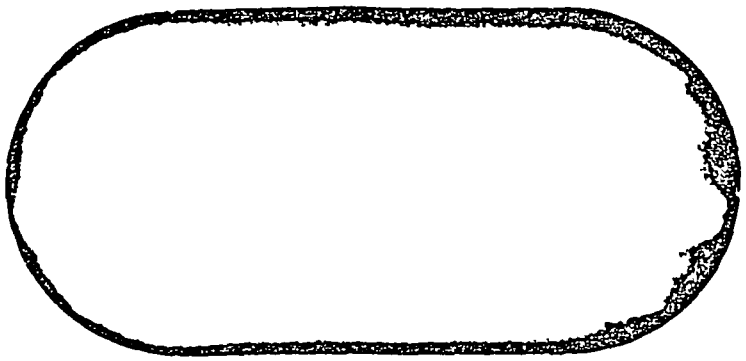
RETURN TO
Joseph J. Gitelman
GSFC Code 402



(NASA-CR-158842) ON-BOARD ATTITUDE DETERMINATION SYSTEM (OADS) Final Study Report (Martin Marietta Aerospace, Denver, Colo.) 277 p HC A13/MF A01 CSCL 22B

N79-28222

Unclas
G3/18 32008

ON-BOARD ATTITUDE DETERMINATION SYSTEM (OADS) FINAL STUDY REPORT

APRIL, 1978

Contract NAS5-23428 Mod. 27

Prepared By:

P. Carney
M. Milillo
V. Tate
J. Wilson
K. Yong

MARTIN MARIETTA AEROSPACE
Denver Division
Post Office Box 179
Denver, Colorado 80201

TABLE OF CONTENTS

	PAGE
LIST OF TABLES	v
LIST OF FIGURES	viii
ACRONYMS	x
FOREWORD	xii
1.0 INTRODUCTION	1-1
2.0 OADS SYSTEM DESIGN CONCEPT SUMMARY	2-1
2.1 OADS System Sensor and Performance	2-1
2.2 On-Board Computation Requirements	2-1
3.0 OADS ORBIT DESCRIPTION AND ATTITUDE DETERMINATION	3-1
3.1 MMS Mission Selected Orbits and Related Features	3-1
3.1.1 Earth Mission	3-1
3.1.2 Stellar Mission	3-1
3.1.3 Solar Mission	3-2
3.2 Coordinate System Definition	3-2
3.3 Attitude Angle Definition and OADS Output Definition	3-9
3.3.1 Attitude Angle Definitions	3-9
3.3.2 OADS Output Definitions	3-15
4.0 OADS SENSOR EVALUATION	4-1
4.1 Inertial Reference Unit	4-1
4.1.1 Description of NASA Standard Unit - DRIRU-II	4-1
4.1.2 Trade Study	4-6
4.1.2.1 Trade Study Between DRIRU-II and Bendix Redundant Strapdown IRU	4-6
4.1.2.2 Trade Study Between SDG-5 and Other Dry Tuned Gyros	4-10
4.1.3 Conclusion of Trade Study	4-13
4.2 Star Tracker System	4-13
4.2.1 Description of NASA Standard Star Tracker (SST)	4-13
4.2.2 Trade Study	4-19

TABLE OF CONTENTS

		PAGE
4.2.2.1	Comparison with Star Mapper	4-19
4.2.2.2	Comparison with Other Fixed Head Star Trackers	4-20
4.2.2.3	Comparison with CCD Unit	4-21
4.2.3	Conclusion of Trade Study	4-22
4.3	GPS Magnavox Receiver/Processor Assembly	4-25
4.3.1	Global Positioning System Summary	4-25
4.3.2	Host Vehicle Command Interface to the Magnavox Receiver/Processor Assembly	4-26
4.3.3	GPS Magnavox Receiver/Processor Assembly Output to the Host Vehicle	4-27
5.0	MISSION PERFORMANCE STUDY	
5.1	Simulator Description	5-1
5.1.1	Inertial Measurement Unit Subsystem	5-1
5.1.2	Star Tracker Subsystem	5-4
5.2	Sensor Error Sources	5-6
5.2.1	Gyro Errors	5-6
5.2.2	Star Tracker Update Error Sources	5-8
5.2.2.1	Star Catalog Error	5-9
5.2.2.2	Star Tracker Errors	5-9
5.2.3	Global Positioning System Orbit Errors	5-10
5.3	Sensitivity Study	5-11
5.3.1	Maneuver Profiles	5-11
5.3.2	Sensitivity Due to Numerical Integration	5-13
5.3.3	Gyro Errors Sensitivity Analysis	5-14
5.3.4	Star Tracker Update Error Sensitivity Analysis	5-22
5.3.5	Orbit Errors Sensitivity Analysis	5-23
5.4	Nominal Performance Results	5-24
5.4.1	Nominal Earth Mission Performance Results	5-28
5.4.2	Nominal Stellar Mission Performance Results	5-29
5.4.3	Nominal Solar Mission Performance Results	5-36
5.4.4	Summary	5-36

TABLE OF CONTENTS - (Continued)

		PAGE
6.0	MICROPROCESSOR SOFTWARE ANALYSIS	
6.1	IRU Analysis	6-1
6.2	Star Tracker Analysis	6-10
6.3	IRU and Star Tracker Integration	6-19
6.4	Orbit Generator and Resolver Analysis	6-22
6.5	OADS Microcomputer System Integration	6-24
7.0	MICROPROCESSOR HARDWARE ANALYSIS	7-1
7.1	Analysis of Baseline Configuration Hardware	7-1
7.2	Available Fabrication Technology	7-2
7.3	Mass Storage Requirements	7-5
8.0	MICROPROCESSOR SYSTEM ANALYSIS	8-1
8.1	Supplementary Design Issues	8-1
8.2	Conclusions	8-3
9.0	NSSC-I SOFTWARE ANALYSIS	9-1
9.1	IRU Software Analysis	9-1
9.2	Star Tracker Software Analysis	9-1
9.3	Orbit Generator and Resolver Software Analysis	9-5
9.4	OADS NSSC-I System Integration	9-5
10.0	NSSC-I PERFORMANCE ENHANCEMENT APPROACHES	10-1
11.0	OADS TESTING, OPERATIONAL UPDATE REQUIREMENTS AND POST LAUNCH VERIFICATION	11-1
11.1	Testing Requirements	11-1
11.1.1	Hardware Components	11-1
11.1.2	OADS Closed Loop System	11-2
11.2	Operational Update Requirements	11-2
11.3	Post Launch Verification	11-6
12.0	RECOMMENDATIONS FOR FUTURE INVESTIGATIONS	12-1
12.1	Attitude Control Law Processing	12-1
12.2	NSSC-I and Bipolar Arithmetic Processing Unit	12-1
12.3	Replacement of SST with CCD Star Tracker	12-2
12.4	Replacement of Star Tracker with Landmark System	12-3

TABLE OF CONTENTS - (Continued)

		PAGE
APPENDIX I	Microprocessor Speficiations	I-1
APPENDIX II	Timing and Sizing Estimates of Common Routines and Algorithms Used in NSSC-I Analysis	II-1
APPENDIX III	Pertinent Reference Material	III-1

LIST OF TABLES

TABLE		PAGE
2.1	Summary of Estimated OADS Processing Time For Baseline Microcomputer System	2-5
2.2	Summary of Estimated OADS Processing Times for NSSC-I Computer	2-5
4.1	DRIRU-II System Evaluation	4-2
4.2	First Difference Drift Rates	4-7
4.3	Gyro Error Characteristics Comparison	4-9
4.4	Comparison of Physical Characteristics and Reliability	4-10
4.5	Comparison of Performance	4-11
4.7	SST System Evaluation	4-17
4.8	Comparison Between SST and Bendix SSA Star Mapper	4-20
4.9	Comparison of Star Tracker Candidates	4-21
4.10	Preliminary Characteristics of CCD Star Tracker	4-23
4.11	Contents of File 7	4-28
5.1	Teledyne SDG-5 TDF Nominal Error Parameters . .	5-7
5.2	SDG-5 Gyro Physical Properties	5-8
5.3	Star Update Error Source Summary Table	5-10
5.4	GPS Orbit Errors Used in Performance Study . .	5-11
5.5	Pointing Error for Various Integrators at Various Integration Stepsizes	5-16
5.6	Pointing Error for Each Individual Gyro Error Term	5-18
5.7	Star Update Error Source Sensitivity	5-22

LIST OF TABLES - (Continued)

TABLE		PAGE
5.8	Comparison of Proposed Onboard Orbit Generators to GTDS	5-26
5.9	LANDSAT-D Orbit Performance Results for Various Maneuver Rates	5-30
5.10	Nadir Tracking Performances Results for Earth Missions	5-31
5.11	Performance Results for a 20 Degree Maneuver at 5°/Minute for the 400 km and 705.3 km Earth Missions	5-32
5.12	Performance Results For a 20 Degree Maneuver at 2°/Second for the 400 km and 705.3 km Earth Missions	5-33
5.13	Performance Results for a 5° Maneuver at 5°/ Minute and 2°/Second for the 2000 km Earth Mission	5-34
5.14	Time Between Star Tracker Updates for the Earth Missions	5-35
5.15	Stellar Mission Performance Results	5-38
5.16	Solar Mission Performance Results	5-39
6.1	Estimated IRU Throughput Using Single APU . .	6-6
6.2	Processor Efficiency In Single APU System . .	6-8
6.3	Estimated IRU Throughput In Dual APU System .	6-9
6.4	Estimated Star Tracker Throughput	6-16
6.5	Multiple Phase Star Tracker Throughput	6-19
6.6	Summary of Estimated OADS Throughput	6-26
7.1	OADS Baseline Microcomputer Configuration Power Consumption and Component Count	7-2
7.2	Performance Comparison of NMOS and Alternate Fabrication Technologies	7-7

LIST OF TABLES - (Continued)

TABLE		PAGE
9.1	Estimated IRU Processing Times for NSSC-I Computer	9-2
9.2	IRU Data Table	9-3
9.3	Estimated Star Tracker Processing Times for NSSC-I Computer	9-6
9.4	Star Tracker Data Table	9-7
9.5	Multiple Phase Star Tracker Throughput for NSSC-I	9-9
9.6	Estimated Orbit Generator and Resolver Processing Times for NSSC-I Computer	9-9
9.7	Orbit Generator and Resolver Data Table. . . .	9-10
9.8	Summary of Estimated Processing Times for NSSC-I Computer	9-11
9.9	Summary of Estimated Memory Size for NSSC-I Computer	9-11
11.1	Major On-Board System Operational Update Requirements	11-4

LIST OF FIGURES

FIGURE		PAGE
2-1	Proposed OADS System	2-2
2-2	OADS Multiple Microprocessor System	2-6
3-1	Inertial and Earth Reference Frame	3-4
3-2	Rotation Between I Frame, O Frame and L Frame .	3-6
3-3	Body Fixed Axes	3-8
3-4	α, δ ,ϕ Attitude Angle Definition	3-10
3-5	Sun Aspect Angle Definition	3-11
3-6	Relation Between the L-Frame and F-Frame	3-13
3-7	Rotation Relation Between F-Frame and B-Frame	3-14
4-1	Schematic Diagram of DRIRU-II System	4-2
4-2	Power Spectral Density Comparison	4-8
4-3	Ball Brother's CT401 Star Tracker	4-15
4-4	Inertial Reference and Earth Centered - Earth Fixed Coordinate Systems	4-30
4-5	Summary of Interface Between OADS and GPS Receiver	4-31
5-1	OADS Computer Simulator	5-2
5-2	Detail IMU Block Diagram	5-3
5-3	Star Tracker Subsystem	5-5
5-4	Maneuver Profiles	5-12
5-5	Integration Error Diagram	5-15

LIST OF FIGURES - (Continued)

FIGURE		PAGE
5-6	Pointing Error (2σ) As a Function of Scale Factor	5-19
5-7	Pointing Error (2σ) As a Function of Gyro-to-Gyro Nonorthogonality Error	5-20
5-8	Pointing Error (2σ) As a Function of Gyro Nonorthogonality Error	5-21
5-9	Poisition and Velocity Propagator Algorithm . .	5-25
5-10	RSS of Position Errors As a Function of Computation Stepsize for the Position and Velocity Propagator	5-27
6-1	Processing Elements In Onboard Attitude Determination System	6-2
6-2	IRU Processing Steps	6-3
6-3	Representative Execution In Single and Dual APU Configuration	6-7
6-4	Star Tracker Processing	6-11
6-5	Star Catalog Search Using Indirect Indexing . .	6-18
6-6	OADS Multiple Microprocessor System	6-25
6-7	Typical OADS Microcomputer Processing Timeline .	6-27
10-1	Existing ACS Module Configuration	10-2
10-2	ACS Module Supplemented by OADS Microcomputer System	10-3
10-3	OADS - Experiment - ACS Relationship	10-5
10-4	NSSC-I Supplemented by Bit-Slice Arithmetic Processing Unit	10-6

ACRONYMS

Arithmetic and Logic Unit

Arithmetic Processing Unit

Charged-Coupled Device

Complementary Metal-Oxide Semiconductor

Digital Redundant Inertial Reference Unit

Earth Centered - Earth Fixed

Global Positioning System

Host Vehicle

Inertial Reference Unit

Integrated Injection Logic

Large Scale Integration

Martin Marietta Aerospace

Multi-Mission Spacecraft

Metal Oxide Semiconductor

Microprocessor Unit

Medium Scale Integration

N-Channel Metal Oxide

Onboard Attitude Determination System

P-Channel Metal Oxide Semiconductor

Power Spectrum Density

Read or Write Random Access Memory

Rate Integration Gyro

Read Only Memory

Receiver/Processor Assembly

Spacecraft

Silicon On Saphire

ACRONYMS - (Continued)

SSI	Small Scale Integration
SST	Standard Star Tracker
SV	Space Vehicles
TDF	Two Degree of Freedom

FOREWORD

This report documents the results of the On-Board Attitude Determination System (OADS) study for advanced spacecraft missions. It is submitted in accordance with contract number NAS5-23428, modification 27, and covers the work performed from 11 October 1977 through 11 April 1978.

1.0 INTRODUCTION

The purpose of the six-month study was to determine the requirements, capabilities and system design for an on-board attitude determination system (OADS) to be flown on advanced spacecraft missions. The specific objectives were directed in two areas; first, an OADS design that was spacecraft independent and second, an OADS design to be incorporated on the multi-mission spacecraft. Each design was required to provide the required attitude and pointing information for three basic missions--Earth, stellar and solar-oriented missions. The related mission parameters were selected by GSFC on those missions that were considered typical for advanced spacecraft missions.

The basic study approach taken to determine the OADS system capability and preliminary design was as follows:

- The first step was to establish a specific set of mission requirements and study ground rules that would permit achieving meaningful results within the time and budget constraints of the contract.
- The second step was to develop the attitude determination algorithms to process the sensor data considering, initially, the NASA Standard star tracker and gyro and the GPS receiver.
- The third step was to evaluate other candidate sensors and compare their characteristics to the NASA Standard components.
- The fourth step was to use a computer simulation to evaluate the OADS performance characteristics and to establish the OADS software and hardware design parameters.
- The fifth step was to design a microprocessor system that would meet the derived requirements established in Step 4.

Following this basic approach, we were able to establish an OADS design for the spacecraft independent concept. The OADS design for the multi-mission spacecraft was established by first, investigating the implementation of the OADS derived software requirements in the NSSC-1 and second, investigating the implementation of the same requirements using the NSSC-1 supplemented by a microcomputer system. The latter is by far the most promising.

A summary of the proposed OADS concept and basic study results is presented in Section 2.0. The content of Section 3 through 8 generally follows our study approach. Section 3 discusses the selected orbit features and the coordinate system definition used in the study. Section 4 discusses the OADS sensor evaluation showing the candidate sensor characteristics and the trade studies that were run to select the OADS attitude sensors. As part of the sensor evaluation, the GPS Magnavox receiver/processor assembly is discussed showing how it will be controlled and used to provide the necessary position and velocity information for the OADS design. Section 5 discusses the mission performance studies showing the sensitivity analysis results and the nominal performance results from the computer simulation. Section 6 through 8 discusses the microprocessor software, hardware and system analysis for the spacecraft independent concept. Section 9 discusses the implementation of the attitude determination algorithms in the NSSC-1 showing the resulting time analysis. Section 10 discusses the investigation of supplementing the NSSC-1 with a multiple microcomputer system and what the impact might be for such a system. Section 11 discusses the OADS testing possibilities. Also, a general description of an OADS hardware test bed presently being pursued at Martin Marietta Aerospace for another program is discussed. Finally, Section 12 presents some recommendations for further studies.

For information purposes, we have provided a number of appendices to help the reader understand what vendor hardware/software reference material we used for the study. We have also included an appendix detailing the timing and sizing analysis for the NSSC-I.

2.0 OADS SYSTEM DESIGN CONCEPT SUMMARY

Based upon the OADS requirements and system performance evaluation, a preliminary on-board attitude determination system is proposed. The proposed OADS system consists of one NASA Standard IRU (DRIRU-II) as the primary attitude determination sensor, two improved NASA Standard star tracker (SST) for periodic update of attitude information, a GPS receiver to provide on-board space vehicle position and velocity vector information, and a multiple microcomputer system for data processing and attitude determination functions. The functional block diagram of the proposed OADS system is shown in Figure 2-1. The computational requirements are evaluated based upon this proposed OADS system. The major conclusions from the OADS study are summarized in the following subsections.

2.1 OADS System Sensor and Performance

Based upon the trade study of current existing sensors, the NASA Standard IRU system--DRIRU-II is far superior to any of the current existing gyro system operating in the strapdown environment. In order to increase the life-time reliability, a backup DRIRU-II unit may be considered in conjunction with the primary unit proposed. The improved NASA standard star tracker is the best candidate for the MMS mission at the present time. The SST should be used only when the spacecraft slew rate is less than 0.5°/sec.

The GPS receiver, with the assistance of an on-board position and velocity propagator, when operating under the spacecraft user environment, is able to provide orbit information for which the impact to OADS system accuracy is insignificant.

The proposed on-board attitude determination system, with the described attitude determination procedure and algorithm, is capable of providing precision on-board attitude information for all three MMS missions.

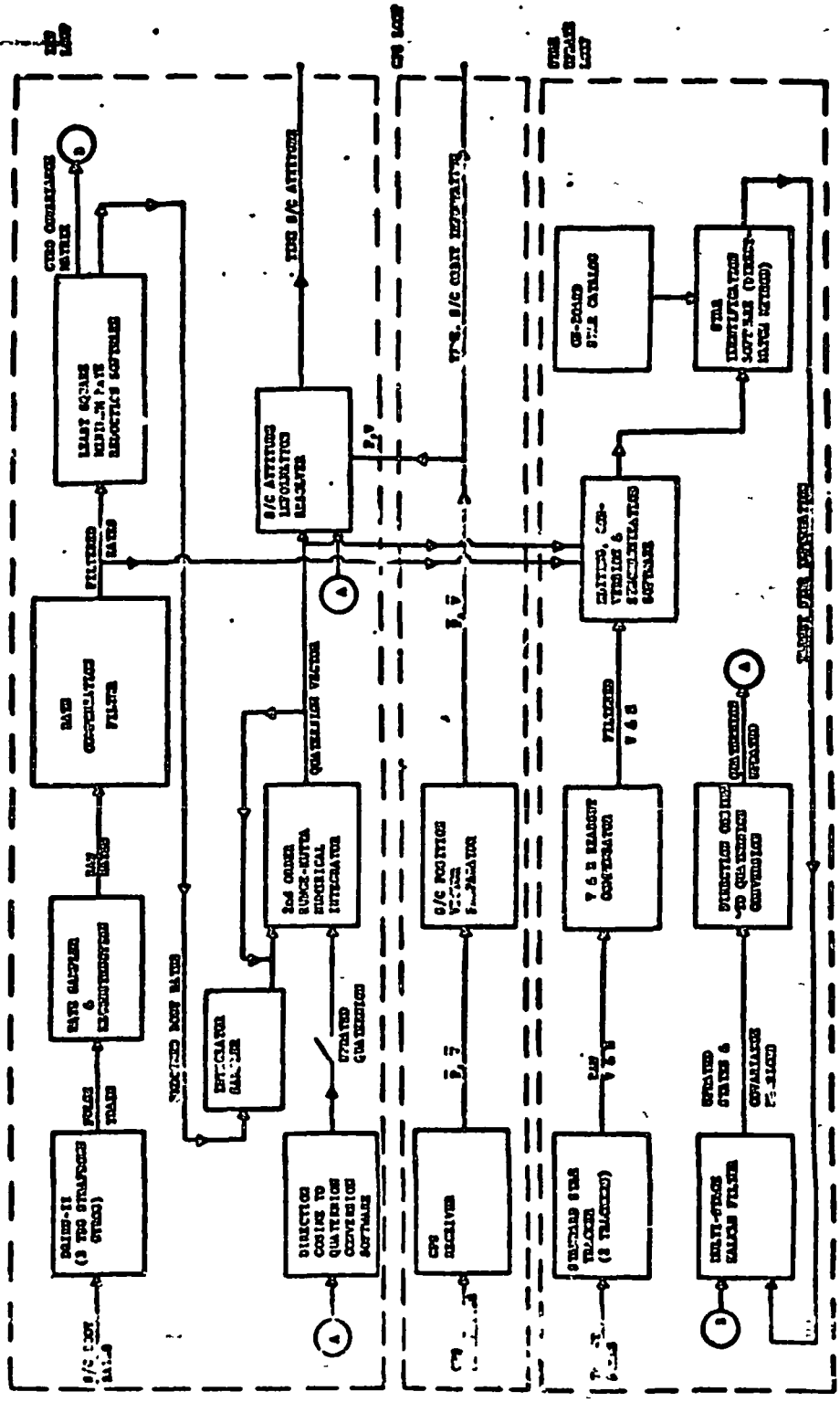


FIGURE 2-1 PROPOSED OADS SYSTEM

As expected, the Earth mission posses the most demanding OADS system processing among the three MMS missions. For the same maneuvering sequence, the 400 km altitude Earth mission requires most frequent update by the star trackers. The OADS Star update frequency decreases as the spacecraft altitude increases and is relatively independent to the orbit plane inclinations.

The most significant IRU errors experienced in the slewing mode environment are scale factor and misalignment and bias drift in the non-slewing environment. Therefore, as the maneuver rate increases, the required time interval between star update becomes shorter. The most significant star tracker error is the tracker boresight axis misalignment. Improvement of overall OADS performance is anticipated if better knowledge of this error is available.

2.2 On-Board Computation Requirements

Our investigations show that use of a multiple microcomputer system for onboard attitude determination is quite feasible from a software performance view. The timing estimates for the baseline microcomputer we examined are summarized in Table 2.1. A system level block diagram of the baseline multiple microcomputer system is shown in Figure 2-2. It should be emphasized that the baseline configuration, which we examined, utilized commercial LSI devices. The performance of these devices have been derated to accommodate operation in an extended temperature range; however, radiation threshold levels for the devices are far below the levels needed for flight quality microcomputer hardware. While performing this OADS analysis, we could not find microcomputer devices which were ideally suited to the spaceborne environment and had the necessary performance to support the onboard attitude determination system. Radiation hardened devices are a

major problem although power consumption considerations come into play, but to a lesser extent. We have, therefore, included in this report recommendations briefly outlining the type of LSI devices which we feel would be appropriate in future spaceborne applications such as OADS.

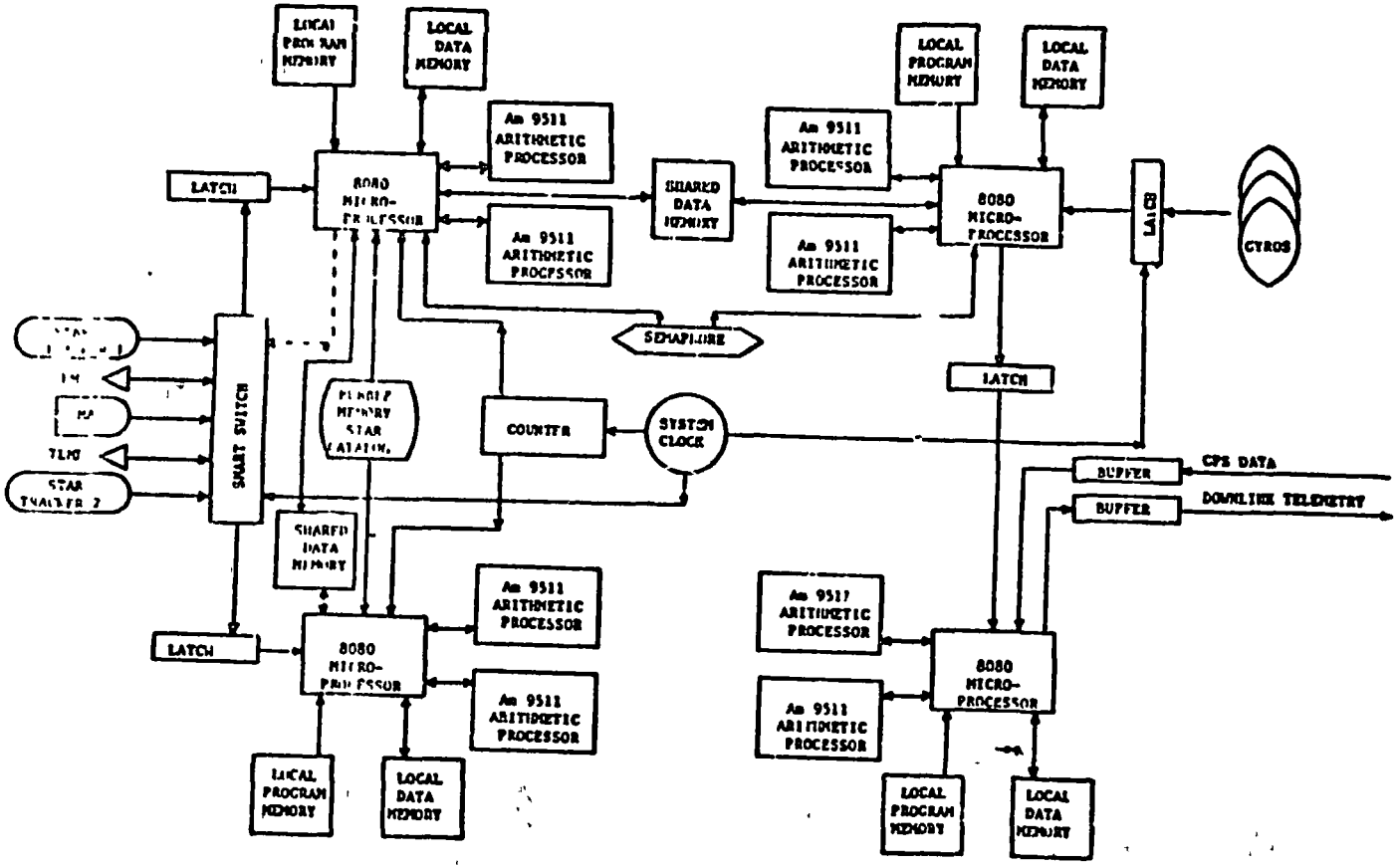
We also performed a timing analysis to determine what performance could be expected if the attitude determination algorithms were executed on the NSSC-I computer. It quickly became obvious that the NSSC-I could not support these operations. This results from the fact that the OADS algorithms are very computation oriented and not like the data management tasks which the NSSC-I is tailored to performing. Table 2.2 summarizes the NSSC-I timing analysis. Because of these results, a top-level investigation was performed to determine the feasibility of supplementing the HSSC-I with an OADS microcomputer system. The results of this investigation tend to indicate that this approach could be very attractive if a flight quality microcomputer system were available.

TABLE 2.1 SUMMARY OF ESTIMATED OADS PROCESSING TIME FOR BASELINE MICROCOMPUTER SYSTEM

	Processing Time (ms)
IRU Processing	29.42
Star Tracker Processing	
Phase I (star identification)	134.3
Phase II (1 tracker - quaternion correction)	152.2
Phase II (2 trackers - quaternion correction)	235.8
Orbit Generator Processing	3.421
Resolver Processing	38.518

TABLE 2.2 SUMMARY OF ESTIMATED OADS PROCESSING TIMES FOR NSSC-I COMPUTER

	Processing Time (ms)
IRU Processing	44.575
Star Tracker - Star Identification	183.98
Star Tracker - Quaternion Correction (1 tracker in use)	314.983
Star Tracker - Quaternion Correction (2 trackers in use)	609.468
Orbit Generator/Resolver	187.527



GADS MULTIPLE MICROPROCESSOR SYSTEM

FIGURE 2-2

3.0 OADS ORBIT DESCRIPTION AND ATTITUDE DEFINITION

3.1 MMS Mission Selected Orbits and Related Features

A set of Solar, Earth, and Stellar missions for the MMS satellite was selected at the first GSFC interchange meeting. The selected orbits for the evaluation and its related features are described in the following sub-sections.

3.1.1 Earth Mission

Orbit: Circular orbit evaluated at altitudes of 400 km, 705.3 km and 2000 km. The orbital inclination angles under consideration are 56° for the 400 km and 2000 km altitudes and approximately 97° (sun synchronous) for the 705.3 km altitude.

Mission Features: For the Earth mission, the spacecraft's nadir axis is tracking along the Earth local vertical direction. That is, the spacecraft nadir axis is rotated at the orbital rate closely aligned with the local position vector direction. Pitch maneuvers of ± 20 degrees at 400 km and 705.3 km altitude and ± 5 degrees at 2000 km altitude should also be considered as the nominal mode for the Earth mission. The spacecraft nominal rate is orbital rate with maneuver slew rates of 5°/min and 2°/sec.

3.1.2 Stellar Mission

Orbit: Circular orbit evaluated at altitudes of 400 km and 2000 km. The orbital inclination angle under consideration is 28.5°.

Mission Features: For the stellar mission, the spacecraft body axis is locked to the stars with only the dwelling motion being considered. Spacecraft maneuvers from one star to a new star with slew rate range from 5°/min to 2°/sec should also be considered in the OADS system accuracy evaluation.

3.1.3 Solar Mission

Orbit: Circular orbit evaluated at altitudes of 400 km and 2000 km. The orbital inclination angle for the solar mission orbit is 28.5°.

Mission Feature: For the solar mission, the spacecraft body axis is mainly locked to the sun line vector with possible small angle scan in the plane perpendicular to the sun line vector. The possible scan rate is 5°/min for OADS evaluation.

3.2 Coordinate System Definition

In order to define the spacecraft attitude angles for a MMS mission, the reference coordinate systems must be defined. The basic reference coordinate systems defined in this report are selected for the convenience of the MMS mission. It is subject to change when other requirements are inserted.

a. Inertial Reference Coordinate - I frame

A 1950 epoch inertial reference is selected as the inertial coordinate. The reason for selecting this coordinate is that most of the existing star catalogs exist using this coordinate, therefore, no extra transformation is required; it also gives a fixed reference in the sun-centered heliocentric frame.

The coordinate frame is defined as follows:

X_I axis along 1950 Epoch vernal equinox direction

Z_I axis along symmetrical earth rotation axis (North pole)

Y_I axis completes a right-hand triplet

Another commonly used inertial frame is the true-of-date Epoch reference in which the true of date (current) vernal equinox direction is used. There is a slight rotation angle between the fixed Epoch and true of date inertial systems. These must be corrected if any

a. Inertial Reference Coordinate - I frame (Continued)

quantity is defined using the true of date Epoch reference (i.e., Keplerian orbital parameters).

b. Earth Reference Frame - E frame

The Earth reference frame is chosen with X_E axis along the zero longitude (Greenwich) with a rotation angle from the X_I axis. Z_E is coincident with Z_I (North Pole). X_E and Y_E are contained in the plane of X_I and Y_I. If we assume a constant earth rotation rate of W_E = 0.41667×10^{-2} deg/sec, then X_E has a rotation angle of $W_E t$ from the X_I axis (where t is measured from Greenwich noon-time). In general, the Earth longitude and latitude is defined in the Earth reference frame.

The I and E frames are shown in Figure 3-1.

If Point P is defined as the target to be tracked on Earth surface with longitude ϕ_p and latitude λ_p, then the tracking station P unit vector expressed in the I frame will be:

$$\{U_I\} = [{}^I C_E] \{U_E\} \qquad (3.1)$$

(From here on, $\{\ \}$ denotes vector quantity, $[\ ]$ the matrix quantity, and the symbol "$\wedge$" indicates unit vector.)

Where:

$$\{U_E\} = \{\cos\lambda_p \cos\phi_p,\ \cos\lambda_p \sin\phi_p,\ \sin\phi_p\} \qquad (3.2a)$$

$$[{}^I C_E] = \begin{bmatrix} \cos W_E t & -\sin W_E t & 0 \\ \sin W_E t & \cos W_E t & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

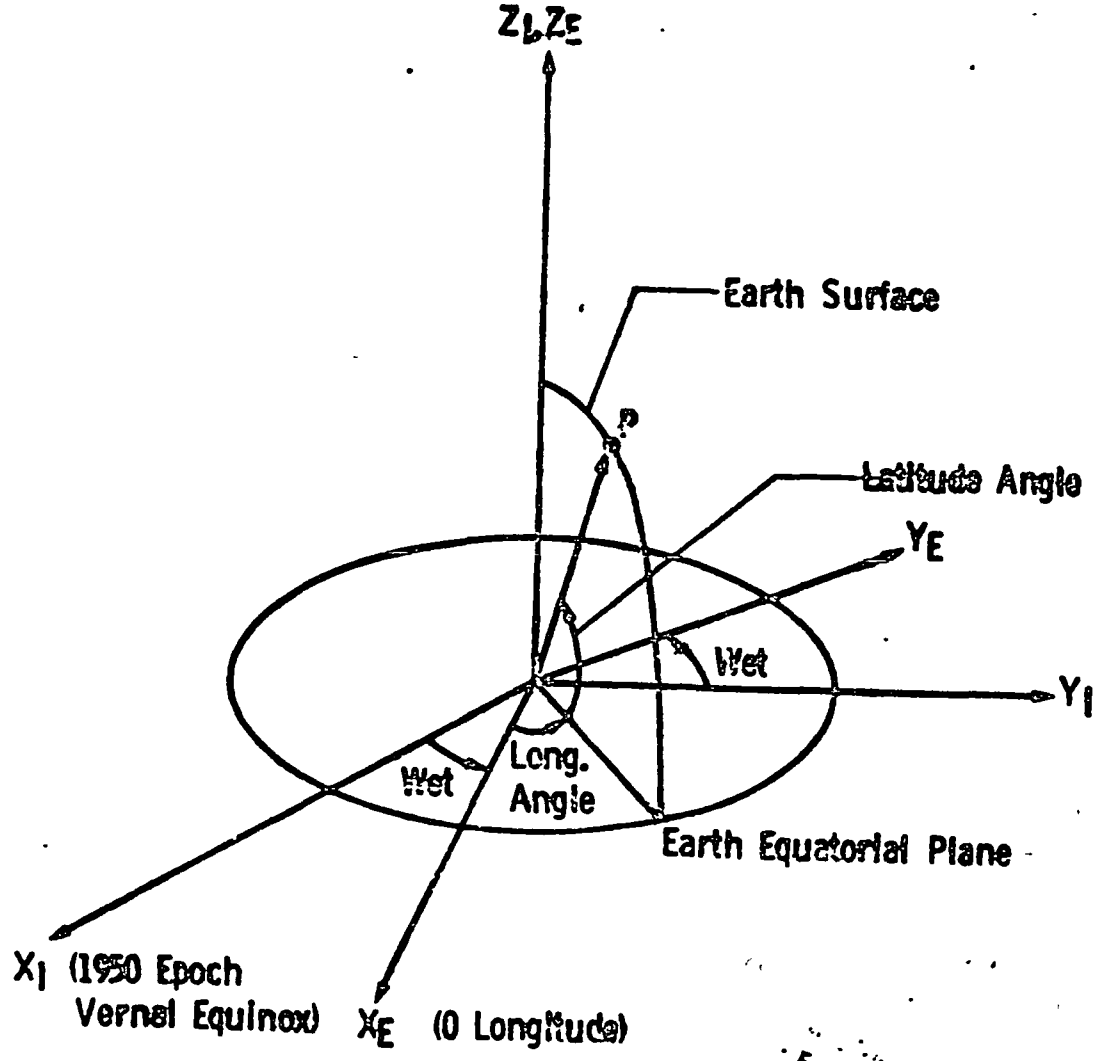


FIGURE 3-1 INERTIAL AND EARTH REFERENCE FRAME

c. Orbit Reference Frame - O Frame

The orbit reference frame defines the orientation of the S/C orbit in the I frame. It is generally defined by (3-1-3) Euler transformation angles between the I and O frames. The relationship between the orbital frame and inertial frame is illustrated in Figure 3-2. The transformation of a vector from O frame to I frame is given by letting

$$\left\{\hat{U}_I\right\} = \left[{}_I C_O\right]\left\{\hat{U}_O\right\} \tag{3.2}$$

$C \Leftrightarrow \cos \qquad S \Leftrightarrow \sin$

$$\left[{}_I C_O\right] = \begin{bmatrix} C\Omega C\omega - S\Omega S\omega Ci & - C\Omega S\omega - S\Omega C\omega Ci & S\Omega Si \\ S\Omega C\omega + C\Omega S\omega Ci & - S\Omega S\omega + C\Omega C\omega Ci & - C\Omega Si \\ S\omega Si & C\omega Si & Ci \end{bmatrix} \tag{3.2a}$$

d. The Local Vertical Frame - L Frame

The local vertical frame is defined by the S/C position along the orbit plane, which is simply a rotation angle of the true anomaly, parameter f, (generally referred to as fast moving Keplerian parameter) from X_o about the orbit normal axis Z_o. The L frame is defined as:

X_L - from center of Earth to S/C

Z_L - normal to orbit in the direction of the orbit angular momentum vector direction.

Y_L - complete a right-hand triplet of X_L and Z_L. For circular orbit, Y_L is in the direction of the velocity vector.

