

N80-15537

NASA Contractor Report 159178

On-Board Multispectral Classification Study – Volume II: Supplementary Tasks

D. Ewalt

**International Business Machines Corporation
Gaithersburg, Maryland 20760**

**CONTRACT NAS1-15119
DECEMBER 1979**

NASA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

BIBLIOGRAPHIC DATA SHEET

1. Report No. 159178	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle On-board Multispectral Classification Study - Volume II: Supplementary Tasks		5. Report Date	
		6. Performing Organization Code	
7. Author(s) D. Ewalt		8. Performing Organization Report No.	
9. Performing Organization Name and Address International Business Machines Corporation 18100 Frederick Pike Gaithersburg, Maryland 20760		10. Work Unit No.	
		11. Contract or Grant No. NAS1-15119	
		13. Type of Report and Period Covered Contractor Report 1/15/79 - 10/31/79	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code	
15. Supplementary Notes Mr. Harry Benz was Technical Monitor for this study.			
16. Abstract This study is a continuation of the On-board Multispectral Classification Study Report No. NAS1-15119. Four tasks were defined to provide back-ground data and information pertinent to the Information Adaptive System (IAS). These four tasks include: <ul style="list-style-type: none"> o Sensing Characteristics for Future Space Applications o Information Adaptive System (IAS) Architectural Approaches o Data Set Selection Criteria o On-board Functional Requirements for Interfacing with Global Position Satellites (GPS). The results of these tasks will aid in establishing the overall IAS requirements necessary for processing remote sensor data on-board a spacecraft.			
17. Key Words (Selected by Author(s)) On-board processing Global Positioning System Data set selection		18. Distribution Statement Unclassified - Unlimited	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages	22. Price*

CONTENTS

Section		Page
1	INTRODUCTION	1
2	TASK 1 SENSING CHARACTERISTICS FOR FUTURE SPACE APPLICATIONS	7
2.1	Solid-State Self-Scanned Arrays	10
2.1.1	Sensor Data Rate	11
2.1.2	Radiometric Calibration	16
2.1.3	Data Format	22
2.1.4	Geometric Requirements	28
2.1.5	Charge-Coupled Imaging Device Chips	31
2.2	Conclusions	34
3	TASK 2 - INFORMATION ADAPTIVE SYSTEM (IAS) ARCHITECTURAL APPROACHES	36
3.1	IAS Architecture Characteristics	47
3.1.1	Interconnection or Bussing Structure	49
3.1.2	Memory Configuration and Type	68
3.1.3	Processing Element Architecture	71
3.1.4	Software Structure	76
3.1.5	System Design Procedure	81
3.2	Conclusions	89
4	TASK 3 DATA SET SELECTION CRITERIA	91
4.1	Sensor Data Preprocessing	96
4.1.1	Data Set Selection	100
4.1.2	Automatic Cloud Cover Detection Techniques	105
4.2	Conclusions	123
5	TASK 4 ON-BOARD FUNCTIONAL REQUIREMENTS FOR INTERFACING WITH GLOBAL POSITIONING SATELLITES (GPS)	124
5.1	SV Time and Data Control for User Acquisition	130
5.2	User Position Processing	136
5.3	Geometric Dilution of Precision (GDOP)	145
5.3.1	Ranging Accuracy	151
5.3.2	User Navigation Error	156
5.4	Relative Pixel Location	158
5.5	Satellite Selection	161
5.6	Tracking and Data Relay Satellite System (TDRSS)	162
5.7	Conclusions	164
	REFERENCES	165

ILLUSTRATIONS

Figure		Page
1	Present NASA Data Flow	1
2	Future NASA Data Flow	2
3	NEEDS Data Flow	4
4	Multispectral Scanner (MSS) and Its Field of View	8
5	Return Beam Vidicon (RBV) Camera and Its Field of View	8
6	Illustration of the Pushbroom Scan Technique	9
7	Line Array Dwell Time	13
8	On-Board Removal of Fixed Pattern Noise	19
9	Sequential/Parallel Data Readout of Line Array	25
10	Integrated Detector Array Chip	26
11	Bilinear Design (Detector and Edge Element Geometry)	27
12	Photodiode Detector Chip Showing Associated Electronics and Geometry	29
13	Solid-State Self-Scanning Array Spatial Separation	30
14	Pushbroom-Mode CCD Orbital Imager	33
15	Custom Designed Computer	37
16	Distributed Signal Processor	39
17	FFP Structural System	41
18	Processing Elements Grouped to Form a Cluster	44
19	Direct Dedicated Loop (DDL)	55

Figure		Page
20	Direct Dedicated Complete (DDC)	57
21	Direct Shared Memory (DSM) - Multiprocessor	58
22	Carnegie-Mellon C.mmp	60
23	Direct Shared Bus (DSB) - Global Bus	60
24	Indirect Centralized Shared (ICS) - Bus With Central Switch	62
25	Indirect Decentralized Shared (IDS) - Bus Window	63
26	Processing Element Performance Comparison	75
27	System Design Procedure	82
28	Partitioning Rationale	86
29	Horizontal Partitioning or Pipeline Processing	87
30	Vertical Partitioning or Parallel Processing	88
31	Multiprocessor Organization for the Maximum Likelihood Algorithm	90
32	Parallel/Pipeline Implementation of Maximum Likelihood Algorithm	90
33	NEEDS Data Flow	92
34	Information Adaptive System Data Flow	94
35	Sensor Data Preprocessing for IAS	97
36	Adaptive System Controller	99
37	Creating Data Flow For Resampling Grids On-Board	101
38	Data Set Selection	104
39	Homomorphic Filtering Method	114
40	Radiative Transfer Atmosphere Model	118

Figure		Page
41	Reflectance Test Method Using Radiative Transfer Model	121
42	Orbital Configuration	126
43	GPS Signal Structure for L1 Signal	130
44	Timing Program for the P Code Components X1, X2, and the Z - Count and HOW-Message Relationship	131
45	C/A Signal and Data Component Timing	132
46	Space Vehicle Timing	133
47	GPS Navigation Message Summary	135
48	Functional System Operation	137
49	Single Channel GPS Receiver Functional Diagram	138
50	Received Multiple Access Signal	140
51	Essential Element of a Delay-Lock Loop (DLL) is the Correlator	141
52	Coherent Delay Lock Loop	142
53	Carrier Recovery of the Received Signal	143
54	Earth Centered Coordinates	146
55	GDOP vs. The Cumulative Probability of a GDOP Less Than the Value Shown, (From Bogen, 1974)	152

TABLES

Table		Page
1	Data Readout Approach Comparisons	23
2	Architecture	50
3	Interconnection Structure Comparison	65
4	Distributed Processing System Comparison	67
5	Projected (1990) Mass Memory Technologies	71
6	Approximate 1σ Accuracy Orbit Determinations	128
7	Summary of GPS Signal Parameters	129
8	System Error Budget	156
9	User Navigation Error (UNE) (Meters)	158
10	Error Source Values for the Simulation	160
11	Maximum Error Over 90 Percent Grid	161

1.0 INTRODUCTION

Figure 1 illustrates the present NASA data flow for processing remote sensor data. The sensor and spacecraft (attitude and position) data are acquired at separate sites. Raw sensor data is either transmitted directly to a receiving site or stored in an on-board tape recorder for subsequent data transmission. Spacecraft tracking and attitude data are recorded at the various tracking sites and forwarded to a support computing center. The support computing center generates orbit and attitude profiles. The received sensor data is formatted, edited, calibrated, and radiometrically corrected. This sensor data is then combined with the orbit and attitude profile data to geometrically convert the sensor data from its sensor frame (line and sample) to a geocentric frame (latitude and longitude). Information extraction and data reduction algorithms are applied to the processed sensor data with each user then further extracting his specific information.

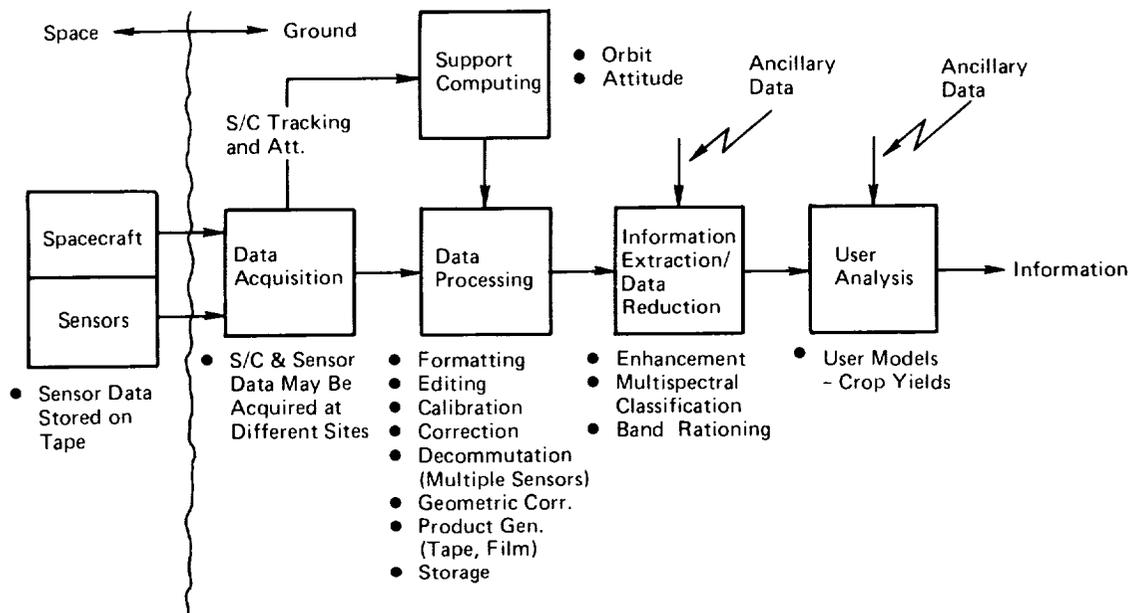


Figure 1. Present NASA Data Flow

Figure 2 illustrates the future NASA data flow for processing remote sensor data. The orbit and attitude profile data, previously computed at the ground, will now be computed on-board and combined with the preprocessed sensor data, also computed on-board. The sensor data will be converted to a geocentric frame, on-board, and the data packetized for transmission to the ground. A user can now process the data directly from the spacecraft.

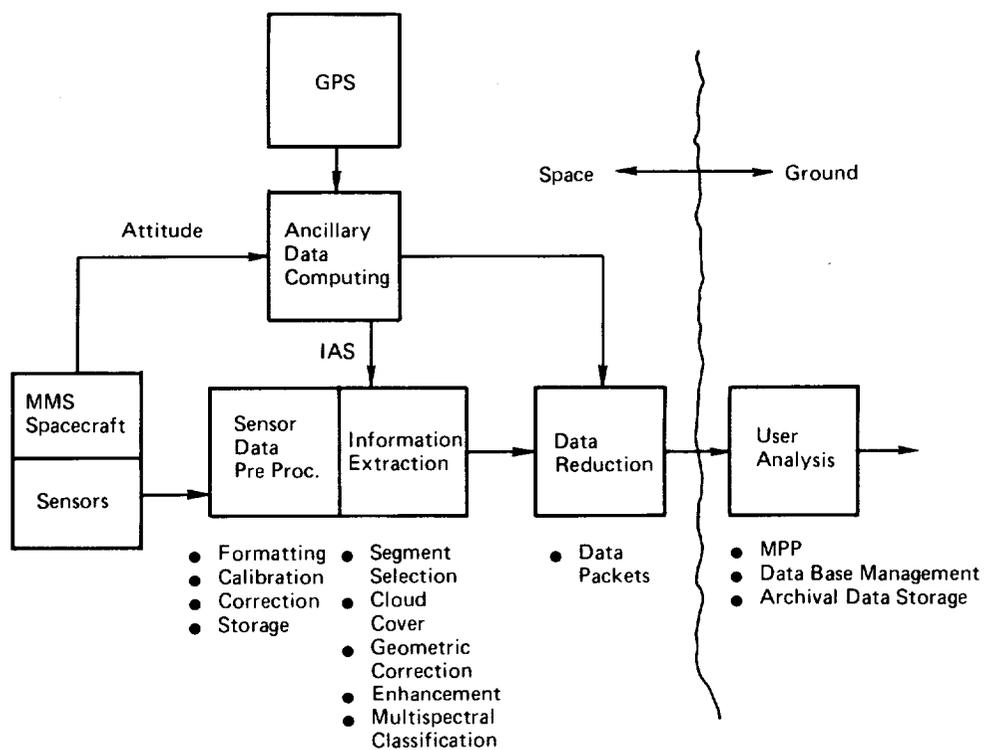


Figure 2. Future NASA Data Flow

NASA, as part of the NASA End-to-End Data System (NEEDS), is developing this on-board processing capability to adaptively control and process sensor data. Figure 3 illustrates the total NEEDS data flow. Of interest is the Information Adaptive System (IAS). This on-board capability will interface directly with remote sensors to provide on-board data control, data formatting, calibration, preprocessing, and data set selection as shown in Figure 3. The on-board system will be capable of processing multiple sensors; each sensor having 4-7 spectral bands, 26-100 detector elements, and a composite sensor data rate of 300×10^6 bits per second. The development of future solid-state sensors with as many as 6000 detector elements with an integral functional capability to adapt or respond either to high level instructions or commands or to information contained within the sensed data itself is expected to profoundly affect the architecture and design of on-board data processing over the coming decade.

The On-Board Multispectral Classification Study, Report No. NASA CR-3134, has been modified to add four additional tasks. These additional tasks will provide background data and information pertinent to the on-board IAS requirements.

The four tasks are as follows:

- Task 1 Sensing Characteristics for Future Space Applications

- Task 2 Information Adaptive System (IAS) Architectural Approaches

- Task 3 Data Set Selection Criteria

- Task 4 On-Board Functional Requirements for Interfacing with
Global Position Satellites (GPS).

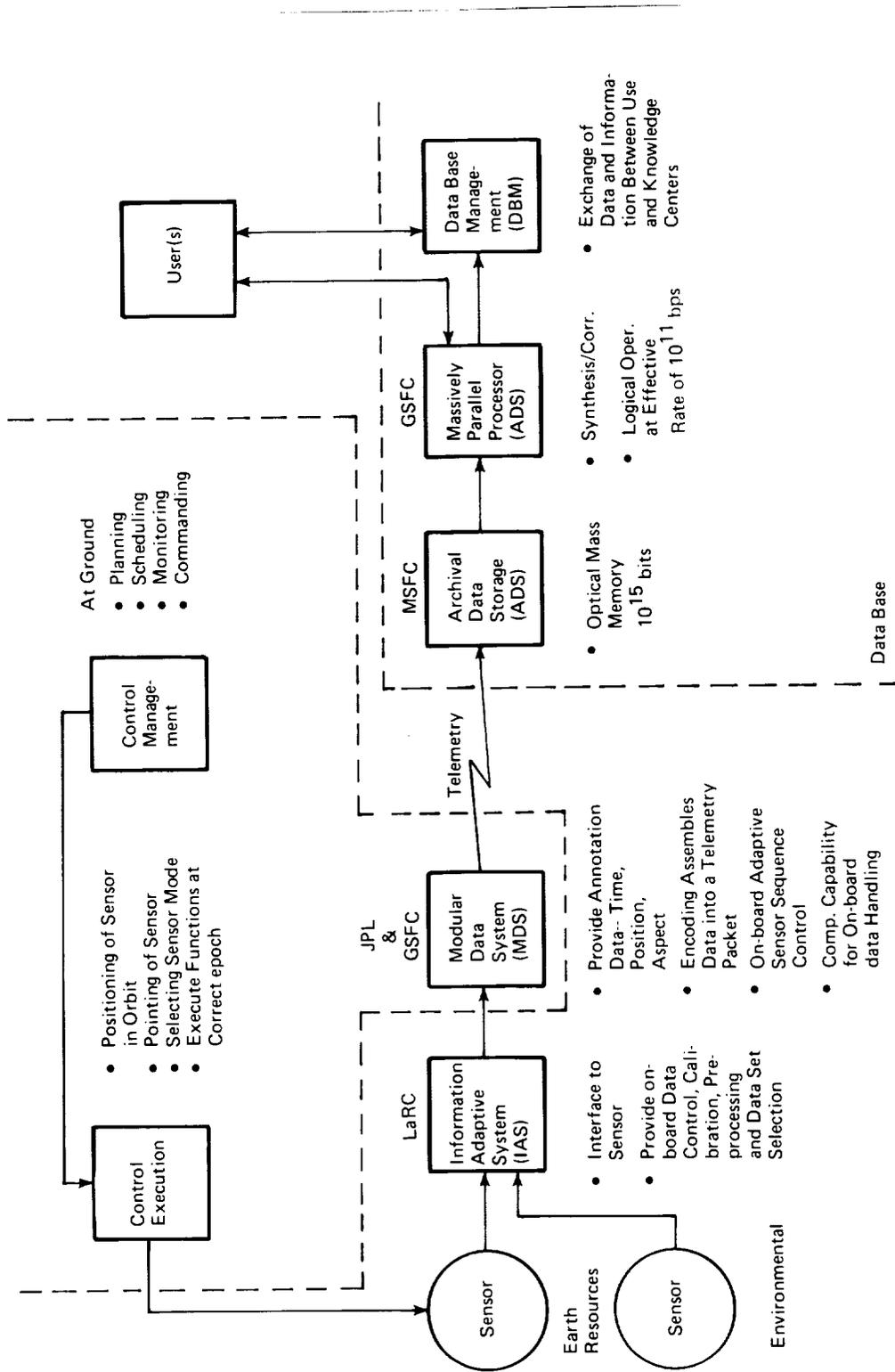


Figure 3. NEEDS Data Flow

The sensing characteristics considered for future planned space applications include data rate, data format, and radiometric techniques.

A data set selection criteria will be defined to determine whether acquired sensor data are of value and warrant further processing (example-cloud cover). Criteria currently used on the ground has been reviewed and where feasible proposed for on-board use.

The IAS architectural approaches considered will perform the following single functions or an appropriate combination of the functions:

- a) Formatting
- b) Radiometric calibration
- c) Data set selection (editing)
- d) Extraction of key data characteristics
- e) Extraction of specific information quantities desired by the user.

Four architectural approaches were reviewed and one analyzed in detail.

At present the precise geometric location of a pixel is determined by post flight analysis as a ground processing function. The pixel location (in geodetic coordinates) can be determined on-board by using position and time information from Global Position Satellites (GPS) and spacecraft attitude. The functional requirements and potential accuracy of using position and time information from GPS along with

spacecraft attitude to compute on-board the ground location to which an imaging system is pointing has been investigated.

The results of these tasks will aid in establishing the overall system requirements necessary for processing remote sensor data on-board a spacecraft.

2.0 TASK 1 - SENSING CHARACTERISTICS FOR FUTURE SPACE APPLICATIONS

The purpose of this task is to define the sensing characteristics for future planned space applications including data rates, data formats, and radiometric calibration techniques.

Currently these are three types of sensors employed for earth applications missions:

- "Whiskbroom" or electromechanical scanners
- Electron beam imagers
- "Pushbroom" or self-scanned solid-state imaging sensor arrays.

Electromechanical scanners accomplish image formation by mechanical scanning. As shown in Figure 4, mechanical motion causes the scene to be sampled in the across-track direction by a detector or array of detectors while spacecraft motion provides the orthogonal scan component. Examples of electromechanical scanners are: the Multispectral Scanner (MSS), and the Thematic Mapper (TM).

Electron-beam imagers, shown in Figure 5, are those devices which, by means of a shutter or similar mechanism, put an image, fixed in time, of an entire scene on a photo-sensitive surface by means of a beam of electrons after the shutter has closed. An example is the Return-Beam Vidicon (RBV).

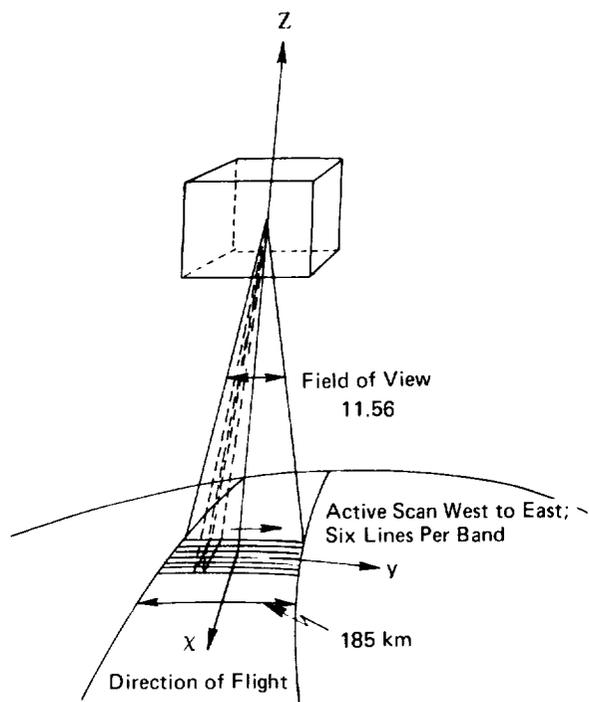


Figure 4. Multispectral Scanner (MSS) and Its Field of View

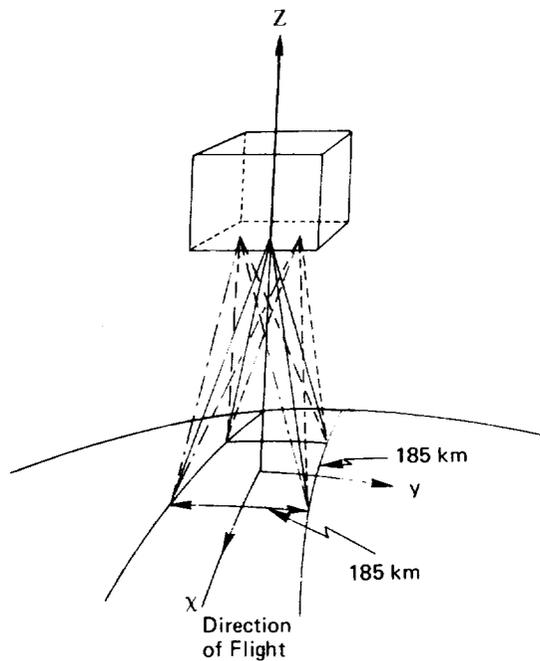


Figure 5. Return Beam Vidicon (RBV) Camera and Its Field of View

Self-scanned solid state imaging sensors use line or area arrays consisting of thousands of detector elements in a "pushbroom" scan mode. As shown in Figure 6 the across track scene is imaged on the array which electronically samples the image (no mechanical scanning), while the spacecraft motion provides the orthogonal scan component. The scanning electronic circuits are integrated on the same semiconductor wafer with the detector elements.

Electromechanical scanners and electron beam imagers have been extensively reviewed elsewhere. Solid-state self-scanned arrays offer important advantages for future earth-orbit satellite missions and will be further discussed within this section.

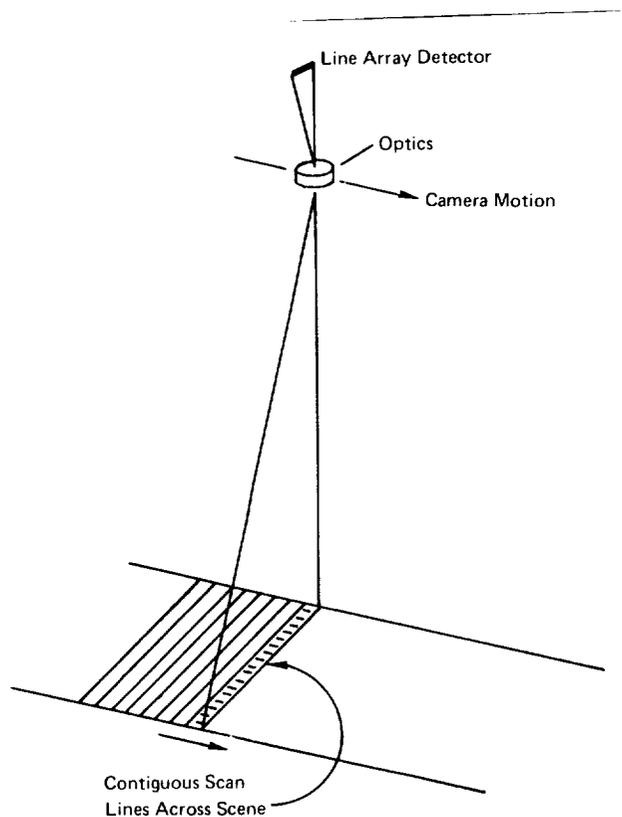


Figure 6. Illustration of the Pushbroom Scan Technique

2.1 Solid-State Self-Scanned Arrays

The advantages of solid-state self-scanned arrays include:

- Precise geometric positioning of the detectors
- Very high sensitivity and favorable signal-to-noise ratio (S/N)
- Low power consumption
- High resolution
- Good stability
- No mechanical scanning.

As with other sensors, solid-state self-scanned arrays require geometric, radiometric, and electrical calibration. Geometric calibration is straightforward and, once established, is subject only to mechanical distortions of the sensor structure. Element-to-element variations in photo-sensitivity or dark current must be calibrated out. Depending on the sensor design, additional calibration must be provided when additional per-element variations exist; for example, phototransistors require gain or linearity calibration, usually at two or more points, plus additional calibration for temperature variations. Generally, determination of photosensor geometry by the photomasking process during fabrication, and the chemical stability of the silicon/silicon dioxide system make the geometric, radiometric, and electrical characteristics of solid-state self-scanned arrays very stable.

The sensor data rate and data format are the characteristics most affecting the sensor data processing requirements.

Three candidate sensors types are: photodiodes, phototransistors and charge-coupled devices (CCD's). All three are similar in mechanism, limitations, and potential of the sensing portions, but differ in the arrangements for signal amplification and scanning. In photodiode and phototransistor arrays, scanning is accomplished by switching, while, in charge-coupled devices, scanning is accomplished by movement of potential wells.

In general, there are two self-scanning modes: digital multiplexing and analog charge-transfer. The photodiodes and phototransistors ordinarily use digital multiplexing. The CCD sensors have analog charge-transfer readout. Since all of the sensors are silicon, their operation is restricted to the visible and near infrared (1.1 μm).

The readout circuitry is generally on the same silicon chip as the array of detector elements (usually 100 or more detectors per chip). These chips can then be assembled into line arrays of several thousand elements for a high-resolution, pushbroom mode of scanning.

2.1.1 Sensor Data Rate

The sensor data rate and sensor data format are the characteristics most affecting the sensor data processing requirements. The sensor data rate is determined based on the following parameters:

- Orbital parameters
- IFOV size (resolution)
- Detector dwell time

- Cross-track swath width
- Number of spectral bands, N_b
- Imager efficiency
- Number of bits per pixel, N_{bp}
- Oversampling factor
- Data overhead percentage.

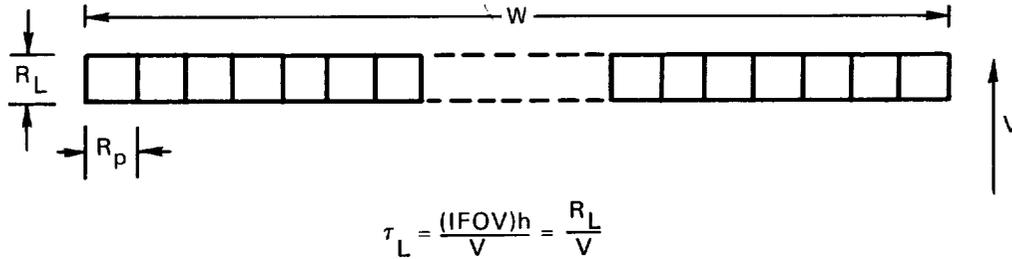
One of the principal advantages to the pushbroom scan technique using long linear arrays of solid-state detectors is that the approach allows the photon flux from the scene to be integrated during the time required for the instantaneous field-of-view (IFOV) to advance the dimension of one resolution element on the ground. Referring to Figure 7 the dwell timer T_L per detector for the line array is

$$T_L = \frac{(\text{IFOV}) h}{V} = \frac{R_L}{V} \quad (1)$$

where R_L = resolution in the along track direction, m
 V = velocity of subsatellite point, m/sec.
 IFOV = instantaneous field-of-view, radian
 h = satellite altitude, m

The data rate for the line array is then determined by the following expression

$$D_L = \frac{N_b W N_{bp}}{T_L R_p} \quad (2)$$



where

- R_L = Resolution in the Along-Track Direction, m
- V = Velocity of Subsatellite Point, m/sec
- IFOV = Instantaneous Field-of-View, Radians
- h = Satellite Altitude, m
- W = Swath Width, m

Figure 7. Line Array Dwell Time

- where
- N_b = number of spectral bands
 - W = swathwidth, m
 - N_{bp} = number of bits per pixel
 - T_L = line array detector dwell time, seconds
 - R_p = resolution in the cross-track direction, m

By way of comparison, the detector dwell time for the mechanical scanning array is dependent upon the scan rate, S , in the cross-track (along-scan) direction.

$$S = \frac{V}{2n (IFOV) h} \quad (3)$$

where V = velocity of subsatellite point in m/sec
 n = detectors per band or the number of lines imaged per scan
IFOV = instantaneous field-of-view, μ radian
 h = satellite altitude, m
and the factor 2 implies a bi-directional scanner.

The dwell time, T_s , of each detector is then

$$T_s = \frac{(\text{IFOV}) h}{2 \theta h S} = \frac{\text{IFOV} h}{2 \theta h V} = \frac{(\text{IFOV})^2 nh}{2n(\text{IFOV})h \theta V} \quad (4)$$

where θ = cross-track swath width, radians

The data rate for the scanning sensor is

$$D_s = \frac{n N_b N_{bp}}{T_s} \quad (5)$$

As an example consider the following parameters

$$\begin{aligned} N_b &= 4 \\ W &= 185 \times 10^3 \text{ m} \\ N_{bp} &= 8 \\ R_p &= 30 \text{ m} \\ n &= 16 \\ \text{IFOV} &= 42 \times 10^{-6} \\ h &= 705 \times 10^3 \text{ m} \\ \theta &= .2606 \\ V &= 7.5 \times 10^3 \text{ m} \end{aligned}$$

$$D_s = \frac{16 \times 4 \times 8}{10.2 \times 10^{-6}} = 50.2 \times 10^6 \text{ bps} \quad (6)$$

$$D_L = \frac{4 \times 185 \times 10^3 \times 8}{30 \times 3.95 \times 10^{-3}} = 50 \times 10^6 \text{ bps} \quad (7)$$

The data rates are essentially identical. This is as it should be since the swath covered, number of bands, number of bits per pixel and the orbital parameters are the same for each sensor.

But note the relative dwell times between the two sensors

$$\frac{T_L}{T_s} = \frac{\frac{(\text{IFOV}) h}{V}}{\frac{(\text{IFOV})^2 n h}{\theta V}} = \frac{\theta}{(\text{IFOV}) n} \quad (8)$$

or for $\theta = .2606$ radian (185 Km at 705 Km altitude)
 $n = 16$
 $\text{IFOV} = 42 \times 10^{-6}$ radian

$$\frac{T_L}{T_s} \approx 400 \quad (9)$$

This indicates one of the major advantages of the line array. The line array allows an increase of about a factor of 400 over the mechanical scanner in signal-to-noise ratio. This improvement is significant, and permits smaller aperture optics to be used with a consequent reduction in size and weight.

2.1.2 Radiometric Calibration

Radiometric calibration must provide for the maintenance of an absolute radiometric accuracy. The calibration must provide precise calibration for band-to-band relative accuracy, and yet more precisely for channel-to-channel relative accuracy within a band.

The critical elements in radiometric correction of detector arrays are:

- Provide a highly stable operating temperature at the array, and stable bias voltage.
- Provide updates of calibration files at the beginning of an orbital pass and at the end to determine if any drifts have occurred.
- Have an extensive ground calibration procedure to catalog array performance under various bias voltage and focal plane temperature configurations.
- Plan to have most elements corrected using a simple equation of a straight line. For the other elements, either linear segment approximations with five or more calibration points per detector or some complex polynomial fit will be required.

Solid-state detector arrays exhibit a systematic error in response uniformity from pixel to pixel known as fixed pattern noise. Pattern noise is a variation in output level from different elements in the array due to differences in element characteristics, such as sensitivity and leakage of the detectors and offset and other variations in associated amplifiers and samplers. Pattern noise can, in principle, be measured and eliminated from the image.

Pattern noise is primarily due to an interelement variation in dark current (detector output in the absence of photoelectrons) and, secondarily, to a variation in sensitivity. The effects of the interelement variation in a linear array sweeping over an area shows up in an image as dark and light lines (stripes) parallel to the track. In a system where the ratio of two outputs is assumed to be the ratio of the input light intensities for special analysis (i.e. classification), the results of low-light levels would be grossly inaccurate without dark-level compensation. To a large extent, the dark-level variation is due to leakage that can be reduced substantially by cooling the array.

The electronic readout for the detector elements is a source of heat resulting in a rise in array temperature along with a subsequent increase in dark current. One published evaluation (Reference 5) of a self-scanned array used in an astronomy application indicates orders of magnitude change in dark current when several readouts were performed within a 100 millisecond time period. A recent study (Reference 2) indicates a change of 10°C results in a 2:1 change in dark current.

Temperature fluctuation in the arrays can also be expected from ambient temperature variations. Responsitivity variations are wavelength dependent so that corrections for temperature changes vary with spectral band.

Providing a highly stable operating temperature at the array is therefore a major factor in reducing the uncompensated pattern noise.

Dark current compensation is certainly a function of temperature and, to a less extent, of time. It is possible to obtain patterns of dark output at various temperatures before launch and use the data on the ground after selecting the appropriate pattern from telemetered temperature data.

For cases where responsivity and dark current variations occur uniformly, only one detector element needs to be monitored to determine correction factors which can be applied to the signals from all elements. Some concepts for this use the measurement of responsivity and dark offset either for an imaging element (during a time when the shutter is over the imaging optics) or a non-imaging element which is blocked from the image and exposed to a known radiant power level. Another concept would involve the addition of a thermocouple to the array to directly measure temperature changes from which correction factors could be determined. Another approach is that instead of measuring temperature directly, temperature changes are inferred from dark current measurements made either when a shutter is over the optics or by masking off one detector element. The masking off of one element is preferable since it involves no sacrifice in image integration time.

