FÁBIO MONTORO
DESENVOLVIMENTO

discutir em sala: LP/HP? conclusão?

Program a spectrum analyzer on a one-chip real-time signal processor

Program it properly, and the single-chip Intel-2920 real-time signal processor becomes a spectrum analyzer capable of covering a 200-Hz to 3.2-kHz range (Table 1). The signal processor contains all the circuitry needed to implement an input low-pass filter, a bandpass filter (BPF), a reconstruction output filter, an input a/d and corresponding output d/a converter, a mixer and full-wave rectifier.

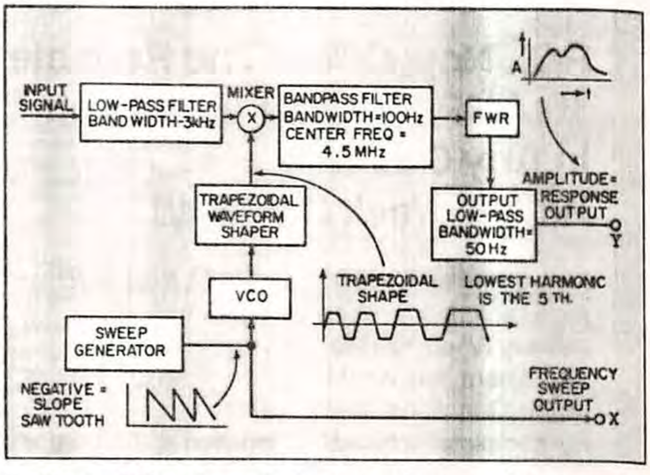
Although a scanning spectrum analyzer could be implemented directly with a single, electronically tunable, narrowband filter, it is impractical to design such a filter (whether analog or digital) to cover a 10-to-1 frequency range, including near dc. The spectrum analyzer digitally implemented through the Intel 2920 employs the easier method of sweeping the input signal past a fixed-tuned narrow bandpass filter (Fig. 1). It uses 107 program instructions and 22 RAM locations (Table 2). (The basic characteristics of the 2920 and a review of the sampling theory involved in digital processing of analog signals appeared in ELECTRONIC DESIGN, Sept. 27, 1979, p. 50.)

A dc control voltage to the 2920 establish the sweep rate of a sweeping local oscillator (SLO). Mixed with the analog input signal, the SLO generates upper and lower sidebands centered about its own frequency (Fig. 2). To cover the specified spectrum range (200-Hz to 3.2-kHz band), the SLO sweeps from 1.3 to 4.3 kHz.

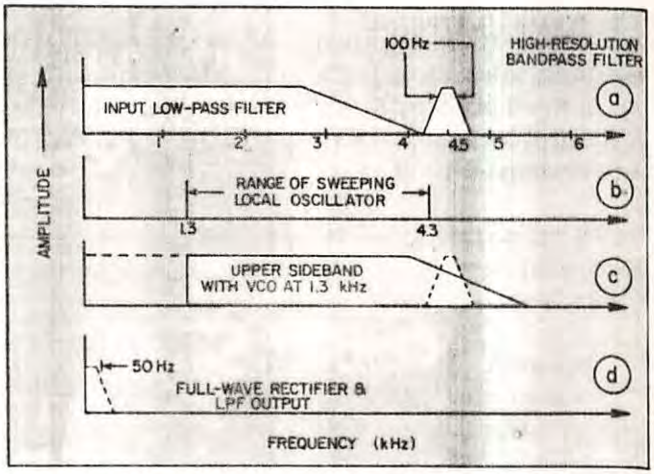
After the SLO and signal are mixed, only the upper sideband is of interest. When the SLO is at 1.3 kHz, the BPF (which is tuned to 4.5 kHz) looks at the high end of the input range (3.2 + 1.3 = 4.5 kHz). As the SLO frequency increases, the input signal frequency "seen" by the BPF descreases until at 4.3 kHz, the BPF "sees" the signal energy at 200 Hz. After the sideband is applied to the bandpass filter, a full-wave rectifier and low-pass filter (50-Hz wide, centered at dc) extract the envelope from the BPF output.

Outputs of the 2920 include the frequency-sweep voltage, which can drive the horizontal-axis (X) of a scope, and the low-pass filter output, which can drive the vertical axis (Y) of the scope. A signal from the input low-pass filter and another from the SLO are used primarily for demonstration.

