intel®

APPLICATION NOTE

AP-92

February 1980

Implementation of a Scanning Spectrum Analyzer Using the 2920 Signal Processor

Robert E. Holm
John S. Rittenhouse
Telecommunications
Product Marketing

FABIO MONTORO

AFN-01386A

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

Implementation of a Scanning Spectrum Analyzer Using the 2920 Signal Processor

Contents

1.0 INTRODUCTION

The purpose of a scanning spectrum analyzer is to determine the long term spectral characteristics of an analog signal. This is done by sweeping the input signal through a bandpass filter and displaying the filter output as a function of time. This application note describes the implementation of a sampled data scanning spectrum analyzer using the 2920 Signal Processor.

A scanning spectrum analyzer embodies many of the functions which can be found in a broad class of analog applications. These functions include: lowpass and bandpass filters, multipliers (mixers), detectors, and oscillators. The spectrum analyzer is a useful circuit which lends itself to applications such as speech processing, industrial control, medical electronics, signal detection, and signal processing.

The implementation of a spectrum analyzer using a sampled data system requires an understanding of sampling theory and digital signal processing as well as the ability to specify the system in analog terms. A basic review of sampling theory is provided in Section 2. Section 3 describes the 2920 Signal Processor. Section 4 describes the block diagram of the spectrum analyzer and discusses the design considerations. Once the block diagram of the application is complete, it is relatively straightforward to implement each subsystem as a block of code in the 2920 Signal Processor. Section 5 describes the implementation of the spectrum analyzer then gives a more detailed look at the actual design process using the signal processor resulting in the final 2920 assembly language program. A complete listing of the spectrum analyzer program is given in Appendix A.

2.0 SAMPLED DATA SYSTEM

Sampled data systems can be implemented using either analog or digital processing techniques or both. Examples of analog processing include transversal filters using CCD or bucket brigade shift registers and analog tap weights to implement transfer characteristics. The identical characteristics can also be implemented using digital processing. Since the use of digital processors are of interest here, it is useful to investigate the elements of a general purpose signal processor using a digital implementation.

2.1 Block Diagram Description

The block diagram shown in Figure 2.1 illustrates the basic blocks of a general purpose sampled data system using a digital signal processor. In this configuration it is assumed that both the input and output signals will be analog. (This is not a necessary condition since digital signals can be considered a special type of analog signal and processed accordingly.) The following paragraphs describe the function of each block in Figure 2.1.

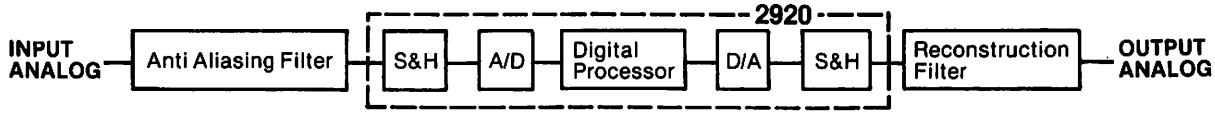
Anti-Aliasing Filter — This filter is used to bandlimit the incoming analog signal prior to sampling (thus a continuous analog filter is used) so that distortion terms (aliasing noise) due to the sampling process are minimized.

Input Sample and Hold (S&H) — The filtered input signal is sampled at an instant in time and the resulting sampled amplitude is held long enough for subsequent processing such as analog/digital conversion (Hold time ≤ the sample period).

Input Analog to Digital Converter (ADC) — The held analog voltage is converted to a digital word by comparing the voltage to discrete thresholds representing the digital words.

Digital Processor — This can be a general purpose processor or one specifically built to perform a predetermined algorithm. Typically, a general purpose microprocessor can be programmed to perform any function but the resulting execution time is too limiting for most real-time applications. A programmable digital signal processor such as the 2920 eliminates this problem because it is designed specifically for high speed signal processing while at the same time preserving its general purpose nature.

Digital to Analog Converter (DAC) — The processed digital words are converted back to analog using the DAC. Again, the analog signal is approximated by



Anti Aliasing Filter: Bandlimits input signal to reduce distortion due to sampling

S & H: Sample and hold performs sampling process and holds data sufficiently long for processing

A/D: Analog to digital conversion, generates a digital word to represent held analog voltage

Digital Processor: Implement transfer function using digital processing (under software control)

D/A: Converts digital words to analog voltages

Reconstruction Filter: Smooths D/A or S & H waveforms to recover continuous analog output signal

Figure 2.1. Elements of a Sampled Data System Using Digital Processing

discrete amplitude levels (as in the ADC). In addition, the DAC output weights the signal output in the frequency domain thereby causing some signal distortion.

Output Sampler — One method of reducing the output frequency distortion is to resample the output signal using a very narrow sample width. The sampler takes the DAC held output and resamples it with narrow sample pulses.

Reconstruction Filter — Since the desired output signal is a continuous representation of the processed input signal, it is necessary to remove high frequency components resulting from the DAC or sampler outputs. This in effect smooths the analog output from sample to sample. A lowpass filter is used to perform the signal "reconstruction."

The process of sampling a signal introduces certain amounts of distortion called aliasing noise. The conversion of the analog samples to digital words is done with an analog to digital converter (ADC). This circuit represents the analog signal with a digital word which corresponds to a discrete amplitude approximation to the signal. This process also introduces a distortion term called quantization noise. By properly designing the sampled data system, these distortion or "noise" terms can be made insignificantly small so that the sampled data system closely represents the analog equivalent system with all the advantages of digital processing.

2.2 Sampling Theory

A digital signal processor requires the conversion of the input analog signal to a digital signal. Inherent in this analog-to-digital conversion is the sampling of the continuous input signal. As would be expected, the method and rate of sampling of the input signal affect the information content of the sampled signal so that some degree of distortion is incurred when the input signal is analog reconstructed from the digital data samples.

A sampling theorem that relates the minimum required sampling frequency to the signal bandwidth can be stated as follows. If a signal f(t) (a real function of time) is sampled instantaneously at regular intervals and at a rate higher than twice the signal bandwidth that includes all the significant frequency components, then the samples contain all the significant information of the original signal. The signal bandwidth, as used above, is the lowpass bandwidth for video signals and the RF bandwidth for signals modulated on a carrier.

Two aspects of the sampling theorem must be investigated prior to the selection of a sampling frequency.

- What is the effect of finite-width samples (vs instantaneous samples) on the information content of the samples?
- How is it determined that the signal bandwidth contains all significant frequency components?

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

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

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

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

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

The time- and frequency-domain plots for the square-topped sampled signals are shown in Figure 2.2. It is clear in 2.2(d) that the sampling process acts as a lowpass filter with a sin x/x amplitude response. If this filter response is not constant across the signal bandwidth, some information content of the signal will be lost due to rolloff distortion.

The amount of spectral overlap seen between the lowpass spectrum and that centered about the sampling frequency is referred to as aliasing noise. The effect of sample rate on aliasing noise for a given input spectrum can be clearly seen in Figure 2.3. Note that the amount of overlap increases as the sampling frequency is

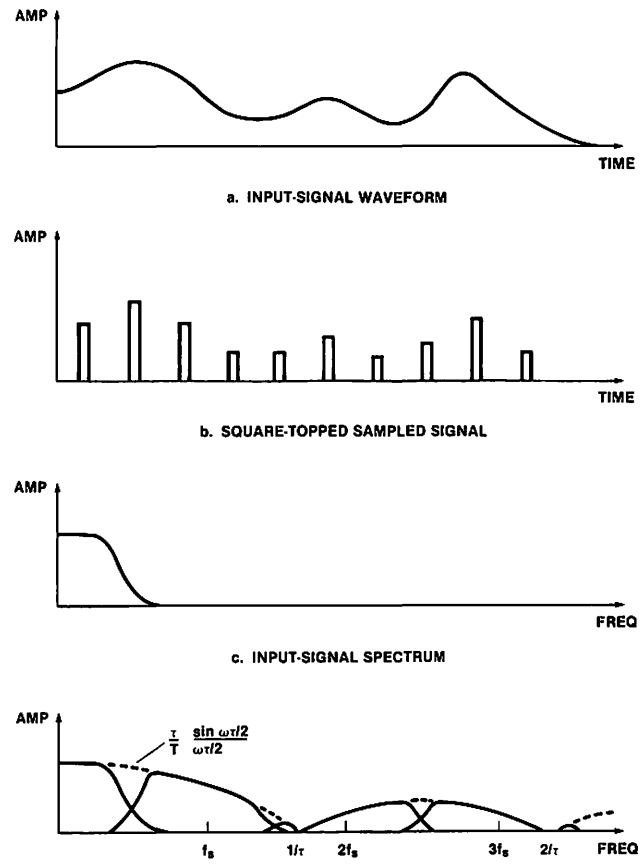


Figure 2.2. Analysis of a Sampled Signal

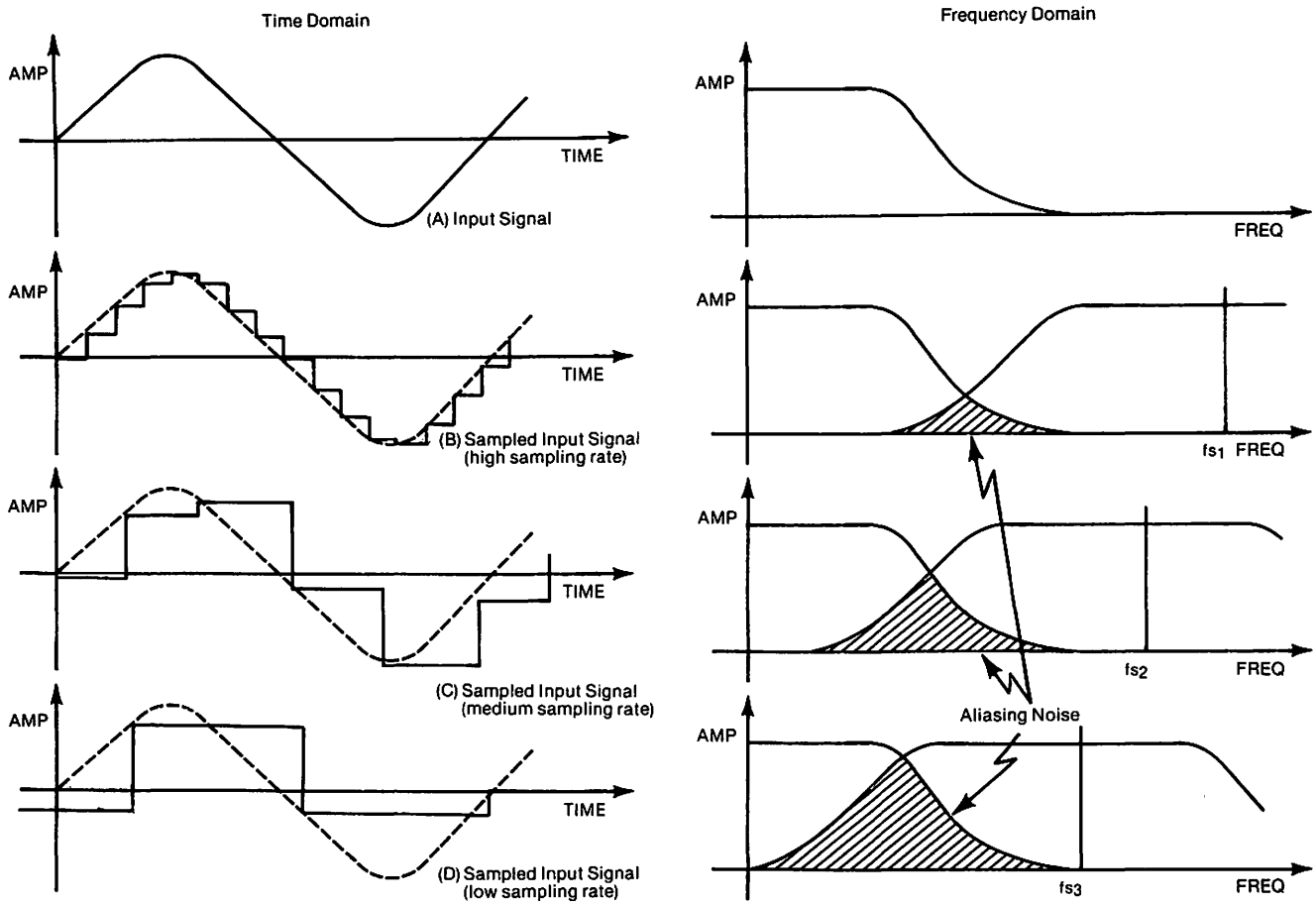


Figure 2.3. Effects of Sampling Rate on Aliasing Noise

decreased. In a similar manner, for a fixed sampling frequency, the overlap could be reduced by increasing the filter rolloff before sampling (anti-aliasing filter). Figure 2.4 illustrates the overlap for several popular filter types. These tradeoffs between filter selectivity and sampling frequency will be used in the spectrum analyzer design.