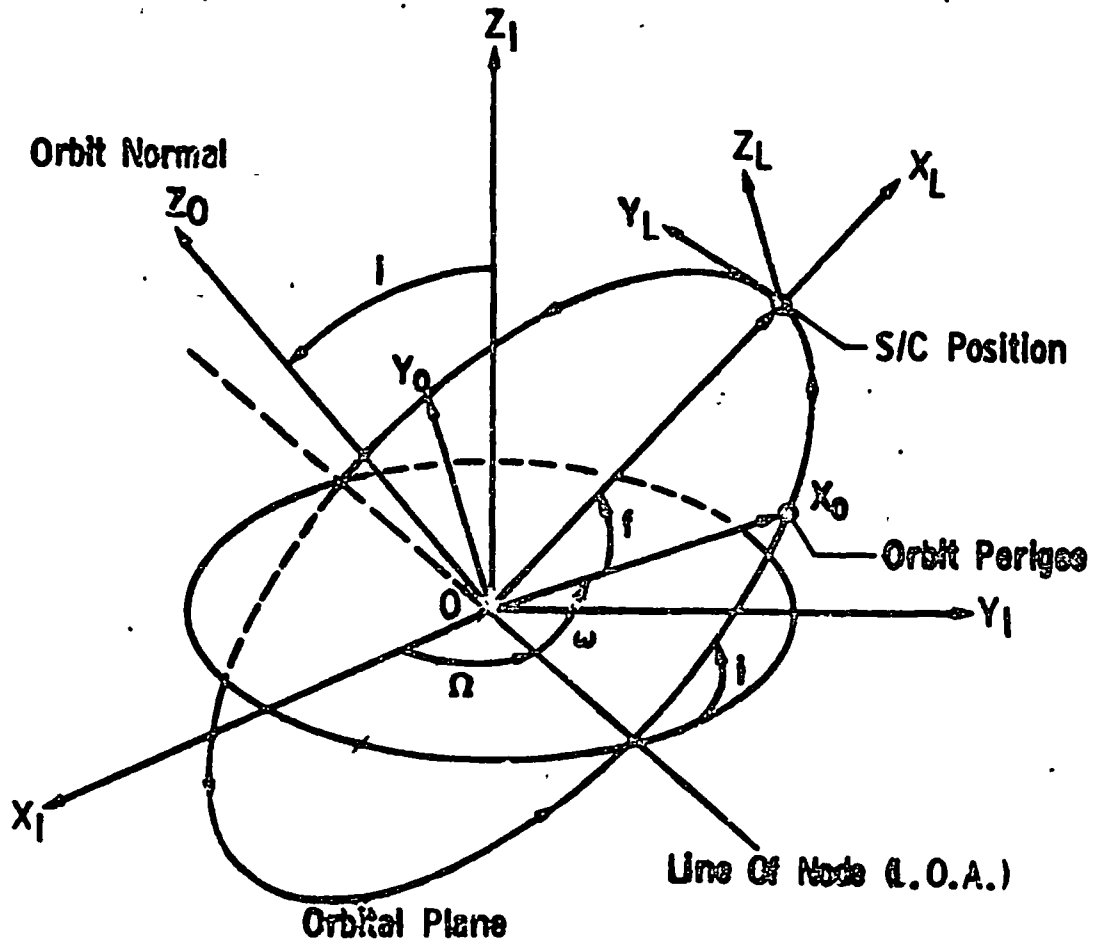


FIGURE 3-2 ROTATION BETWEEN I FRAME, O FRAME AND L FRAME

Where:
- Ω - longitude of ascending node
- i - inclination angle
- ω - argument of perigee
- f - true anomaly

d. The Local Vertical Frame - L Frame - (Continued)

Because the L frame is a simple rotation angle of f about Z_o axis from the O frame, the angles W and f are in the same plane and additive. The transformation from the L frame to the I frame replaces W in (3.2a) by the quantity (W + f); i.e.,

$$\{\hat{U}_I\} = [{}_I C_L]\{\hat{U}_L\} \tag{3.3}$$

Where:

$$[{}_I C_L] = \begin{bmatrix} C\Omega C\theta - S\Omega S\theta Ci & -C\Omega S\theta - S\Omega C\theta Ci & S\Omega Si \\ S\Omega C\theta + C\Omega S\theta Ci & -S\Omega S\theta + C\Omega C\theta Ci & -C\Omega Si \\ S\theta Si & C\theta Si & Ci \end{bmatrix} \tag{3.3a}$$

$$\theta \triangleq \omega + f$$

e. Body Reference Frame - B Frame

The body reference frame is fixed on the S/C body and is rotated with the body. The orientation of the body axes in the I frame or L frame is generally used to define the S/C attitude motion. The B frame is selected usually due to the convenience of mission requirement, S/C geometrical symmetry, and major instrument location. For convenience, we assume the MMS spacecraft is a cylindrical shape with X_B Y_B fixed in the S/C equatorial plane. The Z_B axis is along the cylindrical longitudinal axis as shown in Figure 3-3.

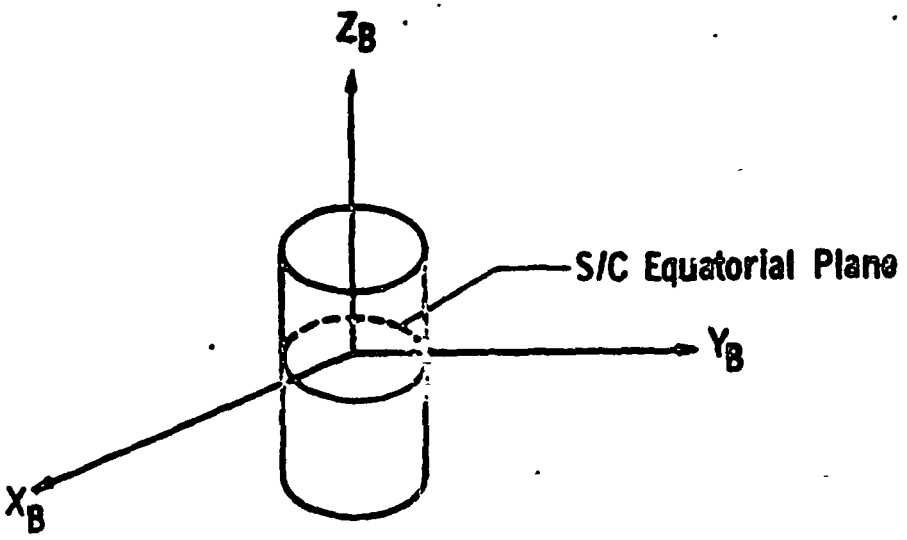


FIGURE 3-3 BODY FIXED AXES

e. Body Reference Frame - B Frame - (Continued)

From the definition described above, the Z_B axis is generally referred to as the nadir pointing axis (especially for Earth Mission). The five reference frames described above are the basic coordinate systems recommended for the MMS mission. To summarize they are:

Inertial Reference Frame	- I Frame
Earth Reference Frame	- E Frame
Orbit Reference Frame	- O Frame
Local Vertical Reference Frame	- L Frame
Body Reference Frame	- B Frame

3.3 Attitude Angle Definition and OADS output definition

3.3.1 Attitude Angle Definitions

As mentioned in 3.1, the S/C attitude is defined as the orientation of the body axes in the inertial frame or in the local vertical reference frame. There are four ways to represent the spacecraft attitude:

Direction cosines

Quaternion

Euler attitude angle

Gibbs vector presentation

The direction cosines is the most straight-forward way to describe the S/C attitude. The other methods can be derived from the direction cosine representation and are directly related to each other. The Euler attitude angle representation gives a much better physical visualization of the attitude and is generally the quantity delivered to the user although the system may not use it as the internal state variables. For MMS missions, there are three sets of attitude angles to be defined and are described in the following paragraphs.

a. Attitude angles in inertial reference frame

A 3-1-3 Euler rotation is used to define the body axes orientation in the inertia frame. The rotation is illustrated in Figure 3-4. The α, δ, ϕ attitude angle set is the right ascension, declination, and phase angle, respectively. This set is convenient for the inertial attitude angle expression for the Stellar and Solar Mission. The inertial position of a star or sun is defined in the frame by right ascension and declination angles. The body Z_B axis orientation is also represented by the same attitude set; therefore, it is very convenient for inertial pointing missions.

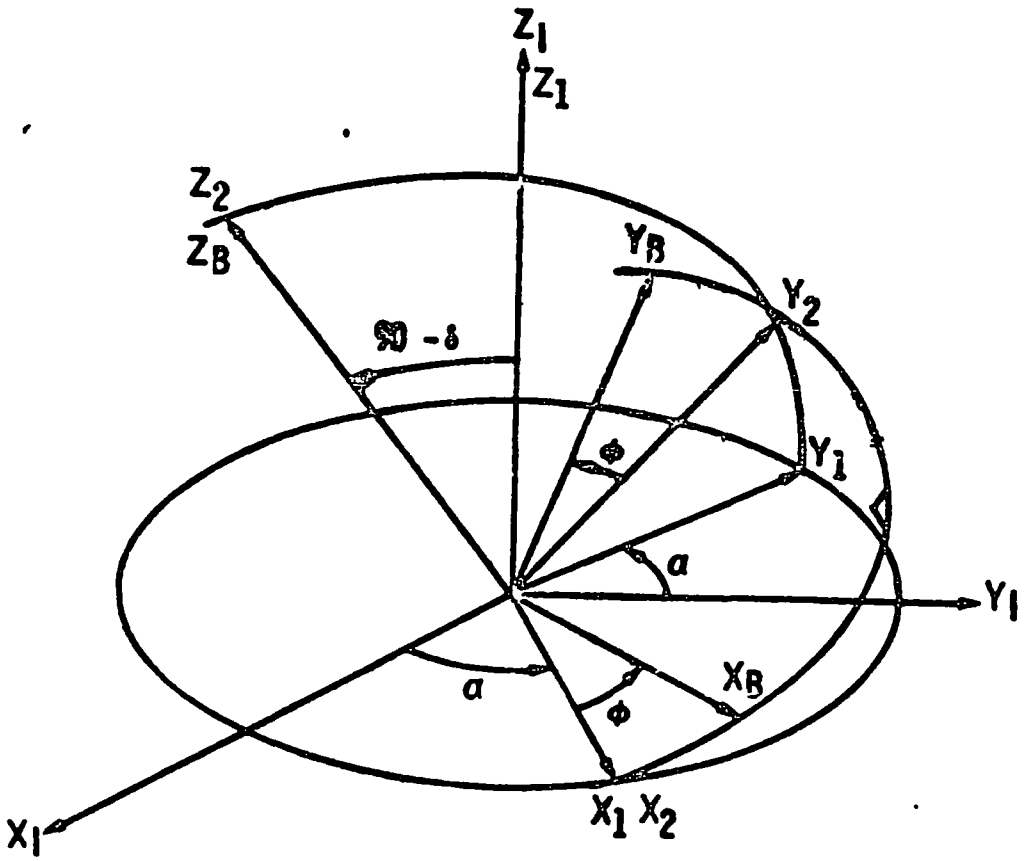


FIGURE 3-4 $\alpha\delta\phi$ ATTITUDE ANGLE DEFINITION

b. Sun Angle Definition

This set of attitude angles is defined strictly for the Solar mission. For the Solar mission, the aspect angles between the line vector and the spacecraft body frame may be useful to the user. The definition of the sun aspect angles is described in Figure 3-5.

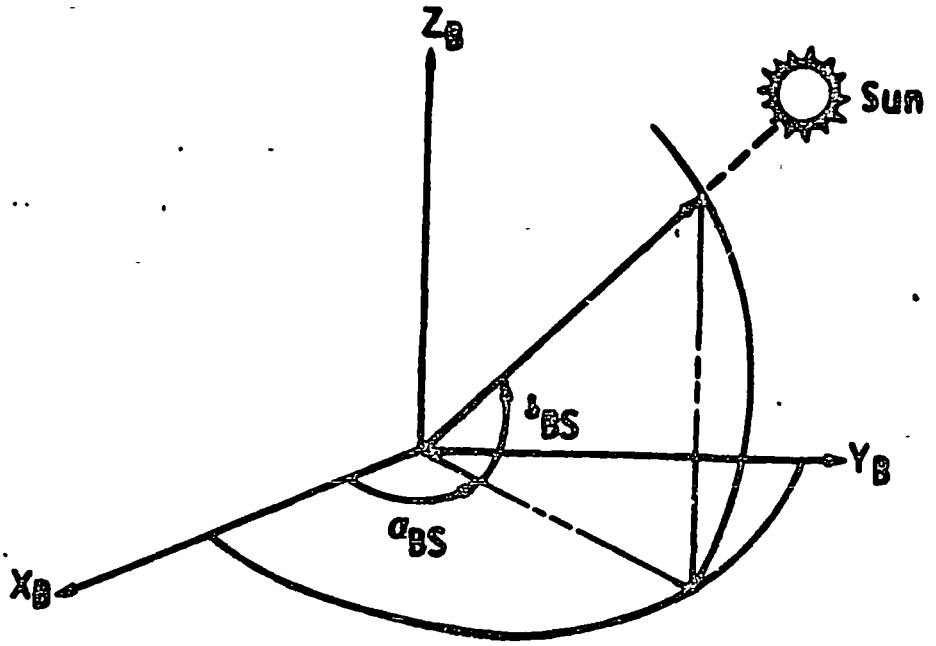


FIGURE 3-5 SUN ASPECT ANGLE DEFINITION

As shown in Figure 3-5, the sun aspect angles α_{BS} and δ_{BS} are defined as the sun line orientation in the body frame, where α_{BS} is the right ascension sun aspect angle and δ_{BS} is the declination sun aspect angle.

c. Attitude Angles in local vertical Reference frame - pitch, roll and yaw angles

This set of attitude angles is very convenient for the Earth mission of the MMS satellite. In order to follow a conventional manner of pitch, roll and yaw angle defintion, an intermediate reference frame called local flight frame, F frame, is introduced and the pitch, roll and yaw angles are actually defined in the F frame. The F frame is obtained by two 90° rotation from the L frame defined in Section 3.1. The relation between F frame and L frame is shown in Figure 3.6.

The F-frame axes are defined as:

X_F - in the + flight path direction

Y_F - opposite to the orbit normal

Z_F - along the local vertical, positive pointing toward the earth center

A 1-2-3 Euler rotation is used to rotate the pitch, roll, and yaw angles from F-frame to B-frame. The rotation is shown in Figure 3-7. This set of attitude angles defines the body axes in the flight path frame. It is used for an Earth pointing spacecraft and is very useful to control system work. For a single rotation sense, the Φ angle is the roll angle along the flight path. The θ angle is the pitch angle along the negative orbital normal. The Ψ angle is the yaw angle defined along the nadir vector.

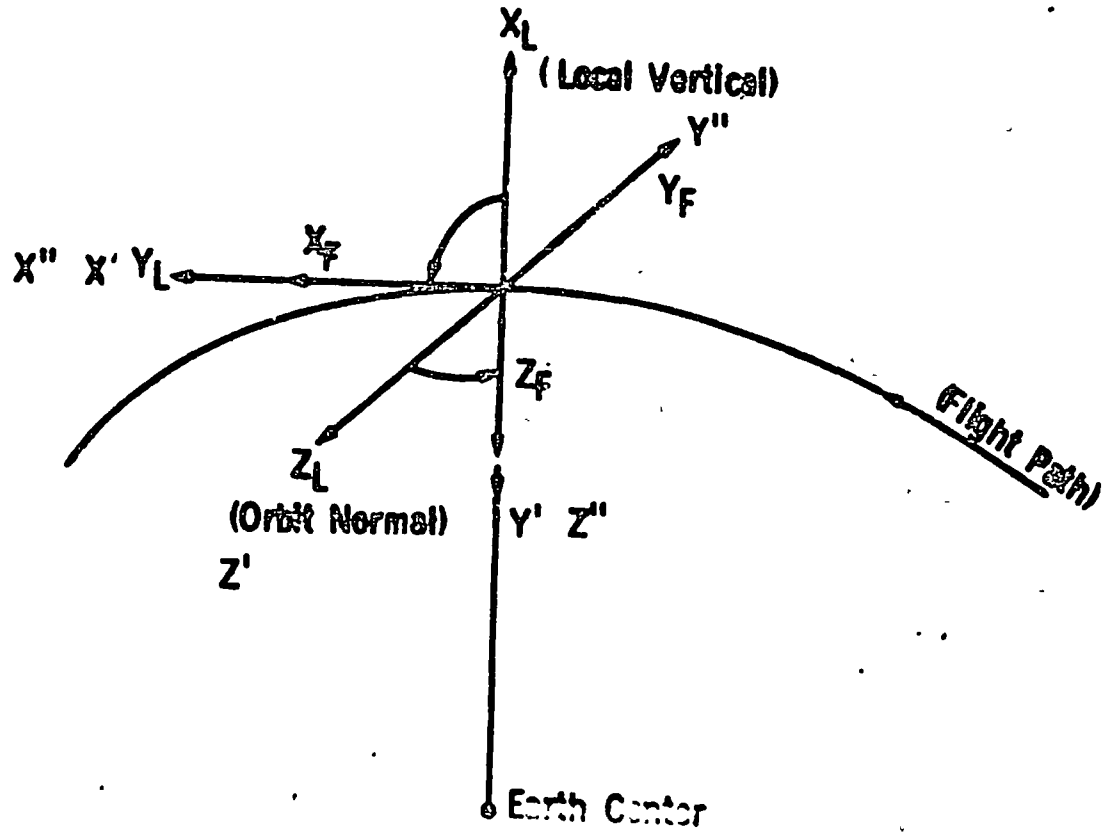


FIGURE 3-6 RELATION BETWEEN THE L-FRAME AND F-FRAME

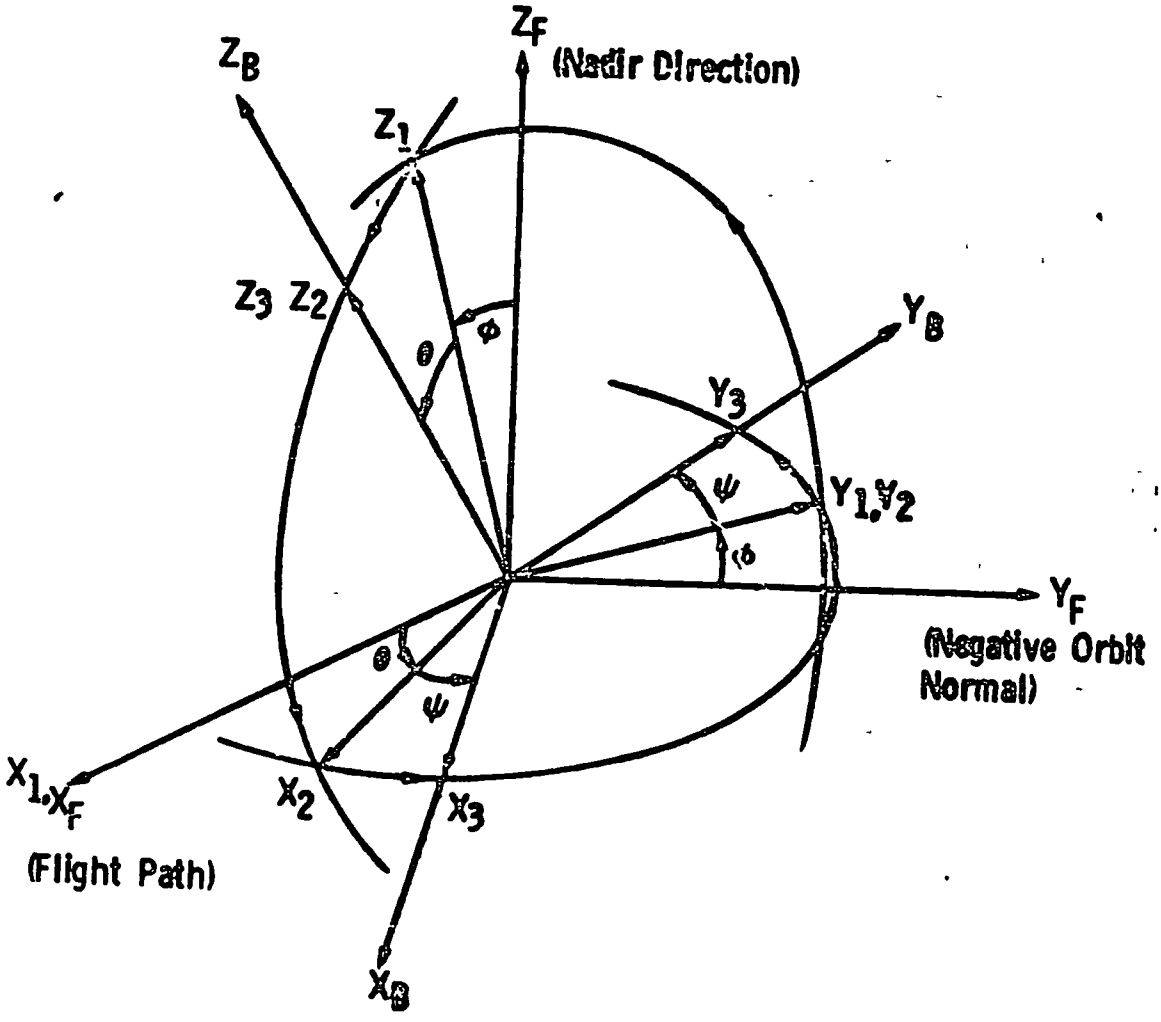


FIGURE 3-7 ROTATION RELATION BETWEEN F-FRAME AND B-FRAME

3.3.2 OADS Output Definition

The required attitude information is different for different MMS mission modes. The intent of this subsection is to define the preliminary OADS attitude output format. The output set is subject to change upon user request or future mission requirements.

Earth Mission

Record Description	Unit	Remarks
Time (t)	Seconds	Seconds from S/C clock epoch
Pitch Angle (θ)	Degrees	As described
Roll Angle (Φ)	Degrees	"
Yaw Angle (Ψ)	Degrees	"
Target Longitude ϕ_p	Degrees	Nadir axis pointing target
Target Latitude γ_p	Degrees	"
Target Altitude (hp)	Km	"

Solar Mission

Record Description	Unit	Remarks
Time (t)	Seconds	From S/C clock epoch
Right Ascension Angle (α)	Degrees	As described
Declination Angle (δ)	Degrees	"
Phase Angle (ϕ)	Degrees	"
Sun Aspect R.A. Angle (α_{SB})	Degrees	"
Sun Aspect Dec Angle (δ_{SB})	Degrees	"

Stellar Mission

Record Description	Unit	Remarks
Time (t)	Seconds	From S/C clock Epoch
Right Ascension Angle (α)	Degrees	As Described
Declination Angle (δ)	Degrees	"
Phase Angle (ϕ)	Degrees	"

Uncertainty (or confidence level) of each output record may be attached with the preliminary output records described above. The orbital information received from the GPS sensor may also be attached. However, it is not directly required for the OADS function.

4.0 OADS SENSOR EVALUATION

4.1 Inertial Reference Unit

4.1.1 Description of NASA Standard Unit - DRIRU-II

The NASA Standard IRU unit is the DRIRU-II (Digital Redundant Inertial Reference Unit - II) manufactured by Teledyne. It consists of 3 SDG-5 two-degree-of-freedom (TDF) dry-tuned gyros mounted orthogonally in a single unit. The schematic diagram is shown in Figure 4-1. Each TDG gyro has its own power supply with independent electronics for its two output channels. Temperature computation of scale factor and bias are included in the electronics. The DRIRU-II unit is to be mounted directly upon the spacecraft body (in strapdown environment); therefore, the IRU gyros sense the body rates in inertial space directly. The body rates are output in the form of digital pulse counts via a Voltage to Frequency (V/F) Converter. The three TDF gyros provide full redundancy of vehicle's body 3-axis rate measurement. In the case of one gyro failure, the remaining two TDF gyros still provide the full 3-axis measurement, therefore, no rate data measurement is lost. In order to reduce the rate errors of the TDF gyro related errors, an on-board software compensator (rate filter) is required other than the temperature compensation provided by the output channel electronics. Periodic update of gyro error parameters are necessary to maintain the rate output accuracy. In-flight recalibration may also be necessary to maintain the knowledge of major-error source uncertainty due to environmental and space vehicle control impact. Because of the on-board compensator and temperature compensation by the output channel electronics, thermal control of the IRU system is not required under nominal temperature range.

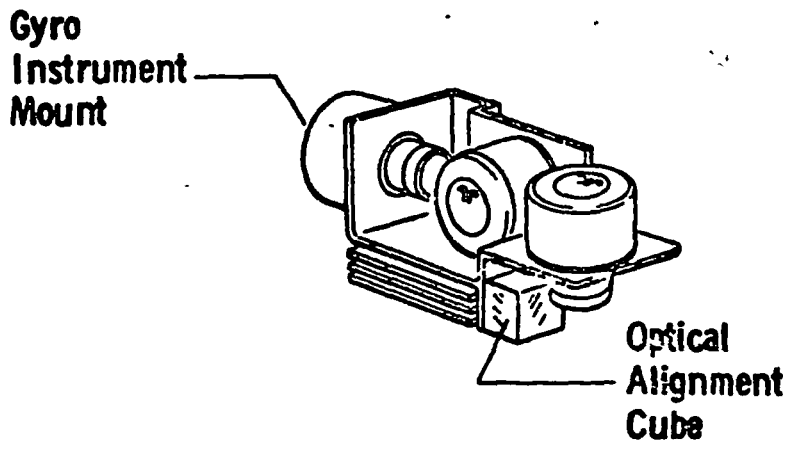


FIGURE 4-1 SCHEMATIC DIAGRAM OF DRIRU-II SYSTEM

A system evaluation of the DRIRU-II system was conducted and the results are presented in the following Table 4.1.

TABLE 4.1 DRIRU-II SYSTEM EVALUATION

CONSIDERATION FACTOR	DESCRIPTION
Life Cycle Costs	200K for a system.
Design and Development	Breadboard was built to verify design requirements.
	Modular assembly and simplified gyro design for production build.
	Test plans and specifications written in accordance with NASA requirements.
	Gyro is two degree of freedom dry tuned gyro.
Build	Most components manufactured at contractor's facility
	Electronic components compliance with MIL-STD-975.
	DMS09306 Rev. A
	MSFC Std 136
	NASA Standard parts

TABLE 4.1 DRIRU-II SYSTEM EVALUATION - (Continued)

CONSIDERATION FACTOR	DESCRIPTION
Test	Complete step-by-step qualification and acceptance test procedures will be available for system test. Test points will be available for verification of system level interface testing.
Weight	System weight 25 lb. each.
Volume	Volume/system 1183 in^3.
Complexity - Passive Electrical	3 gyros/system Motors Hystress Synch.
Mechanical	Mechanically and thermally mounted to vehicle.
Growth	Electrical Boards only 70% full.
Accessibility/Maintainability	Modular design built for maintainability
Potential for 1g demonstration	Verification in 1g field very acceptable
Modularity	Yes
Technical Uncertainty	None
Performance-Functional Attributes	Random Drift .0005°/Hr. Attitude noise .33 arc sec. Continuous rate 100°/sec. First difference for 1 sec sample interval .00425°/Hr.
EMI Susceptibility	Meet MIL-STDs 461, 462, 463 for susceptibility and generation.
Maintainability/Accessibility	Modular design for maintainability and accessibility.
Power Consumption (users)	20 watts/system
Standby power required	None
Average power required	20 watts
Contamination to spacecraft	Materials selected to prevent outgassing.
Mission Effectiveness	Meet overall system requirements.

TABLE 4.1 DRIRU-II SYSTEM EVALUATION - (Continued)

CONSIDERATION FACTOR	DESCRIPTION
Lifetime	5 years
Reliability	System reliability .853 at end of three years. No single point failures.
Survivability	Designed to meet transportation, storage and handling requirements.
Natural Environment	
Manmade - Internal to S/C	TBD
Manmade - External to S/C	Designed to meet vehicle launch vibration, shock and temperature
Technical Uncertainty	Contractor has good track record in the technical field.
Growth	Growth potential is available within present envelope.
Software Complexity	
Airborne	Require on-board data reduction of gyro data, error compensation and numerical integrator
Ground Support - Update	Simple failure detection and parity check
Downlink processing	Required for calibration (bias update)
In-Flight Calibration	Required periodically to update scale factor, misalignment and bias
Ground Checkout	
Complexity	Verification of system operation not complex
Testability	Verification of system is relatively easy. Adequate test outputs.
In-Field Calibration	Not Required.
Schedules	
Design	Design engineering based on DRIRU-II. No facility mod required. Completion by December, 1977.

TABLE 4.1 DRIRU-II SYSTEM EVALUATION - (Continued)

CONSIDERATION FACTOR	DESCRIPTION
Build	Subcontract procurement total DRIRU System. Approximately 12 months from go-ahead to delivery.
Tests	
Ground Checkout	Yes - Procedures exist
Interface Testing	Yes - Procedures exist
Subsystem	Yes - Procedures exist
System Level	Yes - Procedures exist
Qual Article, Structures Test Articles, Launch Activity Timeline	Based on NASA qualification 1977-78

4.1.2 Trade Study

A trade study was conducted between the DRIRU-II system and other candidates. The significant factors considered in the trade study are:

1. The system must be capable of operating under all three MMS mission environments and be able to provide rate information as accurate as possible.
2. The system must be capable of maintaining accuracy and stability under a wide dynamic range of zero rate to 2°/sec with possible acceleration and jerk motion due to the control system. Moreover, a fast settling time is required.
3. The system must be operating continuously during the missions and must have long life time and high reliability.

Based upon the above criteria, a trade study between the DRIRU-II system and a single degree of freedom (SDF) gyro system (Bendix 64 PM) was conducted. A comparison of error budget between the DRIRU-II system gyro (SDG-5) and two other TDF dry tuned gyros, C-6 and G-1200 manufactured by Litton, was also conducted. Results are presented in the following paragraphs.

4.1.2.1 Trade study between DRIRU-II and Bendix Redundant Strapdown IRU

The Bendix Redundant Strapdown IRU System consists of 6 SDF 64 PM RIG (Rate Integration Floated Gyro). Each gyro channel is capable of being operating independently and can be "powered down" in case of failure. The Bendix system has been used in HEAO-A and IUE missions. MMA has conducted the first difference drift rate and PSD (Power Spectrum Density) of the SDG-5 gyro and 64 PM gyro using the facility in our Inertial Guidance Laboratory. These trade study results follow.

(1) Transient Settling Time

The 65 PM-RIG gyro has a drift characteristic associated with output axis slew rates that is undesirable. This drift could induce pointing errors as high as 35 arc seconds after slewing about the output axis at 6 deg/sec. The settling time after slewing about the output axis is about 250 seconds which is excessive. The SDG-5 gyro does not exhibit this slew rate drift or long settling time. This is due to the tight capture loop about each axis of the two degree of freedom gyro.

(2) First Difference Drift Rate Compensation

First difference tests indicated that the SDG-5 gyro has less output noise when the sample interval is short as shown in Table 4.2

TABLE 4.2 FIRST DIFFERENCE DRIFT RATES

Sample Interval Second	Teledyne SDG-5 Gyro	Bendix 64 PM Gyro
.1	0.0432	High
1.0	0.0046	0.038
10.	0.00077	0.0046
100.	0.00035	0.00028
1000.	0.00011	0.0002

(3) Power Spectral Density Comparison

Power spectral density (PSD) was also taken on the candidate gyros and are illustrated in Figure 4-2. The Teledyne PSD is an order of magnitude less than the Bendix unit, also indicating less output noise.

SEE ATTACHED SHEET

(DATA - COMPANY PROPRIETARY)

FIGURE 4-2 POWER SPECTRAL DENSITY COMPARISON

(4) Error Character Comparison

The error character reflects the accuracy and stability of the IRU system. The comparison is summarized in Table 4.3 From the table, one can see that the DRIRU-II system is superior to the Bendix system in almost every category.

TABLE 4.3 GYRO ERPOR CHARACTERISTICS COMPARISON

	DRIRU-II	Bendix 64 PM
Random Drift (deg/hour) - (1σ)	.0005	.0005
Long-Term Bias Stability (deg/hr/yr) - (1σ)	.01	.09
Torquer Scale Factor (deg/hr/ma) - (1)	.6	230.0
Torquer Linearity (PPM) - (1σ)	25	37
Torquer Asymmetry (PPM) - (1σ)	3	27
Angular Rate Capability (deg/sec)	100	20
Angular Momentum (gm-cm^2/sec)	1×10^6	$.43 \times 10^6$
Anisoelastic Drift (deg/hr/g) - (1σ)	.01	.04

(5) Physical Characteristics and Reliability Comparison

The comparison between DRIRU-II and Bendix Redundant Unit is shown in Table 4.4. Again, the DRIRU-II system is superior in almost all respects.

TABLE 4.4 COMPARISON OF PHYSICAL CHARACTERISTICS AND RELIABILITY

	DRIRU-II	BENDIX SYSTEM
Weight (lbs)	25	65
Power (watts)	21	115
Cost	200K	800K
*Reliability (2 years)	0.958	0.914

*Based upon 6-gyro configuration for both systems

4.1.2.2 Trade Study between SDG-5 and Other TDF Dry Tuned Gyros

A comparison of performance between the SDG-5 (DRIRU-II gyro) and other TDF dry tuned gyros was made. The reason of this gyro level comparison rather than the system level comparison is because the other TDF gyros do not exist in a system configuration. The comparison is made between SDG-5 and Litton's G-1200 and G-6 series TDF gyros. Both Litton gyros are dry-tuned gyros; however, the G-1200 was manufactured for the purpose of a 3-axis platform and not for a strapdown usage. Therefore, although G-1200 possess better stability performance than G-6 (a strapdown unit), extensive effort is required in order to convert into a strapdown gyro. The comparison of performance is shown in Table 4.5. It is observed that the SDG-5 gyro is superior to the C-6 gyro in almost all respects (both are strapdown gyros). Although the stability performance of G-1200 gyro is slightly better than SDG-5 gyro, its dynamic range is far too narrow (1^{o}/sec steady state and 2^{o}/sec transient), therefore, it is not applicable in the strapdown environment.

TABLE 4.5 COMPARISON OF PERFORMANCE

Parameter	Units	SDG-5	G-6	G-1200
G-Insensitive Drift				
Absolute value	deg/hr	<.5	4.0	.23
Stability				
Random drift	deg/hr 1σ	.0005	.003	.0009
Shutdown	deg/hr 1σ	.0016	.01	.0023
Long term	deg/hr/yr	.01	.03	.015
Temp Sensitivity				
Uncompensated	deg/hr/^{o}F	.00059	.002	.0014
Compensated	PPM/^{o}F	1.0	---Not Compensated---	
G-Sensitive Drift				
Absolute value	deg/hr/G	<1.0	5.0	.23
Stability				
Continuous operation	deg/hr/G 1σ	.0007	.0003	.0005
Shutdown	deg/hr/G 1σ	.008	.008	.0035
Long term	deg/hr/G/yr	.02	.04	.015
Temp. Sensitivity				
Uncompensated	deg/hr/G/^{o}F	.0032	.02	.0017
Compensated	PPM/^{o}F	<1.0	---Not Compensated---	
Torquer Scale Factor				
Absolute Value	^{o}/hr/MA	160	1250	33
Stability	PPM 1σ	27	50	50
Linearity	PPM peak	25	30	No data
Asymmetry	PPM peak	3	30	No data

TABLE 4.5 COMPARISON OF PERFORMANCE

Parameter	Units	SDG-5	C-6	G-1200
Temp. Sensitivity				
Uncompensated	PPM/°F	229	250	37
Compensated	PPM/°F	< 1.0	---Not Compensated---	
Axis Alignment				
Absolute	arc sec	30	2000	26
Stability	arc sec 1σ	10	10	10
Angular Rate Cap				
Steady state	deg/sec	100	120	1
Transient	deg/sec	500	60	2
Anisoelasticity				
Uncompensated	$deg/hr/G^2$	.01	.02	.008

4.1.3 Conclusion of Trade Study

Based upon the trade study presented above, it is obvious that the NASA Standard IRU system--DRIRU-II--is far superior to the current existing gyro system operating in the strapdown environment. Better performance could be achieved using gyro platforms; however, the gyro platform suffers the long-term reliability problem. Therefore, the performance could be degenerated quite rapidly. Therefore, the NASA Standard Unit of DRIRU-II IRU system is recommended for the proposed MMS mission. It is also recommended that two DRIRU-II systems be used for the mission with one as primary unit and the other one as a backup unit. Under this consideration, the system redundancy is extremely high and the OADS mission successfulness is highly warranted.

4.2 Star Tracker System

4.2.1 Description of NASA Standard Star Tracker (SST)

The selected NASA Standard Star Tracker (SST) is an electro-optical system manufactured by Ball Brothers Research Corporation. The SST is an all electronic strapdown device which automatically searches within its field of view (FOV) for a target star. Once a target is acquired, it provides the target position and star intensity data for spacecraft attitude determination and navigation. The major components consist of a one-inch magnetically focused, magnetically deflected ITT EID F4012RP image dissector tube, a 70 mm f/1.2 lens and associated signal processing electronics.

The SST operation function consists of two modes, the search mode and track mode. When SST is activated, it immediately gets into a raster scan search mode to locate the candidate star target. When a

target is brighter than the commanded threshold setting, the SST will go into track mode and the two axis, 12 bit digital and/or analog output signal represents target position within its $8^{\circ} \times 8^{\circ}$ FOV. The SST resumes the search mode function if either the target leaves the field of view, the amplitude falls below the commanded threshold or it receives a "break command" from an external source. Regardless of the reason for returning to the search mode, it will search the remaining portion of the FOV.

The SST is very similar to the CT401 (See Figure 4-3) star tracker flown on SAS-C mission with minor differences. First, the time required to scan through the entire FOV is 10 seconds for the SST instead of 4 seconds as for the CT401. Secondly, it has the following options:

- Position output calibration
- Self Test
- Internal compensation
- Offset pointing
- Bright object protection
- Sun shade

The NASA SST specified by GSFC-S-712-9 is the basic SST with the position output calibration and bright object projection options. The SST has a ground command star magnitude threshold setting of +6, +5, +4 and +3. When the magnitude is set to acquire +6 or brighter stars, the position inherent output accuracy is 120 $\widehat{\text{sec}}$ RMS with peak errors of 240 $\widehat{\text{sec}}$. The major error components are measurable and repeatable functions of temperature, star location in the FOV and external magnetic fields. A set of calibration data sufficient to effect

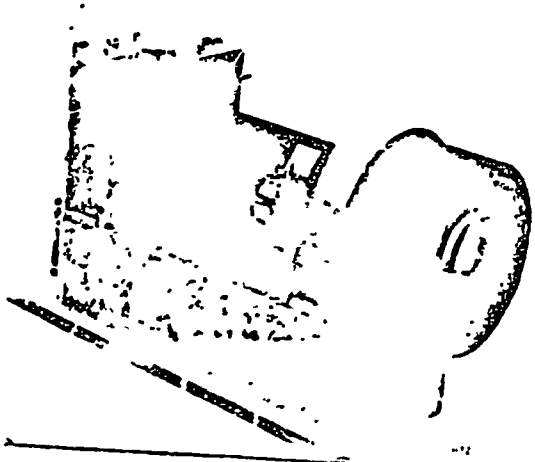


FIGURE 4-3 BALL BROTHER'S CT401 STAR TRACKER

position output accuracy of 10 arc sec RMS is supplied with each tracker having the position output calibration option. Via a ground calibration process, this position output calibration is actually accomplished by a set of third order, ten-term polynomial in each output axis. The position calibration equations are:

$$V_c = C_1 + C_2V + C_3V^2 + C_4V^3 + C_5V^2\,H + C_6VH^2 + C_7H^3 + C_8H^2 + C_9H + C_{10}HV \qquad (4.1)$$

$$Hc = d_1 + d_2V + d_3V^2 + d_4V^3 + d_5V^2\,H + d_6VH^2 + d_7H^3 + d_8H^2 + d_9H + d_{10}HV \qquad (4.2)$$

where Vc = Compensated vertical reading

Hc = Compensated horizontal reading

V,H = Vertical and horizontal readout before compensation

Cn, dn = Coefficients obtained from ground calibration

One should bear in mind that this position output calibration is a software package and can reside in the onboard computer. It is a "must" item in order to maintain the basic SST accuracy to 10 $\widehat{sec}$, therefore, it must be considered in the onboard computation consideration. When the target is acquired and the SST enters the track mode, the V and H readouts are delivered at the rate of every 40[1] ms. Thus, the position output compensator represented in Equation (4.1) and (4.2) is also evaluated at the interval of 40 ms or multiples of this. A SST system evaluation was conducted and the results are presented in Table 4.7.

