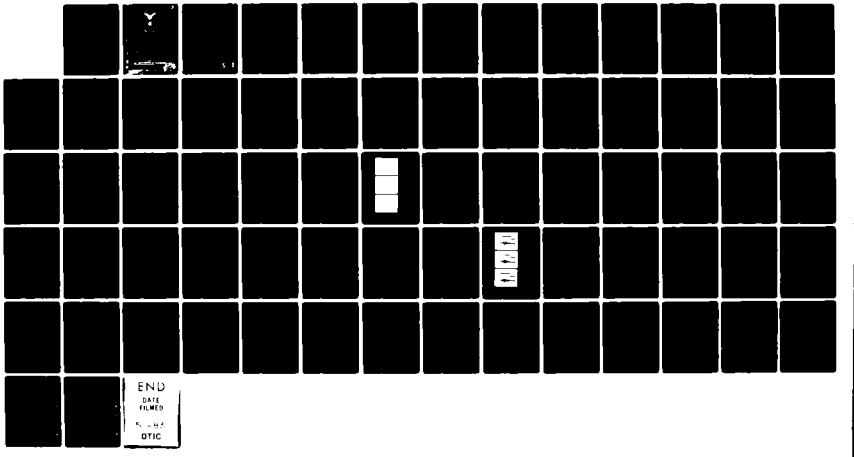
AD-A127 434 DIGITAL CORRELATION OF SPREAD - SPECTRUM SIGNALS VIA FREQUENCY DOMAIN PROCESSING(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. J A CARRETTO 1/1

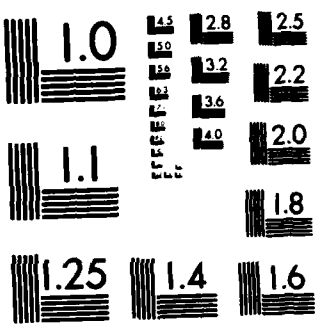
UNCLASSIFIED JUL 82 AFIT/GE/EE/82S-15 F/G 17/2 NL

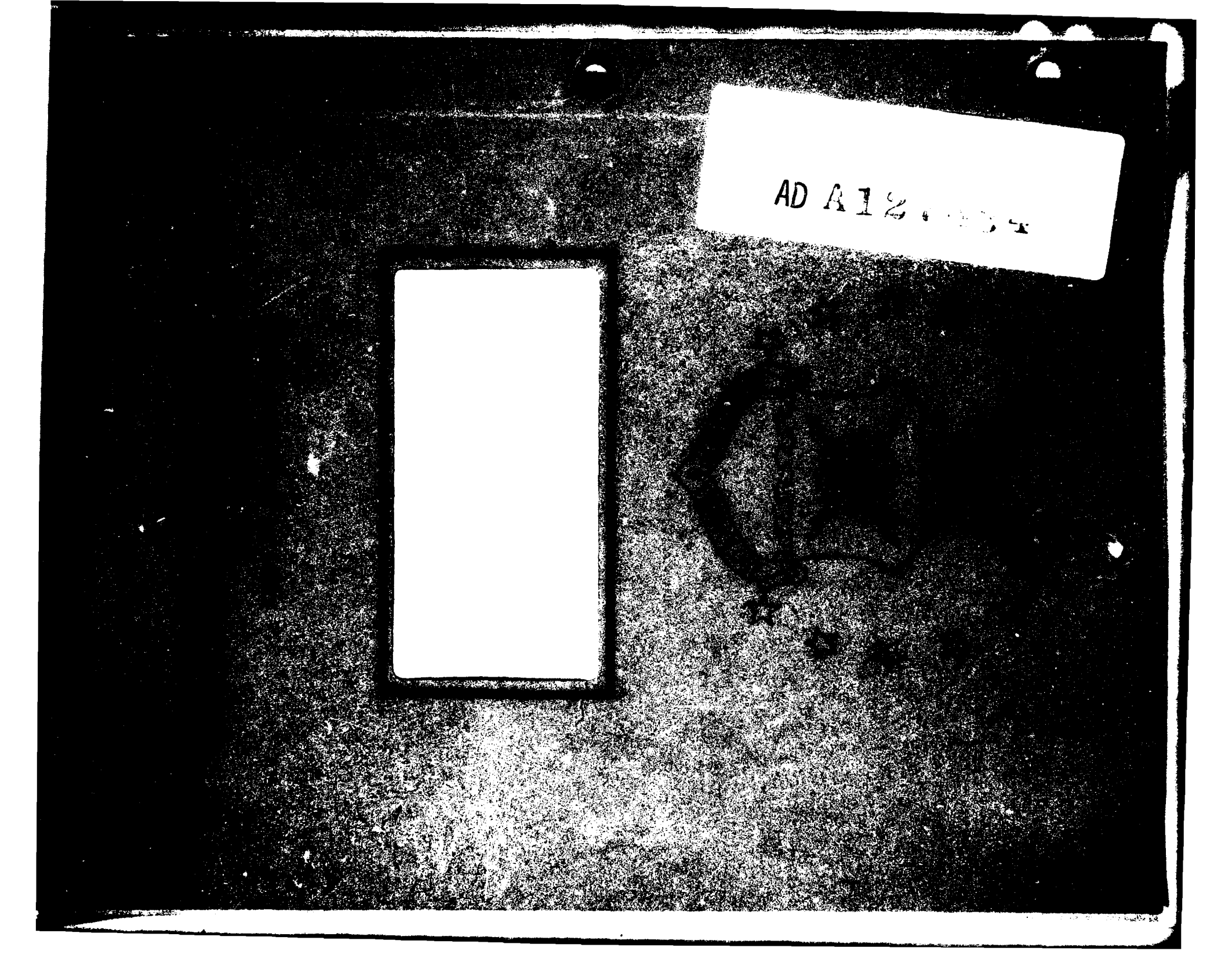


END

DATE FILMED

DTIC





AFIT/GE/EE/82S-15

DIGITAL CORRELATION OF SPREAD - SPECTRUM SIGNALS VIA FREQUENCY DOMAIN PROCESSING

THESIS

Joseph A. Carretto, Jr
1Lt USAF
AFIT/GE/EE/82S-15

AFIT/GE/EE/82S-15

DIGITAL CORRELATION OF SPREAD - SPECTRUM SIGNALS VIA FREQUENCY DOMAIN PROCESSING

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in Electrical Engineering

by
Joseph A. Carretto, Jr, B.S.E.E.
1Lt USAF
Graduate Electrical Engineering
July 1982

Preface

Spread-spectrum anti-jamming techniques are becoming increasingly common in military radio systems. Current methods of processing them center around time domain correlation schemes. An alternative approach is to process the correlation function in the frequency domain as a product of Fourier transforms. Digital processing techniques are then used to extract the correlation data.

This report describes the use of the Frequency Domain Processing (FDP) techniques to process simulated spread-spectrum signals. The emphasis in this project was on the development of hardware to validate the FDP concepts. A complete description of both the hardware and the results obtained with it is included in this report.

I wish to thank my thesis advisor, Major Ken Castor, for his outstanding support throughout this effort. Without his committment and dedication, this report would never have been completed. Thanks are also due to Major Larry Kizer and Lt Col Robert Edwards for their support and responsiveness in a very time critical situation. Finally, I wish to thank my supervisor, Mr Robert Witters, who helped convince management of the need to begin this project, and gave me the freedom and encouragement to finish it.

Joseph A. Carretto, Jr

CONTENTS

List of Figures

Abstract

Backround information on the theory behind processing spread-spectrum synchronization data in the frequency domain is presented. The derivation shows that the time domain psuedo-noise sequence offset is represented in the frequency domain by frequency modulation on a known carrier wave. Following the backround development is a detailed description of the analog hardware used to implement the frequency domain processing techniques. A proposed method of digitally extracting the synchronization data by Fourier transforming the analog output signal is presented. The digital processor designed and built to sample, average, and calculate a Fast Fourier Transform of the analog data is described in detail. Results obtained from both the analog and digital hardware are discussed, comparing the actual results to those predicted for various inputs. The results validate the processing concepts, and indicate areas for further research.

I: Introduction

Under a contract with the Avionics Laboratory, the Charles Stark Draper Laboratory (CSDL) developed a new technique for processing spread-spectrum radio signals, specifically those used in the Global Positioning System (GPS). In the type of spreading used for GPS, the data signal is modulated by a wide bandwidth, psuedo-random code in order to spread the transmitted energy over a band of frequencies. To decode the signal, it must be correlated with a locally generated reference code, which is a duplicate of the transmitted spreading code. This is necessary to synchronize the two signals, and permits decoding the transmitted data. The correlation is normally performed in the time domain as a bit by bit comparison of the received and reference codes. The CSDL approach is to first convert the input and reference signals into their frequency domain representations, through Fourier transformation, and then perform the correlation as a multiplication of the two transforms. Because of the method used for Fourier transformation, the code synchronization data can be extracted from the transform products as a frequency shift of a known carrier signal. This frequency shift information is then used to close the code tracking loops and extract the transmitted data.

Implementing the spread-spectrum tracking loop in this manner is a three fold problem. First of all, some method of forming the Fourier transforms of the received and reference signals is required. Digital computation methods are impossible due to the high frequencies and large bandwidths of the GPS signals. Once the transforms are obtained and multiplied, the product must be sampled and processed to calculate the frequency shift information. As will be shown, this can be accomplished with digital processing techniques, since the frequency and bandwidth are reduced as a result of the transform generation and multiplication. Finally, the digital correlation information must be used to close the tracking loops, synchronize the received and reference codes, and decode the transmitted data.

The effort of this thesis is primarily in the second area: calculating the frequency shift information that represents the code synchronization data. As will be shown in Chapter IV, this requires sampling and averaging the transform product information, calculating a digital Fourier transform, and extracting the frequency shift information from the result. Specifically, this thesis effort involved the design and construction of hardware for sampling the transform product, averaging the samples, and calculating the frequency shift information through a Fast Fourier Transform (FFT) algorithm programmed on a general purpose personal computer. In addition, hardware for producing the simulated GPS received and reference signals was designed and constructed. In direct

support of this effort, hardware for generating the Fourier transforms and their product was constructed, based upon designs developed by CSDL under Air Force contract F33615-78-C-1563 with the Avionics Laboratory. Modifying that design and building the transform generation hardware was a joint effort involving other Avionics Laboratory engineers.

Available hardware was a major constraint throughout this effort, and as a result, several simplifications were made in processing the simulated GPS signals. Due to bandwidth limitations of the transform generation hardware, the simulated GPS psuedo-random codes had a bandwidth of 10 Mhz, instead of the 20 Mhz bandwidth of actual GPS codes. Additionally, the simulated signals did not allow for signal degradation. In other words, the 'received' signal was at full power, with no transmission loss or jamming interference. Finally, because of the slow processing speed of the general purpose computer used to calculate the FFT, the frequency shift information was not provided in the real time necessary to close the tracking loops of an actual GPS receiver.

In summation, this thesis effort validates the theoretical concept of performing correlation in the frequency domain. Hardware was designed and constructed to generate the frequency domain correlation information and to digitally process that data to provide the desired code synchronization information. The results demonstrate the feasibility of using frequency domain processing techniques for processing spread-spectrum signals.

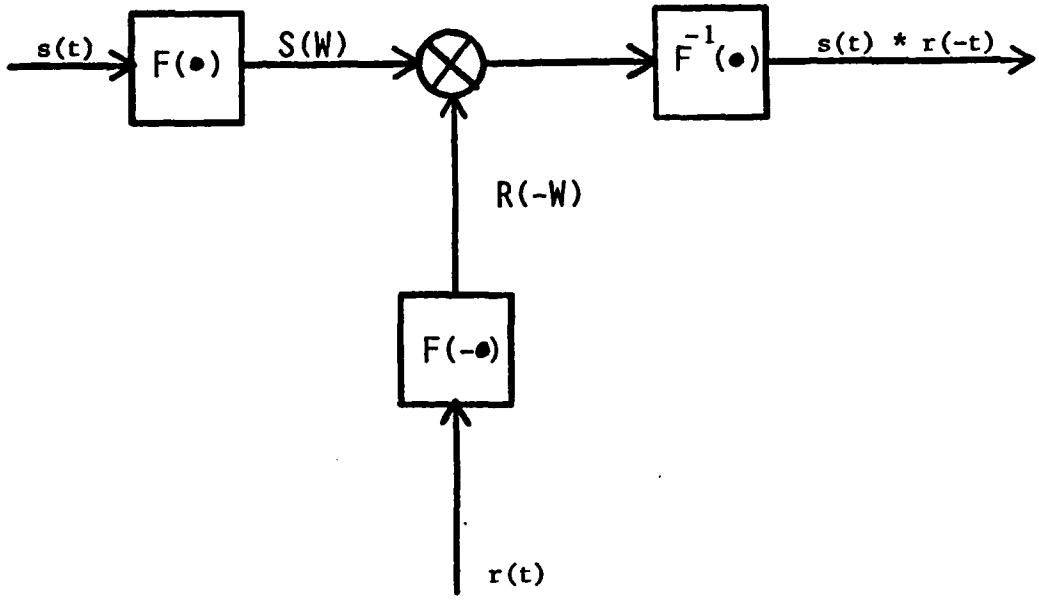
The remainder of this thesis is divided into five sections. The theory behind the generation of the Fourier transforms, and the method used to derive correlation data from those transforms, is discussed briefly in Chapter II. The actual analog hardware used to demonstrate that theory is described in detail in Chapter III. Chapter IV discusses the digital processing hardware built to sample, average, and Fourier transform the correlation data from the transform product produced by the analog hardware. The results of the effort are described in Chapter V, and the final chapter summarizes the effort and provides recommendations for further research.

