IN - 39809

p-34

JPL PUBLICATION 86-33

(NASA-CR-179913) ERS-1 SAR DATA PROCESSING (Jet Propulsion Lab.) 34 p CSCL 05B

N87-12965

Unclas
G3/43 44622

ERS-1 SAR Data Processing

Kon Leung
Thomas Bicknell
Kenneth Vines

September 1, 1986

NASA
National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

JPL PUBLICATION 86-33

ERS-1 SAR Data Processing

Kon Leung
Thomas Bicknell
Kenneth Vines

September 1, 1986

NASA

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

ABSTRACT

To take full advantage of the synthetic aperture radar (SAR) to be flown on board the European Space Agency's Remote Sensing Satellite ERS-1 (1989) and the Canadian Radarsat (1990), the Jet Propulsion Laboratory (JPL) is being directed by the National Aeronautics and Space Administration (NASA) to study the implementation of a receiving station in Alaska to gather and process SAR data pertaining in particular to regions within the station's range of reception. The current SAR data processing requirement is estimated to be on the order of 5 minutes per day. JPL's Interim Digital SAR Processor (IDP) which has been under continual development through Seasat (1978) and SIR-B (1984) can process slightly more than 2 minutes of ERS-1 data per day. On the other hand, the Advanced Digital SAR Processor (ADSP), currently under development at JPL primarily for the Shuttle Imaging Radar C (SIR-C, 1988) and the Venus Radar Mapper (VRM, 1988), is capable of processing ERS-1 SAR data at a real time rate. To better suit the anticipated ERS-1 SAR data processing requirement, both a modified IDP and an ADSP derivative are being examined. For the modified IDP, a pipelined architecture is proposed for the mini-computer plus array processor arrangement to improve throughput. For the ADSP derivative, a simplified version of the ADSP is proposed to enhance ease of implementation and maintainability while maintaining near real time throughput rates. These processing systems are discussed and evaluated here.

CONTENTS

1. INTRODUCTION 1

2. ERS-1 SAR 2

2.1 ERS-1 Orbit 2

2.2 ERS-1 SAR Characteristics 2

2.3 Alaska Facility SAR Data Processing Requirements . . 3

2.4 Processing Algorithm 3

3. SOFTWARE BASED PROCESSORS 3

3.1 Interim Digital SAR Processor (IDP) 5

3.2 Pipelined IDP 6

4. HARDWARE BASED PROCESSOR 18

4.1 System Summary 19

4.2 Algorithm and Data Flow 20

4.3 Throughput Evaluation 24

5. THROUGHPUT AND COST TRADE-OFF 25

6. CONCLUSION 26

ACKNOWLEDGEMENT 26

REFERENCES 26

<u>Figures</u>

1. Processing Algorithm 4

2. Interim Digital Processor Hardware Block Diagram . . 5

3. Processing Module Partitioning 6

4. Pipeline Processing Architecture 7

5. Three-Page Scheme 8

6. Data Flow Diagram 9

PRECEDING PAGE BLANK NOT FILMED

Figures (cont)

7. Relative Hardware Cost vs. Throughput Capability . . . 15

8. ERS-1 Software Based Pipeline Processor Simulation Program Structure 18

9. Advanced Digital SAR Processor Block Diagram 19

10. Block Diagram of Modified ADSP for ERS-1 20

11. Modified ADSP Processor Functional Block Diagram . . 21

12. Algorithm Flow Diagram 22

13. Throughput vs. Cost 26

Tables

I. Execution Times Per Function On The ERS-1 Software Based Benchmark Processor. (FPS-5205 Array Processors) . . 11

II. Pipeline Implementations 16

1. INTRODUCTION

The European Space Agency (ESA) is preparing the launch of its first in a series of remote sensing satellites (ERS-1) in 1989. Among other remote sensing instruments, on board will be a C-Band synthetic aperture radar (SAR). With no on-board data storage capability planned, five SAR data receiving stations are selected to span the northern hemisphere including station sites at Kiruna (Sweden), Fucino (Italy), Maspalomas (W. Africa), Churchill (Canada), and Shoe Cove (Canada).

The coverage of these five stations is mainly North America, Europe, North Africa, and a portion of the Arctic region around Greenland. Clearly, the installation of a receiving station at Alaska will allow the coverage of the whole state of Alaska and the surrounding oceans as well. This will provide an excellent opportunity for researchers to investigate the dynamic behavior of polar ice and oceans in support of overall Earth resource studies. Currently, the National Aeronautics and Space Administration (NASA) has instructed the Jet Propulsion Laboratory (JPL) to pursue the installation of a ground receiving station and an appropriate data processor and archival facility at the University of Alaska (UAL) in Fairbanks. The goal is to receive about five minutes of SAR data per day when the ERS-1 satellite is in view of the receiving station for the full duration of the planned three-year lifetime of the satellite. In addition, it is hoped that some data can also be acquired from the Radarsat satellite which is being planned by the Canadian government for a 1990 launch.

To handle this quantity of data with no daily backlog will require a processor that has throughput rate capability in the neighborhood of 1/230th real time rate and better. Currently, the fastest digital SAR processor in existence at JPL is the Interim Digital SAR Processor (IDP). It is a software based processor, with a mini-computer plus array processors set up, that is capable of running at about 1/500th real time rate with ERS-1 type data. However, with the implementation of a pipelined processing architecture, such a mini-computer plus array processors set up is anticipated to achieve throughput rates up to 1/105th real time rate depending on the level of parallelism of the array processors arrangement. In the mean time, under development currently at JPL primarily for the Shuttle Imaging Radar-C (SIR-C, 1988) and the Venus Radar Mapper (VRM, 1988) is a hardware based processor named Advanced Digital SAR Processor (ADSP) which has the ability to handle ERS-1 type data in close to real time rate. The complexity of the ADSP system compounded with the limited availability of knowledgeable ADSP service personnel poses a problem for the operations of an ADSP unit at UAL. However, there are ways to improve upon the maintainability and reliabilty of the ADSP by sacrificing some throughput speed. This prompts the proposal for a simpler version of the ADSP that can better fit the processing requirement of the Alaska facility.

This publication starts with a description of the ERS-1 SAR and the data processing requirements of the Alaska facility. The applicability of the processing algorithm currently implemented in the IDP and planned for the ADSP is examined. It is then followed by detailed descriptions of the software based pipelined version of the IDP as well as the hardware based ADSP derivative. Special attention is paid to the derivation of the throughput figures for each machine and the trade-off between throughput rates versus costs. Finally, the applicability of these processors to other missions as well as future processor development trends are discussed.

2. ERS-1 SAR

The synthetic aperture radar (SAR) aboard the ERS-1 satellite consists of a 10m X 1m antenna and is designed to operate in C-Band (5.3 GHz). It is capable of two modes of operation, the wave-mode and the image-mode. The wave-mode is designed to produce small SAR images (typically 5 km X 5 km) and is intended to provide estimates of the power spectrum corresponding to the imaged areas. In contrast, the image-mode is designed to yield high resolution images covering a wide area (typical coverage of 100 km X 80 km per frame at ~30 meter resolution). The image-mode data is the type of data planned to be acquired at the Alaska facility.

2.1 ERS-1 Orbit

The ERS-1 orbit is described in detail in the ESA Ground Station Interface Specification document (Ref. 1). The key parameters are listed as follows:

1. Semi-Major Axis	7153.10 km
2. Mean Inclination	98.52 deg
3. Mean Eccentricity	1.165E-3
4. Mean Argument of Perigee	90.00 deg
5. Mean Nodal Period	6027.90 sec (14 1/3 orbits per day)
6. Mean Local Solar Time @ Descending Node	1030 hours +/- 1 minute

2.2 ERS-1 SAR Characteristics

The ERS-1 SAR has the following characteristics (Ref. 1).

Frequency:	5300 +/- 0.2 MHz
Bandwidth:	13.5 +/- 0.06 MHz
PRF range:	1640 - 1720 Hz in 2-Hz steps
Long pulse:	37.1 +/- 0.05 microsec
Compressed pulse length:	64 ns
Peak Power:	4.8 KW (at power amplifier out)
Antenna size:	10m X 1m
Polarization:	Linear Vertical
Incidence angle:	23 deg nominal
A/D complex sampling:	18.96 samples/sec
Sampling window length:	299 microsec
Quantization:	5I, 5Q

2.3 Alaska Facility SAR Data Processing Requirements

The SAR data processing requirements at the Alaska facility for ERS-1 are established to be:

Throughput -- Data Processed/Day	5 min
(Equivalent Throughput Rate <a>)	> 1/230th real time rate
Image	
Spacial Resolution (3-dB width)	
Ground Range	< 30 m
Azimuth (4-look)	< 26 m
Number of Looks	
Range	1
Azimuth	4
Peak Side-Lobe Ratio (PSLR)	<-21 dB
Integrated Side-Lobe Ratio (ISLR)	<-17 dB
Pixel Dynamic Range	> 72 dB
Pixel Spacing	
Ground Range	12.5 m
Azimuth	12.5 m
Frame Size	
Along Track	100 km
Across Track	100 km
Relative Geometric Accuracy	200 m
Operations Duration	36 months

<a>. Over 24 hr. day; including 25% processing overhead.