(1) This time specification corresponds to Ball Brother's anticipated performance of an improved version of the BBRC-SST

TABLE 4.7 SST SYSTEM EVALUATION

CONSIDERATION FACTOR	DESCRIPTION
Life Cycle Costs	200K/systems
Design and Development	Engineering is complete. Flight build in progress for Shuttle.
Build	Most components manufactured at contractor's facility. Assembly is performed at contractor's facility. In production.
Test	Ground checkout procedures and test equipment available at suppliers. System level testing limited. Qualification test to be conducted.
Performance-Physical Attributes	
Weight and Mass Properties	Weight 25 lb including sunshade. Sunshade weight 5 lb.
Volume, Area, Size	14 x 6.5 x 6.5 not including sunshade. Sunshade 16 in diameter x 24 inch length.
Complexity-Passive Electrical	Modular electrical construction non-redundant.
Mechanical	No movable parts except shutter for sun protection.
Accessibility/Maintainability	Designed for accessibility and maintainability
Potential for 1g demonstration	Testing in 1 g field for verification with no problem.
Modularity	Yes
Technical Uncertainty	None
Performance-Functional Attributes	Thermal - 10°C to 50°C .6°/sec rate 16.7 g RMS random 10g sinusoidal
EMI-Susceptibility	SST is designed to meet EMC requirements of MIL-STD-461 equipment

TABLE 4.7 SST SYSTEM EVALUATION

CONSIDERATION FACTOR	DESCRIPTION
Maintainability/Accessibility	Modular design
Power Consumption (users)	19.0 watts
Standby power required	None
Average power required	19.0 watts
Peak power required	22 watt when shutter is operated
Mission Effectiveness	
Targeting, Pointing, Coverage	Accuracy 11.3 arc sec with all errors considered
Lifetime	3 years
Reliability	System reliability 0.9978 at end of 3 years
Survivability	Designed for launch environments and operating orbit environment
Technical Uncertainty	None
Growth	Unknown at this time
Software complexity	Simpler, only periodic update necessary
Ground support-updates	Require storage of star catalog-smaller sub catalog generation
Downlink Processing	Require star identification and sub catalog linking
In-Flight Calibration	Easier under low slew rate
Ground Checkout	Verification of system operation not complex
Testability	Adequate test output available
Schedules	
Design	System developed and in production. No facility mod required.
[illegible]	Subcontract procurement
Tests	Interface available to perform self check and verify system level interfaces

4.2.2 Trade Study

A trade study was conducted between the NASA Standard SST and other celestial sensor systems using celestial stars for the attitude determination function. First, a trade study between the SST and a star mapper is presented. Second, a comparison between the SST and other fixed-head star trackers is presented and finally, a comparison with the next generation star tracker-Charge Coupled Device (CCD) is presented. The two significant factors considered in the trade study are:

(1) The system must be able to operate in both the inertial hold and dynamic environments of MMS' solar, earth, and stellar missions.

(2) The system must possess high accuracy with good data coverage probability.

4.2.2.1 Comparison with Star Mapper

The Star Mapper is generally a slit type sensor which uses the spacecraft rotational motion to provide search and sensing functions. The spacecraft rotation causes the sensor to scan the celestial sphere. As the star image on the focal plane passes a slit, the star is sensed by the detector. If the signal is above the threshold, a pulse is generated by the electronics, signifying the star presence. The crossing time of the first slit and the elapsed time between the crossing of first slit and the following one(s) together with the star catalog provide the target star attitude information.

The trade study was made between the SST and a Bendix SSA star mapper. The trade study results are presented in Table 4.8.

TABLE 4.8 COMPARISON BETWEEN SST AND BENDIX SSA STAR MAPPER

CONSIDERATION FACTOR	SST	BENDIX SSA STAR MAPPER
Dynamic Range (^{o}/sec)	Zero to 0.7	0.05 ~ 6
Accuracy 2σ - Arc Sec)	10	5
Reliability (MTBF-Hrs)	188,680	429,400
Cost (Per Unit)	200K	1,000K
Weight (lbs.)	25	63
Power (watts)	19	7
Size (in^3)	14 x 6.5 x 6.5	12 x 12 x 10
Star Magnitude	+6, +5, +4, +3	+4
Software Requirement	Less Requirement	More Complex
In-Flight Calibration	Easy	Difficult

From Table 4.8, one can see that although the Star Mapper has better accuracy and reliability, the SST is superior in all other aspects. The most important factor is that the Star Mapper cannot be used in the stellar mission and the inertial hold mode of the solar mission because it depends upon the spacecraft motion to acquire stars. Therefore, for MMS mission consideration, the SST should be selected over the star mappers.

4.2.2.2 Comparison with other Fixed Head Star Trackers

A comparison between the SST and other star trackers was conducted. The comparison is shown in Table 4.9. Although both the Honeywell and TRW tracker have better accuracy than the SST, there are two major problems to use these two trackers for MMS Mission. First, the star

tracker sensitivity for those two trackers are +8 and brighter. This creates numerous data and storage and processing problems for onboard systems because there are about 14,000 stars for +8 magnitude and brighter. Secondly, the smaller FOV of those two trackers impose a problem under dynamic environment since fewer good quality star signals may be obtainable before it leaves the FOV. Because of those two restrictions, both the Honeywell and TRW star trackers can be used only during the stellar mission but may have difficulty for the solar and Earth missions. There are other star trackers such as the ITT tracker which has large FOV and dynamic range but do not possess required accuracy. Thus, the SST is still superior for MMS OADS usage.

TABLE 4.9 COMPARISON OF STAR TRACKER CANDIDATES

STAR TRACKERS	SENSITIVITY (MAGNITUDE)	FOV	CALIBRATED ACCURACY (2)
SST	+6, +5, +4, +3	$8^{\circ} \times 8^{\circ}$	10 sec
Honeywell Photon Counting Star Tracker	+8	$2^{\circ} \times 2^{\circ}$	3 sec
TRW PADS Tracker	+10	$1^{\circ} \times 1^{\circ}$	3 sec

4.2.2.3 Comparison with CCD Unit

The CCD Star Tracker uses a charged-coupled imaging array as a detector in place of an image dissector. The detector is a buried-channel, line-transfer, charge-coupled device (CCD), with vertical and horizontal picture elements. A typical detector contains 488 vertical by 380 horizontal picture elements within an active image area of 8.8 mm by 11.4 mm. The detector is cooled to an operating temperature below $0^{\circ}C$ with an array of peltier affect thermo-electric junctions.

The detector array is readout with high speed microprogrammable logic. At those places in the field of view where star energy is detected, the operation is slowed to allow analog to digital conversion of the signal charge of each picture element, or "pixel" in the region. A micro-processor is employed to compute the location of the centroid of the star images to an accuracy of about 1/10 of the inter-pixel distance and to provide sequencing and control functions. The CCD unit posses some distinct advantages over the image dissector star tracker (i.e., SST). Those are: the ability to track multiple stars simultaneously, no sensitivity to magnetic fields, and improved accuracy. At the present time, TRW, BBRC, and Honeywell are evaluating the performance of CCD in the laboratory using experimental breadboard models. The preliminary characteristics of both the BBRC and TRW CCD units are presented in Table 4.10. Because the CCD unit has approximately the same FOV and dynamic range as the SST unit with better accuracy, it can be considered as the primary alternative for the SST for the MMS mission once it is fully tested and qualified.

4.2.3 Conclusion of Trade Study

Based upon the above trade studies, we conclude that the NASA-Standard Star Tracker is the best candidate for MMS mission at the present time. An alternative of using the CCD unit after full development is recommended. It is also recommended that the SST should be used under the condition that the spacecraft slew rate is less than 0.5^{o}/sec in order to maintain its accuracy. Considering most of the MMS mission modes, this restriction is justified and the method of using the SST in combination with DRIRU-II unit will be presented in the next section.

TABLE 4.10 - PRELIMINARY CHARACTERISTICS OF CCD STAR TRACKER

CHARACTERISTIC	UNITS	TRW	BBRC
Field of View			
Total	deg	6.60 x 8.53	7.1 x 9.2
Instantaneous	arc min	.81 x 1.35	
Optical System			
Focal Length	mm	76	70
f/NO		.87	
Transmission		.75	
Detector			
Type		Fairchild CCD	
Number of Elements		488 x 380	488 x 380
Image Area	mm	8.8 x 11.4	8.8 x 11.4
Configuration		Front Illuminated, Interline Transfer	Front Illuminated, Interline Transfer
Electronics			
Integration Time (for +6 M Star)	sec	.100 max	.100 max
Readout Rate (for +6 M Star)	sec	.100	.100
Star Position Output			
Vertical	Digital	12 Bit Serial	
Horizontal	Digital	12 Bit Serial	
Star Magnitude	Digital	12 Bit Serial	
Update Interval (for +6 M star)	sec	.100 max	.100 max

TABLE 4.10 PRELIMINARY CHARACTERISTICS OF CCD STAR TRACKER - (Continue

CHARACTERISTIC	UNITS	TRW	BBRC
Accuracy (1 sigma)			
Vertical	arc sec	2.4	
Horizontal	arc sec	4.1	
Total	arc sec	4.75	5.0
Physical			
Weight	lb	7	7
Volume	in.	6 x 6 x 12	
Power	watts	9.5 @ 28 VDC	26 @ 28 VDC
Development			
Status		Breadboard in Test	Breadboard in Test

4.3 GPS Magnavox Receiver/Processor Assembly

4.3.1 Global Positioning System Summary

The Global Positioning System (GPS) consists of 24 navigation space vehicles (SV) at an altitude of 20,182 km with an orbital period of 12 hours. A minimum of six and a maximum of eleven space vehicles will be visible at one time. The Host Vehicle (HV) GPS Receiver/Processor Assembly (R/PA) will receive data from the GPS space vehicles when they are five degrees above the horizon. The expected position and velocity accuracies for a 500 km circular polar orbit were determined by a computer program at Martin Marietta and are listed below. [Satin, 1]

Position: 12 M (1σ)

Velocity: .0061 M/sec (1σ)

Time: 9 nanosec (3σ)

The above time accuracy was taken from [Birnbaum, 1]. The problem of determining the Host Vehicle's position and velocity was solved in two phases. The first phase was to determine the GPS space vehicle ephemerii using the ground control segment. The second phase was to determine the Host Vehicle's position and velocity accuracies using the GPS pseudo-range measurements for navigation purposes. The a priori user position and velocity errors in the radial (U), downtrack (V) and out-of-plane (W) coordinate are listed below:

U - 9144 M - 3σ	$\dot{U}$ - 21.4 M/S - 3σ
V - 14240 M - 3σ	$\dot{V}$ - 7.6 M/S - 3σ
W - 2134 M - 3σ	$\dot{W}$ - 3.05 M/S - 3σ

The position and velocity errors computed are conservative since it was later determined the error in the Earth's gravitational constant was an order of magnitude too large.

4.3.2 Host Vehicle's Command Interface to the Magnavox Receiver/ Processor Assembly

The Receiver/Processor Assembly receives from the Host Vehicle external control signals, pulse commands and data commands. [APL, 1] gives a complete description of these commands. A summary of the important commands is given below.

The significant external control commands are:

- HV Thrust Flag - The R/PA accepts a command which indicates an adjustment to the HV orbit is in progress.
- Time Strobes - The R/PA can receive four independent time strobes from the Host Vehicle. For each time strobe, the R/PA time code generator contents are saved with the leading edge of the HV time strobe and stored in R/PA memory for transmission to the ground. Each HV time strobe has an identification and quality flag.

The principal function of pulse commands is to operate the system. The maximum number of pulse commands is twelve. Specific functions of the pulse command are:

- R/PA power on/off
- Select R/PA mode of operation. Mode of operation include boot, command and navigate.
- Selection of R/PA oscillator
- Operation of R/PA time code generator

Data commands are used to initialize the system. Specific data commands are listed below.

- Select data file output - The R/PA generates twelve data files for output. The user selects which files are to be sent to the Host Vehicle.
- Initialize the R/PA time code generator.
- HV almanac upload - The HV almanac is required at initialization.
- GPS space vehicles almanac upload - The GPS space vehicles almanac upload is optional. If it is not uploaded, it will be collected from the GPS space vehicles by an almanac collection command. The GPS space vehicle almanac is used by the R/PA for selecting GPS space vehicles for navigation.
- Other data commands are set receiver channel, set mode of operation, and memory dump.

4.3.3 GPS Magnavox Receiver/Processor Assembly Output to the Host Vehicle

The Host Vehicle is required to sample ten analog measurements at least once every eight seconds and 16 binary measurements at least once every four seconds. These measurements represent the health and status of the R/PA.

The R/PA outputs twelve data files. Each data file is output once every six seconds or multiple of six seconds depending upon user requirements. As indicated in the previous section, the user can specify which files to output. File 7 (navigation best estimate) is the file of interest to OADS; although, at the last interchange meeting at GSFC, it was recommended that we use both files 6 and 7 since they may merge sometime in the future.

The contents of File 7 are listed in Table 4.11. Position and velocity in File 7 are given in the Earth Centered - Earth-Fixed (ECEF) Coordinate System which is designed as follows.

TABLE 4.11 CONTENTS OF FILE 7

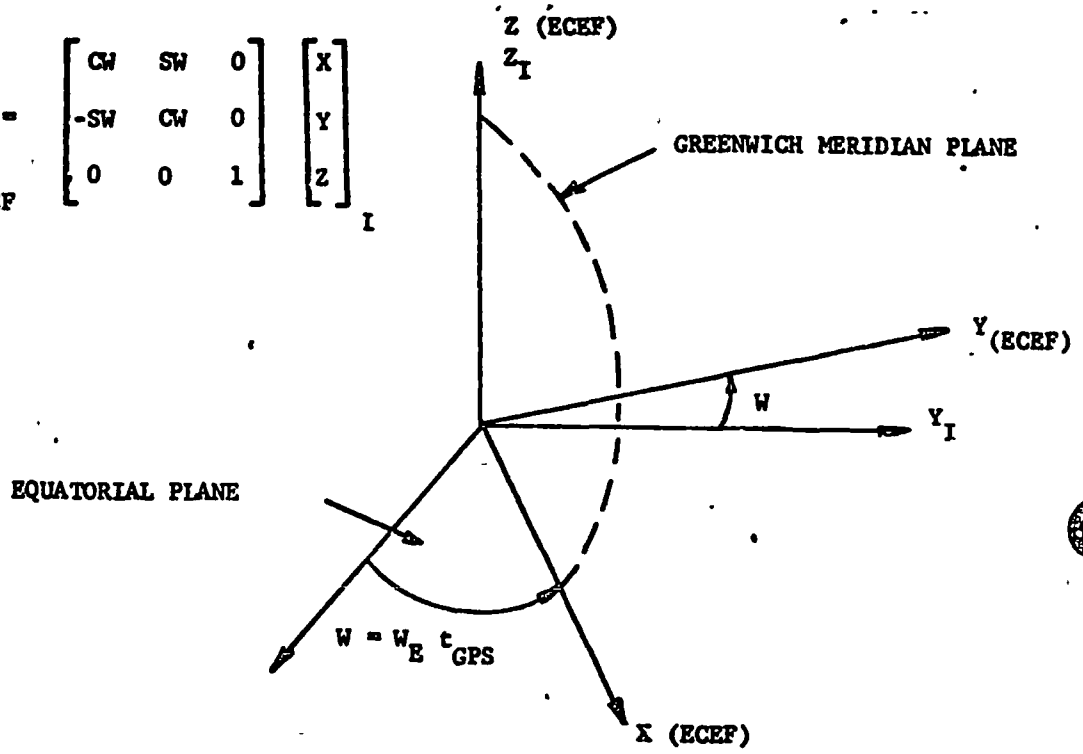
- T_{R1} - USER TIME CODE GENERATOR MARK INDICATING WHEN THE TIME MARKER (EPOCH 1) ON THE GPS SIGNAL ARRIVED
- TIME - RECEIVER'S GPS TIME OF ARRIVAL OF EPOCH 1
- X, Y, Z - USER POSITION IN THE EARTH CENTERED - EARTH-FIXED (ECEF) COORDINATE SYSTEM
- V_x, V_y, V_z - USER VELOCITY IN ECEF COORDINATE SYSTEM
- C_D - USER DRAG COEFFICIENT
- σ_p^2 - USER POSITION VARIANCE
- σ_v^2 - USER VELOCITY VARIANCE

- X is in the true equatorial plane in the direction of Greenwich Meridian
- Z is along the true earth spin axis, positive in the Northern Hemisphere
- $\hat{Y} = \hat{Z} \times \hat{X}$

OADS requires the position and velocity in the inertial reference frame defined in Section 3.2. The transformation between the ECEF and inertial reference coordinate system is shown in Figure 4-4. The ECEF coordinate system is identical with the Earth reference coordinate system defined in Section 3.2.

A summary of the relationship between the GPS Receiver/Processor Assembly and the Host Vehicle is shown in Figure 4-5.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{ECEF} = \begin{bmatrix} CW & SW & 0 \\ -SW & CW & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{I}$$

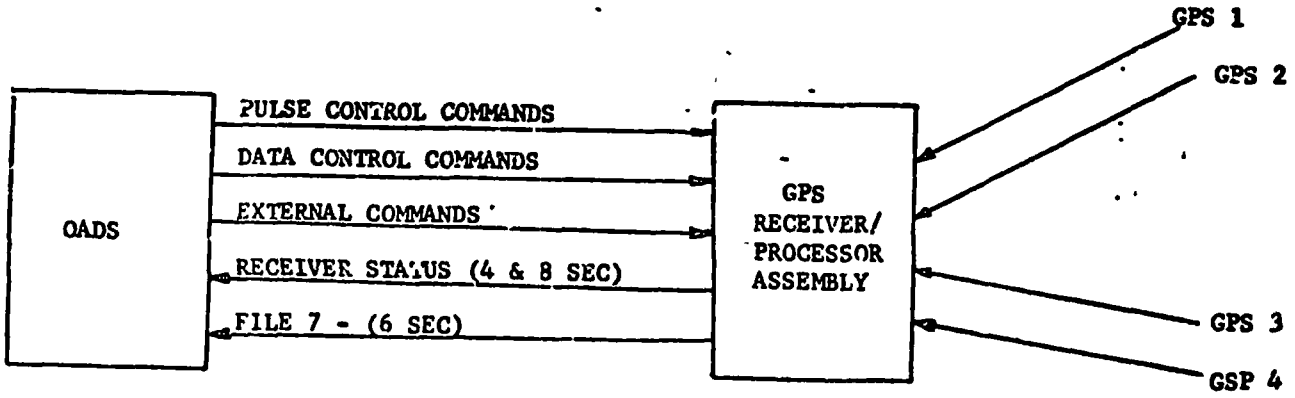


W_E = Earth Rotation Rate

t_{GPS} = GPS Time

INERTIAL REFERENCE AND EARTH CENTERED - EARTH FIXED COORDINATE SYSTEMS

FIGURE 4-4



SUMMARY OF INTERFACE BETWEEN OADS AND GPS RECEIVER

FIGURE 4-5

5.0 MISSION PERFORMANCE STUDY

A performance study was done on the attitude determination system (ADS) for the missions defined in Section 3. Section 5.1 describes the simulator and the IMU and star tracker update algorithms used in this study. Sensors errors are given in Section 5.2. Section 5.3 presents the sensitivity study done on the OADS sensors and onboard integrator. The performance results for each mission are presented in Section 5.4.

5.1 Simulator Description

A schematic diagram of the simulator used in the performance study is shown in Figure 5-1. The system has two basic subsystems: the inertial measurement unit (IMU) subsystem and the star tracker update subsystem. Each subsystem is described in the following sections.

5.1.1 Inertial Measurement Unit Subsystem

A detailed IMU block diagram is shown in Figure 5-2. Three two degree of freedom dry-tuned gyros are modelled to provide redundant vehicle rate measurements. Pulse counts are sampled at 20 Hz frequency and converted into angular rates in deg/sec by rate reconstruction software. The raw channel output angular rates are compensated for static and dynamic errors by compensation software. The 6 compensated angular rates, which represent a set of redundant vehicle 3-axis body rates, will pass through the onboard data reduction processor and the vehicle rate vector $\underline{W}_v$ is obtained. The $\underline{W}_v$ vector will be used for attitude knowledge, star update compensation and the strapdown integrator. As shown in Figure 5-2, the rate vector can also be used for attitude control of the spacecraft. A minimum variance weighted least square processor is used as the data reduction processor [Yong, 1] . The reduced body rate vector $\underline{W}_v$ is fed into a numerical integrator. The quaternion vector is used as the internal state vector with periodic update

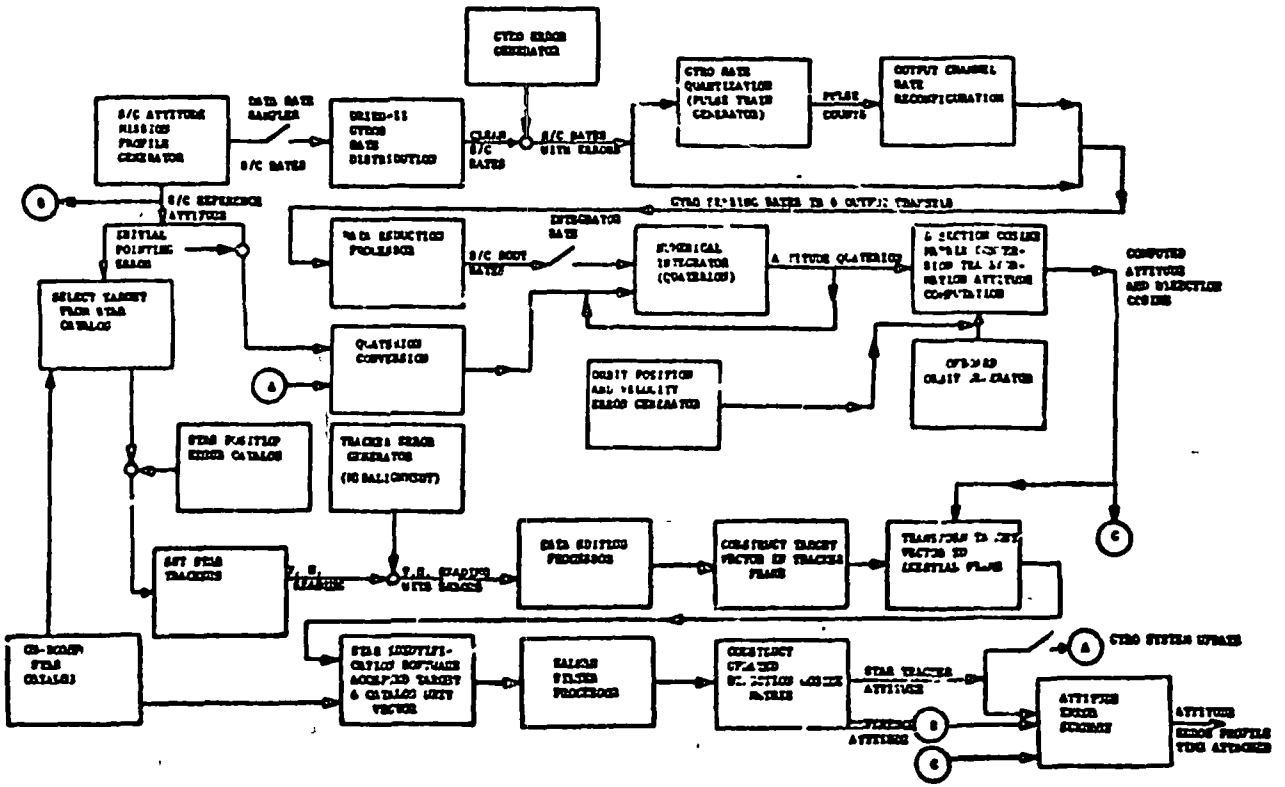


FIGURE 5-1 OADS COMPUTER SIMULATOR

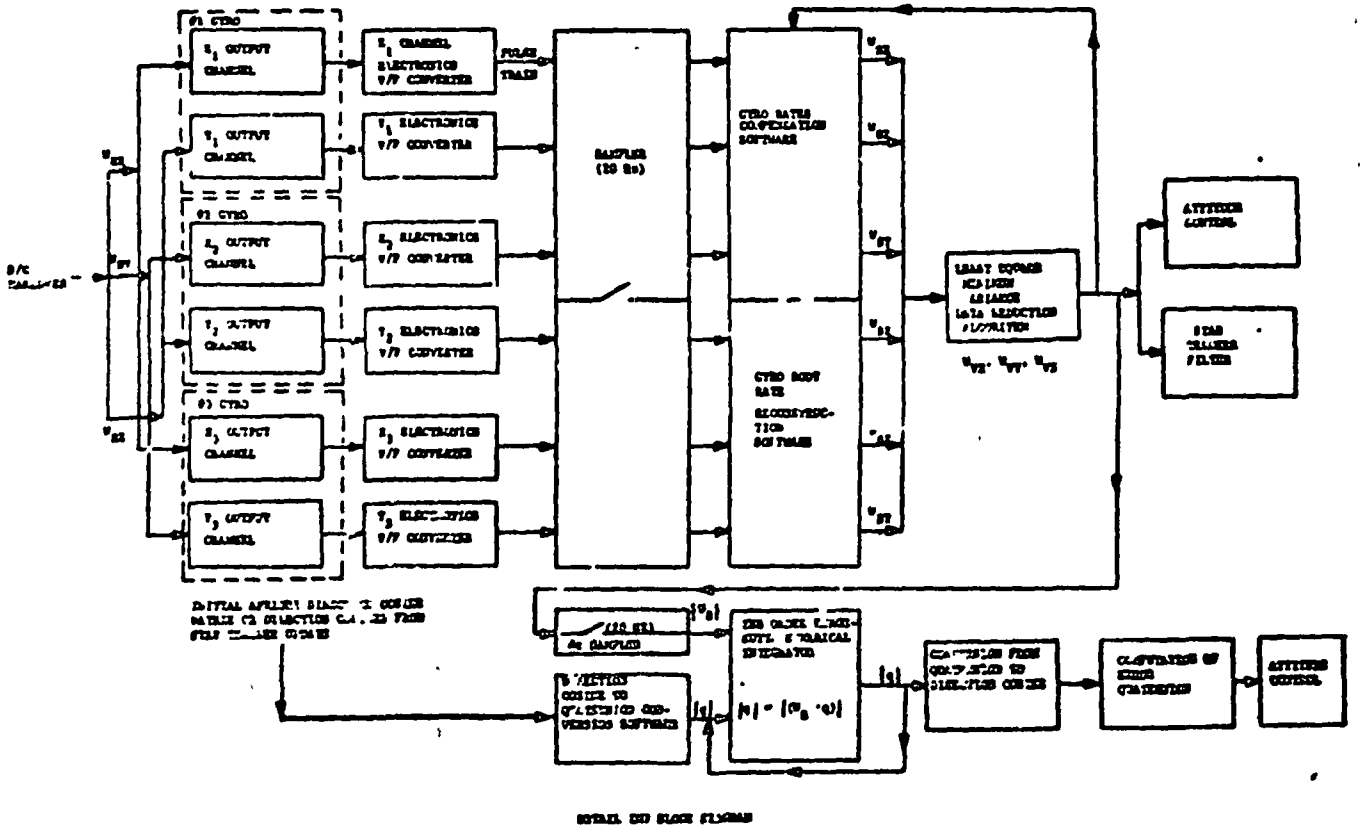


FIGURE 5-2 DETAIL IMU BLOCK DIAGRAM

from the star tracker attitude update software. A second order Runge-Kutta integrator with a 20 Hz integration frequency is currently selected for the strapdown system.

5.1.2 Star Tracker Subsystem

Figure 5-3 shows the detailed star tracker algorithm block diagram. Yong [Yong, 2] discusses the star update analytic details of the star tracker subsystem.

When the star tracker(s) acquire and lock on to target stars, a set of tim-eattached raw two axes V (vertical) and H (horizontal) readouts are generated at the rate of 40 ms. Temperature and magnetic sensitivity to the V and H readouts are compensated by a 10 term 3rd order polynomial onboard compensator. Bad V & H readings are rejected via the data editing software. When more than one tracker acquires a star, the V and H reading from each tracker is synchronized in time using the rate knowledge obtained from the IMU if the satellite possesses angular rates in inertial space. The target unit vectors in the star tracker frame are then constructed for attitude determination processing.

A star catalog is generated from SAO star catalog which contains the right ascension, declination, visual magnitude and other essential information. For star magnitudes of +5.0 and brighter about 1500 stars will be stored onboard if enough space is available. A trade study should be conducted to determine the number of stars to be stored onboard for the MMS missions. Star subcatalogs are generated containing stars within ± 5.6 degrees radius from the boresight of each tracker. A direct matching algorithm is employed for star identification between the acquired target and catalog stars. Once identification is confirmed,

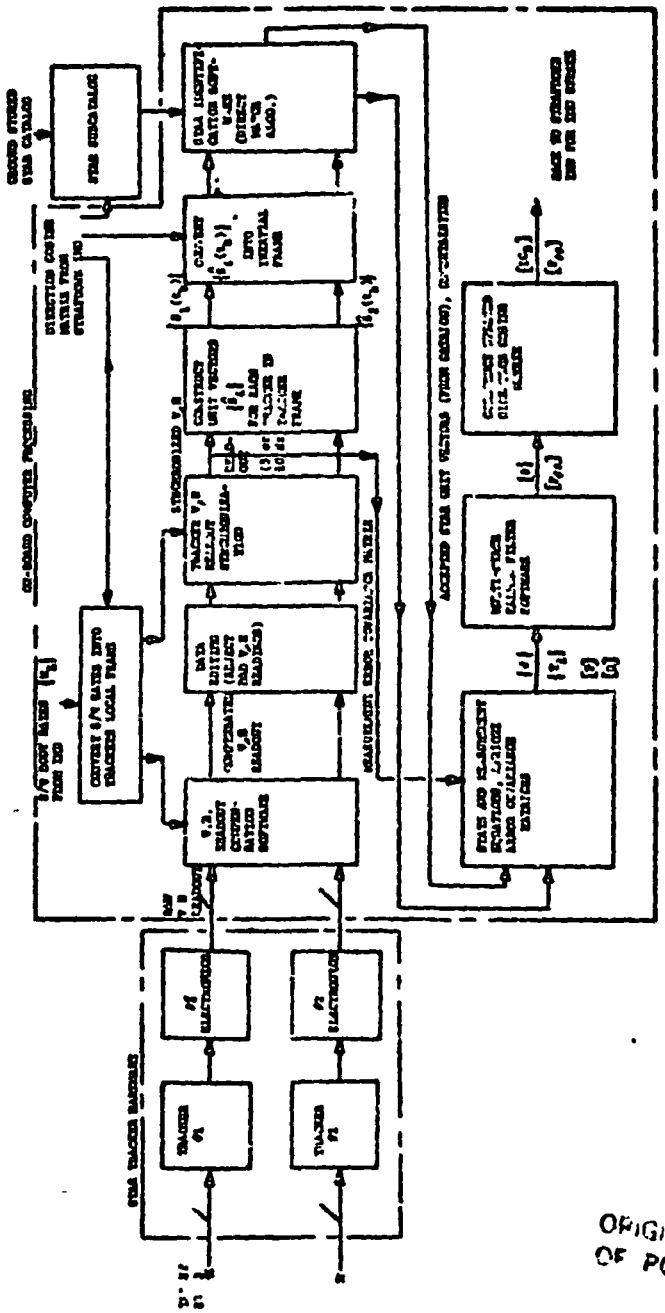


FIGURE 5-3 STAR TRACKER SUBSYSTEM

both the target unit vector in the tracker frame and selected catalog stars in the inertial frame are sent to the filter for the star attitude update process.

A multi-stage sequential Kalman filter was developed for the star attitude determination system. The state variables are defined as the small angle rotations between the computed (from IMU) and the true vehicle body frames. The filter is operated at a frequency of 5 Hz or less. A measurement equation is established relating the measurement quantities (star unit vectors) and state variables. Through filter processing a best estimate of the state variables and its error covariance matrix is obtained. When the update of the IMU is desired, the small error angles are converted into the quaternion errors to update the strapdown IMU system.

5.2 Sensor Error Sources

This section discusses the nominal gyro, star tracker, and GPS errors used in the mission performance study.

5.2.1 Gyro Errors

The nominal gyro errors used in the mission performance study are given in Table 5.1. The nominal errors were given by the vendor [Teledyne, 1]. These nominal errors were based upon the testing of the SDG-5 gyros in the past two years and are believed to be the maximum possible uncertainty errors for the SDG-5 gyros used in the DRIRU-II system. Explanations should be given regarding the nominal parameters shown in Table 5.1. Pertaining to the gyro-to-gyro misalignment in Table 5.1, Teledyne has indicated that the individual gyro to the DRIRU-II mounting block misalignment can be measured on ground to the accuracy of 10 arc seconds. This is an absolute measurement accuracy

Teledyne has been able to achieve on ground; therefore, it should be considered at more than 3σ accuracy value. It is, however, questionable that the misalignment values will remain the same after the launch environment, flight temperature variation, and other environmental impacts. An onboard calibration method has been studied by Martin Marietta [Martin Marietta, 1]. From the simulation results, it is believed that we can calibrate the gyro-to-gyro misalignment to the value of accuracy uncertainty of 20 arc seconds 2σ. This is a rather conservative value, especially if we know the non-orthogonality as accurate as 5 arc-seconds.

TABLE 5.1 TELEDYNE SDG-5 TDF NOMINAL ERROR PARAMETERS (2σ)

ERROR SOURCE ITEMS	ACCURACY UNCERTAINTY
Gyro-to-Gyro misalignment	20 arc-sec
Scale Factor	
Linearity Asymmetry	50 PPM
Temperature Sensitive	2 PPM/^{o}F
Nonorthogonality	20 arc-sec
Bias Drift	0.001^{o}/hr
Temperature Drift	2 PPM/^{o}F
Motor Dynamics	0.006^{o}/hr
Cross Coupling	0.01^{o}/hr
Uncompensable Errors	0.004^{o}/hr
Random Noise Standard Deviation	0.1×10^{-4}
Temperature Variation During Simulation	5^{o}F

The SDG-5 gyro physical properties used for the simulation are given in Table 5.2.

TABLE 5.2 SDG-5 GYRO PHYSICAL PROPERTIES

Transverse Moment of Inertia	2380 gram-cm^2
Polar Moment of Inertia	1600 gram-cm^2
Angular Momentum	1 x 10^6 cgs unit

The 1σ scale factor accuracy given by the vendor is 25 PPM for linearity and asymmetry, and 1 PPM/^{o}F for temperature sensitive. Assuming the accuracy uncertainty follows normal distribution, a 2σ value is then used to represent the scale factor uncertainty. This assumption is also used for the other error source uncertainties, such as bias, dynamic errors, and uncompensable errors, etc. The nonorthogonality accuracy is taken the same as the misalignment value (20 arc-seconds - 2σ). However, it was later learned from Teledyne that this value can be measured to the accuracy of 5 arc-seconds - 10σ uncertainty on ground and little change is anticipated due to launch and flight maneuvering impact. Thus, the 20 arc-second nonorthogonality value is a very much exaggerated value to be used for the nominal error simulation. The random noise value is obtained from the SDG-5 gyro PSD and first difference test value conducted here at Martin Marietta [Martin Marietta, 2]. The value of transverse moment of inertia given in Table 5.2 is computed from the SDG-5 gyro size and configuration [Yong, 3].

5.2.2 Star Tracker Update Error Sources

The star tracker is used to periodically update the IMU reference to maintain attitude determination accuracy. The attitude update error sources can be divided into star catalog and star tracker errors.

5.2.2.1 Star Catalog Error

Star position uncertainty - The average star position error for the SAO star catalog is 0.5 arc-second 1σ. Thus, we assume the 2σ position error is 1 arc-second.

Star Motion Error - The highest annual motion for +5 visual magnitude and brighter stars if 0.75×10^{-3} deg/yr. This error depends upon the frequency of updating the onboard star catalog. If the star catalog is updated every half year, then the maximum possible position error due to star motion is 1.35 arc-second.

Aberration Error - Aberration is the apparent shift of the star position due to S/C motion, and for an earth-orbiting S/C is approximately 5 arc-seconds maximum. However, this error can be estimated by a software relation and can be estimated and reduced to 1 arc-second 2σ.

Other star related errors such as faint background, multiple star mis-identification and quantization errors are relatively small compared to the above three error sources. Therefore, they are assumed negligible and are not included in the simulation error model.

5.2.2.2 Star Tracker Errors

Basic Tracker Accuracy - The BBRC-SST star tracker can achieve 10 arc-seconds (2σ) accuracy after onboard temperature, magnetic and star intensity compensation if the S/C slew rate is lower than 0.5°/sec. For MMS missions, the star tracker update shall always operate during the non-maneuvering mode, therefore, the 10 arc-second basic accuracy can be assumed [Clevinger, 1].

Quantization Error - The star position of BBRC-SST is indicated by a 12-bit digital word for V & H and will produce an error of about 2 arc-seconds (2σ).

Boresight Axis Misalignment - The tracker can maintain a null accuracy of 10 arc seconds (2σ). At the time being, it is not certain how much this misalignment error can be removed from on-orbit calibration. Thus, a 10 arc-second (2σ) misalignment error is assumed.

Time Tagging Error - With the S/C in the Earth mission environment the V & H readout time differential must be compensated by knowing the orbital rate vector in the tracker frame. Assuming the time tagging accuracy of 5 ms and the rate error of 0.1×10^{-4} /sec, this error is negligible.

The star tracker basic accuracy, quantization and boresight misalignment errors are modelled in the simulation program. The time tagging error, although unimportant in the MMS case, is also modelled. The significant star tracker update error sources are summarized in Table 5.3.

ERROR SOURCE	ERROR LEVEL (2σ) ARC SEC
Star Catalog Error	
Position Uncertainty	1.0
Star Motion Error	1.35
Aberration Error	1.0
Star Tracker Error	
Basic Accuracy	10
Quantization	2
Boresight Axis Misalignment	10

TABLE 5.3 STAR UPDATE ERROR SOURCE SUMMARY TABLE

5.2.3 Global Positioning System Orbit Errors

The global positioning system errors used in the mission performance study are listed in Table 5.4 [Sutin, 1]. The satellites position and velocity errors in the performance study were modelled using a Gaussian distribution.

TABLE 5.4 GPS ORBIT ERRORS USED IN PERFORMANCE STUDY

Position	24 m (2σ)
Velocity	.0122 m/sec (2σ)
Time	18 nano sec (2σ)

As indicated in Section 4.3, the GPS Magnavox receiver sends to the user position, velocity, and time every six seconds. The errors in Table 5.4 apply only to the beginning of the six-second interval. Between the six-second intervals, the user satellite's orbital position and velocity deteriorates. The amount of deterioration depends upon the onboard orbit propagator accuracy. How much the user orbit accuracy deteriorates and the effect on the satellite's attitude is discussed in Section 5.4.4.

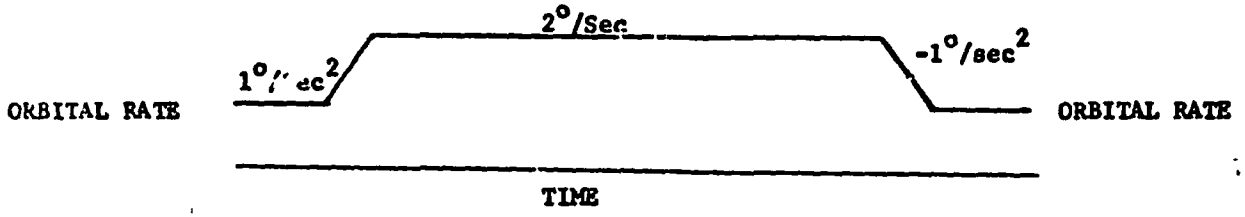
5.3 Sensitivity Study

A sensitivity study was done on several integration methods, gyro errors, star tracker errors, and orbit errors. Results are presented in the following sections. The pointing error angle used in the Earth missions and sensitivity study is defined as the RSS value of the pitch and roll error angles. The pointing error used in the solar and stellar missions is defined as the RSS value of the right ascension and declination error angles.

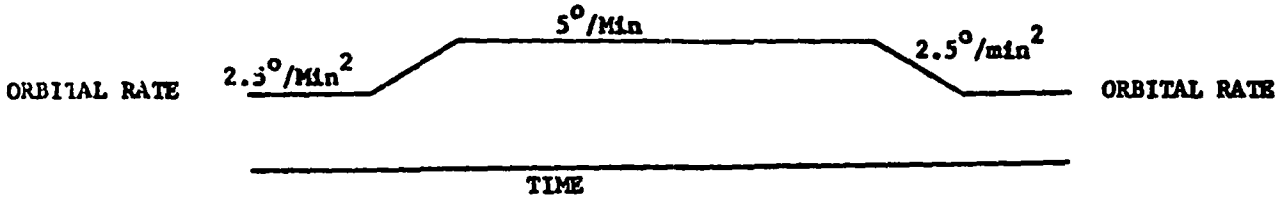
5.3.1 Maneuver Profiles

The maneuver profiles used in the performance and sensitivity studies are shown in Figure 5-4. The sensitivity study maneuver profile was the 2°/sec., 20 degree maneuver for a 400 km circular orbit. The initial attitude state was zero yaw and roll errors with a .004 degree pitch error. All maneuvers were along the pitch axis.