II: Frequency Domain Correlation

Under contract F33615-78-C-1563 with the Avionics Laboratory, the Charles Stark Draper Laboratory developed a technique for accomplishing synchronization of spread-spectrum signals using frequency domain processing techniques. The detailed results of that effort were published in an Air Force technical report (ref 4). This chapter presents a brief review of those concepts. For a more detailed discussion, the reader is referred to that report.

The SAW as a Fourier Transformer

The use of the frequency domain to perform the correlation function on spread-spectrum signals is not a new concept. Recognizing that correlation is a point by point comparison of two signals, it is equivalent to convolution, except that one of the signals is reversed in time with respect to the convolution process. The input and reference signals are Fourier transformed, (one of them is spectrally inverted) and their product is inverse transformed to provide the correlation function. This process is shown in Figure 1. (Note that in Figure 1, and throughout the remainder of this report, the symbol <==> indicates a Fourier transform pair.) Conceptually, several different methods can be used to calculate the transforms and the inverse. However, when



CORRELATION: $R_{sr}(\tau) = \int s(t)r(\tau+t)dt$

CONVOLUTION: $s(\tau) * r(\tau) = \int s(t)r(\tau-t)dt$

$$R_{sr}(\tau) = s(\tau) * r(-\tau)$$

$$s(\tau) * r(\tau) \Leftrightarrow S(W)R(W)$$

$$s(\tau) * r(-\tau) \Leftrightarrow S(W)R(-W) \quad \text{for real } r(\tau)$$

$$R_{sr}(\tau) \Leftrightarrow S(W)R(-W)$$

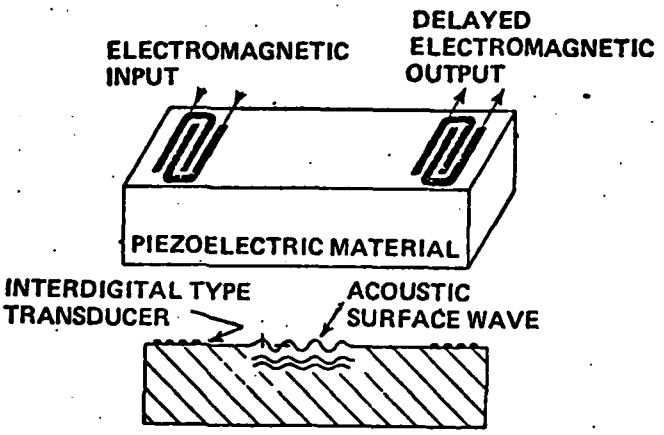
FIGURE 1 - CORRELATION IN THE FREQUENCY DOMAIN

working with wide band or high frequency signals, some methods are impractical. A digital Fast Fourier Transform, for example, cannot process the 20 Mhz Global Positioning System satellite code fast enough to meet system requirements. Other transform methods must be used.

The use of Surface Acoustic Wave (SAW) devices to perform the Fourier transformation has been discussed in numerous publicatons (ref 4, 10, 13). As the name implies, the SAW is an acoustic device, a solid piezoelectric crystal that propogates electrical signals as physical deformations (waves) along its surface (Figure 2a). The response of the SAW to input signals is controlled by the properties of the crystal, including any etchings or deposits on its surface. One type of SAW often used is the Reflective Array Compressor (RAC). A RAC has an etched herring bone pattern on its surface, providing frequency selective reflection of the surface waves (Figure 2b). Its response is such that an input signal is decomposed into frequency components that are output sequentially in time. While this does not result in the Fourier transform of the signal, it does provide the basis for it.

If the RAC is excited with an impulse, a linear frequency-modulated (FM) waveform, or "chirp", results at the output. The chirp begins at some frequency W_1 and increases (or decreases) at a constant rate to frequency W_2. The time it takes to sweep over the entire bandwidth W_1 to W_2 is a function of the length of the crystal, and is called the

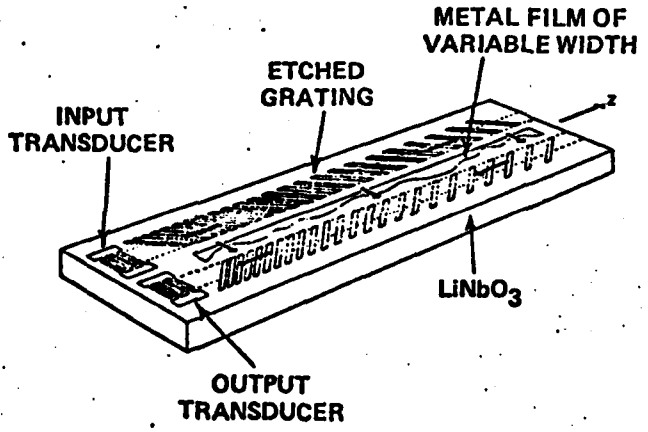
THE LAUNCHING AND PROPAGATION OF A SURFACE ACOUSTIC WAVE



FROM: SLOBODNIK, PROCEEDINGS OF IEEE, Vol. 64, No. 5, May 1976, p. 581.

Figure 2a

REFLECTIVE ARRAY COMPRESSOR (RAC)



FROM: WILLIAMSON, PROCEEDINGS OF IEEE, Vol. 64, No. 5, May 1976, p. 707.

Figure 2b

FIGURE 2: SURFACE ACOUSTIC WAVE DEVICES

dispersion time of the device, T_c. The slope of the linear FM sweep (chirp) is defined as

$$u = \frac{W_1 - W_2}{T_c}$$

so that the instantaneous frequency of the chirp is $W_1 + ut$. The impulse response of the RAC is given by

$$h(t) = \cos(W_1 + ut)t$$

or in complex notation, neglecting the carrier W

$$h(t) = e^{jut^2}$$

Note that this impulse response of the RAC is fixed - i.e. constant. One of the primary advantages to this frequency domain processing scheme is the use of fixed response devices in place of the programmable impulse response devices currently used for correlation. The fixed response devices are inherently simpler to manufacture, and consequently cheaper and more reliable than their programmable counterparts.

The generation of the chirp waveform is only the first step in the Fourier transform process. Two chirp filters, of opposite slopes, are needed to complete the process. If a bandlimited signal, s(t) of duration T is first multiplied by one chirp waveform (a pre-chirp) and then convolved with a

chirp of opposite slope, the Fourier transform of s(t) results (Figure 3).

The complete development of the equations for generating the Fourier transform with SAW devices is beyond the scope of this presentation. However, by using complex notation for the modulation components and neglecting the carriers and delays through the components, the derivation is greatly simplified. From Figure 3, the output of the filter may be written as

$$b(t) = \left[s(t)e^{-jut^2}\right] * e^{jut^2}$$

By replacing the convolution operation with its integral equivalent

$$b(t) = \int s(\tau)e^{-ju\tau^2}e^{ju(t-\tau)^2}d\tau$$

$$= e^{jut^2}\int s(\tau)e^{-2jut\tau}d\tau$$

If the integral above is evaluated only during the time of valid output of the SAW device, then the integral can be written as

$$b(t) = e^{jut^2}\int_{-T_c/2}^{T_c/2}\left(s(\tau)e^{-2jut\tau}d\tau\right)$$

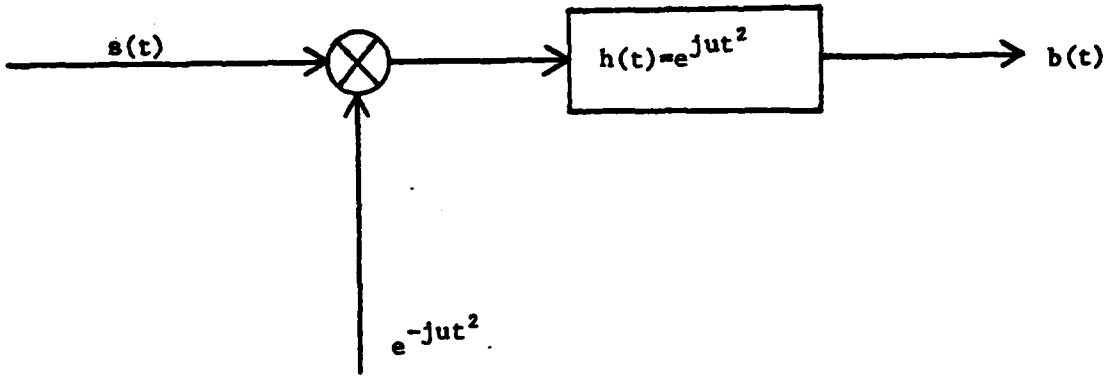


FIGURE 3 - THE FOURIER TRANSFORM PROCESS

Letting w=2ut, the integral can be recognized as the Fourier transform of the T-length segment of the input signal s(t), multiplied by a residual chirp. The residual could be removed by post multiplication, but this is not necessary, as will be shown later. Note that in actuality, w=ut and not 2ut as used in the substitution above. Since u is a constant of the device, the inconsistency can only be resolved with a change of variable t=2t i.e. a change in the time scale. This results in a restriction on the validity of the transform output. Instead of being valid over a time period equal to the dispersion time of the device, the Fourier transform is valid only half of that time, or $T_c/2$. Therefore, in order to output a 20 microsecond transform record, SAW devices with dispersion times of 40 microseconds must be used.

The output of the chirp transform is a time serial record of the magnitude and phase components of the Fourier transform of the input signal. If the chirp transform is implemented at bandpass frequency, the magnitude and phase components appear as magnitude and phase modulation of the transform carrier frequency. At baseband, the in-phase (I) and quadrature (Q) components are the real and imaginary parts of the Fourier transform, respectively.

Correlation in the Frequency Domain

A fundamental property of signal processing is that convolution in the time domain can be performed as a multiplication in the frequency domain of the Fourier

transforms of the two time signals. The inverse transform of the product is identical to the convolution sum. As shown in Figure 1, the process of correlation in the frequency domain is almost identical to convolution, except that one of the time signals is reversed relative to the other. An alternative approach to time reversal is the use of spectral inversion. A well known property of the Fourier transforms says that if $g(\tau)$ is real and

$$g(\tau) \quad <==> \quad G(W)$$

then

$$g(-\tau) \quad <==> \quad G(-W)$$

In other words, if

$$s(\tau) * r(\tau) \quad <==> \quad S(W)R(W)$$

then

$$R_{sr}(\tau) = s(\tau) * r(-\tau) \quad <==> \quad S(W)R(-W)$$

The quantity R(-W) can easily be generated from R(W) by multiplication with a local oscillator and bandpass filtering, i.e. by spectral inversion.

The above analysis is valid for signals in general. The GPS signal, and in fact all psuedo-random spread-spectrum signals, have specific properties that further simplify the correlation process. In psuedo-random spreading, the two signals to be correlated are nearly identical, the only

difference being an unknown time delay bewteen them (neglecting noise and signal strength differences).

Let $c_{ref}(t)$ be the reference code signal and $c_{rec}(t)$ be the received code signal. The relationship between them is then given by

$$c_{ref}(t) = c_{rec}(t-d)$$

which transforms to the frequency domain as

$$C_{ref}(W) = C_{rec}(W)e^{-jwd}$$

or

$$C_{rec}(W) = C_{ref}(w)e^{jwd}$$

where d is the time delay between the two signals.

The purpose in expressing the two time signals in the frequency domain is to find the correlation between them as a function of their frequency domain representations. Letting $R(\tau)$ be the correlation function, we want to find

$$R(\tau) = c_{ref}(-\tau) * c_{rec}(\tau)$$

If M(w) is defined to be the Fourier transform of the correlation function, such that

$$R(\tau) \Longleftrightarrow M(W)$$

then

$$M(w) = C_{ref}(-W) \times C_{rec}(W)$$
$$= C_{ref}(-W) \times C_{ref}(W)e^{jwd}$$

The time domain signal $c_{ref}(t)$ is a real signal, and so its transform $C_{ref}(W)$ will be an even function, such that

$$C_{ref}(-W) = C_{ref}(W)$$

Therefore,

$$M(W) = C_{ref}(W)C_{ref}(W)e^{jwd}$$
$$= C^2_{ref}(W)e^{jwd}$$

In this case, the desired correlation is in fact the autocorrelation function of the signal $c_{ref}(t)$.

Recalling that the chirp transform outputs in time serial, and that the frequency of the chirp, w, equals ut at any given moment, the output of the transform multiplier is just

$$M(W) = C^2_{ref}(W)e^{jutd}$$

This represents a sine wave out of the multiplier, with envelope equal to the PSD of the input signal, and with carrier frequency deviation proportional to the time delay in the input and reference signals. Thus, the task of estimating the time delay between input and reference signal has been reduced to measuring a frequency shift of the carrier.

Finite Length Restrictions

The above analysis is strictly valid only for infinite length time records. The SAW transformer, however, operates on finite length records, and outputs a transform valid over a finite time period. Therefore, the results are slightly different than those presented above. Instead of the full autocorrelation of the signal, each output of the multiplier is only a single occurrence of a random process, the expected value of which represents the full autocorrelation function. If the multiplier output is averaged over sufficiently large samples, the result will be the full autocorrelation with probability approaching one.

Once the transform product is available, signal processing techniques can be used to estimate the frequency deviation of the carrier wave, and therefore the time delay, or code tracking error in the case of GPS. However, as long as the transform product remains at bandpass frequency, digital processing is impossible. Once the signal is converted to baseband and properly bandlimited, however, digital sampling and processing techniques can be used to

estimate the time delay. The actual hardware used to produce the signal transforms, their product, and the final correlation signal is the subject of the next two chapters.