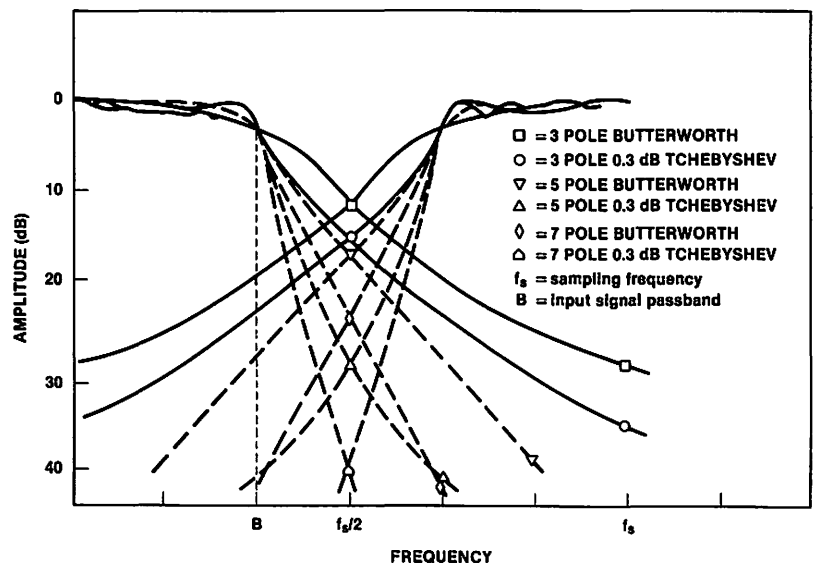


Figure 2.4. Effects of Filtering on Aliasing Noise

2.3 Quantization Noise

The analog-to-digital conversion of a signal implies that at specific times the signal is sampled and a digital word is formed that represents the amplitude of the signal at that time. The effect of the sampling process on the signal has been described, and it has been shown that a minimum loss of information is possible with the proper selection of bandwidths, sampling frequency, etc. The conversion from a continuous signal to a digital signal requires that the signal voltage be divided into M finite intervals which can be represented by an n-bit digital word, where

$$M = 2^n$$

The quantizing error can be expressed in terms of the total mean squared error voltage between the exact and the quantized samples of the signal. With reference to Figure 2.5, a signal voltage V(t) falls between the i^{th} and the $(i-1)^{th}$ levels which define the i^{th} quantizing interval. The error signal e_i is expressed as

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

where

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

$V(t)$ = input signal voltage

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

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

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

or represented as a logarithm

$$S/N_Q = (6)(n)\ dB$$

where

S is the peak signal power

N_Q is the mean quantization noise

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

n is the number of bits in the amplitude word

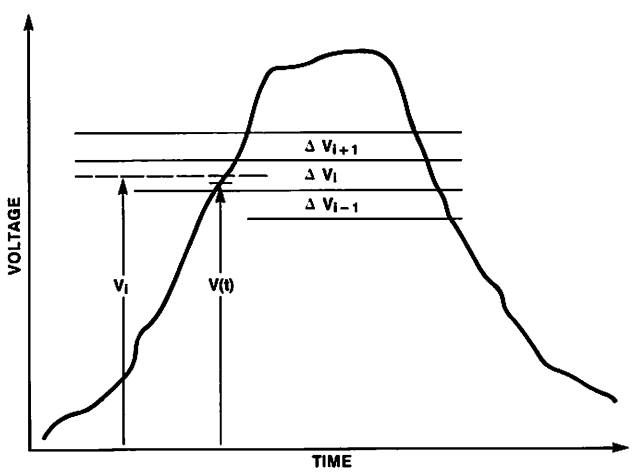
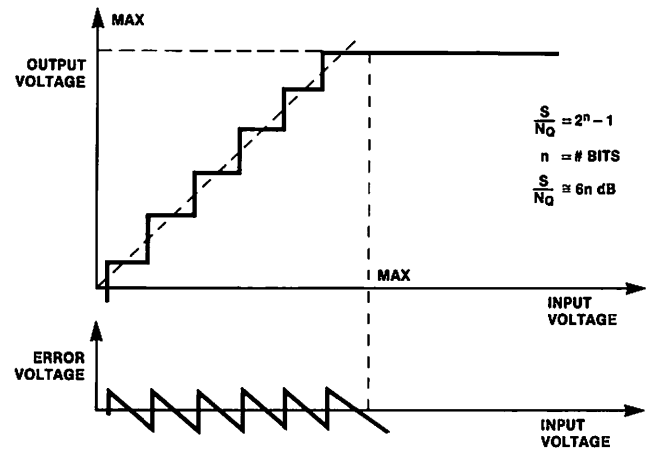


Figure 2.5. Quantization Step

Figure 2.6 illustrates the error voltage due to quantization and the corresponding peak signal to quantization noise ratio as a function of the digital word length (number of bits).



# BITS	2	3	4	5	6	7	8	9	10	11
S/N_Q dB	11.8	18	24	30	36	42	48	54	60	66

Figure 2.6. Quantization Noise

2.4 Signal Reconstruction

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

The basic assumption here is that a signal, which has been sampled and held for digital processing, is now to be converted back to analog form with minimum loss of information. The output of a sample-and-hold circuit (S&H) or a digital-to-analog converter (DAC) has a frequency spectrum as shown in Figure 2.7(a), where the sample width τ is equal to the period of the sample T. The amplitude gain factor is observed to have a noticeable rolloff within the signal spectrum when the sampling period is a significant portion of the shortest signal period. This represents a distortion of the output signal and, unless it is compensated for, it will cause some loss of information similar to that of a lowpass filter with an insufficient bandwidth.

To correct this situation, either the reconstruction sampling pulse width should be made narrow relative to the inverse signal bandwidth 1/B, or a sin x/x correction is needed in the output filter. Figure 2.7(b) shows the effect of resampling with a narrower pulse.

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

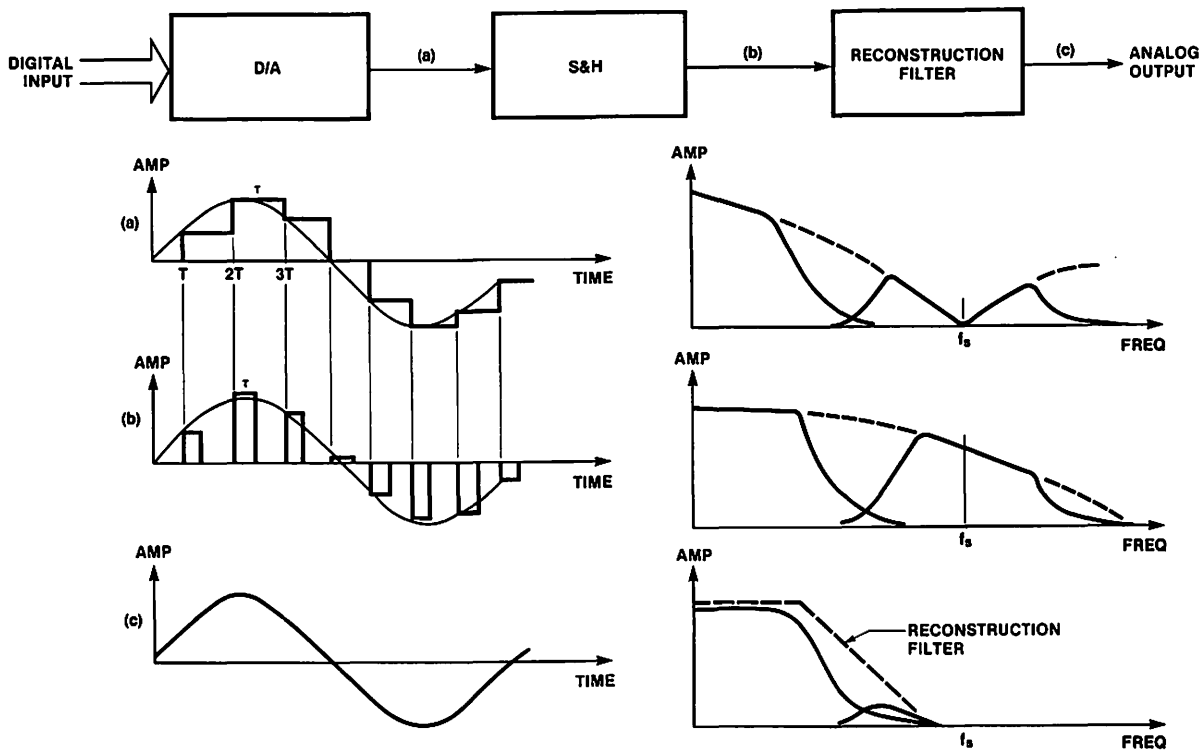


Figure 2.7. Analog Signal Reconstruction

When the data samples have been established, they are passed through a reconstruction lowpass filter that removes the high-frequency components of the sampled signal (Figure 2.7(c)). The purpose of the output lowpass filter is primarily to remove the high-frequency spectra caused by the output sampling. It can also be used to help shape the amplitude and phase response of the output network. The Intel 2912 PCM line filter can sometimes be used for both anti-aliasing and signal reconstruction filtering. The 2912 reconstruction filter also provides sin x/x correction.[1]

3.0 2920 SIGNAL PROCESSOR DESCRIPTION

The 2920 performs all the functions illustrated in Figure 2.1 beginning with the input S&H and ending with the output S&H. In addition, 4 input lines and 8 output lines are multiplexed to give the 2920 the capability of realizing several circuits, or one circuit with multiple inputs and outputs. A functional block diagram of the 2920 is shown in Figure 3.1.

The functions of the 2920 are controlled by the instructions stored in its 192-word EPROM. Each instruction is 24 bits long and is split into 5 fields, with each field controlling a subsystem of the 2920. In order to maintain a constant sample rate, the execution time for each instruction is identical and there is no conditional branching. The sample rate is determined by the program length and the instruction cycle time, which is four clock cycles or 400 ns at the maximum 10 MHz clock rate. A full 192 instruction program, running at 10 MHz, results in a 13 kHz sample rate. This allows for a signal bandwidth of approximately 4 kHz (maximum of 6.5 kHz with a rectangular filter). A shorter program will yield a higher sample rate.

3.1 Analog Operations

The 2920 input and output operations are under program control. To acquire an input signal, one of the 4 input lines is selected and the signal sampled and held. The resulting sample is then converted to digital form using a successive approximation A/D conversion. The result of the conversion can be up to 9 bits (a sign bit and 8 amplitude bits). However, since the A/D conversion is under program control, the conversion could consist of only a single bit which might be used to read a logical input for example.

The result of the A/D conversion is stored in the DAR. This register provides the interface between the analog and digital sections of the 2920. After the A/D conversion, the digital word in the DAR can be moved to a scratch pad RAM location for further processing. The DAR is also used for the output operation where the 9 most significant bits of the value to be output are loaded into the DAR. The DAR drives the D/A converter, which is connected via the output demultiplexer to one of the 8 output S&H circuits.

3.2 Digital Operations

The digital loop, shown in Figure 3.1, includes the 2 port, 40 word scratch pad RAM, a binary shifter, and the ALU. Two 25-bit words are fetched simultaneously from the 40 possible RAM locations. The data from the A port is

(1) R. E. Holm, "Data Conversion, Switching, and Transmission Using the Intel 2910A/2911A Codec and 2912 PCM Filter," Intel AP-64, p. 35.

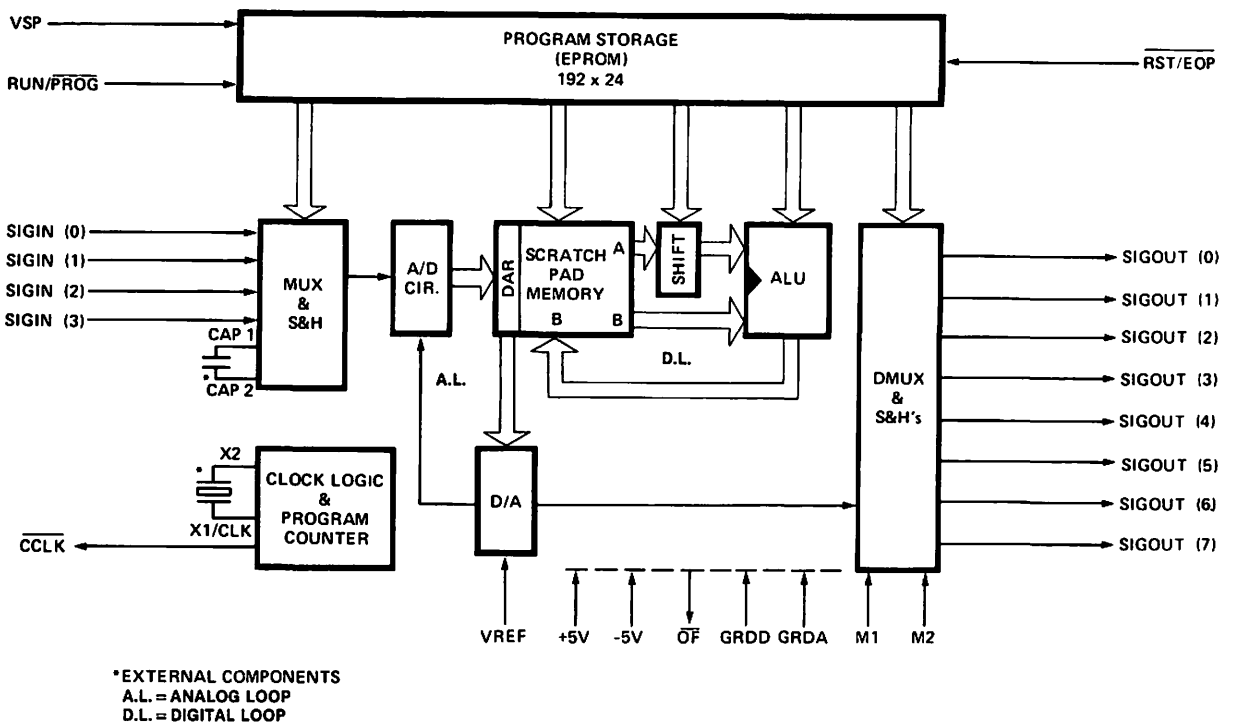


Figure 3.1. Functional Block Diagram (Run Mode)

passed through the binary shifter, which allows scaling from 2^2 (a 2-bit left shift) to 2^{-13} (a 13-bit right shift). The scaled A value and the unscaled B value are then passed on to the ALU. Sixteen internally available 4-bit constants are also accessible via the A port.

The ALU operates on the two values using the digital instructions specified by the program, and produces a 25-bit result. This result is then stored in the RAM location specified by the B address.

Digital and analog operations can execute simultaneously. For example, while doing a 9-bit A/D conversion, it is possible to implement a 5 pole lowpass filter with the digital section of the 2920. What gives the 2920 real-time processing capability is the ability to do the dual memory fetch, binary shift, ALU processing, and write back to memory as well as an analog operation in one instruction cycle (400 ns with a 10 MHz clock).

3.3 The 2920 Instruction Set

The following format is used by the 2920 assembler to specify the 24-bit instruction word stored in the EPROM:

ALU INSTRUCTION	B ADDRESS (DESTINATION)	A ADDRESS (SOURCE)	SHIFT CODE	ANALOG INSTRUCTION

Processing subsystems are implemented using a combination of analog and digital instructions.

The analog input and output instructions are IN(K) and OUT(K). To acquire a sample of the input signal, a sequence of IN(K) instructions is used. The A/D conversion is performed by the sign conversion and amplitude conversion instructions CVTS and CVT(K) respectively. A sequence of OUT(K) instructions will place a 9-bit amplitude on output channel K. Other analog instructions are the EOP instruction which resets the program counter to zero after the next three instructions are executed, NOP which is simply a no-operation, and CNDS or CND(K) which are conditional operators which test a bit of the DAR for the conditional ADD or LDA instruction, or define the destination of the carry bit for the conditional SUB instruction.

