SPECIFICATION

Digital processor for processing analog signals

BACKGROUND OF THE INVENTION:

1. Field of the Invention

The invention relates to the field of digital processors, particularly those structured for processing analog signals.

2. Prior Art

Much of the current commercial LSI technology is directed towards the processing and handling of digital signals. Recently, with the declining cost of integrated circuits and their improved performance, more emphasis is being placed on the digital processing of analog signals. Digital processing of video, speech and other analog signals has become more common.

The efficient processing of these analog signals has required new architecture. In the case of the present invention, a two-port, random access memory and the unique treatment of overflows provide an extremely efficient processor. The closest processors known to applicant are described in *IEEE ICASSP,* April 21—14, 1976, Philadelphia, PA, "A Digital Signal Processing System" by Abraham Peled, pages 636—639, and *IEEE ICASSP,* 1977, "Speed Enhancement of Digital Signal Processing Software Via Microprogramming a General Signal Processing Software Via Microprogramming A General Purpose Minicomputer" by Morris and Mudge, Session 5, Paper No. 5.

SUMMARY OF THE INVENTION:

According to the present invention there is provided a digital processor for processing input analog signals comprising: an analog-to-digital converter for converting said input analog signals into an input digital signal; and arithmetic logic means for performing digital arithmetic, having first and second input terminals and an output terminal; a random-access memory having input lines and first and second output lines, the data stored in the cells of said memory being simultaneously accessable at said first and said second output lines, said first output lines being coupled to said first input terminal of said logic means, said output terminal of said logic means being coupled to said input lines of said memory, said input lines of said memory being coupled to said converter to receive said input digital signal; a scaler means for shifting a digital signal, said scaler means being coupled to said second output lines of said memory and to said second input terminal of said logic means; and a program storage means storing a digital control program, said storage means being coupled to supply program instructionss to said logic means, addresses to said memory, and a bit pattern to said scaler means to control said shifting.

BRIEF DESCRIPTION OF THE DRAWINGS:

FIGURE 1 is a block diagram of the presently preferred embodiment of the invented processor.

FIGURE 2 is a partial circuit diagram of the random-access memory employed in the processor of FIGURE 1.

FIGURE 3 is a block diagram of a well-known recursive filter. This diagram is used to describe the manner in which such a filter is implemented by the processor of FIGURE 1.

FIGURE 4 illustrates the plurality of filtering operations which are performed by the process or of FIGURE 1 with a single program.

DETAILED DESCRIPTION OF THE INVENTION:

A digital processor particularly suited for processing analog signals is described. The processor duplicates filters, waveform generators and non-linear functions, such as rectification. In the following description, numerous specific details, such as specific word lengths, cycle times, etc., are set forth to provide a thorough understanding of the invented processor. However, it will be obvious to one skilled in the art that the invention may be practiced without the use of these specific details. In other instances, well-known circuits are shown in block diagram form in order not to obscure the present invention in unnecessary detail.

In the presently preferred embodiment, the entire processor, shown in block diagram form in FIGURE 1, is fabricated on a single silicon substrate employing metal-oxide-semiconductor (MOS) technology. The circuit is realized with n-channel, field-effect transistors employing polycrystalline silicon gates. The programmable read-only memory (PROM) which stores the processor's program uses floating gate memory devices, specifically cells with double polycrystalline silicon gates. The cells are erased by exposure to ulatraviolet radiation. It will be obvious to one skilled in the art that other integrated circuit technologies may be employed for the fabrication of the processor.

As presently realized, the processor employs an external capacitor 26 for use with the sample-and-hld means 24. While an externally generated clocking signal may be employed, the processor also includes clock generation circuitry which only requires the placement of an external crystal between designated pins. Other inputs to the processor are the +5 volt power supply potential and a +25 volt potential used for the PROM programming.

The master clock is divided into multiple phase-shifted clocks for the internal timing of the processor in a well-known manner. The program counter 33 is incremented one instruction count for every 4 master clock cycles and continues to increment until it reaches a count of 191, unless reset by an "End of Program" (EOP) instruction or reset signal. In its presently preferred embodiment, an instruction cycle takes 400 nsec., and thus the entire program of 192 words stored within the PROM 12 is executed at the rate of 13,020 Hz. The various lines associated with the timing are not shown in FIGURE 1 in order not to complicate the figure.

