Distribuir no curso (Engenharia de Instrumentação)

A new LSI chip called an analog signal processor allows analog-circuit design to be performed quickly and efficiently with μC development systems and high-level software.

FÁBIO MONTORO

Signal processor 'digitizes' analog design problems

A specialized microprocessor brings programmability, flexibility, and efficiency to analog system design. Now analog systems can be designed and developed quickly and easily via microcomputer development systems and high-level applications software and high-level compilers. With the "analog" signal processor using digital sampled-data techniques to implement analog functions, analog systems become much smaller and are able to maintain greater stability with time and temperature variations.

The heart of the digital sampled-data system, the digital processor, eliminates the need for conventional analog components (op amps, transistors, precision resistors, capacitors, and diodes) by implementing in software the hardware devices in which they are normally employed: filters, limiters, oscillators, modulators, and demodulators. In all, there are three major advantages to going digital:

- *Flexibility.* Modifications and design improvements require simple program changes. Hard-wired functions are implemented with flexible software, which means one analog signal processor can be used in thousands of applications. Furthermore, a programmable device eliminates the risk associated with custom ICs; in addition, programmability permits product improvements without redesigning printed-circuit boards.
- *Reproducibility.* Particular analog-signal-processor implementations are repeatable—for example, a breadboard waveform generator designed to oscillate at a particular frequency will work exactly the same in production. Performance specifications are no longer subject to component tolerances. Parts matching and circuit tuning are eliminated, instances of variable circuit performance are reduced, and circuit degradation over time due to interaction or noise is minimized.
- *Ease of design.* Suddenly, a host of development tools that increase the efficiency and reduce the development time for digital designs is available to solve analog design problems. Tools include high-level applications compilers, software simulators, and assemblers. For example, analog filter performance can be verified without building prototype hardware. In fact, nothing need be built until the system has been simulated and found to be completely functional.

Homer Gee, Product Manager, Telecomm Components
Intel Corp.
265 Walsh Ave., Santa Clara, CA 95051
Martin Tanenhaus, Manager, Advanced Prod. for Ordinance
Roland Fritts, Electrical Engineer,
Advanced Software Systems
Magnavox Government and Industrial Electronics Co.
1313 Production Rd., Ft. Wayne, IN 46808

LSI opens the door

Digital sampled-data systems themselves haven't always been so easy to design, though the concept is simple enough. There are seven major elements (Fig. 1). An incoming signal enters an anti-aliasing (low-pass) filter and is sampled by a sample-and-hold circuit. Each acquired sample is then converted to a digital signal and sent to the digital processor, which executes any necessary programmed functions. Next, the processed samples are converted back to an approximation of an analog signal. A smoothing filter serves to recover a continuous signal.

Until recently, however, high implementation cost, considerable system complexity, large size, and substantial power consumption have limited hardware applications severely. Traditionally, digital sampled-data systems have been reserved for applications stressing high stability and accuracy over cost, complexity, size, and power consumption.

Recent advances in large-scale integration (LSI) and digital signal-processing architecture make it possible to implement digital sampled-data systems with just a few components. Cost, complexity, and

power consumption are reduced several orders of magnitude, and systems once requiring many printed-circuit boards are easily constructed with a few ICs. Many of the elements in Fig. 1 can be placed on a single LSI chip.

While there are analog signal-processor ICs that involve only the digital processor, Intel's 2920 incorporates all elements shown except the anti-aliasing filter and the reconstruction filter. The 2920 is housed in a 28-pin IC package and contains on-chip EPROM program storage, four multiplexed input channels and sample-and-hold circuits, a-d and d-a circuitry, scratchpad memory, a 28-bit digital processor, eight demultiplexed output channels with sample-and-hold circuits, plus all necessary program-counter and clock logic (Fig. 2). Table 1 lists the 2920's instruction set.

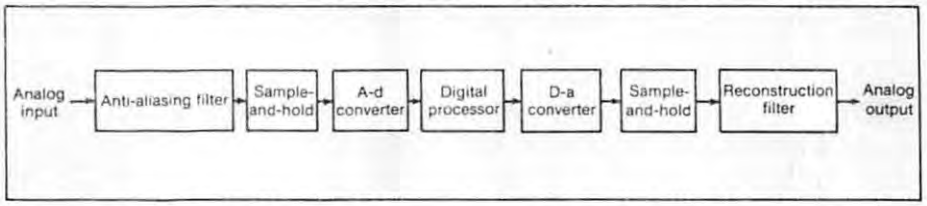
The 2920 provides PLL AM detection

A good example of an analog subsystem built with a 2920 is the phase-locked-loop amplitude-modulation detector. Able to multiplex input and output lines, the 2920 can either be time-shared among several different circuits or provide one circuit function with multiple inputs and outputs.

