HONEYWELL

SYSTEMS & RESEARCH DIVISION

DIGITAL SIGNAL PROCESSOR

ID-C-1

Digital Signal Processor

HONEYWELL Systems & Research Division

Printed in U.S.A. ID-C-1 26 January 1971

FOREWORD

The digital signal processor design discussed in this document was developed at Honeywell's Applied Research Department of the Systems and Research Division by Mr. Robert Berg and Dr. Larry Kinney of the Computer Techniques section. Mr. Ferdinand Ohnsorg developed the Fast Walsh Transform, and Dr. M. Geokezas did the accuracy analysis and application studies. Both men are in the Information Processing section.

This development effort has been sponsored by the Honeywell Research Department and by the Honeywell Ordnance Division, whose support and encouragement is gratefully acknowledged.

George Swanlund
Principal Staff Scientist

INTRODUCTION

Digital signal processing is being used to perform an ever-increasing share of signal processing and spectral analysis tasks in both scientific and operational disciplines. This is because digital processing has advantages not available in conventional analog techniques:

- Output is compatible with digital equipment used in subsequent computations
- Very-low-frequency signals are processed efficiently with much smaller equipment
- Insensitivity to environmental conditions or changes
- Operational stability
- A complete set of operating modes is available in one unit
- Smaller, lighter, less expensive, more reliable, more maintainable
- Can time-share devices to service a number of inputs
- Accurate

The implementation of digital processors has advanced quite rapidly in recent years because of two key developments:

- Fast Fourier and Fast Walsh algorithms can now be used for frequency transforms
- Integrated circuitry now permits low-cost, special-purpose computers

Computational speed has increased dramatically because of these fast transforms. Directly implementing a Fourier transform requires N^2 computations, where N is the number of discrete samples. The fast transform reduces the number of computations required to $N \log_2 N$. As an example, when N = 512, the computations are reduced from 262, 144 to 4, 608, or by a factor of 57.

The advent of large-scale integrated circuits (LSICs) permits the economic realization of parallel arrays of processing modules. An array of 512 arithmetic units can further reduce processing time by a factor of 512. Each arithmetic unit performs only nine computations and the transform can be performed in real time. Furthermore, incorporating microprogramming into each module allows a variety of processing mode uses, which combines the flexibility and speed of special-purpose computation.

Honeywell's DIgital Signal Processor (DISP) incorporates these new algorithms into parallel arrays of identical processing modules. Each module consists of two arithmetic units fabricated on one LSIC chip, resulting in a processor whose size, weight, cost, power dissipation, and reliability are particularly appropriate for:

- General laboratory computations
- Real-time simulations

(either stand alone or tied in to a computer facility)

- Operational hardware
- Portable test equipment

CONTENTS

ILLUSTRATIONS

SECTION I
SUMMARY

The DIgitial Signal Processor (DISP) is comprised of five main units (Figure 1-1):

1. An expandable array of N identical processing modules, where each module performs identical computations simultaneously
2. 2N shift registers
3. Control unit
4. Input buffer
5. Output buffer

Each processing module is, in effect, a small microprogrammed computer with its own input and output registers, memory, arithmetic section and instruction repertiore. All input and output data are represented in 12-bit* fractional 2s complement format. The arithmetic portion of the processing module can:

1. Add
2. Subtract
3. Multiply (simple and complex)

The instruction repertoire permits selecting a complete set of signal processing modes. (These modes are discussed in detail in Section II.) DISP is switched to a new mode simply by a control command.

*12 bits is a nominal value. The number of bits is optional.

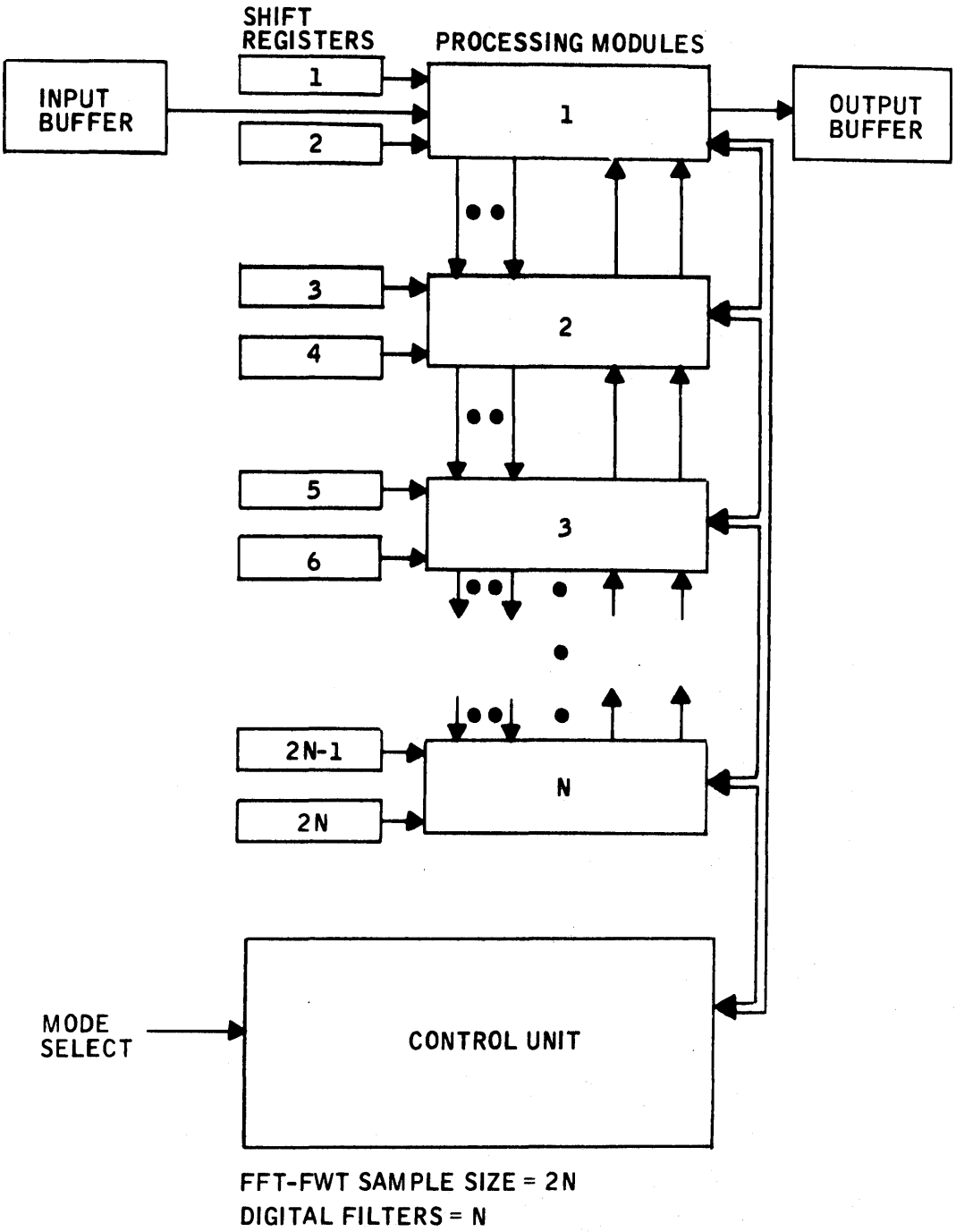

Figure 1-1. Block Diagram of a Digital Signal Processor

The unit scales automatically to maintain full dynamic range. Any arithmetic overflow is detected by the module in which it occurs. The module notifies the control section that overflow has occurred, and the control unit issues a command, correcting the overflow condition and properly scaling the data in all modules.

Each processing module is fabricated on one identical LSIC using bipolar-compatible Metal Oxide Semiconductor (MOS) technology. Each module performs serial arithmetic at a bit cycle of 1 μsec or less. Since the data word used in DISP is 12 bits long and requires overflow detection and correction, a word time consists of 13-bit times (13 μsec).

Shift registers are required to store the weighting factors (W_i) of the Fast Fourier Transform (FFT), and to store the constants of the digital filter. The length of the shift registers increase as the $\log_2 N$ to accommodate the FFT mode (every time N is doubled, another stage is added in the FFT algorithm).

To illustrate the physical characteristics of a DISP, the following estimates are made for a DISP containing 256 processing modules:

Description	Size (cu ft)	Weight (lbs)	Power (watts)
Standard packaging	1.0	40	100
Miniaturized packaging	0.1	10	80

DISP can be tied into a GP computer (Figure 1-2) or it can stand alone in real-time simulations or on-line in an operational system. Each processing module has buffer registers internal to the module. Various groupings of these registers allow output data to be transmitted at a rate compatible with a wide range of I/O devices.

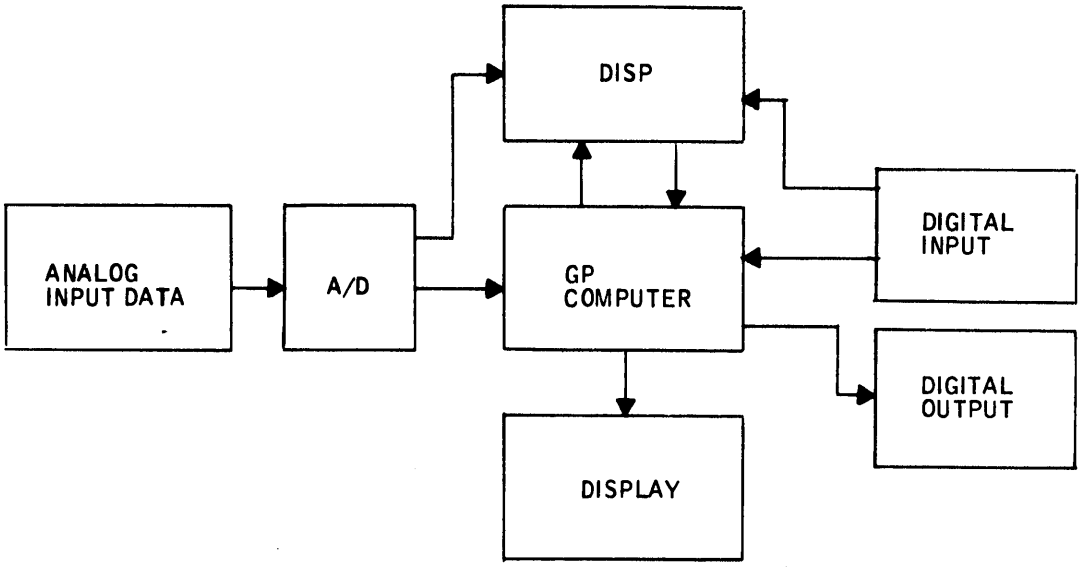

Figure 1-2. DISP-GP Computer Tie-In

DISP VERSUS OTHER DIGITAL PROCESSORS

Appendix A consists of a chart from the IEEE Transactions on Audio and Electroacoustics, Vol. AU-17, No. 2, June 1969, entitled, "FFT Hardware Implementations - A Survey", by Glen D. Bergland. The Honeywell DISP capabilities have been added to this chart. DISP matches or exceeds the capabilities of all other units. In addition, the size, weight and power of DISP are smaller than for any of the other equipment described.

SUMMARY OF REST OF DOCUMENT

Section II discusses the set of operating modes which are available. Some of the complex modes and systems applications require a tie-in to a GP computer.

Section III presents the accuracy results of DISP operating as an FFT and a bank of filters. This analysis establishes that a 12-bit unit will be adequate for the majority of applications.

Section IV presents:

1. A description of the DISP system organization

2. A detailed description of the design of the processing module and how it operates in the system

3. The functions of the DISP control unit

SECTION II
OPERATING MODES

The DISP has three levels of operating modes:

1. The basic modes consist of single operations such as a Fast Fourier Transform (FFT) or a multiplication of two functions.

2. The complex modes consist of two or more basic modes, e.g., the power spectrum mode includes a Fast Fourier Transform and a subsequent squaring of the frequency coefficients.

3. The functional modes consist of some specific signal processing application. Some of the application modes can be performed entirely within DISP, while others assume additional external processing. The functional modes shown are not exhaustive and serve mainly to illustrate typical applications.

BASIC MODES

The list of basic modes and their execution times are given in Table 2-I. A brief discussion of each mode is given below.

Fast Fourier Transform (FFT)

The Fourier Transform is based on sine and cosine functions and is used effectively for spectral analysis of real or complex inputs. The Walsh transform of real inputs is based on rectangular functions analogous to

Table 2-I. Basic Modes

MODES		PROCESSING TIMES, msecs. 256 POINT	512 POINT
1. FAST FOURIER TRANSFORM,	FFT	1.118	1.274
a) INVERSE FFT,	IFFT	1.118	1.274
2. FAST WALSH TRANSFORM,	FWT	.104	.117
a) INVERSE FWT,	IFWT	.104	.117
b) FWT - COMPLEX INPUTS	FWT (C)	.117	.130
c) IFWT - COMPLEX INPUTS	IFWT (C)	.117	.130
3. TIME WINDOW WEIGHTING (COMPLEX)	W	.299	.299
4. SQUARE COMPLEX FUNCTION	SQ	.364	.364
5. MULTIPLY TWO FUNCTIONS (COMPLEX)	MPLY	.400	.400
6. DIGITAL FILTER BANK - 2nd ORDER	DFB	.468	.468
a) DFB 4th ORDER	DFB (4)	.962	.962
b) DFB 6th ORDER	DFB (6)	1.443	1.443
c) DFB 1st ORDER (LOW PASS FILTER)	LPF	.351	.351

hard-clipped sine and cosine functions. The Walsh transform of complex inputs is based on rectangular functions analogous to the hard-clipped exponential representation of the sinusoids.

The DISP easily computes these three transforms because all use the same computational flow algorithm, although requiring different weighting coefficients. (This algorithm is shown in Figure 2-1 for a complex FFT of eight input samples.)

The unique feature of the algorithm (Figure 2-1) is that each of the k columns ($N = 2^k$) requires identical computations, and combines the same samples to derive a new sample.

A solid line to a node represents addition, a dashed line subtraction, and W_i a complex multiplication. The W_i's represent complex weighting factors because the Fourier transform has a sinusoidal basis function:

$$W_i = \cos \frac{2\pi i}{N} - j \sin \frac{2\pi i}{N} .$$

The operations performed in a single module are shown in Figure 2-2. Each module is time shared over all k columns or stages.

Note that the algorithm does not produce the Fourier coefficients in their natural order. The output order can be found by first numbering the outputs in natural order using binary numbers, then reversing the order of the digits of the binary numbers and interpreting the resulting number as the number of the Fourier coefficient.

