## N OTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE


## FINAL REPORT

## ADVANCED DIGITAL SAR PROCESSING STUDY

## BY

LLOYD W. MARTINSON<br>BRIAN P. GAFFNEY<br>BEDE LIU<br>RICHARD P. PERRY<br>ABRAHAM RUVIN

This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronsutics and Space Administration under Contract NAS7-100.

RCA|Government Systems Division Missile and Surface Radar Moorestown, NJ 08057
FINAL REPORT
ADVANCED DIGITAL GAR PROCESSOR STUDY
BY
LLOYD W. MARTINSON
BRIAN P. GAFFNEY
BEDE LIU
RICHARO P. PERRY
ABRAHAM RUVIN
This work was performed for the Jet Propulsion Laboratory,California Instit'te of Technology, sponsored by theNational Aeronautics and Space Administration underContract NAS7-100.
RCA/Government Systems Division Missile and Surface Radar Moorestown, N.J. 08057

A design of a highly programmable, land based, real time synthetic aperture radar (SAR) processor requiring a processed pixel rate of 2.75 MHz or more in a four-look system was conducted. Variations in range and azimuth compression, number of looks, range swath, range migration and SAR mode were specified. A number of alternative range and azimuth processing algorithms were examined and analyzed in conjunction with projected integrated circuit, digital architecture, and software technologies.

The selected design for the Advanced Digital SAR Processor (ADSP) employs an FFT convolver algorithm for both range and azimuth processing in a parallel architecture configuration. This overall design approach met all of the system implementation and performance criteria for programmability, modularity, adaptability to VLSI, low risk, reliability and cost.

The report provides algorithm performance comparisons, a detail design of the selected system, implementation tradeoffs and the results of a supporting survey of integrated circuit and digital architecture tfchnologies. Cost iradeoffs and projections with alternate implementation plans are presented.

## TABLE OF CONTENTS

SECTION TITLC PAGE
1.0 INTRODUCTION ..... 1
2.0 TECHNICAL DISCUSSION ..... 3
2.1ADSP SYSTEM REQUIREMENTS3
2.1.1 Functional Requirements ..... 4
2.1.2 ADSP System Scope ..... 6
2.1 .3 Analysis of ADSP Requirements ..... 7
2.1.3.1 SAR Parameter Variation ..... 7
2.1.3:2 Mode Variations ..... 8
2.1.3.3 Range Cell Migration ..... 11
2.1.3.4 Clutter Lock Requirements ..... 12
2.1.3.5 Automatic Focusing Requirement ..... 12
2.2 PROCESSING ALGORITHMS ..... 15
2.2.1 Basic SAR Processing Functions ..... 15
2.2.2 Time Domain Processing ..... 18
2.2.2.1 General Time Domain Processing Consideration ..... 18
2.2.2.2 Serial Time Domain Processor (Type A) ..... 20
2.2.2.3 Parallel Time Domain Processor ..... 24
2.2.2.4 Type A, B Comparison ..... 26
2.2.2.5 Prefilter and Multilook Integrator ..... 27
2.2.2.6 Time Domain Hardware Summary ..... 30
2.2.3 FFT Convolver ..... 31
2.2 .4 Subarray Processing ..... 34
2.2 .5 Two-dimensional Convolution ..... 39
2.2.5.1 Full Range Azimuth Correlation 2-D Process ..... 39
2.2.5.2 2-D Convolution With Range Compressed Data ..... 40
2.2.5.3 Hyturid 2-D Process ..... 45
2.2.6 Range Correlation ..... 46
2.2.6.1 Range FFT Convolver ..... 46
2.2.6.2 Step Transform Linear Frequency Modulated (LFM) Signal Matched Filter ..... 49
2.2.6.3 Oigital Tapped Delay Line Correlator ..... 51
SECTION TITLE ..... PAGE
2.2.7Algorithm Study Summary52
2.3 Performance Levels ..... 55
2.3.1 Performance Procedure ..... 55
2.3.1.1 Generation of the Time Signal ..... 55
2.3.2 FFT - Convolver Algorithm ..... 57
2.3.2.1 Range Migration Compersation ..... 57
2.3.2.2 Azimuth Matched Filter ..... 59
6.3.2.3 Weighting ..... 59
2.3.2.4 Time Domain Output ..... 61
2.3.2.5 Interpolation and Weighting Simulation Results ..... 61
2.3.3 Step Transform - Subarray Approach ..... 64
2.3.3.1 Description of Subarray Simulation ..... 64
2.3.3.2 Subarray Simulation Results ..... 68
2.3 .4 Performance Comparisons ..... 70
2.4 Technology Survey ..... 72
2.4.1 Digital Integrated Circuits ..... 72
2.4.1.1 Survey of Integrated Circuit (IC) Manufacturers ..... 72
2.4.1.2 IC Technology Trends ..... 73
2.4.1.3 Random Access Memories ..... 74
2.4.1.4 EPROMS, PROMS, ROMS ..... 76
2.4.1.5 IC:-y.c Functions ..... 77
2.4.1.6 VHSIC ..... 78
2.4.2 Digital Architecture ..... 80
2.4.2.1 Pipeline Processor ..... 81
2.4.2.2 Parallel Processor ..... 82
2.4.2.3 Single Instruction - Multiple Data (SIMD) ..... 83
2.4.2.4 Cross-bar ..... 84
2.4.2.5 Multi-bus Architecture ..... 85
2.4.2.6 Programmable Signal Processor Developments ..... 86
2.4.3 Software ..... 87

TABLE OF CONTENTS (continued)
SECTION TITLE ..... PAGE
2.5 ADSP SELECTED DESIGN ..... 92
2.5.1 Selection of Algorithm ..... 92
2.5.1.1 Key ADSP Requirements ..... 92
2.5.1.2 Programmability ..... 92
2.5.1.3 Incremental Implementation and Growth ..... 93
2.5.1.4 Burst Multimode Processing ..... 94
2.5.1.5 Algorithm Cost Comparisons ..... 97
2.5.1.6 Risks ..... 97
2.5.1.7 Summary and Selection ..... 98
2.5.2 Selection of Architecture ..... 99
2.5.3 Design ..... 101
2.5.3.1 Range Correlator Design ..... 101
2.5.3.2 Linear Range Migration Correction ..... 104
2.5.3.3 Corner Turning Memory ..... 104
2.5.3.4 Azimuth Correlation ..... 106
2.5.3.5 Multilook Integrator ..... 107
2.5.3.6 Clutter Lock ..... 108
2.5.3.7 Generation of Focus function ..... 110
2.5.3.8 Control System ..... 111
2.5.3.9 Test Subsystem ..... 113
2.5.3.10 Physical Configuration ..... 113
2.5 .4 Impact of Technology ..... 115
2.6 ADSP SCHEDULE AND COST FACTORS ..... 117
2.7 IMPLEMENTATION PLAN ALTERNATIVES ..... 119
3.0 CONCLUSIONS/RECOMMENDATIONS ..... 122
4.0 NEW TECHNIQUES ..... 124
LIST OF REFERENCES ..... 125
APPENDIX PART I - SAR Signal Generation ..... Al
PA. - FFT Convolver Azimuth Correlator ..... A23
PART III - Subarray Azimuth Processor ..... A36

## LIST OF ILLUSTRATIONS

FIGURE TITLE1 Satellite SAR Parameters3
2 ADSP Elements ..... 6
3
Continuous Mode Integration Requirements ..... 9
4
Interleaved Polarization Burst Mode ..... 1056789101112
Interleaved Bursts at Two Frequencies ..... 11
Focused SAR Image Processing Functions ..... 15
Satellite Radar Swath ..... 16
Azimuth Resolution Element Motion Relative Fixed Antenna ..... 17
Multi-Look Time Domain Processor Functions ..... 18
Serial Time Domain Azimuth Processor ..... 19
Parallel Time Domain Azimuth Processor (Type B) ..... 20
Detail of Type A Processor ..... 21
13 Multiplexed (4X) Type A Correlator Channels ..... 23
14
Detail of Type B Processor ..... 25
15
16
$17 a$
17bMulti-Look Filtering Functions28
Low Pass Filter (32nd Order) Decimate by 4 ..... 29
Multi-Look integrator ..... 30
FFT Convolution-Azimuth Processing ..... 32
18 Unique Relationship Between Range Migration and Frequency ..... 3319202122232425262728
Block Diagram of Subarray Azimuth Processor ..... 35
Continuous Deramping and Subarray Processing ..... 36
Deramp Coarse Resolution ..... 36
Bulk Storage in Subarray Process ..... 38
Two-Dimensional Convolution Using FFT Processing ..... 39
Polynomial Transform Operations. ..... 41
First Stage of Polynomial Transforin ..... 42
Merging of Two MxM/2 Arrays ..... 44
Hybrid SAR Processor ..... 46
FFT Matched Filter System ..... 47

## LIST OF ILLUSTRATIONS (continued)

FIGURE TITLE PAGE
29
FFT Convolution-Range Processing ..... 48
30 Step Transform LFM Pulse Compression Algorithm ..... 50
31 Step Transform LFM Range Pulse Compression Processing ..... 50
32 Time Domain Range Convolver Controls ..... 52
33 Baseline Azimuth Paraneters ..... 53
34 Quadratic and Linear Range Migration ..... 56
35 Generation of the Time Signal ..... 57
36 Range Migration ..... 58
37 Sidelobe Definitions ..... 60
38 Subarray Formation ..... 64
39 First FFT in Subarray Approach ..... 65
40 Subarray Range Migration Correction ..... 66
41 Coefficient Selection ..... 68
42
Memory Chip Capacity Trends ..... 75
43
Memory Costs/Bit Trends ..... 76
44a Comparison of New Technologies on Clock Rate - Gate Basis ..... 79
44b Example ADSP Pipeline Processor (FFT Convolver) ..... 82
45
Example ADSP Paral, 1 Processor (FFT Convc lver ..... 83
46
SIMO (Singie Instruction - Multiple Data Stream) ..... 84
47 Cross-Bar Architecture ..... 85
48 Multi-Bus Architecture ..... 85
49
Burst Processing Delays ..... 95
50 Interrupted Burst Processing with Many Looks ..... 96
51 Advanced Digital SAR Processor Design ..... 102
52 Range Correlator ..... 103
53 Corner Turning Memory Function ..... 105


Corner Turning Memory Organization ..... 105
55
Range Migration Correction and Matched Filter ..... 106
56 Multi-Look Integrator ..... 107
57 FFT Convolver Control ..... 113
58 Test System Concept ..... 114

## LIST OF ILLUSTRATIONS (continued)

FIGURE TITLE ..... PAGE
59 IC Technology Projection ..... 116
60 Signal Processor Architecture Projection ..... 116
61 Effect of Schedule on Cost ..... 117
62 Typical Expenditure Rates ..... 118
63 Recommended Schedule ..... 119
TABLE TITLE PAGE
1 SAR Mission Set ..... 4
2 ADSP Baseline Functional Requirerents ..... 5
3 Type-A Circuits -- Interconnections ..... 23
4 Type-B Circuits .- Interconnections ..... 26
5 Type-A, B Comparisons ..... 27
6 Time Domain Module Summaries ..... 31
7 Omitted
8 Range FFT Convolver Alternatives ..... 49
9 Look-Parameter Azimuth Processor Variations for Baseline SAR System ..... 54
10 Interpolation formulas ..... 59
11 Ideal Characteristics of Weighting Functions ..... 61
12
FFT Convolver interpolator Results ..... 62
13
Effect of Weighting Functions, FFT Convolver ..... 63
14
Subarray Over lip ..... 69
15
Effect of Weighting of First FFT in Subarray Process ..... 70
16 Linear Range Migration (24 cells) ..... 71
17
Linear Range Migration ( 40 cells) ..... 71
18 Random Accesss Memories ..... 74
19
EPROMS, PROMS, ROMS ..... 77
20 IC Logic Functions ..... 78
21 ADSP Functional Characteristics ..... 80
22 Key Characteristics of Candidate HOLs ..... 91
23 Key ADSP Requirements ..... 92
24
Algorithm Programmability ..... 93
25 Incremental Growth and Implementation ..... 94
26 Relative Algorithm Costs ..... 97
27 ADSP Algorithm Development Risks ..... 98
28 Algorithm Summary Comparison ..... 99
29 Pipeline - Parallel Architecture Comparison ..... 100
30 Processor Comparisons (FFT Convolver Algorithm -- FFTs) ..... 100

## LIST OF TABLES (continued)

TABLE TITLE ..... PAGE
31 Clutter Lock Simulation Results ..... 109
32 AOSP System Control Requirements ..... 111
33 FFT Range Correlator Convolver Variations ..... 112
34 Module Summary ..... 115
35 Alternative Träde.off ..... 120

### 1.0 INTRODUCTION

A general practice in synthetic aperture radar (SAR) systems has been to use a specially tailored processor to convert the raw radar signals to imagery. The advent of practical digital signal processing techniques, brought about by LSI technology, has made high performance, multi-mission programmable SAR digital signal processors feasible.
This is the final report of a study to quantify various issues related to the hardware development of such a SAR processor. The generic name for the processor is the Advanced Digital SAR Processor and it shall be referreit to from this point on in the report as the ADSP.

A very br, ad set of performance goals have been set by the Jet Propulsion Laboratory (JPL) for the ADSP. These are discussed in Section 2.1 together with the system performance requirements.

Two principle design facets are examined, processing algorithms and technology. The algorithms are further divided into time and frequency domain techniques; while the broad issue of technolagy includes digital integrated circuits, digital architecture and software. Section 2.2 gives a comprehensive description of the candidate processing algorithms together with hardware implementation approaches for them.

A key factor in the selection of a processing algorithm is the quality of imagery it produces. Extensive computer simulations have been developed and run to evaluate the performance of candidate algorithms. Thi: data, given in Section 1.3, was then applied as appropriate to the hardware sizing of the various implementations.

Technology is addressed in Section 2.4 with the current state of the art and projections provided for digital integrated circuits, architecture and software.

A design recommendation for the AUSP is synthesized and described in Section 2.5. It meets all of the performance and implementation goals of the ADSP.

Key issues in the selection of an algorithm for the ADSP were its effectiveness in compensating for range migration, progammability for multiple modes, performance level and the inherent computation
r.aisirements and memory storage. The selection of an algorithm was closely tied into the technology issues, particularly digital integrated circuits and architecture. The design selected minimizes memory storage and offers simple memory management techniques. It provides a programmable architecture in modular form which offers both incremental development and growth potential with little waste of effort. Finally, the design has inherently high reliability and maintainability features.

Cost factors, aiternative development plans and tradeoffs are presented in jection 2.6 and 2.7.

Software developed under the program is provided in the appendix.

### 2.1 ADSP SYSTEM REQUIREMENTS

A satellite SAR system is depicted in Figure 1. As the radar moves along its flight path, multiple radar pulses are coherently processed to achieve a resolution corresponding to the length of a synthetic antenna. This length is determined by the footprint of the real antenna on the surface. The 1 imiting resolution of the SAR is thus determined by the size of the real anterina and is equal to one half its diameter. To achieve radar returns which depict the contours of the surface, an oblique incidence angle is employed. In addition, a forward look angle off broadside is commen. Resolution in the cross track, or range, dimension is achieved by using conventional radar pulse compression techniques. The total range swath covered is also constrained to the dimensions of the radar footprint.


Figure 1.
SATELLITE SAR PARAMETERS

### 2.1.1 Functional Requirements

The ADSP requirements have been established in anticipation of processing a wide variety of SAR data. SAR misjions which represent a comprehensive set for design parameter purposes are given in Table 1. Both satellite and aircraft SAR are represented.

Tatle 1
SAR Mission Set

| MISSION SET | RESOLITIIN | $\begin{aligned} & \text { SWAD } \\ & \text { WHEH } \\ & (k m \end{aligned}$ | $\begin{aligned} & \text { NUMBER } \\ & \text { OF LOOKS } \end{aligned}$ | $\begin{aligned} & \text { INCIDENT } \\ & \text { ANGLE } \end{aligned}$ | Frequency | $\begin{aligned} & \text { POLARI-: } \\ & \text { ZATION- } \end{aligned}$ | $\underset{\substack{\text { ALIITMOE } \\(\mathrm{mm}}}{ }$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| SEASAT | 25 | 100 | 4 | 22.54 | 1275 | S | 800 |
| SIR REFLIGHT | 40 | 50 | 1 | 20-70 | 1275 | S | 210-280 |
| $\begin{gathered} \text { AIRCRAFT } \\ \vdots \\ x \end{gathered}$ | $10-20$ $10-20$ | $\begin{aligned} & 18-35 \\ & 18-35 \end{aligned}$ | 8 | $0-60$ 0.60 | 1275 9600 | $\begin{aligned} & \mathrm{S} \\ & \mathrm{~S} \end{aligned}$ | $\begin{aligned} & 3-12 \\ & 3-12 \end{aligned}$ |
| $\begin{aligned} & \text { VOIR } \\ & \text { HI-RES } 52^{0} \\ & \text { LO-RES } 52^{0} \end{aligned}$ | 75 300 | $\begin{array}{r} \text { TBD } \\ 30 \end{array}$ | 4 30 | 52 92 | 1275 1275 | 5 | $\begin{aligned} & 250 \\ & 250 \end{aligned}$ |
| ICEX | 150 | 360 | 6 | 36.42 | 9600 | S | 100 |
| ERSAR L-0 | 15 | 40 | 4 | $60^{0}$ | 1275 | 0 | 300 |
| ERSAR X 00 | 15 | 40 | 4 | $60^{0}$ | 9600 | D | 300 |
| - S - SINGLE, D - DUAL <br> -• PRELIMINARY |  |  |  |  |  |  |  |

The most stressing signal processing requirement of the missions in Table 1 is the ERSAR (Earth Resources SAR) which has been selected by JPL for the baseline design mission for the ADSP. Functional requirements for the baseline mission are given in Table 2.

Both range and azimuti processing requirements are included. Specifications are also prosided for range migration correction, reference function update; integrated sidelobe level and output data format.

Table 2
ADSP Baseline Functional Requirements

| RANGE CORRELATOR |  |
| :---: | :---: |
| INPUT SAMPLE RATE ICOMPLEX WORDS) | 11 MHz |
| Range Compression ratio | 410 |
| RANGE PULSE LENGTH | 475 (COMPLEX SAMPLES) |
| range integrated sidelobe ratio due to processing | -15dB |
| AZIMUTH CORRELATOR |  |
| SWATH WIOTH IOUTPUT PIXELSI | 4,000 |
| RADAR PRF | 2,500Hz |
| NUMBER OF AZIMUTH LOOKS ICONCURTEİNITY PROCESSEDI | 4 |
| AZIMUTK BANDWIDTH PER LOOK | 500Hz |
| AZIMUTH COMPRESSION RATIO PER LOOK | 350 |
| RANGE WALK COMPENSATION ACROSS 1 LOOK | 28 SLANT RANGE BINS |
| RANGE WALK COMPENSATION ACROSS 4 LOOKS | 86 SLANT RANGE BINS |
| RANGE CURVATURE COMPENSATION ACROSS ALOOKS | 7 SLANT RANGE BINS |
| REFERENCE FUNCTION UPDATING RATE | CROSS-TRACK EVERY 8 SAMPIES: ALONG TRACK EVERY 12,000 RADAR PULSES |
| azimuth iniegrated sidelobe ratio due IO PROCESSING | -20dB |
| OUTPUT PIXEL FORMAT (BOTH OPTIONS A ÁND B ARE REOUIRED) | A) SELECTABLE 12 BITS IN AMP LITUDE OVER 80dB DYNAMIC RANGE |
|  | $\begin{aligned} & \text { 8) COMPLEX SINGIE LOOK } \\ & \text { PIXELS' } 12 \text { BITS I. } \\ & \text { 12 BIIS } \end{aligned}$ |

Changes in operating mode are anticipated to occur within a SAR mission and these changes could involve, altitude, look angle, resolution, number of looks, swath width and the range and/or azimuth weighting functions. The processor is required to operate in a continuous or burst mode. The former has the sensor emitting non-changing parameters. The latter interleaves alternate bursts with different parameters, yet permits contiguous imagery formation. Alternate bursts may involve different radar frequencies, different polarizations or different antenna pointing angles.

Uesign objectives of the ADSP include programmable features to permit range compression from 10 to 600 , azimuth compression from 20 to 350 per look and swath width from 500 to 4000 pixels. A desired feature provides the capability to tradeoff looks, resolution and swath width within the capabilities of the processor.

Modularity in design was a key functional design objective. This feature could provide greater swath width (up to four times the baseline), more parallel looks and multiple frequencies and polarizations.

### 2.1.2 ADSP System Scope

The principal elements of the ADSP as shown in Figure 2 are an input data handling facility to permit either real time or recorded data to be processed, a range correlator, an azimuth correlator and a multiloo! integrator. All of the high speed processing hardware is encompassed in the pipeline from data input to output. The host computer is used to set up conditions for mission operation. These will include SAR parameters and overall control commands. The hardware itself will respond to the program commands and under firmware


Figure 2
ADSP Elements
control will complete the required operations. A goal of the system design was to maintain software simplicity for the host computer by incorporating as much as possible of the SAR variables in the hardware.

### 2.1.3 Analysis of ADSP Requirements

### 2.1.3.1 SAR Parameter Variation

The azimuth resolution for operation near broadside is given by:

$$
\delta_{A Z}=\frac{K_{a} \lambda^{R_{s}}}{2 V_{s} T_{L}}
$$

where $R_{s}=$ slant range, $\lambda=$ wavelength, $V_{S}=$ satellite velocity and $T_{L}=$ coherent look time. The azimuth beam width factor $K_{a}$, is associated with the receive weighting required to achieve a desired sidelobe level. For -36 dB sidelobe level, $K_{a}=1.2$ for a Taylor $\bar{n}=8$ weighting function. This weighting produces an illegrated sidelobe level, ISL, of -25 dB . When other sources of sidelobes are included such as digital granularity and the lack of interpoiation in range cell migration compensation, an overall ISL of -20 dB is obtained. The chirp bandwidth associated with the coherent look time is given by: given by:

$$
B_{L}=\frac{2 V_{s}^{2} T_{L}}{\lambda R_{s}}=\frac{K_{a} V_{s}}{\delta_{A Z}}
$$

For the ERSAR case $V_{s}=7.73 \mathrm{~km} / \mathrm{sec}, \delta_{A Z}=15 \mathrm{~m}, K_{a}=1.2$ and $B_{L}=618.4 \mathrm{~Hz}$ for each look. At $\lambda=0.235 \mathrm{~m}, T_{L}=0.686 \mathrm{sec} .$, $B_{L} T_{L}=424$ and the angle compression during the look angle $=$ $424 / 1.2=353$.

For four looks the total processed bandwidth $=4 B_{L}=2473.6 \mathrm{~Hz}$. The azimuth beamwidth that just matches this spectrum is given by $\theta_{B}=4 V_{S} T_{L} / R_{S}=0.0375\left(2.15^{\circ}\right)$. The antenna length required for this beamwidth is $L_{A}=.886 \lambda / \theta_{B}=5.55$ meters, assuming uniform illumination. The PRF of the ERSAR $=2500$. The ratio of the PRF to the four-look bandwidth $=1.01$. This ratio is lower than it should be for low doppler aliasing (angle grating lobes). A calculation shows that the integrated grating lobes from the four looks is only -8.2 dB . A higher ratio of PRF to four-look bandwidth would be required for reduced grating lobes; however, the range grating lobes would then be
required for reduced grating lobes; however, the range grating lobes would then be excessive unless the swath were reduced by the same rat to that the PRF was increased. The sampled range to allow for a 40 km swath at $60^{\circ}$ incidence and a $35.2 \mu \mathrm{sec}$ transmitted pulse requires $266.1 \mu \mathrm{sec}$. The ratio of the $2500 \operatorname{PRF}$ period to $266.1 \mu \mathrm{sec}=1.503$ is a reasonable value to inhibit range grating lobes adequately. A lower ratio of periods caused by a higher PRF would cause the range grating lobes to appear.

SAR mission sets given in Table 1 can be evaluated as a function of the r umber of looks, swath width and required resolution for the optimum PRF and antenna parameters. For the case of dual frequency such as ERSAR-L and ERSAR-X, the real antenna lengths, PRFs, and look bandwidths would be identical. However, the antenna widths and angle compression should be scaled by the wavelength. Dual polarization reception could be simultaneous, if desired. However, dual polarization transmission would have to be orthogonal either by frequency or time. The possibility for change in look angle depends upon an adequate ratio of PRF period to A/D sampling time for the desired swath. Electronic scanning or beam switching can be employed to change the look angle if the ratio is less than 1.5 The requirement for a "burst mode" involving interleaving different frequencies, polarization, and pointing angle may alsu require azimuthal electronic scan. This is discussed in Scction 2.1.3.2. Of course, several PRFs may be required to center the swath window between the ambiguous transmission pulses to avoid eclipsing if the look angle or orbit attitude is changed.

### 2.1.3.2 Mode Variations

The ADSP is required to have the flexibility to process data from a wide variety of different. missions described in Table 1 and to accommodate continuous or burst modes.

Figure 3 shows the case for four looks in the continuous mode integration for azimuth cell $j$, where Look 4 coherently integrates the first $N_{A} / 4$ samples and Look 1 cuherently integrates the last $N_{A} / 4$ samples over the azimuth beam. The four looks are noncoherently integrated and
registered at the correct range and azimuth cell. As each new azimuth sample is received, a beam is completed for each look and one final integrated multilook azimuth image sample is completed.

ORIGINAL PAGE 5s OF POOR QUALITY



Figure 3
Continuous Mode Integration Requirements

In the burst mode, a number of variations are possible. When bursts are interleaved and do not involve frequency chancles, the integration patterns shown in Figure 4 are obtained. In the case shown in Figure 4a, two polarizations are interleaved by bursts in a four-look mode. The result is that the output imagery is available on a two-look two-polarization basis. A polarization comparison of image samples can then be made. The processing can be identical in form to the continuous case except that inefficiencies occur in that not all of the antenna coverage is integrated and used in the final imagery. This is because the field must be illuminated for a full look before the data can be processed to obtain a full integrated signal-to-noise ratio with the desired resolution.

A large number of looks can be transmitted on an interrupted basis as shown in Figure 4b. In this case, 32 looks are processed. Look integration procedures may be varied as a function of time, if the burst sequences are short. In the limiting case, each look can be an unfocussed segment of the entire beam. The beam is then split in angle into the number of "looks" processed. The Multiple burst looks must be placed in proper range registration prior to integration.

The conditions change if bursts of different frequencies are transmitted. If we assume that the antenna size remains the same, the high frequency has a proportionally narrower bean. Thus, if full coverage is desired, multiple bursts at different pointing angles must be transmitted. This is illustrated in Figure 5. For simplicity, the example of high frequency is four times the lower frequenry and four looks are employed. The first burst at frequency $F_{1}$ provides the four-look coverage indicated. Quadrupling the frequency narrows the antenna beam proportionally and also increases the ground resolution by the same factor. To obtain the same final image resolution, only one fourth as many pulses need be transmitted per burst. But since the beam is narrower, it must be steered to a total of four angles with a separate burst transmitted at each angle. Four looks are processed at the high frequency also to maintain the final image resolution. A strip map can be generated in this way which has one look at each frequency for each image location.



EDPS !nvalio ou partial output tiageay

FIGURE 4. Coneiguous, fro-polariaction Bursts
".



Figure 4B. Intarrupted Bursts, 3? Looks

Figure 4 Interleaved Polarization Burst Mode

## Ofiginal page is <br> OF POOR PUALITY








Figure 5
Interleaved Bursts At Two Frequencies

### 2.1.3.3. Range Cell Migration

Range cell migration occurs during the synthetic aperture integration due to the radial range change between the radar and a ground scatterer. This migration consists of a linear component with time, $\dot{r} t$, and a quadrature ratio component, $1 / 2$ rit. $^{2}$ The range cell migration function is analagous to the phase focusing function and can be derived in the same manner. The time, $t$, is generally referenced to the point of closest approach, $r_{m}$, for a non-rotating earth. A rotating earth component of velocity toward the radar will actually make the point of closest approach occur slightly later in time. An orbital altitude rate associated with eccentricity will also affect the time of closest approach. The radar bean pointing departure from broadside will shift the time of signal return and thus introduce an $\dot{r}$ term equal to $V_{s} \cos \theta_{v}$, where $\theta_{v}$ is the angle between the velocity vector and the ground scatterer.

Table 2 indicates that for ERSAR there are 86 slant range bins of range walk to be compensated for across t!e four looks. Assuming an incidence angle of $60^{\circ}$, a ground bin of 15 meters results in a slant range bin of $15 \sin 60^{\circ}=12.99 \mathrm{~m}$. Thus, the range walk is $86 \times 12.99=$ $1 i 17 \mathrm{~m}$ or $7.45 \mu \mathrm{sec}$. The range curvature is given as 7 slant range bins $=90.9 \mathrm{~m}$. The maximum range :alk across the farthest look $=1117 / 4+$
$3 / 4 \times 90.9=347.4 \mathrm{~m}$ or 26.7 slant range bin's (slightly less than the 28 specified). The 4 -look time is $4 \times 0.686=2.744 \mathrm{sec}$. Thus, the maximum range rate is assumed to be $407.1 \mathrm{~m} / \mathrm{sec}$. The corresponding acceleration is $96.6 \mathrm{~m} / \mathrm{sec}^{2}$. The coherent integration of each look must take place along the curved range cells to achieve the desired $r$ asolution. The range cell migration infers a two-dimensional matched filter requiremert for which the processing algorithm alternatives in Section 2.2 have been developed.

### 2.1.3.4. Clutter Lock Requirements

The approximate satellite position, velocity and altitude information is provided to the SAR processor. The SAR antenna pointing (roll. pitch and yaw) will be controlled within $\pm 1^{0}$. Uncertainities in the data could easily cause the signal to depart in center frequency by more than the per look bandwidth. For example a yaw angle of $1^{0}$ translates to a cross-cone angle of the slant range vector of $0.83^{\circ}$ from broadside. This results in a shift in duppler of 950 Hz for the L-band 300 km ERSAR satellite. At X-band the same yaw shifts the doppler by 7153 Hz . Earth velocity for a satellite polar orbit has a radial component at the equator of $V_{E Q} \sin$ (incidence angle) $: 463 \times \sin 60^{\circ}$ $=401 \mathrm{~m} / \mathrm{sec}$, causing a doppler shift of 3413 Hz at L-band. For accurate ERSAR mapping such as 10 m absolute location ( $\Delta X$ ), the doppler shift due to the earth must be known to,

$$
\Delta f=\frac{2 V{ }_{s} \Delta X}{R_{s} \lambda^{\lambda}}=\frac{2 \times 7730 \times 10}{564.4 \times .235}=1.17 \mathrm{~Hz} \text { cr } 0.034 \%
$$

Clutter locking should be maintained to within 100 Hz to keep the per look tuning within $20 \%$ of the look bandwidth ( $4 \%$ of the PRF). Errors greater than $50 \%$ of the look bandwidth will degrade the azimuth sidelobes due to azimuth spectral aliasing. Clutter locking errors due to satellite yawing do not cause a mapping error since the doppler used to tune the processing bandwidth to the signal spectrum is employed in forming the map location.
2.1.3.5 Automatic Focusing Requirement

The doppler chirp rate $f_{d}$ of a target viewed by the SAR beam must be
matched by the azimuth correlation process to achieve the azimuth beam compression and desired target resolution. It is equal to $f_{d}=$ $2 \ddot{r} / \lambda$. The radial acceleration $\ddot{r}$ is a function of the satellite velocity $V_{s}$, the slant range $R_{s}$, the ratio of the target distance $y$ from the orbit axis to the orbit radius, $R_{E}+h_{E}$, for near broadside operation for a non-rotating earth. The relation for a circular orbit is

$$
r=\frac{V_{S}^{2} y}{R_{S}\left(R_{E}+h_{E}\right)}=\frac{V_{S}^{2} R_{E} \cos \alpha}{R_{S}\left(R_{E}+h_{E}\right)}
$$

where $a$ is the earth angle between the satellite and the earth target and $h_{E}$ is the orbit height; $R_{E}$ is the earth radius. The nominal value of $\ddot{\dot{r}}=100.84 \mathrm{~m} / \mathrm{sec}^{2}$ at the midpoint of the ERSAR swath ( $V_{s}=7.73 \mathrm{~km} / \mathrm{sec}$, $R_{E}=6373.95 \mathrm{~km}, h_{E}=300 \mathrm{~km}, R_{S}=564.4 \mathrm{~km}, \alpha=4.2^{\circ}$ ). Over the 40 km swath, $\ddot{r}$ varies from $104.03 \mathrm{~m} / \mathrm{sec}^{2}$ at the near edge to $97.8 \mathrm{~m} / \mathrm{sec}^{2}$ at the far edge. The variation $\Delta \ddot{r}=6.23 \mathrm{~m} / \mathrm{sec}^{2}$ occurs across the 40 km swath, resulting in a change in doppler rate

$$
\Delta \dot{f}_{d}=\frac{2 \Delta \ddot{r}}{\lambda}=\frac{2 \times 6.23}{.235}=53 \mathrm{~Hz} / \mathrm{sec}
$$

The depth of focus, expressed in terms of allowable change in doppler rate, can be determined assuming a quadratic phase error allowance of $\pm \pi / 2$ at both ends of the synthetic aperture relative to the center of the aperture. The phase error

$$
\Delta \theta= \pm \frac{\pi}{2}=2 \pi \int_{0}^{T / 2} \int_{0}^{T / 2} \Delta \dot{f}_{d} d t^{2}=\frac{\pi}{4} \Delta \dot{f}_{d} T_{L}{ }^{2}
$$

The term $\Delta \dot{f}_{d}= \pm 2 / T_{L}{ }^{2}$. Thus, the depth of focus, $D O F=\left(4 / T_{L}{ }^{2}\right)$, where $T_{L}$ is the time to form the synthetic aperture for each look. For an azimuth compression ratio of 350 and a bandwidth of 500 Hz , $T_{L}=0.7$ seconds. Thus, the depth of focus $\left.=4 / .7^{2}\right)=8 \mathrm{~Hz} / \mathrm{sec}$. From the point of view of a single look, the reference function does not have to change more often than $40 \mathrm{~km}(8 \mathrm{~Hz} / \mathrm{sec} \div 53 \mathrm{~Hz} / \mathrm{sec})=6.04 \mathrm{~km}$
swath. However, the reference function will have to change more often in a cross-track direction to permit the registration of the four looks within 0.2 azimuth cell.