The arithmetic operations are ADD, SUB, and LDA which are addition, subtraction, and data transfer (load) respectively. These instructions may be modified with a conditional operator and used to perform multiplication or division by a variable or data dependent (conditional) switching. Other digital instructions are the absolute value ABS, the absolute value and add ABA, the ideal limit instruction LIM, and the logical instructions XOR and AND. Two special instructions, ABA CND( ) and XOR CND( ), are used to disable and enable, respectively, the ALU overflow saturation algorithm. The instruction set is given in Table 3.1. The 2920 internal constants and scalar mnemonics are listed in Table 3.2 and Table 3.3 respectively.

TABLE 3.1. INSTRUCTION SET AND OPERATIONS

Mnemonics		Operations		
Code / Condition				
Digital Instructions				
ADD		$(A \times 2^N) + B \longrightarrow B$[1]		
SUB		$B - (A \times 2^N) \longrightarrow B$		
LDA[3]		$(A \times 2^N) + 0 \longrightarrow B$		
XOR[3]		$(A \times 2^N) \oplus B \longrightarrow B$		
AND		$(A \times 2^N) \cdot B \longrightarrow B$		
ABS[3]		$[(A \times 2^N)] \longrightarrow B$		
ABA		$[(A \times 2^N)] + B \longrightarrow B$		
LIM		Sign(A) → ± F.S. → B[4]		
ADD	CND( )[2]	$(A \times 2^N) + B \longrightarrow B$		IFF DAR(K) = 1
		$B \longrightarrow B$		IFF DAR(K) = 0
SUB	CND( )[2][7]	$B - (A \times 2^N) \longrightarrow B$	& CY → DAR(K)	IFF $CY_P = 1$
		$B + (A \times 2^N) \longrightarrow B$	& CY → DAR(K)	IFF $CY_P = 0$ [5]
LDA	CND( )[2]	$(A \times 2^N) \longrightarrow B$		IFF DAR(K) = 1
		$B \longrightarrow B$		IFF DAR(K) = 0
ABA	CND( )[8]	$(A \times 2^N) + B \longrightarrow B$		
XOR	CND( )[8]	$(A \times 2^N) \oplus B \longrightarrow B$		
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		Program Counter to Zero[6]		
NOP		No Operation		
CND(K)		Select Bit K for Conditional Instructions		
CNDS		Select Sign Bit for Conditional Instructions		

Notes:
1. Note that scaling of A always occurs before executing the digital operation.
2. CND( ) can be either CND(K) or CNDS testing amplitude bits or the sign bit of the DAR respectively.
3. Clarification of CY_{OUT} sense for certain operations. For LDA, XOR, AND, ABS: $CY_{OUT} \rightarrow 0$.
4. B is set to full scale (F.S.) amplitude with the same sign as the "A" port operand.
5. The previous carry bit (CY_P) is tested to determine the operation. The present carry bit (CY) is loaded into the Kth bit location of the DAR. "Present carry (CY) is generated independent of overflow. It will represent the carry (CY) of a calculated 28-bit result."
6. EOP will also enable overflow correction if it was disabled during a program pass. The EOP must occur in ROM location 188.
7. For SUB CNDS operation $\overline{CY} \rightarrow$ DAR(S).
8. Does not affect DAR. In this case, CND is used with XOR/ABA to enable/disable the ALU overflow saturation algorithm. Use of either instruction causes the ALU output to roll over rather than go to full scale with sign bit preserved. An EOP instruction will also enable the ALU overflow sturation algorithm.

TABLE 3.2. CONSTANT MNEMONICS

Mnemonic	Value	Bit Sequence
KP0	0.000	0.000
KP1	0.125	0.001
KP2	0.25	0.010
KP3	0.375	0.011
KP4	0.5	0.100
KP5	0.625	0.101
KP6	0.75	0.110
KP7	0.875	0.111
KM1	−0.125	1.111
KM2	−0.25	1.110
KM3	−0.375	1.101
KM4	−0.5	1.100
KM5	−0.625	1.011
KM6	−0.75	1.010
KM7	−0.875	1.001
KM8	−1.0	1.000

The 2920 Assembler accepts these mnemonics as inputs to specify the 4-bit sequences shown in Table 3.2. These constants are stored as read only "RAM" locations in the 2920. The binary point is placed to the right of the most significant bit (MSB). Longer bit patterns may be obtained by shifting and adding several constants. A right shift operation in the 2920 fills the left most bit positions with the sign bit (0 for positive constants, 1 for negative constants). Negative constants are represented in 2's complement notation.

TABLE 3.3. SCALER MNEMONICS

Scaler Mnemonic	Equivalent Multiplier	Scaler Mnemonic	Equivalent Multiplier
L02	$2^2 = 4.0$	R06	$2^{-6} = 0.015625$
L01	$2^1 = 2.0$	R07	$2^{-7} = 0.0078125$
R00	$2^0 = 1.0$	R08	$2^{-8} = 0.00390625$
R01	$2^{-1} = 0.5$	R09	$2^{-9} = 0.001953125$
R02	$2^{-2} = 0.25$	R10	$2^{-10} = 0.0009765625$
R03	$2^{-3} = 0.125$	R11	$2^{-11} = 0.00048828125$
R04	$2^{-4} = 0.0625$	R12	$2^{-12} = 0.000244140625$
R05	$2^{-5} = 0.03125$	R13	$2^{-13} = 0.0001220703125$

4.0 DESCRIPTION OF SPECTRUM ANALYZER

The purpose of this spectrum analyzer is to determine the long term spectral characteristics of a signal in the 200 Hz to 3.2 kHz frequency band. The approach used is to sweep the input signal through a high resolution (narrowband) bandpass filter and observe the filter response as a function of the frequency sweep. First the spectrum analyzer block diagram and parameters are determined. Then sampled data considerations are taken into account, and finally the 2920 signal processor code is developed in Section 5.0.

4.1 Specifications

A spectrum analyzer which covers the audio frequency range of 200 Hz to 3.2 kHz was selected for this design. The specifications of the analyzer are given in Table 4.1.

TABLE 4.1 SPECTRUM ANALYZER SPECIFICATIONS

- INPUT BANDWIDTH: 3 kHz
- RESOLUTION BANDWIDTH: 100 Hz
- SWEEP RATE: 6 kHz/sec or 0.5 sec/Band
- DYNAMIC RANGE: 48 dB
- INPUTS — ANALOG SIGNAL: $-1V \leq SIG \leq 1V$
- OUTPUTS — FREQUENCY RESPONSE LINEAR AMPLITUDE (VERTICAL AXIS)
  - FREQUENCY RESPONSE LOG AMPLITUDE (VERTICAL AXIS)
  - SWEEP WAVEFORM (SAWTOOTH) (HORIZONTAL AXIS)
- OPTIONAL OUTPUTS — VCO (SWEEPING SINUSOID)

4.2 Block Diagram Description

Ideally, a scanning spectrum analyzer could be implemented by simply scanning a tunable narrowband bandpass filter across the input signal band to determine the signal energy at any frequency. Practically speaking it is nearly impossible to design a complex tunable analog filter which can cover a 10 to 1 range of frequencies, especially near DC. Even digital implementations become very complex and hardware inefficient when tuning is required. It is therefore easier to realize the equivalent of the scanning filter by sweeping the signal past a fixed tuned narrowband bandpass filter. This is accomplished by the superheterodyne system illustrated in the block diagram of Figure 4.1.

Additional Functions — The block diagram in Figure 4.1 shows the basic functions or subsystems which must be implemented to operate the spectrum analyzer. In the digital implementation there must also be an input anti-aliasing filter, sample and hold, A/D converter, and the corresponding output D/A converter and reconstruction filter. The analog-digital functions are implemented by a single 2920 signal processor in software.

Block Diagram — The input signal spectrum is first shaped by the input lowpass filter (LPF) (in addition to the anti-aliasing filter shaping) to avoid overlapping spectral components after mixing. The filtered signal then is multiplied (mixed) by the sweeping local oscillator (SLO) to generate upper and lower sidebands centered about the SLO frequency. The spectral characteristics of the system are shown in Figure 4.2. The bandpass filter (BPF) is centered at 4.5 kHz with a 100 Hz bandwidth. Figure 4.2(a) shows the filter characteristics. The SLO sweeps from 1.3 kHz to 4.3 kHz as seen in Figure 4.2(b). After mixing, the upper and lower sidebands are seen in Figures 4.2(c) and (d) for SLO frequencies of 1.3 and 4.3 kHz respectively. Only the upper sideband is of interest however as it is swept across the BPF and the signal energy is extracted. When the SLO is at 1.3 kHz the BPF is looking at the high band (3.2 kHz). As the SLO frequency increases, the apparent signal frequency seen by the BPF decreases until at a SLO frequency of 4.3 kHz, the BPF "sees" the signal energy at 200 Hz (4.5 kHz minus 4.3 kHz).

The block diagram shows that the BPF output is then passed through a full wave rectifier (FWR) and lowpass filter to extract the envelope from the 4.5 kHz carrier which is generated when signal energy is present. The resulting signal spectrum is centered at DC and shown in Figure 4.2(e).

The sweep output provides a horizontal sweep voltage for an X-Y display. The purpose of the delay shown in Figure 4.1 is to synchronize the sweep output with the amplitude response output. This delay should approximately equal the propagation delays of the BPF and output LPF.

I/O — The input to the spectrum analyzer is the analog signal to be analyzed. There are several outputs identified in Figure 4.1. These include the frequency sweep output which becomes the horizontal axis drive to a scope, the VCO output, and the BPF amplitude response (both linear and logarithmic) output which becomes the vertical axis drive to the scope.

4.3. Sampled Data System Considerations

An expansion of the frequency axis in Figure 4.2 to include the sampling frequency at 13 kHz shows the first order aliasing spectra as seen in Figure 4.3. From this figure the limitations and requirements for filter rolloff, bandwidths, and center frequencies become clearer.

