AN INTERFERENCE MONITOR FOR A RADIO OBSERVATORY

by

DAVID N. ROMALO


A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

Department of Electrical Engineering

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

April, 1988

© David N. Romalo, 1988
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia
Vancouver, Canada

Date April 29, 1988
ABSTRACT

This thesis describes the design, construction, and testing of a radio-frequency interference monitoring system for use with the synthesis array telescope at the Dominion Radio Astrophysical Observatory near Penticton, B.C. The system is designed to provide continuous, automated surveillance of the radiospectrum around 408 MHz. Interfering signals are characterized and catalogued according to strength, duration, frequency, and direction. Although the monitor is presently a very useful tool for detecting and finding sources of interference, it is ultimately intended to communicate directly with the telescope's control computer, so that sporadic bursts of interference can be removed automatically.

The system can detect a weak interfering signal that is within 5 dB of the smallest signal that can contaminate the astronomical observations. The smallest signal was calculated based on the following conditions, considered to be the case for which a synthesis telescope is most sensitive to interference, i.e., the worst case:

a) observing at high declination (towards the North Pole), so that the fringes of the synthesis telescope are too slow to reduce the effects of the interference, and

b) with the interference present continuously during the observation.

These weak signals can be detected in the presence of other signals, nearby in frequency, which are up to 40 dB stronger, i.e., the dynamic range of the monitor is 40 dB.
The monitor consists of an antenna system, a computer-tunable radio receiver, a fast Fourier transform (FFT) spectrum analyzer, and a microcomputer for control and data analysis. Everything except the microcomputer hardware was built as part of the project.

A thorough survey of the literature on the design of dedicated FFT machines was required. It was discovered that there had been no investigation of the design details for fixed-point FFT machines which are required to do long integrations. In such situations, fixed-point errors limit the performance of the machine. A computer simulation of the Welch process was developed to analyze the effects of these errors and to optimize the design. Some new results concerning the detectability of small signals are presented.

The FFT spectrum analyzer is used to estimate the power spectrum of 500 kHz-wide sub-bands using Welch's method of modified periodograms. It computes 256-point transforms in real-time with a resolution of 3.91 kHz (corresponding to one FFT every 512 µsec). This is comparable to the speed of a large array processor but at a fraction of the cost. Since the FFT is equivalent to a bank of contiguous filters, it can analyze the spectrum in much less time than the single swept filter found in most commercial spectrum analyzers, i.e., it is much more sensitive.

The analyzer was specially designed and built using recently-available digital integrated circuits. The design draws upon several high-speed architectural concepts including pipelining, parallel arithmetic, and hard-wired control. Except for
expensive array processors, the analyzer is much faster than any commercial FFT processors or FFT-based spectrum analyzers.

As part of the antenna system, an array of helical antennas was designed and constructed, its characteristics were investigated and found to be suitable for the present application, and a method of remotely switching them on and off was devised.

One more note - the radio spectrum is becoming more and more cluttered with man-made signals. Unprotected radio astronomy bands are being adversely affected and radio astronomers are turning to FFT spectrometers to cope with the relatively large interfering signals. The work herein on FFT-based design is applicable in such cases.
TABLE OF CONTENTS

Abstract .......................................................................................................................... ii

List of Tables .................................................................................................................. viii

List of Figures ................................................................................................................ ix

List of Photographs ......................................................................................................... xiii

Acknowledgements ......................................................................................................... xiv

Glossary .......................................................................................................................... xvi

1. INTRODUCTION ........................................................................................................ 1

   1.1. Earth-Rotation Aperture Synthesis ........................................................................ 2

       1.1.1. The Synthesis Telescope at DRAO ............................................................... 10

   1.2. Overview of the Interference Monitoring System .................................................. 15

2. THE INTERFERENCE PROBLEM .............................................................................. 19

   2.1. Sources of Interference ......................................................................................... 20

   2.2. The Effect of Interference on the Synthesis Telescope .......................................... 21

   2.3. Quantification of Interfering Signals .................................................................... 26

       2.3.1. Maximum Signal ......................................................................................... 26

       2.3.2. Minimum Detectable Signal and Dynamic Range ...................................... 29

3. THE FFT SPECTRUM ANALYZER ......................................................................... 31

   3.1. The Modified Periodogram .................................................................................. 32

       3.1.1. Windowing Effects ....................................................................................... 34

       3.1.2. Scaling of the Power Spectrum .................................................................. 40

       3.1.3. Power Spectrum of Real Data ................................................................... 41

       3.1.4. FFT Basics .................................................................................................. 44

   3.2. Survey of Real-Time Narrowband Processing Techniques .................................. 48

       3.2.1. General-Purpose Computers ..................................................................... 50

       3.2.2. Swept IF ..................................................................................................... 50

       3.2.3. One or Two Bit Correlation ........................................................................ 53

       3.2.4. Analog Filter Bank ..................................................................................... 54

       3.2.5. Walsh Functions ......................................................................................... 54

       3.2.6. DSP Chips ................................................................................................... 55

       3.2.7. Custom Hardware ...................................................................................... 55

   3.3. Architectural Alternatives .................................................................................... 58

       3.3.1. Double-Buffering ....................................................................................... 58

       3.3.2. Real-Time Bandwidth ................................................................................ 59

       3.3.3. Speeding Up the FFT ................................................................................ 61

           3.3.3.1. Parallel Butterfly Computations ......................................................... 61

           3.3.3.2. Higher Radix ....................................................................................... 63

           3.3.3.3. Overlapped Memory/Compute Cycles ................................................. 63

           3.3.3.4. Pipelined FFT Stages ....................................................................... 65

       3.3.4. FFT Addressing ......................................................................................... 65

       3.3.5. DIT versus DIF ......................................................................................... 66
Cited References ........................................................................................................ 179

Other References ....................................................................................................... 183

Appendix 1 - FORTRAN Program Listing of FFT Spectrum Analyzer
  Computer Simulation ....................................................................................................... 185

Appendix 2 - Schematic Diagrams of FFT Spectrum Analyzer ...................... 199

Appendix 3 - The Split-Window Normalizer .............................................................. 227
LIST OF TABLES

Table 3-1. Comparison of FFT Spectrum Analyzer Systems (1987) .................. 56

Table 3-2. Number of Occurrences of Unity Values of Unit-Circle Coefficients in 256-Point FFT .................................................................................. 87

Table 3-3. Read and Write Address Sequences for 16-Point Constant-Geometry DIF FFT .................................................................................................. 87

Table 3-4. Coefficient Address Sequence for 16-Point Constant-Geometry DIF FFT ........................................................................................................... 89

Table 3-5. Read and Write Address Sequences for Unscrambling Power Spectrum (N=16) .................................................................................................. 89

Table 3-6. Number of Occurrences of $\theta \mod 2\pi = \frac{\pi}{4}$ or $3\frac{\pi}{4}$ ............................................. 102

Table 4-1. Crossover Gain versus Number of Turns for Helical Array ........... 118

Table 4-2. Measured Terminal Impedance and Return Loss Helices ............. 123

Table 4-3. Measured Crosstalk Between Pairs of Helices ............................. 123
LIST OF FIGURES

Figure 1-1. Basic Interferometer ................................................................. 5
Figure 1-2a. Angular Responses of Various Antennas ........................................... 7
Figure 1-2b. Angular Responses of Various Antennas (cont’d) ............................. 8
Figure 1-3. Geometry and Coordinate Systems for Synthesis Mapping ................. 9
Figure 1-4. SST Antenna Configuration .......................................................... 11
Figure 1-5. Configuration of Interference Monitoring System and Observatory Computers ................................................................. 16
Figure 2-1. Example of Harmonics in Oscillator Output ..................................... 21
Figure 2-2. Example of Map Contaminated by Prolonged, Weak Interference ....... 24
Figure 3-1. Partitioning of Data Blocks in Welch’s Method of Power Spectrum Estimation ................................................................. 33
Figure 3-2. Rectangular Window ...................................................................... 36
Figure 3-3. Kaiser-Bessel (α = 2) Window ....................................................... 37
Figure 3-4. Flow Diagrams for DIT and DIF Butterflies ..................................... 46
Figure 3-5. Flow Diagram for DIF Butterfly as Implemented in the Analyzer ....... 47
Figure 3-6. Flow Diagram for 16-Point In-Place FFT ....................................... 49
Figure 3-7. Flow Diagram for 16-Point Constant Geometry FFT ....................... 49
Figure 3-8. Comparison of Swept IF and FFT Methods ..................................... 52
Figure 3-9. Comparison of Three Schemes for Performing Memory and Compute Cycles ........................................................................... 64
Figure 3-10a. Block Diagram of FFT Spectrum Analyzer ................................... 69
Figure 3-10b. Block Diagram of FFT Spectrum Analyzer (cont’d) ..................... 70
Figure 3-11. Board Layout and Inter-Board Signal Flow in FFT Spectrum Analyzer ..................................................................................... 71
Figure 3-12. Layout of A-D Chip on AD Board .................................................. 80
Figure 3-13. 500 kHz Cauer-Chebyshev Lowpass Anti-Aliasing Filter ............... 83
Figure 3-14. Frequency Response of 500 kHz Anti-Aliasing Filter
Figure 3-15. Kaiser-Bessel (a = 2) Window (8-Bit Quantization)
Figure 3-16. Upper Bounds on Butterfly Outputs
Figure 3-17. Comparison of Two Divide-By-2 Scaling Schemes
Figure 3-18. Simulated Performance of Analyzer for Various BFFT
Figure 4-1. Bird's Eye View of Helices
Figure 4-2. Cross-Section of Mainlobe of Helical Beampattern
Figure 4-3. Theoretical Response of Helix (in One Dimension)
Figure 4-4. FSK Transmitter (for Antenna Switch Control)
Figure 4-5. FSK Receiver (for Antenna Switch Control)
Figure 4-6. Triplexers
Figure 5-1. Generic Receiver System of Cascaded Elements
Figure 5-2. First Part of Receiver and Other Components (on Antenna Pole)
Figure 5-3. Second Part of Receiver (in SST Blockhouse, outside Screened Room)
Figure 5-4. Third Part of Receiver (in SST Blockhouse, inside Screened Room)
Figure 5-5. Imageless Mixer
Figure 5-6. All-Pass Networks in Imageless Mixer
Figure 6-1a. Plots from High-Speed Analyzer Tests
Figure 6-1b. Plots from High-Speed Analyzer Tests (cont'd)
Figure 6-1c. Plots from High-Speed Analyzer Tests (cont'd)
Figure 6-1d. Plots from High-Speed Analyzer Tests (cont'd)
Figure 6-1e. Plots from High-Speed Analyzer Tests (cont'd)
Figure 6-2. \( \sigma/\mu \) of Passband versus Input Level
Figure 6-3a. Plots from System Tests
Figure 6-3b. Plots from System Tests (cont'd) .................................................. 169
Figure 6-3c. Plots from System Tests (cont'd) .................................................. 170
Figure 6-3d. Plots from System Tests (cont'd) .................................................. 171
Figure 6-4a. Plots of Interfering Signals .............................................................. 172
Figure 6-4b. Plots of Interfering Signals (cont'd) ................................................. 173
Figure A2-1. Board Interface and FFT Memories (C&M Board) ......................... 200
Figure A2-2. Analog-to-Digital Converter (AD Board) ........................................ 201
Figure A2-3. Window Multiplication and Digitized Data Display (AD Board) ....... 202
Figure A2-4. Butterfly Circuitry (BUTT Board) ................................................... 203
Figure A2-5. Blowups of Butterfly Modules (BUTT Board) ............................... 204
Figure A2-6. Power Spectrum Computation and Accumulation Circuitry (PS Board) .................................................................................................................. 205
Figure A2-7. Output Buffer and Interface to Microcomputer (PS Board) .......... 206
Figure A2-8. Main Clock Generation (C&M Board) ............................................ 207
Figure A2-9. PS Clock Generation (C&M Board) ................................................ 208
Figure A2-10. AD and BUTT Control Signals, and Some Control Signals for Buses (C&M Board) ........................................................................................................ 209
Figure A2-11. Some Control Signals for Buses (C&M Board) ............................. 210
Figure A2-12. PS Control Signals (C&M Board) .................................................. 211
Figure A2-13. Write and Enable Signals for FFT Memories 1 and 2 (C&M Board) .................................................................................................................. 212
Figure A2-14. Write and Enable Signals for FFT Memory 3 and E5 (C&M Board) .................................................................................................................. 213
Figure A2-15. AD and Window Address Generation (C&M Board) ...................... 214
Figure A2-16. FFT Data Address Generation (C&M Board) .............................. 215
Figure A2-17. Bit-8 Generation for FFT Data Address (C&M Board) ............... 216
Figure A2-18. FFT Coefficient Address Generation (C&M Board) ................. 217
Figure A2-19. PS Address Generation (C&M Board) ........................................... 218
Figure A2-20. Address Multiplexers (C&M Board) .............................................. 219
Figure A2-21. Sample and Stage Counters (C&M Board) ..................................... 220
Figure A2-22. Integration (FFT) Counter (C&M Board) ........................................ 221
Figure A2-23. Reset Generation (C&M Board) ..................................................... 222
Figure A2-24. Overflow Indicators (AD Board) .................................................... 223
Figure A3-1. Double-Boxcar Function in Split-Window Normalizer ...................... 228
LIST OF PHOTOGRAPHS

Photograph 1-1. The Spectroscopic-Synthesis Telescope at DRAO ......................... 13

Photograph 3-1. FFT Spectrum Analyzer and Microcomputer (First View) .............. 73
Photograph 3-2. FFT Spectrum Analyzer and Microcomputer (Second View) .......... 75
Photograph 3-3. FFT Spectrum Analyzer and Microcomputer (Third View) .......... 77

Photograph 4-1. Mounting of Antenna Assembly (First View) .......................... 113
Photograph 4-2. Mounting of Antenna Assembly (Second View) ....................... 119
Photograph 4-3. Preliminary Tuning of Helix ............................................. 122
Photograph 4-4. Helix Feed ........................................................................... 122
Photograph 4-5. RF Switch PC Board (Signal Side) .......................................... 126

Photograph 4-6. Timing Diagram of SW Helix Being Turned On ....................... 132
Photograph 4-7. Timing Diagram of Transmission of Logic '1' and Logic '0' .... 132
Photograph 4-8. Timing Diagram of Transmission of One Logic '1' ................. 132

Photograph 5-1. First Part of Receiver and Other Components (on Antenna Pole) ........................................................................................................... 141

Photograph 5-2. Second Part of Receiver (in SST Blockhouse, outside Screened Room) .......................................................................................................... 143

Photograph 5-3. Third Part of Receiver (in SST Blockhouse, inside Screened Room) .......................................................................................................... 145
ACKNOWLEDGEMENTS

I would like to express my gratitude to those people who contributed to this thesis in one way or another. At the University of British Columbia, I thank my supervisor Dr. Mabo Ito for his support and interest in the project. I also thank Dr. Michael Beddoes for the use of his lab space and Mr. Tony Leugner for sharing his high-speed oscilloscope.

At the Dominion Radio Astrophysical Observatory, I thank my co-supervisor Dr. Peter Dewdney for his guidance, numerous technical suggestions, unwavering enthusiasm, and for making me a better engineer. I thank Dr. Tom Landecker for his continued interest in the project and for his excellent technical advice in the RF laboratory. I thank Mr. Ron Casorso for his help in the lab and for helping to install the Heliax cable, for our fruitful discussions on the design of the RF electronics, and for being a good listener. Thanks also to a host of other observatory employees: Mr. Ed Danallanko constructed and installed the helical antennas and antenna frame, machined parts of the digital cabinetry, and patiently tutored me in the machine shop. Mr. Gary Hovey maintained the C development system and helped with the C software. Messrs. Rod Stuart and Ev Sheehan helped me in the lab. In the office, Mmes. Erika Rohner, Cindy Furtado, and Bette Jones did various administrative work, and Mrs. Rohner handled the many purchase orders. Finally, I would like to express my sincere appreciation to all the members of the observatory for their warmth and general good nature.
I also thank Ms. Patricia Kavanagh for reviewing the thesis.

I received financial support from the Natural Sciences and Engineering Research Council with a Post-Graduate Scholarship and a Research Assistantship.
GLOSSARY

A-D analog-to-digital
AD Board analog-to-digital conversion (plus windowing) board
AF audio frequency
AWG American wire gauge
bin output point in frequency domain after FFT
(e.g. "k" in X(k), P(k))
BUTT Board butterfly board
bw bin-width (Hz)
CG coherent gain
CMOS complementary metal-oxide semiconductor
C&M Board control and FFT memories board
CW continuous-wave
dBi dB relative to isotropic
dBm dB relative to 1 milliWatt (into 50Ω)
DC direct current
DFT discrete Fourier transform
DIF decimation-in-frequency
DIT decimation-in-time
DR dynamic range
DRAO Dominion Radio Astrophysical Observatory
DSP digital signal processing
ENBW equivalent noise bandwidth
FFT fast Fourier transform
FSK frequency-shift keying
glue logic small-scale integration ICs (e.g. flip-flops, AND gates)
IC integrated circuit
IF intermediate frequency
IL insertion loss
I/O input/output
LNA low-noise amplifier
LO local oscillator
M number of stages in FFT (=log₂ N)
N length of FFT
NF noise figure
NMOS n-channel metal-oxide semiconductor
PC printed-circuit
PROM programmable read-only memory
PS Board power spectrum accumulator board
RAM random-access memory
RFI radio frequency interference
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>signal-to-noise ratio</td>
</tr>
<tr>
<td>SPICE</td>
<td>Simulation Program with Integrated Circuit Emphasis</td>
</tr>
<tr>
<td>SQSUM</td>
<td>square-of-sum of window values</td>
</tr>
<tr>
<td>SST</td>
<td>spectroscopic-synthesis telescope</td>
</tr>
<tr>
<td>SUM</td>
<td>sum-of-window values</td>
</tr>
<tr>
<td>SUMSQ</td>
<td>sum-of-squares of window values</td>
</tr>
<tr>
<td>TTL</td>
<td>transistor-transistor logic</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

At the Dominion Radio Astrophysical Observatory (DRAO) near Penticton, British Columbia, detailed maps of the sky are made using a technique known as earth-rotation aperture synthesis. The signals from astronomical radio sources are relatively weak compared with other signals. Therefore, it is important that interfering radiation be suppressed. In particular, the effects of man-made terrestrial and satellite radio frequency interference (RFI) must be minimized. It is for this reason that DRAO is situated in an isolated, radio-quiet mountain valley. However, experience has shown that, in the radio astronomy band 406.1—410 MHz, interfering signals occur sporadically with sufficient strength to distort the maps. The distortions cannot be completely removed by post-processing and hence are a source of error in the observation. Sometimes the distorting signal is so weak that its presence is not detected until the final map has been made after the 35-day observation period.

In view of these facts, it is desirable to have a system capable of continuous, automated surveillance of the RF environment around 408 MHz. The ideal detection system would provide information about the interfering signals such as strength, duration, frequency, and direction. Any signals large enough to distort a map would be quickly detected, tracked down, and (hopefully) eliminated. Also, the contaminated data could be flagged and removed from the observation.

The subject of this thesis is the design, construction, and testing of a radio-frequency interference monitoring system for DRAO. The system consists of
an antenna array, a sensitive RF receiver, a digital fast Fourier transform (FFT) spectrum analyzer, and a microcomputer for system control and data analysis. The system is currently in use at DRAO. It is unique in its construction and application.

The thesis work was carried out in two places. The construction of the FFT spectrum analyzer and the development of the computer simulation were done at the university. The remainder of the work took place at the observatory.

The remainder of this chapter is an introduction to earth-rotation aperture synthesis followed by an overview of the interference monitoring system. In the second chapter, the interference problem in radio astronomy is reviewed, and the application to the DRAO telescope is discussed. The third chapter is a detailed description of the FFT spectrum analyzer including a software simulation. The fourth and fifth chapters cover the antenna array and RF receiver, respectively. The sixth chapter describes testing of the spectrum analyzer, testing of the complete system, and system operation. The seventh chapter contains a summary and conclusions.

1.1. EARTH-ROTATION APERTURE SYNTHESIS [Chri85,Krau66,Thom82a]

The field of radio astronomy has flourished since its inception in 1932 when Karl Jansky observed radio emissions from the center of the Milky Way. At present, there are numerous radiotelescope sites throughout the world including Canada, the United States, England, the Netherlands, West Germany, Australia,
and Japan. Radio† observations are a very important part of the overall process of understanding astronomical objects. Often, they reveal sources and structure that are invisible at optical wavelengths. (In particular, radio emissions usually result from processes involving the gaseous components of the universe.) Two well-known discoveries in radio astronomy are pulsars and quasars.

A problem inherent in radiotelescopes is the lack of resolution due to the long wavelength (compared to optical telescopes). In general, for wavelength $\lambda$ and dimension $D$, an antenna aperture will have angular resolution $\approx \frac{\lambda}{D}$. A good optical telescope can resolve two sources that are as close as .5 arc sec. An equivalent radiotelescope at the shortest useful radio wavelength, a few mm, would be 1 km in diameter. Observations at longer wavelengths require even larger apertures. It is impractical to build receiving dishes‡ of such extent. The size is limited by the surface deformations caused by the dish's own weight. The largest fully-steerable dish in the world is 100 m in diameter (Effelsberg, West Germany). It is representative of the limit in the size of a radiotelescope dish.

A large aperture can be synthesized with one small, fixed antenna and one small, movable antenna. Phase and amplitude data can be recorded at various positions in the large "aperture" and the results added. Provided the sources are invariant over the observation period, the only theoretical difference

†There are only two "windows" in the electromagnetic spectrum for which both the atmosphere and ionosphere are transparent: radio and optical. Ground-based telescopes are limited to these two frequency bands. Orbiting telescopes, although limited in size, have available for observation the entire electromagnetic spectrum.
‡Except for the longest wavelengths, most radiotelescopes consist of one or more, often paraboloidal, reflecting "dishes" with a smaller collecting antenna at the focus.
between this method and that using a large antenna is the smaller collecting area (and hence lesser sensitivity).

In radio astronomy, the signal is usually much smaller than the unwanted noise level. Sources of noise include the receiver electronics, ground radiation, and radiation from atmospheric processes. Telescopes that measure total power require very stable equipment to prevent fluctuations in noise level from masking the signal. Better results can be obtained by correlating (multiplying), rather than adding, the data. The simplest correlating telescope is an interferometer pair (Figure 1-1). Ideally, such a telescope is sensitive only to signals that are received at both antennas simultaneously. Unwanted noise that is uncorrelated will average to zero. In the interferometer, the signals from the two antennas are multiplied and a lowpass filter extracts and averages the difference product. An instrumental delay \( \tau_i \) is inserted for steering. The one-dimensional complex response to a monochromatic plane wave of intensity \( S \) as a function of pointing angle \( \theta \) for \( \tau_i = 0 \) is shown in the figure. It can be modified to represent an actual telescope by including the beampattern of the interferometer \( A(s) \), signals from other sources \( S(s) \), and instrumental delay \( \tau_i \), and then transforming coordinates to two dimensions. The resulting response is known as the visibility \( V \) and is a function of the baseline vector \( B \) [Thom82a]:

\[
V(B) = \int A(s)S(s)Ge^{j2\pi f_0(B \cdot s/c - \tau_i)}d\Omega
\]

where the surface integral is over the celestial sphere, \( G \) is the system gain, \( s \)

\[\dagger\text{The "celestial sphere" is defined as having earth at its center and of arbitrarily large radius such that effects of the telescope's displacement from the center of the earth are negligible.}\]
\[ \tau_g = \frac{\vec{B} \cdot \hat{s}}{c} \]

\[ \tau_i = \text{inserted instrumental delay} \]

\[ G = \sqrt{G_1 \cdot G_2} \]

response to monochromatic plane wave of intensity \( S \):

\[ V(\theta) = S \cdot G \cdot e^{i2\pi f (\tau_g - \tau_i)} = S \cdot G \cdot e^{i2\pi (\theta \sin \theta / \lambda - f \tau_i)} \]

Figure 1-1. Basic Interferometer
is a unit-vector pointing towards the center of the sources, $f_0$ is the center frequency, boldface denotes a vector, and $V$ has units $Wm^{-2}Hz^{-1}$. The quantity we wish to measure is the brightness (intensity) distribution $S$ which has units $Wm^{-2}Hz^{-1}rad^{-2}$.

The response in one dimension of a multiplying interferometer to a point source (along with the response of a single dish) is plotted versus angle in Figure 1-2. Also plotted are the responses of a large dish of diameter equal to the interferometer spacing and an adding interferometer. The response of the multiplying interferometer is a sinusoid of frequency dependent on angle, with envelope equal to the response of a single dish. The lobes in the response are known as "fringes".

Now suppose the baseline is aligned in an east-west direction. As the earth rotates, the baseline sweeps around in an ellipse as viewed from a celestial source. The eccentricity of the ellipse depends on the declination of the source. The plane in which the vector rotates is called the u-v plane.

Figure 1-3 illustrates the geometry and coordinate systems for synthesis mapping. The x-y plane defines the map of the brightness distribution. As the two-dimensional interferometer pattern sweeps across the sky, the real part of the correlator output will undergo quasi-sinusoidal variations, or fringes. Samples of the correlator output, called visibilities, can be collected in the u-v plane by observing at different times and with different baseline lengths.

\[†\] For small angular displacements, the variation is sinusoidal. The frequency changes slowly with angle.
Figure 1-2a. Angular Responses of Various Antennas
Now consider transforming Equation 1-1 to rectangular coordinates by equating x and y to the direction cosines of $s_0$ and setting $w = \frac{f_0 \cdot B \cdot s_0}{c} = \tau_1$:

$$V(u, v, w) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(x, y)S(x, y)Ge^{j2\pi(ux + vy + w\sqrt{1-x^2-y^2})} \frac{dx \, dy}{\sqrt{1-x^2-y^2}}$$

(1-2)

The coordinates $u$ and $v$ are measured in units of wavelength of the center frequency. Normally, $u$ is taken as North and $v$ as East. If we make $w = 0$, i.e., $\tau_1 = \frac{2\pi B \cdot s_0}{c\lambda}$, so that the main beam of the interferometer response is steered towards the center of the region of interest, then Equation 1-2 reduces to a two-dimensional Fourier transform. The equation can be transformed to discrete form and the modified brightness distribution (equal to the product of the brightness distribution and the interferometer response) computed by inverse discrete Fourier transformation. This technique is known as earth-rotation aperture
Figure 1-3. Geometry and Coordinate Systems for Synthesis Mapping (copied from [Thom82a])
synthesis or spectroscopic-synthesis.

1.1.1. The Synthesis Telescope at DRAO

The DRAO spectroscopic-synthesis telescope (SST) is located in a semi-arid mountain valley 25 km south of Penticton. The site has two main advantages for radio astronomy: the mountains serve as a shield from outside RFI, and the flatness of the valley facilitates precise positioning of the antennas. The SST simultaneously conducts three types of observations: 1420 MHz continuum, 408 MHz continuum, and spectral line (narrowband) observations of neutral hydrogen emission at 1420 MHz.

Compared to other radiotelescopes, the DRAO SST has a wide field of view and moderate resolution. This makes it well suited for mapping objects of large angular extent, such as supernova remnants and HII regions in our own galaxy. As well, observations of extragalactic objects are made. Such objects include the Andromeda Galaxy and clusters of galaxies. In general, any process in the interstellar medium which emits radio waves is a suitable candidate for observations.

The telescope consists of four antennas. Each is a 9 m paraboloidal reflecting dish with dual-frequency receiving horns at the focus. The antennas are configured along a 600 m east-west baseline (Figure 1-4). Two are fixed at the ends and two are movable along a 300 m railway track. Photograph 1-1 shows the four antennas of the synthesis array.

†In the jargon of astrophysics, "continuum" implies broadband.
Four (of a possible six) interferometer pairs are implemented, each operating at a different spacing. The dishes track a region of sky for 12 hours (nominal) per day, corresponding to one half-ellipse in the u-v plane. Each day the spacings are changed by moving the middle dishes. After 35 days, 140 baselines have been sampled. The visibility data is then interpolated onto a rectangular grid and inverse (fast) Fourier transformed to yield the synthesized map. The size of the map (field of view) is defined by the main beam of an individual antenna. If desired, the map can be deconvolved with the interferometer response using one of various algorithms, the most popular of which is known as "CLEAN" [Högb74], [Stee84].

†Since the brightness distribution is a real quantity, the visibility matrix is Hermitian. Therefore, only one half of each ellipse needs to be sampled in the complex u-v plane. Visibilities in the other half can be computed by changing the sign of the imaginary part. (The baseline rotates 180° every 12 hours, equivalent to a sign change in the phase of the received signal.)
Photograph 1-1. The Spectroscopic-Synthesis Telescope at DRAO
The specifications of the 408 MHz system are summarized below:

- field of view = $7.4^\circ$ diameter circle
- resolution = $3.5 \times 3.5 \csc \delta$ arc min
- rms sensitivity = $3.3$ mJy/beam at center of field
- integration time per visibility = 90 sec
- baselines = 13 to 600 m in steps of 30/7 m
- bandwidth = 4 MHz

where 1 Jy (Jansky) equals $10^{-2.6}$ Wm$^{-2}$Hz$^{-1}$ and "beam" implies a single dish.

The total integration time $\tau$ for a full survey is:

$$\tau = (140 \text{ spacings})(12 \text{ hr/spacing})(3,600 \text{ sec/hr})$$

$$= 6,048,000 \text{ sec or 70 days}$$

over a period of 35 days.