The presence of a quadratic phase error corresponds to a mismatch of the reference function chirp rate with the target chirp rate. This mismatch has the effect of a doppler error between the four looks, af, producing a misregistration of the inages, $\Delta x$.

$$
\Delta f=\frac{2 V_{s} \Delta x}{\lambda R_{s}}
$$

For $x=0.2, \delta_{A Z}=0.2 \times 15=3$ meters, $\Delta f=\frac{2 \times 7730 \times 3}{.235 \times 564.4 \mathrm{~km}}=0.35 \mathrm{~Hz}$

The time interval between the four looks $=3 T_{L}=2.1 \mathrm{sec}$. The aliowable change in doppler rate cross-range $=(0.35 / 2.1)=0.17 \mathrm{~Hz} / \mathrm{sec}$, Thus, for registration, a new reference function should be applied more often than $40 \mathrm{~km} \times(0.17 \mathrm{~Hz} / \mathrm{sec} ; 53 \mathrm{~Hz} / \mathrm{sec})=126$ meters of swath, corresponding to 8.4 resolution elements. The specification calls for a new reference function cross-track every 8 samples, which is reasonable. Fortunately, the chirp rate variation across the swath is deterministic from the geometrical parameter variations and may be computed from the one or two measured chirp rates refined by auto-focusing techniques.

This section describes the candidate SAR processing algorithm considered for the ADSP and gives pertinent implementation data on them. Performance levels are treated separately in the following section (Section 2.3).

### 2.2.1 Basic SAR Processing Functions

The formation of images from synthetic radar arrays follows the same principles found with optical lenses or real radar antennas. Images can be focused in the far-field when the energy reaches the antenna as a virtual plane wave relative to the antenna diameter or in the near field. The far-field SAR is generally terr od an unfocused SAR and, being relatively simple to process, is not treated in this study. Processing in a focused SAR is essentially a two dimensional matched filter process as indicated in Figure 6.


Figure 6
Focused SAR Image Processing Functions

The range, or cross-track, dimension may be processed by conventional redar pulse compression techniques, the mothod used depending on the type of radar waveform employed. The azimuth, or along track, dimension can be considered in the same functional sense as the range dimension. Here, the radar footprint of the real antenna with its doppler frequency response forms a quadratic phase (or linear FM) function across its width. As the real antenna moves with its scanning piatform the radar footprint "waveform" moves along the image plane in azimuth in the same manner that the uncompressed radar pulse moves across the image plane in range. Thus a filter matched to the radar footprint "waveform" operating in the azimuth dimension will compress the radar footprint to the resolution defined by its time and handwidth parameters. The limiting azimuth resolution for a focused SAR is one half the azimuth dimension of the real antenna.


Figure 7
Satellite Radar Swath

Two other functional requirements are important in SAR processing, range migration and multiple looks. As a SAR satellite platform
moves a long its track, the earth velocity causes the radar return to move linearly with the motion of the spacecraft (S/C) (See Figure 7). In addition, the range of a fixed point on the earth will vary in a quadratic manner as the S/C radar footprint moves past.
Radar returns from objects vary in amplitude as a function of the aspect angle of the radar. This tends to give radar mapping return a speckled appearance. The effect can be reduced by non-coherently summing coharently processed segments of the radar fcotprint "waveform". This multi-look processing reduces the final image resolution in proportion to the number of looks processed. The bandwidth of the processing required for each look is also proportionally reduced. The combined etfect of range migration, linear and quadratic range walk and multiple look processing is illustrated in Figure 8.


Figure 8
Azimuth Resolution Element Motion
Relative Fixed Antenna

### 2.2.2 Time Domain Processing

2.2.2.1 General Time Domain Processing Consideration

The general form of a time domain processor is shown in Figure 9. It essentially implements the SAR processing equations with a time domain analog of the functions. After range correlation, which consists of a tapped delay line finite impulse response filter in a time domain implementation, the signal is filtered (presummed) into bands curresponding to the looks to be processed. This serves to greatly reduce the computation requirements of the azimuth correlators. A simple example will illustrate the effect. Consider an azimuth correlator with 1000 azimuth samples over the total beamuidth with an azimuth doppler bandwidth (and sample rate) of 1000 Hz . The number of complex multiples per second to perform a time domain correlation is then $10^{6}$ per range cell. If the signal is processed as four looks, the bandwidth per look becomes 1000/4 and the number of azimuth samples covering each look is $1000 / 16$ giving a total computation rate for four parallel look iilters of ( $1000 \times \frac{1000}{4} \times 4$ ) = $10^{6} / 16$ per range cell. Thus, the computation rate is ${ }^{4}$ inver ${ }^{16}{ }^{6} 1 y$ proportional to the square of the number of looks processed. The net complex computation rate for a time domain correlator is $C T_{A} B_{A} / N_{L}{ }^{2}$


Figure 9
Multi-Look Time Domain Processor Functions
where; $T_{A}$ is the synthetic aperture integration time, $B_{A}$ is the doppler bandwidth. $N_{L}$ is the number of looks, and $\mathbb{C}$ is the oversampling factor.

After prefiltering and azimuth correlation, the multiple looks are summed. A time delay must be inserted in the look filter prior to summation to compensate for the time offset of the look energy from particular looks.

Two general types of architecture have been identified for time doma in processor and have been evaluated in Reference 1. They have been designated Type A and Type B and we will continue to use those designations for clarity. Type $A$ (Figure 10 ) is a classical tapped delay line matched filter in which the intertap delay lines connected serially contain the entire SAR range swath.


Figure 10
Serial Time Domain Azimuth Processor

Type B (Figure ill) performs azimuth filtering by the construction of parallel azimuth beam correlators, each accumulating the processed beam over the entire range swath. The general simplicity of form for these processors plus their minimization of memory storage makes them candidates for the ADSP. We will develop sufficient detail on these processors to permit evaluation on an equal hardware technology basis with the frequency doma in methods.


Figure 11
Parallel Time Domain Azimuth Processor (Type B)

### 2.2.2.2 Serial Time Domain Processor (Type A)

Figure 12 is a more detailed sketch of a Type A or serial processor. The processing is segmented into processing units labeled as correlator \#1, correlator \#2 ... up to correlator N. Each processing unit has a FIFO (first-in, first-out) memory large enough to hold a single pulse radar sweep. Since the range processing is completed,
the FIFO memory covers the range swath plus any uncompensated range: migration. The FIFO memory can be implemented with shift registers: but random access memories which have more capacity per chip, consume less power and are less expensive are a preferred choice. In addition, the storage length of a FIFO random access memory is easily changed by address control. The address control mechanism conzists of a counter, a reset control word and a comparator. On each count new data is read into the memory and the data stored is read out. When the count reaches the value of the reset control word, the comparator will output a signal which resets the counter to zero. Thus, the delay (FIFO) capacity is easily changed. The memory read-modify-write cycle time must be less than the clock period. If it is not, multiple memory units can be multiplexed to meet the speed.


Figure 12
Detail of Type A Processor

Range migration compensation is accomplished within each correlator by a combination of variable clock interval (range cell) shifts ( $\tau_{v}$ ) plus interpolation. Simulations indicate that a two point interpolator will give adequate performance, but for the highest quality imagery it may be desirable to use a three or four point interpolator. The focusing function storage in each correlator unit is related to the required increments of range migration compensation. Each significant variation in focusing function, nominally every 8 range cells for the baseline SAR mission, has an associated range migration compensation. The total reference storage in each correlator unit is at least 4000/8 $=500$ words of at least 34 bits. These bits would be assigned as 22 bits for focusing function, 7 bits for incremental range migration, and 5 bits for the interpolator.

In addition to the variable time delay in each range migration compensation memory, a fixed incremental delay is inserted to aid in convenfently obtaining an output summation. In effect, each correlator segment is operated offset in time by one range cell. Then, as the partial sums are passed from one unit to the next through register delays they are automatically added with the appropriate delays.

In the baseline SAR system requirements, the four look prefiltering will give a clock rate of $11 / 4=2.75 \mathrm{MHz}$ for each azimuth correlator. Obviously, the computational hardware in each correlator unit can be operated at a higher rate and this is reflected in the three circuit tabulations given in Table 3 for the Type A system. A typical configuration for a $4 x$ multiplexed correlator is shown in Figure 13. Note also in Table 3 that the number of interconnections is modest for the Type A correlator modules.


Table 3
Type-A Circuits -- Interconnections

| Enicioh | CIRCUIT TYPES |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  | $278$ | $\begin{gathered} 2 x \\ 5.5 \text { min } \end{gathered}$ | AxM |
| PRI STORE (AK $\times$ 20) | 4k X | 4anms | 12 | 24 | 4 |
| PRI MEMOAY CONTROL | misc | celunneous | 10 | 10 | 10 |
| Rance COLNIER | coun | hers and registers | 4 | 4 | 4 |
| RMC AND CONIROL | Misctir | ceunatous | 60 | 70 | 0 |
| focus mult | - mu | LP, 2 ADDS | 15 | 17 | 19 |
| ADOER | adot | ERS and registers | 16 | 21 | 26 |
| CONTROL INIERFACE | misc | CEUANEOUS | 10 | 12 | 14 |
|  |  | rotal | 17 | 161 | 211 |
|  | 110 | INPIT output osured OUTPUT Sum CONTROL IN-OUT |  |  |  |
|  |  | rotal |  |  |  |

### 2.2.2.3 Parallel Time Domain Processor (Type B)

A detailed diagram of the Type $\mathbf{B}$ time domain processor is shown in Figure 14. Because of the nature of the architecture, a single common interpolator unit can be used for all of the correlators. However, this step creates a larse I/O pin requirement and the accuracy that can be used is therefore limited. The example shown has eijht, two point interpclators feeding the input bus. Range migration compensation is accomplished for each correlator by selecting the appropriate interpolated point and gating the desired range cell. The entire interpolation control sequence is generated and/or stored in the Common Control Unit. As the first correlator starts its integration process, the required RCM controis as a function of range for the first radar PRI are applied. This control sequence is then passed to the second correlator and the RCM control sequence for the second PRI is applied to the first correlator. This process is repeated until the last PRI to be integrated by the first correlator is received. A similar process is used for the focusing function store. The reference words pass serially through the string of correlators. Wnen a correlator unit completes its integration process, its output is selected and it starts the integration process again.

ORIGINAL FAGLE IS OF POOR QUALITY

Figure 14
Detail of Type B Processor

Table 4 lists the circuit counts for various multiplexing alternatives for the Type B architecture.

Table 4
Type-B Circuits -- Interconnections


### 2.2.2.4 Type A, B Comparison

A qualitative comparison of the Type $A$ and Type $B$ time domain processors is shown in Table 5. There is not a significant difference in the estimated cost or performance of the two approaches. Type B offers more flexibility for variation of SAR parameters. However, this is only true if the number of looks are held constant and the compression factor is less than the established size. For both systems, if the number of looks are reduced it takes significantly more hardware correlator modules to achieve real time processing rates and the system must be reconfigured. Our preference is with the Type B system because of its greater programmability and somewhat less fault sensitivity.

2.2.2.5 Prefilter and ilultilook Integrator

The general multilook (for four looks) prefilter situation is depicted for the input signal to the prefilter shown in Figure 15. The signal spectrum, which has been sampled at the sampling frequency $f_{s}$, is offset from the center by an amount $f_{\text {off }}$. The objective is to filter the signal into the four individual looks and decimate the sampling frequency by a factor of four. The clutter lock circuit will detect the offset frequency and this will be applied to the overall spectrum (point (a)) to center it about zero (point (b)). Each of the four individual look offset frequencies $f_{1}, f_{2}, f_{3}$, and $f_{4}$ are then applied and the result is low pass filtered to isolate the individual look bands. Since the bandwidth for each channel has been reduced by a factor of four, the sampling rate can also be reduced (decimated).


Figure 15
Multilook Filtering Functions

An implementation of the filtering and decimation function is given in Figure 16. A 32nd order low pass filter is used to achieve the desired passband and ripple characteristics. This is obtained from the approximation given in reference 2,

$$
\Delta F \approx \frac{\ln 2-\ln \delta_{2}}{\pi(M-1)}
$$

where $\Delta F$ is the passband to stopband transition width, $M$ is order of the filter, and $\delta_{2}$ is the passband ripple,

If we set $\Delta F=.05$, and $\boldsymbol{\delta}_{2}=.01$ we obtain $N \approx 32$. In the implementation of Figure 16, the decimation is accomplished by the serial to parallel shift register tap registers operating at a reduced (4 to 1) clock rate. The decimation is exploited by operating the filter coefficient multipliers at the input slock rate and accumulating four products prior to the formation of the adder tree.

JRGGINAL PAGE :<br>OF POOR QUALITY



Figure 16
Low Pass Filter
(32nd Order) Decimate by 4

An implementation of the multilook integrator is shown in Figure 17a. The method shown with four individual look integrators is preferred
over the delay 1 ine approach since less memory is used. If $X$ is the product of range cells times azimuth compression samples per look, the storage in Figure 17 a is 4 x . A delay line approach (indicated in inset) would require $3 X+2 X+X=6 X$.


Figure 17 a
ORIGINAL PAGE IS
OF POOR QUALITY
Multilook Integrator
2.2.2.6 Time Domain Hardware Summary

Table 6 lists a hardware summary of the time domain processing options. The totals assume 11 MHz clock rates in the correlator modules. Mthol.gh this speed is achievable, it is somewhat high for TTL application. We feel it is practical in this instance, since it is an on-module clock rate. Most of the module interface rates are at one fourth this rate or 2.75 MHz . Nevertheless, the total number of modules is 3 to 4 times as high as the number expected for an FFT convolver implementation.

Table 6

| FUNCTION | $\begin{aligned} & \text { NUMBER } \\ & \text { TYPEA } \end{aligned}$ | $\begin{aligned} & \text { QUIRED } \\ & \text { TYPE } \end{aligned}$ | REMARKS |
| :---: | :---: | :---: | :---: |
| RANGE CORRELATOR | 130 | 130 | 11 MHz CLOCK, $\sim 200$ CIRCUITS EACH |
| RANGE CONTROL | 1 | 1 |  |
| PREFILIER CHANNEL MODULATCR AND CONTROL | 4 | 4 | $11 \mathrm{MHz} \mathrm{IN}$,2.75 MHz OUT |
| INTERPOLATOR | - | 4 |  |
| AZIMUTH COKRELATOR | 420 | 420 | 11.0MHz CLOCK, 4 CHANNELS IMODULE |
| CONTROLLER | 1 | 1 |  |
| INTERPOLATOR AND MAGNITUDE | 1 | 1 | 4 MODULES/INTEGRATOR |
| MULTILOOK ACCUMULATOR | 16 | 16 | 500K WORDS/MODULE |
| CONTROLIER | 1 | 1 |  |
| SYSTEM CONTROLLER | 1 | 1 |  |
| INTERFACE/TEST | 1 | 1. |  |
|  | 577 | 581 |  |

### 2.2.3 FFT Convolver

Azimuth processing, using an FFT convolver, has been divided into two approaches: Case 1 , discussed here, where the range and azimuth correlation are operated separately and Case 2, discussed in Section 2.2.5, where a two dimensional convolution is performed either including, or not including, the range compression function. Range compression alone using FFT convolution is discussed separately in Section 2.2.6.

FFT azimuth correlation is accomplished by forming an azimuth matched filter moving along constant range lines of the radar return. The
filter is formed by transforming the data to the frequency domain, multiplying by the complex reference of the azimuth focusing function and performing the inverse transform to convert the signal back to the time domain. The frequency transform filter thus formed performs a circular convolution of the azimuth focusing function with the signal data within the FFT window.

With multiple look processing, separate frequency filters must be provided for each look. This is done as shown in figure 17 b with separate spectral references for each look and separate inverse FFT's. Because the frequency band coverage of a single look is a fraction of the total, the frequency samples can be decimated (reduced sample rate) prior to inverse FFT processing. In the example of Figure 17b with four looks, the number of inverse FFT points is reduced by a factor of four to coincide with a four to one reduction in bandwidth.


Figure 17 b
FFT Convolution-Azimuth Processing

The ability to perform range migration compensation in the FFT convolver approach is due to the unique relationship of range migration and frequency coefficients in the azimuth processor as illustrated in Figure 18. As indicated in Figure 8, the range migration correction is a function of beam angle and the frequency coefficients are representetive of beam angle. Therefore, range migration correction by shifting fresuency coefficients in range will correct for all of the targets with energy in that coefficient. We have shown (Section 2.3) that for adequate performance the range migration correction must be refined using interpolated sample points.

FREQUENCY


Figure 18
Unique Relationship Between Range Migration and Frequency

We have selected the FFT convolver algorithm for the recommended ADSP design and discussion of its implementation is contained in Section 2.5.

### 2.2.4 Subarray Processing

Subarray (or step transform) SAR processing is performed by converting segmented received time data to the frequency domain and resolving the individual doppler trajectory of the map elements. Partitioning the data into small groups or subarrays permits range walk and other correction factors to be incorporated. This partitioning also enables a continuous strip map to be processed in an efficient manner by adding only new subarray data to the already accumulated data and dropping the old subarray data.

SAR processing using subarrays consists of the following five steps.

1. Focus data over short subarray time.
2. Store subarray data over large array length.
3. Combine subarray outputs to form large array.
4. Compute new subarray data entering large aperture.
5. Drop old subarray leaving large aperture.

Figure 19 illustrates the step transform algorithm. Since the data is being continuously fed into the step transform processor, the overlapped subarray computation can be performed as a running process. Similarly, the output fine resolution can be computed as soon as a full aperture of subarrays have been computed. This is also done as a running process where new subarray data is added while the old data is dropped.

Factors to be considered in implementing the subarray approach include input spectral aliasing, short aperture sidelobes, and straddling loss. These and other factors have been analyzed and are discussed in Reference 3.

## ORICINAL PAGE IS

OF FOOR QUALITY


Figure 19
Block Diagram of Subarray Azimuth Processor
The functional diagram of Figure 19 illustrates the subarray approach for an infinite deramping function. The azimuth processing begins with the application of the infinite deramping function following range pulse compression. The reference function applied is illustrated in Figure 20. It is essentially a continuous frequency ramp wrapping around at the sample rate. Total frequency coverage within the sample rate will cover the entire SAR antenna beam response including multiple looks and guard band. A single target will encompass the full response.

After demodulation, the data is collected in a subarray memory which forms overlapping subarray segments which are amplitude weighted and processed in the subarray focusing FFT. Data is then placed in a corner turning memory whose output is first processed for range walk compensation prior to final subarray combining. The subarray combining data is integrated along paths as indicated in Figure 21. Coherent integration occurs across a single look at a single coarse azimuth frequency with multiple look, non-coherent integration following along the same frequency.


Figure 20
Continuous Deramping and Subarray Processing


Figure 21
Deramp Coarse Resolution

Buik storage in the subarray processor can be estimated by reference to Figure 22. The integration must occur along horizontal paths for each look. As new subarray data is received integration intervals as a finction of frequency become filled. These are then read out and processed for fine resolution. Since the data can be discarded after integration the minimum storage requirements for each range cell will be contained within the four triangular patches with width covering a look integration time and height corresponding to the look bandwidth. Thus, if there are $M$ subarray coefficients and $N$ subarrays per look the minimum storage is MN/2 or MN if a double buffered implementation is used. For the ERSAR case, the PRF is 2500 and the look integration is 0.7 seconds. The input time-bandwidth product is then 1750. Using a subarray size of 64 points and an overlap factor of 2 to 1 , the number of subarrays per look becomes 54 . The number of spectral coefficients used is $4 / 5$ th of $64=52$. The minimum memory storage is therefore $52 \times 54 / 2=1404$ per range cell or about $5.7 \times 10^{6}$ words to encompass 4086 range cells. A programmable design of the bulk storage unit in a subarray processor is difficult and it may simplify its control if a double-buffer arrangement can be used. In this case the total memory would be $11.4 \times 10^{6}$ words.

The complex computation rate for the baseline can be estimated per look interval per range cell as follows:


The rate of $168 \times 10^{6}$ assumes equal computational load for the weighting function and the FFT butterfly computations. If the weighting is accomplished by combining it with the FFT process, half of the weighting oreration are eliminated and the net computation rate
becomes $148 \times 10^{6}$ complex operations per second.


MINIMUM STORAGE - MNI2
DOUBLE BUFFER STORAGE •MN

Figure 22
Bulk Storage in Subarray Process

### 2.2.5 Two-dimensional Convolution

The 2-dimensional convolution approach to SAR azimuth processing has been divided into two parts, one in which the range correlation is included and the second for azimuth compression only and the 2-dimensional process is applied to correct for range migraticn. The latter approach also includes a hybrid FFT time domain technique.
2.2.5.1 Full Range Azimuth Correlation 2-D Process

Two dimensional convolution as illustrated in Figure 23 is an ideal approach in terms of computational efficicency for a single reference, but the small depth of focus for a single reference spectrum results in reduced efficiency. For example, the azimuth reference function must be updated every 6 km of swath to maintain single look focus. Over a 40 km swath, the reference would have to be adjusted at least 9 times iŕ a $1024 \times 40962-0$ array were used. The process also rules out the incorporation of multilook registration in the process since this requires a reference update every 8 range cells.


Figure 23
Two-Dimensional Convolution Using FFT Processing

### 2.2.5.2. 2-D Convolution With Range Compressed Data

A 2-D convolution azimuth correlation procedure can also be employed on the range compressed data to accomodate range walk. This function can be done using the standard FFT algorithms (4) or other more recent algorithms such as the fast polynomial transform (5). The fast polynomial transform has been applied to SAR processing ( 6,7 ) and a short summary will be provided here.
Consider the convolution of the range compressed data $\{a(m, n)\}$ with reference $\{r(n, m)\}$, each of size $M \times N$. For the SAR processing in mind, $M$ is the number of range cells for which a single focusing function could cover, typically 128 or 256 , and $N$ is the azimuth coverage, typically 4096 or 8192.
From the data array $\{a(m, n)\}$, we generate $(r+1)$ subarrays, where $r=1+\log _{2}(N / M)$. The $i-$ th subarray $\left\{a^{i}(m, n)\right\}$ is of size $M_{x}\left(N / 2^{i}\right)$, $i=1,2, \ldots, 1+\log _{2}(N / M)$ except array $\left\{a^{0}(m, n)\right\}$ which is of size $M_{X}(M / 2)$. To generate $\left\{a^{1}(m, n)\right\}$, we take each row of $\{a(m, n)\}$, partition in the middle, take the sum and difference of the corvesponcing elements:

$$
\begin{aligned}
& a^{1}(m, n)=a(m, n)-a(m, n+M / 2) \\
& a^{-1}(m, n)=a(m, n)+a(m, n+M / 2)
\end{aligned}
$$

$$
0 \leq n<N / 2-1
$$

Of the two arrays generated $\left\{a^{1}(m, n)\right\}$ and $\left\{a_{2}^{1}(m, n)\right\}$. The first one is kept and the second is used to generate $a^{2}(m, n)$, row by row, as follows:

$$
\begin{array}{ll}
a^{2}(m, n) & =a^{-1}(m, n)-a^{-1}(m, n+N / 4) \\
a^{-2}(m, n) & =a^{-1}(m, n)+a^{-1}(m, n+N / 4)
\end{array} \quad 0 \leq n \leq \frac{N}{4}-1
$$

This process is illustrated in Figure 23.

[^0]The process is repeated $r$ times, the last two arrays are $\left\{a^{r}(m, n)\right\}$ and $\left\{a^{0}(m, n)\right\}$ as indicated in Figure 24. The operation takes a total of $2 M N$ additions.
The operation is also precomputed on the reference $\operatorname{array}\{r(m, n)\}$ and stored.

$m^{\text {th }}$ row of $a^{-1}(m, n)$
Gener at ion of $a^{2}(a, n)$
$m^{\text {th }}$ row of a ( $m, n$ )
Generation of al $(m, n)$

Figure 24a. Generation of $\left\{a^{1}(m, n)\right\}$ from $\{a(m, n)\}$ and of $\left\{a^{2}(m, n)\right\}$ from $\left\{a^{-1}(m, n)\right\}$.


Figure 2:3b. Renetition of Figure 24a Process

FIGURE 24. Polynomial Transform Operations

The next step is to perform the polynomial transform of the arrays $\left\{a^{i}(m, n)\right\}$ and $\left\{b^{i}(m, n)\right\}, i=0,1, \ldots, r$. The structure of the polynomial transform is very similar to radix 2 FFT, except each row in the array is treated as a single element. To illustrate with a "DIF" structure applied to the $\left\{a^{l}(m, n)\right\}$ array, we first take two rows M/2 apart, and take the sum and difference of the corresponding elements, i.e..

$$
a^{1}(m, n) \pm a^{1}(m+M / 2, n) \quad 0 \leq N / 2-1
$$

The sum is left alone, but the difference is cyclic shifted (or wrapped around) by a prescribed amount (depending on the corresponding twiddle factor in the FFT structure) and the sign of the urapped around part is changed as shown in Figure 25. These two rows are put bark in the array and two more rows are taken out and cperated on in this manner. After $M / 2$ such operations, the first stage is completed. Another set of operation then commences, as in the $2 n d$ stage of the FFT. That is, rows that are M/4 apart are operated on pairwise.
This operation is repeated for the $\log _{2} M$ stages.


## ELEMENT BY ELEMENT

ADCITION AND SUBTRACTION

Figure 25. First Stage of Polynomial Transform

The "butterfly" operation in the polynomial transform can be pipelined by inserting the results back into the array in a eriscross manner ( 8 ), as follows. For the first stage, the first pair is $\left\{a^{1}(0, n)\right\}$ and $\left\{a^{1}(M / 2, n)\right\}$, the results are, for example $\left\{a_{+}^{1}(0, n)\right\}$. The 2nd pair to operated on by the butterfly should be $\left\{a^{( }(M / 4, n)\right\}$ and $\left\{a^{l}(3 M / 4, n)\right\}$, obtaining the results $\left\{a_{+}^{1}(M / 4, n)\right\}$ and $\left\{a_{-}^{1}(M / 4, n)\right\}$. Now the result $\left\{a_{+}^{1}(0, n)\right\}$ should be stored in row $0,\left\{a_{-}^{1}(0, n)\right\}$ in row $M / 4,\left\{a_{+}^{1}(M / 4, n)\right\}$ in row $M / 2$, and $\left\{a_{-}^{1}(M / 4, n)\right\}$ in row $3 M / 4$.

The total number of additions for transforming the $\left\{a^{1}(m, n)\right\}$ array is $N \log _{2} M$. So the total number of additions for transforming all the $\left\{\left(a^{i} m, n\right)\right\}$ arrays is $2 N \log _{2} M$.

Again, we note that the polynomial transforms of the reference arrays $\left\{r^{1}(m, n)\right\}$ can be precomputed and stored.
We denote the polynomial transforms of $\left\{a^{i}(m, n)\right\}$ and $\left\{r^{i}(m, n)\right\}$ by $\left\{a^{i}(m, n)\right\}$ and $\left\{\tilde{r}^{i}(m, n)\right\}$ respectively.
The next step is a cyclic like convolution of the rows of $\left\{\tilde{a}^{i}(m, n)\right\}$ with $\left\{\sim^{i}(m, n)\right\}$. The difference from the ordinary cyclic convolution is that the wrapped around part has a sign change. References (6) and
suggests the use of an idea proposed by Arambepola and Rayner (9) which accomplishes the convolution via a transform algorithm that is only a slight modification of the FFT algorithm by changing the twiddle factors. One could, on the other hand, use the straightforward FFT to compute the noncyclic convolution and simply do a circular shift and change the signs of part of the result. This step requires $2 M N\left(\log _{2} N-3\right)$ additions and $N\left(\frac{4 N}{3}+\log _{2} N-3\right)$ multiplications.

Finally these $(r+2)$ arrays are combined to give the cyclic convolution oi the data $\{a(m, n)\}$ and the reference $\{r(m, n)\}$. This step involves the merging of arrays of sizes $M x_{x}\left(N / 2^{i}\right)$. The merge start; with the summing and differencing, element by element, two smallest arrays, each of size $M x\left(N / 2^{r}\right)$ (Figure 26 ). The result is a MaM array. This is then merged with the next size array (i=r-1), using the same element by element sum and difference. This process is continued until a final array of $M \times N$ is obtained. This is the result of the cyclic convolution.


Figure 26
Merging of two MxM/2 arrays

The advantages claimed by the fast Polynomial Transform method are:

1. It requires fewer multiplications than using the FFT.
2. The process of Polynomial transformation can be performed in parallel.

The comparison with FFT in terms of arithmetic operations are summarized in Table 1 of reference 6 . The number of additions are comparable and the number of multiplications have a ratio of approximately 5:3 in favor of the polynomial transform. Bergland (10) has shown in 1968 that there is a $30 \%$ saving in multiplications if one uses radix 4 rather than radix 2. Recently Nakayama (11) proposed a mixed decimation FFT algorithm which results in a $15 \%$ saving over the conventional $F F F$ for the radix 2 case. In his algorithm, the two input butterfly computation ker . $\mathrm{P}^{\top}$ is retained. The saving in the radix 4 case is about 5 N8\%.

The computation of 2-dimensional transforms is conventionally carried out by first transforming each row and then transforming each column. It has been shown (12) that a simultaneous row-column decimation would result in a $25 \%$ saving. It is therefre possible to combine the simultaneous 2-dimensional decimation with the other schemes mentioned previously to obtain a substantial saving over the straightforward FFT approach. Indeed, it has been shown (13) that a $40 \sim 50 \%$ saving
possible. Needless to say, the saving is achieved at the expense of increased complexity of the algorithm. But the polynomial transform approach, requiring operations on row vectors of different length also increases the complexity considerably over that of a conventional FFT. Thus, the first purported advantage of savings in arithmetic operations is achievable with other approaches with perhaps a simpler structure of the hardware processor. This latter point definitely would require further study to establish.

The second advantage of the polynomial transform is also not convincingly demonstrated. The arrays $a^{i}(m, n)$ are of differsint sizes, the row sizes of successive arrays are two to one. Although these operations can be carried out in parallel, it is doubtful that the saving would be substantial. A considerable portion of the hardware would be idle. Essentially the speed advantage of using parallel operation would be about 2 to 1 but the hardware would be idling about half of the time.

The tentative conclusion we have reached at this time is that, in terms of simplicity of structure, a processor based on the conventional FFT or some minor modifications appears to be most attractive approach for a two-dimensional convolution.

### 2.2.5.3 Hybrid 2-D Process

A final 2-1 process has been suggested by Wu and Liu (14). This approach, indicated in Figure 27, employs FFT convolution for azimuth correlation and a tapped delay line convolver for the range migration correction. Ordering of data appropriately for the tapped delay line convolver is accomplished by doing the azimuth processing with a multiplexed FFT structure. This method, which requires a larger memory storage, outputs data from the first FFT on a singie frequency coefficient sequential range basis. Range migration correction can then be applied with a tapped delay line of length equal to the range migration. Modularity and an incremental growth and implementation are also difficult to achieve.