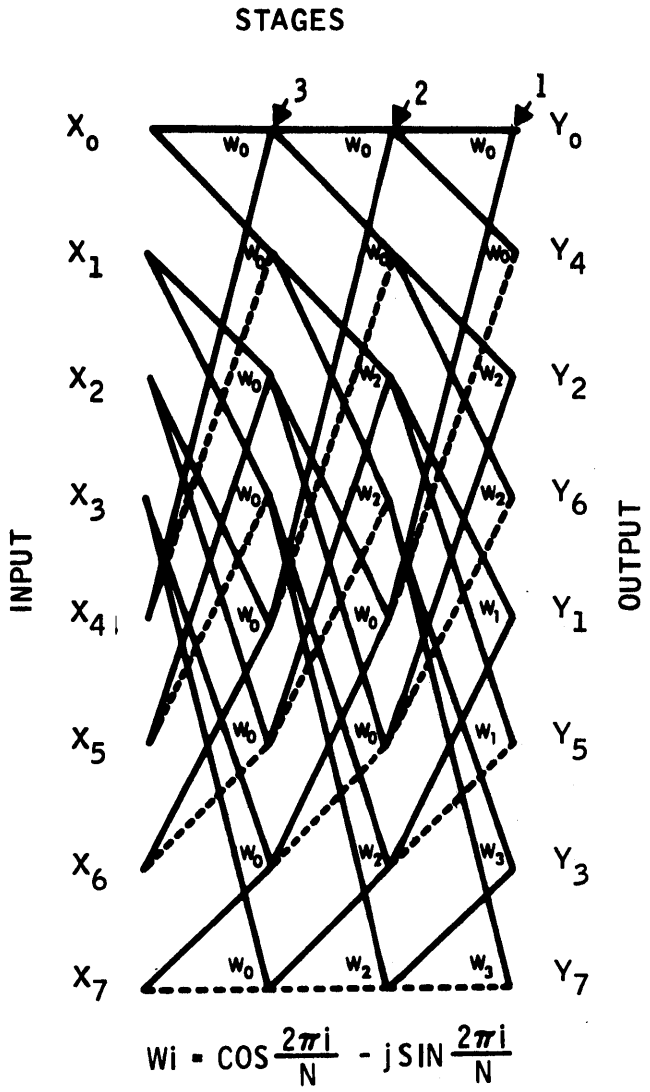

Figure 2-1. Algorithm for the Fast Fourier Transform of Eight Input Samples

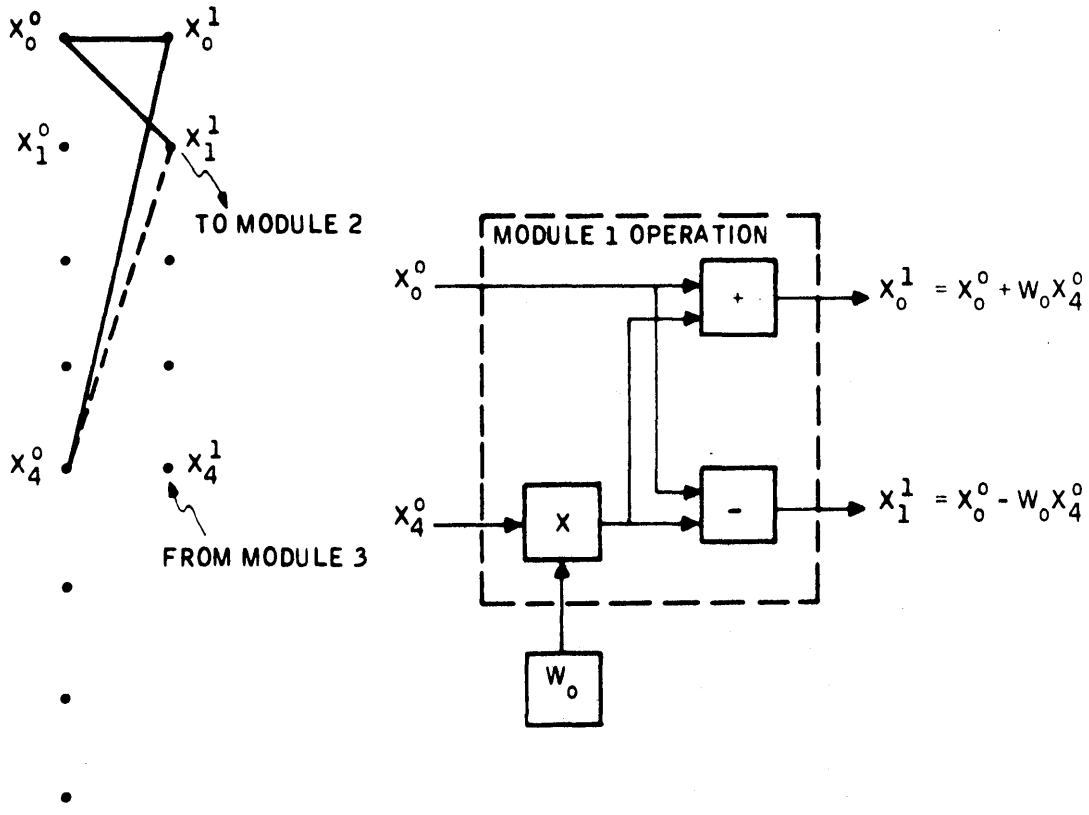

Figure 2-2. Module Operation: FFT Mode

Fast Walsh Transform (FWT)

Since the Walsh transform of real samples is based on rectangular functions, the only weighting coefficients are plus and minus one. These coefficients are processed by addition and subtraction. The combinational algorithm of the FWT is identical to Figure 2-1, if all of the W_i terms are removed. Since FWT computations require no multiplications, they are performed much faster than FFT with the same number of discrete data samples.

The complex Walsh transform algorithm requires multiplying certain data values by the value -j ($j = \sqrt{-1}$) through internally complementing the real portion of the data and interchanging the real and imaginary parts. The algorithm for the complex FWT is the same as that in Figure 2-1 if all values of $W_{n/4}$ are replaced by -j, and all other W_i's removed.

For all FWT algorithms, outputs are ordered differently than shown in Figure 2-1. The FWT output order can be found by numbering the outputs in binary, reversing the digits of the binary numbers, and interpretating the resulting digits as the Gray code for the number of the FWT coefficient. For n = 8, the output order starting at the top of Figure 2-1 is h_0, h_7, h_3, h_4, h_1, h_6, h_2 and h_5.

Time Window Weighting, W

In some cases, it is desired to shape the time representation of the data to achieve a more desireable frequency function. In the cases of coherent detection, multiplication by a reference function is desired. In both of these cases, either real or complex functions are involved for both the input and the time window weighting function.

The time window weighting is accomplished by storing the weighting factors in the module shift registers. The resulting weighted data are retained in the module for subsequent processing.

Square One Function, SQ

The squaring operation is similar to the time window weighting except that the multiplier and multiplicand are the same and are already in the module. Squaring is typically an intermediate operation.

Multiply Two Functions, MPLY

Again the process is similar to the time window weighting except both functions are in the module. MPLY is also usually an intermediate operation.

Digital Filter Bank, DFB

Each module in a DISP is capable of performing second-order digital filtering of the form

$$\begin{bmatrix} X_1(n) \\ X_2(n) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} X_1(n-1) \\ X_2(n-1) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(n)$$

$$Z_1(n) = b_o X_1(n)$$

This is a recursive filter. The state X(n) depends only on the first previous state X(n-1) and the current input u(n). The output $Z_1(n)$ is real and is a function of only $X_1(n)$. The module operation in the filter mode is shown in Figure 2-3. The bandwidth and Q of each filter is determined by the values of the coefficients.

The output $Z_1(n)$ can also be stored in the module. Enough storage space within the module is left to store the states of two other second-order filters. Thus, the module can perform the calculations required of three second-order filters in cascade, thereby simulating a sixth-order digital filter.

COMPLEX MODES

The complex modes consist of two or more basic modes. They are listed in Table 2-II. The processing times shown are for a 512-point transform. Since these modes generally require some interaction with a general purpose digital computer, the operation times are for two different data transfer rates. These rates correspond to two current 16-bit mini-computer, namely 0.286×10^6 s/sec and 1.43×10^6 s/sec; 1 sample = 12 bits.

Power Spectrum, PDF

To compute the power spectrum, the outputs from the FFT are squared. Since the module output Y_i is a complex number, the multiplication is complex. The output is both stored and conjugated (Y_i*). The product YY* is a real, positive number. Also, the power coefficients for positive frequencies (0, N/2-1) are the same as for negative frequencies (N/2, N-1). Thus, only the positive frequencies need to be read out.

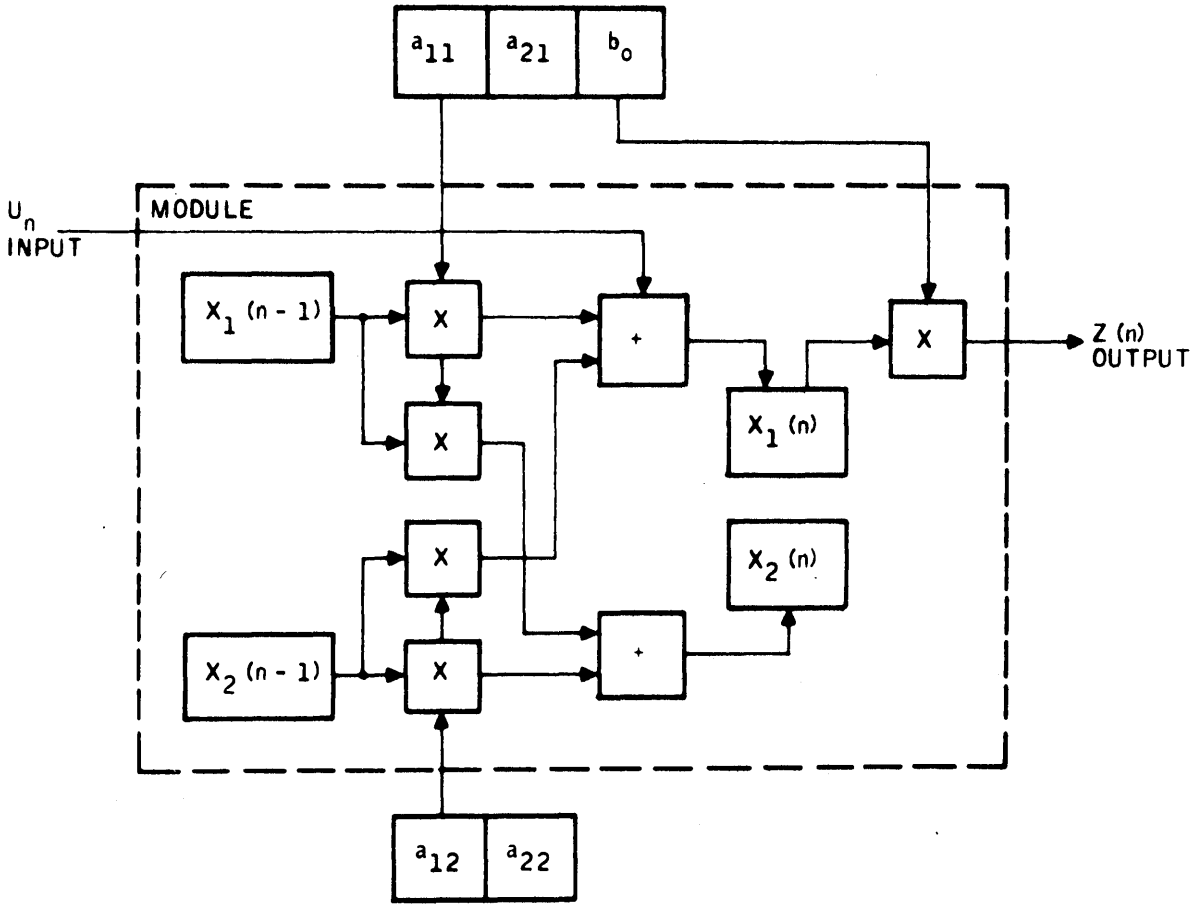

Figure 2-3. Module Operation: Filter Mode

Table 2-II. Complex Modes (512 Points)

MODES		PROCESSING TIME IN MILLISECONDS FOR THE GIVEN TRANSFER RATES	
		4.992×10^6 BIT/SEC	20×10^6 BITS/SEC
1. POWER SPECTRUM	PDF	1.738	1.738
2. CROSS POWER SPECTRUM	XPDF	4.992	2.948
3. AUTO CORRELATION	R_{11}	3.718	3.718
4. CROSS CORRELATION	R_{12}	4.222	4.222
5. CONVOLUTION	H_{12}	4.222	4.222
6. DOUBLE LENGTH/FFT/^{2}	FFT(2)	18.304	4.576
7. QUADRUPLE LENGTH (FFT)^2	FFT(4)	36.608	9.152
8. ENERGY-TIME-FREQUENCY (2nd ORDER)	ETF	0.936	0.936
9. FREQUENCY TRANSLATION	FT	1.738	1.738

Cross-Power Spectrum, XPDF

The operations are the same except that two transforms are required. The first transform outputs are stored in the module while performing the second transform. Also, the power coefficients are now complex. However, only the positive frequency terms need be read out since the negative frequency terms are complex conjugates.

Correlation and Convolution Modes

Correlation and convolution are performed via the Fast Fourier Transform. Both operations require a segment of N/2 zeros adjoining a data segment of N/2 values. Thus, the data sample is only N/2 rather than N.

For correlating two functions $X_1(k)$, $X_2(k)$, the procedure is

1. Adjoin N/2 zeros to $X_1(k)$, $X_2(k)$ as

$$\hat{X}(k) = X(k) \qquad 0 \leq k < N/2$$

$$\hat{X}(k) = 0 \qquad N/2 \leq k < N$$

2. Compute FFT of $\hat{X}_1(k)$, $\hat{X}_2(k)$ to give $\hat{Y}_1(j)$, $\hat{Y}_2(j)$
3. Take Complex Conjugate of $\hat{Y}_2(j)$ or, $\hat{Y}_2(j)^*$
4. Multiply $Z(j) = \hat{Y}_1(j) \cdot \hat{Y}_2(j)^*$
5. Compute FFT^{-1} of Z(j) to obtain $R_{12}(k)$

The output $R_{12}(k)$ represents the correlation over the interval $(-\frac{N}{2}, \frac{N-1}{2})$, i.e.,

$$R_{12}(L) = \frac{1}{N} \Sigma X(k)\, X(k+L) \qquad L = -\frac{N}{2}, -\frac{(N-1)}{2}, \ldots, \frac{N-1}{2}$$

For auto correlation, $\hat{Y}_1(j)$ and its complex conjugate $Y_1(j)^*$ are multiplied, $Z(j) = \hat{Y}_1(j) \cdot Y_2(j)^*$ and transformed to obtain $R_{11}(L)$.

For convolving two functions $X_1(k)$, $X_2(k)$, the procedure is similar.