Robert E. Holm, Product Marketing Manager, Telecommunications, Intel Corp., 3065 Bowers Ave., Santa Clara, CA 95051.



1. **Most of the elements in a spectrum analyzer** can be implemented with the Intel 2920 signal processor.



2. **The high-resolution bandpass filter and baseband low-pass filter** (a) can be implemented with the 2920. After the input signal is mixed with a local oscillator that linearly sweeps from 1.3 to 4.3 kHz (b), the upper sideband (c) is applied to the bandpass filter. A full-wave rectifier and low-pass output filter, also implemented with the 2920 (d), then extract the desired output information.

Table 1. Target specifications of the spectrum analyzer

Input bandwidth	3 kHz
Resolution bandwidth	100 Hz
Sweep rate	5 kHz/s, or 0.6 s/band
Dynamic range	48 dB
Inputs	Analog signal $-1\text{ V} \leq \text{SIG} \leq 1\text{ V}$
Outputs	Frequency-response amplitude (vertical axis) Sawtooth sweep waveform (horizontal axis)

Determined by the sampling frequency (in this case about 13 kHz), the first-order aliasing spectrum (Fig. 3a) establishes the limitations and requirements for the subsequent rolloffs, bandwidths and center frequencies of the system's various filters.

Aliasing sets the limits

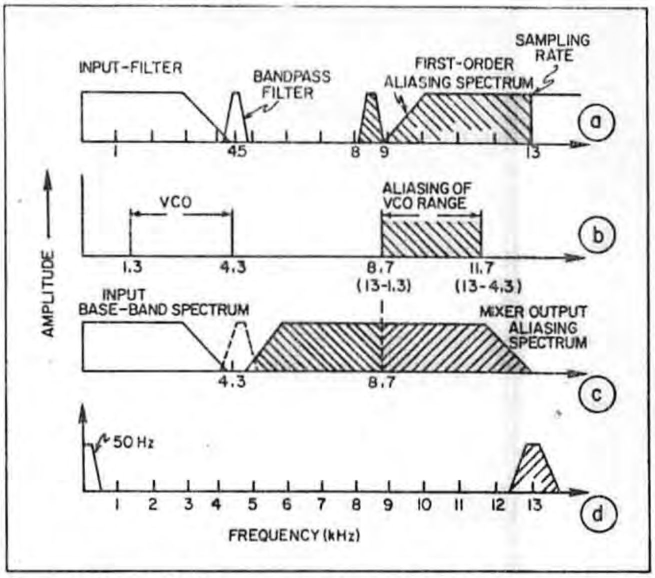
For example, the center frequency of the bandpass filter is set between the spectrum of the input filter and the lower sideband of the aliased spectrum, when the SLO is at 4.3 kHz (Fig. 3b). Consequently, the bandpass filter must have enough attenuation to eliminate at its pass edges any components of both baseband and aliased signals. A three-pole Bessel filter can do the job, if the input low-pass filter does its work properly. The small overshoot of the Bessel makes it a good choice for the bandpass unit.

The input low-pass filter, however, is less demanding. It establishes the baseband spectrum and also the aliased lower sideband of the SLO. The rolloff provided by a simple four-pole, two-zero filter is adequate to keep spurious signals out of the passband of the bandpass filter.

An anti-aliasing input filter is needed only when the input signal has significant frequency components above roughly 7 kHz. Controlled signals, such as sine waves or other narrow-band signals, do not need anti-aliasing filtering.

If needed, the anti-aliasing filter must ensure that unwanted spectrum components are at least 50-dB down (to conform with an input dynamic range of 48 dB) before they center the passband of the input low-pass filter. With a 13-kHz sampling frequency (corresponding to a full 192-instruction program and a 10-MHz clock), the aliasing components are already more than 50-dB down at 3.2 kHz, which is 9.8 kHz from the sampling frequency. Therefore, the anti-aliasing filter needs relatively little rolloff by 3.2 kHz (about 1 dB), but needs the full 50 dB at 9.8 kHz. This level of attenuation requires an advanced six-pole Butterworth or a five-pole, 0.5-dB-ripple Chebyshev filter.