In principal this method is functionally equivalent to the FFT convolver algorithm using range interpolation in the range migration compensation. The

## ORIGINAL PAGE IS OF POOR QUALITY

memory required is at about 2.5 times the minimum memory storage required in the FFT convolver corner turning memory. Since the minimum memory storage is about $17 \times 10^{6}$ words, the total impact on cost is large.


### 2.2.6. Range Correlation

### 2.2.6.1 Range FFT Convolver

The forward-inverse FFT frequency domain matched filter, Figure 28 is now an established technique for digital convolution for many applications (15). Matched filtering is accomplished by spectral domain multiplication with the inverse FFT providing the matched filtered time donain output.

The advantages of the forward-inverse FFT matched filter are well known and include:

- A computation-efficient algorithm.
- Complete waveform flexibility.
- Adaptable to modular construction.
- Precise, predictable performanre level.
- Implementation can be adjusted to accommodate different processing speed requirements.
- Will benefit from advanced technology developments.
- Direct interpolation of output data possible.


Figure 28
FFT Matched Filter System
The input data can lie processed from real sampled data as might be acquired with the intention of processing in an optical processor. Using the FFT for providing the real-complex conversion is the most convenient approach. If this is done the input data window is normally twice as long, but two real channels can be processed simultaneously in ine complex FFT window.

The FFT matched filter (range correlator) will functionally have the form shown in Figure 29. Because the range swath interval is a fraction of the total radar pulse repetition interval (PRI) the range data is normally buffere': after $A / D$ conversion to slow down the data rate to the range correlator.


Figure 29

## FFT Convolution-Range Processing

A tradeoff exists in setting the FFT processor parameters of an FFT convolver. In general, the larger the size of the FFT, the more efficient the computations. For the baseline system range processor, the total number range samples to be processed per pulse is 4000 (range image swath) plus 475 (uncompressed pulse length) plus 86 (range migration) for a total 4561 samples. This can be processed as a sliding aperture convolver with an FFT size of 1024 points or as a batch processor of 8192 or 4096 points. If the maximum FFT size is 4096 , a seperate FFT of 1024 must be processed to cover the total range interval. The net computation rate for the three cases is given in Table 8 which shows that the most computationally efficient size for the baseline system is the 4096 point FFT.

Table 8
Range FFT Convolver Alternatives

| characteristic | SLIDING APERTURE | MAX-SWATH PROCESSOR |
| :---: | :---: | :---: |
| HTT SIZE | $\geq 2$ WAVEFORM | $\begin{gathered} 34000+475+89 \\ 181920 R+096+1024) \end{gathered}$ |
| COMPUTATIONS PER PULSE | $\begin{gathered} 8\left(51212 \log _{2} 1024+2\right) \\ (90.112) \end{gathered}$ | $\begin{gathered} \frac{8192 \text { FFT }}{\left.409612 \log _{2} 8192+2\right)} \\ (114,688) \\ \frac{4096+1094 \text { FFT }}{204812 \log _{2}(1096+2)+512(22)}(164,512) \end{gathered}$ |
| COMPUTATION RATE ICOMPLEX <br> OPERATIONS / sec | $225 \times 10^{6}$ | $\begin{aligned} & 8192 \text { FFT }: 287 \times 10^{6} \\ & 4096+1024: 161 \times 10^{6} \end{aligned}$ |

### 2.2.6.2 Step Transform Linear Frequency Modulated (LFM) Signal Matched Filter

The step transform subarray algorithm was originally conceived as a technique for L.FM matched filter processing (16,17). The algorithm is shown symbolically in Figure 30. A received LFM ramp is demodulated by a sawtooth ramp with the same slope, producing CW frequency segments "stepped" by the frequency span of the sawtooth ( $\Delta f$ ) and of length equal to a tooth ( $\Delta T$ ). A spectral analysis of the successive CW segment produces a set of frequency response functions that are stored in a time-frequency data reorder memory shown in Figure 32. Data as read from the time frequency matrix along diagonals results in a linear phase shift along the diagonal producing the "fine" range resolution at the processor output. Weighting is applied prior to the second FFT to reduce range sidelobes.

The step transform pulse compression technique provides a natural segmentation of data as a function of range. It offers a means of applying range migration compensation to small blocks of range samples. Its main disadvantage is that it is not universally programmable for waveform type or time-bandwidth product.


Figure 30
Step Transform LFM Pulse Compression Algorithm


Figure 31

## Step Transform LFM Range Pulse Compression Processing

(For maximum efficiency, DFTs are implemented as FFTs)

### 2.2.6.3 Digital Tapped Delay Line Correlator

Tapped Delay Line Pulse Compression Filters - The output $y(t)$ of a matched filter implemented via convolution of an input signal $s(t)$ with the impulse response of the matched filter $h(t)=x(-t)$, where $x(t)$ is the transmitted waveform, is:

$$
y(t)=\int_{-\infty}^{\infty} S(\tau) H^{*}(t-\tau) d \tau
$$

The signal and filter functions are generally represented as complex inphase and quadrature samples for sample data operations. Thus, a physical realization of the filter function must accomodate the complex multiolication operation.

$$
(a+j b)(c-j d)=a c+b d+j(b c-a d)
$$

A tapped delay line matched filter then takes the form of Figure 32. The total number multipliers is over $4(T W)^{2}$ where $T W$ is the time bandwidth product of the waveform. With a maximum TW product of 660 for the ADSP, the total number: of taps required is 765 .

A time doma in range processor is very simple to use. It is only necessary to input the radar waveform of arbitrary length into the reference register together with the appropriate number of zeros. If a single unit is used as in Figure 32, however, it must operate at a ciock rate of 11 MHz . The system would be partitioned into 130 modules with up to 200 circuits per module all operating synchronously at an 11 MHz clock rate. The total number of circuits for the range correlator would in this case be about equal to the total number required by both the range and azimuth correlators using FFT convolver algorithm.

ORIGINAL PACE IS
OF POOR QUALITY


Figure 32
-sme Domain Range Convolver Controls

### 2.2.7 Aigorithm Study Summary

A graphical depiction of the SAR baseline system performance parameters for the azimuth processor are given in Figure 33. It shows the relationship of the total beam integration time of 28 seconds to the radar PRF. The time-bandwidth product per look is 0.7 seconds times $500 \mathrm{~Hz}=350$. Using these parameters, a comparison was made of the azimuth processor memory size and computation rate for the three main processing algorithms as a function of the number of looks.


Figure 33
Baseline Azimuth Parameters

The results are given in Table 9. It shows results as generally expected with the subarray approach having the lowest cinputation rate and the time domain processor exhibiting the smallest memory. However, the computation rate of the time domain processor is much higher than the other approaches. This high computation rate translates into hardware as demonstrated in Section 2.2.2 and el iminates the time doma in afproach as a contender. The tradeoff between complexity of control and hardware is a factor between the FFT convolver and subarray approaches.

The 2-D convolution approaches have not been included since they are similar to the FFT convolver algorithm. In the approaches using 2-D convolution of range compressed data the question is of the relative advantage of 2-0 convolution for range migration compensation versus range interpolation. Simulation results presented in Section 2.3 indicate that a simple two point interpolator will give an adequate integrated sidelobe performance level. A four point interpolator is close to the ideal and its implementation in the selected design is not a computational burden.

Based upon its greater ease and completeness of programmability and
its lower cost for development, the FFT convolver algorithm has been selected for the ADSP recommended design. Further details of the design and selection issues are given in Section 2.5. The design and architecture have also lent themselves to the selection of the FFT convolver for the range processor.

Table 9
Look-Parameter Azimuth Processor Variations for Baseline SAR System

| 2 <br> LOOKS | FFIT <br> CONVOLVER |  | SUBARRAY |  | TIME DOMAIN |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | COMP <br> RATE | MEMORY | COMP <br> RATE | MEMCPY | COMP <br> RATE | MEMORY |
| 1 | 306 | 67.1 | 171 | 45.6 | 77,000 | 28.0 |
| 2 | 276 | 33.6 | 160 | 22.8 | 19,250 | 14.0 |
| 4 | 245 | 16.8 | 148 | 11.4 | 4,813 | 7.0 |
| 8 | 215 | 8.4 | 138 | 5.7 | 1,203 | 3.5 |

- COMPUTATION RATE IN MILLIONS OF COMPLEX OPERATIONS PER SECOND
- MEMORY STORAGE IN MILLIONS OF WORDS
- totals exclude range migration compensation


### 2.3 Performance Levels

The selection of a signal processing configuration for the ADSP required the determination and comparison of the relative performance of each candidate algorithm. The analysis/simulation of the algorithms is described in this section. The simulation results permitted cost versus performance tradeofis. A design requirement that was particularly important in the tradeoffs was the integrated sidelobe level (ISL). A design goal of -20 db ISL in two dimensions (range and azimuth) was established as a minimum requirement.

### 2.3.1 Performance Procedure

The organization of the simulation was broken into two parts. The first part is the generation of a representative time signal of interest. The second was the processing of the signal using two different algorithms, the FFT-convolver and the step transform subarray.

### 2.3.1.1 Generation of the time Signal

The objective of the first part of the simulation was to generate the time response of a point target. The simulated target was assumed to have already been matched filtered in the range dimension by FFT convolution. The range response was Hamming weighted and was evaluated exactly at the sampling points. If the target peaken at a nonsampled point, the response at the sampling points was generated re lative to the peak. The phase of the target was generated using

$$
\phi(t)=2 \pi V^{2} t^{2} / \pi R
$$

where $V$ is the velocity of the spacecraft (assumed to be a constant), $t$ is the relative sampling time, $\bar{N}$ is the wavelength of the transmitted pulse, and $R$ is the slant range to the point source.
The effect of the ranige migration was obtained by displacing the Hamming weighted function. Range migration is composed of linear and quadratic components. The linear range migration component is primarily due to the earth's rotation. The quadratic component varies in a nun-linear manner as a function of beam angle. Figure 34 shows the two components of the range migration as they were modeled in the


RANGE

Figure 34
Quadratic and Linear Range Migration
Linear range migration is simply a slope parameter in the simulation. The quadratic displacement is given by

$$
Q=\left(V^{2} t^{2} / 2 R\right) /(C / 2 B)
$$

where $V$ is the velocity of the spacecraft, $t$ is the relative time, $R$ is the slant range, $c$ is the velocity of light, and $B$ is the bandwidth of the system. Figure 35 shows the relationship between the quadratic range migration (35a) and the corresponding linear FM function (35b). The maximum range, Rm, in Figure 35a corresponds to the maximum and minimum frequencies $\left(+f_{m} / 2\right.$ and $\left.-f_{m} / 2\right)$ in Figure 35 b . The minimum range migration corresponds to the zero crossing point of Figure 35b. The signal is present along limited portions of the spectrum as a function of time and generates a linear FM (LFM) response. The generated signai migrates to different range cells as a function of time and beam angle.

Onosiv." incos
of fưR QuAnity

Figure 35a
RANGE R:' GRATION


Figure 35
Generation of the Time Signal

### 2.3.2 FFT - Convolver Algorithm

### 2.3.2.1 Range Migration Compensation

The FFT-Convolver SAR processing algorithm was simulated first. in this approach, the azimuth data is transformed to the frequency domain along constant range lines. In Figure 35a, the transform is performed along the horizontal lines. Since the energy of a point source is dispersed into different range lines, compensation must be provided prior to matched filtering in the azimuth direction. Figure 36 shows the fourier transform of $N$ range cells. The compensation for the range migration consists of moving the fourier coefficients into the proper range cells. The number of ran, migration cells is equal to

$$
\lambda^{2} R n^{2} / 8 V^{2} R_{c}
$$

where $\lambda$ is the wavelength of the transmitter, $R$ is the minimum slant range, $V$ is the velocity of the spacecraft, $R_{c}$ is the range cell width and $n$ is the Fourier coefficient number.

$\triangle$ - FsILENGTH OF THF. FFT
is -- SAMPLING FREQUENCY

- range walk per a - Const $\cdot(\Delta n)^{2}$

CONST $=\lambda^{2} \cdot r_{m} 18 V^{2} \cdot R_{C}$
$\frac{-N}{2} \leqslant n<\frac{N}{2}$
$R_{C}=$ RANGE CELL WIDTIT
$\lambda$ - WAVELENGTH OF RADAR FREQUENCY
$\mathrm{P}_{\mathrm{m}}$ - MINIMUM SLANT RANGE
$v$ - VELOCITY OF SPACECRAFT
N - FFT TRANSFORM SIZE

Figure 36
Range Migration

The amount of range walk as a function of frequency allows for the proper alignment of whe coefficients.

The simpliest adjustment to the coefficient would be to simply slide the data by integral steps (8). For example if the migration for a given coefficient were 5.4 range cells then the data would be obtained from the coefficient 5 range cells offset. For a migration of 6.7 range cells, the data would be obtained from 7 range cells array. This niethod, while simple, would tend to cause some ou'cput distortion.

Interpolation of the data shift eliminates the effects of the discontinuities of integral transfers. Interpolators of two, three and four points were simulated. Table 10 gives the equations of the interpolators used in the simulations.

Table 10
Interpolation Formulas

- $2 p t:(1-p) X_{0}+p X_{1}$
- $3 p \mathrm{p}: \frac{\mathrm{p}(\mathrm{p}-1)}{2} x_{0-1}+\left(1-p^{2}\right) x_{0}+\frac{p(p-1)}{2} x_{1}$
- $4 p t: \frac{-p(p-1)(p-2)}{6} x_{-1}+\frac{\left(p^{2}-1\right)(p-2)}{2} x_{0}+$
$\frac{-p(p+1)(p-2)}{2} x_{1}+\frac{p\left(p^{2}-1\right)}{6} x_{2}$


### 2.3.2.2 Azimuth Matched Filter

Following the range walk compensation, each of the range lines are matched filtered by multiplying the transform of the azimuth focusing function. The azimuth focus function has a linear FM slope of

$$
a=2 \pi v^{2} / R \lambda
$$

and iergth determined by the maximum integration line. The simulation has the option of applying a non-unifcrm weighting across the time response of the azimuth matched filter. The location of the time aperture of the azimuth matched filter depends upon which of the four looks is being processed.

### 2.3.2.3 Weighting

Figure 37 outlines the terms that are used in the discussion of the simulation and resulics. The waveform in Figure 37 represents the compressed response in both range and azimuch. The mainlobe of the response can be defined in a number of ways. The most comnon definitions use widths about the peak to the 3 db points, 6 db points or to the first nulls. Sidelobes are then defined as everything that does not lie within the mainlobe. The peak sidelobe level versus the peak mainlobe (PSL/PML) is defined as the largest value found outside the mainlobe divided by the largest value found within the mainlobe, usually expressed in db . The integrated sidelobes versus the integrated mainlobes (ISL/IML) is defined as the integration of the region not found in the mainlobe divided by the integration of the mainlobe region,
again usually expressed in 1 dB . The PSL/PML and the ISL/IML can be given for one or two dimensions. The one dimension (10) case uses a single range line containing the mainlobe peak. The two dimension case (2D) is the two dimensional plane of azimuth and range. :ising the definitions of figure 37, Table 11 gives a brief review of differe: : $\quad$.ighting functions. Weighting selection represents a compromise between resolution or width of the peak response and sidelobe levels. This is clearly seen in comparing the increase in the mainlobe width ( 3 dB and null columns of Table 11) to the decrease in peak sidelobe and integrated sidelobe levels (the columns labeled PSL/PML and ISL/IML). The results of Table 11 were obtained using a signal aperture of 64 samples and a total time history of 1024 samples (the rest of the signal was padded with zeros).


PSLIML -- PEAK SIDELOBE VERSUS THE MAINLOBE (dB)
ISLIIML -- INTEGRATED SIDELOBES VERSUS THE INTEGRATED MAINLOBE (dB)

- 1D -- ALONG AZIMUTH DIRECTION
- 2D -- ALONG AZIMUTH AND RANGE DIRECTION
- MAINLOBE -- POINTS LYING WITHIN THE FIRST NULLS FROM THE PEAK
- SIDELOBI: -- ALL POINTS NOT IN THE MAINLOBE
- 3dB -- WIDTH OF THE MAINLOBE AT THE 3dB POINTS
- 6dB -- WIDTH OF THE MAINLOBE AT THE 6dB POINTS
- Nillti -- WIDTH OF THE MABNLOBE TO THE FIRST NULLS

Figure 37
Sidelobe Definitions

Table 11
Ideal Characteristics of Weighting Functions
LENGTH: 64 FFT: 1,024

| TYPE | PSL/ML | ISL/IML | 3dB | NULL |
| :---: | :---: | :---: | :---: | :---: |
| RECTANGULAR | 13.26 | 9.68 | 1.0 | 2.2 |
| HAMMING | 42.48 | 34.42 | 1.5 | 4.5 |
| BARTLETI | 26.53 | 25.30 | 1.5 | 4.3 |
| HANNING | 31.50 | 32.91 | 1.7 | 4.5 |
| BLACKMAN | 58.14 | 58.89 | 1.9 | 6.6 |
| 25 dB TAYLOR | 24.34 | 18.96 | 1.2 | 3.0 |
| 30 dB TAYLOR | 30.98 | 23.55 | 1.3 | 3.4 |
| 35 dB TAYLOR | 35.93 | 27.69 | 1.3 | 3.7 |

-PSL/ML -- PEAK SIDELOBE TO MAINLOBE RATIO (dB)
-ISL/IML -- INTEGRATED SIDELOBES TO INTEGRATED MAINLOBE (dB)
-3 dB -- RELATIVE WIDTH OF MAINLOBE TO 3 dB POINTS
-NULL -- RELATIVE WIDTH OF MAINLOBE TO FIRST NULL POINTS

### 2.3.2.4 Time Oomain Output

The final processing of the signal in the azimuth FFT convolver is to apply the inverse Fourier transform to obtain the time domain response. In the simulation, the time domain output is not decimated as would be the case in an actual signal processor. This reduces the measurement error since the nondecimated output provides a higher output sampling rate.

### 2.3.2.5 Interpolation and Weighting Simulation Results

A number of simulation runs were performed to measure the peak sidelobe levels and the integrated sidelobe levels for different order interpolators. The system parameters used in the simulations were a constant spacecraft velocity of $7.45 \mathrm{~km} / \mathrm{sec}$, a slant range of 850 km , 22 MHz sampling rate in the range direction, a pulse repetition rate of 1.6 kHz and an azimuth coverage of 4096 pulses. The signal was placed in the middle of the scan and the scan was processed for four looks. The radar system pulse bandwidth in the range dimension assumed to be 19 MHz (over-sampling in the range direction by 1.16). The transmitter frequency was assumed to be 1250 MHz .

Table 12
FFT Convolver Interpolator Results

- CASE 1: NO INTERPOLATION

|  | 10 | 20 |  |
| :---: | :---: | :---: | :---: |
| L00K | [SL/IML | 15LIIML | PSL/ML |
| 1 | 26.3 | 12.4 | 19.8 |
| 2 | 27.8 | 12.6 | 19.6 |

- CASE 2: FOUR POINT INTERPOLATION

|  | 10 | 20 |  |
| :---: | :---: | :---: | :---: |
| L00K | ISLIMML | ISLIIML | PSLIML |
| 1 | 31.1 | 28.9 | 35.8 |
| 2 | 328 | 32.6 | 36.3 |

- MEASUREMENTS IN DB


## - HAWING WEIGHTIMG

Table 12 summarizes the performance obtained by incorporating a four point interpolator with range migration compensation. Since the signal was placed in the middle of the scan, look 3 has the same results as look 2, and look 4 the same as look 1. Looks 3 and 4 are thus not shown in Tabie 12. Table 12 shows that the ISL/IML performance is improved by the four point interpolator over no interpolator in one dimension only (column labeled '10') by only 5 dt at a 30 db level. The improvement is of the order of 20 db however when a two dimensional integration is examined (column labeled ' $2 D^{\prime}$ '). The peak sidelobe level versus the mainlobe occurs in the one dimensional line and is thus the same for either 10 or $2 D$ (column labeled 'PSL/PML'). The large degradation in the ISL/IML with no interpolator is expected. Shifting of the Fourier coefficients without interpolation results in discontinuities that raise the level of the sidelobes in the range cells surrounding the signal.

The results in Table 12 were obtained using a Hamming weighting function across the azimuth matched filter. Table 13 illustrates the reason for using a weighting function to reduce the sidelobe levels at the expense of a reduction in the resolution in a simulation. If no weighting were employed, the results would be ambigious on the need for an interpolator in the processor. The column labeled 'Rect' shows
only a 1.5 to 2.0 improvement for the interpolated case. What has happened is that the high sidelobes of the rectangular weighting function (window) have contributed to the integrated sidelobe. Using a window with greater sidelobe suppression provides results which are sensitive to processing differences. A triangular window ( column labeled 'Triangle') suppresses the PSL/PML level of the window to 27 db from 13 db for a rectangular window and also the falloff of the sidelobes is at a $12 \mathrm{db} / o c t a v e ~ r a t e ~ v e r s u s ~ 6 ~ d b / o c t a v e ~ f o r ~ r e c t a n g u l a r . ~$ The mainlobe increases by approximately $40 \%$; however, the difference due to processing can now be seen. The ISL/IML difference between no interpolator and a four point interpolator is approximately 12 db . Using Hamming window which has a PSL/ML of 43 db , a falloff rate of $6 \mathrm{db} / o c t a v e$ and an ISL/IML. of 34.42 db , the processing difference is seen to increase to approximately 20 db (column labeled 'Hamming'). Considering that the theoretical ISL/ML for a rectangular window is 9.68 db and a PSL/ML of 13.26 db , the effects being seen in the "rect" column for the no-interpolation case are largely due to processing (the ISL/IML is approximately 8 db ), while for the four point interpolator it is the window (the ISL/IML is 9.7 db ). On the other hand, both processor's performance is being measured when the Hamming window is used.

Table 13
Effect of Weighting Functions, FFT Convolver

- CASE 1: NO INTERPOLATION

|  | LOOK | RECT | IRIANGLE | HAMMING |
| :---: | :---: | :---: | :---: | :---: |
| ISL/IMLL (PSL/ML) | 1 | $7.8(13.51$ | 12.4119 .81 | 12.4 (19.8) |
|  | 2 | 8.1112 .81 | - | 12.7 (19.6) |
| AZIMUTH RESOLUTION |  | 1 |  | 1.4 |
| RANGE RESOLUTION |  |  | TANT |  |

- CASE 2: FOUR POINT INTERPOLATION

|  | L00K | RECT | TRIANGLE | HAMMONG |
| :---: | :---: | :---: | :---: | :---: |
| ISL/DML (PSL/ML) | 1 | 9.8 (13.5) | 24.0126 .71 | 29.1 (35.8) |
|  | 2 | 9.7 (13.2) | 25.1 (27.0) | 32.8 (36.3) |
| AZIMUTH RESOLUTION |  | 1 | 1.4 | 1.4 |
| RANGE RESOLUTION |  |  | TANT |  |

### 2.3.3 Step Transform - Subarray Approach

### 2.3.3.1 Description of Subarray Simulation

The second algorithm simulated was the step transform-subarray. The first step in the process after range compression is the derinping of the complex time signal by the azimuth reference LFM. The deramped signal is then sectioned into subarrays. The number of samples in each subarray and the amount of overlap between the subarrays were systems parameters examined in the simulation. An example of a subarray length of 64 with 2 to 1 overlap is depicted in Figure 38.


- Example
- Length of subarray -- 64
- OVERLAP (112) -- 32
- SUBARRAYS
- S1 (1-64)
- 52 (33-96)
- 53 (65-128)
- Si (97-160)

Figure 38
Subarray Formation
After forming the subarrays, each subarray is transformed into the frequency domain using an FFT. The sudarrays are then phase corrected using a correction factor of exp $\left(-j 2 \pi_{k d / N}\right)$ where $k$ is the FFT coefficient number, $d$ is the displacement between subarrays and $N$ is the length of the FFT. The original signal has been transformed into a three dimensional array with the Fourier coefficients on one axis, the subarray number on the second axis, and the range cell number on the third. Figure 39 shows a two dimensional slice of the
three dimensional array with the range dimension fixed and the FFT length 64.

- COMPUTE THE FIRST FFT (EXAMPLE 64 LENGTH)



SUBARRAY
lst COEFF -- DC TERM
2nd COEFF -- $\Delta$
3rd COEFF -- 2
33rd COEFF -- T/2
$\Delta$ - is/LENGTH OF FFT
is - $1 / \mathrm{T}$ - SAMPLING FREQUENCY

Figure
39
First FFT in Subarray Approach

Assuming that the signal depicted in Figure 39 has no range migration, then the deramped subarrayed signal would appear entirely in the plane. If for example, the difference between the deramping reference and the signal was exactly $3 \Delta$ and the signal began at the 256 sample, then the fifth subarray would be the firsi to see the signal and its transform would have the fourth Fourier coefficient as the highest of the 64 coefficients (assuming no other signals are present). The fourth coefficient would be high for each subarray in which the signal was present. In the case where the signal exactly matches the reference, the first (DC coefficient) would be the highest of the 64 .

However, the signal does not stay within a single range cell. The range migration traverses a quadratic function in the third dimension. Consider three signals, one that exactly matches the reference, one that is delayed in time by an amount which corresponds to the signal peaking in the 2nd Fourier coefficient and the third is delayed by an amount
equal to the third coefficient. These three signals are depicted in Figure 40a as $0, \Delta T$ and $2 \Delta T$ delay from zero time. Assuming that the first signal begins at the first sample, then the first coefficient of the first array would have to be range corrected the maximum amount. The first coefficient of the 2nd subarray would be corrected according to the range migration correction formula used in the FFTConvolver approach. The amount of range migration and correction would diminish as the higher number subarrays are addressed until the subarray corresponding to the zero time of Figure 40a is addressed. This point is shown as the first point on the zero correction line of Figure 40 b . The correction would then increase to the maximum. Singe the signal was assumed to be matched to the reference, then all of the correction is applied to the first Fourier coefficient.
target spacing of jtin same range cell

Figure 40a
Migration of Target Across Subarray


Figure 40b Deramp and FFT the Subarrays


Figure 40
Subarray Range Migration Correction
The second signal in Figure 40a would first appear at a subarray number corresponding to the time delay $\Delta T$. Once the signal starts, the correction function would proceed from the maximum amount down to zero and back up to the maximum in the same quadratic shape as the
first signal. The only difference is that the correction is performed on the 2nd FFT coefficient. The third signal would have the same correction performed on the 3rd coefficient. Figure 40b depicts the correction across the subarrays as the straight lines labeled "contour of maximum correction". "zero correction", and "maximum correction". The slope of the parallel lines would be given by

$$
\text { slope }=\left(\begin{array}{ll}
f s^{2} & T
\end{array}\right) /(N \Delta L F M d)
$$

where $f$ is the azimuth sampling frequency, $N$ is the subarray length, $T$ is the total time that the signal is present, $\Delta$ LFM is the total change in the LFM reference and $d$ is the number of points that the subarrays overlap. The correction scheme used for the range walk is no different than that used in the FFT-Convolver algorithm. The data is shifted in integral amounts for no interpolation and is interpolated between points otherwise.

The final part of the processing in the subarray approach is to perform the focusing FFT. The same slope that is used in the range migration correction is used to address the subarrays for the focusing FFT. In the case of a non-integral slope, the nearest subarray is chosen. The signal is then padded with zeros which minimizes the measurement errors of the analysis. The FFT is computed and the proper coefficients are selected from the FFT. Each set of coefficient are abutted to form the output. The selection of the proper cuefficients is analogous to a comb filter bank where each output filter is tuned to only certain center frequencies and a given bandwidth about them. Figure 41 depicts the process of selecting the output values.


OF TE RMINATION OF VALIO CUEFIICIENIS OF FOCUSING FFT(:-)

-d a namovirlapping displacementi

- $\Delta F^{\prime}=\left(1 / 4 / w^{\prime} /\right.$ if he ouency oisplacement of focusing efti)
- \# = $\mathrm{NH}^{\prime} / \mathrm{N}$

Figure 41
Coefficient Selection

### 2.3.3.2 Subarray Simulation Results

The first issue to be examined in the step transform subarray algorithm is the amount of overlap required between successive subarrays. The results of the simulation of two overlap cases is given in Table 14. Case 1 is an overlap of 4 to 1 and case 2 is an overlap of 2 to 1. That is, case 1 represents a subarray spacing of one fourth of the subarray length and case 2 , a subarray spacing of one half its length.
The column labeled 'COEFF' is the FFT coetricient output by the focusing FFT (second FFT). The parameters for the second or focusing FFT were chosen so that both cases had the same number. The 33rd coefficient in column 'COEFF' corresponds to the DC component of the good coefficients and the 65 th the last valid coefficient. These two coefficients represent the extremes with the other 32 coefficients lying between. As can be seen from the 'ISL/IML' columns, the greater the overlap or subarray sampling rate the better the performance. Increasing the overlap increases the spacing between the main subarray lobe and its grating lobe caused by subarray sampling. However, higher overlap ratios also increase the required amount of computational effort.

Averaging over the 64 coefficients for case 2, the average performance is found to be 25.6 db for 2 D ISL/IML which is well below the 20 db required.

Table 14
Subarray Overlap

| CASE | COEFF | 10 |  | 20 |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | ISLIIML | PSLIML | 151.11ML | PSLIMI |
| 1 | 65 | 32.0 | 40.7 | 31.9 | 40.8 |
|  | 33 | 37.0 | 42.5 | 37.0 | 42.5 |
| 2 | 65 | 21.9 | 19.0 | 18.9 | 16.7 |
|  | 33 | 32.1 | 37.1 | 31.3 | 37.1 |


|  |  | CASE 1 | CASE 2 |
| :--- | :--- | ---: | ---: |
| 1st fft | - | 64 | 64 |
| SUBARRAY | - | 16 | 32 |
| SPACING |  |  |  |
| SUBARRAYS | - | 64 | 32 |
| 2nd FFT | - | 256 | 128 |

-BOTH FFTS HAMMING WEIGHTING -LOOK 2
-AVERAGE ISLIIML FOR CASE 2 OVER 64 COEFICIENTS 25.6 dB

The offects of the grating lobes on the subarrays can also be reduced by a proper choice of the weighting function across the first FFT. The window is chosen to place a null near the grating lobe. Tabie 15 shows that a 30 dB Taylor weighting will meet the requirements. The table also shows that optimum performance with the first FFT weighting function is achieved by a careful balance between mainlobe width, which affects the anplitude of the output grating lobe, and sideiobe level which sets overall ISL/IML.

Table 15
Effect of Weighting of First FFT in Subarray Process

|  |  | 20 |  |
| :---: | :---: | :---: | :---: |
| TYPE | COEF'F | 15L/IML | PSLIML |
| HAMMIING | 65 | 18.9 | 16.7 |
|  | 64 | 18.3 | 17.3 |
|  | 34 | 31.2 | 37.1 |
|  | 33 | 31.3 | 37.1 |
| 25 dB TAYLOR | 65 | 15.5 | 22.9 |
|  | 64 | 15.6 | 2.1 |
|  | 34 | 34.3 | 41.5 |
|  | 33 | 34.8 | 41.9 |
| 30dB TAYLOR | 65 | 21.3 | 24.0 |
|  | 64 | 21.4 | 23.6 |
|  | 34 | 32.7 | 38.4 |
|  | 33 | 32.7 | 38.1 |

## " 30 dB TAYLOR WEIGHTING WILL MEET THE REQUIREMENTS

### 2.3.4 Ferformance Comparisons

Having chosen the parameters wich will meet the requirements for the different algorithm the relative performance between algorithms is shown in Table 16 . Table 16 shows the performance with and without the addition of linear range migration to the quadratic range migration. The effects of interpolator complexity is also summarized in Table 16. For the FFT/Convolver approach a 2-point interpolator would seem to be adequate since 13 db of 18 db of the processing improvement compared to no interpolator has been obtained. For the subarray approach a 3 -point interpolator appears adequate. The integrated sidelobe values in Table 16 for the subarray system have been averaged over all valid coefficients out of the focusing FFT. Table 17 shows that doubling the amount of linear range migration does not affect the performance for the FFT convolver algorithms.

The reason that the subarray case in Table 16 with linear migration and no interpolator provides better performance than the quadratic only case is that the linear migration tends to offset the quadratic in look 2.