1. Repeat steps 1 and 2
2. Multiply $Z(j) = \hat{Y}_1 \cdot \hat{Y}_2(j)$
3. Compute FFT^{-1} of $Z(j)$ to obtain $V(k)$. The output $V(k)$ represents the convolution over the interval $-N/2, \ldots, \frac{N-1}{2}$, i.e., $V(k) = \frac{1}{N} \Sigma X_1(L)\, X_2(k-L) \quad L = -\frac{N}{2}, \ldots, \frac{N-1}{2}$

For continuous inputs, correlation is performed on $X_1(k)$ and $\hat{X}_2(k)$, i.e., $X_1(k)$ is N samples while $\hat{X}_2(k)$ has $\frac{N}{2}$ zeros adjoined. The same steps are followed as described above but only the first $\frac{N}{2}$ output samples are valid. Convolution is performed similarly by the last $\frac{N}{2}$ samples are retained (see Reference 1 for more details).

Multiple Length Sample Size

The number of modules in a DISP is determined by sample size of the FFT (or FWT). Nevertheless, a DISP can compute an FFT (or FWT) of sample sizes either larger or smaller than the one for which it was designed. The

computation for a smaller sample size requires that only part of the algorithm be performed. The computation for larger sample sizes requires dividing the sample set into groups. After performing an FFT on each group, the resulting outputs are reordered (by external computer). These are also divided into groups and a partial transform performed on each group. The flow diagram for the case of a double sized window (2N) is shown in Figure 2-4. For the case of 2N there are two complete transforms and two partial transforms. For the case of 4N there are four complete and four partial transforms.

The procedure for a 2N window is as follows:

1. Perform an N point transform on the even numbered points and shuffle outputs

2. Repeat (1) on the odd numbered points

3. Perform one stage of an N point transform on each half of the outputs from (1) and (2) using the weighting coefficients for the last stage of a 2N transform and then shuffle outputs

The procedure for a 4N window is as follows:

1. Perform an N point transform using every fourth sample. Shuffle outputs.

2. Repeat (1) three times.

3. Perform two stages of an N point transform on each quarter of the outputs from (1) and (2) using the weighting coefficients from the next-to-last and last stages of a 4N transform. Each transform output is one-fourth of the 4N transform.

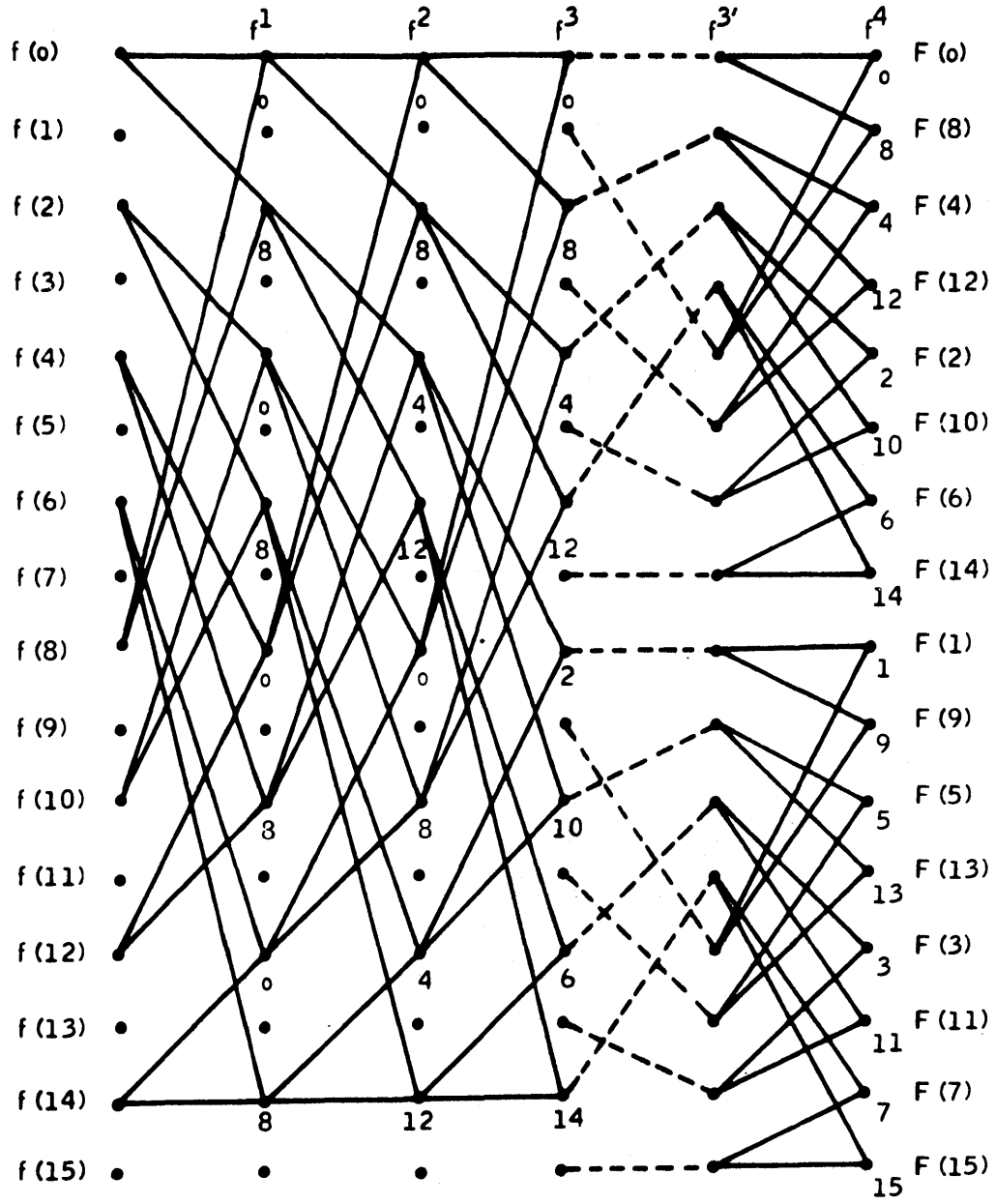

Figure 2-4. Sixteen Point FFT Using an Eight Point Processor

Energy-Time-Frequency

For a continuous output of a filter bank, one generally wants the energy rather than the filter output directly. This is accomplished by squaring the outputs and passing through a low pass filter. Thus, the operations in sequence are DFB, SQ, LPL.

Frequency Translation, FT

Often it is desirable to obtain finer frequency resolution over some portion of the frequency spectrum. This is handled by the frequency translation mode. The procedure is as follows:

1. Select the lower and upper frequency points $Y_L(j)$, $Y_H(j)$. At least four frequency points should be included (two besides $Y_L(j)$ and $Y_H(j)$.

2. Perform FFT on window 1 to obtain Y(j).

3. Perform $(FFT)^{-1}$ on Y(j) within selected interval and store time samples $\overline{X}(k)$. The number of time samples equals the number of Y(j) retained.

4. Repeat steps 2 and 3 until the number of time samples equals N.

5. Perform an FFT on the N sample time function. This provides an N sample resolution of the selected interval.

It is noted that the input/output transfer rates become limiting in some modes. At an effective bit transfer rate of 3.684×10^6 bits/sec, the transfer rate limits the processing for XPDF, FFT(2) and FFT(4). At a rate of 14.736×10^6 bits/sec, the transfer rate limits FFT(2) and FFT(4). In this latter case the computation time is only slightly less than the transfer rate.

Also, one notes that all modes except ETF can handle a 50ks/sec sampling rate. Thus, real time processing can handle a 20 KHz input signal bandwidth.

Some of the complex modes are illustrated in Figure 2-5. These show the repeated application of the basic modes. They also show the relationships between sample lengths and resolution.

FUNCTIONAL MODES

The basic and complex modes can be used to perform a variety of signal processing functions. Some typical examples are listed below. Generally, these require input/output and other processing functions in addition to the DISP. To make the illustration specific we have assumed two different configurations using mini-computers. The DISP would be under control of the computer. The computer would also provide data storage, data reordering, post-processing and data display and output.

The major factor is the transfer rate of the computer. With direct memory access DMA, the rates are:

H316 - 0.312×10^6 sames/sec (16 bit)

Supernova SC - 1.25×10^6 samples/sec (16 bit)

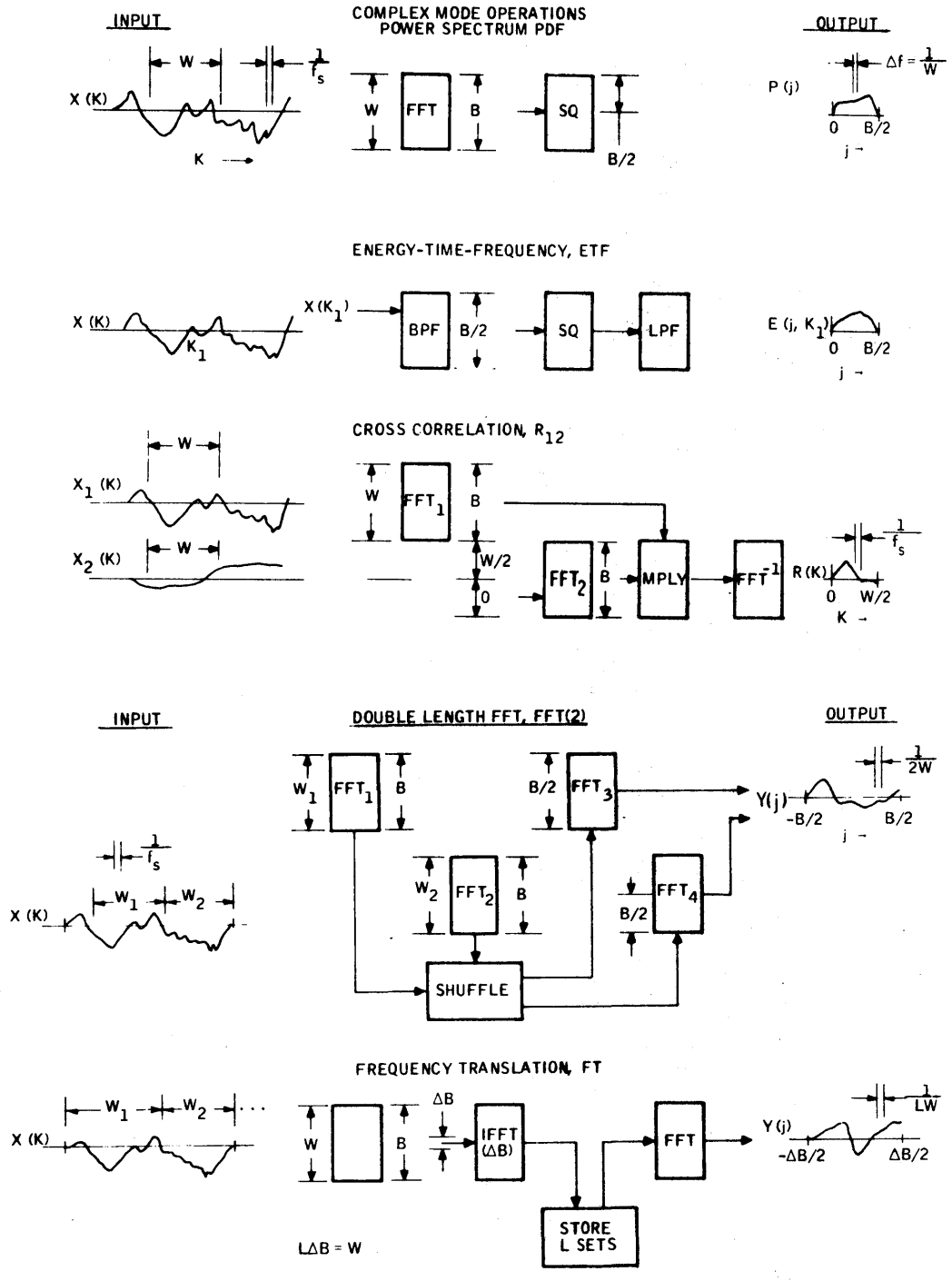

Figure 2-5. Complex Mode Operations

The DISP outputs one 12-bit word every 13 bit times (1 μsec). For the lower transfer rate, 4 output channels would be patched into 3 16-bit words. For the higher rate, 16 channels would be patched into 12 16-bit words. The resulting effective transfer rates are:

H316 - 0.307×10^6 samples/sec (12 bit)

Supernova SC - 1.25×10^6 samples/sec (12 bit)

The minimum time to transfer a set of samples is

	Sample Transfer Rate		
	256 samples	512 samples	1024 samples
H316	0.832 msec	1.664	3.328
Supernova SC	0.208	0.416	0.832

Using these transfer rates, the speeds for specific applications can be determined.

Logrithmic Frequency Analysis, LFA

The first application is for spectral analysis over a wide frequency range. Both proportional and logrithmic frequency intervals are available. We will describe the logrithmic since it is more complex to implement. The input is assumed to be sampled at 50ks/sec and quantized into 12-bit words. Further, each decade in frequency will be sampled separately as shown in Figure 2-6. It is desired to form a time-averaged 1/3-octave power spectrum. The power spectrum is formed in DISP and the frequency and time averaging performed in the GP computer.

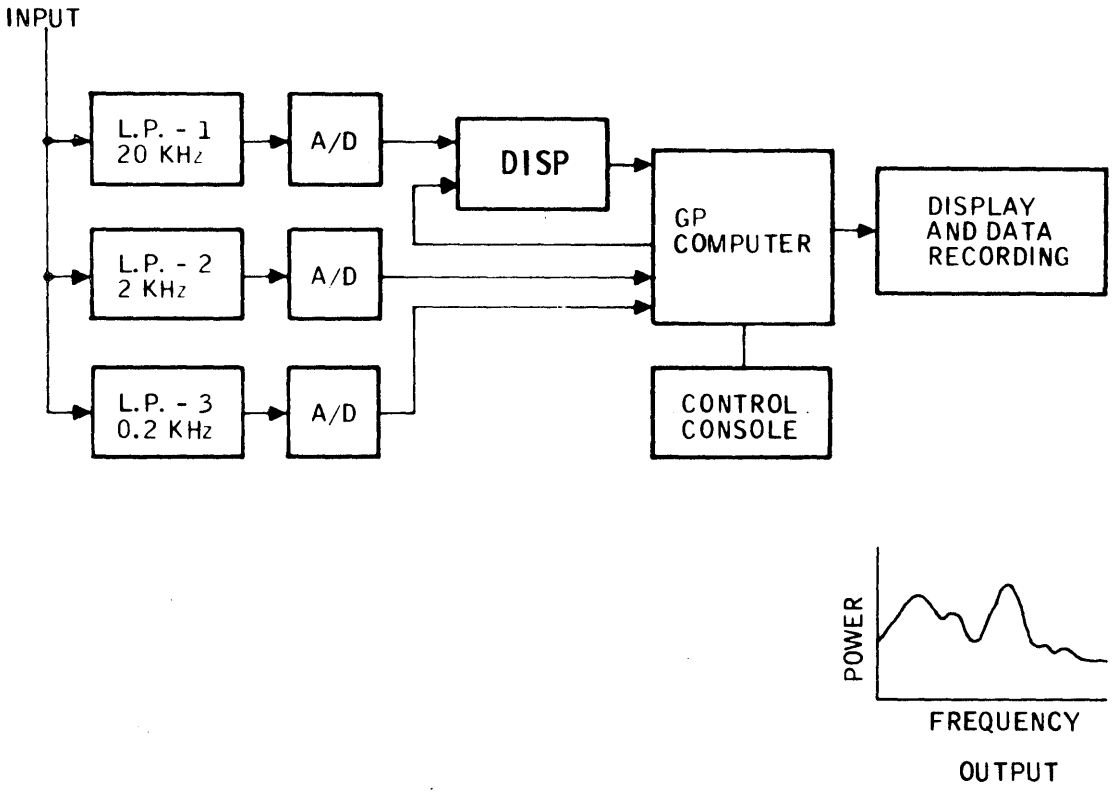

Figure 2-6. Logarithmic Frequency Analyzer

The DISP performs the power spectrum operation on each window of 512 samples from the high speed channel. The slower data channels are fed into the computer. Every 10th window, the 512 samples from the medium speed channel is processed, and likewise for every 100th window for the low speed channel. The resulting spectrum is illustrated in Figure 2-7.