- 2°/SEC MANEUVER PROFILE



- 5°/MINUTE MANEUVER PROFILE



MANEUVER PROFILES

FIGURE 5-4

5.3.2 Sensitivity Due to Numerical Integration

Three numerical integration methods were investigated to determine which integration method provided the best pointing accuracy with a minimum computational requirement. The three integration methods were first order Taylor expansion, second order Runga-Kutta and fourth order Runga-Kutta. The sensitivity study maneuver profile discussed in Section 5.3.1 and nominal gyro and orbit parameters discussed in Section 5.2 were used. The criterion used in determining which integration method and step size would be satisfactory was to maintain a pointing error of less than .01 degree during the 2 deg/sec maneuver, for a 400 km circular orbit.

The results of the study are shown in Table 5.5. From Table 5.5 the first order Taylor expansion method did not satisfy the pointing requirement. The second and fourth order Runga-Kutta methods do satisfy the pointing requirement for integration stepsizes of 0.05 and 0.01 seconds and give nearly identical results.

Because the effect of the control system on the satellite attitude motion is unknown and high frequency motion may exist, the 0.05 second integration stepsize was selected. Since the second order Runga-Kutta method requires less computations and storage than the fourth-order Runga-Kutta, the second order Runga-Kutta method was selected with the fixed integration stepsize of 0.05 seconds, for use in the performance study as well as the OADS onboard integration requirement.

The integration error occurs from the size of the integration stepsize (truncation error) and for a maneuver whether the start time of the integration interval coincides with the start time of the maneuver acceleration phase [Yong, 1]. The latter error is illustrated

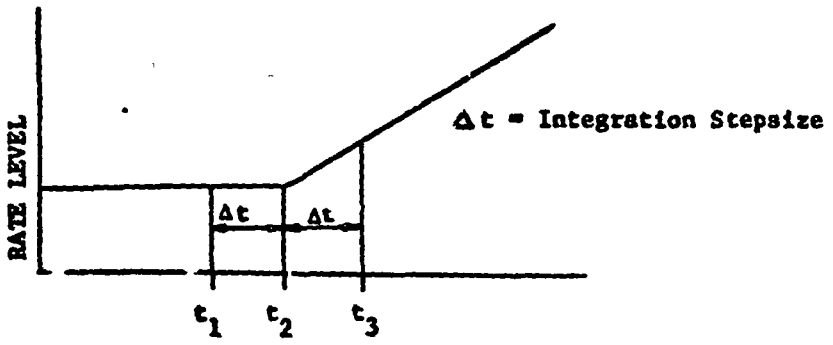
in Figure 5-5. As shown in Figure 5-5 (a) the start of the integration interval coincides with the start of the acceleration phase and no integration error occurs. In Figure 5-5 (b) the start occurs before the acceleration phase and an integration error is introduced. Since it is doubtful the integration interval will start at the acceleration phase or any attempt will be made to coordinate with the start of the integration interval and maneuver acceleration phase, this type of integration error should be considered. This error can be reduced by using a small integration stepsize.

For the OADS study, the 0.05 integration stepsize was chosen to handle the above errors and still maintain a 0.01 degree pointing accuracy for a maneuver rate of 2°/sec. If the maneuver rate is lower or a constant rate exists, a larger stepsize can be used. Therefore, for OADS-type missions it may be desirable to have the capability to change the integration stepsize onboard as a function of sensed rates or by ground command for different maneuver and tracking sequences. Studies will be required to determine the effect of changing integration stepsizes on the satellite computer configuration since less computing power would be required at a larger integration stepsize. This would also have an impact on the operational aspects of the missions.

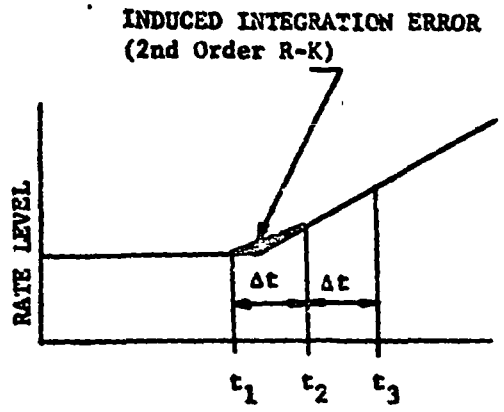
5.3.3 Gyro Error Sensitivity Analysis

Sensitivity of various gyro error sources were studied using the maneuver profile discussed in Section 5.3.1. The gyro error parameter uncertainty values in Table 5.1 were used as the nominal case.

Simulation runs were made by setting each individual error source to zero while the other error sources remained at the nominal value. The result on the pointing error of zeroing each error source is compared



(a)



(b)

INTEGRATION ERROR DIAGRAM

FIGURE 5-5

INTEGRATION STEPSIZE (SECONDS)	MAXIMUM POINTING ERROR (DEGREE)		
	1ST ORDER TAYLOR EXPANSION	2ND ORDER RUNGA-KUTTA	4TH ORDER RUNGA-KUTTA
0.05	0.055	0.007	0.007
0.10	0.105	0.008	0.008
0.20	----	0.024	0.013

TABLE 5.5 POINTING ERROR FOR VARIOUS INTEGRATORS AT VARIOUS INTEGRATION STEPSIZES

with the pointing error of the nominal case. The purpose of this sensitivity analysis is to see what are the dominant error sources for the type of mission being considered. The results are given in Table 5.6. As shown by the table, the scale factor, gyro-to-gyro misalignment and nonorthogonality are the dominant error sources. A sensitivity study on the three dominant error sources was conducted The results are presented below.

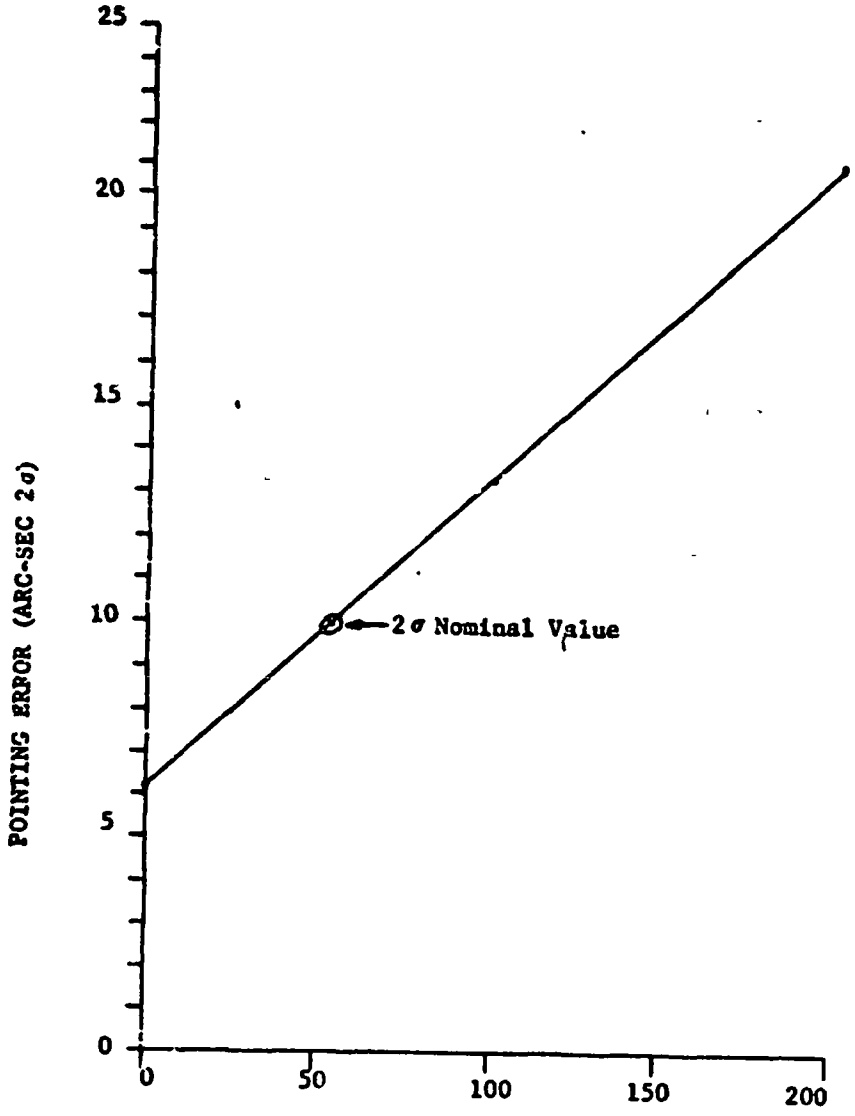
Gyro Scale Factor Error - The nominal 2σ scale factor value is 50 PPM for linearity and asymmetry, 2 PPM/^{o}F for temperature sensitivity. The linearity and asymmetry are the dominant error sources compared to the temperature sensitivity in the scale factor sensitivity study; therefore, only the linearity and asymmetry scale factor is varied except at zero when both are set equal to zero. The range of variation is from 0 to 200 PPM and the result is shown in Figure 5-6.

Gyro-to-Gyro Misalignment Error - The assumed nominal 2σ gyro-to-gyro misalignment value is 20 arc-seconds. As indicated by Teledyne [Teledyne, 1] this is a conservative misalignment accuracy uncertainty from on-orbit calibration. Assumptions are made that the misalignment value can be compensated by the onboard compensator to a value of 20 arc second uncertainty. The range of misalignment variation is from 0 to 40 arc-seconds for all three gyros in the DRIRU-II systems. The results are shown in Figure 5-7.

Nonorthogonality Error - The nominal 2σ nonorthogonality error is 20 arc-seconds. The error was varied from 0 to 40 arc-seconds and the results are shown in Figure 5-8.

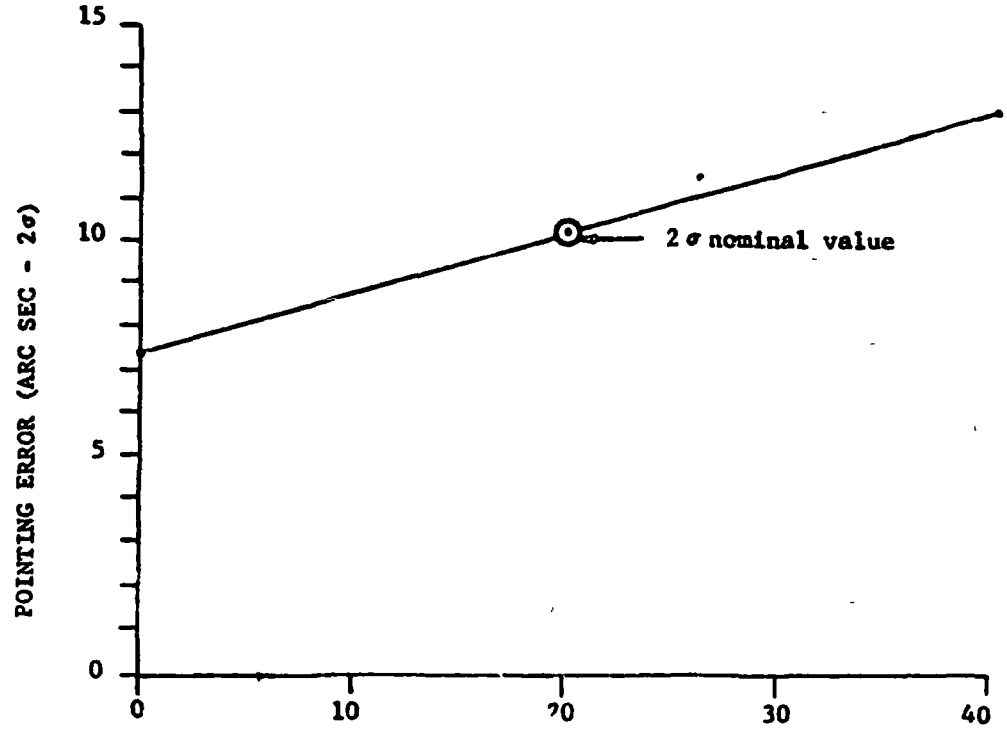
TABLE 5.6 POINTING ERROR FOR EACH INDIVIDUAL GYRO ERROR TERM

Gyro Error Term	Pointing Error Difference From the Nominal Case (arc-sec-2σ)
Scale Factor	3.768
Gyro-to-Gyro Misalignment	2.312
Nonorthogonality	1.270
Drift Bias	0.251
Motor Dynamics	0.0
Random Noise	0.011
Dynamic Errors	0.018



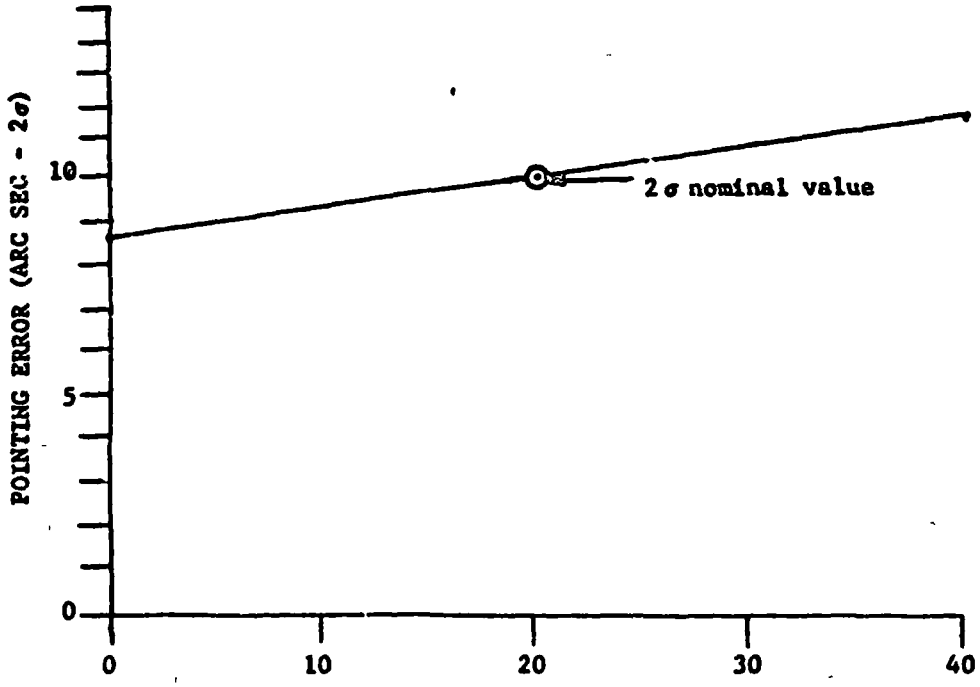
POINTING ERROR (2σ) AS A FUNCTION OF SCALE FACTOR

FIGURE 5-6



POINTING ERROR (2σ) AS A FUNCTION OF GYRO-TO-GYRO MISALIGNMENT

FIGURE 5-7



POINTING ERROR (2σ) AS A FUNCTION OF GYRO NONORTHOGONALITY ERROR

FIGURE 5-8

5.3.4 Star Tracker Update Error Sensitivity Analysis

As indicated in the previous sections, the star tracker update is employed when the pointing error reaches a certain value by using the IMU strapdown integration alone and while the S/C is in orbital rate or in a vertically fixed condition. After the tracker acquires the stars, the compensated and edited V and H readout of the target star is processed by a Kalman filter processor after a star in the catalog is identified. Assume the star tracker update is activated when the pointing error of IMU determination approaches 0.01 degrees and the Kalman filtering update frequency is 5 Hz. After 4 seconds of continuous star acquisition, the pointing error is reduced to 0.004 degrees (14.4 arc sec). Although further star acquisition continues, there is little improvement in the performance results. (The filter converges rather slowly after the first 4 seconds.) This result indicates that if continuous star information is available (for SST tracker, this is highly possible), the star tracker update software package needs to be activated only 4 - 10 seconds after the initial star acquisition.

The star update errors discussed in Section 5.2.2 are all modelled as Gaussian white noise except the boresight axis misalignment, which is treated as bias error. Sensitivity analysis has been conducted on the star update related errors and the result is summarized in Table 5.7.

TABLE 5.7 STAR UPDATE ERROR SOURCE SENSITIVITY

	Percentage of Error Condition
Boresight axis misalignment	72
Basic V & H readout accuracy	17
Star catalog related errors	8
Others	3

As indicated in the table, boresight axis misalignment is the dominant error source for star tracker update. Improvement of performance is anticipated if better information can be achieved from periodic calibration procedure on this bias error.

5.3.5 Orbit Errors Sensitivity Analysis

The orbit errors consist of an error from the onboard orbit generator when it computes the satellite's position and velocity between the GPS receiver six seconds update interval and the GPS receiver time, position, and velocity errors.

Two onboard orbit generators were investigated: a two-body orbit generator and a position and velocity propagator. The two-body orbit generator inputs Keplerian elements and outputs position and velocity. The position and velocity propagator computes an acceleration from the current and previous GPS velocity. Using the current position, velocity and the acceleration, the position and velocity propagator computes the satellite position and velocity. The algorithm for the position and velocity propagator is listed in Figure 5-9.

To determine the accuracy of the above orbit generators, a six-second computer run was done using the Goddard Trajectory Determination System (GTDS) [Goddard, 1] simulation orbit as the true orbit. A 12th order Cowell/Adams predictor-corrector integration scheme with a 9th order polynomial geopotential model was used in GTDS to integrate the Cowell equations of motion. A comparison of the GTDS orbit to the two-body orbit generator and the position and velocity propagator with a stepsize of .05 sec at the end of six seconds given in Table 5.8. As can be seen in Table 5.8, the errors from either method are small. Since the two-body orbit generator requires a large amount of computation,

the position and velocity propagator was selected as the OADS onboard orbit generator.

The results for the orbit position and velocity propagator, as shown in Table 5.8, are based on computing a new position and velocity every 50 ms. Since the orbit position and velocity may not be required every 50 ms (or the onboard computer may not be able to compute position and velocity every 50 ms); a study was done to determine the effect on position and velocity if the stepsize was increased. No difference in the velocity errors occurs for various stepsizes when using the position and velocity propagator. For all cases, the velocity error was 0.4 m/sec. The position errors for various stepsizes are presented in Figure 5-10.

The satellite position and velocity errors from the Global Positioning System listed in Table 5.4 were modelled in the simulation as a Gaussian distribution and added to the position and velocity propagator output. A sensitivity study using the designated maneuver profile was done on the GPS errors by varying the nominal 2σ values from 1 to 10 times the nominal values. Using the nominal orbit errors the effect on the pointing error angle was 0.73 arc sec. Increasing the orbit error 10 times results in a pointing error of 6.83 arc sec. From the above results it was determined that the nominal orbit errors have a relatively insignificant effect on the satellite's pointing error.

5.4 Nominal Performance Results

The proposed OADS attitude determination procedure is to use the IMU strapdown package as primary sensor to continuously provide the reference information from the rate integration. As the reference accuracy gradually

Given:

Position (P_{i-1}), Velocity (V_{i-1}) and time (t_{i-1})

Position (P_i), Velocity (V_i) and time (t_i)

A = Acceleration

Δt = Stepsize

a) Compute Acceleration

$$A_x = (V_x(t_i) - V_x(t_{i-1}))/(t_i - t_{i-1})$$

$$A_y = (V_y(t_i) - V_y(t_{i-1}))/(t_i - t_{i-1})$$

$$A_z = (V_z(t_i) - V_z(t_{i-1}))/(t_i - t_{i-1})$$

b) Compute Velocity

$$V_x = V_x + A_x \Delta t$$

$$V_y = V_y + A_y \Delta t$$

$$V_z = V_z + A_z \Delta t$$

c) Compute Position

$$P_x = P_x + V_x \Delta t$$

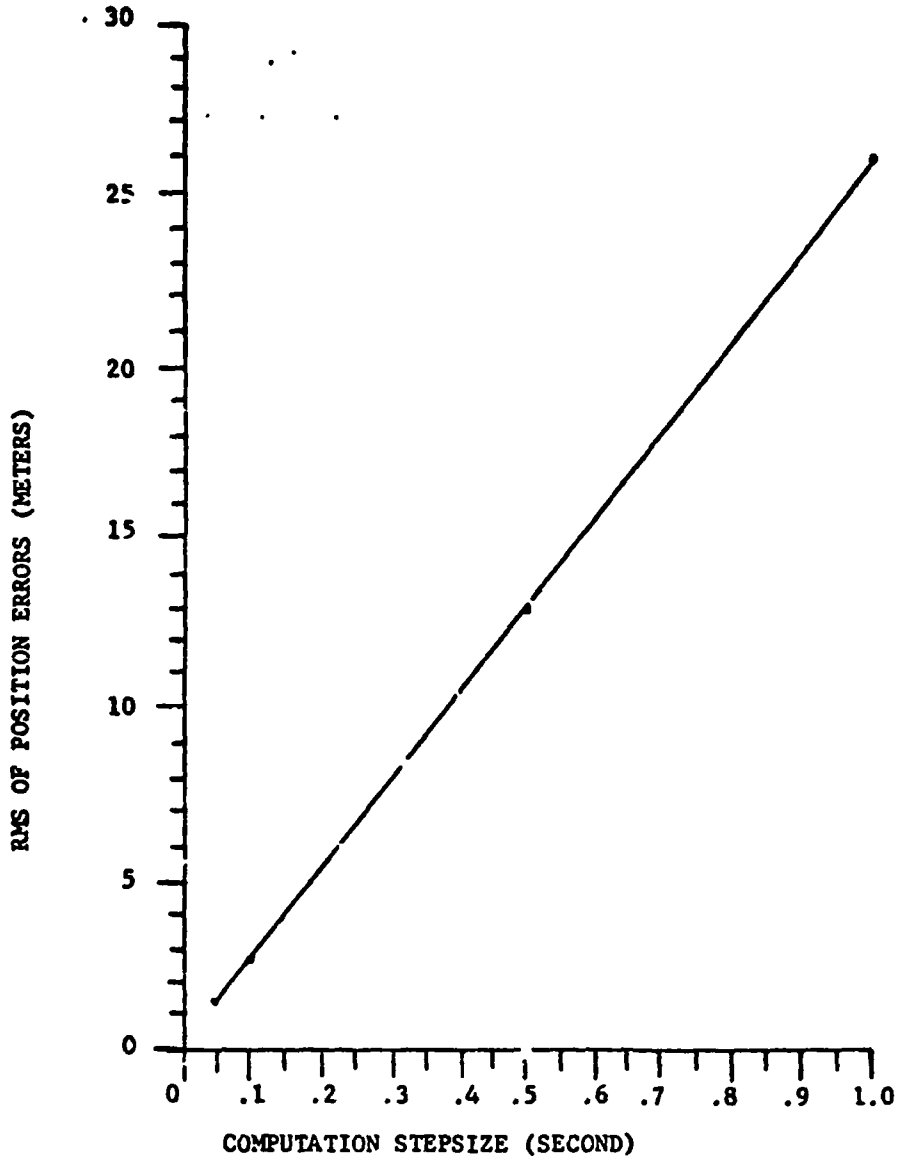
$$P_y = P_y + V_y \Delta t$$

$$P_z = P_z + V_z \Delta t$$

FIGURE 5-9 POSITION AND VELOCITY PROPAGATOR ALGORITHM

	ONBOARD ORBIT GENERATORS	
DIFFERENCE FROM GTDS AT END OF SIX SECONDS	TWO BODY ORBIT GENERATOR	POSITION AND VELOCITY PROPAGATOR
POSITION (ERROR (m)	0.202	1.54
VELOCITY ERROR (m/sec)	0.061	0.40

TABLE 5.8 COMPARISON OF PROPOSED ONBOARD ORBIT GENERATORS TO GTDS



RMS OF POSITION ERRORS AS A FUNCTION OF COMPUTATION STEPSIZE FOR THE POSITION AND VELOCITY PROPAGATOR

FIGURE 5-10

5.4 Nominal Performance Results - (Continued)

deteriorates due to the cumulated gyro errors, the star tracker will be activated to update the reference information after star acquisition and the filtering process. By properly mounting the star trackers such that the tracker data will not be reduced significantly by Earth, sun, or bad geometry the star update performance is relatively independent of the three different MMS missions. However, the orbit rate correction reduced the error in the earth mission, which is relatively small as indicated by the error sensitivity analysis. With the error parameter values described in Table 5.3, the star update gives 0.004° (14.4 arc sec) 2σ pointing accuracy after a continuous sighting of stars. Thus, the performance analysis in the following paragraphs is concentrating on the primary attitude determination sensor, the IMU. The performance study determined what was the resultant error after a certain type of maneuver under different mission environments and the time interval between star tracker updates.

Performance results are presented in the following sections for the Earth, stellar and solar missions. Due to the star tracker update accuracy, an initial attitude error of 0.004 degree along the pitch axis is imposed on all missions being studied. All maneuvers were along the pitch axis using the maneuver profiles discussed in Section 5.4.1. The nominal sensor errors discussed in Section 5.3 were used.

5.4.1 Nominal Earth Mission Performance Results

The earth missions of interest are 400 km, 705.3 km, and 2000 km orbits at inclinations of 56, 98.2, and 56 degrees, respectively. Performance results for nadir tracking, 5°/minute and 2°/sec maneuver profiles defined in Section 2.1 are presented in Tables 5.9 - 5.13. Tables 5.10 - 5.13 present the performance results for

each maneuver profile at the different altitudes. Table 5.9 presents the three maneuvers for LANDSAT-D (705.3 km altitude, 98.2 degrees inclination). As shown in Table 5.9, the pointing error rate decreases as the maneuver rate decreases. This occurs because the effect of the gyro scale factor errors and gyro-to-gyro misalignment errors are directly related to the body rates. From the sensitivity study, Section 5.3.3, the above gyro errors were the most significant.

The pointing error rate provides a way to determine how often the attitude state needs to be updated by the star tracker. Assuming the nominal mission requirement is to maintain a pointing error of .01 degree and a star tracker accuracy of .004 degrees, the pointing error change between star tracker updates is 0.006 degree. The time interval between star tracker updates for each study case was computed and is presented in Table 5.14. Table 5.14 shows that as the maneuver rate increases for a specific orbit, the more frequent attitude updates become. This occurs because the gyro errors are directly related to the satellite body rates as discussed above. As the satellites altitude increases, the orbital rate decreases resulting in a lower pointing error rate and longer time intervals between star tracker updates. Thus, the lower the orbit and the higher the maneuver rate, the more frequent star tracker updates will be required for each Earth type mission.

5.4.2 Nominal Stellar Mission Performance Results

The stellar missions investigated were 400 and 2000 km at an inclination of 28.5 degrees. The star lock and dwell maneuver was simulated by commanding a negative pitch rate. The resulting pointing error, pointing error rates, and time between star tracker updates are presented in Table 5.15.

TABLE 5.9 LANDSAT-D ORBIT PERFORMANCE RESULTS FOR VARIOUS MANEUVER RATES

LANDSAT-D ORBIT	ALTITUDE = 705.3 km INCLINATION = 98.2 Deg		
	NADIR TRACKING 60 SECONDS	5°/MINUTE 20° MANEUVER 245 SECONDS	2°/SECOND 20° MANEUVER 13 SECONDS
FINAL PITCH ERROR (ARC-SEC)	15.11	24.59	19.21
FINAL POINTING ERROR (ARC-SEC)	15.14	25.87	20.33
LATITUDE ERROR (ARC-SEC)	1.88	3.129	2.10
LONGITUDE ERROR (ARC-SEC)	0.14	0.52	0.65
ALTITUDE ERROR (M)	24.0	25.0	25.0
POINTING ERROR RATE (ARC-SEC/SEC)	0.012	0.047	0.46

TABLE 5.10 NADIR TRACKING PERFORMANCE RESULTS FOR EARTH MISSIONS

NADIR TRACKING RESULTS FOR THE EARTH MISSION			
	400 km i = 56° 60 SECONDS	705.3 km i = 98.2° 60 SECONDS	2000 km i = 56° 60 SECONDS
FINAL PITCH ERROR (ARC-SEC)	15.43	15.11	14.84
FINAL POINTING ERROR (ARC-SEC)	15.46	15.14	14.87
LATITUDE ERROR (ARC-SEC)	1.40	1.88	3.60
LONGITUDE ERROR (ARC-SEC)	1.16	0.14	3.57
ALTITUDE ERROR (M)	25.0	24.0	22.0
POINTING ERROR RATE (ARC-SEC/SEC)	0.018	0.012	0.0078

TABLE 5.11 PERFORMANCE RESULTS FOR A 20 DEGREE MANEUVER AT 5°/MINUTE FOR THE 400 km AND 705.3 km EARTH MISSIONS

5°/MINUTE - 20° MANEUVER - 245 SECONDS		
	400 km i = 56°	705.3 km i = 98.2°
FINAL PITCH ERROR (ARC-SEC)	27.63	24.59
FINAL POINTING ERROR (ARC-SEC)	27.67	25.87
LATITUDE ERROR (ARC-SEC)	1.3	3.29
LONGITUDE ERROR (ARC-SEC)	1.56	0.52
ALTITUDE ERROR (M)	25.0	25.0
POINTING ERROR RATE (ARC-SEC/SEC)	.054	.047

TABLE 5.12 PERFORMANCE RESULTS FOR A 20 DEGREE MANEUVER AT 2°/SECOND FOR THE 400 km AND 705.3 km EARTH MISSIONS

2°/SECOND - 20° MANEUVER - 13 SECONDS		
	400 km i = 56°	705.3 km i - 98.2°
FINAL PITCH ERROR (ARC-SEC)	24.33	19.21
FINAL POINTING ERROR (ARC-SEC)	24.35	20.33
LATITUDE ERROR (ARC-SEC)	1.12	2.10
LONGITUDE ERROR (ARC-SEC)	0.79	0.65
ALTITUDE ERROR (M)	26.0	25.0
POINTING ERROR RATE (ARC-SEC/SEC)	.77	.46

TABLE 5.13 PERFORMANCE RESULTS FOR A 5° MANEUVER AT 5°/MINUTE AND 2°/SECOND FOR THE 2000 km EARTH MISSION

5° MANEUVER, ALTITUDE=2000 km i = 56°		
	5°/MINUTE (64 SECONDS)	2°/SECOND (6.5 SECONDS)
FINAL PITCH ERROR (ARC-SEC)	18.06	16.77
FINAL POINTING ERROR (ARC-SEC)	18.09	16.78
LATITUDE ERROR (ARC-SEC)	4.21	3.96
LONGITUDE ERROR (ARC-SEC)	3.96	3.61
ALTITUDE ERROR (M)	25.0	26.0
POINTING ERROR RATE (ARC-SEC/SEC)	0.058	0.37

TABLE 5.14 TIME BETWEEN STAR TRACKER UPDATES FOR THE EARTH MISSIONS

	TIME BETWEEN STAR TRACKER UPDATES (MINUTES)		
ORBIT ALTITUDE (km)	NADIR TRACKING	5°/MINUTE MANEUVER	2°/SECOND MANEUVER
400.0	20.00	6.67	0.47
705.3	30.00	7.66	0.78
2000.0	46.15	*6.21	0.97

*This data is based on a 64 sec 5°/min., 5° maneuver instead of 20° maneuver at 400 km and 705.3 km which has larger scale factor error effect

5.4.3 Nominal Solar Mission Performance Results

The solar missions investigated were 400 km and 2000 km at an inclination of 28.5 degrees. The sun lock and small scan maneuvers were investigated. The sun look is the same as the star lock and the results are presented in Table 5.15. The small scan maneuver was simulated by scanning .5 degrees (the diameter of the sun) at 5°/minute. Associated pointing error, pointing error rate, and time between star updates is presented in Table 5-16.

5.4.4 Summary

Based upon the sensitivity and performance results, the following observations can be made.

(a) The proposed On-board Attitude Determination System consisting of NASA Standard IMU (DRIRU-II), NASA Standard star tracker (SST), and GPS receiver, with the described attitude determination procedure and algorithms, is capable of providing precision on-board attitude information for all these MMS missions (Earth, solar and stellar).

(b) The most significant IMU errors due to maneuvers are scale factors and gyro misalignment. In non-slewing environments, bias drift is more important.

(c) The most significant star tracker error is boresight axis misalignment. Significant improvement of performance is anticipated if better knowledge of this error is available.

(d) If the Global Pointing System is able to maintain its specified accuracy, the impact to the OADS accuracy is insignificant.

(e) As expected, the Earth mission is the most demanding one among the MMS missions for OADS system. For the same maneuvering sequence, the 400 km altitude Earth mission requires the most frequent updates by star trackers.

(f) As the spacecraft altitude increases, the time required for star update to maintain a certain accuracy level also increases. The OADS performance is independent to the change of orbit plane inclination.

(g) As the maneuver rate increases, the require time interval between star update decreases due to large IMU error buildup during maneuvers.

TABLE 5.15 STELLAR MISSION PERFORMANCE RESULTS

ATTITUDE ERROR (ARC-SEC)	400 km, e = 0, i = 28.5			2000 km, e = 0, i = 28.5		
	STAR LOCK AND DWELL (60 SECONDS)	15° MANEUVER 5°/MIN (180 SECONDS)	15° MANEUVER 2°/SEC (10.50 SECONDS)	STAR LOCK AND DWELL (60 SECONDS)	15° MANEUVER 5°/MIN (180 SECONDS)	15° MANEUVER 2°/SEC (10.50 SECONDS)
FINAL POINTING ERROR	7.79	17.26	12.62	7.79	16.17	12.59
Δ POINTING ERROR	0.61	10.08	5.44	0.61	8.99	5.41
POINTING ERROR RATE (ARC-SEC/SEC)	0.01	0.056	0.52	0.01	0.05	0.52
TIME BETWEEN STAR TRACKER UPDATES (MINUTES)	35.50	6.43	0.69	35.50	7.35	0.69

TABLE 5.16 SOLAR MISSION PERFORMANCE RESULTS

ATTITUD ERRORS (ARC-SEC)	400 km, e = 0 i = 28.5°	2000 km, e = 0 i = 28.5°
	5°/minute .5° maneuver (9 seconds)	5°/minute .5° maneuver (9 seconds)
Final Pointing Error	7.56	7.50
Δ Pointing Error	0.387	0.32
Pointing Error Rate (Arc-Sec/Sec)	0.042	0.036
Time Between Star Tracker Updates (Minutes)	8.60	10.03

6.0 MICROPROCESSOR SOFTWARE ANALYSIS

This section describes our software analysis for a microprocessor based spaceborne attitude control system. Figure 6-1 illustrates the OADS elements which we have examined. A timing analysis was performed to determine what throughput could be expected and whether this throughput was consistent with OADS objectives. To perform the detailed timing analysis, the actual code was written for a baseline configuration containing Intal 8080 microprocessors and Advanced Micro Devices AM 9511 arithmetic processor units. There were two reasons for chosing this configuration. First, these devices are representative of current large-scale integration (LSI) fabrication technology. Secondly, because the devices are N-Channel silicon gate metal oxide semiconductor (NMOS) technology, the 8080 and AM 9511 are relatively power conservative, moderate speed devices. This latter characteristic helps to provide conservative timing estimates. Section 7 discusses the hardware aspects of our baseline configuration as well as the use of other types of fabrication technology in spaceborne processing applications such as OADS.

The following paragraphs describe the results of our software analysis of an OADS microprocessor system. Although star tracker processing is only performed a few times per orbit, very fast processing is required when star tracker measurements are being taken. We, therefore, performed a special IRU and star tracker integration analysis to insure that performance objectives could be maintained. Total system integration, involving all OADS processing elements, is described at the end of this section.

6.1 IRU Analysis

Figure 6-2 shows the five steps required for IRU processing. Previous analysis showed that the gyro inputs were to be sampled every 50 milliseconds (ms).

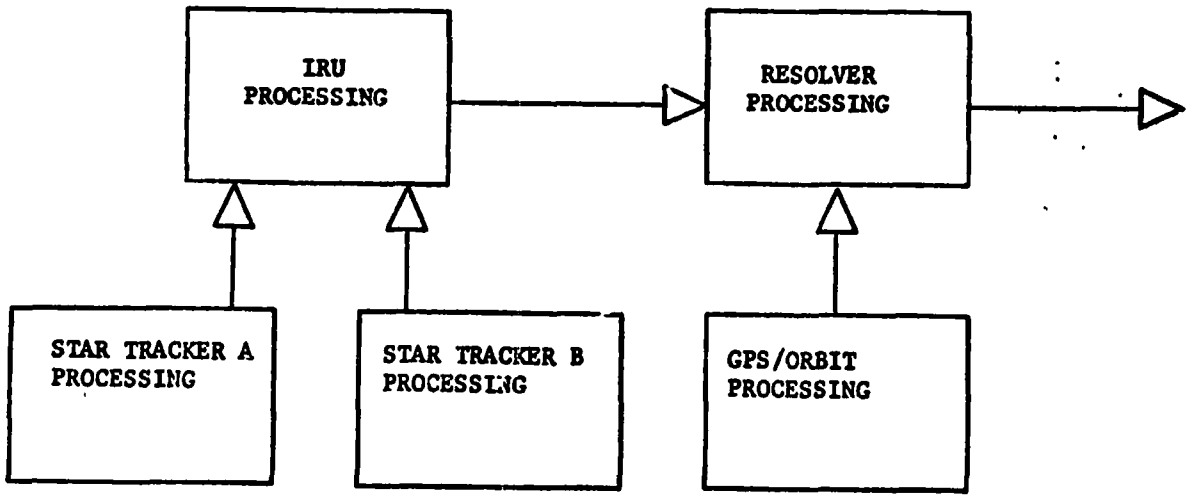
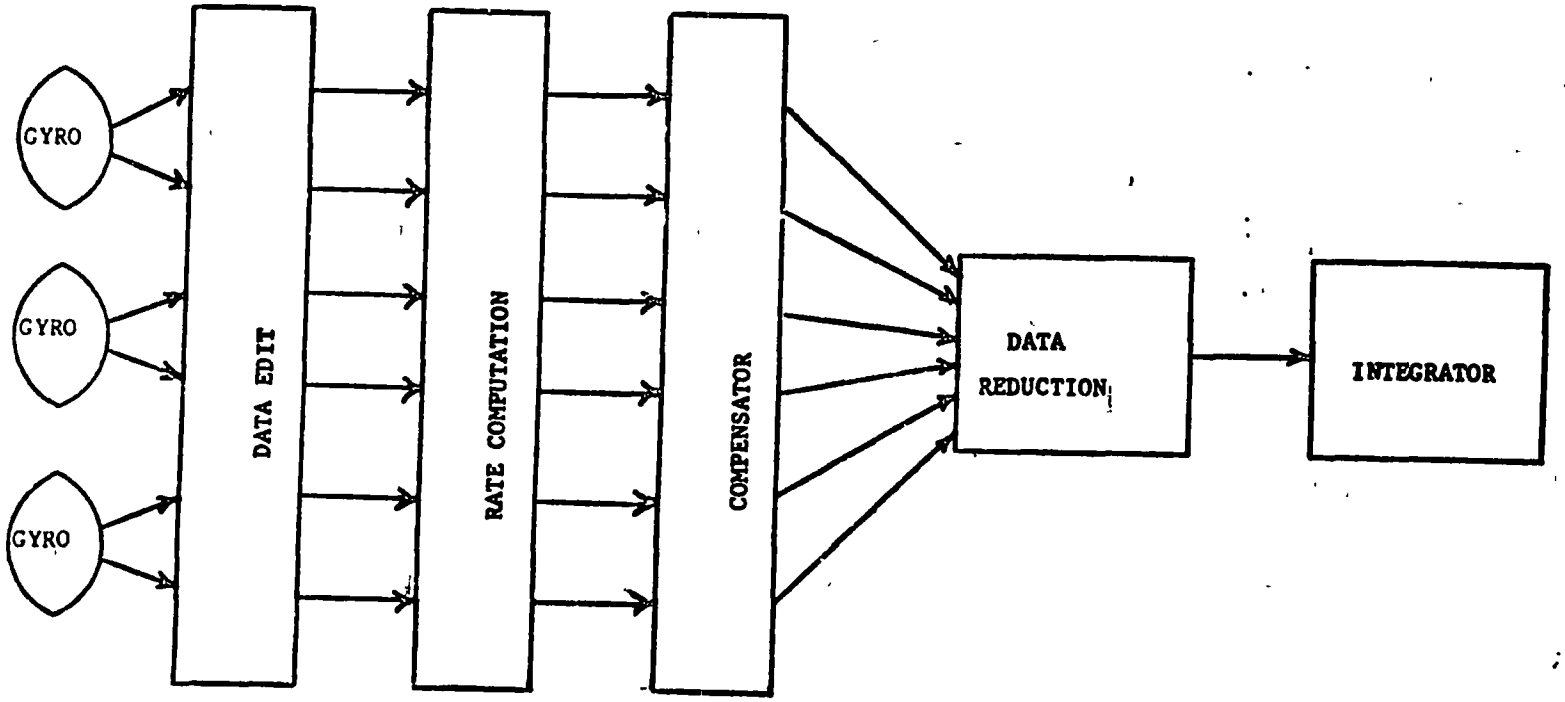


FIGURE 6-1

PROCESSING ELEMENTS IN ON-BOARD ATTITUDE DETERMINATION SYSTEM



IRU PROCESSING STEPS

FIGURE 6-2

This implies a 50 ms cycle time for IRU processing, however, the quaternion must be produced as early as possible to minimize the impact on star tracker processing.