Table 16
Linear Range Migration (24 cells)

- COMPUTATION OF ISLIIML WITH LINEAR RANGE MIGRATION OF 24 CELLS ADDED TO QUADRATIC (7 CELLS)

|  | LINEAR <br> MIGRATION | INTERPOLATOR POINTS |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 0 | 2 | 3 | 4 |
| FFT CONVOLVER | NO | 12.4 | 28.3 | 29.9 | 32.1 |
|  | YES | 12.4 | 25.8 | 27.6 | 30.5 |
| SUBARRAY | NO | 12.5 | - | 24.6 | 25.5 |
|  | YES | 13.6 | 19.1 | 24.4 | 25.4 |

DATA TAKEN FOR LOOK 2

Table 17
Linear Range Migration ( 40 cells)

- ISLIIML WITH A LINEAR RANGE MIGRATION OF 40 CELLS ADDED TO QUADRATIC (7 CELLS)
- FFT/CONVOLVER APPROACH

| $\begin{aligned} & \text { INTERPOLATION } \\ & \text { POINTS } \\ & \hline \end{aligned}$ | LOOK | $\begin{gathered} 2 \mathrm{D} \\ \text { ISLIIML (PSL) } \\ \hline \end{gathered}$ |
| :---: | :---: | :---: |
| 0 | 1 | -12.4 (19.7) |
|  | 2 | -12.4 (19.7) |
|  | 3 | -12.4 (19.8) |
|  | 4 | -12.4 (19.7) |
| 2 | 1 | -25.6 (30.7) |
|  | 2 | -25.8 (31.3) |
|  | 3 | -25.8 (31.4) |
|  | 4 | -24.8 (29.5) |
| 4 | 1 | -29.9 (37.7) |
|  | 2 | -30.5 (38.5) |
|  | 3 | -30.5 (37.3) |
|  | 4 | -28.2 (35.1) |
|  | 71 |  |

### 2.4 Technology Survey

Technology is an important consideration in the design of the ADSP. The three principal areas designated; digital integrated circuits, digital architecture and software encompass the major design issues related to the system. This section summarizes the stuciy, evaluation and projection of these technology issues for the ADSP and future SAR systems.

### 2.4.1 Digital Integrated Circuits

### 2.4.1.1 Survey of Integrated Circuit (IC) Manufacturers

Keys to the determination of IC technology available for the ADSiF are the current and future plans of semiconductor manufacturers. An industry survey was conducted to determine plans and projections in the 1983-85 and post-1985 time frame. The companies included were:

| Advanced Micro Devices | Ravtheon Semiconductor |
| :--- | :--- |
| Fairchild Semiconductor | RCA Solid State |
| American Microsystems | Signetics |
| Intel | Texas Instruments |
| Intersil | Harris Semiconductor |
| Motorola | Synertek |
| Mostek | Monolithic Memories |
| National Semiconductor |  |

Information relative to technologies, memory size and configuration, LSI-VLSI logic functions, speed, power and cost were requested for the 1983-85 time frame. For post 1985 projections on technologies, line widths, gate density, speed, and power were requested with specific pro,jections on memories, arithmetic functions and microprocessors.

Several of the companies surveyed declined to respond due to company policies on the release of long rarige plans. A summary of the results is contained in Tahle 18-20 and includes current technology, firm plans -1983-1985, projections 1983-85, and post 1985 projections. Random access memc.iss, EPROMS/PROMS/ROMS, and IC logic functions are included.

### 2.4.1.2 IC Technology Trends

The IC technologies in current use span a large gamut of semiconductors. These include the high speed emitter coupled logic (ECL) line to the low power, relatively low speed complementary metal oxide semiconductors (CMOS) and integrated injection logic ( $I^{2} \mathrm{~L}$ ). Bipolar transistor-transistor logic (TTL) with Schottky and low power Schottky (STTL, LSTTL) versions are the dominant semiconductor technologies now used in digital logic. N-channel MOS (NMOS) is the dominant large memory technology and silicon on sapphire (SOS) is used primarily for specialized military and space applications.

As digital logic moves to shorter channel lengths we can expect changes in the technologies used. Although we can expect the same technologies to be in general use through 1985 we will see a definite shift toward narrow channel CMOS replacing TTL and ECL because of its iow power, with high speed, capability.

In the post 1985 time frame the principal technologies will be NMOS and CMOS with CMOS/SOS possibly moving into some general conmercial application.

There is general agreement that CMOS will have emerged as the prime digital semiconductor technology because of its inherent low power dissipation. As line dimensions of CMOS circuits decrease, the speed also increases. CMOS sircuits have been fabricated with 2 micron dimensions giving average gate delays of 1 nsec . When poojecting up to 50,000 or more gates per chip, the power dissipation per gate must obviously be low to prevent thermal breakdown. At line dimensions of 1 to 2 microns and less the speed and power advantages of CMOS/SOS over CMOS are diminished, particularly when using oxideisolated CMOS technology.

Although CMOS has been indicated to be a post 1985 technology, it could happen much faster than that. There is a growing realization of the speed-power advantage of advanced CMOS in the industry-and increased investment. When the cost of the CMOS circuits meets the NMOS ?chnology costs, it will definitely be the choice because of its lower power which simplifies ine power supply design, cooling
requirements and packaging system.
The impact of the post 1985 technology on SAR processing will be in the feasibility of on-board processing provided by low power, dense memories, and VLSI with low power per gate provided by CMOS.

Gallium arsenide (GaAs) can also be expected to emerge as the high speed technology in the post 1985 time frame. Research efforts with laAs have produced $G H z$ functional logic eiements and $A / D$ converters.

### 2.4.1.3 Random Access Memories

Random access memories have been broken down into dynamic and static categories. It can be noted that for the ADSP, static RAM's are preferable since the synchronous nature of the input and output data is more easily controlled with static RAM storage. In general, dynamic RAMs are less costly on a per-bit basis. Current static RAM's are much faster, consume more power and, in accordance with general trends, cost more per-bit for the higher speed capability.

Table 18
Randon: Access Memories

|  |  | CURRENT TECHNOLOGY | $\begin{gathered} \text { 1983-1985 } \\ \hline \end{gathered}$ | $\begin{gathered} \text { 1983-1985 } \\ \text { PROJECTIONS } \\ \hline \end{gathered}$ | $\begin{aligned} & \text { POST } 1985 \\ & \text { PROJECTIONS } \\ & \hline \end{aligned}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| DYNAMIC | SIze | T0 64K | 64 K | 128K-512K | 64K-1M |
|  | COnfiguration | xl | X8, x4, xI | x8, x16 | - |
|  | ACCESS (nsec) | 100-300 | 200 | 50-200 | 50-100 |
|  | POWER (mW) (ACTIVE/STANDBY) | 300120 | 300120 | - | - |
|  | COST/BIT ( ${ }^{\text {( })}$ | 0.1 | 0.04-0.1 | 0.02-0.1 | - |
| STATIC | SIZE | TO 16K | 16K-64K | $64 \mathrm{~K}-256 \mathrm{~K}$ | 64K-1M |
|  | CONFIGURATION | xl, x4, x8 | x8, x4, x1 | x8, x16 | - |
|  | ACCESS (nsec) | 50-100 | 25-200 | 50-200 | 50-100 |
|  | POWER | 6001100 | 200-1.000 | - | - |
|  | COST/BIT (1) | 0.1-0.4 | 0.1-0.3 | 0.02-0.1 | - |

The 1983-85 projections generally continue to follow the "Moore curve" of memory development first observed by Gordon Moore of Intel Corporation. It shows memory capacity per chip doubling every year. Perhaps more significantly; the cost per bit is projected to drop significantly, at least by some companies, in this time frame. In addition, the memory size firm committments of manufacturers is probably conservative. Figure 42 shows the state of the art for developmental memory circuits which is currently (mid 1981) 256 K bits for bulk CMOS. There is about a two year delay between development and commercial introduction. Based on the Figure 42 curve, we should see megabit dynamic memories on the market in 1985.


Figure 42
Memory Chip Capacity Trends

Memory cost trends are shown in Figure 43. Both chip costs and installed memory costs are included. Chip costs should be close to the 0.01 cents per bit level in 1985. This translates to about $\$ 50,000$ for the chip costs of the corner turnirig memory of the ADSP processor in the 1985 time frame.


Figure 43
Memory Costs/Bit Trends
2.4.1.4 EPROMS, PROMS, ROMS

An essential part of the ADSP processor are the programmable aemories used in the control firmware of the processor. Projection of circuit capabilities for erasable programmable read only memories (EPROMS), programmable read only memories (PROMS) and mask programmable read only memories (ROMS) are given in Table 19.

Table 19
EPROMS, PROMS, ROMS

|  |  | CURRENT TECH NOLOGY | $\begin{gathered} \text { 1983-1985 } \\ \text { FIRM PLANS } \\ \hline \end{gathered}$ | $\begin{gathered} \text { 2983-1985 } \\ \text { PROJECTIONS } \\ \hline \end{gathered}$ | POST 1985 PROJECTIONS |
| :---: | :---: | :---: | :---: | :---: | :---: |
| EPROMS | CONFIGURATION | 2k 4K 8M WORDS'BY 2. 488 BitsI <br> WORD | 32K, 64K |  | TO 1M |
|  | ACCESS (nsec) | 350-600 | 200 |  |  |
|  | COST/BIT (Q) | 0.2-0.4 | 0.1-0.3 |  |  |
| PROMS | CONFIGURATION | 1 K 2 K WORDS <br> BY 48 BITS PER WORD | T0 $8 \mathrm{~K} \times 8$ |  | TO 1 M |
|  | ACCESS (nsec) | 45-100 | 50 |  |  |
|  | COST:BIT (Q) | 0.3 | 0.1-0.3 |  |  |
| ROMS | SIZE | TO 64K | $64 \mathrm{~K}-128 \mathrm{~K}$ | 512K | TO 1 m |

EPROMS are fabricated with MOS technology and although there are a wide variety of configurations available, they are generally too slow for convenient application to the ADSP. PROMs on the other hand, are made with bipolar technology and are much higher in speed although not as large. However, the maximum size PROM, 16K in up to a $2 \mathrm{~K} \times 8$ configuration is applicable to an FFT signal processor.

A mask programmable read-only memory chip (ROM) may be applicable to the ADSP. The size and speed of ROMs of up to 64 K are applicable, and if the set-up charges are sufficiently low to be offset with the quantities required in the ADSP it is the preferred direction.

### 2.4.1.5 IC Logic Functions

IC logic function projections are summarized in Table 20. Particularly noteworthy to the ADSP (and all other signal processing appicaticn) is the expected emergence of various "FFT chips" in the 1980's. These circuits will make the principal computation requirement in the ADSP much less costly to meet.

Table 20

## IC Logic Functions

| CURRENT <br> TECHNOLOGY | $\begin{aligned} & \text { F1RMP3-1985 } \\ & \text { FIRUNS } \end{aligned}$ | $\begin{aligned} & \text { 1983-1985 } \\ & \text { PROJECTIONS } \end{aligned}$ | POST 1985 PROIETIONS <br> PROJECTIONS |
| :---: | :---: | :---: | :---: |
| ARITHMETIC | - 87016 BIT MULTIPLIERS | - $16 T 03281 T$ MICROPROCESSORS | - 32 BIT MULTIPLIERS AND MICRO- |
| MULTIPLIERS |  |  | PROCFSSOR, 20 MHz |
| $\begin{aligned} & 8 \times 8,50 \mathrm{nsec} \\ & 16 \times 16,100 \mathrm{nsec} \end{aligned}$ | - frt CHIP | - 10,000 GATES I CHIP | - FTT PIAO:ESSOR ON CHIP |
| ALUs | $\begin{aligned} & \text { - 1,000-10,000 } \\ & \text { GATESICHIP } \end{aligned}$ |  | - ${ }^{\text {CHIP }}$ 50,000 GATES। |
| (1981) 16 BIT FLOATING POINT AND PROGRAMMABLE ALUS |  |  |  |
| OTHER |  |  |  |
| 16 BIT ERROR DETECTION AND CORRECTION |  |  |  |
| - 8 bit Comparator |  |  |  |
| - programmable logic ARRAYS |  |  |  |

Most of the new logic circuit development in recent years has been directed toward microprocessor development. High speed LSI arithmetic circuits applicable to signal processors have not been developed to the extent possible. TRW has marketed a number of LSI arithmetic components which meet signal processing needs. Chief among these are a line of parallel multipliers of up to 24 bits. The $16 \times 16$ bit multiplier has a speed oi about 100 nsec. TRW is also planning to release a floating point adder circuit by early 1982 which could be a key component in the ADSP.

Another circuit of special interest to the ADSP is an error detectioncorrection circuit. Because of the large memory in the ADSP, it may be desirable to employ error detection and correction to increase reliability.

### 2.4.1.6 VHSIC

Perhaps the most significant development effort now underway for the future of signal processing technology is the Department of Defense (DOD) VHSIC program - for very high speed integrated circuits.

ORIGINAL PAGE IS OF POOR QUALITY

The product of the number of gates on an IC and the clock frequency is a figure of merit for integrated digital logic that is particularly useful in assessing computing or signal processing power. The long term goals of the DOD VHSIC program is to achieve a gates-times-clock rate of $3 \times 10^{12}$. This goal is compared to a number of advanced technologies in Figure 44a. All of the technologies shown are or were candidates for achieving the VHSIC goals.


Figure 44 a
Comparison of New Technologies on Clock Rate - Gate Basis

The VHSIC program objectives inciude a shrinkage of the circuit geometric dimensions from 5 microns to 1.5 microns over 3 years and to 0.5 microns over 6 years. The scaling theory for circuit performance indicates that this size shrinkage will provide lower power-delay products. For example, in CMOS circuits, the basic geometric performance parameter is the transistor channel length,
$L_{c}$. With power dissipation proportional to $L_{c}{ }^{2}$ and the delay to $1 / L_{c}$, the power-delay product is thus proportional to $L_{c}$.
The projected VHSIC circuit capabilities will permit chip implementation of advanced signal processing functions un a chip. All contractors selected for the VHSIC Phase I programs have objectives which are oriented toward signal processing.

### 2.4.2 Digital Architecture

An overview of digital architecture approaches was developed in relation to the functional characteristics of the ADSP. These functional characteristics are summarized in Table 21 for the range and azimuth processing functions.

Table 21
ADSP Functional Characteristics

|  | RANGE PROCESSOR | AZIMUTH PROCESSOR |
| :---: | :---: | :---: |
| natural partitioning | - radar PrI <br> - PULSELENGTH | - RANGE CEL 1 LOOKS storage range migration span |
| PROGRAMMABILIITY | - BANDWIDTH <br> - TW PRODUCT <br> - WAVEFORM <br> - WEIGHTING <br> - RANGE SWATH | - PRI AZIMUTH COMPRESSION - RANGE MIGRATION CLUTIER LOCK FOCUS |
| BULK Storage | - none required | $a$ RANGE SWATH $\times$ AZJMUTH $\times$ COMPRESSION XLOOKS x OVERLAP FACTOR |
| MAJOR FUNCTIONS | 0 FFT <br> - COMPLEX MULTIPLY <br> - Amplitude weighting <br> - INIERPOLATION | - range processor functions plus: <br> - corner turning <br> - range migration compensation <br> - decimation <br> - magnitude, scaling <br> - integration <br> - doppler (clumer) estimate <br> - Auto-focus |

The first functional feature, natural partitioning, refers to partitioning approaches based on the synthetic aperture radar processing algorithms or the inherent radar operation. For example, a natural partition in the range processor is the radar PRI (pulse repetition
interval) during which a single radar pulse is transmitted and its reflected signals from the range swath coverage are received.

Programmability must be incorporated for the range and azimuth processes as indicated in Table 21. A major requirement is bulk storage which occurs primarily in the azimuth processor. The size of the bulk store, required for range-azimuth corner turning, is determined by the product of the range cells and pulses coherently integrated with the number of looks and signal overlap also involved. It is also dependent upon the algerithm employed.

The major finctions listed in Table 21 constitute the well defined functions for range and azimuth SAR processing. A programmable processor capable of performing all of these functions will by its nature have functional capabilities not listed. Modifications in processing algorithms should therefore be possible.

Several basic architecture approaches were investigated for the ADSㅇ: pipeline, parailel, SIMU (single instruction stream-multiple data stream), cross-bar, and various multi-bus architectures.

### 2.4.2.1 Fipeline Processor

A pipeline processor using the fFT convolver processing algorithm is shown in Figure 44b. It has the primary feature of minimizing hardware both memory and arithnetic processing elements. All elements of the pipeline operate simultaneously and the control is fairly straight forward. It can be programmed for changes in the waveforms or focusing function via the spectral reference functions and the FFT size. The arithmetic and memory can be scaled to match the word size as the signals progress through the pipeline. However, this approach is not usually followed since net cost reductions are generally realized if fewer design variations are used. Techniques for reliability enhancement include providing for interchangeable pipeline segments and the incorporation of extra FFT stages in the pipeline which can be switched into use when a failure occurs at any particular stage. Growth to higher capacity systems must be done by the addition of complete, parallel pipeline systems.


- MEMORY: CTM + RMC + 19K
- fft STAGES: 46
- PROGRAMMABILITY: COMPRESSION RATES, PIPELINE ADJUSTMENTS
- reliability enhancement: switched ffts or fft stages
- modularity: parallel systems

Figure 44b
Example ADSP Pipeline Processor (FFT Convolver;

### 2.4.2.2 Parallel Processor

An approach to implementing the ADSP with parallel processors is shown in Figure 45. In this case, a total of 52 programmable FFT processing units are used to compute the required FFT computations. In the range correlator, each unit is capable of processing a complete single pulse range scan. Successive pulses are inputted to successive processors and the total number is set to operate on the total processing load at an internal clock rate of 8 MHz . The azimuth correlator is different in that each parallel unit handles a fixed segment of range cells. This permits the bulk corner turning memory to be divided equally and allocated to the respective parallel azimuth processors. An FFT convolver function is inherent in the azimuth correlator as in the range correlator, but because of the spectral domain range migration compensation, the forward and inverse FFT's are separated. The range migration compensation unit is then a special processor in between the forward and inverse FFT. The advantage of the parallel processor is in the standardization of
hardware and high reliability. The current thrust of Lsi technology permits implementation of a FFT processor at reasonable cost levels, which when constructed in quantity will further reduce the unit cost. Spare units can be employed to enhance reliability.


- MEMORY: CTM + 15RMC + 238 K
- PROCESSOR UNITS: 52
- PROGRANMABILITY: AT PROCESSING UNIT LEVEL
- RELIABILITY ENHANCEMENT: SPARE PARALLEL PROCESSORS
- mOdULARITY: AdDITIONAL PROCESSORS

Figure 45
Example ADSP Parallel Processor (FFT Convolver)

### 2.4.2.3 Single Instruction - Multiple Data (SIMD)

The SIMD approach shown in Figure 46 is another parallel processor approach with an important difference. All of the procossors operate in-step on the entire processing algorithm. This approach offers a common control approach, but it is not good for the ADSP. Since each unit must do the entire process, it must have a full corner turning memory or at least a significant portion of the range swath. Thus, the total memory requirements of the system are unreasonable.


- MEMORY: - 25 CTM
- PROCESSING UNITS: ~ 50
- PROGRAMMABILITY: COMMON CONTROL
- reliability enhancement: parallel units
- MODILLARITY: ADDITIONAL PROCESSORS

Figure 46
SIMD (Single Instruction - Multiple Data Stream)

## ?.4.2.4 Cross-bar

Criss-bar architecture has long been used in high performance digital processors. This approach, shown in Figure 47, can be used to form a programmable pipeline signal processor. It has particular advantages in cases where several different processing units may be used in a varying sequence. The need for a variation of processors would arise in a signal processing application where computationally intensive algorithms must be performed. It may then be cost effective to construct special purpose processors for these functions. The number and types of processors might vary for each installation. A technical problem with the cross-bar approach is the design of the cross-bar switch itself. Several methods have been suggested in the literature for constructing modified variations of the cross-bar. These approaches are aimed at matching the real switching needs with an eificient switching mechanism, generally a multi-level switch matrix not unlike an FFT flow diagram. However, all of the proposed switching schemes do not provide the flexibility of a cross-bar and thus limit the overall efficiency of a general purpose processor.


- mendary x-ctm



madhanify: paralla svstems
Figure 47
Cross.llar Architecture


### 2.4.2.5 Multi-bus Architecture

A muilti-bus architecture can be used and Figure 48 does not do justice to the many variations which are possible. Recent efforts in this area have been concentrated on heirarchical schemes which partition the hardware structure into various levels of operational control. The control problems of this approach seem to be overly complicated for a signal processing system as essentially well defined as the ADSP.

STRUCTURE -- VARIABLE -- 2, 3 DIMENSIONS


- MEMORY: > MINIMUM STORAGE
- PROCESSING UNITS: > SO
- PROGRAMMABILITY: SEOUENCE, SIGNAL PATH CONTROL
- reliability enhancement: adaptive functional paths
- MODULARITY: ADOITIONAL PROCESSORS

Figure 48
Multi-bus Architecture

2.4.2.6 Programmable Signal Processor DevelopmentsSeveral government-industry efforts to build high performance,programmable signal processors are underway. These developments
are not near the mature stage where one can buy a system. A briefsummary of these developments follows.
Advanced On-Board Signal Processor (AOSP)
AGENCY: DARPA
CONTRACTOR: RAYTHEON
OBJECTIVE: DEVELOP SIGNAL PROCESSING SYSTEM FOR FUTURE SPACE
RADAR, EL[CTRO-OPTIC AND COMMUNICATION SYSTEMS CAPABLE OF
MEETING ON-BOARD SIZE, WEIGHT AND RELIABILITY REQUIREMENTS
ARCHITECTURE: MULTIPLE BUSSES INTERCONNECTING ARRAY COMPUTING
ELEMENTS (ACEs)
CLOCK RATE: $10-40 \mathrm{MHz}$
MEMORY/ACE: > IOM BITS
GATES/ACE: > 100K
ACES/ADSP: 56
STATUS: STUDY/SIZING/SIMULATION/SOME BREADBOARDS UNDERWAY
S-1 Super Computer
DESIGN: LAWRENCE LIVERMORE LABORATORY
SUPPORT: OFFICE OF NAVAL RESEARCH
OBJECTIVE: TO IESIGN AN ADVANCED VECTOR PROCESSING
ARCHITECTURE: CROSS-BAR CONNECTING MEMORIES WITH 16 HIGH SPEED
PROCESSING UNITS (A-BOXES)
CLOCK RATE: ..... 57 MHz
COMPUTATION RATE: $430 \mu \mathrm{sec} / 4096$ POINT FFT PER A-BOX
A-BOXES/ADSP: ..... 10
STATUS: UNDER DEVELOPMENT
Enhanced Modular Signal Processor (EMSP)
AGENCY: NAVAL SHIP SYSTEMS COMMAND
OBJECTIVE: TO DEVELOP AN ADVANEED PROGRAMMABLE SIGNAL PROCESSSOR
FOR NAVY APPLICATIONS TO REPLACE THE AN/UYS-I
ARCHITECTURE: MODIFIEO CROSS-BAR


COMPUTATION RATE: 80 M (REAL MULTIPIESi/sec
EMSPs/ADSP: $\sim 25$
STATUS: STUDY PHASE
Other processors such as the MPP - Massively Parallel Processor (NASA-Goodyear) and MRP - Multi-mode Radar Processor (Hughes) are in various stages of development. The question that must be answered is whether any of these developments are practically upplicable to the ADSP.

The specialized needs of the ADSP can be best met with a processor matched to its requirements. This will also limit long term life cycle costs particularly in the area of mission software development.

### 2.4.3 Software

A study was completed of high order languages (HOLs) suitable for the executive and control software tasks within the ADSP host computer. The languages studied were Fortran, Pascal. SPL/I and Ada. Any one of the candidate languages is adequate for the task, but some present advantages over the others. It is desirable for documentation purposes and ease of understanding to have as much of the control software as possible written in the HOL, as opposed to assembly language subroutines. This gives languages which have real time control features an eage, since a language without them must resort to assembly language for real time control. Another key feature for readability is the degree to which a language permits and encourages structured code. The newer languages, especially Pascal and Ada, are designed to force the programmer into structured pregrams. Manipulation of large data arrays is important and all of the candidate languages have this capability. Finally, the availabi’ity of the language is crucial. Is it supported on the selected ADSP computer? Is it sufficiently mature to be fully documented and readily usable? Is it in widespread use? These are the key questions.

Table 22 summarizes the key characteristics of each language. Ada most closely meets the needs of the ADSP. This conclusion was reached after a careful comparison of the candidate HOLs.

The HOL which has been proven in almost every application is Fortran. In use for twenty years, Fortran is a suitable substitute for almost any language. The newest versions of Fortran include provisions for structured programming, including control constructs such as IF...THEN...ELSE. Most versions have fairly sophisticated 1/O capabilities, although most are uniqua to one machine and rot: readily transportable to another. Most Fortran versions do not have any real time control structures, leaving these functions to the ojera:ing system. Fortran can handle arrays of data, although complex file structures can be difficult to handle. Fortran is less readable than the never languages, although its widespread use may offset this, in that so many programmers readily understand it. In short, Fortran may be an acceptable vehicle for the task, but its limitations were precisely what the newer languages were designed to improve upon. It may pay to take advantage of these improvements. SPL/I, or Signal Processing Language 1 , is a relatively new language developed by Intermetrics, Inc. for Navy programmable signal processing applications. As a language designed expressly for signal processing, it is an immediate candidate for ADSP. It contains many reatures of use $: \cdot$ this application, especially in the number and power of its real time control constructs. These can effectively handle multiple parallel processes and the interfaces between them. SPL/I shares the structured constructs and data typing of Pascal. Ada, and others. Detailed study of the language shows strong structural similarities to other HOLs, including many of their weaknesses. SPI/I, like Pascal and Ada, makes an effort to minimize dependence on a particular macnine or compiler to guarantee software transportability. As such, I/O capabilities are essentially left to the implementation instead of being defined within the language. SPL/I has only four simple library functions defined to handle $1 / 0$. This is less than the other candidate HOLS.
Another major problem with SPL-I is a lack of available implementations. Versions exist for Navy F.N/UYK-7 $^{W}$ and AN/UYK-20 computers, but outside the Navy, it is virtually unused. Much of this lack of use is due to the
expectations in the development of Ada, which has most of the same real time control features, and at the same time provides for more readable code.

Pascal, a language invented in 1970 by Niklaus Wirth of the University of Zurich, is intended primarily as a teaching language. The original intent was to provide the necessary language constructs to force a programmer to write structured programs. Many of the concepts defined by Pascal have been used in later languages, especially Ada and SPL/I. Pascal is a highly readable, easy to understand language. It is a strongly typed language, which means that all variables and constants in a prograin must be explicicly declared as to its type at the beginning of a program. Translation between different types is difficult urioss their relationship is pre-defined. For example, this feature prevents a programmer from accidently equating feet and pounds in a calculation. Strong typing requires a programmer to think through the process he is defining before he actually codes it. Pascal incorporates many program control features which guarantee structured program flow: IF...THEN...ELSE and CASE statements provide explicit conditions for branching, making the reasons for each branch clear in each case. Parcal has array handing features which provide easy set up and manipulation of complex data structures such as linked lists.

Because Pascal was originally intended as a teaching language, it has a number of shortcomirgs when applied to a suphisticated real time application. The I/O defined for Pascal in its original form is severely deficient, amd numerous implementations hive attempted to improve upon its I/O capabilities. This has seriously inhibited Pascal's ability to be implementation dependent, and hence transportable from one compiler to another, and from one machine to another. It also makes it difficult to compare Pascal with other languages. since some versions of Pascai may have adequate $1 / 0$ for the intended application.

Another serious deficiency of Pascal is the absence of any real-time control structures. Pascal is not intended for real time applications. To be readily transportable from ore system to another, many Pascal
compilers, most notably the UCSD version, translate to a universal intermediate level language (termed a p-code), which in turn is readily cranslated to the rachine language of host computer. This two level translation process, while useful, limits the efficiency of a Pascal program in terms of the number of machine instructions, it takes to execute a single Pascal instruction. The two level process also makes the execution of a Pascal program very difficult to predict and/or minimize.

Despite these limitations, Pascal has gained wiocspread acceptance in the industry. Some companies, like Texas Ins cruments, have standardized all of their software development around the use of Pascal as their preferred HOL. There are many versions available, and any final computer system chosen is likely to have at least one commercially available compiler to use. In addition, an intermediate p-code language could be translated to any embedded processor's instruction set, providing HOL capabilities and documentation within the system hardware, provided real-time control and I/O problems can be dealt with.

Many of the limitations of Pascal may be overcome through the use ol Ada, a derivative of Pascal first developed in 1979 by a design team from Honeyweli-Bull, under the direction of Jean Ichbiah. Developed as a joint Army/Air Force project, Ada is intended to be the Department of Defense standard HOL for all major systems of the future. As such, rida was designed to be an all-purpose language, incorporating the key features of Pascal together with real-time processing constructs and improved I/O capabilities. Like Pascal, Ada is a strongly typed language, with all variables defined clearly at the beginning of the program. Ada has all the same structured program constructs as Pascal, and in addition has features to simplify separate compilation of different sections of a large program. The I/O constructs in the original Ada definition are adequate, although still somewhat less sophisticated as most Fortran implementations. This will probably be improved when Ada is actually implemented. Ada is designed to be highly readabie: the design intent was to simplify documentation requirements by using a language as close as possible
to a Program Design Language (PDL) which car be used for documentation instead of flow charts. Ada's multitasking capability provides clear definitions of multipie parallel processes, and their interfaces.

The primary problem with using Ada is that currently its implementation is in its infancy. The most notable development is the Intel VLSI LAPX-432 microcomputer which is designed for the Ada HOL. The initial definition of the language has generated considerable interest within the software industry. A number of projects to develop working compilers has begun, bnth inside and outside the defense industry. RCA has identified 24 such projects for various machines. It is expected that by 1983 most major computer systems will have usable versions of Ada, and that by 1985 Ada will be in widespread use. As such, it appears to be most promising candidate for the ADSP. Ada combines the best features of the other languages into a single language: the versatility of Fortran, the real-time control of SPL/I, and the readability of Pascal. Table 22 summarizes the key characteristics of each language.

Table 22
Key Characteristics of Candidate HOLs

| feature language | SPL-1 | FORTRAN | PASCAL | ADA |
| :---: | :---: | :---: | :---: | :---: |
| READABILITY | FAIR | FAIR | EXCELIENT | 6000 |
| REAL T'ME CONTRO STRUCTURES | YES | IMPLEMENTATION DEPENDENT | No | YES |
| arpay oata HANDLNG STRUCTURES | G000 | FAIR | 6000 | G000 |
| STRUCTURED PROGRAMMING FEATURES | YES | $\begin{aligned} & \text { IMPLEMENTATION } \\ & \text { DEPENDENT } \\ & \text { (FORTRAN } 777 \end{aligned}$ | YES | YES |
| data typing | YES | NO | YES | YES |
| 110 HANCLING | POOR | EXCELIENT | POOR | FAIR |
| CURRENT USE | ANUYK-7. <br> AN/UYK -20 NAVY <br> COMPUTERS ONLY | WIDESPREAD | WIDESPREAD | $\begin{aligned} & \text { FIRST } \\ & \text { COMPRLERS IN } \\ & \text { DEVELOPMENT } \end{aligned}$ |
| $\begin{aligned} & \text { FUTURE USE } \\ & \text { (1983- } \end{aligned}$ | UNPREDICTABLE | WIDESPREAD | $\begin{aligned} & \text { WIDESPREAD } \\ & \text { IALHOUGH MAY } \\ & \text { BERRPLACED } \\ & \text { BYADA) } \end{aligned}$ | WIDESPREAD |

### 2.5 ADSP SELECTED DESIGN

### 2.5.1 Selection of Algorithm

The FFT convolver was selected for the ADSP. The selection of a processing algorithm was made principally between the time domain, FFT convolver and subarray algorithms. The other approaches were considered in relation to these three.

### 2.5.1.1 Key ADSP Requirements

The key requirements for the ADSP are summarized in Table 23.
Continuous variation in processing parameters, with provision for modular construction and future growth are driving factors. In addition, the processor must accommodate continuous or burst modes.

Table 23
Key ADSP Requirements

|  |  | BASELINE | VARIATION |
| :---: | :---: | :---: | :---: |
| - range correlator: | COMPLEX SAmple rate | 11 MHz |  |
|  | PULSE SAMPLES | 475 |  |
|  | PULSE COMPRESSION | 410 | 10-660 |
|  | RANGE SWATH | 4000 | $500-4000$ |
|  | integrated fange sidorobes | -1508 |  |
| - AZIMUTH CORRELATOR: | PRF | 2500 Hz |  |
|  | L00KS | 4 | - |
|  | AZLMUTH COMPRESSION RATIO/LOOK | 350 | 20-3500 |
|  | INTEGRATED AZIMUTH SIOE.OBES | -20d8 |  |
|  | rance migration | 86 PANGE BIN |  |

- trade-OFF among range swath, number of looks, and resolution
- MODULARITY -- PROVISION FOR:
- SWATH WIDTH -- UP TO FOUR TIMES BASELINE
- more parallel looks
- multiple frequencies and polarizations


### 2.5.1.2 Programmability

The variation in JAI $^{\text {p }}$ parameters can be accommodated by the three major processing algorithms shown in Table 24. Both the time domain
and FFT convolver approzches offer full control over the SAR processing parameters such as compression factor. However, the subarray implementation cannot be readily programmed to cover a continuous selection of parameters. The subarray formation process most easily accommodates a stepped parameter control. A disadvantage accrues to the time domain approach with variations in the number of looks. A variable prefilter would be difficult to implement in this case. The Type B time domain processor is used in the comparison because it is preferred over the Type A processor.