The frequency coefficients can be averaged into logrithmic intervals. Two typical intervals, 1/3 and 1/15 octave are shown. For the 1/15 octave, the first (and smallest) band contains one frequency coefficient. The last band contains 10. For the 1/3 octave, there are five times as many coefficients per band. This averaging of coefficients over frequency bands is performed in the GP computer. Also, any time averaging is performed in the GP computer.

If finer frequency resolution is required, multiple windows can be processed. Using a 4-window mode would increase the frequency resolution by four. This increased resolution for the power spectrum is also shown in Figure 2-7.

If much finer resolution is required over some part of the spectrum, the frequency translation mode can be utilized. Suppose the band from 100 Hz to 112.8 Hz is to be expanded. This band contains 16 frequency coefficients saved from each transform of the 512 data window. The coefficients are inverse transformed to form a time sample of 16 points. After 32 such windows (32 seconds), the time sample is 512 points. It is transformed to provide a 256-point frequency set from 100 Hz to 112.8 Hz. The frequency resolution is 1/16 of the previous Δf or 0.05 Hz.

Coherent Detection System

The coherent detector detects the target and estimates its position and velocity. In the case of coherent detection, the transmitted signal $r_T(t)$ is reflected from some target and the received signal s(t) contains range, velocity and acceleration information about the target. For narrow band detection the two operating modes are a) Doppler search and b) Range search.

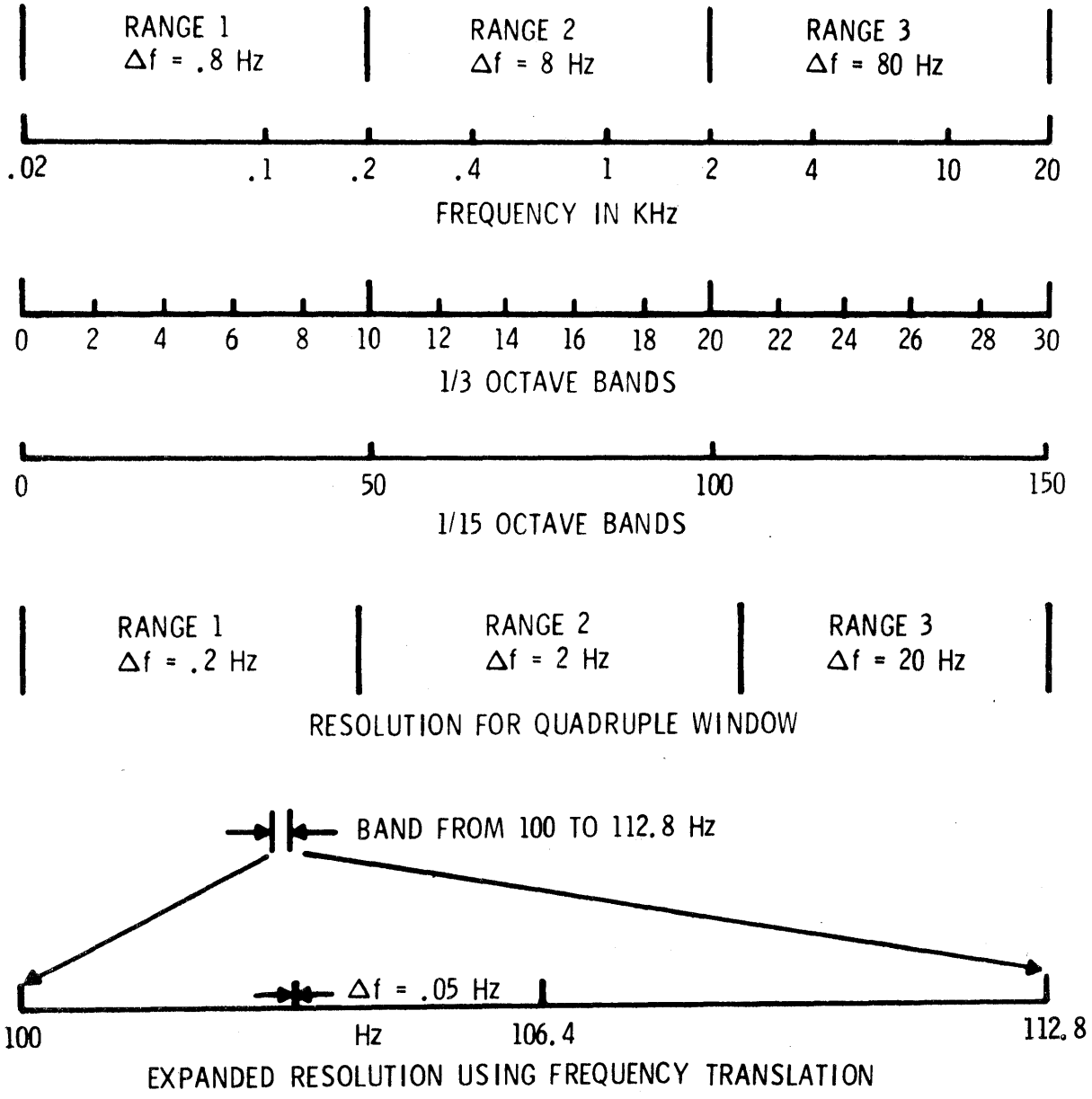

Figure 2-7. Power Spectrum Output Formats

For the narrow band coherent detector in the Doppler Search Mode (Figure 2-8) the received signal s(t)' is quadrature demodulated, lowpass filtered, and converted from analog to digital signal. It then is multiplied by the reference transmitted signal, which is Fourier transformed. The square of the Fourier components represent the ambiguity function for a particular delay (range) as a function of frequency shift (Doppler).

For the narrow band coherent detector in the Range Search Mode (Figure 2-9) the received signal is processed initially as before to form the complex signal, $S_c(n)$. The DISP-FFT is used to Fourier transform 512 samples of $S_c(n)$. The transform $S_c(f_k)$ is multiplied by the Fourier transform of the reference signal $R_o(f_k)$, which may have been stored in the DISP premultiply shift registers. Results are then processed through the inverse FFT. The square magnitude of the output represents the ambiguity function for a particular Doppler (see Figure 2-9).

Wideband coherent detection may require several references because of decorrelation at large Doppler shifts. For example, in the Doppler Search Mode, M reference signals $r_d^*(t)$ with M different Doppler shifts (Figure 2-10). Each reference signal is multiplied by the received signal $s_c(n)$ and then the product is FFT transformed. The magnitude squared represents the ambiguity function about the reference Doppler. Each reference Doppler and FFT transformation may be performed in parallel with M DISP's, or sequentially with one DISP and M reference signals stored in M shift registers.

Walsh-Fourier Signal Representation

The chief advantage of using the Walsh-Fourier representation is the increased speed in performing the transform. The Walsh-Fourier representation may be useful, especially in the area of data compression and

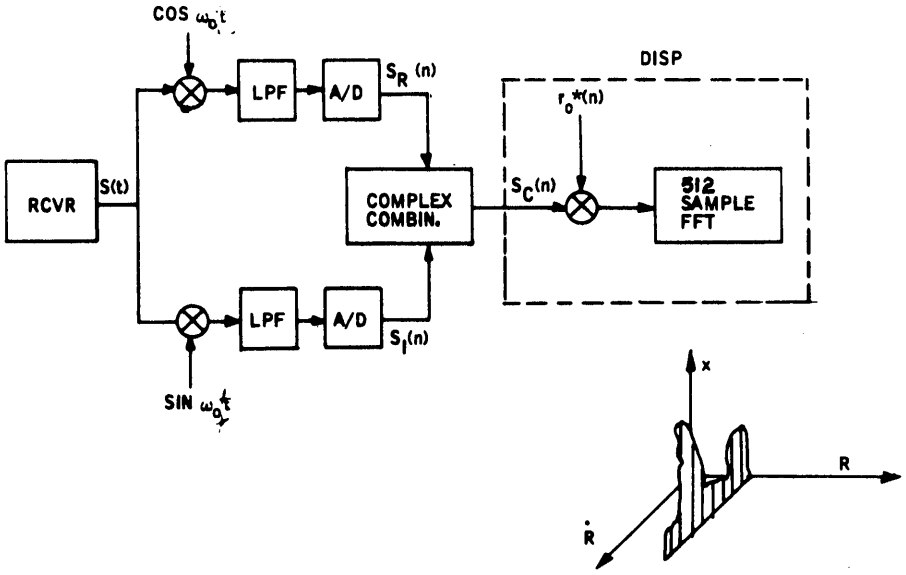

Figure 2-8. Narrow Band Coherent Detector: Doppler Search Mode

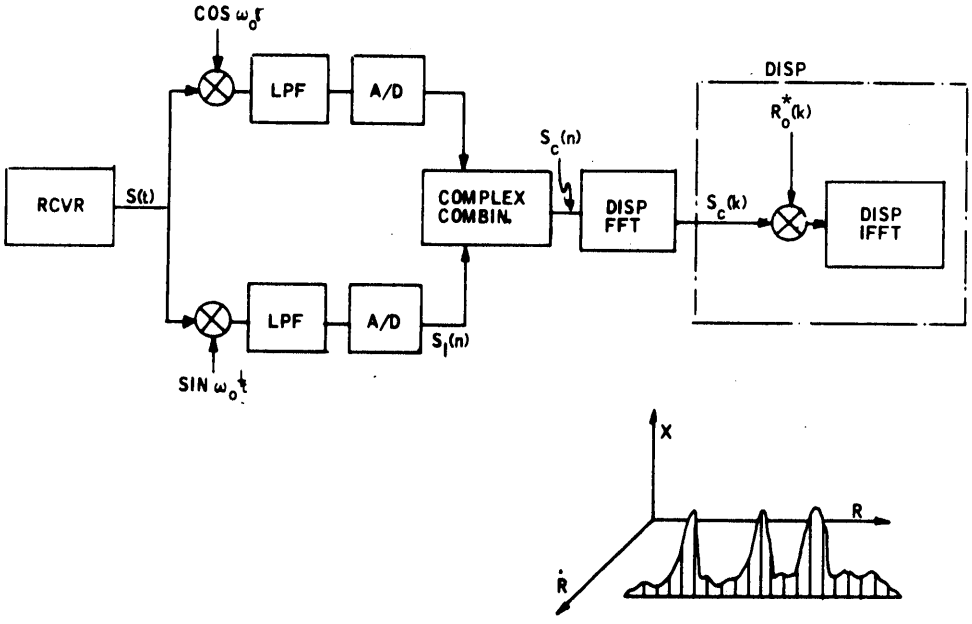

Figure 2-9. Narrow Band Coherent Detector: Range Search Mode

signal classification[2]. Application of the Walsh transform to obtain the power spectral coefficients of the channel vocoder before transmission over a channel has been noted by several authors[3]. Other investigators[4, 5] have studied the merits of the "transformation compression" approach with other methods of data compression, finding it efficient but difficult to implement. Perhaps the Walsh transform with its simple implementation in DISP will make this method practical.

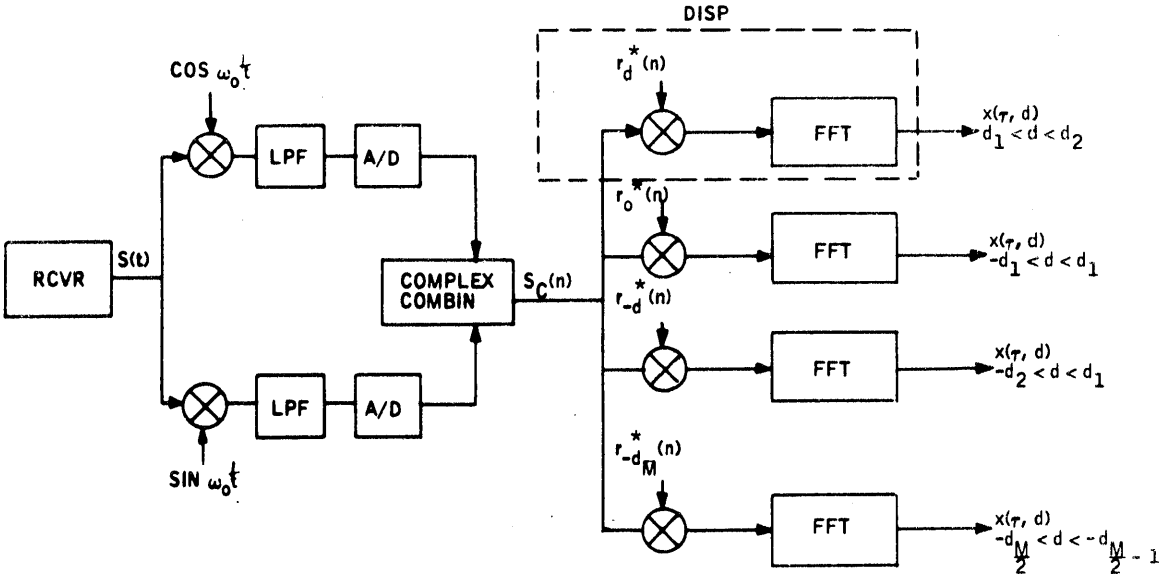

Figure 2-10. Wide Band Coherent Detector: Doppler Search Mode

SECTION III
ACCURACY

Accuracy is critical in digital operations: too few bits lead to erroneous results; too many bits decrease speed and increase costs. Consequently, numerous application studies were made before selecting 12 bits as the nominal word length for DISP. In addition, the accuracy of DISP operating as a Fast Fourier Transform (FFT) and as a Digital Filter Bank (DFB) was evaluated theoretically as well as experimentally. The experiments used an exact simulation of DISP on a general purpose computer.