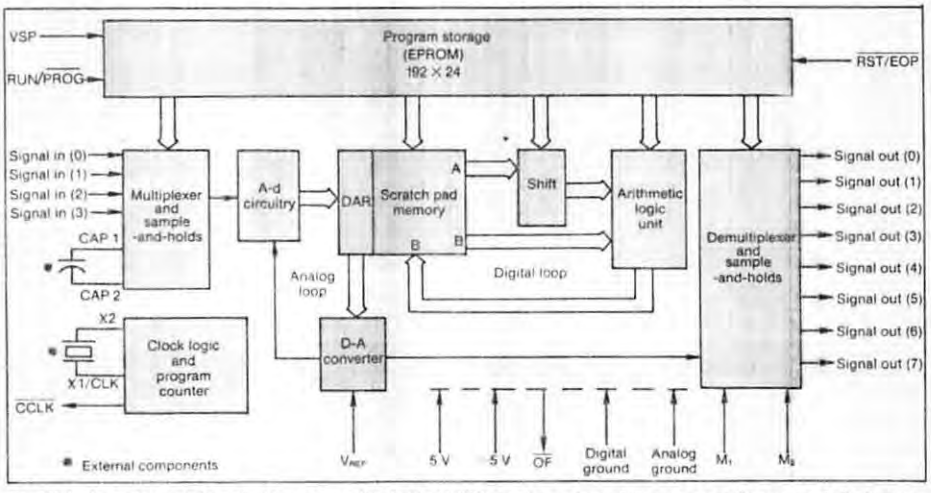
PLL AM detection may be applied to many different electronic processing systems. Simple acquisition systems use the quadrature "lock indicator" technique to indicate whether a signal is present in a noisy environment. When more complex signal processing is involved—say, at the output of an audio receiver—the PLL may be used at a tracking filter. Then the output of the voltage-controlled oscillator (VCO) is processed to determine the frequency content of the audio signal, which may initiate further processing circuitry.

As a PLL AM detector, the 2920 locks to the AM signal's carrier. This produces a VCO output with the same frequency as the AM carrier but no modulation envelope. Multiplying this "in-phase" coherent reference signal (in the phase detector) with the amplitude-modulated input signal produces an output that indicates PLL signal acquisition (Fig. 3, top).

Designing the PLL AM detector involves computer-aided network analysis. First, a macro-



1. A sampled-data system using digital processing contains seven elements: an anti-aliasing filter to band-limit the input signal and reduce sampling distortion, a sample-and-hold that retains the sampled data long enough for processing, an a-d converter that generates a digital word to represent each analog voltage level held, a digital processor to implement the transfer function under software control, a d-a converter that turns digital words back into analog voltages, a second sample-and-hold that removes output frequency distortion by resampling the d-a output using very narrow pulses, and a reconstruction filter that smooths d-a waveforms, producing a continuous analog output.



2. This functional block diagram of Intel's 2920 analog signal processor (run mode) shows that the chip contains all the elements of Fig. 1 except the anti-aliasing filter and the reconstruction filter.

Table 1. The 2920 instruction set

Digital instructions		Operations		
ADD		$(A \times 2^N) + B \rightarrow B$		
SUB		$B - (A \times 2^N) \rightarrow B$		
LDA		$(A \times 2^N) \rightarrow B$		
XOR		$(A \times 2^N) \oplus B \rightarrow B$		
AND		$(A \times 2^N) \cdot B \rightarrow B$		
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. $\rightarrow B$		
ADD	CND( )	$(A \times 2^N) + B \rightarrow B$		IFF DAR(K) = 1
		$B \rightarrow B$		IFF DAR(K) = 0
SUB	CND( )	$B - (A \times 2^N) \rightarrow B$	& CY→DAR(K)	IFF $CY_P = 1$
		$B + (A \times 2^N) \rightarrow B$	& CY→DAR(K)	IFF $CY_P = 0$
LDA	CND( )	$(A \times 2^N) \rightarrow B$		IFF DAR(K) = 1
		$B \rightarrow B$		IFF DAR(K) = 0
ABA	CND( )	$(A \times 2^N) + B \rightarrow B$		
XOR	CND( )	$(A \times 2^N) \oplus B \rightarrow B$		

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	Reset program counter to zero
NOP	No operation
CND(K)	Select bit K for conditional instructions
CNDS	Select sign bit for conditional instructions

model of the circuit is generated and analyzed using the NET-2 computer-analysis code (Fig. 3, bottom).[1] Only after such analysis is the detector implemented on the 2920. Implementation is fully simulated on a microcomputer development system prior to programming the 2920. A hardware checkout matches the results predicted by NET-2 macromodel analysis surprisingly well (Fig. 4).[2]

Using the 2920's a-d conversion power, 8-bit multiply, algorithm processing, and digital output, the PLL AM detector is realized in a single chip. What's more, by using Intel's 2920 simulator to help test 2920 implementations, any parameter of the PLL AM detector can be viewed while it is operating.

Functional blocks simplify design problems

One of the most attractive and powerful features of building the PLL AM detector with a 2920 is the ability to view the design problem in functional blocks and develop the code in function modules. These modules can be tested separately and then integrated sequentially to achieve the overall subsystem. After the code is assembled, debugged, and simulated in modular form, it is then compressed, exploiting the 2920's ability to process analog and digital instructions simultaneously.

The functions to be realized are limiters, mixers, filters, and VCOs (Fig. 4). Each part of the block-model implementation is tested separately, then modified, and finally "glued together" to achieve a working PLL. Sample-rate constraints for the 2920's 192-step program are overcome by using two PLLs cascaded together. This yields twice the original sample rate, so high-frequency information can be processed.

After a-d conversion, the signal amplitude is limited to a specific plus or minus value. For normal input levels, a slightly clipped output results. As the input amplitude increases, the clipped output approximates a square wave.

The limiter checks the input to determine if it is positive or negative. Negative input signals are limited, positive input signals are passed. The absolute limit value is either added or subtracted. These conditional checks on the input signal's sign "bit" make the limit function powerful and flexible.

Digital implementation of an analog mixer simply requires that two signals be combined with an 8-bit multiplication. The 2920 can provide the necessary 8-bit multiplication because it features 8-bit multiply code. Details about this multiply code are found in the Intel *2920 Assembly Language Manual.*

The digital filters (e.g., loop filter) required for the PLL and quadrature-detector portions of the PLL AM detector can be easily achieved with the 2920 because code can be developed from the s-plane representation of the filter. Or, one of Intel's many development tools specifically made for digital filter design can be used.