Because Penticton is in the northern hemisphere (latitude = $49^\circ$) and the baselines are all oriented East-West, the telescope is suitable only for observing the northern sky. The practical range of source declinations$^\dagger$ is $20^\circ \leq \delta \leq 90^\circ$. (For low declinations, the resolution in the y dimension ($3.5 \csc \delta$ arcmin) is too coarse and the view of more southerly objects is blocked by the earth.)

$^\dagger$Declination is defined such that it is $0^\circ$ at the equator, $+90^\circ$ at the North Pole, and $-90^\circ$ at the South Pole.
1.2. OVERVIEW OF THE INTERFERENCE MONITORING SYSTEM

A block diagram of the interference monitor and the observatory computers is given in Figure 1-5. The interference monitor is a self-contained unit. It consists of an array of small, electronically-selectable antennas, a computer-tunable radio receiver, a fast Fourier transform (FFT) spectrum analyzer, and a microcomputer for control and data analysis. Except for the microcomputer, all of the hardware was built as part of this project.

The system provides continuous, automated surveillance of the radio band 398—418 MHz with 4 kHz resolution. Narrowband interfering signals are characterized and catalogued according to strength, duration, frequency, and direction. A resolution of 4 kHz is adequate for detecting continuous-wave (CW) and voice-modulated signals (the most predominant types of interference) and is partly the result of design constraints.

The monitor can detect interference in much less time than the 35-day integration period of the synthesis telescope because it improves the signal-to-noise ratio by:

1) looking directly at the source of interference rather than through the sidelobes of the telescope’s antennas and
2) analyzing a much narrower bandwidth.

The antenna system includes eight directional helices for scanning in azimuth and a quarter-wave whip for omni-directional surveillance. The antennas and an electronics box are located on a pole 400’ away from the building that contains the rest of the system. Via the use of special filters, the DC power,
Figure 1-5. Configuration of Interference Monitoring System and Observatory Computers
antenna control signal, and RF signal are frequency-shared on a single cable that runs between the pole and the building.

The receiver is based on the conventional superheterodyne design and includes a variable (computer-programmable) local oscillator. The receiver amplifies and shifts a 500 kHz sub-band within the band of interest down to baseband for spectrum analysis.

The FFT spectrum analyzer computes and integrates 500 kHz-wide spectra in real-time. It was built using recently-available digital integrated circuits. It estimates the power spectrum using Welch’s method of modified periodograms. It performs 256-point transforms with a real-time bandwidth of 500 kHz and a resolution of 3.91 kHz (corresponding to one FFT every 512 μsec). This is comparable to the speed of a large array processor. Since the FFT is equivalent to a bank of contiguous filters, it can analyze the spectrum in much less time than the single swept filter found in most commercial spectrum analyzers. Or, in other words, the analyzer is much more sensitive per unit time.

The 68008-based microcomputer is the system controller. Programmed in the C language, it accepts spectra from the analyzer, processes the data, and controls the other hardware. Presently, the data processing includes normalization followed by a threshold-based search for narrowband signals. The microcomputer controls the other hardware by:

a) electronically selecting one of 8 helices or the quarter-wave whip,

b) tuning the receiver to a desired frequency band, and
c) specifying the sampling rate and integration time of the analyzer (the latter over a range from a few seconds to many hours).

If desired, information can be sent to the synthesis telescope's control computer. Presently, a listing of the detections, or "hits", is automatically stored in the control computer. Data in the control computer can be manually transferred over a remote link to the main observatory computer (in the main building) for further analysis and plotting.
2. THE INTERFERENCE PROBLEM

To measure the weak radio signals from outer space, radiotelescopes must use very sensitive instruments. The 408 MHz SST system at DRAO has a sensitivity of 3.3 mJy/beam. This is equivalent to the level received from a typical FM radio station 10,000,000,000 miles away!

Unfortunately, the extreme sensitivity of the radiotelescope leaves it prone to interference. In particular, the multiplying interferometer is adversely affected by interfering signals that are correlated between antennas. For this reason, the SST is located in a radio-quiet mountain valley far from any major cities. However, experience has shown that interfering signals occur often enough to impede and contaminate the observations. Interference is a significant problem for the 408 MHz array. It is less significant for the 1420 MHz array. Strong interference is easily found in the observational data, but is very laborious to remove. Prolonged weak interference is equally serious, and much more difficult to remove.

In this chapter, the 408 MHz interference problem is examined. In particular, the sources of possible interference are listed, the effect of interference on the synthesis telescope is discussed, and the levels of large and small interfering signals are estimated.
2.1. SOURCES OF INTERFERENCE

Experience has shown that RFI at DRAO can originate from many different sources. Some examples of sources that have caused problems in the past include the police transmitter on nearby Mount Kobau, radar equipment at Penticton airport, transmitters aboard orbiting satellites, and the Sun. Unfortunately, the band 406.1–410 MHz is not reserved solely for radio astronomy. Included in the allocation for the band are various mobile and mobile-satellite transmitters [NTC82]. Interference problems due to legal, in-band transmitters occur now and then. The observatory’s only resort in these cases is to request that the party not use the band.

In addition to interference from sources located outside the valley, interference from DRAO equipment can be a problem. Fundamental and higher harmonic frequencies and their intermodulation products are generated from telescope instrumentation and electronics under development in the lab (e.g. Figure 2-1). It is impractical to shield all possible sources of interference at the observatory.

Other sources of RFI in the valley are vehicles that travel to and from DRAO. Transceivers in vehicles have caused problems in the past. Also, it is conjectured that ignition noise may cause significant interference.

Except for ignition noise and the Sun, the common characteristic of most interfering signals is that they are narrowband (narrow relative to the 4 MHz band of observation).
2.2. THE EFFECT OF INTERFERENCE ON THE SYNTHESIS TELESCOPE

There are four ways in which the synthesis telescope discriminates against interference:

1. the fringe oscillations attenuate signals not in phase with those from the region of interest;
2. the sidelobes of the primary (single dish) beampattern suppress signals that are outside the main lobe;
3. the front-end filters suppress out-of-band signals; and
4. broadband interfering signals are decorrelated between antennas.

These will be discussed below. In so doing, the effect of interference on the synthesis telescope, and its susceptibility to interference, will be elucidated.

1) In general, a signal that is not in phase with the signals from the
region of observation will sweep through the fringes of the interferometer response (Figure 1-2). For a source of interference that is stationary relative to Earth, the attenuation due to fringing is:

\[
\text{fringe attenuation} = \frac{1}{\tau} \int_0^\tau \cos(2\pi ft) dt = \frac{\sin(2\pi f\tau)}{2\pi f\tau}
\]

where

\[
\tau = \text{integration time for visibility point (sec)}
\]

\[
f = \text{natural fringe frequency (Hz)}.
\]

The fringe frequency is given by:

\[
f = \omega u \cos \delta
\]

where

\[
\omega = \text{angular rotation velocity of earth (rad/sec)}
\]

\[
u = \text{coordinate in u-v plane (East direction) (\lambda)}
\]

\[
\delta = \text{declination of source of observation (deg)}
\]

The reduction in interference due to fringing goes away as \( f \) nears 0. This occurs for:

a) points near the v-axis (\( u \) near 0), i.e., hour angle near \( \pm 90^\circ \), and

b) any point in the u-v plane when the source of observation is at a high declination.

Thus, when observing away from the pole, prolonged stationary interference will
contaminate visibilities along the v-axis. A two-dimensional inverse Fourier transformation into the map (x-y) plane will create an east-west interference structure (e.g. horizontal stripes). This is equivalent to the response to a source close to the North celestial pole,† the stripes being the sidelobes of the polar source.

Figure 2-2 shows prolonged weak interference in a map of the supernova remnant, HB3. The concentric rings are artifacts due to instrumental errors in phase and amplitude, not interference.

If the interference is randomly intermittent, the u-v plane will be contaminated in a random fashion, and the map plane distortion will not be predictable.

2) The main lobe in the beampattern of a single antenna defines the field of view. For the 408 MHz system, this is a 7.4° diameter circular region. In general, only airborne or spaceborne interference can enter the mainlobe. Other interfering signals are attenuated relative to the main lobe. The directivity of the 408 MHz beampattern is 32 dB above isotropic. The average sidelobe level is approximately 35 dB [Land88]. Therefore, the attenuation of a signal moving through the sidelobes (as the antennas track the source) is about 35 dB = −3 dBi.

3) In the 408 MHz SST, the electronic system has bandpass filters for image rejection (before down-mixing). They have a bandwidth of 10 MHz.

†The precise position of the fictitious polar source depends on the exact frequency of the interference.
Figure 2-2. Example of Map Contaminated by Prolonged, Weak Interference
Subsequent filtering and amplification confines the bandwidth to 4 MHz, equivalent to the radioastronomy band 406–410 MHz. Interference outside the astronomy band will be attenuated. Strong interference outside the astronomy band but within the 10 MHz initial bandwidth can cause problems (e.g. saturation of amplifiers.)

Also, there is a notch filter of bandwidth 120 kHz centered at 408 MHz (implemented at baseband). This suppresses a harmonic of 204 MHz generated by the system electronics.

4) Since the geometrical time delays introduced between each antenna and the correlators are, in general, different, broadband interference is attenuated by decorrelation. For this reason, narrowband interference is usually more harmful. A rough calculation below estimates the minimum bandwidth of an interfering signal necessary for significant decorrelation:

\[
\text{max. time delay between antenna pair} = \frac{\text{maximum baseline}}{c} = \frac{600 \text{ m}}{3 \times 10^8 \text{ m/s}} = 2 \mu\text{s}
\]

\[
\text{bandwidth} \approx \frac{1}{\text{correlation time}}
\]

\[
\text{minimum bandwidth} \approx \frac{1}{2 \mu\text{s}} \approx 500 \text{ kHz}.
\]
2.3. QUANTIFICATION OF INTERFERING SIGNALS

In order to design the monitoring system, it is necessary to know the amplitudes of the signals it should be capable of detecting. The ratio of the largest to smallest signals of interest specifies the dynamic range (DR) of the system. The smallest signal determines how sensitive the monitor should be.

2.3.1. Maximum Signal

A simple experiment was performed to estimate the largest interfering signal the analyzer should be able to handle without distortion. The experiment also served as a useful test of part of the receiver.

A λ/4 "stub" antenna (wire monopole with 4-wire ground-plane) [Jasi61] connected to a signal generator was used to transmit continuous-wave (CW) interference at 408.5 MHz into the sidelobes of the array's parabolic dishes. (The frequency was not exactly 408 MHz because the array system incorporates a notch filter at that frequency to suppress a harmonic of 204 MHz generated by the system electronics.) The transmitter was set up about 90 m south of the east-west baseline of the array. The telescopes were tracking a point fixed on the sky at the time so that the signal was sweeping through the sidelobes of the antennas in the array. The average output of the telescope receiving system would correspond to a rough average of the sidelobe level.

The signal was also observed with a receiver system comprising a helical antenna, a superheterodyne receiver, and a commercial spectrum analyzer. The receiver system was used to measure, at the center of the array, the power
being transmitted. The superheterodyne receiver was a partial prototype of that now used in the interference monitor. In particular, it consisted of everything before the imageless mixer (section 4.3). The receiver was set up in the spectroscopic-synthesis telescope (SST) control building located at the center of the array.

The signal level was increased until the SST visibility traces were just noticeably distorted; the traces, plotted on a chart recorder, display the complex cross-power of signals received from the four interferometer pairs in the array. This level represents a very strong interfering signal. A signal this strong is easily detected by someone scanning the chart recorder output traces. The level was measured to be $-65$ dBm into the stub.

The following shows a breakdown of the various gains and losses† in the transmitter/receiver combination. The flux at the array center is:

$$P_i = P_G \quad \text{(generator power in dBW)}$$

- $-0.36$ (generator-stub mismatch loss in dB)

+ $2.15$ (directivity of stub in dB)

$- 11.0 - 20 \log r$ (spherical spreading loss in dBm$^{-2}$, $r \approx 176$ m)

$$P_i = P_G - 54.12 \quad \text{dBWm}^{-2}.$$ 

The gain of the helix-plus-receiver is:

†A note on units - "dBW" implies that the value in Watts has been changed to $10\log(\text{value})$, ditto for dBm$^{-2}$, etc.
GR = -3.01 (polarization loss of helix in dB)
- 1.10 (helix-receiver mismatch loss in dB)
+ .26 (aperture area of helix in dBm²)
+ 73.7 (gain of superheterodyne receiver in dB)

GR = 69.85 dBm².

(In the above equation, the transmitter is assumed to be linearly polarized.) The signal level at the output of the receiver is then:

PR = Pi GR

PR = PG + 15.73 dBW.

The measured level verified this calculation to within 4 dB:

PR = PG + 12 dBW (measured).

(In the Pi and GR equations, generator power and receiver gain were measured; the other values were calculated.)

The generator power level of -65 dBm corresponds to a power flux density at the array center equal to:

Pi max = -149 dBW/m².

In the 4 MHz array bandwidth, this corresponds to an equivalent broadband signal with spectral power flux density at the array center equal to:
\[
P_{\text{max}} / \beta = S_{\text{max}} = -215 \text{ dBWm}^{-2}\text{Hz}^{-1} \approx 31,000 \text{ Jy}
\]

where 1 Jy (Jansky) equals \(10^{-26}\) Wm\(^{-2}\)Hz\(^{-1}\) and bandwidth \(\beta\) is 4 MHz.

### 2.3.2. Minimum Detectable Signal and Dynamic Range

We shall define the minimum detectable signal to be the smallest flux at the center of the array that may distort the map. To calculate the minimum detectable signal, we shall assume the worse-case condition: observing at high declination (towards the North Pole, so that averaging over large fringe rotation angles does not reduce the interference (section 2.2)) with the interference in the radioastronomy band and continuously present for the full survey. The rms sensitivity of the array is given by [Thom82a]:

\[
\Delta S = \frac{2 \sqrt{2} k T_s}{\sqrt{\beta \tau \eta_c \eta_a A_a}} \text{ Wm}^{-2}\text{Hz}^{-1}
\]

where \(k = 1.38 \times 10^{-23} \text{ JK}^{-1}\) (Boltzmann’s constant), \(\tau = \text{total (effective) integration time, and for the 408 MHz array [Veid84]:}

\[
T_s = \text{system noise temperature} = 104 \text{ K}
\]

\(\beta = \text{bandwidth} = 4 \text{ MHz}
\]

\(\eta_c = \text{correlator efficiency} = .89
\]

\(\eta_a = \text{ aperture efficiency} = .6
\]

\(A_a = \text{aperture area of dish} = \pi(4.5 \text{ m})^2.
\]

For a full survey, \(\tau = 6 \times 10^6 \text{ sec (section 1.1.1). The effective directivity of a dish is } D_e = 4\pi \eta_a A_a / \lambda^2\). We can rewrite the sensitivity as:
Assuming the response in the sidelobes of the dish beampattern is the average sidelobe level, i.e., setting $D_e = -3 \text{ dBi} = .5$, and assuming that the smallest, continuously present, bothersome signal level $S_{i_{\text{min}}}$ is equal to the sensitivity $\Delta S$, we calculate for a full survey:

$$S_{i_{\text{min}}} = -254 \text{ dBWm}^{-2}\text{Hz}^{-1} \approx 4 \text{ Jy}.$$  

Thus,

$$S_{i_{\text{min}}} \beta = P_{i_{\text{min}}} = -188 \text{ dBWm}^{-2}.$$  

This represents the smallest flux at the array center that may distort the map.

We shall define "dynamic range" $\text{DR}$ to be the range of amplitudes over which signals can be accurately measured, including the case when the largest and smallest signals are both present. Based on the above results, the dynamic range of the spectrum analyzer should be:

$$\text{DR} = \frac{P_{i_{\text{max}}}}{P_{i_{\text{min}}}} = \frac{-149 \text{ dBWm}^{-2}}{-188 \text{ dBWm}^{-2}} \approx 39 \text{ dB}.$$  

The dynamic range is a major consideration in the design of the various components in the interference monitor.
3. THE FFT SPECTRUM ANALYZER

As stated earlier, almost all interfering signals are narrowband. The basic problem, then, is the detection and estimation of narrowband signals in broadband noise. Therefore, some form of narrowband spectrum analysis is required. Such analysis has three main advantages over a broadband approach: it

1. increases signal-to-noise ratio, thus decreasing integration time;
2. discriminates between signals of different frequencies; and
3. can extract spectral details of a modulated signal (e.g. shape of sidebands).

Assuming that the interfering signals are smoothly-varying sinusoids or modulated sinusoids, a "natural" method to use is Fourier analysis (natural because it uses sinusoids as the basis vectors).† An approximation to the continuous Fourier transform is the discrete Fourier transform (DFT).‡ An efficient DFT algorithm amenable to digital hardware implementation is the fast Fourier transform (FFT). The power spectrum can be estimated by squaring and averaging the complex spectra produced by FFTs of successive time segments.*

The interference monitor incorporates an FFT spectrum analyzer to perform the narrowband processing. The analyzer is a digital hardware

†Indeed, when the signal is narrowband and has constant amplitude, random phase, and unknown frequency, an infinite set of Fourier filters followed by power integration and thresholding is the optimum detector (in the maximum likelihood sense) [Will72].
‡With a concomitant degradation in performance (relative to optimum).
*Averaging of the complex spectra is pointless because the relative phase of the signal varies randomly so the complex spectral values would average to zero.
implementation of Welch's modified periodogram power spectrum estimator [Welc67]. It was specially designed and constructed by the author and includes over 200 integrated circuits. The decision to build an FFT analyzer was made only after careful consideration of other processing methods and a study of products available on the market.

The sections in this chapter contain the following: a review of Welch's method of modified periodograms, comments on windowing and scaling, a derivation showing how to compute the power spectrum of real data based on the standard "two-in-one-FFT" trick, and a review of some basic FFT concepts; a survey of real-time narrowband processing techniques; a discussion of alternatives in the design architecture; details on the design and construction of the analyzer; a discussion of the errors due to fixed-point number representation and computation in the analyzer; and results of a computer simulation of the analyzer.

3.1. THE MODIFIED PERIODOGRAM

The following is a description of Welch's method of power spectrum estimation using modified periodograms [Welc67].

Given an input signal $d(t)$, consider sampling it (equally-spaced samples) and partitioning the resulting data sequence into $K$ blocks $d_i(n)$, possibly overlapping, each of length $N$ (Figure 3-1). Now, multiply each block by a window $w(n)$, $n=0,1,...,N-1$, and take the DFT of each windowed block $x_i(n) = d_i(n)w(n)$ to get the complex spectra:
Figure 3-1. Partitioning of Data Blocks in Welch's Method of Power Spectrum Estimation

\[ X_j(k) = \sum_{n=0}^{N-1} x_j(n) e^{-j2\pi kn/N} \quad ; \quad k = 0, 1, \ldots, N-1. \]  

(3-1)

Then, take the magnitude-squared of the complex spectra and average the resulting power spectra (modified periodograms)† to get the estimate of the power spectrum of \( d(t) \):

\[ P(k) = \frac{1}{K} \sum_{i=0}^{K-1} |X_i(k)|^2 \quad ; \quad k = 0, 1, \ldots, N-1. \]

\[ \]

\[ P(k) \] is a "good" estimate in that it is asymptotically unbiased and

†"Periodogram" is a historical term referring to the magnitude-squared of the complex spectrum estimate. "Modified" means that a non-rectangular window is applied.
consistent. For input noise with stationary, second-order statistics and non-overlapping data blocks, the variance of the estimate is given by:

\[ \sigma^2(k) = \frac{P^2(k)}{K} \quad ; \quad k=0,1,...,N-1. \]  

(3-2)

3.1.1. Windowing Effects

There is one main reason for applying a tapered (non-rectangular) window to the data blocks in the power spectrum estimation process: reduction of spectral leakage. Leakage occurs because the continuous power spectrum \( P(f) \) of the estimate equals the convolution of the continuous power spectrum \( D(f) \) of the sampled data with the magnitude-squared of the finite Fourier transform \( W(f) \) of the window function:

\[
P(f) = \frac{1}{2} \int D(f') |W(f-f')|^2 df' - \frac{1}{2}
\]

where

\[
W(f) = \text{finite Fourier transform of } w(n) = \sum_{n=0}^{N-1} w(n)e^{-j2\pi fn}
\]

and \( f = \) normalized frequency (relative to sampling rate). If the input is a complex sinusoid at the center of bin \( k_0 \), the sampled power spectrum estimate \( P(k) \) equals \( |W(k-k_0)|^2 \), not an impulse. The leakage from a large signal can mask a small nearby signal. The lower the sidelobes of the window, the less the leakage.
The window used in the analyzer is the Kaiser-Bessel (sometimes called Taylor). It is designed to maximize the energy in the mainlobe for a given first (highest) sidelobe level.† This is a suitable selection criterion for the detection of narrowband signals. The window values are defined in terms of the zero-order modified Bessel function of the first kind [Harr78]:

\[ w(n) = \frac{I_0[\pi a \sqrt{1 - (2n/N - 1)^2}]}{I_0[\pi a]} \quad ; \quad n = 0, 1, \ldots, N - 1 \]

where \( a \) = design parameter. The window is "DFT-even" [Harr78], i.e., \( w(1) = w(N-1), \ w(2) = w(N-2), \) etc., such that its DFT is real. The level of the first sidelobe is given by:

\[ \text{first sidelobe level} = -10 \log [4.603 \sinh[\pi a] \frac{\pi a}{\pi a} ]^2 \ (\text{dB}). \]

The parameter \( a \) was chosen to give a first sidelobe level commensurate with the analyzer's dynamic range DR.

The chief detriment to using a tapered window is the reduction of frequency resolution due to the widening of the window's mainlobe. The rectangular window offers the best resolution. It has a 3 dB-bandwidth of .89 bin and first sidelobe level \(-13.46 \) dB. The corresponding parameters for the Kaiser-Bessel \((a = 2)\) window are 1.43 bin and \(-45.85 \) dB. Plots of the power spectra of the two windows are shown in Figures 3-2 and 3-3. Each plot was generated by padding the 256-point window with 768 zeros, FFTing, then taking the magnitude-squared. Included in the figures is a comparison of various

†The optimization assumes continuous time and frequency variables. Eberhard has derived an optimization procedure based on discrete time and continuous frequency variables [Eber73]. Unfortunately, the method is tractable only for small N.
Rectangular Window (Time Domain)

Rectangular Window (Magnitude-Squared, Frequency Domain)

Figure 3-2. Rectangular Window
Figure 3-3. Kaiser-Bessel ($\alpha = 2$) Window
parameters.

A side benefit of using a tapered window is the reduction in scalloping loss. Scalloping refers to the loss in power when the signal is not located at the center of a bin. The worst case occurs when the signal is halfway between centers. The crossover point of adjacent filters determines the scalloping loss. As the mainlobe is widened, the scalloping loss decreases.

It should be pointed out that, in theory, spectral leakage can be eliminated by deconvolving the spectrum with the power spectrum of the window. However, this can be difficult when there are multiple signals, and nearly impossible when there is noise. Also, in theory, scalloping loss can be recovered by interpolating the spectrum. However, in a real-time system, it may be impractical to do post-processing. In such a case, the windowing effects are irrecoverable.

Another detriment to using a tapered window is the reduction in SNR for bin-center signals, with a concomitant increase in integration time. There are two effects to consider. First, the widened mainlobe means more broadband noise will appear within each bin. A measure of this noise increase is equivalent noise bandwidth (ENBW). ENBW is defined as the width of a rectangular-shaped filter that would accumulate the same noise power as the actual window filter (each filter having the same peak power gain). Harris [Harr78] shows that:

\[
\text{ENBW} = \frac{N \text{SUMSQ}}{\text{SQSUM}}
\]
where it is convenient to define the following terms:

\[
\text{SUM} = \sum_{n=0}^{N-1} w(n)
\]

\[
\text{SQSUM} = (\text{SUM})^2
\]

\[
\text{SUMSQ} = \sum_{n=0}^{N-1} w^2(n).
\]

The second effect is the decrease in energy (for both signal and noise) due to the window taper, i.e., amplitude reduction. A measure of this energy reduction is coherent gain \(CG\). Harris shows that:

\[
CG = \frac{\text{SUM}}{N}.
\]

The loss in SNR for a bin-center signal due to windowing is then:

\[
\text{loss in SNR (bin-center)} = \frac{\text{signal reduction}}{\text{noise increase}} = \frac{(CG)^2}{(CG)^2 \text{ENBW}} = \frac{1}{\text{ENBW}}.
\]

The loss in SNR for a bin-center signal for rectangular and Kaiser-Bessel (\(a=2\)) windowing are 0.00 dB and 1.75 dB, respectively. For a signal halfway between centers:

\[
\text{loss in SNR (worst-case)} = \frac{1}{\text{ENBW}} + \text{scalloping loss}.
\]

The worst-case losses in SNR for rectangular and Kaiser-Bessel (\(a=2\)) windowing are 3.92 dB and 3.21 dB, respectively. We see that a tapered window degrades the SNR for bin-center signals but actually improves the SNR for signals halfway between centers.

Another benefit of tapered windows is realized when the input data
segments are overlapped. Overlapping allows more averages per unit time. Averaging $K$ identically-distributed independent measurements reduces the noise variance by a factor $1/K$. However, overlapped data blocks are not independent so the variance reduction is somewhat less, i.e.,

$$\text{variance reduction} = cK ; 0 < c \leq 1$$

where $c$ is a constant whose value depends on the window, the amount of overlap, and the number of spectra in the average [Harr78]. $c$ is an indirect measure of the correlation between data segments due to overlap. It decreases as the amount of tapering increases. For overlapped processing, a tapered window allows greater variance reduction than a rectangular window. With 50% overlap and large $K$, $c = .67$ for the rectangular window and .95 for the Kaiser-Bessel ($\alpha = 2$) window.

In section 3.5.3, it is noted that a tapered window reduces the probability of overflow in the FFT computations.

### 3.1.2. Scaling of the Power Spectrum

To obtain an absolute (properly scaled) measurement from the Welch spectrum, a number of effects must be taken into account. Some important points are listed below.

1. There is an $N$-fold inflation in the power spectral values inherent in the DFT, re. Parseval’s theorem:

$$N\left[ \sum_{n=0}^{N-1} |x(n)|^2 \right] = \left[ \sum_{k=0}^{K-1} |X(k)|^2 \right]$$

2. The window taper in the time-domain deflates the energy by a factor
3. The widened mainlobe (relative to rectangle window) in the frequency domain inflates the energy of broadband noise by a factor ENBW.

4. A sinusoid will appear in both sides of the spectrum, so the energy observed will be deflated by 2.

5. White noise will be evenly distributed amongst the frequency bins.

To obtain an absolute measurement, the power spectral estimate must be scaled to account for the above effects. In the case of white noise, $P(k)$ should be multiplied by:

$$\frac{1}{N(CG)^2 ENBW} = \frac{1}{N SUMSQ}.$$

In the case of a bin-center sinusoid, $P(k)$ should be multiplied by:

$$\frac{2}{N(CG)^2} = \frac{2N}{SQSUM}.$$

3.1.3. Power Spectrum of Real Data

We want to compute and average the power spectra of data blocks which can be represented as real numbers. We shall show that we can compute the sum (average) of the power spectra of two real data blocks using one DFT followed by a simple "unscrambling" of the power spectrum. It is not necessary to first unscramble the individual complex spectra (using formulas available in some DSP textbooks) and then add their squared-magnitudes. (Welch has shown this result before [Welc67], but his derivation is somewhat brief and also contains three errors.)
Since the input data is real, an improvement in speed can be realized by packing two data blocks in the real and imaginary parts of memory and reconstructing the output spectra. Incidentally, other methods of exploiting the realness of the data were investigated: packing the even and odd samples of one data block (less bandwidth), packing the two halves of one data block (cannot be unscrambled), or complex-mixing to yield real and imaginary input data streams thus halving the sampling rate (too costly).

Consider two real data blocks, \( f(n) \) and \( g(n) \). Form the complex array \( x(n) \):

\[
x(n) = f(n) + jg(n) \quad ; \quad n = 0, 1, \ldots, N - 1.
\]

The DFT of each side is:

\[
X(k) = F(k) + jG(k) \quad ; \quad k = 0, 1, \ldots, N - 1.
\]

By explicitly writing \( F(k) \) and \( G(k) \) in terms of sine and cosine series and remembering that \( f(n) \) and \( g(n) \) are real, it is straightforward to show that:

\[
F(k) = X_{ER}(k) + jX_{OI}(k)
\]

\[
G(k) = X_{EI}(k) - jX_{OR}(k)
\]

where \( E, O, R, \) and \( I \) denote even, odd, real, and imaginary parts, respectively. The even and odd symmetries (modulo \( N \)) of \( X_{ER} \) and \( X_{OI} \) enable us to write:
\[ F(k) = \frac{1}{2} \left[ X_R(k) + X_R(N-k) + jX_I(k) - jX_I(N-k) \right] \]
\[ = \frac{1}{2} \left[ X(k) + X^*(N-k) \right]. \]

Similarly,
\[ G(k) = \frac{1}{2} \left[ X_I(k) + X_I(N-k) - jX_R(k) + jX_R(N-k) \right] \]
\[ = \frac{1}{2} j \left[ X(k) - X^*(N-k) \right]. \]

The sum of the power spectra of the two data blocks is:
\[ P(k) = F(k) F^*(k) + G(k) G^*(k) \]
\[ = \frac{1}{4} \left[ X(k) + X^*(N-k) \right]\left[ X^*(k) + X(N-k) \right] + \frac{1}{4} \left[ X(k) - X^*(N-k) \right]\left[ X^*(k) - X(N-k) \right] \]
\[ = \frac{1}{4} X(k) X^*(k) + \frac{1}{4} X(N-k) X^*(N-k) + \frac{1}{4} X(k) X(N-k) \]
\[ + \frac{1}{4} X^*(k) X^*(N-k) \]
\[ + \frac{1}{4} X(k) X^*(k) + \frac{1}{4} X(N-k) X^*(N-k) - \frac{1}{4} X(k) X(N-k) \]
\[ + \frac{1}{4} X^*(k) X^*(N-k) \]
\[ = \frac{1}{2} \left[ X(k) X^*(k) + X(N-k) X^*(N-k) \right] \]
\[ = \frac{1}{2} \left[ X_R^2(k) + X_I^2(k) + X_R^2(N-k) + X_I^2(N-k) \right]; k = 0,1,\ldots,(N/2-1) \]

where we only consider the first N/2 points because the power spectrum of real signals is symmetric.