FFT

The accuracy analysis of the fast Fourier transform mode included both statistical and deterministic effects. The statistical analysis evaluated the effects of roundoff and truncation. The theoretical[8] values are:

Roundoff Error

$$NSR_r = 2\,n\,\sigma_\epsilon^2 = 2\,\sigma_\epsilon^2\,\log_2 N$$

where

$N = 2^n$ is the sample size

σ_ϵ^2 = error variance

Truncation Error

$$NSR_t = (2n + 81)\ \sigma_\epsilon^2$$

For a white noise input, the value of σ_ϵ^2 is $\frac{2^{-2N}}{12}$ or about 10^{-7} for N = 256. The noise-to-signal ratio from both sources is, therefore, about 10^{-5}.

A simulation using a sinusoidal input gave a noise-to-signal ratio of 2×10^{-5}. Theoretical analysis shows that the sinusoidal input should produce noise 15% greater than for a white noise input. Thus, the simulation results agree closely with the theoretical predictions[6] (2×10^{-5} vs. 1.15×10^{-5}). A complete analysis is given in Reference 8.

Dynamic Range

Dynamic range can be measured two ways. One is the ratio of maximum to minimum values of input. This is the inverse of the quantization accuracy or 2^N = 66 db. (Note that DISP scales automatically so that the full dynamic range is always utilized.)

A second way to measure dynamic range is to insert two signals, A_1 and A_2. As A_2 is decreased in magnitude, the error in its FFT representation will increase. This error was determined experimentally by introducing an input signal,

$$X(t) = A_1 \cos 2\pi f_{63} t + A_2 \cos 2\pi f_k t$$

The ratio of A_2/A_1 was varied and the FFT computed over all values f_k. The resulting deviation in the estimated value A_2 from the actual value is shown in Figure 3-1 for 12-bit accuracy and in Figure 3-2 for 16-bit accuracy.

The experimental results for 12-bit words show that a dynamic range of 40 db in A_1/A_2 gives a maximum error of 2.5 db in estimating A_2.

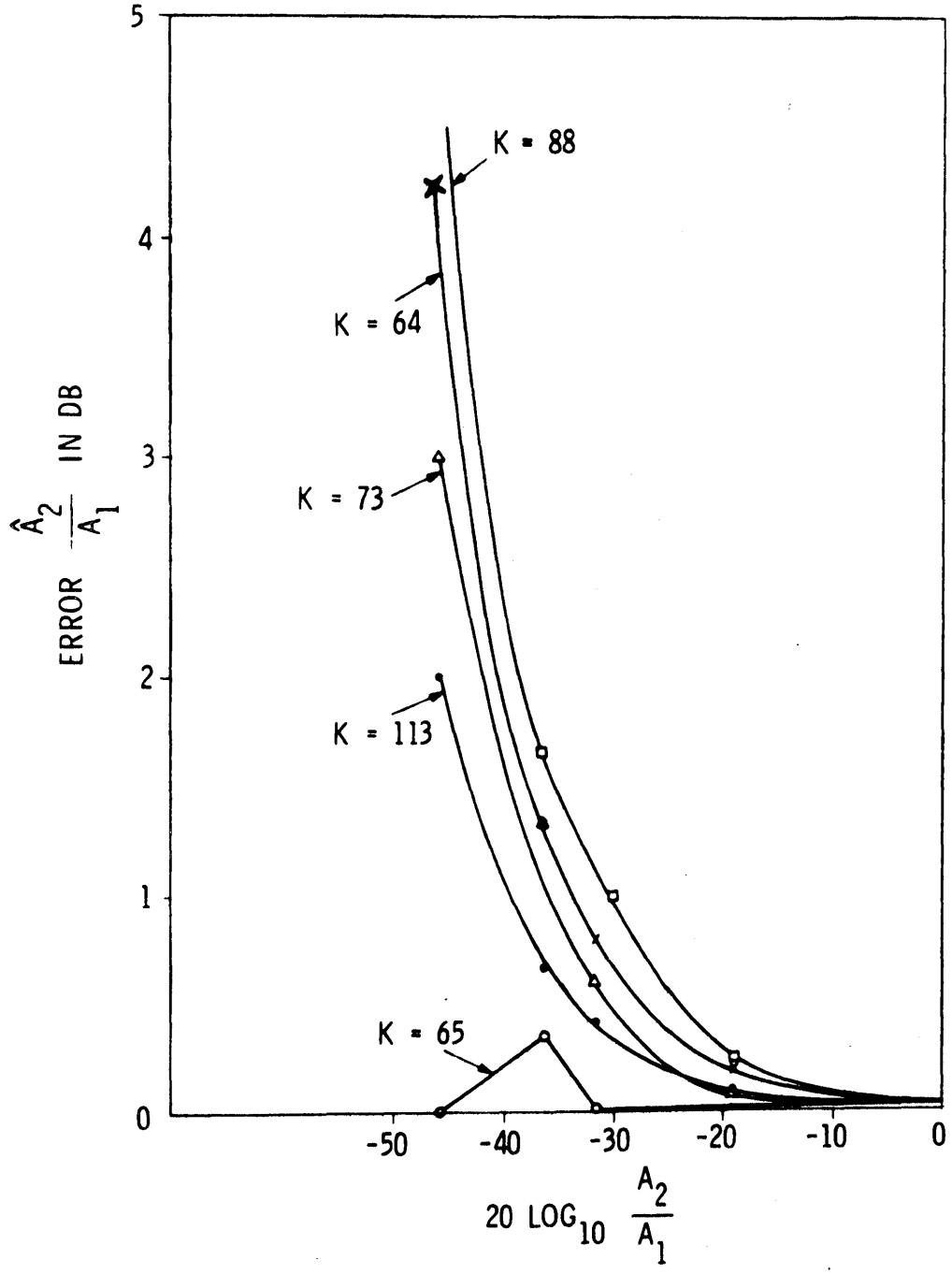

Figure 3-1. Percent Error versus Dynamic Range with Input $A_1 \cos 2\pi f_{63} f + A_2 \cos 2\pi f_K t$, a 256 Sample Window, and 12 Bits/Word

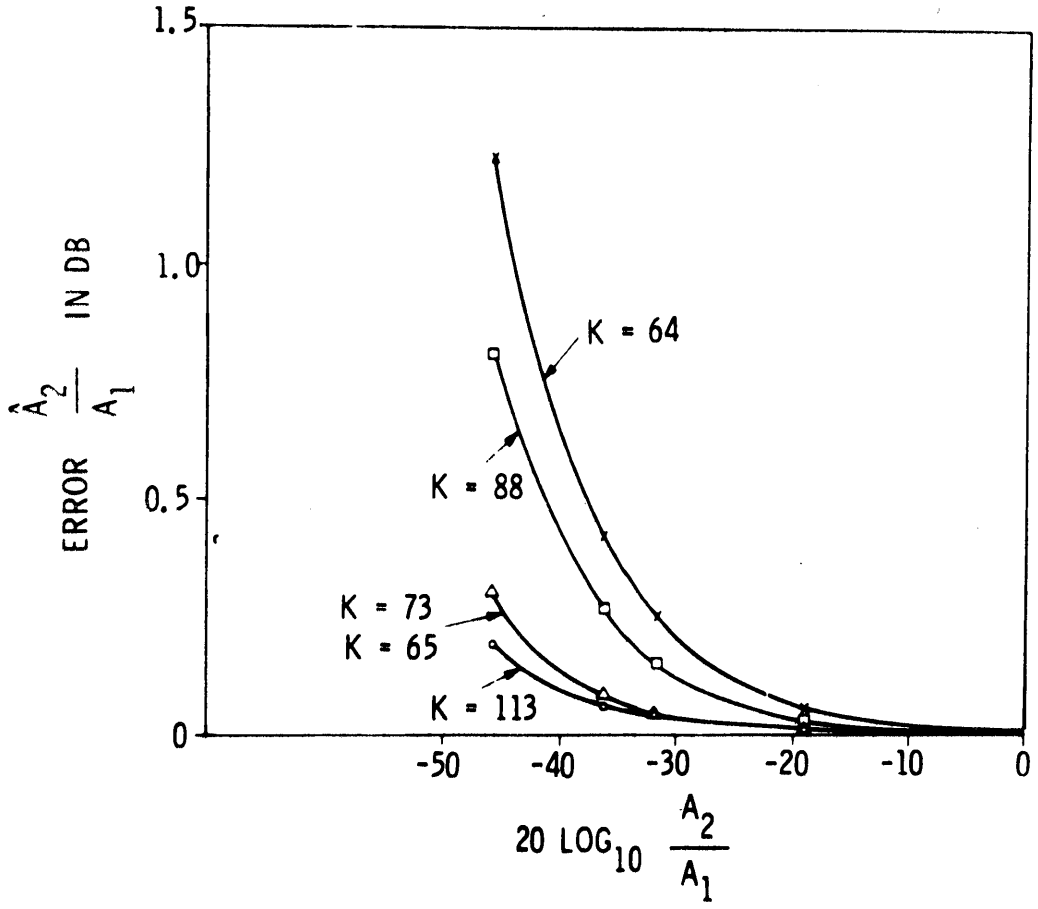

Figure 3-2. Percent Error versus Dynamic Range with Input $A_1 \cos 2\pi f_{63} t + A_2 \cos 2\pi f_K t$, a 256 Sample Window, and 16 Bits/Word

SECTION IV
DISP ORGANIZATION

SYSTEM DESCRIPTION

A DISP consists of a control unit a number of identical Processing Modules and 2 shift registers permodule. Each module can process 2 samples of an FWT or FFT or can implement one bandpass filter.

Referring to the DISP block diagram in Figure 1-1, data inputs are loaded into the input buffer bit serially, with the real and the imaginary portions in parallel. Interconnecting the input and output pins of the Processing Modules properly allows samples to flow down through the modules of the DISP to permit serial-by-word loading and/or moving window operations. Size of the window is governed by the number of load instructions preceeding a computation.

Since the DISP operates in parallel, all outputs are available simultaneously bit serially, word parallel. These outputs can be accepted in this form, or can be stored in the buffer registers of each processing module. If outputs are stored internally, output instructions will feed the contents of the imaginary part of the word into the real buffer register, while the contents of the real register are output. Thus, the external buffer register is a 24-bit serial in/ parallel out shift register and a 24-bit holding register. The number of these registers used determines the output rate.

Figure 1-1 shows the slowest method of obtaining the outputs since it uses only one output buffer. The input and output pins of the buffer registers can be properly connected to feed the computed outputs up through the modules into the external buffer in unshuffled order for either the FFT or the FWT. If the unit interfacing with DISP is capable of high-speed operation,

more external buffers can be added. For example, a computer which can multiplex 24 bit I/O transfers at a rate of 1 MHz could use 24 output buffers. Unloading the results of a complex 256-point FFT would then require 512/24 or 22 word times, or 286 μ sec. Since this is less than the computation time of the FFT, 1118 μ sec, the FFT could be run at top speed. The output would always be completed before the next set of data was ready.

A 256-point FWT requires only 9 word times, and the last word loads new data into the internal buffers. Thus, only 8 output instructions could be performed during the next computation. The output of this computation would have to be delayed while the remaining 14 output instructions are performed.

The fixed interconnections of the processing modules are shown in Figure 4-1, for the FFT algorithm of Figure 2-1. Four modules are required as well as 8 shift registers. The 8 shift registers hold the two words required for the premultiplications, and the three weighting coefficients required for each module. Registers one through four hold real components, while five through eight contain imaginary components.

Each processing module receives two complex inputs representing the i^{th} and the $i + N/2$ data samples. Each module is identical in construction and the arithmetic operations are performed serially, bit by bit, with all modules computing in parallel.

Each module performs the computations indicated by two rows of the transform algorithm. As seen from Figures 2-1 and 4-1 module number 1 receives inputs F_0 and F_4 and forms the sum (S) and difference (D) operations of the top two rows of the algorithm. Module 2 receives inputs F_1 and F_5 and computes the operations of the next two rows, etc. Thus, after one iteration time, the outputs of the modules represent the nodes in column 3 of Figure 2-1. During subsequent iteration times the outputs of columns 2 and 1 are formed. Thus, with all N/2 modules, each operating in parallel, an entire

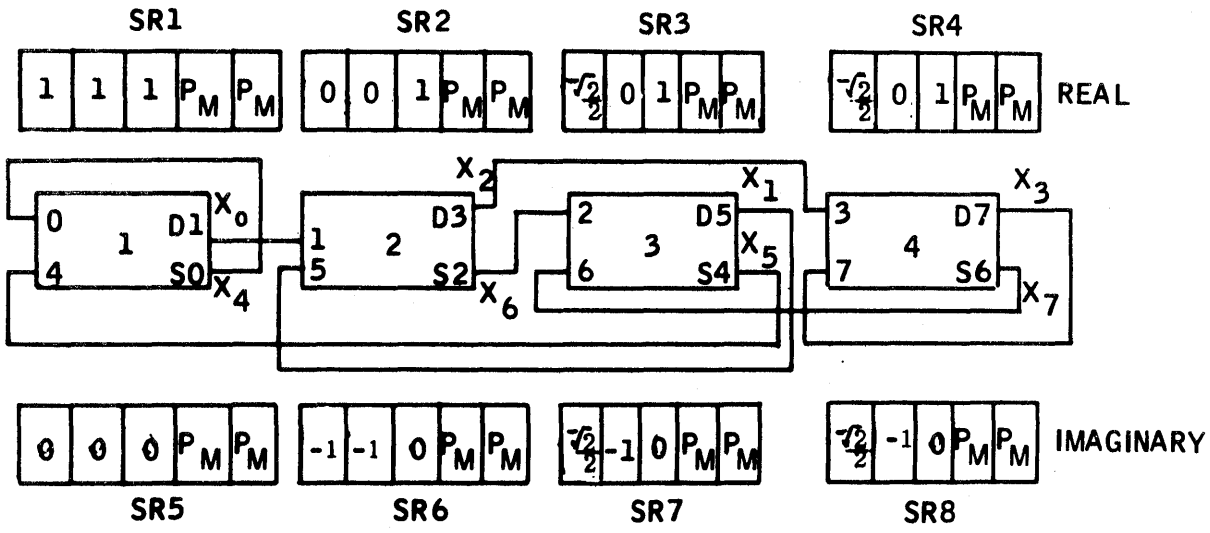

Figure 4-1. Modular Implementation of the FFT

column is computed at once. The number of iteration times is k, where the sample size $N = 2^k$.