The output low-pass filter is the least demanding of all the filters. It eliminates the harmonic output of the full-wave rectifier (and corresponding aliased

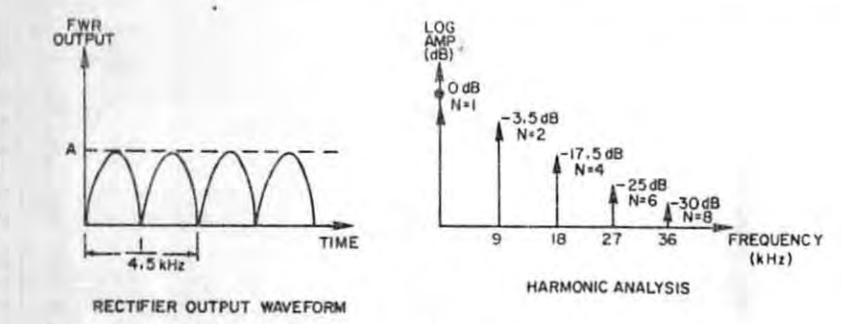


3. **Aliasing with the sampling frequency** (a) sets the limits for placing the center frequencies, and the bandpass and rolloff requirements of the system's filters. The first-order aliased VCO extends from 8.7 to 11.7 kHz (b) with a 13-kHz sampling frequency. When mixed with the aliased input signal, the resulting aliasing spectrum forces the bandpass filter to locate between the aliased spectrum of the mixer-output and the baseband spectrum (c). The unwanted aliased output signals, if any, center around 13 kHz, well outside the range of the output low-pass filter (d).

Table 2. Spectrum analyzer instruction summary

Subsystem	No. Instructions
Input low-pass filter	29
Multiplier (mixer)	13
Sweep generator	8
VCO (SLO & wave shaping)	8
Bandpass filter	33
Output rectifier & filter	16
Total Instruction	107
No. RAM locations	22

Note: With a 10-MHz clock, an instruction cycle takes 400 ns; therefore, the sample rate is $1/(400 \times 10^{-9}) \times$ (number of instructions). If the full complement of the 2920's 192 instructions are used, the sampling rate is 13.02 kHz. Shorter programs yield a higher sample rate, with a maximum of 75 kHz. Bandwidth is approximately 1/3 the sampling rate.



(a)

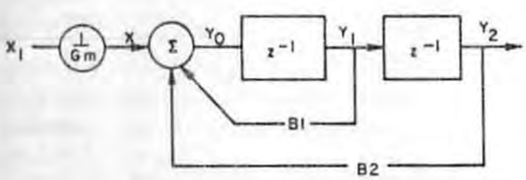
$$G(S) = \frac{1}{S^2 + 1.41S + 1}$$

$$|G(\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{B_3}\right)^4}$$

(b)

$$G(Z) = \frac{1}{1 - B_1 Z^{-1} - B_2 Z^{-2}}$$

(c)



$B_1 = 2e^{-0.707B_3T}(\cos B_3T)$

$B_2 = -e^{-1.414B_3T}$

B_3 = 3-dB bandwidth in radians/s

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

$$= [(1+B_2)\sin B_3T]^{-1}$$

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

$B_3 = 2\pi \times 50 = 314.16$ rad/s

$B_3T = 0.02413$

$B_1 = 1.9663$

$= 1.1111011101010 = 2^1 - 2^{-5} - 2^{-9} - 2^{-11}$

$B_2 = -0.9529$

$= -0.11110011111101 = 2^0 - 2^{-5} - 2^{-6} - 2^{-13}$

$G_{max} = 192.2$

$1/G_{max} = 0.0052 = 0.000000010101$

$= 2^{-8} + 2^{-10} + 2^{-12}$

(d)

4. **An analysis of the output-signal spectrum** (a) provides the basis for selecting a two-pole Butterworth filter with a transfer function (b), which is converted to the Z domain (c). The Butterworth can then be implemented with the 2920 by employing the calculated constants and configuration (d).

components). Its passband is at least half that of the bandpass filter; however, it should be as simple as possible to keep the amplitude and phase distortion low. Because of its relative simplicity, the output lowpass filter exemplifies the method of implementing a filter digitally in the 2920.

The full-wave rectifier's spectral output (Fig. 4a) shows the desired signal information located between dc and 50 Hz; all other signal components should be removed. A two-pole Butterworth filter can provide the filtering.

Implementing a digital filter

An approximation to a Butterworth filter can be implemented digitally with the 2920 by converting the filter's transfer function from analog to digital; or rather, from the S-plane to the Z-plane.[1]

The real and imaginary parts of Z can be calculated from S (Fig. 4b), where $Z = e^{-\sigma T} + e^{-j\omega T}$ and $S = (\sigma + j\omega)T$. The function, G(Z), can then be realized digitally as a two-stage recursive transversal filter, as shown in Fig. 4c. Input signal values must be normalized by a gain, $1/G_{max}$, to prevent overflow in the filter calculations.