Table 24
Algorithm Programmability

- ALL APPROACHES CAN ACCOMMODATE VARIATIONS IN RANGE SWATH THROUGH MEMORY CONTROL AND SCALING

| ALGORITHM | REQUIREMENT VARTATION | METHOD | IMPACT ON HARDWARE |
| :---: | :---: | :---: | :---: |
| $\begin{aligned} & \text { TIMED DOMAIN } \\ & \text { TYYE E) } \end{aligned}$ | COMPRESSION | VARY NUMBER OF ACTIVE CORRELATORS | MODEST INCREASE IN CONTROL FUNCTION |
|  | LOOKS | - SWITCH CORRELATORS <br> - VARY PREFILTERING | LARGE INCREASE IN CONTROL AND SWITCHING HARDWARE |
| TT CONVOLVER | COMPRESSION | CONTROL COMMANDS AND REEERENCE FUNCTION | MODEST HARDWARE INCREASE AND CONTROL FUNCTION SOFTWARE |
|  | L00\%S | REPROGRAM CONTROL SEQUENCE | MINIMUM HAROWARE INCREASE |
| SUBARRAY | COMPRESSION | VERY DIFFICULT TO OBTAIN CONTINUOUS CONIROL: CONTROL IN STEPS | LaRGE INCREASE IN COMPLEXITY OF |
|  | LOOKS | REPROGRAM CONTROL SEQUENCE | CO |

### 2.5.1.3 Incremental Implementation and Growth

Incremental growth or implementation can refer to variation in swath width, number of looks, or the number of processing chainels. In addition, the capability of the processor to nandic data at real time rates is a factor. All of these issues are in a sense related. As the swath width and number of channels increases, the processor memory and computation can be expected to increase linearly. As the number of looks increases the computation requirements generally decrease and drop dramatically for the time domain processor with a prefilter.

The algorithm and architecture selection are related and in anticipation of the selection of a parallel architecture that implementation is assumed for the FFT convolver and subarray algorithm in Table 25.

Table 25
Incremental Growth and Implementation

## PARAMETER

INCREMENTAL GROWTH

| ALGORITHM | SWATH WIDTH | PARALLEL LOOKS |
| :---: | :---: | :---: |
| $\begin{aligned} & \text { TIME DOMAIN } \\ & \text { TYPE B) } \end{aligned}$ | ADO PARALLEL SYSTEM (S) | AUD ELEMENTS TO REDUCE LOOKS |
| $\begin{aligned} & \text { FT CONNOLVER } \\ & \text { PAPALLEL } \\ & \text { ARCHITECTURE) } \end{aligned}$ | ADD RANGE AND AZIMUTH PROCESSORS | $\begin{aligned} & \text { REDUCING LOOKS } \\ & \text { INCREASE FF } \\ & \text { SIZE AND NEMORY } \\ & \text { STORAGE } \end{aligned}$ |
| AZIMUTH SUBARRAY (PARALLEL ARCHITECTURE) | $\begin{aligned} & \text { ADD RANGE } \\ & \text { AND AZIMUTH } \\ & \text { PROCESSORSS } \end{aligned}$ | $\begin{gathered} \text { IS MOST EFFECTIVE } \\ \text { FOR LARGE } \\ \text { COMPRESSIONRATIOS } \\ \text { (FEM LOOKS) } \end{gathered}$ |

NUMBER OF CHANMELS
(FREQUENCYAND
POLARIZAIIONI
ADO PARALIEL SYSTEMS

ADC PARALIEL SYSTEMS

ADD PARALLEL SYSTEMS

INCREMENTAL IMPLEMENTATION

| TIME DOMAIN TYPE BI | NOT PRACTICAL BECARSE OF POOR MODUL MEMORY | - Programmable PREFILTEP <br> - Increages real tIME RAIE | N/A |
| :---: | :---: | :---: | :---: |
| FFT CONVOLVER Pár RALLEL ARCHITECTURE) | REDUCE NUMBER OF RANGE AND AZIMUTH PROCESSORS | MORE LOOKS REDUCE SIZE OF FFTS AND MEMORY | N/A |
| AZIMUTH subarray (PARALlel ARCHITECTURE) | REDUCE NUMBER OF RANGE AND AZTMUTH PROCESSORS | LOSES EFFECTIVENESS AS NUMBER OF LOOKS BECOMES LARGE | N/A |

### 2.5.1.4 Burst Multimide Processing

The issues in burst mode processing fall into two categories; 1) the ability of the processor to accommodate mode changes with different SAR parameters with no loss of imagery and 2) the ability to process specia? modes such as a large members of looks in a burst mode (See Figure 5). The first case is illustrated by Figure 49. A SAR processor will have an


Figure 49
Burst Processing Delays
inherent delay ( $\tau_{D}$ ) from input to output which is greater than the integration time ( $\tau_{A}$ ). Radar bursts can switch instantaneously from one frequency to another as illustrated. A processor can have either a switched mode control or a traveling mose control. With a switched mode control, all of the parameters of the processor change together so the data within the processor when the mode swtich is changed is all lost. On the other hand, with a traveling control that moves with the data no additional invalid data appears beyond that equal to the integration time. The traveling mode control is obviously preferable and the time domain processor (Type B) is unique in its adaptability to a traveling mode control. The lost data can be avoided by increasing the bulk memory size by $50 \%$ for the FFT convolver and $100 \%$ for the subarray technique.

The second burst processing problem can be considered in the extrenie case where the length of a burst is equal to the integration time and the bursts are of arbitrary spacing. In this situation there
can be essentially one image resolution element per look and the FFi convolver in its normal form is extremely inefficient. The time doriain type $B$ process is well matched to the case because of its mulitplier - accumulator structure. The type B processor can be used with each processing module handling a separate look for the fuil burst range swath. The FFT convoiver and subarray systems can process this mode with some variation in their structure. The basic approach is to deramp the input signal, corner turn, FFT, correct for range all migration and integraice the appropriate multiple looks as indicated in figure 50.


FIGURE 50. INTERRUPTED BURST PROCESSING WITH MANY LOOKS

### 2.5.1.5 Algorithm Cost Comparisons

Relative costs were developed for the three principal ADSP algorithms and are summarized in Table 26 . The design costs of the subarray technique overshadow its lower materials costs. 'faterials for the time domain processor reflecting the large number of computations required dominate its cost profile.

Table 26
Relative Algorithm Costs

|  | $\begin{aligned} & \text { TIME } \\ & \text { DOMAIN } \end{aligned}$ | $\begin{gathered} \text { FFT } \\ \text { CNUVOLYR } \end{gathered}$ | $\begin{aligned} & \text { SUB- } \\ & \text { ARRAY } \end{aligned}$ |
| :---: | :---: | :---: | :---: |
| PROGRAM CONTROL MONITORING AND DESIGN assurance | 0.246 | 0.207 | 0.220 |
| DESIGN | 0.416 | 0.384 | 0.626 |
| MATERIALS/SERVICES | 1.214 | 0.263 | 0.174 |
| ASSEMBLY AND TEST | 0.419 | 0.146 | 0.191 |
| TOTALS | $\begin{aligned} & 2.295 \\ & 12.31 \end{aligned}$ | $\begin{aligned} & 1.000 \\ & (1.0) \end{aligned}$ | $\begin{aligned} & 1.211 \\ & (1.2) \end{aligned}$ |

### 2.5.1.6 Risks

Development risks associated with the candidate algorithms are indicated in Table 27. The time domains approach has problems with reliability and the control of a large hardware system at: the nominal system clock rate of 11 MHz . The complexity of the subarray processor design is its major risk factor while the FFT convolver algorithm has no serious drawbacks. Its only questionable area is in achieving corvenient modularity of the hardware elements - a concern that also applies to the other algorithms. General algorithm-independent risks are also listed in the table and the major one affecting the lifetime of the ADSP is meeting the correct mix between hardware and software controls.

Table 27
ADSP Algorithm Development Risks

|  | SPECIFIC |  |
| :---: | :---: | :---: |
| TIME DOMAIN | - control of large BUS SYSIEM multiple CAbinets <br> - 11 MHz CLOCK |  |
|  | - RELIABILITY | SCHEDULE: TIMELY AVAILABILITY OF PURCHASED ICS |
| fficonvolitr | - achieving module partitioning | COST: - UNPREDICTABLE INFLATIONARY |
| Subarray | - achiving required programmability <br> - DEVELOPMENT OF SIMPLETEST procedure <br> - CONTROL SYSTEM | - MEETING COST PROJECTIONS OF ICS IF PURCHASES MUST BE SPLIT INTINE |

### 2.5.1.7 Summary and Selection

A summary of the general characteristics of the various processing algorithms studied is given in Table 28. Particular algorithms have features which may be useful in certain applications. For example, the subarray process, which offers a minimal hardware implementation, is advantageous for a single dedicated processor for low power and size on-board applications. However, the FFT convolver algorithm was seiected for the ADSP because:

- Programming for multiple modes is straightforward,
- the hiaher cost of memory and computations (relative to subarray) are offset by savings in control system design costs,
- a lower risk is associated with the approach,
- integrated circuit iechnology thrust is reducing the cost of memory and computations,
- it lends itself to modular growth,
- it provides a continuous selection of SAR parameters and,
- the ADSP has no requirements for small size, low power or special environmental conditions.

Table 28

| ALGORITHM | MAJOR FEATURE | DISADVANTAGES |
| :---: | :---: | :---: |
| FFT CONVOLUTION (1D) | EASIER TO PARTITION AND CONTROL | NOT MINIMUM HARDWARE |
| BLOCK FFT APPROACH | NO CORNER TURNING MEMORY | MORE MEMORY REQUIRED |
| $\begin{aligned} & \text { 2D FAST FOURIER } \\ & \text { TRANSFORM } \end{aligned}$ | MOST FLEXIBLE | HIGHER COMPUTATIONAL RATE DUE TO 2 D OVERLAP LIMITED BY DEPTH OF FOCUS |
| 20 FAST POLYNOMIAL TRANSFORM | LESS MULTIPLICATIONS | MORE DIFFICULT TO PARTITION |
| SERIAL TIME DOMAIN CORRELATION | VERY FLEXIBLE <br> EASY REFERENCE GENERATION <br> BEST FOR BURST MODE | REQUIRES MOST COMPUTATIONS |
| PARALLEL TIME DOMAIN CORRELATION | NO ADVANTAGE OVER SERIAL SYSTEM | REC UIRES MOST COMPUTATIONS <br> REFERENCE FUNCTION DIFFICULT TO GENERATE |
| SUBARRAY PROCESSING | MINIMAL HAROWAFİ | MORE DIFFICULT TO PROGRAM |

### 2.5.2 Selection of Architecture

A comparison of pipeline and parallel architecture is given in Table 2 ?
The preponderance of key favorable factors make the parallel approach the optimum selection. A parallel structure is also characteristic of a number cf specific processors including the Massively Parallel Processor = MPP (19), the Advanrad On-Board Signal Processor - AOSP (20) and S-1 (21). Table 30 shows how these systems could be configured for the ADSP. Their use in the ADSP would depend upon the achievement of their development goals in a timely manner. In addition, they do not have the incremental implementation and growth characteristics desired. For those reasons we have recommended a parallel design tailored specifically to the ADSP requirements.

Table 29
Pipeline - Parallel Architecture Comparison
ADSP

| CHARACTERISTIC | PIPELINE | PARALLEL |
| :---: | :---: | :---: |
| NATURAL PARTITIONING | EACH PIPELINE ELEMENT DOE.: OWN FUNCTION | - RANGE -- PRI <br> - AZIMUTH -- RANGE SLICE |
| PROGRAMMABILITY | FFTS -- LENGTH CHANGE <br> CONTROL - TIMING MUST BE ADAPTABLE | - firmware programmable (NO HAFIDWARE RESTRUCTORING) |
| MEMORY SYSTEM | - CAN TAILOR EACH ELEMENT TO MINIMUM SIZE <br> - MUST OPERATE AT HIGH SPEED | - LOW SPEED OPERATION <br> - MODULAR, CONVENIENT SIZES |
| RELIABILITY\| <br> ENHANCEMENT | - MED!UM RELIABILITY <br> - EXTRA PIPELINE UNITS BUT CONTROL IS DIFFICULT | - IMHERENT HIGH RELIABILITY <br> - SPARES ARE INDEPENDENY AND EASILY INSERTED |
|  | - tESTING IS DIFFICULT | - STATUS TEST IS STRAIGHTFORWARD |
| MODULARITY | - MODULARITY REDUCES EFFICIENCY | - MODULARITY INHERENT IN DESIGN |
| RISK | - moderate | - LOW |

Table 30
Processor Comparisons
('FT Convolver Algorithm -- FFTs)

|  | $\begin{aligned} & \text { ADSP } \\ & \text { REQUIRED } \end{aligned}$ | MPP. PARALLEL | $\begin{aligned} & \text { AOSP } \\ & \text { MULTIBUS } \\ & \text { (56 ACES -- } \\ & \text { eSt) } \end{aligned}$ | $\begin{gathered} \text { S-1 } \\ \text { CROSSBAR } \\ \text { (10A-BOXES) } \end{gathered}$ |
| :---: | :---: | :---: | :---: | :---: |
| REAL <br> MULTIPLICATIONS <br> PER SECOND $\div 106$ | 2,240 | 910 | 2,240 | 2,240 |
| REAL ADDS $\text { PER SECOND } \div 10^{6}$ | 3,360 | 4,428 | 3,360 | 3,360 |
| APPROXIMATE FMEMORY $\begin{aligned} & \text { STORAGE } \div 106 \\ & \text { (BITS) } \end{aligned}$ | 416 | $16^{\circ 0}$ | $560{ }^{\circ} 0$ | - |

- MULTIPLIES AND ADDS ARE NOT SIMULTANEOUS FOR MPP
* EXTENDABLE
** ASSUMES 5 MEMORY CHIPS PER ACE


### 2.5.3 Desion

The design selected for the ADSP features a parallel processor implementation of the FFT convolver algorithm for both the range and azimuth correlators. It is progranmable and adaptable to multiple missions. The parallel architecture allows a modular implementation suitable for expansion or reduced system capability. The parallel design achiever high system reliability and graceful performance degradation.

The system is designed with TTL compatible logic, has an internal 8 MHz operating clock rate, has floating point arithmetic and employs 64 K dynamic RAMs for the bulk memory. It is controlled by a host computer for mission set up and performance monitoring, but because most of the control reqirements are embedded in firmware, a minimum of software effort is anticipated for a new mission set-up.

Figure 51 shows the arrangement of the parallel processing modules. For the assumed 8 MHz clock rates, 22 parallel processors are used for the range correlator and 15 parallel channels are employed in the azimuth correlator.

The architecture and algorithm permits a maximum degree of mission programmability with a minimum of softivare set-up effort. The host computer is not involved in any of the real-time computations as a process proceeds bur is used for configuration and test functions. real time controls are embedded in the signal processing elements. Commands from the host computer set up the SAR processing parameters.

### 2.5.3.1 Range Correlator Design

The range correlator is illustrated in Figure 52. Each processing module handles a full radar PRI. The PRI/FFT unit, whose primary function is FFT processing, is the same module used for FFT processing in the azimuth correlator.

Each PRI/FFT processor consists of an input/output (I/O) memory, a working memory for computations, an arithmetic computation unit, a reference function store, and a controller. The I/O memory is large

Figure 51
Advanced Digital SAR Processor Design
enough to hold the full range swath window prior to range compression. Data is transferred from the $1 / 0$ memory through the data switch to the working memory for processing. The computation unit can be programmed to do the FFT butterfly, weighting, reference multiplication, magnitude calculation ( $I^{2}+Q^{2}$ or $\sqrt{I^{2}+Q^{2}}$ ), normalization, scaling, and either real or complex FFT computations.

### 2.5.3.2 Linear Range Migration Correction

From the point of view of control simplicity it appears preferable to compensate for all of the linear components of range migration separately from the quadratic terms. This is the approach employed in the recommended design where the correction is applied after range pulse compression. The implementation is simply an incremental delay plus an interpolated fractional sample delay. Our simulation showed that, with the FFT convolver processing algorithm, adequate performance could be obtained without the use of a separate linear range migration unit if interpolation were used in the frequency domain correction. However, we have included the function since it may contribute to higher quality imagery in some cases.

### 2.5.3.3 Corner Turning Memory

The corner turning memory module functions are given in Figure 53. A PRI buffer is used to capture the range swath to be processed. The net data rate through each parallel channel is then only about $22 / 15 \mathrm{MHz}$. This low data rate permits the use of slow 64 K dynamic RAMs for the bulk store.

The organization of the corner turning memory modules is shown in figure 54. With a 4096 point azimuth processing window, the windows must be repeated each 2048 points. This is accomplished by using an overlap - save sequence in the memories. After the full 4 K samples are stored in two memories the next $2 K$ of data is read in to one-half the memory while the 4 K window is read out. Operation in this manner means that the total bulk corner turning memory storage is approximately the product of the range swath and the azimuth correlation extent.


Figure 53
Corner Turning Memory Function


Figure 54
Corner Turning Memory Organization

### 2.5.3.4 Azimuth Correlation

After the first FFT is taken to convert the constant range lines to the spectral domain, the quadratic range migration correction (and any residual linear component) is applied together with the spectral reference focusing function. Bearing in mind that the data rate at the interface to this function shown in Figure 55 is less than 2 MHz , the implementation of the functions can be streamlined by time-sharing hardware elements. Most of the adaptive processes in the azimuth correlator are centered here. A separate clutter lock estimator is used. Although the focusing function generation is shown on the module, our selected approach for this function is to employ a separate FFT processor module for computing all of the references as discussed later in this section.


Figure 55
Range Migration Correction and Matched Filter

### 2.5.3.5 Multilook Integrator

Figure 56 shows the design approach for the multilook integrator. Input data is first interpolated, if necessary, for image registration or geometric mapping correction, or changes in the output image pixel density. If the autofocus is not accomplished with modification to the focusing function, multilook registration corrections can be applied here. The multiple looks are integrated after the magnitude of the complex samples are obtained. The multiple looks are integrated after the magnitude of the complex samples are obtained. The multilook memory store is approximately one-eighth as large as the corner turning memory and all of the associated functions can therefore be included on the module.


Figure 56
Multilook Integrator

### 2.5.3.6 Clutter Lock

A clutter lock technique was described in (22). It relies on a measure of symmetry of the mean image energy of the four looks. The normalized error is determined from

$$
\frac{E_{1}+E_{2}-E_{3}-E_{4}}{E_{1}+E_{2}+E_{3}+E_{4}}
$$

where $E_{1-4}$ are the energy le,els in the respective looks. A modification of this approach is suggested here based on the following two observations. First, the referenced approach assumes that the actual mean image energy of a particular look, say look 3, does not fluctuate too much as the terrain is changed. Although this is generally valid, there may be specific peculiarities, e.g. transition from ocean to land, that will violate this assumption. Secondly, as the frequency offset ( $\mathrm{Fc}-\mathrm{fc}$ ) approaches zero, the difference between the mean image energy looks 2 and 3 would appear to be insensitive to changes in ( $\mathrm{Fc}-\mathrm{fc}$ ).

The suggested modification uses two different segments of the range compressed data in such a way that the image of look 1 cf the first segment coincides with the image of look 4 of the second segment. Furthermore, only the spectra of looks 1 and 4 are used to establish the symmetry, as these are much more sensitive to changes in the frequency offset ( $\mathrm{Fc}-\mathrm{fc}$ ).

Two measures of symmetry can be used. The first is the difference of the total image energy of each look, i.e. E1-E4. Another measure is obtained by comparing each spectral line in the two looks. Let $\left\{S_{1}(k)\right\}$ and, $\left.\mathrm{S}_{4}(k)\right\}$ be the $k^{\text {th }}$ component of looks 1 and 4 respectively. Then a good measure is

$$
D=\frac{\varepsilon}{k}\left|S_{1}(k)-S_{4}(k)\right|
$$

This quantity appears to be much more sensitive to changes of the frequency offset ( $\mathrm{Fc}-\mathrm{fc}$ ), as can be seen from the result of a typical simulation presented in Table 31. The disadvantage of using 0 to perform clutter lock is that one has to seek the minimum of $D$ as ( $\mathrm{Fc}-\mathrm{fc}$ ) is varied whereas the use of ( $E_{1}-E_{4}$ ) needs only to search for the zero crossing.

Table 31
Clutter Lock Simulation Results

## - POINT TARGET

- TWO POINT INTERPOLATOR IN RANGE COMPENSATION ALGORITHM
- YAW IS APPROXIMATELY 1/2 A LOOK

RESULTS:

|  |  | $F_{C}-f_{C}$ |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| MEASURE | 0.008 | 0.006 | 0.004 | 0.002 | 0 | -0.002 | -0.004 |  |
| $E_{1}$ | 5.057 | 5.055 | 5.061 | 5.070 | 5.076 | 5.082 | 5.089 |  |
| $E_{4}$ | 5.094 | 5.087 | 5.089 | 5.083 | 5.076 | 5.070 | 5.060 |  |
| $E_{1}-E_{4}$ | -0.0304 | -0.0325 | -0.0286 | -0.0131 | -0.0004 | +0.0125 | +0.0287 |  |
| 0 | 3.989 | 3.003 | 1.904 | 1.442 | 0.343 | 1.468 | 1.985 |  |

### 2.5.3.7 Generation of Focus Function

The reference signal needed for the azimuth matched filter is of the form $\left\{e-j a\left(n-n_{0}\right)^{2}\right\}_{n}$ where the constant a depends on the range as well as other system parameiers. This dependence on range demands that new reference functions be generated fairly frequently for proper focusing, and the Fourier transform of the newly generated reference function be calculated if the matched filtering is carried out in the frequency domain. The required spectral focusing function can be generated by taking the FFT of a linear frequency ramp function which has been adjusted for the correct slope. While we have initially selected this first method, it is worthwhile to consider alternates which reduce the amount of computations. One method for reducing this computation task is described here. Assume that $\left\{e^{\left.-j a\left(n-n_{0}\right)^{2}\right\}}\right.$ is the focusing function in use, and the value of $a$ is changed to $a+\Delta$ for focusing at a different range. Rather than generate $e^{-j(a+\Delta)\left(n-n_{0}\right)^{2}}$ anew, we rewrite $e^{-j(a+\Delta)\left(n-n_{0}\right)^{2}}$ as follows:

$$
\begin{aligned}
e^{-j(a+\Delta)\left(n-n_{0}\right)^{2}} & =e^{-j a\left(n-n_{0}\right)^{2}} e^{-j \Delta\left(n-n_{0}\right)^{2}} \\
& =e^{-j a\left(n-n_{0}\right)^{2}}\left[1-j \Delta\left(n-n_{0}\right)^{2}-\frac{\Delta^{2}\left(n-n_{0}\right)^{4}}{2}+\cdots\right]
\end{aligned}
$$

If is sufficiently small, the first two terms in the series expansion are sufficiently accurate for our purpose. In the transform domain:

$$
\begin{aligned}
\because\left\{e^{-j(a+\Delta)\left(n-n_{0}\right)^{2} ;} \approx\right. & \because\left\{-j a\left(n-n_{0}\right)^{2}\right\} \\
< & -j \Delta-\left\{\left(n-n_{0}\right)^{2} e^{-j a\left(n-n_{0}\right)^{2}},\right. \\
& -\frac{\Delta^{2}}{2} \because\left\{\left(n-n_{0}\right)^{4} e^{-j a\left(n-n_{0}\right)^{2}}\right\}
\end{aligned}
$$

Thus, one needs to store only the three transforms on the right side of the equation and combine them using whatever value of $\Delta$ necessary to obtain the transform of the desired new reference.

To determine one range of $\Delta$ for which this approach offers sufficient accuracy, we senerated a reference with $\Delta$ varying and used it for matched filtering the returns from a point target. The results were compared with those obtained when $\Delta=0$.

If there is no yaw, then the proposed alternate will produce less than 0.5 dB change in the response when $\Delta$ is kept within $\pm 50$ range cells. If a yaw corresponding to half a look is present, then $\Delta$ has to be within $\pm 30$ range cells in order to keep the change to within 0.5 dB . If the yaw is increased to one look, then $\Delta$ has to be within $\pm 20$ range cells.

While the alternate method offers significantly fewer computations, the availability of FFT processing modules makes the direct reference computation cost efficient. With a new reference required every eight range cells, two FFT processing modules can be applies to the task of computing the references for 15 parallel channels.

### 2.5.3.8 Control System

System requirements which affect the range and azimuth processor controls are listed in Table 32. These general requirements impact in various ways on the processor depending upon the algorithm and architecture emp lo yed.

Table 32
ADSP System Control Requirements
Range Processor
Range Compression Ratio: 10 to 660
Swath Whath: 500 to 4000

Aztmuth Processor
Azimuth Compression: $20 \cdot 350$ par look
Reference Function Update: Cross track - every 8 range cells
Along track - every $\mathbf{1 2 , 0 0 0}$ radar pulses

## Required Modes

Duai Frequency
Dual Poiartzation
Burst
Contingus
Variations within Mission in altitude, look angle, resolution, number of looks, swath width, and azimuth weighting

The FFT convolver is relatively easily controlled for the variation of system parameters. The length of the FFT can be changed to accommodate varying time bandwidth products and swath width. This is especially easily accomplished in a serial FFT where the length change is by firmware control. In addition, waveform and weighting function variations can be accommodated by either storing the additional functions in ROM or by inserting them into a temporary storage RAM via the control computer. A continuous selection of swath widths and waveform length can be batch processed with the FFT convolver within the limits of the maximum size FFT. The FFT can also be operated in a slidinc aperture mode. Then successive FFT windows are processed at intervals of the difference between the FFT size and waveform length. Table 33 lists a number of variations in FFT and waveform size and resulting complex computations per output point. The table illustrates the logarithmic increase in the required computations as the waveform and FFT size increase. It also illustrates the computational advantages of increasing the FFT size relative to the waveform length.

TABLE 33

| SIZE | $\begin{aligned} & \text { WAVEFORM } \\ & \text { SIIE } \\ & \hline \end{aligned}$ | RANGE COVERAGE PER FFT | COMPLEX <br> COMPUTATIONS <br> PER POINT |
| :---: | :---: | :---: | :---: |
| 512 | 256 | 256 | 20 |
| 1024 | 512 | 512 | 22 |
| 2048 | 1024 | 1024 | 24 |
| 4096 | 2048 | 2048 | 26 |
| 8192 | 4096 | 4096 | 28 |
| 8192 | 2048 | 6144 | 18.67 |
| 8192 | 1224 | 7168 | 16 |
| 8192 | 512 | 7680 | 14.9 |

## ORIGINAL PAGE IS OF POOR QUALITY

Controls required on the FFT convolver to meet the total programming requirements are indicated in Figure 57. Some of these controls may be most appropriately placed on a common control module fo: multiple range processors. The tradeoff in this regard is the consideration for multiple uses of the module, as in the azimuth correlator.

Controls for the azimuth processor are distributed to each functional element. For the FFT convolver algorithm all required controls for the FFT are embedded in the FFT module used in the range processor. The primary control centers are then in the corner turn memory, the range migration-matched filter module and the multi-look integrator. Linear range migration is a separate function in the design as is the clutter lock estimator.


Figure 57 FFT Convolver Control

## ORIGINAL PAGE IS OF POOR QUALITY

### 2.5.3.9 Test Subsystem

The complexity of the ADSP dictates the implementation of a diagnostic, status test subsystem in conjunction with the operating system. A conceptual test system is shown in Figure 58. The system elements would be exercised with a test signal sequence and check-sums would be obtained at selected locations in the processor. These check sums are read back to 'he control computer for failure localization.

In addition to the built-in-test system, a test fixture should be developed for testing and debugging functional modules. This test unit could be used both for production testing and long term maintenance of the equipment.

### 2.5.3.10 Physical Configuration

The ADSP system was sized for its physical configuration. The module complement for the system is shown in Table 34 . Large $15^{\prime \prime} \times 17^{\prime \prime}$ modules were assumed capable of housing 200 or more standard dual-in-line circuits. Using an average of 170 circuits per module the total number of circuits in the system is about 33,000 including $5 \%$ spares. Five standard cabinets can be used to house the ADSP with about 2000 watts dissipation per cabinet.


Figure 58 Test System Concept

Wirewrap incerconnections were assumed. Because of the large number of circuits in the system it is recommended that they be acquired with a burn-in specification of 60 hours, particularly if plastic circuits are used. A PDP-11/23 computer with 256 K bytes of memory, dual disk, and hardware floating point was selected for the host computer although any of a large number of computers would be adequate.

TABLE 34. MODULE SUMMARY

| MODULE TYPE | FUNCTIONAL UNITS | OPERATING SPARES | OTHER SPARES (OPTIONAL) |
| :---: | :---: | :---: | :---: |
| PRI/FFT PrC essor | 54 | 4 | 4 |
| Corner Turn Memory | 60 | 4 | 4 |
| Range Migration Correction | 15 | 1 | 1 |
| Multi-look Integrator | 15 | 1 | 1 |
| Linear Range Migration | 1 |  | 1 |
| Timing Unit | 1 |  | 1 |
| Clutter Lock and Focus | 1 |  | 1 |
| Test Probe Interface | 6 |  | 1 |
| Controllers | 3 |  | 3 |
|  | 156 | 10 | 17 |

### 2.5.4 Impact of Technology

The specific impact of future technology on the implementation of the ADSP can be deduced from Figures 59 and 60 which show key integrated circuit and architectural developments over the ADSP development time. All of the DC developments shown should serve to minimize the ADSP cost if they are available in a timely manner. Either 256 K bit dynamic RAM's or 64 K bit static RAMs will be available by 1983. Wide word configurations of up to 8 bits will reduce costs. Particular logic developments which can be anticipated are highlighted by a coming selection of chips specifically designed to reduce the cost of implementing the fast fourier transform (FFT). The culmination of logic will come with the advent of VHSIC technology which is primariiy aimed at signal processing applications. In the microcomputer field the recent announcement of the $\mathfrak{i}$ APX-432 Ada 32 bit microcomputer by Intel Corporation is just the forerunner of an expected set of competitive VLSI computer systems.

Specific signal processors should also be completed within the general time frame of the ALSP as indicated in Figure 60. These developments, however, as previously indicated are not immediately applicable to the ADSP unless they have sufficient maturity, which should come at about the time the ADSP is completed.


Figure 59
IC Technology Projection


Figure 60
Signal Processor Architecture Projection

### 2.6 ADSP SCHEDULE AND COST FACTORS

The nominal development time for the ADSP has been established as three years. However, changing the program schedule will affect the expected cost as illustrated in Figure 61 which was derived from outputs of the RCA PRICE* program for the ADSP design. However, at shorter schedules both costs and risks tend to increase until at about 18 months the risks are too high for rational consideration. A 36 month schedule provides a higher cost, but the additional time gives a higher success expectation.


Figure 61
Effect of Schedule on Cost

If a two phase program is planned for the three year effort, the expenditure rates in Figure 62 will be obtained. The first 24 months of the job includes all of the design and first piece test work plus the integration of all the elements into a partial working system capable of doing all ADSP functions at a non-real time rate. The two-year design costs include all of the cabinetry and control for the full performance system. The final 12 monch period then involves purchasing additional parts and modules and integrating them into a full performance system.


Figure 62
Typical Expenditure Rates

### 2.7 IMPLEMENTATION PLAN ALTERNATIVES

The recommended schedule for the ADSP is given in Figure 63. It reflects the two phase deielcpment outlined in Section 2.6.


Figure 63
Recommended Schedule

The question of structuring the ADSP as a partial (non-real time) implementation with a four year development cycle was addressed. Full capability would be achieved with later add-ons. Alternatives for this
scenario are listed in Table 35. The first option is essentially to stretch the two-year development cycle of Figure 63 to four years. While this would achieve all program performance objectives, it would result in a cost increase (not including inflation) of 20 percent. In addition, the four year program could produce a technology deficient design unless the hardware design details were delayed until the last two years.