The control unit contains in the memory all programs required by the DISP. When a given computation is required, a section of this memory is read out sequentially. Each memory word is decoded into an instruction and distributed to each of the processing modules. The control unit also sends the proper timing information to each module, causing each module to execute this instruction.

PROCESSING MODULE DESCRIPTION

In the processing module (Figure 4-2) the logic gates interconnecting the various module elements are not shown because of their complexity. These gates are defined by logic gate-enable equations (Appendix C) written using the notation shown in Figure 4-2 (e.g., $I_{A_1}{}^R$ represents the input to the register $A_1{}^R$). The notation is identified in Table 4-I.

Table 4-I. Notation for DISP Module

A^R	Intermediate register - real word
A	Adder
B^R	Output register - real word
C	Complementer
D^R	Input register - real word
T^R	Premultiply register - real word
$I_A{}^R$	Input to register A^R
E_{C_1}	Enable complementer 1
$O_A{}^R$	Output from register A^R
OV	Overflow
R^R	Reference register - real word

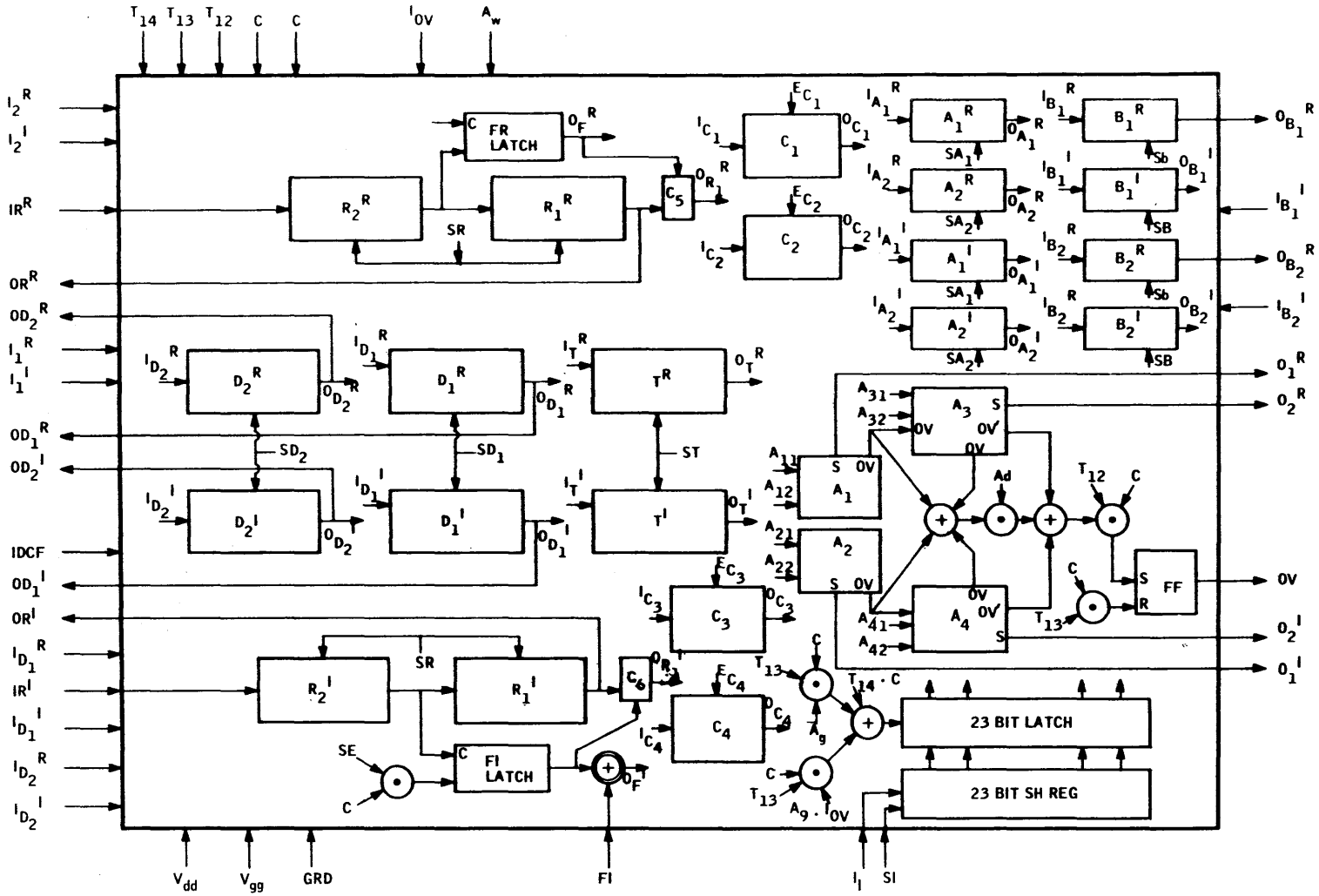

Figure 4-2. DISP Module Block Diagram

The module contains 18 12-bit shift registers designated as A, B, D, T, and R, as well as 6 conditional complementers designated $C_{1\rightarrow 6}$ (Figure 4-3). Figure 4-4 presents serial adders A_1 and A_2, while Figure 4-5 shows adders A_3 and A_4. These adders are designed to detect and correct all arithmetic overflow which may occur during computation. A detailed explanation of their operation is presented in Appendix D.

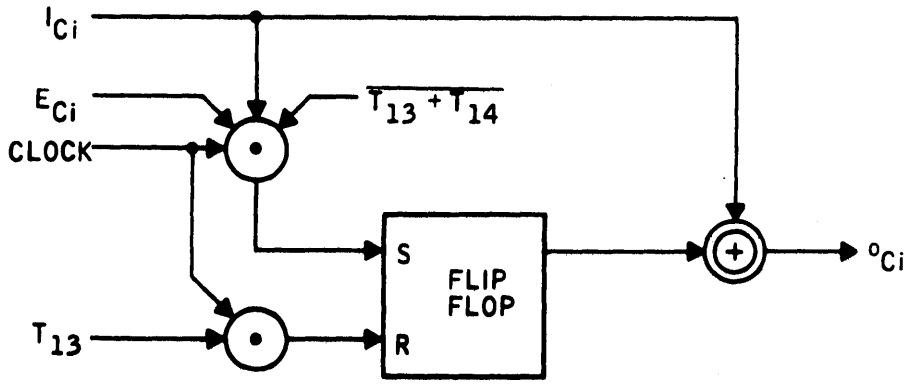

Figure 4-3. Complementer

The complexity of the processing module in equivalent AND/OR gates is shown in Table 4-II. When implemented in MOS technology, approximately 3.5 devices are required for the average gate.

Thus, these 891 logic gates would require approximately 3100 MOS devices. Two builders of semiconductor devices have assured Honeywell that this module can be fabricated on one low threshold (bipolar compatible) LSIC.

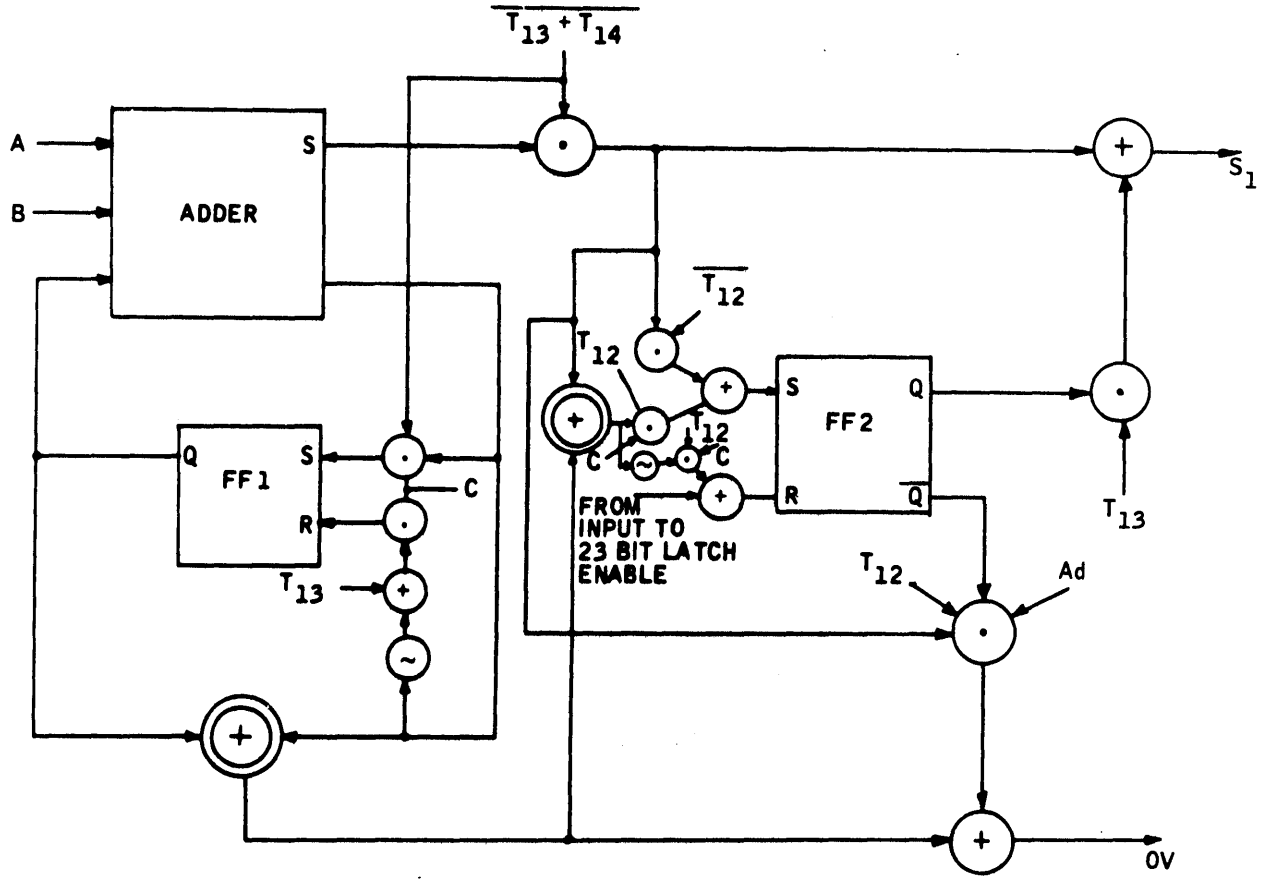

Figure 4-4. First Adder with Overflow Detection

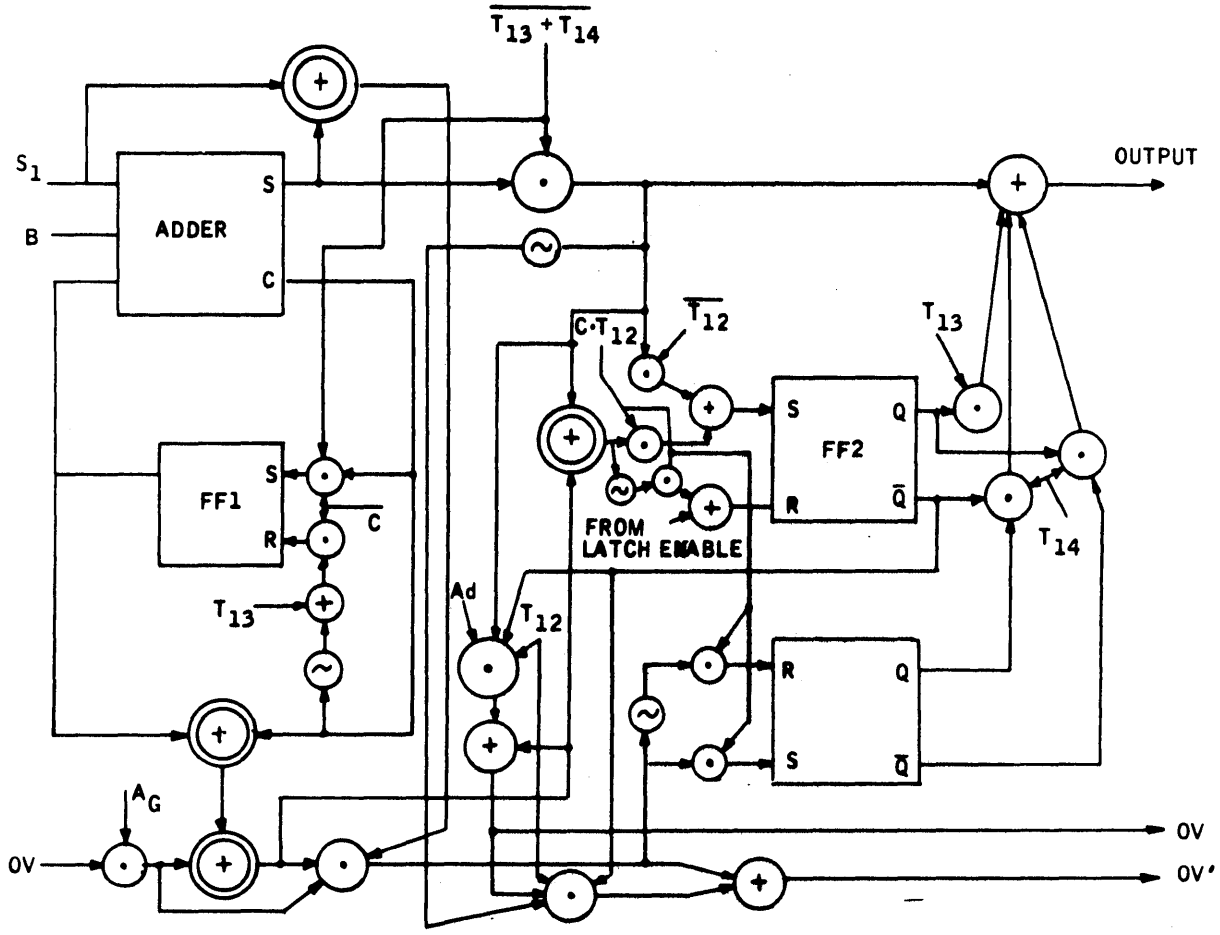

Figure 4-5. Second Adder with Overflow Detection

Table 4-II. Equivalent AND/OR Gates of the Processing Module

Quantity	Description	Estimated Gates
18	12-Bit Shift Registers	432
4	Adders with Overflow Detection	156
6	Complementers	48
3	Flip-Flops and Latches	12
1	23-Bit Shift Register	46
1	23-Bit Latch	46
	Miscellaneous Gates (Appendix B)	151
	Total	891

The module can be housed in a 40-pin package, using 13 pins for outputs and 26 for inputs.