This approach is illustrated in Figure 8 for an on-board implementation. A commandable shutter is used to provide dark current data at any time in orbit and a capability to record dark current on-board. Two storage registers of dimensions equal to that of the detector array for storing dark current values and response nonuniformity values are required on-board. The measured dark current will be subtracted and the proper gain coefficient applied to normalize the response. Rather than perform multiplicative operations, a table-lookup may be involved to linearize any nonlinear effects, or normalize nonlinear differences. One consideration is that any linear or nonlinear correction of discrete data values implies an output quantization level higher than that of the input data being corrected. This may not be desirable from a data volume standpoint but performing a nonlinear correction with the same output quantization involves some information loss, as well as some compromise in the data certainty.

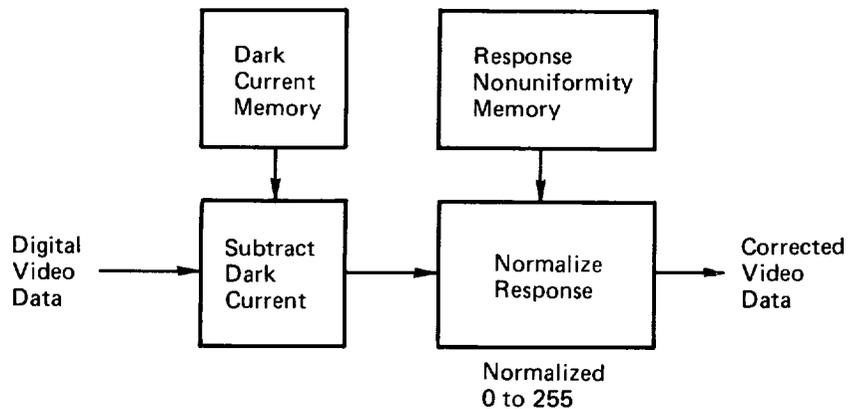


Figure 8. On-Board Removal of Fixed Pattern Noise

The simplest correction involves a gain coefficient (G) and bias (B).

$$S_{out} = (S_{in} + B)G \quad (10)$$

When the sensor response is nonlinear, or if different detectors have different nonlinearities, higher-order corrections are involved, such as:

$$S_{out} = (S_{in} + B)^{\gamma} \quad (11)$$

or more generally

$$S_{out} = S_{in} + G_1 (S_{in})^2 + G_2 (S_{in})^3 + \dots + B \quad (12)$$

The above approaches have assumed that dark currents and responsivity have occurred uniformly. Concepts for compensating non-uniform temporal changes use the same principles as before but require that the output of an entire array be monitored to develop correction factors unique to each image (detector) element.

Radiometric specifications for future platforms should be made sufficiently complete to assure meeting user expectations. Distinctions among the various types of calibration should be explicitly defined. These types include:

- Absolute
- Relative spectral
- Temporal stability.

Absolute calibration requires maintenance of an extremely stable reference. A 5 percent absolute accuracy is a suitable goal. A relative spectral radiometric accuracy of 1 percent is achievable using a straight line correction. A relative accuracy of 1/4 of 1 percent will require some complex polynomial or nonlinear correction.

The radiometric calibration and correction process selected for any operational on-board system can seriously affect both the computational and operational requirements. As previously mentioned, each individual detector of a given array may have its own response characteristic. This requires that the entire array output be monitored to develop correction factors. By observing the response of each detector at various input radiance levels one can see the variation across the array. Also, by observing the same line number and noting the variation in channel number, one can see the different spectral responses.

Nonlinear sensor response presents another level of on-board complexity. More calibration data must be collected to determine the correction coefficients implying more on-board storage and computational power required for implementing the radiometric correction.

The following represents a procedure currently employed to determine the radiometric coefficients for a 1728 detector element line array.

Calibration data consists of array outputs for each element at five pre-determined radiance levels. 576 samples are run for each element at each level. The 576 values are averaged and the results stored for use in normalizing the picture data. Any missing element data is generated by averaging between two adjacent elements to compensate for the missing data.

The image data is read and stored. During the reading of the data, the inoperative elements are also removed by adjacent element averaging. Two 576 element data lines (full line) are read in. The even numbered elements are stored for one access interval and then placed into the storage array with the data from the odd numbered elements read during the next interval forming a 576 element line. This removes the two pixel displacement (across scan) inherent in the staggered photodiode array. The elemental data values are compared to the calibration data and, by interpolation between the nearest higher and lower calibration values, normalized to a range of 0 to 255. One thousand, seven hundred twenty-eight (1728) lines of normalized data from each 6-chip section are accumulated in a storage area until the scene data from all 3 sections have been read.

The final step of a scene data process is to retrieve the data from storage, taking a line from each 6-chip section, to form a single 1728 element data line.

A major advantage of the solid-state array is that the one-dimension optical image (line image) and the sensor array, consisting of a series of small chips, can be made to follow any image curvature in a piecewise approximation manner. This high cross-track geometric fidelity is achieved along each linear array to the extent that the position of each individual detector is precisely known. Spacecraft motions will limit the ability to attain geodetic fidelity along track.

Besides the geometric accuracy within a single array, accurate positioning of arrays for each spectral band in the image plane with respect to each other allows very close multispectral registration of the resulting images.

2.1.3 Data Format

The array data format is determined by the data readout approach. There are various approaches to the readout of a line array. These include a pure sequential readout, a combination sequential/parallel, and an all-parallel approach as summarized in Table 1.

The simultaneous parallel readout of all array detectors is difficult to achieve due to excessive video bandwidth requirements and to the fact that the detector's dual role of integration and intermediate storage would be lost.

Table 1. Data Readout Approach Comparisons

TYPES OF READOUT	DESCRIPTION
<ul style="list-style-type: none"> • Sequential All N sensor channels in one analog multiplexer and A/D converter 	<ul style="list-style-type: none"> • Would require very high speed, complex A/D converter and analog multiplexer • Not readily adaptable to modular growth
<ul style="list-style-type: none"> • Sequential/Parallel One multiplexer and A/D converter per band or multiple chips within a band 	<ul style="list-style-type: none"> • Analog multiplexer and A/D converter more state-of-the-art • Dedicated circuitry allows graceful degradation • Modularity allows easy accommodation of growth
<ul style="list-style-type: none"> • Parallel Analog multiplexer and A/D converter per chip 	<ul style="list-style-type: none"> • Excessive complexity, bandwidth

A fully serial sequential readout of all detectors appears to be a good candidate approach. The advantages are that it truly represents a line-scan system and the output video would require no reconstruction, interleaving, or reformatting. The disadvantage is that the monolithic circuitry associated with the sequential access and readout of 6000 or more detectors requires the basic sampling interval be 1 Mhz whereas, inherent solid-state properties of silicon dioxide CMOS switching and sensing arrays limit the sequential sampling rate to lower rates.

The sequential/parallel data readout provides a compromise solution. Here one analog multiplexer and A/D converter is used for multiple chips. The multiplexer and A/D converter represent state-of-the-art designs while still providing a modular growth capability to accommodate longer arrays with more elements.

Figure 9 illustrates a simplified block diagram utilizing this sequentially/parallel approach. As shown, each detector array chip contains 200 detector elements. Four of the detector array chips are input to a single analog multiplexer and A/D converter. The A/D output is serial-to-parallel converted and radiometrically corrected, then output to a digital multiplexer.

Each processing element contains 2K words of storage (32 bits per word). The storage is split between the radiometric calibration coefficients and the sensor data. There will be a gain and bias coefficient (8 bits each) for each of the 800 elements being multiplexed or .4K words of coefficient storage. Each element will store eight successive pixel values for a total of 1.6K words (8 pixel/element x 800 elements x 8 bits). Accumulating the 8 pixels allows the processing element to be more efficiently utilized.

The linear array photodiode (or phototransistor) chip is an integrated circuit and contains a staggered linear array of detectors and the first

$$\frac{800 \text{ Samples} \times 8 \text{ Bits}}{4 \times 10^{-3} \text{ sec.}} = 1.6 \text{ Mbps}$$

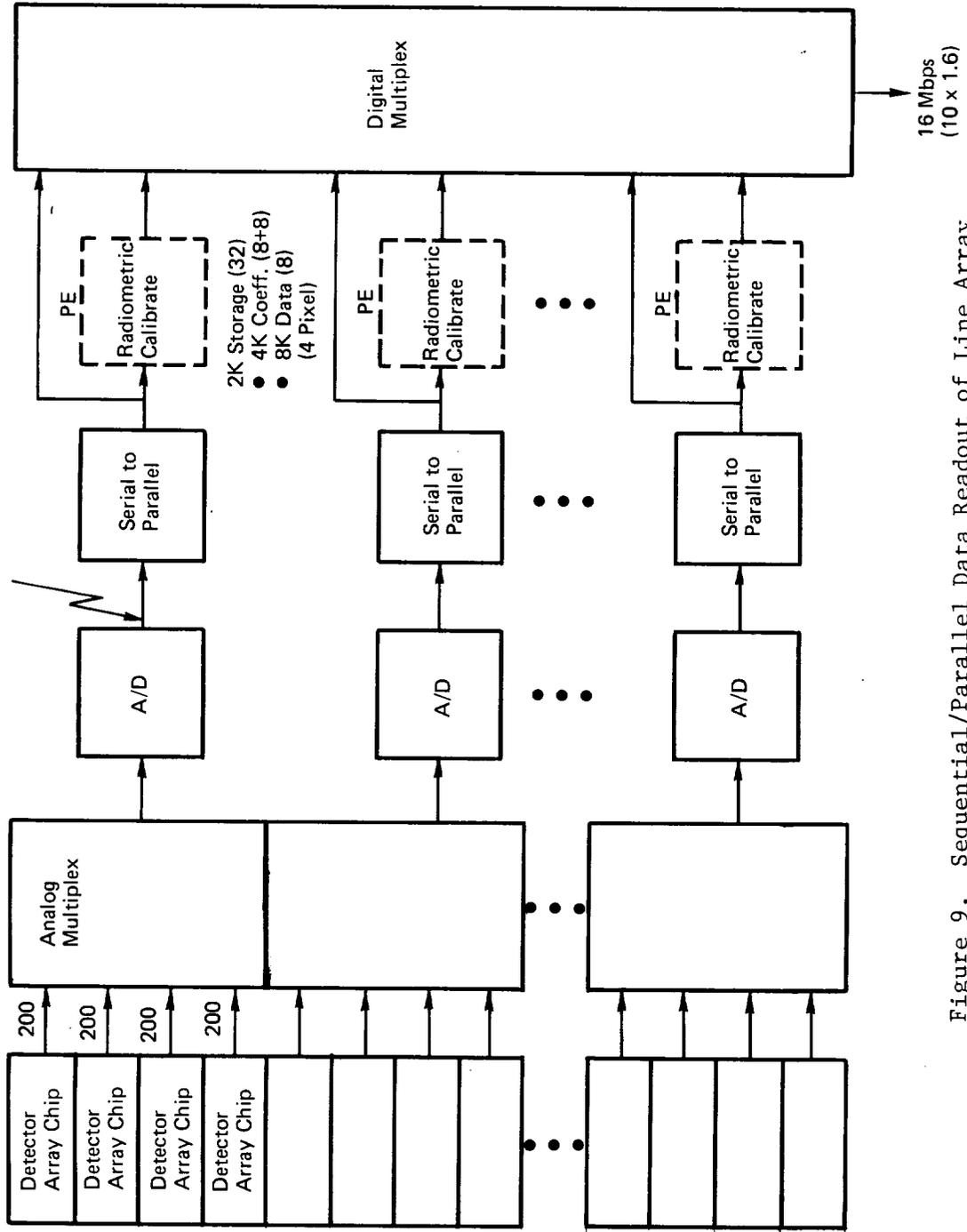


Figure 9. Sequential/Parallel Data Readout of Line Array

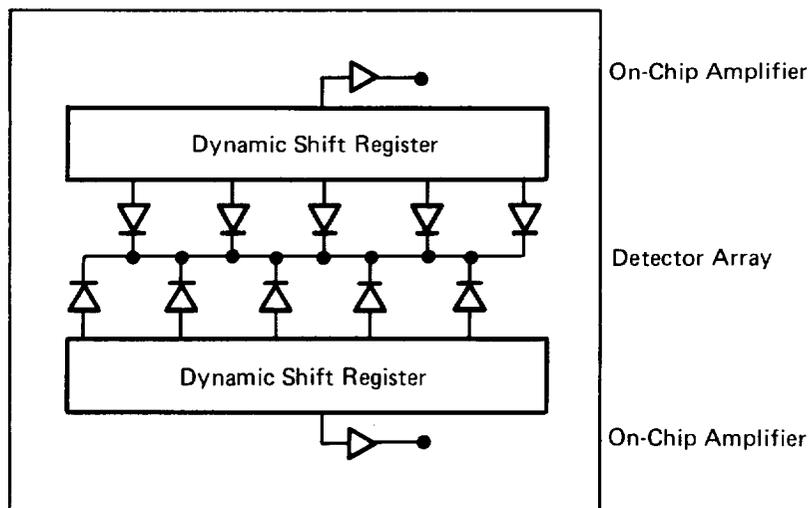


Figure 10. Integrated Detector Array Chip

levels of preamplification and multiplexing. Figure 10 illustrates one approach to the array organization showing how the detector signals are sampled and read off the chip. Each detector is sequentially connected, one at a time, to an on-chip amplifier. A dynamic shift register controls the sequence of the connections to the amplifier. The analog signal is converted to digital form, encoded, and the signals multiplexed out.

Because of the difficulty in making very long-line arrays monolithically, smaller arrays are used; however, smaller arrays pose problems in maintaining image continuity. To achieve image continuity, the edge of the chip must be placed very close to the end diodes, generally to within 0.3 mil.

Detector elements are usually in a bilinear configuration as shown in Figure 11 for the line array (i.e., the even numbered pixels are offset

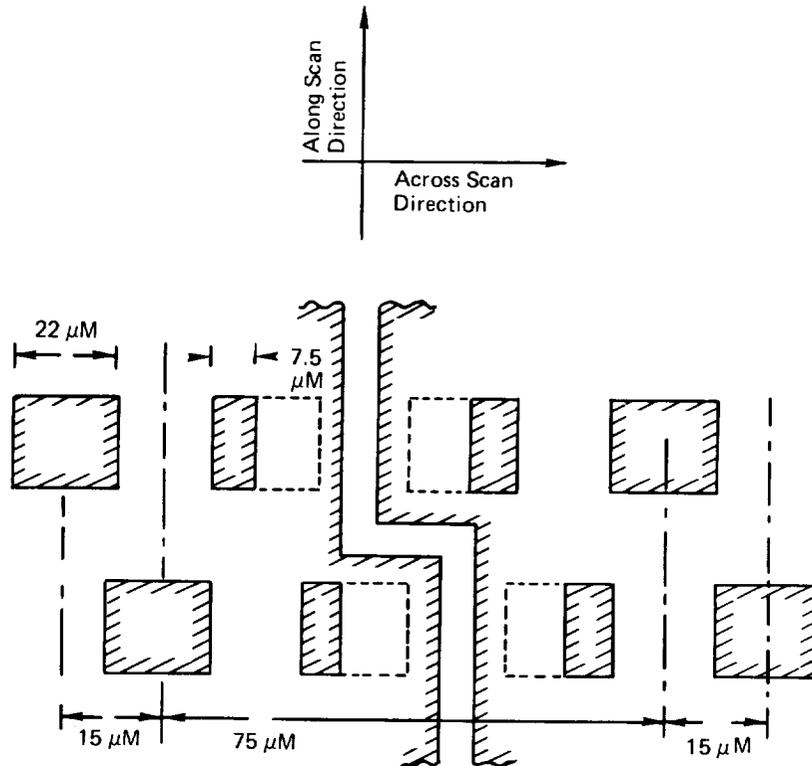


Figure 11. Bilinear Design (Detector and Edge Element Geometry)

from the odd numbered pixels). This allows sufficient physical space for each detector element and reduces the amount of crosstalk between adjacent detector elements.

Other suggested approaches end up with bilinear configuration where the chips are spatially staggered over distances equivalent to 200 pixels. Another advantage of a segmented contiguous array is that a piecewise approximation to curved focal surfaces is permitted. In high resolution systems, this may be required. The singular disadvantage of the contiguous approach is that typically one 15μm space is required at the chip edges. This is equivalent to having a dead element in the array. If there are many small chips used, a number of elements will require smoothing.

2.1.4 Geometric Requirements

Three geometric characteristics of the output images and image data are fundamental to the purposes of multispectral remote sensing:

- Spatial Registration - elemental registration of the scene information in different spectral bands-for image interpretation and spectral data analysis.
- Geographic Registration - determining the geographic coordinates of the scene information-for locating and delineating observed phenomena on the earth's surface.
- Temporal Registration - elemental registration of scene information from different coverages-for interpretation and analysis of time-dependent phenomena.

Spatial Registration

The correction required to achieve spatial registration of the sensor data in different spectral bands generally involves fixed offsets between bands due to the physical size of the integrated detector chip in the sensor image plane. Figure 12 illustrates this for the one detector array configuration. Here, the overall physical length of the detector chip is 210 mils with the detector elements spanning a distance of 57 mils or about a 4:1 ratio. Thus, if several bands were butted together, there would be a physical separation of the bands equivalent to this ratio.

Since the data from the solid-state array is output a full line at a time and the various bands are separated in the along-track direction by a large amount (i.e. as much as 200 equivalent pixels), the total storage

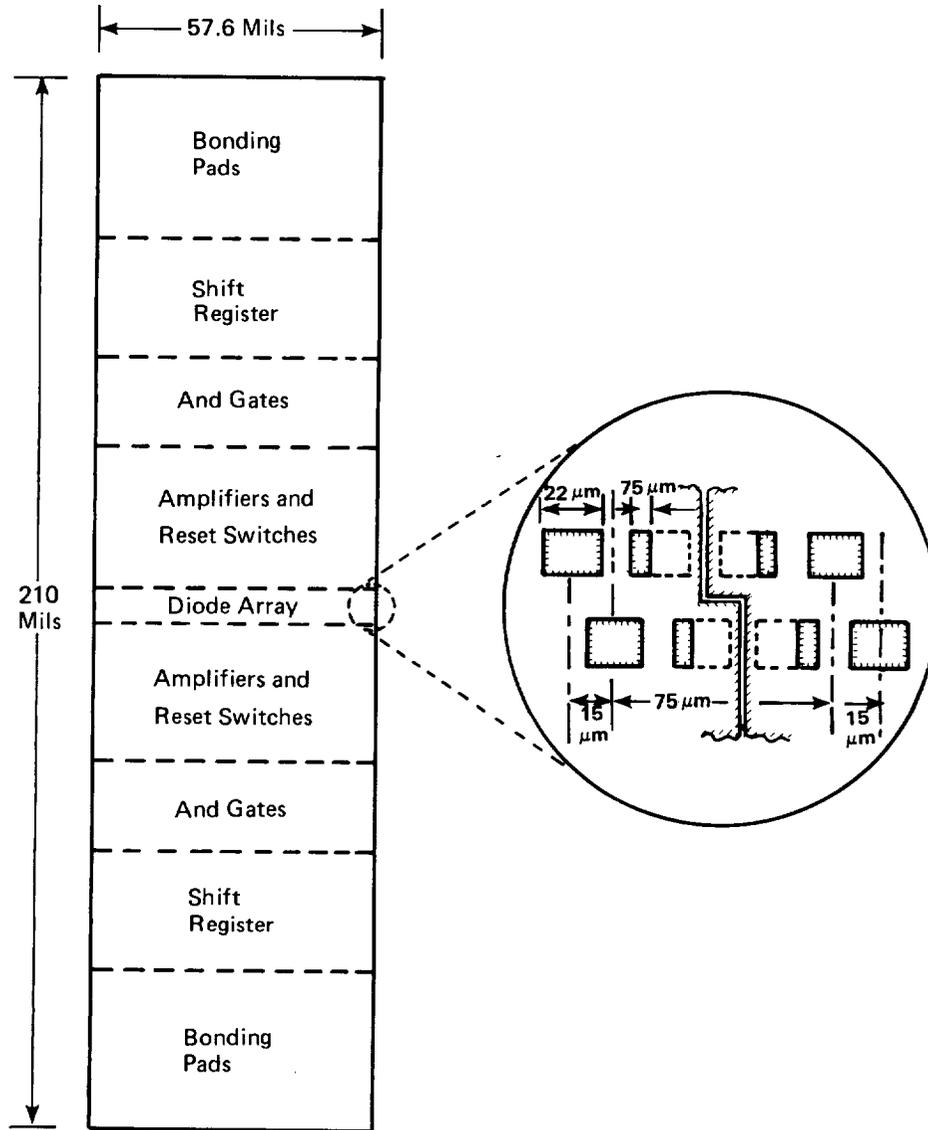


Figure 12. Photodiode Detector Chip Showing Associated Electronics and Geometry

required to register all bands can become significant. Figure 13 illustrates a four band linear array with 2000 pixels per line and a 200 pixel offset between bands. The on-board storage for achieving spatial registration would be 2.4 Mbits.

Geographic Registration

The geometric functional requirements can be considered two ways. It can involve: (1) identification of earth coordinates in images and/or data remaining in or recorded in the sensor-coordinate frame or (2) transformation of the sensor data into images and/or data in an earth-based coordinate system. The first approach requires determination of the sensor-to-earth coordinate relationship in a way unique to that image and/or data set. This does not transform the data itself but determines certain earth locations (e.g. grid intersections), determining the position of points other than those identified requires a nonuniform spatial interpolation unique to that image.

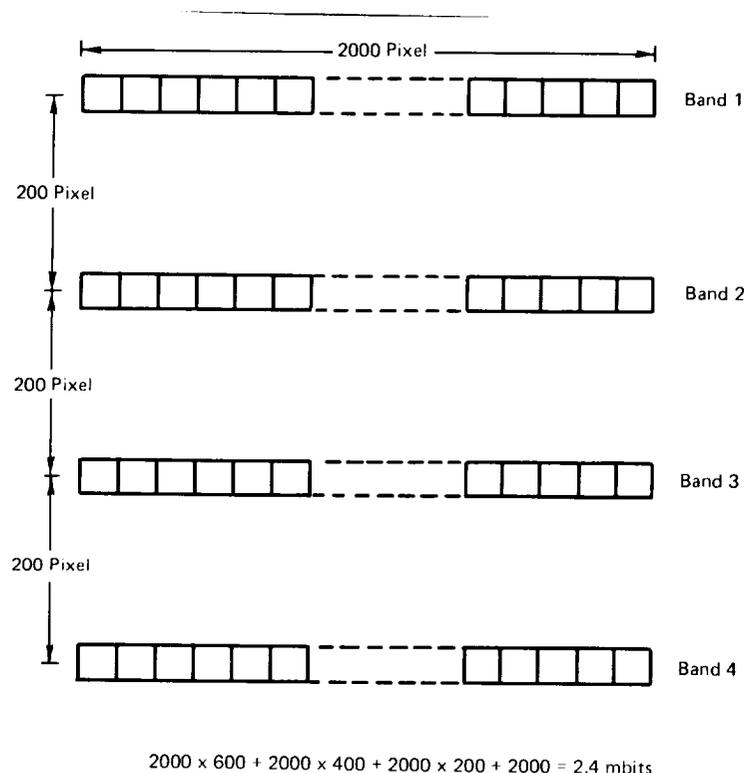


Figure 13. Solid-State Self-Scanning Array Spatial Separation

The second approach involves a transformation of the data according to the known projection geometry and orientation of the sensor into a standard projection, and then identifying the position and orientation of the transformed data set on the earth's surface.

A version of the first approach is generally used. This mapping process is simplified for the solid-state array due to the high cross-track geometric fidelity achievable along each linear array and the fact that the complex mechanical scan mechanism has been eliminated. This eliminates sensor related errors (in the cross-track direction) of

- Mirror scan linearity
- Scan line length
- Detector sampling delay
- Scan skew.

Although the processing of the sensor data is less due to fewer sensor related corrections required, the geometry of the line arrays is such that much more storage of data is required if the data is to be interleaved by pixel.

2.1.5 Charge-Coupled Imaging Device Chips

It appears that it will be possible to fabricate charge-coupled imaging device chips with considerably more sensing elements per chip than is possible with photodiode and phototransistor arrays. This means fewer interfaces for a given total number of detectors, and therefore a smaller fraction of the image data which would require special processing to give a smoothed video signal, regardless of the specific requirements on smoothing at the interface.

One chip-butting concept of interest for a pushbroom-mode CCD orbital imager is shown in Figure 14. In the concept each chip has a parallelogram shape and consists of a main sensor array and a secondary sensor array. The displacement of the secondary array frame, the axis of the main array, and the length of the secondary array are minimized. The direction of image motion is such that, to read one line of the image, the signal from the smaller array is stored (off-chip) until the longer array scans that same line. This minimizes the amount of memory storage required. An advantage of this concept is that the ends of the sensor array are more accessible for charge detection and amplification.

Linear imaging arrays (using CCD's) can be designed in 3 ways:

- (1) A simple CCD shift register can be used if it is clocked in such a way that the shift-out time is very much less than the integration time. This condition reduces image smear caused by shifting pixels through light-sensitive regions.
- (2) An image can also be designed with separate sensors and shielded readout register. After integration in the sensor, the charge configuration is shifted into the shift register, i.e. a parallel-to-series transformation is effected. The line of video is then shifted out via the shift register while a new line is being integrated. This effectively eliminates the smear problem.
- (3) The third approach, uses a line of sensors and two shielded readout registers. After integration, odd-numbered pixels are shifted into one readout register and even-numbered pixels are shifted into the other readout register. The information in the two vertical registers is clocked into a two-bit

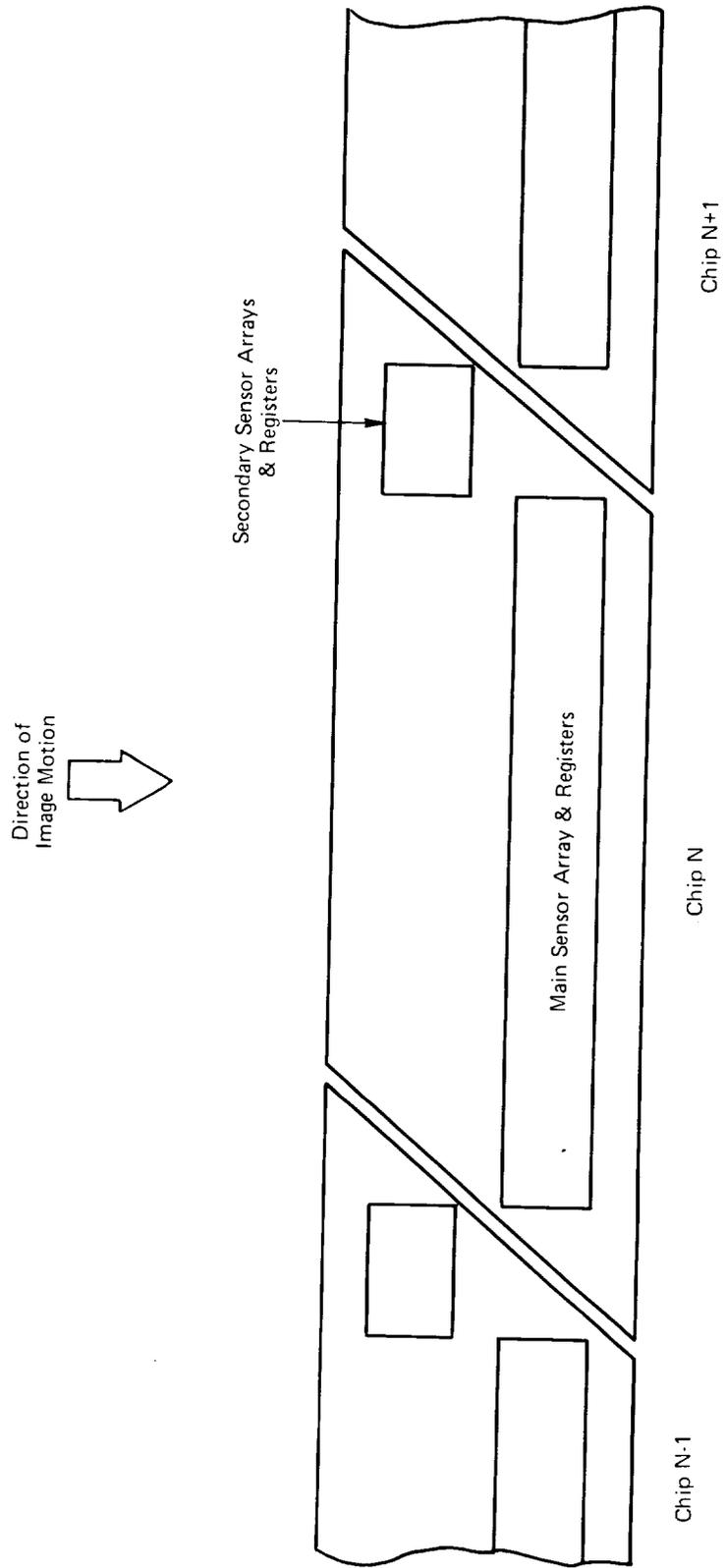


Figure 14. Pushbroom-Mode CCD Orbital Imager

horizontal register thus reforming the pixels in the order in which they were formed. The number of bits in each vertical register is half the total number of pixels. Thus, for two-phase vertical registers, the number of transfers required to clock out the pixel farthest from the output is equal to the number of pixels.

The primary advantages of this approach are higher sensor packing density and fewer transfers to read out a pixel.

Time Delay and Integration (TDI) Arrays

In applications where the CCD imaging chip has a velocity relative to the object to be imaged, the CCD can be used in a time delay and integration (TDI) mode to enhance the signal-to-noise ratio. In such applications which lend themselves to this mode of operation, a CCD composed of N_x columns each containing N_y bits is oriented in such a direction and clocked at such a rate that the transfer of pixels down the CCD column compensates for the movement of the image along the CCD columns due to the relative velocity of the chip and the object.

2.2 Conclusions

The sensors utilized for future planned space applications will most likely be solid-state self-scanned arrays offering the following advantages:

- Higher sensitivity and favorable signal-to-noise ratio (S/N)
- Low power and volume, with no moving parts, therefore higher reliability.

The arrays contain many more detector elements (thousands) than electro-mechanical sensors, each detector requiring radiometric correction. Procedures for accomplishing this radiometric correction without excessive computation or storage are currently under investigation. In addition, a large amount of on-board storage is required if the sensor data is to be processed in a BIP format due to the physical placement of chips forming a long array. As LSI and VLSI techniques mature these problems become less severe and the advantages of solid-state self-scanned arrays can be better realized.

3.0 TASK 2 - INFORMATION ADAPTIVE SYSTEM (IAS) ARCHITECTURAL APPROACHES

Several architectural approaches exist for the implementation of the required IAS functions. These include:

- Custom Designed Computer
- Federation of Functional Processors
- Distributed Microprocessor System
- Distributed Signal Processor System.

Figure 15 illustrates the more conventional central-processing approach. Here, a single high-speed processor carries the entire processing burden of the system. The programs for the processing tasks are contained in a single large memory and share the single CPU. The Executive Program sequences the tasks on the CPU in accordance with the requirements of the problem. Certain disadvantages do exist with the centralized approach:

- 1) High-speed data links are required to carry information between the central processor and the input channels. This high speed is generally achieved by paralleling a number of input devices (i.e. disks) resulting in high interconnection hardware costs.
- 2) The addition of input channels or the use of more sophisticated algorithms is limited by the thruput capability of the CPU making modular expansion difficult at best.

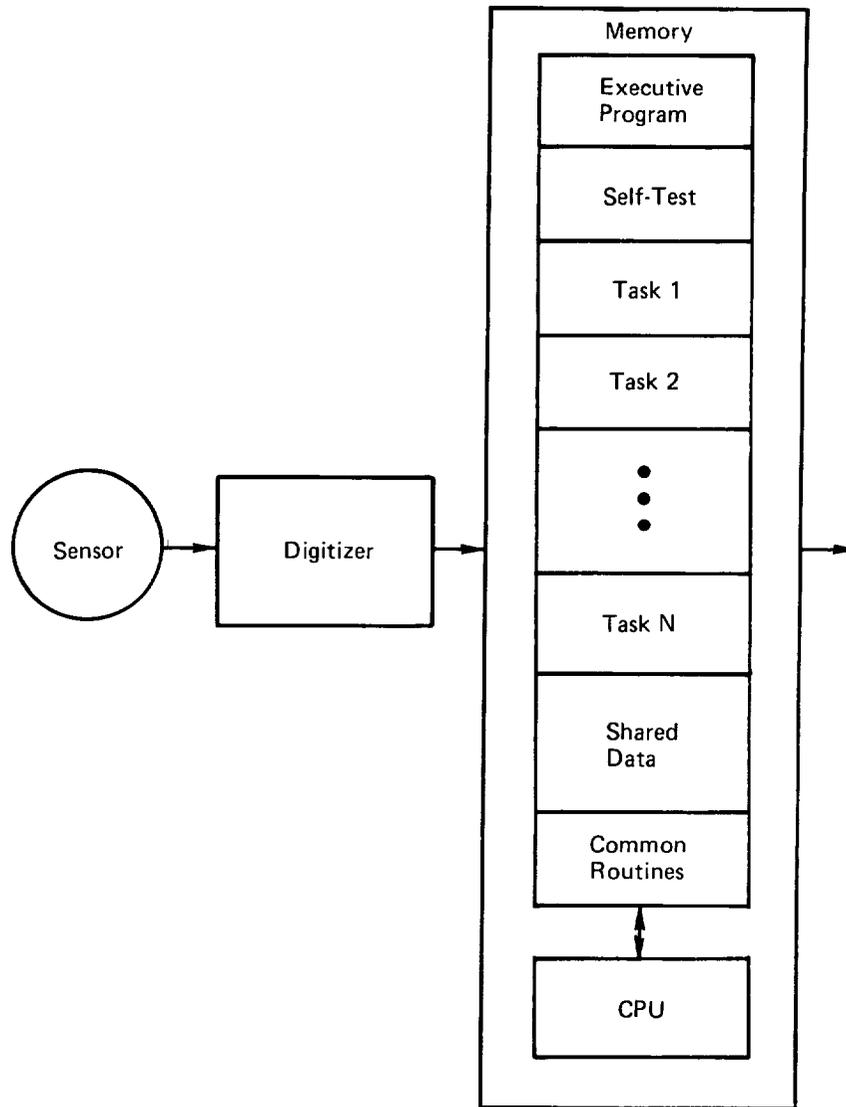


Figure 15. Custom Designed Computer

- 3) Because all input channels are controlled by a single processor, the software for the CPU must be carefully designed to provide for various interrupt conditions.