As presently implemented, the processor includes (on the chip) an analog-to-digital and digital-to-analog converter. Input analog signals are coupled on input lines 18 to a multiplexer 21. This multiplexer, under commands from the PROM 12, selects predetermined signals for coupling to the sample-and-hold means 24. The signals sampled by the sample-and-hold means 24 are coupled to a comparator 29 and compared with signals provided at the output of the digital-to-analog converter 31 in an ordinary manner to provide an analog-to-digital conversion. The digital-to-analog converter employs words stored within the input/output register 16b for the comparison function during the analog-to-digital conversion. This conversion and the digital-to-analog conversion for the output is controlled by the PROM 12 and occurs while the processor is performing other functions. The output signals are coupled through the multiplexer 22 to selected output lines 19.

The specific digital-to-analog converter employed in the processor is described in detail in copending application Serial No. 714,422, filed August 24, 1976 entitled "Digital-To-Analog Converter Array" and assigned to the assignee of the present invention. An MOS comparator circuit which may be employed for the comparator 29 of FIGURE 1 is described in copending application Serial No. 828,254, filed August 26, 1977, entitled "MOS Analog-To-Digital Converter" and also assigned to the assignee of the present invention. In the presently preferred embodiment, the reference potential employed by the comparator 29 is externally generated.

The digital portion of the processor generally comprises a random-access memory 16, a scaler 14, an arithmetic logic unit (ALU) 10 and a memory storage means (PROM 12) with its program counter 33. In its presently preferred embodiment, this processor employs twos complement arithmetic.

In the presently preferred embodiment, the PROM 12 stores 192, 24-bit instructions. Each instruction consists of 6 fields. The first field of 3 bits is coupled by lines 43 to the ALU 10. In the ALU, these three bits are decoded and used to select the various operations performed by the ALU as will be described. A field of 4 bits is coupled by lines 44 to the scaler 14. The control signals on these lines are decoded by the scaler decoder 14a and used to control the scaler as will be described in greater detail. Two address fields, each of 6 bits, are transmitted via lines 45 and 46 to the decoding sections of the RAM 16. The signals on lines 46 are referred to as the "A-address" and are coupled to the A-decoder 51; the signals on the lines 45 are referred to as the "B-address" and are coupled to the B-decoder 52. As illustrated in FIGURE 1, 4 lines of lines 46, shown as lines 47, are coupled to a constant register 16a which is part of the RAM 16. The last two fields (total of 6 bits) of each instruction word are coupled via lines 48 to the input/output means which primarily consists of the multiplexers 21 and 22, the sample-and-hold means 24, and of the analog-to-digital and digital-to-analog converters. The signals on these lines control the input/output functions in a well-known manner.

During the operation of the processor, the program counter 33 sequentially accesses each of the instructions. No program jumps are employed. As mentioned, the program returns to the first word upon completion of the 192nd word or for an EOP instruction which is contained in the input/output instruction field. The EOP instruction allows the program to be terminated at its useful end. As employed by the processor, the PROM 12 may be considered as a clock-controlled cycle generator which has a direct effect on the sample rate of the analog signals. If, by way of example, a given input is sampled once per program pass, the sample rate is 13,020 Hz. This provides an effective bandwidth of greater than 5 kHz, assuming an anti-aliasing filter is employed or no higher frequency components are present. The presently preferred embodiment for such a filter is disclosed in copending application Serial No. 868,976, filed January 12, 1978 entitled "L—C Ladder Simulator With Transmission Zeros" assigned to the assignee of this application. The bandwidth of the processor is thus sufficiently broad for use in telephone line communications, and an application in that field will be discussed in conjunction with FIGURE 4. Greater bandwidth may be obtained by sampling the input signal more frequently. This is done by using a shorter program with an EOP instruction or by repeating the program in the PROM.

As presently implemented, the PROM fetch/execute cycle is pipelined. That is, the next instruction is fetched while the previously fetched instruction is being executed. This substantially eliminates the PROM access time as a consideration in the processor speed.