The module can perform 23 basic subinstructions (see Appendix E). A number of these subinstructions are enabled during a given word time to form an instruction. Subinstructions perform the following functions:

- Add for real multiply (AF)
- Add for complex multiply (AG)
- Add for forming sum and difference ($\overline{\text{ADAW}}$)
- Add for forming sum and difference of (A) and the complex conjugate of (B) (ADAW)
- Load Reference and Data register (LDR and LDD)
- Load buffer registers (LDB)
- Output buffer registers (OB)
- Exchange contents of A registers (EXA)

- Various transfers of Data to A registers
- Various transfers of Data to T registers
- Various transfers of Data to D registers

Instructions are received serially by the processing modules into the 23-bit shift register of Figure 4-2. After this register is loaded, the data is transfered in parallel to the 23-bit latch. Each bit of this latch corresponds to one of the subinstructions which may be included in the instruction. An instruction is thus represented by the subinstructions which have a logic "one" stored in the 23 bit latch. While one instruction is being executed, another is being entered serially into the 23-bit shift register of each processing module from the control unit. At the end of each word time the contents of this register is gated into the latch where it presents the proper gate enables for the next word. Note that the logic-enable equations of Appendix C include the appropriate subinstructions as gate inputs.

Prior to modifications to expand the capabilities of DISP, a complete logic level simulation was performed on the processing module design. The simulation verified all logic-enable equations, adder operation, and overflow detection and correction within the module. The functional test written for the module was also verified (see Appendix F). Subsequent changes to DISP leaves the design approximately 95% verified by simulation. The functional test will also have to be expanded to check the new instruction LDTR2.

CONTROL UNIT DESCRIPTION

The processor control unit (Figure 4-6) is not yet designed in detail. The read-only memory will contain the coded instructions of all programs which can be computed by the DISP. The control programs required by DISP (Appendix G) consist of instructions made up of various combinations of the

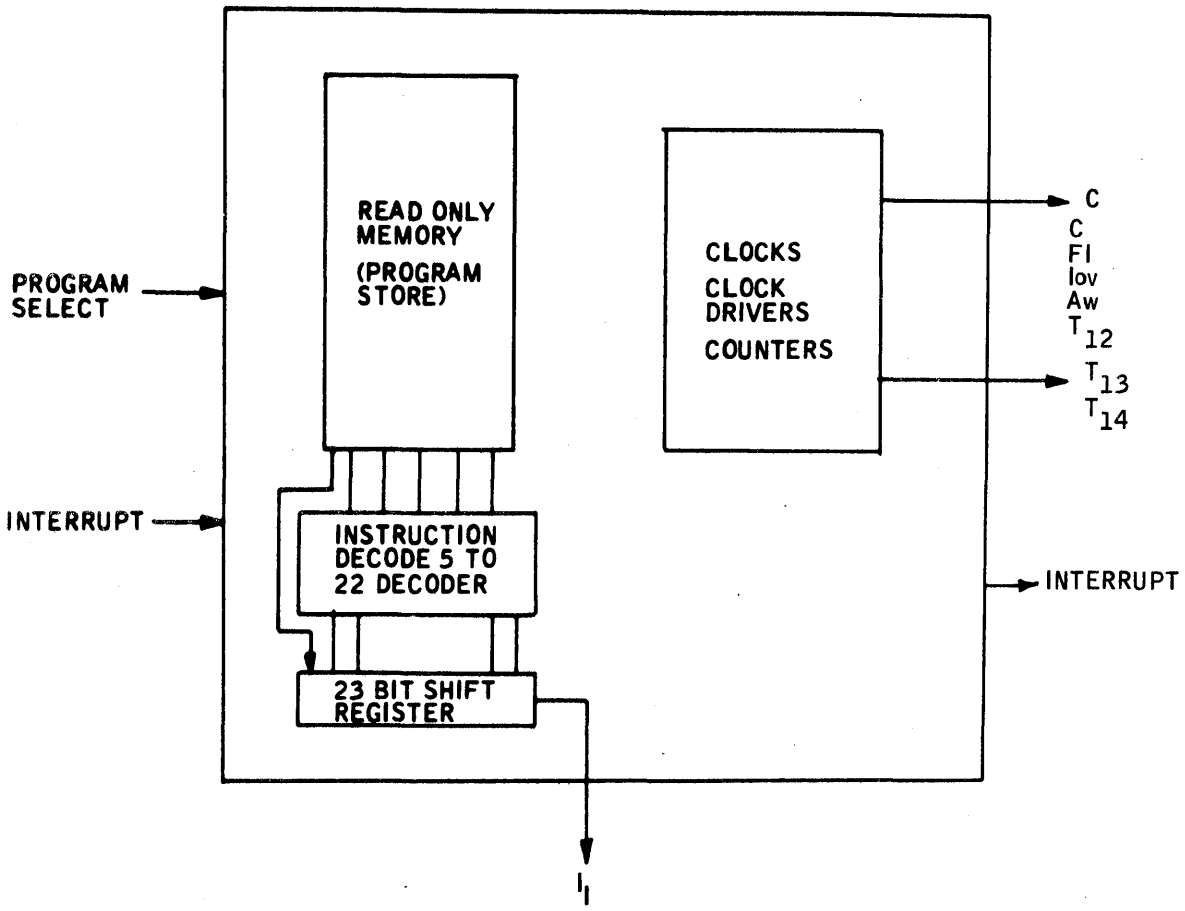

Figure 4-6. Processor Control

23 basic subinstructions listed in Appendix E. The number of unique instructions used in these programs is found to be the 26 shown in Table 4-III.

The only subinstruction not included in any of these instructions is the OB instruction (Output Buffer). It is planned that the program store will consist of a read-only memory of 6-bit words. Five bits will be used to encode the 26 unique instructions, and the sixth bit will be used for OB subinstruction. The number of storage words required will be a function of the number of processing modules in the DISP, and the number of external buffers. In any case, this memory should not exceed 512 words.

The instruction decoder decodes the 5-bit memory word into the proper set of subinstructions and loads them into the shift register. This register then transfers this instruction to all the processing modules simultaneously.

The control unit will also include clocks, drivers, counters and other logic required to generate other outputs to the modules. The processor control will also detect overflow in any module, and notify all modules that such has occurred. A count of overflow occurances is maintained during a computation such that the proper scale factor can be applied to the output. Upon notification that overflow has occurred, each module will scale its data down by one half.

The operation of DISP is determined by a program-select input which defines the area of program storage containing the instructions for the desired computation. This block of memory is sequentialy read out from the memory, decoded and transmitted to all processing modules. Interrupts allow DISP to function in a system containing other devices.

Table 4-III. Control Instructions

Unique Instruction	Arith	A Reg	T Reg	R Reg	Shift	D Reg	B Reg
1	A_g	LDA2					
2	A_g	LDA2	LDT3		SHR1		
3		EXA					
4	A_g	LDA2			SHR1		
5	Ad	LDA5	LDT1				
6	Ad		LDT2				
7		LDA7				LDD	
8	Ad						LDB
9	Ad	LDA6, LDA5					
10		LDA7					
11	Ad					LDD	LDB
12	Af	LDA3					
13	Af	LDA4			SHR1		
14	Af	LDA3			SHR1		
15	Ad	LDA4, LDA8					
16	Ad		LDT5				
17	Ad						
18	A_g	LDA2	LDT4		SHR1		
19			LDT1	LDR			
20				LDR			
21	Ad					LDD1	
22	Ad	LDA4					
23				LDR		LDD	
24						LDD	
25			LDT2				
26				LDTR2			

SECTION V
REFERENCES

1. Bergland, G. D., a guided tour of the fast Fourier Tansform. IEE Spectrum, July 1969, pp. 41-52.

2. Whelchel, J. E., D. E. Guinn, The Fast Fourier-Hadamard Transform and its use in signal representation and classification. EASCON Record Sept. 9-11, 1968, pp. 561-573.

3. Rader, C. M., W. R. Crawther Efficient Coding of Vocoder Chamel Signals using Linear Transformations. Proc. IEEE, Nov. 1966, pp. 1594-pp. 1594-95.

4. Goodman, L. M., A binary Linear Transformation for Redundancy Reduction. Proc. IEEE, Vol 55 No. 3, March 1967, pp. 467-67.

5. Andrews, C. A., J. M. Davies, G. R. Schwartz, Adaptive Data Compression. Proc. IEEE, Vol 55, No. 3, March 1967.

6. Weinstein, C. J. Roundoff Noise in Floating Point Fast Fourier Transform Computation. IEEE Trans. on Audio and Electracoustics. Vol. 17, No. 3, Sept. 1969, pp. 209-215.

7. Liu, B., T. Kaneko, Error Analysis of Digital Filters Realized with Floating Point Arithmetic. Proc. IEEE, Vol. 57, No. 10, Oct 1969, pp. 1735-47.

8. Geokezas, M., Error Analysis of Fast Fourier Transform and Digital Filter Bank, Honeywell Document. SRM-119, June 1970.

APPENDIX A
COMPARING DISP WITH OTHER IMPLEMENTATIONS

APPENDIX A
COMPARING DISP WITH OTHER IMPLEMENTATIONS

The following tables were reproduced from the article in the June 1969 Transactions of IEEE Audio and Electroacoustics by Glen Bergland, "Fast Fourier Transform Implementations, A Survey", pp. 109-117. The DISP characteristics are shown generally for a 128-module processor. Exceptions are: For the case of maximum number of samples, 1024 modules are assumed. The maximum throughput for N = 1024 assumes 512 modules and a clock rate of 1.4 MHz.

TABLE A1

	Bell Telephone Labs.[1]	FFTP	Computer Signal Processors[9]	CSS-3	Computer Signal Processors[9]	CSP-30	Control Data Corp.[3]	FFT Processor	Dept. of Defense[4]	COMP-II	Emerson[6]	Digital Signal Processor	Emerson[6]	MM DSP	IBM[6]	Array Proc. 2938-1	IBM[6]	Array Proc. 2938-2	M.I.T. Lincoln Labs.[7]	FDP	McCullough Engrg.[8]	Continuous Proc. I	DISP
DESIGN STATUS																							
Status																							
Paper Design				×						×	×		×						×				×
Breadboard Model		1–69	7–69								×										×		
Date Operational (Past or Future)	5–67	3–69	9–69[3]	3–69	7–69		9–68[9]		12–69[10]		7–68		8–68		9–70		6–69		1 yr[1]				
Objective																							
Research Model	×				×		×		×						×								
Built for a Specific Application	×						×		×						×				×				×
Production Model		×	×								×		×						×				×
Commercially Available		×	×	×							×		×										
Application																							
Off-Line Signal Processing		×	×								×		×		×								×
Real-Time Signal Processing	×	×	×	×			×		×						×				×		×		×
General Scientific					×						×		×										×
ARCHITECTURE																							
Classification																							
Stand Alone		×	×		×		×		×		×		×								×		×
G P Computer Attachment	×			×	×										×		×		×				×
G P Computer Modification											×												
Other							×		×														×
Structure																							
Sequential	×	×	×	×											×		×		×				
Cascade																							
Parallel-Iterative					×		×		×														×
Array																							
Other																						×	
FUNCTIONAL CHARACTERISTICS																							
Arithmetic Unit																							
Static (or Combinatorial) Multiplier															×		×		×				
Parallel-Iterative Multiplier		×	×	×																	×		
Serial Multiplier																							×
Real Multiplier		×	×	×											×		×		×				×
Complex Multiplier	×				×	×	×														×		×
Log and Log^{-1} Conversions						×	×																
Multiplicand (bits)	12	16	16	16	12	6	6	32[12]		32[12]		32[12]		18		9							12
Multiplier (bits)	12	16	16	16	12	6	6	32[12]		32[12]		32[12]		18		9							12
Product (bits)	12	32	32	16	12	7	7	32[12]		32[12]		36		9									12
Truncated Result		×	×		×	×	×																×
Rounded Result	×		×	×				×		×				×									
Floating Point Numbers									×														
Automatic Scaling																							×

	McCullough Engrg.[5]	Continuous Proc II	Raytheon[9]	SRFFT	Stanford Res. Inst.[14]	Project CRANE	Sylvania[11]	ACP	Sylvania[11]	ASP	Texas Instr.[13]	System I	Texas Instr.[13]	System II	Texas Instr.[13]	System III	Time/Data[15]	Model 100	Time/Data[15]	Model 90	Washington University[16]	MM FFT	Washington University[16]	MM FFT	Westinghouse[18]	FFT-1	Westinghouse[18]	FFT-2	Westinghouse[18]	FFT-3
DESIGN STATUS																														
Status																														
Paper Design	×														×						×		×						×	
Breadboard Model			×																×						×					
Date Operational (Past or Future)			3–69		1–68		3–68		2–69		12–68[47]		3–69[51]		12–69[52]		10–67		6–69		7–69		7–69				12–69		6–70	
Objective																														
Research Model											×		×		×						×		×						×	
Built for a Specific Application			×		×				×		×		×		×										×		×		×	
Production Model							×										×		×		[40]		[40]							
Commercially Available			×				×		×								×		×		[40]		[40]							
Application																														
Off-Line Signal Processing							×		×		×						×		×		×		×							
Real-Time Signal Processing	×		×		×		×		×		×		×		×		×		×		×		×		×		×		×	
General Scientific							×										×		×		×		×							
ARCHITECTURE																														
Classification																														
Stand Alone	×		×				×		×		×		×		×		×		×											
G P Computer Attachment							×		×		×		×				×		×		×		×		×		×		×	
G P Computer Modification					× [38]																									
Other																														
Structure																														
Sequential					×		×														×		×							
Cascade			×																×						×		×		×	
Parallel-Iterative																			×											
Array																														
Other	× [24]						×		× [48]		× [48]		× [48]		× [48]		× [53]													
FUNCTIONAL CHARACTERISTICS																														
Arithmetic Unit																														
Static (or Combinatorial) Multiplier											×		×		×															
Parallel-Iterative Multiplier	×		×				×		×								×		×		×		×							
Serial Multiplier																														
Real Multiplier			×		×		×		×								×		×		×		×							
Complex Multiplier	×										×		×		×										×		×		×	
Log and Log^{-1} Conversions																														
Multiplicand (bits)	15		12		16		12		16 [43]		18		18		12		8		12		24		12		10		10		10	
Multiplier (bits)	15		12		16		12		16 [44]		18		18		12		8		12		24		12		10		10		10	
Product (bits)	15		12		31		12		16 [43]		18		18		12		8/15		12/23		24		12		10 [63]		10 [63]		10 [63]	
Truncated Result			× [29]		×				×												×		×							
Rounded Result	×						×		× [49]		× [49]		× [49]		×		×		×											
Floating Point Numbers									×																					
Automatic Scaling																														