The algorithms associated with IRU processing are: data edit, rate computation, compensation, data reduction, and integration. Although there are many methods of editing gyro data and isolating bad readings, the algorithm used in our timing analysis simply consisted of reading six gyro input channels and checking that the readings be between predefined maximum and minimum values. Rate computation consist of differencing consecutive gyro readings and multiplying this value by the rotation rate. Since the gyros are sampled at fixed time intervals, the rate conversion factor is a constant. The equation therefore becomes:

$$W = (P_i - P_{i-1})\, K_w$$

For six values of W, compensation is necessary to correct channel rate measurements for static and dynamic gyro errors (see Section 5). Two equations must be evaluated:

$$W_{xc} = A' \left(W_x - \rho_z W_y + \rho_y W_z - B_y - B' W_x W_z + C'_\alpha W_y\right)$$

and

$$W_{yc} = E' \left(W_y + (\rho_z + U_{ez}) W_x - \rho_x W_z - B_x - B' W_y W_z + C'_\alpha W_y\right)$$

for three values of W_{xc} and three values of W_{yc}. The data reduction algorithm is a least squares algorithm for computing body rates from the corrected gyro measurements. The equation is of the form:

$$\left[W_B\right] = \left(\left[H\right]^T \left[W_M\right]^{-1} \left[H\right]\right)^{-1} \left[H\right]^T \left[W_M\right]^{-1} \left[W_C\right]$$

Once data reduction is performed, it is then possible to evaluate the quaternion differential equation using a second order Runga-Kutta integration.

The equations involved with this step are of the form:

$$\left[\delta_{i-1}\right] = \tfrac{1}{2}\left[\Omega_{i-1}\right]\left[q_{i-1}\right]$$

$$\left[Y_{i-1}\right] = \left[q_{i-1}\right] + \Delta t \left[\delta_{i-1}\right]$$

$$\left[\delta_i\right] = \tfrac{1}{2}\left[\Omega_i\right]\left[Y_i\right]$$

$$\left[q_i\right] = \left[q_{i-1}\right] + \frac{\Delta t}{2}\left(\left[\delta_i\right] + \left[\delta_{i-1}\right]\right)$$

where

$$\left[\Omega\right] = f\left(\left[W_B\right]\right)$$

For a more detailed explanation of the algorithms associated with IRU processing, the reader should consult Section 5.

As can be easily seen from the algorithms just summarized, IRU processing is a computational oriented problem. Microprocessors have for some years been used as circuit simplification devices as well as in many small process control applications. Only recently have the computational capabilities of these these devices been examined for use in applications such as OADS. The Intel 8080 microprocessor is an 8-bit general-purpose processing unit. Its relatively primitive instruction set (as compared to minicomputers) makes the 8080 undesirable for performing the arithmetic computational requirements needed for on-board attitude determination. The Am9511, on the other hand, is tailored to performing arithmetic computations (refer to Appendix I) but its data management capabilities are extremely limited. IRU processing was, therefore, initially analyzed for a system containing both an 8080 and an Am9511.

In our analysis, worst case timing was always used. For example, a floating point addition in the Am9511 requires between 28 and 175 microseconds depending on normalization. A high degree of confidence can be placed in the results we obtained since our analysis used not only conservative timing

estimates for individual instructions but also because all of the algorithms were coded at the instruction level without any attempt to optimize the code. The following table shows the results of our timing analysis.

TABLE 6.1 ESTIMATED IRU THROUGHPUT USING SINGLE APU

ALGORITHM	ALGORITHM PROCESSING TIME (ms)	ACCUMULATIVE PROCESSING TIME (ms)
Data Edit	.132	.132
Rate Computation	2.157	2.289
Compensation	13.242	15.531
Data Reduction	21.534	37.065
Integration	12.924	49.989

We determined that all nominal IRU processing can be accomplished by one 8080 microprocessor and one Am9511 arithmetic processor in 49.989 ms. Although it is interesting to note that this NMOS configuration is capable of meeting IRU throughput requirements (50 ms), the analysis shows that there is virtually no room for expansion.

It was initially felt that OADS processing could use a multiple microprocessor configuration to improve throughput by parallel processing. Assuming that parallelisms exist in the IRU algorithms, dual microcomputers, each consisting of an 8080 and Am9511, could be implemented to improve IRU processing throughput. There is, however, an alternate configuration which proved to be much more favorable. Figure 6-3a shows the microprocessor unit (MPU) and arithmetic processing unit (APU) activities during a segment of IRU processing. The gaps in this timeline figure represent periods during which a device (MPU or APU) cannot perform useful work because it is waiting on data being used by the other device. To illustrate how much this configuration

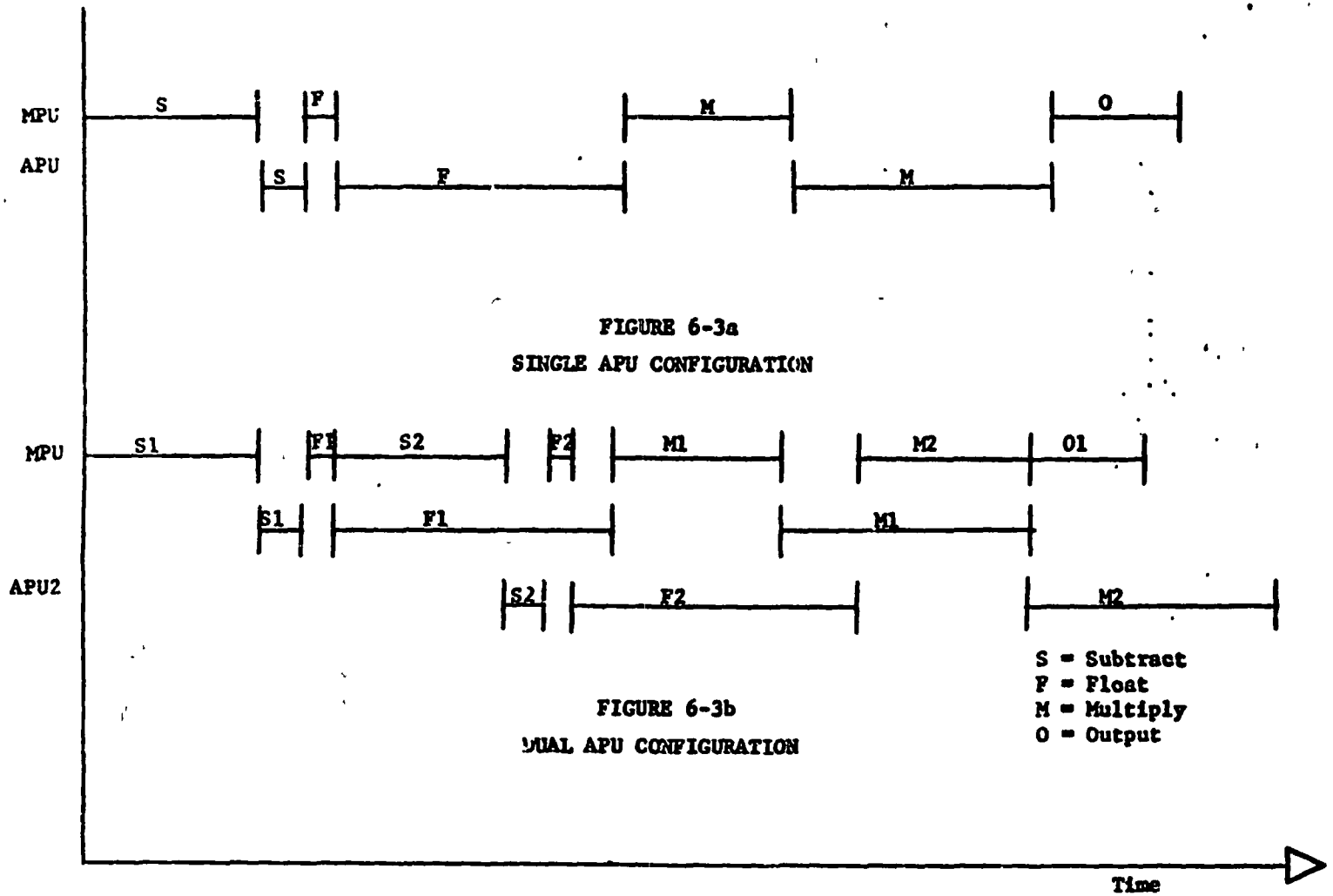


FIGURE 6-3a
SINGLE APU CONFIGURATION

FIGURE 6-3b
DUAL APU CONFIGURATION

FIGURE 6-3
REPRESENTATIVE EXECUTION IN SINGLE AND DUAL APU CONFIGURATIONS

characteristic was degrading throughput, a processor efficiency term was calculated. This term, E, was defined as the amount of time a processor was performing useful work divided by the total algorithm processing time. Processor efficiency in the single MPU, single APU configuration is shown in the following table.

TABLE 6.2 PROCESSOR EFFICIENCY IN SINGLE APU SYSTEM

ALGORITHM	ALGORITHM EFFICIENCY EMPU	EAPU	ACCUMULATIVE EFFICIENCY EMPU	EAPU
Data Edit	1.0	0.0	1.0	0.0
Rate Computation	.46	.54	.49	.51
Compensation	.33	.67	.36	.64
Data Reduction	.34	.66	.35	.65
Integration	.38	.62	.36	.64

As can be seen in the table, during IRU processing the 8080 is performing useful work only 36% of the time while the Am9511 is busy 64%. This fact, combined with the timeline characteristics (see Figure 6-3a), suggest that throughput could be increased significantly by the addition of a second APU. Such a multiple APU configuration can take advantage of parallelisms in the IRU algorithms but would be much less complex, in both hardware and software terms, than a dual microcomputer configuration. Figure 6-3b illustrates the overlapped processing that can be obtained in the single MPU, dual APU configuration. The following table shows the results of the software timing analysis of the dual APU system.

TABLE 6.3 ESTIMATED IRU THROUGHPUT IN DUAL APU SYSTEM

ALGORITHM	ALGORITHM PROCESSING TIME (ms)	ACCUMULATIVE PROCESSING TIME (ms)
Data Edit	.132	.132
Rate Computation	1.245	1.377
Compensation	7.198	8.575
Data Reduction	13.521	22.096
Integration	7.324	29.42

In our timing analyses, numerous ways were found to code the software and obtain different degrees of overlapped execution. No attempt was made to optimize parallel execution and the results of our analysis for the dual APU configuration are even more conservative than for the single APU configuration. Two performance items are important in the dual APU configuration. First, MPU and APU efficiency are much closer. Accumulative Empu is .61 and accumulative Fapu is .56. This means that although performance may be further improved by adding another microprocessor or arithmetic processor, the improvement would not be dramatic. Secondly, the 29.42 millisecond accumulative IRU processing time represents 60% machine utilization. Although many flight computer designs are based on timing estimates showing 50% machine utilization, we do not feel this extra safety margin would be necessary in an IRU microcomputer system. Most flight computers are intended to be general purpose systems whose processing requirements are dependent on total spacecraft engineering and science needs. These needs may change during the mission and the general purpose flight computer must accommodate these changes. An IRU microcomputer system is dependent only on IRU processing requirements. The microcomputer handles a very limited and (hopefully) well-defined requirement.

Furthermore, changes in these requirements during a mission are limited by the fact that the IRU microcomputer will have limited interfaces to other spacecraft subsystems. For these reasons and because very conservative timing estimates were used, 60% microcomputer utilization appears to be very reasonable.

6.2 Star Tracker Software Analysis

Figure 6-4 shows the steps involved in processing data from a single star tracker. There are basically ten algorithm sets involved. The first of these algorithms is to simply read the star tracker and convert V & H grid values to engineering units. This is a straightforward conversion process:

$$V_E = V * K_V$$

$$H_E = H * K_H$$

Once V and H are computed in engineering units, it is necessary to synchronize the readout by "moving" the V and H values forward or backward in time. This is necessary if a quaternion, as supplied by the IRU, is to be used in star tracker processing. To synchronize the V and H reading, the average gyro rate must be computed,

$$\left[W_{AVG}\right]_B = \tfrac{1}{2} \left(\left[W_i\right]_B + \left[W_{i-1}\right]_B \right)$$

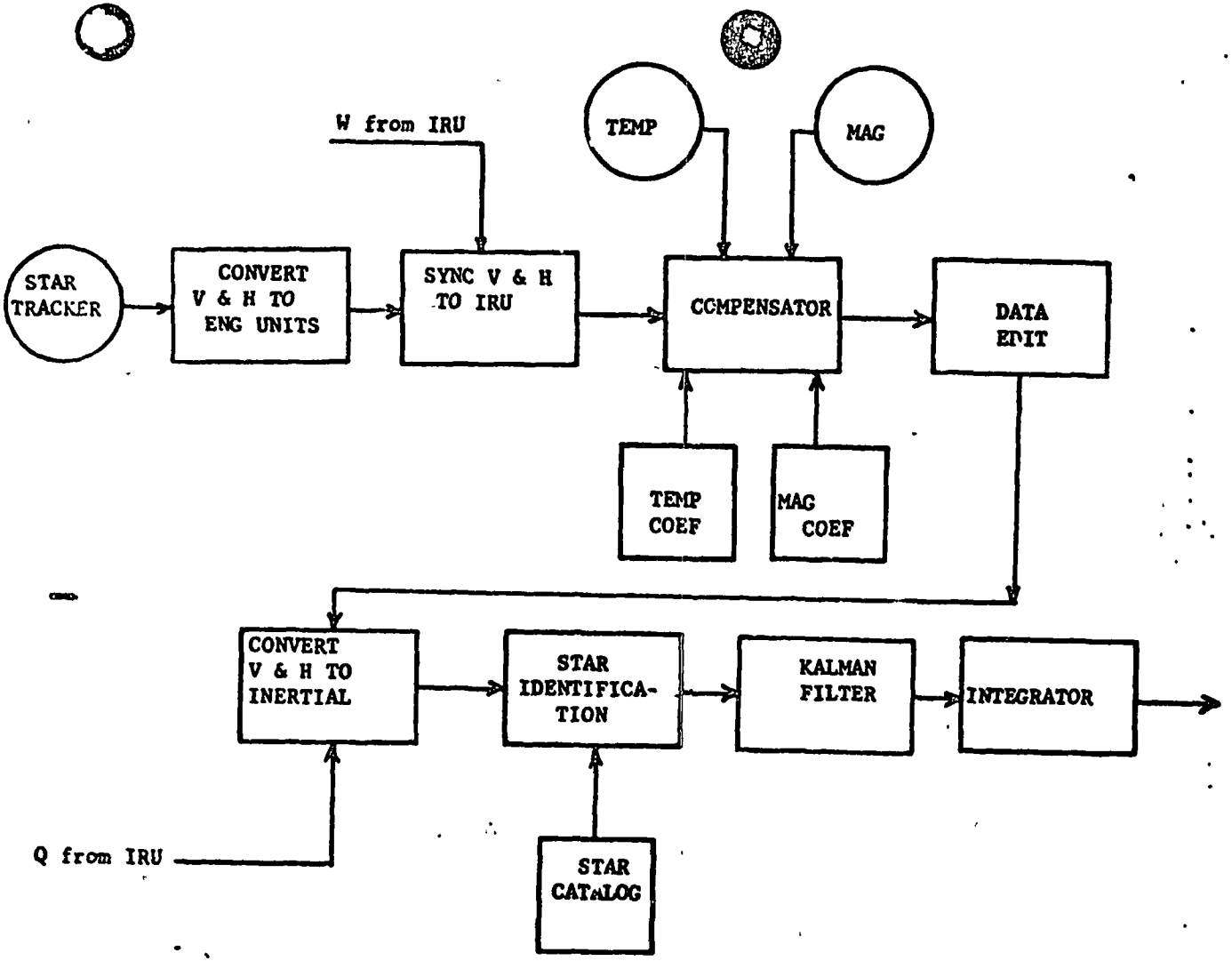
and converted into the star tracker reference frame.

$$\left[W_{AVG}\right]_S = \left[{}^S C_B\right] \left[W_{AVG}\right]_B$$

The synchronized V and H values may then be calculated.

$$V_S = V_E + W_{ZS}\, \Delta t$$

$$H_S = H_E + W_{XS}\, \Delta t$$



STAR TRACKER PROCESSING

FIGURE 6-4

Since star tracker readouts are sensitive to both temperature and magnetic field, compensation is required. To accomplish this temperature and magnetometer reading must be made and using these values interpolation in lookup tables is performed to determine twenty correction coefficients. Compensated V and H values may then be obtained using the following equations:

$$V_C = C_1 + C_2V_S + C_3V_S^2 + C_4V_S^3 + C_5V_S^2H_S + C_6V_SH_S^2 + C_7H_S^3 + C_8H_S^2 + C_9H_S + C_{10}H_SV_S$$

$$H_C = d_1 + d_2V_S + d_3V_S^2 + d_4V_S^3 + d_5V_S^2H_S + d_6V_SH_S^2 + d_7H_S^3 + d_8H_S^2 + d_9H_S + d_{10}H_SV_S$$

where C_i and d_i are compensation coefficients computed by linear interpolation in lookup tables.

After compensation, the data should be edited as was done in IRU processing. This simply involves insuring that Vc and Hc lie between minimum and maximum limits. To perform star identification, it is necessary to convert Vc and Hc values to inertial right ascension and declination. This is accomplished as follows:

$$\begin{bmatrix} S_S \end{bmatrix} = \begin{bmatrix} -\sin H_C \cos V_C \\ \cos H_C \cos V_C \\ -\sin V_C \end{bmatrix}$$

$$\left[{}^{I}C_{B}\right] = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix}$$

where qi is an element of quaternion to which V and H were previously synchronized. Then,

$$\left[S_I\right] = \left[{}^{I}C_{B}\right]\left[{}^{B}C_{S}\right]\left[S_S\right]$$

and

$$\alpha_{VH} = \tan^{-1}(S_2/S_1)$$

$$\delta_{VH} = \sin^{-1}(S_3)$$

Knowing the right ascension and declination associated with V and H readings, it is possible to search a star catalog and determine what star is in the tracker's field of view. This is accomplished by determining which star has a right ascension and declination closest to that predicted by the V and H readings. Therefore, the equation:

$$D_E = (\alpha_S - \alpha_{VH})^2 + (\delta_S - \delta_{VH})^2$$

must be evaluated for every candidate star in the catalog. After the proper star has been identified, a Kalman filter process is used to correct the quaternion for the difference between the known attitude frame and the ideal attitude frame. The equations associated with the Kalman filter process are shown below:

$$\left[G\right] = \begin{bmatrix} -\cos H_C \cos V_C & \sin H_C \sin V_C \\ -\sin H_C \sin V_C & -\cos H_C \sin V_C \\ 0 & -\cos V_C \end{bmatrix}$$

$$\left[{}^{I}C_{S}\right] = \left[{}^{I}C_{B}\right]\left[{}^{B}C_{S}\right]$$

$$\left[A\right] = \left(\left[{}^{I}C_{S}\right]\left[G\right]\left[E_{1}\right]\left[G\right]^{T}\left[{}_{S}G_{I}\right]\right) + \left[I\right]$$

$$\left[E_{2}\right] = \begin{bmatrix} -\sin\alpha_S & \cos\delta_S & -\cos\alpha_S & \sin\delta_S \\ \cos\alpha_S & \cos\delta_S & -\sin\alpha_S & \sin\delta_S \\ 0 & & -\cos V_C & \end{bmatrix}$$

$$\left[D\right] = \begin{bmatrix} \sigma_{11}^{2} & 0 \\ 0 & \sigma_{22}^{2} \end{bmatrix}$$

$$\left[R\right] = \left[E_{2}\right]\left[D\right]\left[E_{2}\right]^{T}$$

$$\left[H_{I}\right] = \begin{bmatrix} 0 & -S_{3I} & S_{2I} \\ S_{3I} & 0 & -S_{1I} \\ -S_{2I} & S_{1I} & 0 \end{bmatrix}$$

$$\left[P\right] = \left[M\right] - \left[M\right]\left[H_{I}\right]\left[\left(\left[H_{I}\right]\left[M\right]\left[H_{I}\right]^{T} + \left[A\right] + \left[R\right]\right)\right]^{-1}\left[H_{I}\right]\left[M\right]^{T}$$

$$\left[K\right] = \left[P\right]\left[H_{I}\right]^{T}\left[\left(\left[A\right] + \left[R\right] + \left[I\right]\right)\right]^{-1}$$

$$\left[Y_{M}\right] = \left[T\right] - \left[S_{I}\right]$$

where $$\left[T\right] = \begin{bmatrix} \cos\delta_S & \cos\alpha_S \\ \cos\delta_S & \sin\alpha_S \\ & \sin\delta_S \end{bmatrix}$$

$$\left[V^1\right] = \left[V\right] + \left[K\right]\left(\left[Y_M\right] - \left[H_I\right]\left[V\right]\right)$$

$$\left[{}^{I}C_B\right] = \left[DMA\right]\left[{}^{I}C_B\right]$$

where

$$\left[DMA\right] = \begin{bmatrix} 1.0 & v_3' & -v_2' \\ -v_3' & 1.0 & v_1' \\ v_2' & -v_1' & 1.0 \end{bmatrix}$$

$$q_4 = \tfrac{1}{2}\left(1.0 + C_{11}' + C_{22}' + C_{33}'\right)^{\frac{1}{2}}$$

$$q_1 = (C_{23}' - C_{32}')/4q_4$$

$$q_2 = (C_{31}' - C_{13}')/4q_4$$

$$q_3 = (C_{12}' - C_{21}')/4q_4$$

Once the updated quaternion has been produced, it will be necessary to integrate it forward in time to catch up with the IRU processing that has been going on asynchronously. The same Runge-Kutta integrator that was used in IRU processing is used again in this phase. In this case, however, an average gyro rate evaluated over the last N IRU processing cycles will be used. For a more detailed discussion of the star tracker algorithms, consult Section 5.

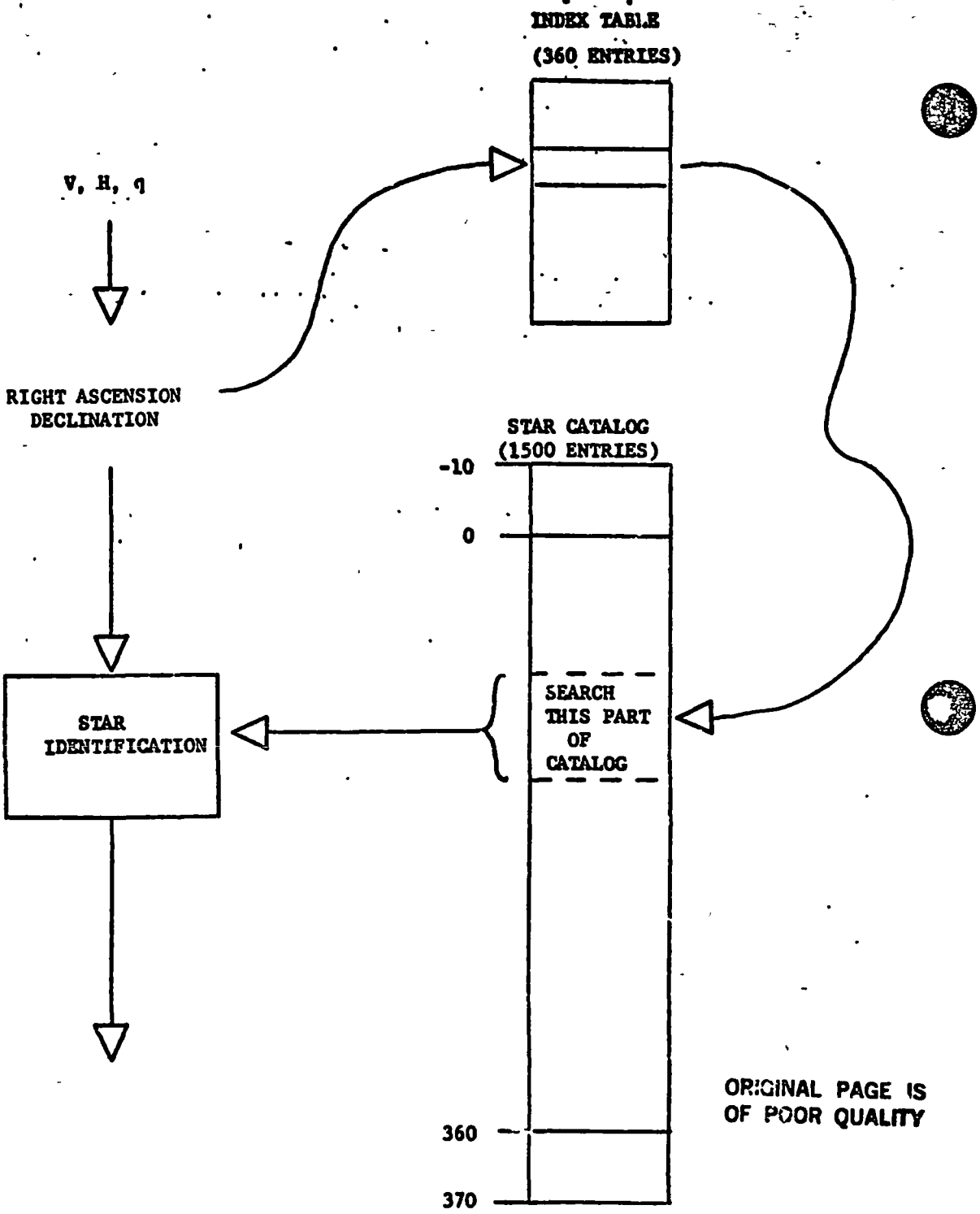
We performed the software timing analysis by writing the actual code needed in star tracker processing. Baseline configurations containing a single 8080 microprocessor and one or two Am9511 arithmetic processors were evaluated just as in the IRU timing analysis. Worst case instruction execution times were used and no attempt was made to optimize overlapped execution in the dual APU configuration. The individual processing times for the star tracker algorithms are as follows:

TABLE 6.4 ESTIMATED STAR TRACKER THROUGHPUT

ALGORITHM	SINGLE APU PROCESSING TIME (ms)	DUAL APU PROCESSING TIME (ms)
Units Conversion	20.612	20.3605
Synchronization	4.233	2.551
Compensation	27.970	19.987
Data Edit	.094	.094
Inertial Conversion	29.92	16.83
Star Identification	142.776	74.526
Kalman Filter	133.0745	83.179
Integration	15.676	9.159

The following characteristics were assumed in generating these timing estimates. First, there is a 20 millisecond latency between V and H readouts. Secondly, the lookup tables used for compensation contained 100 entries each. Finally, the star identification algorithm is based on a search of 100 star catalog entries. Although there are approximately 1500 stars of magnitude five, it will be shown in a later paragraph that searching the entire catalog is not necessary.

The primary difference between IRU processing and star tracker processing is that star tracker processing is a two-phase problem. The first phase involves determining what star is in the tracker's field of view. Our analysis shows that it requires over 1100 milliseconds to accomplish a linear search of a star catalog containing 1500 stars (using a dual APU configuration). This time might be reduced by using a binary search, however, the 8080 microprocessor instruction set does not lend itself to this type of algorithm for large tables. Another approach we considered is to use an indirect indexing table. If the star catalog is sorted by right ascension, an indirect indexing table containing 360 entries is constructed. Each entry in the index table corresponds to one degree of right ascension and points to the star catalog where stars of corresponding right ascension are stored (see Figure 6-5). For example, assume that a given V and H reading and a quaternion produce a predicted star right ascension of 263.875°. This right ascension value is truncated and used as an index table position. The 263 entry in the index table contains an address in the star catalog where stars whose right ascension is 263.xxx degrees are stored. The star catalog entries around this point are then searched (using the star identification algorithm previously described) to determine which star best fits the predicted star right ascension and declination. It is estimated that this may require searching 100 star catalog



STAR CATALOG SEARCH USING INDIRECT INDEXING

FIGURE 6-5

entries and use 114 milliseconds (in the dual APU configuration).

Once a star in the tracker's field of view has been identified, searching the star catalog is not required again until that star leaves the field of view. Phase two of star tracker processing involves using V, H and IRU quaternion, and star catalog data to produce an updated quaternion. This updated quaternion is then integrated forward in time and used in succeeding IRU processing. Our analysis shows that for a single star tracker, generation of a corrected quaternion will require 152 milliseconds in the dual APU configuration. If two trackers are used, each having their own microcomputer system, star identification processing can be overlapped but most of phase two processing must be executed sequentially. This results in 235 millisecond processing time when two star trackers are active. The following table summarizes this information.

TABLE 6.5 MULTIPLE PHASE STAR TRACKER THROUGHPUT

	PROCESSING TIME WITH SINGLE APU (ms)	PROCESSING TIME WITH DUAL APU (ms)
Phase 1	265.6	134.3
Phase 2 - Single Star Tracker	231.6	152.2
Phase 2 - Dual Star Tracker	366.0	235.8

6.3 IRU and Star Tracker Integration

From the beginning of this study it was felt that the combined IRU and Star Tracker processing would place the greatest demands upon an onboard microcomputer system. IRU processing time is constrained by the fact that gyro readings are to be made every 50 milliseconds. Star tracker processing

must be synchronized to IRU processing and must be performed in a sufficiently short time to correct the quaternion to the required accuracy. Because of the high demands on the IRU and star tracker microcomputers, it seemed appropriate to integrate these subsystems before investigating the remaining elements of the OADS system.

Since IRU data is required for star tracker processing and vice versa, a shared memory unit is anticipated for the two microcomputer systems. It would have been possible to directly connect the microcomputers using input and output ports; however, this technique would require higher software overhead. Because of the independent processing of the IRU and star tracker microcomputer systems, a synchronization mechanism is required to insure reliable results. Two levels of synchronization are anticipated. First, a hardware semaphore is needed to prevent inconsistencies in shared data memory. For example, the IRU microcomputer must not be allowed to modify rate and quaternion data the star tracker microcomputer is reading. The hardware semaphore would prevent this by permitting only one microcomputer system to access shared memory at a time. To avoid long access delays, individual microcomputers could move shared data into local memory and then operate upon that data while it is in local memory. The hardware semaphore need not be complex circuitry; in fact, it need only emulate a slow input/output port. The techniques for implementing such microcomputer logic are well known.

The second level of synchronization is required to logically associate IRU and star tracker data. For example, the time between gyro readouts and star tracker readouts must be known to associate V & H readouts with IRU ...vity... Obviously, a common clock and time tagging hardware is an essential element in this synchronization. It will also be necessary in software to

carry time tags (in the form of counter values) along with rate values, V & H values, quaternion values, etc. This type of software logic is common in most process control applications.

It was mentioned earlier that if two star trackers are operating simultaneously, star identification may be performed in parallel but that correction of the quaternion (using two stars) is basically a sequential process. For this reason, it is desirable to put the two star tracker microcomputers in a master-slave relationship. This can be achieved by means of a "smart" switch. The function of the smart switch is to direct the data generated by the first tracker to lock on to a star to the master microcomputer system. The master microcomputer system may then proceed to identify the star and correct the quaternion. Should the second tracker acquire a star during this time, its data would be directed by the smart switch to the slave microcomputer system which would then proceed to identify the star. When the master microcomputer has finished correcting the quaternion based on data from the first star tracker, it would check with the slave microcomputer to determine if a second quaternion correction can be performed. If the Kalman filter can be run again, it is done at this time by the master microcomputer using data supplied by the slave microcomputer.

The master-slave relationship between star tracker microcomputers is suggested because it reduces the complexity of software needed for star tracker processing and because it minimizes the interfaces between the IRU subsystem and the star tracker subsystem. Implementation of the smart switch is not envisioned to be a difficult problem. Even if star tracker electronics cannot be extended to make a smart switch, it is possible to use normal switching logic driven by the master microcomputer system.

6.4 Orbit Generator and Resolver Software Analysis

The final elements of the OADS technique we examined were the orbit generator and the resolver. A detailed timing analysis of these two elements was conducted using the single and dual APU configuration described earlier. The orbit generator consists of accepting a spacecraft state vector from GPS and then propagating this state vector forward in time to the points at which spacecraft attitude is being generated. A brief analysis showed that for the missions being examined, only very slight errors resulted from using linear progagation as opposed to two body (conic) propagation. This fact may, of course, not be true in other missions having highly eccentric orbits. Linear interpolation can be very simply performed. First, an acceleration vector is computed based on consecutive GPS supplied velocity vectors. Once this is done, the state may be propagated using the equations:

$$\vec{V}_{i+1} = \vec{V}_i + \Delta t \, \vec{A}$$

$$\vec{P}_{i+1} = \vec{P}_i + \Delta t \, \vec{V}_i$$

where $\vec{A}$ is the acceleration vector. Analysis shows that these operations require 5.498 milliseconds in the single APU configuration, and 3.421 milliseconds in the dual APU configuration. These times include calculation of the acceleration vector which in actuality is performed only once in every six seconds.

The function of the resolver module is to calculate spacecraft attitude and orbit in the format of the end user. This data is then relayed to Earth as part of the downlink telemetry. The resolver software timing analysis was performed for an Earth mission since it was felt that these algorithms would place the greatest processing demands on the resolver

microcomputer system. The first algorithm in the resolver is to transform the GPS provided state vector to the proper reference frame:

$$\left[P_R\right] = \left[R\right]\left[P_{ECEF}\right]$$
$$\left[V_R\right] = \left[R\right]\left[V_{ECEF}\right]$$

where $\left[R\right]$ is a function of GMT and Earth's rotation rate. Next, the $\left[{}^{I}C_{L}\right]$ and $\left[{}^{I}C_{B}\right]$ matrices are constructed:

$$\vec{Z} = \vec{P} \times \vec{V}$$
$$\vec{Q} = \vec{Z} \times \vec{P}$$
$$\hat{P} = \vec{P}/|\vec{P}|$$
$$\hat{Z} = \vec{Z}/|\vec{Z}|$$
$$\hat{Q} = \vec{Q}/|\vec{Q}|$$

$$\left[{}^{L}C_{I}\right] = \begin{bmatrix} P_X & P_Y & P_Z \\ Q_X & Q_Y & Q_Z \\ Z_X & Z_Y & Z_Z \end{bmatrix}$$

and $\left[{}^{I}C_{B}\right]$ is a function of the quaternion as discussed earlier (see Star Tracker Software Analysis). $\left[{}^{F}C_{B}\right]$ may then be constructed using the equation:

$$\left[{}^{F}C_{B}\right] = \left[{}^{F}C_{L}\right]\left[{}^{L}C_{I}\right]\left[{}^{I}C_{B}\right]$$

where ${}_{F}C_{L}$

$$\begin{bmatrix} 0 & 0 & -1.0 \\ 1.0 & 0 & 0 \\ 0 & -1.0 & 1.0 \end{bmatrix}$$

Pitch, roll and yaw are then:

$$\text{Roll} = \tan^{-1} \left\{ {}_{F}C_{B}\,(2,3) \; {}_{F}C_{B}\,(3,3) \right\}$$
$$\text{Yaw} = \tan^{-1} \left\{ {}_{F}C_{F}\,(1,2) / {}_{F}C_{B}\,(1,1) \right\}$$
$$\text{Pitch} = \tan^{-1} \left\{ {}_{F}C_{L}\,(1,3) / \left[{}_{F}C_{B}^{-2}\,(2,3) + {}_{F}C_{B}^{2}\,(3,3) \right]^{\frac{1}{2}} \right\}$$

Finally, altitude, latitude, and longitude are calculated:

$$H = |\vec{P}| - R_E$$

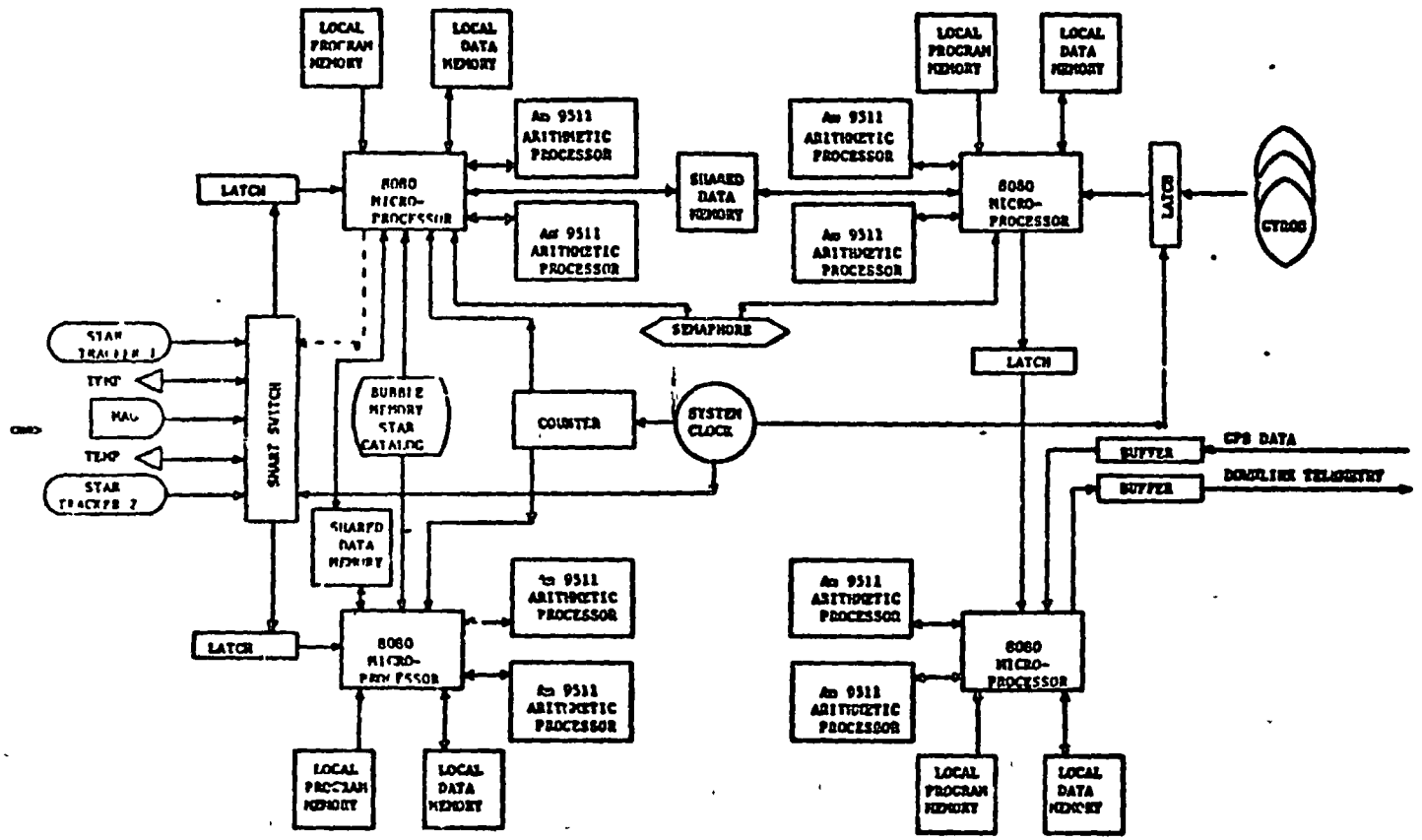
$$L_A = \sin^{-1} (\hat{P}_{Rz})$$

$$L_O = \tan^{-1} (\hat{P}_{Ry}/\hat{P}_{Rx})$$

For a more detailed discussion of resolver algorithms for the Earth mission and other missions, consult Section 3. Our timing analysis showed that it requires 59.637 milliseconds to perform resolver processing using a single APU configuration. Since this exceeds the maximum allowable processing time (50 milliseconds), this configuration is not satisfactory. Resolver timing in the dual APU configuration was 38.518 milliseconds. Performing both orbit generation and resolver algorithms in the same dual APU microcomputer would result in 41.939 milliseconds throughput or 84% system utilization. This was considered tolerable since worst case timing estimates were used throughout the analysis. It would also be possible to unload some orbit generator and resolver processing into the IRU microcomputer, if necessary. The primary reason for long processing times in the resolver algorithms is the heavy use of sine, cosine, arc tangent and square root functions.