On the plus side the task coordination and scheduling, system data base management, I/O data management, and interrupt handling are all solved problems for the single CPU system. The implementation of these functions within Distributed Processing Systems is not a solved problem. A family of compatible peripheral and I/O devices, while costly, do exist for the centralized approach thereby reducing development costs. In addition, there are in existence extensive application routines for the centralized approach.

Distributed processing techniques, when coupled with microprocessor technology, provide a cost effective alternative to the use of custom designed computers. A fully distributed processing system consists of a multiplicity of processors that are physically and logically interconnected to form a single system in which overall executive control is exercised through the cooperation of decentralized system elements.

The processing element is a single LSI device, physically smaller and more reliable than custom designed CPU. It is quite likely that fault tolerance techniques can be developed which exploit the parallelism inherent in a distributed processing system.

Figure 16 illustrates a distributed signal processing system containing a number of processing elements (PE's) with each PE performing a portion of the processing required by the application. The PE can be either a microprocessor or a small signal processor depending on the requirements for a given application. IAS requirements would dictate the PE design be a small signal processor to accommodate the high thrupt.

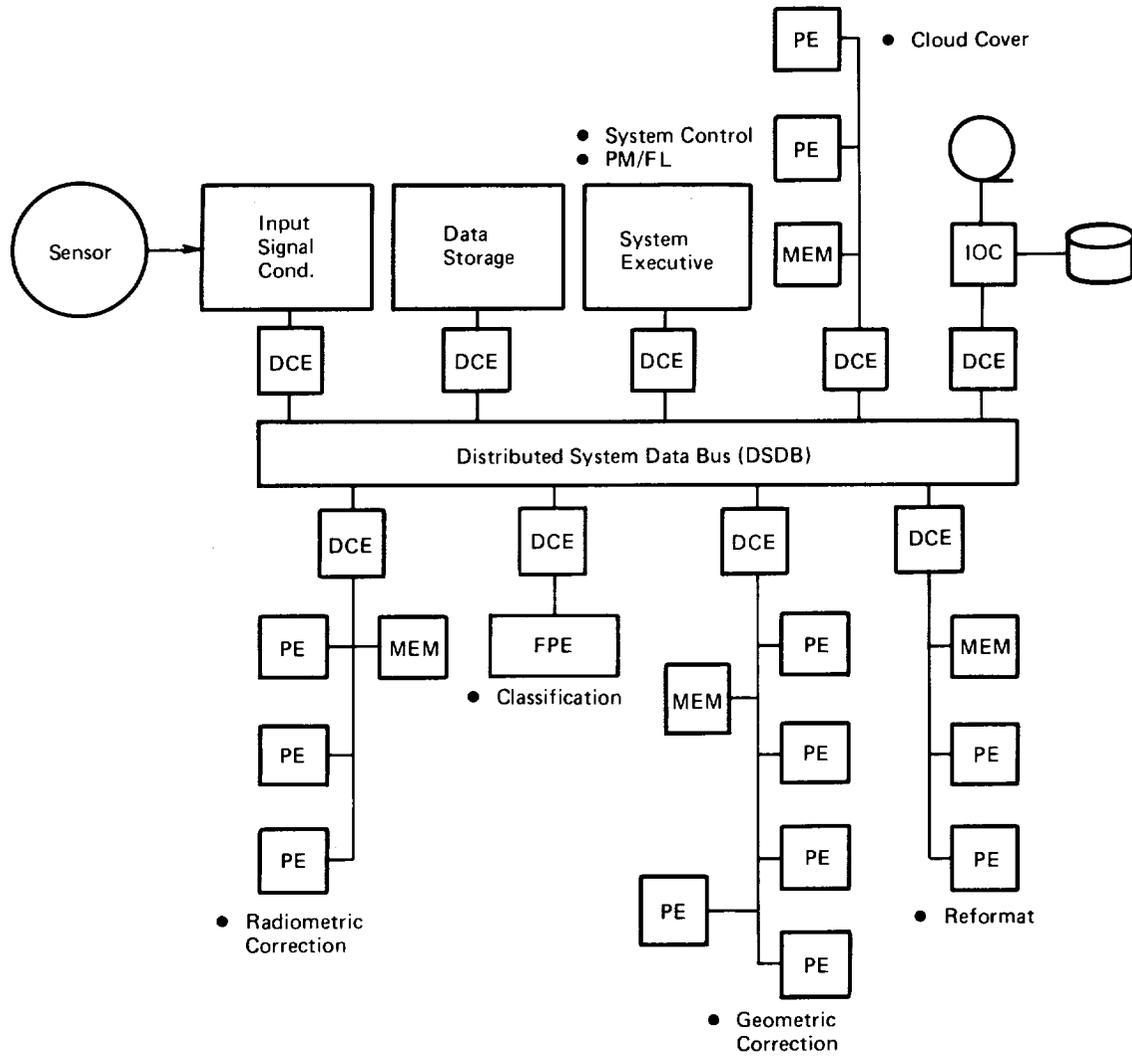


Figure 16. Distributed Signal Processor

The distributed architecture has three levels of processor hierarchy. At the lowest level is the processing element with its own local memory. At the next higher level, a cluster is constructed by connecting a number of processing elements via a cluster bus with a common memory for the cluster controlled by a distributed control element (DCE). Finally, a number of clusters are connected via a distributed system data bus (DSDB) such that all application functions can be accomplished. It is anticipated that some functions may require a computational capability beyond that of a cluster. Thus the architecture is configured to interface with special purpose hardware or functional processing elements (FPE). The classification function is shown as one requiring a FPE.

A distributed system has certain inherent advantages. Among the advantages are:

- Fault tolerance
- Lower life cycle costs
- Improved maintenance
- Standardization
- Resource sharing
- Reconfiguration ability.

The major disadvantages of distributed processing systems is that there are no proven approaches to implementing the multitasking executive programs for a distributed processing system. In addition, new system analysis and design verification and validation tools must be developed and put into place to reduce distributed processing to practice.

A distributed signal processor, currently under development, will be described in the next section.

A federation of functional processors (FFP) represents another type of distributed processing system where each processor represents a higher order of integration than that of a microprocessor. Each processor is itself composed of modules that implement high-level "operators" such as Multiply, Sum Accumulate, and Trig. Function Generate. Extensive use is made of modules of the same design. Use of these "off-the-shelf" modules assures that custom processors can be rapidly assembled at low cost to implement new or improved processing algorithms.

Figure 17 illustrates the FFP structural system. The illustrated FFP system includes input and output signal conditions, signal processor, data processor, and a dedicated PM/FL processor. These processing elements are interconnected by a multiple bus structure consisting of a control bus, interprocessor bus and a PM/FL bus. Additional processors can be attached to this bus structure as needed to satisfy specific applications.

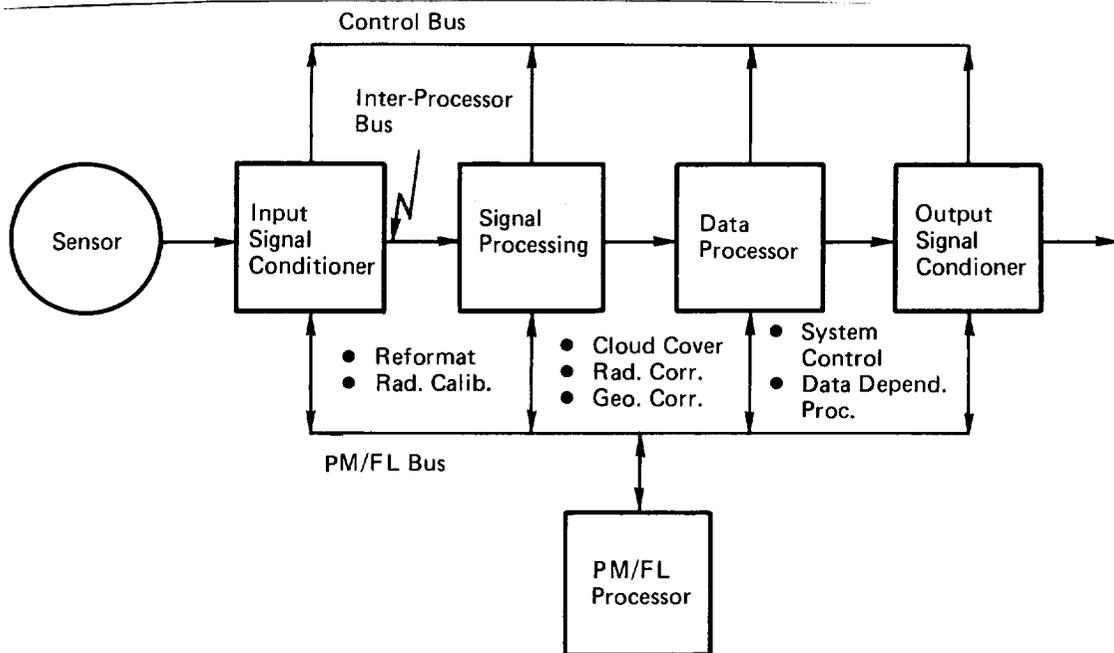


Figure 17. FFP Structural System

The input signal conditioner is applications dependent coupling external inputs such as the following the FFP processors:

- Sensor subsystem inputs
- Telemetry receiver inputs.

The signal processor implements the high speed, periodic processing functions. Examples of signal processor functions include:

- Cloud cover evaluation
- Radiometric correction
- Geometric correction.

The data processor implements the data-dependent processing and provides for the system control.

The PM/FL processor monitors processor status and assesses processor faults. Each processor will perform its own performance monitoring. The PM/FL processor reads status registers of the processing elements over a dedicated PM/FL bus, assesses faults, and issues reconfiguration requests to the system controller (data processor) when reconfiguration is required. In a backup mode, the PM/FL processor (or any other processor) may act as the system controller.

The modularity of the FFP system is supported by the multiple bus structure. All communication between processors is over three independent busses:

- Control
- PM/FL
- Inter-processor.

The control bus transmits system level commands to all processors. The PM/FL bus monitors status registers built into each processor. The inter-processor bus connects the output of one processor to the input of the next processor.

Currently, much interest is focused on distributed processing systems due to their potential for lower system cost coupled with higher fault tolerance when compared with the centralized computer approach. Because of this the remainder of this section will present an example of a distributed signal processor with subsequent sections concentrating on factors affecting the design of a distributed processing system.

Distributed Signal Processor

An example of a distributed signal processor is the Future Signal Processor (FSP) currently under development. The FSP is built around a single operand high speed processing element with a unique data flow and instruction set, which facilitates the implementation of signal processing algorithms. Instructions are pipelined through functional phases of execution. All instructions are defined to require a fixed amount of cycles although they do not make use of all the cycles. Pipelining allows the effective rate to become one instruction per cycle, thus increasing the throughput speed over conventional non-pipelined architectures.

The basic processing element has abundant CPU resources that include numerous registers, multiple stacks, 32 bit ALU, circular buffering, an instruction set tailored for signal processing and numerous user-selectable addressing modes. The instruction set has been tailored to support the implementation of the signal processing arithmetic functions as well as the control of the typical data structures encountered in signal processing applications.

The processing element architecture is tailored for configuring an array of processing elements in an optimum manner for large signal processing applications. Figure 18 illustrates how the basic processing elements can be grouped together to form a cluster of processing elements. Currently a cluster can support up to 16 PE's. One or more functions can be accommodated by a cluster.

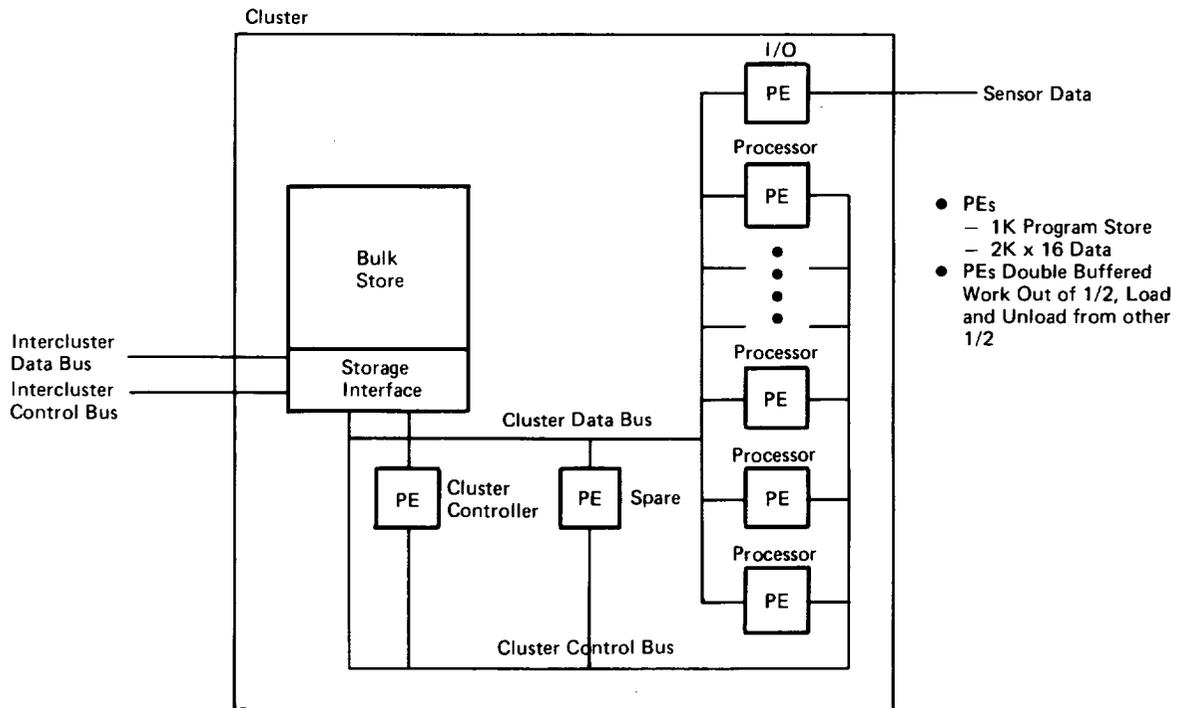


Figure 18. Processing Elements Grouped to Form a Cluster

The cluster is comprised of the following elements:

- Bulk Store (BS)
- Processing Elements (PE)
- Cluster Data Bus
- Cluster Control Bus
- Intercluster Data Bus
- Intercluster Control Bus
- Bulk Storage Interface.

Bulk storage is the main storage element for the cluster. It can be accessed by all the PE's in the cluster via the cluster data bus. The bulk store is used to hold all data which is not actually being processed. Data is moved from BS to the PE's to be processed, then returned to BS to await further processing or outputting.

The PE's are used, in a cluster, to perform several functions: I/O, processing, and control. All PE's in a cluster are considered to be identical. All are assumed to have 2K data store and 1K program store. A cluster is architecturally limited to 16 PE's on the busses.

The I/O PE will move data between its I/O interface (could be sensor) and data store as a series of one word transfers; and between its high speed data store and BS in block transfers. The processing PE's perform the arithmetic processing to accomplish the desired functions. Data is moved from BS to the high speed data store of the PE under control of the cluster controller. The PE then executes one or more arithmetic processes on the data and the results are moved back to BS. The PE will operate in a double buffered mode, i.e., one half of the high speed data store will be used to process one block of data while the other half is unloaded from the previous block and loaded for the next. The cluster controller is the only PE active on the cluster control bus. The controller samples the control bus input lines from the BS. These lines contain the "service request" signals from each of the PE's. When service is required for a PE, the appropriate transfer command is issued by the cluster controller. The controller contains a program describing the required transfers for each PE. A spare PE is shown. This spare can replace either a processing PE or the cluster controller by simply loading the appropriate program.

The cluster data bus connects BS to all of the PE's in the cluster. The cluster control bus is connected to all PE's in a cluster except the I/O PE. The input side of the control bus is used to pass control flags to the cluster controller, while the output side is used by the cluster controller to issue transfer commands to the BS to move data and initiate processing.

The Distributed System Data Bus (DSDB), shown in Figure 16, functionally accommodates both the intercluster data bus and intercluster control bus. It provides the communication path from each cluster to the intercluster controller (shown as the system executive in Figure 16).

The bulk storage interface functions include: address generation, contention, command buffering, and interface protocol. The address generation portion generates the addresses for each data transfer to and from BS. The addressing is characterized by a starting address within BS and the type of transfer, contiguous addressing, addressing by variable step size, or circular addressing. The BS interfaces with both the cluster data bus and the intercluster data bus, thus a contention can exist. A transfer in progress is always allowed to complete. The next allowable transfer is from the intercluster bus if present, otherwise the cluster bus is serviced. One command from each bus can be buffered allowing the immediate initiation of a command when a transfer completes. Control line protocol will be accomplished by the BS interface logic. The BS interface logic forms a conduit between the cluster controller and the selected processing or I/O PE's for the purpose of control of the transfers.

IPL Sequence

The initial program load (IPL) sequence begins with an initial transfer from the system controller to the intercluster controller. The first program is the control program to conduct intercluster communications. The subsequent programs cause the clusters to be IPL'd. The initial programs for all PE's in a cluster are transferred to a cluster BS via the intercluster controller. Once these programs are resident in BS, the system controller is instructed to command that cluster controllers program be loaded from BS to a selected PE. All other PE's are loaded by command from the cluster controller.

Process Flow Description

The I/O (synchronous) interface of the I/O PE is the entry point of the cluster for sensor data. Sensor data is accumulated in the I/O data store. When the input buffer is filled, the I/O switches to the alternate input data buffer and signals the cluster controller. The cluster controller updates and issues transfer commands to the BS interface to cause data to be moved to the appropriate input buffer in BS. The control program in the cluster controller keeps track of the input buffer space in BS to recognize when the input buffers are filled. The cluster controller then initializes the transfers to move the data from BS to the PE. Following the data transfer, the BS interface logic will load a predetermined location in the PE data store causing the program in the PE to begin processing the data. While the PE provides the data in one-half data store, the other half is loaded with the next data to be processed. Following completion of processing, PE issues a "signal" to alert the cluster controller and goes on to process the next data. After a known number of executions through a process, the input buffer for the next process becomes mature. The maturing intervals and sequence is fixed and is programmed into the cluster controller. The larger problems require, for any given process, that several PE's implement identical time lines on different blocks of data.

3.1 IAS Architecture Characteristics

When designing (or evaluating) a multi-processor system architecture capable of accommodating a wide range of applications, the following system architectural characteristics and tools must be considered:

- Interconnection or bussing structure
- Memory configuration and type
- Processing element architecture
- Software structure
- System design tools.

Applications may be represented by three characteristic types:

- Single processor system
- A memory dominated system
- A thrupt dominated system.

How then does one establish the "best" set of architectural characteristics capable of "best" accommodating these three application types?

Certain system features are, of course, desirable regardless of the system architecture selected. These include:

- 1) Modularity - the ultimate goal would be to provide a common system level architecture supported by a modular family of hardware and software system building blocks with common user interfaces.
- 2) High System Reliability/Availability/Fault Tolerance - the systems must have a range of capability offering degraded mode operation; subsystem redundancy; enhanced and standardized test and debug; error recovery and retry; decoupled performance monitoring and fault location.
- 3) Low Cost - it goes without saying that the approach should minimize initial design and life cycle costs.
- 4) Ease of programmability - a high order language (HOL) should exist for describing system functions within systems elements and the data exchange between the elements.
- 5) Extensive system design tools - simulators and performance measuring tools must be available to allow the system

designer to evaluate the loading of each processing element and all interconnecting busses.

- 6) Extensive software - set of executive control and general purpose building blocks to allow a variety of applications to be addressed.
- 7) Small size (LSI) - the extensive use of LSI with its attendant characteristics of small size and low power is a must for on-board implementations.
- 8) Technology Independent - To the greatest extent possible, the architecture must be independent of technology so that newer technologies may be accommodated as they mature and become available.

The system architectural characteristics will be discussed in more detail in the following sections. Table 2 summarizes some of the more important system features for the four architectures considered.

3.1.1 Interconnection or Bussing Structure

One of the most active areas in system architectural studies is the interconnection of processing elements to form systems called "distributed processors". These systems can range in organization from two processing elements sharing a memory to large numbers of relatively independent processors with localized memory for clusters of these processors.

Table 2. Architecture

Feature	Dist Signal Processor	Dist Micro-Processor	Federation of Functional Processors	Custom Designed Computer
Modularity	<ul style="list-style-type: none"> Additional functionality & performance of each increment can be added without major change in hardware or software. Add more PE's. 	<ul style="list-style-type: none"> Low limit on max. number of processors if primary memory shared. 	<ul style="list-style-type: none"> Existing processor or can be upgraded by adding only the modules (mult. accumulator, FFT, etc) req'd. to implement new reqm'ts. 	<ul style="list-style-type: none"> Compatible Family of computers & I/O devices. Capability & cost increments few & costly.
Fault Tolerance Reliability/ Availability	<ul style="list-style-type: none"> Errors are confined to a subset of the system. No one PE has control of the system. 	<ul style="list-style-type: none"> Multiple processors raises the opportunity for detection, diagnosis & recovery thru cooperation among processors but centralized procedure & repair impair effectiveness. Need dynamic reconfiguration capability. 	<ul style="list-style-type: none"> Separate PM/FL processor monitors processing elements & issues reconfiguration requests when req'd. 	<ul style="list-style-type: none"> Achieved by voting either within the processor or among synchronized copies of processor - Expensive.

Table 2. Architecture (Cont.)

Feature	Dist Signal Processor	Dist Micro-Processor	Federation of Functional Processors	Custom Designed Computer
Performance	<ul style="list-style-type: none"> ● Quality & functionality of smaller processor logic more amenable to larger scale int. ● Smaller processors can be designed & implemented more quickly with latest tech. 	<ul style="list-style-type: none"> ● Response time & thruput can be improved by a multiplicity of processors executing concurrently. Have separate jobs to separate processors or multiple processors on one job. Partitioning a problem. 	<ul style="list-style-type: none"> ● Added performance achieved by incremental expansion of any processor. 	<ul style="list-style-type: none"> ● Ceiling imposed by given hardware implementation tech. ● Process multiplexing degrades response time due to overhead. ● Exponential use in software cost as max. perf. of processor is approached.
Software Executive Control	<ul style="list-style-type: none"> ● Decentralized control for task coordination & scheduling, system data base management, I/O data management, interrupt handling <u>not</u> solved for large number of PE's. 	<ul style="list-style-type: none"> ● Same as dist. signal processor. 	<ul style="list-style-type: none"> ● System control resides at data processor. ● Local executive for each functional processor. 	<ul style="list-style-type: none"> ● Centralized control for task coordination & scheduling system data base management, I/O data management, interrupt handling <u>is</u> solved.

Table 2. Architecture (Cont.)

Feature	Dist Signal Processor	Dist Micro-Processor	Federation of Functional Processors	Custom Designed Computer
<p>Software Executive Control</p> <p>Application</p>	<ul style="list-style-type: none"> ● Total system state not visible at one PE. ● Tasks execute in parallel on different hardware. 	<ul style="list-style-type: none"> ● Same as dist. signal processor. 	<ul style="list-style-type: none"> ● No software interaction between processors. ● Processing tasks partitioned to control number of units in 8K max. 	<ul style="list-style-type: none"> ● Total system state visible ● Extensive application routines available.
<p>Programmability</p>	<ul style="list-style-type: none"> ● Functions can be added or expanded without affecting total system timing. 	<ul style="list-style-type: none"> ● Processor may perform a single function thereby simplifying the updating of any processor program. 	<ul style="list-style-type: none"> ● Change only made to program affected ● NO HOL used ● Each processor has own hardware macro controller. 	<ul style="list-style-type: none"> ● Change made to any function or the addition of a function requires the processor program to be recompiled.

The method for identifying the interconnection structure of a system is to isolate the major hardware units involved in transferring information between processes in different processing elements (PEs). This would be a message transmission where a message can be a data block or control request. The path over which the message is transmitted can be a wire or a bus. The message being transmitted can be acted upon by a switching element that affects the destination of the message in some way (e.g. changing its destination address), by routing it to an alternative path, or both.

The first design decision for the interconnection or bussing structure is the transfer strategy; direct or indirect. A direct transfer implies a message is sent directly from a source processing element to a destination processing element. An indirect message transmission involves an intervening operation that alters the message (e.g. address transformation) or routes the message onto an alternate path.

Choosing indirect transmission involves a second decision concerning the switching method; centralized or decentralized. Centralized switching involves a single entity to switch all messages. Decentralized switching involves multiple intervenors.

The third design decision involves the choice of dedicated or shared message transfer path. A dedicated path can be unidirectional point-to-point with no contention; or bidirectional point-to-point with some contention possible. Paths that are shared are bidirectional, visiting more than two points with contention problems. Again, a path does not imply an implementation, and both busses and memories can be paths.

Keeping these design decisions in mind, six system design types will be defined and discussed.

These are:

- 1) Direct Dedicated Loop (DDL)
- 2) Direct Dedicated Complete (DDC)
- 3) Direct Shared Memory (DSM) - Multiprocessor
- 4) Direct Shared Bus (DSB) - Global Bus
- 5) Indirect Centralized Shared (ICS) - Bus with Central Switch
- 6) Indirect Decentralized Shared (IDS) - Bus Window.

The characteristics considered in comparing these designs are:

- Modularity - the ability to make incremental changes in system capability.
- Failure reconfiguration - cost of fault tolerance and the method by which a system is reconfigured to mask faults in processors and intercommunication paths.
- Logical complexity - characteristic affected by the architecture, but its major effect is on software cost.

Direct Dedicated Loop (DDL)

This interconnection structure is illustrated in Figure 19. Loop architectures consist of a number of individual processing elements (PEs), each of which is connected to two neighboring PEs. The loop is unidirectional, one neighbor of a PE is regarded as the source neighbor and the other as the destination neighbor. Messages circulate around the loop from source to destination with intermediate PEs acting as relay or buffer units.

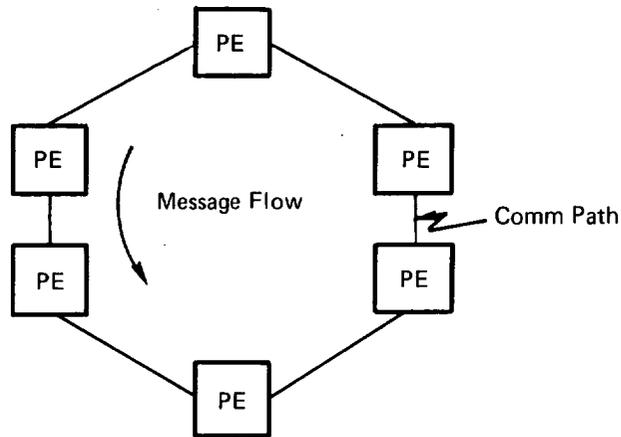


Figure 19. Direct Dedicated Loop (DDL)

Both the cost-modularity and the place-modularity of DDL systems are very good. An additional PE can be inserted anywhere in the loop with the addition of a single communication path, and the flow of messages is not significantly affected by its presence. The failure-effect and failure-reconfiguration characteristics of DDL systems are poor. A single failure in a path or a PE interface causes intercommunication to stop (at least between PEs separated by the failed resource). The logical complexity of communications in a DDL system is low; a PE must only relay messages, originate messages and transmit them to a single destination, recognize messages destined for itself, and strip off messages according to the discipline.

The bandwidth of the single loop is, of course, a potential bottleneck, as communication rates increase. DDL architectures proposed have almost all used bit-serial data links as the communication paths between PEs. This, together with the delay involved in relaying messages, has resulted in significant increases in message transit times around the loop.

The best-known example of a DDL computer system is the Distributed File System at the University of California, Irvine. This system consists of five mini-computers and a number of peripheral devices looped around the Irvine campus.

Direct Dedicated Complete (DDC)

The DDC architecture, shown in Figure 20, is the simplest design type conceptually. Each processor is connected by a dedicated path to every other processor in the system, with messages between processors transferred only on the path connecting them. The source processor must choose the path to the destination processor from the alternative paths available, and all processors must be equipped to handle incoming messages on a multiplicity of paths.

The DDC system has poor cost-modularity. The addition of the n^{th} processor to a DDC system requires not only the addition of $n-1$ paths between it and the other processors, but also, all processors in the system must have facilities for accepting the incremental PE as a data source. Thus, their interfaces must have at least $M-1$ ports, where M is the maximum size of the system.

Modularity of the DDC relative to placement is good, as are failure-effect and failure-reconfiguration characteristics. The DDC architecture

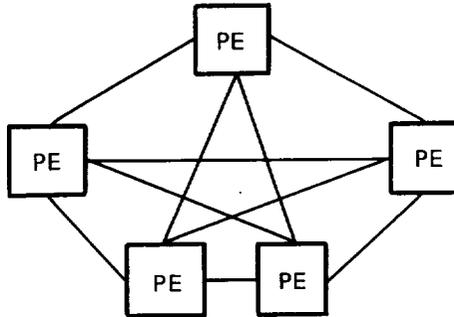


Figure 20. Direct Dedicated Complete (DDC)

is one which can be easily degraded in the event of a failure - a failed processor, or one of the two processors terminating a failed path, can simply be disconnected from the system.

DDC systems have no obvious bottlenecks, and their logical complexity is relatively low. The best-known instance of a localized DDC architecture is a fully connected version of the IBM Attached Support Processor System, in which up to four System/360 or/370 computers may be linked through I/O channel couplers.

Direct Shared Memory (DSM) - Multiprocessor

The most common way to interconnect processing elements is the DSM or multiprocessor architecture shown in Figure 21. Here, two or more

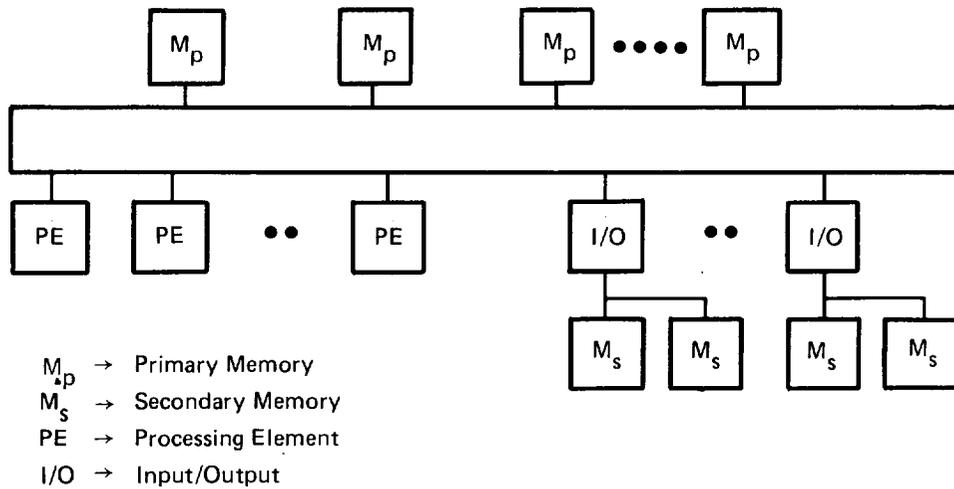


Figure 21. Direct Shared Memory (DSM) - Multiprocessor

processors communicate by leaving messages for one another in a commonly-accessible memory. The key characteristic of the DSM architecture is that the memory is, or can be, used as a path rather than solely as storage.

The modularity of DSM systems is very good; it is possible to add processors arbitrarily, and it is also possible to increase the in-transit message capacity of the path simply by increasing the size of the memory. Cost-wise, it depends almost completely on the path structure by which the processors access the memory system. If each processing element is provided with a direct path, it can be costly since an incremental processor could bring the total to greater than the number

of available memory ports. Alternatively, if the memory is accessed through a single bus with a suitable allocation mechanism, cost-modularity can be very good.

A DSM system is vulnerable to a bottleneck in which the memory bandwidth becomes a restriction on communication rates. Logical complexity of DSM system is quite low. Reconfiguration characteristics of DSM systems are good in the case of processor failures, but poor in the event of failure of the central memory unit or of a shared access bus.

In implementing the multiprocessor sharing, it has been found the systems performance has increased more slowly as the number of processors increased. Systems consisting of more than about four processors have not been cost-effective. The reason for this has been the extreme contention for memory bandwidth when the single memory must serve for all purposes. The most challenging problem for a large number of PEs is the PE/memory switching structure.

An example of a contemporary multiprocessor is the Carnegie-Mellon Cmmmp (Carnegie multi-mini-processor) which allows up to 16 processors to share up to 16 memory modules through a crossbar switch as shown in Figure 22.

Direct Shared Bus (DSB) - Global Bus

The DSB architecture, shown in Figure 23, comprises a number of processing elements interconnected by a common, or global bus. Access to this bus is shared among the processors by some allocation scheme, and messages are sent directly from the source PE onto the bus, to be recognized and accepted by the proper destination(s).