III: The Analog Hardware for Correlation

The circuit shown in Figure 4 is the analog portion of the frequency domain correlator developed under this effort. It is a modification of the one used by Draper laboratory in the original effort (ref 4). The modifications were made as a joint effort with other Avionics Laboratory engineers, and were primarily intended to reduce the parts count and increase the gain of the output to permit sampling with the A/D converters available. In addition, the CSDL design permitted simultaneous correlation of two seperate signal channels. While the hardware needed to track two channels is only slightly more complicated than for tracking only one, the two channel circuit can not correlate over the same signal error range. Since the purpose of this effort was to demonstrate the ability to digitally correlate, it was more important to show a wide correlation range than to demonstrate multiple channel tracking. Therefore, I modified the circuit design to provide wider range, single channel tracking.

The circuit is divided into five basic blocks, as indicated by the dashed boxes. Block 1 is the basic chirp generator, used to form the pre-chirp of the Fourier transformer. Block 2 uses that chirp to generate the transform of the input signal, while block 3 forms the transform of the locally generated reference signal. Block 4

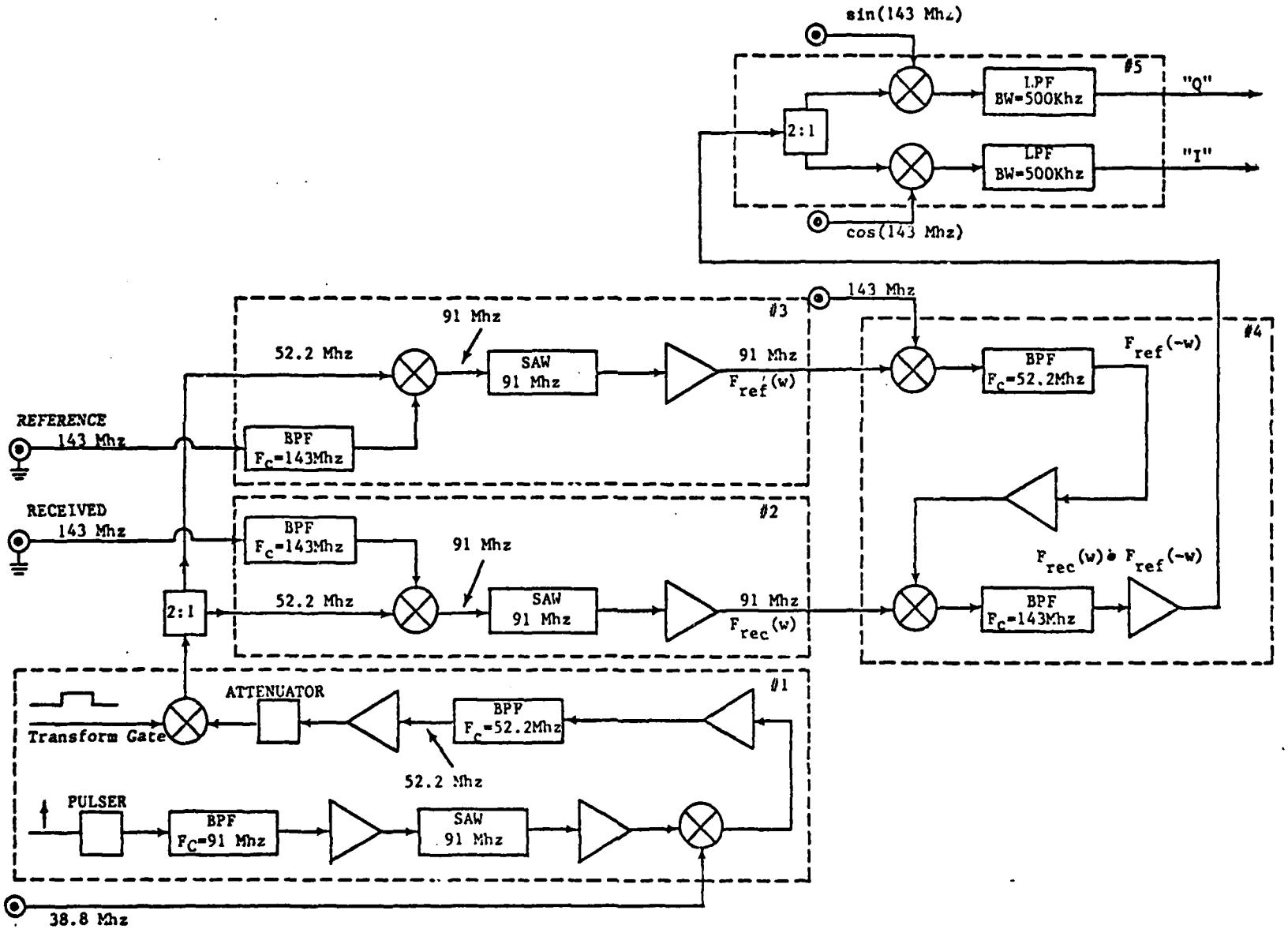


FIGURE 4 - ANALOG HARDWARE FOR FREQUENCY DOMAIN CORRELATOR

multiplies the two transforms together, and block 5 down converts the product to baseband frequency, resulting in the in-phase and quadrature components of the transform product. Those signals are then sampled, averaged, and effectively inverse transformed to find the correlation of the input and reference signals. The sampling, averaging, and digital processing circuit will be explained fully in the next chapter.

The Transform Generators and Multiplier

The transform circuit uses standard radio frequency building block circuits for ease of implementation. Each of the circuit elements shown on the diagram is, in fact, a one-to-one representation of an actual module used to build the circuit. The transform implementation repeatedly uses heterodyning to center the intermediate signals at the appropriate frequency and to insure proper spectral polarity of those signals. By properly choosing the heterodyning frequencies, a minimal component circuit can be designed. Figure 4 represents such a circuit. This circuit is a modified version of one developed by the Charles Stark Draper Laboratory. The modifications were relatively minor, primarily regrouping certain functions to minimize the required modulations, and consequently reducing the total hardware.

The pre-chirp signal is generated as the response of the SAW to an "ideal" impulse, which is provided by the pulser

circuit in block 1. The pulser is triggered by the rising edge of a square wave, and outputs a short burst of energy. This signal is then bandpass filtered and amplified, concentrating the impulse energy at a center frequency of 91 Mhz. This frequency is the center frequency of the SAW delay lines, which, when injected with an impulse, outputs a linear FM sweep, or "chirp", beginning at a frequency equal to the center frequency minus one-half the bandwidth, and increasing linearly to a final frequency equal to the center freqeuncy plus one-half the bandwidth. For the particular SAW devices used in this circuit, the SAW output is an FM waveform, sweeping in frequency from 81 to 101 Mhz over a 40 microsecond time period. After amplification, the chirp is modulated with a 38.8 Mhz carrier, amplitude limited, and bandpass filtered to provide a 52.2 Mhz "up-chirp". The up chirp refers to the positive slope of the linear frequency sweep. The signal is finally gated with a 20 microsecond rectangular time "window" to eliminate the non-linearites that occur at the upper and lower edges of the chirp. This chirp signal is then applied identically as the pre-chirp to both the input and reference transform blocks.

The input and reference signals are both present on a carrier wave, centered at 143 Mhz. This frequency is approximately 14 times the GPS code frequency of 10.23 Mhz, and was chosen to be compatible with available GPS receiver hardware and test equipment. In the input and reference transform circuits (blocks 2 and 3 in Figure 4), these

signals, band limited to 10 Mhz, are individually multiplied by the pre-chirp, and applied to another SAW device. The resulting SAW output is a 20 microsecond time signal, representing the Fourier transform, superimposed on a residual chirp waveform. The amplitude is equal to the magnitude of the input (or reference) transform. The transform phase angle is represented by the phase modulation on the SAW output.

Block 4 is the transform multiplier circuit. The reference signal transform is first modulated by a 143 Mhz carrier, and the result is filtered to provide only the lower sideband. This signal, spectrally inverted with respect to the original transform, is centered at 52.2 Mhz and contains a residual chirp with a negative slope. When multiplied by the reference signal transform, centered at 91 Mhz and containing a positive slope residual chirp, the resulting product is the power spectral density (PSD) of the input (and reference) signal, centered on a carrier whose frequency is equal to 143 Mhz plus a deviation proportional to the time delay between the input and reference signals. The residual chirps of both transforms cancel as a result of the spectral inversion. As explained in Chapter II, it is therefore unneccesary to post-multiply the individual transforms by a negative slope chirp waveform in order to remove the residual chirp.

The circuit of block 5 serves to down-convert the transform product to baseband, providing the in-phase ("I") and quadrature ("Q") components of the transform product. When sampled, averaged, and Fast Fourier Transformed, the

frequency deviation of the transform product, and therefore the time delay or tracking error in the input and reference signals, can be determined. The method of accomplishing this is the subject of the remaining chapters of this report.

The Psuedo-Random Code Generator

One additional circuit is necessary to implement the frequency domain correlator: the psuedo-random input and reference signal generators. A well known method for generating a psuedo-random binary code involves the use of digital delay elements and adders to implement an appropriate feedback polynomial. In terms of real hardware, this is easily accomplished with digital shift registers and logic gates. The circuit of Figure 5 was used to generate the psuedo-random codes. The shift registers and inverted XOR (exclusive OR) gate implement a polynomial of the form

$$1 + x^3 + x^{28}$$

The resulting code is then input to a shift register with switch selectable outputs. By properly setting the switches, an "input" signal can be formed with an adjustable delay (or advance) relative to the reference signal. These signals are then Bi-Phase Shift Key (BPSK) modulated onto a 143Mhz carrier and applied as the input and reference signals to the circuit of Figure 4.

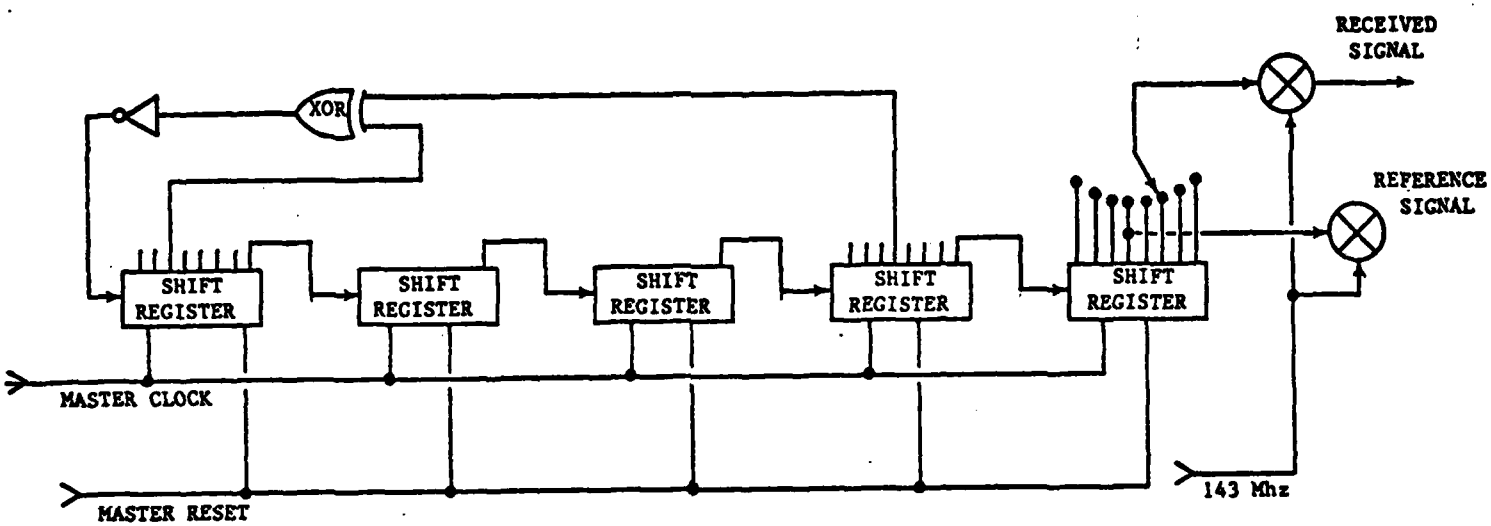


FIGURE 5 - PSUEDO-RANDOM CODE GENERATOR

Analog Results

With the aid of a spectrum analyzer, the frequency domain correlation process is easily demonstrated. The photographs of Figure 6 show the output of the transform multiplier circuit, before down-conversion to baseband. The outputs for several combinations of input/reference relative timing are shown. The signal shown is the Power Spectral Density (PSD) of the input (and, therefore, the reference) signal, with the horizontal axis representing the frequency axis. As shown previously, the frequency deviation is a measure of the input versus reference signal misalignment. In the photographs of Figure 6, taken from the spectrum analyzer, the center of the screen is the zero deviation reference. A PSD centered to the right is an "early" input signal relative to the reference, and a PSD to the left of center is a "late" input signal.

