

# A Wide-Band, High-Resolution Spectrum Analyzer

M. P. Quirk, H. C. Wilck, and M. J. Grimm  
Communications Systems Research Section

*This article describes a million-channel, 20 MHz-bandwidth, digital spectrum analyzer under development at JPL for use in the SETI Sky Survey and other applications in the Deep Space Network. The analyzer digitizes an analog input, performs a  $2^{20}$ -point, Radix-2, Fast Fourier Transform, accumulates the output power, and normalizes the output to remove frequency-dependent gain. The effective speed of the real-time hardware is 2.2 GigaFLOPS.*

## I. System Overview

### A. Spectrum Analyzer Description

This article describes a wide-band, high-resolution, digital spectrum analyzer design for use in the Search for Extraterrestrial Intelligence (SETI), spacecraft telemetry, radio astronomy, planetary radar, Radio Frequency Interference (RFI) monitoring, and frequency analysis of wide-band systems in the Jet Propulsion Laboratory's Deep Space Network. The primary radio astronomy user will be the Search for Extraterrestrial Intelligence (SETI). The analyzer samples an input signal with up to a 20 MHz IF bandwidth, performs contiguous real-time Fast Fourier Transforms (FFTs) of up to  $1,048,576 = 2^{20}$  points corresponding to a resolution of 20 Hz. Special-purpose digital hardware provides real-time processing of the output spectra, including power accumulation, baseline rectification, matched filtering, and thresholding.

The output spectra can be recorded, displayed, and sent to a processing computer over a high-speed data bus. The system is microprocessor controlled, software reconfigurable, and includes built-in module and system-level diagnostics. The unit will be one seven-foot rack for the high-speed digital hard-

ware plus a half rack for the analog components, and a workstation computer for system control, spectrum displays, and post-processing.

The system contains two RF input channels, enabling dual-polarization analysis or the simultaneous analysis of two analog input channels.

The various components of the system are described below. Table 1 lists all the modules in the spectrum analyzer system.

### B. System Block Diagram

Figure 1 is a functional block diagram of the spectrum analyzer system. The Intermediate Frequency (IF) signals come into the IF Input Signal Conditioner where they are mixed to baseband and sampled at the desired sampling frequency. The resultant digital samples are complex numbers with a total of 16 bits, 8 bits real and 8 bits imaginary.

The Input and Timing module collects sufficient data for a spectrum in one of the input buffers, implements the delay for the first stage of an  $N$ -point FFT, then sends the delay pairs to the window module. The Input and Timing module also gener-

ates all the synchronization signals for the spectrum analyzer system. The Window module applies a real, symmetric window to the data to broaden the main lobe of the frequency response and suppress the sidelobes. An FFT is then performed on the windowed data. Each FFT stage performs an FFT butterfly operation and then implements the delay for the next stage. It is also possible to transform two channels simultaneously, by putting one real channel into the real components and another real channel into the imaginary components of the complex FFT. The two channels are sorted out in the Real-Adjust board of the FFT processor. An unscrambler module bit-reverses the frequency bin indices of the FFT output so that the output data are in natural order.

The final FFT data then goes to the power accumulator which calculates the power of the complex spectrum and samples and accumulates them on a frequency-by-frequency basis. There are actually two independent accumulators so that two separate data streams can be accumulated. The accumulated power spectra are then baselined to remove frequency-dependent system gain. The baseline is derived from an exponentially updated accumulation for each frequency. Again two baseline accumulation memories are needed for the separate accumulations.

The final processing involves combining successive spectra using a five-coefficient filter which operates on a frequency-by-frequency basis. This spectrum filter can be a matched filter to any desired single-frequency time signature of duration equal to five accumulation intervals. For example, as a source transits across an antenna beam, its amplitude traces the shape of the antenna beam response as shown in Fig. 2. If the coefficients of the spectrum filter are chosen to match the beam shape, the filter acts as a matched filter for transit sources. The output of the spectrum which passes a threshold test is then sent to the Spectrum Processor.

An output processor is used to select data for display on the system graphics display. The data can be from the spectrum filter or from the baseline module. The output processor can be used for test output by bypassing all the modules between it and the module under test. The spectrum analyzer system is controlled by the microprocessor computer system whose primary function is data processing and graphics handling.

### C. Configuration, Control and Post Processing

The system controller and post processor will be a microprocessor-based computer system with high-resolution graphics, and sufficient storage and processing power to control and configure the spectrum analyzer, and perform additional data processing on the output spectra. The computer system will generate graphics displays, archive data, provide antenna monitoring and control, and perform special-purpose processing.

Examples of special-purpose processing include signal identification, signal tracking, and the removal of undesired interfering signals.

The spectrum analyzer can be configured by a few dozen control bytes in the memory address space of the control microprocessor. Configuration parameters which can be changed include the sampling rate, whether the input is dual-channel or single-channel, real or complex, the length of the FFT, accumulation frame lengths, the baseline decay factor, the length of the spectrum filter, and the processing of the output data. The RAM coefficient memories for the window, matched filter, or initial baseline can be read in from the control microprocessor. Any module can be bypassed during operation or testing.