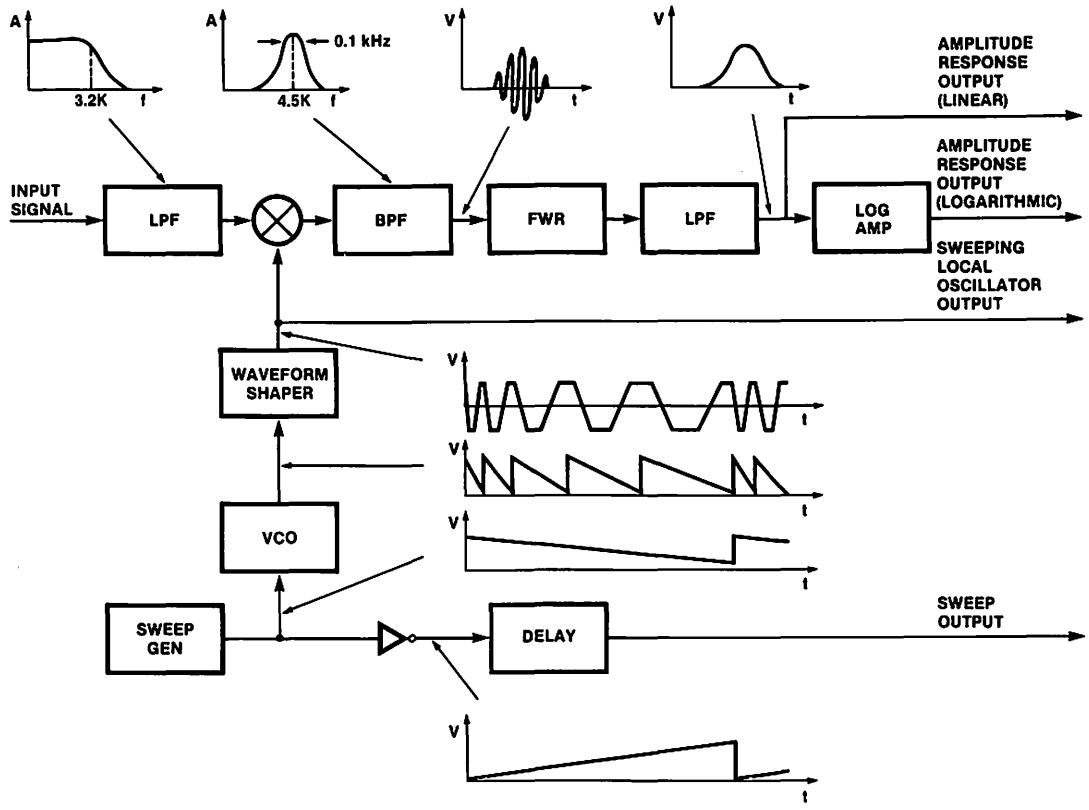


Figure 4.1. Spectrum Analyzer Block Diagram

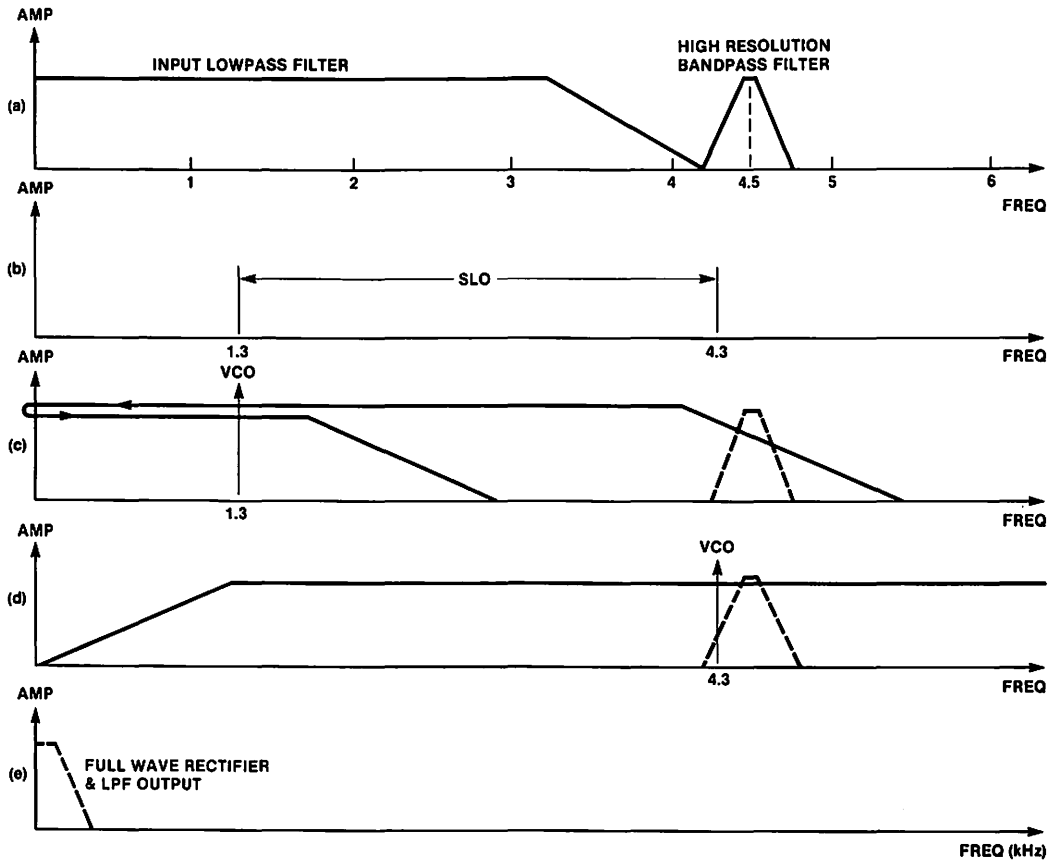


Figure 4.2. Frequency Domain Analysis of Spectrum Analyzer

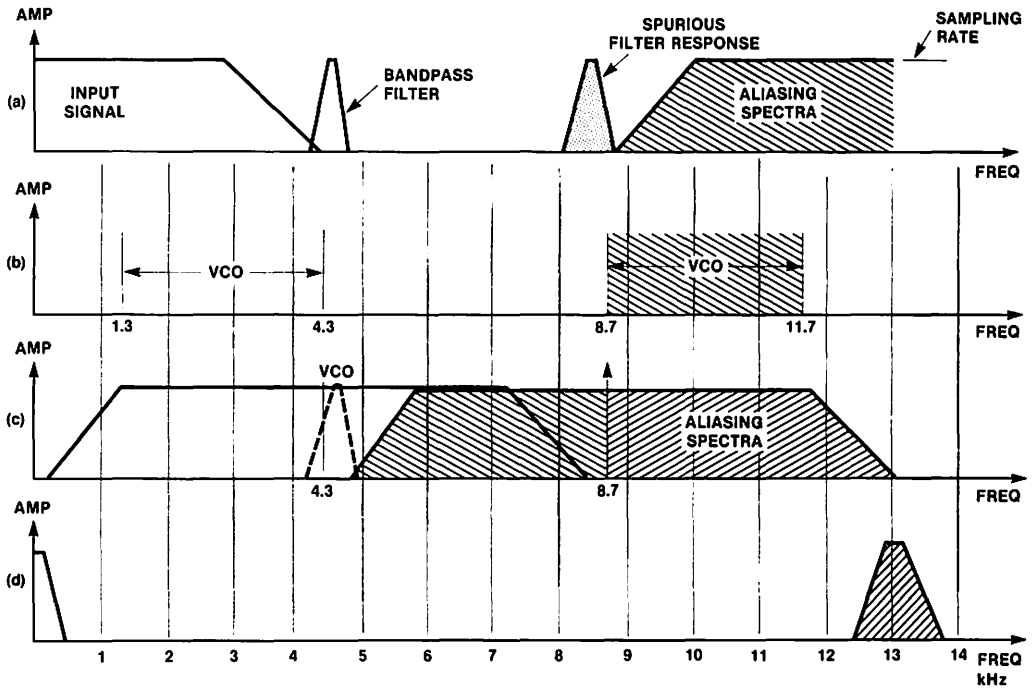


Figure 4.3. Frequency Domain Analysis of Sampled Data Spectrum Analyzer

Bandpass Filter — The location of the bandpass filter is determined by the input lowpass filter bandwidth and rolloff (Figure 4.3(a)) and the aliased spectrum of the lower sideband resulting when the SLO is at 4.3 kHz (Figure 4.3(c)). The BPF must have enough rolloff to eliminate both the baseband and aliased out-of-band signal components that are present. Analysis shows that a 3 pole pair Bessel filter will suffice if the input LPF is designed properly. The Bessel filter also has ideal transient response (no overshoot) so that the resulting output will not have overshoot and ringing.

Input LPF — This filter determines not only the baseband (centered about DC) spectrum but also that of the aliased lower sideband of the SLO. It was found that a 4 pole, 2 zero filter provides adequate rolloff to keep spurious signal (and aliased) components of significant amplitude (less than 48 dB down) out of the BPF passband.

Output LPF — This filter is used to remove the harmonic content of the FWR output (and the associated aliased components) before the signal is converted back to analog and outputted.

5.0 DESCRIPTION OF SUBSYSTEM IMPLEMENTATION

This section develops 2920 assembly code for several subsystems of the spectrum analyzer and discusses various aspects of the total program. The instruction set is given in Section 3 as well as a description of the 2920 device. The digitally implemented subsystems described here are: (1) sweep rate generator and SLO, (2) multiplier, (3) output lowpass filter, and (4) log amplifier. These are discussed following a discussion of the input anti-aliasing filter.

5.1 Anti-aliasing Filter

The basic function of the anti-aliasing filter is to attenuate the out-of-band spectral components of the input signal in order to reduce the effects of aliasing. From Figure 4.3 it is seen that with a 13 kHz sampling frequency (corresponding to a full 2920 program and a 10 MHz clock) the aliasing components must be below −50 dB at 3.2 kHz or 9.8 kHz from the sampling frequency. Therefore, the anti-aliasing filter attenuation characteristics are: (1) relatively little rolloff by 3.2 kHz (say 1 dB) and (2) 50 dB by 9.8 kHz. Filter curves (readily available in the literature) show that this would require a 6 pole Butterworth, or a 5 pole 0.5 dB ripple Tchebyshev, or equivalent.

Note that this filter is only needed if the input signal has significant frequency components above about 7 kHz. If a controlled signal is to be processed by the spectrum analyzer (such as sine waves or narrowband signals) then an anti-aliasing filter may not be needed.

5.2. Sweep Rate Generator (SRG) and Sweeping Local Oscillator (SLO)

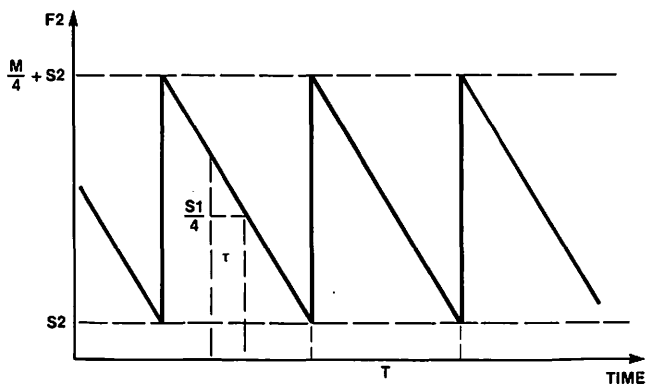
Development of the SRG and SLO are good examples of using time domain processing to avoid some of the problems which result from nonlinear processes creating aliasing components within the sampled system and beyond the help of an anti-aliasing filter. The purpose of the SRG is to provide the horizontal sweep output for an oscilloscope and to create an input to a voltage controlled oscillator (VCO) which will result in a linear frequency sweep as a function of time. The SLO is then a combination of the SRG and VCO, e.g. the sawtooth wave of the SRG drives the VCO resulting in a linear sweeping local oscillator which sweeps between

predetermined frequencies (1.3 kHz and 4.3 kHz) with a sweep rate determined by the period of the sawtooth wave.

Sweep Rate Generator — A sawtooth wave with an offset is the required output of this subsystem. The slope of the sawtooth determines the rate of change of frequency of the VCO, the voltage excursion is proportional to the frequency range of the VCO, and the offset represents the minimum frequency. As an output signal, the sawtooth provides a linear sweep for the horizontal axis of a scope (X-Y display) which is synchronized with the frequency sweep of the VCO. Based on the input specifications, a repetition rate of 2 sweeps/sec is needed.

The sawtooth wave is simply generated by continuously decrementing a register with a fixed value and thereby generating a linear negative slope. When the voltage changes sign (crosses zero) a constant equal to the sawtooth peak amplitude is added. This is accomplished by using an add (ADD) instruction conditioned on the sign bit. Once the sawtooth waveform is generated, it is scaled and a constant offset is added to provide a minimum voltage corresponding to the minimum frequency of the VCO. The resulting waveform and the 2920 program to generate this function is shown in Figure 5.1. Because of the low frequency of this signal (2.0 Hz) compared to the sampling frequency (13 kHz), all aliasing components are negligible. Therefore no action is needed to control them.

The program of Figure 5.1 presupposes the existence of two constants S1 and S2. These constants must be generated by the program prior to their use. Since each constant represents a sequence of ones and zeros, they can be generated several ways. One is to use a combination of shifts and adds of the constants KPx or KMx (see Section 3) to the register S1 or S2. Another approach is to read in a value from outside the chip by performing an A/D conversion of a DC voltage. This would allow both the sweep rate and the frequency range to be controlled externally. Figure 5.2 gives an example of creating the constant S1.



OP	DEST	SOURCE	SHF	COND	COMMENTS
SUB	F1	S1	R00	—	DECREMENT REGISTER
LDA	DAR	F1	R00	—	LOAD DAR
ADD	F1	M	R00	CNDS	CONDITIONED ADD
LDA	F2	F1	R02	—	$F_2 = F_1$ (SCALED)
ADD	F2	S2	R00	—	ADD OFFSET

Figure 5.1. VCO Input Waveform and 2920 Program Example

ASSUME FROM FIGURE 5.1 THAT

$\tau = 76.8\ \mu sec$
$M = 1.0$ VOLT
$T = 0.5$ sec

THEN $S1 = \frac{M}{(T/\tau)} = 15.36 \times 10^{-5} \frac{VOLTS}{STEP}$

CONVERTING TO BINARY YIELDS

$$S1 = 15.36 \times 10^{-5} = 2^{-13} + 2^{-15} + 2^{-20}$$
$$= (2^{-1} + 2^{-3} + 2^{-8})2^{-12}$$
$$= (0.10100001)2^{-12}$$

WE SEE THAT BY SPLITTING THE BINARY WORD INTO GROUPS OF 4 BITS YIELDS

$$S1 = \left[\boxed{0.101} + \boxed{0.001}\,2^{-5}\right]2^{-12}$$

FROM SECTION 3.0 TABLE 3.2 THIS EQUALS $[KP5 + KP1 \times 2^{-5}] \times 2^{-12}$ OR IN 2920 ASSEMBLY LANGUAGE

OP	DEST	SOURCE	SHF	COND
LDA	S1	KP5	R00	—
ADD	S1	KP1	R05	—
LDA	S1	S1	R12	—

LESS THAN ONE PERCENT ERROR IN THE SWEEP RATE IS INTRODUCED BY NEGLECTING KP1 R05. THEREFORE, 2 PROGRAM STEPS CAN BE SAVED AND THE CONSTANT CAN BE GENERATED WITH A SINGLE INSTRUCTION

LDA S1 KP5 R12

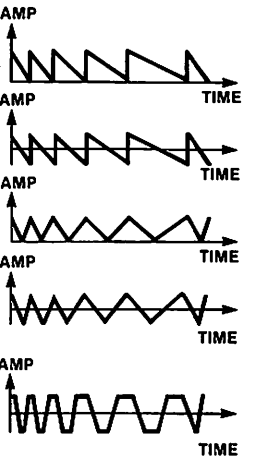
Figure 5.2. Developing the Constant S1 in the 2920

Voltage Controlled Oscillator — The VCO is developed in the same way as the SRG except that the decrement value is not a constant but rather is determined by a scaled version of the SRG input waveform. The calculation would be the same as shown in Figure 5.2 for both ends of the VCO frequency range. An offset would be determined by the low frequency and the scaling factor by the high frequency. The net result would be a sawtooth wave with a varying period as a function of time.

This high frequency sawtooth wave (1.3 kHz to 4.3 kHz) has significant harmonic content which will be reflected by the sampling frequency harmonics and cause distortion of the desired input to the mixer. Digital filters cannot be used here because they are susceptible to the aliasing components also. Some means must therefore be found to reduce the harmonic content of this signal. One approach would be to filter the VCO output using an external filter. This would involve additional hardware plus many extra instructions for I/O and A/D conversion. An alternative is to shape the waveform in the time domain to look more like the desired sinusoid.

By investigating the Fourier Transforms of various symmetric waveforms it is noticed that a trapezoidal waveform can be adjusted so that even harmonics are eliminated and the first odd harmonic is the fifth. This is done by selecting the top of the trapezoid to be 2/3 of the peak of a corresponding triangle wave. The program to accomplish this transformation is shown in Figure 5.3.

	OP	DEST	SOURCE	SHF	COND	COMMENTS
SLO	SUB	OSC1	F2	R00	—	
	LDA	DAR	OSC1	R00	—	
	ADD	OSC1	M	R00	CNDS	
WAVEFORM SHAPING	LDA	OSC	OSC1	R00	—	
	SUB	OSC	M	R01	—	CENTER WAVEFORM ABOUT 0
	ABS	OSC	OSC	L01	—	DOUBLE & TAKE ABS VALUE
	SUB	OSC	M	R01	—	RECENTER ABOUT 0
	ADD	OSC	OSC	L01	—	MULTIPLY BY 3. IF WE SET M = 1 THEN WAVEFORM WILL TRY TO REACH 1.5 AND WILL LIMIT AT 1 (60°) WHICH MINIMIZES 3RD HARMONICS



DEFINITIONS

F1 = SAWTOOTH WAVE VOLTAGE = SOURCE OF VARIABLE STEP SIZE TO SLO

M = CONSTANT VALUE ≤ 1.0

S2 = MINIMUM STEP SIZE FOR SLO

F2 = F1/4 + S2 = SLO STEP SIZE

OSC1 = BASIC SLO OUTPUT BUT WITH VARIABLE FREQ SAWTOOTH WAVE

OSC = FINAL SLO OUTPUT AFTER WAVESHAPING

$\text{SWEEP RATE} = \frac{S2}{T} \leq \text{RATE} \leq \frac{M/4 + S2}{T}$

T = SAMPLE PERIOD

Figure 5.3. Sweeping Local Oscillator (SLO) Program

5.3 Implementation of a 4 Quadrant Multiplier

The mixer shown in Figure 4.1 which multiplies the filtered input signal times the SLO waveform must be implemented as a 4 quadrant multiply since both waveforms have positive and negative values. A microprocessor implementation of this multiply might use a shift and add algorithm to determine the magnitude of the product and separate logic to determine the sign. A more direct algorithm is used in the 2920 to avoid the necessity of dealing with the sign bit separately.