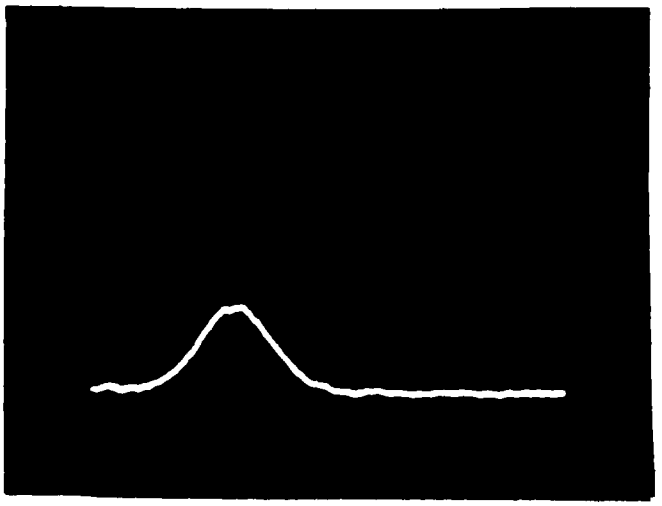
The relationship between signal misalignment and frequency deviation is a straightforward one.

$$\text{delta } f = B/T * M$$

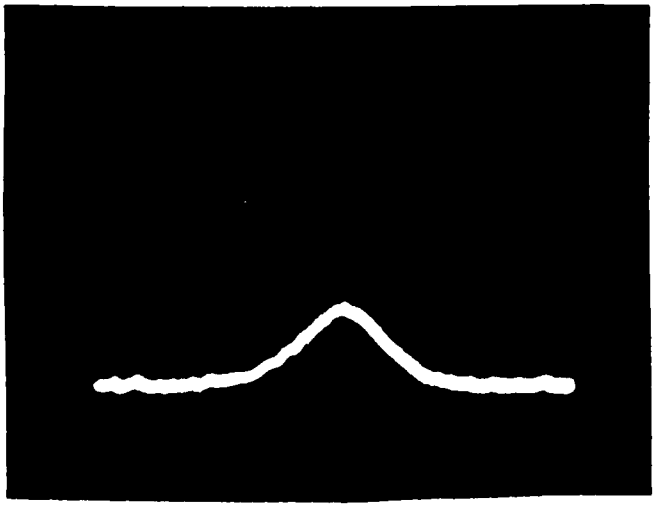
where B = analysis bandwidth of the transformer

T = the output record length of the transform multiplier

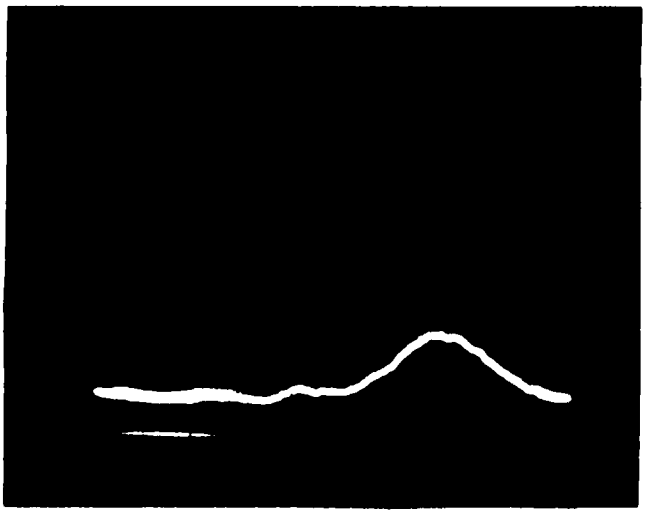
M = misalignment time of input and reference signals. Since the signals used are binary psuedo-random codes, the misalignment time is the number of code bits ("chips") times



RECEIVED CODE
2 CHIPS EARLY



RECEIVED CODE
ON TIME



RECEIVED CODE
2 CHIPS LATE

FIGURE 6 - SPECTRUM ANALYZER RESULTS

the length of a single code chip. For the devices used, the bandwidth B is 10 Mhz, the record length T is 20 microseconds, and the code chipping rate is 5 Mhz.

10Mhz * (1/5Mhz) / 20microseconds = 100Khz

Therefore, a code misalignment of 1 code chip is represented by a frequency shift of 100 Khz, and a total output bandwidth of 1 Mhz on the transform multiplier will allow measurement of +/- 5 code chips of tracking error.

An alternative to measuring the frequency shift on the spectrum analyzer is to sample the signal using an analog to digital converter and process the result by computer. This is the topic of the remaining chapters.

IV: The Digital Processing Hardware

Overview

The analog circuitry described in the previous chapter results in the multiplication of the Fourier transforms of the reference and received signals. The correlation information is represented in that product as a frequency deviation of the carrier signal from an expected frequency. This frequency deviation is directly proportional to the time delay between the received and reference signals, and is totally dependent on the physical characteristics of the transform circuitry. In order to determine that time delay, and therefore the signal tracking error, some method of measuring the frequency shift is required. As shown in the previous chapter, a spectrum analyzer correctly shows the deviation, but its use is impractical for anything but a laboratory demonstration. A digital signal processing approach is more practical, since it permits using the information for computational purposes such as navigation.

The block diagram in Figure 7 represents one possible implementation of a digital sampling and processing scheme which could be used to extract the desired frequency shift information from the transform product. Only one channel, the in-phase channel, is shown. The other channel would be identical, sharing the same master control unit and main

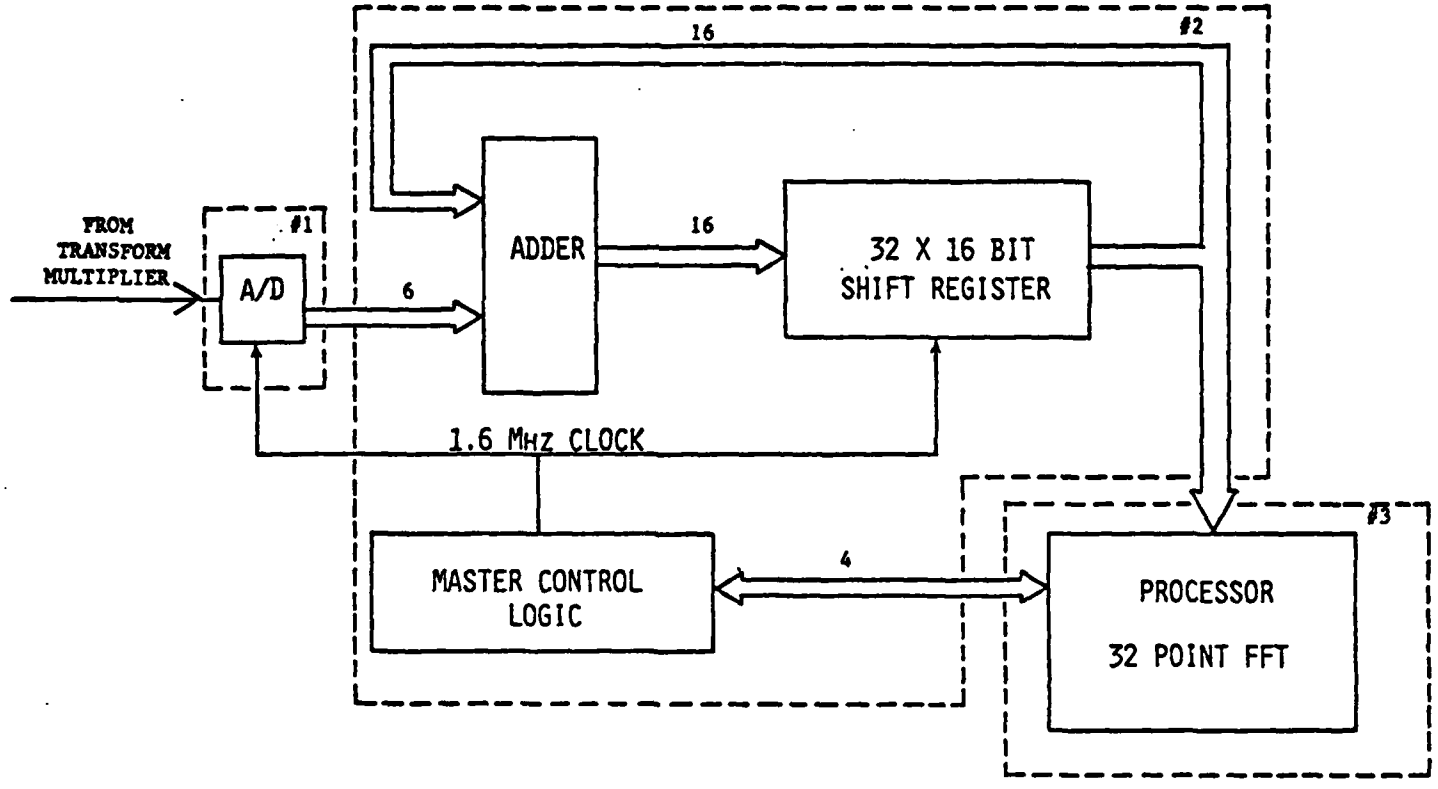


FIGURE 7 - DIGITAL PROCESSOR FOR FREQUENCY DOMAIN CORRELATOR

processor. Since the output of the transform multiplier represents only one occurrence of the autocorrelation random process, the desired signal is the expected value of that process, and hence the average of many transform product frames. If the transform product were converted to baseband (which would be required for accurate sampling), this averaged signal could then be fed directly to a processor running a Fast Fourier Transform (FFT) algorithm to determine the frequency of the baseband signal, and therefore the deviation of the carrier waveform.

The circuit outlined in Figure 7 accomplishes the sampling, averaging, and FFT processing of the in-phase channel of the baseband signal from the transform multiplier shown in Figure 4. The diagram is broken into three sections, one each for sampling, averaging, and calculating the FFT. Due to funding limitations, only one channel of the circuit could be built, so that only the in-phase component of the baseband signal is actually sampled and Fourier transformed. However, the in-phase and quadrature components of the baseband signal are nearly identical with respect to the frequency shift information. Phase tracking of the carrier waveform would require both I and Q channels, but determinaton of the frequency shift information does not. The only difficulty comes in trying to measure the frequency shift resulting from a negative time delay between reference and received input signals, i.e. a 'negative' frequency shift. With a bandpass signal of 143 Mhz, a code shift error of -1

chip would result in a new carrier frequency of 142.9 Mhz (with the 100Khz per chip error frequency shift resulting from the hardware used). Heterodyning that with a 143 Mhz local cosine oscillator and low pass filtering the result yields a signal of the form

$$A\cos(142.9 - 143) = A\cos(-.1) = A\cos(.1)$$

In other words, the frequency shift of the in-phase component is identical for positive or negative code shift errors. This problem is seemingly eliminated by using the quadrature component, since the down conversion by the sine wave local oscillator retains the positive / negative frequency relationship as a magnitude sign change, i.e.

$$A\sin(142.9-143) = A\sin(-.1) = -A\sin(.1)$$

Unfortunately, the phase of the input signals cannot be determined, and so down conversion by a sine wave cannot be distinguished from down conversion with a cosine wave plus phase shift. In order to properly determine the sign of the frequency shift it is necessary to use both the I and Q channels, detecting the sign of the frequency shift as a phase angle - i.e.

$$\text{phase} = \arctan(Q/I)$$

A phase angle of greater than 180 degrees would imply a 'negative' frequency shift, and hence a negative time delay between received and reference signals. Note that an alternative would be to down convert with a heterodyning frequency of less than 143 Mhz. For example, to detect a maximum code chip error of +/- 5 chips (at 100 Khz per chip), the down conversion to baseband would use a 142.5 Mhz local oscillator. A zero code chip error would then be detected as a 500 Khz frequency at baseband, while a -5 code chip error would show zero frequency. Unfortunately, this method requires higher bandwidth processing and faster sampling. This method was not used primarily due to the bandwidth limitations of the hardware available. As a result, the FFT algorithm is unable to detect the sign of the code chip error, only its magnitude.

The A/D Converter

The analog/digital (A/D) converter used is a commercial device, available as off the shelf hardware. It includes the buffer circuitry for setting offsets, gain, and input impedance. The unit was configured to operate over a .5 volt input range and provide a 6 bit digital word as output. The 6 bit A/D was selected as adequate since dynamic range of the circuit was not required for breadboard demonstration. The sampling clock is provided by the master control logic, built as part of the second block along with the averaging circuit.

The Averaging Circuit

The second block is the major component of the digital correlation hardware. It receives the samples from the A/D converter, averages those samples, and provides the result to the FFT processor for calculation of the frequency deviation. Block two is itself divided into two components - the averaging circuit and the master timing circuit. The detailed schematic diagrams of both are shown in Appendix A. The averaging circuit is configured as a recirculating shift register 32 words long, each word consisting of 16 bits. In operation, 32 samples of the transform product (at baseband) are sampled at a 1.6 Mhz clock rate. Each sample is 6 bits wide. After a complete transform frame has been sampled, 32 words, 6 bits each, reside in the shift register, representing one complete frame sample from the transform product. Upon command from the master timing circuit, the transform circuit forms another transform frame, and the A/D begins sampling another 32 words. This time, each word is added to the previous sampled word in that stage of the shift register, and then re-stored in the shift register. In other words, the first sample from frame two is added to the first sample from frame one, and the result is stored in position one of the shift register bank. This process continues for all 32 samples of each transform frame. As each new frame is added sample by sample to the previous frame, the bit length of the sum increases. For 256 additions of 6 bits (maximum) each, at least 14 bits of storage are needed in the shift register.

The averaging function can be obtained by selecting only the eight most significant bits for transfer to the FFT processor. Alternately, the processor can use full precision from the averaging hardware and divide as necessary in software. Both approaches were tried for this effort, with no noticeable difference in performance. In terms of implementation speed, it would obviously be better to perform the divide by 256 function by transfering only the 8 most significant bits.

The Master Control Unit

The master control logic not only controls the sampling of the transform product, but the generation of the chirps and timing gates needed to form the signal transforms and their product. The master control logic insures that an average of 256 independent transform product frames is calculated, then signals the processor that the data is ready for FFT processing.

The master timing in the control logic is such that 32 samples adequately sample the transform product, guarding against aliasing and insuring that the sampling takes place during valid output frames from the transform product generator. Since the transform product is low pass filtered to 500Khz, a sampling rate of 1.6 Mhz eliminates aliasing problems. At the same time, the 32 samples take place within 20 microseconds (32 * 1/1.6Mhz = 20 microseconds) insuring that the transform frame, which is less than 20 microseconds long, is completely sampled. The choice of 256 samples of the

transform frame was largely based on ease of implementation, since 8 bit counters digital are readily available and the divide by 256 function is so trivial. Additionally, an average of 256 samples is generally enough to insure a value close to the probabilistic expected value. Finally, 256 frames at 20 microseconds per frame yields 5.16 milliseconds, roughly one quarter of the data bit duration in the 50 Hz GPS data signal. If desired, the integration time could easily be increased so as to provide maximum processing gain (which is essential to GPS operation) but this was unnecessary for breadboard demonstration.