Thus, we see that it is possible to compute the sum of the power spectra of two real data blocks using one DFT followed by a simple "unscrambling" of
the power spectrum. We don't have to compute \( F(k) \) and \( G(k) \) explicitly from \( X(k) \) and then add the individual power spectra.

In the implementation, the two data blocks are adjacent, i.e.,

\[
\begin{align*}
  f(n) &= x_i(n) \\
  g(n) &= x_{i+1}(n) \\
  &; n = 0,1,\ldots,N-1 \\
  &; i = 0,2,4,\ldots,K-2
\end{align*}
\]

where \( K \) is the number of blocks per estimate (an even number equal to twice the number of actual FFTs).

3.1.4. FFT Basics

In this section, some basic concepts needed to understand the implementation of the FFT algorithm are presented and some relevant terms are defined.

The FFT is a computationally efficient method of evaluating the DFT in Equation 3-1 (section 3.1). In the derivation of the radix-2 FFT, in which \( N \) must be an integer power of 2, redundant\(^\dagger\) multiplications are removed by decomposing the DFT into successively smaller DFTs until the computation consists solely of 2-point DFTs, or "butterflies". There are two classes of FFT: "decimation-in-time (DIT)" and "decimation-in-frequency (DIF)". In the DIT algorithm, the decomposition begins with the data \( x(n) \). In the DIF algorithm, the decomposition

\(^\dagger\)Redundancy is present in the DFT because of the modularity and symmetry of the complex exponential.
begins with the complex spectrum $X(k)$. In either case, the number of complex
multiplications and additions is reduced from $N^2$ in the DFT to $N \log_2 N$. For
more details on the derivation of the FFT, the reader is referred to the excellent
textbook by Oppenheim and Schafer [Oppe75].

The DIT and DIF butterflies are illustrated in Figure 3-4. We see that
the butterfly process has two complex inputs, $A$ and $B$, two complex outputs, $X$
and $Y$, and a complex multiplier $C$ (unit-circle coefficient). The DIT and DIF
butterflies each require the same number of arithmetic operations.

Figure 3-5 is a diagram of the DIF butterfly showing the real arithmetic
operations as implemented in the FFT spectrum analyzer. The divide-by-2
elements are explained in section 3.5.3.

The complete FFT consists of a series of butterflies arranged in groups
called "stages" or "passes". Figures 3-6 and 3-7 are flow diagrams for two FFTs
having different addressing schemes. Each node represents one butterfly. The
number before/after each node is the exponent in the unit-circle coefficient for the
DIT/DIF algorithm. In each stage, the butterflies are computed in order from top
to bottom.

There are $M = \log_2 N$ stages in the FFT and $N/2$ butterflies per stage.
Each stage can be considered as a transformation of one $N$-point input array
into another $N$-point array, the first input array being the original data $x(n)$ and
the last output array being the DFT result $X(k)$. 
\[ \cos \theta - j \sin \theta \quad \theta = \frac{2\pi kn}{N} \]

Figure 3-4. Flow Diagrams for DIT and DIF Butterflies
$C_R = \cos \theta$

$C_I = -\sin \theta$

$X_R = [A_R + B_R]/2$

$X_I = [A_I + B_I]/2$

$Y_R = [(B_R - A_R)(-C_R) + (B_I - A_I)C_I]/2$

$Y_I = [(B_I - A_I)(-C_R) - (B_R - A_R)C_I]/2$

Figure 3-5. Flow Diagram for DIF Butterfly as Implemented in the Analyzer
A phenomenon known as "bit-reversal" is inherent in the FFT algorithm. The FFT results $X(k)$ are produced in bit-reversed order, i.e., if $k$ is written as a binary number and its bits are reversed to form a new index, say $k_{rev}$, then the results $X(k_{rev})$ will be in the order $k_{rev} = 0,1,...,N-1$. Some versions of the FFT algorithm call for bit-reversed inputs. These produce normally-ordered outputs.

The DFT in Equation 3-1 is equivalent to a bank of finite-impulse response filters.† The shape of an individual filter is defined by the window function. The filters are spaced in frequency by "bin-width" $bw$, defined as:

$$bw = \frac{f_s}{N} = \frac{1}{T} \text{ (Hz)}$$

where $f_s$ = sampling rate (samples/sec), $N$ = length of DFT, and $T$ = block duration (sec).

The minimum sampling rate (Nyquist rate) for avoiding aliasing is:

$$f_s = 2BW$$

where $BW$ = bandwidth of input signal.

### 3.2. SURVEY OF REAL-TIME NARROWBAND PROCESSING TECHNIQUES

Before choosing to build a special-purpose FFT spectrum analyzer, numerous methods of real-time narrowband processing were considered. These included general-purpose computing, swept IF, one or two bit correlation, analog

†This can be shown by considering the DFT as a "complex heterodyne to baseband and then low-pass filter" process.
Figure 3-6. Flow Diagram for 16-Point In-Place FFT
(from [Rabi75], courtesy of L.R. Rabiner)

Figure 3-7. Flow Diagram for 16-Point Constant-Geometry FFT
(from [Rabi75], courtesy of L.R. Rabiner)
filter banks, Walsh functions, DSP chip-based systems, and special-purpose custom hardware. The latter was chosen for the interference monitor. Details of the advantages and disadvantages of the various methods are given in the following sections. We shall define "real-time" to mean that, for a given analysis bandwidth, no data is discarded in the processing, e.g. in the Welch process, where blocks of data are processed continuously, real-time implies there are no gaps between blocks.

3.2.1. General-Purpose Computers

Since we require a real-time wide-bandwidth system, the narrowband processor must be dedicated to the task at hand. This requirement coupled with cost-efficiency rules out mini- and larger computers as well as array processors. The remaining possibility is a high-performance microcomputer. Its speed can be used efficiently if extra hardware is added to double-buffer the I/O. Algorithms other than the ubiquitous FFT can be programmed, e.g. maximum entropy, Prony, or other "exotic" estimation techniques. However, a survey of available processors showed that the bandwidth of such a system is too low for our purposes (≈ 100 Hz – 1 kHz using the FFT, less bandwidth with other algorithms).

3.2.2. Swept IF

Almost all commercial spectrum analyzers use the "swept IF" method to measure the power in a narrow band. As shown in Figure 3-8, the image-suppressed analog signal is mixed down to an intermediate frequency (IF) and passed through a narrow bandpass filter. The filtered signal is then detected
and sampled. The frequency of interest is selected by setting the frequency of the local oscillator (LO). To obtain a power spectrum, the LO is swept over a range of frequencies. Swept IF methods are convenient for analyzing high frequencies because of their relative simplicity.

A display can be generated by synchronizing the LO with the display scan rate. The vertical deflection is produced by a DC level from a lowpass filtered version of the detected power, i.e., there is no explicit sampling.

The range of signals we want to detect includes signals that are buried in the noise. Averaging of the power spectra must be done to reduce the fluctuations due to noise. Averaging $K$ independent samples reduces the noise variance by a factor $1/K$. The swept IF method can produce independent measurements spaced no closer than $1/BW = T$ seconds apart, where $BW =$ bandwidth of the IF filter in Hz. This is true regardless of whether the LO is fixed or sweeping. On the other hand, the FFT method (also shown in Figure 3-8) produces $N/2$ independent samples every $T$ sec where $BW$ equals the bin-width. Therefore, for a given resolution and level of detectability, the swept IF approach requires $N/2$ times more averaging time than the FFT. In other words, for a given resolution and integration time, the FFT is much more sensitive (can detect smaller signals). Since we want to detect signals quickly, the swept IF method is less suitable than a "multi-filter" method such as the FFT.
Swept IF Method

image-suppressed real analog data

\[ \text{bandpass filter} \quad (BW = 1/T) \]

step frequency every T sec

power detect

sample every T sec

1 power spectrum sample every T sec

FFT Method

(lowpass-filtered real analog data)

\[ (BW = N/(2T)) \]

A/D \[ \rightarrow \] N-point FFT \[ \rightarrow \] \[ \left| \right|^2 \]

N/2 power spectrum samples every T sec

Figure 3-8. Comparison of Swept IF and FFT Methods
3.2.3. One or Two Bit Correlation

Rather than averaging the power spectra of successive FFTs, it is possible to average auto-correlations and then do one FFT to get the power spectrum estimate. However, except for small values of \( N \), correlation is much less efficient than direct FFT analysis \( (O(N^2) \) versus \( O(N \log_2 N) \) computations).

A technique often used in radiotelescopes is "one or two bit correlation". The incoming signals are quantized to a few levels and the correlation (both cross and auto) and integration are done with special-purpose hardware. Because the wordlength is very short, the circuitry can use very simple multipliers and adders. High-speed logic allows large real-time bandwidths to be achieved, e.g. 10–20 MHz.

Coarse quantization does not degrade the SNR excessively if the input signal is much smaller than the noise [Vlec66]. For example, one bit (two level) quantization degrades the SNR by only 1.96 dB [Bowe74] when the \( \text{SNR} \ll 1 \).

The interference monitor, however, must handle a wide dynamic range of signal levels (from \( \text{SNR} \ll 1 \) to \( \text{SNR} \gg 1 \)). Under circumstances of one or more powerful signals, coarse quantization and correlation would cause two problems. First, large signals would be severely clipped. This would destroy the magnitude information and generate numerous harmonics. The harmonics (in-band and aliased counterparts) would clutter the spectrum and mask smaller signals. "Decluttering" the spectrum is thought to be infeasible. Second, the statistics of the auto-correlator would be distorted from Gaussian, thereby producing an
irreducible "noise".

3.2.4. Analog Filter Bank

The power spectrum can be measured by detecting and integrating the outputs of a bank of contiguous, fixed-frequency bandpass filters. A large bandwidth can be achieved. However, there are drawbacks. First, it can be difficult to realize the filters if the ratio (filter bandwidth)/(center frequency) is small. Second, analog circuitry is prone to drift problems due to aging and temperature variations. Third, the amount of hardware grows linearly with the number of channels. To match the analyzer used in this project, 128 4 kHz-wide bandpass filters (plus detectors and integrators) would have to be built, and the problem of sampling their outputs would have to be solved. This would be both expensive and time-consuming.

3.2.5. Walsh Functions

The Walsh transform [Beau84,Blac74] is similar to the Fourier transform, except that the basis set consists of binary waveforms, not complex sinusoids. It is conceivable that fairly simple hardware could be designed to perform the Walsh transform very quickly. Multiplication is not necessary because the basis set only takes on the values +1 and −1.

Unfortunately, the Walsh transform is not amenable to the processing of sinusoids. It is not surprising that the Walsh transform of a sinusoid yields a similar result to the Fourier transform of a square wave, i.e., a spectrum cluttered with odd harmonics or their aliased counterparts. For this reason, the
design of a Walsh processor was not pursued.

3.2.6. DSP Chips

The last decade has seen the advent and growth of the digital signal processing (DSP) chip, a microprocessor tailored to rapidly execute DSP routines, most notably those algorithms involving sum-of-products expressions like

$$\sum_{n=0}^{N-1} a(n)x(n)$$

(e.g. the DFT and filter convolution). If I/O double-buffering circuitry is added to a DSP chip-based system, the throughput is limited only by the load-compute-unload time of the processor. A survey of available processors when the project began showed that a fixed-point FFT spectrum analyzer could be built with real-time bandwidth in the range 10—100 kHz.† A two processor system that doubled the bandwidth appeared feasible. Adding more processors made the system too complex. Some commercially-available DSP chip-based FFT processors are compared in Table 3-1.‡ Although there are faster DSP chips than those in the table, they have not yet been incorporated into commercial machines.

3.2.7. Custom Hardware

In general, for any DSP problem, the highest throughput solution will be to build highly-specialized hardware. The window-FFT-power spectrum process* can be hard-wired with high-speed logic chips such as multipliers, adders, counters,

†A note of caution: most published specifications of FFT execution times do not include the time to load and unload the data.
‡The bandwidths of the systems in Table 3-1 and those designed in the author's surveys of digital technology are compared based on the following assumptions: bin-width ≈ 4 kHz; windowing, FFT, and power calculation all done by processor; "two-FFTs-in-one-trick" used on real data; and anti-aliasing filter ignored.
*In general, other power spectrum estimation methods are too complex to hard-wire.
<table>
<thead>
<tr>
<th>System</th>
<th>Architecture</th>
<th>Real-Time</th>
<th>Arithmetic</th>
<th>Cost (Canada)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rapid Systems R340 IBM PC FFT Peripheral</td>
<td>DSP chip-based (TMS32010)</td>
<td>25 kHz</td>
<td>16-bit fixed</td>
<td>$2,400</td>
<td>hardware addressing, N = 1024 only, not programmable</td>
</tr>
<tr>
<td>IQS System 416-2-210 Vector Signal Processor</td>
<td>DSP chip-based (ZORAN ZR34161)</td>
<td>28 kHz</td>
<td>16-bit block floating-point</td>
<td>$11,900</td>
<td>requires IBM PC host, some programming, 2 channels available</td>
</tr>
<tr>
<td>Burr-Brown SPV120 VMEbus FFT Module</td>
<td>DSP chip-based (TMS32020)</td>
<td>90 kHz</td>
<td>16-bit fixed</td>
<td>$5,300</td>
<td>A-D not included, programmable</td>
</tr>
<tr>
<td>Elsin Corp. FFT-101 VMEbus FFT Module</td>
<td>multipliers with hardwired microcode</td>
<td>262 kHz</td>
<td>floating-point</td>
<td>$48,000</td>
<td>A-D not included, programmable, available 4Q 1987</td>
</tr>
<tr>
<td>Motorola T-ASP II Array Processor (4 AU’s)</td>
<td>TTL bit-slice, parallel pipelines</td>
<td>2,000 kHz</td>
<td>floating-point</td>
<td>$700,000</td>
<td>A-D not included, multi-tasking, vector programmable</td>
</tr>
<tr>
<td>FFT spectrum analyzer built by author</td>
<td>multipliers with hardwired microcode</td>
<td>500 kHz</td>
<td>16-bit fixed</td>
<td>$1,500 (parts only)</td>
<td>highly specialized, N = 256 only, not programmable</td>
</tr>
</tbody>
</table>

* rolloff of anti-aliasing filter not considered

Table 3-1. Comparison of FFT Spectrum Analyzer Systems (1987)
memories, PROMs, etc. Such a machine is essentially non-programmable.

A number of special-purpose non-commercial FFT processors have been built previously [Ali78,Kasp81]. The specialized machines in the literature were not suitable of directly applicable to the present application, e.g. they were too slow, too complex and expensive, and/or performed only the FFT. Also, no thorough treatment of all facets of the design of a Welch processor was found in the literature. It was decided to build a special-purpose machine that executes the Welch process at high speed.

The FFT spectrum analyzer in the interference monitor is an original design by the author. Custom hardware was designed and built using recently-available off-the-shelf small-scale and medium-scale integrated circuits (actually, the multipliers are large-scale ICs). It has a real-time bandwidth of 500 kHz (without considering the roll-off of the anti-aliasing filter). The design is a compromise between speed and simplicity, the latter determining cost and development time.

Digital technology is evolving at a startling pace. The speed and density of integrated circuits increase by significant amounts every year. New signal processing chips are continually making their debut on the commercial market. Therefore, the design of custom hardware requires a "jump in and build" attitude, i.e., a line must be drawn and the system designed from available components. This was certainly the case in the design of the analyzer.
In the next section, various concepts in the design of custom hardware are expanded upon.

3.3. ARCHITECTURAL ALTERNATIVES

During the development of the FFT spectrum analyzer, a number of different design options were considered. Most choices involved the ubiquitous tradeoff between cost/complexity and speed. Unfortunately, since the analyzer was entirely a "one-man project", the more complex (and costly) schemes had to be forgone in lieu of reality. In fact, a detailed early design ended up being scaled down to the present version. Nevertheless, the investigation into various architectural alternatives was both instructive and fruitful.

The following sections describe some of the design options considered in the development of the analyzer. Included are a few well-known schemes and some new ideas.

3.3.1. Double-Buffering

By adding extra memory banks (buffers), a process can be divided into sub-processes that run simultaneously. The throughput is then limited by the slowest sub-process.

In Welch's periodogram method, the processes "A-D and windowing", "FFT", "power spectrum computation/accumulation", and "data upload to host" can all be double-buffered to form a pipeline. For example, two memories can be placed between the A-D/window and FFT processes. When one memory is filling
up with windowed data, the other is being FFT’d (Figure 3-10).

Double-buffering requires the ability to switch data lines, and, in some cases, control and address lines. The difficulty incurred in swapping data lines can be ameliorated by using memory chips that have separate input/output (I/O) data lines. Less switching hardware is required compared to systems with memory chips having common I/O. Unfortunately, separate I/O memory chips could not be used in the analyzer because they did not fit standard socket spacings!

When pipelined in the above fashion, the throughput of the analyzer is limited by the FFT time, i.e., the FFT is the bottleneck.

### 3.3.2. Real-Time Bandwidth

When double-buffering is used, the speed of the FFT limits the data rate (bandwidth). If a single butterfly unit is implemented in hardware, and if the real input data is packed into complex arrays, two blocks each, and the data is sampled at the Nyquist rate, then the unit must process $1 \text{FFT} = (N/2)\log_2 N$ butterflies in the time it takes for two data blocks to be collected ($2N/f_s$ seconds). For this case, it then follows that:

$$T_{\text{BUTT}} = \frac{4}{f_s \log_2 N} \text{ sec} = \frac{4}{N \text{ bw} \log_2 N} \text{ sec} \quad (3-3)$$

where $N =$ length of FFT, $bw =$ bin-width (Hz), $f_s =$ sampling rate (Hz), and $T_{\text{BUTT}} =$ butterfly execution time (sec). This equation can be used to select $N$, $bw$, and $T_{\text{BUTT}}$. Since $N = 2^M$ in the FFT algorithm, the choices for $N$ and
bw vary in jumps for a given $T_{\text{BUTT}}$.

For a given bin-width, decreasing $T_{\text{BUTT}}$ causes a less than linear increase in bandwidth $\text{BW} = f_s/2$ because of the logarithm in Equation 3-3. For example, to double the bandwidth, $T_{\text{BUTT}}$ must be reduced by more than a factor of two:

for $\text{bw} = 4 \text{ kHz}$:

- $\rightarrow N = 128, T_{\text{BUTT}} = 1116 \text{ ns}, \text{ BW} = 256 \text{ kHz}$
- $\rightarrow N = 256, T_{\text{BUTT}} = 488 \text{ ns}, \text{ BW} = 512 \text{ kHz}$

$\frac{512}{256} = 2.04\approx 2.3$  
$\frac{512}{1116} = 0.5$  
$\frac{512}{1116} > 1.0$  
$\frac{512}{1116} > \frac{2.3}{0.5}$

The 20 MHz bandwidth of interest is too wide to process in real-time. For a given frequency resolution, there are three practical alternatives for processing the data in such a case: use a long FFT such that there are gaps between blocks; use non-overlapped, contiguous blocks; or use overlapped blocks. The second method, in which the band is broken up into sub-bands each of size equal to the maximum-attainable real-time bandwidth† of the processor, is the most efficient.‡ The first method is less efficient because of the logarithm in Equation 3-3. In the third method, the sub-bandwidth is traded off with overlap.

†Maximum-attainable real-time bandwidth is defined as the maximum bandwidth that can be processed using non-overlapped, contiguous data segments.
‡Here, efficiency is defined as the percentage, on average, of time samples used in the FFT computation for a given frequency bin.
Recall that by overlapping data blocks, more averaging per unit time can be obtained (section 3.1.1). However, since the overlapped blocks are correlated, the variance reduction is not quite linear with the number of averages. Therefore, it is less efficient to overlap the data blocks.

3.3.3. Speeding Up the FFT

The FFT is the bottleneck in a double-buffered Welch processor and hence it was the object of most of the efforts to boost throughput. The butterfly is the basic task in the FFT. A butterfly involves: reading two complex words from memory, computing the butterfly, and writing two new complex words to memory. The computation consists of 10 arithmetic operations: 4 multiplications, 3 additions, and 3 subtractions. The minimum custom hardware needed to do an FFT is: one arithmetic processor, one (complex) data memory, one (complex) coefficient memory, and a control and address generator. The processor does the 10 operations sequentially and the memory is continuously overwritten with new results. There are a number of schemes used to boost speed as described in the following sub-sections.

3.3.3.1. Parallel Butterfly Computations

The butterfly can be hard-wired with individual units for each of the 10 operations. This requires adding I/O latches to hold data and tri-state buffers for sharing data on each of the real and imaginary buses. If the word length is long, it may be necessary to use bit-slice designs, e.g. cascading four 4-bit adder chips to form a single 16-bit adder.
Some of the calculations can be performed with the same hardware. In particular, multiplier/accumulator chips can be used to do multiply/add or multiply/subtract operations. The extra time to do an addition (or subtraction) after a multiplication is usually small.

Also, the computations can be pipelined, i.e., divided into sub-computations operating simultaneously, e.g. in the DIF butterfly:

| scaling | adds, subs | mults | adds, subs |

This requires putting latches between stages. The limited access speeds of present-day memories imply that such an architecture requires two memories operating simultaneously to make the scheme efficient, one memory providing input data, the other accepting results. Three memories would then be required to do the double-buffering between the A-D/window process and the FFT. With present technology, the DIF butterfly is particularly well-suited to a three-section pipelined design because it is "better balanced" (the adds and subs before and after the mults can be combined with the scaling and I/O latch delays). This scheme is the basis of a higher-bandwidth design considered earlier in the project. The control circuitry proved to be too complex (time-consuming) for a one-man project, although a detailed and feasible design was worked out. Some of the necessary control functions are: flush butterfly pipeline, context switch of triple memory buffer, and address pipelining.
3.3.3.2. Higher Radix

A radix higher than two can be used to reduce the number of butterflies in the FFT algorithm [Rabi75]. For example, radix-4 requires \((N/4)\log_4 N\) butterflies. However, the butterfly is now a 4-input/4-output process with 34 arithmetic operations and the hardware is far more complex than radix-2. The memory access time becomes a major bottleneck in such a system. Also, the choices of \(N\) are more restricted, i.e., \(N = 4^M\).

3.3.3.3. Overlapped Memory/Compute Cycles

The simplest butterfly processor performs the read-compute-write operations sequentially. When the memory and compute cycle times are comparable, an improvement in throughput can be realized by overlapping the cycles. Of course, extra control is needed, the trickiest problem being the design of circuitry to produce interleaved timing signals at the FFT stage boundaries (e.g. the memory and compute cycles must end at slightly different times).

Figure 3-9 illustrates three schemes: no overlap, total overlap when cycle times are equal, and partial overlap.

In the analyzer, a compromise is made between total overlap and no overlap. Referring to Figures 3-5 and A2-4, we observe that results \(X_R\) and \(X_I\) are derived from additions only. \(Y_R\) and \(Y_I\) must each go through two stages of additions (or subtractions) and one stage of multiplications. Therefore, after both inputs \(A\) and \(B\) have been latched, \(X\) may be written out before \(Y\) has finished.
Non-Overlapped Cycles

<table>
<thead>
<tr>
<th>read</th>
<th>read</th>
<th>write</th>
<th>write</th>
<th>read</th>
<th>read</th>
<th>write</th>
<th>write</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$B_1$</td>
<td>$X_1$</td>
<td>$Y_1$</td>
<td>$A_2$</td>
<td>$B_2$</td>
<td>$X_2$</td>
<td>$Y_2$</td>
</tr>
</tbody>
</table>

Compute cycles:

| compute butterfly 1 | compute butterfly 2 |

Overlapped Cycles

<table>
<thead>
<tr>
<th>read</th>
<th>read</th>
<th>read</th>
<th>read</th>
<th>write</th>
<th>write</th>
<th>read</th>
<th>read</th>
<th>write</th>
<th>write</th>
<th>read</th>
<th>read</th>
<th>write</th>
<th>write</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$B_1$</td>
<td>$A_2$</td>
<td>$B_2$</td>
<td>$X_1$</td>
<td>$Y_1$</td>
<td>$A_3$</td>
<td>$B_3$</td>
<td>$X_2$</td>
<td>$Y_2$</td>
<td>$A_4$</td>
<td>$B_4$</td>
<td>$X_3$</td>
<td>$Y_3$</td>
</tr>
</tbody>
</table>

Compute cycles:

| compute butterfly 1 | compute butterfly 2 | compute butterfly 3 |

Partially-Overlapped Cycles (as used in Analyzer)

<table>
<thead>
<tr>
<th>read</th>
<th>read</th>
<th>write</th>
<th>write</th>
<th>read</th>
<th>read</th>
<th>write</th>
<th>write</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$B_1$</td>
<td>$X_1$</td>
<td>$Y_1$</td>
<td>$A_2$</td>
<td>$B_2$</td>
<td>$X_2$</td>
<td>$Y_2$</td>
</tr>
</tbody>
</table>

Compute cycles:

| compute butterfly 1 | compute butterfly 2 |

Figure 3-9. Comparison of Three Schemes for Performing Memory and Compute Cycles
being computed† A similar overlap is possible in the DIT algorithm in which A is read in while the product BC is being computed.

3.3.3.4. Pipelined FFT Stages

The FFT can be divided into $M = \log_2 N$ pipelined processors, each performing one pass (stage) in the algorithm [Rabi75]. When double-buffered, the throughput is limited by the time to process one stage of butterflies. Because results within one FFT are not overwritten, constant geometry (section 3.3.4) can be used to simplify data addressing. One processor-memory unit can be designed and then replicated. The only difference between processors is the contents of the unit-circle coefficient PROMs. The only global control required is the distribution of a master clock.

The cost of such systems increases by the number of processors.

3.3.4. FFT Addressing

There are two basic schemes for data addressing in the FFT: in-place and constant-geometry [Rabi75]. These are illustrated in Figures 3-6 and 3-7. In the in-place method, the butterfly outputs are written to the same locations from where the inputs were read. This is convenient, particularly in software implementations. The address sequence changes from stage to stage. In constant-geometry, the outputs are, in general, not written to the input locations. The address sequence is the same for each stage.

†After the analyzer was built, it was noticed that Ali used a similar technique in his FFT processor [Ali78].
To prevent unprocessed data from being overwritten, the results of a constant-geometry stage must be placed in a second memory, i.e., constant-geometry requires twice the memory of in-place. The advantage of constant-geometry is simpler data address generation circuitry.

In both schemes, the coefficient address sequences change from stage to stage, and the circuitry for their generation is comparable in complexity.

Bit-reversal of the data addresses is inherent in the FFT algorithm. Normally-ordered inputs result in bit-reversed outputs, and vice-versa. Care must be taken to select an addressing scheme commensurate with the input order, i.e., a normally-ordered input algorithm operating on bit-reversed data will produce garbage. Bit-reversal is unique in that it is easier to perform in hardware (simply reverse address lines) than in software!

3.3.5. DIT versus DIF

From a design viewpoint, the DIT and DIF algorithms are quite similar. The final choice for the implementation was a flip of a coin. Each requires the same number of multiplications, additions, and subtractions. The address sequences are different but the generation circuitry is basically the same. However, there are a few noteworthy differences. These are summarized below.

DIT:
- partial overlap of memory read and compute cycles (section 3.3.3.3)
- less chance of overflow (section 3.5.3)

DIF:
- partial overlap of memory write and compute cycles (section 3.3.3.3)
- evidence of less roundoff error [Sund77]
- more efficient when butterfly is pipelined (section 3.3.3.1)
3.4. DESIGN AND CONSTRUCTION OF THE FFT SPECTRUM ANALYZER

The FFT spectrum analyzer is a digital hardware implementation of Welch's modified periodogram power spectrum estimator. The 500 kHz bandwidth-limited analog signal is digitized, windowed, FFT'd, squared, and averaged, all in real-time.

To achieve wide bandwidth, custom hardware was designed and built "from the ground up". The design is synchronous and incorporates a central clock from which all other signals are (directly or indirectly) derived. Hard-wired and PROM microcoding is used for control sequencing. The circuitry includes 221 IC's and occupies three 7"x16" wire-wrap boards (total number of wraps = 8384).

The performance specifications of the FFT spectrum analyzer are listed below.