Each module of the analyzer is designed to support self-tests and diagnostics. Data vectors can be read into or out of any module for testing. The test output vectors are compared to the expected response to check the performance of the board. Since any module can be bypassed as the controller chooses, each module can be tested in isolation or with any combination of other modules. The executive board is controlled to generate test synchronization signals, and transmit and receive test vectors through the executive board buffers. The test software will reside in the control computer.

## D. Hardware Description

The hardware design of the megachannel analyzer is based on design experience with previous instruments of a similar nature. The 65,536-channel spectrum analyzer (Refs. 2, 3) is used for radio spectroscopy, Radio Frequency Interference (RFI) surveillance, to aid the DSN in trouble-shooting, and for the SETI All-Sky Survey (Refs. 4, 5). Recently a digital convolver was developed to process Synthetic Aperture Radar (SAR) data. The SAR processor contains two forward FFTs and two inverse FFTs, each 16,384 points long.

The new megachannel analyzer prototype will use TTL logic on large 16-inch by 17-inch wirewrap boards with eight boards in a cage. The entire system will be housed in a single 7-foot rack. Extensive use is made of the TRW TDC1022 22-bit floating-point adders and AMD MPY016-type 16-bit multipliers. The machine processes 20 complex megasamples per second and performs more than 2.2 GigaFLOPS.

## II. Module Descriptions

Table 1 lists all the modules in the spectrum analyzer system. Except for the Output Buffer and Interface Board, all the modules are described below.

## A. Intermediate Frequency Input Signal Conditioner

The IF Input Signal Conditioner filters and samples the analog input data to provide digital samples for the spectrum analyzer. Figure 3 shows the block diagram for one of the two IF sections. The input is bandpass filtered to a 20 MHz bandwidth and 55 MHz center frequency. An attenuator is used to adjust the IF level. This signal is downconverted to baseband by the complex mixer. The 3 dB points of the inphase and quadrature low-pass filters are at 10 MHz. These filters are matched in gain and phase to provide image rejection of greater than 30 dB. The complex baseband signal is sampled at 20 MHz by the 8-bit Analog-to-Digital Converters (ADCs). Synthesizers generate the sampling frequency and center frequency reference tones.

Two IF Channels can be processed simultaneously by the spectrum analyzer, but the maximum bandwidth is 10 MHz, half the maximum for a single input. Each signal is applied to the input of a complex mixer and the ADCs sample one of the arms of each mixer. Thus one of the real sampled signals is in the real components and the other is in the imaginary components of the complex FFT input. Since the spectrum of the real component is conjugate symmetric and the spectrum of the imaginary component is conjugate antisymmetric, the two signals can be separated by forming the conjugate symmetric and antisymmetric components of the output spectra. This operation is performed in the Real-Adjust Module.

For sampling frequencies  $F_s$  other than 10 or 20 MHz bandwidth, filters with adjustable bandwidths are used after the matched low-pass filters. The current design does not allow for sampling rates greater than 20 MHz, but this could be implemented by adding a higher bandwidth IF section, faster ADCs, and a fast input buffer which would read into the current input buffers at 20 MHz. Contiguous spectra could not be processed at a rate higher than 20 MHz, so the duty cycle would be the ratio of the fast sampling rate to the 20 MHz processing rate.

## B. Input and Timing

The Input and Timing module triple buffers the input data using three, 2-Mbyte RAM buffers, implements the delay operation for the first stage of the FFT, and generates all the synchronization signals for the spectrum analyzer. The synchronization signals, explained below, include the start-of-spectrum indicator, valid spectrum indicators, and frame triggers. The synchronization signals are sent down the processing pipe on two serial lines. Each module delays the synchronization signals by an amount equal to the module pipe delay before sending the signals to the next module in the pipe.

The current buffer is read out by the delay commutator which outputs pairs of complex data samples,  $x(k)$  and  $x(k + N/2)$ ,  $0 \leq k \leq N/2 - 1$  which are to be combined in the first butterfly stage of an  $N$ -point FFT. The start-of-spectrum indicator coincides with  $k = 0$  and occurs every  $N$  samples, or every  $N/2$  sample pairs. Because of the periodicity of the delay commutator, the beginning of a new spectrum can occur only every  $N$  samples times. Thus, if the sampling rate is less than the 20-MHz processing rate, and the next buffer is not full when the commutator is reset to  $k = 0$ ,  $N$  ticks of the 20-MHz processor clock must be skipped before the buffer can begin to be processed. The first two time lines in Fig. 4 show the write and read times for a 16 MHz sampling rate. Since the sampling rate is four-fifths the processing rate, every fifth spectrum does not contain valid data, and a valid spectrum indicator does not appear in the third time line. Since the processing pipe cannot be stopped to wait for the buffer to fill, the modules continue cycling through their processing, but the absence of the valid spectrum indicator prevents the data from being added into the accumulation, used to update the baseline, or entered into the spectrum filter.

The frame trigger signals control the lengths of the spectrum accumulations. The bottom time line in Fig. 4 is for accumulation frames of 3 spectra. Since two input channels can be processed simultaneously there are two separate accumulator memories. The valid spectrum indicator identifies which accumulation each spectrum belongs to, and there are two frame triggers, one for the  $\alpha$  accumulator and one for the  $\beta$  accumulator so that they can be controlled independently. For dual-channel input, the Real-Adjust module interleaves a spectrum of the real input with a spectrum of the imaginary input. Thus the even-numbered spectra belong to the real channel and are accumulated separately from the odd spectra which belong to the imaginary channel.