2.4 Processing Algorithm

The processing algorithm implemented on the IDP and planned for the ADSP is depicted in Figure 1. The algorithm utilizes the frequency domain fast correlation approach (Refs. 2 and 3). The data (range echoes) is first correlated in the range dimension with the range pulse replica. The range compressed data is then corner-turned to make them easily accessible in the azimuth dimension. Azimuth compression is then performed by correlating the azimuth data with azimuth reference functions having the appropriate Doppler characteristics. Range migration effects are compensated for in the azimuth frequency domain with range cell selection and interpolation. Correlation in both the range and the azimuth dimensions is performed efficiently with the help of Fast Fourier Transforms (FFTs). This algorithm has proven to provide high fidelity and efficiency through Seasat and SIR-B.

3. SOFTWARE BASED PROCESSORS

Software based SAR processors have undergone continual development at JPL for the past decade. The original Interim Digital SAR Processor (IDP) was completed in 1979 to digitally correlate Seasat (1978) SAR data (Ref. 4). The system consisted of a mini-computer (Gould SEL 32/55) and an array processor (Floating Point Systems AP-120B). Its throughput capability was in the neighborhood of 1/3000th real time rate. The system has since been

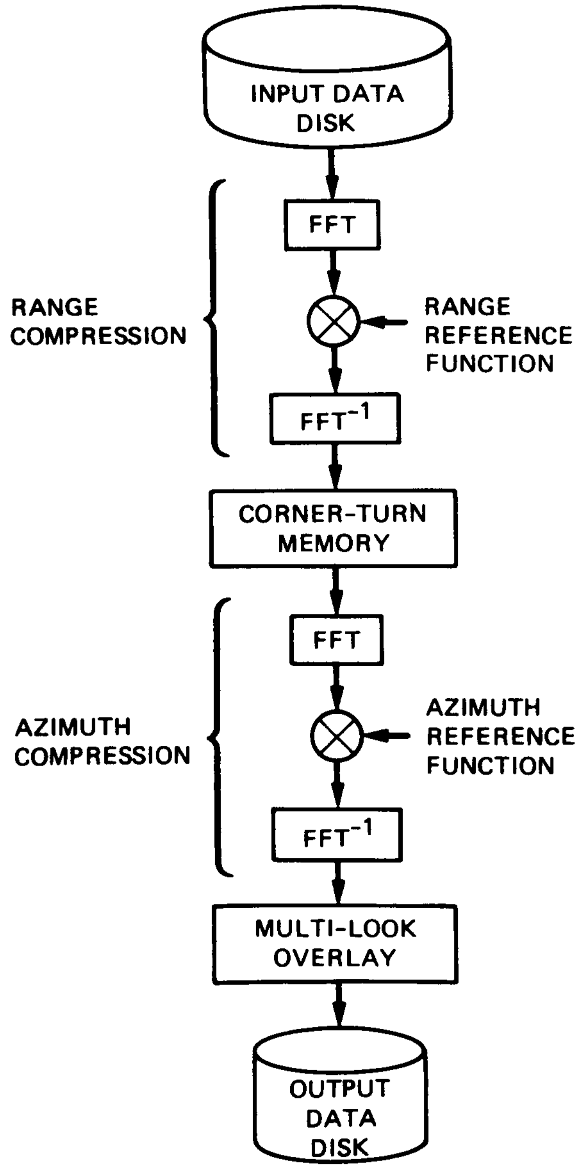


Figure 1. Processing Algorithm

upgraded to a Gould SEL 32/77 mini-computer with three Floating Point Systems (FPS) AP-120Bs in parallel. The throughput rate increased to 1/600th real time rate in 1980, and with more improved software and an additional FPS AP-120B in parallel, the throughput rate is currently at 1/500th real time rate. The IDP is still in active support of the Seasat (1978) and SIR-B (1984) data processing function to date.

3.1 Interim Digital SAR Processor (IDP)

The IDP hardware configuration is depicted in Figure 2. The array processor(s) takes on the computation intensive load which is principally the vector arithmetics associated with FFT correlation. The SAR processing algorithm is partitioned into three major processing modules (see Figure 3): range correlation, corner-turn, and azimuth correlation. The IDP executes each module sequentially, using disks to store intermediate results between modules. For the initial IDP system that utilizes a single array processor (AP), the throughput was bounded by array processing. That is, the AP processing times were much longer than input/output (I/O) times, thus making the IDP computation bound. The system is then augmented with multiple array processor units arranged in parallel, each performing the identical function in each of the processing modules to allow an increase in throughput (Ref. 5).

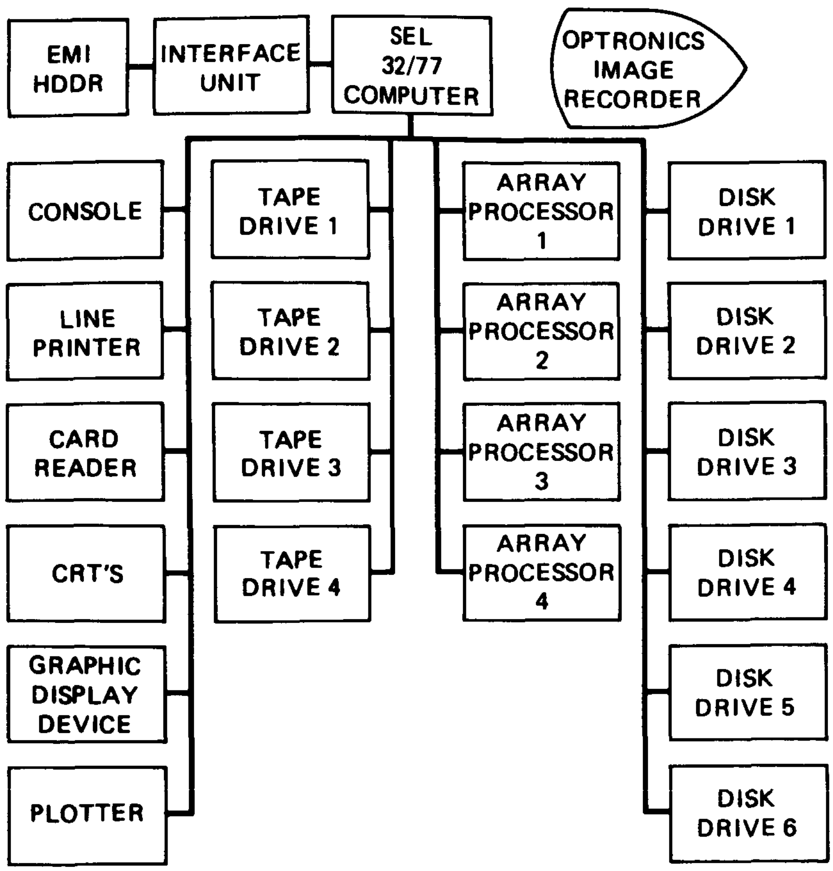


Figure 2. Interim Digital Processor Hardware Block Diagram

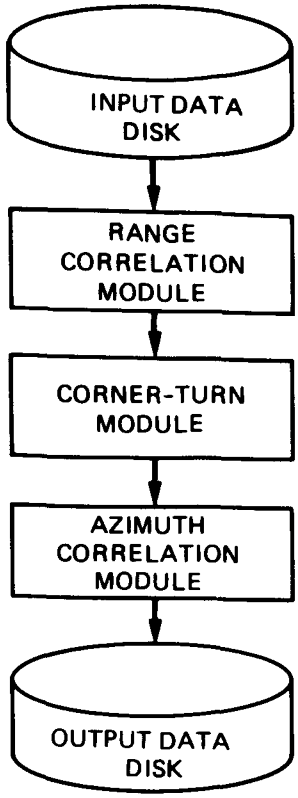


Figure 3. Processing Module Partitioning

3.2 Pipelined IDP

The throughput rate of the IDP in its state can no longer be improved upon appreciably because the AP processing time is closely matched to the I/O times. Its current configuration uses array processors in parallel in each of the program modules to maximize efficiency, but each program module is executed sequentially due to limited computer core memory (512 KBytes) and other hardware constraints. An alternative is to consider performing the three major SAR processing functions of range correlation, corner-turn, and azimuth correlation concurrently. Data is then essentially pipelined continuously through the system. While such a system demands more hardware to implement, the advent of relatively inexpensive memory modules and low AP costs certainly makes this a very cost effective means to increase throughput. An estimated fourfold increase in throughput is possible with such a pipelined arrangement.