- FFT length \( N = 256 \)
- resolution = 3.91 kHz  
  = 244 Hz (ZOOM mode)
- real-time bandwidth = 500 kHz  
  = 31.25 kHz (ZOOM mode)
- sampling rate = 1 Megasamples/sec  
  = 62.5 ksamples/sec (ZOOM mode)
- dynamic range = 46 dB
- integration time = 2–32 sec  
  = .125–2 sec (ZOOM mode)
- power spectrum = 128 32-bit samples
- A-D word length = 8 bits (2's complement)
- window word length = 8 bits (unsigned)
FFT word length = 16 bits (2's complement)

power spectrum word length = 16 bits (unsigned)

accumulator width = 32 bits

power requirements = 10 A @ 5 V, 40 mA @ 12 V

3.4.1. Design Overview

The analyzer is divided into four boards: analog-to digital conversion (plus windowing) board (AD Board), butterfly board (BUTT Board), power spectrum accumulator board (PS Board), and control and FFT memories board (C&M Board). The C&M Board houses the control circuitry and the FFT memories. The other three boards are "slaves" that perform the computations. Figure 3-10 is a block diagram of the process. It illustrates the flow of data but does not include address and control blocks. Figure 3-11 illustrates the physical layout of the boards and shows the flow of inter-board signals.

High speed was an important design goal. This was achieved by double-buffering the processing sections and using recently available high-speed hardware. Double-buffering is used between the following pairs of processes: A-D conversion (plus windowing) and FFT, FFT and power spectrum computation, and between power spectrum computation and the microcomputer. The three main computations proceed simultaneously, i.e., they are pipelined.

The following sections contain detailed descriptions of the parts of the FFT spectrum analyzer. Schematic diagrams of the circuitry are contained in Appendix 2 and are referenced in the following text. The first seven schematics
Figure 3-10a. Block Diagram of FFT Spectrum Analyzer
Figure 3-10b. Block Diagram of FFT Spectrum Analyzer (cont'd)
Figure 3-11. Board Layout and Inter-Board Signal Flow in FFT Spectrum Analyzer
Photograph 3-1. FFT Spectrum Analyzer and Microcomputer (First View)
(showing inter-board cables and front panel)
Photograph 3-2. FFT Spectrum Analyzer and Microcomputer (Second View) (showing back panel and fans)
Photograph 3-3. FFT Spectrum Analyzer and Microcomputer (Third View)  
(showing AD and PS Boards on top, C&M Board in middle, and BUTT Board on bottom)
correspond to the major data flow blocks in Figure 3-10. The remainder show control and address circuitry.

3.4.2. Control and FFT Memories

The C&M Board contains the control circuitry and the three FFT memories. It generates all the control signals and memory addresses, and sends to and receives data from the other boards. Fast bipolar PROMs, delay lines, and small-scale integration ICs (glue logic) are used to generate the control sequences (e.g. Figure A2-10). Addresses are generated with counters and glue logic (e.g. Figure A2-18). All events are synchronized to the main clock, a 32.000 MHz crystal oscillator (e.g. Figure A2-8).

Data flows between the boards via dedicated buses. Bus interface circuitry on the C&M Board allows selectable access to the three FFT memory banks (Figure A2-1). Each bank comprises four 2Kx8, 35 ns, CMOS static RAM chips. In each bank, the real and imaginary parts operate in parallel.

The flow of data is as follows (Figure 3-10): The analog signal is digitized, windowed, and then stored in either FFT Memory Bank 1 or 2. While data is collected in one of these banks, the previous data block in the other bank is FFT'd. The results of each stage of the FFT are stored in and retrieved from the same bank. On the last stage of the FFT, the results are copied into FFT Memory Bank 3. During the next FFT, the power spectrum of the data in Bank 3 is computed and added to the spectrum in the Power Spectrum Memory Bank. Once the specified number of FFTs have been averaged, i.e., at the end of the
integration period, the spectrum is copied into the Output Memory Bank from which the microcomputer may retrieve the final averaged power spectrum.

3.4.3. Analog-to-Digital Conversion

The analog signal is sampled at 1 Megasamples/sec. Each sample is quantized to 8-bit 2's complement numbers.

The AD Board is carefully laid out to avoid ground loops [Shei86]. The analog and digital circuitry occupy separate halves of the board. The power and ground planes were milled out such that each is common at only one point. The A-to-D chip (Micropower MP7684KD flash converter) straddles the analog-digital boundary as shown in Figure 3-12.

The analog side consists of gain and offset circuitry followed by a high-speed operational amplifier (Harris HA2541-5) to drive the high capacitance input of the converter (Figure A2-2). The digital side includes a short and long term overflow detector and window circuitry (Figures A2-3 and A2-24). The output of the A-D converter is in offset binary form. It is converted to 2's complement form by inverting the sign bit.

The aperture uncertainty time of the converter is an important specification. It is the maximum variation in the time at which the converter samples the incoming signal and hence limits the bandwidth that can be processed without significant quantization error. For a full-scale 500 kHz sinusoidal input, the maximum rate of change in levels/sec is:
Figure 3-12. Layout of A-D Chip on AD Board
The MP7684KD has an aperture uncertainty of 60 psec representing an error of:

\[(8.04 \times 10^8 \text{ levels/sec})(60 \text{ psec}) = 0.05 \text{ levels.}\]

This is much less than its \(\pm \frac{1}{2}\) LSB linearity error.

### 3.4.3.1. Anti-Aliasing Filters

A lowpass filter is needed to bandlimit the signal to prevent aliasing. Towards achieving maximum bandwidth, the sampling rate used (1 Megasamples/sec) is exactly twice the maximum frequency 500 kHz, i.e., the signal is not "over-sampled". It is therefore important to use an anti-aliasing filter that has a rapid transition from passband to stopband. The Cauer-Chebyshev filter has the fastest roll-off for specified ripples in the passband and stopband. Tables of normalized L and C component values are given in [Zver67]. A seven-pole lowpass Cauer-Chebyshev filter was constructed using silver-mica capacitors and specially-wound high-Q inductors. (High-Q inductors are required to achieve the rapid rolloff of the transition region.) The filter was designed to have a transition region of 433 kHz to 500 kHz, a pass-band ripple of 1.35 dB, and a maximum stop-band response of \(-57.84\) dB. A schematic and the measured and theoretical frequency responses of the filter are shown in Figures 3-13 and 3-14. The theoretical response was calculated with the SPICE program [SPIC81]. The measured response is \(-56\) dB (below in-band signals) at 500 kHz, the stopband attenuation is \(\geq 52\) dB, and the passband ripple is 1.3 dB. The ferrite cores of the inductors have a limited frequency range. This is
probably the reason the resonance at 900 kHz is not present.

The analyzer can also be programmed to operate at a sampling rate of 62.5 ksamples/sec (ZOOM mode). The ZOOM mode requires a separate anti-aliasing filter. If the output passband of the imageless mixer went down to DC, a lowpass filter with cutoff at \( (500 \text{ kHz})/16 = 31.25 \text{ kHz} \) would suffice. However, the imageless mixer passband begins at 20 kHz. To overcome this problem, we can instead build a bandpass filter centered at \( 2(31.25 \text{ kHz}) = 62.5 \text{ kHz} \) with a bandwidth of 31.25 kHz and invoke the sub-sampling theorem, i.e., sample the band-limited signal at 62.5 kilosamples/sec.

An active (operational amplifier) design would probably be most suitable. The ZOOM anti-aliasing filter has not yet been included in the interference monitoring system.

3.4.4. Windowing

After analog-to-digital conversion, each datum is multiplied by an 8-bit unsigned window coefficient. A 256-point Kaiser-Bessel window stored in PROM is addressed sequentially by a counter. Its first sidelobe level is down \(-46 \text{ dB}\) from the main lobe. The full 16-bit 2's complement product is retained and buffered onto the AD–C&M data bus (Figure A2-3).

The window values were calculated on a mainframe computer in double-precision FORTRAN, quantized, and written to floppy disk (for subsequent PROM programming).
Figure 3-13. 500 kHz Cauer-Chebyshev Lowpass Anti-Aliasing Filter
Figure 3-14. Frequency Response of 500 kHz Anti-Aliasing Filter
A second 256x8 PROM containing a rectangular window (every bit equal
to logical 1) was used for testing purposes.

3.4.5. Fast Fourier Transformation

The fast Fourier transform is computed using a constant-geometry,
decimation-in-frequency algorithm. Compared to the in-place algorithm,
constant-geometry requires 512 complex words instead of 256 per bank. However,
since the RAMs come in 2Kx8 packages, the extra 256 words are "free".

The heart of the FFT process is the BUTT Board. It performs one DIF
butterfly in 500 nsec using 16-bit, 2's complement arithmetic. One FFT takes
1024 butts x 500 nsec/butt = 512 \(\mu\)sec. The butterfly is the rate-determining process
in the analyzer. Via real and imaginary bi-directional buses (operating
simultaneously), the board receives two complex words, A and B (each has 16-bit
real, 16-bit imaginary parts), from the C&M Board, computes the butterfly (with
partial overlap), and sends the results, X and Y, back. Each bus works at an
8 MHz word rate (16-bits per word).

In each stage of the FFT, the data are pre-scaled down by two to
prevent word growth and subsequent overflow. In the divide-by-2 operation, the
least-significant bit of the result is the logical OR of the two least-significant bits
of the input. This causes numbers to be rounded up or down pseudo-randomly
and thus does not introduce the DC bias created by conventional "round up
only" or "round down only" schemes.
The butterfly computation is hard-wired with four 16-bit, 85 nsec, CMOS parallel multipliers plus 74F' series, TTL bit-slice adders and subtractors (Figures A2-4 and A2-5). Latches and tri-state latches are used to hold data and buffer data onto the buses. The data lines are terminated at both the C&M Board and the BUTT Board to reduce ringing.

The unit-circle coefficients $C_j$ are stored in a 128x32 PROM bank as $-\cos$ and $-\sin$. The negative of $\cos$ is stored because the number $-1.000$ can be represented exactly while $+1.000$ cannot. As seen in Table 3-2, $+1.000$ is the only unity-valued cosine coefficient occurring in the DIF FFT. Thus, storing $-\cos$ reduces the computational noise. Similarly, $-j$ occurs while $+j$ does not, so $-\sin$ is stored.

The sin/cos values were calculated on a mainframe computer in double-precision FORTRAN, quantized, and written to floppy disk (for subsequent PROM programming).

### 3.4.5.1. FFT Data and Coefficient Addressing

Data address generation for constant-geometry is very simple. Table 3-3 lists the data address sequences for a 16-point transform. The read addresses are generated by rotating the output of a counter by one bit (Figure A2-16). The write addresses are the outputs of a counter.

Table 3-4 lists the coefficient addresses for a 16-point transform. They are generated by masking the lower $s$ bits of a counter during the $s^{th}$ stage.
Table 3-2. Number of Occurrences of Unity Values of Unit-Circle Coefficients in 256-Point FFT

Table 3-3. Read and Write Address Sequences for 16-Point Constant-Geometry DIF FFT
During the first stage, the data is read from the top half of memory and written to the bottom half. In subsequent stages, the direction of data flip-flops back and forth. The swap is implemented by toggling the most significant bit (A₈) of the address (Figure A2-17).

During the final stage, the data is copied to FFT Memory Bank 3. Hardwired bit-reversal is applied at this point (Figure A2-20).

3.4.6. Power Spectrum Accumulation

The 128-point power spectrum of the complex FFT data is computed by squaring and adding X_R(k) and X_I(k) values. The elements are addressed such that the power spectrum is "unscrambled" as described in section 3.1.3. (Recall that the computation actually yields the sum of two power spectra.)

The circuitry centers around a 16-bit, 95 nsec, CMOS parallel multiplier for squaring and a 32-bit, 74F' series, TTL bit-slice adder for accumulation (Figure A2-6). The data rate is such that only one multiplier is required to do the four squares per bin. The spectrum can accumulate up to 32 bits in a 128x32 NMOS memory bank.† Further averaging can be done by the microcomputer.

The power spectrum computations for a particular FFT take place during

†The data rate is too great for the microcomputer to do all the accumulation.
### Table 3-4. Coefficients Address Sequence for 16-Point Constant-Geometry DIF FFT

<table>
<thead>
<tr>
<th>Stage</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>000</td>
<td>000</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td></td>
<td>001</td>
<td>000</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td></td>
<td>010</td>
<td>010</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td></td>
<td>011</td>
<td>010</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>000</td>
</tr>
<tr>
<td></td>
<td>101</td>
<td>100</td>
<td>100</td>
<td>000</td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>110</td>
<td>100</td>
<td>000</td>
</tr>
<tr>
<td></td>
<td>111</td>
<td>110</td>
<td>100</td>
<td>000</td>
</tr>
</tbody>
</table>

(binary addresses)

### Table 3-5. Read and Write Address Sequences for Unscrambling Power Spectrum (N=16)

<table>
<thead>
<tr>
<th>Read addresses (binary)</th>
<th>Write addresses (binary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000, 0000</td>
<td>0000</td>
</tr>
<tr>
<td>0001, 1111</td>
<td>0001</td>
</tr>
<tr>
<td>0010, 1110</td>
<td>0010</td>
</tr>
<tr>
<td>0011, 1101</td>
<td>0011</td>
</tr>
<tr>
<td>0100, 1100</td>
<td>0100</td>
</tr>
<tr>
<td>0101, 1011</td>
<td>0101</td>
</tr>
<tr>
<td>0110, 1010</td>
<td>0110</td>
</tr>
<tr>
<td>0111, 1001</td>
<td>0111</td>
</tr>
</tbody>
</table>
the first seven stages of the next FFT. (Recall that, during the final stage, the new complex spectrum is being loaded into FFT Memory Bank 3.) The power spectrum clock is inhibited during the final stage by an end-of-FFT pulse (Figure A2-9).

3.4.6.1. Power Spectrum Addressing

The read and write address sequences for calculating an unscrambled power spectrum for a 16-point transform are listed in Table 3-5. The first read address and the write address are the output of a counter. The second read address is generated by a PROM addressed by the counter (Figure A2-19).

3.4.7. Interface to Microcomputer

The FFT spectrum analyzer is connected to the microcomputer via ports on the microcomputer (Figures A2-7, A2-8, A2-22, and A2-23). The interface signals are summarized as follows.

1. NAVG0-7 = Number of averages per integration period. The microcomputer can set NAVG0-7 at any time. The range of integration times is 2–32 sec (normal mode) and .125–2 sec (ZOOM mode).

2. MODE0-1 = Clock mode. The microcomputer can set MODE0-1 to "external", "single step", "normal", or "ZOOM".

3. EXTCLK = External clock signal from microcomputer.

4. SYSRESET = System reset from microcomputer. This signal resets the analyzer by zeroing the accumulator and restarting the processing.

5. OUTPUT0-7 = Output data byte from analyzer. The 128-point spectrum (32-bit values) is uploaded to the microcomputer in bytes.

6. DATAREADY = Data ready signal from analyzer. The microcomputer polls this line. It indicates when data is ready for upload.

7. EMUX, A0-1 = Enable output bus and byte address. During an upload,
the microcomputer manipulates these signals.

8. 1ARDY, 2ARDY, 2BRDY = Microcomputer port ready lines.

During the final accumulation, the averaged power spectrum is copied into a 128x32 NMOS memory bank that serves as an output buffer (Figure A2-7). The microcomputer can then read the data any time before the end of the next integration period. One upload (128 32-bit words) takes 92 ms.

3.4.8. Considerations in the Circuit Design

The overall design goals included high speed, reasonable power consumption, moderate cost, modularity, reliability, and serviceability. In this section, the attainment of these goals will be elaborated upon.

High speed is desired to achieve wide bandwidth. When digital logic is operated at high speeds, e.g., 10's of MHz and up, three basic problems arise: noise generation, crosstalk, and ringing [DeFa70, Morr71]. To reduce noise, the wire-wrap boards have ground and $V_{cc}$ planes to confine the electric and magnetic fields generated by the fast switching signals on the wires. Also, decoupling capacitors are used to reduce current spikes by providing local, short-term power to chips. In the analyzer, each chip is accompanied by a 0.1 $\mu$F Z5U-grade ceramic decoupling capacitor. The make and value were selected after measuring the impedance of numerous capacitors at various frequencies. As well, several 10–15 $\mu$F tantalum capacitors are placed on each board to handle longer-term power demands.

Crosstalk on adjacent wires and ringing on a single wire increase with
frequency and length of wire. To reduce crosstalk, signal lines can be paired with ground lines to confine their fields. Ringing is a transmission line effect and can be reduced, ideally, by providing a known, controlled characteristic impedance \( Z_0 \) along the signal path and terminating (matching) both ends with resistances equal to \( Z_0 \). A rule of thumb says termination is needed when the path length is long enough such that:

\[
t_d \geq \left( \frac{1}{3} \text{ to } \frac{1}{2} \right) \min(t_r, t_f)
\]

where \( t_d \) = one-way transit time, \( t_r \) = rise-time of logic, and \( t_f \) = fall-time of logic. For the inter-board cables, the propagation speed is 1.7 ns/foot. The 74F' drivers have rise- and fall-times in the range 2—4 ns. The rule of thumb implies that termination is necessary for wire lengths \( \geq 5'' \). A controlled impedance can be effected by running signal wires on top of a ground plane or along side a ground wire in ribbon cable (\( Z_0 \approx 100\Omega \) in either case). Even better results are obtained with twisted-pair wires where one wire is grounded at each end. (The characteristic impedance of 30 AWG twisted-pair wire-wrap wire was measured to be 111\Omega.) However, a single terminating resistor in series/parallel at the source/destination requires more driving voltage/current than is available in currently-available TTL drivers. A compromise solution to the termination problem is to use a split-resistor (two resistors) combination at the destination whose Thévenin resistance is approximately \( Z_0 \).

All three ground schemes are used in the analyzer. Wires on a board are run as close to the ground plane as possible. Sensitive signals, e.g., edge-triggered inputs, that run more than a few inches are twisted around
ground wires (grounded at each end). All inter-board signals are paired with
ground wires (grounded at each end) in ribbon cable. As well, all inter-board
signals are destination terminated with split-resistor pairs to reduce ringing. (The
bidirectional data buses are terminated at each end.) There are a few different
sets of values of terminators available. The greatest reduction in ringing (with
reasonable current requirements) for 74F' series drivers and loads is achieved
with 180Ω-390Ω terminators. This was determined by using reflection diagrams
[FAST85].

Board layout was an important and time-consuming task. Countless
variations on IC placement were tried with the following considerations in mind:
minimization of the critical wire lengths, the boardspace was at a premium (esp.
the C&M Board), and the interboard ribbon cables must line up. A drawing of
the boards is shown in Figure 3-11. The C&M Board is located in the middle to
minimize the lengths of the numerous interconnecting buses.

Low power consumption is important because the interference monitor will
eventually be powered by a battery-based uninterruptable power supply (section
7.1). Also, the monitor might one day be made portable to facilitate tracking
down of interfering sources. It would then require a battery power supply. In
general, the IC technology with the lowest power consumption was used for a
given speed requirement. The analyzer consists of CMOS multipliers, CMOS and
NMOS static RAMs, a CMOS A-D converter, and bipolar PROMS, as well as
74F', 74LS', and 74' series TTL chips. The analyzer consumes 10 A @ 5 V and
40 mA @ 12 V.
Cost was minimized because of budget constraints aggravated by drastic cuts in federal funding to the observatory. The total cost of the parts used in the project was $4,000. The spectrum analyzer parts cost $1,500. As shown in Table 3-1, the spectrum analyzer built for this project achieves a high bandwidth at a fraction of the cost (albeit parts only) of other FFT systems.

The analyzer was made modular to facilitate testing and servicing. It is divided into four boards: analog-to-digital conversion (plus windowing) board C&M Board. On each board, special microcomputer ports were wired for testing sub-sections of the analyzer. Sub-sections can be tested separately by running test routines on the microcomputer after rearranging the ribbon cables. In a typical test routine, the microcomputer provides test data, addresses, and control signals to the boards and retrieves and checks the results.

Since the analyzer is part of a working, useful system, it is important that it be reliable and serviceable. It is possible that some chips will eventually fail. By using the test software, a fault can be traced to an individual board. Also, the timing is designed to accommodate worst-case chip delays over the entire 0 to 70 °Celsius temperature range. Thus, if a chip needs replacing, any off-the-shelf replacement will suffice (no screening of chips is necessary). The design is specified over a wide temperature range so that the interference monitor may be operated outdoors. (This is a desirable option that would allow better localization of interfering sources.)

Special attention is given to bus conflict, i.e., the situation when opposing
tri-state buffers are both enabled. Unlike many commercial digital designs, in which 1–100 ns-wide glitches occur due to bus conflicts during context switching, there is no bus conflict at all in the analyzer. There are two reasons for doing this. Short-term bus conflict, although not catastrophic: a) degrades the life of the driver chips, and b) generates noise spikes on the power lines. In the analyzer, all potentially conflicting enable signals are strictly† non-overlapping, at the expense of slightly reduced throughput. To ensure non-overlap upon power-up, NAND-gate configurations are added to the enable control lines, e.g. as in Figure A2-10.

3.5. ERRORS DUE TO FIXED-POINT NUMBER REPRESENTATION AND COMPUTATION IN THE FFT SPECTRUM ANALYZER

In order to maximize speed, fixed-point arithmetic is implemented in the FFT spectrum analyzer. Since we want to process a wide dynamic range of signals, it is important to understand the effects of fixed-point number representation and computation in the analyzer. Representing continuous variables as digital words introduces quantization noise. Computing with digital words produces roundoff noise.

In the following sections, the sources of fixed-point error in the analyzer are pinpointed, characterized, and, where possible, quantified. For comparative purposes, all noise levels are referenced to the output power (in a single frequency bin) of a full-scale, bin-center input sinusoid. Full-scale is normalized to ±1.0 Volt. Ideally, any noise introduced by the analyzer is much less than the

†A safety cycle is inserted between enable transitions.
input noise from the receiver.

To establish a point of reference, we calculate the power in one bin of a full-scale, bin-center sinusoid:

\[
\frac{A^2}{2} \left( \frac{1}{N} \right)^2 N \frac{1}{2} (CG)^2 \right]_{A=1V} = \frac{SQSUM}{4N^3}
\]

where the terms correspond to, respectively, power in a sine wave of amplitude \(A\), divide-by-2 scaling at each stage (section 3.5.3), Parseval energy growth, bin-distribution, and window tapering (section 3.1.3) (The terms CG and SQSUM were defined in section 3.1.1.)

After applying the appropriate scaling, the receiver noise at the input of the analyzer (referred to an output bin) works out to be \(-32\) dB (measured).

3.5.1. A-D Conversion Noise

Assuming the A-D converter has no non-linearities (i.e. it is a perfect sampler and quantizer) and the quantization error is uniformly distributed, then the only noise produced by a \(B\)-bit conversion is white noise with power:

\[
\frac{(\Delta V)^2}{12}
\]

where \(\Delta V = \frac{\text{full-scale voltage}}{2^B} = \frac{2.0}{2^B} = 2^{-(B-1)}\). Thus, the referenced noise level (after scaling the noise and referring it to a full-scale sinusoid) is:

\[
\text{A-D noise level} = \frac{4 \cdot 2^{-2B}}{3N} \text{ENBW}
\]

(ENBW was defined in section 3.1.1.) For \(N = 256\), \(B = 8\), and the Kaiser-Bessel
window, the level is \(-69 \text{ dB}\).

The dynamic range of the A-D converter in terms of number representation is:

\[
A-D \text{ dynamic range} = 20 \log_2 B \text{ (dB)}.
\]

For \(B = 8\), this is \(48 \text{ dB}\). For us, the maximum SNR (section 2.3.1) is about \(9 \text{ dB}\). So, if the noise spans a moderate number of quantization levels, the dynamic range of an 8-bit converter is more than adequate. Note that the overall dynamic range \(DR\) (39 dB) does not have to be less than the A-D dynamic range because small signals "ride" on top of the noise.

3.5.2. Windowing Noise

Since the A-D converter produces 8-bit words, it is convenient to quantize the window to 8 bits and perform the window multiplication with an 8x8 parallel multiplier. Because the window coefficients are all greater than zero, we can represent them as unsigned integers. This increases the accuracy by a factor of two.

Intuitively, we might expect 8-bit windowing to produce about the same level of quantization noise as 8-bit A-D conversion. To determine the windowing quantization noise, the 256-point quantized time-domain window was padded with 768 zeros, FFT'd (1024-point transform), then magnitude-squared. The zero-padding interpolates the spectrum to afford a more detailed view. A plot of the result is shown in Figure 3-15. We see that it differs from the unquantized
window (Figure 3-3) by having somewhat higher sidelobes. A rough calculation shows that the noise added to the distant sidelobes is $\approx -80 \, \text{dB}$. This is quite satisfactory.

![Figure 3.15. Kaiser-Bessel \((a = 2)\) Window (8-Bit Quantization)](image)

3.5.3. Overflow in the FFT

Parseval's theorem indicates that the DFT process introduces an N-fold inflation in the power spectral values. The word growth is monotonic from stage to stage. For both the DIT and DIF butterflies, it is easy to show that:

$$|X|^2 + |Y|^2 = 2|A|^2 + |B|^2$$

where \((A,B)\) and \((X,Y)\) are the butterfly input and output pairs, respectively (each of \(A, B, X, \text{and } Y\) is a complex number). That is, the mean-square value
of the data increases by a factor of two every stage. The magnitudes of the butterfly outputs are bounded as follows [Pele76]:

\[ \max(|A|,|B|) \leq \max(|X|,|Y|) \leq 2\max(|A|,|B|). \] (3-4)

That is, the largest magnitude in the butterfly output is non-decreasing.

If unchecked, this word growth can cause overflow in the fixed-point butterfly computation. Down-scaling of the data can prevent overflow. Oppenheim and Weinstein [Oppe72] show that overflow can be prevented by scaling the input data such that:

\[ |x(n)| < \frac{1}{N} \quad ; \quad n=0,1,...,N-1. \]

The upper bound in Equation 3-2 implies that the scaling can be divided into \( M = \log_2 N \) scalings, i.e., one divide-by-2 at each stage such that:

\[ |A| < \frac{1}{2} \]
\[ |B| < \frac{1}{2}. \]

This distribution of scaling does not change the overall scaling of the input data. However, it reduces the output noise due to the FFT because errors in the early stages are attenuated by scaling in the later stages.

Although the above scaling procedure ensures that overflow can never happen, it is impractical because the complex data is scaled. It is more practical to scale the real and imaginary parts such that:
\[ |A_R| < \frac{1}{2} \]
\[ |A_I| < \frac{1}{2} \]
\[ |B_R| < \frac{1}{2} \]
\[ |B_I| < \frac{1}{2} \]

Unfortunately, this does not eliminate the possibility of overflow. In Figure 3-16, the upper bounds on the butterfly outputs are derived for the DIT and DIF algorithms (result for DIF not in other literature). We see that overflow is most probable when the unit-circle angle \( \theta \mod 2\pi = \frac{\pi}{4} \) or \( 3\frac{\pi}{4} \). This condition is not possible in all stages. For example, the constant-geometry flow diagram (Figure 3-7) indicates that in the last stage it does not occur (all \( \theta = 0 \)). In fact, overflow cannot occur in the last stage.

The number of occurrences of \( \theta \mod 2\pi = \frac{\pi}{4} \) or \( 3\frac{\pi}{4} \) in each stage are listed in Table 3-6.

Using the simulation (Appendix 1), the number of overflows were counted for various levels of input noise. For a level similar to that in the actual system, no overflows were observed during a run of one million averages.

Besides scaling, the other major reason for the infrequent occurrence of overflow is the tapered window coupled with the nature of the addressing in the FFT algorithm. Referring to Figure 3-6 or 3-7, we observe that for each butterfly in the first stage, the data for the A and B inputs are multiplied by different window weights, in particular, one small and one large weight. This reduces the
1. DIF

\[ X = A + B \]
\[ Y = (A - B)C \]

consider real parts of \( X \) and \( Y \):

\[ X_R = A_R + B_R \]
\[ Y_R = (A_R - B_R) \cos \theta - (A_I - B_I) \sin \theta \]

Now, if \( |A_R| < \frac{1}{2} \), \( |A_I| < \frac{1}{2} \), \( |B_R| < \frac{1}{2} \), \( |B_I| < \frac{1}{2} \)

Then

\[ |X_R| = |A_R + B_R| \]
\[ |Y_R| = |(A_R - B_R) \cos \theta - (A_I - B_I) \sin \theta| \cdot \]

\[ |X_R| < |A_R| + |B_R| < \frac{1}{2} \quad \text{(triangle inequality)} \]
\[ |Y_R| < |\cos \theta \pm \sin \theta| < \frac{1}{\sqrt{2}} \quad \text{or} \quad \frac{3\pi}{4} \]

(similar for \( X_I, Y_I \))

Result: \( |X_R| < 1 \), \( |Y_R| < 1.414 \)
\( |X_I| < 1 \), \( |Y_I| < 1.414 \)

2. DIT

\[ X = A + BC \]
\[ Y = A - BC \]

(similar to DIF)

Result: \( |X_R| < 1.207 \), \( |Y_R| < 1.207 \)
\( |X_I| < 1.207 \), \( |Y_I| < 1.207 \)

Figure 3-16. Upper Bounds on Butterfly Outputs
Table 3-6. Number of Occurrences of $\theta \mod 2\pi = \frac{\pi}{4}$ or $\frac{3\pi}{4}$

<table>
<thead>
<tr>
<th>stage</th>
<th>$\pi/4$</th>
<th>$3\pi/4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

probability of two large inputs occurring and hence the chance of overflow.