The ALU 10 performs a number of well-known operations on the operands delivered via lines 35 and 37. The output data of the ALU 10 (lines 39) are written into the RAM 16. The operations performed by the ALU 10 are summarized in Table I below.

TABLE I

Code ALU 2	1	Ø	MNEM	OPERATION		DESCRIPTION/COMMENTS	
Ø	Ø	Ø	XOR	A (+) B → R			
Ø	Ø	1	AND	A . B → R			
Ø	1	Ø	LIM	A_S → ±F.S. → R		Sign of A saturates output	
Ø	1	1	ABS	A + Ø → R		Absolute Value	
1	0	0	ABA	A + B → R		Absolute Value and Add	
1	0	1	SUB	B − A → R			
1	1	Ø	ADD	B + A → R			
1	1	1	LDA	A + Ø → R			
1	1	Ø	CND(K)	A + B → R	ADD	If DAR(K)=1	Conditional ADD
1	1	Ø	CND(K)	B → R	or NOP	If DAR(K)=Ø	
1	1	1	CND(K)	A → R	LDA	If DAR(K)=1	Conditional LDA
1	1	1	CND(K)	B → R	or NOP	If DAR(K)=Ø	
1	0	1	CND(K)	B − A → R; Cy → DAR(K)	SUB	If prev. Cy=1	Conditional SUB
1	0	1	CND(K)	B + A → R; Cy → DAR(K)	or ADD	If prev. Cy=Ø	

Most of the operations set forth above are self-explanatory. The absolute value (ABS) and absolute add (ABA) convert the operand on lines 37 to an absolute value before performing any calculations. Load (LDA) and ABS are treated as arithmetic operations in the ALU. That is, the operand on lines 37 is added to zero which replaces the B-operand. This is necessary to avoid the detection of a left shift overflow condition which will be described below. Limit (LIM) sets the result to + or − the maximum value which may be stored in the 35-bit word lengths of the RAM 16 based on the sign of the A-operand.

The output of the ALU 10 is always a 25-bit result which is written into the RAM location determined by the B-address. This writing occurs towards the end of each of the instruction cycles. The two operands entering the ALU as 28-bit words. This 28-bit word length is used to accommodate the left shifting or scaling performed by the scaler 14. In the case of the operand on lines 35, the sign bit is copied onto the three remaining lines. Thus, if the sign bit is a binary one, the three lines which do not receive stored signals from the RAM couple binary ones to the ALU.

The scaler 14 is able to scale any A-operand to a magnitude between 2^2 and 2^{-13}. Therefore, with a 25 word input to the scaler, an overflow from a 28-bit result can never occur since the maximum shift to the left of the 25-bit word input is two places. However, the result may be larger than 25 bits, and thus may not be stored as a 25-bit word in the RAM 16. The overflow tester 10a determines when the result is too large to be stored as a 25-bit word. In this case, the result is corrected to the most positive or most negative twos complement number attainable in 25 bits. This number is then stored in the RAM 16. This "saturation" algorithm protects the continuity of the digitized analog signals and provides stability to the implemented signal processing functions. Note that typically in the prior art when an overflow occurs, a somewhat complex, time consuming procedure is required to handle the overflow. With the implemented algorithm, no time is lost.

The scaler 14 is a digital shifting means which in the presently preferred embodiment, shifts the 25-bit word from the A-ports of the RAM 16 to any magnitude between 2^2 and 2^{-13} (left 2, right 13), as mentioned. The operation of the scaler 14 is summarized in Table II.

TABLE II

SHF 3	SHF 2	SHF 1	SHF ∅	MNEM	OPERATION	DESCRIPTION/COMMENTS
1	1	0	0	R13	$A \times 2^{-13}$	Right Shift 13
⟆	⟆	⟆	⟆			
0	0	0	0	R01	$A \times 2^{-1}$	Right Shift 1
1	1	1	0	L01	$A \times 2^{1}$	Left Shift 1
1	1	0	1	L02	$A \times 2^{2}$	Left Shift 2
1	1	1	1	R00	$A \times 2^{0}$	No Shift

The RAM 16, in the presently preferred embodiment, is a static memory organized as 40 words of 25 bits each. The A-address selects stored data for the A-ports (lines 25) and similarly, the B-address selects stored data for the B-ports (lines 28).