Dicke switching also makes use of the two separate accumulators. In Dicke switching the antenna switches from looking at a source for  $K$  spectra, to looking at the noise for  $K$  spectra. The switching is driven in synchronization with the spectrum analyzer. A counter can be set to discard a specified number of samples to allow synchronization with a pulsing signal, or settling time for the Dicke switch.

## C. Executive Board

The Executive board generates the clock signals for the high-speed digital components. It also contains a test input buffer and response buffer which are read only and write only respectively to the microprocessor bus. The buffers are in the memory address space of the microprocessor. Test data vectors are generated by the microprocessor and written to the test input buffer at microprocessor speed and then transferred to the Input and Timing module at the full hardware processing

speed. The Executive board generates the appropriate test synchronization signals as commanded by the microprocessor. Test output vectors are returned to the microprocessor via the Response Bus and Response Buffer. The Response buffer is a First-In-First-Out (FIFO) buffer. The data streams through until a programmed stop signal freezes the contents. The contents can then be read out and analyzed by the microprocessor at its slower processing rate.

Test vectors can be applied to any module or group of modules in the spectrum analyzer by bypassing all preceding boards in the pipe so that the test vectors are passed on unchanged to the input of the first board under test. Every module has the ability to send output on the Response Bus to the Response Buffer. The bypass data path is 46-bits wide to accommodate 44-bit data vectors plus two synchronization lines.

#### D. Window Module

The window module applies a real, symmetric window to the fixed-point, complex input pairs for each spectrum. The operation is a simple real times complex multiply:

$$y_m(n) = w(n)x_m(n) \quad 0 \leq n \leq N-1, 0 \leq m \quad (1)$$

The subscript will be used to index successive spectra, or equivalently, segments of  $N$  samples. Thus  $x_m(n)$  refers to the  $(mN + n)$ th time sample.

Since a window is a point-by-point multiplication in frequency, by the convolution theorem of the Discrete Fourier Transform (DFT) (Ref. 1; p. 59) the DFT of a windowed signal is the convolution of the spectrum of the unwindowed signal with the frequency response of the window. Thus the window  $\{w(n)\}$  shapes the frequency response of the FFT. If no window is applied, performing an FFT on a finite data segment of length  $N$  is equivalent to using a rectangular window. The frequency response of the rectangular window is shown in Fig. 5. Since the FFT samples the spectrum at frequency points  $kF_s/N$ ,  $0 \leq k \leq N-1$ , the FFT of a frequency component depends on how close it is to a multiple of  $F_s/N$ . The FFT for a frequency which is an exact multiple of  $F_s/N$  is shown in Fig. 6(a). Figure 6(b) shows the FFT of a frequency exactly half way between two sample points. For Fig. 6(b) the peak magnitude is 3.9 dB less than for Fig. 6(a), and the first sidelobe at -13.5 dB is only 9.5 dB down from the peak. The frequency response can be improved by using a window like the Hanning window shown in Fig. 7. This window broadens the main lobe and suppresses the sidelobes. A frequency halfway between the FFT sample point would have a peak magnitude

only 1.4 dB down from the peak, and the first sidelobe is 31.5 dB down from the peak. Frequency responses of many different windows are given in Ref. 1.

The hardware for this module consists of an arithmetic unit for the window multiply, and a RAM coefficient memory which is loaded in from the control microprocessor during system setup. The window length is the same as the FFT length, and can be any power of two from  $2^4 = 16$  to  $2^{20} = 1,048,576$ .

#### E. FFT Module

The FFT module transforms the time domain signal into the frequency domain by means of an FFT. The FFT of the input signal is:

$$Y_m(k) = \frac{1}{N} \sum_{n=0}^{N-1} y_m(n) \exp(-j2\pi nk/N), \quad 0 \leq k \leq N-1 \quad (2)$$

where  $\{y_m(n)\}$  is the output of the window module. The length of the FFT,  $N$ , can be any power of two from  $2^4 = 16$  to  $2^{20} = 1,048,576$ .

The FFT is implemented using the Radix-2, Decimation-In-Frequency (DIF), pipelined FFT algorithm. This algorithm is derived in detail in Ref. 1 (pp. 368-371). The Radix-2 form was chosen because it has the least complexity for addressing data and coefficients. The post-multiply butterfly of the DIF algorithm was preferred by the hardware designers.

In the Radix-2, DIF algorithm the FFT is computed in  $M = \log_2 N$  stages. Each stage consists of a "butterfly" operation on  $N/2$  pairs of complex points. The butterfly, shown in Fig. 8, takes the sum and difference of the complex pairs and then multiplies the difference by the appropriate "twiddle" factor. The twiddle factors are complex roots of unity  $W_N^{nk} = \exp(-j2\pi nk/N)$ . At the  $i$ th stage,  $1 \leq i \leq M$ , the pairs for each butterfly operation are the results from the previous stage taken  $2^{M-i}$  apart,  $z_{i-1}(n)$  and  $z_{i-1}(n + 2^{M-i})$ , and the twiddle factors are addressed by  $n2^{i-1}$ . The butterfly operation is:

$$\left. \begin{aligned} z_i(n) &= \{z_{i-1}(n) + z_{i-1}(n + 2^{M-i})\} \\ 0 \leq n \leq N-1 \\ z_i(n + 2^{M-i}) &= \{z_{i-1}(n) - z_{i-1}(n + 2^{M-i})\} W_N^{n2^{i-1}} \\ 1 \leq i \leq M \end{aligned} \right\} \quad (3)$$

with

$$z_0(n) = y_m(n)$$

and

$$Y_m(U(n)) = z_M(n)$$

where  $U(n)$  is the bit reversal of  $n$ . The bit reversal is performed in the Unscrambler.