6.5 QADS Microcomputer Systems Integration

The composite microcomputer system's block diagram is shown in Figure 6-6. For more information on the hardware aspects of this configuration, the reader should consult the next section. The following table summarizes the software timing analysis for the entire system.



OASIS MULTIPLE MICROPROCESSOR SYSTEM

FIGURE 6-6

TABLE 6.6 SUMMARY OF ESTIMATED OADS THROUGHPUT

ALGORITHM	PROCESSING TIME (ms)
IRU Processing	29.42
Star Tracker Processing	
Phase I (star identification)	134.3
Phase II (1 tracker - quaternion correction)	152.2
Phase II (2 trackers - quaternion correction)	235.8
Orbit Generator Processing	3.421
Resolver Processing	38.518

Since parallel, processing is used quite heavily in this system it is not possible to sum processing time and compute the throughput. The reader should consult the timeline diagram shown in Figure 6-7. The sum of IRU, orbit generator, and resolver processing times is equal to the latency time from when the gyros were read to when the downlink telemetry is available. This time is slightly greater than 71 milliseconds. As can be seen in Figure 6-7, all star tracker processing can be performed in parallel with IRU processing. Quaternion correction, Phase 2 star processing, requires 152 milliseconds for single tracker updates and 236 milliseconds for dual tracker updates. These time periods are noted on the figure as points q1 and q2, respectively. It should also be noted that Phase 1 star tracker processing is only required when a star is first identified. Therefore, consecutive star tracker measurements only require Phase 2 processing.

Our investigations show that use of a multiple microcomputer system for onboard attitude determination is quite feasible from the software timing view. Although our analysis did not cover some necessary housekeeping softwa-

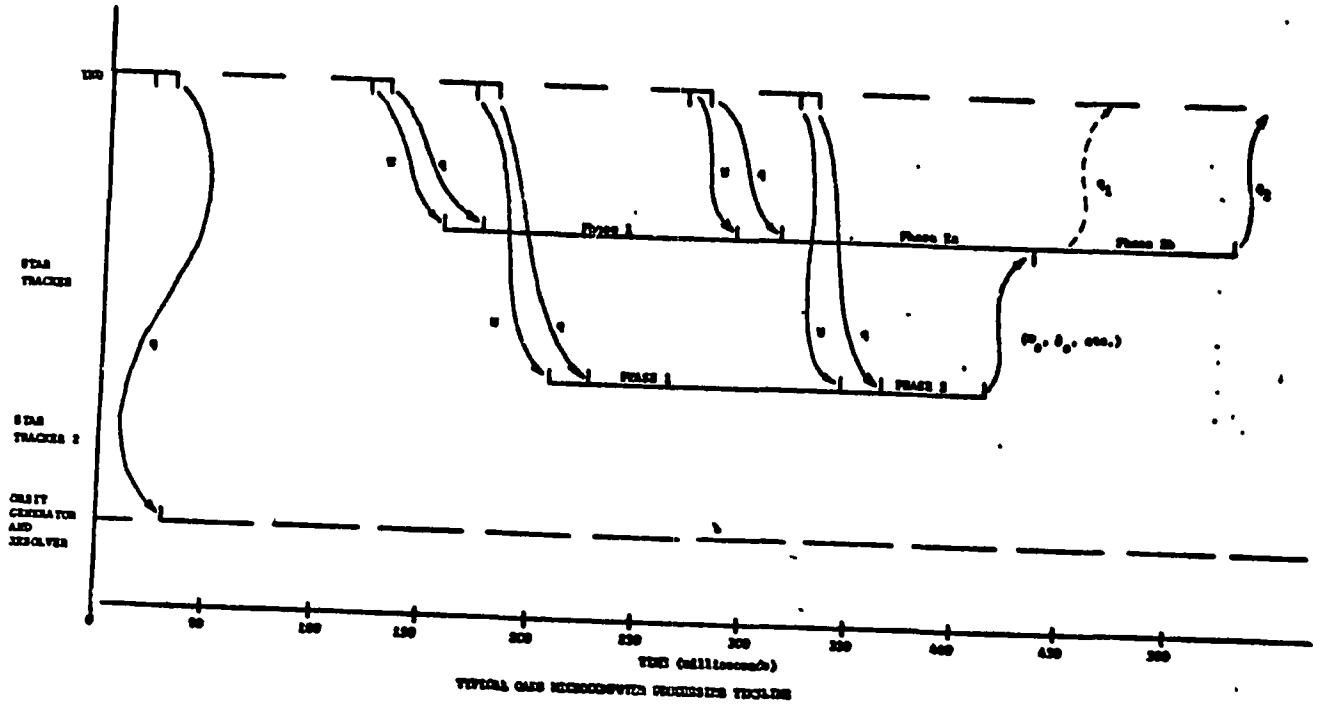


FIGURE 6-7

these items are considered minor compared to the computational processing that must be performed. It should also be remembered that in our analysis, worst case instruction times were used and no attempt was made to optimize overlapped execution in the arithmetic processing units. Timing analysis is, therefore, very conservative.

7.0 MICROPROCESSOR HARDWARE ANALYSIS

The purpose of this section is to provide some insight into the feasibility of using currently available LSI hardware in spaceborne applications. Also included in this section is a discussion of OADS mass storage requirements and LSI technology trends.

7.1 Analysis of Baseline Configuration Hardware

A system level block diagram of the baseline OADS microcomputer system used in the software timing analysis is shown in Figure 6-6. Table 7.1 shows estimated parts count and power consumption for the different elements used in the baseline configuration. Maximum power consumption occurs during Phase 1 star tracker processing and is estimated to be 363 watts. This may appear to be a very high power requirement; however, it must be realized that the duration of this peak power consumption lasts for only a few seconds, three or four times per orbit. Average power consumption estimates are considerably less, 39 watts, assuming that the star tracker microcomputers and bubble memory star catalog are powered down when not in use. It should be noted that the component count and power consumption estimates are very rough numbers since a detailed hardware schematic was not developed during this study. The component count is only based on SSI, MSI, and LSI devices and does not include discrete devices such as resistors, capacitors, etc. There are numerous semiconductor memory devices available on the market and semiconductor memory density is rapidly increasing. Since semiconductor memory is the largest contributor to both power consumption and component count, detailed component count and power consumption estimates should be delayed until breadboard prototying is initiated.

The primary components of the baseline configuration are Intel 8080 microprocessors and Am9511 arithmetic processors (see Appendix I).

TABLE 7.1 OADS BASELINE MICROCOMPUTER CONFIGURATION POWER CONSUMPTION ANT COMPONENT COUNT

	COMPONENT COUNT	TYPICAL POWER (WATTS)	MAX POWER (WATTS)
IRU	77	13.5	26.4
Star Tracker (Master)	653	107.6	213.3
Star Tracker (Slave)	149	25.3	49.7
OG/Resolver	77	13.5	26.4
Bubble Memory	30	48	N/A

As was previously noted, these devices are fabricated using N-channel metal oxide semiconductor (NMOS) technology. This technology is considered feasible for spaceborne applications from the viewpoints of throughput, power consumption and temperature hardening. We have, however, discovered that NMOS devices are very susceptible to radiation contamination. Depending on orbit geometry, a total dosage of over 10^7 Rad (si) can be expected in a multi-year mission. The radiation dosage threshold for NMOS devices is approximately 1.5×10^3. Upwards of 3 gm/cm^2 of aluminum would be required to shield NMOS devices on long-duration missions. This is considerably more than the shielding requirements on most present-day flight computers. Circuit redesign of NMOS components could increase their radiation tolerance by an order of magnitude; however, even this dosage does not appear to be satisfactory for many spacecraft requirements.

7.2 Available Fabrication Technology

The following paragraphs qualitatively summarize the advantages and disadvantages of other fabrication techniques. Table 7.2 is included at

the end of this discussion so that the reader may quickly compare the performance features of alternate fabrication techniques. The reader should also consult Appendix I.for greater detail on the specifications of individual LSI components. Section 8 contains supplementary information on the requirements for manufacturing LSI products specifically designed for spaceborne applications.

P-Channel Metal Oxide Semiconductor (PMOS) - During discussions with Bob Stermer (NASA Langley) regarding bubble memory technology, mention was made of the radiation hardening problems associated with microprocessors. Because of NMOS vulnerability, Bob Stermer has been examining PMOS devices for spaceborne processing applications; specifically, the Western Digital LSI 11. This device is sold in packaged form by Digital Equipment Corporation (DEC) as the PDP 11/03. The LSI 11 is a very powerful multi-chip microprocessor. It is a 16-bit device that executes the PDP 11/40 instruction set including 32 bit floating point arithmetic. Since the LSI 11 is custom made for DEC, detailed data on the microprocessor's architecture and circuitry are not generally available. It is known that the processor is microprogrammed and hence it could be customized for certain applications such as OADS. PMOS, although using less power than NMOS, also results in slower machine cycle times. Furthermore, Bob Stermer indicated that he felt the radiation dosage threshold was around 5×10^4 RAD (si) which is still far below the requirements needed for five-year or longer missions.

Integrated Injection Logic I^2L - Texas Instruments was the first company to announce a major microprocessor family based on I^2L technology, the SBP 9900. Typically, I^2L has greater power consumption than NMOS, however, slightly

higher machine cycle times should be possible. In its commercial form, the SBP 9900 has only slightly better radiation hardening than the NMOS 8080. Texas Instruments, under contract by the Navy, has been able to produce the SBP 9900 with a radiation tolerance of better than 2×10^6 RAD (si). With minimal shielding, such a device could be used in spaceborne processing applications such as OADS.

Bipolar - Although there are many vendors of bipolar LSI devices, the AMD 2903 is representative of microprocessor technology in this area. The most obvious disadvantage of bipolar processors is their very high power consumption. This, of course, must be weighed against processing speed and environment considerations. Bipolar microprocessors are five to ten times faster than NMOS and can withstand total radiation dosages of better than 10^7 RAD (si). The shielding supplied by the spacecraft structure itself is probably sufficient for five-year missions. Unlike the microprocessors previously mentioned, the bipolar AMD 2903 is supplied as a 4-bit wide slice of an arithmetic and logic unit (ALU). This has two important impacts. First, by cascading slices together, an arbitrarily wide processor can be built. Very long word length processors, however, become slow because of the latency associated with propagating signals through the bit slices. The other important aspect to the AMD 2903 bit slice is that the processor instruction set is tailored by the user to his application by means of microcode. Since the Interdata 8/16 minicomputer is made from bipolar bit slice architecture, very realistic software estimates can be made using the Interdata 8/16 instruction times.

Silicon on Saphire (SOS) - For the past two years, the industry has been awaiting the release of LSI devices based on SOS technology. Articles published in technical journals and other trade literature indicate that SOS devices

are capable of having most of the advantages associated with other fabrication techniques and few of the disadvantages. Unfortunately, the production development process required for SOS appears to be moving very slowly. Hewlett Packard announced an SOS microprocessor last year but the processor is imbedded in other Hewlett Packard equipment and detailed information is not generally available. Rumors have been circulating that RCA will release a version of their 1802 microprocessor using SOS; however, no such announcement has been made by RCA nor does one appear to be forthcoming in the near future. One very significant item discovered during our study was an ALU slice developed by Rockwell using CMOS/SOS technology. Rockwell, under the Advanced Computer Technology (ACT-I) contract from SAMSO, developed their ALU specifically for use in military applications such as spaceborne processing. The 8-bit wide ALU is radiation hardened and presently being evaluated for use on the MX Program. Rockwell also has an internal study task in progress that is building a breadboard computer using the CMOS/SOS ALU. This computer will be evaluated for GPS applications.

7.3 Mass Storage Requirements

As part of this task, we performed an analysis of the feasibility of using bubble memory mass storage for saving the star catalog. To minimize star tracker processing, it is advantageous to not only save right ascension and declination in the star catalog but also matrices which are functions of sine and cosine. This results in a total of 14 parameters per entry. Assuming 1500 stars and 4 bytes per entry, then 84,000 bytes of storage are needed for the star catalog. Since 84,000 bytes is beyond the direct addressing range of present microprocessors, memory paging hardware would be required to

store the star catalog in main memory. This, of course, increases both hardware and software complexity. A second disadvantage is that if the star catalog were stored in read/write random access memory (RAM), then power would have to be maintained since this is a volatile storage medium Bubble memory, on the other hand, will not lose its contents when power is removed. The primary disadvantage to bubble memory is the long access time. Up to 2 milliseconds latency is associated with a random access and approximately one tenth millisecond latency for queued access. If the indirect indexing table technique (described in Section 6) were used, queued access into a bubble memory star catalog is possible. It would also be possible to overlap processing of star identification algorithms with access to the bubble memory star catalog. Since Phase 1 star tracker processing is only performed once for each series of tracker measurements, the worst impact that a bubble memory star catalog would have would be a slight increase in the lead time required for quaternion correction.

Two vendors are predominant in the bubble memory area: Texas Instruments and Rockwell. A Texas Instrument brassboard system is presently under evaluation at the Air Force Avionics Laboratory, Wright Patterson Air Force Base. Another prototype system based on the Rockwell unit is under evaluation at NASA Langley. In very general terms, power consumption is around 6 watts per 100,000 bits. Hence, a 100,000 byte unit (the size needed for the star catalog) would require about 48 watts of power. Also of interest is the fact that radiation hardening of the bubble memory itself is not a problem. The sense amplifiers, however, are radiation sensitive. It was learned in discussions with Texas Instrument engineers that they are presently conducting internal studies to determine if alternate fabrication techniques, such as I^2L, can be used to eliminate this problem area.

The primary advantage of bubble memory is that it is a non volatile, read or write storage medium. The analysis currently being done by NASA and the Air Force is to study the feasibility of using bubble memory to replace spaceborne tape recorders and drum memory systems. Bob Stermer of Langley feels that flight quality systems should be available as early as 1980. He has indicated that work is presently going on to define a NASA standard low-cost bubble memory system.

TABLE 7.2 PERFORMANCE COMPARISON[1] OF NMOS AND ALTERNATE FABRICATION TECHNOLOGIES

DEVICE	PERFORMANCE FACTOR[2]
PMOS[3]	1.29
I^2L	.489
BIPOLAR	.07
CMOS/SOS	Not Available

NOTES:

1. Comparison was done using a portion of IRU algorithms and should not be construed as a benchmark test.
2. Performance factor is defined as alternate device timing divided by NMOS device timing. Hence, the smaller the number the better the relative performance.
3. Comparison based on commercial versions.

8.0 MICROPROCESSOR SYSTEMS ANALYSIS

The two previous sections described the results of our analysis of a baseline microprocessor configuration executing the nominal processing associated with on-board attitude determination. This section discusses other aspects associated with this application, the conclusions we have reached after performing this study, and suggestions as to what activities should be pursued to realize a flight quality OADS microcomputer system.

8.1 Supplementary Design Issues

There is one important operational aspect of an OADS microcomputer system which has not been addressed: In-flight software updates. Most special purpose ground based microcomputers have their programs stored in read only memory (ROM). The reason for this is that ROM is non volatile and will not lose its contents when power is removed. The contents of read/write RAM will be lost if power is not maintained. If in-flight reprogramming is a requirement in the OADS application, it will not be possible to use the presently available forms of ROM. The power consumption estimates shown in Table 7.1 are based on the use of RAM. If ROM could have been used, these estimates would be reduced by approximately fifty percent.

Assuming that it is desirable to maintain an in-flight reprogramming capability, there are several implementation strategies available to the designer. First, if RAM is used throughout the system, it will be necessary to either maintain power to all subsystems at all times or to reload memory after power is first applied. Since the star tracker is used cyclically, it would not be economical to keep power applied to this subsystem; a memory reload strategy would be much more desirable. To reload memory, it is necessary to have a mass storage medium, such as a tape recorder, available. If a bubble memory unit is used to store the star catalog, this device could

also be used to save the program during power down.

Two other design possibilities may be possible. The first alternative is to use core memory. This technique has the advantage of being non-volatile and has already been used in flight computers. The primary drawbacks to core is that it requires considerable expense to construct it, it is much less modular than semiconductor, and it is slower than many of the presently available semiconductor units. The second design possibility is the use of electrically reprogrammable read only memory. There are many reprogrammable read only memories currently available. These devices are programmed electrically but can be erased by exposure to high intensity ultraviolet light. Recently, the industry has been investigating the design of electrically reprogrammable read only memory. Should such a technology come into production, it would be highly desirable in many spaceborne applications.

A design issue which comes into play from the moment flight hardware is being considered is quality assurance. The testing that semiconductor vendors are doing for their commerical components is completely inadequate for flight use. JPL has started a testing program for the 8080, Am2900 and RCA 1802. Thus far, they have found many variations between manufacture's specifications and test results. Perhaps, more importantly, RCA is the only semiconductor vendor to take advantage of the JPL findings.

Many of the present testing techniques used with today's flight computer can be applied to spaceborne microcomputer systems. These techniques include factory acceptance testing and burn-in testing at both the component and system level. Component traceability must be maintained throughout the testing phase. Self test software must also be written for flight qualification. Unlike present flight computers using SSI and MSI components, microprocessors

and other LSI devices cannot be decomposed to force some error conditions. Because of this characteristic, the test software must be much more extensive and detect input patterns to which the devices are sensitive. This same software, or elements of it, can possibly be executed as a background task during the flight.

8.2 Conclusions

The primary conclusion we reached after performing our analysis of a microcomputer based on-board attitude determination system is that presently available commercial LSI devices can be configured to obtain the required performance. The important phrase in the previous sentence is commercial LSI devices. During this study we were not able to find any microcomputer system which was ideally suited for the spaceborne environment and had the necessary performance for applications such as OADS. The major problem associated with most LSI hardware was lack of radiation hardening. Power consumption problems also influence the hardware selection but to a lesser extent. We feel that no significant benefits will be derived from further microprocessor performance analysis until standard flight quality LSI hardware has been defined. The following paragraphs describe what components are necessary to realize the benefits of spaceborne microcomputer systems. Power consumption problems also influence the hardware selection but to a lesser extent. We feel that no significant benefits will be derived from further microprocessor performance analysis until standard flight quality LSI hardware has been defined. The following paragraphs describe what components are necessary to realize the benefits of spaceborne microcomputer systems. These recommendations are based not only on our OADS analysis, but also on previous investigations we have performed and on discussions with other industry and DOD personnel.

It is unlikely that one microcomputer system will be sufficient to properly handle all the spaceborne applications that are becoming evident. Therefore, it would be advantageous to have a number of low-cost standard systems from which a designer could select. We envision that three types of microcomputer systems are needed. At the low of the performance spectrum, a single chip microprocessor should be available for non time-critical process control applications. By supplementing this microprocessor with an arithmetic processor, many computational oriented applications such as OADS could be handled. For high performance process control and computational oriented applications, a multiple chip bit slice processor sppears to be appropriate. I^2L technology could possibly be used for both the single chip microprocessor and arithmetic processor. The SBP 9900 processor is, in fact, very close to the desired hardware. Its primary drawbacks are its register and CRU architecture and the limitation of memory expansion beyo 32K words without mapping hardware. The Rockwell CMOS/SOS arithmetic and logic unit appears to be the most likely candidate for use in a high performance bit slice processor. Its primary drawback is that there is not an extensive selection of SOS support devices that would be needed to construct a full microcomputer.

The microcomputer systems just outlined would probably be considered conservative development efforts. There have been many exaggerated claims by quasi technical observers of the semiconductor industry that today's most powerful computers will one day be available on a single chip. While it is true that the semiconductor industry is only in its infancy and that significant LSI performance gains will be obtained, it is unlikely that even moderate performance minicomputer systems on a chip will be available in the next few years. It should also be remembered that semiconductor

vendors are primarily commercial oriented. Their earnings are based on sales of calculators, digital clocks, and home video games. Without the proper technical guidance and financial encouragement, semiconductor vendors will not produce flight-quality LSI hardware.

9.0 NSSC-I SOFTWARE ANALYSIS

This section describes our software analysis for the NSSC-I based spaceborne attitude control system. The OADS elements which we have examined are described in Section 6, Microprocessor Software Analysis.

Timing analysis and memory size analysis was done for each element of the system. Data used in the analysis was assumed to be a double precision (36 bits) fixed point format. It was also assumed that data did not exceed this range and that the wide range of values of data could adequately be handled by scaling. The timing analysis draws heavily from work done by Computer Sciences Corporation (see reference CSC 1976 and Appendix II). These estimates assume a NSSC-1 cycle time of 1.25 microseconds. The memory size analysis consists of both data and code estimates. The data required are listed in the data table. The code estimates were determined by changing all macros for common routines (see CSC 1976) into subroutines and then estimating the code necessary to invoke these subroutines.

For more information on the algorithms used in this section, the reader is referred to Sections 3 and 5.

9.1 IRU Software Analysis

There are five steps required for IRU processing. The algorithms involved are: data editing, rate computation, rate compensation, data reduction, and integration. The timing is shown in Table 9.1. Memory estimates are shown in Table 9.2.

9.2 Star Tracker Software Analysis

There are eight steps required for Star Tracker processing. The algorithms include: conversion to engineering units, synchronization to the IRU, compensation, data editing, conversion to inertial system, star identification, Kalman filter correction, and integration.

TABLE 9.1 ESTIMATED IRU PROCESSING TIMES FOR NSSC-I COMPUTER

ALGORITHM	ALGORITHM PROCESSING TIME (ms)	ACCUMULATIVE PROCESSING TIME (ms)
Data Edit	.378	.378
Rate Computation	1.860	2.238
Compensation	10.703	12.941
Data Reduction	19.770	32.711
Integration	11.864	44.575

TABLE 9.2 IRU DATA TABLE

DATA ITEM(S)		NSSC-I SIZE (18 BIT WORDS)
P_i, $i = 1,6$		12
W_i, $i = 1,6$		12
$\frac{\Delta\theta}{\Delta t}$		2
Wx_{ci} $i = 1,3$		6
Wy_{ci} $i = 1,3$		6
ρ_x, ρ_y, ρ_z		6
A', B', E'		6
By, Uez, C'_α		6
$[H^T WM^{-1} H]^{-1}$	(3 x 3)	18
$[W_M^{-1}]$	(6 x 6)	72
$[H^T]$	(3 x 6)	36
$[W_B]$	(3 x 1)	6
$\Delta t/_2$	2	
$[M_i]$	(4 x 1)	8
$[\delta_i]$	(4 x 1)	8
$[\Omega_i]$	(4 x 4)	32
$[q_i]$	(4 x 1)	8
$[Y_i]$	(4 x 1)	8
$[\delta_{i+1}]$	(4 x 1)	8

TABLE 9.2 IRU DATA TABLE - (Continued)

DATA ITEM(S)		NSSC-I SIZE (18 BIT WORDS)
$[\Omega_i + 1]$	(4 x 4)	32
$[q_i + 1]$	(4 x 1)	8
Total Data		302 Words
Code Estimate		300 Words

The timing is shown in Table 9.3. The memory estimates are shown in Table 9.4. The multiple phase star tracker throughput timing is shown in Table 9.5. The description of each phase is found in Microprocessor Software Analysis, Section 6.

9.3 Orbit Generator and Resolver Software Analysis

The timing analysis of these two steps are shown in Table 9.6. The primary factor causing large processing times for the resolver algorithms is the use of trigometric functions. If less accuracy were required or fast table lookup operations could be used, resolver processing could be significantly enhanced. Memory estimates for orbit generator and resolver processing are shown in Table 9.7.

9.4 OADS NSSC-I System Integration

Tables 9.8 and 9.9 summarize the timing and memory requirements needed for processing the OADS algorithms. These estimates do not include overhead used by any NSSC-I executive software. The largest single factor in memory usage is the star catalog. This table requires 42,000 words or 82% of the total memory requirement. It is suggested that the full star catalog be placed in a mass storage system, such as a drum or bubble memory system (see Section 7).

Because the NSSC-I must perform all processes sequentially, it is obvious from the timing analysis summary (see Table 9.8) that it it not possible to execute the OADS algorithms on the NSSC-I in 50 millisecond cycles. The reader should consult the following section for a discussion of how a multiple microcomputer system could be integrated into the Multi Mission Spacecraft and interfaced to the NSSC-I to provide the onboard attitude determination capability.

TABLE 9.3 ESTIMATED STAR TRACKER PROCESSING TIMES FOR NSSC-I COMPUTER

ALGORITHM	ALGORITHM PROCESSING TIME (ms)
Convert to engineering units	20.466
Synchronize to IRU	4.555
Compensator	14.539
Data Edit	.274
Convert to Inertial	58.546
Star Identification	85.600
Kalman Filter	196.105
Integration	20.498

TABLE 9.4 STAR TRACKER DATA TABLE

DATA ITEM(S)		NSSC-I SIZE (18 BIT WORDS)
V, H, Vc, Hc, C		10
$[W_B]$ (previous value held)	(3x1)	6
$[W_{BS}]$	(3x1)	6
$[S^C_B]$	(3x3)	18
$[W_S]$	(3x1)	9
Δt		2
V2, V3, H2, H3, VH2, V2H, VH (Temporary Variables)		14
T1, T2, T3, T4, T5, T6, T7, T8, T0		18
[Temperature]	(100x10)	2000
[Magnetic]	(100x10)	2000
[Temp Slopes]	(100x1)	200
[Mag Slopes]	(100x1)	200
VMAX, HMAX	4	
$[S_{S_i}]$	(3x1)	6
$[I^C_B]$	(3x3)	18
$[B^C_S]$	(3x3)	18
$[S_{I_i}]$	(3x1)	6
α, δ, X2, Y2, Z2		10
[Star Cat]	(1500x14)	4200
[Indices]	(360x1)	720
[ ]	(3x2)	12

TABLE 9.4 STAR TRACKER DATA TABLE - (Continued)

DATA ITEM(S)		NSSC-I SIZE (18 BIT WORDS)
$[I^{C}_{S}]$	(3x3)	18
[AAMAT]	(3x3)	18
[E]	(3x3)	18
[Y]	(3x1)	9
[PMAT]	(3x3)	18
[MMAT]	(3x3)	18
[RMAT]	(3x3)	18
[I]	(3x3)	18
[VAR]	(3x1)	6
[QMAT]	(3x3)	18
[DMA]	(3x3)	18
[Q]	(1x3)	6
K_q		2
$[W_1]$	(3x7)	42
TOTAL DATA		47,522 words
CODE ESTIMATE		2,030 words

TABLE 9.5 MULTIPLE PHASE STAR TRACKER THROUGHPUT FOR NSSC-I

	PROCESSING TIME (ms)
Phase 1	183.98
Phase 2 -	
Single Star Tracker	314.983
Phase 2 -	
Dual Star Tracker	609.468

TABLE 9.6 ESTIMATED ORBIT GENERATOR AND RESOLVER PROCESSING TIMES FOR NSSC-I COMPUTER

	PROCESSING TIME (ms)
Orbit Generator	9.608
Resolver	177.919
Subtotal	187.527

TABLE 9.7 ORBIT GENERATOR AND RESOLVER DATA TABLE

DATA ITEM(S)		NSSC-I SIZE (18 BIT WORDS)
[A]	(3x1)	6
[V]	(3x1)	6
[P]	(3x1)	6
[I]	(3x1)	6
$[I^{C}_{L}]$	(3x3)	18
$[F^{C}_{L}]$	(3x3)	18
$[F^{C}_{B}]$	(3x3)	18
Roll, Yaw, Pitch, Dum		8
Altitude		2
Latitude		2
Longitude		2
TOTAL DATA		92 WORDS
CODE ESTIMATED		350 WORDS

TABLE 9.8 SUMMARY OF ESTIMATED PROCESSING TIMES FOR NSSC-I COMPUTER

	PROCESSING TIME (ms)
IRU Processing	44.575
Star Tracker - Star Identification	123.98
Star Tracker - Quaternion Correction (1 Tracker in Use)	314.983
Star Tracker - Quaternion Correction (2 Trackers in Use)	609.468
Orbit Generator/Resolver	187.527

TABLE 9.9 SUMMARY OF ESTIMATED MEMORY SIZE FOR NSSC-I COMPUTER

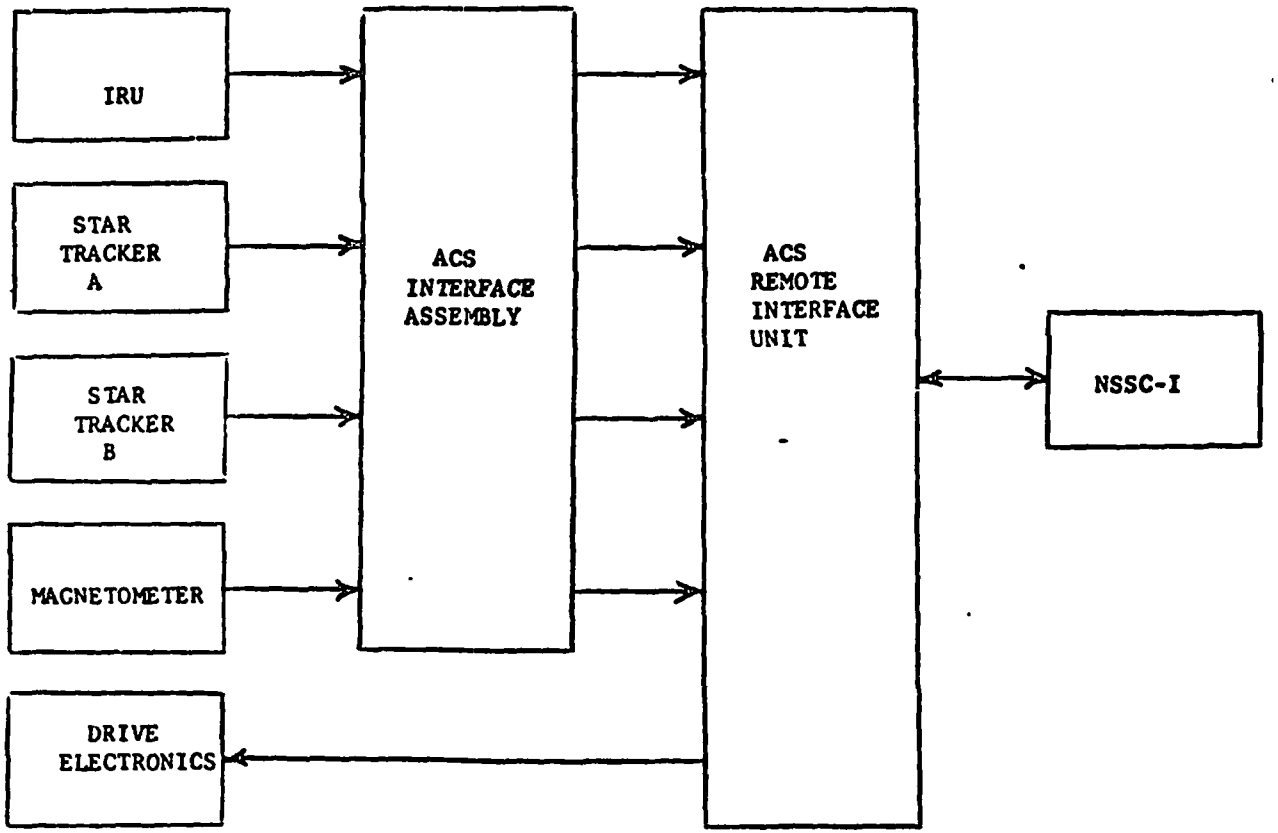
		NSSC-I SIZE (18 BIT WORDS)
IRU Processing	Data	302
	Code	300
Star Tracker	Data	47,522
	Code	2,030
Orbit Generator/Resolver	Data	92
	Code	350
General Routines	Code	670
TOTAL DATA		47,888 WORDS
TOTAL CODE		3,350 WORDS
TOTAL MEMORY SIZE		51,238 WORDS

10.0 NSSC-I PERFORMANCE ENHANCEMENT APPROACHES

Our analysis showed that it was not feasible to implement the attitude determination algorithms, previously described, in the NASA Standard Computer -I. We have, therefore, investigated the concept of supplementing the NSSC-I with microcomputer equipment. This concept is in complete agreement with the specification for MMS attitude control subsystem which suggested the use of a dedicated processor if it is determined that the NSSC-I is not capable of supporting the total ACS computational requirements. The following paragraphs describe some design approaches which could be used to enhance the NSSC-I for use in computational oriented applications such as OADS.

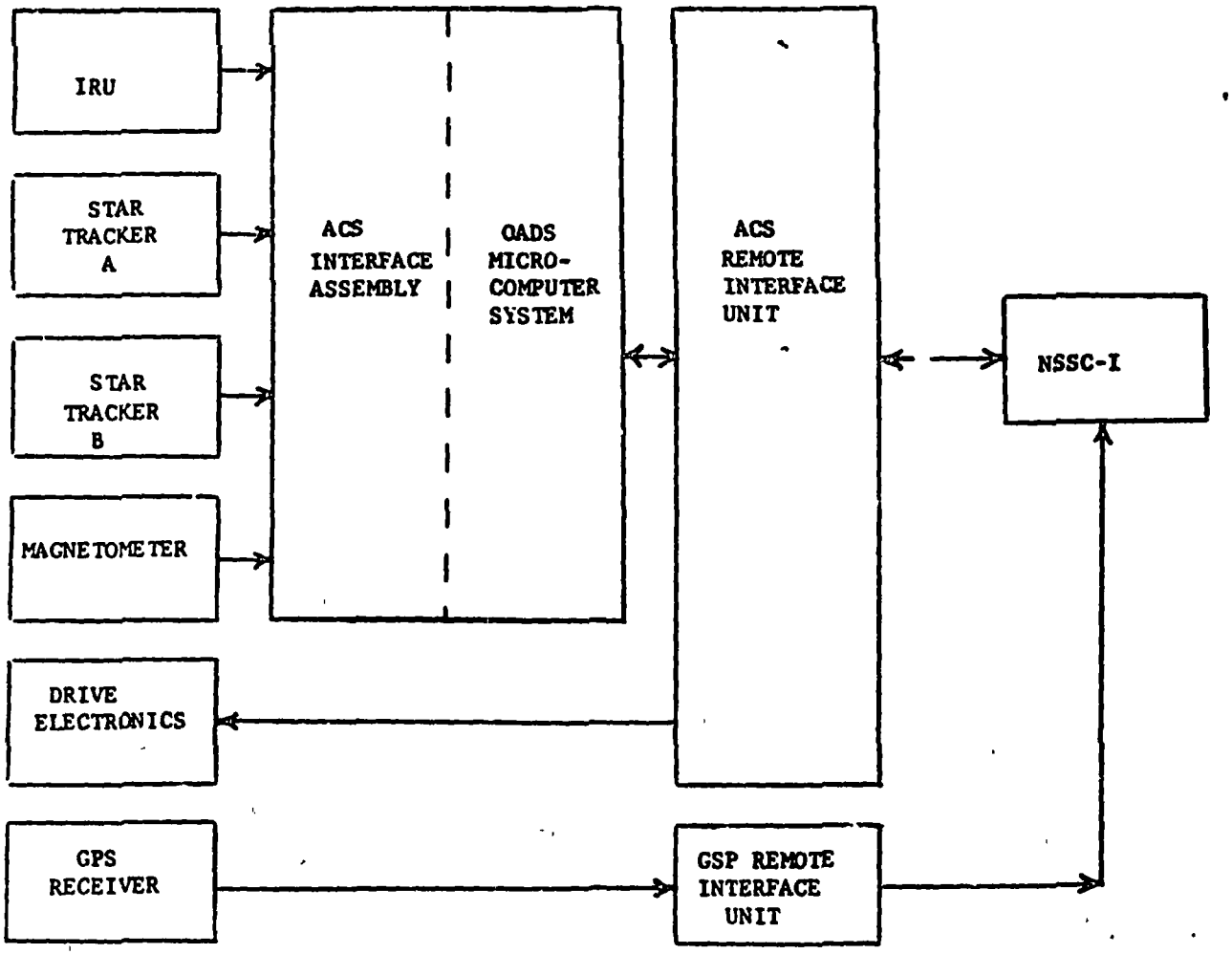
Figure 10-1 is a block diagram of the present ACS module configuration for the Multi Mission Spacecraft. It appears that a multiple microcomputer system, such as the baseline configuration described earlier, could be added for on-board attitude determination without a large impact. Although the system's physical layout has not been examined, it may be possible to replace or expand the ACS Interface Assembly with the OADS microcomputer system. Such a configuration is shown in Figure 10-2. Functionally, the NSSC-I would be responsible for major data management operations such as control of the interface between GPS and the OADS computer and the control of the telemetry to and from the OADS microcomputers. It appears that the attitude control algorithms could be implemented on the NSSC-I or a dedicated microcomputer system could be assigned this task.