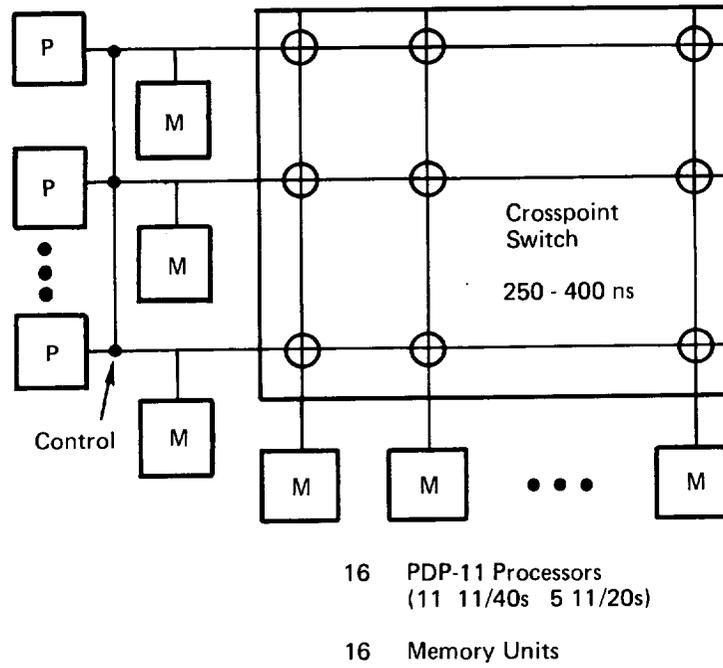


Figure 22. Carnegie-Mellon C.mmp

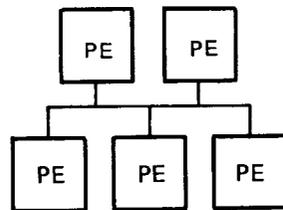


Figure 23. Direct Shared Bus (DSB) - Global Bus

Depending on the choice of bus allocation scheme, it can be possible to add a processor to the system in any position with little or no effect on the other PEs. It is not possible, however, to increase the bandwidth easily as needed, nor is it often possible to increase performance only where needed. Rather, to increase performance it is usually necessary to change the implementation of the entire bus or to replicate the bus.

The DSB architecture requires no overt hardware reconfiguration activity to continue operation for PE failures. Failures of the bus, however, are inevitably catastrophic, and replication is required if the DSB architecture is to be retained. The global bus is, of course, a potential bandwidth bottleneck. Replication of the communications path is the predominant technique for mitigating both the bandwidth restriction and the fault vulnerability of the shared bus.

A pure DSB architecture is typified by one utilizing word-wide busses. An example of a global bus is the bit-parallel word-serial data bus developed by IBM for use inside the AN/BQQ-5(X). This new distributed system data bus (DSDB) has an effective data capacity of 120 million bits per second. The DSDB has been designed to provide a flexible, high-speed data path for moving large blocks of data; it provides a distributed allocation mechanism that does not degrade bus data bandwidth.

Indirect Centralized Shared (ICS) - Bus With Central Switch

The ICS architecture is shown in Figure 24. Here all processors share a path to the central switch. When a PE wishes to transmit a message over the shared bus, it must first acquire the bus, then transmit the message to the switch. From the switch, the message is retransmitted

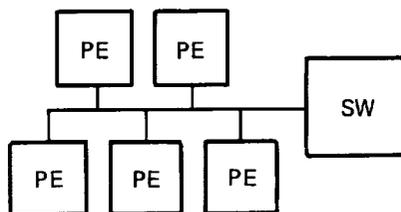


Figure 24. Indirect Centralized Shared (ICS) - Bus With Central Switch

over the same bus to its proper destination. This retransmission is the characteristic by which ICS systems can be distinguished from DSB organizations and from DSM systems using a single bus to memory.

Modularity and failure reconfiguration are good with respect to the PEs, and poor with respect to the central switch and the shared path. Both features can be improved for the central switch if the PEs and central switch are interchangeable.

The existence of the shared path to the switch need not contribute significantly to bottlenecking since it is quite feasible to balance its performance with that of the switch.

Indirect Decentralized Shared (IDS) - Bus Window

The IDS architecture is shown in Figure 25. Here, access to the switching resources is via a path shared by multiple PEs. Switching is performed by more than one resource, and messages may be retransmitted onto the path from which received or onto another path.

The failure-reconfiguration characteristics are poor because multiple PEs and switches can be affected by the failure of a single path. Also, systems of this type are not easily dispersible due to the shared busses.

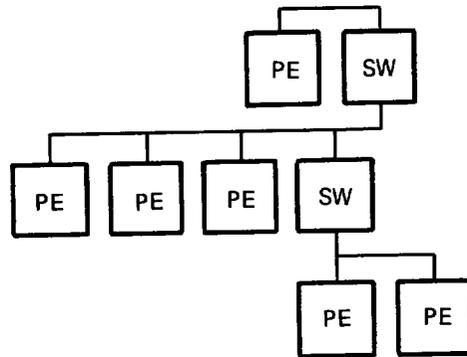


Figure 25. Indirect Decentralized Shared (IDS) - Bus Window

Modularity tends to be good with both processors and paths added as, and where needed. A number of paths may be used to construct hierarchies of processing elements. The logical complexity of this approach grows rapidly as the number of translation levels increased, and as the translation binding becomes more dynamic. This type of interconnection is subject to deadlock.

An example of this architecture is the Carnegie-Mellon Cm*.

Table 3 presents a comparison of these various interconnection approaches.

Table 4 compares four experimental distributed architectures each having unique characteristics. These four architectures include the following:

- Honeywell Experimental Distributed Processor (HXDP)
- Future Signal Processor (FSP)
- Carnegie-Mellon Cm*
- Modular Design using the TI9900 Microprocessor.

The FSP concept has been previously described. The Honeywell Experimental Distributed Processor (HXDP) is a vehicle for research in the science and engineering of processor interconnection, executive control, and user software. A fundamental thesis of the HXDP project is that the benefits and cost-effectiveness of distributed processing systems depend on the judicious use of hardware to control software costs.

Cm* is intended to be a testbed for exploring a number of research questions concerning multiprocessor systems, for example: potential for deadlocks, structure of inter-processor control mechanisms, modularity,

Table 3. Interconnection Structure Comparison

Type	Modularity		Failure Effect	Failure Reconfiguration	Logical Complexity	Bottleneck
	Cost	Place				
DDL	Very Good	Very Good Add PE anywhere in loop with single comm. path	Poor	Poor Need fully redundant path	Low PE relays messages one way	Yes
DDC	Poor Add nth PE requires addl. n-1 paths	Good	Good Just disconnect failed PE	Good	Low	No
DSM	Very Good Mem accessed thru a single bus with suitable allocation mech. up to max. bandwidth	Very Good Add PE's arbitrarily	Good for PE failure Poor for failure of central mem or shared bus	Good for PE failure Poor for mem.	Low	Yes
DSB	Good for PE's Poor for comm. path can't easily increase bw	Good for PE's Poor for comm. path	Very Good for PE's Poor for bus	Very Good for PE's Poor for bus Need redundant bus	High	Potential

Table 3. Interconnection Structure Comparison (Cont.)

Type	Modularity		Failure Effect	Failure Reconfiguration	Logical Complexity	Bottleneck
	Cost	Place				
ICS	Good for PE's Poor for central switch	Good for PE's Poor for central switch	Good for PE's Poor for switch	Good for PE's Poor for switch	Moderate	Potential at switch
IDS	Good PE's need only one or two connections to rest of system	Very Good PE's & comm. paths added as needed	Moderate Multiple PE's & switches can be affected	Moderate	Very High Routing decisions based on knowledge of overall system	Not Likely

Table 4. Distributed Processing System Comparison

Characteristic	Honeywell Exp. Distributed Proc. HXDP	FSP	CM*	TI 9900 Multimicroproc.
Interconnection Structure	Direct Shared Bus	Direct Shared Bus	Indirect Decentralized Shared	Direct Shared Bus
Bus Hierarchy	One level	Two level	Mult. level	One level
Data Flow	Bit serial	16 bit parallel	16 bit parallel	Full duplex serial
Interprocess Comm.	Loose, messages 16 bit logical	Loose, messages	Tight, packet switching	Circuit switching
Data Transfer Rate	1.25 Mbps	128 Mbps	N/A	4800 bps
Addressing	Destination name	Data multiplexing	Virtual destination	Virtual destination
Allocation Scheme	<ul style="list-style-type: none"> o No control lines o Predetermined allocation pattern o Modification very time consuming 	<ul style="list-style-type: none"> o Control lines o Fully distributed bus allocates with variable priority 	Round robin priority scheme	Asynchronous rotating priority technique. Equal priority all users
Shared Memory	No	No	Yes	Yes
Automatic Retry	3 times	On parity error	On parity error	N/A
Processing Elements	Up to 64 PE memory pairs	64 ports/16 PE's each on one bus	10,000	5

reliability, and techniques for decomposing algorithms into parallel cooperating processes.

The final architecture consists of a set of modules based on the Texas Instruments TMS 9900 microprocessor. This design allows the designer to configure a variety of systems including closely coupled multiprocessors and loosely coupled networks. The configuration referenced consists of five processors communicating over an asynchronous bus in order to achieve realtime performance. A feature of the design is bus arbitration hardware that uses an asynchronous rotating priority technique to allocate use of the interprocessor data, address, and control busses.

3.1.2 Memory Configuration and Type

Generally, the most expensive portion of any processing system is the memory. The memory size must be in excess of the requirements set forth by the average job that is anticipated, must be fast to prevent a limiting of the system performance, and must be low power to reduce power supply costs.

Basic semiconductor memory-chip technology involves variations of random-access memory (RAM) and read-only memory (ROM). RAM allows binary data to be written in, and to be read out. New and different programs and data can be loaded and stored in RAM as needed by the processor.

Because information is stored electrically in RAM its contents are lost whenever power goes down or off. When fixed, or unchanging, programs and data are needed by the processor, they are loaded into some form of ROM. In ROM, information is physically (permanently) embedded; therefore, its contents are preserved whenever power is off or momentarily interrupted.

Semiconductor memory chips are normally manufactured using either bipolar or metal-oxide semiconductor (MOS) technologies. Bipolar and MOS memories implement bipolar transistor and MOS field-effect transistor (MOSFET) arrangements, respectively, to store addressable sequences of binary 1's and 0's. MOS memories are either static or dynamic. Static memory depends on a dc level for operation; it is easier to implement in many cases, but requires more power. Dynamic memory requires clock signals and special refresh circuitry for operation; thus more external circuitry may be needed. However, chip size and thus cost is reduced as is power dissipation.

Major types of read-only memory (ROM) are: basic mask programmed ROM; electrically programmable, ultraviolet erasable (EPROM); electrically alterable (EAROM); electrically erasable (EEROM); and the field programmable (p/ROM). EPROM is electrically programmable, then erasable by ultraviolet (UV) light, and programmable again. The EEROM can be erased (all bits) by electrically pulsing the device. The EAROM utilizes special processing techniques that allow bit locations to be reprogrammed at any time. Both EEROM & EAROM are used mostly in specialized applications where nonvolatility and electrical erasability are requirements.

A variety of technologies exist for memory devices. These include, in addition to standard NMOS and bipolar, I^2L , CMOS, DMOS, VMOS, SOS, and a number of CCD structures. The overwhelming bulk of users utilize NMOS silicon gate or some variant thereon (such as two level poly).

The memory requirements normally falls into four types:

- Working
- Control store

- Microstore
- Bulk store.

The trend today is toward the "processor on-a-chip" concept with many designs being microprogrammable. The working memory (or local memory) tends to be a 1K to 2K (16 or 32 bit words) static RAM containing the data (i.e. sensor data) to be processed by the processing element. The control store is 1K/4K/8K static RAM generally for defining the program to be executed. The program includes the functions to be performed, the control of the data between the working store and the logic unit plus the control of all I/O. The microstore is the ROM/PROM/EPROM used to execute the control store program for those logic devices that are microprogrammable. For those logic devices having a combinatorial logic internal structure rather than microprogrammable, the program to be executed is contained in the control store.

Applications requiring multiple processors to share the total processing load will have another level of memory (storage) **introduced-program** memory. This will provide the global control for all processors with the control store local to each processor (could be the same in each processor) and providing control for a single processor.

Bulk storage is required for those applications where direct communication links are not always available and data must not be lost or where the processing is data dependent (i.e. data is reformatted, radiometric calibration coefficients computed, etc.).

Bulk memory could be magnetic bubble, charge-coupled devices (CCD), magnetic disk, magnetic tape, or dynamic RAM depending on the application requirements. Volatility and speed are the major differences between bubble and charge-coupled devices. CCD memories offer speeds

up to several megahertz compared with the several hundred kilohertz typical of bubble-chip data rates. But CCD's do not hold their stored information when power supplies are removed (volatile), thus requiring use of a standby power source. Also, CCD memories require continual refresh to maintain data. The bubble mass memory has the advantage of non-volatility, ease of fabrication (hence low cost), high radiation resistance, and high reliability. Power requirements are also lower for bubbles than for CCD's. Magnetic recording devices (tape and disk) have been the major medium for large-scale digital data storage for some time. In addition to low cost per bit, this technology offers immediate playback and repeatable use. The characteristics of these bulk memory devices are shown in Table 5.

Table 5. Projected (1990) Mass Memory Technologies

Device	Capacity (Bits)	Access Time	Transfer Rate (Mbps)	Cost (Cents/Bit)
*CCD	10^9	0.5 ms	6	0.001
Bubble	10^9	2 ms	1.5	0.003
*Dynamic RAM	10^6 /Chip	50 ns	20	.005
Mag Tape	10^{10}	50 sec	2.5	5×10^{-5}
Mag Disk	10^{10}	10 ms	15	0.005

*Volatile - Requires Standby Power for Refreshing

3.1.3 Processing Element Architecture

The performance of a Distributed Processing System is determined primarily by the number of processing elements used for a specific application. The processing element architecture should be applicable to a wide breadth of programs via a single processing element for those programs

involving a lower performance boundary to a multiple application of the processing element as program performance/thruput and functional requirements vary. The software support for the PE should strive to provide ease of programmability and be efficient on a wide variety of algorithms. LSI implementation will optimize the density, power and performance of the PE by tailoring the design and physical layout of the individual circuits specific to the requirements of the PE architecture and logic design.

What, then, should be the features of the PE to accomplish these goals? The desire is to always get as much performance as possible out of the smallest amount of real estate. There is, however, always an upper limit to the complexity of the chip design. The more complex designs for a given instruction set may require a larger die than is economical to make today. These larger dice may result in an unacceptably low yield. Also, as more digital circuits are added, there is ultimately a point of diminishing returns in performance. Up to this point of diminishing returns, circuits can be found that speed up the entire instruction set. Then more chip area is required to speed up an increasingly smaller number of instructions. An example might be the amount of real estate required to speed up the MULTIPLY instructions with hard-wired logic. If the majority of applications did not require MULTIPLY operations, this would not be a cost-effective tradeoff.

The IAS functions do require many MULTIPLY operations, thus the addition of a high speed multiplier would be beneficial even at the expense of more real estate.

Features to be considered when comparing or evaluating PE design include:

- Register set
- Memory interface

- Internal structure.
 - Combinatorial logic
 - Microprogrammed.
- 4,8,16, etc. bit arithmetic
- 32 bit accumulate
- 8,16,32 bit logicals
- Instruction set
- Hardware multiplier
- Chip characteristics.
 - Technology
 - Add time
 - Power.
- Pipeline instructions
- Performance.

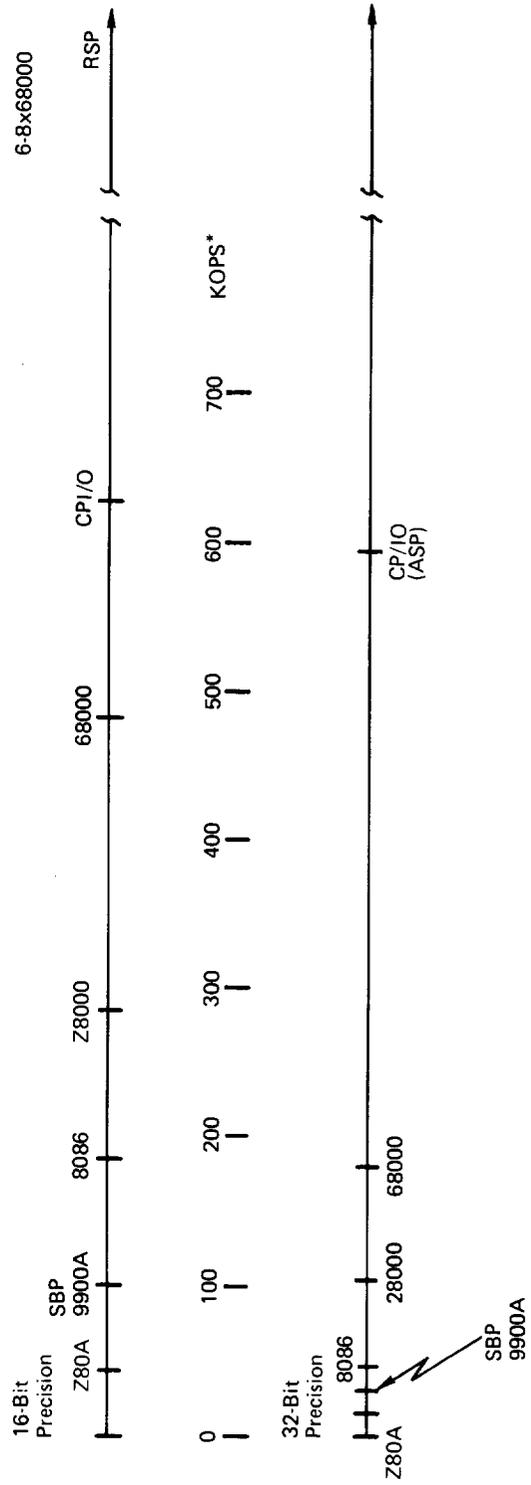
A sufficient number of registers is desirable to allow as many register-to-register operations as possible. Eight to sixteen 16 bit general purpose registers are standard. The memory interface could be a multiplexed address/data bus or a separate bus for data and address. The internal structure could be microprogrammable allowing some tailoring of the basic instruction set depending on the application. The arithmetic, accumulate, and logical bit lengths should be compatible with the application requirements and incremented to provide efficiency. The instruction set should be applicable across the range of applications, preferably tailored to support the implementation of the IAS signal processing arithmetic functions as well as the control of the typical data structures encountered. A multiply capability is a must for the IAS functions. This capability can be part of the basic instruction set or an off-chip hardware multiplier could be provided. Many standard microprocessors do not contain a multiply capability. The thruput of the microprocessor when implementing the IAS functions would be greatly enhanced by the addition of a hardware multiplier. The PE design should be

technology transparent to allow the graceful integration of new technologies as they mature. Pipelining instructions is a form of parallel processing in which portions of each instruction executed are overlapped. Suppose the execution of each instruction required four fundamental operations of one clock cycle. Instructions could be executed by loading them into the front end of a circuit "pipe" consisting of four stages. The instruction would pause at each of the stages for one clock cycle before passing to the next. In this way the front end could be loaded with a new instruction each clock period rather than every third clock period. Pipelines are expensive, typically requiring more digital circuitry than that required to execute instructions individually. Jumps and breaks in the sequential flow of instructions causes the pipe to momentarily lapse to that of a machine that executes only one instruction at a time.

Figure 26 illustrates the performance, in thousand operations per second (KOPS), for seven processing elements. These seven include:

- Z80A
- SBP 9900A
- Intel 8086
- Z 8000
- Motorola 68000
- Advanced Signal Processor (ASP) CP/IO
- Research Signal Processor (RSP).

The RSP is currently being developed for use in the 1981 time frame. The RSP is being designed specifically for signal processing applications and is one of the reasons the indicated performance is much greater (for 16 bit precision) than for the first five microprocessors shown. The addition of a MULTIPLY capability to these microprocessors would greatly increase their capability relative to signal processing applications.



* Based on a 94% Add Type to 6% Multiply Instruction Mix with Operand in a Register and the Other Operand in Main Storage.

Figure 26. Processing Element Performance Comparison

3.1.4 Software Structure

The total system software consists of two parts: Executive and Applications software. Executive software is a system overhead function; the time and memory resources used for Executive Control do not contribute directly to the applications processing requirements.

Currently there are no standards for Executive programs for Distributed Processing Systems and no body of expertise that can be used to estimate or evaluate their memory or time requirements. There is a great deal of experience related to multitasking executives for centralized computer systems developed over the last 20 years, however, this experience cannot be applied directly to distributed systems.

Executive software functions include the following:

- Task scheduling and dispatching
- Interrupt handling
- I/O data management
- System initialization and exceptional condition handling
- Self test
- Problem recovery for detected failures
- PE-PE communications
- Cluster-to-cluster communications.

The task scheduling strategy must guarantee that the processing of each iteration of a task will be completed within the allotted time or time frame of that iteration. It does not matter when the processing is performed within the time frame, only that it be completed by the end of the time frame. The time needed to execute an iteration of a task is predictable, however the iteration rates required by different tasks will differ.

Scheduling strategies could include:

- Fixed preplanned (nonpreemptive) scheduling
- Priority scheduling (GP system)
- Deadline scheduling
- Simply periodic scheduling.

The fixed preplanned scheduling, while simple to implement, does not allow any flexibility in scheduling. Each iteration of a task is run to completion.

The priority scheduling strategy, commonly used in general purpose systems, assigns the highest priorities to the tasks with the fastest iteration rates.

The deadline scheduling strategy always runs the task whose deadline is closest. The scheduling overhead time for short tasks is large in comparison to executive time negating the advantages of this strategy.

The simply periodic strategy is similar to the priority scheduling strategy, but the iteration rates of the tasks are constrained such that each iteration rate is an integral multiple of the next smaller rate.

The executive software consists of both global and local executive control and is structured into three parts:

- The global executive task
- The local executive
- The local-global communication.

Global Executive

The global executive is responsible for the execution of all application tasks and implements the overall system error detection and reconfiguration mechanisms. The global executive provides the inputs to a cluster for a given task or function and then receives the output values from each cluster iteration.

Local Executive

The local executive is a collection of routines to perform the following functions:

- 1) Run each task allocated to it at the tasks specified iteration rate
- 2) Provide input values to, and receive output values from each task iteration
- 3) Report errors to the local executive task.

A scheduler routine is responsible for scheduling the execution of a task or tasks. Each task can be run at a prespecified iteration rate that defines a sequence of time frames within which the task must be run. The scheduler is invoked by a clock interrupt or by the completion of a task. It always runs the highest priority task allocated to a processing element that has not yet finished executing the iteration for its current time frame. Execution of a task may be interrupted by the clock, in which case its state is preserved until execution is resumed—possibly after the execution of a higher priority task. A task

that has completed its current iteration is not executed again until after the start of its next time frame.

In general, a processing element will operate on a number of units of data, performing the same task or tasks for each unit of data (sensor channel). The processing element will operate in a double buffered mode. In any one time frame, one of the buffers is available for new data being generated by the task (output data) while the other contains the data generated the last time frame. The output values then become input to other tasks. At the start of the next time frame, the buffers are switched in a ping-pong fashion.

When partitioning a problem, several PEs may be required to accomplish a given task or tasks for the total units of work within a given time frame. Program storage for all tasks can reside in the program store of each PE or program store overlay can occur to conserve total program store. This loading of program store for each task results in some additional executive overhead, however if many units of work are operated on by each task this additional overhead is generally negligible. This loading of the program store is just another task to be scheduled in the time line program.

As previously mentioned, a "cluster" of PE's may be required to accomplish a task in a given time frame. This represents a second level of local control, that of controlling the proper input values to and output values from a cluster of PE's. Clusters will generally contain 1 or more redundant PE's. Thus, the cluster controller must be able to detect a bad PE and switch in the redundant PE.

Local-Global Communications

Communication among the clusters of PEs of a distributed processing system is generally accomplished over a global bus. Tasks are performed sequentially in a pipeline fashion. If an individual cluster controller wants to initiate a transfer to another cluster via the global bus, the following sequence would occur: the cluster controller issues the command to initiate an external transfer. The global bus will recognize the request and depending on the bus allocation mechanism suspend or acknowledge the request. If acknowledged, the transfer can be completed.

The cluster controller will initiate an external transfer when the contents of a buffer have been filled signifying the completion of a task by the cluster.

Application Software

The application software is structured as a set of iterative tasks each task described by one or more algorithms. Each task is run with a fixed iteration rate which depends upon its priority. The iteration rate of a higher priority task is an integral multiple of the iteration rate of any lower priority task.

The fact that a task is executed by several processors is transparent to the application software. In each iteration, an application task obtains its inputs by executing calls to the local executive software. After completing its outputs, the results of each processing element are combined in buffer storage, and made available as input to another task.

The application software will be tailored to support the implementation of signal processing arithmetic functions as well as the control of the typical data structures encountered in signal processing applications. Functions are written in a high order language (HOL) to aid the application software designer and provide efficient, verified code. The verified function or macro will provide a building block basis from which many algorithms can be developed. A library of these macros will be developed consisting of: a brief functional description of the process, a mathematical description, and sizing information including number of instructions required and execution times in machine cycles based on a per input word or per output word basis.

3.1.5 System Design Procedure

The design of a distributed processing system is an iterative process in the beginning eventually culminating in the compiled output code. As shown in Figure 27, a definition of the system requirements is essential. The System Engineer will translate the requirements into a functional flow defining the functions to be performed, the input data rates, output data rates, and available processing time. The System Engineer and System Programmer, aided by a description of the distributed processing architecture will translate the requirements into system configuration complete with processor and bus loading. This process is iterated until a configuration results capable of meeting the system requirements plus a margin to account for any modifications that may be made to the system at a later time. The selected system configuration results in a complete system data flow and algorithm specification defining the functions, all interfaces and a complete execution time line. At this point the application programmer converts the algorithm specifications into coded programs. The final output becomes the compiled application programs for the processing elements along with the compiled programs for the distributed control.

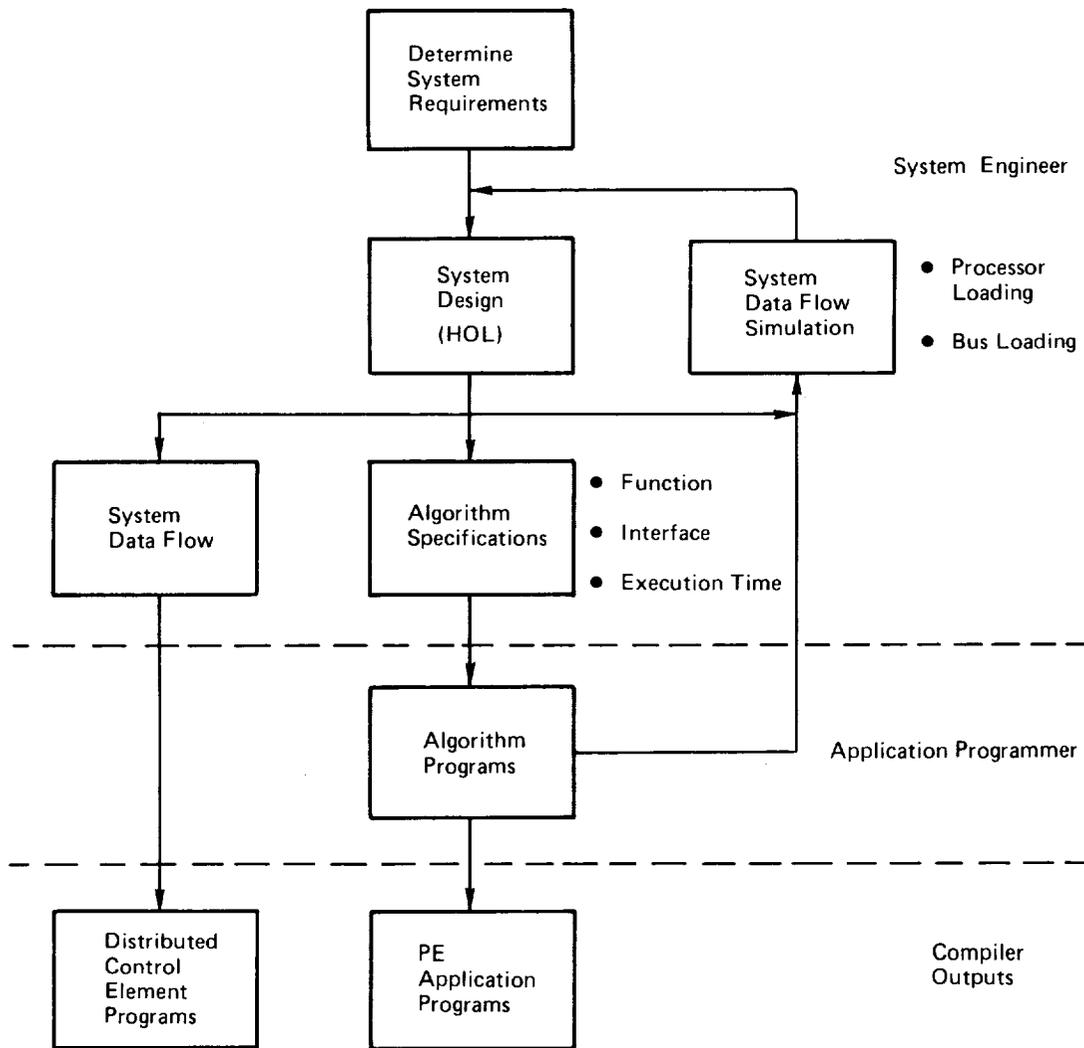


Figure 27. System Design Procedure

Key to the system design is the system design tools available to translate system requirements into compiler outputs in an orderly and timely manner.

System Design Tools

While a wealth of experience exists for the design of single CPU systems, new system analyses and design verification and validation tools are required to reduce distributed processing to practice. These system tools include the following:

- System Description
- System Analysis/Evaluation
- Executive Control
- Software Design Verification and Validation
- Hardware/Software Integration
- Self Test
- Reliability/Availability Enhancement Techniques.

The System Description is a user oriented description of the Distributed Processing architecture which is defined without reference to any specific processing application. Included within this System Description is:

- Communications architecture
 - Task-To-Task
 - Task-To-Processing Element (PE)
 - PE-To-PE.
- PE architecture
- Bus or memory protocol connecting PEs & clusters of PE's.

The System Analysis/Evaluation provides a method of insuring that the loading on each processing element and the interconnecting bus will be within acceptable limits. The problem is first decomposed into a series of tasks. A good decomposition is often specific to a problem and the processing resources being considered. A problem can be partitioned horizontally (pipeline processing), vertically, or a combination of the two. This will be described in more detail in later paragraphs. Having decided upon an initial approach to the problem decomposition, the PE loading (for as many as 1000 processors) and the total bus loading will be established. The partitioning task can be automated or performed manually, but in either case the partitioning is followed by the performance modeling. Manual performance modeling is not satisfactory for most applications of interest because of the many PEs involved.

As mentioned previously, there are no standards for executive control programs for Distributed Processing Systems and no body of expertise that can be used to estimate or evaluate their memory and time requirements. Executive overhead is affected by the number and type of task as well as the architecture and performance of the processing element. There is a requirement for a method of controlling the operation and interaction of the processing elements that does not require permanent Master-Slave relationships among the processing elements.

The Software Design Verification and Validation involves a method of developing the software for the individual processing elements that will insure that the overall system function is performed. This involves the use of a software design language, system software simulator, and performance measuring. The software design language (an executable high order language) is used by system engineers and software engineers to describe the functions within partitions and the data exchange

between partitions. The system software simulator is used to verify & validate both the HOL description of partitions and the operational software for partitions. Performance measuring is used to insure that the software and hardware partitions are consistent and satisfy the system time line requirement. The loading on each PE and the total bus loading is established based on a mathematically rigorous model of bus traffic dispatching and PE algorithms.

Successful Hardware/Software Integration requires external test equipment and built in test facilities to provide sufficient visibility into the internal operation of a distributed system during checkout.

Self Test programs and/or built in test equipment will establish the operational readiness status of the system to a very high level of confidence.

Fault tolerance techniques can be developed for Distributed Processing Systems to provide a high level of system availability by exploiting the parallelism inherent in Distributed Systems. Reliability/Availability Enhancement Techniques provide this capability.

Partitioning

One of the major problems confronted by the design engineer when developing a Distributed Processing System concept is that of problem decomposition or partitioning. How should the required functions for a particular application be spread across many individual processing elements?

Consider an application made up of four functions and executed for many elements—sensor channels for example. The functions are executed on a regular basis, once every t seconds, each function being executed for N units of processing (N pixels) as shown in Figure 28. Functions considered are:

- Reformatting
- Radiometric correction
- Geometric correction
- Classification.

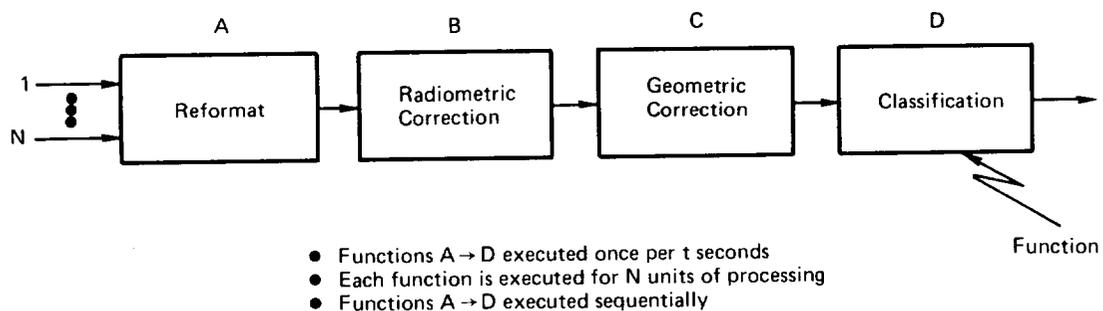


Figure 28. Partitioning Rationale

The data can be partitioned vertically or horizontally. Vertical partitioning, shown in Figure 29, accommodates all four functions in each processor (four are shown) each time period for $N/4$ of the N units of processing. Horizontal partitioning, shown in Figure 30, dedicates a processor to each function and processes all N units in t seconds. Here each processor operates on data from a different time period with buffers generally provided between each function. For functions that vary in complexity, it is difficult to balance the load between functions. Function D may require four processors to accommodate the N units if processing in t seconds.