3.5.4. FFT Noise

There are three sources of noise in a fixed-point FFT: quantization of the unit-circle coefficients, roundoff in the butterfly multiplications, and, if implemented, roundoff in scaling. Each is discussed below.

3.5.4.1. Fixed-Point Representation of Unit-Circle Coefficients

Tufts et. al. have studied the effect of fixed-point representation of the unit-circle coefficients on FFT performance [Tuft72]. They quantized the coefficients and transformed† sets of unit-pulse sequences using floating-point

†They do not say whether DIT or DIF was used. However, it should make little difference in the results.
("exact") arithmetic. The deviation of the spectrum from white gave the noise due to inexact representation of the coefficients. They found that the spectrum contained spurious sidelobes. The largest spurious sidelobe decreases by approximately 6 dB per bit and is independent of N for N≥64. For 16-bit coefficients, the highest spurious response is −99.7 dB.

3.5.4.2. Roundoff Noise in the Butterfly

In the butterfly computation, each multiplication produces noise due to rounding of the product. Rounding is needed to limit the size of the word, e.g., in the analyzer, the 16 most significant bits is retained from the 32-bit product of two 16-bit numbers. Numerous studies in the literature estimate the mean noise level due to roundoff in the FFT [Oppe72,Sund77,Thôñ76]. In almost all cases, the roundoff error probability density is assumed to be uniform and the errors uncorrelated.

Some typical results for FFT roundoff noise are shown below. BFFT is the number of bits in the FFT arithmetic. The values were calculated from the indicated references and scaled to suit our reference level.

<table>
<thead>
<tr>
<th>BFFT</th>
<th>[Thôñ76]</th>
<th>[Sund77]</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>−28.8 dB</td>
<td>−27.8 dB</td>
</tr>
<tr>
<td>10</td>
<td>−40.8</td>
<td>−39.8</td>
</tr>
<tr>
<td>12</td>
<td>−52.9</td>
<td>−51.9</td>
</tr>
<tr>
<td>14</td>
<td>−64.9</td>
<td>−63.9</td>
</tr>
<tr>
<td>16</td>
<td>−77.0</td>
<td>−76.0</td>
</tr>
</tbody>
</table>
(Butterfly roundoff noise is mentioned again in section 3.6).

3.5.4.3. Scaling Noise

To combat overflow in the analyzer, divide-by-2 scaling is used at the beginning of each butterfly. Each of the real and imaginary parts of each complex input datum is scaled. In the divide-by-2 operation (Figure A2-5), the 16-bit input word is shifted right by one bit and the sign bit is copied into the most-significant location. The least-significant bit of the result is produced by logically ORing the two least-significant bits of the input [Abru85]. This "pseudo-random rounding" has the advantage over conventional "round up only" or "round down only" truncation (hard scaling) schemes that, at the expense of slightly higher noise variance, it does not introduce a DC bias. A comparison of two schemes is illustrated in Figure 3-17.

During the simulation, it was observed that truncation added a lowpass shape, i.e., colored noise, to the spectrum. Intuitively, this can be understood by considering the DC bias introduced by the truncation at each stage of the FFT. The scaling in the first stage adds noise to the DC frequency bin. Subsequent scalings affect higher bins, up to the last stage where noise is added to all bins.

3.5.5. Power Spectrum Noise

Each power spectrum word is the sum of four squared numbers (Figure 3-10). Intuitively, we can expect the mean roundoff noise to be approximately the same as the roundoff noise generated by one stage of the FFT (since each butterfly contains four multiplications). A theoretical value of
\[ \varepsilon = \text{error, } p = \text{normalized probability} \]

\[ B = \text{number of bits (including sign)} \]

input word = \( b_{15} \cdot b_{14} \ldots b_2 \cdot b_1 \cdot b_0 \)

1. **HARD SCALING (TRUNCATION)**

\[
\begin{array}{c|c|c|c|c}
\varepsilon & b_1 & b_{15} & b_{30} = 1 & \text{new } b_1 \\
\hline
0 & 1 & 1 & -1 & 1 \times 2^{15} \\
1 & 0 & 1 & -1 & 0 \times 2^{15} \\
1 & 1 & 1 & -1 & 1 \times 2^{15} \\
1 & 1 & 0 & -1 & 0 \times 2^{15} \\
\end{array}
\]

mean \( \mu = \left( \frac{\frac{1}{2} \left(-2^{15} + 4(0)\right)}{8} \right) = -\frac{1}{2} \times 2^{15} \)

variance \( \sigma^2 = \frac{4}{8} \left(-2^{15} - \mu \right)^2 + \frac{4}{8} (0 - \mu)^2 + \frac{4}{8} (2^{15} - \mu)^2 = \frac{1}{2} \times 2^{28} \)

for complex numbers: \( \mu = -\frac{1}{2} \times 2^{15}, \sigma^2 = \frac{1}{2} \times 2^{28} \)

2. **PSEUDO-RANDOM Rounding**

\[
\begin{array}{c|c|c|c|c}
\varepsilon & b_1 & b_{15} & b_{30} = 1 & \text{new } b_1 \\
\hline
0 & 1 & 1 & -1 & 1 \times 2^{15} \\
1 & 0 & 1 & -1 & 0 \times 2^{15} \\
1 & 1 & 1 & -1 & 1 \times 2^{15} \\
1 & 1 & 0 & -1 & 0 \times 2^{15} \\
\end{array}
\]

mean \( \mu = \left\{ \frac{1}{2} \left(2^{15} + 4(0) + 2^{15} \right) \right\} / 8 = 0 \)

variance \( \sigma^2 = \frac{2}{8} \left(-2^{15} - \mu \right)^2 + \frac{8}{8} (0 - \mu)^2 + \frac{2}{8} (2^{15} - \mu)^2 = \frac{1}{2} \times 2^{28} \)

for complex numbers: \( \mu = 0, \sigma^2 = 2^{28} \)

Figure 3-17. Comparison of Two Divide-By-2 Scaling Schemes
power spectrum roundoff noise does not exist in the literature and no attempt
was made to derive one.

However, a more important problem arising from the computation of the
power spectrum was observed. Using the result for a square-law detector with
Gaussian input [Papo84], the approximate distribution of a bin in the power
spectrum is $\chi^2$. The standard deviation approximately equals the actual level
(Equation 3-2 with $K = 1$). Unfortunately, the distribution of the unsigned numbers
cannot fall below 0. It is possible for a small, non-zero number to be squared
and rounded to 0. Thus, there is a non-linear effect (clipping) in the power
spectrum computation.† If the input level to the analyzer is too small, then
zeros will occur. For perfectly white noise, this is not important because each
bin will have, on average, the same number of zero contributions. However, for
colored noise, the valleys in the spectrum will be de-accentuated. This distortion
of the spectrum can limit the detectability of small signals.

In Chapter 6, a suitable input level is found by experiment so that the
distortion is negligible. Note that the effect does not significantly "amplify" the
relatively small fixed-point noise contributions from earlier processing, although
they are still the fundamental limit on the detectability of small signals. (This
was confirmed by running the simulation with and without the up-scaling of the
FFT data described in section 6.2.)

†This effect is much less prevalent in the FFT, especially the DIF FFT, because
the multiplications involve numbers from the unit-circle (real and imaginary),
which are unity or exactly zero in the last two stages.
3.6. COMPUTER SIMULATION OF THE FFT SPECTRUM ANALYZER

It is extremely difficult to mathematically calculate the exact noise spectrum of fixed-point errors in the analyzer. Even with only white noise input, the error probability densities are not uniform and errors are not totally uncorrelated as assumed in nearly all theoretical calculations. Indeed, when narrowband signals are present, the probability densities are further distorted. This is particularly true for butterfly roundoff noise. The only paper that calculates the butterfly roundoff noise in individual frequency bins is that by Thong and Liu [Thon76]. However, they make numerous assumptions: the unit-circle coefficients are represented exactly, the multiplication roundoff error is uniformly distributed, and multiplications by +1, −1, +j, and −j are noiseless. Also, only the FFT is analyzed, i.e., A-D conversion, windowing, and power spectrum computation are not considered.

If the noise is perfectly white, then, theoretically, an arbitrarily small signal can be detected by performing enough averages. Unfortunately, the noise spectrum is not white. We want to detect very small signals (sometimes in the presence of large signals) by integrating for long periods of time, so we must know how the fixed-point error spectrum behaves. In particular, we need to know how much the error spectrum deviates from white. Signals smaller than the size of the deviations will not be detectable, i.e., further averaging of the spectra will not reduce the variance. This is one of three reasons a FORTRAN program (Appendix 1) was written to simulate the computations in the FFT spectrum analyzer. The simulation serves the following purposes: to

1. determine the behavior of the error spectrum with white noise and
sinusoids at input;

2. help estimate the word lengths required in the various processes, thereby minimizing hardware; and

3. verify the results of low-speed analyzer tests.

The simulation was not a trivial program to write. Great care was taken to ensure that it mimics the hardware computations exactly and scales the results properly. The program computes and averages successive, non-overlapping power spectra of an input signal equal to the sum of two sinusoids plus Gaussian noise. Both floating-point and fixed-point processing are done in parallel for comparison. The word lengths in the fixed-point process are user-selectable. Because the simulation runs can be quite lengthy (e.g. up to two weeks of processing time on a VAX mainframe computer), the program is restartable.

The processing sequence for one loop is outlined below, including comments specific to the software.

1. Generate Gaussian noise vector (each value is the sum of six uniformly-distributed numbers) and add to signal(s). Check the random number generator seed each time for repetition.

2. Quantize the data using an ideal A-D converter. Count the number of clipped values.

3. Window the time samples. Use one of rectangular or Kaiser-Bessel \((a = 2)\) windows.

4. Pack two data blocks into a complex vector. If necessary, align word sizes to match FFT word sizes.

5. If desired, pre-scale data down by two.

6. FFT the data. The floating-point transform is DIT, in-place. The fixed-point transform is DIF, constant-geometry with pseudo-randomly rounded scaling. For every fixed-point addition or subtraction, count the number of overflows and underflows.
7. Bit-reverse the data.

8. Compute the sum of the two power spectra by unscrambling the transform.

9. Add the power spectrum to the running average.

10. After a specified number of averages, calculate various signal and noise statistics. The signal bins are removed from the noise calculations.

For more details of the simulation program, refer to the program listing in Appendix 1.

The simulation was run for different combinations of word lengths and input signals. While the results in section 3.5 were useful in determining "ballpark" values for the various word lengths in the analyzer, the final choice was based on the simulation results. Of particular importance is the word length in the FFT (BFFT). A plot of the reduction in standard deviation versus the number of averages for various BFFT is shown in Figure 3-18. For this result, the input signal was noise (of a level similar to that in the final system) plus one small sinusoid, and the other word lengths are as implemented. In Chapter 6, the integration time required to detect the minimum detectable signal is shown to be about 1.2 hours. In the plot, we see that for the fixed-point error not to limit the detectability, BFFT should be 15 or more.

[It should be noted that an attempt was made to normalize (whiten) the spectrum by dividing by the error spectrum predicted by Thong and Liu. This was not successful because the actual error spectrum changed (in an unpredictable manner) with the different input signals.]
Decrease in Standard Deviation Versus Number of Averages

Figure 3-18. Simulated Performance of Analyzer for Various BFFT
4. THE ANTENNA SYSTEM

The antennas comprise the first element in the block diagram of the monitor (Figure 1-5). A picture of the antenna assembly is shown in Photograph 4-1. There are nine antennas in the interference monitor: a \( \lambda/4 \) "stub" (monopole over ground plane) and eight helices. The helices are configured in an octagon, each helix pointing horizontally outward. The stub is mounted on top of the helix frame. The antennas and an electronics box are mounted on a 35' pole that is 500' due west of the main building and 400' due south of the SST control building. The location was chosen for its unobstructed view of the observatory buildings and the horizon.

Referring ahead to Figure 5-2 in Chapter 5, an RF switch selects one of the antennas and the RF signal is amplified and sent down the cable to the SST control building. By the use of frequency-multiplexing, the RF signal shares the cable with two other signals: DC power for the amplifier on the pole and an audio frequency (AF) signal that controls the switch.

In this chapter, the antenna system in the interference monitor is described. First, the design considerations for the antennas are discussed. Then, an account of the construction, matching, and testing of the antennas is given. Next, the remote-controlled RF switch and signalling hardware are described. Finally, the "triplexer" filters are described.
Photograph 4-1. Mounting of Antenna Assembly (First View)
(electronics box installed later)
4.1. DESIGN OF THE ANTENNA ARRAY

An attractive approach to RFI monitoring is to use two modes of operation: an "omni-directional surveillance" mode and a "scan and locate" mode. The stub and helix antennas are suitable choices to implement these modes. The stub can monitor all horizontal directions simultaneously. Each helix has a directional beam and an array of them can scan and locate interference by switching from helix to helix.

A stub (gain = 2.2 dBi) was chosen for the omni-directional antenna because it is easy to build. Given more time, a better choice might have been a vertical phased-array of dipoles (e.g. Franklin array [Jasi61]). Such an antenna has greater gain (≈ 6.4 dBi for 3 elements, ≈ 8.7 dBi for 6 elements) and a narrower beam than the stub. In each case, the main beam is toroidal in shape with the maximum response on the horizon.

When designed to operate in end-fire mode, the helix has numerous advantages over other "uni-directional" antennas. It has wide bandwidth, high gain, high directivity, and does not require highly accurate dimensions. It is circularly polarized† and hence capable of receiving signals of arbitrary polarization. Also, there are fairly accurate quasi-empirical formulas describing the beam shape. Some of these are given below [Krau50]:

\[-3\text{dB beamwidth } \theta_{3\text{dB}} \approx (52/C)\sqrt{\lambda^3/nS} \text{ degrees}\]

†This is strictly true on axis. Off axis, the polarization is elliptical.
first nulls beamwidth $\theta_{1}\text{st nulls} \approx 2.21\theta_{3\text{dB}}$ degrees

directivity $D = 15nSC^2$

where

$n =$ number of turns  
$C =$ circumference in meters  
$S =$ turn spacing in meters  
$\lambda =$ wavelength in meters.

The $-3\text{dB}$ beamwidth of a moderately-sized helix is about $45^\circ$. Eight such helices cover $360^\circ$ (Figure 4-1). Rather than simply setting the $-3\text{dB}$ response for each helix at $\theta = 45^\circ/2 = 22.5^\circ$, we should try to maximize the worst-case response so as not to miss any signals. To do so, we shall consider the tradeoff between directivity (maximum gain) $D_0$ and gain $D_C$ at the crossover point $\theta_C = 22.5^\circ$ (Figure 4-2). As the mainlobe is made narrower, $D_0$ increases and $D_C/D_0$ decreases. However, the variation in $D_C$ is not readily apparent. It is best to estimate $D_C$ for various values of $n$ and choose the value that gives the largest value of $D_C$. To do this, we shall use the empirical formulas for $\theta_{3\text{dB}}$ and $\theta_{1}\text{st nulls}$ given above and approximate the response between these angles with a straight line. The latter should be a good approximation because a plot of the purely theoretical response shows that between the two angles the response is quite linear (Figure 4-3).
Figure 4-1. Bird's Eye View of Helices

Figure 4-2. Cross-Section of Mainlobe of Helical Beampattern
Theoretical Response of Helical Antenna for Various Numbers of Turns

\[ E(\theta) = \sin\left(\frac{\pi}{2n}\right) \cos \theta \frac{\sin(n \psi)}{\sin(\psi/2)} \]

where
\[ \psi = 2\pi \left[ \frac{S}{\lambda} \left(1 - \cos \theta\right) + \frac{1}{2n} \right] \]

reference: [Bala B2]

Figure 4-3. Theoretical Response of Helix (in One Dimension)
Setting \( C = \lambda \) and \( a = \tan^{-1}(S/C) = 14^\circ \) (optimum values as given in [Jasi61]) and linearly approximating the response between \( \theta_{3\text{dB}} \) and \( \theta_{1\text{st\ nulls}} \) we compute values of \( D_c \) versus \( n \):

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \theta_{3\text{dB}} )</th>
<th>( \frac{1}{2} \theta_{3\text{dB}} )</th>
<th>( D_o )</th>
<th>( D_c/D_o )</th>
<th>( D_c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>60.1</td>
<td>30.1</td>
<td>10.5dB</td>
<td>-1.37dB</td>
<td>9.13dB</td>
</tr>
<tr>
<td>4</td>
<td>52.1</td>
<td>26.0</td>
<td>11.7</td>
<td>-2.08</td>
<td>9.62</td>
</tr>
<tr>
<td>5</td>
<td>46.6</td>
<td>23.3</td>
<td>12.7</td>
<td>-2.76</td>
<td>9.94</td>
</tr>
<tr>
<td>6</td>
<td>42.5</td>
<td>21.3</td>
<td>13.5</td>
<td>-3.44</td>
<td>10.06</td>
</tr>
<tr>
<td>7</td>
<td>39.4</td>
<td>19.7</td>
<td>14.2</td>
<td>-4.09</td>
<td>10.11</td>
</tr>
<tr>
<td>8</td>
<td>36.8</td>
<td>18.4</td>
<td>14.8</td>
<td>-4.77</td>
<td>10.03</td>
</tr>
<tr>
<td>9</td>
<td>34.7</td>
<td>17.4</td>
<td>15.3</td>
<td>-5.45</td>
<td>9.85</td>
</tr>
<tr>
<td>10</td>
<td>32.9</td>
<td>16.5</td>
<td>15.7</td>
<td>-6.15</td>
<td>9.55</td>
</tr>
</tbody>
</table>

Table 4-1. Crossover Gain versus Number of Turns for Helical Array

We see that for maximum crossover gain, we should choose \( n = 7 \). A helix with 7 turns has a directivity of 14.2 dB and a beamwidth of 39.4°.

4.2. CONSTRUCTION AND MATCHING OF THE ANTENNAS

An octagonal 6'x6' frame of welded angle iron was constructed to accommodate the eight helices and stub antenna. The top and sides are covered in aluminum mesh to form good ground planes for, respectively, the stub and helix antennas. The stub is an 18.4 cm (nominal) length of \( \frac{1}{8} \)" copper-clad steel rod soldered onto an N-type connector and mounted on top of the frame. Each helix consists of 7 turns of \( \frac{1}{2} \)" O.D. copper tubing supported by a plastic pipe and guy rope. The dimensions of an individual helix are as follows: turn-to-turn
spacing = 17.3 cm, circumference = 73.5 cm (1 wavelength at 408 MHz), overall length = 128 cm, and ground plane = 30"x30". The construction work was done by the DRAO machinist, Ed Danallanko. Two photographs of the assembly (while being mounted on the pole) are shown in Photographs 4-1 and 4-2. (The coaxial antenna cables can be seen within the framework in the first photograph; the electronics box was installed later.)

Photograph 4-2. Mounting of Antenna Assembly (Second View)

The terminal impedance of a helix with a conventional feed arrangement
is about 140 $\Omega$ (real). Since the receiving electronics presents a 50 $\Omega$ load, a mismatch loss of 1.1 dB will occur. It is possible to modify the helix conductor near the ground plane to achieve a 50 $\Omega$ (real) terminal impedance at the expense of reduced bandwidth [Krau77]. Kraus did so by bonding a metal strip to the helix conductor near the feed point. As described below, by using a similar technique, close to 50 $\Omega$ impedance was achieved for each of the 8 helices. The bandwidth we require (20 MHz) is small compared to the bandwidth of a conventional helix ($\approx$ 200 MHz) so a reduction can be afforded.

A network analyzer was used to measure the impedance by sourcing RF energy into the antenna (i.e. transmitting) and measuring the reflected signal. An initial experiment was done to determine the approximate geometry of the feed to achieve a match at 408 MHz. A single helix mounted on the frame (with one ground plane in place) was pointed towards the sky and the last turn of the helix was bent towards the ground plane until a rough match was obtained. The impedance changes from inductive to capacitive as the feed is moved closer to the ground plane. When the feed is brought in parallel to the plane about $\frac{1}{4}$" away, a transmission line of impedance close to 50 $\Omega$ is formed. Decreasing the spacing further adds capacitance and the impedance becomes capacitive; increasing the spacing makes the impedance inductive. The impedance is fairly sensitive to the spacing: a change of only $\frac{1}{16}$" produces a noticeable effect. An acrylic standoff was machined to hold the feed in place and the other 7 helices were then replicated and added to the frame along with standoffs, ground mesh, and the stub antenna. Photograph 4-3 shows the preliminary tuning set-up. Photograph 4-4 shows the feed arrangement including the standoff and a $\frac{1}{3}$"
diameter braid used to connect the N-connector to the end of the copper conductor. The braid allows for flexion (e.g. due to wind).

The final matching was done by tilting each helix in turn towards the sky and bending the feed slightly at the standoff. The results are shown in Table 4-2. Included is the return loss at three frequencies.†

During the matching trials, a resonance at 290 MHz was observed. The wavelength at this frequency is 103 cm. There is no apparent similar dimension on an individual helix or the frame and so this response is left unexplained. However, since it is far from the band of interest, it should not cause any problems.

After the helices were matched, the crosstalk (coupling) between pairs was measured. The network analyzer was used to transmit from one helix and measure the received signal at another. The results for a typical trial are shown in Table 4-3. It should be noted that, for this trial, helices 3, 4, and 5 were pointing towards the ground so the coupling measured between them and helix 1 may be inaccurate.

The decoupling at 400 MHz between two parallel helices has been measured as a function of separation in [Jasi61]. In our case, adjacent helices are not parallel but a comparison is still of interest. The feed-to-feed spacing of adjacent helices is 74 cm. For this spacing, the decoupling measured by Jasik

†Return loss is a measure of the reflected power. 0 dB is total reflection; ∞ dB is total transmission.
Photograph 4-3. Preliminary Tuning of Helix

Photograph 4-4. Helix Feed
Table 4-2. Measured Terminal Impedance and Return Loss of Helices

<table>
<thead>
<tr>
<th>Helix</th>
<th>Impedance At 408 MHz</th>
<th>398 MHz</th>
<th>408 MHz</th>
<th>418 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>54 - j4 Ohm</td>
<td>27 dB</td>
<td>25 dB</td>
<td>20 dB</td>
</tr>
<tr>
<td>1</td>
<td>57 - j10</td>
<td>21</td>
<td>20</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>58 + j3</td>
<td>22</td>
<td>21</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>56 + j0</td>
<td>20</td>
<td>24</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>61 - j10</td>
<td>18</td>
<td>17</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>61 + j2</td>
<td>27</td>
<td>.26</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>58 - j5</td>
<td>21</td>
<td>21</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>59 + j0</td>
<td>22</td>
<td>21</td>
<td>19</td>
</tr>
</tbody>
</table>

Table 4-3. Measured Crosstalk Between Pairs of Helices

<table>
<thead>
<tr>
<th>Transmitting Helix</th>
<th>Receiving Helix</th>
<th>Transmission Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>35 dB</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>46</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>42</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>46</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>43</td>
</tr>
<tr>
<td>0</td>
<td>6</td>
<td>49</td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>34</td>
</tr>
</tbody>
</table>
was 42 dB. This is comparable to the 35 dB and 34 dB decouplings for helix pairs (0,1) and (0,7) in Table 4-3. It is speculated that the difference may be due to differences in number of turns and ground plane arrangement (details of which are not given in the reference).

The stub antenna was tuned by connecting it to the network analyzer and trimming the conductor until the return loss was maximum at 408 MHz.

Finally, the feeds were weather-proofed by applying black RTV silicone sealant to the braid connections and painting the standoffs with diluted tar. Then the antenna assembly was hoisted up with the DRAO "cherry-picker" truck and seated upon the 35' wooden pole. The pole is guyed for strength in wind and fenced in for protection from cattle.

4.3. THE REMOTE-CONTROLLED RF SWITCH

In order to switch the output to either the stub or one of the eight helices, a 9-to-1 RF switch is required. Because the switch is before the first LNA, the switch must have a very low insertion loss (IL) to keep the system noise figure as low as possible (section 5.1). Commercial switches were either too expensive, had too much insertion loss, or did not have electronic control. It was decided to build a switch with low IL and electronic control.

A double-sided PC board was designed and fabricated (Photograph 4-5). (In the photograph, the components are on the reverse side.) One side is ground plane and the other contains signal lines. Teledyne 732TN-5 relays were chosen
Photograph 4-5. RF Switch PC Board (Signal Side)
as the switching devices. At 408 MHz they have a specified IL of .3 dB and 35 dB isolation. 50 Ω striplines are used for the RF signal paths. The RF signal lines are connected to the PC board via \( \frac{1}{8} \) 24 AWG wire stubs soldered to a panel of BNC connectors. In addition to the RF lines, +5V and control lines go to each relay. Any of the nine antennas can be switched onto a common signal line. When an antenna is turned off, it is terminated in 51Ω with a chip resistor.

Since the circuit must work at RF, grounding is very important. As much as possible, unused copper area is grounded. Numerous feedthroughs connect the grounds on each side of the board. The BNC-plus-stub signal lines are shielded by specially-made brass screw-on cylinders which make a pressure contact with the PC board ground. These turned out to be a crucial ground connection between the outer case and the circuit board itself. A cylinder can be considered as continuing the outer conductor of the signal coaxial cable. Without them, there was considerable parasitic inductance in each stub.

The center (common) signal conductor has a capacitance to ground that was calculated to be about 17 pF. At 408 MHz this would attenuate the signal significantly due to mismatching. However, since the center piece dimensions are small relative to one wavelength, it can be considered a lumped capacitance and resonated with an inductor added in parallel. The theoretical value of the inductance for resonance at 408 MHz is 9 nH. It was found that a \( \frac{3}{8} \) 24 AWG wire to ground, plus two turns \( \left( \frac{5}{16} \right. \) dia.) of 20 AWG wire to ground for fine tuning, worked best. (The latter was installed after the photograph was taken.)
They were mounted away from each other to eliminate coupling. (The theoretical inductance of these wires in parallel is $7 \text{nH} || 37 \text{nH} = 6 \text{nH}$.)

The insertion loss was measured to be .8 dB. Part of this amount is due to the IL of the relay. The remainder is probably due to loss in the conductor traces.

The isolation between lines was measured to be ≥ 30 dB. Such a value is satisfactory because the sidelobe levels in the beam pattern of a helix are down to about −15 dB. An ambiguity in the signal direction can be resolved by simply switching helices and taking the maximum response.

An FSK transmitter/receiver pair plus a digital interface were built to control the RF switch. The schematic diagrams are shown in Figures 4-4 and 4-5. In the SST control building, the microcomputer toggles a single port line which goes to the transmitter. Nine relay control bits and one parity bit are sent serially using a pulse-width modulation scheme. The transmitter converts the digital data to a phase-continuous audio frequency (AF) signal centered at 20 kHz. A simple common-collector (emitter follower) amplifier drives the 50Ω line. At the antenna pole, the receiver demodulates the signal and shifts the data into a register. If the parity of the control word does not match the received parity the FSK receiver turns off the first LNA. This causes a large reduction in the spectrum analyzer output which can be easily detected so that the control data can be sent again. It was observed that errors do occur, but they are quite rare.
Figure 4-4. FSK Transmitter (for Antenna Switch Control)
Figure 4-5. FSK Receiver (for Antenna Switch Control)
Photographs 4-6, 4-7, and 4-8 show some typical timing diagrams of the digital data. In Photograph 4-6, the SW helix is being switched on. Photograph 4-7 is a close up of logic "1" and logic "0" pulses. Photograph 4-8 shows a single "1". The antennas can be switched on in sequence by transmitting single "1" pulses. This extends the life of the switches (a full 10-bit sequence causes all relays to switch momentarily).

In a similar way to the analyzer functional tests, the transmitter/receiver pair (w/o RF switch) can be tested using the microcomputer. Via ports, the microcomputer sends digital data (with parity) and then checks the received data.

### 4.4. THE TRIPLEXERS

Since 700' of cable is required to connect the antenna pole with the SST control building, it is convenient to run only one cable between them. It was decided to run a single Heliax coaxial cable. Heliax is a solid-shielded coax with almost 100% effective shielding. Excellent shielding is needed to prevent the interference monitor from broadcasting interfering signals from the cable. As well, Heliax has extremely low loss (1.8 dB/100' at 400 MHz).

Two identical triplexers (four-port filters) were built to combine/decombine the following signals onto the cable: the RF signal from the antenna, the AF signal from the FSK transmitter, and the DC power for the electronics on the pole. The schematic diagram is shown in Figure 4-6. In each triplexer, the three signals are separated from each other by a combination of highpass, bandpass, and lowpass filters.
data sent
data rec'd
latch pulses

Photograph 4-6. Timing Diagram of SW Helix Being Turned On

ditto
close up of "1" and "0"

Photograph 4-7. Timing Diagram of Transmission of Logic '1' and Logic '0'

ditto

single "1" pulse

Photograph 4-8. Timing Diagram of Transmission of One Logic '1'
Figure 4-6. Triplexers
5. THE RECEIVER

The receiver is the second component in the block diagram of the monitor (Figure 1-5). In order to analyze the 20 MHz-wide band 398–418 MHz with the 500 kHz baseband spectrum analyzer, a sensitive receiver "front-end" is required. The chief functions of the receiver are to amplify the RF signal from the antenna and to select (convert to baseband) a 500 kHz sub-band to be analyzed. The receiver is divided into three parts, each in its own cabinet: a low-noise amplifier located on the antenna pole, a first mixer stage located in the unshielded room in the SST control building, and second mixer stage located in the shielded room in the SST control building. Diagrams of the receiver are given later in Figures 5-2, 5-3, and 5-4. The receiver is constructed from commercially-available components with the exceptions of the IF amplifiers (earlier DRAO projects), imageless mixer (modification by the author of an earlier DRAO design), 100 MHz lowpass filter (made by the author), and two of the baseband amplifiers (modification by the author of an earlier DRAO design).