RAM 16 includes a constant register 16a which receives a 4-bit data word from the A-address field. The number in this register is accessed through the A-ports. The four data bits on lines 47 are placed into the four most significant bit positions of the register 16a with the remaining positions set to binary zeros. This is equivalent to a number between

$$\frac{8}{8} \text{ to } +\frac{7}{8}$$

before scaling.

The input/output register 16b is also part of the memory 16 and performs the interfacing function with the analog-to-digital and digital-to-analog converters. Each bit position of this register is selectable for processing through the ALU 10 for conditional arithmetic operations. This 9-bit wide register is accessible in a plurality of ways. The 9 most significant bits of the 25-bit words stored within the memory can be placed in this register via lines 41 from either the A-ports or the B-ports of RAM 16.

Referring now to FIGURE 2, a typical memory cell 55 employed within the RAM 16 of FIGURE 1 is shown. This memory cell comprises a well-known bistable circuit. One output from this cell is coupled through transistor 58 to a line 61. The other output is coupled through a transistor 59 to a line 60. The gate of transistor 57 is coupled to the A-address decoder 51. Similarly, the gate of transistor 58 is coupled to the B-address decoder 52. When an address is received by the memory, the decoders 51 and 52 decode this address in ordinary manner and select the appropriate stored data which is then coupled to the A-ports and the B-ports. For example, if cell 55 is selected by both the A-address and the B-address, both transistors 57 and 58 are selected and the state of the memory cell 55 is coupled through lines 60 and 61 to the A-ports and B-ports, respectively. The memory includes logic which compensates for the fact that sensing of the bistable circuits is performed from the complementary outputs of these circuits. As mentioned towards the end of each instruction cycle, the B-address is employed to select the cells into which data from the ALU is written. For this purpose, the B-address effectively overrides the A-address; this is shown by the line 63 in FIGURE 2.

It should be noted that the memory cells in the RAM may be simultaneously addressed for the A-ports and B-ports. This permits the furnishing of different operands from a single memory without two separate address cycles. A substantial amount of substrate area is saved by this technique in addition to a gain in speed.

All variables employed by the processor are fixed point values of 25 bits, using twos complement arithmetic. For purposes of explanation, it is convenient to assume that a binary point is present just right of the highest order bit (the sign bit). Then the range of any value of x can be considered as $-1.000...0 \leq 1 < +1.000...0$. With this convention, the overflow tester 10a of FIGURE 1 need only examine the most significant 4 bits of each 28-bit word to determine if an overflow (with respect to the 25-bit word lengths) has occurred. If all 4 most significant bits are the same, then the word may be shifted into the RAM 16 by ignoring the 3 most significant bits. On the other hand, if the 4 most significant bits are not the same, then the result is too large to fit within the allocated 25 bits. In this case, the result is replaced with a value which has the sign of the correct result (which is equivalent to the highest order bit of the 28-bit word), but has the largest magnitude, as mentioned.

Before describing a specific use for the processor of FIGURE 1, a brief description of the manner in which the scaler and ALU cooperate is helpful. Assume it is necessary to multiply x by the coefficient 1.11100011. In a typical algorithm, the result is computed by employing six addition steps

corresponding to each of the binary ones in the coefficient. However, the result y may be represented by:

$$y = (x \cdot 2^{1}) - (x \cdot 2^{-3}) + (x \cdot 2^{-6}) - (x \cdot 2^{-8})$$

which requires four addition/subtraction operations. If x is communicated to the scaler 14 from the memory, the shifts of x by the magnitudes 2^{1}, 2^{-3}, 2^{-6}, and 2^{-8} are easily performed by the scaler. After each shift, the appropriate addition or subtraction is performed by the ALU 10 to provide the result. In some cases, a combination of the above algorithm along with the more standard technique provides the most efficient way of reaching the result. For example, is a coefficient is 1.111010101, eight additions or subtractions would be required for the above-described method. However, the same result can be reached with five steps as shown by the following equation:

$$y = (x \cdot 2^{1}) - (x \cdot 2^{-3}) + (x \cdot 2^{-5}) + (x \cdot 2^{-7}) + (x \cdot 2^{-9})$$