3.2.1 Pipelining Architecture

The pipelined configuration is depicted in Figure 4. The array processors are arranged in sequential stages so that data is pipelined through each AP stage in sequence, accomplishing the range correlation, corner-turn and azimuth correlation in succession. Within each stage array processors are arranged in parallel, much like in the existing IDP configuration. The list of functions performed in each of the AP stages is listed below:

1. Stage AP1 - Range Compression

   a. Input Data Unpack
   b. Forward FFT
   c. Range Reference Multiply
   d. Inverse FFT
   e. Output Data Pack

2. Stage AP2 - Azimuth Forward FFT

   a. Input Data Unpack
   b. Forward FFT
   c. Output Data Pack

3. Stage AP3 - Range Migration Compensation, Azimuth Compression, and Multi-look Overlay

   a. Input Data Unpack
   b. Range Cell Interpolation
   c. Azimuth Reference Multiplies
   d. Inverse FFTs
   e. Magnitude Detect
   f. Multi-look Overlay
   g. Output Data Pack

Data is packed between the stages to facilitate I/O and to reduce memory requirement for intermediate data storage.

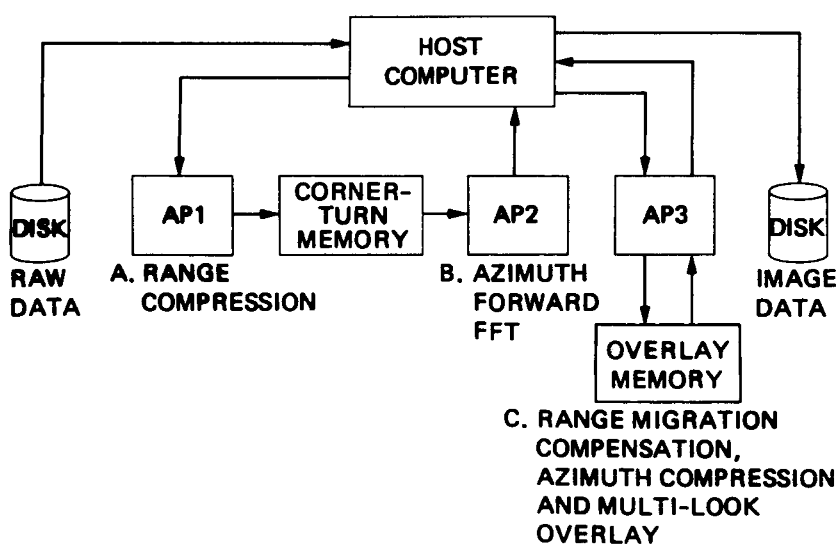


Figure 4. Pipeline Processing Architecture

As with the IDP processor, the ERS-1 software based processor requires range compressed lines to be transposed before azimuth processing can begin. The IDP utilizes a limited amount of memory plus extensive disk storage and I/O to accomplish this function in two sequential phases of processing (Refs. 8 and 9). The recent availability of low cost memory enables the transpose (corner-turn) operation to be performed more efficiently in memory alone. The ERS-1 software based processor uses a three-paged system to perform the corner-turn in memory (see Figure 5). The three-page scheme utilizes double buffering with three separate blocks (pages) of memory. As shown in Figure 5, the first 2K lines of range compressed data (16I, 16Q) are stored into two pages of the shared attached memory (SAM). At this point, enough lines are available (2K lines) to be read from blocks A and B, transposed, and input to the azimuth processor. While these two pages are read, the third page (page C) is written with the next 1K of range compressed lines. After the 1K range lines are written and 2K azimuth lines are read from SAM, new range compressed lines are then written into page A, while pages B and C are read transposed into the azimuth processor, and so on. This scheme provides the 1K samples overlap in azimuth that is necessary to allow continuous pixel output. This read-write process continues throughout processing by switching the page pointers as the buffers are filled and emptied. Care must be taken when implementing this algorithm to ensure that all page accesses are completed before the page pointers are switched.

3.2.2 Algorithm and Data Flow for ERS-1

A detailed data flow diagram in Figure 6 illustrates the data precision and memory requirements at various locations along the pipeline. A typical ERS-1 processing run will go through the following steps:

1. Raw data range lines (~28K lines for each 100 km frame) are transferred from high density digital tapes (HDDTs) through an input interface and the host computer onto disk units.

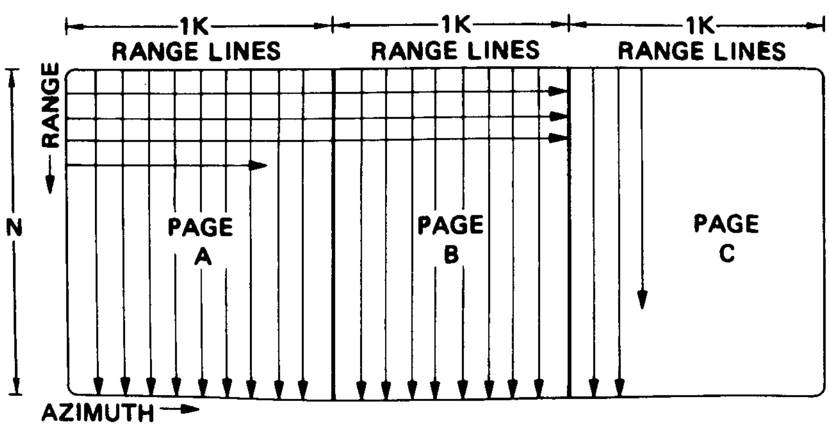
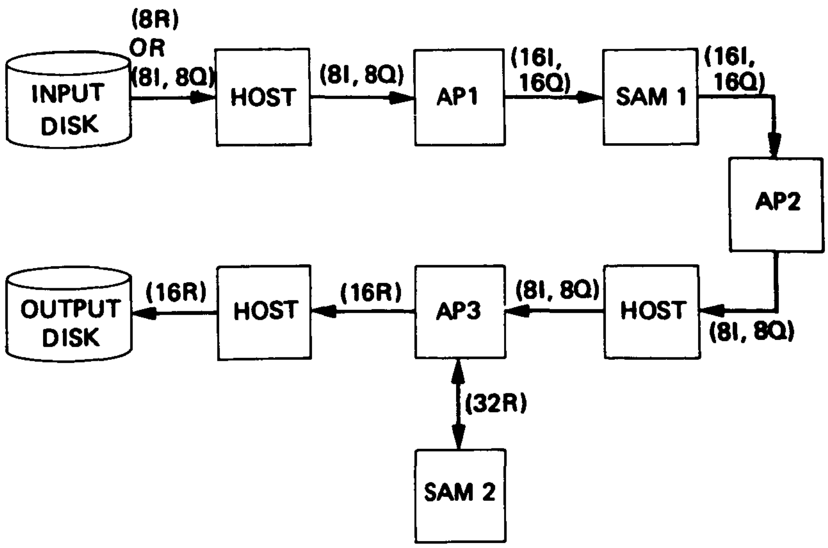


Figure 5. Three-Page Scheme



● MEMORY CAPACITY:

INPUT DISK	4K (8I, 8Q) x 28K RANGE LINE ≃ 224 MB		
OUTPUT DISK	256 x 24(16R) x 3400 AZ LINES ≃ 40 MB		
AP1	INPUT DOUBLE BUFFER	(2KW + 8KW) x 2	≃60 KW
	OUTPUT DOUBLE BUFFER	(8KW + 4KW) x 2	
	CFFT + RANGE REF	8KW + 8KW	
AP2	INPUT DOUBLE BUFFER	(2KW + 4KW) x 2	≃26 KW
	OUTPUT DOUBLE BUFFER	(4KW + 1KW) x 2	
	CFFT	4KW	
AP3	INPUT DOUBLE BUFFER	(1KW + 4KW) x 2	≃70 KW
	INTERPOLATION COEFF	32KW	
	INTERPOLATION STACK	16KW	
	$CFFT^{-1}$	1KW x 4	
	CVMAGS	1KW	
	SAMGET	4KW + 1KW	
	OUTPUT DOUBLE BUFFER	(256W + 128W) x 2	
SAM 1	3400 (16I, 16Q) x 3K		≃40 MB
SAM 2	OVERLAY	3400 x 1KW	≃22 MB
	AZIMUTH REF.	512 x 4KW	
HOST	128KW x 20		≃10 MB

Figure 6. Data Flow Diagram

2. Ephemeris and other pertinent engineering data are extracted either at the interface or by a host-resident program. Initial estimates of processing parameters are derived based on the decoded results.

3. Pre-processing for Doppler parameters (Doppler frequency, Fd, and Doppler frequency rate, Fr) is initiated as follows:

   (i) 5K raw data range lines are range compressed at reduced resolution resulting in 5K range-compressed data lines stored in the first corner-turn memory (SAM-1). Each range-compressed line contains 1K samples at 16I, 16Q.

(ii) Azimuth correlation is initiated using processing parameters determined in Step 2. Auto-focus and clutterlock techniques are used to derive refined Doppler frequency and Doppler frequency rate.

(iii) Step 3ii is repeated iteratively until certain Doppler parameter accuracies are met.