Each FFT board, except the final board, contains two butterfly stages of a pipe-lined, Radix-2 FFT. The final board contains four butterfly stages. Each stage consists of an arithmetic unit (AU) and a PROM coefficient memory followed by a RAM delay memory with a cross switch. The cross switch is programmed to choose the correct data pairs for the next stage. The first six stages of the FFT are implemented using fixed-point arithmetic and data. The final fourteen stages use floating-point arithmetic and data. The last four stages are implemented on one board. There are no delay memories since only pipelined delays are used. In addition, the last two stages require no complex multiplies. The third card contains Response Bus connection for the first cage. The FFT module can be configured to produce any power-of-2, complex FFT from a 16-point up to a 1,048,576-point ( $2^{20}$ ).

## F. FFT Unscrambler

The FFT Unscrambler puts the complex, floating-point FFT data stream in natural order. The ordering involves bit reversing the addresses such that the input data with address:

$$\begin{aligned} n = & n_0 + 2n_1 + 2^2n_2 + \dots + 2^k n_k \\ & + \dots + 2^{M-2}n_{M-2} + 2^{M-1}n_{M-1} \end{aligned} \quad (4)$$

becomes the output address:

$$\begin{aligned} U(n) = & n_{M-1} + 2n_{M-2} + 2^2n_{M-3} + \dots \\ & + 2^k n_{M-k-1} + \dots + 2^{M-2}n_1 + 2^{M-1}n_0 \end{aligned}$$

The bit reversal is performed by writing the data into a double-buffered memory in order  $n$ , and reading them out in order  $U(n)$ . Two parallel output lines concurrently carry the upper and lower halves of the spectrum in sequential order. The module consists of two 1,048,576 by 44-bit memories and a crossbar switch. It also contains the control bus and response bus interfaces for the FFT cage. The board can unscramble any transform length from 16 to  $1,048,576 = 2^{20}$ .

## G. FFT Real-Adjust Stage

The FFT Real-Adjust stage is used when the input is one or two real channels,  $a_m(n)$  and  $b_m(n)$ ,  $0 \leq n \leq N-1$ , instead

of the normal input of a single complex channel. Two real channels can be transformed simultaneously if one channel is put into the real components and another channel into the imaginary components of the complex FFT input:

$$x_m(n) = a_m(n) + j b_m(n), \quad 0 \leq n \leq N-1 \quad (5)$$

Since the spectrum of a real signal is conjugate symmetric and the spectrum of an imaginary signal is conjugate antisymmetric, the two channels can be separated by forming the conjugate symmetric and antisymmetric components:

$$A_m(k) = \frac{1}{2} \{Y_m(k) + Y_m^*(N-k)\}, \quad 1 \leq k \leq N-1 \quad (6)$$

and

$$B_m(k) = -\frac{j}{2} \{Y_m(k) - Y_m^*(N-k)\}, \quad 1 \leq k \leq N-1$$

with

$$A_m(0) = \text{Re} \{Y_m(0)\}$$

$$B_m(0) = \text{Im} \{Y_m(0)\}$$

## H. Power Accumulator

The power accumulator module computes the power of each of the complex points output by the FFT and accumulates the power values into one of the channel accumulation memories. The output,  $P_k(n)$ , is

$$P_k(n) = \sum_{m=L(k-1)}^{Lk-1} |Y_m(n)|^2, \quad k \geq 0 \quad (7)$$

where  $L$  is the length of the accumulation. The average power in each spectrum is computed and accumulated in place of the zero-frequency component. The two accumulation memories,  $A$  and  $B$ , are controlled independently, and can be used to accumulate two different channels, or to switch the accumulators as in Dicke switching. Every valid spectrum entering the Power Accumulator module is identified as a type  $A$  or  $B$  by the frame synchronization indicator. The maximum accumulation length is  $2^{16}$ , and is controlled by frame trigger synchronization signals generated by the Input and Control module.

The power is computed in floating-point arithmetic and then converted to fixed-point in order to provide sufficient precision during the accumulation. The accumulations are performed with fixed-point arithmetic and then converted to floating-point before being output to the baseline module. The

module consists of two arithmetic units, one for the power calculation and one for the accumulation operation, two barrel shifters to fix the input and float the output, and a RAM memory for the 60-bit wide, 1,048,576-point accumulated spectra. The power accumulator board also provides the multi-bus connection for the signal processing cage. The input and output are 22-bit floating-point numbers.