Another spaceborne microprocessor application which is related to OADS is the Instrument Telemetry Packet (ITP) concept developed by Albert Ferris and Edward Greene of GSFC. Their concept is to improve onboard data management so that ground telemetry processing functions may be made more



EXISTING ACS MODULE CONFIGURATION

FIGURE 10-1

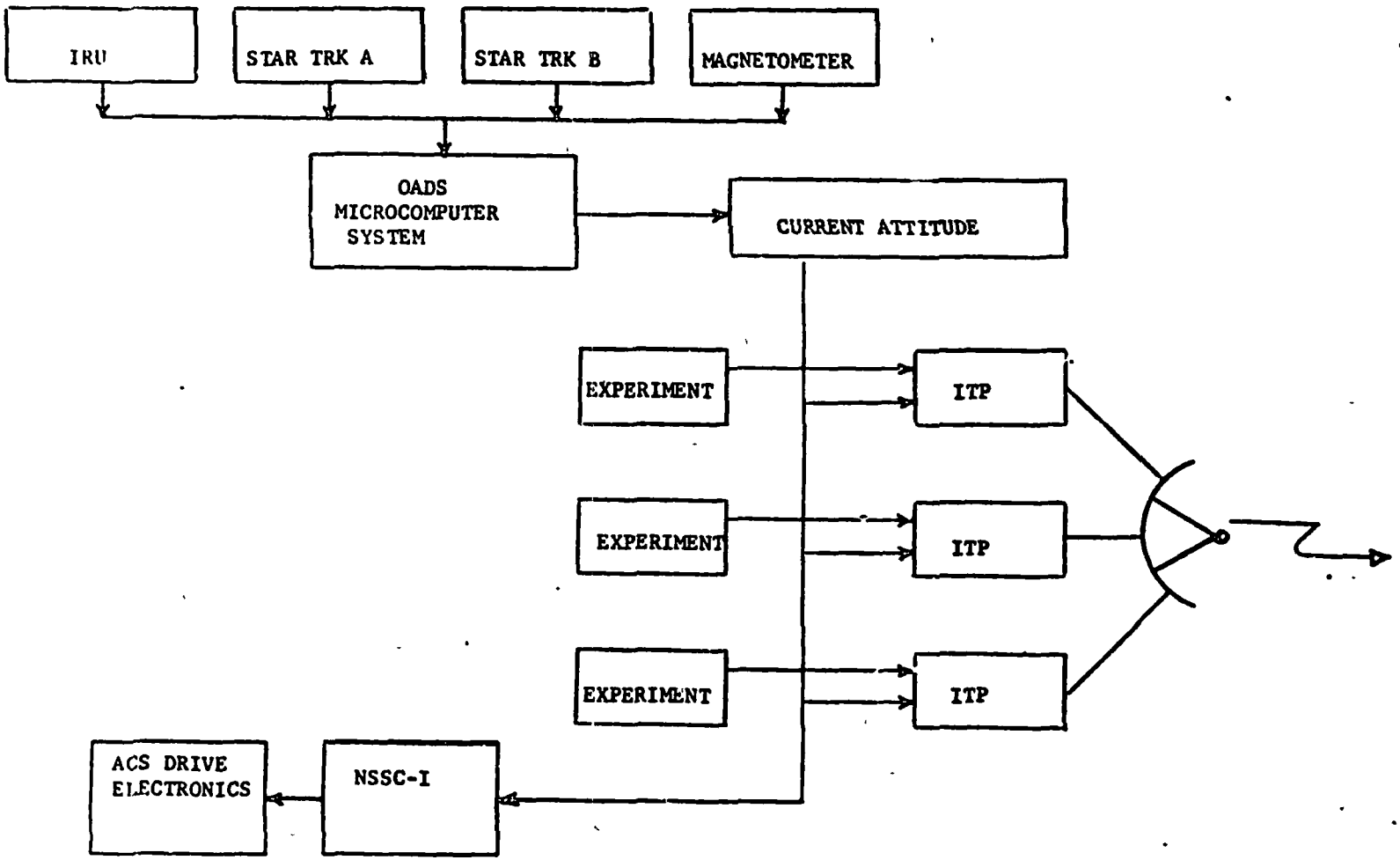


ACS MODULE SUPPLEMENTED BY OADS MICROCOMPUTER SYSTEM

FIGURE 10-2

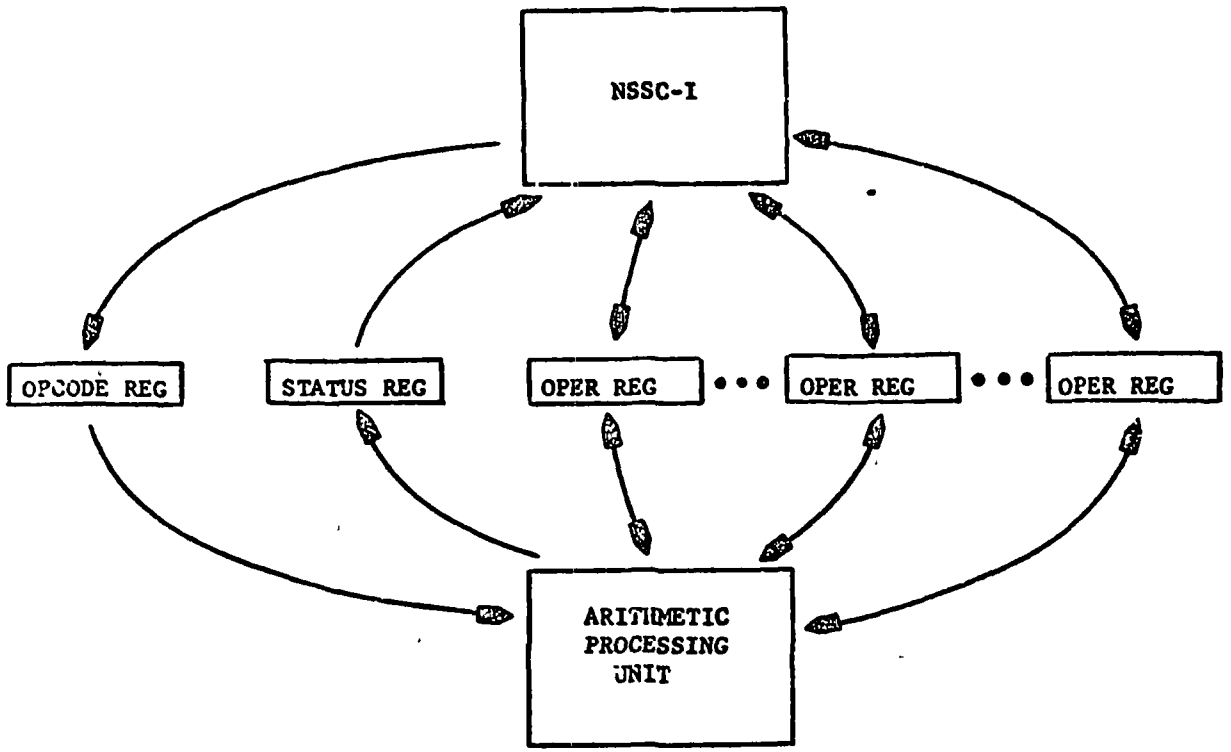
responsive to scientific investigators. ITP requires that the telemetry data for a single spaceborne instrument or subsystem be assembled into a telemetry packet containing only the data from a single instrument along with any required ancillary data such as spacecraft clock and possibly the spacecraft state and attitude vectors. The MMS data management architecture does not appear to be well suited to this approach since the central computer is not the focal point for the downlink telemetry. If such a configuration is considered in the future, an OADS microcomputer system could be conveniently incorporated into the design. Figure 10-3 illustrates such a system.

As mentioned in Section 8, we feel that microcomputers having the necessary characteristics for the OADS spaceborne environment do not presently exist. Although construction of such devices are well within the state of the art, it could very well require three or more years before flight quality hardware was ready. It, therefore, appeared judicious to investigate what short-term solutions exist for improving the NSSC-I performance to the point where it could handle applications such as OADS. Figure 10-4 shows an NSSC-I computer interfaced to an arithmetic processing unit. The APU could be constructed using bipolar bit slice processors such as the AMD 2900 series. These devices are very fast and radiation hardened to over 10^7 Rad (Si). The bipolar bit slice processors are very power consumptive devices and if an entire spaceborne computer were built around these devices, power requirements could be too great. By limiting their use to the arithmetic processing function, this drawback is tolerable. It appears that the bipolar bit slice processor would require approximately 40 to 50 watts but could be powered down when not in use. Floating point adds



OADS - EXPERIMENT - ACS RELATIONSHIP

FIGURE 10-3



NSSC-I SUPPLEMENTED BY BIT-SLICE ARITHMETIC PROCESSING UNIT

FIGURE 10-4

would require five to six microseconds and floating point multiplies could be accomplished in 13 to 20 microseconds depending on normalization and precision. Using these performance numbers, it was determined that the IRU algorithms could be processed in eleven milliseconds, which is four times the performance of the NSSC-I computer by itself. If trigometric functions are placed in microcode, even greater performance benefits could be obtained for star tracker and resolver processing.

It may be possible to interface the bipolar APU to the NSSC-I in a number of different ways. The most attractive manner, because of its simplicity, is the use of an external register file. The NSSC-I could load or read operand registers using its standard input and output facilities. It would also load an opcode register which informs the APU what function is to be performed Upon completion of the operation, the APU loads the status register for the NSSC-I.

Should NASA decide to postpone usage of stand-alone microprocessors until more desirable devices are available, the bipolar APU appears to be a reasonable interim solution.

11.0 OADS TESTING, OPERATIONAL UPDATE REQUIREMENTS AND POST LAUNCH VERIFICATION

11.1 Testing Requirements

In order to insure the success of the MMS mission using OADS, performance testing is vitally important. The OADS testing shall contain two levels:

- Hardware component testing under the MMS environment
- Overall OADS system closed-loop testing

The testing requirements of these two test levels are discussed in the following paragraphs.

11.1.1 Hardware Components

The study results presented in this report indicated that the NASA Standard IMU (DRIRU-II), NASA Standard tracker (SST), GPS, and microprocessors are recommended for the on-board attitude determination system. To insure performance, each hardware unit of the OADS system must be fully tested.

NASA Standard IMU (DRIRU-II) - Testing procedure and facility proposed by Teledyne Teledyne, I is sufficient to verify the DRIRU-II system specification. Additional emphasis should be placed on the MMS environment and requirements. The error parameters of scale factor, spin axis orthogonality, misalignment and bias are crucial to the MMS mission successfulness and, therefore, should be verified carefully. The on-board rate compensator accuracy should be fully investigated of its error reduction capability.

NASA Standard Star Tracker (SST) - Testing procedure and facility proposed by Ball Brother is sufficient to verify the SST performance. Additional emphasis should be placed on the star signal acquisition

and accuracy when the tracker is operating under the MMS orbital rate, and the V, H readout compensator due to temperature, magnetic field and star intensity variation of the MMS missions.

GPS - The testing of the GPS Receiver/Processor Assembly with OADS consists of testing and interface between the GPS Receiver/ Processor Assembly and OADS and investigating the accuracy of the onboard orbit propagator. Once the receiver is integrated with OADS, the onboard orbit propagator accuracy can be investigated on the ground using real GPS data.

Microprocessor - Testing of the microprocessor is discussed in Section 8.1

11.1.2 OADS closed-loop system

A complete OADS closed loop system testing is recommended to test the interface among OADS components, data processing, and system capability under MMS operational environment. An overall OADS testing set up flow diagram, shown in Figure 11.1, is suggested. A major portion of this breadboard will be completed in the later part of 1978 and ready for system testing in 1979 by Martin Marietta Aerospace for another spacecraft application. With minimum modification, this breadboard can immediately be used for OADS overall system testing.

11.2 Operational Update Requirements

The operational update requirements of OADS can be separated into two categories; the onboard system update and the ground operational update. The preliminary onboard update requirements were discussed in Section 5 based upon the study results of the OADS performance of various MMS missions. The major update requirements are summarized in the following table.

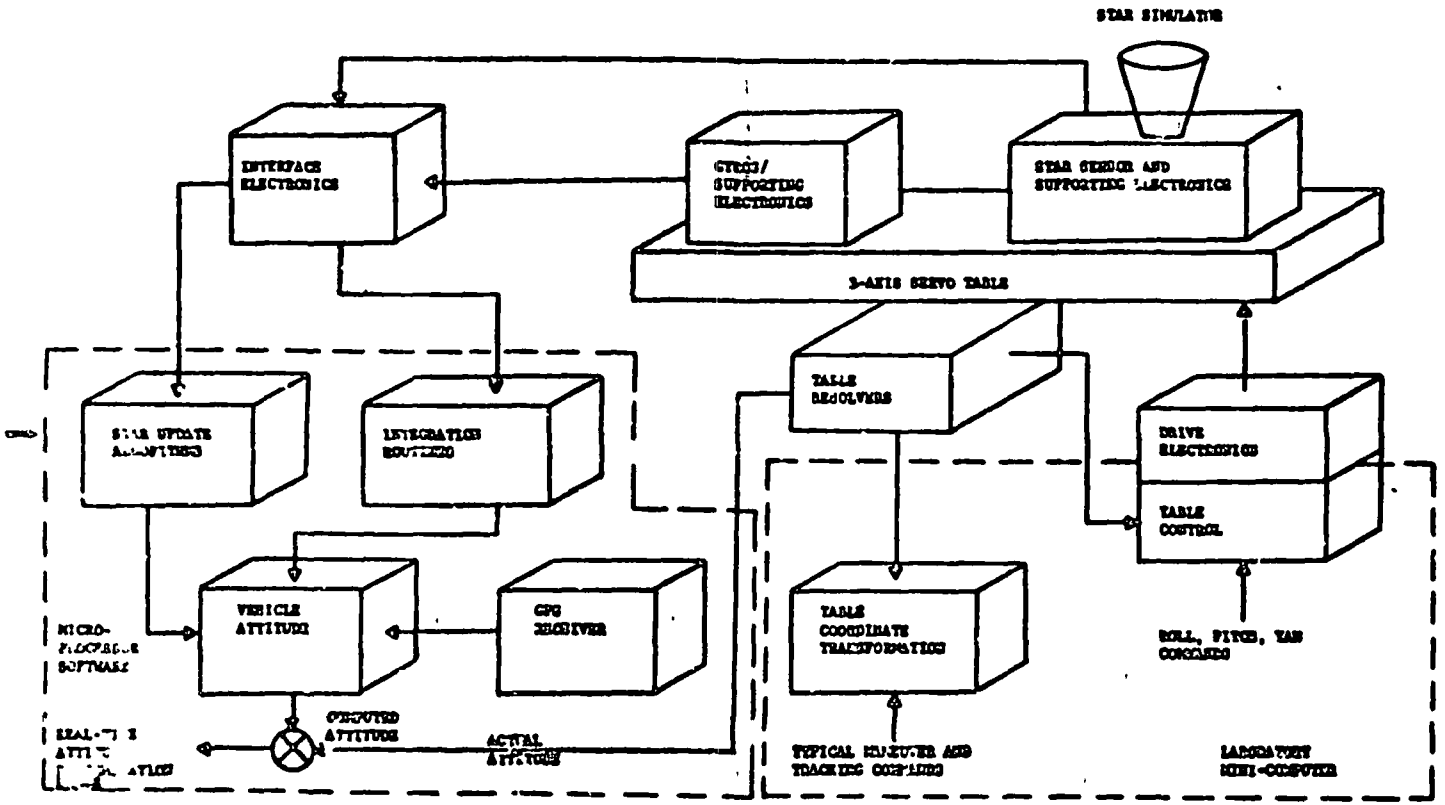


FIGURE 11-1 TESTING PROCEDURE FLOW DIAGRAM

TABLE 11.1 MAJOR ON-BOARD SYSTEM OPERATIONAL UPDATE REQUIREMENTS

FUNCTIONS	TIME INTERVAL BETWEEN UPDATE (sec)
IMU Sample Rate	0.05
Strapdown Integrator Sample Rate	0.05
Star Tracker readout sample rate	0.20
Filter processor rate	0.2
IMU updat · from star trackers	
Earth mission - nadir pointing	1200.0
- 5°/min maneuver	400.0
- 2°/sec maneuver	28.0
Stellar mission	2130.0
Solar mission	2130.0
GPS Orbit information update rate	6.0
Internal orbit propagator update rate	1.0

Because of the on-board autonomous attitude determination and orbit information system concept for the OADS, the ground operational update requirement should be minimized. When a ground update is necessary because of the on-board storage problem or system degeneration, the update interval should be long such that it will not lose the meaning of on-board autonomous systems. The OADS system should have the capability of updating the data base parameters on the ground and uplinking to the spacecraft on-board system processors. There are two major items required for periodic updates from the ground station. One is the update of IMU error parameters and the star tracker boresight axis misalignments. This will be discussed in the next subsection. For the time being, it is felt that those error parameters should be updated depending upon the mission ruggedness (for instance, how frequent the spacecraft is in and out of the earth shadow, how much environmental vibration occurs due to VCS burn, how tight is the onboard system temperature control, etc). This update frequency can be varied from a few days to one month, and further study should be conducted when more mission phase knowledge is available. The other sets of information which require periodic update is the on-board star catalog, if the annual mean motion cannot be included due to the restriction of space. By using +5.0 visual magnitude and brighter stars, for example, if we do not include the mean motion parameters, then the star catalog needs to be updated every six months to keep the star position error less than 3 arc seconds. If the mean motion parameters for each star is included, then this periodic update procedure can be neglected.

11.3 Post Launch Verification

The OADS requires knowledge of the form and magnitude of specific error contributions to provide precise attitude information to the mission phase. The ground preflight testing and calibration of the sensors can provide initial values, but extended operations can result in long-term variations in the parameters. To maintain the precision of the system, some means of post launch verification, such as on-orbit calibration of critical error sources in IMU, star trackers, GPS systems is required. Methods have been developed Headley, 1 to provide the on-orbit calibration of individual gyro scale factor, bias drift, gyro-to-gyro misalignment, and star tracker to IMU misalignments. Specific spacecraft maneuver sequences are required for the collection of most calibration data. Using ground computational facilities as mentioned in 11.2, the data is reduced to provide refined compensation coefficients. The GPS receiver data can be telemetered to the ground station to compare with the tracking data processed by the ground computer. By periodically resolving and updating the error parameters of each OADS components, the OADS accuracy can be maintained throughout the mission period.

12.0 RECOMMENDATIONS FOR FUTURE INVESTIGATIONS

In the course of this study, several system design alternatives presented themselves. We would, therefore, like to describe to GSFE topics which we feel are of special interest but were not included in the Statement of Work. The following paragraphs highlight four areas concerning OADS instrumentation and processing alternatives which should be considered in future investigations.

12.1 Control Law Processing

The proposed OADS concept provides attitude information as a function of time to be derived on-board and made available as downlink data to the payload user. A natural follow-on, or subset, to the proposed OADS concept would be to modify the existing OADS algorithms to generate spacecraft attitude and attitude rate error signals. These signals would be used to drive the associated control laws which would be incorporated in the OADS multiple microcomputers. In the case of the MMS, this would off-load the control law processing burden presently being accomplished in the NSSC-I computer, thus allowing the NSSC-I computer to be used for additional processing and/or switching functions for scientific payloads. To make this realizable on a multiple microcomputer system, flight quality LSI hardware would have to be developed. Since this may take several years, we suggest that GSFC consider the bipolar arithmetic processing unit approach outlined in the following recommendation p ragraphs.

12.2 NSSC-I and Bipolar Arithmetic Processing Unit

Section 11 described several approaches for enhancing the performance of spacecraft systems which use the NSSC-I. One of the approaches outlined is to supplement the NSSC-I with a bipolar arithmetic processing unit. This concept has several advantages, however, the most important feature is that such a high performance system could be developed in a very short period of time.

Also important is the fact that such a configuration could be utilized in a number of different spaceborne processing applications and not just OADS. Since it may be several years before high performance flight qualifiable microprocessor hardware is available, we feel that the development of an NSSC-I/bipolar APU would be the most direct mechanism for improving spaceborne processing systems.

12.3 Replacement of the SST with the CCD Star Tracker

As mentioned in Section 4.2.2.3, the CCD unit currently being developed has certain advantages which make it a prime alternative to the Standard Star Tracker for MMS OADS missions. The advantages of the CCD unit over the SST are:

- The CCD has a self contained heating unit and is insensitive to magnetic field intensity variations. The on-board temperature and magnetic compensation algorithms used with the SST for better accuracy can be completely omitted for the CCD unit. This results in a significant reduction in star tracker update software.
- Use of the star field brightness map in the CCD $8^{\circ} \times 8^{\circ}$ field of view can significantly reduce the star identification software. Furthermore, the probability of false or ambiguous star identification can be reduced and hence, star tracker update accuracy can be improved.
- The CCD's self contained microprocessor can be programmed to simplify the user supplied data editing and synchronization processor.

Thus, the CCD unit, which is considered to be the next generation of star tracker, can not only provide better accuracy but also simplify the OADS software. It has the same dynamic range and field of view as the standard

star tracker and, therefore, can be used in the same manner as the SST in MMS missions.

12.4 Replacement of Star Tracker with Landmark Tracking System

Attitude determination for Earth viewing satellites; e.g., LANDSAT, SEASAT, requires accuracies consistent with the imagery devices (e.g., Multispectral Scanner). Star tracker systems, in general, are not accurate enough due to sensor inaccuracies, misalignments, and the difference between the Earth image sensors and the star system. A possible solution to the above problem is to replace or supplement the star tracker with a landmark tracker as a source of attitude information. Studies [Martin Marietta, 3] have shown from temporal registration of images; i.e., pixel, that attitude accuracy of 7 arc-sec (14 M on Earth surface - 2σ) can be achieved for a LANDSAT type orbit. The star tracker system, at best, can achieve 14.4 arc-sec (30 M on Earth surface - 2σ) accuracy.

Currently, one of the major data processing log jams in the processing of Earth resource-type data is the necessity for calibration, correction and reformating of orbit, attitude and scientific image data by ground facilities before it is available for recognition processing. Pre-processing this data on-board the spacecraft in real time would provide a significant reduction in the cost of processing Earth resource-type data as well as reducing the end-to-end processing time. Once the data is processed on-board, a direct link to the user of attitude, orbit, and image data could be established creating a real-time system.

Areas of study for a Landmark OADS concept are:

- Landmark tracker configuration interfaces with the NSSC-I, microprocessors and a NSSC-I microprocessor hybrid system.

- Landmark/star tracker combination configuration trade study for attitude determination.
- Landmark OADS configuration using the landmark tracker for both attitude and orbit determination.
- Processing algorithms and hardware configuration for the landmark OADS concept.
- A computer simulation study of the landmark OADS concept.

APPENDIX I

MICROPROCESSOR SPECIFICATIONS

DISCLAIMER

The microprocessor description material contained in this appendix has been reproduced from literature supplied by various vendors with their permission. This information is supplied for the sole purpose of giving the reader greater insight into the LSI devices discussed in the text of this report. This information is NOT to be used as a detailed device specification. Detailed device specifications may be obtained by directly contacting the vendors whose addresses are listed at the end of this section.

Am9080A

The 8080 microprocessor first produced by Intel Corporation is now available from a number of different semiconductor vendors. The following data sheet describes the Am9080A microprocessor produced by Advanced Micro Devices. This device is functionally equivalent to the Intel 8080 microprocessor.

Am9080A

8-Bit Microprocessor
Advanced Micro Devices
Advanced MOS/LSI

Distinctive Characteristics

- Plug-in replacements for 8080A, 8080A-1, 8080A-2
- High-speed version with 1μsec instruction cycle
- Military temperature range operation to 1.5μsec
- Ion-implanted, n-channel, silicon gate MOS technology
- 3.2mA of output drive at 0.4V (two full TTL loads)
- 700mV of high, 400mV of low level noise immunity
- 820mW maximum power dissipation at ±5% power
- 100% reliability assurance testing to MIL STD-883

GENERAL DESCRIPTION

The Am9080A products are complete, general-purpose, single-chip digital processors. They are fixed instruction set, parallel, 8 bit units fabricated with Advanced N-Channel Silicon Gate MOS technology When combined with external memory and peripheral devices, powerful microcomputer systems are formed. The Am9080A may be used to perform a wide variety of operations, ranging from complex arithmetic calculations to character handling to bit control. Several versions are available offering a range of performance options

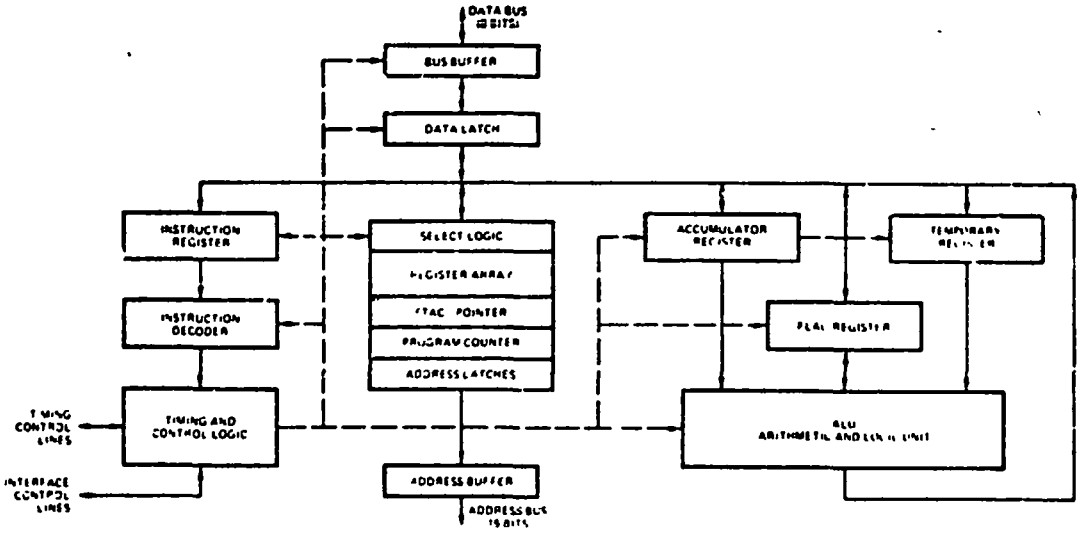
The processor has a 16 bit address bus that may be used to directly address up to 64K bytes of memory The memory may be any combination of read/write and read only Data are transferred into or out of the processor on a bi-directional 8-bit data bus that is separate from the address lines. The data bus transfers instructions, data and status information between system devices All transfers are handled using asynchronous handshaking controls so that any speed memory or I/O device are easily accommodated.

An accumulator plus six general purpose registers are available to the programmer. The six registers are each 8 bits long and may be used singly or in pairs for both 8 and 16 bit operations. The accumulator forms the primary working register and is the destination for many of the arithmetic and logic operations.

A general purpose push-down stack is an important part of the processor architecture The contents of the stack reside in R/W memory and the control logic, including a 16 bit stack pointer, is located on the processor chip. Subroutine call and return instructions automatically use the stack to store and retrieve the contents of the program counter Push and Pop instructions allow direct use of the stack for storing operands, passing parameters and saving the machine state

An asynchronous vectored interrupt capability is included to allow external signals to modify the instruction stream. The interrupting device may specify an interrupt instruction to be executed and may thus vector the program to a particular service location, or perform some other direct function Direct memory access (DMA) capability is also included.

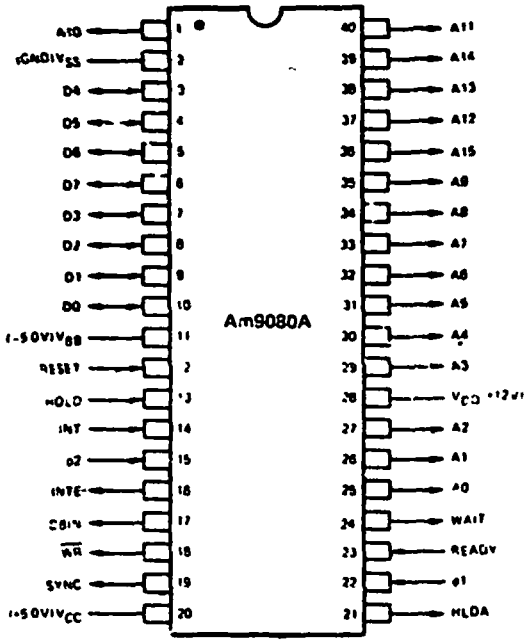
BLOCK DIAGRAM



ORDERING INFORMATION

Package Type	Ambient Temperature Specification	Clock Period 250ns	320ns	380ns	480ns
Hermetic DIP	0°C ≤ T_A ≤ +70°C	AM9080A-4DC	AM9080A 1DC C8080A 1	AM9080A 2DC C8080A 2	AM9080ADC C8080A
	−55°C ≤ T_A ≤ +125°C			AM9080A 2DM	AM9080ADM
Molded DIP	0°C ≤ T_A ≤ +70°C		AM9080A 1PC	AM9080A 2PC	AM9080APC

CONNECTION DIAGRAM
Top View



Note Pin 1 is marked for orientation

INTERFACE SIGNAL SUMMARY

TYPE	PINS	ABBREVIATION	SIGNAL
INPUT	1	V_{SS}	Ground
INPUT	3	V_{DD}, V_{CC}, V_{BB}	+12V, +5V, −5V Supplies
INPUT	2	Φ_1, Φ_2	Clocks
INPUT	1	RESET	Reset
INPUT	1	HOLD	Hold
INPUT	1	INT	Interrupt
INPUT	1	READY	Ready
IN/OUT	8	D_0–D_7	Data Bus
OUTPUT	16	A_0–A_{15}	Address
OUTPUT	1	INTE	Interrupt Enable
OUTPUT	1	DBIN	Data Bus In Control
OUTPUT	1	$\overline{WR}$	Write Not
OUTPUT	1	SYNC	Cycle Synchronization
OUTPUT	1	HLDA	Hold Acknowledge
OUTPUT	1	WAIT	Wait

INTERFACE SIGNAL DESCRIPTION

Φ1, Φ2 The Clock inputs provide basic timing generation for all internal operations They are non overlapping two phase, high level signals All other inputs to the processor are TTL compatible

RESET The Reset input initializes the processor by clearing the program counter, the instruction register the interrupt enable flip-flop and the hold acknowledge flip-flop. The Reset signal should be active for at least three clock periods The general registers are not cleared

HOLD The Hold input allows an external signal to cause the processor to relinquish control over the address lines and the data bus When Hold goes active the processor completes its current operation, activates the HLDA output, and puts the 3 state address and data lines into their high impedance state The Holding device can then utilize the address and data busses without interference

READY The Ready input synchronizes the processor with

INT The Interrupt input signal provides a mechanism by

handled efficiently with the vectored interrupt procedure and the general purpose stack Interrupt processing is described in more detail on the next page

D_0-D_7 The Data Bus is comprised of 8 bidirectional signal lines for transferring data, instructions and status information between the processor and all external units

A_0-A_{15} The Address Bus is comprised of 16 output signal lines used to address memory and peripheral devices

SYNC The Sync output indicates the start of each processor cycle and the presence of processor status information on the data bus

DBIN The Data Bus In output signal indicates that the bidirectional data bus is in the input mode and incoming data may be gated onto the Data Bus

WAIT The Wait output indicates that the processor has entered the Wait state and is prepared to accept a Ready from the current external operation

$\overline{WR}$ The Write output indicates the validity of output on

HLDA The Hold Acknowledge output signal is in response to a Hold input It indicates that processor activity has been suspended and the Address and Data B

INTE The Interrupt Enable output signal shows the status of the interrupt enable flip-flop and indicates whether or not the processor will accept interrupts

INSTRUCTION SET INTRODUCTION

The instructions executed by the Am9080A are variable length and may be one, two or three bytes long. The length is determined by the nature of the operation being performed and the addressing mode being used

The instruction summary shows the number of successive memory bytes occupied by each instruction, the number of clock cycles required for the execution of the instruction, the binary coding of the first byte of each instruction, the mnemonic coding used by assemblers and a brief description of each operation Some branch type instructions have two execution times depending on whether the conditional branch is taken or not Some fields in the binary code are labeled with alphabetic abbreviations That shown as vvv is the address pointer used in the one byte Call instruction (RST) Those shown as ddd or sss designate destination and source register fields that may be filled as follows

111 A register
000 B register
001 C register
010 D register
011 E register
100 H register
101 L register
110 Memory

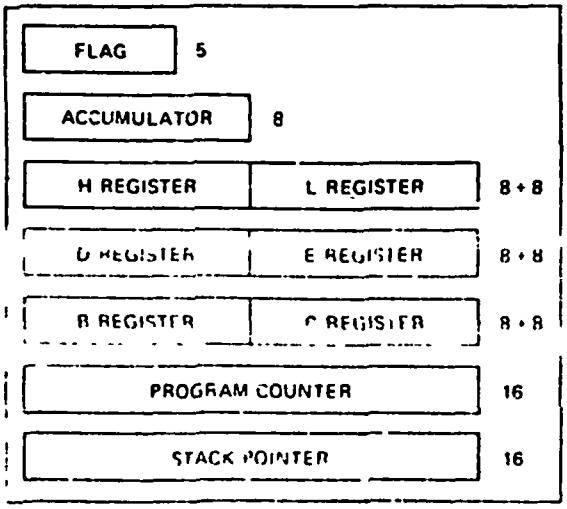
The register diagram shows the internal registers that are directly available to the programmer. The accumulator is the primary working register for the processor and is a specified or implied operand in many instructions All I/O operations take place via the accumulator Registers H, L, D, E, B and C may be used singly or in the indicated pairs The H and L pair is the implied address pointer for many instructions

The Flag register stores the program status bits used by the conditional branch instructions carry, zero, sign and parity The fifth flag bit is the intermediate carry bit The flags and the accumulator can be stored on or retrieved from the stack with a single instruction. Bit positions in the flag register when pushed onto the stack (PUSH PSW) are

7	6	5	4	3	2	1	0
S	Z	0	CY1	0	P	1	CY2

where S = sign, Z = zero, CY1 = intermediate carry, P = parity, CY2 = carry

REGISTER DIAGRAM



During Sync time at the beginning of each instruction cycle the data bus contains operation status information that describes the machine cycle being executed. Positions for the status bits are.

7	6	5	4	3	2	1	0
MEMR	INP	M1	OUT	HLTA	STK	$\overline{WO}$	INTA

STATUS DEFINITION:

INTA Interrupt Acknowledge. Occurs in response to an Interrupt input and indicates that the processor will be ready for an interrupt instruction on the data bus when DBIN goes true

$\overline{WO}$ Write or Output indicated when signal is low When high, a Read or Input will occur

STK Stack indicates that the content of the stack pointer is on the address bus.

HLTA Halt Acknowledge

OUT Output instruction is being executed

M1 First instruction byte is being fetched

INP Input instruction is being executed

MEMR Memory Read operation.

INTERRUPT PROCESSING

When the processor interrupt mechanism is enabled (INTE = 1), interrupt signals from external devices will be recognized unless the processor is in the Hold State In handling an interrupt, the processor will complete the execution of the current instruction, disable further interrupts and respond with INTA status instead of executing the next sequential instruction in the interrupted program.

The interrupting device should supply an instruction opcode to the processor during the next DBIN time after INTA status appears.

Any opcode may be used except XTHL If the instruction supplied is a single byte instruction, it will be executed (The usual single byte instruction utilized is RST) If the interrupt instruction is two or three bytes long, the next one or two processor cycles, as indicated by the DBIN signal, should be used by the external device to supply the succeeding byte(s) of the interrupt instruction. Note that INTA status from the processor is not present during these operations

If the interrupt instruction is not some form of CALL, it is executed normally by the processor except that the Program Counter is not incremented The next instruction in the interrupted program is then fetched and executed Notice that the interrupt mechanism must be re-enabled by the processor before another interrupt can occur

If the interrupt instruction is some form of CALL, it is executed normally The Program Counter is stored and control transferred to the interrupt service subroutin This routine has responsibility for saving and restoring the machine state and for re-enabling interrupts if desired When the interrupt service is complete a RETURN instruction will transfer control back to the interrupted program

INSTRUCTION SET SUMMARY

Op Code [7][6][5][4][3][2][1][0]	No. of Bytes	Clock Cycles	Assembly Mnemonic	Instruction Description
DATA TRANSFER				
01dddsss	1	5	MOVr r	Move register to register
01110sss	1	7	MOVm, r	Move register to memory
01ddd110	1	7	MOVr, m	Move memory to register
00ddd110	2	7	MVI, r	Move to register, immediate
00110110	2	10	MVI m	Move to memory immediate
00111010	3	13	LDA	Load Acc direct
00001010	1	7	LDAX B	Load Acc indirect via B & C
00011010	1	7	LDAX D	Load Acc, indirect via D & E
00101010	3	16	LHLD	Load H & L direct
00100001	3	10	LXI H	Load H & L, immediate
00010001	3	10	LXI D	Load D & E immediate
00000001	3	10	LXI B	Load B & C immediate
00110001	3	10	LXI SP	Load stack pointer immediate
00100010	3	16	SHLD	Store H & L direct
00110010	3	13	STA	Store Acc direct
00000010	1	7	STAX B	Store Acc indirect via B & C
00010010	1	7	STAX D	Store Acc indirect via D & E
11111001	1	5	SPHL	Transfer H & L to stack pointer
11101011	1	4	XCHG	Exchange D & E with H & L
11100011	1	18	XTHL	Exchange top of stack with H & L
11011011	2	10	IN	Input to Acc
11010011	2	10	OUT	Output from Acc
CONTROL				
01110110	1	7	HLT	Halt and enter wait state
00110111	1	4	STC	Set carry flag
00111111	1	4	CMC	Compliment carry flag
11111011	1	4	EI	Enable interrupts
11110011	1	4	DI	Disable interrupts
00000000	1	4	NOP	No operation
BRANCHING				
11000011	3	10	JMP	Jump unconditionally
11011010	3	10	JC	Jump on carry
11010010	3	10	JNC	Jump on no carry
11001010	3	10	JZ	Jump on zero
11000010	3	10	JNZ	Jump on not zero
11110010	3	10	JP	Jump on positive
11111010	3	10	JM	Jump on minus
11101010	3	10	JPE	Jump on parity even
11100010	3	10	JPO	Jump on parity odd
11001101	3	17	Call	Call unconditionally
11011100	3	17/11	CC	Call on carry
11010100	3	17/11	CNC	Call on no carry
11001100	3	17/11	CZ	Call on zero
11000100	3	17/11	CNZ	Call on not zero
11110100	3	17/11	CP	Call on positive
11111100	3	17/11	CM	Call on minus
11101100	3	17/11	CPE	Call on parity even
11100100	3	17/11	CPO	Call on parity odd
11001001	1	10	RET	Return unconditionally
11011000	1	11/5	RC	Return on carry
11010000	1	11/5	RNC	Return on no carry
11001000	1	11/5	RZ	Return on zero
11000000	1	11/5	RNZ	Return on not zero
	1	11/5	RM	Return on minus
	1	11/5	RPE	Return on parity even
	1	11/5	RPO	Return on parity odd
		5	PCHL	Jump to address in H & L
				register

Op Code [7][6][5][4][3][2][1][0]	No. of Bytes	Clock Cycles	Assembly Mnemonic	Instruction Description
ARITHMETIC				
10000sss	1	4	ADDr	Add register to Acc
10001sss	1	4	ADCr	Add with carry register to Acc
10000110	1	7	ADDm	Add memory to Acc
10001110	1	7	ADCm	Add with carry memory to Acc
11000110	2	7	ADI	Add to Acc immediate
11001110	2	7	ACI	Add with carry to Acc, immediate
00001001	1	10	DAD B	Double add B & C to H & L
00011001	1	10	DAD D	Double add D & E to H & L
00101001	1	10	DAD H	Double add H & L to H & L
00111001	1	10	DAD SP	Double add stack pointer to H & L
10010sss	1	4	SUBr	Subtract register from Acc
10011sss	1	4	SBBr	Subtract with borrow register from Acc
10010110	1	7	SUBm	Subtract memory from Acc
10011110	1	7	SBBm	Subtract with borrow memory from Acc
11010110	2	7	SUI	Subtract from Acc immediate
11011110	2	7	SBI	Subtract with borrow from Acc immediate
00100111	1	4	DAA	Decimal adjust Acc
STACK OPERATIONS				
11000101	1	11	PUSH B	Push registers B & C on stack
11010101	1	11	PUSH D	Push registers D & E on stack
11100101	1	11	PUSH H	Push registers H & L on stack
11110101	1	11	PUSH PSW	Push Acc and flags on stack
11000001	1	10	POP B	Pop registers B & C off stack
11010001	1	10	POP D	Pop registers D & E off stack
11100001	1	10	POP H	Pop registers H & L off stack
11110001	1	10	POP PSW	Pop Acc and flags off stack
LOGICAL				
10100sss	1	4	ANA r	And register with Acc
10100110	1	7	ANA m	And memory with Acc
11100110	2	7	ANI	And with Acc immediate
10101sss	1	4	XRA r	Exclusive or register with Acc
10101110	1	7	XRA m	Exclusive Or memory with Acc
11101110	2	7	XRI	Exclusive Or with Acc immediate
10110sss	1	4	ORA r	Inclusive Or register with Acc
10110110	1	7	ORA m	Inclusive Or memory with Acc
11110110	2	7	ORI	Inclusive Or with Acc immediate
10111sss	1	4	CMP r	Compare register with Acc
10111110	1	7	CMP m	Compare memory with Acc
11111110	2	7	CPI	Compare with Acc immediate
00101111	1	4	CMA	Compliment Acc
00000111	1	4	RLC	Rotate Acc left
00001111	1	4	RRC	Rotate Acc right
00010111	1	4	RAL	Rotate Acc left through carry
00011111	1	4	RAR	Rotate Acc right through carry
INCREMENT/DECREMENT				
00ddd100	1	5	INR r	Increment register
00110100	1	10	INR m	Increment memory
00000011	1	5	INX B	Increment extended B & C
00010011	1	5	INX D	Increment extended D & E
00100011	1	5	INX H	Increment extended H & L
	1	5		Increment stack pointer
00ddd101	1	5	DCR r	Decrement register
00110101	1	10	DCR m	Decrement memory
00001011	1	5	DCX B	Decrement extended B & C
00011011	1	5	DCX D	Decrement extended D & E
	1	5		Decrement extended H & L
	1	5		Decrement stack pointer

MAXIMUM RATINGS (Above which useful life may be impaired)

Storage Temperature	−65°C to +150°
Ambient Temperature Under Bias	−55°C to +125°
All Signal Voltages With Respect to V_{BB}	−0.3V to +20
All Supply Voltages With Respect to V_{BB}	−0.3V to +20
Power Dissipation	1 5\

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations o static charge It is suggested nevertheless, that conventional precautions be observed during storage, handling and use in order to avou exposure to excessive voltages.