(iv) Proper azimuth reference functions are generated and stored in SAM-2.

4. Normal processing begins. Post processing operations (slant range to ground range conversion and output pixel spacing resampling) are performed in the host in conjunction with the image pixel corner-turn.

5. Final image data are stored in the output disk.

3.2.3 Throughput Estimates

We shall estimate the throughput capability of the pipelined processor by analyzing the processing times and I/O rates based on a set of benchmark hardware. An attempt is made to formulate the throughput estimate as a function of the level of parallelism achieved in each of the AP stages. An actual timing exercise is also performed using available limited hardware to validate the throughput estimate calculations.

3.2.3.1 Basic Benchmark Hardware. The basic benchmark hardware units are as follows:

a.	Host	Gould SEL 32/97 mini-computer with 16 MBytes of 32 bit memory.
b.	Array Processor	FPS 5205 (equivalent to the FPS AP-120B in terms of processor speed and data I/O rates) with 1 MWord of 38 bit memory.
c.	Memory Module	Texas Memory Systems SAM-400/600 shared attached memory.
d.	I/O Disks	CDC 9766, 300-MByte Storage Module Device.

3.2.3.2 Basic Processing Times. The basic processing time at each array processor stage (AP1, AP2, and AP3) is compiled based on the benchmark array processor, the FPS 5205 (see Table I). In summary, at stage AP1 where range compression takes place, the worst case time taken to process each 4K complex samples long (or 8K real samples long) range line is 77.07 msec. At stage AP2, a worst case time of 23.89 msec is required to complete eack 2K complex azimuth forward FFT. At stage AP3, a worst case time of 64.43 msec is taken to accomplish four 256 complex point azimuth compressions and 4-look pixel overlays.

Table I. Execution Times Per Function On The ERS-1 Software Based Benchmark Processor. (FPS-5205 Array Processors)

Function	Worst Case (ms)	
Range Compression (Full Swath)		
8K Unpack (8 to 32 Bits)	4.75	
4K Complex FFT/Scaling	30.84	
4K Complex Multiply	6.14	
4K Complex Inv. FFT	27.44	
6800 Pack (32 to 16 Bits)	6.80	
SUBTOTAL	75.97	
Overhead (Apex Call)	1.10	
TOTAL	77.07	
Range Compression (Partial Swath)		
4K Unpack (8 to 32 Bits)	2.38	
2K Complex FFT/Scaling	14.93	
2K Complex Multiply	3.07	
2K Inverse FFT	13.23	
2688 Pack (32 to 16 Bits)	2.69	
SUBTOTAL	36.30	
Overhead (Apex Call)	1.10	
TOTAL	37.40	
Azimuth Compression Phase 1		
4K Unpack (16 to 32 Bits)	3.77	
2K Complex FFT/Scaling	14.93	
4K Pack (32 to 8 Bits)	4.09	
SUBTOTAL	22.79	
Overhead (Apex Call)	1.10	
TOTAL	23.89	
Azimuth Compression Phase 2 and Overlay		
4K Unpack (8 to 32 Bits)	2.37	
4 * 4K SAMGET	2.00	* 4
4 * 4K Multiply	4.096	* 4
4 * 4K Add	4.096	* 4
4K SAMGET	2.00	
2K Complex Multiply	3.07	
4 * 512 Complex Inv. FFT	2.86	* 4
1K Complex Magnitude Squared	0.85	
1K SAMGET	0.50	
1K Add	1.02	
256 Square Root	0.47	
256 Pack (32 to 16 Bits)	0.256	
256 Vector Clear	0.084	
1K SAMPUT	0.50	
SUBTOTAL	63.33	
Overhead (Apex Call)	1.10	
TOTAL	64.43	

3.2.3.3 Basic Data Transfer Times Among Devices. The basic I/O rates between various devices are listed below. These rates have been verified by actual timing exercises and therefore do include overheads.

I/O rates between :

a.	Host / disk	0.8 MByte/sec
b.	Host / AP	3.2 MByte/sec
c.	AP / SAM	8.0 MByte/sec

Using these I/O rates, the basic data transfer times for one "line" are compiled as follows:

1. Input Disk to Host
   8K (8 bit real) line @ 0.8 MBytes/sec — 10.24 msec

2. Host to AP1
   4K (8I,8Q) line @ 3.2 MBytes/sec
   + 1.1 msec overhead — 3.55 msec

3. AP1 to SAM-1
   400 (16I,16Q) line at 8.0 MBytes/sec
   + 0.02 msec overhead — 1.65 msec

Corner-turn in SAM-1

4. SAM-1 to AP2
   2K (16I,16Q) line @ 8.0 MBytes/sec
   + 0.02 msec overhead — 1.00 msec

5. AP2 to Host
   2K (8I,8Q) line @ 3.2 Mbytes/sec
   + 1.1 msec overhead — 2.33 msec

6. Host to AP3
   2K (8I,8Q) line @ 3.2 MBytes/sec
   + 1.1 msec overhead — 2.33 msec

7. AP3 to and from SAM-2 data transfer time included in processing time of AP3

8. AP3 to Host
   256 (16 bit) line @ 3.2 MBytes/sec
   + 1.1 msec overhead — 1.26 msec

Corner turn in Host

9. Host to Output Disk
   3400 (16R) line @ 0.8 MBytes/sec — 8.11 msec

3.2.3.4 Basic Pipelined System. The basic pipelined IDP system is depicted in Figure 6. It is defined to be the benchmark system (as discussed in Section 3.2.1) with only one FPS 5205 array processor in each AP stage. The system consists of the following:

Item	Model	Memory Size
1. Host Computer	Gould SEL 32/97	16 MByte
2. Input Disk	CDC 9766	300 MByte
3. Output Disk	CDC 9766	300 MByte
4. SAM-1	TMS SAM-600	40 MByte
5. SAM-2	TMS SAM-400	22 MByte
6. AP1	FPS 5205	1 MWord
7. AP2	FPS 5205	1 MWord
8. AP3	FPS 5205	1 MWord

It is evident from the basic processing times in Section 3.2.3.2 and the basic data transfer times in Section 3.2.3.3 that the pipeline speed is bounded by the computations in AP1 and AP3. Specifically, the processing time in AP1 of 77.07 msec per "line" is larger than the data transfer times of 10.24 msec, 3.55 msec, and 1.65 msec in Section 3.2.3.3 items 1, 2, and 3 respectively. Also, since it takes 78.92 sec (77.07 msec X 1K) to fill 1 of the 3 pages of memory in SAM-1 (see Figure 5), it means an azimuth line is available for azimuth compression every 23.22 msec (78.92 sec/3400). This transfer rate is much slower than those listed in Section 3.2.3.3 items 4 through 8. However, it is faster than the 23.89 msec and 64.43 msec processing times in AP2 and AP3 (see Section 3.2.3.2). Therefore, the bottleneck exists in AP3.

3.2.3.4.1 Total Correlation Time Estimate. Based on a typical image frame comprising of 28K raw data range lines (equivalent to ~15 sec of raw data covering roughly 100 km along track), the correlation time using the basic pipeline structure is estimated as follows:

1.	Time for initial fill of 2 pages of memory in SAM-1: (77.07 msec X 2K)	2.63 min
2.	Correlation time: (64.43 msec X 3400 X 27)	98.58 min
	Total Correlation Time	101.21 min

3.2.3.4.2 Pre-processing Time Estimate. Using the pre-processing procedures outlined in Section 3.2.2 item 3 and allowing, on the average, 4 iterations for Doppler parameters to be refined, the pre-processing time is estimated as follows:

a.	Range Compression: (77.07 msec X 5K)	6.58 min
b.	Azimuth Correlation: (64.43 msec X 1K X 4)	4.40 min
c.	Doppler parameter estimation and azimuth reference generation	1.10 min
	Total time based on 4 iterations =	a + 4(b + c)
	=	28.58 min

3.2.3.4.3 Total Processing Time and Throughput Rate. Counting a combined data transfer (from HDDT to disk) and ephemeris decode time of 15 minutes using Seasat and SIR-B operations experience, the total processing time is summed as follows:

1.	Data Transfer and Ephemeris Decode	15.00 min
2.	Pre-processing	28.58 min
3.	Correlation	101.21 min
	Total Processing Time per Frame	144.79 min
	Equivalent Throughput Rate	1/580th real time rate

3.2.3.5 Parallel Pipelined System. Re-examining the processing times in Section 3.2.3.2 and the data transfer times in Section 3.2.3.3, it is apparent that the pipeline can be sped up considerably if the processing times can be cut down to match the data transfer speeds. To achieve this, we can either select faster array processors or install parallel APs at each AP stage. We shall examine the fastest throughput achievable for the pipeline based on I/O rates alone.