### I. Baseline Module

The baseline module removes the frequency-dependent system gain by multiplying each spectrum channel by the inverse of a baseline value. The floating-point baseline can be loaded in by the system controller during system setup, or computed by inverting a baseline formed from the input channels. The baseline is formed by applying an exponentially decaying update filter to the frequency components of the power accumulator. Components greater than a specified threshold value are not used to update the baseline. A local spectrum average is used instead. The baseline algorithm is:

$$S_k(n) = P_k(n)/B_{k-1}(n) \quad (8)$$

$$B_k(n) = \begin{cases} (1 - \alpha)B_{k-1}(n) + \alpha P_k(n) & \text{if } S_k(n) \leq T \\ (1 - \alpha)B_{k-1}(n) + \alpha \bar{P}_k(n-1) & \text{if } S_k(n) > T \end{cases} \quad (9)$$

where  $\bar{P}_k(n)$  is a local average of the spectrum values formed by another exponentially updated filter:

$$\bar{P}_k(n) = (1 - \beta)\bar{P}_k(n-1) + \beta P_k(n), \quad \text{if } S_k(n) \leq T \quad (10)$$

If the baseline spectrum value  $S_k(n)$  is above the threshold, the spectrum average is not updated. Since there can be two distinct sequences of spectra from the Power Accumulator, there are two baseline memories to accommodate them.

The inversion of the floating-point baseline is performed by table lookup on the mantissa values, and negation of the exponents. The baseline module consists of an arithmetic unit (AU) to implement the inverse baseline multiply and update filter, a floating-point table-lookup inverter, and RAM memories for the baseline.

### J. Spectrum Filter Module

The Spectrum Filter module contains a five-coefficient Finite Impulse Response filter which filters each baselined frequency component across successive spectra. The spectrum filter can be a matched filter to any desired single-frequency

time signature of duration less than or equal to five accumulation intervals. The example in Fig. 2 was for a source transiting an antenna beam. Another example is needed. The filtering operation is:

$$F_k(n) = \sum_{i=0}^4 \alpha_i S_{k-i}(n) \quad (11)$$

where the  $\alpha_i$ ,  $0 \leq i \leq 4$  are input parameters. To implement a filter of length less than 5, simply set the end coefficients to zero.

The filtered output which passes a threshold test is sent to the Spectrum Processor via a high-speed data bus for additional processing. The output can also be sent to the Output processor. A bit mask suppresses specified components so that they do not pass the threshold test. The mask is input from the control computer. The spectrum filtering operation requires an arithmetic unit and four 1,048,576 by 22-bit memories.

### K. Output Processor

The Output Processor controls the amount and form of the spectrum analyzer output. A full buffer of data is collected and then sent to the system Response Buffer on the Executive board for transfer to the processing computer via a microprocessor bus. The output data is used for the spectrum displays and can also be used for further processing. The output data can be a snapshot of a full spectrum, or a selected window of data from successive spectra. The Output Processor can divide a spectrum into sections and choose the  $k$ th value, select the peak value, or compute the average value for each section. The Output Processor consists of an arithmetic unit and a 1,048,576 by 50-bit buffer memory. It also contains the Control Bus and Response Bus connections for the signal processing cage.

## III. Scaling

The 22-bit floating-point format represents numbers with exponents from -32 to 31, and mantissas from -1 to  $1-2^{-15}$ . Numbers between  $(-2^{-33} + 2^{-47})$  and  $2^{-33}$  will all be represented as zero. Negative overflow occurs for numbers less than  $-2^{31}$ , and positive overflow occurs for numbers greater than  $2^{31} - 2^{16}$ . Scaling is used to keep the numbers computed in the spectrum analyzer within the dynamic range of the floating-point format.

To determine the range of numbers the analyzer will encounter, we consider two different types of input signals, a pure sinusoid, and Gaussian white noise.

The samples of the complex sinusoid with amplitude  $A$ , initial phase  $\phi$ , and frequency  $f$  are:

$$x(n) = A \exp(j\phi) \exp(j2\pi fn/F_s) \quad (12)$$

where  $F_s$  is the sampling frequency. The FFT of this signal, as defined in Eq. (2), using a rectangular window is:

$$Y(k) = \frac{A \sin \pi N f / F_s}{N \sin \pi (f / F_s - k / N)} \quad (13)$$

with power  $A^2$  at  $k/N = f/F_s$  and less than  $A^2$  elsewhere. If we accumulate over  $L$  spectra, the maximum power will be  $LA^2$ .

For complex Gaussian white noise, the real and imaginary components are independently, identically distributed Normal ( $0, \sigma^2$ ):

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp[-x^2/(2\sigma^2)] \quad (14)$$

The frequency bins of the FFT from Eq. (2) will be identically distributed Normal ( $0, \sigma^2/N$ ). Taking the sum of the squares

of the imaginary and real components, and accumulating  $L$  of these power values results in a chi-square distribution of channels with  $2L$  degrees of freedom:

$$p(x) = \frac{1}{2^L (\sigma^2/N)^L \Gamma(L)} x^{L-1} \exp[-xN/(2\sigma^2)] \quad (15)$$

The expected value of this distribution is  $2L\sigma^2/N$  with variance  $4L(L+1)\sigma^2/N$ .

Since the spectrum analyzer input consists of 8-bit two's complement numbers, the range for both  $A$  and  $\sigma$  is  $2^{-7}$  to 1. As seen from the above discussion, the FFT output range is much larger than the input range, and the power of a sinusoidal signal in a frequency bin is approximately  $N$  times as large as the expected power of Gaussian white noise. In order to keep the accumulated output from overflowing the floating-point range, and at the same time keep the noise power from underflowing, we provide a scale factor in the form of an exponent bias applied at the input to the first floating-point FFT stage. The bias exponent,  $B$ , will be based on the length of the FFT,  $N$ , and the number of accumulations,  $L$ , in order to retain as much precision as possible while avoiding overflow.