The 2920 uses RAM locations to implement the Z delays. In the transfer of data from one location to another, the sample period, T, is the delay. Fig. 4c shows the result of multiplying the delayed values (tapped from appropriate memory locations) by the appropriate coefficients (B_1 and B_2), with an efficient shift-and-add software algorithm. Fig. 4d contains the calculations leading to both the filter coefficients and the gain-weighting factor; Table 3 gives the 2920 assembly-language program.

In addition to filters, the spectrum analyzer needs a sweep-rate generator and a mixer. Both are also implemented in the 2920.

A sawtooth wave starts it all

The sweep-rate generator starts with a sawtooth wave that drives a voltage-controlled oscillator (VCO), which sweeps linearly between the desired predetermined frequencies (1.3 and 4.3 kHz) at a rate determined by the period of the sawtooth wave. The sawtooth's slope determines the rate at which the frequency of the VCO changes; the voltage excursion determines the VCO's frequency range; and the offset represents the VCO's minimum frequency.

The sawtooth wave has a negative slope generated simply by continuously decrementing a register from a fixed constant. When the voltage crosses zero, a constant equal to the sawtooth peak amplitude is loaded with an LDA instruction. A constant offset to the waveform provides a minimum voltage that corresponds to the minimum frequency of the VCO (Fig. 5). The frequency of the sawtooth signal is low relative to the sampling frequency; therefore, any aliasing effects it generates are negligible.

The VCO, like the sawtooth, can be implemented with a decrementing register, except that the decrementing value is not constant; rather, it is determined from a scaled version of the sawtooth input waveform. The method for calculating both ends of the VCO frequency range is similar to that for calculating the sawtooth; the sweep-rate generator's offset determines the low frequency, and the scaling factor determines the high frequency. The result is a period for the VCO sawtooth wave that varies as a function of the sweep-generator sawtooth.

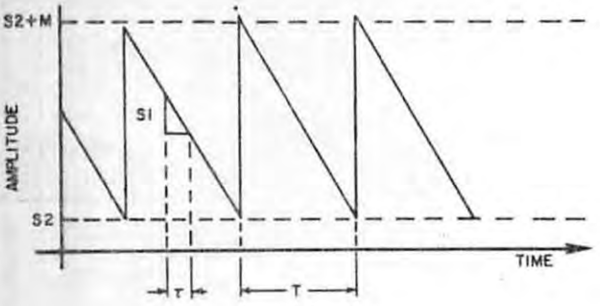
Unfortunately, such a variable-frequency sawtooth signal (1.3 to 4.3 kHz) has high harmonic content, which can distort the input signal to the mixer. Digital filters cannot be used to remove these harmonics, because they are susceptible to the aliasing components. The VCO output could be filtered with an external analog filter, but this would involve additional hardware and many extra instructions for I/O and a/d conversion.

Table 3. Program to implement the output low-pass filter

OP	DEST	SOURCE	SHF	CND	COMMENTS
ADD	X	X1	R08	—	Adjust X_1 by gain $1/G_m = 2^{-8} + 2^{-10} + 2^{-12}$
ADD	X	X1	R10	—	
ADD	X	X1	R12	—	
LDA	Y2	Y1	R00	—	Propagate samples through delay Line
LDA	Y1	Y0	R00	—	
LDA	Y0	X	R00	—	Replaces Y_0 with X
SUB	Y0	Y1	R05	—	Feedback B_1Y_1 TO Y_0 where $B_1=2-2^{-5}-2^{-9}-2^{-11}$
SUB	Y0	Y1	R08	—	
ADD	Y0	Y1	R10	—	
ADD	Y0	Y1	R12	—	
ADD	Y0	Y1	L02	—	
SUB	Y0	Y2	R00	—	Feedback B_2Y_2 TO Y_0 where $B_2= 1-2^{-5}-2^{-6}-2^{-13}$
ADD	Y0	Y2	R05	—	
ADD	Y0	Y2	R06	—	
ADD	Y0	Y2	R13	—	
LDA	DAR	Y2	R00	—	Load filter output to DAR for outputting from 2920