3.2.3.5.1 Throughput Estimate. Suppose the processing times are no longer a factor, 1 page of the SAM-1 memory can be filled in 10.49 sec (10.24 msec X 1K). This means an azimuth line can be read from the 2 page buffer in SAM-1 every 3.09 msec (10.49 sec/3400). Since this line rate is slower than the data transfer times in Section 3.2.3.3 items 4 through 8, the pipeline becomes I/O bound at the input disk to the host bus. The correlation time in this case is therefore ~5.08 minutes (10.24 X 2K + 3.09 X 3400 X 27 msec). The pre-processing time is estimated as follows:

a. Range Compression:

(10.24 msec X 5K)	0.88 min

b. Azimuth Compression:

(3.09 msec X 1K X 4)	0.21 min

c. Fd, Fr estimation and azimuth reference generation — 1.10 min

Total based on 4 iterations =	a + 4(b+c)
=	6.12 min

So the total processing time becomes:

1.	Data Transfer Time per Frame	15.00 min
2.	Pre-processing	6.12 min
3.	Correlation	5.08 min
	Total Processing Time per Frame	26.20 min
	Equivalent Throughput Rate	1/105th real time rate

3.2.3.5.2 Hardware Requirement. To allow the pipeline to run at the input disk to host rate of 10.24 msec per range line, AP1 has to run at an effective speed ~7.5 times (77.07/10.24) faster than the baseline machine (an FPS 5205). At the same time, AP2 and AP3 also have to run at ~7.7 (23.89/3.09) and ~20.9 (64.43/3.09) times faster respectively.

3.2.3.6 Throughput vs. Cost Options. As illustrated in Section 3.2.3.5.2, the hardware cost of the pipelined system varies a great deal depending on the degree of parallelism in the AP stages along the pipeline (ie., the throughput capability). The normalized costs for the two systems described in Sections 3.2.3.4 and 3.2.3.5 are roughly 1.0 and 1.8 respectively. The software development effort for the pipelined processor is estimated to be around 25,000 lines of code (FORTRAN and ASSEMBLY combined) using prior Seasat and SIR-B processor development experience. So assuming an average of 10 lines per man-day and 240 man-days per man-year, the software effort is estimated to be roughly 10 man-years.

Table II contains a list of implementation alternatives between the basic and the fully paralleled systems. The cost of each alternative clearly depends on the throughput capability and they generally follow the curve graphically depicted in Figure 7.

3.2.3.7 Throughput Simulation. To simulate the ERS-1 SAR pipelined processing algorithm, a test program was written and implemented at the IDP facility at JPL. Since the equipment necessary to perform the simulation on the benchmark processor (see Section 3.2.3.4) was not available, the simulation was confined to a timing exercise. Actual simulation with image data will be performed as soon as the necessary hardware is in place and the simulation results will be reported at that time.

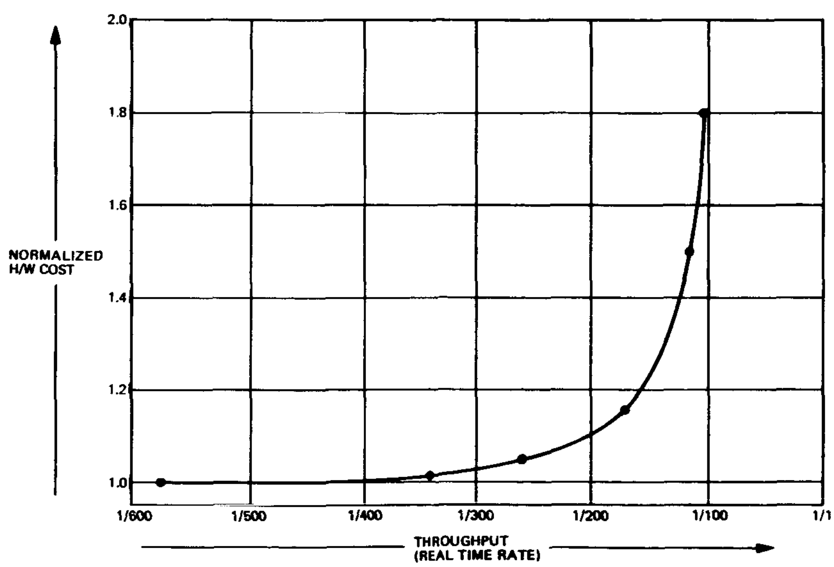


Figure 7. Relative Hardware Cost vs. Throughput Capability

Table II. Pipeline Implementations

AP3	Line Rate Post SAM-1	AP2	Line Rate Prior SAM-1	AP1	Correlation Time	Pre-Process Time	Total Process Time and Thruput Rate	Normalized H/W Cost
a	b msec	c	d msec	3	f min	g min	h min	
1	64.43	1	213.93	1	101.21	28.57	144.78 1/580	1.00
2	32.22	1	106.96	2	51.93	19.77	86.70 1/347	1.01
3	21.48	2	71.31	2	35.29	16.35	66.64 1/267	1.05
6	10.74	3	35.65	3	17.65	10.37	43.01 1/172	1.15
15	4.30	6	14.26	6	7.06	6.79	28.85 1/115	1.51
21	3.09	8	10.26	8	5.08	6.12	26.20 1/105	1.81

b = max 64.43/a, 3.09 msec

c = 23.89/1

d = max 3400"b/1024, 10.24 sec

e - 7.07/d

f = min 77.07,d •2048 + (3400)(27)b x 10^3/60 min

g = min 77.07,d •5120 + 4 4096 b + 66000 x 10^3/60 min

h = f + g + 15 min

i = 60 h/15

3.2.3.7.1 Simulation Environment. The present IDP facility makes use of a Gould SEL 32/77 mini-computer. It contains 512 KBytes of memory with a 26.67 MByte/sec, 32-bit internal data bus and is capable of executing 3 million instructions per second. Four Floating Point Systems AP-120B array processors are interfaced to the SEL 32/77 through one High Speed Data (HSD) 32-bit parallel interface. Each AP-120B is capable of performing 12 million floating point operations per second. These array processors each contain 64K words of 38 bit data memory. Three of the AP-120B array processors are independently interfaced with a Texas Memory Systems SAM-400 shared attached memory which has a 12-MByte 32-bit memory and an internal bus bandwidth of 16 MByte/sec. The Gould SEL 32/77 also has direct access to the SAM-400 through an independent HSD interface.

Although the ERS-1 benchmark processor described in Section 3.2.3.4 requires much more memory in the host computer and an additional SAM unit, the pipelined algorithm can still be simulated with the existing IDP hardware with the following qualifications:

1. Memory limitation in each device is circumvented by re-using memory buffer locations. However, double buffering is still maintained to minimize I/O overhead.

2. With only one SAM unit available at present, it is assigned to SAM-1 in the pipeline for this simulation. However, appropriate I/O times are included in the AP3 processing time, but no I/O to SAM-2 actually takes place.

3. The actual data corner-turning in SAM-1 utilizing the 3-page memory arrangement is not carried out due to memory contraint. Also, post-processing functions in the host are not simulated.

With the aforementioned limitations, SAR data was not used in the simulation exercise and no image was generated. However, as soon as sufficient hardware is available to form the basic ERS-1 benchmark processor, real SAR data (either Seasat or SIR-B data) will be used to validate the pipelined processor system.

3.2.3.7.2 Simulation Software. The pipeline simulation program consists of a control program and subtasks as illustrated in Figure 8. The control program regulates the execution of all the subtasks (T1, T2, and T3) by polling them in turn to determine which one is ready to be activated and also by keeping track of the number of times each subtask has executed. Each subtask involves a sequence of arithmetic operations (performed in the array processors) and I/O operations in the pipeline (see Table I). Direct memory access (DMA) commands are used to initiate I/O between the array processors, SAM, and disks. Furthermore, the correlation functions that are performed in the array processors are written in Array Processor Assembly Language (APAL) using parallel coding techniques (Ref. 6). These repeatedly used functions, which are mostly FPS supplied library routines, are then grouped into convenient host callable subroutines using an FPS supplied Vector Function Chainer to eliminate the overhead associated with multiple AP calls from the host.