This then leads us to a combination of the two approaches. Take, as an example, the microprocessor organization for implementing classification using the Maximum Likelihood algorithm. An initial approach to the implementation would be to divide the algorithm into four parts

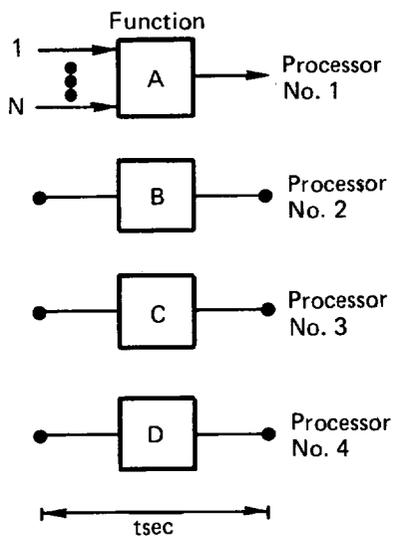


Figure 29. Horizontal Partitioning or Pipeline Processing

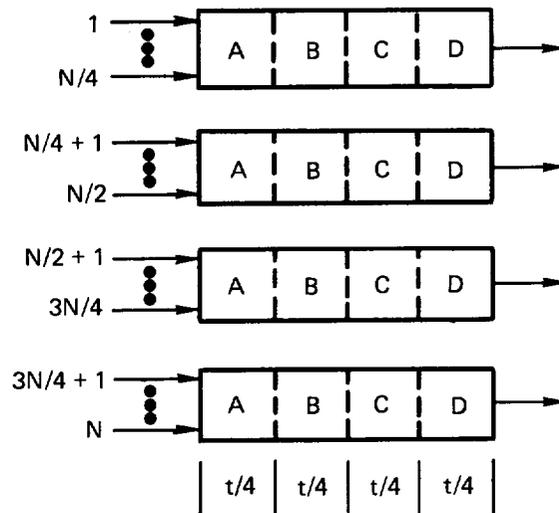


Figure 30. Vertical Partitioning or Parallel Processing

as shown in Figure 31. Using a ROM for multiplication and assuming a 100 nsec Add time produces the configuration shown in Figure 32 for an imaging sensor with 7 spectral bands, a resolution of 40 meters, and 12 classes. Nine of these configurations would be paralleled to accommodate the total input rate for the sensor of 31.6×10^6 bps.

The choice of the problem decomposition is critical; different decompositions often induce different requirements on available resources.

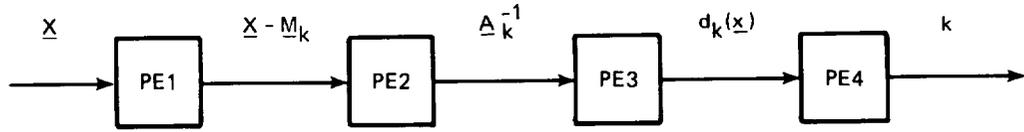
3.2 Conclusions

Four architectural approaches capable of implementing the IAS functions were proposed and discussed. These include:

- Custom Designed Computer
- Federation of Functional Processors (FFP)
- Distributed Microprocessor System
- Distributed Signal Processor System.

The last three approaches all represent a distributed architecture with the FFP having a high level of logic integration at the processor level. Much interest is currently focused on the last two distributed approaches due to a potential for lower overall system cost coupled with high system fault tolerance.

The distributed signal processor system best suits the IAS high throughput requirements. Key to the design of any distributed architecture is the interconnection or bussing structure and the available system design tools necessary for properly partitioning the required system functions among several processing elements. The Future Signal Processor (FSP) is representative of the system architecture best suited to IAS requirements.



$$\Delta_k^{-1} = (\underline{X} - \underline{M}_k)^T C_k^{-1}$$

$$d_k(\underline{X}) = C_k - 1/2 \Delta_k^{-1} (\underline{X} - \underline{M}_k)$$

Figure 31. Multiprocessor Organization for the Maximum Likelihood Algorithm

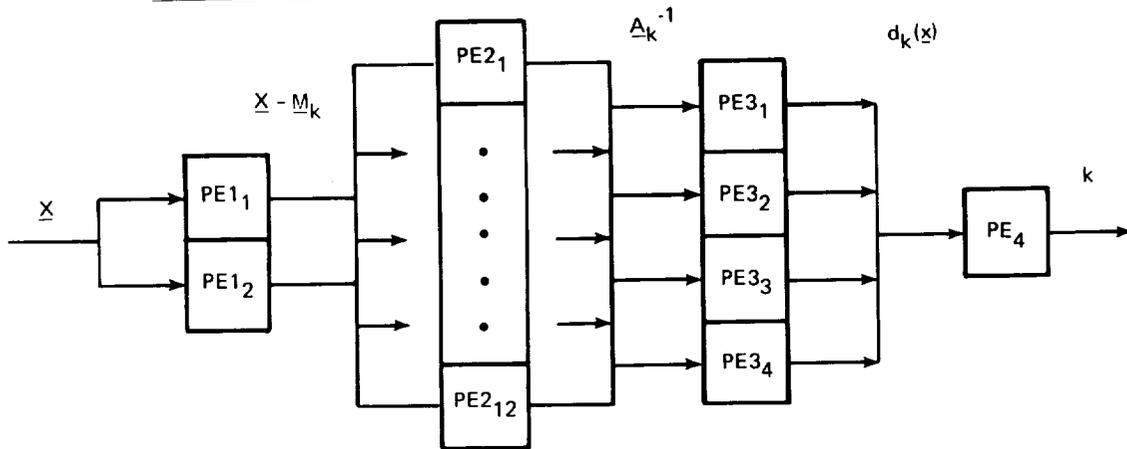


Figure 32. Parallel/Pipeline Implementation of Maximum Likelihood Algorithm

4.0 TASK 3 - DATA SET SELECTION CRITERIA

Presently, a large amount of unusable data or data not of interest to a user, are transmitted, processed, and stored because most spacecraft do not have on-board data set selection capability. The purpose of this task is to define a data set selection criteria to be used to determine whether data are of value and warrant further processing (example- cloud cover).

The Information Adaptive System (IAS) is the NEEDS element that will implement this data set selection capability. As shown in Figure 33, IAS is only one element of the total NEEDS data flow. Other elements of NEEDS include:

- Modular Data System (MDS)
- Archival Data Storage (ADS)
- Massively Parallel Processor (MPP)
- Data Base Management (DBM).

The IAS and the MDS will be on-board the spacecraft with the IAS interfacing directly with the spacecraft sensors. The greatest gain, therefore, in terms of the transmission of useable data can be obtained by the proper design of the IAS. The IAS will allow a user or users to adapt the characteristics of the spacecraft sensors to the observed data based upon the user's knowledge of what is occurring. There are some scientific missions, where the useless data are quite evident.

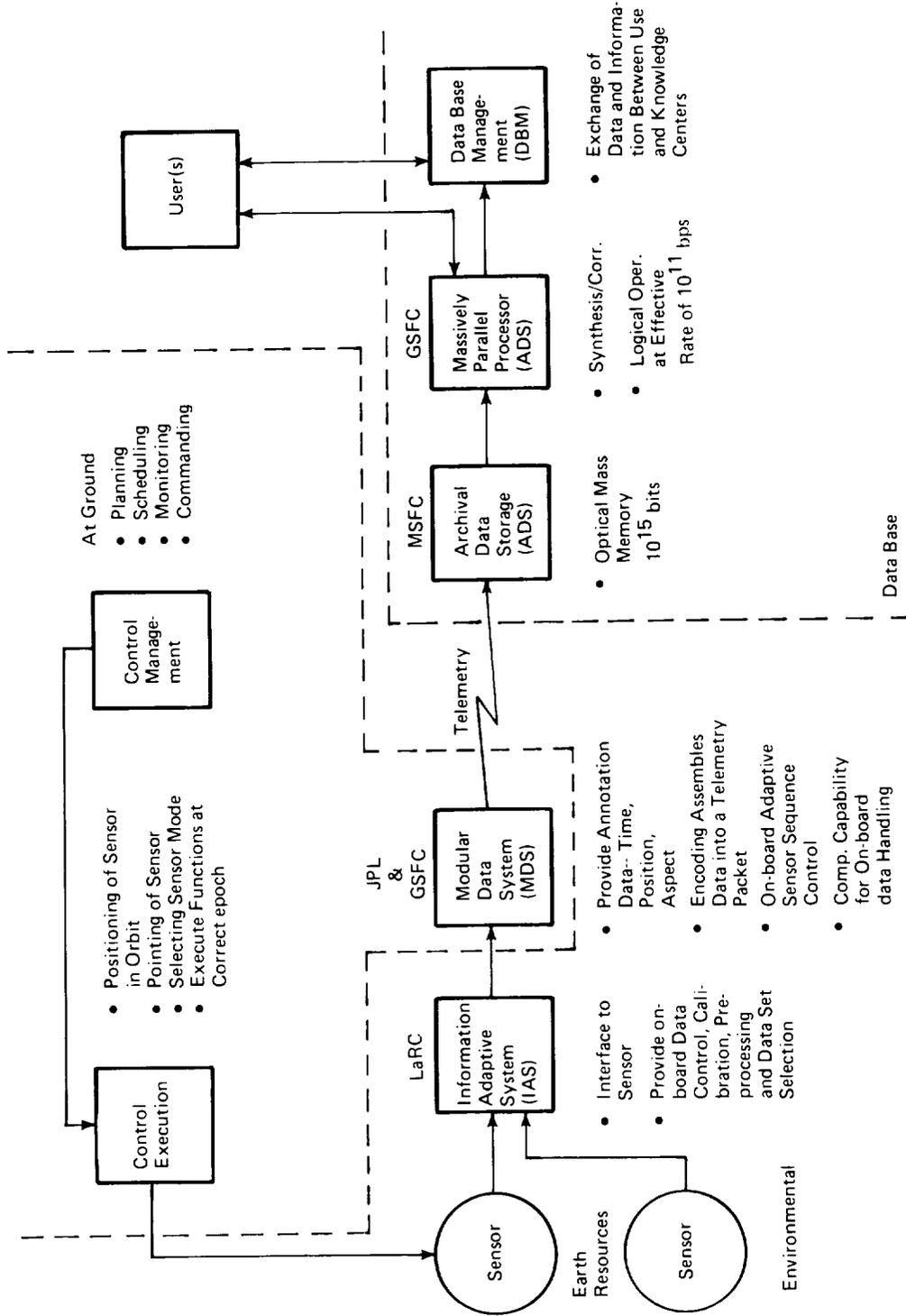


Figure 33. NEEDS Data Flow

For example, an instrument that is prepared to observe a solar burst with great resolution could discard all data until the burst occurred and then save the desired amount of data before, during, and following the event. Another example would be the detection of forest fires. Forested areas, especially remote areas, could be monitored using the output of IR sensors. The IR data could be thresholded on-board. If a threshold is exceeded, indicating a significant change in background temperature, a forest fire is indicated. Data would be transmitted only when a threshold had been exceeded thereby reducing the downlinked data load. In addition to pinpointing fires at a very early stage, surveillance could be maintained automatically over large fires so that hot spots, perimeters, new flare-ups and progress of fire-fighting efforts can be determined. A third example would be the detection of ocean oil spills and well blowouts. The location of known well and shipping lanes could be constantly monitored with data transmitted only if a spill were detected. Once detected, the progress of the spill could be monitored to aid in any cleanup operations.

The overall IAS data flow, shown in Figure 34, consists of:

- Sensor data preprocessing
- Information extraction
- Data prioritization
- Adaptive system control.

Sensor data preprocessing refers to the data processing tasks applied to the sensor data that correct or eliminate radiometric and geometric distortions, transform the data to a standard format, and provide the data set selection criteria.

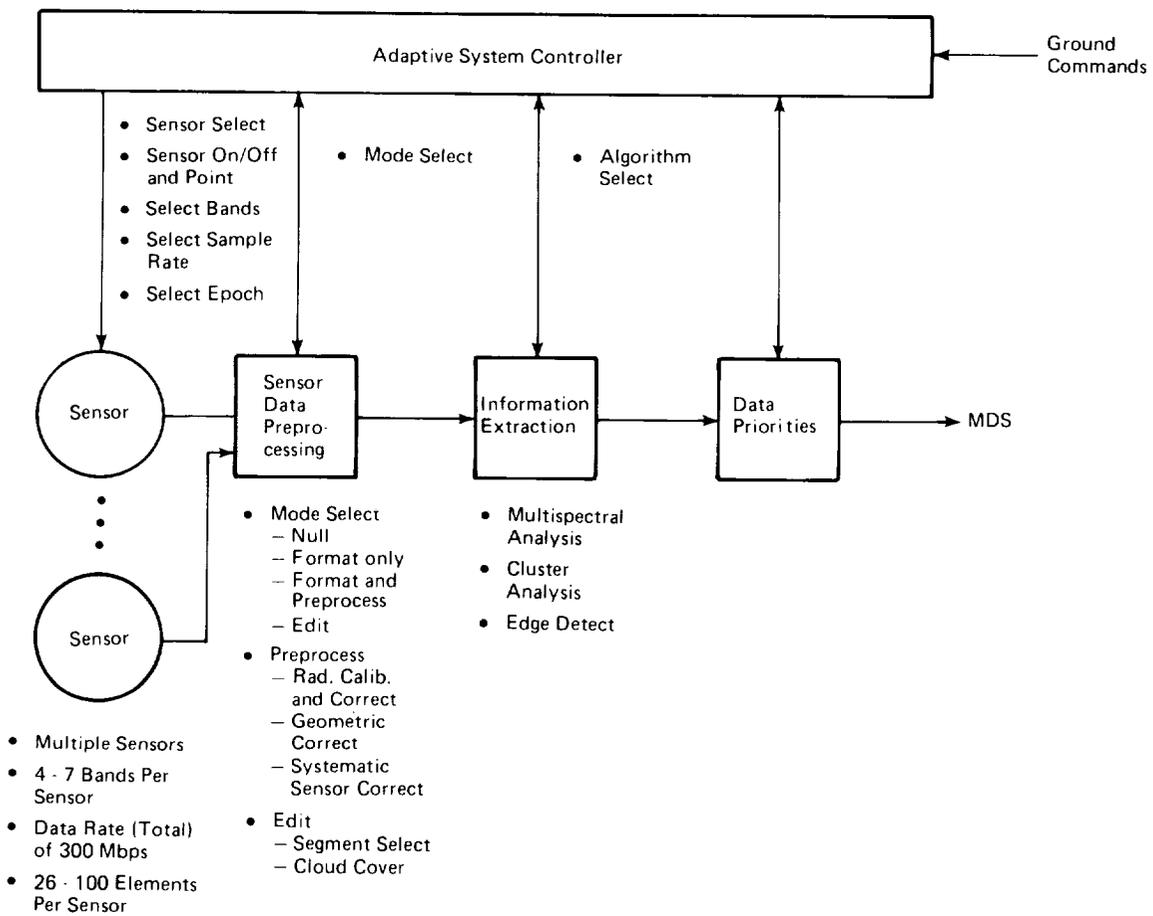


Figure 34. Information Adaptive System Data Flow

Information extraction transforms the preprocessed data of interest (user dependent) into information products relative to a given application. Examples would be the application of multispectral analysis, cluster analysis, or edge detection information extraction algorithms. The data prioritization will establish the sequence of sensor data or information sent to the MDS for packetization will establish the sequence of sensor data or information sent to the MDS for packetization. For example, BIP formatted, visible sensor data may be routinely transmitted to a ground station. However, if a potential forest fire or oil spill is detected, this information would have priority over the visible data. Adaptive system control of mission and sensor operations is any technique used for control which is dependent on the information content of the sensor data. Examples of control functions could include:

- Sensor selection
- Sensor on/off and pointing
- Band selection
- Sample rate selection
- Epoch selection
- Sensor data preprocessing mode selection
- Information extraction algorithm selection.

4.1 Sensor Data Preprocessing

The greatest gain in terms of data reduction will occur by implementing the sensor data preprocessing functions on-board the spacecraft. Since extracted information is a very low data volume compared to the raw sensor data, transmission problems are greatly reduced. In addition, several users often require the same preprocessing operations prior to information extraction. Performing these operations on-board would allow the same data to be broadcast directly to several users, thus eliminating redundant processing at each of the user receiving sites.

Sensor data preprocessing functions generally include the following:

- Data reformatting
- Radiometric calibration and correction
- Create resampling grids for registration
- Data set selection.

Figure 35 illustrates, in functional form, the data flow of the IAS with the sensor data preprocessing functions highlighted. The IAS may be required to process the output from several sensors. The sensors will be imaging and environmental. A user can request, for his particular application, the output from more than one sensor. Figure 35 indicates the preprocessing functions for two sensors, sensor 1 and sensor n. Sensor 1 data is first reformatted and spatially registered into a band-interleaved-by-pixel format. The reformatted data is then stored in a suitable mass storage device, disk, bubble memory, or perhaps tape. Radiometric calibration data is

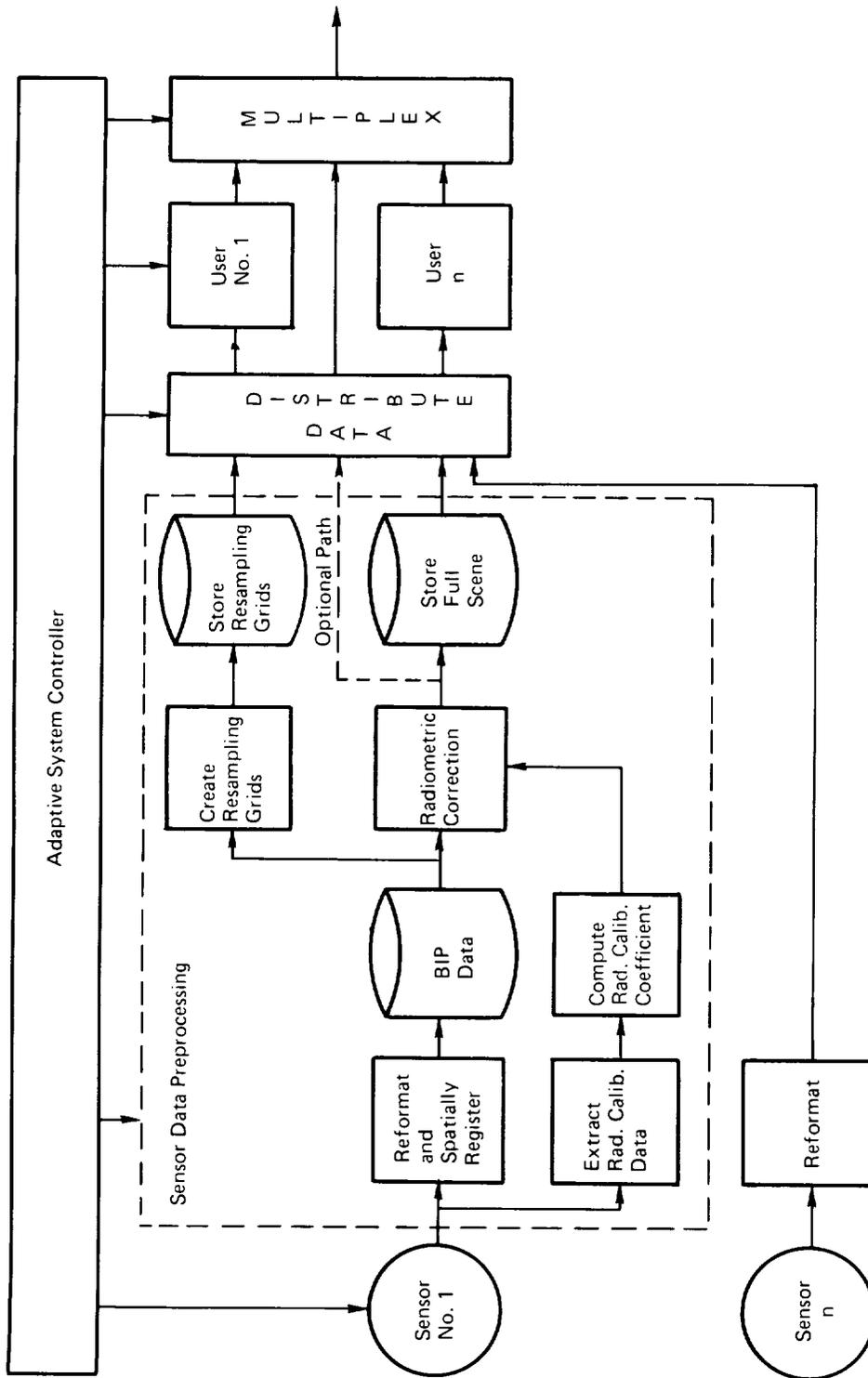


Figure 35. Sensor Data Preprocessing for IAS

accumulated and processed to generate radiometric calibration coefficients. The data is then radiometrically corrected and a full scene stored, conceivably using the same storage but most likely a separate storage. Storing the full scene data allows a user to access any portion of the scene or to go back and access other regions based on ground command. It is expected that an operational system will have changing requirements for the location of an area of interest. Thus, some provision must be made for updating the commands that identify to the on-board processor the locations of the samples to be extracted. If a user does not foresee any change in his region of interest, the radiometrically corrected data could be directly distributed to the user. The resampling grids necessary for mapping the sensor data to a standard projection will be computed and stored for distribution to those users requiring geometrically corrected data.

These preprocessing functions would be performed once with each user then applying his own information extraction algorithm to his area of interest. This avoids duplicating the functions by each user.

If, on the other hand, only a small region of a scene is to be processed by the users, it would be more efficient, both from a computational and storage standpoint, as shown in Figure 36, not to radiometrically correct the data prior to data distribution. Here, the individual user performs the radiometric correction and if registration of the sensor data is also desired, the two operations can be combined into one, gaining even more computational efficiency.

The question to be addressed is: how much of the data acquired by the sensor or sensors will actually be used by the users? If there are no multiple users for the same geographical region of a scene, if the regions of interest never change, then each user should radiometrically correct his data. If the converse is true, radiometric correction is

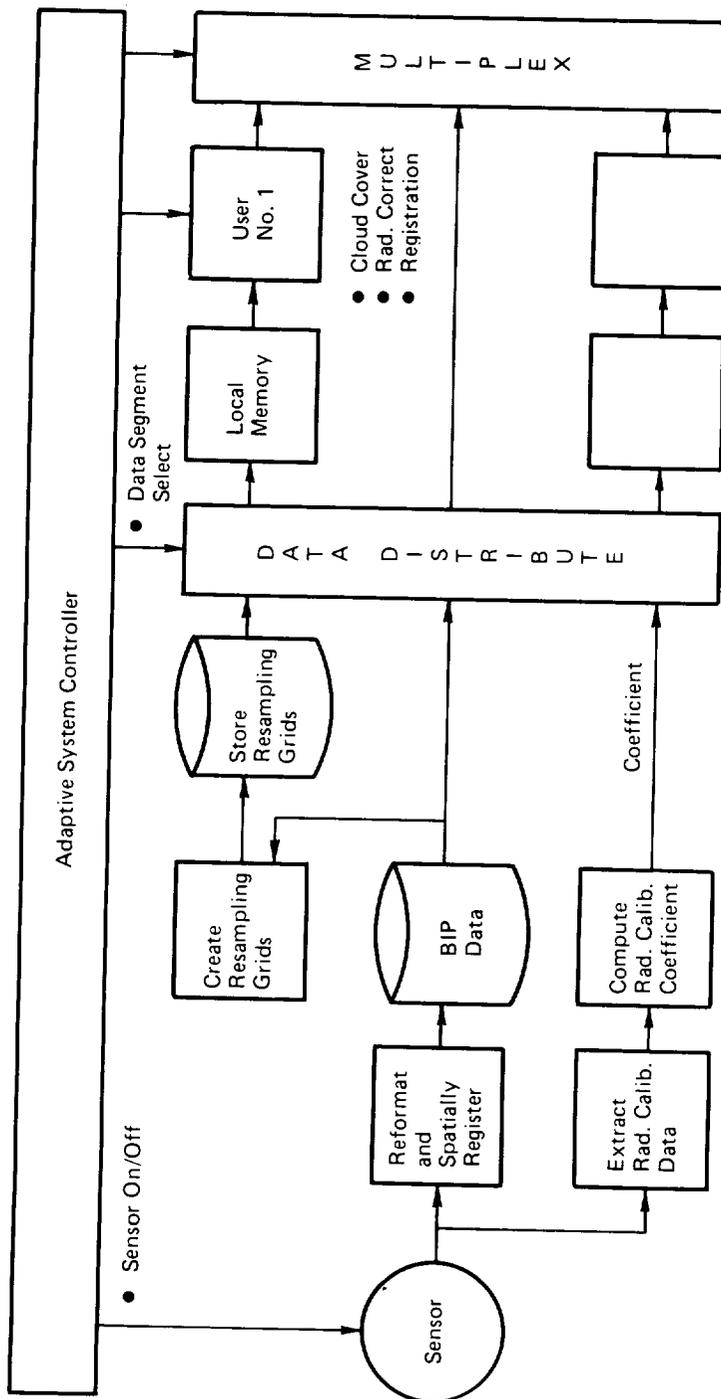


Figure 36. Adaptive System Controller

best accomplished as a common function as part of the sensor pre-processing functions.

Figure 37 illustrates the data flow for creating the resampling grids on-board the spacecraft. Currently, the spacecraft attitude and positional data are computed at the ground at a computing facility remote from the sensor data receiving site. This smoothed attitude and position data are transmitted to the sensor data receiving sites and combined with ground control point data to produce the desired resampling grids. This same process can now be performed on-board the spacecraft as shown in Figure 37. Here, ground control points are located within the image data, combined with GPS positional information and MMS attitude information, and processed by a Kalman filter to provide attitude and altitude coefficients or estimates. This data is then mapped via the Space-to-Space mapping algorithm to generate the line and pixel coordinates of the resampling grid points.

4.1.1 Data Set Selection

Having reviewed the purpose of the IAS let us return to the purpose of this task, which is to define data set selection criteria to be used to determine whether data are of value and warrant further processing.

A number of criteria can be employed to reduce the data volume through the elimination of extraneous data. Criteria for data selection could involve:

1. Cloud cover
2. Sample segment extraction

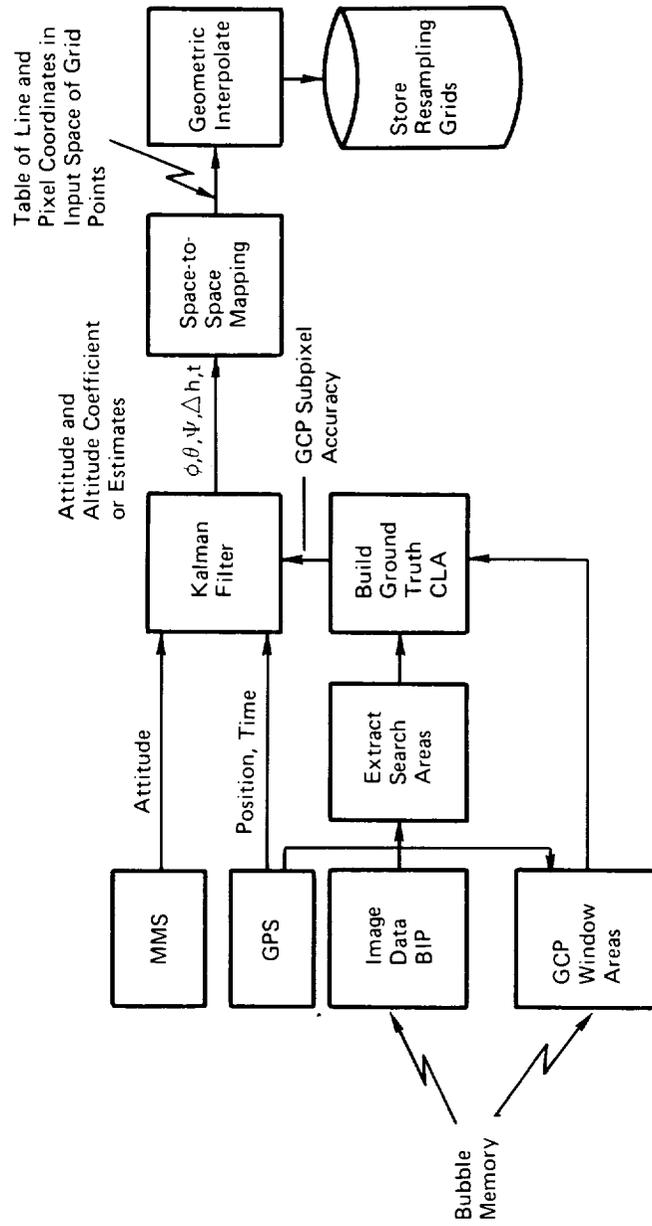


Figure 37. Creating Data Flow For Resampling Grids On-Board

3. Spectral band selection
4. Pixel resolution
5. Availability of ancillary data
6. Elimination of redundant data
7. Target or feature selection
8. Multiple samples.

Currently, nearly half of the scenes taken by the Landsat Multi-Spectral Scanner (MSS) have so much cloud cover that they are not useful. Yet, all MSS data (cloud covered included) is transmitted through the system until a preprocessing function has determined the extent of cloud cover. Significant savings could be realized if these images could be discarded at the sensor before transmission to the ground. The following section will describe five potential automatic cloud cover detection techniques.

Many applications only require information from specific regions. Monitoring forest fires requires land areas known to contain timber. Monitoring oil spills requires water areas known to contain shipping lanes and oil wells. Specific regions, or sample segments, can be identified by a latitude/longitude grid and only the data within that grid transmitted.

A programmable mode of operation that allows the selection and transmission of only a selected set of spectral bands is desirable for certain applications. For example, the Corps of Engineers uses only the MSS bands 4 and 7 or TM bands 1 and 4 as input for some of their

flood control programs. Many geologic investigations (i.e., discriminating rock types) only utilize the IR spectrum.

The pixel resolution for some applications could be lowered to reduce data volume without sacrificing the quality of the desired information. In some cases, the ancillary data to achieve a required geometric accuracy may not be available, rendering the sensor data less useful.

A significant amount of swath overlap occurs on successive spacecraft earth orbits resulting in redundant data for applications requiring large temporal separation between samples. This overlap represents redundant data that could be eliminated. Conversely, for applications where the sample segments are required with minimum temporal separation, this overlap would be beneficial.

If target boundaries and coordinates are well known, all data outside the target boundary could be deleted.

The peculiarities of agricultural remotely sensed data requirements evoke special sensor requirements. Vegetative species do not possess significantly different spectral signatures at given phases of their development cycle. Hence, the key to their discriminability is the phasing of the phenologic cycle of the subject species. Significant improvements in classification can be obtained by consistently employing multi-temporal observations taken at specific times during the year.

Figure 38 now illustrates how this data set selection criteria would be used for two different users. Two criteria expected to be used by most users will be cloud cover and data segment select. As shown in the Figure, the data distribution function will cause the sample

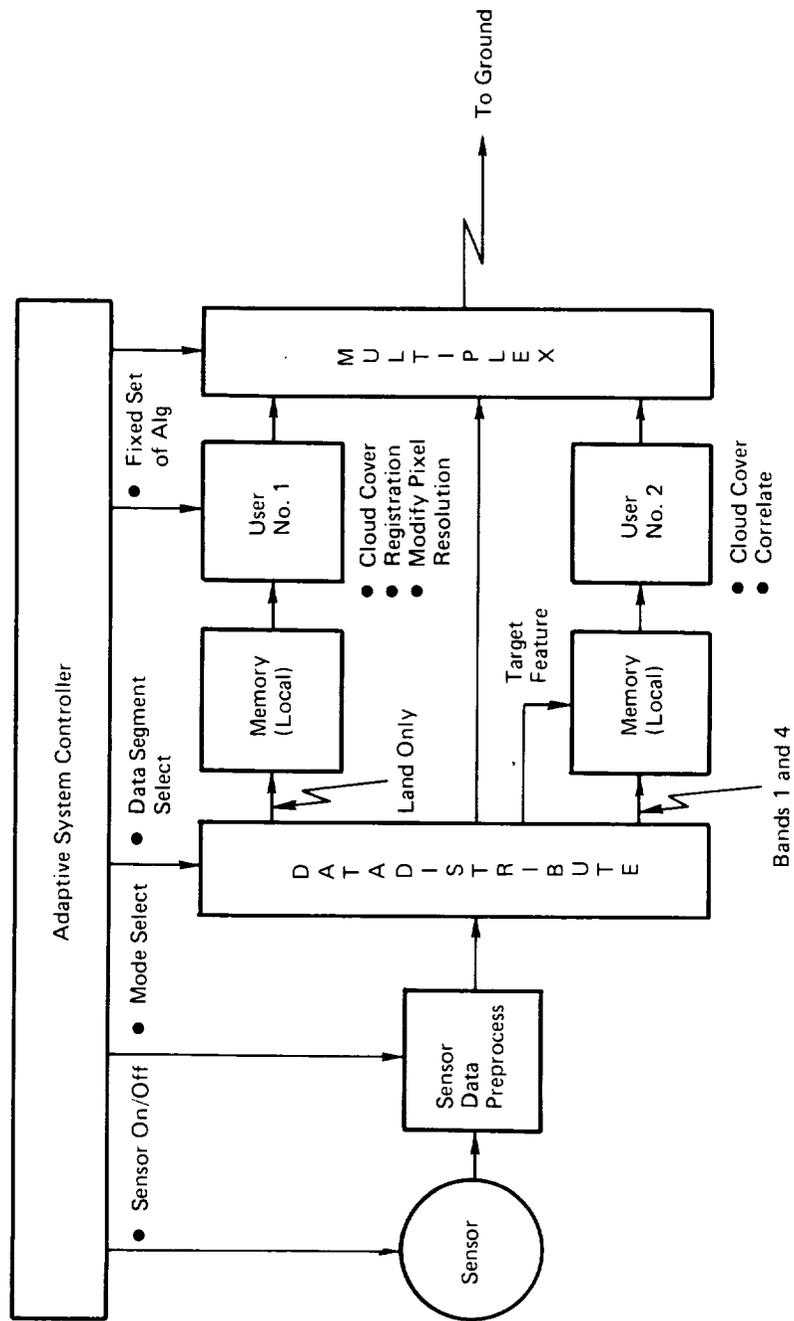


Figure 38. Data Set Selection

segments of interest to each user to be loaded into a local memory. User #1 wants only land areas. The data will be evaluated for cloud cover, registered, and the pixel resolution modified to reduce the downlinked data volume. User #2 is interested in only bands 1 and 4 of the image data. His data will be evaluated for cloud cover provided the cloud cover is less than 50% correlated with the intensity profile of a known target. As shown, an alternate path would allow some other selected data segment to be directly downlinked to ground.