The longhand method for designing the PLL filters involves a few simple mathematical steps (Fig. 5, top). After determining the s-plane equation of each

filter, the bilinear transform is used to calculate the normalized z-plane equation. This z-plane equation provides coefficient values useful in the digital realization of the filters. When the normalized z-transform has been developed and expressed in binary numbers, a filter can be coded using general digital realization derived from the Intel assembler's realization of poles and zeros (Fig. 5, bottom).

The VCO consists of a sawtooth waveform generator whose frequency is controlled by a variable ramp input (Fig. 6). To vary the ramp, the filtered error voltage from the loop filter is integrated and summed with a constant voltage representing the free-running VCO frequency. The sawtooth's variable-frequency output is converted into sine and cosine functions. The sine function is approximated as a clipped triangle, while the cosine function is a 90°-shifted square wave.

Mating the modules

The function modules are combined using the macromodel approach. (Incidentally, this computer modeling technique was developed to simulate ICs from information in vendor specification sheets. Though this PLL macromodel is fashioned after the 565's device specifications, it is a general model and applies to virtually all currently available devices.)

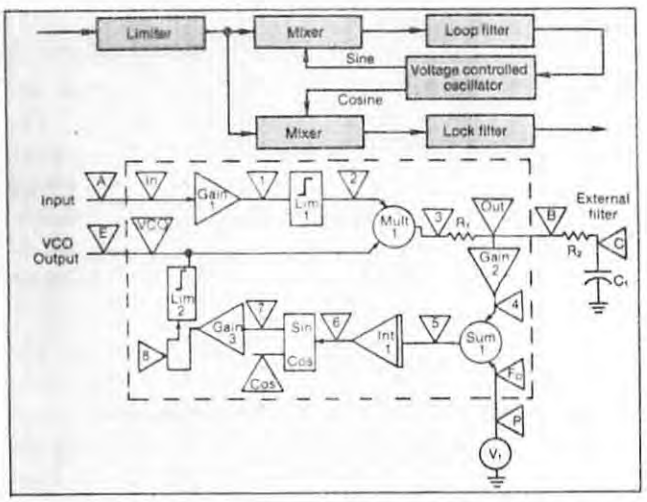
The PLL macromodel description specifies the dc performance (loop gain) of the phase detector (K_D) and VCO (K_0), and defines the external discrete components (resistors and capacitors) representing the loop filter. Transient analysis establishes the PLL's acquisition and track characteristics, which are governed by the loop-filter components between the phase detector and VCO. Key dynamic characteristics include capture range (ω_c), lock range (ω_L), and damping factor (δ).

Each section of the PLL is modeled and evaluated with the Intel simulator. The macromodeling approach makes it easy to check code functionality and to modify the original design so that the modules work together. To realize the PLL with the 2920, a center frequency of 5 kHz with a lock range of approximately ±3 kHz is needed. Two loops, cascaded together, produce a sample rate that allows frequency processing at the high end of the lock range.

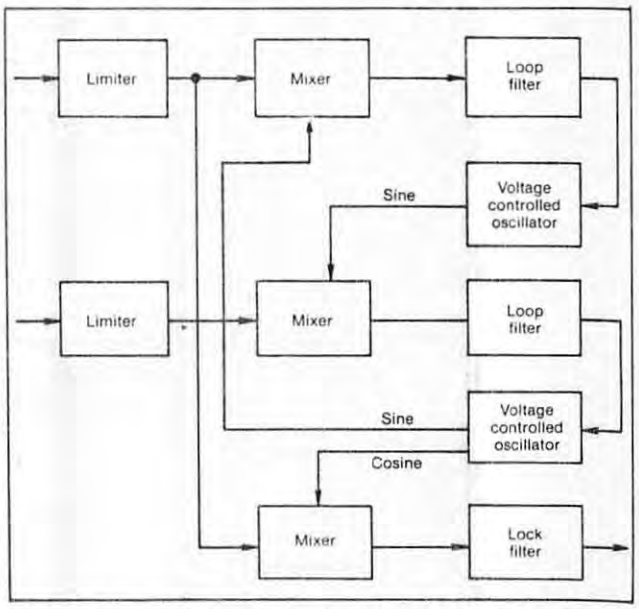
At this point, the designer's software skill becomes an important factor. When first combined and tested as a PLL and quadrature detector, the code representing all the function modules does not operate in the frequency range desired. Modular code alone does not use the 2920's ability to process discrete analog and digital functions simultaneously.

Code compression removes inefficiencies

The desired operating-frequency range is obtained by compressing and selectively doubling the sample rate of all functions. Using a top-down sequential approach, the PLL AM detector requires 140 lines of code to operate from 2 to 8 kHz. But a 2920 chip (13-kHz sampling rate nominal) can only achieve a maximum 17-kHz sampling rate, little more than twice the highest frequency of interest. Therefore,



3. When a PLL AM detector is built around the 2920, lock indication is provided by multiplying an "in-phase" coherent reference signal (VCO cosine output) with the amplitude-modulated input signal (top). Actual design involves computer-aided network analysis that generates a macromodel of the circuit (bottom). Triangle symbols with numbers are test points; triangle symbols with single letters show where external test parameters are applied; and all multiple-letter triangles identify I/O lines.