Table 35
Alternative Trade-off

## RELATIVE <br> COST

OPTION (AYEAR) COMMENTS

1. NONREAL TIME

FULL FUNCTIONS
2. (a) NONREAL TIME, REDUCED

PROGRAMMABILITY
(b) CASE (a) PLUS REDUCED GROWTH

- WOULD DEMONSTRATE ALL PROGRAM OBJECTIVES - FOUR YEAR PROGIRAM IMPOSES 20\% COST PENALTY - FOUR YEAR PROGIRAM MAY BE TECHNOLOGY DEFICIENT
0.73
- REDUCING PROGRAMMABILITY EIIMINATES ONE OF KEY PROGRAM OBJECTIVES
- TEST FACILITIES, BOTH BUILT-IN AND MODULE, ELIMINATED
0.60 - LOSS OF PROGRAM OBJECTIVE OF EXPANDABILITY TO FLLL REAL TIME SYSTEM WITHOUT ANY DESIGN
- FILL SYSTEM CAPABILITY WOULD REQUIRE RE'PACKAGING OF HARDWARE

3. DELAY HARDWARE ? TOTAL COST IS UNKNOWN -- COULD BE HIGHER THAN ALTERNATIVES

- APPLIES MOST ADVANCED TECHNOLOGY
- hardware development better matched to CURRENT MISSION TIMING

The second option in Table 35 would involve reducing the level of programmability and the built in test and module test facilities.

This would eliminate one of the key program objectives, programmability, and the elimination of test facilities would make the equipment very difficult to maintain. A substantial cost reduction (.40) is achieved if the previous two factors are eliminated plus growth capability. That is a partial system, with its limited control, would be designed and built. A large disadvantage to this approach (2b) is that future expansion would require design effort and thus cause much greater ultimate costs.

The final choice would be to delay the hardware development. While the ultimate cost of this approach is not known, it does have the advantage of applying the most advanced technology, either custom built or purchased. Also since the SAR missions which might use the ADSP do not occur until 1986 and beyond, a delayed program would be more in line with mission timing.

### 3.0 CONCLUSIONS/RECOMMENDATIONS

## General Recommendations

1) The FFT convolver algorithm is the recommended choice for a programmable SAR processor not having specific environmental requirements.
2) Parallel architecture is recommended since it provides modular structures which are matched to program needs in terms of incremental implementation and growth and the thrust of technology.

## Performance Results

1) Equivalent performance can be achieved with the FFT convolver, step transform subarray, or time domain azimuth processing algorithms.
2) The performance of the step transform subarray technique is a function of the subarray overlap factor.
3) Range migration compensation without interpolation is not a recommended procedure.
4) Adequate interpolation can be achieved with a four, three, and possibly a two point inte:polator.
5) Very little range spreading is observed with any processing alqorithm, although some skewing of the resolution cell can occur in the subarray process.
6) The integrated sidelobe levels are comparable with either the subarray or FFT convolver approaches.
7) The mainlobe width (resolution cell) is determined by the weighting function as long as interpolation is employed.

## Algorithms

1) The FFT convolver algorithm offers a high degree of programmability with moderate hardware complexity.
2) Time domain approaches, while aroviding some good control features, require by far the most computations which translate directly to hardware size and cost.
3) Subarray processing offers a minimum hardware approach, but its control system complexity outweighs the hardware reduction for a highly programable, laboratory based system.
4) The subarray technique may be most appropriate for a dedicated, onboard processor where size, weight, and power requirements predominate.

## Architecture

1) Parallel architecture offers the desired system characteristics of modularity, modular implementation, modulier growth, high reliability, maintainability, programmability, and accommodation to new technology.
2) Pipeline architectures minimize hardware, but are difficult to make reliable and do not iend themselves to modular implementation and growth.
3) Current government and industry advanced signal processor developments are not ready for immediate use in a real SAK processor.

Integrated Circuit Technology

1) VLSI, VHSIC technology trends will have a profound impact on signal processor implementations such as the ADSP in tine 1985 time frame.
2) Between 1981 and 1985 a continuing array of improved integrated circuits will become commercially available including: 256K dynamic RAMs, $8 K \times 8$ static RAMs, 64 K PROMs, FFT chips, and advanced microcnmputers.
3) These developments will make on board processors realizable with modest size and power.

### 4.0 NEW TECHNIQUES

A modular parallel SAR signal processing design was conceived and designed which meets the severe performance requirements of the ADSP. The system, described in Section 2.5, is capable of being programmed to cover a wide, continuous range of SAR parameters by virtue of its use of the FFT convolver algorithm. The algorithm has been implemented in a parallel architecture which allows incremental implementation and growth, high reliability, programability, maintainability, and low cost. The design is adaptable to subsequent advanced technology infusion.

## REFERENCES

1. W. E. Arens, "DMSP Azimuth Correlator Implementation Architecture", JPL Interoffice Memor andum 3626-77-017, 12 July 1977
2. Rabiner, L. R. and Gold, B., Theory and Application of Digital Signal Processing, Prentice Hall 1975, p. 153.
3. RCA, "A Study of Linear Approximation Techniques for SAR Azimuth Processing", Final Repr $\bullet$ t, JPL Contract 955067, 1 February 1979.
4. J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Computation of Complex Fourier Series", Math Comp Vol. 19, 1965, pp 297-301
5. H. J. Nussbaumer and P. Quandalle, "Compilation of Convolutions and Discrete Fourier Transforms by Polynomial Transforms", IBM Tr. Res. Dev. Vol. 22, May 1978, pp 134-144
6. T. K. Troung, R. Lipes, J. S. Reed, and C. Wu, "On the Application of a Fast Polynomial Transform and the Chinese Remainder Theorem to Compute a Two Dimensional Convolution", JPL-TDA Progress Report 42-57, 1980 (submitted to IEEE ASSP Trans.)
7. J. S. Reed, T. K. Troung, and K. Y. Liu, "A Parallel Pipeline Architecture of the Fast Polynomial Transform for a Real Time Synthetic Aperture Radar Processor"
8. A. Peled and B. Liu, "Digital Signal Processing - Theory, Design and Implementation", John Wiley \& Sons, 1976, Appendix 5.3
9. B. Arambepola and P. J. W. Rayner, "Efficient Transforms for Multidimensional Convolution", Electronic Let.ters, Vol. 15, 15 March 1979, pp 189-190
10. G. D. Berglund, "A Fast Fourier Transform Algorithm Using Base 8 Iterations", Math. Comp. Vol. 22, 1967, pp 236-238
11. K. Nakayama, "A Mixed Decimation FFT Algorithm", submitted to the IEEE AASP Trans.
12. E. A. Hoyer and W. R. Berry, "An Algorithm for the Two Dimensional FFT", 1977 IEEE Int. Conf. ASSP, pp 552-555
13. C. Caraiscos and B. Liu, "Two Dimensional FFT Using Simultaneous Mixed Decimation", manuscript in preparation
14. C. Wu and K. Y. Liu, " A Pipelined Digital Signal Processor for Producing Real Time SAR Imaging" (Unpublished paper)
15. L. W. Martinson and R. J. Smith, "Digital Matched Filtering with Pipelined Floating Point Fast Fourier Transforms (FFTs)", IEEE Trans. Acoust., Speech, Sianal Processing, Vol. AASP-23, No. 2, April 1975, p. 222
16. R. P. Perry, H. W. Kaiser, "Digital Step Transform Approach to Airborne Radar Processing", NAECON '73 Convention Record, April 1973
17. L. W. Martinson, "A Programmable Digital Processor for Airborne Radar", 1975 IEEE International Radar Conference Proceedings, April 1975
18. C. Wu, "A Digital System to Produce Imagery from SAR Data", AIAA Systems Design Driver by Sensors Conference, Pasadena, CA., October 1976
19. T. M. McWilliams, L. C. Widdoes, L. L. Wood, "Advanced Digital Technology Base Development for Navy Applications: The S-1 Project" UC1D-1770S, Lawrence Livermore Laboratory, September 1977
20. Raytheon Co., "Advanced Onboard Signal Processor (AOSP), RADC-TR-80-243 Air Force Systems Command, Griffiss Air Force Base, N.Y., July 1980 (portions classified SECRET)
21. K. E. Batcher, "Design of a Massively Parallel Plocessor", IEEE Trans. Computers, Vo1. C 29 No. 9, Sept. 1980, pp 836-840
22. C. Wu, "Electronic SAR Processors for Space Missions", Synthetic Aperture Radar Technology Conference, New Mexico State University, l.as Cruces, N.M., March 1978

# APPENDIX - PERFORMANCE SIMULATION PROGRAMS 

## PART I - SAR SIGNAL GENERATIOM

ORIGINR PRATE
OF POUR QUALity
6
6
SARG.FOR
pinguam ro genehate phe mange mabs ano
PME phase mistony of a targei in a
SAR TYPE ENVINONMENT
VIFTUAL As (4996), Ag(4090)
VIRTUAG PEIGNA(2n48), TEMHB(2048) ORANGEN(4a96)
common Is.jpa/A(30)

CALL INSTA
CALL INSPAA
GALL MAPPOI,AG, PEMPA, PEMPB,RANGER,LMNS
CALL OUTP(TEMH2, TEMHYORANGER)
CALL MFIT(AI, AO, TEMPA,TEMPG,RANGER,LMN)
ENO


```
SAHGI.FOR
```

SAHGI.FOR
INSPRUCTION SET FOR TAE SAR
INSPRUCTION SET FOR TAE SAR
C IS PME SPEEO UF LIGNT E S.E ( M/SEC)
C IS PME SPEEO UF LIGNT E S.E ( M/SEC)
P! = 4.*ATAN(1.)
P! = 4.*ATAN(1.)
A(l) TRANSM\&TPEO FHEGUENCY ( Mmz ) Fi
A(a) WAVEGENGFM OF FF (M) C/FP b
*(3) velocily of suacechaft ( mM/SEC ) v
M(4) MANGE PO TAAGEP ( KM), R
A(S) gAMPLING HATE ( MAZ) FS
-(0) PAF ( NHZ)
A(Y) NO. OF MULSES THANSMITPED N
T(a) aANOMIDTM OF a mangt cegl ( maz) OW
A(H) Rativ Or FS/um
A(IA) POTAL TIME OF FLIGHT ( SEC ) NJPPF
a(li) CUNSTANY POR IHE MANGE WALK CNW
v**2/(eqR) (M)
-(IE) hange qeSUluTiON RH
Cノ(2n**) (M)

```

```

        HI/RATIO
        a(l4) CONSTANT FOH PHE PHASE CORREGTION
        4*RI*CHW/WL ( 1/SEC:*2 )
        A(IS) HAMM!NG WEIGHTING ACROSE TME ANTENNA WINDOW
        YES : & NO = O
    A(1b) NUMBER UF GANGE CELLS LOOREO AT
    A(II) range walk yES : NO : I
    A(i&) BANOASDTM OF THE SYSTEM ( MH2)
    A(lq) iange cell fMaY faHgef IS IN
    A(cも) NO. OF SAMPLES PER DOPPLEN FILPER
    A(2I) p&NGE CUQHECTION OPIION
        - NO CORNECTION
        WU SCHEME
        PhENCA
            3 STEP TRANSFORM
    A(2E) RATIO OH PME SAMPLING NATE TO TME LFM RATE
    A(25) LINEAR NANGE WALK
        NO : D YES : mAX AMOUNT
    A(20) mAXIMUM AMOUNP UF RANGE WALK
    A(2I) TYPE OF INTERPOLUTION FOR FNENCN SCMEME
        2 15 2 PT
        3 IS 5 PT
        4 IS a PT
    A(30) TEMPORY STORATE USED FGR TME DYNAMIC WANGE
        *****ーロ***********
        sugqOUTINE INSTR
    ******************

```
\[
\Delta(s) 01.05
\]
\[
\operatorname{AEAO}(0,11) \quad(1)
\]

WHIPE(B,30)
fogmapi: mange to pme parget (nm) (old
A(0)assu.
AEAU(0.11) A(a)
0
    - (5) e2e
    AEAO(0.11) A(s)
    WRITE(0.50)
so FORMAP(\% PKF OF BEAMS (KMZ)" 1
    (10) 180
    NEAU(0,11) A(0)
    WHITE(0,00)
00 FOGYati POYAL NO. of beams layed UUWN • /)
    (1) 1004 b .
    AEAO (0,1i) (1)
        CUMMON /INSTH/A(I)

        Cabl assigncs, LP:')
        CALL ASSIGV(2,'SANG,SNCP, O, 'SCRP, I2)
    DISR FILE FOH PME PFT GAW UAPA
    WNIIE(0,5)
    FORMATE NAME OF PHE OISR FILE */J

    4SA the guestions
    WHITE(0.10)
    funmar ( pranymitito pheouency (mhz)'ノ
    A(1)alasu.
    NeAu(0.11) a(1)
    FOMMAT(F10.0)
WAVELENPM C/A(I)
    -(2)as.tez/a(1)
0
6
6
\(c\)
6
0
6
6
```

        NN1\E(6.76)
        *UHMAPP' GANONTUPIA UF A RANGE CEGL (MHZ)",)
        A(A)=14.
        MEAU(0.11) (t)
        WHIPE(B,BO)
        FURMATP' MAMMING NEIGMT.NG ACROSS ANYENNA WINDOW %/
    l
        A(15)0i.
        MEAO(6,11) (13)
    WHITE(6,90)
    funmati* NO. OF RANGE CEGGS lOOKEO AT PER BEAM */J
    A(10)=20.
    4(|0)-b.
    NE.AD(0,11) 4(10)
        WRITE (0,95)
        FUWMATI: DO VOU WANP RANGE NALK */
        1 - YES NO NO 1 1/J
        A18)=0.
        NEAU(0.11) (187)
        w^{1&(0,4)
        *DDMAT(" GANOWIOPM OF THE SYSTEM (MH6) "1)
        m(10) E1%.
        NEAO(0.88) A(10)
        WMPE(0.90:
        PONMAPIO GOCAIION OF THE TANGEP WHICN RANGE CELG '/I
        4(19)-3.
        *&4J(b,81) 4(89)
        WNITE(6,00)
        Funmart* pange cORRECTION ORTION O/
        & NONE O WU & CANO 2STEP I P/1
            A(21)El.
            A(c|)|e.
        GEAO(0,11) M(21)
        IP(A(2I).NE.2.\ 60:0 110
        wRITE(b,ise)
        FONMAY(" PYPE OF INYEWPOLAPION Z OR a P/I
        日gAU(H,1:1)A(4T)
        CONTINIJE
        GAPIO OP PHE SATSLING MATE TO PME DANGE CELL BANDWIOPM
        A(9)0A(5)/A(0)
    TOYAL PIME OF TME PLGGMY (SECS)

```

```

    DIEG. MTTAN(!.)
    A(13)=P!/A(D)
    C
CONSTANT FOQ TME PNASE COHRECTIDN
A(14)=0.*P\&|(A\&\&)/A(2)
A(2y)00.
WNITE(S,1050)
FUHMAT\&' GINEAR RANGE NALR NO O YES MAX AMOUNF '/)
REAU(0,11) A(25)
C
NE TURN
END

```

\title{
ORIGINAL PAEE IS OF POOR QUALITY
}

```

    S|NG|A.FOR
    ```

```

                                    SUGROUTINE INSTRA
    ```

```

    COMMON/INSTH/A(I)
    WRITE(S.!(#0%) (A(8).8:8.7)
    l
    VEGIJCITY OF PME SPICECNAFP (KMISEC J F.IPEIS.E/
    - hange to ime ranger ( KM) ',iveds,g/
    - SAMPLING HATE IN NANGE (MHZ) OIPEIS.A/
    - PMF IN ALSMUPM ( KMZ J O,IPEID.B1
    - NO. UF PULSES LAYEO DOWN O,FT.1/I
    C

```

```

FUDMAT(" HANDNTOTM OF A RANGE CEGL (MNZ) 'IPEIS.8/
1

```

```

    * RANGE WALK CONSTANT i M/SEC**E J P,IPEd5.8/
    * hange mesOquTION ( M ) '.dPEIS.B/

```


```

C
1020
NW\&TE(5,102g) (A(I),I=15,89)
FGRMAT(" MAMMING NEIGMTING ACROSS ANTENNA APERTURE*,

- ves e l NO © *,Fg.il
- NO. of range cellss TO PHOCESS 0.f5.I/
* mange halr yes e a no l %fS.l/
- UNUWIOTM OF THE SYSTEM ( MHZ % O,IPESS.8f
- LOGAT:ON OF PARGET RANGE CELG NO. ',IPEIS.8//J
C
DELLFMsz.*A(3)|*2*A(1G)/(A(2)*A(4))
OH=A(O)/A(1)
UP\&こDEGGPM/OF
A(?N)= JPd/4.
4(2己):A(0)/DELLFM


```
FOWMAPG GMANGF IN FREQUENCY ACNOSS W&NDOW & RMZ % '
    IPL15,日,* SAMPLING NATE (KMZ?!.
    &PE&5.0/
    - NO. OF AMZ PER FFY CELG O,IPE&5.8.
        - NO. GELLS PKN DOWPLEN FILPEH P,IPEIS.O/
        - HATIO OR SAMPLING MAPE TO LPM RATE .JPEI5.8/1)
```


# ORG: ${ }^{2}$. in. <br> of pock quality 

```
C
```



```
C
l04S FIHMAP(' PYOE OF INTENHOLATION TWO ON FOUN ',F5.1//I
C
1050 FUNMATP' LJNEAR RANGE WALK AOOEO O/
AHIPE(5.1050) A(25)
    I NO : VES m max amuUNT', FT.E/I
C
        RETUNN
        ENO
```


# ORIGINAL PAGE IS <br> OF POOR QUALITY 



```
C
C********************
    VINPUAL AI(I), QANGEG(I),AG(I)
    VIATUAL PEHPI(d),TENPQ(I)
    CUMAUN /INSTM/A(I)
C
    TjME=-d(10)/2.
    TSTEP={.E-3/4(6)
    YPINALSOPIME
C
```



```
CUNWMa(1&)
OEtPG:EA(4)
OGLPE#OELTAR#R.
CONSR=a(l2)
COMSA=A(13)
CONS*A(80)
C
NPULSEEA(1)
INANGEAM(10)
|NALNEEA(IP)
IOFFSYaA(19)
INUQPIVA(d&)
ILINHEA(8S)
IF(ILINN,NE.O)CONLO-Z.由A(25)/(A(T):TSTEP)
```



```
\W=O
1 CONPINUE
5
```

ONIOINAL PAGE IS
Of POOR QUALITY

 －No．wanteo ．\＆B．NO phocesseo ．IO／／l
IF (ILINR.NE.OS HANWLRARANWLR • CONLITBME
PIND the phase
DNASEEPSORCONS
8nElw -
A1(IW) APMASE
GG(IN)ANANWLK
TIME TIME PSTEP
SP(:N.LT.NPULSES GOPO :
Fino the min amo max ranee malk
$x \operatorname{MAX} \operatorname{AO}(1)$
XHINaAO(i)
6max=l
LMINEI
00 182,NPULSE
lF(AO(I).bT.xMAX) GOTO 2
xMax=AO(b)
LmAXel
lF(Ag(l).GT.XMSN) GOTO a
xMINAAO(I)
LMINEI
cuntinue
normalsze phe map to the minfmax of the range walk
nRANGEPIRANGE
LMNEI
IF (XMAXEEO.D.) GOTO 9
XXMAX:IF\&X(XMAX)
IF (XXMAX, NE, XMAX) XXMAXEXXMAX - 1.
NEANGEBNHANGE IfIX(XXMAX)
if (XMIN:GEDO.) GOTO 0
$\mathrm{XSa-XMIN}$
XXMINIFIX(XS)
IF (XXMIN,NE,XS) XXMINEXXMIN + 1 .
NRANGEQNRANGE XXMIN
LMNE-XXMIN * 1.
IF(ImunPt.éo.l) guro 9
NHANGEENRANGE • 2
LMNELMN - 8
CONIINUE
WRITE(O,OUBO) LMIN,XMIN,LMAX,XMAX,LMN,IRANGE, NAANGE

かのnののロ
COMPUPE THE SIGNAG FOR RACN GEAM POSITION
C AS FUNGTION OF RANGE
C
C
is conitnue
6
【W8!
It Walom
PMASEEAI(L)
CSACOS(PMASE!
SSasIn(HASE)
KANINLKJAD(L)
CONTINUE
${ }_{c}^{82}$
TEMPI(IN) 0 .
6
xIWEItW- LOFFSp
XXIWOXIW - HANWLR
CabL SINX(VI, RXIW, CON $3 x$ )
xyIwExXIW + OELTAR
Cabl Sinx (V2, xxiw, CONSX)
x×twexxiw - OELP
CALL SINX (V3,XXIW,CONSX)
$c$
$c$
$c$
6
${ }^{6}$
20

| $c$ |
| :--- |
| $C$ |

$c$
$c$
6

LEL 1
[F(L.LE.NPUGSE) GNYO 10
$\stackrel{C}{c}$
100
${ }_{c} \mathrm{C}$
c rearrange the range
GENPULSt/E
UU 110 JE\&,NPULSE
6.6.1
SP(L.EPGAPULSE) LEI
TEMPPEAO(J)
RANGER(L) ETEMPP
cuntinue
A(30) enkAnge

```
            A(2n)=xMAx
    WRITE(0,18Yq) LXLX,NHANGE,NPULSE
    POHMAY(//" NU. PHOC ',16," NHANG ',10." NPULSE *,80/18
        be Purn
        ENID
C
c
C
    SUBROUPINE SINX(V,X,CONST)
C
vei.
DOECONST:X
Datssames(00)
IP(UAESS,GT.i.E-8) VaSIN(OD)/00
        NEPUNN
    ENO
```

```
O
C SARLS.POH
```



```
c
                                    SUGFOUIINE OUPP(AI,AO,NANGER)
C
G**&*******************
c
c
    VIOPUAL AI(90),AO(50),RANGER(1)
    REAL IA(ab)
    COMMON /INSTR/A(1)
6
```



```
    xMAXOS.
    REAO(0.11) XMAX
    fonmat(ri0.0)
l
    IF(MMAX.NE.O.) GOPO 10000
c
0012 1:8,20
lc ba(l)=100200.
C
        PENINO Z
        NPulSEEA(1)
        mHangega(30)
FINU PME mAXIMUM KEEP THE AZIMUPA ANO HANGE BIN
        xMAX=O.
    DO :U0 JEI,NPULSE
        READ(E) (AI(I),ICI,NQANGE),(AQ(I),I=1,NAANGE)
C
C
        UO 50 I:I,NRANGE
                    A&(I):AI(1)**2* AO(8)**己
                    IF(AI(I).bl.xMAX) GOFO 50
                    EMAXEAL(d)
                    MJaJ
                    K&=1
S
go coniINUE
C
I00 CONTINUE
ののாのロの
SCALE ALL PHE POINTS TO MAX PUT IN LOG SCALE
```


# omgral thle is OF POOR QUALITY 

```
            NHANENRANGF
        IF (NHAN,GP.EJ) NHANEEO
C
```



```
    & ISF MAMIMUM VALUE (O OD REFEHENCES OIPEIS.O/1
    5x,ब与(2x,!2,2x),0 WA6R:1)
C
C
C
C
C
C
C
O
150
100%
C
200 conpinue
C
C
80000 CONTINUE
C
    REPURN
    ENO
```



```
    SAH6G,FOR
```

    SAH6G,FOR
            WR&TES OUP A PSLE IN PHE FOLGOWING PORMAT
    REC0RO \&
GRAAY I INIPIAL CONFIGUNAYION (ZO REAG ELEMENTS)
RECORDS E TO RANGE CHLLS \& I
\& COMPDNENT ANRAY OF PFT ANO O COMDONENT AMGAY OF FFT
(PON TME SPED TRANSFOHM IT IS NUN FFY BATA?
EAGM AXRAY IS OF LENGTM OF SEAM POSITIONS NEAG AHNAYS
IP NP HEAMS IS NOT NOWER OF Z THEN THE LENGTM
OP INE AMHAYS IS NEXY HIGHEST POVEA OF \&
REGORO RANGE CELLS * 2
RANGE WALK FON EACH BEAM POSITION
SUQNOUIINE PFPR(AI,AO,XYI,XXO,MANEER,GNN)
C
C*****************
VIRTUAL XXI(50,42), XX0(50,40)
VIRTUAL MANGEG(1),AL(1),AC(1)
OEAL XI(50), x(4)50)
COMMON/INSYH/A(SO)
monnom
GRING IN EACN RANGE ROA TO MAKE UP
THE AZIMUTN ARRAY
NPULSE:ACTI
SWALKEA(II)
{W\&NU=A(15)
NRANGEEA(16)
1SGHMSA(21)
INTHEA(27)
IRANGEEA(30)
Q\triangleT\0:4(22)
KMAX*\&A(26)
MNPOFLOAP(NPULSE/ZJ/HATIO
JNPEXNP - %.5
O NNITEIE,I8999% JNP,NPULSE,XNP,NATIO

```

```

C
CONSPa|(2)**2*\&(4)/(8.*A(3)**2*\&(12))
CONST*CONSTM\&.E-S
PSTED=A(B):1.E+3/FLOAP(A(T))
OWHITE(t.5500) CONST,FSTEP
SY08 FORMATG'COST .1PEIS.8.'FSTEO O,IPESS.8/1)

```

\section*{ORIGINAL PAEE IS \\ OF POOR QUALITY}
©

WAIPE UUT THE CUNFIGURATION
WRITE(B) A
WHITE(O,BEAC) NPULSE, IRANGE, \(\triangle(1)\)

PUP ME GANGE WALK AMOUNT IN TO FILE POR TME STEP

\(00100,6 \mathrm{I}\) IRANGE
REWINO ?
00 EO JU\&,NPULSE
READ(E) (XI(K), K@I,IRANGE), (XQ(K),K日I,IRANGE) Af(J) \(x\) (fl)
AO(J) axoll
6
50
50 CONTINUE
WINOOW THE DATA
IF (IWINO.NE.I) GOTO 5S
XKK=0.000679y
XNN:=NPULSE/E
DO S2 J: 1. NPULSE
XXEXNN XKR

AI(J)EAI(J)*VAL
\(\triangle C(J)=A D(J)\) VAC
XNNEXNN 1.
CONPINUE
52

CONPINUE
CUMPUTE PME FFP O: PII , ANGE CEGG
IF (ISCHM,NE, 3 ) \(\quad\) R,L. FFTV(NPULSE, AI, \(\triangle Q\}\)

WAITE OUY TME ANRAYS TO PME FILE
 WHITE(B, OQUE) NPULSE, 1
```

BI.% CUNTINIJE
IFIISCHM.EG.OS GOTO 18OOU
IF(ISCWM.tG.S) GOTO 13000
C
C
\&\&9 . OO 120 J=1,NPP
REWINO !
REAU(8)
OO \&UG KEI,IRANGE
IS\#:STAAT
gEAO(l) (AI(L),L=!,NPULSE),(AG(L),LE\&,NPULSE)
C
c
00 105 LLEI.NPASS
xxI(x,LL)=A!(\$3)
XXO(K,LL)=AO(IS)
|SH!\& |
C
l0S CONIINUE
C
106
continue
c
gorregt tme araay in range
C
C
JJ|ISTART
WALKH=O.
IF(IWALK.EO.1) GOPO 20S
dF(JJ.GT.NH) JJENPPE - JJ
valuta(FLOAT(JJ - i)mFSTEP):-2\#CONST
wAGKHEvALUE
\$03 IFIILINH.EG.0) GOTO lO4
\&F(ISTANT.GT.NP) JJ= ISTAMT - NMULSE
VALE=CONL*FLOAT(JJ - \&
NALKHawALKR + VALE
104
CONPINUE
C
IF(ISCHM.tQ.2) GOTO 100

```

```

                    ORIGINAL FACL: IS
        If(mALAM.ti.0.) {H|Ee|wU
            ISE1517. IWU
            IEOIENA *WU
        IF(IS.GT.0) GOPO 1000
                1301
                IEONHANGE
            (i)TO :G7
    1060 \&F(IE,LE.IHANGFS GOTO 107
IESIRANGE
ISEIMNGR

```

```

\$01 WGITE(2) (XXI(LG,K),GLEIS,IE),(XXO
6060 FOPMAP(It,IX,IPEIS,0,J(ix,I5)j
G010 100
cONTINUE
R6GEK
GALL CAND(XXI,XXA,WALKR,NHANGE,KLL,INPP,ISTR,INANGES
C
WFITE(Z) (XXI(LL,K)OLLEI,MRANBES,(XXO(LLOK)ILLSI,NNANGE)
C
IQQ CONPINUE
WKIPE(0,QOIQ) ISTARP,JJ,VALUE,VALZ,WALKR,IS,IE
FORMAP(C(1x,14),3(8x,1P(15.8),2(1x,14))
ISTANTEISTARY - I
C
18 CUNPINUE
C
1\& CONPINUE
C
\&SO IF (NOVEN.LE.D) GOFO 130
C NOOEI
NPASSANOVER
NOVEHEO
6070 189
C
:39 CONTINUE
C
REWINO I
@EAO(8)
C
C
DO \&SI JUINNPULSE
GAU(己) (XI(L),LE\&,MMAMGE),(XO(L),G:I,NRANGE)
A!(J:@x{(K)
AO(J):xO(K)
C
IDI CONTINUE
C
WR\&TE(\&) (AI(L),LE\&,NMULSE),(AO(L),L-8,NPULSE)
C
140 CONTINUE
IWOQJ CONIFNUE
C
Hug\& FN{TE(0,GASI) NPULSE
C
C
calG ClOSE(!)
GETUNN
ENO

# ORTBNALDNE PG <br> OF POOR QUALIT: 

```
C
C
C
c
C
0
5050
5055
    SARG4A.FOR
    SUYROUPINE CANO(AI,AG,NW,N,K,INTP,ISPH,IRANGE)
    Y&RPUAL A&(50,40),A0(50,40)
    spNaRW
    UEGIAERM - FLUAT(IRW)
        LE!
    jelsip - gam
    dF(INTO.EM,u) J.J -!
    &゙(INPQ.EQ:3) JaJ - I
    jP(1)ELTA.LP,0.) JaJ - 1
        BFCOELTA.GP.O.) OELTAES. * DELTA
            I+(J.6T.&) J&I
            IAAN(jeIFANLEE - N
    WMITE(5,5&50) K,J,ISTR,NW
    WMIPE(S,SOSS) (JK,AI(JK,K),JK=I,IHANGE:)
    FORMAP(S(IB,|x),IPEI5,8)
```



```
    IF(INTROEG.E) GOTO ita0
    IF(INPP.EQ.3) GOTO 300
        IRANG=IRANG - ?
        IF(J.GT.JAANG) JaIHANG
    OLLTAESUELTA#UELTA
    OELPASaDELTAC*DELTA
C
C
C
&
C
C
C
C
c
100
C
            OEbTAEz&. - OELTA
        jF(J.ST.iNANG) JaIRANG
    CONYINUE
108
C
    valuegaugbiazaal(J.K) * OtLTA*Al(f*&,K)
    VALUE({aUELTAE*AG(J,K) * DELTA*AG(J*&,R)
C
```

SF(L.LE:N) GOTO 101

## 300

G010 10000
${ }^{6}$
c
IkANGEINANG - I
1F(J.GT.JYANG) JaIRANG
$c$
AMg EDELTA* (OELTA - 1,)/2.
AEn: - DEGTAREZ
Asogliancortra + 1.)/2.
6
501
c
CONTINUE

VALUEGAMI*AS(J,K) - $A$ AAAG(J+b,K) + AbAGU(J+C,K)
c
Af(L, $x)=v a l u z i$
AG(L, X) EVALUEO
c

$$
J=J \bullet 1
$$

6.6.1

IFPL.LE,NJ GUPO 301
6
6
6
6
6
deare CONTINUE
O WKIPE(5,jESS) (JK,AI(JK,K).JNAIRN) hetuan ENU

```
C:
        FFIV,FOR
        pfr algutagfmm fur vietual ahrays
        SUEROUT&NE: PFTV (NUNGR,XT,YY)
        VIRTUAL XT(I),YT(8)
    Numge is phe vo. or points of govo data
    IF NOT A PONEH OF TNO PAO TO NEXT HIGHEST
    WGTH Z&kuS
        NCPON=1
        dtempenumgh
        |TEMP:ITEMP/E
        &F(ITEMP.LE.l) GOTO $0
        MPPOW=NCPOW + \
        GOTO &
        GONPINUE
```

```
C
13
C
C
C
15
C
2a cuntinue
C
C
    MONEPOW
    OO ong Loaf,M
    LMX=2:-(M-Lも)
    LIXEd*LMX
    SCL80.2#3145/PLOAT(LIX)
    OO OUO LMa&,LMX
    AHG=(LM-I)ESCL
    CECOS(AHG)
    SaSIN(ARG)
    00 O0| LIaGIX,N,GdX
    J&=LJ-6dx+LM
    Je=J&+LMx
    F1:xT(j) -xT(J2)
    TE=\T(J!)-VT(JE)
    xT(Ji)=xT(J&)+xT(se)
    YY(Jil)=Y(J&)+YT(J2)
    xT(Jこ)=C+i&*S-f!
```



```
O00 CONPINUE
    NuEEN/E
    NM& =N-I
    J|!
GS! GONVE
635
    NE2**NRPOW
    IP(N.GE,NUMUR) GOTO |3
        NaNe2
        NEPOWENEPOW * 1
    if(NUMAN.GE.N) GOTO 20
        N!anUMHR * !
    OU &5 GMENI,N
    XY(LM)=0.
    Y(LLM)=0.
    guntinue
    cuntinue
```

```
    00 b35 da&.NM!
```