Number Representation — It is convenient to form a representation of a number in 2's complement notation since this notation is hardware efficient and is used in the 2920. Assume that X is the multiplier number (sign and magnitude) and Y is the multiplicand. We can represent X in 2's complement as

$$X = (-1)s + s\sum_{i=0}^{n} \bar{b}_i 2^{-i} + \bar{s}\sum_{i=0}^{n} b_i 2^{-i} + s2^{-n}$$

$$= s\left[-1 + \sum_{i=0}^{n} \bar{b}_i 2^{-i} + 2^{-n}\right] + \bar{s}\sum_{i=0}^{n} b_i 2^{-i}$$

where s is the sign bit; 0 is positive, 1 is negative
b_i is the weighting and is either 1 or 0

This can be rewritten as

$$X = -s + x$$

where x represents the magnitude of the amplitude bits excluding the sign bit.

$$X = \sum_{i=0}^{n} \bar{b}_i 2^{-i} + 2^{-n} \quad \text{for } X < 0, \; s = 1$$

$$X = \sum_{i=0}^{n} b_i 2^{-i} \quad \text{for } X \geq 0, \; s = 0$$

Product Implementation — The product, $Z = X \times Y$, can now be determined as follows:

LET $X = (s,x) = -s + x$
$Y = (t,y) = -t + y$

WHERE s = SIGN BIT OF X
t = SIGN BIT OF Y
x = MAGNITUDE OF X
y = MAGNITUDE OF Y

THEN $Z = X \cdot Y$
$= (-s + x)(-t + y)$
$= st + xy - sy - tx$

Now the 2920 can easily implement the product of a positive multiplier and a bipolar multiplicand using a simple shift and conditional add algorithm. The add is conditioned on the value of the multiplier bit located in the DAR.

IF THE SIGN BIT IS IGNORED IN THE MULTIPLIER, X, THE RESULTING PRODUCT WILL BE

$Z' = (x)(-t+y)$

$= xt + xy$

THIS EXPRESSION LACKS THE TERMS

$st - sy = s(-Y)$

WHICH CAN BE ADDED TO FORM THE ENTIRE PRODUCT $Z = Z' + s(-Y)$ BY PERFORMING A CONDITIONAL ADD OF $-Y$ BASED ON THE VALUE OF "s."

The resulting 2920 Assembly code is shown in Figure 5.4 along with comments.

OP	DEST	SOURCE	SHF	COND	COMMENTS
LDA	DAR	X	R00	—	SET UP DAR FOR CONDITIONAL ADD'S, X IS MULTIPLIER
ADD	Z	Y	R01	CND 7	MULTIPLY Y BY THE MAGNITUDE OF X, THAT IS x WHERE $Z = x(-t+y)$
ADD	Z	Y	R02	CND 6	
ADD	Z	Y	R03	CND 5	
ADD	Z	Y	R04	CND 4	
ADD	Z	Y	R05	CND 3	
ADD	Z	Y	R06	CND 2	
ADD	Z	Y	R07	CND 1	
ADD	Z	Y	R08	CND 0	
SUB	Y	Y	L01	—	DEVELOP $-Y$ $Y \leftarrow Y - 2Y = -Y$
ADD	Z	Y	R00	CNDS	CONDITIONAL ADD OF "$-Y$" IF SIGN OF X IS NEGATIVE
SUB	Y	Y	L01	—	RESTORES ORIGINAL SIGN OF Y IF NEEDED

Figure 5.4. 4 Quadrant Multiply Program

5.4 Design and Implementation of the Output Lowpass Filter

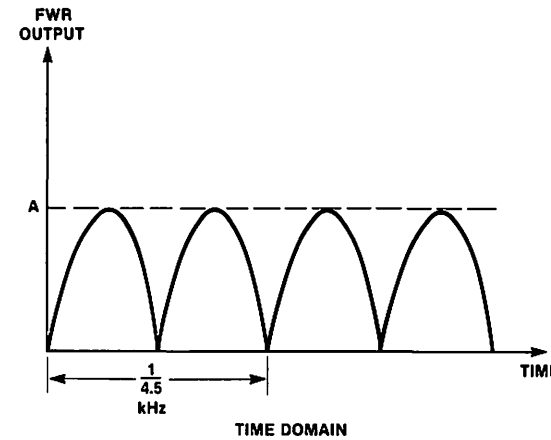
The primary purpose of the output lowpass filter is to eliminate the harmonic content of the full wave rectifier (FWR) output and the corresponding aliased components. The filter passband must be at least half that of the narrowband BPF (preferably wider) and the filter complexity should be minimized to reduce amplitude/phase distortion of the signal and ease implementation.

Design Considerations — The FWR spectral output is shown in Figure 5.5 along with the corresponding time domain waveform assuming quasi-static amplitude variation (relatively little change in amplitude over several 4.5 kHz carrier cycles). The desired signal information is located from DC to 50 Hz. All other signal components should be removed by filtering.

The spectral components illustrated in Figure 5.5 will also be centered about multiples of the sampling frequencies (aliasing noise) and must be considered before selecting a filter. This process is tabulated in Table 5.1 where $\Delta f = |Mf_s - Nf_{FWR}|$, N is the FWR harmonic, M is the sampling frequency harmonic and the Δf amplitude is determined by N from Figure 5.5.

From Table 5.1 it is clear that the 1 kHz component ($N = 6$, $M = 2$) is the most critical since it is closest to the filter passband and also requires a full 25 dB of attenuation by the filter. The 4 kHz component ($N = 2$, $M = 1$) requiring 46.5 dB must also be considered.

A look at the attenuation characteristics of standard filters shows that both criteria are met with a 2 pole Butterworth filter with a bandwidth of 50 Hz.



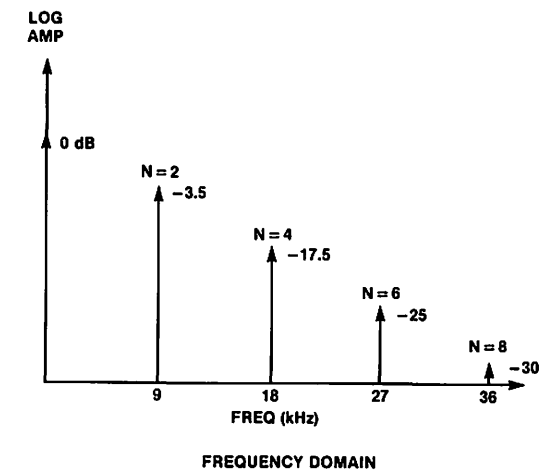


Figure 5.5. Harmonic Analysis of the Full Wave Rectifier (FWR)

TABLE 5.1. ALIASING COMPONENTS OF FWR OUTPUT

FWR[1] HARMONIC N	f_s[2] HARMONIC M	ALIAS[3] FREQ (kHz)	AMPLITUDE[4] LEVEL OF COMPONENT	ATTN REQ'D (to reach 50 dB)
2	1	4	−3.5 dB	46.5 dB
	2	17		
	3	30		
	4	43		
4	1	−5	−17.5 dB	37.5 dB
	2	8		
	3	21		
	4	34		
6	1	−14	−25 dB	25 dB
	2	−1		
	3	12		
	4	25		
8	1	−23	−30 dB	20 dB
	2	−10		
	3	3		
	4	16		
10	1	−32	−34 dB	16 dB
	2	−19		
	3	6		
	4	7		
12	1	−41	−37 dB	13 dB
	2	−28		
	3	−15		
	4	−2		

1. FWR Fundamental is 4.5 kHz.
2. Sample frequency is 13 kHz.
3. The alias frequency is the absolute value of the value shown. Negative frequencies fold around DC to become positive.
4. Determined from spectrum of FWR output (Figure 5.5).

Filter Implementation — The transfer function and s-plane pole-zero plot of the 2 pole Butterworth lowpass filter are shown in Figure 5.6(a) as a function of the 3 dB bandwidth B_3. Figure 5.7(a) shows a lumped parameter LC filter realization of this transfer function where L, C, and R are normalized lowpass prototype values which must be scaled by the actual resistive load and bandwidth of the filter. An approximation to this filter can be implemented digitally using the 2920 once the conversion from analog to digital or from s-plane to z-plane is performed.

The matched z – transform is defined as $z = e^{st}$ where s is a complex frequency defined by its real and imaginary parts (see Figure 5.6(a)) and T is the sample period of the sampled data system (the 2920 in this case). Expanding s into its components yields $z = e^{(\sigma T \pm j\omega T)} = e^{\sigma T}\ e^{\pm j\omega T}$ where σ is the real part and ω is the imaginary part of s. This final expression is recognized as a magnitude and a phase which is plotted in Figure 5.6(b) for the 2 pole Butterworth case ($|\sigma| = |j\omega| = 0.707B_3$). The real and imaginary parts of z can now be calculated and the resulting z transform transfer function G(z) determined as seen in Figure 5.6(b).

The transfer function G(z) can be realized digitally with the 2 stage recursive transversal filter shown in Figure 5.7(b) with the feedback coefficients B_1 and B_2 determined from the plot in Figure 5.6(b). The maximum gain through this filter configuration is given by the equation for G_{MAX} (Figure 5.7(b)). Input signal values must be normalized by a gain = $1/G_{MAX}$ or there will be overflows in the filter calculations.

The digital implementation using the 2920 Signal Processor uses RAM locations as the tap points and the transfer of data from one location to another each sample period T as the delay. The tap values (taken from the appropriate memory locations) are then multiplied by the appropriate coefficients using an efficient shift and add software multiply algorithm.

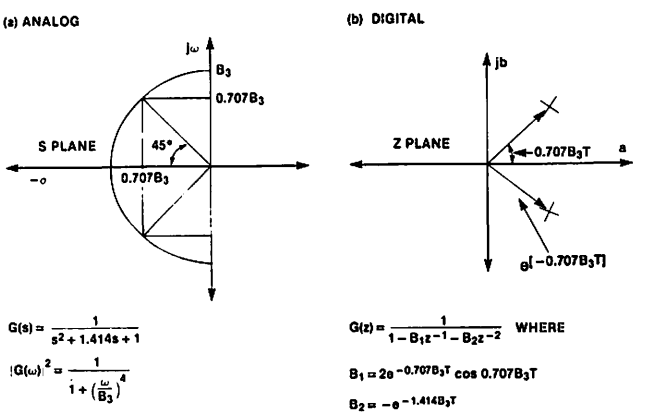


Figure 5.6. LPF Design Filter Transfer Function 2 Pole Butterworth LPF

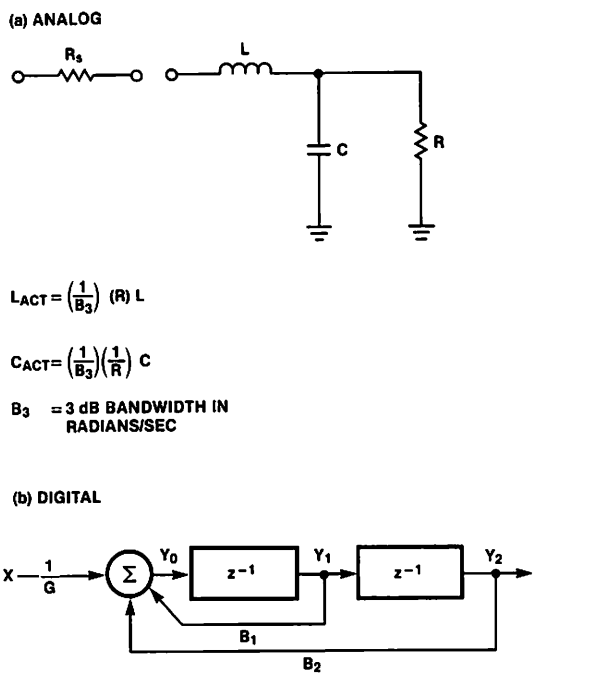


Figure 5.7. Analog and Digital Implementation of Butterworth 2 Pole Lowpass Filter

Filter Calculations — Based on the z-plane plot of poles for the Butterworth filter and the selection of a 50 Hz bandwidth and a 13 kHz sampling frequency, the coefficients of the digital filter can now be implemented. Figure 5.8 shows the calculations leading to both the filter coefficients and the gain weighting factor. The values of B_1 and B_2 after coding should be used to calculate the gain.

COEFFICIENTS

$T = 76.8\ \mu sec\ (f_s = 13\ kHz)$

$B_3 = 2\pi\ (50\ Hz) = 314.16\ rps$

$B_3T = 0.02413$

$B_1 = 1.9659$
$= 01.11110111010000$
$= 2.0 - 2^{-5} - 2^{-9} - 2^{-10}$

$B_2 = -0.966452$
$= -[0.11110111011010]$
$= -[1.0 - 2^{-5} - 2^{-9} - 2^{-12} - 2^{-13}]$

GAIN

$$G_{MAX} = \left[(1 + B_2)\sqrt{1 + \frac{B_1^2}{4B_2}}\right]^{-1}$$
$$= \frac{1}{0.000609}$$
$$= 1640$$

$$G = \frac{1}{G_{MAX}}$$
$$= 0.000609$$
$$> 2^{-11}\ (= 0.000488)$$

Figure 5.8. Digital Filter Calculations

2920 Assembly Language Program — Based on these binary values and their corresponding bit sequences, the lowpass filter can now be implemented digitally using 2920 assembly code. Figure 5.9 shows the program listing and comments which describe what each section of code is accomplishing. The filter variables are shown in Figure 5.7(b).

It can be noted from Figure 5.9 that the feedback of Y2 was begun before the feedback of Y1 was completed. This was done to avoid overflows during the summing of Y0. Although the maximum gain of the filter is known, and has been compensated for, the filter may still overflow during intermediate calculations for certain sequences of instructions.

Also, it should be noted that narrow band filters are very sensitive to coefficient precision. For example, the representation for B_2 required five terms. By omitting only the last term of this coefficient ($Y2 \cdot 2^{-13}$), the cutoff for this filter moves from 53 Hz to 65 Hz.

5.5 Logarithmic Amplifier

The logarithmic amplifier is a function which is usually included as part of a spectrum analyzer. Its purpose is to amplify low level signal components for easier comparison with larger signals. Furthermore, the log amplifier described here provides an example of the use of 2920 code to implement a piecewise linear approximation of a general function. The dynamic range of the amplifier is 50 dB with an error of less than 1 dB for signal levels to −30 dB. The transfer characteristic is shown in Figure 5.10.