## References

1. Rabiner, L. R., and Gold, B., *Theory and Application of Digital Signal Processing*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1975.
2. Emerson, R. F., "Biplex Pipelined FFT," *The Deep Space Network Progress Report 42-34*, pp. 54-59, Jet Propulsion Laboratory, Pasadena, Calif., 1976.
3. Morris, G. A., and Wilck, H. C., "JPL 2<sup>20</sup> Channel 300 MHz Bandwidth Digital Spectrum Analyzer," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tulsa, Oklahoma, pp. 808-811, 1978.
4. Olsen, E. T., Lokshin, A., and Gulkis, S., "An Analysis of the Elements of an All Sky Survey," *Proc. IAU Symposium No. 112*, Boston, MA, June 1984 (in press).
5. Solomon, J., Quirk, M. P., Olsen, E. T., and Lawton, W., "A Signal Detection Strategy for The SETI All Sky Survey," *The Telecommunications and Data Acquisition Progress Report 42-83*, Jet Propulsion Laboratory, Pasadena, Calif., 1985.

**Table 1. Spectrum analyzer modules**

Module	Quantity
IF Signal Input	2
Executive Board	1
Input and Timing	2
Window	1
Fixed-Point FFT	3
Floating-Point FFT	5
4-stage FFT	1
Real-Adjust Board	1
Unscrambler	1
Power Accumulator	2
Baseline	2
Spectrum Filter	2
Output Processor	2
Output Buffer	1
Interface Board	1
Power Supplies	2
Cages	5
Rack	1
Microprocessor System and Software	1