Assume now that the processor of FIGURE 1 is to be employed as a filter and more specifically, a filter with a complex conjugate pair of poles at

$$s = \sigma \pm j\omega_0$$

It is well-known that the sample data equivalent of a "continuous" filter may be represented by one or more recursive filter sections such as the one shown in FIGURE 3. For each sample, y_1 propagates to y_2, and y_0 to y_1. The new value for y_0 is represented by the equation:

$$y_0 = gx + B_1 y_1 + B_2 y_2$$

Thus for each sample, the input shown on line 65 is multiplied by the value g as shown by the multiplier 67 and is then communicated to the summer 72. The multiplier 68 multiplies y_1 by B_1 and communicates the result to the summer 72. Similarly, the multiplier 69 multiplies y_2 by the coefficient B_2 and the result is communicated to the summer 72. The blocks 70 and 71 represent the unit delays, more specifically, the rate at which the analog input signal is sampled by the sample-and-hold means 24 of FIGURE 1.

For the above example,

$$B_1 = 2C^{-\sigma T}\cos w_0 T$$

and

$$B_2 = -C^{-\sigma T}$$

where

$$\sigma T = \frac{w_0 T}{2Q}$$

Assume further that the filter is to have resonance at 1,000 Hz ± 0.5% with a Q between 75 and 100 and a midband gain of 1 ± 10%, for a sample rate of 76.8 microseconds.

Then $\omega_0 t = 0.48255 \pm .0024$ and $.002412 \leq \sigma T \leq .003217$

B_2 in binary form is given by:

$$.1111110010111011 \leq -B_2 \leq 1111111011000100$$

A value which falls within this range and which only requires three terms to express is:

$-B_2 = .111111101$, that is, $-B_2 \mp 0.9941406$

B_1 must fall within the range of:

$$1.764205 \leq B_1 \leq 1.768646$$

In binary form, this may be written as:

$$1.1100010011001 \geq B_1 \geq 1.110000111010$$

The value $B_1 = 1.110001$ meets this requirement. With these binary values for B_1 and B_2,

$$f_0 = 1001.8 \text{ Hz and } Q = 82$$

The value for the coefficient g may be found by evaluating the midband gain of the filter. This relatively complex computation assumes that $y_0 = \sin \omega t$. It can be shown that

$$.002452 \leq g \leq .00299$$

A binary value for g which falls within this range is:

$$g = .0000000011$$

To implement the above recursive filter in the processor of FIGURE 1, consideration must be given first to the fact that both $B_1 y_1$ and $B_2 y_2$ may contribute to an immediate overflow. To handle this potential overflow, a fraction of y_0, for example

$$\frac{y_0}{4}$$

is used. Then

$$\frac{y_0}{4}$$

is shifted left when propagating y_0 to y_1 during the next pass. If overflow occurs, it will be when y_1 is propagated, rather than for some intermediate value. Table III below shows the assembly code for implementing this recursive filter in the processor of FIGURE 1. The variable assigned for y_0 is shown as y_{00} to note that is represents only one-fourth of the desired value.

TABLE III

ALU Operation	Destination	Source	Shift by Scaler 14
LDA	y_2	y_1	0
LDA	y_1	y_{00}	2
LDA	y_{00}	y_1	–1
SUB	y_{00}	y_1	–4
ADD	y_{00}	y_1	–8
SUB	y_{00}	y_2	–2
ADD	y_{00}	y_2	–9
SUB	y_{00}	y_2	–11
ADD	y_{00}	x	–10
SUB	y_{00}	x	–12

In understanding the above table, the additional shift of –2 must be considered, note

$$\frac{y_0}{4}$$

is generated.

The filter thus takes 10 words of PROM to implement. The program uses 4 words of the RAM 16 for the variables y_{00}, y_1, y_2, and x. However, it is possible that x and y_2 be stored in temporary memory locations.