One additional constraint on the master control logic is the requirement to provide chirp impulses and timing gates to the transform generating circuit. Since the transform product is not valid until some time after the impulse to the chirp generator, sampling, and even transform multiplication, cannot begin until after some delay. Theoretically, this delay can be calculated and is roughly equivalent to the dispersion time of the SAW delay lines. In practice, however, it was easier to include a variable time delay for the transform gate and use a trial and error method to optimize the transform product. This variable delay circuit is included in the master control logic.

The FFT Processor

The final section of the digital correlation circuit is the processor itself. The processor has as its input the 32

samples from the shift register averaging circuit, representing the expected value of the transform product. As shown in Chapter I, this product represents the Fourier transform of the correlation between two signals which are identical except for a time delay. This transform product, if inverse Fourier transformed, yields the desired autocorrelation of those two signals. The transform product can also be viewed as a time serial version of the Power Spectral Density of the input signal (the psuedo random code in this case) amplitude modulating a carrier wave whose frequency represents the time delay between the received and reference codes. If the averaged samples of the transform product are Fourier transformed, the resulting signal will be the autocorrelation function centered about a frequency which can be directly related to the code error between the two input signals. The principle of duality between Fourier transformation and inverse Fourier transformation insures that both of the above interpretations are correct. Since the transform product is output as a time serial frequency domain signal, either interpretation can be readily applied. However, since the desired result of the processing is the frequency deviation representing the code racking error, the second interpretation was implemented.

The processor implements a Fast Fourier Transform (FFT) algorithm, a highly efficient digital calculation of the discrete Fourier transform. Since the FFT is essentially a baseband algorithm (i.e. the output always starts at zero

frequency), it would be highly inefficient to perform the FFT at bandpass. Therefore, as shown in Figure 4 (Chapter III), the transform product is converted to baseband prior to sampling and averaging. The processor used in the breadboard demonstration was an APPLE II+ computer with a peripheral interface to the sampling and averaging circuitry. The peripheral interface was a general purpose parallel interface plug-in card for the APPLE II. The choice of this personal computer was based on the ease of interfacing, programming, and availability of software support tools. The FFT was programmed in the BASIC language, and based upon a decimation - in - time algorithm originally written in the FORTRAN language (ref 7). (For a complete discussion of the Fast Fourier Transform algorithm, the reader is referred to references 1, 2, 3, 7, and 11.) The alogorithm presented in ref 7 contained several errors that had to be corrected prior to implementation on the processor, including an erroneous control loop and modification to run in a language without complex arithmetic operations. Once implemented, the algorithm was tested with several known inputs, such as a sum of sine waves. In addition to the FFT algorithm, the program in the processor also includes the necessary code for communication to and from the interface (and hence the averaging circuitry) as well as to and from the user. This interactive capability includes plotting the input (sampled) data if desired, as well as plotting the results of the FFT calculation. During development, the processor ran the

program as an interpreter (BASIC is an interpretive computer language). In final form, however, the program was compiled to increase program speed. The interpreter version (and therefore the compiler source code) is listed in Appendix B, along with a simplified software flow chart.

V: Results

The results of this effort are divided into two areas: analog and digital. Prior to even sampling the transform product, it was necessary to demonstrate that the analog portion of the hardware worked properly. Since several modifications were made to the CSDL design, it was imperative to prove that the modifications worked as predicted. In addition, the design was based on theoretical or ideal specifications for the hardware. Before any digital processing was attempted, therefore, those ideal parameters had to be verified. The result was the series of photographs shown in Figure 6, Chapter III. As explained previously, those photographs show that the code timing error between the received and reference signals does, in fact, translate to a shift in frequency detectable on the spectrum analyzer.

Testing and debugging the digital hardware was considerably more difficult, since there was no previous work attempted in this area with which to compare results. The single factor which contributed most to the debugging effort was the processor used in the project. The ease of programming and the interactive nature of the APPLE II computer permitted very rapid analysis and correction of both hardware and software errors. By proceeding through increasingly more sophisticated simulations of the intended

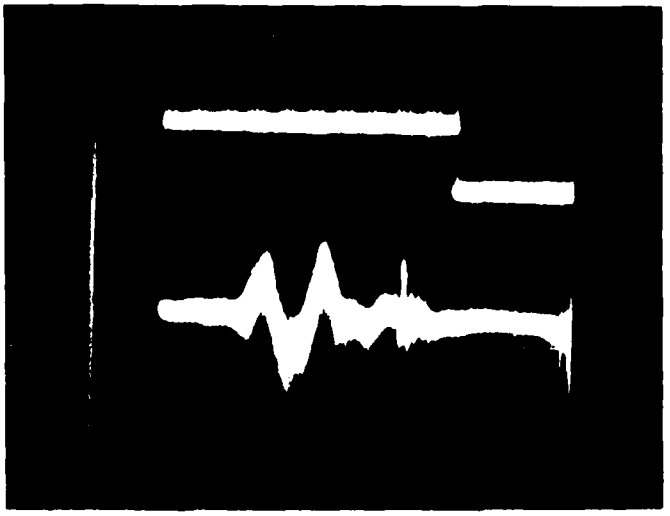
input signal, the complete debugging was accomplished in a surprisingly short time.

One particular problem in debugging the hardware is worthy of special note. Once the sampling and averaging circuit was operational, and working properly with the FFT algorithm on simulated input signals, the circuit was connected to the output of the transform multiplier. The multiplier output had been converted to baseband and amplified sufficiently to permit sampling with the A/D converter, but the lowpass filter had not yet been received and was consequently not in the circuit as originally planned. Since the conversion to baseband through heterodyning will produce both a baseband signal and a signal at twice the input carrier frequency, the lowpass filter is required to filter out the high frequency components of the signal to be sampled. This prevents aliasing in the sampling process. A discrete component lowpass filter was built to replace the modualr filter originally intended for the circuit. The output was displayed on the oscilloscope, where the expected shift in frequency was apparent, though not as clearly defined as expected. The processor, however, could not obtain good data from the sampling and averaging hardware, and the results from the FFT were inconsistent. Several days were spent trying to locate the problem, since reason indicated that if the frequency shift was apparent on the oscilloscope, then it could be detected by the processor. Fortunately, the lowpass filters arrived at that point, and provided the necessary

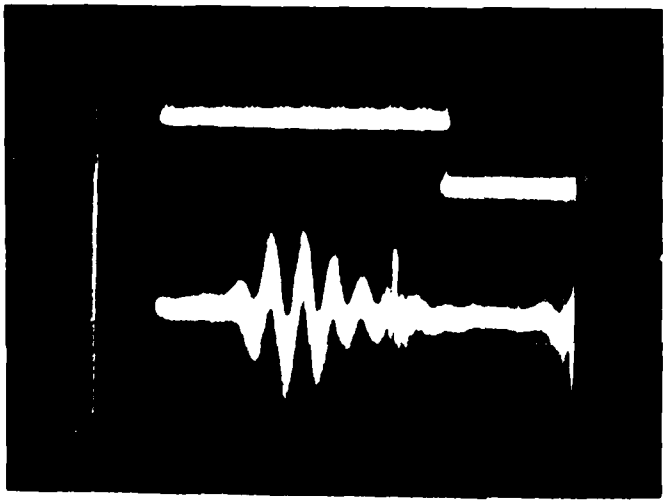
filtering to the signals to provide clean sampled data to the processors. The FFT processor worked immediately, providing exactly the results expected. Apparently, some high frequency signals were passing through the home-built filter, causing aliasing of the low frequency data and corrupting the input to the FFT processor. The oscilloscope display was less corrupted, however, since its bandwidth limitation formed an effective lowpass filter. The resulting display was deceptively 'cleaner' than the actual data going into the sampling and averaging circuit.

Figure 8 shows oscilloscope photographs of the baseband signal for different values of code tracking error. The expected sinusoidal frequency is clearly visible in all three photographs. As predicted, the frequency increases for increasing code tracking errors. Also note that it is impossible to determine if the received code is early or late (a negative or positive frequency shift) since only the in-phase baseband signal is displayed.

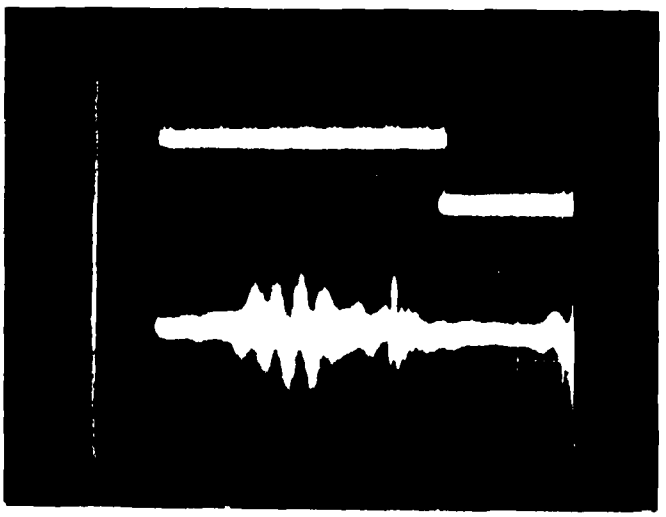
The final output of the processor was intended only to show the code tracking error between the received and reference codes. The program displays the result of the FFT calculation, as a magnitude plot, and picks the frequency value of the signal peak as the carrier frequency deviation. Dividing this deviation by 100Khz per code chip error yields the code tracking error between the two input signals. The output from several runs of the program is shown in Figures 9, 10, and 11. The figures represent the output from the



RECEIVED CODE
1 CHIP LATE



RECEIVED CODE
2 CHIPS EARLY



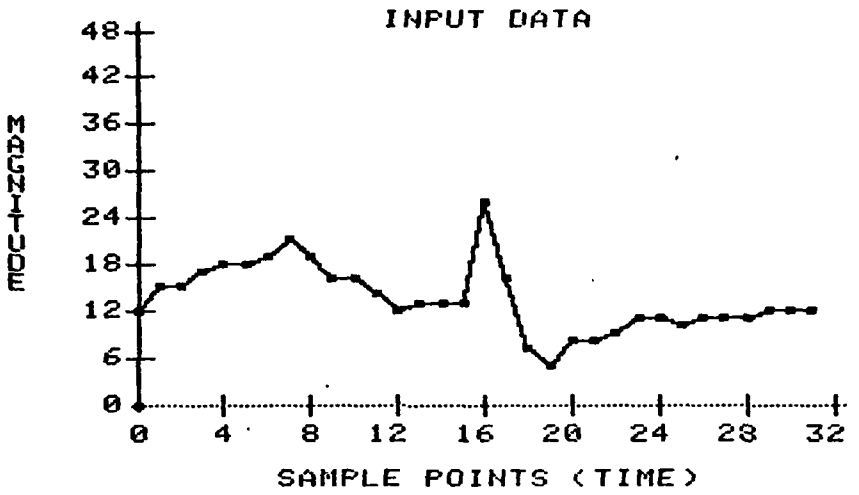
RECEIVED CODE
4 CHIPS LATE

FIGURE 8 - OSCILLOSCOPE RESULTS AT BASEBAND

computer for input code tracking errors of 0, 4, and -2 chips, respectively. Note the presence of a strong direct current (DC) component in the transform plot. That component is, in fact, present in the sampled data, since the A/D converter was configured with a DC offset built into the input signal. This permitted the FFT processor to work with positive sample values only, resulting in faster execution of the program, and eliminating the need to calculate and subtract out the DC component for each of the 32 sample points.

In operation, the physical characteristics of the transform device caused some deviation from the expected output. During the design of the hardware, a constant dispersion time of 20 microseconds for the SAW delay lines was assumed. The proportionality constant between the code error and resulting frequency shift was based on that value. In actuality, the dispersion time is not exactly 20 microseconds, nor is it precisely stable. The primary result was that the 100 Khz per chip ratio was wrong, with a value of 85 Khz per chip closer to correct, and still not exactly constant. A minor change was made to the software calculations to correct for the constant error, but the stability error would require some sort of calibration scheme to periodically test the frequency shift by injecting a known code error, and including the result in future error calculations.