[1] - DISP could be built and tested in a one year program

TABLE A1 *(Cont'd)*

	Bell Telephone Labs.[1] FFTP	Computer Signal Processors[2] CSS-3	Computer Signal Processors[2] CSP-30	Control Data Corp.[3] FFT Processor	Dept. of Defense[4] COMP-II	Emerson[5] Digital Signal Processor	Emerson[5] MM DSP	IBM[6] Array Proc. 2938-1	IBM[6] Array Proc. 2938-2	M.I.T. Lincoln Labs.[7] FDP	McCullough Engrg.[8] Continuous Proc. I	DISP
Arithmetic Unit *(Cont'd)*												
Fixed Point Numbers		×	×	×		×	×	×[12]	×[12]	×		×
Fixed Point with Common Exponent	×				×						×	
One's Complement				×								
Two's Complement	×	×	×		×	×	×	×[12]	×[12]	×		×
Sign Magnitude								×[12]	×[12]		×	×
Number of Real Multipliers/A.U.	4	1	1	1	4	4	4	1	1	1	4	N
Number of Arithmetic Units	1	1	1	1	1	4	4	1	1	4	1	N
Technology Used	DTL	DTL	TTL-MSI	TTL	TTL	TTL	TTL	DTL	DTL	MECL	TTL	MOS
Logic Characterized by Average Propagation Delay/Node (ns/node)	30			5	10	5	5	5–8	5–8	7.5		
Clock Rate (MHz)	5	2	10	10	25	5	5	5	5	6.6	10	1
Algorithms												
Cooley–Tukey (Decimation in Time)	×	×	×		×			×[13]	×[13]	×		
Sande–Tukey (Decimation in Frequency)		×	×	×		×	×	×[13]	×[13]	×		
Danielson–Lanczos										×		
Radix-2	×	×	×	×	×	×	×	×[13]	×[13]	×		
Radix-4		×	×					×[13]	×[13]	×		
Mixed Radix		×	×					×[14]	×[14]	×		
Other								×[15]	×[15]	×	×[37]	×
Internal Control												
Hard-Wired	×			×	×	×	×				×	
Microprogrammed								×[11]	×[11]			×
Software		×	×			1/2	×			×		
Macromodular												
Other Properties												
Stored Trig. Coefficients	×	×	×	×	×	×	×	×	×	×	×	×
Computed Trig. Coefficients										×		
In-Place Reordering		×	×	×						×		
Reordering on I/O	×	×	×			×	×			×	×	
Other					×[7]			×[16]	×[16]			×
Batches of Data Sent to Processor	×	×	×	×						×		×
Stream of Data Sent to Processor		×	×			×	×	×	×	×	×	×

	McCullough Engrg.[8] Continuous Proc. II	Raytheon[9] SRFFT	Stanford Res. Inst.[10] Project CRANE	Sylvania[11] ACP	Sylvania[11] ASP	Texas Instr.[13] System I	Texas Instr.[13] System II	Texas Instr.[13] System III	Time/Data[15] Model 100	Time/Data[15] Model 90	Washington University[14] MM FFT	Washington University[14] MM FFT	Westinghouse[16] FFT-1	Westinghouse[16] FFT-2	Westinghouse[16] FFT-3
Arithmetic Unit *(Cont'd)*															
Fixed Point Numbers		×	×	×	×				×	×	×	×	×[62]	×[62]	×[62]
Fixed Point with Common Exponent	×										×	×			
One's Complement		×[30]													
Two's Complement			×	×	×	×	×	×	×	×	×	×			
Sign Magnitude	×	×[31]													
Number of Real Multipliers/A.U.	4	1	1	1	4	1	1	1	2	2	1	1	4	4	4
Number of Arithmetic Units	20	$\text{LOG}_2 N$	1	1	1–8	1	1	1	1	1	1	1			
Technology Used	TTL-LSI	TTL		TTL	TTL	TTL	TTL	TTL	CTL/DTL	TTL	ECL	ECL			
Logic Characterized by Average Propagation Delay/Node (ns/node)		6		10	10	10	10	10	15/45	29	4	4			
Clock Rate (MHz)	10	1.5	6	15	5	2.85	2.85	2.85	5/2.5	2	none	none			
Algorithms															
Cooley–Tukey (Decimation in Time)		×	×	×[40]	×[40]	×	×	×			×	×	×	×	×
Sande–Tukey (Decimation in Frequency)				×[40]	×[40]										
Danielson–Lanczos				×[40]	×[40]				×[54]	×[54]					
Radix-2		×	×	×[40]	×[40]	×	×	×			×	×	×	×	×
Radix-4				×[40]	×[40]										
Mixed Radix				×[40]	×[40]										
Other	×[37]			×[40]	×[40]	×[50]	×[50]	×[50]							
Internal Control															
Hard-Wired	×	×[28]				×	×	×	×	×			×	×	×
Microprogrammed					×										
Software			×	×	×										
Macromodular											×	×			
Other Properties															
Stored Trig. Coefficients	×	×	×	×[40]	×[40]	×	×	×	×	×	×	×	×	×	×
Computed Trig. Coefficients				×[40]	×[40]	×	×	×							
In-Place Reordering				×[40]	×[40]				×						
Reordering on I/O	×	×	×	×[40]	×[40]	×	×	×		×	×	×	×	×	×
Other															
Batches of Data Sent to Processor				×[40]	×[40]	×	×	×	×	×	×	×			
Stream of Data Sent to Processor	×	×	×	×[40]	×[40]				×	×			×	×	×

(Cont'd)

1D-C-1

TABLE A1 (Continued)

	Bell Telephone Labs.[1] FFTP	Computer Signal Processors[2] CSS-3	Computer Signal Processors[2] CSP-30	Control Data Corp.[3] FFT Processor	Dept. of Defense[4] COMP-II	Emerson[5] Digital Signal Processor	Emerson[5] MM DSP	IBM[6] Array Proc. 2938-1	IBM[6] Array Proc. 2938-2	M.I.T. Lincoln Labs.[7] FDP	McCullough Engrg.[8] Continuous Proc. I	DISP
PERFORMANCE CHARACTERISTICS												
Timing												
Execution Time for $N=1024$ (ms)[A]	31	600	27	22	10	10	1	56[17]	31[17]	3.75		1
Execution Time $=k\ N \log_2 N$ (μs)[A]; $k\simeq$	3	59	2.7–3.3	2.2	1	1	0.1	5.5[17]	3[17]	0.366		
Throughput for $N=1024$[B] Max (Complex Samples/Second)	24 000	1560	26 400	36 300	100 000	1 000 000	1 000 000	18 000[17]	33 000[17]	200 000	1 000 000	10^6
Execution Time One Radix-2 Basic Operation (μs)	6	170	9	4.4	1.3	0.5	0.5	22[18]	12[18]	0.75		156
One Radix-4 Basic Operation (μs)		490	28					44[17]	24[17]			
Precision												
Bits Input (Fixed Point)	12	16	16	16	12	6	6			18	9	12
Bits Output (Fixed Point)	12	16	16	16	12	12	16			18	9	12
Bits Mantissa (Floating Point)								24[12]	24[12]			
Ratio of rms Error/rms Result for Random Number Input	0.001									[21]		0.001
FUNCTIONS PERFORMED (H—Hardware; HS—Hardware Aided by Software; S—Software Only)												
Fast Fourier Transform	H	S	S	H	H	H	HS	HS[13]	HS[13]	S	H	H
Inverse Fast Fourier Transform	HS	S	S	H	H			HS[13]	HS[13]	S	H	H
Convolution	HS	S	S	H	H			H	H	S		H
Correlation	HS	S	S	H	H	H	HS	H	H	S	H	H
Weighting Input by an Arbitrary Data Window	S	S	S	H	H	H	HS	H	H	S		H
Weighting Input by a Specific Data Window		S	S	H						S		H
Squaring to Form Power Spectrum	S	S	S	S	H			H	H	S	H	H
Recursive Filtering	S	S	S	S	HS			H	H	S		H
Vector Multiplications	S	S	S	S	H	H	HS	H	H	S		
Multidimensional FFT	S	S	S	HS		H	HS	HS[13]	HS[13]	S		
Spectral Averaging	S	S	S	S	H	H	HS	HS	HS	S		
Convolutional Form of FFT (Bluestein)	S	S	S		HS			HS	HS	S		
Expanded Resolution in Any Range	S	S	S		HS	H	HS	HS	HS	S		
Diagnostic Tests	HS	S	S	S	HS	H	HS	H	H	S		
Other		S	S	H[6]				[15]	[15]			HS

	McCullough Engrg.[8] Continuous Proc. II	Raytheon[9] SRFFT	Stanford Res. Inst.[10] Project CRANE	Sylvania[11] ACP	Sylvania[11] ASP	Texas Instr.[12] System I	Texas Instr.[12] System II	Texas Instr.[12] System III	Time/Data[12] Model 100	Time/Data[12] Model 90	Washington University[14] MM FFT	Washington University[14] MM FFT	Westinghouse[15] FFT-1	Westinghouse[15] FFT-2	Westinghouse[15] FFT-3
		8	215	45	0.19	9	9	9		25	51	51	4	4	0.5
		$8\div\log_2 N$	21	4.4	$1.5\div$ #A.U.[44]	0.9	0.9	0.9	30	2.5	5	5	0.39	0.39	0.049
1 000 000		125 000		20 800	66 000· #A.U.	56 832	56 832	56 832	1500	16 000	10 000	10 000	250 000	250 000	2 000 000
		8[23]		13	3[44]	1.92	1.92	1.92	8	4					
15		12	14[37]	12	16[42]	8	8	8	8	12	24	12	10	10	10
15		12	8[28]	12	16[42]	18	12	12	16	12/24	24	12	10	10	10
					16[42]										
				[40]	[40]										

H	H	S	S	S	H	H	H	H	H	H	H	H	H	H
H	H	S	S	S				H	H	H	H	H	H	H
H			S	S				H	HS	HS	HS	H	H	H
H	H		S	S				H	HS	HS	HS	H	H	H
		S	S	S	H	H	H	H	HS	S	S	H	H	H
H	H	S	S	S				H	H	S	S			
H	H	S	S	S		H	H	H	H	S	S	H	H	H
			S	S						S	S	H	H	H
			S	S				H		H	H	H	H	H
	H		S	S				HS	HS	H	H	H	H	H
H	S		S	S	H		H	H	H	H	H	H	H	H
			S	S								HS	HS	HS
H		S	S	S				HS	HS			HS	HS	HS
		S	S	S				H	H			H	H	H
								H[55]	H[67]					

(*Cont'd*)

1D-C-1

TABLE A1 (Concluded)

	Bell Telephone Labs.[1] FFTP	Computer Signal Processors[2] CSS-3	Computer Signal Processors[2] CSP-30	Control Data Corp.[3] FFT Processor	Dept. of Defense[4] COMP-II	Emerson[5] Digital Signal Processor	Emerson[5] MM DSP	IBM[6] Array Proc. 2938-1	IBM[6] Array Proc. 2938-2	M.I.T. Lincoln Labs.[7] FDP	McCullough Engrg.[8] Continuous Proc. I	Raytheon[9] Continuous Proc. II	Raytheon[9] SRFFT	Stanford Res. Inst.[10] Project CRANE	Sylvania[11] ACP	Sylvania[11] ASP	Texas Instr.[12] System I	Texas Instr.[12] System II	Texas Instr.[12] System III	Time/Data[13] Model 100	Time/Data[13] Model 90	Washington University[14] MM FFT	Washington University[14] MM FFT	Westinghouse[15] FFT-1	Westinghouse[15] FFT-2	Westinghouse[15] FFT-3	DISP
SYSTEM HARDWARE FEATURES																											
Maximum Value of N Processed	8192	8192	16 384	4096	1024	64	16 384	32 768[19]	32 768[19]	512[25]	16	1024	512[25]	4096[39]	1024	65 536	1024	4096	4096	1001	2048	4096	4096	2048	2048	2048[44]	2048
Internal Buffer Size (Words)	8192	32 768[1]	65 536[1]	8192	8192			[20]	[20]	2048	32	1536	4096		2048	4096 – 65 536	16 384	16 384	16 384	4096	2048	4096	4096	2048	2048	2048[46]	4N
Internal Word Size (Bits)	24	16	16	16	48	6/18	6/18	[20]	[20]	18	18	1020	12	16	12	8–20	24	24	24	8/18	12	24	12	10	10	10	12
Multiplexed I/O Channels		8–64	8–64		2[3]		6			2	q		8	16	[41]	[41]	40	3	8	2	2	[61]	[61]				*
A/D Bits Converted	9–14	8–15	8–15		12	6	7–8				9	15	9	14	[41]	[41]	8	8	8	8	10	[61]	[61]	10	10	10	
A/D Samples/Second (Maximum)	20 000	200 000	10 000 000		1 000 000	7 000 000	10 000 000				1 000 000[25]	1 000 000[25]	1 000 000	25 000	2 330 000	1 660 000				20 000	100 000	[61]	[61]	250 000	250 000	2 000 000	
D/A Bits Converted	16	8–15	8–15		8	3	6				9	15	10	8	[41]	[41]				10[58]	10[58]	[61]	[61]	10	10	10	
Volume of Processor (ft³)	3	24[2]	25[4]	52	50	3	3	153	153				24		25	50				65	6	10	10	4	1		.05R
Weight (lb)		350		750	1000	75	75	3450	3450				250		500	1000				1400	400	heavy	heavy				5R
Max. Ambient Temp. (°F)	140	120	120		167	150	150	90	90				140		100	120				110	110	105	105				
Min. Ambient Temp. (°F)	32	40	40		32	0	0	60	60				0		40	40				50	50	60	60				
Circuit Count (Equivalent Gates)	4500				7000	5000	5000			100 000			3100[34]		65 000	48 000				12 500	5400	10 000[62]	10 000[62]				128,000R
Power Consumption (Watts)	1200	450			175	150	150	9500	9500				750		1000	4000				1250	500	3000	2500				40R
COST																											
Cents/1024 Point Complex Transform[C]	0.04¢	0.08¢	0.01¢		0.01¢			0.056¢[21]	0.034¢[21]	0.0015¢			0.03¢		[42]	[42]				0.064¢	0.0033¢[59]	[61]	[51]	[42]	[42]	[42]	
Monthly Rental, Processor								$9810[22]	$10 460[22]						[42]	[42]								[42]	[42]	[42]	
Purchase Price, Processor Only			$85 000					$356 550[22]	$397 900[22]				$100 000[35]		[42]	[42]								[42]	[42]	[42]	
Purchase Price, Entire System		$45 000	[5]			$100 000	$100 000	[22]	[22]					$125 000	[42]	[42]				$66 925	$45 000	$50 000	$40 000	[42]	[42]	[42]	
Approximate 1968 Parts Cost of Processor (If machine is not commercially available)	$70 000				$150 000	$20 000	$20 000			$100 000	$4000		$20 000		[42]	[42]								[42]	[42]	[42]	
Monthly Rental Entire System								[22]	[22]						[42]	[42]								[42]	[42]	[42]	

* Variable

$$R = \frac{N}{256}$$