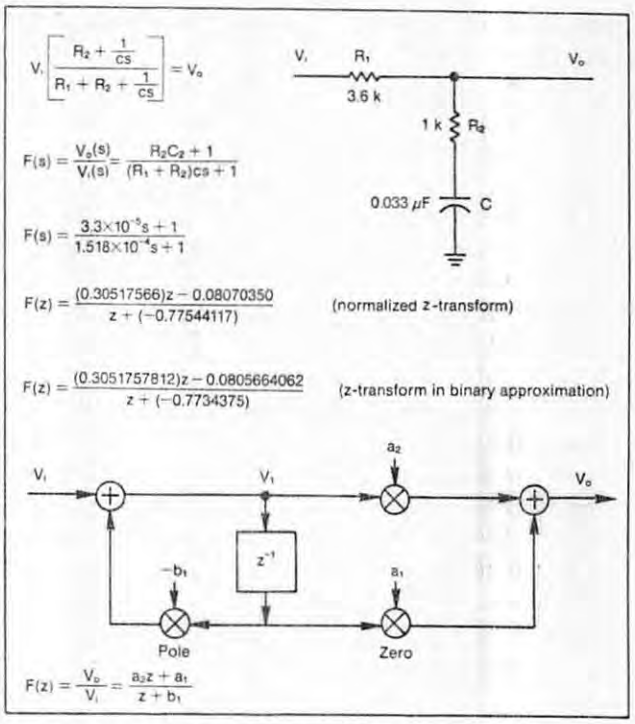
4. Implemented in hardware, the phase-locked loop with a lock indicator matches the results predicted by the macromodel analysis in Fig. 3 surprisingly well. Sample-rate constraints for the 2920's 192-step program are overcome by using two PLLs cascaded together.

the modular code is compressed by paralleling any constant generation with signal input operation, and analog output operations are paralleled with digital calculations (Fig. 7).

Doubling the modules requiring a higher sampling rate produces a system with two rates. The PLL operates at a sampling rate of 26 kHz and the quadrature detector operates at 13 kHz. The operational PLL AM detector, implemented with the 2920, uses 190 lines of code running at almost double the nominal sampling rate.

Since the VCO and the loop-filter designs depend on sample rate, and they appear twice, the sample rate used to design them is twice the sample rate of one program pass. The lock filter is designed for a 13-kHz sample rate, and the loop filters and VCOs are designed for 26 kHz. The loops are cascaded so that the VCO output of the first loop is mixed with the input signal to the second loop.

It follows that the VCO output for the second loop should be mixed with the input for the first loop. The cosine value being mixed with the input of the first loop input is taken from the second VCO. (The lock indicator will operate unchanged if the opposite VCO and input loop lines are used to generate the lock indicator.) Table 2 presents a source-code listing for the PLL AM detector.



5. The longhand design for a digital loop filter involves determining the s-plane equation for the filter and using the bilateral transform to calculate the normalized z-plane equation (top). This provides coefficients used in developing a coded general digital realization of the filter (bottom).

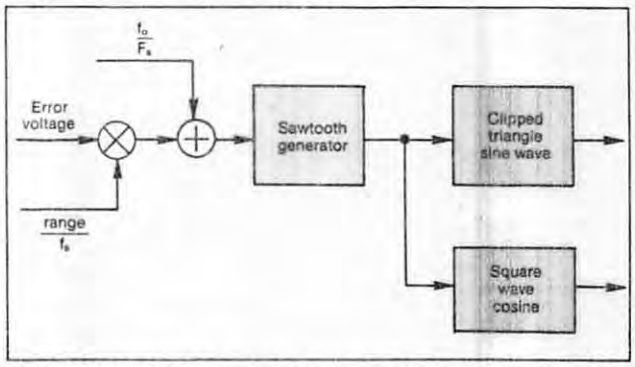
Error, lock, capture, clock

The VCO signal is mixed with the input signal to produce an error voltage, which is added to the constant voltage that drives the VCO sawtooth generator. If the sum voltage and constant voltage are equal, the VCO stays at its free-running frequency. For instance, when there is no input signal, the error voltage coming out of the loop filter will be approximately zero.

Also, if the VCO and input signals differ too greatly in phase (frequency), then the integrated error voltage at the filter output will again be near zero. But when the input frequency enters the capture range of the loop, the error voltage developed causes the VCO frequency to increase or decrease (as necessary) to match the input frequency. Depending on the direction that the input signal is sweeping, the phase-locked loop exhibits both a lock range and a capture range (Fig. 8).

When the input signal and VCO signal are in phase, the input signal and cosine signal are 90° out-of-phase, and the lock filter's output voltage reaches an absolute maximum. By monitoring this lock signal, the quality of lock can be determined. In Fig. 9, a "clean lock" is indicated as the VCO tracks either a sweep up or a sweep down in frequency.

These tests, when run at a 100-kHz/s sweep rate, sweeping from 9 kHz to 1 kHz (and back), give a



6. The PLL AM detector's VCO contains a sawtooth generator that is frequency-controlled by a variable-ramp input voltage. The ramp is varied by error voltage from the loop filter, which is integrated and summed with a constant voltage derived from the VCO's free-running frequency.

continuous sweep pattern. The sweep rate is then increased to find the maximum rate the 2920 PLL can handle. At 1.0 MHz/s, the PLL still tracks the signal. Its response will not break down until the sweep rate approaches 1.4 MHz/s.

Once the PLL is working properly with a 10-MHz clock, some experiments with various clock rates can be conducted and the effect on lock range noted. A test of the loop at different clock rates discloses that the lock range and VCO free-running frequency will change proportionally and predictably as the clock rate changes. With a 5-MHz clock, the free-running

frequency and the lock range are about half of that observed at 10 MHz.