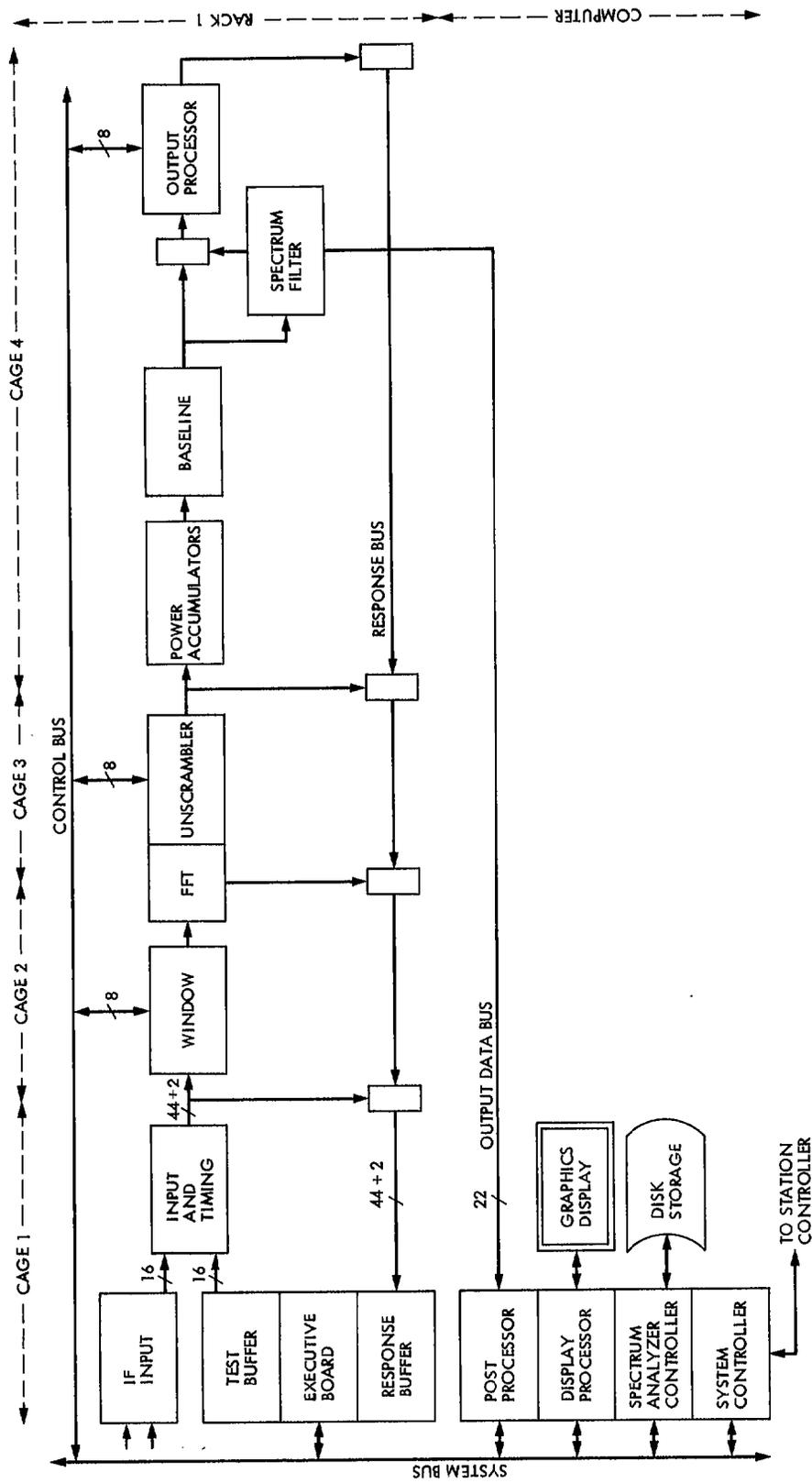


Fig. 1. Spectrum analyzer system

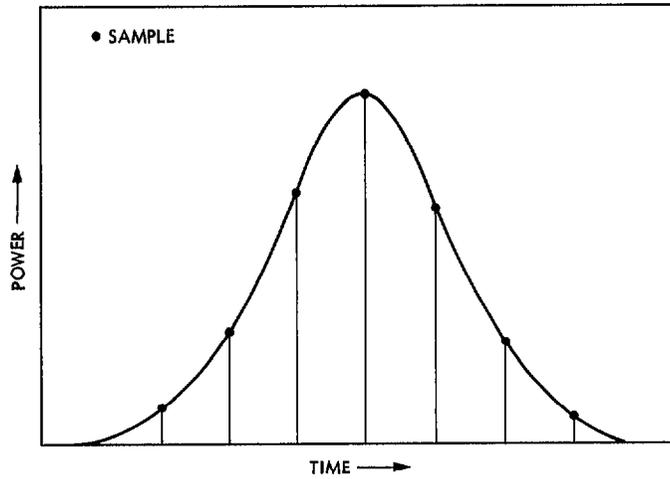


Fig. 2. Frequency response of a source transiting an antenna beam

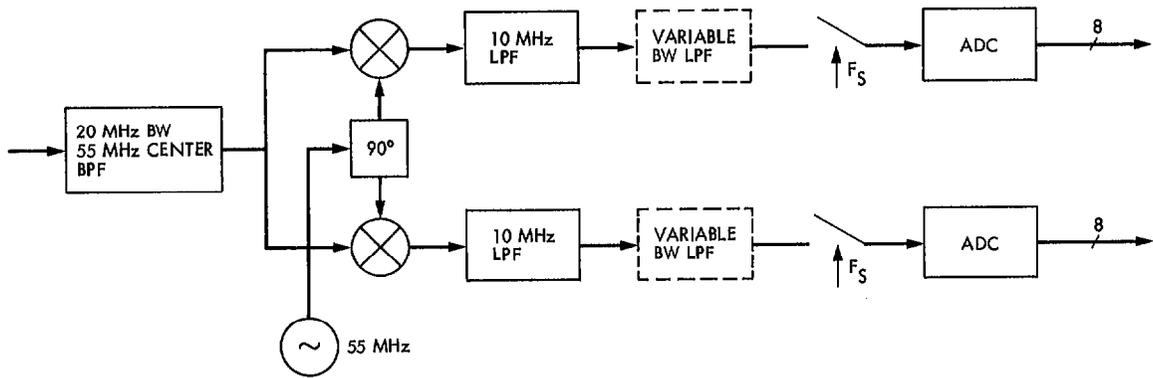


Fig. 3. Intermediate frequency signal conditioner

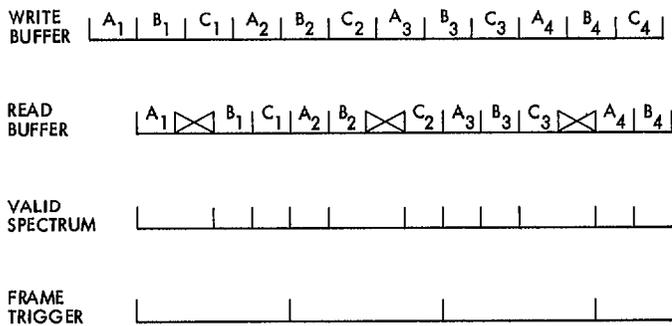


Fig. 4. Timing diagram

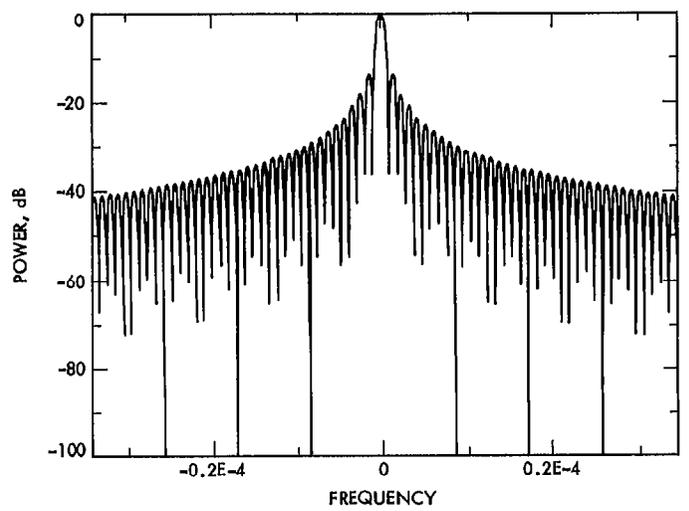


Fig. 5. Rectangular window response

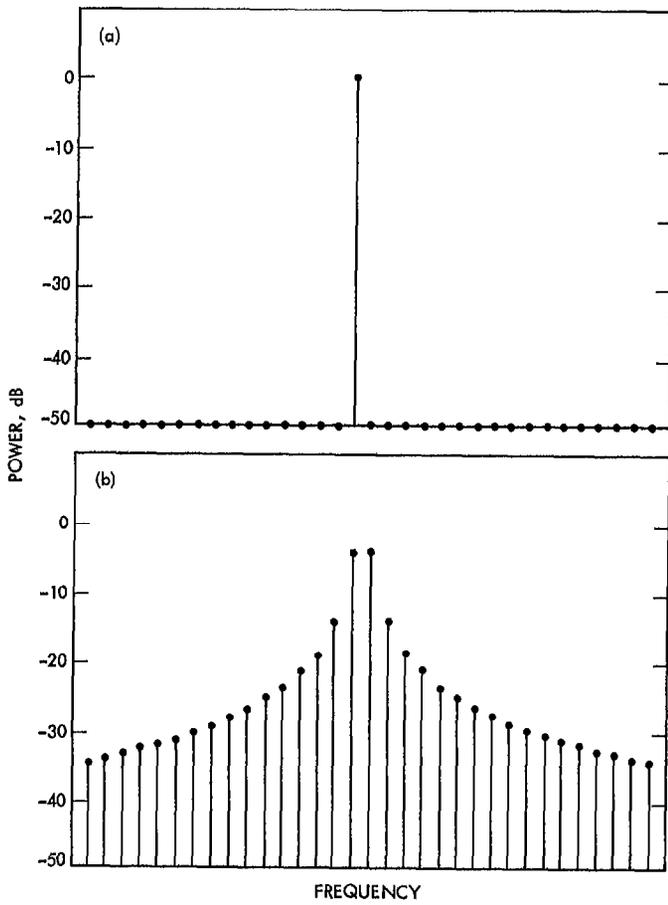