The calibration factor mentioned above was also factored into the FFT output displays of Figures 9, 10, and 11. Theoretically, with a 20 microsecond transform window, the



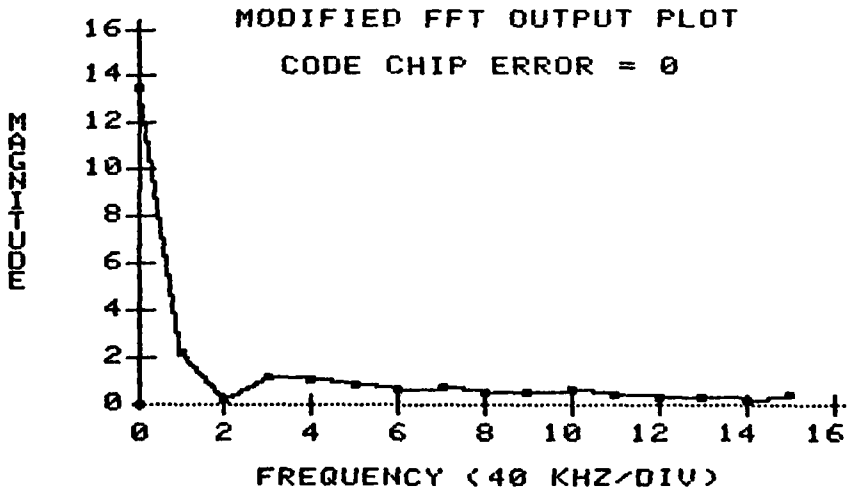
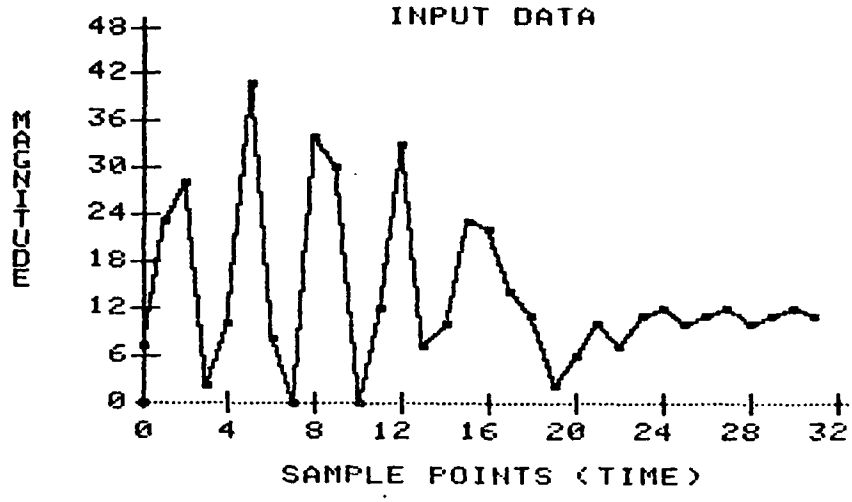


FIGURE 9: DIGITAL PROCESSOR RESULTS (0 CHIP ERROR)



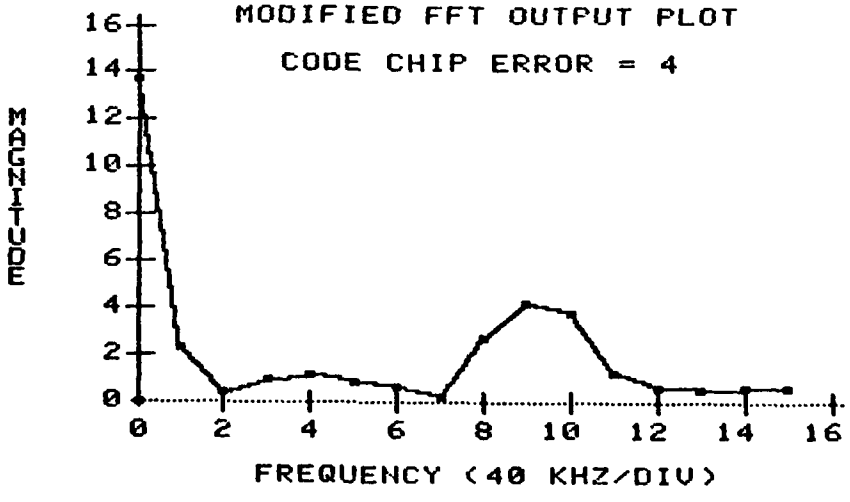
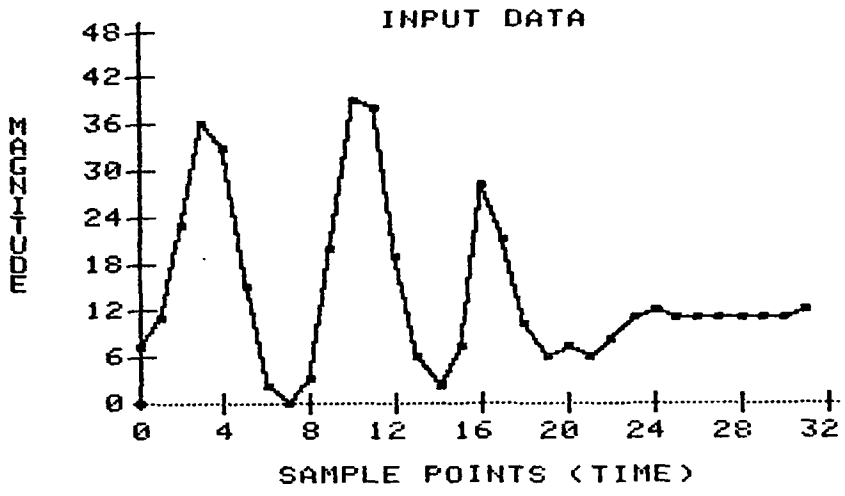


FIGURE 10: DIGITAL PROCESSOR RESULTS (4 CHIP ERROR)



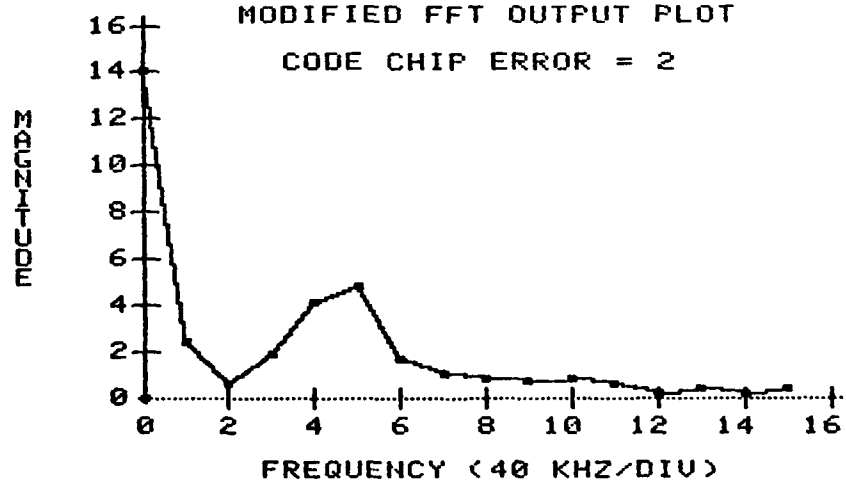


FIGURE 11: DIGITAL PROCESSOR RESULTS (-2 CHIP ERROR)

frequency resolution of the FFT should be 50 Khz per division, instead of the 40 Khz per division shown in the figures. However, for the sake of consistency, the frequency scale was compressed to maintain the 85 Khz per chip frequency deviation relationship that was based upon the actual SAW dispersion time. This in no way changes the accuracy of the results, but simply makes it easier to relate frequncy deviation to code tracking errors.

It should be pointed out that the discrepancy in the frequency deviation proportionality constant was first detected on the spectrum analyzer, and originally attributed to an uncalibrated analyzer display. Only after some analysis and research into the actual characteristics of the delay lines was the true cause determined.

Refering specifically to Figure 11, notice that the FFT algorithm incorrectly estimates the sign of the code error for one input. This is due to the problem discussed previously, where the FFT cannot distinguish between a positive and 'negative' frequency without both the I and Q channels of the baseband signal.

In this effort, no attempt has been made to close the tracking loop automatically. The process would be relatively straightforward, requiring only a digital / analog (D/A) converter connected to a variable frequency oscillator (VFO). The computer would drive the D/A converter based upon the code tracking error, and the VFO would drive the pseudo-random code generator. Funding and time limitations prohibited

implementing this function, which would also require that both the I and Q channels be processed to obtain the sign of the frequency shift.

VI: Summary and Recommendations

The purpose of this effort was to build and demonstrate a digital processing scheme for correlating on psuedo-random spread spectrum signals via frequency domain processing techniques. The effort was divided into two distinct areas: 1) modifying and demonstrating the analog portions of the frequency domain hardware and; 2) designing, building, and programming the digital hardware for processing the signals from the analog circuit to form the desired correlation information. The majority of effort was concentrated in the second area.

The analog hardware was modified from a design developed under Air Force contract. The circuit demonstrated that spread spectrum synchronization information can be represented by the frequency deviation of a carrier signal in the frequency domain. As shown in the photographs of Figure 5, the frequency shift is directly proportional to the time delay between the received and reference signals in a spread spectrum system.

The digital processing scheme developed under this effort was intended to translate the correlation information from the analog hardware into a digital format, i.e. a digital number representing the time delay between received and reference signals. That was accomplished by sampling the

analog correlation signal, averaging the samples, and performing a Fast Fourier Transform calculation to determine the frequency shift information representing the time delay of interest. Figures 9, 10, and 11 in Chapter V shows the computer output for several different time delays between received and reference signals.

Several compromises were made during the design and construction of the hardware, generally simplifying its implementation. Although none of those compromises invalidate the overall results, they do suggest several areas where further research might be warranted. The quadrature baseband signal component should be included in the FFT calculation. This would require building a second averaging circuit and A/D converter. Availability of the Q channel information would provide the FFT with the capability for calculating the sign of the desired time delay. This could not be done under this effort, because the hardware for the Q channel was not available. In addition, the processor could perform carrier phase tracking as well as code tracking. This would certainly be required for high performance GPS type navigation receivers.

Another logical step would be to use the code (and carrier phase) tracking information to close the loops in a tracking system. As explained previously, this would be relatively straightforward, but it too is an essential part of a real GPS receiver.

Further investigation into the area of processing loss

due to the digital processing scheme is needed. For example, is the ultimate tracking ability of the system degraded because of the frequency domain approach, and how might this be corrected? Specific research areas might involve the sampling rate of the A/D converters, the number of bits in those converters, the use of other than rectangular windows in the transform gate, and the use of different length SAW devices in forming the Fourier transforms.

The above suggestions represent only a small fraction of potential areas for further research into frequency domain processing (FDP) techniques. The theory describing various aspects of FDP is fairly well developed, but the practical aspects have not been investigated due to limitations in available hardware for implementation. Only recently has the required hardware become available. Current efforts at the Avionics Laboratory indicate that frequency domain processing techniques are being applied to a continually expanding spectrum of military data transmission hardware, including navigation, communication, and radar systems. Any research in the area will advance the technology and serve to broaden its applicability.

BIBLIOGRAPHY

1. Carretto, J., "An Implementation of the Fast Fourier Transform", unpublished class presentation for EE 6.64, Digital Control Systems, School of Engineering, Air Force Institute of Technology, Wright Patterson AFB, Ohio, Spring 1981

2. Cooley, J., J. Tukey, "An Alogorithm for the Machine Calculation of Complex Fourier Series", Mathematics of Computation, Vol 19, No. 90, PP 297-301, 1965

3. Gentleman, W., G. Sande, "Fast Fourier Transforms - For Fun and Profit", AFIPS Proceedings 1966 Fall Joint Computer Conference,

4. Guinon, W., et al., Reference System Technology Development, AFAL-TR-79-1189, Vol 2, pp 9-34, November 1980

5. Houpis, C., G. Lamont, class notes from EE 6.64, Digital Control Systems, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio , Spring 1981

6. Liu, D., J. Liu, Linear Systems Analysis, New York, McGraw-Hill, Inc., 1975

7. Oppenhiem, A., R. Schafer, Digital Signal Processing, Englewood Cliffs, New Jersey, Prentiss Hall, Inc., 1975.

8. Papoulis, A., Probability, Random Variables, and Stochastic Processes, New York, McGraw-Hill, Inc., 1965

9. Skolnik, M., Introduction to Radar Systems, Chapter 11, McGraw- Hill, Inc., New York, 1980

10. Slobodnik, A., "Surface Acoustic Waves and SAW Materials", Proceedings of the IEEE, Vol 64, No. 5, May 1976

11. Stanley, W., S. Peterson, "Fast Fourier Transforms on Your Home Computer", BYTE Magazine, Vol 3, No. 12, pp 14-25, Dec 1978

12. Van Broekhoven, P, miscellaneous notes and briefing material from Air Force contract F33615-78-C-1563, September 1980-November 1981