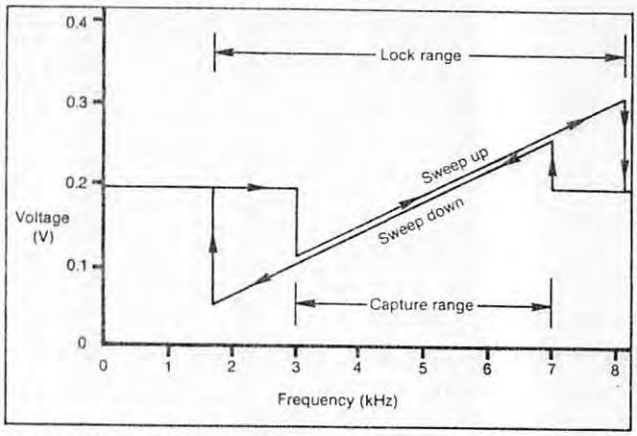
The loop also tends to lock on multiples of the free-running VCO frequency. A bandpass or low-pass filter is needed on the input line to prevent this.

Digital implementation allows the loop filter's characteristics to be varied easily.[3] For example, a two-mode, self-adjusting, acquisition and tracking PLL may be created by modifying the original loop filter's coefficients to adjust loop gain, loop damping, and natural frequency. New, conditional quality-of-lock tests will also be needed.

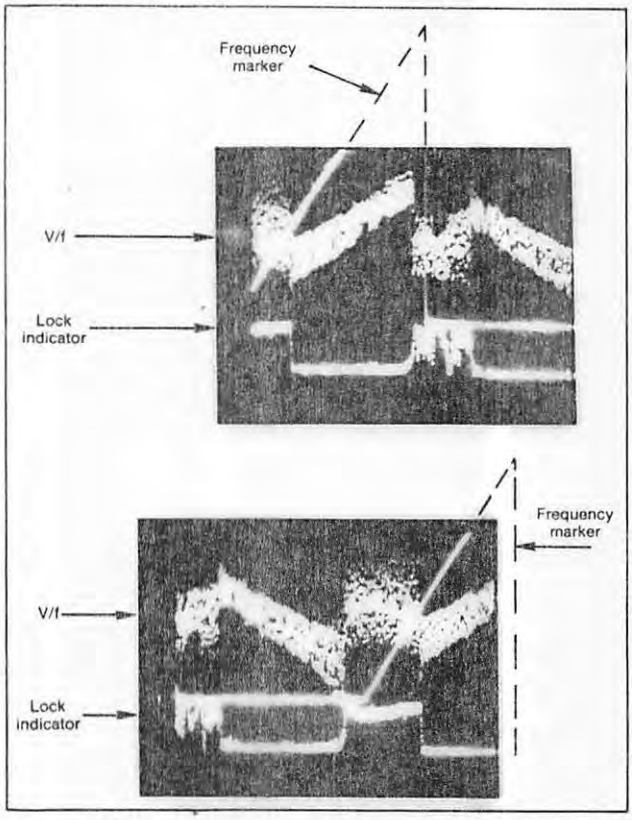
No discussion of the PLL macromodel is complete without describing its dc characteristics[4] and diagramming their relationship (Fig. 10). The phase-detector gain factor (K_D) is defined as the conversion factor between the phase-detector output voltage, $V_e(t)$, and the phase difference between input and VCO signals. The value of K_D is represented by the combined limiter and mixer functions in the 2920; K_{Dmax} is simulated by the limiter.

Program Step	Digital Function	Analog Function
0		
5	Loop filter	Input signal
10		Sign bit converter
15	Lock filter	8-bit A/D
20		
25	Load constants	
30		
35		
40	Limiter	Limiter
45	8-bit multiply	8-bit multiply
50		
55		
60	Loop filter	Output sine
65		
70	Generate sawtooth	Generate sawtooth
75	Generate sine	DAR settle time
80	Generate cosine	Output voltage proportional to frequency
85		
90	8-bit multiply	8-bit multiply
95		
100	Loop filter	Input signal
105		Sign bit converter
110	Lock filter	8-bit A/D
115	Load constants	
120		
125		
130	Limiter	Limiter
135		
140	8-bit multiply	8-bit multiply
145		
150		
155	Loop filter	Output sine
160		DAR settle time
165		Output cosine
170		
175	Generate sawtooth	Generate sawtooth
180	Generate sine	DAR settle time
185	Generate cosine	Output voltage proportional to frequency
190		
192		

7. Code compression allows analog operations to be performed in parallel with digital operations inside the 2920. Blank spaces are purposely inserted in the digital software-functions column so that the simultaneous operations can exchange information efficiently while the program runs.



8. The PLL exhibits a lock range considerably wider than its capture range. This means once an input signal has been captured, the PLL remains locked to it even if it drifts out of the frequency range in which it was initially acquired.



9. Bench tests of lock-signal quality show that a "clean lock" is obtained as the VCO tracks a 1-kHz to 9-kHz sweep "up" (top) or a 9-kHz to 1-kHz sweep "down" (bottom). In both cases, the sweep rate is 100 kHz/s.