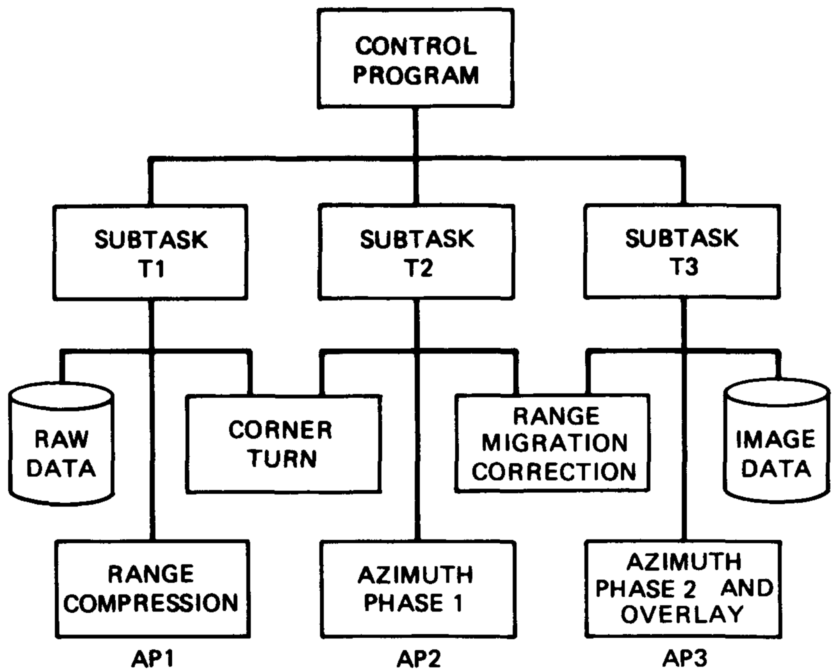


Figure 8. ERS-1 Software Based Pipeline Processor Simulation Program Structure

3.2.3.7.3 Simulation Results. The pipelined processing simulation as described in Section 3.2.3.7.1 and Section 3.2.3.7.2 was executed with the amount of random data equivalent to that of a typical ERS-1 100km X 80km frame (~27 blocks as discussed in Section 3.2.3.4). The execution time was determined to be 93.51 minutes. As expected, the simulation result is below the worst case estimate of 98.58 minutes obtained from analysis in Section 3.2.3.4.1. Moreover, the two results agree within 5.14%, thus validating the accuracy of the throughput analysis presented in Section 3.2.3.

4. HARDWARE BASED PROCESSOR

With the growing interest in near real time and eventual on-board SAR data processing, the hardware based processor is receiving a lot of attention as a viable means to achieve those goals. Under development currently at JPL is the Advanced Digital SAR Processor (ADSP) which is a hardware based processor capable of achieving real time data processing rate for ERS-1 type SAR data. In the ADSP (see Figure 9), all of the data processing functions are performed with high speed dedicated custom hardware. The processing functions themselves are arranged in a pipeline fashion with micro-processor control to maintain high efficiency. The ADSP system comprises 85 VLSI and MSI circuit boards in 27 designs. It is designed to be a development model and is not amenable to function in a field operations environment. To better suit the ERS-1 requirement, a modified version of the ADSP is proposed (see Figure 10). The modified version consists of a mini-computer as host, some commercial array processors to handle the lower rate processing functions and some custom high speed hardware (22 VLSI and MSI circuit boards in 8 designs) to take care of the high data rate functions.

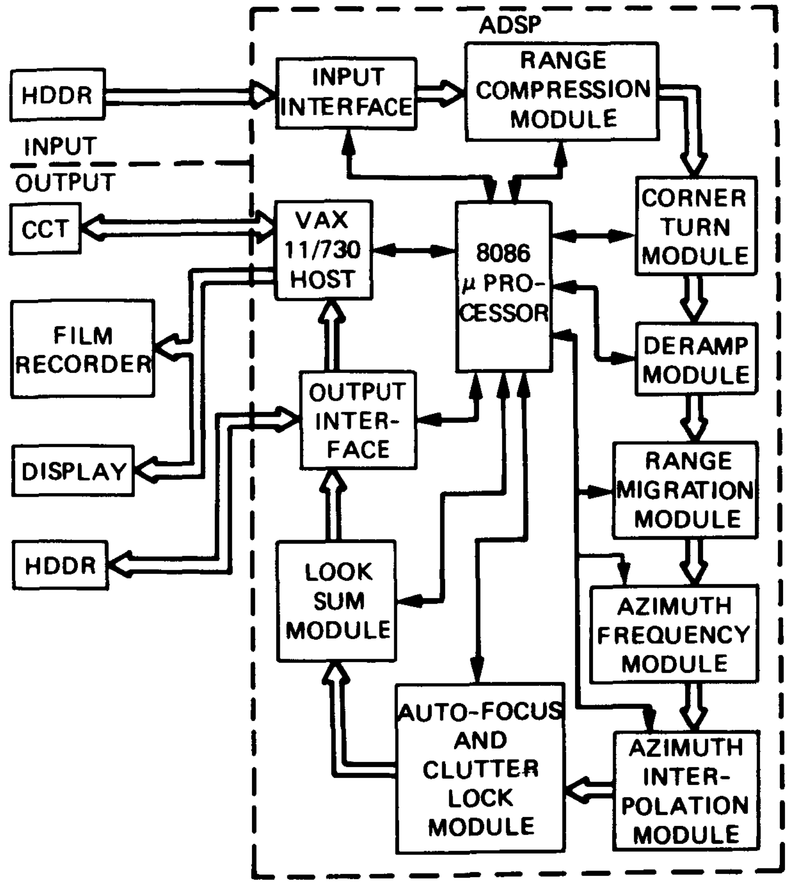


Figure 9. Advanced Digital SAR Processor Block Diagram

The proposed machine is capable of about 1/10th real time rate and is designed with self tests and diagnostic functions to suit a field operations environment.

4.1 System Summary

The processor is composed of a combination of commercial and custom digital processing, communications, and I/O equipment. Figure 11 shows the system diagram. The commercial equipment includes high density digital tape recorders (HDDR) for input and output, VAX control computer, and APTEC communications processor, disk and CCT drives, two array processors with a SAM (Shared Attached Memory), a laser beam film recorder, and an on-line image display. The custom hardware includes an input interface with buffer memory, an FFT with complex multiply, a corner-turn memory, an interpolator, and communications processors. An FFT convolution algorithm is implemented for both range and azimuth processing. The throughput rate is 1 mega-complex samples per second (input), or about 1/10th real time rate. The internal system clock is 10 MHz (complex sample rate). The required image output sampling rate and dynamic range will result in an output image data rate of about 1 MByte per second (for 16-bit output pixels).

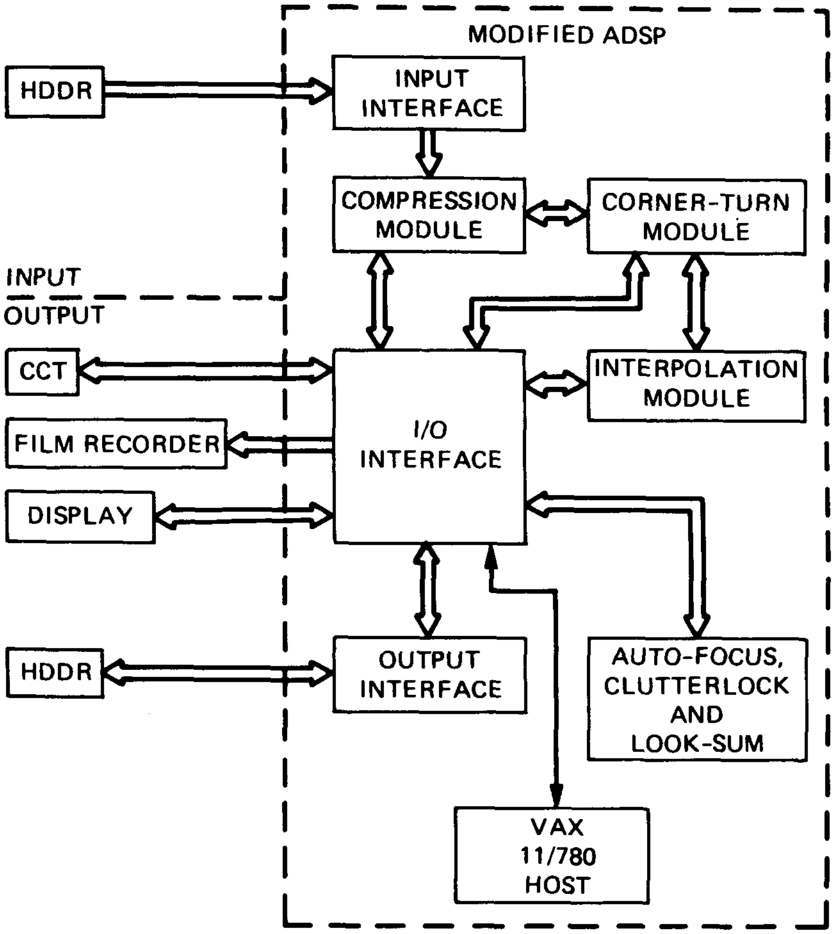


Figure 10. Block Diagram of Modified ADSP for ERS-1

The custom hardware design is based upon algorithms and techniques developed for the ADSP (Advanced Digital SAR Processor, see Ref. 7). At the reduced data rate (compared to the real time rate of ADSP) a significant reduction in the number of unique boards is possible, resulting in a much simpler, more maintainable system. Communications processors have been added to permit multiple passes of the same data through each module, reducing the total number of modules required to implement the algorithm. Significant improvements in fault tolerance, reliability, and testablity can be made with this approach, as opposed to a straight pipeline architecture (see Figures 9 and 10).