13. Williamson, J., "Reflective-Array Devices", Proceedings of the IEEE, Vol 64, No. 5, May 1976

Appendix A: Hardware Schematic Diagrams

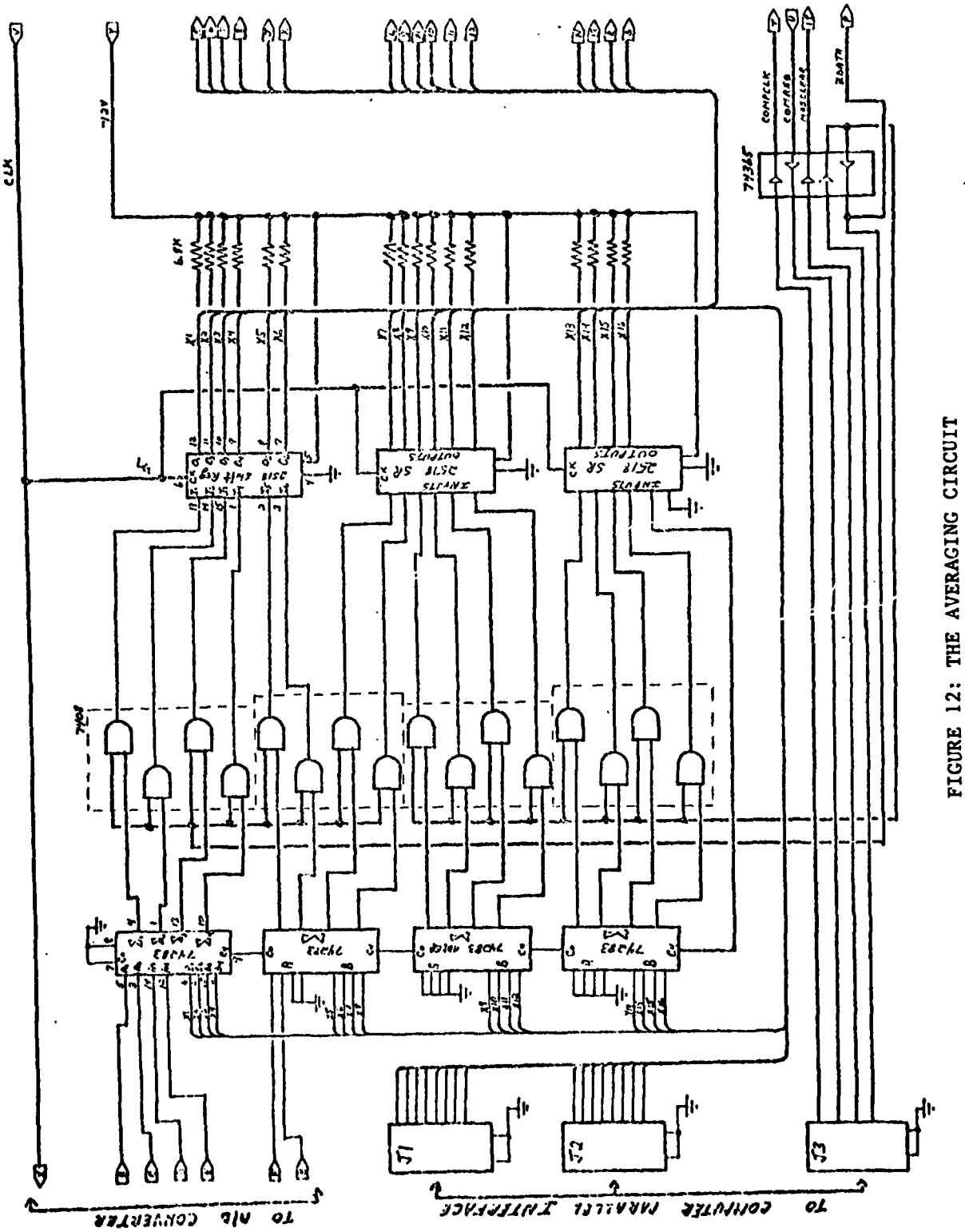


FIGURE 12: THE AVERAGING CIRCUIT

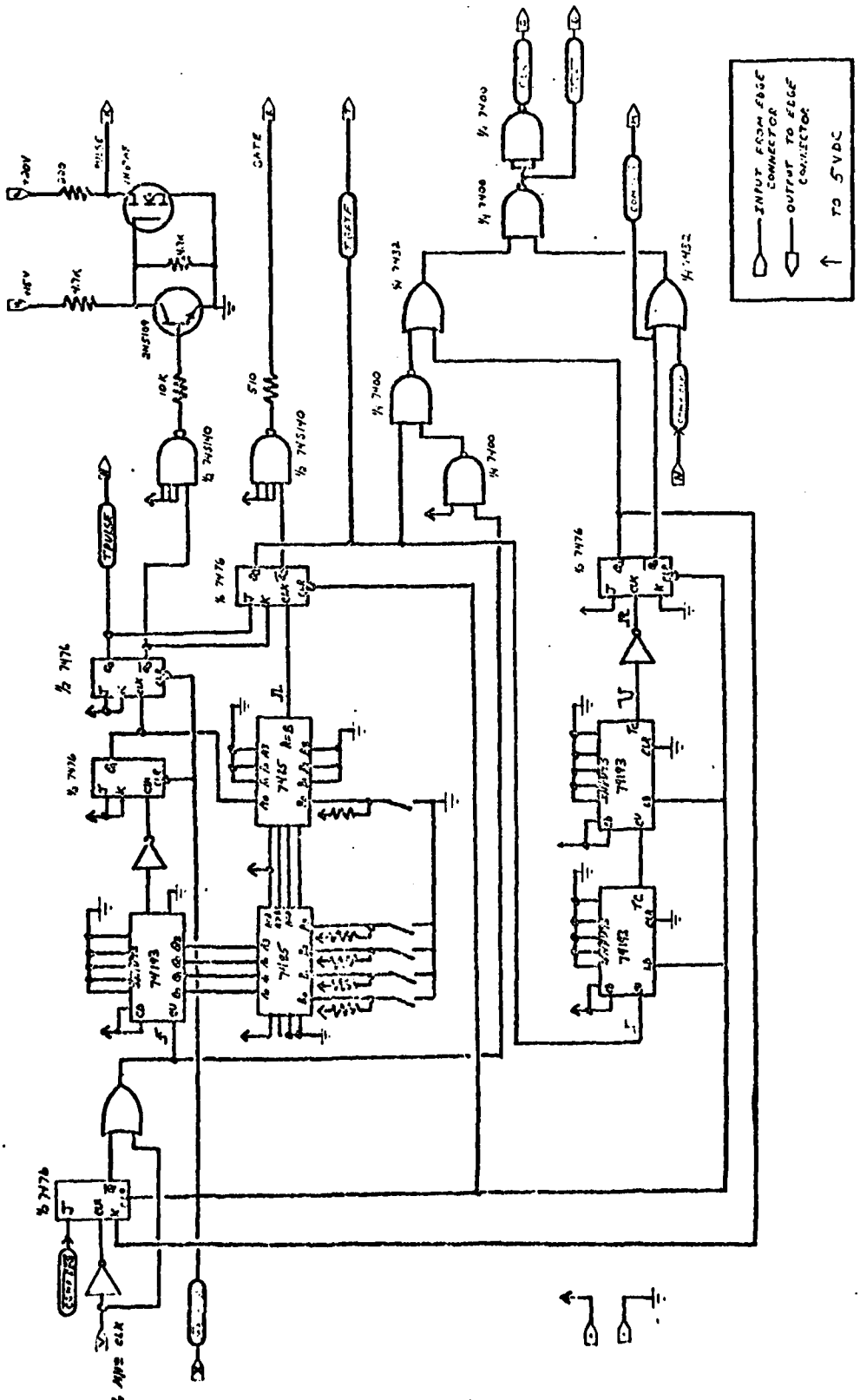
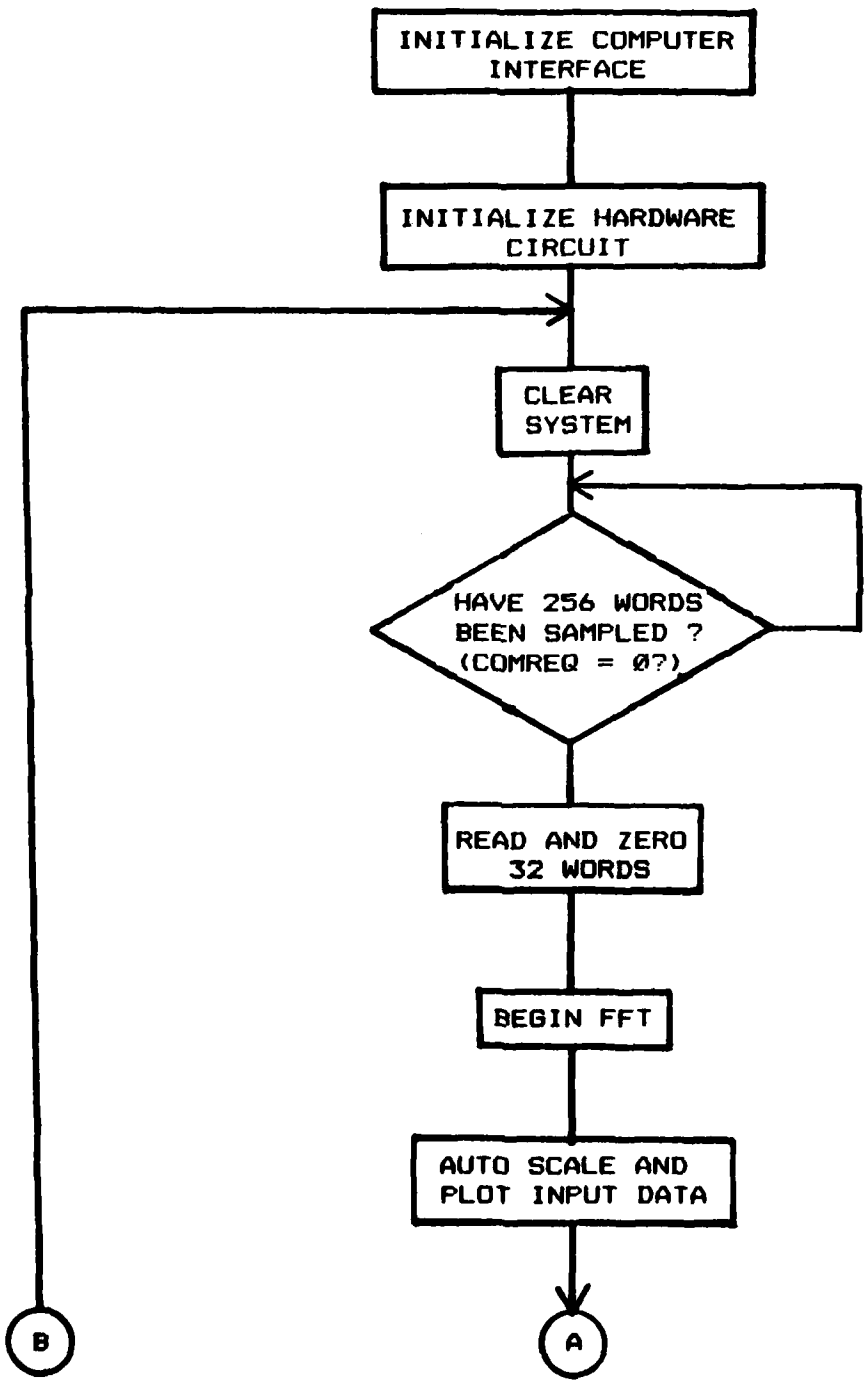
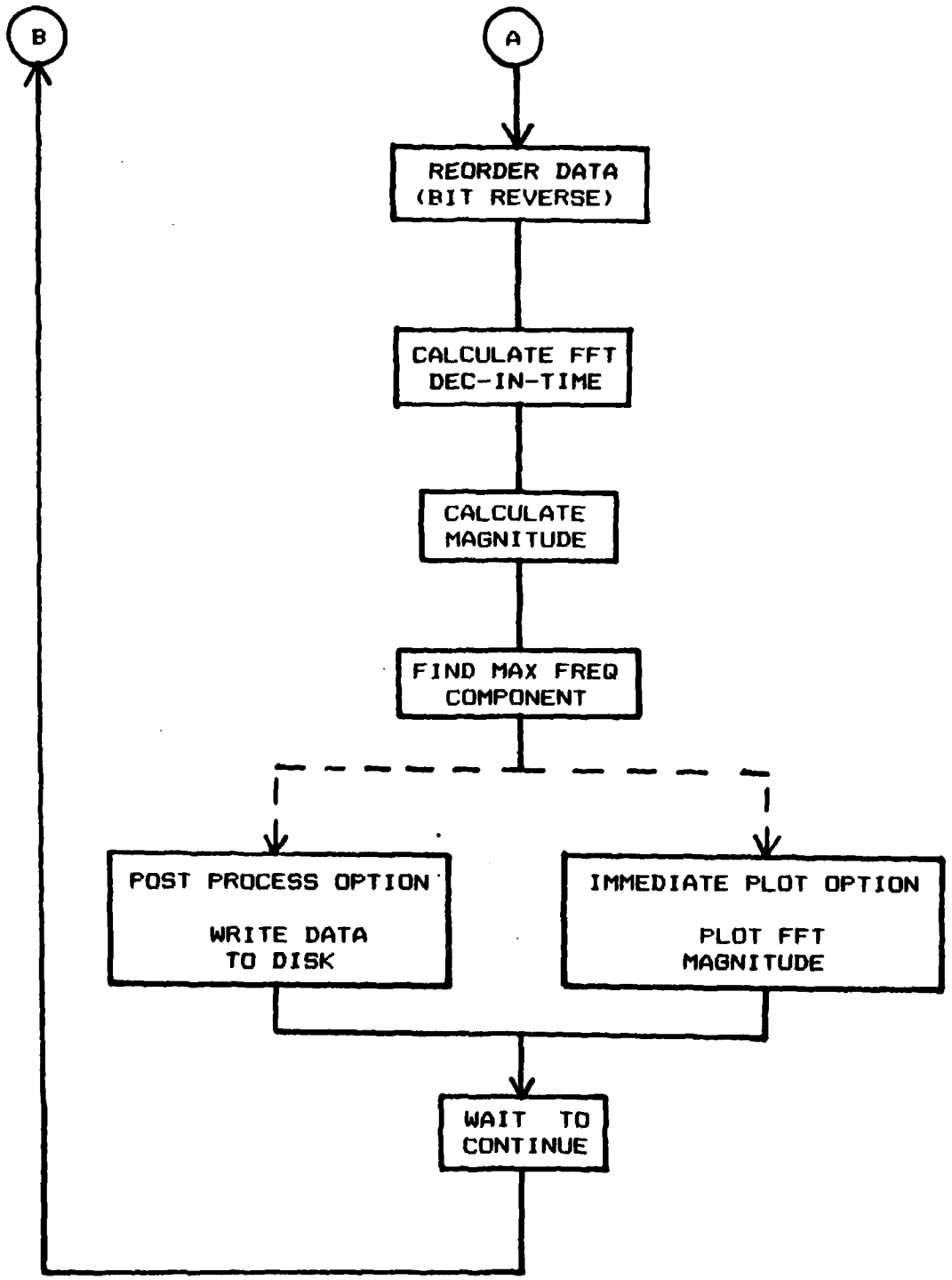