Table 2. PLL AM detector source-code listing

```
ISIS-II 2920 ASSEMBLER V1.0

ASSEMBLER INVOKED BY:  AS2920  F1 ILOOP SRC DEBUG
 LINE  LOC OBJECT SOURCE STATEMENT

    1    0 00C6EF LDA DAR,KP0,IN0          ;
    2    1 0008EF LDA REG,V2,IN0           ;
    3    2 0000FF LDA V2,REG,IN0           ; LOOP FILTER FEEDBACK
    4    3 00003A SUB V2,REG,R02,IN0       ;
    5    4 00009C ADD V2,REG,R05,IN0       ;
    6    5 0000FA SUB V2,REG,R08,IN0       ;
    7    6 44B2EF LDA V0,KP0,R00           ;
    8    7 64006A SUB V0,REG,R04,CVT5      ;
    9    8 4400AA SUB V0,REG,R06           ; LOOP FILTER
   10    9 EBE6ED ADD DAR,KM2,R00,CND6     ;  FEED FORWARD
   11   10 44000B SUB V0,REG,R09           ;
   12   11 75004B SUB V0,REG,R11,CVT7      ;
   13   12 4C00FF LDA DEL,VL,R00          ;
   14   13 4218EF LDA VL,DEL,R00          ;
   15   14 6318BA SUB VL,DEL,R05,CVT6     ; LOCK FILTER FEEDBACK
   16   15 4218AA SUB VL,DEL,R06          ;
   17   16 42180D ADD VL,DEL,R09          ;
   18   17 5318BE LDA VG,DEL,R06,CVT5     ;
   19   18 4218DC ADD VG,DEL,R07          ; LOCK FILTER
   20   19 4218DB SUB VG,DEL,R10          ;  FEED FORWARD
   21   20 459AEF LDA VV,KP1,R00,CVT4     ;
   22   21 449A0C ADD VV,KP1,R01          ; LOAD VOLTAGE PROP
   23   22 449ABC ADD VV,KP1,R05          ;  TO FREE RUN FREQ.
   24   23 3592FF LDA P,KP0,R00,CVT3      ;
   25   24 40C2EF LDA VN,KP0,R00          ;  CLEAR MULTIPLIERS
   26   25 C160EF LDA VN,VN,R00,CND4      ;
   27   26 2100EF CVT2                    ;
   28   27 C160EF LDA VN,VN,R00,CND4      ;
   29   28 4000EF NOP                     ;
   30   29 1100EF CVT1                    ;
   31   30 C160EF LDA VN,VN,R00,CND4      ;
   32   31 4000EF NOP                     ;
   33   32 0100EF CVT0                    ;
   34   33 4062DF LDA X,DAR,L01           ;
   35   34 4468E7 ABS L,X                 ;
   36   35 4ECAEB SUB L,KP7               ;
   37   36 4EC24A SUB L,KP6,R03           ;
   38   37 4ECAFF LDA Z,KP7               ; LIMITER
   39   38 4EC25C ADD Z,KP6,R03           ;
   40   39 74EADD ADD  Z,KM7,L01,CNDS     ;
   41   40 76E23C ADD Z,KM6,R02,CNDS      ;
   42   41 4264EF LDA DAR,L               ;
   43   42 7468FF LDA Z,X,R00,CNDS        ;
   44   43 4864EF LDA DAR,FS              ;
   45   44 F73B1C ADD P,Z,R01,CND7        ;
   46   45 E73B3C ADD P,Z,R02,CND6        ;
   47   46 D73B5C ADD P,Z,R03,CND5        ;
   48   47 C73B7C ADD P,Z,R04,CND4        ;
   49   48 B73B9C ADD P,Z,R05,CND3        ; 8 BIT MULTIPLY
   50   49 A73BBC ADD P,Z,R06,CND2        ;  SIN*INPUT=MIXER
   51   50 973BDC ADD P,Z,R07,CND1        ;
   52   51 873BFC ADD P,Z,R08,CND0        ;
   53   52 668BDB SUB Z,Z,L01             ;
   54   53 763BFD ADD P,Z,R00,CNDS        ;
   55   54 D000EF OUT5                    ;
   56   55 A668DB SUB Z,Z,L01,OUT2        ;
   57   56 AA083C ADD V2,P,R02,OUT2       ;
   58   57 A4082C ADD V0,V2,R02,OUT2      ;
   59   58 A4086C ADD V0,V2,R04,OUT2      ; FINISH LOOP FILTER
   60   59 A408CA SUB V0,V2,R07,OUT2      ;
   61   60 A4084D ADD V0,V2,R11,OUT2      ;
   62   61 A610ED ADD VV,V0,R00,OUT2      ;
   63   62 A6100C ADD VV,V0,R01,OUT2      ;
   64   63 AA70FB SUB OSC,VOUT,R00,OUT2   ; GENERATE SAWTOOTH
   65   64 AE50EF LDA VOUT,VV,OUT2        ;  FREE+ERROR PROP  V
   66   65 486CEF LDA DAR,OSC,R00         ;
   67   66 78D2DD ADD OSC,KP4,L01,CNDS    ;
   68   67 4A64CF LDA DAR,VOUT,L01        ;
   69   68 487BEF LDA FS,OSC              ;
   70   69 48D2EB SUB FS,KP4              ;
   71   70 4870C7 ABS FS,FS,L01           ; GENERATE CLIPPED SINE
   72   71 48D2EB SUB FS,KP4              ;  WAVE FROM SAWTOOTH
   73   72 4870CD ADD FS,FS,L01           ;
   74   73 44D2FF LDA FC,KP0,R00          ;
   75   74 DC78FB SUB FC,OSC,OUT5         ;  GENERATE SQUARED
   76   75 BCD2FD ADD FC,KP4,OUT0         ;   COSINE WAVE FROM
   77   76 BE78F5 LIM FC,FC,R00,OUT0      ;    SAWTOOTH
   78   77 8A21CF LDA VPRF,VOUT,L01,OUT0  ;
   79   78 8000EF OUT0                    ;
   80   79 8000EF OUT0                    ;
   81   80 8000EF OUT0                    ;
   82   81 8000EF OUT0                    ;
   83   82 8000EF OUT0                    ;
   84   83 8000EF OUT0                    ;
   85   84 8000EF OUT0                    ;
   86   85 426CEF LDA DAR,Z,R00           ;
   87   86 F1CB0C ADD VN,FD,R01,CND7      ;
   88   87 E1CB2C ADD VN,FD,R02,CND6      ;
   89   88 D1CB4C ADD VN,FD,R03,CND5      ;
   90   89 C1CB6C ADD VN,FD,R04,CND4      ;
   91   90 B1CB8C ADD VN,FD,R05,CND3      ; 8 BIT MULTIPLY
   92   91 A1CBAC ADD VN,FD,R06,CND2      ;  COS*INPUT=LOCK MIXER
   93   92 91CBCC ADD VN,FD,R07,CND1      ;
   94   93 81CBEC ADD VN,FD,R08,CND0      ;
   95   94 4089DB SUB FD,FD,L01,NOP       ;
   96   95 70CBED ADD VN,FD,R00,CNDS      ;
   97   96 00C6EF LDA DAR,KP0,IN0         ;
   98   97 0008EF LDA REG,V2,IN0          ;
   99   98 0000FF LDA V2,REG,IN0          ; LOOP FILTER FEEDBACK
  100   99 00003A SUB V2,REG,R02,IN0      ;
  101  100 00009C ADD V2,REG,R05,IN0      ;
  102  101 0000FA SUB V2,REG,R08,IN0      ;
  103  102 44B2EF LDA V0,KP0             ;
  104  103 64006A SUB V0,REG,R04,CVT5     ;
  105  104 4400AA SUB V0,REG,R06          ; LOOP FILTER
  106  105 EBE6ED ADD DAR,KM2,R00,CND6    ;  FEED FORWARD
```