Fig. 6. Frequency responses: (a) a frequency halfway between two FFT frequencies; (b) a signal with frequency coinciding with an FFT frequency

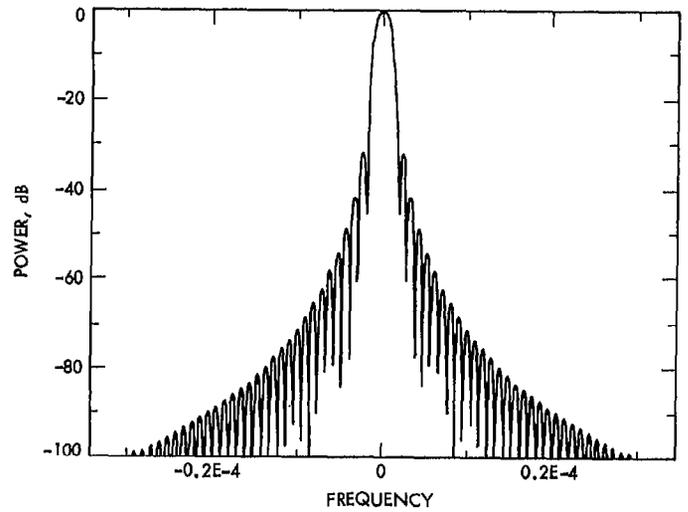
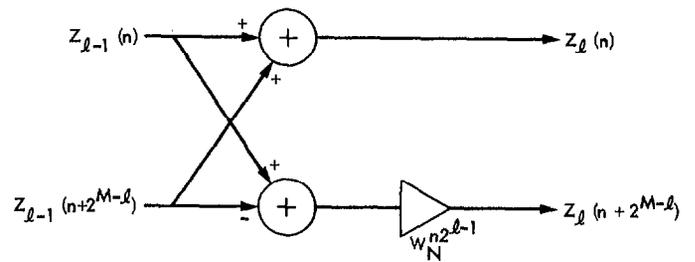


Fig. 7. Hanning window response



NOTE: ADDITIONS AND MULTIPLICATIONS ARE COMPLEX

Fig. 8. Decimation-in-frequency butterfly stage

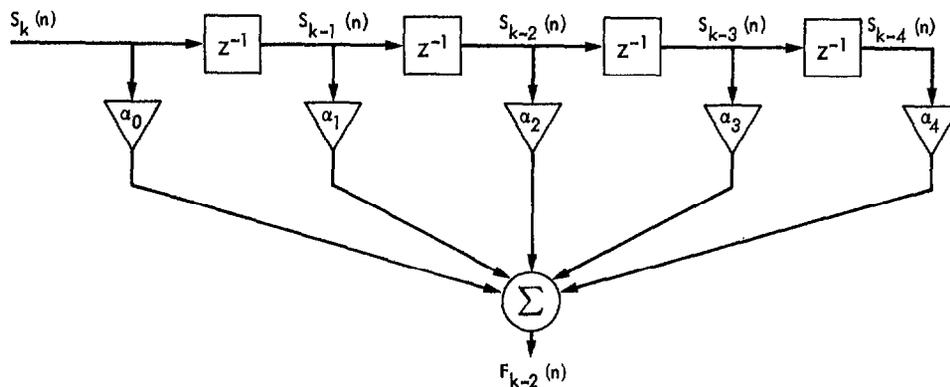


Fig. 9. The 5-stage finite impulse response filter