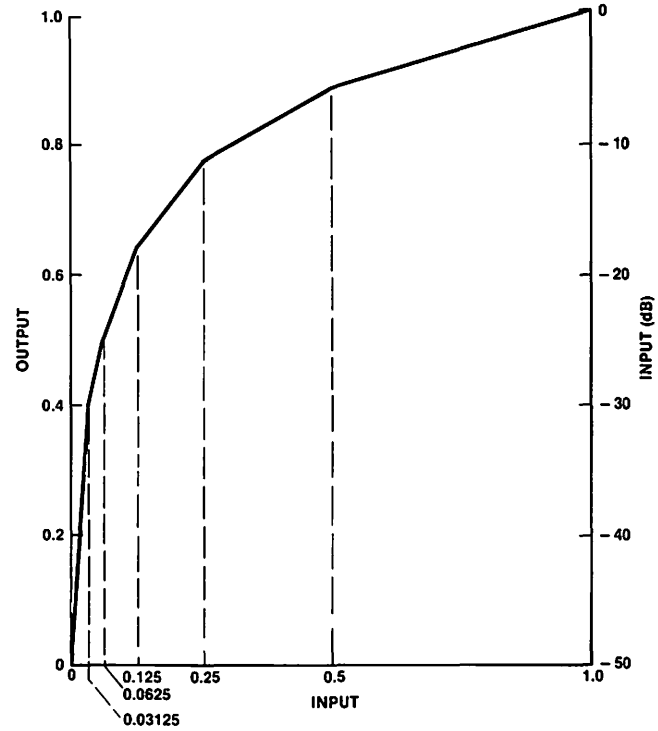


Figure 5.10. Transfer Characteristic of Piecewise Linear Log Amplifier

OP	DEST	SOURCE	SHF	CND	COMMENT
LDA	Y2	Y1	R00	—	PROPAGATES SAMPLES THROUGH DELAY LINE
LDA	Y1	Y0	R00	—	
SUB	Y0	Y1	R05	—	$1 - 2^{-5}$ (Y0 STILL CONTAINS Y1)
SUB	Y0	Y1	R09	—	FEEDBACK $B_1 Y_1$ TO Y0 WHERE $B_1 = 2.0 - 2^{-5} - 2^{-9} - 2^{-10}$
SUB	Y0	Y1	R10	—	
SUB	Y0	Y2	R00	—	
ADD	Y0	Y1	R00	—	
ADD	Y0	Y2	R05	—	FEEDBACK $B_2 Y_2$ TO Y0 WHERE $B_2 = -[1.0 - 2^{-5} - 2^{-9} - 2^{-12} - 2^{-13}]$
ADD	Y0	Y2	R09	—	
ADD	Y0	Y2	R12	—	
ADD	Y0	Y2	R13	—	
ADD	Y0	X	R11	—	ADJUST X BY GAIN $G = 2^{-11}$
LDA	DAR	Y0	R00	—	LOAD FILTER OUTPUT TO DAR FOR OUTPUTTING FROM 2920

Figure 5.9. Program for Digital Implementation of 2 Pole Butterworth Lowpass Filter

Six linear sections are used to approximate the log amplifier. The equations for these sections and the range of inputs for which each equation is used are given in Figure 5.11. The equations were obtained graphically, and then adjusted for coding efficiency. The input for the log amplifier must be positive and less than or equal to 1V. To simplify matters, the endpoints for the linear sections were chosen as powers of two. This way, only one bit of the number to be processed need be checked to determine whether that number falls within an input range. The constant multipliers (slopes) of the linear sections were chosen to minimize error while at the same time allowing the multiplications to be efficiently handled in 2920 code.

X = INPUT Y = OUTPUT

EQUATION	INPUT RANGE
$Y = 0.219(X) + 0.781$	$0.5 \leq X < 1$
$Y = 0.5(X) + 0.641$	$0.25 \leq X < 0.5$
$Y = X + 0.516$	$0.125 \leq X < 0.25$
$Y = 2(X) + 0.391$	$0.0625 \leq X < 0.125$
$Y = 4(X) + 0.270$	$0.03125 \leq X < 0.0625$
$Y = 12.75(X)$	$0 \leq X < 0.03125$

Figure 5.11. Breakpoint Equations for the Piecewise Linear Log Amplifier

The outputs for the log amplifier are also less than or equal to 1V, and positive. An output of 1V corresponds to 0 dB, 0.8V to −10 dB, 0.6V to −20 dB, and so on. An output of 0V corresponds to −50 dB or below. For example, for a device with a maximum output of 1V, an output of 0.7V indicates a signal level of −15 dB. Regardless of V_{REF}, a 2920 output which is 70 percent of full scale represents −15 dB. Any DC offset which may exist at the output of the part should be taken into account when interpreting the output in dB.

A flow chart of the log amplifier program is shown in Figure 5.12, and the assembly code is given in Figure 5.13. The first linear section of the amplifier to be implemented is the sixth section, which corresponds to inputs less than 1/32V. However, all input signals, regardless of amplitude, are processed by the equation for this section initially. The original signal is then placed in the DAR. All the following operations are conditional, and are performed only if the tested bit of the DAR is a "one." Otherwise, a NOP is performed. Each bit of the DAR is tested, starting with the least significant bit, until a "one" is found. Once a "one" is located, the multiplier and offset corresponding to the indicated range of the input are used to compute the result. This result replaces any previously computed result. If no "ones" are encountered, the input is less than 1/32V, and only NOP's are performed. The value computed for the sixth section then remains unmodified. Since the program starts checking for small signals and progresses to large signals, the computed value which corresponds to the signal range into which the input signal falls will be the final result.

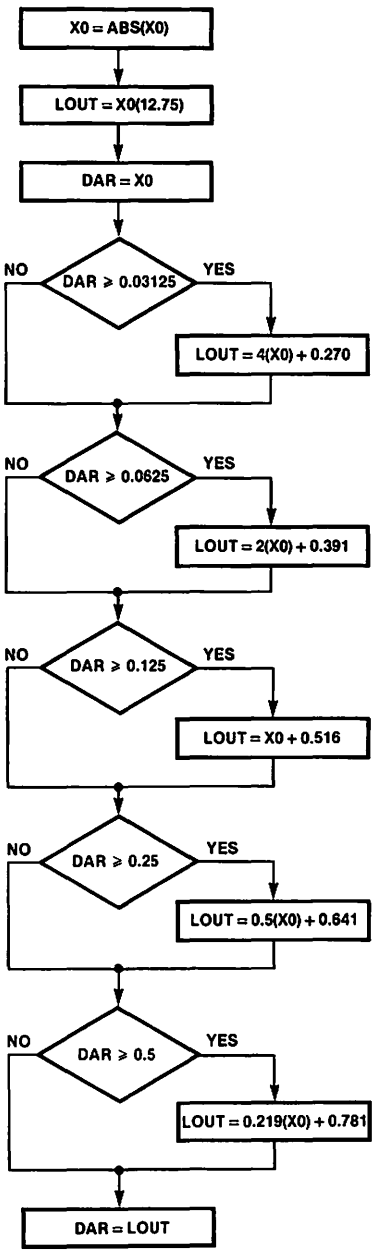


Figure 5.12. Flowchart for Implementation of Piecewise Linear Log Amplifier

```
***LOG AMP*****

ABS YO, YO, ROO                 ;PREVENT PROCESSING OF NEGATIVE NUMBERS

;   SECTION 6

LDA LOUT, YO, LO2               ;LOUT = 12.75(XO)  ,  0 < XO < 0.03125
ADD LOUT, YO, LO2
ADD LOUT, YO, LO2
ADD LOUT, YO, RO1
ADD LOUT, YO, RO2

LDA DAR, YO, ROO                ;TRANSFER INPUT TO DAR TO DO CONDITIONAL ARITHMETIC

;   SECTION 5

LDA LOUT, YO, LO2, CND3         ;LOUT = 4(XO) + 0.270  ,  0.03125 < XO < 0.0625
ADD LOUT, KP2, ROO, CND3
ADD LOUT, KP5, RO5, CND3

;   SECTION 4

LDA LOUT, YO, LO1, CND4         ;LOUT = 2(XO) + 0.391  ,  0.0625 < XO < 0.125
ADD LOUT, KP3, ROO, CND4
ADD LOUT, KP2, RO4, CND4

;   SECTION 3

LDA LOUT, YO, ROO, CND5         ;LOUT = XO + 0.516  ,  0.125 < XO < 0.25
ADD LOUT, KP4, ROO, CND5
ADD LOUT, KP2, RO4, CND5

;   SECTION 2

LDA LOUT, YO, RO1, CND6         ;LOUT = 0.5(XO) + 0.641  ,  0.25 < XO < 0.5
ADD LOUT, KP5, ROO, CND6
ADD LOUT, KP2, RO4, CND6

;   SECTION 1

LDA LOUT, YO, RO3, CND7         ;LOUT = 0.219(XO) + 0.781  ,  0.5 < XO < 1
ADD LOUT, YO, RO4, CND7
ADD LOUT, YO, RO5, CND7
ADD LOUT, KP6, ROO, CND7
ADD LOUT, KP4, RO4, CND7

LDA DAR, LOUT, ROO              ;TRANSFER RESULT TO DAR TO OUTPUT OR OTHER
                                ;REGISTER FOR FURTHER PROCESSING
```

Figure 5.13. Log Amplifier Assembly Code

6.0 SUMMARY

The examples given in Section 5 illustrate some of the design techniques used when implementing an analog function with the 2920 signal processor. The complete spectrum analyzer program listing is given in Appendix A. This listing contains subsystems which were not given as examples including the input lowpass and bandpass filters. Table 6.1 summarizes the number of instructions needed for each subsystem and the total number of instructions and scratch pad RAM locations needed for this implementation. The object code listing is also given in Appendix A. It contains the actual bit sequences loaded into the EPROM.

SUBSYSTEM	# INSTRUCTIONS
MISCELLANEOUS I/O	19
INPUT LOWPASS FILTER	28
MULTIPLIER	12
SRG	18
VCO	10
BPF	31
FWR + LPF	13
LOG AMPLIFIER	24
TOTAL INSTRUCTIONS	155
# RAM LOCATIONS	25

APPENDIX A. COMPLETE SPECTRUM ANALYZER ASSEMBLY LISTING

The spectrum analyzer program listed in this appendix (Figure A.2) was coded in a structured form, with each functional block coded separately and the blocks arranged to follow the signal paths shown in the block diagram of Figure 4.1. This was done for clarity in describing the program. It is not necessary to implement the code one functional block at a time or in any specific order as long as the relationships between the inputs and outputs of the functional blocks remains unchanged. In fact, it is usually more efficient to program the 2920 in a less structured form. For example, because each functional block is executed in its entirety before proceeding to the next functional block, it was not possible to execute all input and output instructions simultaneously with digital instructions. To take advantage of the fact that analog and digital instructions can execute simultaneously, portions of the program could be rearranged, and these analog instructions combined with digital instructions, thus reducing the program length.

The first functional block of the spectrum analyzer program is the 4 pole, 2 zero input filter. The sections titled Pole 1 and Pole 3 each represent a complex pole pair. The filter stage propagation is executed after the input signal is obtained. Stage propagation must be done before the complex zero pair can be implemented.

After the input filter program, the sweep waveform is generated to drive the VCO. This waveform is also inverted and delayed to form the horizontal output of the spectrum analyzer. The delay of 10 msec with respect to the VCO input compensates for the propagation delay of the bandpass and output filters. This delay is implemented in the time domain by simply subtracting a constant from the sawtooth waveform which corresponds to the change in amplitude of the waveform during a 10 ms period of time. The two NOP's which appear in the sweep oscillator sequence are part of the output sequence and are used to settle the D/A converter.

The VCO is implemented next. The sweeping sawtooth is set to zero at the beginning of each sweep so that the VCO output can be more easily observed with an oscilloscope. Once both the VCO waveform and the input signal have been obtained, they are multiplied together using the four quadrant multiply algorithm. With regard to Figure 5.4, OSC = X, MPL2 = Y, and MPL1 = Z.

The signal from the multiplier (mixer) is then passed to the 6 pole bandpass filter. Portions of the output sequences for the VCO and linear and log response outputs are also executed at this time. Executing these sequences simultaneously with the digital instructions saves program steps.

The signal is then processed by the full wave rectifier and output lowpass filter. (Referring back to Figure 5.9, BP50 corresponds to the input X. Also, in step 129, the right shift 11 of Figure 5.9 was replaced with a right shift 9. Since the output of the BPF is not full scale, less gain compensation is needed.) The output of this filter is the linear amplitude response of the spectrum analyzer. The log amplifier is the final section of the program, and provides a log amplitude response output. All unused program steps are NOP's. The symbol table used by the assembler is shown in Figure A.1, and a listing of the spectrum analyzer object code is given in Figure A.3.

Acknowledgement — The authors would like to thank Wallace Li for the work he did in developing the initial spectrum analyzer program and demonstrating its operation in a 2920.