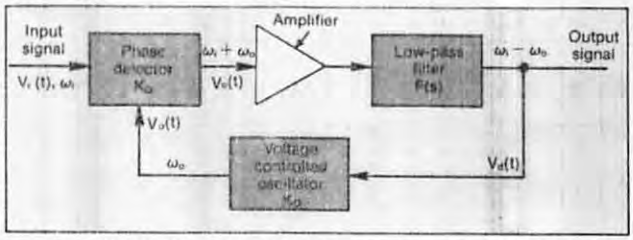
```
 LINE  LOC OBJECT SOURCE STATEMENT
  107  106 44000B SUB V0,REG,R09          ;
  108  107 75004B SUB V0,REG,R11,CVT7     ;
  109  108 40302C ADD VL,VN,R02           ;
  110  109 4B10BC ADD VG,VL,R06           ; FINISH LOCK FILTER
  111  110 6910DC ADD VG,VL,R07,CVT6      ;
  112  111 48103B SUB VG,VL,R10           ;
  113  112 4492FF LDA P,KP0,R00           ;  CLEAR MULTIPLIER
  114  113 559AEF LDA VV,KP1,R00,CVT5     ;
  115  114 449A0C ADD VV,KP1,R01          ; LOAD FREE RUN VOLTAGE
  116  115 449ABC ADD VV,KP1,R05          ;  PROP TO FREQ
  117  116 4100EF CVT4                    ;
  118  117 4818BF LDA VG,VG,L02           ;
  119  118 4000EF NOP                     ;
  120  119 3100EF CVT3                    ;
  121  120 CF10EF LDA VV,VV,R00,CND4      ;
  122  121 4000EF NOP                     ;
  123  122 2100EF CVT2                    ;
  124  123 CF10EF LDA VV,VV,R00,CND4      ;
  125  124 4000EF NOP                     ;
  126  125 1100EF CVT1                    ;
  127  126 CF10EF LDA VV,VV,R00,CND4      ;
  128  127 4000EF NOP                     ;
  129  128 0100EF CVT0                    ;
  130  129 4062DF LDA X,DAR,L01           ;
  131  130 4468E7 ABS L,X                 ;
  132  131 4ECAEB SUB L,KP7               ;
  133  132 4EC24A SUB L,KP6,R03           ;
  134  133 4ECAFF LDA Z,KP7               ; LIMITER
  135  134 4EC25C ADD Z,KP6,R03           ;
  136  135 74EADD ADD Z,KM7,L01,CNDS      ;
  137  136 76E23C ADD Z,KM6,R02,CNDS      ;
  138  137 4264EF LDA DAR,L               ;
  139  138 7468FF LDA Z,X,R00,CNDS        ;
  140  139 4864EF LDA DAR,FS              ;
  141  140 F73B1C ADD P,Z,R01,CND7        ;
  142  141 E73B3C ADD P,Z,R02,CND6        ;
  143  142 D73B5C ADD P,Z,R03,CND5        ;
  144  143 C73B7C ADD P,Z,R04,CND4        ;
  145  144 B73B9C ADD P,Z,R05,CND3        ; 8 BIT MULTIPLY
  146  145 A73BBC ADD P,Z,R06,CND2        ;  SIN*INPUT=LOOP MIXER
  147  146 973BDC ADD P,Z,R07,CND1        ;
  148  147 873BFC ADD P,Z,R08,CND0        ;
  149  148 D68BDB SUB Z,Z,L01,OUT5        ;
  150  149 763BFD ADD P,Z,R00,CNDS        ;
  151  150 AA083C ADD V2,P,R02,OUT2       ;
  152  151 A4082C ADD V0,V2,R02,OUT2      ;
  153  152 A4096C ADD V0,V2,R04,OUT2      ; FINISH LOOP FILTER
  154  153 A408CA SUB V0,V2,R07,OUT2      ;
  155  154 A4084D ADD V0,V2,R11,OUT2      ;
  156  155 A610ED ADD VV,V0,R00,OUT2      ;  FRONT END OF
  157  156 A6100C ADD VV,V0,R01,OUT2      ;   SAWTOOTH GENERATOR
  158  157 A000EF OUT2                    ;
  159  158 A000EF OUT2                    ;
  160  159 A000EF OUT2                    ;
  161  160 484CAF LDA DAR,V0,L02          ;
  162  161 4000EF NOP                     ;
  163  162 4000EF NOP                     ;
  164  163 4000EF NOP                     ;
  165  164 4000EF NOP                     ;
  166  165 4000EF NOP                     ;
  167  166 4000EF NOP                     ;
  168  167 D000EF OUT5                    ;
  169  168 9000EF OUT1                    ;
  170  169 9000EF OUT1                    ;
  171  170 9000EF OUT1                    ;
  172  171 9000EF OUT1                    ;
  173  172 9000EF OUT1                    ;
  174  173 9000EF OUT1                    ;
  175  174 9A70FB SUB OSC,VOUT,R00,OUT1   ;
  176  175 9E50EF LDA VOUT,VV,OUT1        ;
  177  176 486CEF LDA DAR,OSC,R00         ; SAWTOOTH GENERATOR
  178  177 78D2DD ADD OSC,KP4,L01,CNDS    ;
  179  178 4A64CF LDA DAR,VOUT,L01        ;
  180  179 487BEF LDA FS,OSC              ;
  181  180 48D2EB SUB FS,KP4              ;
  182  181 4870C7 ABS FS,FS,L01           ; GENERATE CLIPPED SINE
  183  182 48D2EB SUB FS,KP4              ;  WAVE FROM SAWTOOTH
  184  183 4870CD ADD FS,FS,L01           ;
  185  184 84D2FF LDA FC,KP0,OUT0         ;
  186  185 8C78FB SUB FC,OSC,OUT0         ;
  187  186 8CD2FD ADD FC,KP4,OUT0         ;  GENERATE SQUARED
  188  187 8E78F5 LIM FC,FC,OUT0          ;   COSINE WAVE FROM
  189  188 5A29FF LDA FD,FC,EOP           ;    SAWTOOTH
  190  189 8A21CF LDA VPRF,VOUT,L01,OUT0  ;
  191  190 8000EF OUT0                    ;
  192  191 8000EF OUT0                    ;
  193              END

SYMBOL:          VALUE:

REG                 0
V2                  1
V0                  2
DEL                 3
VL                  4
VG                  5
VV                  6
P                   7
VN                  8
X                   9
L                  10
Z                  11
FS                 12
OSC                13
VOUT               14
FC                 15
VPRF               16
FD                 17
V0                 18

                    ASSEMBLY COMPLETE
                    ERRORS     =     0
                    WARNINGS   =     0
                    RAMSIZE    =    19
                    ROMSIZE    =   192
```