4.1.2 Automatic Cloud Cover Detection Techniques

Currently, much of the earth resources sensor data received at the ground are unuseable due to cloud cover. A capability to automatically evaluate cloud cover on-board a spacecraft would greatly improve the percentage of cloud-free usable data downlinked to the ground. This section presents the results of an IBM study evaluating various automatic cloud cover techniques.

The fundamental problem facing an automatic cloud cover detection technique is the definition and characterization of pixel spectral signatures allowing differentiation between clouds, snow and highly reflective sand. The reflectance of a ground surface or of an interface such as the top of a cloud, can be defined as the ratio of the reflected power to the total power incident to the interface per unit area of irradiated surface. Each channel of an earth resources sensor measures total irradiance with contributions reflected from the ground or from cloud tops plus atmospheric backscattered irradiance. Therefore, the discrimination between clouds, snow, ice or highly reflective sand generally requires the use of radiative transfer models either for a standard atmosphere or for an atmosphere which includes cloud formations. Moreover, the estimation of reflective

signatures demands the use of radiative transfer models, since irradiance as measured by an earth resources channel is not directly proportional to ground or cloud reflectance, due to the interaction effects between clouds or the cloud free atmosphere.

The inclusion of an atmospheric interaction model is then a firm requirement for cloud detection/discrimination methods based on radiance data that has been sensed after its interaction with the atmosphere.

Analysis of five different candidate cloud detection techniques was undertaken:

1. A ratioing method
2. ERIM's method
3. Infrared data method
4. Homomorphic Wiener Filtering in the frequency domain
5. Reflectance test method using a radiative transfer model.

The ratioing method does not include an atmospheric interaction model and therefore may fail to detect clouds. The ERIM method has not been validated for cloud detection and has a rather large CPU requirement (relative to an IBM/S370 M168 Model III). Homomorphic filtering, in a version slightly different from that described, has been applied to remove thin cloud cover in small areas of a scene. It has not been tested as a cloud cover estimator. The infrared data method will detect some transparent (invisible) cirrus and in some cases it may fail to give accurate estimates of cloud cover. The reflectance test method does not appear to have severe physical limitations. The atmospheric

interaction model used by this method is more realistic and accurate than the atmosphere cloud interaction model required in the homomorphic filtering method.

Ratioing Method

Given that ground reflectances (a local property of the ground) can be estimated by accounting for clouds or atmospheric interactions, the ratio of reflectances in two bands and the two reflectances in each band can be used to define a reflectance signature vector which is generally characteristic of a type of ground surface.

A spectral signature given in terms of spectral radiance (watts/steradian-square meter) is not invariant for a given ground feature since it depends on the varying spectral irradiance due to the sun and sky which in turn are functions of the sun's elevation angle β . The ratioing method is a procedure designed to construct signatures which are independent of atmospheric effects after appropriate allowances are made for them.

It can be shown that the virtual pixel reflectance for band i , denoted ρ_{vi} is

$$\rho_{vi}(p,q) = \frac{1}{\tau_{\beta i} \tau_{zi}} \left[\frac{L_i(p,q)}{I_i} - \frac{\rho'_{\beta i}}{\sin \beta} \right] \quad (13)$$

where p, q are pixel coordinates

$L_i(p,q)$ is the sensed digital radiance output

$\tau_{\beta i}$ is the monochromatic one-way transmissivity of the atmosphere at elevation angle β .

τ_{zi} is the monochromatic transmissivity of the atmosphere in the zenith direction for solar radiation reflected by the surface to the nadir viewing sensor.

β is the sun elevation angle

I_i is the modulated incident flux for channel i

$\rho_{\beta i}$ is the monochromatic atmosphere reflectance at angle β .

Neglecting the atmospheric effects, i.e., assuming

$\tau_{\beta i} \tau_{zi} = 1$, $\rho'_{\beta i} = 0$ produces

$$\rho_{vi} = \frac{L_i}{I_i}$$

(14)

$$\frac{\rho_{v1}}{\rho_{v2}} = \frac{L_1}{L_2}$$

which are very poor estimates of a spectral reflectance signature.

Thus in applying the ratioing technique it is essential to use accurate atmospheric models to allow the correction of atmospheric transmission effects.

ERIM's Method

This method has been developed by the Environmental Research Institute of Michigan (ERIM) and is described in References 8, 9, 10, and 11. The method is based on the use of the likelihood function $L(u/w)$ as the conditional probability density of the signal u given the state w for automatic pixel classification.

This method has not been fully validated, however it is of interest to describe its processing steps.

STEP 1 - Radiometrically correct each channel

Let χ_i denote the signal in channel i , then if radiometric correction is necessary, compute for each channel

$$\chi'_i = A_i \chi_i + B_i \quad (15)$$

$$\chi_i = \chi'_i$$

where A_i is the gain and B_i the bias correction necessary for channel i .

STEP 2 - Perform sun angle correction

Let χ_i denote the signal in channel i , following Step 1. Let θ represent the solar zenith angle for the acquisition. Then for each channel of the acquisition compute

$$\chi'_i = \frac{\cos \theta_o}{\cos \theta} \chi_i \quad (16)$$

where θ_0 is the solar zenith angle

then set $\chi'_i = \chi_i$.

STEP 3 - Identify bad data, clouds, water, and cloud shadows

Let χ_i denote the signal in channel i , following Step 2. Let i and j correspond to rows and columns of a rotation matrix R , where R is known. For each pixel of an acquisition, and for each value of j , compute

$$z_j = \sum_{i=1}^4 R_{ij} \chi_i \quad 1 \leq j = 4 \text{ for MSS} \quad (17)$$
$$z = R^T \chi$$

Depending on various threshold values for z_1 , z_2 , z_3 , and z_4 , the pixel is identified as bad, cloud, water, or cloud shadow.

The required CPU time for the IBM/S370M168 assuming a full frame of 2240 lines and 3270 samples (MSS) is

$$\text{CPU time} = 353.8 \text{ sec} + (1.17 \text{ sec}) C_c$$

where C_c is the fraction of cloud cover ($0 < C_c < 1$).

While this is a significant computational time, the potential advantage of the ERIM process is its use of all the spectral data in determining cloud cover.

Infrared Data Method

The MSS infrared channel in the 10.5 μm to 12.5 μm wavelength band measures radiance emanating from the earth-atmosphere system, in a spectral "window" where the atmospheric contribution to the sensed radiance is relatively small and hence negligible for the determination of pixel signatures.

In case of a clear or cloud free line of sight, the radiance observations provide good estimates of the ground surface temperature since a clear atmosphere is nearly transparent in the 10.5 μm to 12.5 μm window. Consequently this channel of radiance data, R, can be considered as giving either estimates of ground surface temperatures, T_g , under clear skies or cloud top temperatures, T_c , under overcast conditions. The relation between R and temperature T is given by Planck's formula. An approximation to Planck's formula was derived by W. Wien and is

$$R(x,y) = 2\epsilon hc^2 \int_{\lambda_a}^{\lambda_b} \lambda^{-5} e^{-\left(\frac{hc}{\lambda kT}\right)} d\lambda \quad (18)$$

or

$$R(x,y) = \epsilon c_1 \int_{\lambda_a}^{\lambda_b} f(\lambda, T) d\lambda, \quad f(\lambda, T) = \lambda^{-5} e^{-c_2/\lambda T}$$

where

$$c_1 = 2hc^2 \quad \& \quad c_2 = \frac{hc}{k}$$

The integral can be computed in closed form as

$$R = \frac{\epsilon C_1}{C_2} T^4 \left\{ e^{-(k_1+1)} \left[k_1^3 + \frac{3}{e} k_1^2 + \frac{6}{e} (1+k_1) \right] - e^{-(k_2+1)} \left[k_2^3 + \frac{3}{e} k_2^2 + \frac{6}{e} (1+k_2) \right] \right\} \quad (19)$$

where

$$k_1 = \frac{c_2}{\lambda_b T} \quad \& \quad k_2 = \frac{c_2}{\lambda_a T}$$

Using Newton's method, $T(x,y)$ will be determined for a given $R(x,y)$; however, the solution is required for each pixel.

The computed temperature will establish whether the pixel represents a cloud top or not.

Typical cloud top temperature is 255°K

Typical ground temperature is 295°K .

The solution for T given R requires a fast and efficient algorithm for a systematic implementation. It is difficult, at present, to estimate computer requirements if this algorithm is unavailable. However, if a table look-up for T as a function of R is sufficiently accurate, the required CPU time for the IBM/S370 M168 (assuming a 3200 x 2340 array) is about .5 sec. plus 3 sec. for radiometric correction or 3.5 sec. per scene.

Homomorphic Filtering Method

This technique illustrated in Figure 39 maps the signal from its original representation space into another space where desired processing operations are performed.

If we assume that the cloud reflectance of sunlight plus the cloud transmissivity equals 1, then the intensity measured by a single channel of a scanner is:

$$L(x,y) = \alpha I \rho(x,y) T(x,y) + I[1 - T(x,y)] \quad (20)$$

where $\rho(x,y)$ is the reflectance of the ground surface

$T(x,y)$ is the clouds transmissivity

I is the sunlight incident flux

α is the sunlight attenuation

and $\rho(x,y)$, $T(x,y)$ & α range between 0 & 1.

A homomorphic transformation can be performed by subtracting $L(x,y)$ from I and taking the logarithm

$$\begin{aligned} d(x,y) &\triangleq \log [I - L(x,y)] \\ &= \log [I - \alpha \rho(x,y)] + \log [T(x,y)] \triangleq s(x,y) + n(x,y) \end{aligned} \quad (21)$$

A best (least squares) separation of $s(x,y)$ and $n(x,y)$ can be estimated by Wiener filtering. In order to apply Wiener filtering, the power spectral density of $n(x,y)$ (which is related to the cloud transmittance

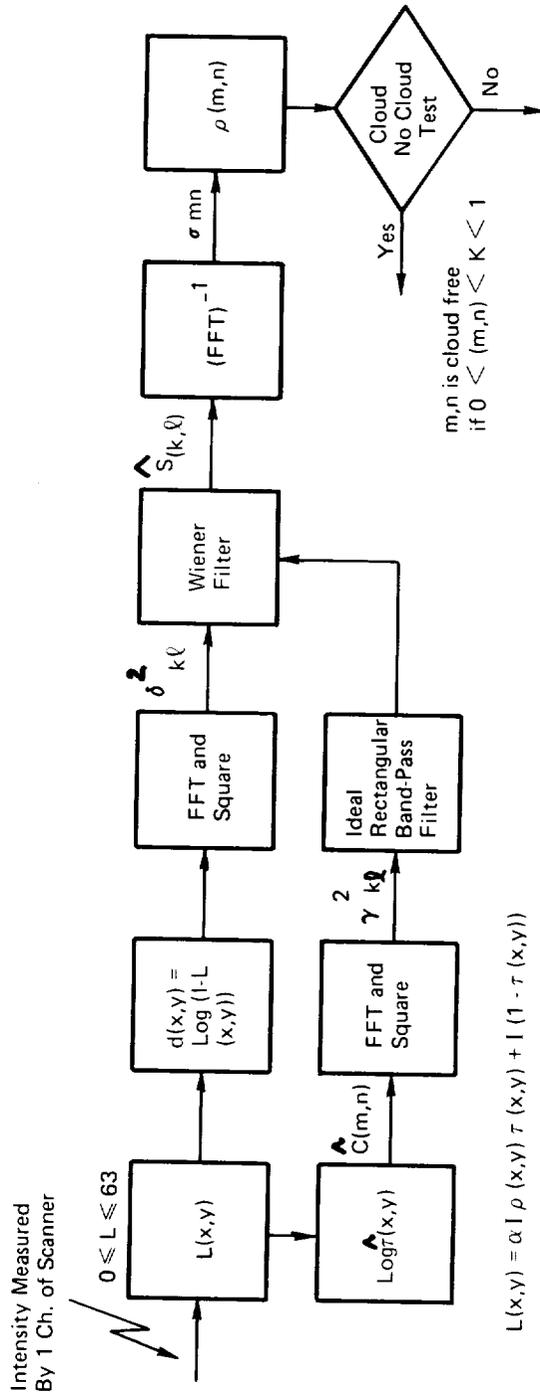


Figure 39. Homomorphic Filtering Method

T) is estimated using a standard FFT procedure.

$$\hat{T}(x,y) = \frac{1 - L(x,y)}{1 - \alpha \rho_{av}} \quad , \quad \hat{C}(x,y) = \log \hat{T}(x,y) \quad (22)$$

where ρ_{av} is the average value of $\rho(x,y)$, dependent on the date of year and WRS frame.

Now take the Fourier Transform of $\hat{C}(m,n)$ & square

$$\begin{aligned} \text{where} \quad x = m \quad & 0 \leq m \leq M-1 \\ y = n \quad & 0 \leq n \leq N-1 \end{aligned} \quad (23)$$

$$\text{FFT } \hat{C}(m,n) \triangleq \gamma_{kl} e^{j\phi_{kl}} \quad \begin{aligned} 0 \leq k \leq M-1 \\ 0 \leq l \leq N-1 \end{aligned}$$

and Power Spectrum is $(\gamma_{kl})^2$

After computing the noise power spectrum, the high-frequency and very low-frequency parts of this power spectrum are removed using an ideal rectangular band-pass filter. High-frequency components are most likely due to ground reflectance (concrete roads, parking areas, etc.). Low-frequency components are most likely due to extended areas such as ice or snow.

The same FFT procedure is applied without the rectangular window band-pass filter to the transformed data $d(x,y)$ (i.e., signal plus noise), to obtain the power spectrum δ_{kl}^2 .

$$\text{FFT } d(m,n) = \delta_{kl} e^{j\theta_{kl}} \triangleq \hat{d}(k,l) \quad (24)$$

The Wiener filter separating signal from noise is

$$H(k, \ell) = \frac{\sigma_{k, \ell}^2 - \gamma_{k\ell}^2 - \eta_s \eta_c R_{mn}(k, \ell)}{\sigma_{k, \ell}^2} \quad (25)$$

where η_s is the mean of signal

η_c is the mean of noise

$$R_{MN}(k, \ell) = \begin{cases} 1, & 0 \leq k \leq M-1, 0 \leq \ell \leq N-1 \\ 0, & \text{otherwise} \end{cases} \quad (26)$$

The least square estimate of the FFT $S(m, n)$ is

$$\hat{S}(k, \ell) = H(k, \ell) \hat{d}(k, \ell) \quad (27)$$

The Wiener estimate of the signal is

$$S(m, n) = [\text{FFT}]^{-1} \hat{S}(k, \ell) = \sigma_{mn} e^{j\psi_{mn}} \quad (28)$$

The estimated signal is

$$S(m, n) = \log [I - \alpha I \rho(m, n)] \quad (29)$$

from which $\rho(m, n)$ is determined.

Cloud free areas are defined by $0 < \rho(m, n) < K < 1$ and cloud boundaries by $\rho(m, n) = K$, where K is an a priori value of cloud reflectance.

The computational time relative to the IBM/S370 M168 for a frame with $M=N=400$ and one spectral band is 160 sec. If four iterations are required, the CPU time is 640 seconds.

Reflectance Test Method Using Radiative Transfer Model

This method invokes the use of atmospheric radiative transfer models allowing a more accurate estimation of pixel spectral reflectance signatures in two bands. The corrected pixel spectral values are ratioed to define a reflectance signature vector.

The total radiance measured by a sensor channel for a band is shown in Figure 40 and is

$$L_i = L_{si} + L_{Ai} = S_i \left[\rho_i \tau_{\beta i} \tau_{zi} \sin \beta + \rho'_{\beta i} \right] R_i \quad (30)$$

where

L_{si} = surface reflected radiance

L_{Ai} = scattered radiation detected by sensor

S_i = solar irradiance of sunlight outside atmosphere in
 β direction

ρ_i = ground reflectance

$\tau_{\beta i}$ = zenith transmissivity

β = solar elevation angle

$\rho'_{\beta i}$ = atmospheric reflectance at angle β

R_i = average spectral response function for channel i .

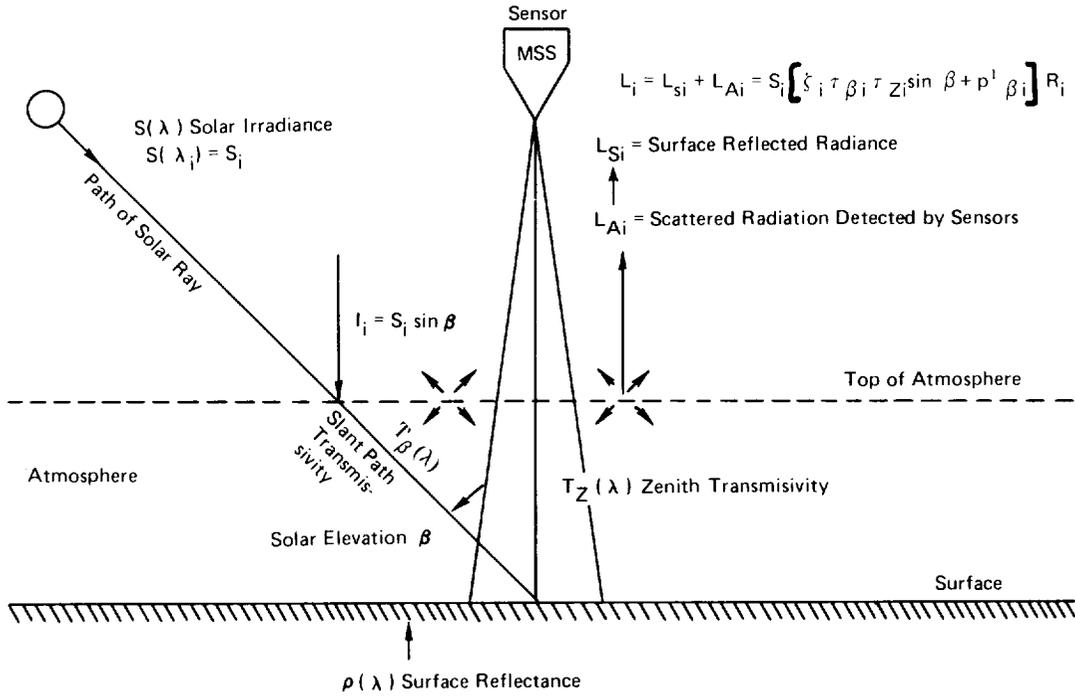


Figure 40. Radiative Transfer Atmosphere Model

Now, let $I_i = S_i R_i \sin \beta$

$$L_i(K, y) = I_i \left[\tau_{\beta}(\lambda_i) \tau_z(\lambda_i) \rho_i(\lambda_i) + \frac{\rho'_{\beta}(\lambda_c)}{\sin \beta} \right] \quad (31)$$

where

$$1 \leq x \leq 2340, 1 \leq y \leq 3264$$

For a given best estimate of I_i , the above equation can be solved for ρ_i , now denoted as ρ_{vi} , the virtual pixel reflectance.

$$\rho_{vi}(p, q) = \frac{1}{\tau_{\beta i} \tau_{zi}} \left[\frac{L_i(p, q)}{I_i} - \frac{\rho'_{\beta L'}}{\sin \beta} \right] \quad (32)$$

where

$$p, q = \text{pixel coordinates } 1 \leq p \leq M$$

$$1 \leq q \leq N$$

$$L_i(p, q) = \text{sensor digital radiance output}$$

A pixel is classified as cloud-free if $\rho_{vi} < K_i < 1$
 and the pixel is considered to be cloud covered if $\rho_{vi} > K_i$.

Using these criteria, the pixels which are classified as cloud covered or cloud free will be determined by the choice of atmospheric attenuation and upward reflectance models. Since the aerosol distribution and turbidity of the atmosphere is locally dependent on air pollution, seasonal and geographical factors, it may be pertinent to use $\tau_\beta(\lambda_i)$, $\tau_z(\lambda_i)$ and $\rho'_\beta(\lambda_i)$ data which are geographical and time dependent and stored in a data base.

Determining the virtual pixel reflectance values requires an input value I_i . The values I_1 & I_2 can be estimated from the scene channel $i=1, i=2$ data

$$\therefore I_i = \frac{L_{i \max}}{\rho_i \tau_{\beta i} \tau_{z i} + \frac{\rho'_{\beta i}}{\sin \beta}} \quad (33)$$

The value $L_{i \max}$ over a scene is determined by means of a histogram program where the 10% upper percentile is excluded to allow for random, signal dependent errors which saturate the signal so that $L_{i \max} = 63$

According to IBM-FSD studies of cloud detection using MSS data on a CRT display, the best choice of channels for a two channel cloud detection system was that of channels 4 and 7 with $\lambda_2=0.55\mu\text{m}$ and $\lambda_1=0.95\mu\text{m}$. Band 4 imagery showed snow and clouds to be bright and served to identify thick, highly reflecting clouds. Band 7 background was less bright allowing a better discrimination in the case of thin, almost transparent cloud formations.

Cloud-free pixels exist if the following conditions exist

$$\rho_{v1} < K_1, \rho_{v2} < K_2$$

where K_1 and K_2 are positive input parameters less than or equal to one.

Cloud covered pixels are all those pixels which do not satisfy conditions for cloud-free pixels.

This procedure shown in Figure 41 allows eight surface types to be distinguished based on the ratio of the virtual reflectances and the value of the virtual reflectances. The eight distinguishable surface types are:

- a. Snow-ice
- b. Crushed limestone
- c. Desert sand
- d. Weathered tuff bedrock
- e. Vegetation
- f. Black loam soil
- g. Flood plain gravel
- h. Water.

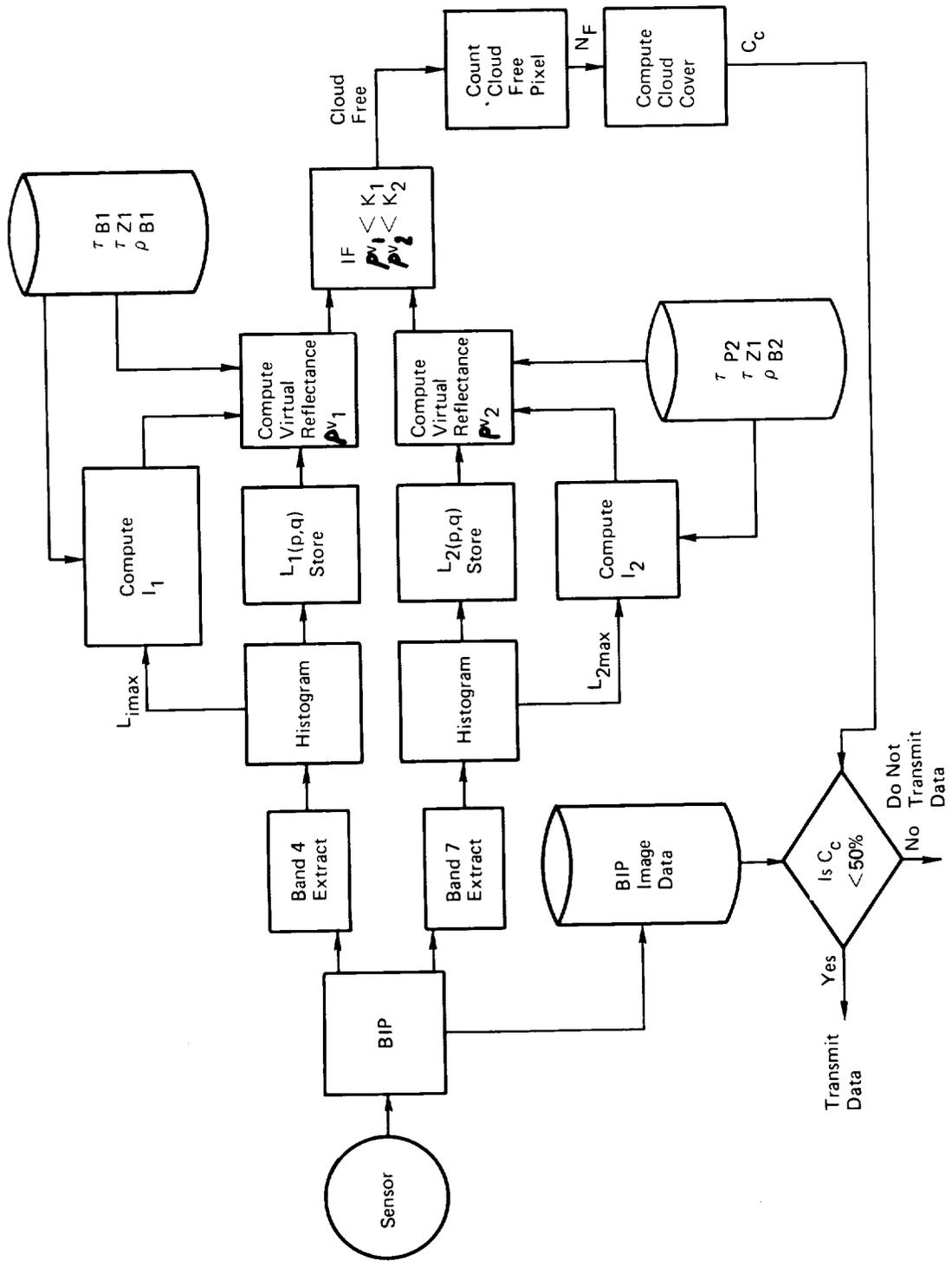


Figure 41. Reflectance Test Method Using Radiative Transfer Model

The percent cloud cover can now be estimated in the following manner. Let N_f be the number of pixels found to be cloud-free in an $M \times N$ frame of a scene. The expected cloud cover in percent is then

$$C_c = 100 \left(1 - \frac{N_f}{M \times N} \right) \quad (34)$$

The computational time based on runs using the IBM/S370 M168 have shown CPU time per pixel of 6×10^{-5} sec. Thus for a full picture with $N=2340$ lines and $M=3264$ pixels, the CPU time is 7 min. 39 sec. For $M=N=400$, the CPU time is 9.6 sec.

Comparative Analysis

The ratioing method does not include an atmospheric interaction model and therefore will fail to detect clouds. The ERIM method, while making use of all available spectral data, has not been validated for cloud detection and has a rather large CPU requirement.

The infrared method will detect some transparent (invisible) cirrus but in some cases may fail to give accurate estimates of cloud cover. Homomorphic filtering has been applied to remove thin cloud cover in small areas of a scene. It has not been tested as a cloud cover estimator. Its major drawback would appear to be the CPU requirements.

The reflectance test method using a radiative transfer model does not appear to have any physical limitations, uses an atmospheric interaction model more realistic and accurate than the atmosphere cloud interaction model used in the homomorphic filtering method, and requires minimum CPU time. These factors suggest the reflectance test method as the current best compromise to estimate cloud cover.

4.2 Conclusions

Presently, large amounts of unusable image data or data of no interest to a user, are transmitted, processed, and stored because most spacecraft do not have an on-board data set selection capability. A number of criteria for data selection were suggested and discussed. These criteria include:

- a. Cloud cover
- b. Sample segment extraction
- c. Spectral band selection
- d. Pixel resolution
- e. Availability of ancillary data
- f. Elimination of redundant data
- g. Target feature selection
- h. Multiple samples.

The first three criteria offer the greatest potential for reducing the volume of image data transmitted to the ground. Five candidate cloud detection techniques capable of on-board implementation were presented and reviewed. The reflectance test method using a radiative transfer model represented a computationally efficient, yet accurate technique for cloud detection. Sample segment extraction and spectral band selection allows a user to access that information of interest to him. This data selectivity is very dependent upon the capability inherent in the adaptive system controller. If the data of interest by a user remains fixed, the system controller can be configured to provide the proper image data according to a fixed, predetermined timeline. If a user desires to change his data of interest, then the system controller must have the capability to reconfigure the processing timeline. The degree of data set selection, therefore, is directly dependent on the capability of the adaptive system controller.

5.0 TASK 4 - ON-BOARD FUNCTIONAL REQUIREMENTS FOR INTERFACING WITH GLOBAL POSITION SATELLITES (GPS)

The purpose of this task is to investigate the functional requirements and potential accuracy of using position and time information from Global Position Satellites and spacecraft attitude to compute, on-board, the ground location to which an imaging system is pointing.

Background information will be presented for the GPS system, characteristics of the system will be described and the expected accuracy of the system determined. This position and time information will be combined with altitude errors plus other system errors to determine pixel ground location accuracy.

The Global Positioning System is a space-based radio navigation and positioning system that will provide very accurate, three dimensional position and velocity information and system time (GPS time) to equipped users worldwide.

The GPS consists of 24 satellites broadcasting coded navigation messages continuously on two different radio frequencies, L_1 & L_2 . The user equipment receives and processes the navigation messages from four satellites and derives its position, velocity and system time.

The GPS is divided into three major components: a Space Segment (SS), a User Segment (US) and a Control Segment (CS). All three segments are interrelated but each has its own particular function.

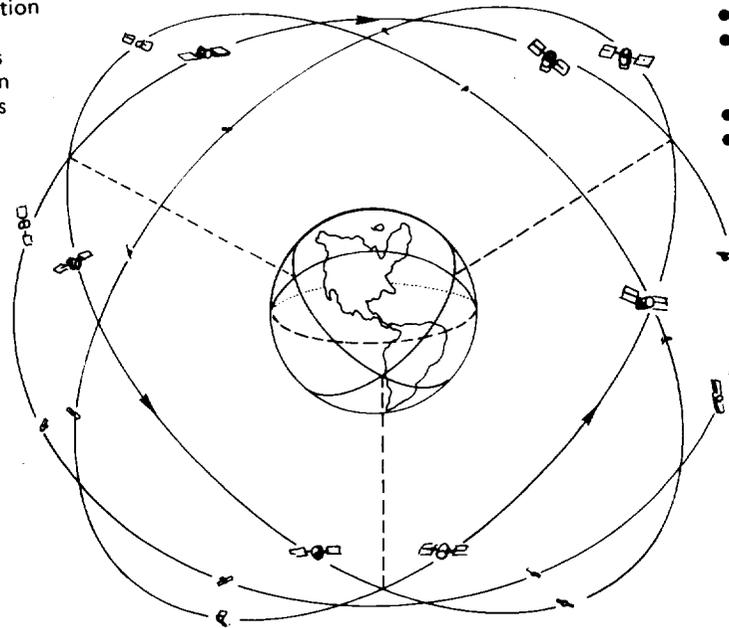
The Space Segment consists of the GPS satellites and the launch vehicle required to place the Space Vehicles (SV's) in orbit. In the operational configuration, the GPS space segment will consist of 24 operative satellites in a 20,183 km orbits. The satellites will be distributed in three circular polar orbits phased 120° apart as shown in Figure 42, and have 12-hour periods.

The User Segment refers to the receiver/processors used by the various users to receive the GPS signals; process the navigational data transmitted from the satellite to derive position, velocity and time.

The Control Segment includes all ground facilities required to support the 24 satellites. Each satellite must be uploaded at least once per day with new navigation information. A Master Control Station (MCS) is the facility that houses the personnel and equipment necessary to generate this daily navigation message. A Ground Control Station (GCS) is co-located with the MCS and consists of the equipment necessary to transmit all generated messages to the satellites. An Alternate Control Center (ALT) will be used in the operational phase to back up the MSC and GSC functions in case of an emergency. The ALT will be geographically separated from the MCS and GCS. Monitor Sets (MS) complete the Control Segment. These are passive, unmanned van-type facilities that are spread geographically throughout the world and report to the MCS on the performance of each space vehicle currently in view. There will be five to eight Monitor Sets deployed during the operational phase.

Because of the clear potential of GPS, the General Accounting Office in a report issued in March 1978, has recommended that, except for self-contained inertial and doppler radar navigation systems and the Coast Guards marine non-directional beacons, all other navigation systems

- Orbital Configuration**
- 12-hr Period
 - Circular Orbits
 - 55° Inclination
 - Eight Satellites Per Plane



Spacecraft Configuration

- 950 lb (Typical)
- 300 Watts (Typical)
- Nav Signals
 - 1227 MHz
 - 1575 MHz
- One Way Tracking
- 10^{-13} Clock Stability

Figure 42. Orbital Configuration

besides GPS should be phased out by the end of the 1980's. The systems which they recommend phasing out include:

- VHF Omnidirectional Range (VOR)
- Tactical Air Navigation (TACAN)
- Loran - A
- Loran - C
- Loran - D
- Omega
- Transit
- Differential Omega.

The Department of Transportation (DOT) has a slightly different viewpoint. They feel that NAVSTAR is not accurate enough for aircraft landings, and anticipate international use of a microwave landing system (MLS) which is expected to replace the instrument landing system (ILS) which has been in operation at commercial airports for over 30 years. In contrast to the GAO recommendation, DOT does not have any specific plans for phasing out the Loran-C and Omega navigation systems. They feel it is unwise for us to depend solely on a satellite system without the backup of a land based combination such as Omega and Loran-C, even if the cost of GPS receivers for civil use becomes competitive.

Depending on the accuracy required, a receiver can be selected to provide 10 meter 3-dimensional position accuracy, or 100 meter accuracy for orbit determination. Table 6 shows the accuracy for orbit determination available from the NASA Spacecraft Tracking and Data Network (STDN), using timely data and also showing the deterioration of predictions at various altitudes. Note that even the coarse GPS component offers the best accuracy. Since it is always available, there is no need for a predicted ephemeris to be transmitted to the user spacecraft, nor for

Table 6. Approximate 1σ Accuracy Orbit Determinations

	<u>Geostationary</u>	<u>6000 km</u>	<u>1000 km</u>	<u>300 km</u>
	<u>Altitude</u>	<u>Altitude</u>	<u>Altitude</u>	<u>Altitude</u>
NASA Spacecraft Tracking and Data Network (STDN)				
Within arc of data ⁽¹⁾	0.05 km	0.02 km	0.03 km	0.5 km ⁽²⁾
One week prediction	4 km	1 km	1 km	20 km ⁽²⁾
24 hour prediction	0.5 km	0.2 km	0.2 km	2 km ⁽²⁾
NAVSTAR ₍₃₎ Global Positioning System	0.01 km	0.01 km	0.01 km	0.01 km

Note 1 -- Assumes an optimum length data arc with good S-band tracking distribution as well as a dynamically clean spacecraft.

Note 2 -- Assumes a moderate level of solar activity as well as a dynamically clear spacecraft.