In the above example, the blocks 70 and 71 of FIGURE 3 represent the delay associated with the execution of the 192 instructions contained within the PROM. Where a faster sample rate (broader bandwidth) is required, the instructions implementing the filter, or the like, may be repeated within the PROM or an "EOP" instruction may be employed causing the instructions to be reexecuted more frequently, as mentioned.

In a similar manner to that used to implement the filter having complex conjugate pole pairs, complex conjugate zero pairs may be implemented. Moreover, in the presently preferred embodiment, four basic sources of non-linear operations are available, specifically, ABS, LIM, conditional arithmetic and the overflow algorithm. The use of the ABS and the overflow operations produce continuous functions while the LIM and conditional arithmetic provide discontinuous functions. The ABS operation is equivalent to an ideal full-wave rectifier. Rectification and filtering may be combined using the ABS operation for the filter input. Oscillators may be similarly implemented by the processor. One method utilizes a simple relaxation technique to implement a sawtooth waveform generator. The sawtooth waveform may then be altered using "piecewise" linear transforms to simulate a sinusoidal function. Another method consists of implementing an unstable second order filter with poles on the $j\omega$ axis.

A typical application for the processor of FIGURE 1 is shown graphically in FIGURE 4. In this application, a telephone line signal is examined to detect the 8 separate signal tones commonly used in a touch-tone dialing system. The high frequency components of the signal are first removed to prevent aliasing. The signal is then applied to the analog-to-digital converter 74 of FIGURE 4. This function is performed by the analog section of the processor of FIGURE 1. At the beginning of the 192 words of the instruction set, the PROM instructions implement a 6-pole, low bandpass filter and a 6-pole, high bandpass filter. Quadrature limiting is also implemented as shown by the blocks 76 and 77. Then the program implements 6 single tone filters ranging in frequency from 697 Hz to 1633 Hz. The program also implements rectification, averaging and level detection. The output is then converted to analog form as shown by digital-to-analog converter 80 and multiplexed on the 8 output lines 19. All the operations shown in FIGURE 4, including the filtering, rectification, averaging, comparison and conversion may be performed within the confines of the 192 word program.

Thus, a digital processor has been described which is particularly adaptable for implementing analog functions, such as filtering, rectification, etc. The processor employs a programmable memory and thus may be programmed and reprogrammed.

CLAIMS

1. A digital processor for processing input analog signals comprising: an analog-to-digital converter for converting said input analog signals into an input digital signal; an arithmetic logic means for performing digital arithmetic, having first and second input terminals and an output terminal; a random-access memory having input lines and first and second output lines, the data stored in the cells of said memory being simultaneously accessable at said first and said second output lines, said first output lines being coupled to said first input terminal of said logic means, said output terminal of said logic means being coupled to said input lines of said memory, said input lines of said memory being coupled to said converter to receive said input digital signal; a scaler means for shifting a digital signal, said scaler means being coupled to said second output lines of said memory and to said second input terminal of said logic means; and a program storage means storing a digital control program, said storage means being coupled to supply program instructions to said logic means, addresses to said memory, and a bit pattern to said scaler means to control said shifting.

2. The processor defined by Claim 1 wherein said storage means communicates a first and a second address field to said memory for selecting stored data for coupling to said first and second output lines, respectively.

3. The processor defined by Claim 2 wherein said first address field determines the location in said memory into which data is written.

4. The processor defined by Claim 3 wherein said memory includes a register for storing a constant, said constant being coupled to said register from said second address field.

5. The processor defined by Claim 1 wherein said memory stores words of n-bits and wherein said logic means performs said digital arithmetic on words of greater than n-bits.

6. The processor defined by Claim 5 including means for detecting words of effective width greater than n-bits which result from said digital arithmetic performed by said logic means and for substituting a predetermined word of n-bits for said words of effective width greater than n-bits.

7. The processor defined by Claim 6 wherein said scaler means shifts words in both directions.

8. A digital processor substantially as hereinbefore described, with reference to and as illustrated in the accompanying drawings.