4.2 Algorithm and Data Flow

The algorithm is depicted in Figure 12. The numbers in the lower right corners of the boxes correspond to the hardware block numbers from the system diagram within which the functions are performed. The data block is formatted as is common in most variations of the FFT-convolution algorithm. A block is clocked through range processing (requiring two passes through the

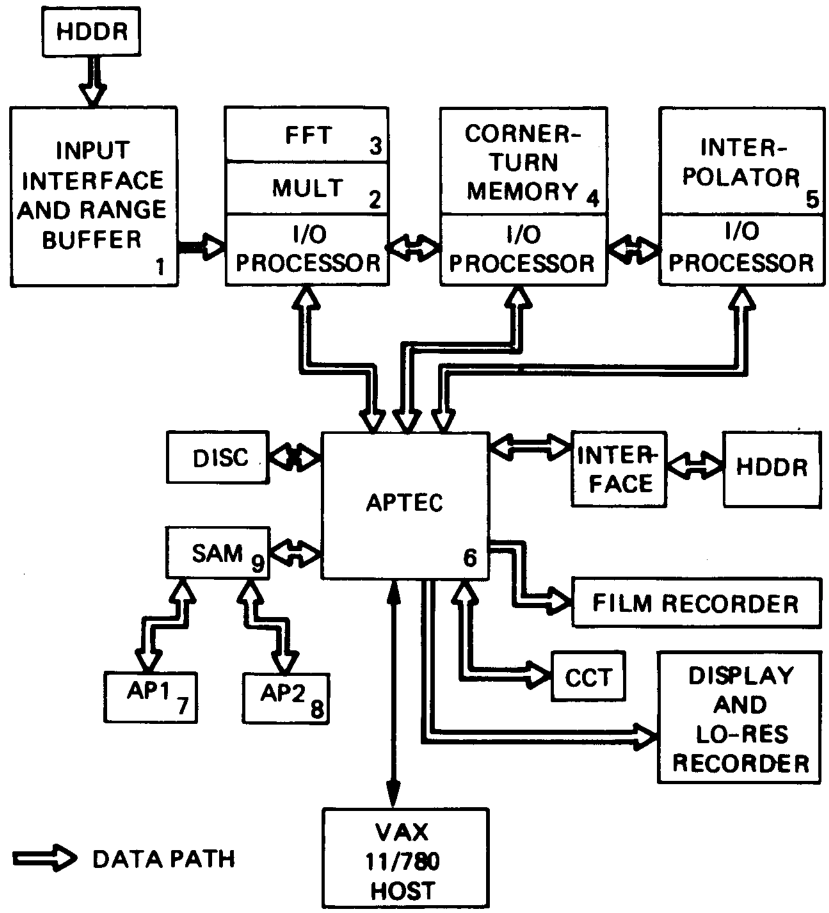


Figure 11. Modified ADSP Processor Functional Block Diagram

FFT module), and then azimuth processing begins. Of course, two of these range processing blocks are used together in azimuth processing. All the azimuth processing functions time-share the various modules. The details of the data flow are explained below.

4.2.1 Range Processing

Data is input from the HDDR as a serial bit stream and converted into parallel format by the input interface. The header data is extracted and sent to the control computer, and the SAR signal data is converted to an 8I, 8Q format and stored as range lines in the range buffer. The range buffer stores a full block of data; that is, the number of range lines is equal to one half the forward FFT length to be performed in azimuth. Actual memory size is 8K samples per line by 1152 lines, allowing up to 1K azimuth reference function length. The extra 128 lines is to allow continuous input while a block of data is being range processed.

When the range buffer is full, the data is sent to the first communications processor. The processor is essentially a staging buffer for

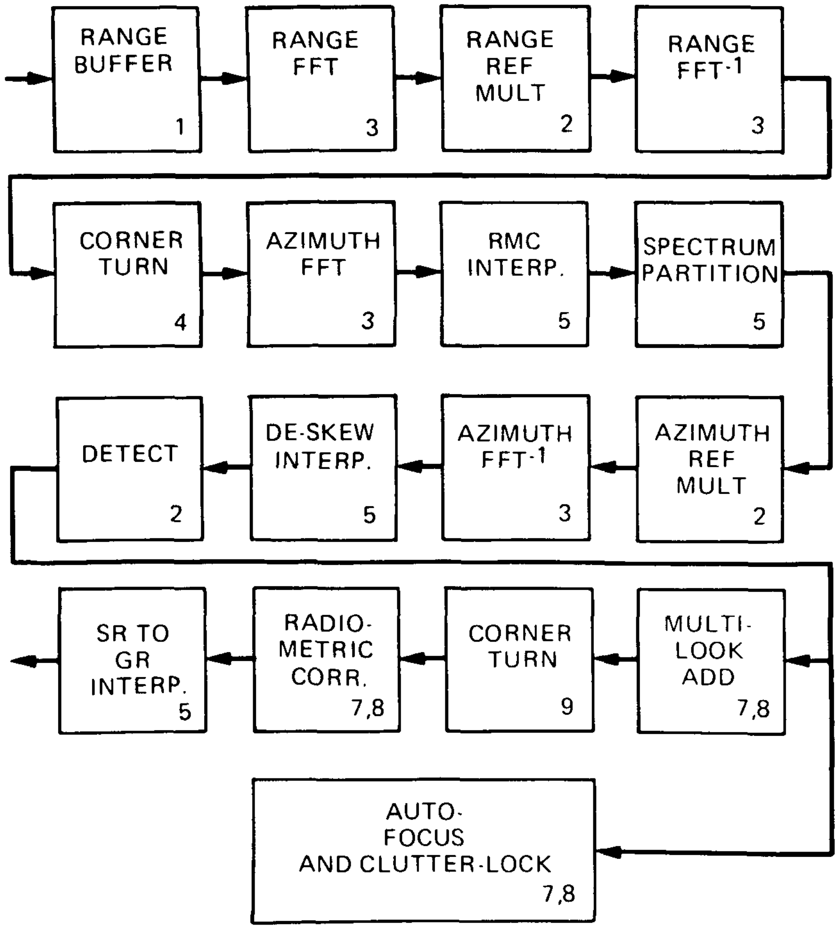


Figure 12. Algorithm Flow Diagram

the FFT (and complex multiplier). The forward FFT is performed on the data, which is then passed by the communications processor into the complex multiply. The output of the multiplier goes directly into the FFT where the inverse range FFT is performed. The data is then fully range processed before being sent to the corner-turn memory.

4.2.2 Corner-turn Memory

The corner-turn memory consists of two pages (range blocks) of memory, each having 8 megawords by 32 bits (complex - 16I, 16Q) for a total of 64 megabytes. Each 16 bit component of the complex word is composed of 10 bits of mantissa and 6 bits of exponent. The normal page format will be 1K range lines by 8K samples per line. When a page is filled (after range processing), both the new page and the previous page are read in azimuth order. This process generates the 50% overlap between azimuth blocks required for continuous processing with the FFT convolution.

4.2.3 Azimuth Processing

4.2.3.1 Forward FFT. The first operation on the data from the corner-turn memory is the forward azimuth FFT (the multiplier is bypassed by this data). The data will be processed through this module three times during azimuth processing for the following three operations (forward FFT, reference function multiply and inverse FFT, and detect). After the forward FFT is performed, the data is sent to the communications processor of the interpolator module for range migration correction.

4.2.3.2 Range Migration Correction. The interpolator module contains 128 lines of memory each 2K points long, allowing for range migration of up to 128 complex range pixels. A (range migration) path address vector is also input into the module and is updated each time the path changes. The address vector contains the path to the nearest eighth of a pixel. The integer portion selects the four points (in range) surrounding the desired location and the fractional bits select one of the eight sets of interpolation coefficients. The coefficients and corresponding data points are multiplied and added, thus performing a four point interpolation.

The interpolator module will also be performing azimuth deskew and slant range to ground range conversion. To minimize main-lobe broadening and ISLR degradation, the data should be interpolated to two times the Nyquist rate before detection. If the original sampling rate in range was 1.22 (time Nyquist), then the sampling rate must be increased by 65%.

4.2.3.3 Multi-look Spectral Division. The azimuth spectral line will be subdivided into (typically four) vectors for multi-look. The lowest point of the Doppler spectrum (start of the first look) is always selected first and written into an output buffer the length of the azimuth inverse FFT. The starting address within the buffer will correspond to the spectral line address (original frequency position) of the first spectral point (modulo the FFT length). Preserving the original frequency position will preserve the phase of the data for spectral applications requiring complex output. Since the spectrum will be less than the FFT length, there will be some zero data points added to the buffer. This process is continued for each look of a particular azimuth spectral line, and the completed azimuth lines are sent to the FFT module for reference function multiply and FFT. The data is now range migration corrected, spectrally separated into looks, and circularly shifted within each look to preserve phase.

4.2.3.4 Azimuth Reference Multiply and Inverse FFT. The data from the interpolator module is sent to the FFT module, which also contains the complex multiply. The azimuth reference function (generated by the array processor) is also sent to the module as the other input to the complex multiply. The reference (vector) memory is double buffered so that it can be updated "on the fly" as the reference function changes. The output of the multiplier is sent directly into the FFT for the azimuth inverse FFT.