Table 4. Algorithm for multiplication

OP	DEST	SOURCE	SHF	CND	Comments
LDA	DAR	X	R00	—	Set up DAR for conditional adds, X is multiplier
SUB	Z	Z	R00		clear product register 2
ADD	Z	Y	R01	CND 7	
ADD	Z	Y	R02	CND 6	
ADD	Z	Y	R03	CND 5	Multiply Y by the magnitude of X where
ADD	Z	Y	R04	CND 4	Z = X (−t+y)
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 ← 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 by calculating −Y= Y−2Y



If:

$$\tau = 76.8\ \mu s/\text{step (13,020-Hz clock)}$$

$$M = 1\ V$$

$$T = 1\ s,$$

then:

$$S1 = M/(T/\tau) = 7.68 \times 10^{-5}\ V/\text{step}$$
$$= (KP5 + KP1 \times 2^{-5}) \times 2^{-13}$$

Program for generating a sawtooth wave

OP	DEST	SOURCE	SHF	COND	Comments
SUB	H1	S1	R00	—	Decrements H1 register
LDA	DAR	R1	R00	—	Loads DAR for conditional
LDA	H1	KP4	L01	CNDS	Adds 1.0 when H1<0
LDA	H2	H1	R01	—	Sets up new register H2 equal to 1/2 H1
ADD	H2	S2	R01	—	Adds offset S2 to H2

Program for generating the constant in S1 from standard binary constants

OP	DEST	SOURCE	SHF	COMMENTS
LDA	S1	KP5	R00	Loads 0.101 into register S1
ADD	S1	KP1	R05	Shifts 0.001 right five places, adds to S1
LDA	S1	S1	R13	Shifts S1 right 13 places

$$\begin{array}{ll} 0.101\ldots\ldots\ldots & = KP5 \\ +\ 0.00000001 & = KP1 \times 2^{-5} \\ \hline (0.10100001) & \times\ 2^{-13} = 7.68 \times 10^{-5} \end{array}$$

Standard 2920 binary constants

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

Notes: KP-positive constants; KM-negative constants, or the 2's complements of KP.

5. **The sweeping local oscillator** starts with a register programmed to decrement linearly, which produces a negatively sloping sawtooth wave. The constants for each decrement (S1) and for the offset (S2) are generated by adding or subtracting standard 2920 internal constants. Subtraction is performed by adding two's complements.

Converting the VCO sawtooth to a trapezoidal waveform, however, eliminates all even harmonics. Moreover, the first odd harmonic that remains is the fifth. To get this result, the top of the trapezoid is programmed to be 2/3 of the peak of a corresponding triangle wave.

The resulting SLO output then feeds into the system's mixer, which also must be programmed.

A mixer multiplies

The mixer multiplies the filtered and sampled input signal with the SLO waveform. Mixing must be implemented as a four-quadrant multiplier, since both its input waveforms can have positive or negative values. A general-purpose microprocessor might use a shift-and-add algorithm to determine the magnitude of a product and use a separate logic to determine the sign. However, the 2920 employs a more direct algorithm to avoid dealing with the sign bit separately. Bit manipulation is comparatively inefficient.

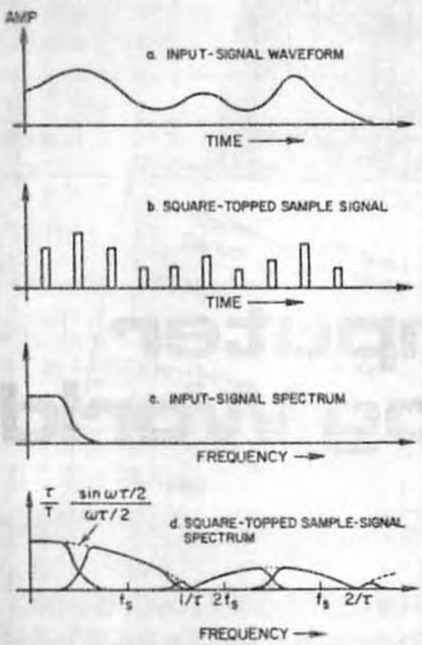
With X as the multiplier number, which includes both sign and magnitude, and with Y as the multiplicand, the algorithm employed follows the ADD-SUB routine listed in Table 4. Negative numbers in the 2920 are represented in two's complement notation, because it is hardware efficient (see Fig. 5).■■