Note 3 -- Coarse system would be 0.1 km.

the ground computer time to calculate one. Having accurate orbit data on board permits its transmission from the user spacecraft along with accurate time received from the GPS signal.

In a 20,183 km circular orbit, each of the 24 nadir-pointing NAVSTAR spacecraft will make 2 revolutions of the earth per day, broadcasting continuous, accurate positioning data through its 12-element helix-shaped beam antenna. The key to the system is having atomic clocks in each spacecraft which will be updated at intervals, from ground stations, with exact timing information on the vehicles course around the earth.

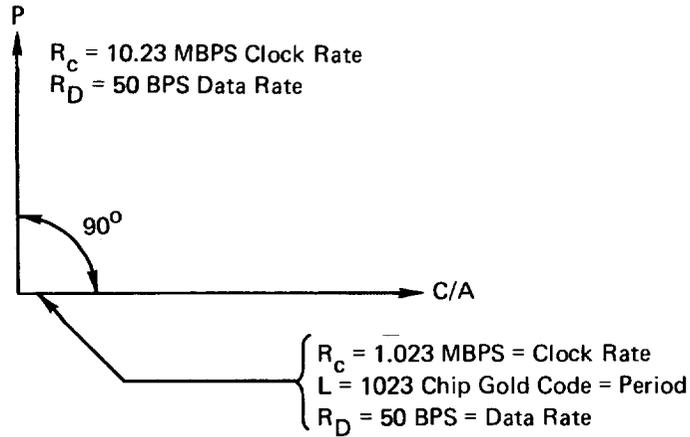
Each NAVSTAR spacecraft simultaneously transmits navigation signals on two channels. The primary purpose of having two frequencies is to permit the user to correct for the signal delay through the ionosphere. The L_1 (1575.42 MHz) is biphase shift key modulated (BPSK) with two codes: a 10.23 MHz Protected (P) pseudorandom noise (PRN) code and a 1.023 MHz Clear/Acquisition (C/A) PRN code. The C/A signal is modulated in phase quadrature to the protected signal as shown in Figure 43. The L_2 channel (1227.6 MHz) carries either P or C/A signals with constant envelope properties. Table 7 summarizes the P and C/A signal characteristics.

5.1 SV Time and Data Control for User Acquisition

It is essential that the GPS SV's timing be under strict control to enable the Users to acquire their signals. This requirement is in the form of a definition of the Navigation Signal Structure. The waveform is designed to allow system time to be conveniently and directly extracted in terms of standard units of days, hours, minutes and integer multiples and submultiples of the second. The P-code for each satellite i is the product of 2 P codes, X1 and X2 (shown in Figure 44) where X1 has a period of 1.5 seconds and X2 has a period 37 chips longer. Both sequences are reset to begin the week at the same epoch time.

GPS Signal

$$S_{L1_i}(t) = \underbrace{AXP_1(t)D_i(t)\cos\omega_1 t}_P + \underbrace{2AXG_i(t)D_i(t)\sin\omega_1 t}_{C/A}$$



P Signal = Long Secure Code with 50 BPS Data
 C/A Signal = 1023 Chip Gold Code with 50 BPS Data

Figure 43. GPS Signal Structure for L1 Signal

Table 7. Summary of GPS Signal Parameters

Parameter	C/A Signal	P Signal
Code Clock Rate, R_c	1,023 Mbps	10.23 Mbps
Code Length	1023	$\approx 6 \times 10^{12}$
Code Repetition Period	1 msec	7 days
Data Rate, R_d	50 bps	50 bps
Transmission Freq.	L1	L1, L2
Frame Length (Bits)	1500	1500
Data Includes:	Telemetry Satellite Ephemeris Satellite Clock Correction Ionospheric Model P-Signal Acquisition Word Almanac	

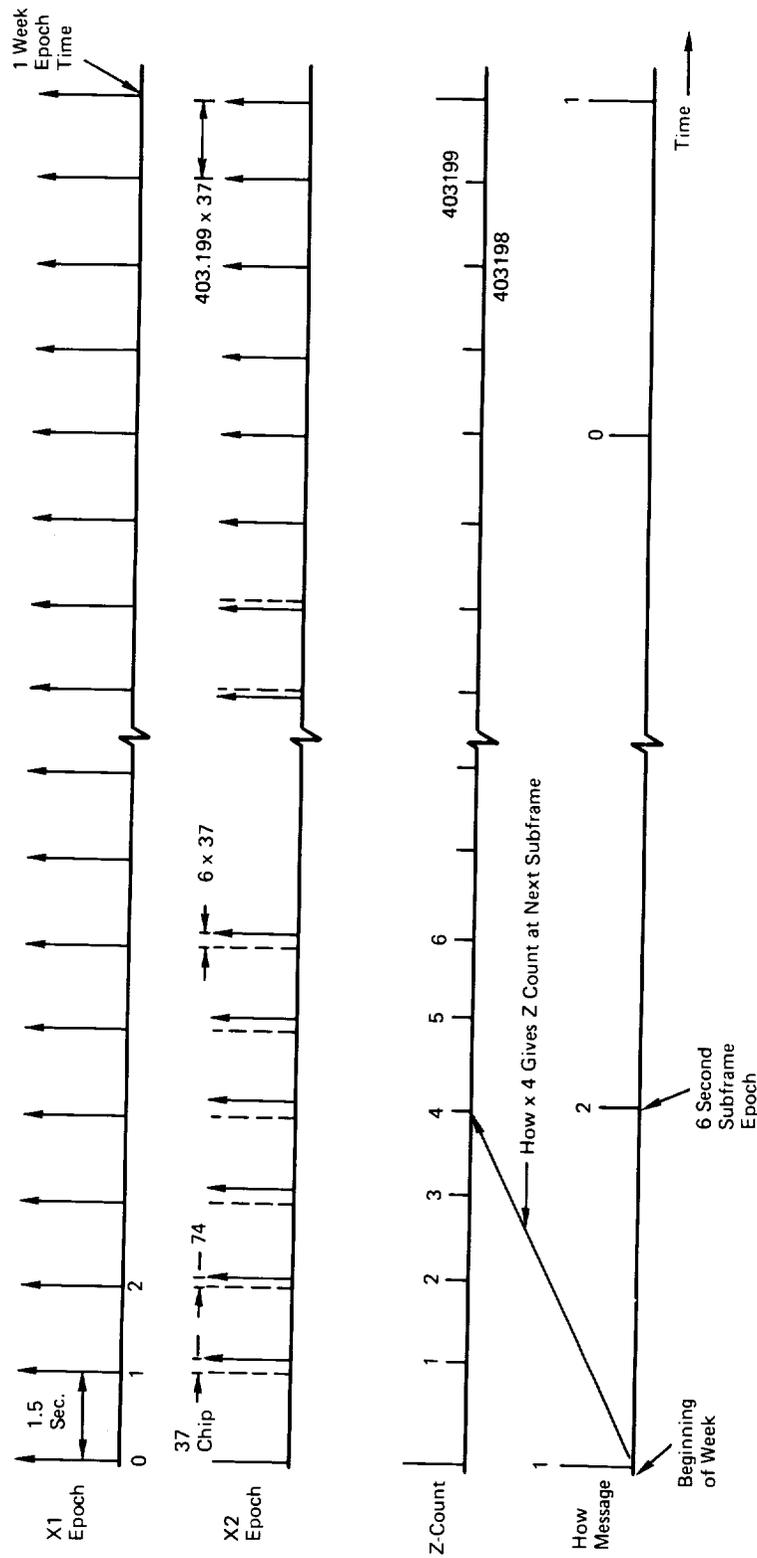


Figure 44. Timing Program for the P Code Components X1, X2, and the Z - Count and HOW-Message Relationship.

The measure of this elapsed time is the number of X1 epochs, termed "Z" count, which have been counted since the P code epoch. The time between X1 epochs is exactly 1.5 seconds of SV time. Thus, a Z count is worth approximately 1.5 seconds of GPS time.

The Z count is transmitted every 6 seconds, and contained in the Handover Word (HOW) of the synchronous data bit stream D, and represents the system time at the start of the next data subframe (a subframe is 6 seconds long).

The C/A (clear/acquisition) signal code shown in Figure 45 has a chipping rate of 1.023 megabits per second. Its XG code epoch occurs every millisecond. A navigation data bit transition occurs every 20 milliseconds, providing data at a rate of 50 bits per second. These bit transitions, the P code epochs, the XG code epochs, the X1 code epochs and the D data bit stream epochs all occur in synchronization at their respective integral multiple rate. These space vehicle timing relationships are shown in Figure 46.

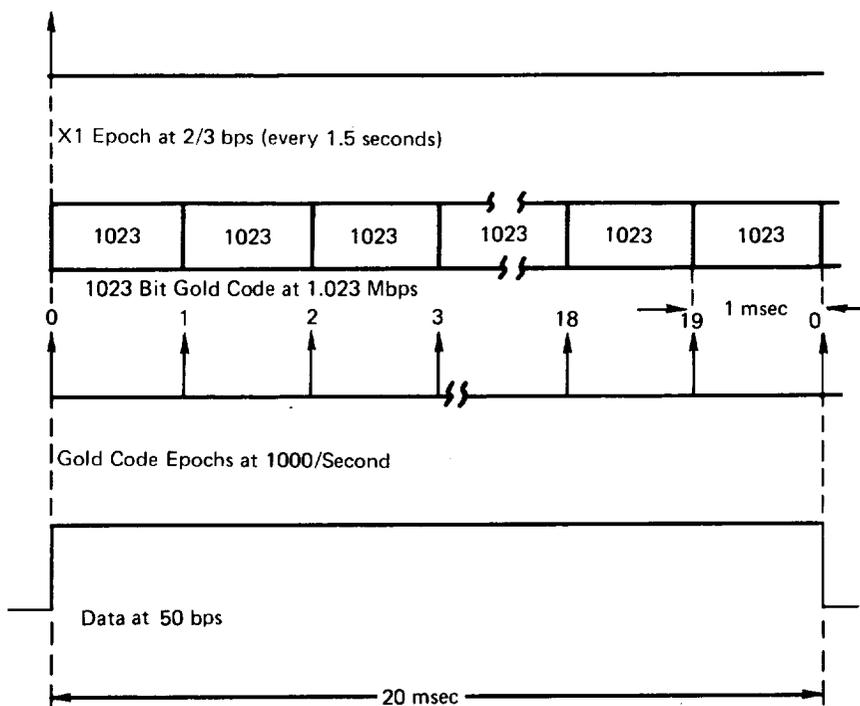


Figure 45. C/A Signal and Data Component Timing

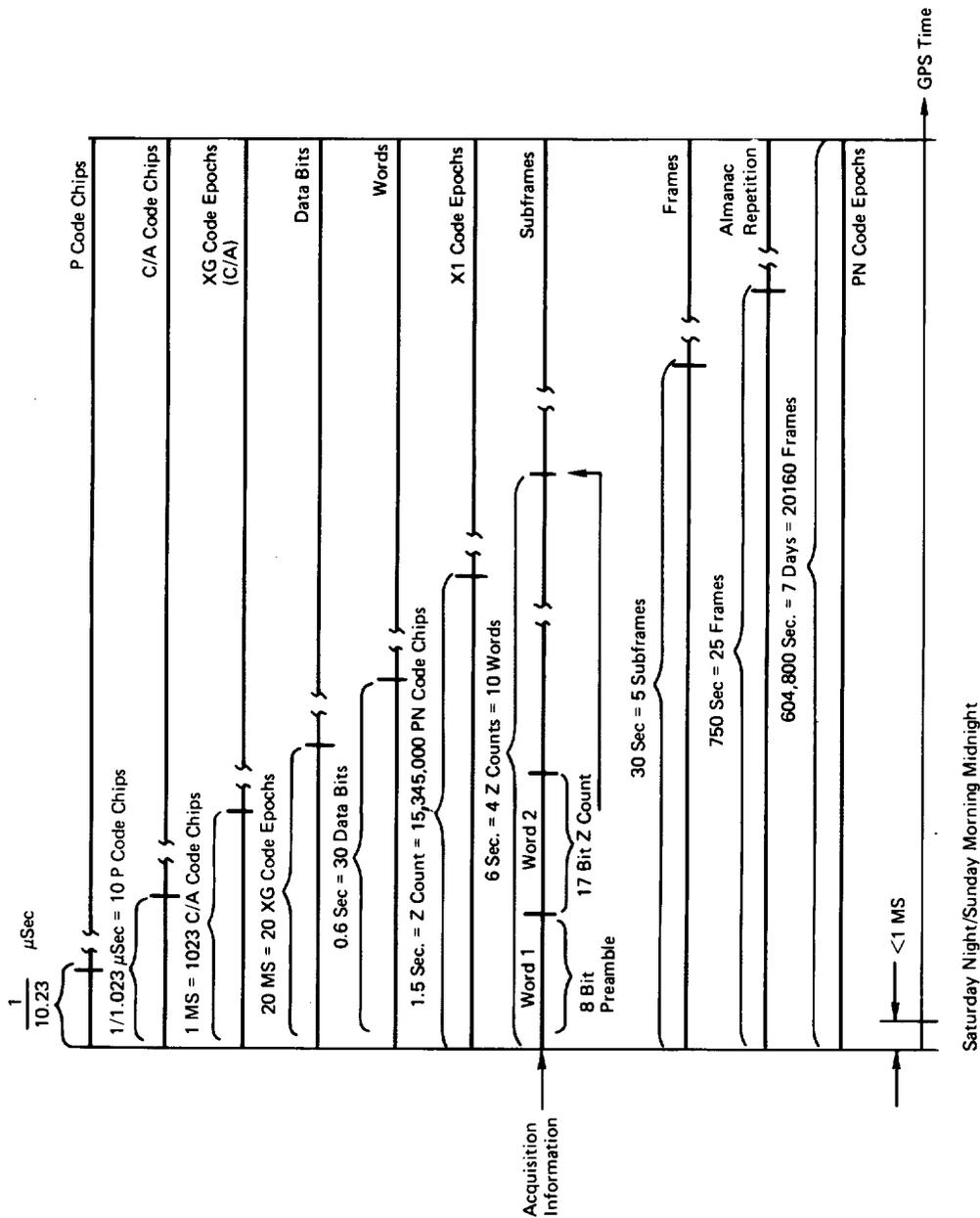


Figure 46. Space Vehicle Timing

The 50 bps data stream modulated on the GPS navigation signal containing the data message with the following information:

- SV ephemeris
- System time (GPS time)
- SV clock behavior data
- Transmitter status information (health)
- C/A to P signal handover information (HOW)
- SV almanac.

The data message is in a data frame 1500 bits long. Each data frame consists of 5 subframes, each 300 bits long. A subframe contains 10 words of 30 bits. The first 2 words of each subframe are the telemetry (TLM) and the handover word (HOW). Subframe 1 contains the space vehicle clock correction parameters and the ionospheric propagation delay parameters. Subframes 2 and 3 contain the space vehicle ephemeris described by Keplerian orbital parameters. Subframe 4 contains a message block. Subframe 5 the space vehicle almanac for up to 24 SV's. The data message is summarized pictorially in Figure 47.

The signal data stream D has two functions for the User. One is to allow the User to navigate continuously, and the second is to aid the User to acquire the transmitted signals. Certain timing information in the data is essential for User acquisition:

- (1) System time
- (2) A preamble for synchronization
- (3) Subframe identification
- (4) SV clock information for all SV's.

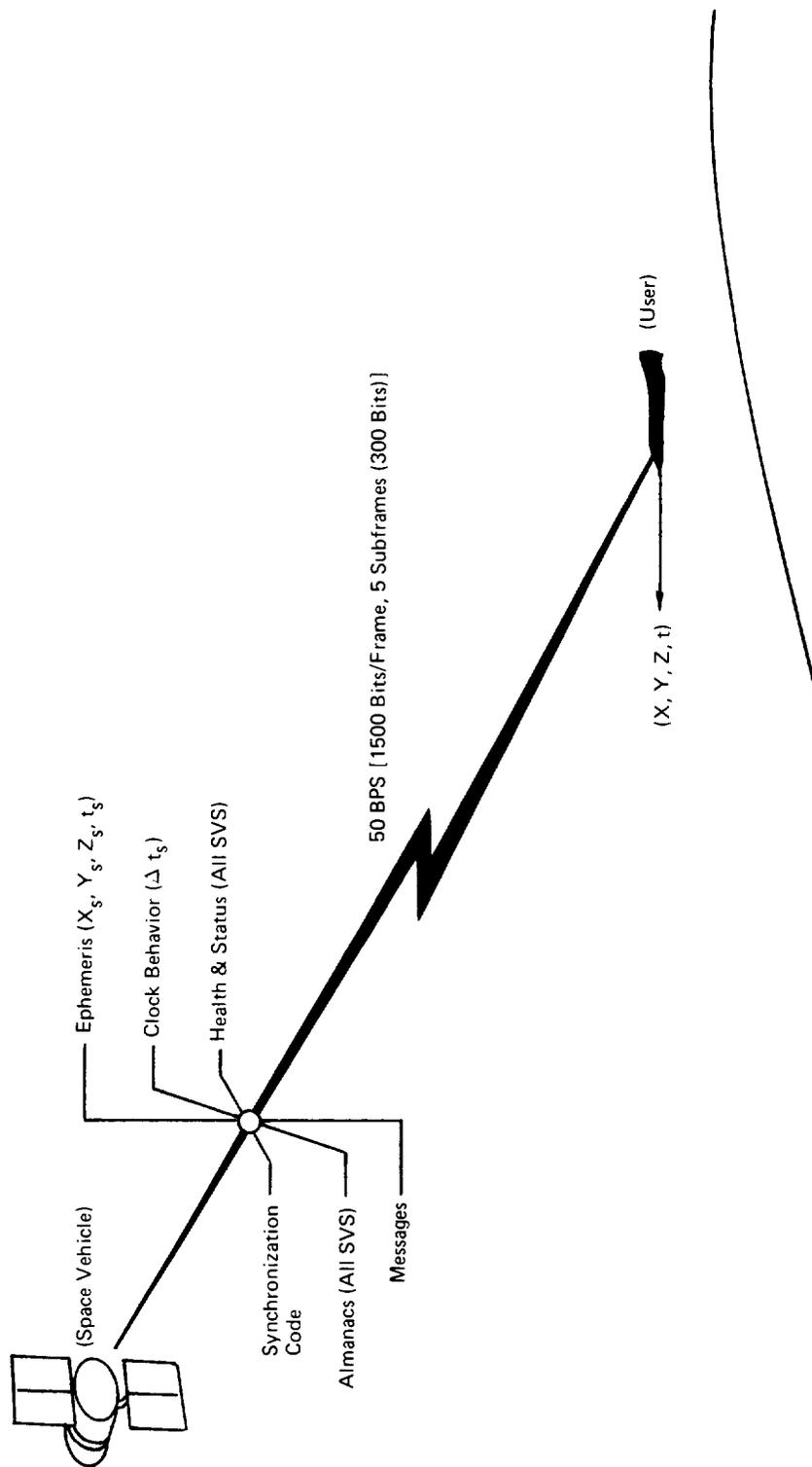


Figure 47. GPS Navigation Message Summary

The system time is in the form of the Z count. The preamble is a fixed 8-bit word that appears at the start of each subframe (every 6 seconds) which also coincides with an X1 code epoch and occurs at the Z count transmitted in the previous HOW word. The subframe identification is also in the HOW word.

This is all the timing information required by the User for normal acquisition on "C/A". Once he has obtained synchronization with the data stream and received the Z count, he is able to transfer to the "P" signal tracking and collect data required for navigation.

Direct acquisition on the "P" signal requires the User have a priori information on the SV signal to be acquired. All SV data streams thus contain information on all SV positions and clocks (almanacs). This information appears in subframe 5 of the data stream, providing almanacs on the SV's on a rotating basis. The clock information required in the almanac is the SV's time offset and drift, which provides the offset to within an accuracy dictated by derived requirements.

5.2 User Position Processing

The User equipment consists of an antenna, receiver, and processor. The functional system operational flow is shown in Figure 48. The processor determines the optimum GPS satellites for processing by the User. The satellite signal is acquired and the data message recovered by the receiver. Pseudorange and range rate are determined by the receiver and passed over to the processor where the User position and velocity are determined.

Figure 49 illustrates the functional flow for a single channel GPS receiver. The receiver is implemented using dual correlator channels.

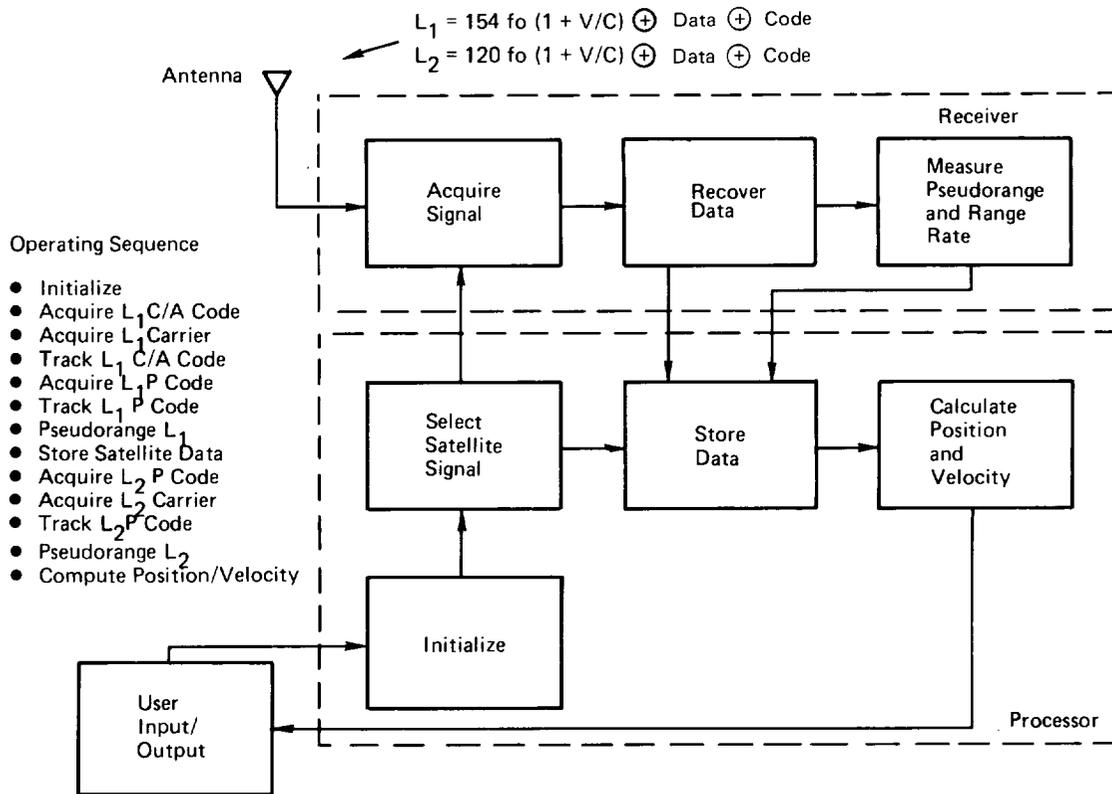


Figure 48. Functional System Operation

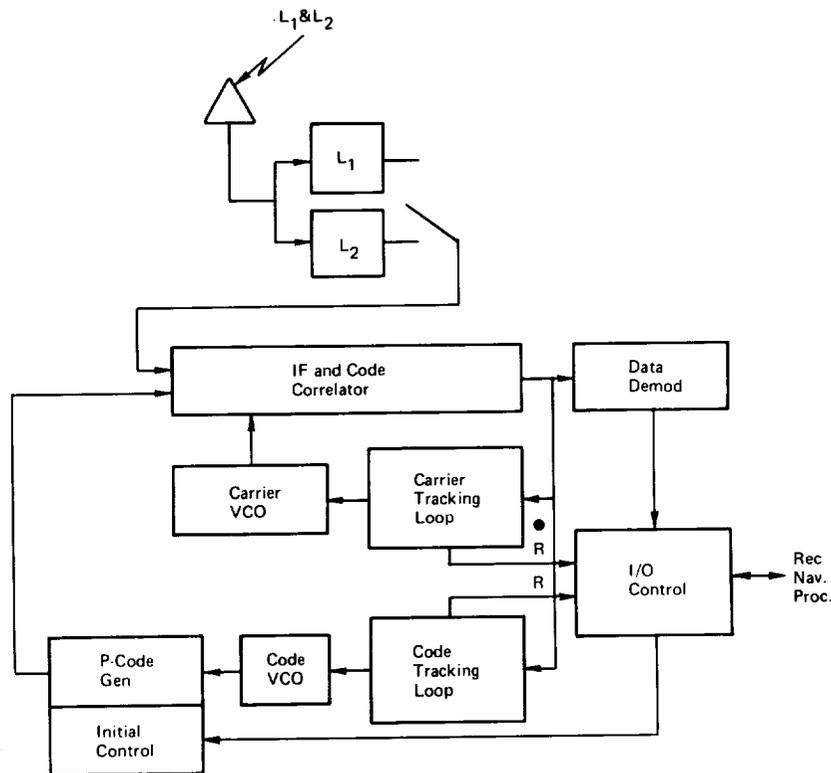


Figure 49. Single Channel GPS Receiver Functional Diagram

This allows, except in a high interfering signal level environment, the code- and carrier-tracking loops to operate on independent channels. The heart of the receiver is the code generator and digital VCO's which control the code position and carrier frequency.

The normal mode processing sequence is as follows:

- 1) Frequency-lock to carrier, and phase-lock to C/A code.
- 2) Phase-lock to carrier
- 3) Bit-synchronize to SV data (search for sign reversals)
- 4) Frame-synchronize to SV data (search for identifiable bit pattern in TLM)
- 5) Obtain P-code handover word (HOW)

- 6) Phase-lock to P-code
- 7) Pseudorange and delta pseudorange measurement
- 8) Pseudorange corrections
 - Ionospheric
 - Tropospheric
 - Clock bias.
- 9) User position and velocity computed.

The User processor, when provided with a valid almanac, an approximate knowledge of its own position and time-of-day, can compute a satellite selection algorithm. The processor then designates to the receiver not only the GPS satellite to be acquired, but an estimate of the doppler shift on the signal. The block diagram of Figure 50 shows a coherent correlation which correlates the incoming signal against a local replica consisting of the chosen C/A sequence, $X_h(t)$, modulated on the receiver local oscillator. The time phasing of the C/A sequence is varied slowly until the post-correlation power exhibits a use above that which might be attributable to noise alone. Having established synchronization of the pseudo-noise sequence, tracking is begun in both code and carrier.

The essential elements of the code tracking loop is a correlator shown in Figure 51. A received code is multiplied by a reference code time effort by $T < T$ where T is a code chip interval. The multiplier output V_3 is averaged by a low-pass filter having an integration time $T_m = 1/B \gg T$, i.e., much greater than the chip interval. The correlation output itself is not sufficient for code tracking since it does not provide an indication of the sign of the delay error of a tracking reference signal.

The sign of the delay error is determined as shown in Figure 52. The outputs V_1 & V_2 of early and late correlation ($\frac{1}{2}$ chip) are subtracted to form a correlation signal, $V_3(T)$, which is then used to drive a

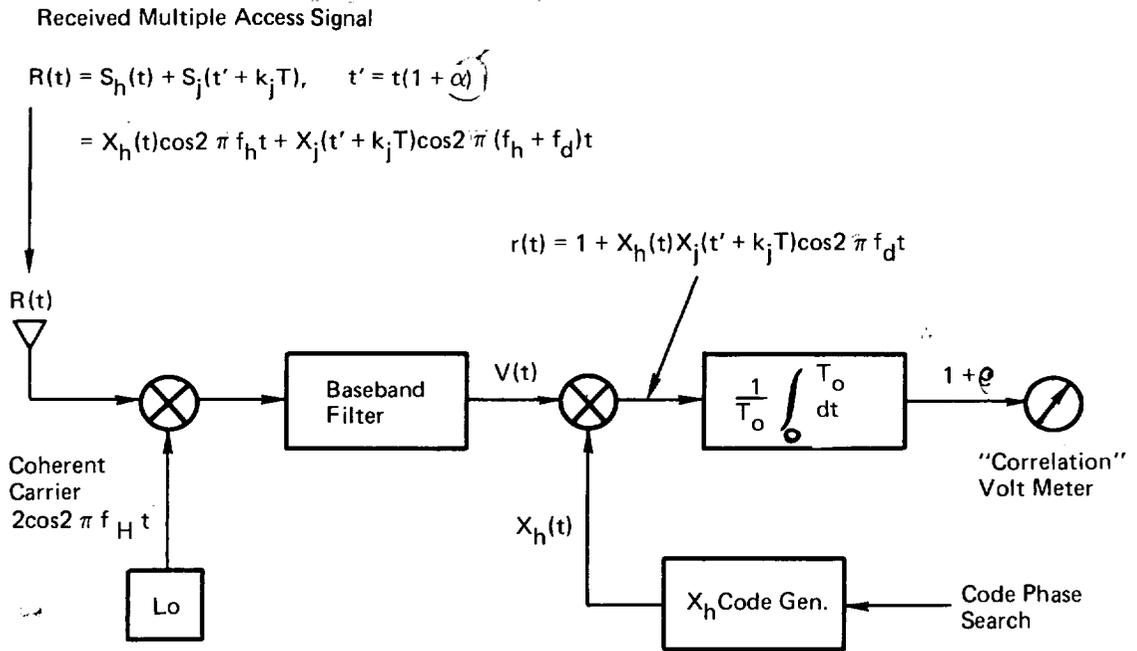


Figure 50. Received Multiple Access Signal

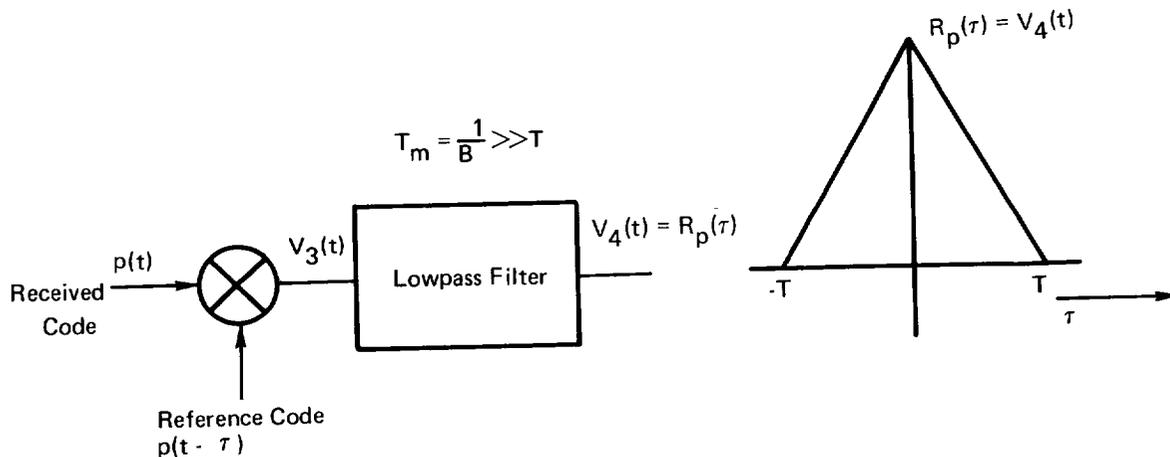


Figure 51. Essential Elements of a Delay-Lock Loop (DLL) is the Correlator

voltage-controlled-oscillator (VCO) or clock. This clock in turn drives the PN generator in such a manner that if the clock is lagging in phase, the correlation signal, V_3 , drives the clock faster and the reference code speeds up. The receiver also contains a coincident channel shown in the top portion of the block diagram.

The actual received signal, of course, arrives at the receiver at RF and has data modulation in addition. A coherent carrier for the down-conversion operation can be generated as shown in Figure 53.