SYMBOL:	VALUE:
TEMP	0
IF11	1
IF10	2
IF31	3
IF30	4
MPL2	5
S1	6
M	7
F1	8
SWP	9
F2	10
S2	11
OSC1	12
OSC	13
MPL1	14
BP11	15
BP10	16
BP31	17
BP30	18
Y0	19
BP51	20
BP50	21
LOUT	22
Y2	23
Y1	24

Figure A-1. Spectrum Analyzer Symbol Table

```
ISIS-II 2920 ASSEMBLER X102                                    PAGE    1

ASSEMBLER INVOKED BY: AS2920 SPEC4 DEBUG


 LINE  LOC OBJECT SOURCE STATEMENT

    1    0 3066EB SUB DAR,  DAR,  R00, IN3      ;CLEAR DAR FOR A/D CONVERSION
    2    1 3000EF IN3
    3    2 3000EF IN3
    4    3 3000EF IN3
    5    4 3000EF IN3
    6    5 3000EF IN3
    7    6 4000EF NOP
    8    7 4000EF NOP
    9    8 6000EF CVTS
   10    9 EBE6ED ADD DAR,  KM2,  R00, CND6     ;A/D CONVERSION INSTRUCTION
   11   10 4000EF NOP
   12   11 4000EF NOP
   13   12 7100EF CVT7
   14   13 4000EF NOP
   15
   16              ;*****INPUT FILTER*****
   17
   18              ;POLE 1
   19   14 4008EF LDA TEMP, IF11, R00, NOP
   20   15 6300FF LDA IF11, IF10, R00, CVT6
   21   16 46002A SUB IF10, IF10, R02, NOP
   22   17 4600AA SUB IF10, IF10, R06, NOP
   23   18 57000D ADD IF10, IF10, R09, CVT5
   24   19 44002A SUB IF10, TEMP, R02, NOP
   25   20 4400AC ADD IF10, TEMP, R06, NOP
   26   21 45000D ADD IF10, TEMP, R09, CVT4
   27   22 44002D ADD IF10, TEMP, R10, NOP
   28   23 44006B SUB IF10, TEMP, R12, NOP
   29
   30              ;POLE 3
   31   24 3308EF LDA TEMP, IF31, R00, CVT3
   32   25 4C00FF LDA IF31, IF30, R00, NOP
   33   26 40100F LDA IF30, TEMP, R09, NOP
   34   27 21100A SUB IF30, TEMP, R01, CVT2
   35   28 40104A SUB IF30, TEMP, R03, NOP
   36   29 48106C ADD IF30, IF30, R04, NOP
   37   30 13184C ADD IF30, IF31, R03, CVT1
   38   31 42188A SUB IF30, IF31, R05, NOP
   39   32 4218CC ADD IF30, IF31, R07, NOP
   40   33 03182D ADD IF30, IF31, R10, CVT0
   41
   42              ;STAGE PROPAGATION
   43   34 44224C ADD IF10, DAR,  R03          ;ADD INPUT TO INPUT FILTER
   44   35 4210ED ADD IF30, IF10, R00          ;GAIN=4. 21/2**3
   45
   46              ;ZERO 5
   47   36 4810FF LDA MPL2, IF30, R00
   48   37 4218FD ADD MPL2, IF31, R00
   49   38 42185C ADD MPL2, IF31, R03
   50   39 4218FC ADD MPL2, IF31, R08
   51   40 42181D ADD MPL2, IF31, R09
   52   41 4010FD ADD MPL2, TEMP, R00          ;INPUT FILTER OUTPUT IN MPL2
```

Figure A.2. Complete Spectrum Analyzer Assembly Listing

```
LINE  LOC OBJECT SOURCE STATEMENT

  53
  54
  55                ;*****SWEEP OSC*****
  56
  57
  58    42 4C9A6F LDA S1,    KP5,  R12            ;DEFINE S1
  59    43 4C92DF LDA M,     KP4,  L01            ;DEFINE M
  60    44 4A40EB SUB F1,    S1,   R00
  61    45 4064EF LDA DAR,   F1,   R00
  62    46 7A48ED ADD F1,    M,    R00,  CNDS
  63    47 4ACAF5 LIM SWP,   KP7,  R00
  64    48 4060FB SUB SWP,   F1,   R00            ;INVERT SLOPE
  65    49 406CEF LDA DAR,   SWP,  R00            ;SWEEP TO DAR TO OUTPUT
  66    50 48CE8A SUB DAR,   KP5,  R05
  67    51 78C6CD ADD DAR,   KP4,  L01,  CNDS     ;10 MS DELAY FOR FILTER RISE TIMES
  68    52 44602E LDA F2,    F1,   R02,  NOP      ;SAWTOOTH SCALING
  69    53 4460AA SUB F2,    F1,   R06,  NOP
  70    54 46606B SUB F2,    F2,   R12,  NOP
  71    55 4460EA SUB F2,    F1,   R08,  NOP
  72    56 4000EF NOP
  73    57 4000EF NOP
  74    58 86CA3E LDA S2,    KP3,  R02,  OUTO     ;DEFINE S2
  75    59 86CABC ADD S2,    KP3,  R06,  OUTO
  76    60 84CA1D ADD S2,    KP1,  R09,  OUTO
  77    61 8668ED ADD F2,    S2,   R00,  OUTO     ;ADD OFFSET
  78
  79
  80                ;*****VCO*****
  81
  82
  83    62 8000EF OUTO
  84    63 8270EB SUB OSC1,  F2,   R00,  OUTO
  85    64 4864EF LDA DAR,   OSC1, R00
  86    65 7A58ED ADD OSC1,  M,    R00,  CNDS
  87    66 4870FF LDA OSC,   OSC1, R00
  88    67 4A581A SUB OSC,   M,    R01
  89    68 4878D7 ABS OSC,   OSC,  L01
  90    69 4A581A SUB OSC,   M,    R01
  91    70 4064EF LDA DAR,   F1,   R00
  92    71 70D2EF LDA OSC1,  KP0,  R00,  CNDS     ;SET VCO TO 0 TO SYNC WITH SWEEP
  93    72 4878DD ADD OSC,   OSC,  L01            ;VCO OUTPUT IN OSC
  94
  95
  96                ;******MULTIPLY*****
  97
  98
  99    73 4E70EB SUB MPL1,  MPL1, R00            ;CLEAR MULTIPLY OUTPUT REGISTER
 100    74 486CEF LDA DAR,   OSC,  R00            ;LOAD DAR WITH MULTIPLIER
 101    75 FD580C ADD MPL1,  MPL2, R01,  CND7
 102    76 ED582C ADD MPL1,  MPL2, R02,  CND6
 103    77 DD584C ADD MPL1,  MPL2, R03,  CND5
 104    78 CD586C ADD MPL1,  MPL2, R04,  CND4
 105    79 BD588C ADD MPL1,  MPL2, R05,  CND3
 106    80 AD58AC ADD MPL1,  MPL2, R06,  CND2
```

Figure A.2. Complete Spectrum Analyzer Assembly Listing (cont'd)

```
 LINE  LOC OBJECT SOURCE STATEMENT

  107   81 9D58CC ADD MPL1, MPL2, R07, CND1
  108   82 8D58EC ADD MPL1, MPL2, R08, CND0
  109   83 4818DB SUB MPL2, MPL2, L01          ;DEVELOP -Y
  110   84 7C58ED ADD MPL1, MPL2, R00, CNDS    ;ADD -Y IF MULTIPLIER IS NEGATIVE
  111
  112
  113            ;*****BAND-PASS FILTER*****
  114
  115
  116            ;POLE 1
  117   85 4A28EF LDA TEMP, BP11, R00, NOP
  118   86 44D0FF LDA BP11, BP10, R00, NOP
  119   87 4A298E LDA BP10, BP11, R05, NOP
  120   88 4A29EB SUB BP10, BP11, R00, NOP
  121   89 40814C ADD BP10, BP10, R03, NOP
  122   90 4081EA SUB BP10, BP10, R08, NOP
  123   91 A001EB SUB BP10, TEMP, R00, OUT2    ;OUTPUT VCO SINE WAVE
  124   92 A0018C ADD BP10, TEMP, R05, OUT2
  125   93 A001CA SUB BP10, TEMP, R07, OUT2
  126
  127            ;POLE 3
  128   94 A088EF LDA TEMP, BP31, R00, OUT2
  129   95 A281FF LDA BP31, BP30, R00, OUT2
  130   96 A401AE LDA BP30, TEMP, R06, OUT2
  131   97 4401EB SUB BP30, TEMP, R00
  132   98 42CCEF LDA DAR,  YO,   R00          ;LINEAR OUTPUT TO DAR
  133   99 4681CA SUB BP30, BP30, R07, NOP
  134  100 4489EB SUB BP30, BP31, R00, NOP
  135  101 44894A SUB BP30, BP31, R03, NOP
  136  102 44898A SUB BP30, BP31, R05, NOP
  137  103 4489CC ADD BP30, BP31, R07, NOP
  138
  139            ;POLE 5
  140  104 4880EF LDA TEMP, BP51, R00, NOP
  141  105 C899EF LDA BP51, BP50, R00, OUT4
  142  106 C0119E LDA BP50, TEMP, R05, OUT4
  143  107 C0113D ADD BP50, TEMP, R10, OUT4
  144  108 C011FB SUB BP50, TEMP, R00, OUT4
  145  109 C899FC ADD BP50, BP50, R08, OUT4
  146  110 C891FB SUB BP50, BP51, R00, OUT4
  147  111 48915A SUB BP50, BP51, R03
  148  112 4AC4EF LDA DAR,  LOUT, R00          ;LOG OUTPUT TO DAR
  149  113 4891BC ADD BP50, BP51, R06, NOP
  150  114 48911B SUB BP50, BP51, R09, NOP
  151
  152            ;STAGE PROPAGATION
  153  115 4A21AC ADD BP10, MPL1, R06, NOP
  154  116 44816C ADD BP30, BP10, R04, NOP
  155  117 42917C ADD BP50, BP30, R04, NOP
  156
  157
  158            ;*****LOW PASS FILTER*****
  159
  160
```

Figure A.2. Complete Spectrum Analyzer Assembly Listing (cont'd)

```
 LINE  LOC OBJECT SOURCE STATEMENT

  161  118 44B1FF LDA Y2,    Y1,    R00,  NOP
  162  119 E2C9EF LDA Y1,    Y0,    R00,  OUT6
  163  120 E4A19A SUB Y0,    Y1,    R05,  OUT6
  164  121 E4A11B SUB Y0,    Y1,    R09,  OUT6
  165  122 E4A13B SUB Y0,    Y1,    R10,  OUT6
  166  123 EE89FB SUB Y0,    Y2,    R00,  OUT6
  167  124 E4A1FD ADD Y0,    Y1,    R00,  OUT6
  168  125 4E899C ADD Y0,    Y2,    R05
  169  126 4E891D ADD Y0,    Y2,    R09
  170  127 4E897D ADD Y0,    Y2,    R12
  171  128 4E899D ADD Y0,    Y2,    R13
  172  129 4C8919 ABA Y0,    BP50,  R09          ;FULL WAVE RECTIFIER OPERATION
  173  130 4689FF LDA Y0,    Y0,    R00
  174
  175
  176              ;*****LOG AMP*****
  177
  178
  179  131 4689F7 ABS Y0,    Y0,    R00          ;PREVENT PROCESSING OF NEGATIVE NUMBERS
  180  132 4699AF LDA LOUT,  Y0,    L02          ;SECTION 6
  181  133 4699AD ADD LOUT,  Y0,    L02
  182  134 4699AD ADD LOUT,  Y0,    L02
  183  135 46990C ADD LOUT,  Y0,    R01
  184  136 46992C ADD LOUT,  Y0,    R02
  185  137 42CCEF LDA DAR,   Y0,    R00
  186  138 B799AF LDA LOUT,  Y0,    L02,  CND3   ;SECTION 5
  187  139 B793ED ADD LOUT,  KP2,   R00,  CND3
  188  140 BD9B8C ADD LOUT,  KP5,   R05,  CND3
  189  141 C799CF LDA LOUT,  Y0,    L01,  CND4   ;SECTION 4
  190  142 C79BED ADD LOUT,  KP3,   R00,  CND4
  191  143 C7936C ADD LOUT,  KP2,   R04,  CND4
  192  144 D799EF LDA LOUT,  Y0,    R00,  CND5   ;SECTION 3
  193  145 DD93ED ADD LOUT,  KP4,   R00,  CND5
  194  146 D7936C ADD LOUT,  KP2,   R04,  CND5
  195  147 E7990E LDA LOUT,  Y0,    R01,  CND6   ;SECTION 2
  196  148 ED9BED ADD LOUT,  KP5,   R00,  CND6
  197  149 E7936C ADD LOUT,  KP2,   R04,  CND6
  198  150 F7994E LDA LOUT,  Y0,    R03,  CND7   ;SECTION 1
  199  151 F7996C ADD LOUT,  Y0,    R04,  CND7
  200  152 F7998C ADD LOUT,  Y0,    R05,  CND7
  201  153 FF93ED ADD LOUT,  KP6,   R00,  CND7
  202  154 FD936C ADD LOUT,  KP4,   R04,  CND7
  203  155 4000EF NOP
  204  156 4000EF NOP
  205  157 4000EF NOP
  206  158 4000EF NOP
  207  159 4000EF NOP
  208  160 4000EF NOP
  209  161 4000EF NOP
  210  162 4000EF NOP
                •
                •
                •
  236  188 5000EF EOP
  237  189 4000EF NOP
  238  190 4000EF NOP
  239  191 4000EF NOP
  240              END
```