The next section discusses the significance of noise in the receiver. The remaining sections describe the receiver components.

5.1. RECEIVER NOISE

Since the signal levels we wish to detect can be very small, it is important that the receiver introduce as little noise as possible, i.e., it should have a small noise figure (NF). The noise figure of a system is defined as [Pett84]:

134
The NF of an ideal (noiseless) system is 1.0 (0 dB). Noise figures of practical RF systems range from 1.1 (0.4 dB) and up.

In a system that consists of $n$ cascaded, matched elements with individual gains $G_i$ and noise figures $F_i$ (Figure 5-1), the overall system gain is:

$$G_{sys} = G_1 G_2 G_3 \ldots G_n$$

and the overall system noise figure is:

$$F_{sys} = F_1 + \frac{(F_2-1)}{G_1} + \frac{(F_3-1)}{G_1 G_2} + \frac{(F_n-1)}{G_1 G_2 \ldots G_{n-1}}.$$
SNR of a signal entering the spectrum analyzer. This in turn determines the integration time needed to detect the signal.

There are two major sources of noise that contribute to the overall noise level at the output of the receiver: thermal radiation from the ground that enters through the antenna and noise generated by the receiver electronics. These contributions were found by measurement to be roughly equal.

5.2. THE FIRST MIXER STAGE

The receiver has two mixer stages. Two stages were chosen because of component availability and the fact that it is safer (with regard to stability), and easier, to provide amplification at three frequencies rather than two.

The antenna is linked to the SST control building by 700' of FHJ4 Heliax cable (section 4.4). Two Trontech L410A 410 MHz wide-band LNA's boost the signal before and after it is attenuated by the cable (Figures 5-2 and 5-3). The first LNA (located on the antenna pole) has 31.9 dB gain and 2.06 dB noise figure at 408 MHz. Although the first LNA defines the major portion of the system noise figure, the second LNA helps to render insignificant the noise contributions of the remaining components.

Referring to Figure 5-3, a bandpass filter centered at 408 MHz (K&L 5B120-408/20-0/0) is used to suppress the image band 338–358 MHz before down-mixing. It is placed before the second LNA to prevent the latter from saturating due to possible large out-of-band signals. The mixer (HP 10514A) and
Figure 5-2. First Part of Receiver and Other Components (on Antenna Pole)
AF INPUT

+24 VDC

HELIAX CABLE

AF triplexer

RF

CABLE

(G = -.9 dB (RF))

3 dB pad

408 MHz bandpass filter

(BW = 20 MHz)

(G = -1.9 dB)

3 dB pad

2nd LNA

(G = 34.4, NF = 2.14 dB)

378 MHz local oscillator

(+13 dBm)

6 dB pad

1st mixer

(G = -9 dB)

100 MHz lowpass filter

(G = -.1 dB)

IF amplifier (Dewdney)

(G = 27 dB, F = 6 dB)

IF amplifier (Sheehan)

(G = 19 dB, F = 6 dB)

IF amplifier (Sheehan)

(G = 19 dB, F = 6 dB)

IF OUTPUT (20-40 MHz)

Figure 5-3. Second Part of Receiver (in SST Blockhouse, outside Screened Room)
**Figure 5-4. Third Part of Receiver (in SST Blockhouse, inside Screened Room)**
Photograph 5-1. First Part of Receiver and Other Components (on Antenna Pole) (showing RF switch, first LNA, triplexer, and FSK receiver)
Photograph 5-2. Second Part of Receiver (in SST Blockhouse, outside Screened Room)
(showing triplexer, first mixer stage, and IF amplifiers)
Photograph 5-3. Third Part of Receiver (in SST Blockhouse, inside Screened Room) (showing imageless mixer stage, frequency synthesizer, baseband amplifiers, anti-aliasing filter, and FSK transmitter)
378 MHz crystal oscillator (Vectron CO-233FW-3YR) shift the band of interest down to an intermediate frequency (IF) band centered at 30 MHz. Further amplification of the signal is then provided by three IF amplifiers. The first is a resonance-tuned, multi-stage amplifier with passband 20—40 MHz (DRAO project, T. Landecker). The other two are single-chip, wide-band amplifiers (DRAO project, E. Sheehan). The 100 MHz lowpass filter is a 5-pole Butterworth design. It consists of silver-mica capacitors and inexpensive coils mounted on a small PC board. The board includes 50 Ω striplines and ground planes on both sides. The filter removes the relatively large 378 MHz signal that leaks through the mixer from the LO.

5.3. THE IMAGELESS MIXER STAGE

Referring to Figure 5-4, the imageless (single-sideband) mixer shifts the sub-band of interest to baseband. A general description of imageless mixers can be found in [Pett84]. A programmable frequency synthesizer (Syntest SM-160-05) serves as the local oscillator. The microcomputer can select a 500 kHz sub-band by setting the frequency of the synthesizer within the range 20—40 MHz.

Because both the sub-band of interest and the image band are variable in frequency, an imageless mixer is required. The alternative to an imageless mixer is an electronically-tunable, high-order bandpass filter for image rejection followed by a regular mixer. Such a filter would be more expensive and time-consuming to build than the imageless mixer. Another alternative is a bandpass filter after the first mixer (of bandwidth equal to the sub-bandwidth). Such a filter would be very difficult to build because of its very narrow relative bandwidth.
A diagram of the imageless mixer is shown in Figure 5-5. The design is based on an earlier DRAO project by B. Penny. It was modified for use at a lower output frequency band. It works as follows: In-phase and quadrature signals are generated by mixing the signal with the LO and a 90°-shifted version, respectively. Low-pass filters remove unwanted products. Balanced all-pass L-C networks introduce another 90° shift (relative to each other). Transformers balance/unbalance the signals at the inputs/outputs of the networks. A resistive bridge adds the two signals to generate a baseband version of the lower sideband of the original input signal. Image suppression occurs when the unwanted signals are subtracted (cancelled) in the bridge. Potentiometers are included so that the image suppression can be fine-tuned. Padding is used at various points throughout the circuit to improve matching and to reduce the effects of reflections.

The all-pass network component values were calculated using formulas given in [Albe69]. The SPICE program was used to select the bridge resistors needed to present each network with a 50 Ω load.

The output of the imageless mixer does not go down to the lower cutoff frequency of the transformers (3 kHz) because the coils in the all-pass networks are too lossy at low frequencies. Impedance measurements determined that the lowest cutoff that could be achieved with off-the-shelf coils is 20 kHz. (At lower frequencies, the resistive part of the coils dominates the impedance.) The networks were then designed for a passband of 20–600 kHz.
LO INPUT
20–40 MHz

IF INPUT 20–40 MHz

3 dB pad
splitter
(Mini-Circuits PSC 2-1)

6 dB pads
mixers
(Mini-Circuits SRA-1)

6 dB pads
lowpass filters
(3-pole Butterworth, \( f_c = 8.1617 \) MHz)

1:1 transformers
(Mini-Circuits T1-6T)

network A
network B

balanced all-pass L-C networks
(A lags B by 90°)

balanced bridge

BASEBAND OUTPUT 20–600 kHz
(lower sideband)

Figure 5-5. Imageless Mixer
Figure 5-6. All-Pass Networks in Imageless Mixer
The potentiometers were adjusted to obtain maximum image suppression without significant reduction in the wanted signal. A suppression of 40 dB was achieved. This is close to the specified value of dynamic range (section 2.3.2).

The insertion loss of the mixer is 35 dB. This is large and requires that we include baseband amplification before the signal is digitized by the spectrum analyzer. Referring to Figure 5-4, after the imageless mixer, baseband amplification is applied to the signal. Two amplifiers are versions of the single-chip IF amplifiers, modified for use at lower frequencies, and the third is a commercial power amplifier. The signal is then fed into the anti-aliasing filter and on to the FFT spectrum analyzer. (The A-D circuitry in the analyzer includes an operational amplifier, section 3.4.3.) The overall receiver gain plus A-D gain were designed and adjusted to provide a suitable noise level for A-D conversion (section 6.2).
6. TESTING AND SYSTEM OPERATION

Testing was an important facet in the development of the interference monitor. Each component was tested as thoroughly as possible within the limitations of available equipment and time. Particular attention was given to the testing of the FFT spectrum analyzer as well as the complete system. In this chapter, the test procedures for the analyzer and the monitor are described, and some results are presented. Also described is the operation of the system under microcomputer control.

The basic questions that we want to answer are: How does the monitor respond to narrowband signals, i.e., does it produce any spurious peaks? Can a small signal be detected in the presence of a large signal? What limits the detectability of very small signals? Do cross-products from multiple signals affect the performance? And how does the monitor respond to broadband noise, i.e., how much does averaging of the power spectra reduce the noise variance? The answers to these questions indicate the level of minimum detectable signal and the dynamic range that can be achieved.

Before proceeding with the description of the testing and system operation, we shall calculate the integration time required to detect a signal of a given strength. Recall that the variance of the Welch estimate for noise is \( \sigma^2(k) = P^2(k)/K \) (Equation 3-2). Let \( N_{\text{bin}} \) be the noise power level in a frequency bin and \( N_{\text{rec}} = -81 \text{ dBWHz}^{-1} \) (measured) be the noise power density at the analog input of the spectrum analyzer. Then, considering noise only:
\[
\sigma(k) = \frac{N_{\text{bin}}}{\sqrt{K}}
\]

and

\[N_{\text{bin}} = N_{\text{rec bw}}.\]

If we assume that a signal can be detected by thresholding when it is ten times greater than the standard deviation of the noise, then:

\[10\sigma = \text{signal level} = P_i G_R\]

where

\[P_i = \text{flux of signal (dBWm}^{-2})\]
\[G_R = \text{gain of receiver} = 115 \text{ dBm}^2.\]

The integration time is:

\[T_i = KN/f_s = K/\text{bw} = \frac{\text{bw} 10[N_{\text{rec}}(\text{SNR loss})]^2}{(P_i G_R)^2} \text{ sec}\]

where "SNR loss" includes any loss due to windowing and scalloping. If the signal is CW and is on-axis for a helix, and assuming the worst-case loss in SNR, the time required to detect the minimum detectable signal (\(-188 \text{ dBWm}^{-2}\)) is 1.2 hours.
6.1. TESTING OF THE FFT SPECTRUM ANALYZER

Testing of the analyzer was divided into two categories: low-speed and high-speed tests. The low-speed tests were used during development to check the functioning of the boards. High-speed tests were conducted to determine the performance with real-time signals.

6.1.1. Low-Speed Functional Tests

Ideally, the best way to test the digital portion of the analyzer (and parts thereof) would involve inputting a known ("canned") data sequence in real-time, uploading the results, and comparing them to the correct results. For example, data could be generated on a mainframe computer, downloaded to a high-speed RAM buffer, and then fed to the analyzer in real-time. The results could then be read out to the buffer in real-time, transferred to the mainframe, and checked. Unfortunately, a high-speed buffer (with appropriate interface circuitry) was not available and there was not enough time to build one. Instead, low-speed tests with canned data were conducted by using the microcomputer to exercise the boards. (The microcomputer turned out to be a very useful tool.) On each board, special headers are wired for connection to the microcomputer ports. Sub-sections were tested separately by running C-coded test routines on the microcomputer. In a typical routine, the microcomputer provides test data, addresses, and control signals to the board(s) and then retrieves and checks the results. The different tests are outlined briefly below.

1. Test BUTT Board. Micro places input data on bi-directional data buses and sin/cos addresses on address bus, board does butterfly under micro control (micro generates control signals), micro retrieves output from data buses. User then compares results to those from FORTRAN butterfly routine.
2. Test PS Board. Micro places input data on data bus, board squares and accumulates under micro control (micro generates address and control signals), upload "power spectrum", micro checks results.

3. Test PS Board Under Control of C&M Board. Same as above except control and address generated by C&M Board which is externally-clocked by micro.

4. Quick Test of Analyzer (Excluding AD Board). Micro provides "windowed data" to analyzer while clocking it externally. Upload power spectrum when pipeline finished. User checks results.
   (Data = DC level + two sinusoids.)

5. Extensive Test of Analyzer (Excluding AD Board). Same as above except more extensive.
   Data = one sinusoid (bin-center or halfway between centers).
   Frequency is incremented after each upload until entire band is covered.
   Send results to mainframe computer for comparison to results of FORTRAN simulation.

The above tests were valuable verification and debugging tools during the development of the analyzer. Wiring and logic errors were quickly tracked down; the wire-wrap connection technique allowed easy modifications.

The same tests can be used in the future to isolate a fault (e.g. a bad chip). In addition, special jigs were wired and software developed to test 8x8 and 16x16 multipliers, 4-bit adders, and PROMs.

6.1.2. High-Speed (Real-Time) Tests

In this section, the FFT spectrum analyzer is tested with sinusoidal inputs. Two high-speed tests were performed and the simulation program was run. In each case, the result is a plot of the power spectrum. The three sources of results are summarized below:

1. FFT processor output power spectrum
2. floating-point power spectrum of A-D samples

3. simulation power spectrum.

The first test checked the performance of the entire analyzer (plus anti-aliasing filter). A spectrally-pure test signal of precise frequency was inputted, the power spectrum was computed and averaged, and the output was uploaded and plotted. The procedure was done with two different integration times, corresponding to 19,968 and 199,680 averages (9,984 and 99,840 FFTs). The longer integration was done to check the statistical accuracy of the shorter integration.

The second test measured the integrity of the A-D converter. The same set-up was used except that the digitized data samples (output of the A-D converter) were captured with a logic analyzer and then transferred to a mainframe computer via an RS-232 (terminal line) link. There, the Welch process was performed on the sample blocks (256-points each) in floating-point arithmetic. A 1024-point FFT (256-point block padded with 768 zeros) was used to yield a better view of the spectrum. The procedure was done for both 1 and 7 averages. No more than 7 averages were computed because the process of capturing and uploading the data was laborious.

The simulation program was run to see how the analyzer should perform with an ideal signal and perfect A-D conversion.

In each case, three kinds of input signal were considered:

1. single sine-wave at various amplitudes
2. single sine-wave at two different frequencies (bin-center or halfway between centers)

3. two sine-waves.

In the case of two sine-waves, the outputs of two generators were combined with a splitter and then fed to the anti-aliasing filter. The difference frequency was chosen to lie at the center of a bin so that any cross-products (or aliased versions) would lie at the center of a bin. (This is a property due to the segmentation of the data in the Welch process and results because bin-centered frequencies always contain an integer number of signal cycles in a segment.) This should yield the worst-case (largest possible) cross-product levels.

A typical set of results for a moderately-sized sinusoid and a pair of sinusoids is presented on the following pages (Figures 6-1a,b,c,d,e). The plots are scaled such that 0 dB is the level of a full-scale (5 Vpp) sinusoid. The peak on the far left of the spectra is the DC component. Some observations drawn from these plots, and other plots not included, are listed below:

1. The FFT spectrum analyzer produces the correct Kaiser-Bessel shape.

2. In either of the high-speed tests, the number of averages is adequate because there is little difference between plots of different averages.

3. Spurious peaks are present in both high-speed tests and the simulation. In general, the largest peaks have similar magnitudes (≈ −55 to −65 dB) but do not necessarily occur at the same locations.

4. The spurious peaks are bigger for very large signals.
5. For either high-speed test, there is little difference in the responses to a bin-center signal and a signal halfway between bins. However, the simulation produced no spurious peaks for a signal halfway between centers.

6. In the case of two large sinusoids, the level of the cross-products are no higher than the spurious peaks.

We can conclude from the tests that the digital portion of the FFT spectrum analyzer is working well for sinusoidal signals. However, the A-D converter produces spurious peaks in the spectrum, i.e., it limits the performance in these tests. For very large signals, these peaks can limit the dynamic range of the interference monitor.

6.2. A-D INPUT LEVEL AND POWER SPECTRUM DISTORTION

There are conflicting requirements in setting the level of the noise at the input to the A-D converter:

1. A high level is needed to minimize the distortion of colored noise.

2. A low level is needed to leave enough "headroom" for large signals and to minimize the noise generated by clipping in the A-D converter.

In section 3.5.5, it was observed that the power spectrum can be distorted (the valleys de-accentuated) when small numbers are rounded to 0 after squaring. To determine a suitable level of noise to input to the A-D converter, a simple experiment was performed. A "radio-quiet" band was selected (with a helix pointing away from the observatory buildings) and the spectrum was integrated until the shape of the anti-aliasing filter was visible in the spectrum (e.g. Figure 6-3a (top)). Spectra were recorded for various levels of noise. (The level was changed by adjusting the gain of the A-D circuit and by inserting
FFT Processor Power Spectrum of A-D Samples (1996800 averages)

Kaiser-Bessel (α=2) window
freq. = 351.5625 kHz
ampl. = 9 Vpp

Floating-Point Power Spectrum of A-D Samples (7 averages)

Kaiser-Bessel (α=2) window
freq. = 351.5625 kHz
ampl. = 9 Vpp

Figure 6-1a. Plots from High-Speed Analyzer Tests
Figure 6-1b. Plots from High-Speed Analyzer Tests (cont'd)
Figure 6-1c. Plots from High-Speed Analyzer Tests (cont'd)
Figure 6-1d. Plots from High-Speed Analyzer Tests (cont'd)
Simulation Power Spectrum of Signals (2 averages)

Figure 6-1e. Plots from High-Speed Analyzer Tests (cont'd)
padding after the anti-aliasing filter.) The mean and standard deviation of the ripple in the passband was calculated. A plot of the ratio $\sigma/\mu$ versus A-D input level is shown in Figure 6-2. We see that for small gain, the ratio is large, i.e., the spectrum is distorted. For large gain, the graph begins to flatten. However, at the largest values of gain, the clipping noise is severe (observed as a dramatic rise in the stop-band of the spectrum).

It was decided to improve the spectrum distortion without introducing too much clipping noise. A number of hardware changes were considered; the best choice, given the time constraints, is to scale the output of the FFT (X(k)) upwards to reduce the number of zeros. The ribbon cables connecting the PS and C&M Boards were modified to shift the data up by four bits. This reduces the headroom but fairly large signals can still be accommodated without overflow in the accumulator.

The experiment was repeated with the modified ribbon cables and a plot of the results is included in Figure 6-2. We see that the ratio $\sigma/\mu$ flattens out before significant A-D clipping occurs. A padding of 6 dB and an A-D gain of 10 were chosen for the final implementation.

6.3. SYSTEM OPERATION

At this point, it is appropriate to describe how the interference monitor operates under microcomputer control. This will prepare the reader for the next section on system tests.
Figure 6.2. $\sigma/\mu$ of Passband versus Input Level

Input Level in dB rel. unity A-D gain and no padding
Via a terminal link with the microcomputer, the user is presented with a menu of programs including one general "surveillance" routine. This program monitors a frequency band for signals that exceed a given threshold. The user first inputs the bandwidth, integration time, and threshold. The program then sweeps through the helical antennas one at a time, each time scanning the sub-bands for interference. A sub-band is selected by setting the frequency of the second LO. After each LO change, the program waits for the LO to settle and then resets the spectrum analyzer. The reset causes the data pipeline to be flushed. The microcomputer then polling the DATAREADY line until it is lowered by the analyzer, indicating the end of an integration period. The microcomputer then uploads the power spectrum and, if required, continues the integration. After integration, the data is normalized with a spectrum from a "radio-quiet" band, split-window normalized to remove any trends (slopes and curvatures), and finally checked for peaks that exceed the threshold. Any detections ("hits") are uploaded to the SST control computer and the cycle begins anew.

The other routines allow the user more manual control of the system, e.g., there are routines to turn on one antenna, set the second LO, set the FFT parameters, capture one spectrum, etc. These routines were used during testing and are useful when manual control is desired.

6.4. SYSTEM TESTS

To test the interference monitoring system, a stub antenna connected to a signal generator was set up on the roof of the main building and a CW signal was broadcasted (similar to the set-up in section 2.3.1) The helix that points
towards the main building was used to observe the signal. Two kinds of signals were considered:

1. single sine-wave at various levels
2. two sine-waves (one large, the other at various levels).

In the case of two sine-waves, the outputs of two generators were combined with a splitter and then fed to the stub antenna. The testing process was complicated by the presence of other signals in the sub-band of interest (generated by electronics in the main building). However, the presence of the signals do illustrate the need for a system such as the one considered in this thesis.

Some of the results for very large and very small signals are presented in the following pages (Figures 6-3a,b,c,d). The plots are scaled such that the background noise level is 0 dB. In the headings of the plots, "normalized" means the spectrum has been divided by the spectrum of the "radio-quiet" band; "normalized**" means that the normalized spectrum has been further processed with the split-window normalizer. Also, the edges of the normalized band are not shown because they contain artifacts of the normalization process. Some observations drawn from these plots, and others not included, are listed below.

1. The variance of the background noise decreases with averaging in agreement with the simulation for an integration time up to 1.4 hours (20 million averages). (Further averaging was not done due to time constraints.)

2. A $-183 \text{dBWm}^{-2}$ signal can be detected after both normalizations.

3. The presence of a large signal slightly degrades the detectability of the minimum detectable signal by creating spurious peaks.
4. The largest cross-products from two large signals is 31 dB below the signal levels. This limits the detectability of small signals in the presence of multiple large signals.

We can conclude from the tests that a $-183 \text{ dBWm}^{-2}$ signal can be detected if there are no large signals in the passband. This level is 5 dB below the minimum detectable signal. If there is a large signal, the minimum detectable signal is a few dB higher. Also, the split-window normalizer must be applied before a small signal can be detected by a threshold test.

6.5. EXAMPLES OF INTERFERING SIGNALS OBSERVED WITH MONITOR

A number of signals in the radio astronomy and neighboring bands were detected with the interference monitor. In particular, numerous signals from the observatory’s main building and SST blockhouse were observed in the radio astronomy band. Plots of some of the signals are presented in the following pages (Figures 6-4a,b). These signals are probably contaminating the SST’s observations at this time. They are at a low enough level that they would not be detected by other techniques. With the aid of the monitor, these signals will be tracked down and eliminated.

†The transmitter happened to lie at the crossover point of the helices, therefore the dBm values in the plots should be decreased by 4 dBm to compare them to the minimum detectable signal level derived in Chapter 2.
Figure 6-3a. Plots from System Tests
Figure 6-3b. Plots from System Tests (cont’d)
Figure 6-3c. Plots from System Tests (cont'd)
Normalized Power Spectrum of -55 and -95 dBm Test Signals (9,984,000 FFTs)

**Figure 6-3d. Plots from System Tests (cont’d)**
Figure 6-4a. Plots of Interfering Signals
Figure 6-4b. Plots of Interfering Signals (cont'd)
7. SUMMARY AND CONCLUSIONS

In summary, this thesis describes the engineering of a system to monitor interfering signals, in particular, those signals that can contaminate the maps produced by the 408 MHz aperture synthesis telescope near Penticton, B.C. The resulting system provides continuous, automated surveillance of the radio spectrum around 408 MHz. Small signals can be detected quickly by averaging spectra in real-time. Valuable information about the interference is recorded and made available to the astronomers. The system is a useful tool with which the astronomers can improve the accuracy of their observations. To the knowledge of the author and the members of the observatory, no other similar system exists.

The concept of the interference monitor relies on the assumption that most interfering signals are narrowband. The design is based on the estimates of the signal levels derived in Chapter 2.

The project required knowledge in a number of diverse areas: antennas, RF electronics (amplifier and filter design, design of receivers), high-speed digital electronics, signal processing, and software. The most innovative work was in the FFT spectrum analyzer and the antenna system.

A full discussion of the details required to build a fixed-point FFT machine for any situation requiring long integration is given. This detailed work on the FFT spectrum analyzer that comprises Chapter 3 draws on many sources of reference and includes some innovations. No equivalent compilation in the
literature is known to the author. Chapter 3 can therefore serve as a reference for those who want to develop a high-speed FFT-based spectral processor.

Numerous aspects of the design, construction, and performance of an FFT spectrum analyzer were studied. Included is a survey of real-time narrowband processing techniques and a study of various alternatives in the processor architecture. The most efficient procedure for processing sub-bands of a larger band using a bandwidth-limited processor was noted. The sources of fixed-point error and their effects in a Welch processor were compiled. Some new results concerning the detectability of small signals were presented (e.g. the limitation due to fixed-point errors, power spectrum distortion). A computer simulation of the Welch process was developed and used in various aspects of the design and testing of the analyzer. The special-purpose hardware that comprises the analyzer was designed and built from scratch. A high-speed processor was achieved at a fraction of the cost of commercial machines.

In the antenna system, an array of eight helical antennas was designed and constructed, its characteristics were investigated and found to be suitable for the present application, and a method of remotely switching them on and off was devised. The antennas and electronics box, designed to operate in an outdoor environment, have survived a winter and high winds in the spring.

Since the interference monitor was intended to be, and is, a working, useful system, testing was a major part of the work. Each component in the system was tested as thoroughly as possible within the limitations of available
equipment and time. Particular attention was given to the testing of the FFT spectrum analyzer. These tests were divided into two classes: low-speed functional tests and high-speed (real-time) tests. As well, extensive testing of the system was conducted with transmitted test signals. The performance came within 5 dB of the minimum detectable signal criterion and met the dynamic range specification. Also, numerous real interfering signals have been observed, particularly from the observatory's own buildings.

Perhaps the most satisfying conclusion, at least in the opinion of the author, is that a sophisticated system has been designed and actually works!

7.1. FURTHER WORK

Although the present interference monitor is a useful tool in its own right, it is possible to make a few improvements. These are summarized as follows:

1. Add a second anti-aliasing filter and a switch to implement the ZOOM mode (section 3.4.3.1).

2. Develop software to increase the accuracy of the estimate of the interfering signal's frequency. Three possibilities are: a) interpolate between bins, b) deconvolve the power spectrum with the window, or c) shift the frequency of the second LO in increments much smaller than the bin-width to obtain a maximum response.

3. Develop software to increase the accuracy of the estimate of the interfering signal's direction. Two possibilities are: a) interpolate between antenna beams and/or b) beamform with two adjacent helices operating simultaneously.

4. Develop more sophisticated normalization routines to increase the detectability of very small signals.

5. Use an uninterruptable power supply to power the monitor. This would help in finding the sources of interference; by shutting off the observatory's main power (and hence all electronics), it could be determined if the source
of an interfering signal was on site or off.

Although the monitor is already a very useful tool for detecting and finding sources of interference, it is ultimately intended to communicate directly with the telescope's control computer, so that most interference can be removed automatically. This "ultimate" system could be realized fairly easily by writing appropriate software routines. However, the decision to implement it would be based on a lengthy term (a few more months) of reliable, successful operation in the present mode, i.e., the current system must achieve a "proven track record".

### 7.2. VALUE OF THESIS OUTSIDE OF PROJECT

Parts of this thesis are useful in areas outside of the original project. First, there are numerous radio observatories throughout the world, many of which observe at frequencies below 1 GHz and so are prone to interference. The results in this thesis can help them to design their own interference monitors. Second, the material is of interest to parties concerned with the use or management of the electromagnetic spectrum, especially at radio frequencies. And third, with the advent of new, high-speed digital technology, a number of observatories are considering, planning, or already building FFT-based spectrometers to perform the cross-correlation in the interferometer. The material in Chapter 3 is of value to these people and to anyone involved with FFT-based high-speed systems.

The work has been presented at two conferences. This has generated a fair amount of interest, expressed by requests for further information by
members of other observatories in the United States, Puerto Rico, and Brazil, as well as by others concerned with the use or management of the electromagnetic spectrum.

The radio spectrum is becoming more and more cluttered with man-made signals. In the future, radio astronomy observations may have to be made with large in-band interfering signals, especially in cases where natural transitions of molecules lie in unprotected frequency bands. Current spectrometer designs use very coarse quantization (section 3.2.3) under the assumption that no such interference is present. Future designs will have to be cognizant of the techniques used in this thesis to be able to observe weak signals in the presence of in-band interference.
CITED REFERENCES


[Harr78]

[Högbo74]

[Jasi61]

[Kasp81]

[Krau50]

[Krau66]

[Krau77]

[Land88]

[Morr71]

[NTC82]
*Table of Frequency Allocations 9kHz to 275GHz*, National Telecommunications Branch, Government of Canada, Jan. 1, 1982.

[Oppe72]

[Oppe75]

[Papo84]

[Pett84]


[Zver67]
[Blak79]  

[Coll85]  

[Croc83]  

[FAST87]  

[Hayw82]  

[Horo86]  

[Lanc85]  

[Lars67]  

[Morr86]  

[Parr84]  

[Peat80]  

[Robi82]  

[Ston82]  

[Stut81]  
[Thom82b]

[Tsui83]
J.B. Tsui, Microwave Receivers and Related Components, NTIS, 1983.