4.2.3.5 Azimuth Deskew Interpolation. The output of the inverse FFT is sent back to the interpolator module for azimuth deskew interpolation and look alignment. The module contains four 8K-long vector memories in addition to the larger range migration memory. The vector memories are used for interpolation in the data direction (as opposed to the cross direction like range migration). Normally only half of an inverse FFT output will be interpolated for deskew, except when positive Doppler shifts occur between blocks and "extra" good data must be saved to fill in the gap. After interpolation, the data is sent back to the FFT module for detect. It is important to note that the multiplier is dual ported with a bypass so that the detect function can be performed simultaneously with the forward azimuth FFT.

4.2.3.6 Multi-look Overlay, Autofocus, and Clutter-lock. After detection the data will be in 16-bit floating point intensity, and will be sent to the two array processors with a Shared Attached Memory (SAM) system. One AP will work on the first half of the range data while the other AP will work on the second half. The multiple look image line is input into the array processor, 512 lines of four-look data are stored in each AP memory (for subsequent cross correlation with look one in a later block), all four looks are individually accumulated for clutter-lock, and the intra-line add function is performed. The corresponding line from the previous block is input to the processor from the SAM and the inter-line add is performed. After inter-line add the data is sent back to the SAM. When the block is completed, the portion of data that has been completed (multi-look) will be read out in range line order, radiometrically corrected, and sent to the interpolator for slant range to ground range interpolation. After this interpolation is complete, the data is merged with header information and sent to the display, output HDDT, and film recorder.

4.3 Throughput Evaluation

As in most data processing systems, the key to achieving high performance is the ability to handle both the I/O and computation rates. At the 1/10th real-time rate (about 1 MHz complex sampling rate input), it is not difficult to design computational modules such as FFTs or interpolators to process the data. In fact, a single FFT module and an interpolator module can process the four FFT and three interpolate operations, respectively, required by the algorithm. The I/O management required to keep the modules running efficiently is not so simple, but can be accomplished as will be described below.

A 13-stage pipelined FFT (sufficient for accommodating 8K complex FFTs), operating at a 10 MHz clock rate can perform all the required FFT functions in the algorithm. The forward and inverse range FFTs must each be performed at the average input data rate of 1 MHz. Together, they use up 20% of the FFT module capacity. The forward azimuth FFT is performed at an average rate of 2 MHz (due to the 50% overlap) and therefore uses up an additional 20% of the capacity. The inverse azimuth FFT is performed after range migration correction, during which the sampling rate in range can increase by as much as 65% (to an average data rate of 3.3 MHz), requiring 35% of the FFT capacity. The total usage of the FFT module comes to 75%, a very reasonable figure for customized hardware. The multiplier in the FFT module is only used as a reference function multiplier just before the inverse FFTs in range and in azimuth. It also performs the magnitude detect function after azimuth compression.

Approximately the same efficiency is required of an interpolation module operating at 10 MHz. The module has a continuous input and output rate of 10 MHz (complex), performing a real four point interpolation on the complex data. The range migration interpolation is performed at the highest azimuth pixel rate of 3.3 MHz. However, the interpolation needs only to be performed on the valid data out of the azimuth inverse FFTs. Although the input data rate is only about 1.65 MHz, the smaller output sample spacing desired (12.5m typically) causes an increase to the output data rate to as much as 2.8 MHz at the lower PRFs. The last interpolation is in the slant range to ground range conversion process which requires only 0.5 MHz since it is performed after multi-look overlay. The total usage of the interpolator comes out to about 83%.

All remaining functions are performed in the array processors with required computation rates as given below. The two array processors perform the identical operations in parallel with one processing the near-range half of the data and the other the far-range half of the data. The rates given are for each AP (ie., half of the total required): 0.25 MHz real adds for clutter-lock; 0.12 MHz real multiplies and 0.11 MHz real adds for auto-focus; 0.2 MHz adds and 0.3 MHz multiplies for reference function generation (table look-up is used for evaluating trigonometric functions); 0.2 MHz real adds and real multiplies are used in interpolation for range migration correction. The total comes out to about 1.4 MHz real operations or about 70% usage of the array processors (based on a typical real operations throughput of about 2 MHz for an array processor).

The input/output processors are required to have data available to the processing modules when they need it so as to prevent loss of efficiency due to I/O waits. The data busses are 32 bits wide (16I, 16Q) so that the data rates are essentially four times the word rate in terms of bytes. Both the FFT and the interpolator modules are required to handle data rates on the order of 100 MBytes per second when input and output of both data and reference functions are considered. Since the input and output busses are separate, the actual clock rate on on the busses is therefore only about 12.5 MHz.

5. THROUGHPUT AND COST TRADE-OFF

The throughput capability and cost trade-offs between the software pipelined processors and the hardware modified ADSP described earlier in Sections 3 and 4 are summarized in Figure 13. The IDP system is included for comparison. It is evident that as the throughput rate approaches about 1/200th real time rate or better, the software based processor cost rises sharply as a function of further increase in throughput. From a cost effectiveness standpoint, it is therefore more advantageous to consider hardware based processors to satisfy throughput requirements of 1/200th real time rate or faster.

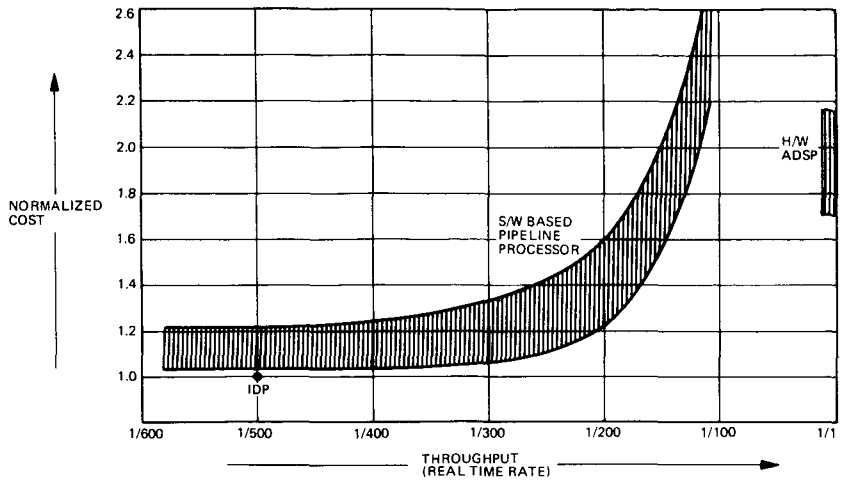


Figure 13. Throughput vs. Cost

6. CONCLUSION

In the previous sections, two types of processors are described for ERS-1 SAR data processing. The software based pipeline processor is flexible and upgradable. It is best suited for applications with processing throughput requirements from 1/500th to 1/200th real time rate. For applications that demand throughput rate higher than 1/200th real time rate, the hardware based processor is clearly the more cost-effective alternative. It is noted that both the software pipeline processor and the modified ADSP processor described in this paper are easily adaptable to handle almost any type of SAR data. The hardware based processor is also more readily adaptable to future on-board processing applications with the help of rapidly advancing integrated circuit technology.

ACKNOWLEDGEMENT

The authors wish to express appreciation to their colleagues and in particular, M. Jin and S. Pang.

REFERENCES

1. "ERS-1 Ground Station Interface Specification," ESA D/APP Earth Observation Department, Doc. No.ER-ESA-IS-GS-0001, Issue No.1, Rev. Letter:0, June 1985.

2. K. Leung, C. Wu, "Processing Techniques For Software Based SAR Processors," AIAA Computers In Aerospace Conference, Hartford, Connecticut, Paper No. AIAA-83-2381, pp.478-486, Oct. 24-26,1983.

3. C. Wu, "A Digital System To Produce Imagery From SAR," Proc. AIAA Systems Design Driven By Sensors, Paper No. 76-968, Oct. 1976.

4. C. Wu, et. al., "An Introduction To The Interim Digital SAR Processor And The Characteristics Of The Associated Seasat SAR Imagery," JPL Publication No. 81-26, April 1981.

5. B. Barkan, "Parallel Processing In A Host Plus Multiple Array Processor System For Radar," JPL Publication No. 83-54, Sept. 1983.

6. Technical Publication Staff, "FPS-5000 Control Processor Programmer's Reference Manual," FPS Publication No. 860-7437-045A/B, March/July 1984.

7. T. Bicknell, et. al., "ADSP Preliminary Design Review," JPL Internal Document, November 1982.

8. B. Barkan, C. Wu, "Transpose Of Externally Stored Matrices," Proceedings of the 1982 Array Conference, pp.179-185, March 1982.

9. B. Barkan, S. Pang, "Transpose Of Externally Stored Matrices-II," Proceedings of the 1983 Array Conference, pp.107-114, April 1983.