Figure A.2. Complete Spectrum Analyzer Assembly Listing (cont'd)

```
:18000000F3F0F6F6FEFBF3F0F0F0FEFFF3F0F0F0FEFFF3F0F0F0FEFFE0
:18001800F3F0F0F0FEFFF3F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFCE
:18003000F6F0F0F0FEFFFEFBFEF6FEFDF4F0F0F0FEFFF4F0F0F0FEFF8B
:18004800F7F1F0F0FEFFF4F0F0F0FEFFF4F0F0F8FEFFF6F3F0F0FFFF8A
:18006000F4F6F0F0F2FAF4F6F0F0FAFAF5F7F0F0F0FDF4F4F0F0F2FAA7
:18007800F4F4F0F0FAFCF4F5F0F0F0FDF4F4F0F0F2FDF4F4F0F0F6FB8C
:18009000F3F3F0F8FEFFF4FCF0F0FFFFF4F0F1F0F0FFF2F1F1F0F0FA5D
:1800A800F4F0F1F0F4FAF4F8F1F0F6FCF1F3F1F8F4FCF4F2F1F8F8FA50
:1800C000F4F2F1F8FCFCF0F3F1F8F2FDF4F4F2F2F4FCF4F2F1F0FEFD28
:1800D800F4F8F1F0FFFFF4F2F1F8FFFDF4F2F1F8F5FCF4F2F1F8FFFCF0
:1800F000F4F2F1F8F1FDF4F0F1F0FFFDF4FCF9FAF6FFF4FCF9F2FDFFCB
:18010800F4FAF4F0FEFBF4F0F6F4FEFFF7FAF4F8FEFDF4FAFCFAFFF599
:18012000F4F0F6F0FFFBF4F0F6FCFEFFF4F8FCFEF8FAF7F8FCF6FCFD7E
:18013800F4F4F6F0F2FEF4F4F6F0FAFAF4F6F6F0F6FBF4F4F6F0FEFAA8
:18015000F4F0F0F0FEFFF4F0F0F0FEFFF8F6FCFAF3FEF8F6FCFAFBFC65
:18016800F8F4FCFAF1FDF8F6F6F8FEFDF8F0F0F0FEFFF8F2F7F0FEFB49
:18018000F4F8F6F4FEFFF7FAF5F8FEFDF4F8F7F0FFFFF4FAF5F8F1FA24
:18019800F4F8F7F8FDF7F4FAF5F8F1FAF4F0F6F4FEFFF7F0FDF2FEFF1C
:1801B000F4F8F7F8FDFDF4FEF7F0FEFBF4F8F6FCFEFFFFFDF5F8F0FCE0
:1801C800FEFDF5F8F2FCFDFDF5F8F4FCFCFDF5F8F6FCFBFDF5F8F8FCC1
:1801E000FAFDF5F8FAFCF9FDF5F8FCFCF8FDF5F8FEFCF4F8F1F8FDFBA9
:1801F800F7FCF5F8FEFDF4FAF2F8FEFFF4F4FDF0FFFFF4FAF2F9F8FE9D
:18021000F4FAF2F9FEFBF4F0F8F1F4FCF4F0F8F1FEFAFAF0F0F1FEFBBE
:18022800FAF0F0F1F8FCFAF0F0F1FCFAFAF0F8F8FEFFFAF2F8F1FFFF94
:18024000FAF4F0F1FAFEF4F4F0F1FEFBF4F2FCFCFEFFF4F6F8F1FCFA79
:18025800F4F4F8F9FEFBF4F4F8F9F4FAF4F4F8F9F8FAF4F4F8F9FCFC59
:18027000F4F8F8F0FEFFFCF8F9F9FEFFFCF0F1F1F9FEFCF0F1F1F3FD3F
:18028800FCF0F1F1FFFBFCF8F9F9FFFCFCF8F9F1FFFBF4F8F9F1F5FA18
:1802A000F4FAFCF4FEFFF4F8F9F1FBFCF4F8F9F1F1FBF4FAF2F1FAFC15
:1802B800F4F4F8F1F6FCF4F2F9F1F7FCF4F4FBF1FFFFFEF2FCF9FEFFF4
:1802D000FEF4FAF1F9FAFEF4FAF1F1FBFEF4FAF1F3FBFEFEF8F9FFFBCB
:1802E800FEF4FAF1FFFDF4FEF8F9F9FCF4FEF8F9F1FDF4FEF8F9F7FDA5
:18030000F4FEF8F9F9FDF4FCF8F9F1F9F4F6F8F9FFFFF4F6F8F9FFF797
:18031800F4F6F9F9FAFFF4F6F9F9FAFDF4F6F9F9FAFDF4F6F9F9F0FC8A
:18033000F4F6F9F9F2FCF4F2FCFCFEFFFBF7F9F9FAFFFBF7F9F3FEFD5A
:18034800FBFDF9FBF8FCFCF7F9F9FCFFFCF7F9FBFEFDFCF7F9F3F6FC2A
:18036000FDF7F9F9FEFFFDFDF9F3FEFDFDF7F9F3F6FCFEF7F9F9F0FE1A
:18037800FEFDF9FBFEFDFEF7F9F3F6FCFFF7F9F9F4FEFFF7F9F9F6FCFC
:18039000FFF7F9F9F8FCFFFFF9F3FEFDFFFDF9F3F6FCF4F0F0F0FEFFF9
:1803A800F4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFF39
:1803C000F4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFF21
:1803D800F4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFF09
:1803F000F4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFF1
:18040800F4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFD8
:18042000F4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFC0
:18043800F4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFA8
:18045000F4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFF90
:18046800F5F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFFF4F0F0F0FEFF77
:00000001FF
```

Figure A.3. Spectrum Analyzer Object Code

intel®

U.S. AND CANADIAN SALES OFFICES

3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

January 1980

ALABAMA
Intel Corp.
3322 S. Parkway, Ste. 71
Holiday Office Center
Huntsville 35802
Tel: (205) 883-2430

Pen-Tech Associates, Inc.
Holiday Office Center
3322 Memorial Pkwy., S.W.
Huntsville 35801
Tel: (205) 881-9298

ARIZONA
Intel Corp.
10210 N. 25th Avenue, Suite 11
Phoenix 85021
Tel: (602) 997-9695

BFA
4426 North Saddle Bag Trail
Scottsdale 85251
Tel: (602) 994-5400

CALIFORNIA
Intel Corp.
7670 Opportunity Rd.
Suite 135
San Diego 92111
Tel: (714) 268-3563

Intel Corp.*
1651 East 4th Street
Suite 105
Santa Ana 92701
Tel: (714) 835-9642
TWX: 910-595-1114

Intel Corp.*
15335 Morrison
Suite 345
Sherman Oaks 91403
(213) 986-9510
TWX: 910-495-2045

Intel Corp.*
3375 Scott Blvd.
Santa Clara 95051
Tel: (408) 987-8086
TWX: 910-339-9279
TWX: 910-338-0255

Earle Associates, Inc.
4617 Ruffner Street
Suite 202
San Diego 92111
Tel: (714) 278-5441

Mac-I
2576 Shattuck Ave.
Suite 4B
Berkeley 94704
Tel: (415) 843-7625

Mac-I
P.O. Box 1420
Cupertino 95014
Tel: (408) 257-9880

Mac-I
11725 Espen Circle
P.O. Box 8763
Fountain Valley 92708
Tel: (714) 839-3341

Mac-I
110 Sutter Street
Suite 715
San Francisco 94104
Tel: (415) 982-3673

Mac-I
20121 Ventura Blvd., Suite 240E
Woodland Hills 91364
Tel: (213) 347-5900

COLORADO
Intel Corp.*
650 S. Cherry Street
Suite 720
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289

Westek Data Products, Inc.
25921 Fern Gulch
P.O. Box 1355
Evergreen 80439
Tel: (303) 674-5255

Westek Data Products, Inc.
1322 Arapahoe
Boulder 80302
Tel: (303) 449-2620

Westek Data Products, Inc.
1228 W. Hinsdale Dr.
Littleton 80120
Tel: (303) 797-0482

CONNECTICUT
Intel Corp.
Peacock Alley
1 Padanaram Road, Suite 146
Danbury 06810
Tel: (203) 792-8366
TWX: 710-456-1199

FLORIDA
Intel Corp.
1001 N.W. 62nd Street, Suite 406
Ft. Lauderdale 33309
Tel: (305) 771-0600
TWX: 510-956-9407

FLORIDA (cont.)
Intel Corp.
5151 Adanson Street, Suite 203
Orlando 32804
Tel: (305) 628-2393
TWX: 810-853-9219

Pen-Tech Associates, Inc.
201 S.E. 15th Terrace, Suite K
Deerfield Beach 33441
Tel: (305) 421-4989

Pen-Tech Associates, Inc.
111 So. Maitland Ave., Suite 202
P.O. Box 1475
Maitland 32751
Tel: (305) 645-3444

GEORGIA
Pen Tech Associates, Inc.
Cherokee Center, Suite 21
627 Cherokee Street
Marietta 30060
Tel: (404) 424-1931

ILLINOIS
Intel Corp.*
2250 Golf Road, Suite 815
Rolling Meadows 60008
Tel: (312) 981-7200
TWX: 910-651-5881

First Rep Company
9400-9420 W. Foster Avenue
Chicago 60656
Tel: (312) 992-0830
TWX: 910-227-4927

Technical Representatives
1502 North Lee Street
Bloomington 61701
Tel: (309) 829-8080

INDIANA
Intel Corp.
2212 Maplecrest Rd.
Ft. Wayne 46815
Tel: (219) 493-2571

IOWA
Technical Representatives, Inc.
St. Andrews Building
1930 St. Andrews Drive N.E.
Cedar Rapids 52405
Tel: (319) 393-5510

KANSAS
Intel Corp.
9393 W. 110th St., Ste. 265
Overland Park 66210
Tel: (913) 642-8080

Technical Representatives, Inc.
8245 Nieman Road, Suite #100
Lenexa 66214
Tel: (913) 888-0212, 3, & 4
TWX: 910-749-6412

KENTUCKY
Lowry & Associates Inc.
P.O. Box 1827
Lexington 40593
Tel: (606) 273-3771

MARYLAND
Intel Corp.*
7257 Parkway Drive
Hanover 21076
Tel: (301) 796-7500
TWX: 710-862-1944

Glen White Associates
57 W. Timonium Road, Suite 307
Timonium 21093
Tel: (301) 252-6360

Mesa Inc.
11900 Parklawn Drive
Rockville 20852
Tel: Wash. (301) 881-8430
Balto. (301) 792-0021

MASSACHUSETTS
Intel Corp.*
27 Industrial Ave.
Chelmsford 01824
Tel: (617) 667-8126
TWX: 710-343-6333

EMC Corp.
381 Elliot Street
Newton 02164
Tel: (617) 244-4740

MICHIGAN
Intel Corp.*
26500 Northwestern Hwy.
Suite 401
Southfield 48075
Tel: (313) 353-0920
TWX: 910-420-1212
TELEX: 2 31143

Lowry & Associates, Inc.
135 W. North Street
Suite 4
Brighton 48116
Tel: (313) 227-7067

Lowry & Associates, Inc.
3902 Costa NE
Grand Rapids 49505
Tel: (616) 363-9839

MINNESOTA
Intel Corp.
7401 Metro Blvd.
Suite 355
Edina 55435
Tel: (612) 835-6722
TWX: 910-576-2867

Dytek North
1821 University Ave.
Room 163N
St. Paul 55104
Tel: (612) 645-5816

MISSOURI
Technical Representatives, Inc.
320 Brookes Drive, Suite 104
Hazelwood 63042
Tel: (314) 731-5200
TWX: 910-762-0618

NEW JERSEY
Intel Corp.*
Raritan Plaza
2nd Floor
Raritan Center
Edison 08817
Tel: (201) 225-3000
TWX: 710-480-6238

NEW MEXICO
BFA Corporation
P.O. Box 1237
Las Cruces 88001
Tel: (505) 523-0601
TWX: 910-983-0543

BFA Corporation
3705 Westerfield, N.E.
Albuquerque 87111
Tel: (505) 292-1212
TWX: 910-989-1157

NEW YORK
Intel Corp.*
350 Vanderbilt Motor Pkwy.
Suite 402
Hauppauge 11787
Tel: (516) 231-3300
TWX: 510-227-6236

Intel Corp.
80 Washington St.
Poughkeepsie 12601
Tel: (914) 473-2303
TWX: 510-248-0060

Intel Corp.*
2255 Lyell Avenue
Lower Floor East Suite
Rochester 14606
Tel: (716) 254-6120
TWX: 510-253-7391

Measurement Technology, Inc.
159 Northern Boulevard
Great Neck 11021
Tel: (516) 482-3500

T-Squared
4054 Newcourt Avenue
Syracuse 13206
Tel: (315) 463-8592
TWX: 710-541-0554

T-Squared
2 E. Main
Victor 14564
Tel: (716) 924-9101
TWX: 510-254-8542

NORTH CAROLINA
Intel Corp.
154 Huffman Mill Rd.

Tel: (919) 584-3631

Pen-Tech Associates, Inc.
1202 Eastchester Dr.
Highpoint 27260
Tel: (919) 883-9125

Glen White Associates
4009 Barrett Dr.
Raleigh 27609
Tel: (919) 787-7016

OHIO
Intel Corp.*
6500 Poe Avenue
Dayton 45415
Tel: (513) 890-5350
TWX: 810-450-2528

Intel Corp.*
Chagrin-Brainard Bldg. #210
28001 Chagrin Blvd.
Cleveland 44122
Tel: (216) 464-2736

Lowry & Associates Inc.
1440 Snow Road
Suite 216
Cleveland 44134
Tel: (216) 398-0506

Lowry & Associates, Inc.
1524 Marsetta Drive
Dayton 45432
Tel: (513) 429-9040

Lowry & Associates, Inc.
2735 Cleveland Ave.
Columbus 43224
Tel: (614) 436-2051

OREGON
Intel Corp.
10700 S.W. Beaverton
Hillsdale Highway
Suite 324
Beaverton 97005
Tel: (503) 641-8086

PENNSYLVANIA
Intel Corp.*
275 Commerce Dr.
200 Office Center
Suite 300
Fort Washington 19034
Tel: (215) 542-9444
TWX: 510-661-2077

Lowry & Associates, Inc.
Seven Parkway Center
Suite 455
Pittsburgh 15520
Tel: (412) 922-5110

Q.E.D. Electronics
300 N. York Road
Hatboro 19040
Tel: (215) 674-9600

TEXAS
Intel Corp.*
2925 L.B.J. Freeway
Suite 175
Dallas 75234
Tel: (214) 241-9521
TWX: 910-860-5487

Intel Corp.*
6420 Richmond Ave.
Suite 280
Houston 77057
Tel: (713) 784-3400

Industrial Digital Systems Corp.
5925 Sovereign
Suite 101
Houston 77036
Tel: (713) 988-9421

Intel Corp.
313 E. Anderson Lane
Suite 314
Austin 78752
Tel: (512) 454-3628

VIRGINIA
Glen White Associates
Route 2, Box 193
Charlottesville 22901
Tel: (804) 295-7686

Glen White Associates
P.O. Box 10186
Lynchburg 24506
Tel: (804) 384-6920

Glen White Associates
Rt. #1, Box 322
Colonial Beach 22442
Tel: (804) 224-7764

WASHINGTON
Intel Corp.
Suite 114 Bldg. 3
1603 116th Ave. N.E.
Bellevue 98005
Tel: (206) 453-8086

WISCONSIN
Intel Corp.
150 S. Sunnyslope Rd.
Brookfield 53005
Tel: (414) 784-9060

First Rep Company
9401 W. Beloit Rd.
Suite 304
Milwaukee 53227
Tel: (414) 546-2033

CANADA
Intel Semiconductor Corp.*
Suite 233, Bell Mews
39 Highway 7, Bells Corners
Ottawa, Ontario K2H 8R2
Tel: (613) 829-9714
TELEX: 053-4115

Intel Semiconductor Corp.
50 Galaxy Blvd.
Unit 12
Rexdale, Ontario
M9W 4Y5
Tel:(416) 675-2105
TELEX: 06983574

Multilek, Inc.*
15 Grenfell Crescent
Ottawa, Ontario K2G 0G3
Tel: (613) 226-2365
TELEX: 053-4585

Multilek, Inc.
Toronto
Tel: (416) 245-4622

Multilek, Inc.
Montreal
Tel: (514) 481-1350

*Field application location

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

Printed in U.S.A.

Printed in USA/K83/0380/15K/FL