Analog signal processor



10. This block diagram of the PLL macromodel highlights its dc characteristics and their relationship within the loop.

The product of the sine-wave input and the square-wave VCO signal contains a $2/\pi$ factor. This value is modified by $\pi/2$ because of the inherent gain of the mixer (represented by the multiplier). The limit value given by the 2920 specification sheet for K_{Dmax} is modified by $\pi/2$ and becomes approximately ± 0.96875. (Refer again to Fig. 3, bottom.)

The variable K_D is represented by the gain stage ahead of the limiter. Its gain is the ratio of the limit and the peak input signal voltages when K_D reaches its maximum value. Though omitted in the macromodel, this gain scales down the input-signal sensitivity by a factor of 45 from its ± input level to fill the 2920's digital-to-analog register.

The phase-detector output is filtered so that the VCO input contains only the difference frequency component of the first harmonic. This VCO input (volts) is multiplied by the VCO gain constant, K_o, (radians/volt), giving an output in radians. This output, when divided by 2π, represents the difference, Δf, between the input and VCO frequencies.

The VCO frequency and frequency difference summation ($f_o + \Delta f$) must be integrated to derive the phase difference between the input and VCO signals. This integrated signal, representing the instantaneous phase difference between input and output signals, drives the sawtooth generator to produce the VCO frequency.[2]□

References

1. Malmberg, A.F., *User's Manual ET-2 Network Analysis Program*," Braddock, Dunn and McDonald, Inc., El Paso, TX, September, 1973.

2. Tanenhaus, Martin E., and Martin, Robert L., "Linear Integrated Circuit Models and Applications for Computer-Aided Circuit and Systems Analysis and Design," *Proceedings of the Eighth Asilomar Conference on Circuits, Systems, and Computers*, December, 1974.

3. *The 2920 Analog Signal Processor Design Handbook*, Intel Corp., Santa Clara, CA, August, 1980.

4. Tanenhaus, Martin E., and Martin, Robert L., "Linear Integrated Circuit Models and Applications for Computer-Aided Circuit and Systems Analysis and Design," Naval Surface Weapons Center, White Oak Lab, Silver Spring, MD," April, 1976.

How useful?	Circle
Immediate design application	550
Within the next year	551
Not applicable	552