[Whit82]

[Wiat80]
### SAMPLE RESULT FILE

run parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>126.95313 kHz</td>
</tr>
<tr>
<td>F2</td>
<td>177.73438 kHz</td>
</tr>
<tr>
<td>FS</td>
<td>1000.0000 kHz</td>
</tr>
<tr>
<td>N</td>
<td>256</td>
</tr>
<tr>
<td>BADC</td>
<td>8</td>
</tr>
<tr>
<td>BW</td>
<td>8</td>
</tr>
<tr>
<td>BFFT</td>
<td>8</td>
</tr>
<tr>
<td>BOUT</td>
<td>31</td>
</tr>
<tr>
<td>A1</td>
<td>0.00000000</td>
</tr>
<tr>
<td>A2</td>
<td>0.00000000</td>
</tr>
<tr>
<td>C</td>
<td>1.00000000</td>
</tr>
<tr>
<td>S</td>
<td>4.00000000</td>
</tr>
<tr>
<td>NAVG</td>
<td>1048576</td>
</tr>
<tr>
<td>DSEED</td>
<td>1234567891</td>
</tr>
<tr>
<td>ISCALE</td>
<td>1</td>
</tr>
<tr>
<td>ISKIP</td>
<td>1</td>
</tr>
<tr>
<td>GAP</td>
<td>7</td>
</tr>
<tr>
<td>IDC</td>
<td>6</td>
</tr>
<tr>
<td>ISP</td>
<td>0</td>
</tr>
<tr>
<td>ISEED1</td>
<td>987654321</td>
</tr>
<tr>
<td>ISEED2</td>
<td>112233441</td>
</tr>
<tr>
<td>REBOOT</td>
<td>6</td>
</tr>
<tr>
<td>DRAO</td>
<td>VAX 11/780</td>
</tr>
</tbody>
</table>

window parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>coherent gain</td>
<td>-6.21 dB (float)</td>
</tr>
<tr>
<td>equivalent noise</td>
<td>-6.21 dB (fixed)</td>
</tr>
<tr>
<td>bandwidth of window</td>
<td>1.75 dB (float)</td>
</tr>
<tr>
<td></td>
<td>1.75 dB (fixed)</td>
</tr>
<tr>
<td>NAVG</td>
<td>2</td>
</tr>
<tr>
<td>BA</td>
<td>16</td>
</tr>
<tr>
<td>CARRY1,2</td>
<td>0</td>
</tr>
</tbody>
</table>

noise statistics (at output):

<table>
<thead>
<tr>
<th></th>
<th>floating-point</th>
<th>fixed-point</th>
<th>theoretical</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard deviation</td>
<td>-30.25 dB</td>
<td>-29.38 dB</td>
<td>-29.86 dB</td>
</tr>
<tr>
<td>decrease in std dev</td>
<td>2.01 dB</td>
<td>3.17 dB</td>
<td>1.51 dB</td>
</tr>
<tr>
<td>signal 2: SNR</td>
<td>-351.76 dB</td>
<td>-353.79 dB</td>
<td>-6.68 dB</td>
</tr>
<tr>
<td>SNvR</td>
<td>-349.75 dB</td>
<td>-350.82 dB</td>
<td>-5.17 dB</td>
</tr>
<tr>
<td>PNSR</td>
<td>-1.26 dB</td>
<td>-2.26 dB</td>
<td>-2.26 dB</td>
</tr>
<tr>
<td>PN level</td>
<td>-30.50 dB</td>
<td>-30.50 dB</td>
<td>-30.50 dB</td>
</tr>
<tr>
<td>mse</td>
<td>-30.19 dB</td>
<td>-30.19 dB</td>
<td>-30.19 dB</td>
</tr>
</tbody>
</table>

using 110 noise values out of 129 samples
PROGRAM SIMULATION

Program summary:
- Compute and average successive, non-overlapped power spectra of input signal = 2 sinusoids + Gaussian noise.
- Pack one data block into the real input array and one into the imaginary input array; unscramble the output to get the average of the two power spectra.
- Both floating-point and fixed-point processing are done in parallel so they can be compared.
- The floating-point FFT is decimation-in-time, in-place.
- The fixed-point FFT is decimation-in-frequency, constant geometry.
- Each sampled input data block is quantized to BADC bits.
- Each power spectrum sample is BSPEC bits which in turn accumulates up to BOUT bits via integration. (BA is calculated by program.)
- BOUT bits are kept for each averaged power spectrum sample.
- Output data to a file (channel 7) and a plotfile (channel 9).
- Window coefficients are in a file (channel 4).
- (Set BW=0 for rectangular windowing.)
- Copy the clipping and overflow results to a file (channel 18).
- Copy the run parameters and noise statistics to another file (channel 11).
- Restrictions on word lengths:
  - BADC < 31
  - 2+BFFT < 32
  - BFFT < BSPEC < 2+BFFT
  - BADC+OW < 31
  - 2*BFFT < 32
  - BSPEC < 2*BFFT-2
  - BADC < 31

Input file parameters:
- FI = frequency of sinusoid 1 (kHz)
- F2 = frequency of sinusoid 2 (kHz)
- BADC = length of 2's complement A-D word (bits)
- BW = length of unsigned window word (bits)
- (set = 0 for rectangle window; otherwise, Kolsa-Beissel (alpha=0))
- BFFT = length of 2's complement FFT word (bits)
- BSPEC = length of unsigned power spectrum word (bits)
- BOUT = length of unsigned final averaged power spectrum word (bits)
- A1 = peak amplitude of sinusoid 1 (linear, relative to unit half full-scale)
- A2 = peak amplitude of sinusoid 2 (linear, relative to unit half full-scale)
- C = std. dev. of noise (linear, relative to unit std. dev.)
- S = half full-scale (linear, relative to unit half full-scale)
  (e.g. A1=1, A2=2, C=1, S=2 produces sinusoids with peak amplitudes .85 and .1 and noise with std. dev. .5)
- OSEED - seed for random number generator
- I SCALE - scale factor for Integer data just before FFT
- ISKIP = use to copy overflow results to a file (no=0,yes=1)
- GAP = total number of points straddling signal bin to remove from noise calculations
- IDC = number of points on each side of DC bin to remove from noise calculations (total number = 2-IDC - 1)
- I3H = total number of points straddling a problem bin to be removed from noise calculations
  (e.g. 3rd harmonic of large sinusoid)
- I SEED1 - not used
- I SEED2 - not used
- J1 = number of FFT's until first printout
- J2 = number of FFT's for last printout (end of run)
  (note: the number of FFTs (= number of averages) is twice the number actually performed due to real data "trick")
- J3 = printout spacing factor
  (e.g. J1=2, J2=32, J3=4 will produce printouts after 2, 8, and 32 FFT's)
- IPLOT = use to plot power spectrum (no=0,yes=1)
- REBOOT = use to reboot program (e.g. after system crash) (no=0,yes=1)

Type and range of input file parameters:
- F1 = REAL*4 (.GT. 0.)
- F2 = REAL*4 (.GT. 0.)
- BADC = INTEGER*4 (.GE. 1) (see above restrictions)
- BW = INTEGER*4 (.GE. 0) (see above restrictions)
- BFFT = INTEGER*4 (.GE. 2) (see above restrictions)
- BSPEC = INTEGER*4 (see above restrictions)
- BOUT = INTEGER*4 (see above restrictions)
- A1 = REAL*4 (.GE. 0.)
- A2 = REAL*4 (.GE. 0.)
- C = REAL*4 (.GE. 0.)
- S = REAL*4 (.GT. 0.)
- DSEED = INTEGER*8 (large and odd)
- I SCALE = INTEGER*4 (1 or 2)
REAL DATA(256), DATA1(256), DATA2(256), POWERF(129)
& .FREQ(129), AVGF(129), AVOD1(129), AVODF(129)
& .NOISE(129), NOISEF(129), NSD, NSDF, NSD, NSD, MSE
& .WINDF(256)

INTEGER IDATA(256), IDATA1(256), IDATA2(256), IPower(129)
& .IDATARS(256), IDATAIS(256)
& .IAVG(129), IAVGL(129)
& .BADC, BW, BFFT, BSPEC, BA, BOUT
& .NSKIP(129), GAP, TR1, TR1, TR2, TR2
& .BADCW, ALIGN, SHIFT, SHIFTA
& .WINOW(256), DSEED, DSEED1, REBOOT

COMPLEX CDATA(256)

CHARACTER*10 FNAME, FILE12

CALL CPUTIME(8)
RTIME = SECONDS(8.)

OPEN(UNIT=10, FILE='XCLIST.OUT', STATUS='UNKNOWN', FORM='FORMATTED')

PI = 3.1415927
TWOP = 6.2831853
N = 256
M = 8

C sampling rate in kHz:
FS = 1000.

C read in parameters for this run

WRITE(6,99)

OPEN(UNIT=5, FILE='XRUN.DAT', STATUS='OLD', FORM='FORMATTED')

READ(5,*) F1, F2, BADC, BW, BFFT, BSPEC, BOUT
& .DSEED1, DSEED2

WRITE(6,100)

READ(5,*) A1, A2, C, S, DSEED, IScale, ISKIP, GAP, IDC, I3H
& .ISEED1, ISEED2

DSEED1 = DSEED
NNPTS = N/2 + 1 - 2-IDC - I3H
IF(A1.NE.8.) NNPTS = NNPTS - GAP
IF(A2.NE.8.) NNPTS = NNPTS - GAP

WRITE(6,112)

READ(5,*) J1, J2, J3, IPLOT, REBOOT

CLOSE(UNIT=5)

J1B2 = NINT( ALOG(J1+1.)/ALOG(2.) )
J2B2 = NINT( ALOG(J2+1.)/ALOG(2.) )
J3B2 = NINT( ALOG(J3+1.)/ALOG(2.) )
NOUT = (J2B2-J1B2)/J3B2 + 1

OPEN(UNIT=11, FILE='XNLIST.OUT', STATUS='NEW', FORM='FORMATTED')

WRITE(11,113) NOUT

NAVQ = J2

WRITE(11,101) F1, F2, FS, N, BADC, BW, BFFT, BSPEC, BOUT, A1, A2, C, S, NAVQ
& .IDC, ISKIP, IScale, ISKIP, GAP, IDC, I3H, ISEED1, ISEED2
& .REBOOT

NLOOP = J2/2

C set up signal locations vector

NF1 = NINT(N*F1/FS +1.)
NF1L = NF1 - (GAP-1)/2
NF1H = NF1 + (GAP-1)/2
NF2 = NINT(N*F2/FS +1.)
NF2L = NF2 - (GAP-1)/2
NF2H = NF2 + (GAP-1)/2
NF3H = NINT(3*N*F1/FS +1.)
DO 1 = 1, N/2 + 1
J = 0
1 IF( ( 1 .GE. NF1L .AND. 1 .LE. NF1H .AND. A1.NE.0.)
& .OR. ( 1 .GE. NF2L .AND. 1 .LE. NF2H .AND. A2.NE.0.)
& .AND. GAP.NE.0 )=J
1 NSkip(J) = J
DO 1111 I=1,1DC
1111 NSKIP(I) = 1
1111 NSKIP(N/2 + 1 - 1 - 1) = 1
1111 IF(13H.NE.0) THEN
1112 NSKIP(NF13H) = 1
1112 DO 1112 I=1,((13H-1)/2)
1113 NSKIP(NF13H-1) = 1
1113 IF(I3H.NE.0) THEN
1114 NSKIP(NF13H-1) = 1
112 ELSE
113 END IF
114 206 NF1MN = NF1 - 4
115 NF1MX = NF1 + 4
116 NF1MNR = N/2 - NF1MN + 2
117 NF1MXR = N/2 - NF1MN + 2
118 NF2MN = NF2 - 4
119 NF2MX = NF2 + 4
120 NF2MNR = N/2 - NF2MN + 2
121 NF2MXR = N/2 - NF2MN + 2
122 C define x-axis vector for plot and initialize average vector
123 DO 2 =1,N/2 +1
124 FREQ(I) = ((I-1) * FS/N
125 IAVGH(I) = 0
126 IAVGL(I) = 0
127 AVGF(I) = 0
128 260 NF1MN = NF1 - 4
129 NF1MX = NF1 + 4
130 NF1MNR = N/2 - NF1MN + 2
131 NF1MXR = N/2 - NF1MN + 2
132 NF2MN = NF2 - 4
133 NF2MX = NF2 + 4
134 NF2MNR = N/2 - NF2MN + 2
135 NF2MXR = N/2 - NF2MN + 2
136 C read In window coefficients and calculate their sum SUM
137 C and the sum-of-squares SUMSQ
138 IF(BW.NE.0) THEN
139 OPEN ( UNIT-4, FILE='KALIBESS.DAT', STATUS='OLD', FORM='FORMATTED')
140 SUM = 0.
141 SUMSOI = 0.
142 SUMF = 0.
143 SUMSOF = 0.
144 SCALEW = 2.*BW - 1.
145 DO 21 =1,N
146 READ(4,115)W
147 IW = NINT(W.SCALEW)
148 SUM = SUM + IW
149 SUMSOI = SUMSOI + FLOAT(IW) ** 2
150 SUMF = SUMF + W
151 SUMSOF = SUMSOF + W ** 2
152 WINDOW(I) = IW
153 WINDWF(I) = W
154 SUM = SUM/SCALEW
155 SUMSOI = SUMSOI/(SCALEW ** 2)
156 WFACE = SCALEW/(SCALEW+1.)
157 2137 SUMSQ = SUMSG1/(SCALEW+2)
158 WFACE = SCALEW/(SCALEW+1.)
159 239 END IF
159 ELSE
160 SUM = N
161 SUMSOI = N
162 SUMF = N
163 SUMSOF = N
164 WFACE = 1.
165 END IF
166 ELSE
167 SUM = N
168 SUMSOI = N
169 SUMF = N
170 SUMSOF = N
171 WFACE = 1.
172 END IF
173 ELSE
174 JOUT = J1
175 JSTART = 1
176 ADJ1 = 0.
177 ADJ2 = 0.
178 F1INC = TWOPI+F1/FS
179 F2INC = TWOPI+F2/FS
180 IFLP = 0
181 NSAVE = 0
182 ELSE
183 IF(REBOOT.EQ.1) THEN
184 OPEN(UNIT=14, FILE='XYING.OUT', STATUS='OLD', FORM='FORMATTED')
185 READ(14)JSYING
186 CLOSE(UNIT=14)
187 IF(REBOOT.EQ.1) THEN
188 OPEN(UNIT=14, FILE='XYANG.OUT', STATUS='OLD', FORM='FORMATTED')
189 READ(14)JSYANG
190 CLOSE(UNIT=14)
191 IF(JSYING.GT.JSYANG) THEN
192 FILNAM = 'XYING.OUT'
193 ELSE
194 FILNAM = 'XYANG.OUT'
195 END IF
196 OPEN(UNIT=14, FILE=FILNAM, STATUS='OLD', FORM='FORMATTED')
197 READ(14)JUNK, JSTART, JOUT, DSEED, ISEED1, ISEED2, SE
198 & JSTART = JSTART + 1
199 DO 218 I=1,N/2 +1
200 READ(14)AVGF(I), IAVGH(I), IAVGL(I)
201 CL0SE(UNIT=14)
202 ELSE
DO 283 J=START,NLOOP
285 DO 578 K=1,2
288 NSAVE = NSAVE + 1
289 JSPEC = 2*J + K
290 C WRITE(6,182)JSPEC
292 C WRITE(10,182)JSPEC
295 C generate Gaussian noise vector (unit variance)
296 C (check for repetition of seed DSEED)
297 DO 22 I=1,N
298 SUMRN = 0.
299 DO 221 L=1,12
300 SUMRN = SUMRN + RAN(DSEED)
301 IF(DSEED.EQ.DSEED1) THEN
302 OPEN(UNIT=12, FILE="XSEED.OUT", STATUS='UNKNOWN', FORM='FORMATTED')
303 WRITE(12,116)JSPEC, I, L
304 CLOSE(UNIT=12)
305 ELSE
306 END IF 221 CONTINUE
307 22 DATA(I) = SUMRN + 6.
308 C generate sampled input data = signal(s) + noise
309 DO 3 I=1,N
310 INDEX = I - 1
311 ARG1 = F1INC*INDEX + ADJ1
312 ARG2 = F2INC*INDEX + ADJ2
313 SIGNAL = A1*SIN(ARG1) + A2*SIN(ARG2)
314 3 DATA(I) = ( SIGNAL + CM DATA(I) ) / S
315 ADJ1 = AMDG(ARG1, TWOP1) + F1INC
316 ADJ2 = AMDG(ARG2, TWOP2) + F2INC
317 C analog-to-digital conversion
318 C (clip the data and convert to 2's complement integers)
319 NNCLIP = 8
320 NPCLIP = 0
321 SCALEI = 2.***(BADC-1) - 1.
322 DO 4 I=1,N
323 IF(DATA(I).LT.-1.)THEN
324 DATA(I) = -1.
325 NNCLIP = NNCLIP + 1
326 ELSE
327 END IF
328 4 CONTINUE
329 C window the time samples
330 IF(BW.NE.8) THEN
331 DO 33 I=1,N
332 DATA(I) = DATA(I)*WINDWF(I)
333 DATA2(I) = DATA2(I)*WINDWF(I)
334 33 DATA1(I) = IDATA(I)*WINDWF(I)
335 ELSE
336 END IF
337 33 CONTINUE
338 C pack two data blocks into (floating-point) complex array
339 DO 3 1=1,N
340 CATA(I) = CMPLX(DATA1(I), DATA2(I))
341 C If, after windowing, data word size does not match BFFT,
342 C scale up or down accordingly (if down, then round result)
343 BADC = BADC + BW
344 IF(BADC.LT.BFFT) THEN
345 C window the time samples
346 IF(BW.NE.8) THEN
347 DO 33 I=1,N
348 DATA1(I) = DATA1(I)*WINDWF(I)
349 DATA2(I) = DATA2(I)*WINDWF(I)
350 33 DATA1(I) = IDATA1(I)*WINDWF(I)
351 ELSE
352 END IF
353 33 CONTINUE
354 C pack two data blocks into (floating-point) complex array
355 DO 3 1=1,N
356 CATA(I) = CMPLX(DATA1(I), DATA2(I))
357 C If, after windowing, data word size does not match BFFT,
ALIGN = 2*(BFFT-BADC)

DO 6 I = 1,N
  IDATAR(I) = IDATAR(I) + ALIGN
  IDATAI(I) = IDATAI(I) + ALIGN
  ELSE
  IF(BADC.GT.BFFT) THEN
    ALIGN = 2*(BADC-BFFT)
    DO 61 I = 1,N
      T = IDATAR(I) + ALIGN/2
      IDATAR(I) = (T + MIN0(0,ISIGN(1,T)))/ALIGN
      T = IDATAI(I) + ALIGN/2
      IDATAI(I) = (T + MIN0(0,ISIGN(1,T)))/ALIGN
    61 ELSE
      END IF
  END IF
  END IF

C If desired, pre-scale the data down by 2 before FFTing
IF(ISCALE.EQ.2) THEN
  DO 7 I = 1,N
    T = IDATAR(I)
    IDATAR(I) = (T + MIN0(0,ISIGN(1,T)))/2
    T = IDATAI(I)
    IDATAI(I) = (T + MIN0(0,ISIGN(1,T)))/2
  7 ELSE
    END IF

C FFT the windowed time samples
CALL FFTI(IDATAR,IDATAI,IDATARS,IDATAIS,M,N,BFFT,ISKIP,JSPEC)
CALL FFTF(COATA,M,N,N)

C compute power spectrum (while unscrambling data)
SHIFT = 2**((2*BFFT - 1 - BSPEC)
TR1 = IDATAR(1)**2 + SHIFT/2
TI1 = IDATAI(1)**2 + SHIFT/2
TR1 = TR1/SHEFT
TI1 = TI1/SHEFT
POWER1 = 2*(TR1 + TI1)
POWERF = REAL(COATA(1))**2 + AIMAG(COATA(1))**2
DO 9 I = 2,N/2 + 1
  TR1 = IDATAR(I)**2 + SHIFT/2
  TI1 = IDATAI(I)**2 + SHIFT/2
  TR1 = TR1/SHEFT
  TI1 = TI1/SHEFT
  POWER1 = TR1 + TI1 + POWER1
  POWERF = REAL(COATA(I))**2 + AIMAG(COATA(I))**2
  END IF
9 CONTINUE

C add power spectrum to running average
C (fixed-point accumulator spans two integer*4 numbers)
DO 10 I = 1,N/2 + 1
  IAVGL(I) = IAVGL(I) + POWER1
  IF(BTEST(IAVCL(I),38)) THEN
    IAVGL(I) = IBCLR(IAVGL(I),38)
    IAVGH(I) = IAVGH(I) + 1
  ELSE
    END IF
10 CONTINUE
C calculate coherent gain CG of window
IF(JSPEC.EQ.J1.OR.REBOOT.EQ.1) THEN
  IF(REBOOT.EQ.0) REBOOT = 1
  CGF = SUMF/N
  CGI = SUMI/N
  WRITE(6,106) 10*ALOG10(CGF),10*ALOG10(CGI)
  WRITE(11,106) 10*ALOG10(CGF),10*ALOG10(CGI)
C calculate equivalent noise bandwidth ENBW of window
ENBF = N*SUMG0/(SUMF+2)
ENBW = N*SUMG0/(SUMI+2)
C calculate squared error of complex spectrum
SCE = 2**((BFFT-1) + SCALE1/(SCALE1+1) * WFAC / SCALE
DO 100 I = 1,N/2 + 1
  IF(NSKIP(I).EQ.0) THEN
    SE = SE
    + CABS(COMPLEX(IDATAR(I),IDATAI(I)))/SCE - COATA(I))**2
  ELSE
    END IF
100 CONTINUE
C calculate scaling factor so maximum possible signal
C comes out at 6dB
IF(JSPEC.EQ.JOUT) THEN
  SCALEO = (.5*CGF)**2 * JSPEC
calculate sealing factors for converting fixed-point data
C to floating-point (note: *AOCFAC* is needed because *SCALE* compensates for fixed-point unscrambling)
C (not 2*(BADC-1); ditto for *WFAC*)

\[ BA = BSPEC + 1 + NINT(ALOG(JSPEC+1.)/ALOG(2.)) \]
\[ SHIFTA = 2**\left( MAX(0, BA-BOUT) \right) \]
\[ ADFAC = 1 - SCALEI/(SCALEI-2**\left( BSPEC-1 \right) + ADFAC*WFAC) \]
\[ SCALE = 2**\left( BADC-1 \right) + ADFAC*WFAC \]

C calculate mean-squared error of complex spectrum
C (fixed-point relative to floating-point)
\[ MSE = SE/(\left( 0.5*CGF \right) + 2) )/NNPTS \]
C compute noise level
\[ IF(C.NE.8.) THEN \]
\[ NLI = 0, NLF = 0 \]
\[ DO 11 = 1, N/2 + 1 \]
\[ T = IAVGL(1)/SHIFTA + IAVGH(1)*(2**36/SHIFTA) \]
\[ TEMPI = T/SCALE /SCALEO \]
\[ TEMPF = AVGF(1)/SCALEO \]
\[ IF(NSKP(1).EQ.0) THEN \]
\[ K = NLI + TEMPI \]
\[ NLI = NLI + TEMPI \]
\[ NLF = NLF + TEMPF \]
\[ ELSE \]
\[ NLI = NLI + TEMPI \]
\[ ENDIF \]
\[ CONTINUE \]
\[ NLI = NLI/NNPTS \]
\[ NLF = NLF/NNPTS \]
C calculate noise-to-maximum signal ratio
C and noise-to-signal ratio
C where signal = signal 1 ;A1.NE.8
C = input noise ;A1.EQ.0
C and noise = noise due to fixed-point processing
\[ PNSR = NLI - NLF \]
\[ IF(A1.EQ.0.) THEN \]
\[ PNSR = (NLI-NLF) / NLF \]
\[ ELSE \]
\[ PNSR = (NLI-NLF) / (AVGF(NF1)/SCALEO - NLF) \]
\[ ENDIF \]
C calculate rms noise = standard deviation
\[ NSDI = 0, NSDF = 0 \]
\[ DO 12 = 1, NNPTS \]
\[ NSDI = NSDI + (NOISEI(I)-NLI)**2 \]
\[ NSDF = NSDF + (NOISEF(I)-NLF)**2 \]
\[ NSDI = SORT(NSDI/NNPTS) \]
\[ NSDF = SORT(NSDF/NNPTS) \]
C calculate theoretical values of noise level and standard deviation
C relative to maximum possible signal
\[ NLT = C**2/S**2 *N/ENBWF \]
\[ NSDT = NLT/SORT(FLOAT(JSPEC)) \]
C calculate resulting decrease in noise standard deviation
C due to averaging
\[ DNSDF = 10**ALOG10(NSDF/NSDI) \]
\[ DNSDT = 10**ALOG10(NSDT/NSDI) \]
C output noise data to terminal and file
\[ WRITE(6,187)JSPEC,BA,IAVGH(NF1),IAVGH(NF2) \]
\[ ,18*ALOG18(NLF),18*ALOG18(NLI),18*ALOG18(NLT) \]
\[ ,DNSDF,DNSDI,DNSDT \]
\[ IF(A1.NE.8.) THEN \]
\[ SI = T/SCALE /SCALEO - NLI \]
\[ IF(SI.LT.1E-38) SI = 1E-38 \]
\[ SF = AVGF(NF1)/SCALEO - NLF \]
\[ IF(SF.LT.1E-38) SF = 1E-38 \]
\[ SNR = SI - 10*ALOG10(NLI) \]
\[ SNRF = SF - 10*ALOG10(NLF) \]
\[ SNRT = 10**ALOG10((A1/S)**2) - 10**ALOG10(NLT) \]
SNSDRI = SI - 10*ALOG10(NSDI)
SNDRF = SF - 10*ALOG10(NSDF)
SNDRD = SI - 10*ALOG10((A/S)**2) - 10*ALOG10(NSDT)

C WRITE(6,108)1,SNRF,SNRI,SNRT
C WRITE(6,109)SNSDFR,SNSDRS,SNSDRT
WRITE(11,108)1,SNRF,SNRI,SNRT
WRITE(11,109)SNSDRF,SNSDRI,SNSDRT

ELSE
IF(A2.NE.0.)THEN
T = IAVG1(NF2)/SHIFTA + IAVGH(NF2)*(2**30/SHIFTA)
SI = T/SCALE /SCALEO - NLI
SI = 10*ALOG10(SI)
SF = AVGF(NF2)/SCALEO - NLF
SF = 10*ALOG10(SF)
SNRT = SI - 10*ALOG10(NLI)
SNRT = 10*ALOG10((A2/S)**2) - 10*ALOG10(NLT)
SNDR = SI - 10*ALOG10(NSDI)
SNDRF = SF - 10*ALOG10(NSDF)
SNDRD = SI - 10*ALOG10(NSDT)

C WRITE(6,108)2,SNRF,SNRI,SNRT
C WRITE(6,109)SNSDRF,SNSDRI,SNSDRT
WRITE(11,108)2,SNRF,SNRI,SNRT
WRITE(11,109)SNSDRF,SNSDRI,SNSDRT

ELSE
IF(PRNSR.LT.1E-38)PNSR = 1E-38
IF(PNMSR.LT.1E-38)PNMSR = 1E-38
MSE = 1E-38
C WRITE(6,110)10*ALOG10(PNSR),10*ALOG10(PNMSR),10*ALOG10(MSE)
WRITE(11,110)10*ALOG10(PNSR),10*ALOG10(PNMSR),10*ALOG10(MSE)
C WRITE(6,111)NNPTS,N/2 +1
WRITE(11,111)NNPTS,N/2 +1

ELSE
ENDIF

C for each of the two signals:
C output to a file two portions of power spectrum: one in the
C neighborhood of signal and one in the neighborhood of the
C "image" signal that appears due to fixed-point processing
C also, output the values near DC
DO 1210 1=1,4
T = IAVG1(I)/SHIFTA + IAVGH(I)*(2**30/SHIFTA)
TDMP = T/SCALE /SCALEO
TDMP = AVGF(I)/SCALEO
SAVE = TDMP
SAVE = TDMP
IF(TDMPL.IE.1E-38)TEMPI = 1E-38
IF(TDMPL.LE.1E-38)TEMPI = 1E-38
IF(TDMPL.IE.1E-38)TEMPI = 1E-38
IF(TDMPL.LE.1E-38)TEMPI = 1E-38
TDMP = 10*ALOG10(TDMPL)
TDMP = 10*ALOG10(TDMPL)
WRITE(11,106)FRE0(I),TEMPF,TEMPI,SAVEF,SAVEI
C * ,IAVGH(I),IAVGL(I)
WRITE(11,106)FRE0(I),TEMPF,TEMPI,SAVEF,SAVEI
C * ,IAVGH(I),IAVGL(I)

DO 121 I=1,NF1MN,NF1MX
T = IAVG1(I)/SHIFTA + IAVGH(I)*(2**30/SHIFTA)
TDMP = T/SCALE /SCALEO
TDMP = AVGF(I)/SCALEO
SAVE = TDMP
SAVE = TDMP
IF(TDMPL.IE.1E-38)TEMPI = 1E-38
IF(TDMPL.LE.1E-38)TEMPI = 1E-38
TDMP = 10*ALOG10(TDMPL)
TDMP = 10*ALOG10(TDMPL)
WRITE(11,106)FRE0(I),TEMPF,TEMPI,SAVEF,SAVEI
C * ,IAVGH(I),IAVGL(I)
WRITE(11,106)FRE0(I),TEMPF,TEMPI,SAVEF,SAVEI
C * ,IAVGH(I),IAVGL(I)