Reference

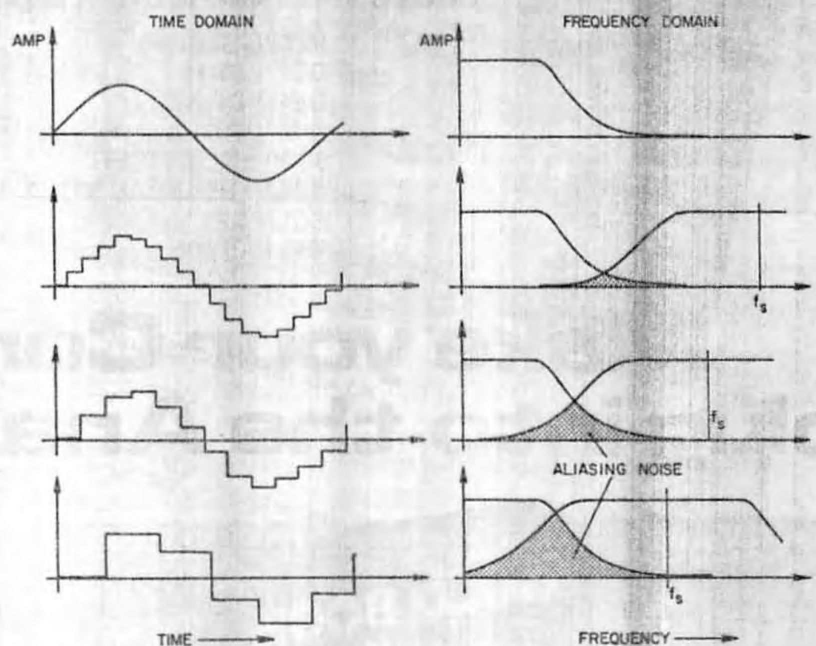
1. Karwoski, R.J., "Implement Digital Filters Efficiently," ELECTRONIC DESIGN, Sept. 1, 1979, p. 110.

Looking for some drawings?

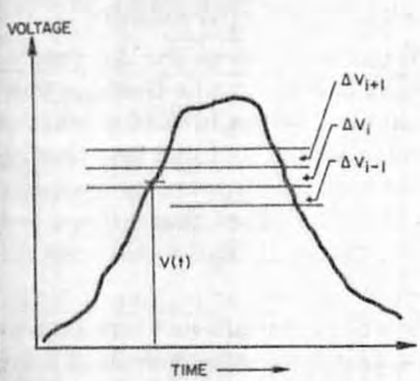
In our excitement in rushing the first details of the 2920 to print (ELECTRONIC DESIGN, Sept. 27, 1979, p. 50), we left behind a number of key figures. We apologize for the confusion and thank the throng of loyal readers who were kind enough to call our attention to the missing figures, which are reproduced below.



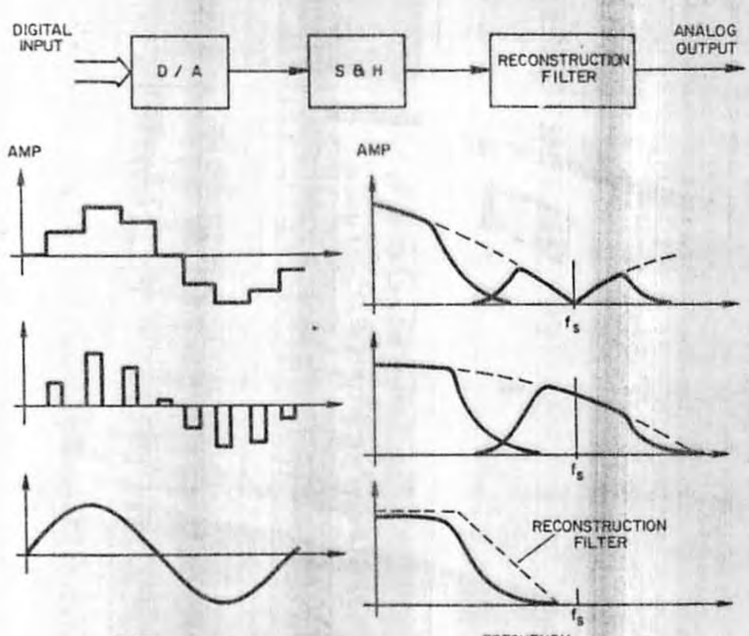
2. Analysis of sampled signal



3. How sampling rate affects aliasing noise



4. Signal quantizing intervals



5. Reconstruction of the analog signal