    00 b35 da&.NM!
    IF(I.GE.J) GO TO 638
    IF(I.GE.J) GO TO 638
    T!axf(J)
    T!axf(J)
    TEarp(J)
    TEarp(J)
    xT(J)axf(I)
    xT(J)axf(I)
    yr(J)eyr(I)
    yr(J)eyr(I)
    NT(I):T&
    NT(I):T&
    YT(1)=P2
    YT(1)=P2
    A3NVZ
    A3NVZ
    fr(K.GE.d) GO TO 63S
    fr(K.GE.d) GO TO 63S
    j=J-K
    j=J-K
    k=k/2
    k=k/2
    6% 10 620
    6% 10 620
    ```
    Saj*R
```

    Saj*R
    numeran
    numeran
    GETUNN
    GETUNN
    ENO
    ```
    ENO
```

```
        AINUOV,FON
    VIKTUAL VEHSION
    mINIJOWING FUNGTION
    ARGUMENTS
    AHRAY IS THE AHRAY TO GE WIGJOWEO PINST LOCATION
```



```
        NUMGM IS THE NUMdER OR ELEMENTS TO UE WINDOWEO STARTING AT ARUVE
            IOPY IS TYPE OF WINDOW
    O NO WINDONING (RECT FUNCTION )
    I HAMMING WINOOW
    E GARTLETY #SNDOW (THIANGLE)
    3 mANNJNG WINUOW
    4 dLACRMAN wINDOW
```

                    SU甘ROUTINE WINOUV(ANRAY,ISTART,NUMHR, IOHY)
    6
Cッ*****************************
c
VIATUAL AKHAY(I)
6
6
IP (IOPY.EG.6) 601084000
XNENUMAH - !
ISPAPsISTART
IF(SOPP,EG.2) GOTO 200
c
PIEG.:ATAN(8.)
PIze己.\#P!
phaseupicixn
IP(IOPP.EO.4) GOTO 100
c
C1:9.s
C20.6.5
1P(IUPT.EQ.S) GOTO 10
c
C8:0.54
cさed.40
6
It CONTINUE
${ }_{6}^{6}$
00 20 Ias, NUMEA
$c$
yNEI - 1
WUAC: - CE-COS(PWASE:XN)


```
                    |STATEISTAP * !
C
C
GOTO 10000
C
IvO CUNITNUE
C
PmasEzopmasegz.
c ou l20 dei,numbr
C
YN:I - !
W026.42-0.50.COS(PMASE:XN) + 0.08*COS(PMASE&#XN)
Amtar(ISTAT)amD*AHhay(ISTAT)
    csrarelstap * b
C
c
    cONIINUE
    G0PO 10000
    6
2vo CONTINUE
6
C
C
    OO 2!tc I=\.IUP
        WOarguat(e*(I - l))/XN
        AW\tilde{AM(ISTAT):WD|ARKAY(ISTAT)}
        ISTATzaSTAT + &
C
210
    cONTINUE
        IUPIIUP * &
C
C
    00 己दg I=IUP,NUMGR
```



```
        AEAAY(ISTAT)BND*AHKAT(ISTAT)
            gSPapEISTAP | |
C
2ag gontinue
C
        GOPO 10000
C
10000 CONTINUE
6
                HETURN
                ENO
```


## PART II - FFT CONVOLVER AZIMUTH CORRELATOR

```
n@nornco
    MAPAFOR
    HYOGQAM TO MROCESS THE SAN OATA FROM SARG PROGQAN
        VIRTUAL AS((4096),ASC(41090)
        V&HPUAL FS(a*MB),F&(0096)
        CUMMON IINSTR/A(30)
        COMMON /PEMP/AT.(10)
C
6
C
    DO 1 101,9
        CALL UOPPLR(FI,PG,I)
        CALL SIGNALSASI,AS(G,FI,PO,I)
        CALL UUPPD(FA,FI,ASI,ASG)
C
l
c
    CONTINUE
    ENU
```


## page is of PCOR QUALIT：

MAPIOFOR
INSTRUCYION SEP FOR PHE SAR
CIS PME SPEEO OF LSGMT © S.E B (M/SEC)
P1 4. ATAN(I,
A(I) PGANSMITTEO FREGUENGY (MAZ) FT
( (己) WAVELENGTH OFFT (M) C/FP b
(s) VELOCPIY OF SHACECQAFY (KM/SEC) V
(4) RANGE PU TANGET (KM) $\quad$ (
A(5) SAMPLING HATE (MHZ) FS
A(6) PNF (KHZ)
(1) NO, OF MULSES PRANSMITPEO N
( (8) GANOWIDIM OF GANGE CEITL ( AMZ) BH
A(9) Hailo OF ES/Bm
A(IG) TOTAL P\&ME OF FLIGMT \& BEC ) NiONF
(II) CONSTANT FOG THE RANGE WALK CHW
$v * 2 /(2+N)$ (M)
- (\& 2 ) RANGE PLSULUTION RR
$C /(2 \oplus \otimes W)$ (M)
A(IS) CONSTANT FON TME NANGK $A M P L I T U O E ~ F U N C T I O N$
PI/QAF80
(10) CQNSTANT POR TME WNASE CORAECTION
OHIOCAN/HL (1/SECOH2)
A【IS) WAMMING WEIGNTING ACHOSS THE ANTENNA WINOOW
Y\&S - 8 NO - 0
A(10) NUMEEN UF AANGE CELLS LOOKED AT
A(II) AANGE WALK YES N NO I
(18) BANOWIOTH OF TWE SYSTEM (MWZ)
$\triangle(I 9)$ LOCAYION OF TME PARGET
A(20) NO. UH CELLS FOW EACH OOPPLER FILTER
(2!) RANGE CORHEGTION UPTIUN
O NDNE
1 WU
C CANUIANS
-(2己) RATIO OF THE SAMPLING *ATE TO THE LFM RATE
-(25) LINEAR PM WALK ADUEU NO OES MAX AMOUNT
A(ET) TYPE WINOOW ACRUSS FILTER
- (29) STAMTINF LOCATION IN FREQ OF PME FILPER
(30) STOPPING LOCATION OF TME MAYCHEO FILTER

SUBROUTINE INSTR
CUMMON /INSTR/A(30)
CALL $4 S 3 I G N(B$, •TY:')
NAME OF＇SBSK FIGE TO EE PRUCESSEO

```
```

WN{PE(0,5)

```
WN{PE(0,5)
FORMAT(" NAME OF O&SK FILS TO BE MROCESSED * OS
FORMAT(" NAME OF O&SK FILS TO BE MROCESSED * OS
CALL ASSTGN(1, 'OUMHY*,01)
CALL ASSTGN(1, 'OUMHY*,01)
nn
1000
WR1TE(5,1000) (4:[8),1:1,7)
```



```
    1
    3 WAVELENGTW (M) IIPEIS.8/
    - VELOCITY OR TME SPACECNAFP ( KM/SEC) OIPEIS.8/
    - qaNGE TO THE PARGET ( RM ) OIPE!S.g/
        - SAMOLING MATE IN RANGE (MH2%)!IPE&S.8/
    - Diff Of ine alimula ( KHz j P,dPEDS.8/
    - NO. UF PULSES GAYEO DOWN ,FF.:/I
C
WHITE(5,1010) (4(1).158,14)
1010 FOHMATP HANOWIDTMOF A PANGE CELL (MMZ % OIPEIS.8/
* GIIIO OF TME SAMPGING HATE IN AAMGE TO QANOWIOTM OF A O
*GGLLOOIPLIS.y/' TOTAL TIME OF LOUR (SEC) ',1PEIS.8/
```



```
- mange megOLUIIUN & M J ',IPEI5.8/
- CONSTANT HON AMPLITHDE PUNCTION O&PEIS.G.' (PI/NATIO) -/
```



```
C
C
OZO OUMATG' MAMMING WEIGMTING AGROSS ANTENNA APENTURE *
-ves l NO - ,F5.1/
* NO. OP RANGE CELLS TC PNOGESS *,FS.I%
* mANGE wALK YES a 0 NO . 1..F5.&/
* GANDWIOTM OP THE SYSTEM (MHZ % O,IPE&5.B/J
C
OELbFM*e.*A(3)**2*A(10)/(A(2)*A(4))
OL:A(G)/A(1)
*(20) =0ELGFM/OL
A(26)-A(20)/4.
C
C ABOVE ASSUMES WE HAVE POUR OUPMLER FILTERS
8030
WHITE(S,IWSOY DELGFM,G(G),DGOA(2O)
```

```
        10t.3.4/
    - NO. OF KMZ PEH FFY SAMPLE P,DPEIS.0.
        - NO. Of CELLS PER OUPPLER FILTEK •IPE\5.d//I
C
    WR!PE(5,1H40) A(2!)
1040 FONMATP' HANGE COHAECTION FACTON NONE:G WU: : ,
C
WFIPE(5,IRSU) A(25)
C
1855
C
1400 FORMAPG: TYPE OH WINODN ACHOSS MATCNEU FILPEN O,
```



```
    e ULAGKMAN a 4 011)
        \Delta(2!)=1.
O EEAO(0.106i) A(27)
106: FUEMAT(FIO.0)
C
IU6C FOQMATP: TYPE OF WINDOW ACROSS MATEHED FILTER P,
    1 , NONE : MAMMING : OAHTLETY : 2 MANNINE: 3 ",
    eglacmman a -,F5.2/1)
        RETURN
        END
```

```
Onnoncon
    MATR,FON
C*************&******
S Sugroutsne gen(ast,asos
C
C****&*ロ**************
C
NPULSE&A(T)
TSTEFE1,E-3/A(0)
TlME=-A(10)/E.
TFINALE-TIME
CONS=A(14)
C
c conTINUE&
C
                                    IF(S.GT.NPULSE) GOTO 100
C
TSG-TIME=TIME
                                    PWASE=TSG*CONS
                                    XX=COS (PNASE)
                                    ASI(l)*xx
                                    xxesiN(PHASE)
                                    ASE(t)exx
C
                                    TIME:TIME + TSTEP
                                    I=! * l
                    60101
C
10% CONTINUE
C
C
G WRITE OUT THE TOTAL WAVE TO A TEMPORY FILE
WRITE(2) (ASI(J),Ja!,NPULOSE),(ASO{J),JaI,NPULSE)
C
return
                    END
```


# ORIGINAL PAGE IS OF POOR QUALITY 

```
C
C
                                    SIJHOOUTINE OUPPGR(PI,FG,ITAG)
C
```



```
    vinfual fi(i),po(i)
    COMMON /INSTM/A(1)
C
            NPULSE:A(7)
            NUMOENPULSE/A
            IwlNOwoA(2%)
c
C
    NULL OUP PHE TIME mISTORy Of no INTEHEST
            KLEl
        Ir(IPAG.EO.I) GOTO 28
C
            REWINO 2
            AEAD(E) (FI(!),IEI,NPULSE),(FO(&),IEI,NPULSE)
のnonのnのna

        RUE(ITAG - I)*NUMB
        MLEKU * &
        OO 10 REI,RU
            FI(K)=0.
            FO(K)00.
10
        GONTINUE
        IF(ITAG.EO.4) GOTO 50
C
20 contINUE
C
            ku:ITAG*NUMG * &
            DO 23 KEKU,NPULSE
C
            FI(K)=0.
            FG(K)=0.
C
            CONTINUE
C
So CUNTINUE
```

```
C
C
C PUP MAMMg:S NEIGHIING ACROSS PILPER
C
        ISTARTERL
        CALL N(HOOV(FI,ISTAHI,NUME,IWINOTA)
        IgTAKPaKL
    CALL WINOCV(FG,ISTART,NUMG,IWINOW)
C
c finu tme spectrum of tme neference
C
4551 CONTINUE
                            NUMGF ENPULSE
C
CALL FFTV(NUMBR,F&,FOS
C
80000 CONPINUE
                KKKERU - KL * I
            WHITE(G,TAY) KKK,KU,KG
709 FORMAT(S(&X,IO)! DEL UP LOW 'S
CC
G STORE AWAY THE STARIING ANIS STUPPPINIG LOCATIONS
                A(29)=!
                            A(s0) anumBa
かのnの%
            mETURN
            ENU
```

```
C
c mathofur
C
c
C*******************
C
C********************
C
C
        VIRTUAG ASI(I),ASG(l)
        v!NPUAL F!(1),Fg(!)
        COHMON IINSTRIA(S)
        CUMMUN/IEMP/TEMP(&)
6
C
        NPULSEEA(7)
        INANGEOA(16)
                        ISTARP=A(2y)
                            IENUaA(30)
C
        REWINO :
        HEAO(1)
                        QEWINO 3
C
C
C
            mMAX=0.
            14N=0
                &NG=0
C
C
C
            00 80| d:SgIRANGE
        WNITE(O,B0日O) ITAG,I
8080 FURMAPGESIG DOPF PIIG.O HANGE ,IG)
C
    REAC(8) (ASI(J),J.&,NPULSE),(ASO(J),J,&,NPULSE)
C
C mulTIPLY GY PME qEPERENGE SIGNAL SPECPRUM
C COMPLEX CONJUGATE THE RESULT POR A SUUSEQUENT
c lFFiv
C
    OD IG JEISTART,IENO
C
    xx&*ASI(J)*FI(J) & ASG(J)#FO(J)
```



```
    ASI(J):xxl
    ASO(j)E=x\timesO
C
80 continue
C+
C
C NULG OUT THE NON-OVERLAPPING REGION OF THE SPECTRUM
```

```
C
C
\i CONYINUE
C IF(IEND,EO,NPULSE) GOTO SO
C
C
C
2)
C
30
C
    gQmpuTE PHE IFFf
        NUMABNPULSE
        CALG FFTV(NUMB,ASI,ASOS
つのロロ
C
xmaxaeg.
                    1XM00
C
C
    OO 40 J:1,NPULSE
        ASg(J)*ASI(J)**己 * ASU(J)**2
        lF(ASI(J).LT.XMAX) GUTO 40
            xMaxEASI(J)
            1\times140J
C
4O CONTINUE
C
    WRITE(S) (ASI(J),JEI,NPULSE)
    WGITE(S,In00) I,IXM,XMAX
    WHITE(b,1000) D.IXMOXMAX
```



```
        IP(XMAY.GPgmMAX) GOTO $00
            RMAX=XMAX
            IAR=I
            INGOIXM
C
        CONTINUE
8
        PrMp(1) anPul.SE
        itMN(2)a!AN
        TEMP(3)=ING
        TEMP(4)=RMAX
narr
```

g`!g%?!
NumbR \&S TmE NO. Or points of gools Data
IF NUP A PONER UF TWU PAO TO NEXP HIGHEST
WITH 2EROS
NPFOWEI
IIEMPONUMBR
ITEMPEITEMN/Z
LF(ITEMP.LE.I) GOTO 10
N2POWaNZPOW * I
GOTO 1
\$0 CONTINUE
NEZ**N2POW
IF(N.GE.NUMGR) GOTO \$3
NEN*E
NCPOWENZPOW * \&
CUNTINUE
lr(NUMAN.gE.N) GOTO 20
NIBNUMBR - I
DO \$S LMEN\&,N
XT(LM)=0.
V「(LMJ=0.
gonfInue
CONTINUE
MONZPOW
OO hen L0.g.M
LMXaE*(M-L0)
LIXac:LMx
SCL36.2Ag805/FLOAT(LIX)
00 EGX LME\&,GMX
ANG=(LH-1)|SCL
C=COS(ARG)
SESIN(ANG)
OU BAD LJ=LIX,N,LIX
JI=Ll-LIX\&LM
JesJ!+LMX
Tf=xT(J\&)=xT(Jこ)
Tc*ri(j1)=YT(de)
xT(J\&)*xT(JI)+xT(s5)
YT(J1)=Y(1J\&)*YT(J2)
x(JEJac*is*S*TC

```

```

    CONIINUE
    NVEEN/E
    NM&ON-I
    Jal
    ```
```

OU 03. lal.NMI
1P(l,CiE.J) 60 T0 \$3)
180x(J)
ferf(J)
y((J)Ex(1)
YI(J)EYT(I)
xi(d)arl
YT(I):TC
git KENVE
O20 CONTINUE
IP(M.GE.J) 60 P0 035
J=J=k
k=k/己
go 10 bed
65
jejrk
NUMGRaN
return
END

```
```

O

```

```

    WINUOV.FOR
    virfual version
    WINUOWING FUNCPION
    ARLUMENTS
    akRay dS phe amnay to be winouneo fihst location
        ISTART IS PME first element of the arkay to be winooweo
        NUMYR IS THE NUMdER OF ELEMENTS TO BE WINDOWED STARTING AT ABOVE
        SOPP IS PYME OF WINDOW
    0 NO WINDOWING (RECT FUNCTION)
    & HAMMING NINDOW
    E BARTLETY WINOOW (THIANGLE)
    3 MANNING WINUOW
    a alACxMaN WINOOW
    C***********************************
C
SUHHOUYINE WINDOV(AHRAY,ISTART,NUMAR,JOPT)
C
C***********************************
c
VIRTHAL ARRAY(I)
C
\$F(IOPT.EQ.0) GOTO 10000
XNENUMAR - \&
ISTATEISTART
1F(IOFP.EO.C) GOTO 200
C
P{|a.\#ATAN(\&.)
P\&zac.\#\#!
pmAsE:PITIXN
IH(IOPP.EQ.4) GOTO \$00
C
C8-0.5
ca=0.y
IF(IOPY.EO.3) GOTO \$0
C
C8=0.54
C200.46
C
CONIINUE
80
C
DO 20 8:8:NUMEP
C
XNEI - \&
WUECS - CE*LOS(PHASEMXN)
ACRAY(ISPAT):WD*AHNAY(ISTATG

```

\title{
SSPAPUZSAP*!
}


\section*{ORIGINAL PAGE IS OF POOR QUALITY}

PART III - SUBARRAY AZIMUTH PROCESSOR
```

ncnnnnnono
OT.FOR
PhOgRaM TO pROCESS THE SAH UATA FROM SAHG PGOGRAM
TMIS progham pquceSS THE vata uSINg THE STEP TGANSFURM
VIQPUAL ASI(4496),ASQ(4098)
VINTUAL F$4046).F[(4070)
        COMHON IINSTA/A(30)
        compuon /fay/P(l2s)
        COMMON /TEMP/TEMP(IO)
            CALG INSTR
        conTINuE
            IF(PEMP(9).GT.TEMP(8)) GOTO 80
                CALL GEN(FI,FQ)
                CaLL UEAAMP(FI,FG,ASG,ASB)
                CALL CORNCT(FI,FG,ASI,ASOS
                CALL FINE(FI,FO)
                CALL OUTHD(FI,ASI,ASU,FQ)
            TEMP(6)aTEMP(6) & &.
            TEMP(9)a TEMP(9) - TEMP(10)
        GOTO:
        continue
c
            AVEBTEMP(7)/TEMD(6)
        WHITE(b)IU00) TEMP(6)OPEMP(7),AVE
        WHITE(5,10U0) TEMP(6),TEMP(7),AVE
```

```
        & . &uE C.&PE&S.8//J
C
ENO
```
        DTIOFON
        Instruction sef fur ine sar
        C IS THE SPEEO OF LIGHT - 3.E A (M/SEC)
        PI - ©.AATAN(8.)

    COMmON GLUCK INSTR

    a(l) fransmitieu freguency (maz) fi
    A(2: NAVELENGPK OF FY (M) C/FT \({ }^{6}$
a(5: VELUCTIY OF SPACEGRAFI (KM/SEC) $V$
A(4) RANGE PU PARGET (KM) R
a(s) SAMDLING RATE (MHZ) PS
( $(0)$ PHF ( KHZ)
a(t) No. OF pulses pannsmitieo n
A(O) HANOWIOPM OF A NANGE CELL (MHZ ) BW
4(8) Watio or fis/am
A(ID) POYAL Plat OF FLIGHT (SEC J N/HaF
A(II) CONSTANT FOR PHE haNGE WALK CNW
$y=A 己(d e$ (i) (M)
-(se) qange resolution an
C/(द. CW ) ( M )
-(IS) CONSPANT fOH TME NANGE AMPLIPUOE PUNETION
plohatio
A(IU) CONSTANT FOR PHE PHASE CORRECTIUN

a(IS) MAMMING WEIGMTING GCROSS PME ANTENNA WINUOW
VES I NO - O
If Yes option Queytion prpe of window
a(sB) number uf pange cells loored ap
A(1) GANGE MALR YES a 8 NO 1
(1(A) GANOWSOTM OF THE SYSTEM (MHZ)
(19)
(20) No. Of cells for each dopplef filter
a(el) GANGE CORHECIION OPIION
- NONE
1 WU
canulans
3 STEP
- (22) qapio or fhe sampling frequency fo pme lfm siwepp freo
A(2S) CUHNECTION OF RANGE MALK 4 PT INTERP: 1 NUNE: 2
C PT BNTEHDOLATOH: 3 PT INTEH ©
A(24) STANTING SUGANAAY WHICH LOOR
1 IS FINST LOON SUB - !
2 IS 2NU LOUK SUB : TOPAL/A - I
A(25) SIZE GF PAE FIRST FFT
A(20) NO. OF POINTS IN THE OVERLAP OF SUBARRAYS
A(CT) NO. OF SU日AhHAYS POR TME ZNO PFT
(2(2y) SIZE OF THE ZNO OH rine fFi
A(29) NO. OF SugARRAYS CONTAgNING DATA

# ORIGINAL PACL IS OF POOR QUALITY 



```
C
```

NH\{PF(A,b)
FIKMAT(" S\&ZE OF PWE PINSTFFT * 11
AEAI) (6, 1) (23)

WNIIE ( 0,8 )
FORMATE NO, UP POZNYS UF PME OVENGAAM "/I
At.AU( 0,7$)$ (20)
midfte( 6,9$)$

MeAU(0,7) (21)
witre ( 6,10 )
FURMAP(* NU, UF POINTS \&N 2ND PFT • /
REAO ( 0,7 ) (20)

```
WHITE(S,IGOQ)(A(I),IE&,7)
WHITE(S,IUQQ) (A(I),IE&,7)
    * WAVELENGTM (mj OIPE&S.8/
    - VELOLITY OF YME SNACECNAFT (KM/SEC) O.OE&5.8/
```




```
    - PHF OF TME AZIMUTM (KHL ) PIPEIS.O/
    O ONO. UF PULEES LAYEO SOWN PFF.INI
```

${ }_{6}^{C}$
1
2
5
4
5
6
1010
1

- WANGE HESOLUTIUN (M) ', IPEIS.8/


$C$
1020
WHilé(5,1010) (a(1), 8=8,14)


- MaIIO OP THE SAMPLING WATE IN WANGE TO BANOWIOTM OF A.

GOMAT( BANONIDIM OF HANGE CELL (MHZ) *IPESS.8/
FOGMAT ${ }^{\circ}$ HAMMING WEIGETING AGROSS ANTENNA APGRTURE .
    - ves 1 NO © CFり.11
    - NO. OP RANGE CELGS TO PNOCESS OFS. $1 /$

    - BANOWIOTH OF THE SYSTEM (MM2 J 0.8PE15.8/7
$c$
1
3
3

DLSA(A)/A(7)
    - (20) = DELFM/UL
$\Delta(20)=A(80) / 4$.
$C$
$C$
ABOVE ASSUMES WE WAVE FO': O OOPPLER FILTERS
8030
WHITE(5,103E) OEGLFM, $4(6), O L, A(20)$

1
2
3
\&PEIS.G. SAMPLING MAPE (KMZ) O
1PE\&S.81
        - ND. OF MMZ PER FFY SAMPLE OIPEIS.A.
        - NO OF CELGS PEH OUPNLER FILTEK PIPEI5.8/1)
6
e

```
    WHIPE(5,1040) A(2)
1040
    1
```



```
1600
    l
    C - E MLACAMAN Y 2S DS TAYLORE.
```



```
C
0
8070
    FGRMATP//F IYPE OF WINOUW ACROSS EACH SUBAMHAY P/
    1. I MAMMING EHAHTLETY I NANNJNGO.
        & MAMMING E HAHTLETY S NANNING
```



```
C
```




```
        1 : NO. OF PUINTS UF UVEHGAR OF SUGARRAYS "FGOZ/
        Z * NUM&EN OF SUQAHAAYS USEU IN FINE FFT ",FG.2/
        - NU. OF POINIS IN THE SECOND FFT ',FO.2/1%
C
IvOOQ CONTINUE
C
    WYITE(4,1000)
    FUNMAT(//O TPME OF NINOOW AGROSS EACG APERTURE */
    - & NAMMING 2 GARTLEPY 3 MANNING */
                4(15) B%.
    RtAU(8,7) A(15)
    WPITE(5,1070) (15)
A(ZC):A(6)/DELLFM
ZgOO FJMMAT:'RATIO OF SAMPLING FREO TO LFM NATE 'IPEIS.8/I)
C
    WHITE(6,1090)
$O%% FORMAT! CORPECTION SCHEME FOR NANGE NALK O/
    & OFTINTEGP I NLSEE E
        2 PT INPENP S 3 PTINTER : 0/1
        A(2)g=1.
            GEAU(b,1) (23)
            WHITE(b,1095)
            FOGMAT(% SUGAHRAY STANTING POINT WHICM LOOK */)
                A(24)=8.
            REAO(B,7) A(24)
C
C
&O96 FUHMAT:" COHRECTIONSCHEML FOR RANGE & a PT 2 % WU -
    W*19E(5.1894) A(23),A(24)
    l FS.E/* SUOAHHAY STANTING DOINT.,PS.E//J
ッロッパ
    CALCULATIONS FOM THE OIPFERENT STARTING TIMES
C SLOPE TIME OVER LFM EXFANOEU
```

$$
\begin{aligned}
& \text { ORIGMIR FA, } \\
& \text { OF PCOR QUAL: }
\end{aligned}
$$

c

```
SGOREAA(IO)/OELGFM
```



```
    PME I2 IS UO PO OVEKSAMPLING PAOOING IIIN IEROS
        TWICE THE LEGNTN
        PEMN(10)-3LOPEAA(b)/(A(25)-4(20)/2.)
    stamtivg time
    PHE/4 IS POR PME PAOOING WIPM LEAOS
    aNU ENOING PIME
        TEMP(8)=TEMP(10)|A(20)/4.
        WHITE(0.8697)
        FURMAP(' SIARTING TIME ( b - UP )*/)
            xx&0.
        NEAD(0,7) xx&
        PEMP(9):-TEMP(8) XX1*PEMP(10)
0ncos
    PuT IN POR EXPGO
        PEM0(8)=(A.
        TEMD(10)=TEMP(80)=8.
C
    RuNNING SumS ano cOUNTEH
            TEMD(7)=0.
                TEMP(6):0.
C
L2O0 WRIPE(S,l2QU) (TEMP(I):I#n,10)
    l
        FONMAT(: COUNTENS P.E(IX,F5,0)
    { - ENOING AND STAHPINGGTYMES O,2(1X,1PE15.8)/
    C INCNEMENP EACN TIME OIPE&S.E/IJ
```

6
6
6
ISTEA(15)
1FमTI:A(25)
\& (HST.GT.0) CALL WP(TAYLOQ, \&FPTL.IST)
wWIPE(6,7891) A(30)
WHIPE(5.7日91) A(30)

WPITE(6, 345T) A(23)
3u5\% FUGMA: (I, TYPE OF INTENPOLATION OF5.I.


6
GE PURN

```
C OLGNZ,HON
C********************
C
C********************
C
            viRTUAL FI(I),FO(I)
            GOMMON /INSIH/A(I)
            COMMON IPEIRPIEMP(I)
のnnonncon
    GENENAIE OUP TME REPERENCE SIGNAL
    WH MAPGAED FIGIEKING IN THE ANGLE DOMABN
        GINEAH FM Wave PONM
            NPULSEEA(7)
                    TSTEP:I,E-3/A(6)
                    OELAYEPEMP(9)
                    1!MEE-a(|の)/E.
                    THINALE-TIME
                    IIME =TIME + OELAY
                                    IFINALEPFINAL - DELAF
                    CONS:A(14)
C
            WHITE(S,IDOO) TEYP(6),TSTEP,OELAY,TIME,TFINAL
            WHITE(O,IUQO) TEMP(O),TSTEP,OEGAY,TIME,TFINAL
1000 FUNMAPG/O STEP 10. H,FO.E,P PIME STEP:,IPEIS.B/
            | COAMSPONUSNG DELAY (SEC ) ',&PEIS.8/
                - INITIAL PIME |.IPEI5.H.C FINAL FIME 0,IPEIS.8//J
C
C
d=l
C CONIINUE
C
C
                TSOETIME*TIME
                    PHASEETSQ*CONS
                    xxecos(pmaseg
                        FI(1):xx
                    MxSSIN(PNASE)
                    6
                    TIMEETIME P TSTEP
                    1:& - 1
            6010 1
C
&0O CONIINUE
C
s
C
                NETURN
                        ENO
```

```
crmanano
    OYCN3.FOR
                                    SUGHOUTINE DERAMP(PI,F0,SI,SOS
C
        VIRTuAL FI(I),FO(1),SI(I),SQ(1)
        CUMMON /INSTR/A(I)
        NHULSEEA(1)
        IWINIOA(IS)
        NKANGEEA(30)
        IHANGEGA(86)
C
                    MEWIND 2
                    HEAO(1)
            HEAU(!) (SI(J),J=!,NPULSE)
        WH!TE(6.234) S!(1),5!(5!2),S!(!024),51(2048).
            S!(3040),S1(4098)
    1
        S\(3096),S((4096)
        OO SOD 12&,NRANGE
    get the sIgNal from the itm range cell
        READ(1) (SI(J),JE!,NPULSE),(SG(J),J={,NPULSE)
        DERAMP PME SIGNAL
            0O 5 J=&,NPULSE
        xxIa + Si(s)afl(J) + Sa(j).FO(J)
            xxG: - Si(J)*FQ(J) * SG(J)=FI(J)
            SI(J):xxd
            SO(J)exx(4
            CONTSNHE
    PERYONM THE ffi gTORE RESULTS AWAY ON OISK FILE & E
    USE. 50 % OHGOGAP
    PUI THE PHAEE CORNEGIION ON OUE TO OVERLAPPING APPERTURES
            CALL OVEM(S1890)
E
Soo CONTINUE
```

NPULSEAA(7)
ILEvEA(2S)
ILENEEA(26)
IWINO:A(ID)
15:1 dEBILEN numesu


Das(2b)/A(25)
pI=a.atan(s.)
PI2日2.*PI
PICD=PI2*O
6
6
6
8

## CONTINUE

$\mathrm{K}=1$
c
C
OU 10 J. 13.15
$A(x)=5 i(J)$
$A R(k)=S 0(J)$
KEK - 1
c
10 gontinue

1s:IS + ILENZ
IEAIE - ILENE
C

```
nnorin
C
C
C
le
if
6
C
C
C
C
C
20
C
C
C
C
C
C
C
C
C
C
C
C
put winvonlng across fag sugarmays
    IP(IWINO.EO,O) GOTO 1:
    DO &C RISOILEN
        Af(K)=AI(K)*TAYLUR(K)
        AG(x)EAG(K)-IAYLUR(X)
    continue
    GONTINUE
        CALL FFT(ILEN,AI,AO)
        If(NUMB.EQ,0) GOTO 20
            PHASE=PI2D*FLOAT(NUMB)
        00 18 KaR,ILEN
            XK=PNASE*FLOAT(K=1)
                SNSSIN(XK)
                CS=cas(xK)
                XXI&G:HAI(K) * SN&AD(K)
                XXG:CS*AG(K) - SN*AI(K)
                        AI(K):xxI
                        AO(K)=4xO
                CONTINUE
    conifnue
store away on oay slut * a
        WRITE(Z) (AI(K)gKa&OILEN),(AO(K),KI&&[LEN)
            NUME=NUME + !
    IF(JE.LE.NPULSES GOTO I
        A(c9)=Nums
        RETURN
            ENO
```