DO 1212 I=1,NF2MN,NF2MX
T = IAVG1(I)/SHIFTA + IAVGH(I)*(2**30/SHIFTA)
TDMP = T/SCALE /SCALEO
TDMP = AVGF(I)/SCALEO
SAVE = TDMP
SAVE = TDMP
IF(TDMPL.IE.1E-38)TEMPI = 1E-38
IF(TDMPL.LE.1E-38)TEMPI = 1E-38
TDMP = 10*ALOG10(TDMPL)
TDMP = 10*ALOG10(TDMPL)
WRITE(11,106)FRE0(I),TEMPF,TEMPI,SAVEF,SAVEI
C * ,IAVGH(I),IAVGL(I)
WRITE(11,106)FRE0(I),TEMPF,TEMPI,SAVEF,SAVEI
C * ,IAVGH(I),IAVGL(I)

DO 1212 I=1,NF2MN,NF2MX
T = IAVG1(I)/SHIFTA + IAVGH(I)*(2**30/SHIFTA)
TDMP = T/SCALE /SCALEO
TDMP = AVGF(I)/SCALEO
SAVE = TDMP
SAVE = TDMP
IF(TDMPL.IE.1E-38)TEMPI = 1E-38
IF(TDMPL.LE.1E-38)TEMPI = 1E-38
TDMP = 10*ALOG10(TDMPL)
TDMP = 10*ALOG10(TDMPL)
WRITE(11,106)FRE0(I),TEMPF,TEMPI,SAVEF,SAVEI
C * ,IAVGH(I),IAVGL(I)
WRITE(11,106)FRE0(I),TEMPF,TEMPI,SAVEF,SAVEI
C * ,IAVGH(I),IAVGL(I)
DO 1222 I=N/2 +1
   T = IAVGL(I)/SCALE + IAVGH(I)*(2**38/SHIFTA)
   TEMPI = T/SCALE /SCALE0
   TEMPF = AVGF(I)/SCALEO
   SAVEI = TEMPI
   SAVEF = TEMPF
   IF(TEMPI.LT.1E-38)TEMPI=1E-38
   IF(TEMPF.LT.1E-38)TEMPF=1E-38
   AVGDI(I) = 10+AVGDI(I)
   AVGDF(I) = 10+AVGDF(I)
   123 WRITE(7,186)FREQ(I),AVGDF(I),AVGDI(I),SAVEF,SAVEI,
   ,IAVCH(I),IAVGL(I)
   1222 CONTINUE
   CL0SE(UNIT-7)
   C convert averaged power spectrum to dB and output to a file
   OPEN(UNIT=7,FILE='XPLIST.OUT',STATUS='NEW',FORM='FORMATTED')
   WRITE(7,118)JSPEC
   DO 123 =1,N/2 +1
      T = IAVGL(I)/SHIFTA + IAVGH(I)*(2**38/SHIFTA)
      TEMPI = T/SCALE /SCALE0
      TEMPF = AVGF(I)/SCALEO
      SAVEI = TEMPI
      SAVEF = TEMPF
      IF(TEMPI.LT.1E-38)TEMPI=1E-38
      IF(TEMPF.LT.1E-38)TEMPF=1E-38
      AVGDI(I) = 10+AVGDI(I)
      AVGDF(I) = 10+AVGDF(I)
      123 WRITE(7,186)FREQ(I),AVGDF(I),AVGDI(I),SAVEF,SAVEI,
      ,IAVCH(I),IAVGL(I)
   1222 CONTINUE
   CL0SE(UNIT=7)
   C clip the large negative dB values before plotting
   IF(IPLOT.EQ.1)THEN
      DO 13 =1,N/2 +1
         IF(AVGDI(I).LT.-188.)AVGDI(I)=188.
         IF(AVGDF(I).LT.-188.)AVGDF(I)=188.
      13 CONTINUE
   ELSE
      END IF
   ICTIME = 1
   CALL CPUTIME(ICTIME)
   CTIME = FLOAT(ICTIME)/1000.
   OPEN(UNIT=13,FILE='XTIME.OUT',STATUS='UNKNOWN',FORM='UNFORMATTED')
   WRITE(13,117)CTIME,SECONDS(RTIME)
   CLOSE(UNIT=13)
   JOUT = JOUT * J3
   ELSE
      END IF
   IF(NSAVE.EQ.1500)THEN
      NSAVE = 0
      IF(IFLIP.EQ.0)THEN
         FILE12 = 'XYING.OUT'
         FILE12 = 'X2.OUT'
      ELSE
         FILE12 = 'XYANG.OUT'
         FILE12 = 'X2.OUT'
      END IF
      OPEN(UNIT=14,FILE=FILE12,STATUS='UNKNOWN',FORM='UNFORMATTED')
      WRITE(14,14)JSPEC
      CLOSE(UNIT=14)
      IFLIP = NOT(IFLIP)
      OPEN(UNIT=16,FILE=FILE12,STATUS='UNKNOWN',FORM='UNFORMATTED')
      WRITE(14,14)JSPEC, J,JOUT,ISEED,ISEED1,ISEED2,SE
      & ,ADJ1,ADJ2,IFLIP
      DO 14 =1,N/2 +1
         WRITE(14,AVGDF(I),AVGDI(I),SAVEF,SAVEI,
         ,IAVCH(I),IAVGL(I)
      14 CONTINUE
      CLOSE(UNIT=14)
      ELSE
      END IF
   1234 CONTINUE
   C ...finished averaging successive power spectra
   CLOSE(UNIT=10)
STOP

100 FORMAT(
 & '1(X, A1, A2, G, S, DSEED, ISCALE, ISKIP, GAP, IDCC, ISH',
 & ), ISEED1, ISEED2')

101 FORMAT( 'run parameters:/',//,
 & ' F1 = ',F18.5, ' kHz',',F2 = ',F18.5, ' kHz',/,
 & ' FS = ',F9.4, ' kHz',',4.*',N, ' = ',14,/,,
 & ' BADC = ',12,13X,'BW = ',12,16X,'BFFT = ',14,/,,
 & ' BSPEC = ',12,12X,'BOUT = ',12,/,,
 & ' A1 = ',F15.8,2X,' A2 = ',F15.8,/,,
 & ' C = ',F15.8,3X,' S = ',F15.8,/,,
 & ' NAVG = ',111,4X,'DSEED = ',111,/,,
 & ' ISCALE = ',11,12X,'ISKIP = ',11,14X,' GAP = ',12,/,,
 & ' IDCC = ',12,14X,'ISH = ',12,/,,
 & ' ISEED1 = ',111,2X,' ISEED2 = ',111,/,,
 & ' REBOOT = ',11,/,,
 & '
 & DRAV 117/80 computer')

102 FORMAT( '/',' —> block number ',12,'...')

103 FORMAT( '/',' no. values clipped to -1 = ',14,/,,
 & '
 & no. values clipped to +1. - ',14)

104 FORMAT( '/',' window parameters:/',//,
 & '
 & coherent gain of window = ',F5.2, ' dB (float)',/,,
 & ' equivalent noise bandwidth of window = ',F5.2, ' dB (fixed)'./,
 & ' noise statistics (at output):',/,,
 & ' noise level = ',F7.2, ' dB',/,,
 & ' standard deviation = ',F7.2, ' dB',/,,
 & ' decrease in std dev = ',F7.2, ' dB',/,,
 & ' theoretical',//,
 & 'noise level = ',F7.2, ' dB',/,,
 & ' standard deviation = ',F7.2, ' dB',/,,
 & ' decrease in std dev = ',F7.2, ' dB',/,,
 & ' theoretical',//,
 & 'SNR = ',F7.2, ' dB',/,,
 & 'SNV = ',F7.2, ' dB',/,,
 & 'PNR = ',F7.2, ' dB')

105 FORMAT( '/',' using ',14, 'noise values out of ',14, 'samples')

111 FORMAT( 'CPU time - ',1PE15.8, ' seconds',/,
 & ' real time - ',1PE15.8, ' seconds')

110 FORMAT(1PE14.7)

116 FORMAT( '/',' **** WARNING ****',//,
 & 'DSEED REPEATS AT BLOCK ',110, 'SAMPLE',14,' LOOP ',12,//)

119 FORMAT(18X, 'signal ',11':',//,
 & 'SNR - ',2X,F7.2, ' dB',/,,
 & 'SNV - ',2X,F7.2, ' dB',/,,
 & 'PNR - ',2X,F7.2, ' dB')

114 FORMAT(1PE14.7)

SUBROUTINE FFTF(DATA,M,N,ISCALE)

COMPLEX DATA(N),U,W,T

PI = 3.1415927

N = 2**M

M2 = N/2

NM = N-1

J = 1

C do in-place bit-reversing shuffle on input data

DO 1 N=1,NM

T = DATA(J)

DATA(J) = DATA(I)

C**********************************************************************************

C Perform a radix-2, decimation-in-time, in-place FFT

C on complex data vector "DATA" of length N=2**M.

C (Algorithm from Cooley, Lewis, and Welch.)

C The output spectrum is scaled down by ISCALE.

SUBROUTINE FFTF(DATA,M,N,ISCALE)

COMPLEX DATA(N),U,W,T

PI = 3.1415927

N = 2**M

M2 = N/2

NM = N-1

J = 1

C do in-place bit-reversing shuffle on input data

DO 1 N=1,NM

T = DATA(J)

DATA(J) = DATA(I)
C do the FFT using recursion to update W
DO 2 L=1,M
LE = 2*L
LE1 = LE/2
U = (1.,0.)
W = CMPLX(COS(PI/LE1),-SIN(PI/LE1))
DO 2 J=1,LE1
DO 3 I=J,LE
IP = I+LE1
T = DATA(IP)*U
DATA(IP) = DATA(IP)-T
DATA(I) = DATA(I)+T
2 U = U*W
C scale the spectrum down by lSCALE
DO 4 I=1,N
DATA(I) = DATA(I)/SCAL
4 RETURN
END

C******************************************************
C Perform a radix-2, decimation-in-frequency, constant geometry FFT
C on complex data vector of length N=2**M.
C The addressing scheme is as implemented in hardware.
C The data is in two vectors (real and imaginary parts).
C each datum being a B-bit (1<8<17) 2's complement integer.
C Each butterfly is pre-scaled down by 2 using "randomized" rounding
C based on ORing of least significant bits.
C The arithmetic is quantized to B bits.
C The unit-circle coefficients are quantized to B bits.
C SUBROUTINE FFTI(IDATAR,IDATAI,IDATARS,IDATAIS,M,N,B,JSPEC)
INTEGER SCOUNT,DCOUNT,STAGE,SCADO,XADD,YADD,AADD,BADD
& ,DMASK,SCMASK
& ,AR,AI,BR,BI,TR,TI,SHIFT,OVA,OVS,T
& ,XR,XI,YR,YI
PI = 3.1415927
N = 2**M
NV2 = N/2
NM1 = N-1
C do the FFT...
SCALE = 2.***(B-1) -1.
SHIFT = 2***(B-1)
NOVAT = 0
NOVST = 0
SCCOUNT = 0
DCOUNT = 0
DMASK = N - 1  I used to make data addresses modulo-N
DO 2 L=1,M  I stage loop
NOVA = 0
NOVST = 0
DO 3 J=1,NV2  I butterfly loop
C calculate the sin/cos coefficient address
STAGE = L - 1
SCMASK = NV2 - 1  I used to generate sin/cos addresses
IF(STAGE.GT.0)THEN
DO 4 K=0,STAGE-1
4 ELSE
ENDIF
SCADD = IAND(SCCOUNT,SCMASK)
THETA = PI*SCADD/NV2
IMR = INT(-COS(THETA)*SCALE )
IMI = INT(-SIN(THETA)*SCALE )
C calculate the data read and write addresses
C (+1) adjusts for vector starting at 1 instead of 0)
XADD = IAND(DCOUNT,DMASK) + 1
YADD = IAND(DCOUNT+1,DMASK) + 1
AADD = IAND(ISHFTC(DCOUNT,-1,M),DMASK) + 1
BADD = IAND(ISHFTC(DCOUNT+1,-1,M),DMASK) + 1
C pre-scale (divide by 2 by right shifting 2's complement number;
C the result is "randomly" rounded up or down by ORing the two least significant bits of the original number to produce the least significant bit of the result.

AFT - IDATAR(AADD)
AI - IDATAI(AADD)
BR - IDATAR(BADD)
BI - IDATAI(BADD)

IF (BTEST(AR,1).AND.BTEST(AR,8)) .EQ. .FALSE. )AR=AR+1
IF (BTEST(AI,1).AND.BTEST(AI,9)) .EQ. .FALSE. )AI=AI+1
IF (BTEST(BR,1).AND.BTEST(BR,8)) .EQ. .FALSE. )BR=BR+1
IF (BTEST(BI,1).AND.BTEST(BI,9)) .EQ. .FALSE. )BI=BI+1

AR - (AR + MIN8(6,ISIGN(1,AR)))/2
AI - (AI + MIN8(9,ISIGN(1,AI)))/2
BR - (BR + MIN8(8,ISIGN(1,BR)))/2
BI - (BI + MIN8(1,ISIGN(1,BI)))/2

C do the butterfly
XR = AR + BR
NOVA = NOVA + OVA(AR,BR,XR,B)

XI = AI + BI
NOVA = NOVA + OVA(AI,BI,XI,B)

TR = BR - AR
NOVS = NOVS + OVS(BR,AR,TR,B)

TI = BI - AI
NOVS = NOVS + OVS(BI,AL,TL,B)

T = TR + MR + SHIFT/2
NUM1 = (T + MIN8(6,ISIGN(SHIFT-1,T)))/SHIFT
T = T + MR + SHIFT/2
NUM2 = (T + MIN8(9,ISIGN(SHIFT-1,T)))/SHIFT

YR = NUM1 + NUM2
NOVA = NOVA + OVA(NUM1,NUM2,YR,B)

T = TR + MR + SHIFT/2
NUM1 = (T + MIN8(6,ISIGN(SHIFT-1,T)))/SHIFT
T = T + MR + SHIFT/2
NUM2 = (T + MIN8(8,ISIGN(SHIFT-1,T)))/SHIFT

YI = NUM2 - NUM1
NOVS = NOVS + OVS(NUM2,NUM1,YI,B)

IDATARS(XADD) = XR
IDATAIS(XADD) = XI
IDATARS(YADD) = YR
IDATAIS(YADD) = YI

C update counters (sin/cos and data)
SCCOUNT = SCOUNT + 1
DCOUNT = DCOUNT + 2

C copy results of stage back to original data vector
DO 5 1=1,N
IDATAR(I) = IDATARS(I)
IDATAI(I) = IDATAIS(I)
5 1 = 1 + K
K = K/2
GOTO 29

C do in-place bit-reversing shuffle on output data
J = 1
DO 1 = 1,N
IF(I.GE.J)GOTO 10
TR = IDATAR(J)
TI = IDATAI(J)
10 IDATAR(J) = IDATAR(I)
IDATAI(J) = IDATAI(I)
11 TR = TI
12 J = J+K
13 K = K/2
GOTO 28
14 J = J-K
15 K = K/2
GOTO 29

16 IF((NOVAT+NOVST).NE.0)THEN WRITE(6,101)J,NOVAT,NOVST
17 ELSE ENDIF
18 RETURN

19 FORMAT(12,E12.12,12,E12.12)
20 FORMAT(12,E12.12,12,E12.12)
21 FORMAT(12,E12.12,12,E12.12)
22 FORMAT(12,E12.12,12,E12.12)
23 FORMAT(12,E12.12,12,E12.12)
INTEGER FUNCTION OVA(NUM1, NUM2, RESULT, B)

INTEGER NUM1, NUM2, RESULT, B, SI, S2
LOGICAL SR

SI = ISIGN(NUM1)
S2 = ISIGN(NUM2)
SR = BTEST(RESULT, B - 1)

OVA = 0
IF (SI.EQ.-1.AND. S2.EQ.-1.AND. .NOT. SR)
& OR.
& (SI.EQ.+1. AND. S2.EQ.+1. AND. SR)
OVA = 1

RETURN
END

INTEGER FUNCTION OVS(NUM1, NUM2, RESULT, B)

INTEGER NUM1, NUM2, RESULT, B, SI, S2
LOGICAL SR

SI = ISIGN(NUM1)
S2 = ISIGN(NUM2)
SR = BTEST(RESULT, B - 1)

OVS = 0
IF (SI.EQ.-1.AND. S2.EQ.+1. AND. .NOT. SR)
& OR.
& (SI.EQ.+1. AND. S2.EQ.-1. AND. SR)
OVS = 1

RETURN
END

C********************************************************************
C CHECK IF OVERFLOW HAS OCCURRED UPON ADDITION OF TWO B-BIT
C 2'S COMPLEMENT INTEGERS (RESULT=NUM1+NUM2).
C IF OVERFLOW, SET OVA = 1; OTHERWISE, SET OVA = 0.
C******************************************************************************

THIS SUBROUTINE HAS BEEN CHANGED TO SUIT VAX FORTRAN.

SUBROUTINE TO FIND CPU TIME
(Subroutine by Subroto of UBC Elec. Eng. Dept.)

SUBROUTINE CPU TIME(CTIME)
INTEGER CTIME, CODE
DATA CODE/2/

IF (CTIME.EQ.0) THEN

CALL LIB$INIT_TIMER

ELSE

CALL LIB$STAT_TIMER(CODE,CTIME)
CTIME = CTIME*10

ENDIF

RETURN
END
APPENDIX 2 - SCHEMATIC DIAGRAMS OF FFT SPECTRUM ANALYZER

The schematic diagrams of the FFT Spectrum Analyzer are contained in the following pages. The first seven schematics correspond to the major data flow blocks in Figure 3-10. The remainder show control and address circuitry. Some notes regarding their interpretation follow.

1. The figure captions include (in brackets) the board on which the circuit lies. The analog-to-digital conversion (plus window), butterfly, power spectrum, and control (plus FFT memories) boards are abbreviated by "AD Board", "BUTT Board", "PS Board", and "C&M Board", respectively.

2. Positive 5 Volts (digital) is denoted by a triangular hat. Digital ground is denoted by three horizontal lines.

3. To reduce complexity, not all individual integrated circuit packages are drawn separately. When more than one package makes up a larger module that is similar in function, they are combined into one and labelled accordingly, e.g., "2 74F244" means two 74F244 packages.

4. Each inter-board connection (data, address, and control) is paired with a ground wire (terminated at each end) for reduction of crosstalk and ringing. Also, numerous sensitive signals (e.g. edge-triggered clocks) on the boards are twisted around ground wires (terminated at each end) for the same purposes. These ground wires are not shown.

5. All data lines have the suffix "data". All address lines have the suffix "add". All other lines are control.

6. Inter-board control signal names include a suffix (AD, BUTT, or PS) denoting the destination board, e.g., "CADAD" denotes CAD sent to the AD Board. (Warning: on the C&M Board, there are some intra-board signals with these suffixes which simply denote an association.)

7. Unless otherwise denoted (by arrows), signals on the left or top of a diagram are inputs while those on the right or bottom are outputs.

Also, a list describing each signal is included at the end.
Figure A2-1. Board Interface and FFT Memories (C&M Board)
Figure A2-2. Analog-to-Digital Converter (AD Board)
Figure A2-3. Window Multiplication and Digitized Data Display (AD Board)
Figure A2-4. Butterfly Circuitry (BUTT Board)
Figure A2-5. Blowups of Butterfly Modules (BUTT Board)
Figure A2-6. Power Spectrum Computation and Accumulation Circuitry (PS Board)
Figure A2-7. Output Buffer and Interface to Microcomputer (PS Board)
Figure A2-9. Main Clock Generation (Including Some Interface Signals) (C&M Board)
Figure A2-9. PS Clock Generation (C&M Board)
Figure A2-10. AD and BUTT Control Signals, and Some Control Signals for Buses (C&M Board)
Figure A2-11. Some Control Signals for Buses (C&M Board)
Figure A2-12. PS Control Signals (C&M Board)
Figure A2-13. Write and Enable Signals for FFT Memories 1 and 2 (C&M Board)
Figure A2-14. Write and Enable Signals for FFT Memory 3 and E5 (C&M Board)
Figure A2-15. AD and Window Address Generation (C&M Board)
Figure A2-16. FFT Data Address Generation (C&M Board)
Figure A2-17. Bit-8 Generation for FFT Data Address (C&M Board)
Figure A2-18. FFT Coefficient Address Generation (C&M Board)
Figure A2-19. PS Address Generation (C&M Board)
Figure A2-20. Address Multiplexers (C&M Board).
Figure A2-21. Sample and Stage Counters (C&M Board)
Figure A2-22. Integration (FFT) Counter (Including Some Interface Signals) (C&M Board)
Figure A2-23. Reset Generation (Including Some Interface Signals) (C&M Board)
Figure A2-24. Overflow Indicators (AD Board)
Description of Signals in FFT Spectrum Analyzer

<table>
<thead>
<tr>
<th>Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1ARDY</td>
<td>microcomputer port ready line</td>
</tr>
<tr>
<td>2ARDY</td>
<td>microcomputer port ready line</td>
</tr>
<tr>
<td>2BRDY</td>
<td>microcomputer port ready line</td>
</tr>
<tr>
<td>A0-1</td>
<td>OUTPUT byte address</td>
</tr>
<tr>
<td>ADadd0-7</td>
<td>A-D data address</td>
</tr>
<tr>
<td>ADdata0-7</td>
<td>A-D data</td>
</tr>
<tr>
<td>ANALOG IN</td>
<td>analog input</td>
</tr>
<tr>
<td>C</td>
<td>main clock</td>
</tr>
<tr>
<td>C1BUTT</td>
<td>clock butterfly latches 1 and 4</td>
</tr>
<tr>
<td>C1PS</td>
<td>clock PS latch</td>
</tr>
<tr>
<td>C2BUTT</td>
<td>clock butterfly latches 2 and 3</td>
</tr>
<tr>
<td>C2PS</td>
<td>clock PS resettable latch</td>
</tr>
<tr>
<td>C3BUTT</td>
<td>clock butterfly tri-state latches 1 and 4</td>
</tr>
<tr>
<td>C4BUTT</td>
<td>clock butterfly tri-state latches 2 and 3</td>
</tr>
<tr>
<td>CAD</td>
<td>clock A-D</td>
</tr>
<tr>
<td>CADAD</td>
<td>clock A-D (buffered)</td>
</tr>
<tr>
<td>CAD(w/o delay)</td>
<td>clock A-D (w/o delay)</td>
</tr>
<tr>
<td>CCadd</td>
<td>clock coefficient address</td>
</tr>
<tr>
<td>C+4</td>
<td>main clock divided-by-4</td>
</tr>
<tr>
<td>CMBUTT</td>
<td>clock butterfly multipliers</td>
</tr>
<tr>
<td>CMPS</td>
<td>clock PS multiplier</td>
</tr>
<tr>
<td>CPS</td>
<td>clock PS control generator</td>
</tr>
<tr>
<td>CPSadd</td>
<td>clock PS address</td>
</tr>
<tr>
<td>CRadd</td>
<td>clock FFT data read address</td>
</tr>
<tr>
<td>CRWadd</td>
<td>clock read/write FFT data address</td>
</tr>
<tr>
<td>CSdata0-15</td>
<td>complex spectrum data</td>
</tr>
<tr>
<td>CWadd</td>
<td>clock FFT data write address</td>
</tr>
<tr>
<td>E1</td>
<td>enable output for FFT memory swapping</td>
</tr>
<tr>
<td>E1BUTT</td>
<td>enable output butterfly tri-state latches 1 and 4</td>
</tr>
<tr>
<td>E1PS</td>
<td>enable output PS latch</td>
</tr>
<tr>
<td>E2</td>
<td>enable output for FFT memory swapping</td>
</tr>
<tr>
<td>E2BUTT</td>
<td>enable output butterfly tri-state latches 2 and 3</td>
</tr>
<tr>
<td>E3</td>
<td>enable output for FFT memory swapping</td>
</tr>
<tr>
<td>E4</td>
<td>enable output for FFT memory swapping</td>
</tr>
<tr>
<td>E5</td>
<td>enable output for copy to FFT memory bank 3</td>
</tr>
<tr>
<td>E6</td>
<td>enable output for reading complex spectrum</td>
</tr>
<tr>
<td>E7</td>
<td>enable output for reading complex spectrum</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>EAD</td>
<td>enable A-D data</td>
</tr>
<tr>
<td>EBUSBUTT</td>
<td>enable output butterfly buffers 1 and 2</td>
</tr>
<tr>
<td>EMUX</td>
<td>enable OUTPUT bus</td>
</tr>
<tr>
<td>EOBLOCK+2</td>
<td>end-of-block pulse divided-by-2</td>
</tr>
<tr>
<td>EFFT</td>
<td>end-of-FFT pulse</td>
</tr>
<tr>
<td>EOINTEG</td>
<td>end-of-integration pulse</td>
</tr>
<tr>
<td>EOINTEGRPS</td>
<td>end-of-integration pulse (buffered)</td>
</tr>
<tr>
<td>EOSTAGE</td>
<td>end-of-stage pulse</td>
</tr>
<tr>
<td>EXTCLK</td>
<td>external clock</td>
</tr>
<tr>
<td>FFTCOEFFadd0-6</td>
<td>FFT coefficient address</td>
</tr>
<tr>
<td>FFTDATAadd0-7</td>
<td>FFT data address</td>
</tr>
<tr>
<td>MEM1add0-7,8</td>
<td>FFT memory bank 1 address</td>
</tr>
<tr>
<td>MEM2add0-7,8</td>
<td>FFT memory bank 2 address</td>
</tr>
<tr>
<td>MEM3add0-7</td>
<td>FFT memory bank 3 address</td>
</tr>
<tr>
<td>MODE0-1</td>
<td>clock mode</td>
</tr>
<tr>
<td>MRFS</td>
<td>clear PS accumulator</td>
</tr>
<tr>
<td>NAVGO-7</td>
<td>number of averages per integration period</td>
</tr>
<tr>
<td>OE1</td>
<td>output enable for memory swapping</td>
</tr>
<tr>
<td>OE1MEM</td>
<td>output enable FFT memory bank 1</td>
</tr>
<tr>
<td>OE1PS</td>
<td>output enable PS memory bank</td>
</tr>
<tr>
<td>OE2</td>
<td>output enable for memory swapping</td>
</tr>
<tr>
<td>OE2MEM</td>
<td>output enable FFT memory bank 2</td>
</tr>
<tr>
<td>OE3</td>
<td>output enable FFT memory bank 3 (unsynchronized)</td>
</tr>
<tr>
<td>OE3MEM</td>
<td>output enable FFT memory bank 3</td>
</tr>
<tr>
<td>OEBUTT</td>
<td>output enable butterfly data</td>
</tr>
<tr>
<td>OVERFLOW</td>
<td>A-D overflow</td>
</tr>
<tr>
<td>PSACCUMadd0-6</td>
<td>power spectrum accumulation address</td>
</tr>
<tr>
<td>PSadd0-7</td>
<td>power spectrum address</td>
</tr>
<tr>
<td>PSdata0-31</td>
<td>power spectrum data</td>
</tr>
<tr>
<td>RESET</td>
<td>reset pulse</td>
</tr>
<tr>
<td>RESETPS</td>
<td>reset PS counter</td>
</tr>
<tr>
<td>S2</td>
<td>select PS address</td>
</tr>
<tr>
<td>SRWadd</td>
<td>select read/write FFT data address</td>
</tr>
<tr>
<td>STAGE0-2</td>
<td>stage count</td>
</tr>
<tr>
<td>SYSRESET</td>
<td>system reset pulse</td>
</tr>
<tr>
<td>T/R</td>
<td>transmit/receive (FFT data direction)</td>
</tr>
</tbody>
</table>
WE2I write-enable FFT memory bank 1 (imag)
WE1PS write-enable PS memory bank
WE1R write-enable FFT memory bank 1 (real)
WE2I write-enable FFT memory bank 2 (imag)
WE2R write-enable FFT memory bank 2 (real)
WE3 write-enable FFT memory bank 3
WEAD write-enable A-D data
WEBUTT write-enable butterfly data
WINDOWadd0-7 window address
WINDOWEDdata0-15 windowed data
Z0 start-of-integration pulse
The split-window normalizer [DREA84] is used in the interference monitor to whiten the background noise before threshold detection. It is applicable for the case of narrowband signals in broadband, colored noise. It is essentially a method of lowpass filtering the power spectrum data to estimate the shape of the background noise and then dividing (normalizing) the original data by the noise estimate.

A double-boxcar function (depicted below) is passed over (convolved with) the data to yield an initial estimate. (The noise estimate at the center of the gap is calculated by averaging the values of the data points that fall inside the windows.)

\[
\begin{align*}
N1 &= \text{window width} \\
N2 &= \text{gap width} \\
A &= \text{clipping constant} \\
B &= \text{replacement constant} \\
NREFIT &= \text{number of passes after first two passes}
\end{align*}
\]

Then, those points in the original data that lie above the estimate are clipped (set equal to the estimate) and points below the estimate are replaced with the estimate. If desired, the clipping and replacement levels can be modified by scale factors A and B. The double-boxcar is then passed over this new dataset to yield the final estimate.
Figure A3-1. Double-Boxcar Function in Split-Window Normalizer

The split-window normalizer was refined by the author to yield better noise estimates. In the refined normalizer, the "clip, replace, and re-estimate" process is repeated NREFIT times, where the clipping and replacement process uses the original data. This yields a better estimate, especially when a large signal is present.