After carrier recovery the recovered baseband code $P(t)$ can be fed to the coherent delay-lock loop for code tracking. Once the tracking loops pull in, the data format features (bit edges, word starts, subframe starts

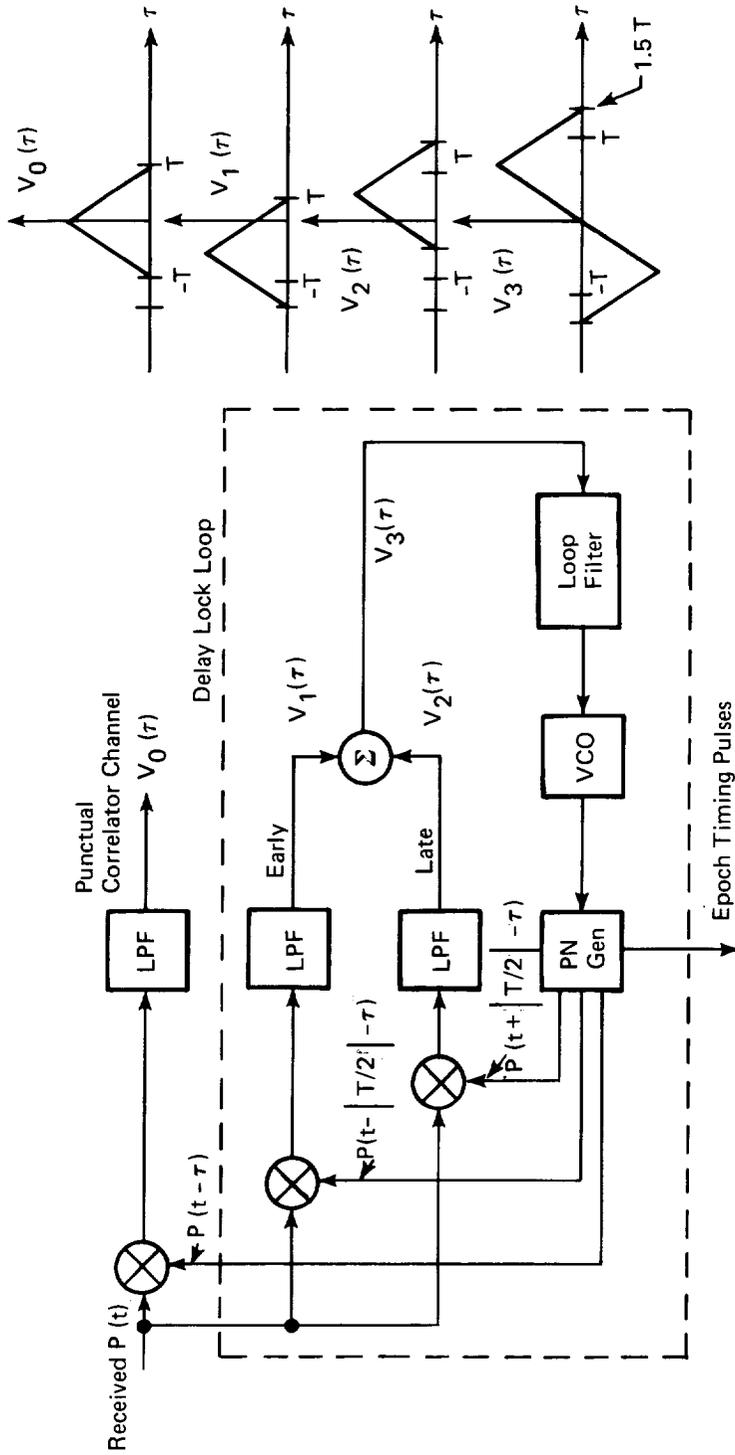


Figure 52. Coherent Delay Lock Loop

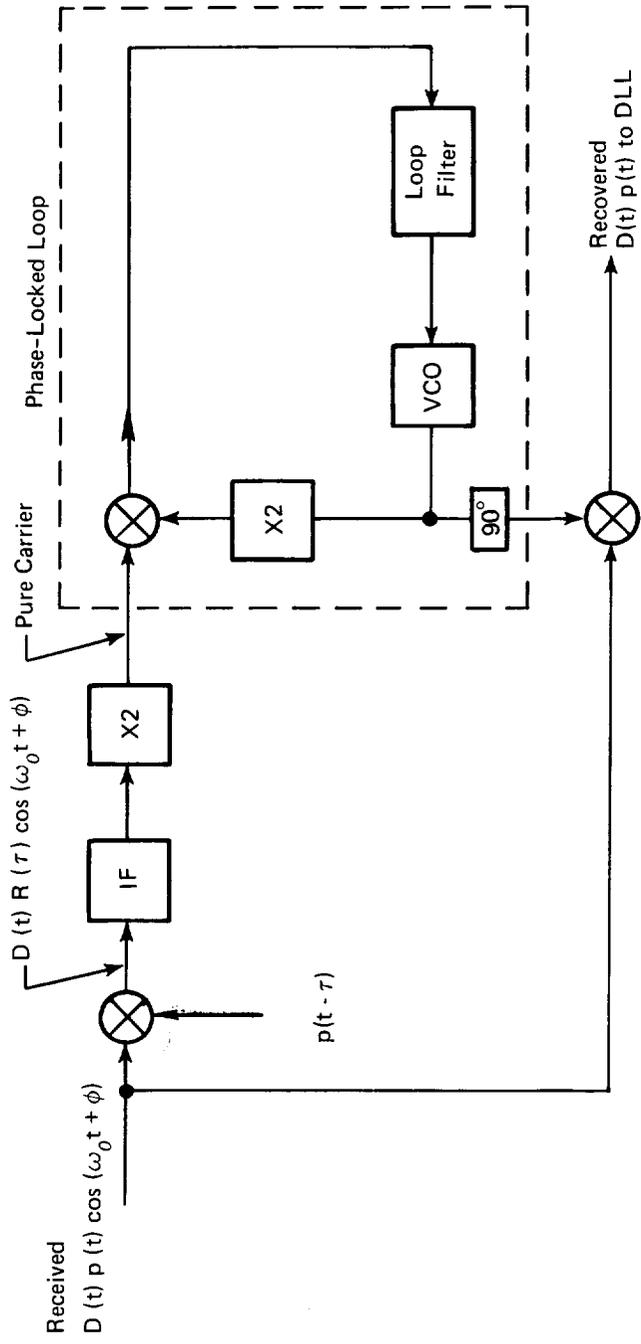


Figure 53. Carrier Recovery of the Received Signal

and Z counts) may be recognized to provide unambiguous time-of-arrival and time-of-day indication. While the almanac is sufficiently accurate for acquisition, much more accurate information on satellite position and the offsets of its clock is needed to achieve the desired navigation accuracy. These data, called the ephemeris, are contained in about twenty 24-bit words which are part of the data format transmitted by each satellite. Having recognized the format identifiers, the ephemeris can be recovered.

Measurements

The User receiver maintains a time reference used to generate a replica of the code transmitted by the GPS satellite. The amount of time skew the receiver must apply to correlate the replica with the code received from the satellite provides a measure of the signal propagation time between emitter and User. This time of propagation is called the pseudo-range measurement since it is in error by the amount of time synchronization error between the emitter and receiver clocks (bias error). The receiver also measures the doppler shift of the carrier signals from the emitter. By measuring the accumulated phase difference in this doppler signal over a fixed interval, the receiver can infer the range change increment. This measurement is called the delta pseudo-range measurement and is in error by an amount proportional to the relative frequency error between the emitter and receiver clocks. Since the carrier wave-length is short, the pseudo-delta range is a finely quantized measurement.

The satellites also transmit precise ephemeris and satellite clock calibration data. The User equipment is thus able to obtain measures of pseudo-range and delta-range reception of these measurements, ephemeris data and emitter clock calibration data.

These measurements, pseudo-ranges, are next adjusted for propagation effects. The delay in traversing the ionosphere is best estimated by observing the difference in arrival time of the P signals on the L1 and L2 frequencies. Tropospheric delay also requires correction. The use of a simple altitude and elevation angle dependent model is sufficient to reduce the error to negligible levels.

The delay compensated pseudo range measurement is then corrected for the satellite clock offset. The coefficients for the clock offset correction are part of the transmission from satellite to User.

The position of the satellite is computed from the ephemeris parameters received from the satellite using a Keplerian ellipse with some correction terms for oblateness of the earth and rotation of the orbit plane. At each time of pseudorange measurement, the corresponding position of the satellite is computed by inserting the value of time in the equations of the corrected ellipse.

At this point, there are 4 unknowns (three coordinates of User position and User clock time offset) and four eqn's. each involving the measured, compensated pseudo-range. The equations are nonlinear but capable of solution by a number of techniques. The usual formulation is to make the position solution part of a Kalman filtering operation.

5.3 Geometric Dilution of Precision (GDOP)

The accuracy with which one can measure position and time is related to the accuracy in radial range measurement by factors known as the GDOP or Geometric Dilution of Precision. Referring to Figure 54 GDOP is developed in the following manner. Shown on the figure is a User position x , y , and z and the position of GPS satellite No. X_1 , Y_1 , and Z_1 . The range distance between the User and the GPS satellite is shown as R_1 .

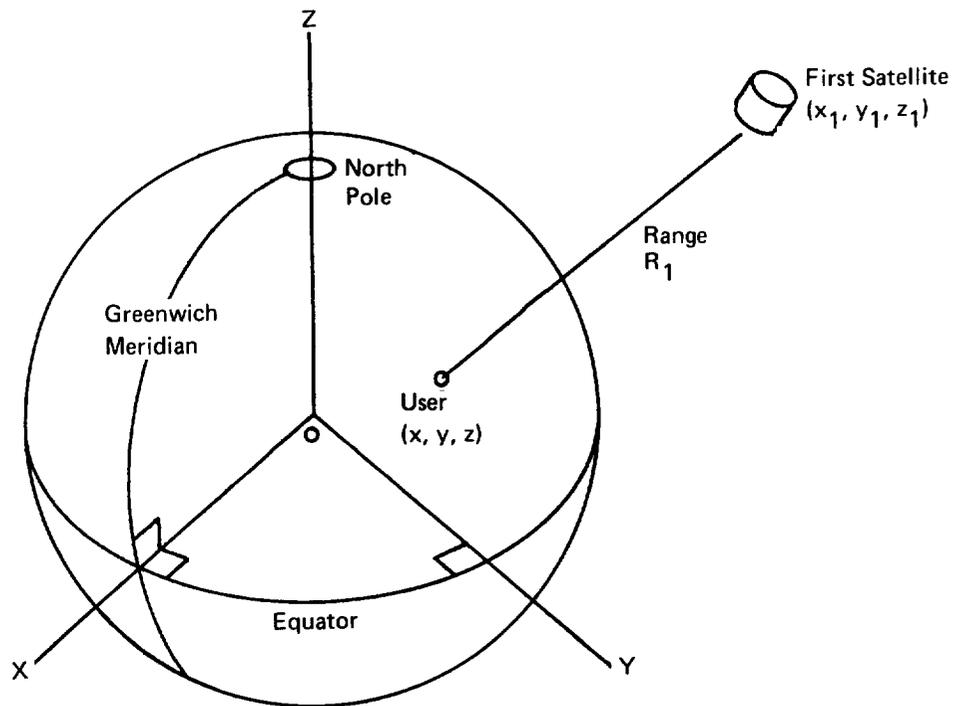


Figure 54. Earth Centered Coordinates

The basic equation is then as follows:

$$\left\{ (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 \right\}^{1/2} + T = R_i$$

where x, y, z are user position

T is user clock bias

x_i, y_i, z_i are SV positions ($i = 1, \dots, 4$)

R_i is the pseudorange measurement.

The above equations are nonlinear and although it is possible to solve the equations directly as shown, user equipments employ a linearized version of these equations.

The basic navigation equations can be linearized by employing incremental relationships resulting in the following matrix notation.

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & 1 \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & 1 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 1 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & 1 \end{bmatrix} \times \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta T \end{bmatrix} = \begin{bmatrix} \Delta R_1 \\ \Delta R_2 \\ \Delta R_3 \\ \Delta R_4 \end{bmatrix} \quad (36)$$

where the known quantities of the right-hand side of the equation are the incremental pseudo-range measurements or the differences between the actual measured pseudo-ranges and the measurements predicted by the user's computer based on the knowledge of satellite position and the User's most current estimate of his position and clock bias. The quantities to be computed, Δx , Δy , Δz , and ΔT , are corrections that the user will make to his current estimate of position and clock biases. The α_{ij} values are the direction cosines of the angle between the range to the i^{th} satellite and the j^{th} coordinate.

Now, let

r = the four element pseudorange measurement difference vector

x = the user position and time correction vector

A = 4 x 4 solution matrix.

Then

$$Ax = r \quad \text{or} \quad x = A^{-1}r. \quad (37)$$

The above relationship is linear and can be used to express the relationship between the errors in pseudo-range measurement and the user quantities.

Thus,

$$\epsilon_x = A^{-1}\epsilon_r \quad (38)$$

Where ϵ_r is the pseudorange measurement error

ϵ_x the errors in user position and clock bias.

The covariance matrices are given by

$$\text{Pseudorange:} \quad \text{COV}(r) = E \left\{ \begin{matrix} \epsilon_r & \epsilon_r^T \end{matrix} \right\} \quad (39)$$

$$\text{User Position \& Time:} \quad \text{COV}(x) = E \left\{ \begin{matrix} \epsilon_x & \epsilon_x^T \end{matrix} \right\}$$

Where $E \left\{ \quad \right\}$ designates "expected value" of the quantity inside the braces.

Upon substitution, the matrix relationship between the two covariance matrices becomes:

$$\text{COV (x)} = \text{A}^{-1} \text{COV (r)} \text{A}^{-\text{T}} \quad (40)$$

or
$$\text{COV (x)} = \left[\text{A}^{\text{T}} \text{COV (r)}^{-1} \text{A} \right]^{-1}$$

It can be seen from the above expression that the error relationships are a function only of satellite geometry. An important user consideration is that the geometry of the four satellites used possess good geometric properties. This leads to the concept of "Geometric Dilution of Precision": it is a measure of how satellite geometry degrades accuracy.

The "Geometric Dilution of Precision" (GDOP) is defined as the square root of the trace of COV (x) when COV (r) is the identity matrix (i.e. 1σ error of unity and expected mean is zero).

Therefore,

$$\text{GDOP} = \left\{ \text{Trace} \left[(\text{A}^{\text{T}}\text{A})^{-1} \right] \right\}^{1/2} \quad (41)$$

Where

$$(\text{A}^{\text{T}}\text{A})^{-1} = \begin{bmatrix} \sigma_{\text{xx}}^2 & \sigma_{\text{xy}}^2 & \sigma_{\text{xz}}^2 & \sigma_{\text{xT}}^2 \\ \sigma_{\text{yx}}^2 & \sigma_{\text{yy}}^2 & \sigma_{\text{yz}}^2 & \sigma_{\text{yT}}^2 \\ \sigma_{\text{zx}}^2 & \sigma_{\text{zy}}^2 & \sigma_{\text{zz}}^2 & \sigma_{\text{zT}}^2 \\ \sigma_{\text{Tx}}^2 & \sigma_{\text{Ty}}^2 & \sigma_{\text{Tz}}^2 & \sigma_{\text{TT}}^2 \end{bmatrix} \quad (42)$$

Some properties of GDOP can be summarized as follows:

1. GDOP is, in effect, the amplification factor of pseudo-range measurement errors into user errors due to the effect of satellite geometry.
2. GDOP is independent of the coordinate system employed.
3. GDOP is a criterion for designing satellite constellations.
4. GDOP is a means for user selection of the four best satellites from those which are visible.

By letting σ_x , σ_y , σ_z , σ_T be the variances of user position and time, we have

$$\text{GDOP} = (\sigma_x^2 + \sigma_y^2 + \sigma_z^2 + \sigma_T^2)^{1/2} \quad (43)$$

As an alternative to GDOP as a criteria for selecting satellites or evaluating satellite constellations only some of the variances of user position and time might be used. These are defined as follows:

PDOP The square root of the sum of the squares of the three components of position error.

$$\text{PDOP} = (\sigma_x^2 + \sigma_y^2 + \sigma_z^2)^{1/2} \quad (44)$$

HDOP The square root of the sum of the squares of the horizontal components of the position error.

$$\text{HDOP} = (\sigma_x^2 + \sigma_y^2)^{1/2} \quad (45)$$

VDOP The altitude error, σ_z^2

$$\text{Note: } PDOP^2 = HDOP^2 + VDOP^2 \quad (46)$$

TDOP The error in the user clock bias multiplied by the velocity of light, σ_T^2

$$\text{Note: } GDOP^2 = PDOP^2 + TDOP^2 \quad (47)$$

The alternative criterion most frequently used is PDOP, The "Position Dilution of Precision". PDOP is also invariant with the coordinate system and is used because the most important consideration in any navigation system is position accuracy; knowing time is generally a secondary by-product. Another alternative is HDOP, "Horizontal Dilution of Precision," which is most meaningful for users who are using the system primarily to obtain only horizontal position.

Figure 55 illustrates the various GDOP factors vs. the cumulative probability of achieving the given GDOP or lower. The values shown are for a 5° elevation mask, i.e. only satellites in the 24 satellite constellation above 5° elevation angle are assumed to be in view. Clearly one has a high probability of a PDOP of 3 or less. Thus if one is to have a 10 meter accuracy goal the desired accuracy in range measurement should be on the order of $(1/3)$ 10 meters or roughly $\sigma_r < 10$ n sec.

5.3.1 Ranging Accuracy

The user position accuracies will be a function of :

- The uncertainties in satellite ephemeris and clock bias
- Atmospheric delay

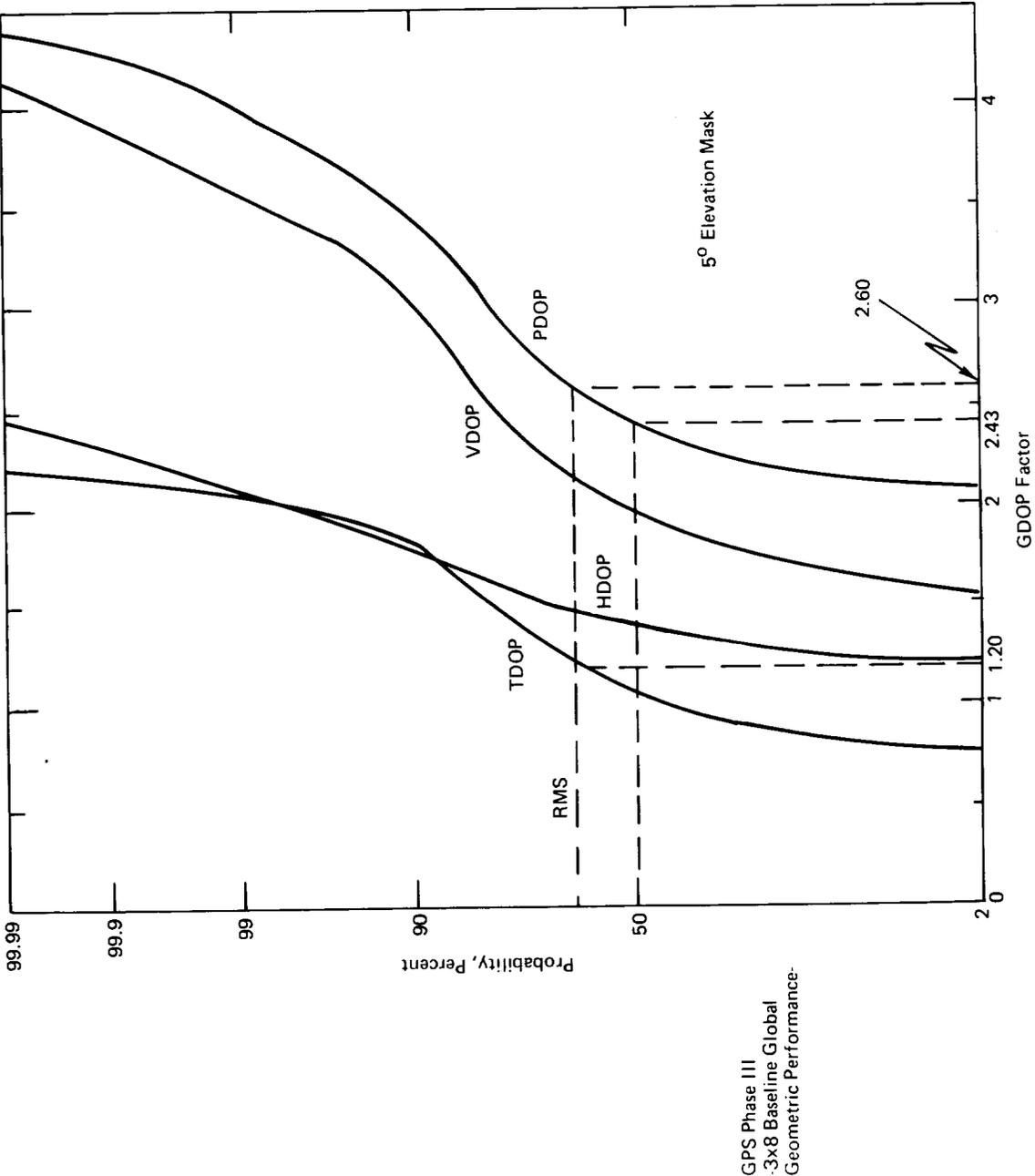


Figure 55. GDOP vs. The Cumulative Probability of a GDOP Less Than the Value Shown, (From Bogen, 1974)

- Satellite group delay
- Receiver noise and resolution
- Multipath.

The orbit determination process derives progressively refined information defining the gravitational field influencing the spacecraft motion, solar pressure parameters, the locations, clock drifts, and electronic delay characteristics of the ground stations, and other factors found to be significant.

Satellite position errors are, however, still on the order of several meters and would appear to be too large for achieving predicted user accuracies. However, it is the ranging error that is important. The factors making this error sufficiently small are twofold: projection geometry and correlations. The in-track and cross-track errors will be projected onto the satellite/user line through small angles for satellites at GPS altitudes (4 m in-track projects to 1 m or less in ranging error). The primary error is then from the projection of the radial position uncertainty. Satellite error correlations reduce the effective error as follows: errors common (deterministically or statistically) to the four range measurements will cause a user clock calibration error but not a user position error. These four radial position and clock errors are highly correlated between measurements, due primarily to the common effect of uncertainties in the earth's gravitational constant. Furthermore, for each satellite the residual radial and satellite clock errors have near unity negative correlation, again primarily due to uncertainties in the earth's gravitational constant. The combined effects of the projection geometry and the various correlations will reduce the equivalent ranging error from the satellite ephemeris and clock uncertainties to about 1.50 meters.

During transit through the ionosphere an RF signal will experience a time delay due to a reduction of the speed of the signal propagation and the bending of the ray, both effects arising from refraction. The ionospheric time delay is essentially inversely proportional to the square of the frequency. Transmission of coherent L1 and L2 signals will therefore permit calibration of the delay to sufficient accuracy.

The ionospheric delay model is

$$\Delta R = \frac{ck}{f^2}$$

where c is the speed of light

k is a constant scale factor

f is the frequency of the rf carrier.

The mechanization uses the difference between L1 and L2 pseudorange measurements to estimate k .

The tropospheric delay is independent of frequency. Studies have shown that the delay can be modeled by a fairly simple algorithm of the following form

$$\Delta R = \frac{2.4224}{0.026 + \sin E} \text{ Exp } \left[-0.13346 h \right] \quad (49)$$

where h is the altitude (km) of the user

E is the surface reflectivity.

This effect, together with errors in modeling the tropospheric delay, results in a total atmospheric delay (ionosphere plus troposphere) error of 2.4 to 5.2 meters.

The satellite group delay error is defined as the summation of delay uncertainty due to effects in the space vehicle such as uncalibrated delay in signal equipment. One meter uncertainty has been budgeted to these satellite induced effects.

Receiver noise and resolution errors in the user navigation receiver hardware and software perturb the correct navigation solution. With a high performance, simultaneous 4 or 5 channel receiver, under typical user dynamics and signal-to-noise conditions, the total error contribution from the receiver is about 1.50 meters.

Multipath error is introduced by the combination of several propagation paths from SV to the user which corrupt the measurement of line-of-sight distance. The magnitude of the resulting ranging error depends strongly on the location and nature of reflecting surfaces in the user environment. The expected range of the multipath error is 1.2-2.7 meters.

The previously discussed error sources that contribute to the pseudo-range measurement errors are summarized in Table 8. The total expected rms 1σ error is 3.6 to 6.3 meters. This level of performance is expected for the fully developed Phase III operational system for users having the best quality dual frequency equipment.

Also shown in Table 8 is the performance level expected for users having stable orbits allowing some error sources to be reduced by smoothing. This results in an expected error of 2.0 meters.

The expected error is the "user equivalent range error" (UERE) in that it is based on the assumption that there is no correlation between satellite measurement errors.

Table 8. System Error Budget

<u>Sources of Error</u>	<u>UERE¹ (Meters)</u>	<u>Smoothed</u>
Satellite Ephemeris & Clock	1.5	1.5
Atmospheric Delay	2.4-5.2 ²	0.5
Satellite Group Delay	1.0	1.0
Receiver Noise & Resolution	1.5 ²	0.5
Multipath	<u>1.2-2.7²</u>	<u>0.5</u>
RSS	3.6-6.3	2.0

1 User Equivalent Range Error

2 Error can be smoothed

5.3.2 User Navigation Error

The User navigation accuracy (UNE) is normally estimated as the UERE times GDOP where GDOP is uniquely established by the geometric relationship between the User's position, and the specific positions of the four satellites utilized for the observations. The optimum GDOP occurs when one satellite is at the

user's zenith and the other three are separated by 120° and are as low on the horizon as permitted by the user's antenna elevation angle.

Figure 55 illustrated the GDOP values for cumulative proportions of user's evenly distributed over the world and around the clock who select the best 4 satellites from those that are visible 5 degrees or more above the horizon.

The values of PDOP for cumulative proportions of users evenly distributed over the globe and around the clock who select the best 4 satellites from those that are visible 5 degrees or more above the horizon. The rms value of PDOP is in the neighborhood of 2.6 which, when combined with the range errors of from 3.6 to 6.3 meters, gives user three-dimensional position errors of from 9.4 to 16.4 meters, 1σ . The horizontal component of the position error reflected by HDOP is usually less. The corresponding rms value of HDOP is about 1.45, which yields horizontal position errors of from 5.2 to 9.1 meters, 1σ .

The corresponding rms value of TDOP is about 1.2 which, when range is converted to time, yields a 1σ time error of from 14 to 25 nanoseconds.

The value of GDOP itself is a composite measure that reflects the influence of satellite geometry on the combined accuracy of the estimate of user time (user clock offset) and user position.

$$\begin{aligned} \text{GDOP} &= (\text{PDOP}^2 + \text{TDOP}^2)^{1/2} \\ \text{GDOP} &= \{(2.6)^2 + (1.2)^2\}^{1/2} = 2.86 \end{aligned}$$

This gives the user navigation error (UNE) as 10.3 meters to 18 meters.

Table 9 summarizes the expected UNE values for both the nominal UERE low value (discussed above) and the optimistic smoothed UERE value of 2 meters.

Table 9. User Navigation Error (UNE) (Meters)

GDOP Factor	ONE	
	Nominal UERE	Smoothed UERE
PDOP	9.4	5.2
HDOP	5.2	2.9
GDOP	10.3	5.7

$$\text{UNE} = \text{PDOP} \times \text{UERE}$$

5.4 Relative Pixel Location

Two tables, generated as part of the first phase of this study, are reproduced here to indicate the expected pixel location error when using GPS position and time estimates in conjunction with best estimate attitude errors and other expected S/C error sources. Run No. 5 in Table 10 lists the input error source values with Table 11 illustrating the resulting maximum error.

The results of run No. 5, shown in Table 11, indicate that the location of a pixel on the ground is known to within 1.22 equivalent pixels (relative to a 30 meter TM pixel size) in the Δy direction and about 5 equivalent pixels in the Δx direction. As previously stated, the larger Δx error is due to the effect of S/C velocity errors. The run utilized a conservative GPS position error estimate of 10 meters in the along-track, cross-track, and radial directions with an optimistic attitude error estimate of 0.00148 degrees in each axis. As shown in Table 9 the User Navigation Error (three dimensions) using the GDOP factor and nominal UERE value is expected to be 10.3 meters total (all three axes).

Augmenting these results with a sparse field of ground control points (GCP's) will allow very accurate pixel position estimates (1 pixel error or less) to be attained.

Table 10. Error Source Values for the Simulation

ERROR SOURCE

Run No.	Attitude (degree)							Ephemeris (m)		
	Mirror Scan (rad)	Timing (sec)	S/C Velocity (m/sec)	ρ_{ij}	Roll	Pitch	Yaw	Along Track	Cross Track	Radial
1	1×10^{-5}	1×10^{-3}	7.5	0	0.01	0.01	0.01	34.3	19	16.1
2	2×10^{-5}	1×10^{-3}	7.5	0	0.079	0.079	0.079	34.3	19	16.1
3	1×10^{-5}	1×10^{-3}	7.5	0	0.01	.01	0.01	10	10	10
4a	0	0	0	0	0.001	0.001	0.001	0	0	0
4c	0	0	0	0	0.0014	0.0014	0.0014	0	0	0
4f	0	0	0	0	0.00148	0.00148	0.00148	0	0	0
5	1×10^{-5}	1×10^{-3}	7.5	0	0.00148	0.00148	0.00148	10	10	10

Table 11. Maximum Error Over 90 Percent Grid

Run No.	ΔX		ΔY	
	Max Value (m)	Equiv. Pixels	Max Value (m)	Equiv. Pixels
1	256.67	8.56	208.36	6.95
2	1619.59	54.00	1624.91	54.16
3	250.93	8.36	206.64	6.89
4a	20.41	0.68	20.56	0.69
4c	28.57	0.95	28.79	0.96
4f	30.20	1.01	30.43	1.01
5	149.11	4.97	36.60	1.22

5.5 Satellite Selection

Based on the user's position estimate, a search is made in the user's navigation computer to determine those GPS satellites in view. Only those satellites above a masking elevation angle of 5° are selected, since satellite signal power rapidly attenuates due to atmospheric delays and noise at the lower elevations. Typically, from 5 to 9 satellites are always in view above 5° . Two approaches can be taken to select the 4 satellites for navigation: (1) a suboptimal approach based on GDOP and (2) an optimal GDOP criteria. The suboptimal algorithm first selects those 3 satellites having maximum range vector components in the vertical, east, and north directions. A fourth satellite is selected from the remaining satellites in view which produces minimum GDOP. The advantage of the suboptimal approach is

computational efficiency since only $n-3$ computations of GDOP are required where n is the number of satellites in view.

The optimal procedure will require $\binom{n}{4}$ computations of GDOP to select those 4 satellites which provide the absolute minimum GDOP. The required frequency of satellite selection is a function of time and User velocity to insure that the satellites remain in view and the GDOP remains good. It is expected that this computation will occur about every 15 minutes.

5.6 Tracking and Data Relay Satellite System (TDRSS)

An alternate technique for obtaining satellite position is to utilize data from the geosynchronous relay satellite system such as TDRSS. The TDRSS approach makes use of one-way range and doppler data along with a time transfer mechanism to determine satellite position and time.

TDRSS consists of two operational satellites placed in geosynchronous orbits. The satellites will be positioned 130 degrees apart in longitude and placed such that they are in constant communication with a ground terminal located at White Sands, New Mexico.

The fundamental differences between GPS & TDRSS are summarized as follows:

- The GPS system transmits a self-contained coded message directly from each NAVSTAR to the user satellite. The TDRSS system will relay ephemeris and timing information from White Sands through TDRSS to the User satellite.

- The NAVSTAR satellites transmit on 2 L-band frequencies whereas the general TDRS User would receive S-band data.
- Due to the dynamics of the GPS orbits and the number of GPS satellites, the measurement geometry provided to the user satellite by GPS will be superior to that provided by TDRSS.
- The data gaps of the GPS are smaller than the data gaps occurring with the TDRSS.

TDRSS provides range and range rate tracking to each user with an accuracy comparable to that currently available from the ground-based network. This tracking information is not comparable to that obtainable with GPS, being at best only equal to the coarse accuracy (100 meters) of the GPS system. It does not, however, require any additional equipment on-board the satellite beyond the TDRSS transponder/antenna system.

Current planning is for the data received at White Sands to be retransmitted via Domsat to GSFC for processing. However, as the trend for the processing function to be accomplished on-board increases, then the direct delivery of information from the satellite to the User can increase. The routing then would very likely be from the satellite to White Sands through Domsat directly to the User.

Although the TDRSS does not provide range and range-rate information as accurate as GPS, it could provide an interim capability for allowing verification of any on-board implementation requiring satellite position and time information. As the GPS becomes operational, this is not expected until 1987, the more accurate position and time information provided by GPS can be integrated into the on-board system.

5.7 Conclusions

GPS position and time information combined with spacecraft attitude will allow the ground position of a pixel to be computed on-board a spacecraft. As shown in Table 11, the location of a pixel on the ground is known to within 1.22 equivalent pixels (relative to a 30 meter TM pixel size) in the Δy direction. These results are based on a GPS position error estimate of 10 meters in the along-track, cross-track, and radial directions coupled with an attitude error estimate of 0.00148 degrees in each axis.

An alternate technique for obtaining the satellite position would be to utilize data from the TDRSS geosynchronous relay satellite. This positional information is not comparable to that obtained with GPS, being at best an order of magnitude less accurate (i.e., 100 meters versus 10 meters). TDRSS could provide an interim capability for allowing verification of any on-board implementation requiring satellite position and time. As GPS becomes operational, the more accurate position and time information provided by GPS can be integrated into the on-board system.

REFERENCES

1. Thompson, L. L., Remote Sensing Using Solid-State Array Technology. Photogrammetric Engineering and Remote Sensing, Vol. 45, No. 1, January 1979, pp. 47-55.
2. Final Report for Breadboard Linear Array Scan Imager Program. Contract NAS5-21806, Westinghouse Defense and Electronics Systems Center, 25 April 1975.
3. Advanced Scanners and Imaging Systems for Earth Observations. NASA SP-335, 1972.
4. Barbe, D. F., Imaging Devices Using The Charge-Coupled Concept. Processings of the IEEE, Vol. 63, No. 1, January 1975.
5. Jobson, D. J., On-Board Compensation for Temporal Changes in Pushbroom Scanner Detector Arrays. AIAA/NASA Conference on "Smart" Sensors, November 1978.
6. Wolfe, R. H., Jr., Cheru, W. S., Data System New Technology Requirements for Shuttle Era Payloads. Contract NASA9-14350, IBM Federal Systems Division, Houston, Texas, September 1977.
7. Anderson, G. A., Jensen, E. D., Computer Interconnection Structures: Taxonomy, Characteristics, and Examples. Computing Surveys, Vol. 7, No. 4, December 1975.
8. Lambeck, P. F., Implementation of the XSTAR Haze Correction Algorithm and Associated Preprocessing Steps for Landsat Data. ERIM Memo No. IS-PFL-T272, ERIM, Ann Arbor, MI., 18 March 1977.

9. Kauth, R., Hemdal, J., Landsat 1 to Landsat 2 Data Conversion. ERIM Memo No. IS-RK/JH-1216, 7 February 1977.
10. Rice, D., Calibration of Landsat 2. ERIM Memo No. IS-DR-935, 13 December 1976.
11. Partitioning Requirements for Signature Extension. ERIM Technical Memo 122700-13-R, ERIM, Ann Arbor, MI, October 1976.
12. Navigation Planning -- Need for a New Direction. Report to the Congress, GAO Report No. LCD-77-109, 21 March 1978.
13. Department of Transportation National Plan for Navigation. November 1977.
14. System Specification for the NAVSTAR Global Positioning System, Phase I. SS-GPS-1018, 15 April 1974.
15. Jorgensen, P. S., Normalized Accuracy Analysis of the NAVSTAR/GPS. Aerospace Report No. TOR-0078 (3475-10)-2, 28 February 1978.
16. Spilker, J. J., Jr., Global Positioning System Signal Structure and Performance Characteristics, STI-TM-8802, 28 February 1978.
17. Navigation, GPS Special Issue, Vol. 25, No. 2, Summer 1978.
18. Van Dierendonck, A. J., Melton, W. C., Bernbaum, M., and Harkins, M. D., The Approach to Satellite Ephemeris Determination for the NAVSTAR Global Positioning System. Navigation, the Journal of the Institute of Navigation, Vol. 23, No. 1, pp. 76-86, Spring 1976.