## ORIGINAL PASE RG OF POOR QUALITY



```
    OTO.FON
    cokmection of hange walk in the step pransform
                                    SUQROUTINE CORKCT(HANGER,A&,AO,OUMMY)
C
        VIATISAL AI(128,32),AO(123,32),RANGER(1),OUMMY(1)
        AEAL FF(250)
        CUMMON /INSTR/A(I)
nonnon
    vO HANGE COPRECIION ?
        IF(A(1)
        HaT10a.|(2己)
        mmgVERAT!O*A(25)/A(27)
        NHULSE:A(7)
        maxSUB=A(29)
        mxsmaxsue - a
        MINSJ#E&
        NMCOEFEA(25)
        NMC3NMCOEF/Z
C
C
            NEwIND &
            READ(1) XXI
            NEAO(1) (OUMMY(I),IE&ONPULSES
E
C
c reqroer ime array
c Fino the min
C
        XMIN:OUMMY(I)
            LENPULSE/C
        0051E&ONPULSE
                LBL & &
            IF(L.GT.NPULSE) LE&
                HANGER(I):OUMMY(L)
            {F(XMIN.GT.UUMMY(L)) XM&NADUMMY(L)
        CONTINUE
C
c
c
    Fino fme location of pme zero fange celb
    LMNEI
        lF(XMINOGE.O.) GOTO a
            XS=-XMIN
        XXMINEIFIX(XS)
        IP(XXMIN.NE.XS) XXMINEXXMIN * 1.
        LMNEOXXMIN * &.
        LMNELMN - &
6
```

```
のにの
C
C
C
S
80
C
C
C
C
12
C
C
6
C
C
C
C
```

```
    NHaNGEA(30)
```

    NHaNGEA(30)
    IHANGEa(10)
    IHANGEa(10)
    ISCEMEaA(23)
    ISCEMEaA(23)
        RENINO 4
        RENINO 4
        OO 10M ICl,MAXSUB
        OO 10M ICl,MAXSUB
    read in THE \& TH subarray fon each ranges cell ano fhen
read in THE \& TH subarray fon each ranges cell ano fhen
OU IME CUNAECTING WRITE THE RESULTS OUT TO * 4
OU IME CUNAECTING WRITE THE RESULTS OUT TO * 4
NEW\&NO 2
NEW\&NO 2
1!|!-1
1!|!-1
IF(1I.LT.I) GOPO 10
IF(1I.LT.I) GOPO 10
OO 5 ja8.18
OO 5 ja8.18
REMO(2)
REMO(2)
CONPINUE
CONPINUE
DO दQ J:I.NNANGE
DO दQ J:I.NNANGE
REAU(2) (A\&(K,J),KEI,NMCOEF),(AG(K,J),K={,NMCOEF)
REAU(2) (A\&(K,J),KEI,NMCOEF),(AG(K,J),K={,NMCOEF)
IF(J.EEQ,NKANGE) GOTO 20
IF(J.EEQ,NKANGE) GOTO 20
vO l2 kas,MX
vO l2 kas,MX
fEAO(2)
fEAO(2)
CONTINUE
CONTINUE
DO PHE CORAECPION WHOLE ARHAY AT A TIME
DO PHE CORAECPION WHOLE ARHAY AT A TIME
STANPOI
STANPOI
INNGENRONGE IRANGE
INNGENRONGE IRANGE
IHNGCOINNG * \&
IHNGCOINNG * \&
INNGLEIHNG
INNGLEIHNG
I\#NGEIRNG - E
I\#NGEIRNG - E
IRNGLJIIRNGL - \&
IRNGLJIIRNGL - \&
ISTA=-LMN - 2
ISTA=-LMN - 2
00 30 J=1,NMCOEF
00 30 J=1,NMCOEF
ISTARTESTAAT * D.S
ISTARTESTAAT * D.S
IP(START,LP.D.) ISTANTESTART - O.5
IP(START,LP.D.) ISTANTESTART - O.5
IF(LSTAET.LT.I) GOTO SS
IF(LSTAET.LT.I) GOTO SS
IELMENMEAISTART
IELMENMEAISTART
HANWLKEHANGER(IELM)
HANWLKEHANGER(IELM)
{F(l{SCEME.EG.1).DR.(ISCEME.EQ.3)) GOTO 24
{F(l{SCEME.EG.1).DR.(ISCEME.EQ.3)) GOTO 24
IF(ISCEME.EW.4) 150TO 24
IF(ISCEME.EW.4) 150TO 24
INWEABS(AANWLK) + O.S
INWEABS(AANWLK) + O.S
IF(A\&NALK.LT.J.) IRW=-IRW
IF(A\&NALK.LT.J.) IRW=-IRW
KKE:STM + dRw
KKE:STM + dRw
\&F(KK.LT.!) GOTO 28

```
            &F(KK.LT.!) GOTO 28
```

If（KK．GT．IMNGC）KRAIHNGE
OO 己己 Rel．dKange
6

$$
A(J, K)=A \subseteq(J, K R)
$$

$A Q(J, K)=A D(J, K K)$ KKEKK＋ 1
2e
cuntinue
GOPO 28
24
CONTINUE
IRWERANWLK
DELTABHANWLK－FLOAP（IRW）
KEISTM＋INW
1F（！SCEHE．EQ．1）K\＆K－1
IF（1SCETE．EU．4）K＝K－ 1『F（DEL．TA．LT．刀．）KミK－ 1 JF（DELTA．LT．0．）UELTAEI．DELTA 1F（K．LT．l）KE』

L－1
1F（ISCEME．EO．3）GUTO 808刀
IF（IPCEME．EQ．4）GOTO S090
LF（K．GT，IRNG）KEIRNG
OEGTA己z JELTA\＆OEGTA
UELTA3EOEGTAC＊UELTA
$c$
AMIs－OELTA＊（OELTAC－3．＊OELTA＋2．）／b． AGECDELTAB－2，MUEGTAE－DELTA＋E．）／E． Als－IELTA＊（OELTAE－UELTA－2．J／Z． A己sOELTA＊（OELTAC－1．）／16．
continue
C

Aट:AI $(\mathrm{J}, \mathrm{K}+3)$


```
A1($,6)=VALUE&
AG(J,L) 3VALUEG
                        L=L + !
                        KaK + 1
                IF(bolE.jrange) GOTO 26
```

                    c
    3080
e ft interp
c
c
8001
c
valug gathrazali(J,K) - DELTA*AI(J,K+\&)
valuedentltaza@ (J, K) DELPA*AQ(J, K+\&)
-I(J.K)sVALUE!
AO(J.K) BVALUE S
b-L
к=x +
I+(t.LE.INANGE) GOTO UD88
c
6070 28
8090
6

```
MMIBOELTAE(OELTA - 1.)/2.
AB&. UELTA*EE
OF POOR QUALITY
    AIsOCLCPA*SOLL:A + 1.d/2.
C
80
C
                STARTESTART - HMOV
    C
    30
    C
    c
    WRIPE OUT THE REWULTS TO * a
    35 CONTINUE
C
DO 4O KES,IRANGE
90 WR&TE(a) (A[(J,K),J={,NMCOEF),(AQ(J,K),J={,NMCOEF)
C
100 CONTINUE
C
G
WRITE PHE FILE GACK TO + 2 IN THE FORM IT GANTS
C
        ISKIPIRANGE - I
                NMEZ +NMCOEF
                GEWIND 2
C
        DO 200 NES,IHANGE
        REWIND 4
        1F(K.EO.8) GOPO 110
                        df=k - &
                00 105 6a1,8%
                    READ(,a)
805
$80 CONTINUE
C
C
    DO &S0 Is&,maxSus
c
                GEAO(4) (PF(J),JE2,NM)
                WFITE(E) (PF(J),JEI,NM)
            IF(I.EQ,MAXSUE) GOTO $50
                OO 14D J=1.15K&P
            846 READ(4)
C
80 continue
C
20O CONTINUE
c
SOOOS CONISNUE
C
    HETUHN
    ENO
```



```
    OPB,FOM
C&*********************SUBROUPINE トINE (F8,FB)
C**********************
C
        VIRPUAL FI(1),F(W(1)
        COMMON /COEFG/AI(1024)
        ctmMMON/INSTR/A(i)
        CUYMCN /PEMP/TEMP(1)
        COMMON /WAM/HAM(256)
のnのロのnのno
    CUMPUTE TAE FINE HESOLUTION FHT
        LRANSESA(18)
        RaTiO:A(ce)
        NMCOEF:A(25)
        NH2=NMCOEP#Z
        iFINEBA(27)
        maxSua=A(29)
        mINSHE=!
            jpauea(28)
            HMOV=AaliOnA(T)/(A(2b):A(25))
C
                    STARTS:&.
            IF(A(C4).tQ.己.) STAKTSa(T)/(4.*A(20)) + 1.
        WH!PE(6,7#9) STARTS,A(24)
```



```
C
        get phe window wegghis
            15T-1
        CalL wT(WAM,IFINE,IST)
        WRITE(h,Sa0&) IR\triangleNGE,NMCOEF,IFINE,IPAD,
        l
            MINSUB,MAXSUE
SuOg Furmat!/| FINE IHANGE *IGI
    1 &STFFP ".860' OH SUGS IN 2NO *.86,
```



```
C
            WHITE(6,5AU&) RATIO,RMOV
SUO& FUFMATGO HATIU OF SF/LFMF OIPEIS.g,
            & INCHEMENT OF SUB -.&PE\S.BN%
C
            FMAX=0.
            1az:0
            INNG=0
C
```

```
c
C
C
C
TRANSFER OATA FOR IHE K PH HANGE TO .DAT SlOT * 4
    00 4 def.maxSua
        meau(C) (Al(L),LEI,NME)
        wmjTE(4) (AI(b),Gal,NME)
    CONIINUE
OO PME COEFFICIENPS 1 TO MALF * I
                                    IAREO
    CALG POSHLF(FI,NMUV,STARTS,IAR,COPMAX,ICOF)
do ime negative cotfficIENTS
    CALL NEGHLF(FI,MMOV,STARTS,IAR,COFMAX,ICOF)
6
G WAITE IT OUT PO THE LO
    NHITE(5.1EO(4) ICOF,COFMAX,R
    wH!TE(holgO(C) ICOF,COFMAX,K
```



```
1000
C
        IF(COFmAX.LY.HMAX) GOPO COO
            mMAX=COFMAX
            jAl:jcof
            IRNGER
        CONTINUE
C
    TEMP(1)EIAR
    TEMP(2)=IKNG
    PEMN(3)=IAZ
    PEMP(4)=RMAX
nnnnonoo00nn
TEMP(I)=1920.
TEMH(2)=2.
TEMP(S)E35.
                    TEMP(4)=2.20944556t5
gefunN
```



```
    OPYA.FOR
C**********************
SUOROUTINE POSHLP (FI,RMOV,STA:ITS,IAR,COFMAX,ICUF)
C**********************
C
        VIRTUAL FI(I)
        CUMMON COEFS/ Al(128),AU(I2A%,WOHKI(256),WONKG(256)
        GOMMON /INSPH/A(I)
no
                NMCOEFEA(2.)
            IFINE:A(27)
            max$u5ma(29)
            M!NSUAM!
            IPavea(28)
            NUMDE:IHAU=IFIX(A(2b))/NMCOEF
        W#ITE(6,450) NUMUR,IPAO,NMCOEF,A(26)
        FUHMAT(S(&b,Ix),\PE{5,0)
    c
C
    DO PHE COEFEICIENPS & TO HOGN & :
        ICCanMGOEP/E * I
                RUNSUYESPARTS
                SSTAHTESTAMPS
            IENIIISTANT * IFINE - I
C
C
c
    0050 [=1,ICE
    IF(ISTARP.GT,MAXSUE) GOTO 60
    REWINO 4
    IGEISTART - I
    \F(IG.LT.&)GOTO 10
            OO 5 Jai,IG
        acau(a)
    CONTINUE
80
C
            IEBIENO
        IF(IE.GTOMAXSUQ) JEamAXSUB
C
            JJ=0
            OO 15 JSISTART,IE
C
            MEAD(4) (AS(L),L=&,NMCUEF),(AD(L)OLO&ONMCOEF)
                    JJEJJ - &
                    wowkI(JJ)aAI(I)
            woexG(JJ)=AG(I)
C
```

```
ib
小又=3J * 1
00 1% Jejx,lfin%
                    wORKI(J):G.
                WORRO(J)=0.
            CONPINUE
17
C
        CINTINUE
    PUT ESMMING WEIGGPING ACHOSS SUBARKAYS
        NULL PME AEST OF AMAAY
        COMPUPE PHE PINE STAUCTURE FFT ZNO
        Pino phe magnituoE OP tME PFT ALONG largest, eget
            CALL PAOFFT(FI,GOPMAX,ICOF,IFAD,IFINE,IAR,NUM&A)
                C
            munSUegrunslue * rmov
            ISTAHTANUNSUS. 0.5
            IEND:ISTAKT IFINE - I
nnognugnn
    conigNuE
        CONTINUE
            GETURN
            ENO
```



# ORIGINAL PAGE is <br> OF POOR QUALITY 

```
つな
        0O SS degy,IENO
        REAO(4) (A&(6),G81,NMCOEF),(AU(6),LE{,NMCDEF)
        JJEJJ * 1
        wonml(JJ)=A!(b)
        WORNO(JJ)EAG(8)
        conTINUE
C
    CALL PAUFPP(FI,COFPAX,ICOF,IPAD,IFINE,\AR,NUMORJ
C
60
C
        RUNSIJEAGUNSUG * MMOV
            xXeAES(RUNSUG) * O.S
        IF(RUNSJH,LP,0.) XXE=XX
        ISTAHIEXX
    IENDOISTART - IFINE - &
100 CONTINUE
C
REPURN
ENO
```


# ORIGINAL PACE IS OF PODR QUALITY 

PUT PHE WEIGHTING ACROSS PME WINOOW
005 JE\&.j」
nORKI(J)anam(J)*WORKI(J)
WORKO(J)SHAM(J)WWONKG(J)
CONPINUE
NuLl out the rest of pme arkay
1F(BPAO.LE.JJ) GOPO 23
」ロシJ」 * 8
0020 JEJX,IPAO
WORKI(J)=0.
wORKC(J) al.
continue
compute tme fine structure
CALG FFTGIHAD, NOHKI, WOKKGS
Compute tme magnitude of the squared
PEOROER THE GRGAY
Since malf fhe flne coefricitats ane pu the leff and
PME OTHER HALF TO PME HIGHT FOR EACH GROSS CUEFFICIENT
OVEN SAMPLJNG by faciuk or 2 to 1 imRow ouf half the coeff


MICROCOPY RESOLUTION TEST CHART NATIONAL BUREAU OF STANDARDS STANDARD REFERENCE MATERIAL 1010a (ANSI and ISO TEST CHART No. 2)

## ORIGINAL PAGEE IS OF POOR QUAEE IS




```
のnのnのrncos
SUHNOUPINE OUPMO(A1,AR,A3,A4)
V84PUAL A1(1),A2(1),As(1),A4(1)
            cummon IINSTR/AA(I)
            COMMUN /TEMP/TEMP(1)
C
gRINP oUf phe vata sCale to pme max passeu dn amguments
            NHANGE=AA(80)
            NPGLSEPEMP(1)
            {A-PEMH(2)
            dAaPEMD(s)
            XMAXITEMP(4)
C
            NNIPE(A,10GG) IN,IA,XMAX,NPULSE
            WNIPE(S,IWNAS IN,DA,MMAX,NHULSE
IU0O FORMAT(//' NANGE CELL ',IO/" ANGLE .,IG/" VAGHE ',IPEIS.O,
            l NO. PutSE P.10/1/
C
C
mEUNGANIZE PAE FILE I'S PME AZIMUITM VERSUS NANGE
```



```
            abso fuf en peak in the miUOLE OP THE MILE
                    CALG HEURO(NPULSE,&A,VRANGE,A1,A2,A3)
C
G find fme sum in fme malnluge ano tme sidloge for
    mANÖC ANU AZIMUPM UIREGTIONS
```



```
6
FiNO PME LIMITS OF TME MABNLOUE ALONG PNE A&BMUTN SLICE
                    CALL SLICE(IR,JA,NPULSE,IGF,IPT,XMAM,AIS
C
C
        return
    ENO
```

```
        VIRTUAL Al(B)
    *inu pme limifS of ple masingobe ano Tme masnlobe
    SUM ALONG THE ANGLE SLIGE
```

        WEWINO 3
        KOIR - 1
        IF(K.LY.1) GOPO 2
            001 jei,k
            HEAD(3)
        CONTINUE
        WEAO(3) (A!(8),IE!,NPULSE:
    UO RIGAP side off the pegax
        RGela
        KHaIA - I
        SUMNED.
    CONTINUE
    IF (KG.GT,NPULSE) RGEI
    IP (NR,GT,NPULSES RHAI
    fH(A1(KE).GE, AI(KL)) GOTO O
            SUMA \(\operatorname{SUMA}\) - AI(RL)
        MLEKL - 1
        K月日KN - \&
        GOPO 3
        conisnue
            gntakl
            SUMNESUMR - AI (KL)
    6
6
5
5
(INI) PME LEFT SIOE POINT
KROIA
KLEBA - 1
suma ${ }^{\circ} 0$ 。

```
    cunitnue
    IP(K#.LT.:) NGENPULSE
    IP(KLGGT.I) KGONPULSE
    (P(A)(KL).\sigmaC.al((RR)) GOTO
        SUMGESIMML AISKR)
        AROKN - I
        MGERL - I
        GOTO S
        CONTINUE
        ILFOKR
        SUMLOSUML - AI(RR)
        SUMAZESUML - SUMR - AI(IA)
6
201 PUPGPOTL * XX
C
C
    FINO PME S&OELOBE
    TOPLOS.
    OO EAI IEI,NPULSE
    ymaci(s)
        TOPSLED.
            PEARSLON.
        SPEAKEN
C
        1F(ILF.GY.$HT) GOTO 209
        3P(BLF.GP.2) GOPO 209
        N0ILF-1
        DO dde lad,k
        YOPSL:POT3L AlPIj
        IP(AICl).GF.PEAKSL; GOPO 200
            PEinSLaA&i!)
                IPEARE\
C
20
C
209 PONYSNUE
C
C
        NEINT * &
            kKENPULSE
        IP(IGF.GT.IRT) RKMILF - I
            OQ 210 INK,KK
                TORSLETURSL. Al(s)
            8*(A|(b).LT.PEAN9L) GOPO 210
                    peAKSLeAl(If
                    IPEAREI
G
210 CONTINUE
の\piのo
```

```
RESEPOTL - SURIAZ
DHABO.
        1F(SUMAZ.GP.O.) DBAE&G.GALUGIG(NES/SUMAE)
            0d|AE|.
        IF(XNAX.GT,O.) OUPAEIO.WALOGIO(HESIXMAX)
C
    l
218
C
C
0
C
    l
2e%
    !
C
n=n
    WAITE(6,CIS) ILF,IA,IHT,SUML,SUMR,SUMAZ,FOTL,
RES,OHA,DGPA
WHITE{5,2&&) \LF,IA,I&T,SUML,SUMM,SUMAZ,TOTLO
                                RES,OHA,DEPA
```



```
    1. MIU PUINT ',$5/' HMT MOINT ',I51' SUM LEFT *IPE&5.8/
    & SUM RIMI .IPEIS.G/. MAINLOSE SUN OIPEIS.,8/
    * TOTAL SUM O&PEIS.6/O SIUELOBE SUM P.IPE:S.8/
    - SIDFLDAE PO MAINLOAE (UBJ, IIPEIS.O/
```



```
        DEA0.
        IF(SUMAZ,GY.E.) DBEIO.*ALOG&O(TOHSL/SUMAZ)
            U日SL#0.
        IF(YMAM.GT.0.) OGSL:IO.*ALOG:O(PEAKSL/XMAX)
        WHITE(G,ZZG) SUMAZ,TOTSL,PEAKSL,IPEAK,XMAX,IA,
        OO,OHSL
        WHITE(S,E<OJ SUMAZ,IUISL,PEAKSL,IOGAK,YMAX,IA,
        OU,DNSL
        GORMAT(1/O ANGLE SUM OIPEIS.8," MABN SUM OIPEIS.B/
        * PEAR SlOE LOEt *IPEIS.d." LOCAFION*,&G/
        - maximum valuEE .,IPEI5.8," location .igel
        - inpeghateo (0,o) oipi;is.0/
        * pol/pear (or) oipeis.8/ij
        RETURN
        ENO
```

```
*のnのo
    OSCNEO.FDN
C*********************
C
    SugROUTINE REOMA(NHULSE,IA,NHANGE,A1,AR,A3)
C
        VINPUAL AI(1),A2(20,200),AS(20,200)
    reunganize fme file in azimuta vemius mangs
        matmet iman nange azimuin
    PUP fME PEAR IN PME mIDOLE OF PNE FILE IN AZIMUTN OIRECTION
                REWINO O
C
                    IMSOANPULSE/E
                            ISTAMTEIA - JMSO
                IF(ISTAMT.LT.J) ISPARTEISTANT * NHULSE
C
                    NPass=00%
                    NPASE:NPASS/E
                    NPASEzanPASE
C
                                    NPANPULSE/NPASS
                    NOVENONPULSE - NPASA*NP
                    IP(NP.BT.J) GOTO 850
C
            O0 8а% Ja&,NP
            WEWINO 3
                            ISOESTART
c
C
            00 8&O I=I,NRANGE
                    AEAO(3) (AI(R),K=1,NPULSE)
                                    ISREIS
C
C
            00 40S NEB,NPASE
                    IF(&SR,GT.NPULSE) ISRAI
                                    A己(I,K)EAI(ISN)
                                    ISREISR - I
C
80s
            CONTINUE
C
    IP(NPAS2z.LT.1) SOTO 810
C
C
    00 000 KEI.NPASCZ
                                &F(ISN.GT.NPULSE) &SN:%
                                    Ag(d,H)aAl(ISR)
                                    ISNEIyN!!
C
```

ORIGINAL PACE: :
OF POOR QUAMIT


```
                                    ORIGINAL PACE IS
                                    OF POOR QUALITY
```

```
000n
```

000n
C
C
C*********************
C*********************
SU\#ROUPINE VALUE\&NPULSE,INANGE,IA,IGF,IAT,NRANGE,A\&,XMAXI
SU\#ROUPINE VALUE\&NPULSE,INANGE,IA,IGF,IAT,NRANGE,A\&,XMAXI
C
C
C**********************
C**********************
C
C
VIPPUAL A\&(1)
VIPPUAL A\&(1)
C
C
PGWINO
PGWINO
C
C
IM1OENPULSE/E
IM1OENPULSE/E
IDELTAE\&A ILF
IDELTAE\&A ILF
IP(IDELTA.LT.O) IOSLTAEIDELTA NPULSE
IP(IDELTA.LT.O) IOSLTAEIDELTA NPULSE
LIMLEIM\&O - 10ELIA - \&
LIMLEIM\&O - 10ELIA - \&
IOELPABIRT - IA
IOELPABIRT - IA
IF(IJELPA.GP.O) IOGLPAEIDELPA * NPULSE
IF(IJELPA.GP.O) IOGLPAEIDELPA * NPULSE
L\MNEIMIO \&OEGTA * I
L\MNEIMIO \&OEGTA * I
C
C
IPNGOLIMN* 200
IPNGOLIMN* 200
IPRLEGIML - 20O

```
                IPRLEGIML - 20O
```




```
        IP(IPNN,GT,NPULSE) IPIRONPULSE
```

        IP(IPNN,GT,NPULSE) IPIRONPULSE
    C
C
POTALED.
POTALED.
C
C
TOPSa0.
TOPSa0.
TOTL**
TOTL**
YSIOEOO.
YSIOEOO.
IVSREG
IVSREG
IVSASO
IVSASO
C
C
DO 120 JE\&,NPULSE
DO 120 JE\&,NPULSE
REAU(4) (4\(%),I\&I,NMANGE)
REAU(4) (4\(%),I\&I,NMANGE)
C
C
C TAG M MAINLORE WITM IT - \&
C TAG M MAINLORE WITM IT - \&
|TEO
|TEO
IF((J.GE., IML6),ANOU.
IF((J.GE., IML6),ANOU.
(J.bE.LJMMJ) IFES
(J.bE.LJMMJ) IFES
C

```
C
```

```
            viadme.
        00 1:8 \=1,NMANGE
C
        POPabsal(1) POTAL
C
112
C
SUMT80.
6
        SUMLLED.
        1P(&P.E|,O) GOTO 1/0
        Iv:IHANGE
            SUMLOE.
        IVLEIV
        IUN:IV - !
    IIJ CONTINUE
C
    IF(IVA.GT.N4ANGE) GOTO IIA
    IF(AI(IVA),GT.A&(IVL)) GOPO $84
    SUMGESUML A/(IVL)
    IVREIVA I
    IVGEIVG I
    gorg ils
C
ila continue
C
C
    IF(IVR,GT,NRANGE) SUMLESUML - AI (NHANGE)
        IVWROIVL
        SUMREA.
        IVGIV - I
        Iva=Iv
1/5 CONTINUE
    8F(IVL.bT.1) OOPO 180
    IF(&1(IVL).GY.A&GIVR)) GOPO 116
C
            SUMGESUMM *I(IVR)
            IVNAIVH - !
            IVLEIVL-I
        6010 125
C
116
    continue
C
C
C
C
C
6
                    vSIDEEA&(1)
                |VSRE!
                ivsAas
80
    SUMR=SUMR * AI(IVH)
    IP(IVL.LY.I) Sumigesuma - 4i(b)
        IVLLミIVN
    SUMTOSUML * SUMR - Al(IVJ
        IF(IVLL.LE.A) GOTO IS
            Ive!IVGL-1
            OD10 8=1,IVL
                SUMLLESUMLL * Al(i)
                &F(A)(d).GT,VSIDE) GOTO $0
```

ORIGINAL PAGE IS OF POOR QUALITY

```
C
ly continue
C
C
    IF(IVNH.(BE,NNANGE) GOPO 20
    IVAIIVAH * I
    OU $1 ISIVH,NKANGE
                SUMLLESUMLL * Al(1)
                If(AI(I).tP.VSIOE) GOPO 87
                    vsive=As(8)
                    IVSNEI
                    IVSAEJ
C
8%
20 LIJNIINUE
                            G0:0 1:%
C
$80 CONTINUE
c 00 30 IEI,NRANGE
C
            LlosumLl - Al(s)
            IF(AI(s).LP.VSgOt) GUTO 30
                    vosoegal(f)
                        :USN=:
                    IVSAOJ
6
30 rJNiINUE
I:9 CONTINUE
6
            YOPSaPOTS * SUMT
            TOPGE%OPL SUMLL
C
C IF(GJ,LT.IPNLS.OR.(J.GY.IFRR)) GOPO 92
c OO 89 IsI,NRANGE
                    Va6-*999.
                    xx+al(i)
                    YX=XX/XMAX
            IF(xx.fif.0.) VAL=EID.*ALOGIO(XX)
            Al(b)OVAL
8.
C
    CONPINUE
FWRIPE(3.808&) J,(A1(8).IEI,NNANGE)
806{ FUMMF(16,10(18,F7,2))
C
9E CONTINUE
C
C
6
leg continuse
```

ORIGINAL PAGE IG

## OF POOR QUALITY

```
C
                uy-2.
        &F(PGPL,GP.0.) DuASH,*ALOGIO(PIJTL/IOPS)
            NOPsJ.
                IF(YOTL.GP.O.) UOP:IO,*AGOGIE(YOPL/XMAX)
            TOPSGETOYS * TOFL
        WHITE(S,BOBG) TOTAL,TOTSL,TOPS,TOTL,OB,OEP
        WHIPE(O,BQOUS POTAL,PUTSL,TOPS,IOIL,UQ,OBP
        FURMAPT/IO TUPAL IN NANGE ANO ANGLE ,IPEIS,OI
        - tural in mange anu angle sug o,dpeig.ol
        - toral uf sum .ifneis.0%
        - toial of siot .,IPEis.8/
        - siot/sum (0hi PiliPEIS.or
            - slue/mear (00) ',ipesis.0/1%
            vSDusvSIDE/XMAX
            IF(VSUH.GY.0.) VSOOEIO.*ALOGIO(VSDS)
        WR&TE(5.8U8E) IVSR,IVSA,VSIOE,VSOO
        WRIILG(b,GGOZ) IVSR,IVSA,VSIDE,VSOO
C
0082
Fummat(%/" peak sloe ludt bevel '/
    -gocatiun in hange P.I4," location in az& 0,Io/
        - MAGNEIUNE ',INEIS.8,' ( OY ) 'IPE&5.8/IS
        RETURN
        ENO
```


## ORIGINAL PAQE IS

 OF POOR QUALITY```
C*
C*
C
C*
C*
    8UBNOUPINE FFP(NUMBH,XT,YT)
    OLMEVSION XP(I),VT(I)
NUMER IS PHE NO, OF POSNTS OF LOOD OAY'A
IF NOT A PONER OF IWO PAD TO NEXT MIGHEST
WIPM 2EROS
    NCPOWEI
        IIGMDBNUMBR
        ITEMPEITEMO/Z
    IF(ITEMP.LE.|) GUTO &0
        NCPOWENEPOW - 1
        GOPO:
        continue
c
    N=2**N2POW
    IP(N,GE.NUMBR) GOTO 13
    N=N*2
    NEPOWENZPOW * 1
    CUNIINUE
    gM(NUMGR.GE.N) GOTO 20
        Niznumga - I
    DO 15 LMANI,N
        XT(LM)日®.
        YT(LM)=0.
is cunismue
C
20 CONTINUE
C
    MzNEPCW
    OO DELE L0.8,M
    LMXEC**(N-LO)
    LIX=2*LMX
```



```
    OU 6iOU LMES,LMX
    AHGS(LMOI)*SCL
    Cacus(ANG)
    SESIN(AHGJ
    OO GO0 LI=LIX,N,WIX
```



```
    JCeJI+LMX
    T1=at(Jl)=xp(je)
    Te=YT(J&)Ory(Jद)
    xT(J))ExT(JI)*xT(J己)
    YT(J&)*YP(J&)+Y!(sa)
    xT(J?)=G#il+S-fz
    Y!(J2)=C*P2-S*「!
    gunTINUE
    NVEEN/E
    NMI=N-I
    J!8
```

c*:
of BuR Qunhty

```
    DU mSS d=1,NNI
1+(1.fig..J) 00 90 031
1!日\(J)
T&Ev(J)
x(J)0ar(I)
vi(J)0ri(i)
xT(1)-il
YT(1)-1it
6 }
020
xinve
020 CuNigIJE
IF(K.Gi`.J) 60 P0 035
J.J.K
kek/2
GO 10 0<0
Og5 J.jok
NUMUREN
gE TUNN
eno
```



```
C***********************************
c
C************************************
C
    qEAl manay(1)
C
        IF(IOPP.EG.0) GOTO 600
            MNanumBR - l
        IF(IOHP.EG.ट) GOPO 200
            PI=a.*ATAN(I,J
            Pl2:E.0P!
            PHASE:PLC/XN
            IF(IOPT.EO.4) GOTO 100
            IF(IOPI.EG.b) GOFO 300
            IP(IOHP.EO.6) GOTO 400
            IP(IOPIVEO.I) GOTO 500
                    6
            C1:0.5
            C己=n.5
    IP(IOPT.EO.3) GOTO $0
c
            C1:0.54
            C200.40
                    &0 continue
C
    OU 20 IEI,NUMER
            XNE& - 1
```

```
            W0WCI - C&*COS(PAASE-EN)
            AHAAY(B):WD
C
20 CUNIINUE
GOPO 10000
C
Ive CONIINUS
c pmasedornaseve.
C OO 1208:1,NUMBR
C
            XN:& - !
```



```
            AnHar(8)=m0
C
ide Cunisnue
C GOTO IHEOO
    NUMGR &S PME NGSGER OF ELEMENTS IN THE WINDOW STARTING
    figm the fias: ADONESS OF AmRAY
gut CuNT:NUE
C
                    IIPPNUMER/2
C
C
            wuafloap(z+(1 - |))/XN
            amhay(t):wo
Civ Sunfinue
: INPaIUP 1 I
C OO z20 Iat'JP,NUMAN
C wusz. PLOAP(z-(I - 1))/XN
            anwar(t)DND
&&O CONTENUE
C
            G0%O 10u00
3u0 cuniINUE
C
2S DO IAMLOR WESGNIINE
            STARPO-P!
C
C
    OO 310 801,NUMEH
```




```
C
lio Cunilnue
        6070 10804
```

```
avs cuvTIN|t*
C
G SS OB PAYGOR WEIGHTING
    STAHTGOP!
    OU 410 dEI,NUMBN
%
    WUSN.898650.CUS(START) - 0.0857830.COS(2.*STANFj * 
                0.AM21*1w4.cos(3.0STAMP)
            anmar(s)at. - 2mwu
            STANTESIANT ( PMASE
C
AIO CONTINUE
C
            goto 14000
gos CONTINUE
C
    3S OB PAYLON WEIGMPING
        STAHPE-P!
        UU SIO IEI,NUMER
C
```




```
            ANPAYIIIEI. - EmwD
            giariegtahy P Phase
c
sio cunPinut
C
                    Garo 10000
bios continue
C
C UNIFUAM WEIGMTING
- 00 6i0 inionuMBR
c anrav(i)as.
C
Os0 CONTINUE
$0000 CONTINUE
C
                    RETURN
                    ENO
```


[^0]:    * The notation is somewhat different from references (6) and (7).