Printed for Her Majesty's Stationery Office by the Courier Press, Leamington Spa, 1983. Published by the Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

In the above example, the blocks 70 and 71 of FIGURE 3 represent the delay associated with the execution of the 192 instructions contained within the PROM. Where a faster sample rate (broader bandwidth) is required, the instructions implementing the filter, or the like, may be repeated within the PROM or an "EOP" instruction may be employed causing the instructions to be reexecuted more frequently, as mentioned.

In a similar manner to that used to implement the filter having complex conjugate pole pairs, complex conjugate zero pairs may be implemented. Moreover, in the presently preferred embodiment, four basic sources of non-linear operations are available, specifically, ABS, LIM, conditional arithmetic and the overflow algorithm. The use of the ABS and the overflow operations produce continuous functions while the LIM and conditional arithmetic provide discontinuous functions. The ABS operation is equivalent to an ideal full-wave rectifier. Rectification and filtering may be combined using the ABS operation for the filter input. Oscillators may be similarly implemented by the processor. One method utilizes a simple relaxation technique to implement a sawtooth waveform generator. The sawtooth waveform may then be altered using "piecewise" linear transforms to simulate a sinusoidal function. Another method consists of implementing an unstable second order filter with poles on the $j\omega$ axis.

A typical application for the processor of FIGURE 1 is shown graphically in FIGURE 4. In this application, a telephone line signal is examined to detect the 8 separate signal tones commonly used in a touch-tone dialing system. The high frequency components of the signal are first removed to prevent aliasing. The signal is then applied to the analog-to-digital converter 74 of FIGURE 4. This function is performed by the analog section of the processor of FIGURE 1. At the beginning of the 192 words of the instruction set, the PROM instructions implement a 6-pole, low bandpass filter and a 6-pole, high bandpass filter. Quadrature limiting is also implemented as shown by the blocks 76 and 77. Then the program implements 6 single tone filters ranging in frequency from 697 Hz to 1633 Hz. The program also implements rectification, averaging and level detection. The output is then converted to analog form as shown by digital-to-analog converter 80 and multiplexed on the 8 output lines 19. All the operations shown in FIGURE 4, including the filtering, rectification, averaging, comparison and conversion may be performed within the confines of the 192 word program.

Thus, a digital processor has been described which is particularly adaptable for implementing analog functions, such as filtering, rectification, etc. The processor employs a programmable memory and thus may be programmed and reprogrammed.

CLAIMS

1. A digital processor for processing input analog signals comprising: an analog-to-digital converter for converting said input analog signals into an input digital signal; an arithmetic logic means for performing digital arithmetic, having first and second input terminals and an output terminal; a random-access memory having input lines and first and second output lines, the data stored in the cells of said memory being simultaneously accessable at said first and said second output lines, said first output lines being coupled to said first input terminal of said logic means, said output terminal of said logic means being coupled to said input lines of said memory, said input lines of said memory being coupled to said converter to receive said input digital signal; a scaler means for shifting a digital signal, said scaler means being coupled to said second output lines of said memory and to said second input terminal of said logic means; and a program storage means storing a digital control program, said storage means being coupled to supply program instructions to said logic means, addresses to said memory, and a bit pattern to said scaler means to control said shifting.

2. The processor defined by Claim 1 wherein said storage means communicates a first and a second address field to said memory for selecting stored data for coupling to said first and second output lines, respectively.

3. The processor defined by Claim 2 wherein said first address field determines the location in said memory into which data is written.

4. The processor defined by Claim 3 wherein said memory includes a register for storing a constant, said constant being coupled to said register from said second address field.

5. The processor defined by Claim 1 wherein said memory stores words of n-bits and wherein said logic means performs said digital arithmetic on words of greater than n-bits.

6. The processor defined by Claim 5 including means for detecting words of effective width greater than n-bits which result from said digital arithmetic performed by said logic means and for substituting a predetermined word of n-bits for said words of effective width greater than n-bits.

7. The processor defined by Claim 6 wherein said scaler means shifts words in both directions.

8. A digital processor substantially as hereinbefore described, with reference to and as illustrated in the accompanying drawings.

Printed for Her Majesty's Stationery Office by the Courier Press, Leamington Spa, 1983. Published by the Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.