FIGURE 13: THE MASTER CONTROL UNIT

Appendix B: *The Software for the Processor*

SOFTWARE FLOW CHART




B
A
REORDER DATA
(BIT REVERSE)
CALCULATE FFT
DEC-IN-TIME
CALCULATE
MAGNITUDE
FIND MAX FREQ
COMPONENT
POST PROCESS OPTION
WRITE DATA
TO DISK
IMMEDIATE PLOT OPTION
PLOT FFT
MAGNITUDE
WAIT TO
CONTINUE

```
30  REM  ************************************************
40  REM  **       TITLE:  THESIS SOFTWARE                **
50  REM  **     LT JOSEPH A. CARRETTO, JR                **
58  REM  ************************************************
60  REM  **      THESIS SOFTWARE - 19 JUNE 1982          **
70  REM  ** PROGRAM INITIALIZES PARALLEL INTERFACE       **
75  REM  ** AND DIGITAL HARDWARE, THEN READS 32 WORDS    **
80  REM  ** REPRESENTING THE AVERAGE OF 256 TRANSFORM    **
90  REM  ** FRAMES.  INPUT DATA CAN BE PLOTTED IF        **
95  REM  ** DESIRED.  FFT IS CALCULATED IN PLACE VIA     **
100  REM  * DECIMATION-IN-TIME ALGORITHM.  OUTPUT        **
110  REM  * MAGNITUDE IS EITHER PLOTTED IMMEDIATELY,     **
115  REM  * OR STORED ON DISK FOR LATER POST PLOTTING    **
120  REM  * VIA APPLE PLOT PROGRAM.                      **
130  REM  * CODE CHIP TRACKING ERROR IS CALCULATED       **
140  REM  * AND DISPLAYED                               **
145  REM  ************************************************
150  REM  ** PROGRAM WRITTEN FOR THE APPLE II+ WITH      **
160  REM  ** PARALLEL INTERFACE IN SLOT #2.  SPECIFIC    **
165  REM  ** INTERFACE USED WAS MANUFACTERED BY JOHN     **
170  REM  ** BELL ENGINEERING, INC.                      **
175  REM  ************************************************
180  DIM DA(32): DIM XP(32): DIM XM(32)
190  DIM XR(32): DIM XI(32)
195  REM     **       SET INTERFACE ADDRESSES           **
200 J1 = 49665
220 J2 = 49664
240 J3 = 49793
260 D1 = 49667
280 D2 = 49666
300 D3 = 49795
310  REM    **     INITIALIZE PARALLEL INTERFACE        **
320  POKE D1,00
340  POKE D2,00
360  POKE D3,253: REM  11111101
380  REM   ** SET J3 TO OUTPUT, EXCEPT BIT 2 (COMREQ)   **
400  POKE J3,133
410  REM   **  SET COMPCLK, ZDATA, AND MASCLEAR HIGH     **
440  POKE J3,5: REM   ** RESET MASCLEAR                 **
460  POKE J3,133: REM    ** MASCLEAR HIGH               **
540  REM  ** RESET ZDATA AND CLOCK 32 PULSES TO ZERO     **
560  INVERSE : PRINT "ZEROING": NORMAL
580  POKE J3,129
600  GOSUB 1400
620  POKE J3,133
630  REM   ************************************************
640  REM   **BEGIN SAMPLING CLOCK - WAIT UNTIL COMREQ    **
660  REM  ** TOGGLE MASCLEAR  **
670  PRINT "CLEARING SYSTEM": PRINT
```

```
680  POKE J3,5
700  POKE J3,133
740 PT = 135
760 S =  PEEK (J3)
780  IF S = PT GOTO 760
800  PRINT "COMREQ LOW - ";S: PRINT
810  REM   ************************************************
820  REM   ** READ DATA AND AUTO ZERO SHIFT REGISTERS   **
840  PRINT "READING AND ZEROING 32 WORDS"
860  PRINT
880  POKE J3,S - 1
900  POKE J3,S
920  REM   **  CLOCK TO SYNC SHIFT REGISTERS             **
940  POKE J3,129
960  FOR E = 1 TO 32
980 T1 =  PEEK (J1)
1000 T2 =  PEEK (J2)
1020 DA(E) = T2 * 256 + T1
1040 DA(E) =  INT (DA(E) / 256)
1060 S =  PEEK (J3)
1080  POKE J3,S - 1
1100  POKE J3,S
1120  NEXT E
1140  POKE J3,133
1400  REM  ********************************************************
1410  REM  ** CLOCK ROUTINE - SENDS 32 PULSES TO HARDWARE **
1415  REM  ********************************************************
1420 S =  PEEK (J3)
1440  FOR CK = 1 TO 32
1460  POKE J3,S - 1
1480  POKE J3,S
1500  NEXT CK
1520  RETURN
2000  REM    ************************************************
2020  REM    ** PLOT ROUTINE - USED FOR INPUT AND OUTPUT **
2030  REM    ************************************************
2520 EMAX = 1:MN = 0
2530  REM  *** FIND MIN AND MAX FOR AUTO SCALE ***
2535  FOR Q = 1 TO 32 / O
2542  IF XP(Q) > EMAX THEN EMAX = XP(Q)
2545  IF XP(Q) < MN THEN MN = XP(Q)
2550  NEXT Q
2560  HGR
2580 KNST = 159 / (EMAX - MN)
2600  HCOLOR= 3
2620 XP = 0:YP = 159
2640  HPLOT XP,YP TO XP,YP
2660 IN = (270 / 32) * O
2680  FOR Q = 1 TO (32 / O)
2700 XP = Q * IN
```

```
2720 YP = 159 - (XP(Q) * KNST)
2740  HPLOT  TO XP,YP
2760  NEXT Q
2820  RETURN
3000  REM  ************************************************
3010  REM  **  FFT CALCULATION - PERFORMS IN-PLACE      **
3020  REM  ** DECIMATION-IN-TIME ALGORITHM              **
3040  REM  ************************************************
3080  TEXT
3100 E = 0
3120 M = 5
3140 N = 32:NV = 16:NM = 31
3200  FOR Q = 1 TO 32:XR(Q) = DA(Q):XI(Q) = 0
3210  NEXT Q
3220 DT = 1 / 1.6E6
3255 O = 1
3300  FOR Q = 1 TO N:XP(Q) = XR(Q): NEXT Q
3320 V = DT
3340  REM  ** PLOT INPUT DATA   **
3360  PRINT "INPUT MAGNITUDE PLOT"
3370  GOSUB 2000
3375  PRINT "CALCULATING FFT"
3380  FOR Q = 1 TO N:XR(Q) = XR(Q) / N: NEXT Q
3400  REM  ****** BIT REVERSING ROUTINE       ****
3410  REM  ****** USED TO RE ORDER DATA FOR   ****
3415  REM  ****** IN-PLACE FFT CALCULATION    ****
3420 J = 1
3440  FOR I = 1 TO NM
3460  IF I >  = J GOTO 3540
3480 TR = XR(J)
3500 XR(J) = XR(I)
3520 XR(I) = TR
3540 K = NV
3560  IF K >  = J GOTO 3640
3580 J = J - K
3600 K = K / 2
3620  GOTO 3560
3640 J = J + K
3660  NEXT I
3680  REM  ** PERFORM ACTUAL FFT CALCULATION   **
3700 PI = 3.1415927
3720  FOR L = 1 TO M
3740 LE = 2 ^ L
3760 L1 = LE / 2
3780 UR = 1:UI = 0
3800 WR =  COS (PI / L1):WI =  - 1 *  SIN (PI / L1)
3820  FOR J = 1 TO L1
3840  FOR I = J TO N STEP LE
3860 IP = I + L1
3880 TR = XR(IP) * UR +  - 1 * (XI(IP) * UI)
```

```
3900 TI = (XR(IP) * UI) + (XI(IP) * UR)
3920 XR(IP) = XR(I) - TR:XI(IP) = XI(I) - TI
3940 XR(I) = XR(I) + TR:XI(I) = XI(I) + TI
3960  NEXT I
3980 TR = (UR * WR) - (UI * WI)
4000 TI = (UR * WI) + (UI * WR)
4020 UR = TR:UI = TI
4040  NEXT J
4060  NEXT L
4070  REM  **** CALCULATE MAGNITUDE   **
4080  FOR Q = 1 TO N
4100 XM(Q) =  SQR (XR(Q) ^ 2 + XI(Q) ^ 2)
4120 XP(Q) = XM(Q)
4140  NEXT Q
4160 O = 2
4200  PRINT "OUTPUT MAGNITUDE PLOT"
4205 Z = 2
4208  REM  ** FIND FREQUENCY OF MAXIMUM MAGNITUDE  **
4210  FOR Q = 2 TO 16: IF XP(Q) > XP(Z) THEN Z = Q
4215  NEXT Q
4217  REM  *** REAL TIME PLOT OPTION - GOSUB 2000 **
4218  REM  ** POST PROCESS PLOT OPTION - GOSUB 5000 **
4220  GOSUB 2000
4230  REM  ***** CALCULATE CODE TRACKING ERROR  **
4240  INVERSE : PRINT
4243 K = 2.1: REM  ** DISPERSION TIME SCALE FACTOR **
4245 A =  INT (.5 + (Z - 1) / K)
4250  PRINT "CODE CHIP ERROR OF "A
4260  NORMAL
4270  PRINT : PRINT "TYPE ANY CHARACTER TO CONTINUE"
4280  GET W$
4300  TEXT : GOTO 660
4420  REM   ********************************************
4430  REM   **                    END                  **
4440  REM   ********************************************
4500  END
5000  REM  *** SUBROUTINE TO STORE PLOT DATA ON DISK  **
5020  REM  *** FOR POST PROCESS WITH APPLE PLOT        **
5100  INPUT "FILE NAME ? ";NA$
5120 D$ = "": REM  CTRL-D
5140  PRINT D$;"OPEN ";NA$
5160  PRINT D$;"WRITE ";NA$
5180  PRINT TT
5200  PRINT 0
5220  FOR Q = 1 TO TT
5240  PRINT Q - 1
5260  PRINT XP(Q)
5280  NEXT Q
5300  PRINT D$;"CLOSE ";NA$
5400  RETURN
5500  REM   **          END OF SUBROUTINE             **
5600  REM   ********************************************
```

]

VITA

Joseph Anthony Carretto, Jr, was born on 30 January 1957 in West Hempstead, New York. He graduated from Sidney Lanier High School in Montgomery Alabama in 1974. He attended the Massachusetts Institute of Technology from 1974 to 1978, receiving the degree of Bachelor of Science in Electrical Engineering in June 1978. He attended M.I.T. on a full Reserve Officers Training Corps (ROTC) scholarship, serving as the Cadet Corps commander his final year and graduating as a Distinguished Graduate. Before entering active duty with the Air Force, he worked for the Lockheed Electronics Company at the Johnson Space Center, Houston, Texas. He was responsible for developing redundancy management algorithms for portions of the Space Shuttle simulation software. Upon entering active duty, he was assigned to the Reference Systems Branch of the Avionics Laboratory at Wright-Patterson Air Foce Base, working as a project manager for Navigation Technology Development. He entered the School of Engineering of the Air Force Institute of Technology as a part-time student in July 1979.

Permanent Address: 1595 Parkridge Drive
Springfield, Ohio 45506

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/82S-15	2. GOVT ACCESSION NO. AD-A12743	3. RECIPIENT'S CATALOG NUMBER 4
4. TITLE (and Subtitle) DIGITAL CORRELATION OF SPREAD-SPECTRUM SIGNALS VIA FREQUENCY DOMAIN PROCESSING		5. TYPE OF REPORT & PERIOD COVERED MS THESIS 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) JOSEPH A. [illegible]FTTO, JR 1LT, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AIR FORCE INSTITUTE OF TECHNOLOGY (AFIT-EN) WRIGHT-PATTERSON AFB OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS ELECTRICAL ENGINEERING DEPARTMENT AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB, OH 45433		12. REPORT DATE July 1982 13. NUMBER OF PAGES 66
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release: IAW AFR 190-17.

LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

7 APR 1983

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

FFT
FDP
FREQUENCY DOMAIN PROCESSING
SPREAD SPECTRUM PROCESSING
DIGITAL SIGNAL PROCESSING

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

BACKGROUND INFORMATION ON THE THEORY BEHIND PROCESSING SPREAD-SPECTRUM SYNCHRONIZATION DATA IN THE FREQUENCY DOMAIN IS PRESENTED. THE DERIVATION SHOWS THAT THE TIME DOMAIN PSUEDO-NOISE SEQUENCE OFFSET IS REPRESENTED IN THE FREQUENCY DOMAIN BY FREQUENCY MODULATION ON A KNOWN CARRIER WAVE. FOLLOWING THE BACKGROUND DEVELOPMENT IS A DETAILED DESCRIPTION OF THE ANALOG HARDWARE USED TO IMPLEMENT THE FREQUENCY DOMAIN PROCESSING TECHNIQUES. A PROPOSED METHOD OF DIGITALLY EXTRACTING THE SYNCHRONIZATION DATA BY FOURIER TRANSFORMING

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

THE ANALOG OUTPUT SIGNAL IS PRESENTED. THE DIGITAL PROCESSOR DESIGNED AND BUILT TO SAMPLE, AVERAGE, AND CALCULATE A FAST FOURIER TRANSFORM OF THE ANALOG DATA IS DESCRIBED IN DETAIL. RESULTS OBTAINED FROM BOTH THE ANALOG AND DIGITAL HARDWARE ARE DISCUSSED, COMPARING THE ACTUAL RESULTS TO THOSE PREDICTED FOR VARIOUS INPUTS. THE RESULTS VALIDATE THE PROCESSING CONCEPTS, AND INDICATE AREAS FOR FURTHER RESEARCH.

END

DATE
FILMED

5 -83

DTIC