OPERATING RANGE

Part Number	T_A	V_{DD}	V_{CC}	V_{BB}	V_{SS}
An 9080A XDC C8080A X	0°C to +70°C	+12V ±5%	+5 0V 5%	−5 0V ±5%	0V
Am9080A XDM	−55°C to +125°C	+12V ·10%	+5 0V ·10%	−5 0V ±10%	0V

No signal or supply vo'tage should ever be greater than 0 3V more negative than V_{BB}.

ELECTRICAL CHARACTERISTICS over operating range (note 1)

Parameters	Description	Test Conditions			C8080A-X Min.	C8080A-X Typ.	C8080A-X Max.	Am9080A-XDC Min.	Am9080A-XDC Typ.	Am9080A-XDC Max.	Am9080A-XDM Min	Am9080A-XDM Typ	Am9080A-XDM Max	Units
V_{IL}	Input LOW Voltage				−1 0		0 8	−1 0		0 8	1 0		0 8	Volts
V_{IH}	Input HIGH Voltage				3 3		V_{CC} +1	3 0		V_{CC} +1	3 0		V_{CC} +1	Volts
V_{ILC}	Input LOW Voltage Clock				−1 0		0 8	−1 0		0 8	1 0		0 8	Volts
V_{IHC}	Input HIGH Voltage, Clock		Am9080A 4		9 0		V_{DD} +1	9 0		V_{DD} +1	V_{DD} 2		V_{DD} +1	Volts
								V_{DD} 2		V_{DD} +1				
V_{OL}	Output LOW Voltage	I_{OL} · 3 2mA								0 40				Volts
		I_{OL} · 1 9mA					0.45						0 40	Volts
V_{OH}	Output HIGH Voltage	I_{OH} · 200µA						3 7			3 7			Volts
		I_{OH} · −100µA			3 7									Volts
$I_{DD(AV)}$	V_{DD} Supply Current Average	Operating Minimum Clock Period	Am9080A Am9080A 2 Am9080A 1	T_A · +25 C		40			30	45		30	50	mA
				T_A · 0 C			70		35	50		3	55	
				T_A · −55 C								45	70	
			Am9080A 4	T_A · +25 C					45	60				
				T_A · 0 C					55	70				
$I_{CC(AV)}$	V_{CC} Supply Current Average	Operating Minimum Clock Period	Am9080A Am9080A 2 Am9080A 1	T_A · +25 C		60			25	30		15	35	mA
				T_A · 0 C			80		20	35		20	40	
				T_A · −55 C								25	50	
			Am9080A 4	T_A · +25 C					35	50				
				T_A · 0 C					40	60				
$I_{BB(AV)}$	V_{BB} Supply Current Average	Operating Minimum Clock Period					1 0			1 0			1 0	mA
I_{IL}	Input Leakage Current	(Note 4)					10			10			10	µA
I_{CL}	Clock Leakage Current	V_{SS} · V_{IN} · V_{DD}					10			·10			10	µA
I_{DL}	Data Bus Current Input Mode (Note 2)	V_{IN} · V_{SS} + 0 8V					−100			100			1001	µA
		V_{IN} · V_{SS} + 0 8V					−2 0			2 0			20	mA
I_{FL}	Address and Data Bus Leakage in OFF State	$V_{A,D}$ V_{CC}					10			10			10	µA
		$V_{A,D}$ V_{SS}					100			100			100	µA

CAPACITANCE

[illegible] 0V T_A 25 C

[illegible] 50

Parameters	Description	Typ	Max	Units
C	Clock Input Capacitance	12	20	pF
C_I	Input Capacitance	4 0	8 0	pF
C_O	Output Capacitance	8 0	15	pF
$C_{I/O}$	I/O Capacitance	10	[illegible]	[illegible]

[illegible]

SWITCHING CHARACTERISTICS over operating range

Boldface numbers are 8080A specs which are exceeded.

Parameters	Description	Test Conditions	Am9080A-4 Min.	Am9080A-4 Max.	Am9080A-1 / C8080A-1 Min.	Am9080A-1 / C8080A-1 Max.	Am9080A-2 / C8080A-2 Min.	Am9080A-2 / C8080A-2 Max.	Am9080A / C8080A Min.	Am9080A / C8080A Max.	Units
t_{DA}	Clock $\phi 2$ to Address Out Delay	Load Capacitance = 100pF		125		150		175		200	ns
t_{DD}	Clock $\phi 2$ to Data Out Delay			140		160		200		220	ns
t_{DI}	Clock $\phi 2$ to Data Bus Input Mode Delay	(Note 5)		t_{DF}		t_{DF}		t_{DF}		t_{DF}	ns
t_{DS1}	Data In to Clock $\phi 1$ Set-up Time	Both t_{DS1} and t_{DS2} must be satisfied	10		10		20		30		ns
t_{DS2}	Data In to Clock $\phi 2$ Set up Time		110		120		130		150		ns
t_{DC}	Clock to Control Output Delay	Load Capacitance = 50pF		100		110		120		120	ns
t_{RS}	Ready to Clock $\phi 2$ Set up Time		80		90		90		120		ns
t_H	Clock $\phi 2$ to Control Signal Hold Time		0		0		0		0		ns
t_{IS}	Interrupt to Clock $\phi 2$ Set-up Time		90		100		100		120		ns
t_{HS}	Hold to Clock $\phi 2$ Set-up Time		100		120		120		140		ns
t_{IE}	Clock $\phi 2$ to INTE Delay	Load Capacitance = 50pF		100		200		200		200	ns
t_{FD}	Clock $\phi 2$ to Address/Data OFF Delay		100			120		120		120	ns
t_{DF}	Clock $\phi 2$ to DBIN Delay	Load Capacitance = 50pF	25	110	25	130	25	140	25	140	ns
t_{DH}	Clock $\phi 2$ to Data In Hold Time	(Note 5)	–	–	–	–	–	–	–	–	ns
t_{AW}	Address Valid to Write Delay	(Note 8)	–	–	–	–	–	–	–	–	ns
t_{DW}	Output Data Valid to Write Delay	(Note 8)	–	–	–	–	–	–	–	–	ns
t_{KA}	Address Valid to Write Increment	(Note 8)		90		110		130		140	ns
t_{KD}	Output Data Valid to Write Increment	(Note 8)		130		150		170		170	ns
t_{WA}	Write to Address Invalid Delay	(Note 8)	–	–	–	–	–	–	–	–	ns
t_{WD}	Write to Output Data Invalid Delay	(Note 8)	–	–	–	–	–	–	–	–	ns
t_{HF}	HLDA to Address/Data OFF Delay	(Note 9)	–	–	–	–	–	–	–	–	ns
t_{WF}	Write to Address/Data OFF Delay	(Note 9)	–	–	–	–	–	–	–	–	ns
t_{KH}	HLDA to Address/Data OFF Increment	(Note 9)		40		50		50		50	ns
t_{AH}	DBIN to Address Hold Time		0		-20		-20		-20		ns

Notes 1 Typical values are at $T_A = 25^\circ C$ nominal supply voltages and nominal processing parameters.

2 Pull up devices are connected to the Data Bus lines when the input signal is high during DBIN time. When switching the input from HIGH to LOW a transient current must be absorbed by the driving device until the input reaches a LOW level.

3 Timing reference levels –
- Clocks HIGH = 8.0V LOW = 1.0V
- Inputs HIGH = 3.3V LOW = 0.8V
- Outputs HIGH = 2.0V LOW = 0.8V

4. Control inputs impress currents on the driving signal during HIGH to LOW transitions. Values shown are for logic high or logic low levels. Peak current during transition is as much as 1.0mA.

5 Bus contention cannot occur and data hold times are adequate when DBIN is used to enable Data In. t_{DH} is the smaller of 50ns or t_{DF}.

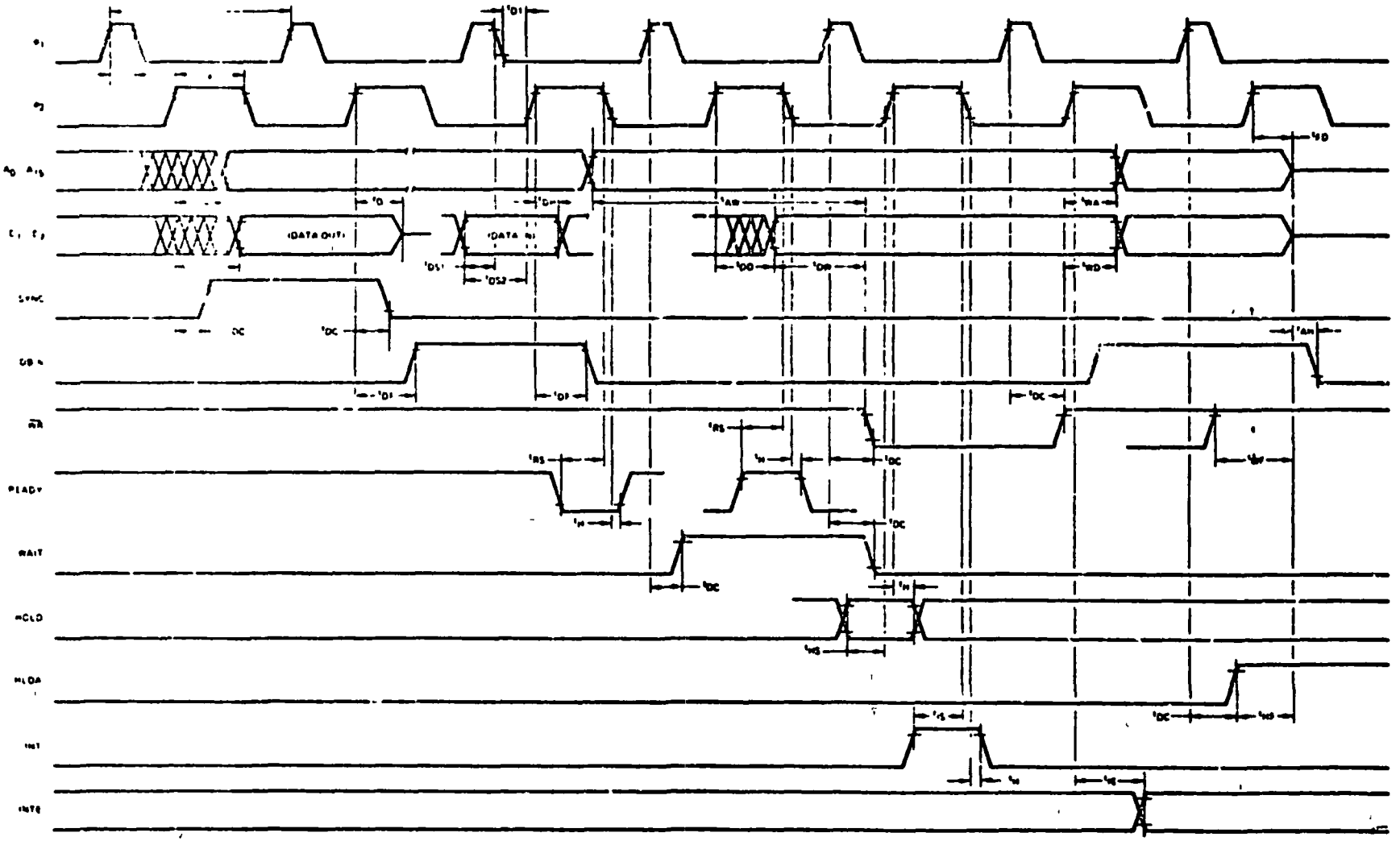
6 RESET should remain active for at least three clock periods.

7 With interrupts enabled, the interrupted instruction will be one with an interrupt input stable during the indicated interval of the last clock period of the preceeding instruction. Additional synchronization not necessary.

8 $t_{AW} = 2\,t_{CY} - t_{D3} - t_{r\phi 2} - t_{KA}$
$t_{DW} = t_{CY} - t_{D3} - t_{r\phi 2} - t_{KD}$
For HLDA Off $t_{WD} = t_{WA} = t_{D3} + t_{r\phi 2} + 10ns$
For HLDA On $t_{WD} = t_{WA} = t_{WF}$

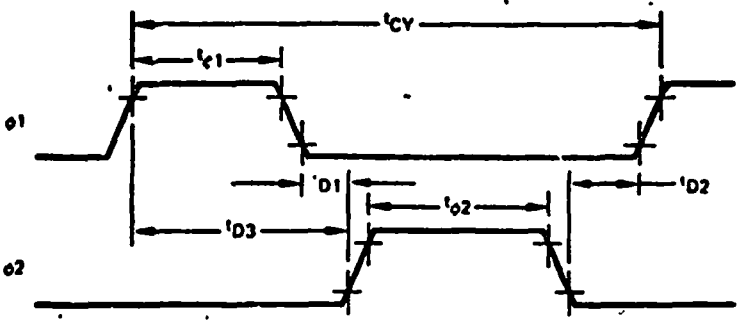
9 $t_{HF} = t_{D3} + t_{r\phi 2} - t_{KH}$
$t_{WF} = t_{D3} + t_{r\phi 2} - 10ns$

SWITCHING WAVEFORMS SUMMARY



This chart presents relative timing waveform relationships and does not show actual processor operating cycles.

CLOCK WAVEFORM DETAIL

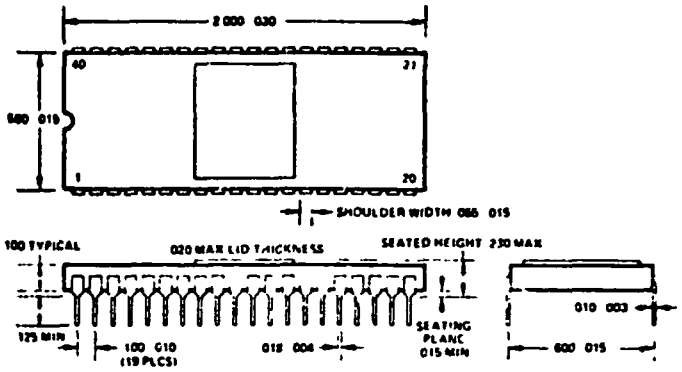


$t_{CY} = t_{D3} + t_{r\phi2} + t_{\phi2} + t_{f\phi2} + t_{D2} + t_{r\phi1}$

CLOCK SWITCHING CHARACTERISTICS over operating range

Parameters	Description	Am9080A-4 Min.	Am9080A-4 Max.	Am9080A-1 C8080A-1 Min	Am9080A-1 C8080A-1 Max.	Am9080A-2 C8080A-2 Min.	Am9080A-2 C8080A-2 Max.	Am9080A C8080A Min.	Am9080A C8080A Max.	Units
t_{CY}	Clock Period	250	2000	320	2000	380	2000	480	2000	ns
t_r, t_f	Clock Transition Times	0	15	0	25	0	50	0	50	ns
$t_{\phi1}$	Clock φ1 Pulse Width	50		50		60		60		ns
$t_{\phi2}$	Clock φ2 Pulse Width	120		145		175		220		ns
t_{D1}	φ1 to φ2 Offset	0		0		0		0		ns
t_{D2}	φ2 to φ1 Offset	50		60		70		70		ns
t_{D3}	φ1 to φ2 Delay	50		60		70		80		ns

PHYSICAL DIMENSIONS
40-Pin Hermetic DIP



Metallization and Pad Layout

Pin 1 identification is substrate
DIE SIZE 0.131 X 0.169

ADVANCED MICRO DEVICES INC
901 Thompson Place
Sunnyvale
California 9408
(408) 732 2400
TWX 910 339 9280
TELEX 34-6306

Am9511

The following data sheet describes the Am9511 arithmatic processing unit produced by Advanced Micro Devices. The vendor is presently sampling the unit and is expected to be in full production later this year.

Am9511

Arithmetic Processing Unit
Advanced Micro Devices
Advanced MOS/LSI

PRELIMINARY INFORMATION

DISTINCTIVE CHARACTERISTICS

- Fixed point single and double precision (16/32 bit)
- Floating point single precision (32 bit)
- Binary data formats
- Add, Subtract, Multiply and Divide
- Trigonometric and inverse trigonometric functions
- Square roots, logarithms, exponentiation
- Float to fixed and fixed to float conversions
- Stack oriented operand storage
- Direct memory access or programmed I/O data transfers
- End of execution signal
- General purpose 8 bit data bus interface
- Standard 24 pin package
- +12 volt and +5 volt power supplies
- Advanced N-channel silicon gate MOS technology
- 100% MIL STD 883 reliability assurance testing

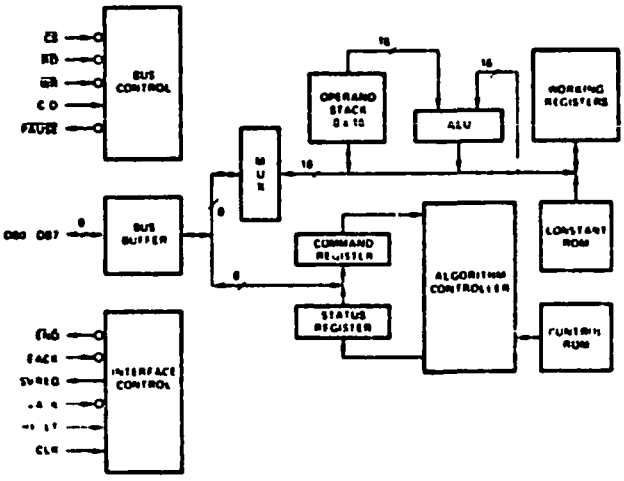
GENERAL DESCRIPTION

The Am9511 Arithmetic Processing Unit (APU) is a monolithic MOS LSI device that provides high performance fixed and floating point arithmetic and floating point trigonometric operations. It may be used to enhance the mathematical capability of a wide variety of processor-oriented systems. Chebyshev polynomials are used in the implementation of the APU algorithms

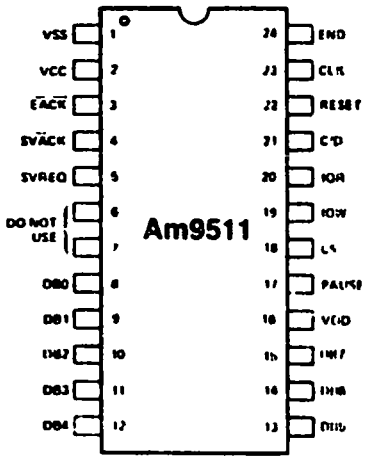
All transfers, including operand, result, status and command information, take place over an 8-bit bidirectional data bus Operands are pushed onto an internal stack and commands are issued to perform operations on the data in the stack Results are then available to be retrieved from the stack

Transfers to and from the APU may be handled by the associated processor using conventional programmed I/O, or may be handled by a direct memory access controller for improved performance Upon completion of each command, the APU issues an end of execution signal that may be used as an interrupt by the CPU to help coordinate program execution

BLOCK DIAGRAM



CONNECTION DIAGRAM



Top View

Pin 1 is marked for orientation

ORDERING INFORMATION

Package Type	Ambient Temperature	Clock Speed 2mHz	Clock Speed 4mHz
Hermetic DIP	0°C ≤ T_A ≤ +70°C	Am9511DC	Am9511 DC
	-55°C ≤ T_A ≤ +125°C	Am9511DM	

INTERFACE SIGNAL DESCRIPTION

VCC: +5 Volt power supply

VDD: +12 Volt power supply

VSS: Ground

CLK (Clock, Input)

An external, TTL compatible, timing source is applied to the CLK pin.

RESET (Reset, Input)

The active high reset signal provides initialization for the chip. RESET also terminates any operation in progress. RESET clears the status register and places the Am9511 into the idle state. Stack contents and command registers are not affected.

$\overline{CS}$ (Chip Select, Input)

$\overline{CS}$ is an active low input signal which selects the Am9511 and enables communication with the data bus.

C/$\overline{D}$ (Command/Data, Input)

In conjunction with the $\overline{RD}$ and $\overline{WR}$ signals, the C/$\overline{D}$ control line establishes the type of communication that is to be performed with the Am9511 as shown below:

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	Function
0	1	0	Enter data byte into stack
0	0	1	Read data byte from stack
1	1	0	Enter command
1	0	1	Read status

$\overline{RD}$ (Read, Input)

This active low input indicates that data or status is to be read from the Am9511 if $\overline{CS}$ is low.

$\overline{WR}$ (Write, Input)

This active low input indicates that data or a command is to be written into the Am9511 if $\overline{CS}$ is low.

$\overline{EACK}$ (End Acknowledge, Input)

This active low input clears the end of execution output signal ($\overline{END}$). If $\overline{EACK}$ is tied low, the $\overline{END}$ output will be a pulse that is one clock wide.

$\overline{SVACK}$ (Service Acknowledge, Input)

This active low input clears the service request output (SVREQ).

$\overline{END}$ (End Execution, Output)

This active low, open-drain output indicates that execution of the previously entered command is complete. It can be used as an interrupt request and is cleared by $\overline{EACK}$, RESET or any read or write access to the Am9511.

SVREQ (Service Request, Output)

This active high output signal indicates that command execution is complete and that post execution service was requested in the previous command byte. It is cleared by $\overline{SVACK}$, the next command output to the device, or by RESET.

$\overline{PAUSE}$ (Pause, Output)

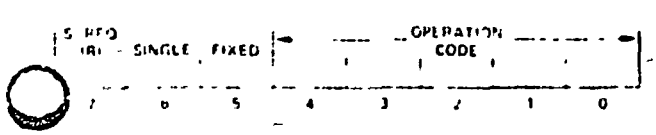
This active low output indicates that the Am9511 is unable to accept communication with the data bus. When an attempt is made to read data, write data or to enter a new command while the Am9511 is executing a command, $\overline{PAUSE}$ goes low until execution of the current command is complete. (See Pause Operation, p. 5)

D80-D87 (Bidirectional Data Bus, I/O)

These eight bidirectional lines provide for transfer of commands, status and data between the Am9511 and the CPU. The Am9511 can drive the data bus only when $\overline{CS}$ and $\overline{RD}$ are low.

COMMAND STRUCTURE

Each command entered into the Am9511 consists of a single 8 bit byte having the format illustrated below:



bits 0-4 select the operation to be performed as shown in the [illegible]. Bits 5-6 select the data format appropriate to the selected operation. If bit 5 is a 1, a fixed point data format is specified. If bit 5 is a 0, floating point format is specified. Bit 6 selects the precision of the data to be operated upon by fixed point commands only (if bit 5 = 0, bit 6 must be 0). If bit 6 is a 1 single precision (16 bit) operands are indicated. If bit 6 is a 0 double precision (32 bit) operands are indicated. Results are undefined for all illegal combinations of bits in the command byte. Bit 7 indicates whether a service request is to be issued after the command is executed. If bit 7 is a 1 the service request output (SVREQ) will go high at the conclusion of the command and will remain high until reset by a low level on the service acknowledge pin ($\overline{SVACK}$) or until completion of execution of the succeeding command where service request (bit 7) is 0. Each command issued to the Am9511 requests post execution service based upon the state of bit 7 in the command byte. When bit 7 is a 0, SVREQ remains low.

COMMAND SUMMARY

Command Code								Command Mnemonic	Command Description (1)
7	6	5	4	3	2	1	0		
FIXED POINT SINGLE PRECISION									
R	1	1	0	1	1	0	0	SADD	Adds TOS to NOS Result to NOS Pop Stack
R	1	1	0	1	1	0	1	SSUB	Subtracts TOS from NOS Result to NOS Pop Stack
R	1	1	0	1	1	1	0	SMUL	Multiplies NOS by TOS Result to NOS Pop Stack
R	1	1	0	1	1	1	1	SDIV	Divides NOS by TOS Result to NOS Pop Stack
FIXED POINT DOUBLE PRECISION									
R	0	1	0	1	1	0	0	DADD	Adds TOS to NOS Result to NOS Pop Stack
R	0	1	0	1	1	0	1	DSUB	Subtracts TOS from NOS Result to NOS Pop Stack
R	0	1	0	1	1	1	0	DMUL	Multiplies NOS by TOS Result to NOS Pop Stack
R	0	1	0	1	1	1	1	DDIV	Divides NOS by TOS Result to NOS Pop Stack
FLOATING POINT									
R	0	0	1	0	0	0	0	FADD	Adds TOS to NOS Result to NOS Pop Stack
R	0	0	1	0	0	0	1	FSUB	Subtracts TOS from NOS Result to NOS Pop Stack
R	0	0	1	0	0	1	0	FMUL	Multiplies NOS by TOS Result to NOS Pop Stack
R	0	0	1	0	0	1	1	FDIV	Divides NOS by TOS Result to NOS Pop Stack
DERIVED FLOATING POINT FUNCTIONS (2)									
R	0	0	0	0	0	0	1	SQRT	Square Root of TOS Result in TOS
R	0	0	0	0	0	1	0	SIN	Sine of TOS Result in TOS
R	0	0	0	0	0	1	1	COS	Cosine of TOS Result in TOS
R	0	0	0	0	1	0	0	TAN	Tangent of TOS Result in TOS
R	0	0	0	0	1	0	1	ASIN	Inverse Sine of TOS Result in TOS
R	0	0	0	0	1	1	0	ACOS	Inverse Cosine of TOS Result in TOS
R	0	0	0	0	1	1	1	ATAN	Inverse Tangent of TOS Result in TOS
R	0	0	0	1	0	0	0	LOG	Common Logarithm (base 10) of TOS Result in TOS
R	0	0	0	1	0	0	1	LN	Natural Logarithm (base e) of TOS Result in TOS
R	0	0	0	1	0	1	0	EXP	Exponential (e^x) of TOS Result in TOS
R	0	0	0	1	0	1	1	PWR	NOS raised to the power in TOS Result to NOS Pop Stack
DATA MANIPULATION COMMANDS (3)									
R	0	0	0	0	0	0	0	NOP	No Operation
R	0	0	1	1	1	1	1	FIXS	Converts TOS from floating point to single precision fixed point format
R	0	0	1	1	1	1	0	FIXD	Converts TOS from floating point to double precision fixed point format
R	0	0	1	1	1	0	1	FLTS	Converts TOS from single precision fixed point to floating point format
R	0	0	1	1	1	0	0	FLTD	Converts TOS from double precision fixed point to floating point format
R	1	1	1	0	1	0	0	CHSS	Changes sign of single precision fixed point operand on TOS
R	0	1	1	0	1	0	0	CHSD	Changes sign of double precision fixed point operand on TOS
R	0	0	1	0	1	0	1	CHSF	Changes sign of floating point operand on TOS
R	1	1	1	0	1	1	1	PTOS	Push single precision fixed point operand on TOS to NOS.
R	0	1	1	0	1	1	1	PTOD	Push double precision fixed point operand on TOS to NOS
R	0	0	1	0	1	1	1	PTOF	Push floating point operand on TOS to NOS
R	1	1	1	1	0	0	0	POPS	Pop single precision fixed point operand from TOS NOS becomes TOS
R	0	1	1	1	0	0	0	POPD	Pop double precision fixed point operand from TOS NOS becomes TOS
R	0	0	1	1	0	0	0	POPF	Pop floating point operand from TOS NOS becomes TOS
R	1	1	1	1	0	0	1	XCHS	Exchange single precision fixed point operands TOS and NOS
R	0	1	1	1	0	0	1	XCHD	Exchange double precision fixed point operands TOS and NOS
R	0	0	1	1	0	0	1	XCHF	Exchange floating point operands TOS and NOS
R	0	0	1	1	0	1	0	PUPI	Push floating point constant "π" onto TOS Previous TOS becomes NOS

Notes 1 NOMENCLATURE TOS is Top Of Stack NOS is Next On Stack

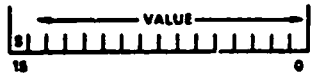
2 All derived floating point functions destroy the contents of the stack Only the result can be counted on to be valid upon command completion

3 Format conversion commands (FIXS, FIXD FLTS, FLTD) require that floating point data format be specified (command bits 5 and 6 must be 0)

DATA FORMATS

The Am9511 arithmetic processing unit handles operands in both fixed point and floating point formats. Fixed point operands may be represented in either single (16-bit operands) or double precision (32 bit operands), and are always represented as binary, two's complement values.

SINGLE PRECISION FIXED POINT FORMAT



DOUBLE PRECISION FIXED POINT FORMAT

The sign (positive or negative) of the operand is located in the most significant bit (MSB) Positive values are represented by a sign bit of zero (S = 0) Negative values are represented by the two's complement of the corresponding positive value with a sign bit equal to 1 (S = 1) The range of values that may be accommodated by each of these formats is −32,768 to +32,767 for single precision and −2,147,483,648 to +2,147,483,647 for double precision

Floating point binary values are represented in a format that permits arithmetic to be performed in a fashion analogous to operations with decimal values expressed in scientific notation

$$(5\ 83 \times 10^{2})\ (8\ 16 \times 10^{1}) = (4\ 75728 \times 10^{4})$$

In the decimal system, data may be expressed as values between 0 and 10 times 10 raised to a power that effectively shifts the implied decimal point right or left the number of places necessary to express the result in conventional form (e g , 47,572 8) The value portion of the data is called the mantissa The exponent may be either negative or positive.

The concept of floating point notation has both a gain and a loss associated with it The gain is the ability to represent the significant digits of data with values spanning a large dynamic range limited only by the capacity of the exponent field For example, in decimal notation if the exponent field is two digits wide, and the mantisa is five digits, a range of values (positive or negative) from 1.0000×10^{-99} to $9\ 9999 \times 10^{+99}$ can be accommodated. The loss is that only the significant digits of the value can be represented. Thus there is no distinction in this representation between the values 123451 and 123452, for example, since each would be expressed as 1.2345×10^{5} The sixth digit has been discarded In most applications where the dynamic range of values to be represented is large, the loss of significance, and hence accuracy of results, is a minor consideration. For greater precision a fixed point format could be chosen, although with a loss of potential dynamic range

The Am9511 is a binary arithmetic processor and requires that floating point data be represented by a fractional mantissa value between .5 and 1 multiplied by 2 raised to an appropriate power. This is expressed as follows

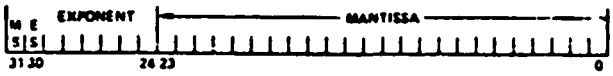
$$\text{value} = \text{mantissa} \times 2^{\text{exponent}}$$

For example, the value 100 5 expressed in this form is 0 1100 1001 $\times 2^{7}$ The decimal equivalent of this value may be computed by summing the components (powers of two) of the mantissa and then multiplying by the exponent as shown below:

$$\begin{aligned}\text{value} &= (2^{-1} + 2^{-2} + 2^{-5} + 2^{-8}) \times 2^{7} \\ &= 0\ 5 + 0\ 25 + 0.03125 + 0.00290625) \times 128 \\ &= 0.78515625 \times 128 \\ &= 100\ 5\end{aligned}$$

FLOATING POINT FORMAT

The format for floating point values in the Am9511 is given below. The mantissa is expressed as a 24 bit (fractional) value, the exponent is expressed as a two's complement 7-bit value having a range of −64 to +63. The most significant bit is the sign of the mantissa (0 = positive, 1 = negative), for a total of 32 bits. The binary point is assumed to be to the left of the most significant mantissa bit (bit 23) All floating point data values must be normalized Bit 23 must be equal to 1, except for the value zero, which is represented by all zeros

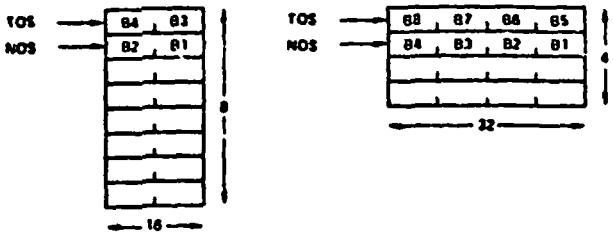


The range of values that can be represented in this format is ± ($2\ 7 \times 10^{-20}$ to $9\ 2 \times 10^{18}$) and zero

FUNCTIONAL DESCRIPTION

Stack Control

The user interface to the Am9511 includes access to an 8 level 16-bit wide data stack. Since single precision fixed point operands are 16-bits in length, eight such values may be maintained in the stack. When using double precision fixed point or floating point formats four values may be stored. The stack in these two configurations can be visualized as shown below



Data are written onto the stack, eight bits at a time, in the order shown (B1, B2, B3). Data are removed from the stack in reverse byte order (B8, B7, B6). Data should be entered onto the stack in multiples of the number of bytes appropriate to the chosen data format.

Data Entry

Data entry is accomplished by bringing the chip select ($\overline{CS}$), the command/data line (C/$\overline{D}$), and $\overline{WR}$ low, as shown in the timing diagram. The entry of each new data word "pushes down" the previously entered data and places the new byte on the top of stack (TOS). Data on the bottom of the stack prior to a stack entry are lost.

Data Removal

Data are removed from the stack in the Am9511 by bringing chip select ($\overline{CS}$), command/data (C/$\overline{D}$), and $\overline{RD}$ low as shown in the timing diagram. The removal of each data word redefines TOS so that the next successive byte to be removed becomes TOS. Data removed from the stack rotates to the bottom of the stack.

Command Entry

After the appropriate number of bytes of data have been entered onto the stack, a command may be issued to perform an operation on that data. Commands which require two operands for execution (e.g., add) operate on the TOS and NOS values. Single operand commands operate only on the TOS.

Commands are issued to the Am9511 by bringing the chip select ($\overline{CS}$) line low, command/data (C/$\overline{D}$) line high, and $\overline{WR}$ line low as indicated by the timing diagram. After a command is issued, the CPU can continue execution of its program concurrently with the Am9511 command execution.

Command Completion

The Am9511 signals the completion of each command execution by lowering the End Execution line ($\overline{END}$). Simultaneously, the busy bit in the status register is cleared and the Service Request bit of the command register is checked. If it is a "1" the service request output level (SVREQ) is raised [illegible] active low End Execution ($\overline{END}$) pulse. Similarly, the service request line is cleared by the [illegible] of an active low Service Acknowledge ($\overline{SVACK}$) [illegible].

Pause Operation

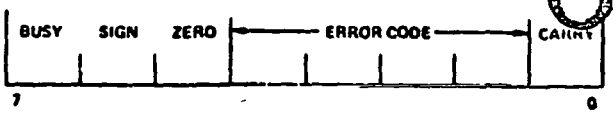
An active low pause ($\overline{PAUSE}$) is provided. This line is high in its quiescent state and is [illegible] by the Am9511 during the following conditions:

1. A previously initiated operation is in progress (device busy) and Command Entry has been attempted. In this case, the $\overline{PAUSE}$ line will be pulled low and remain low until completion of the current command execution. It will then go high, permitting entry of the new command.
2. A previously initiated operation is in progress and data access has been attempted. In this case, the $\overline{PAUSE}$ line will be pulled low, will remain in that state until execution is complete, and will then be raised to permit completion of the stack access.
3. The Am9511 is not busy, and data removal has been requested. $\overline{PAUSE}$ will be pulled low for the length of time necessary to transfer the byte from the top of stack to the interface latch, and will then go high, indicating availability of the data.
4. The Am9511 is not busy, and a data entry has been requested. $\overline{PAUSE}$ will be pulled low for the length of time required to ascertain if the preceding data byte, if any, has been written to the stack. If so $\overline{PAUSE}$ will immediately go high. If not, $\overline{PAUSE}$ will remain low until the interface latch is free and will then go high.
5. When a status read has been requested $\overline{PAUSE}$ will be pulled low for the length of time necessary to transfer the status to the interface latch, and will then be raised to permit completion of the status read. Status may be read whether or not the Am9511 is busy.

When $\overline{PAUSE}$ goes low, the APU expects the bus control signals present at the time to remain stable until $\overline{PAUSE}$ goes high.

Device Status

Device status is provided by means of an internal status register whose format is shown below



BUSY Indicates that Am9511 is currently executing a command (1 = Busy)

SIGN Indicates that the value on the top of stack is negative (1 = Negative)

ZERO Indicates that the value on the top of stack is zero (1 = Value is zero)

ERROR CODE This field contains an indication of the validity of the result of the last operation. The error codes are

- 0000 – No error
- 1000 – Divide by zero
- 0100 – Square root or log of negative number
- 1100 – Argument of inverse sine, cosine, or e^x too large
- XX10 – Underflow
- XX01 – Overflow

CARRY Previous operation resulted in carry or borrow most significant bit. (1 = Carry/Borrow, 0 = No Carry, No Borrow.)

If the BUSY bit in the status register is a one, the other status bits are not defined. If zero, indicating not busy, the operation is complete and the other status bits are defined as given above.

Read Status

The Am9511 status register can be read by the CPU at any time (whether an operation is in progress or not) by bringing the chip select ($\overline{CS}$) low, the command/data line (C/$\overline{D}$) high, and lowering $\overline{RD}$. The status register is then gated onto the data bus and may be input by the CPU.

EXECUTION TIMES

Timing for execution of the Am9511 command set is contained in Table 1. All times are given in terms of clock cycles. Where substantial variation of execution times is possible, the minimum and maximum values are quoted, otherwise, typical values are given. Variations are data dependent.

Total execution times may require allowances for operand transfer into the APU, command execution, and result retrieval from the APU. Except for command execution, these times will be be heavily influenced by the nature of the data, the control interface used, the speed of memory, the CPU used, the priority allotted to DMA and Interrupt operations, the size and number of operands to be transferred, and the use of chained calculations, etc.

COMMAND EXECUTION TIMES

Command Mnemonic	Clock Cycles	Command Mnemonic	Clock Cycles	Command Mnemonic	Clock Cycles	Command Mnemonic	Clock Cycles
SADD	17	FADD	56-350	LN	4478	POPF	12
SSUB	30	FSUB	58-352	EXP	4616	XCHS	18
SMUL	92	FMUL	168	PWR	9292	XCHD	26
SDIV	92	FDIV	171	NOP	4	XCHF	26
DADD	21	SQRT	800	CHSS	26	PUPI	16
DSUB	38	SIN	4464	CHSD	34		
DMUL	208	COS	4113	CHSF	16		
DDIV	208	TAN	5754	PTOS	16		
FIXS	92 216	ASIN	7668	PTOD	20		
FIXD	100 346	ACOS	7734	PTOF	20		
FLTS	98 186	ATAN	6006	POPS	10		
FLTD	98 378	LOG	4490	POPD	12		

MAXIMUM RATINGS above which useful life may be impaired

Storage Temperature	−65°C to +150°C
Ambient Temperature Under Bias	−55°C to +125°C
VDD with Respect to VSS	−0.5V to +15.0V
VCC with Respect to VSS	−0.5V to +7.0V
All Signal Voltages with Respect to VSS	−0.5V to +7.0V
Power Dissipation	2.0W

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of static charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

OPERATING-RANGE

Part Number	Ambient Temperature	VSS	VCC	VDD
Am9511DC Am9511-4DC	$0°C \le T_A \le +70°C$	0V	+5.0V ± 5%	+12V ± 5%
Am9511DM	$-55°C \le T_A \le +125°C$	0V	+5.0V ± 10%	+12V ± 10%

ELECTRICAL CHARACTERISTICS Over Operating Range (Note 1)

Parameters	Description	Test Conditions	Min.	Typ.	Max.	Units
VOH	Output HIGH Voltage	IOH = −200μA	3.7			Volts
VOL	Output LOW Voltage	IOL = 3.2mA			0.4	Volts
VIH	Input HIGH Voltage		2.0		VCC	Volts
VIL	Input LOW Voltage		−0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VI ≤ VCC			±10	μA
IOZ	Data Bus Leakage	VO = 0.4V			−100	μA
		VO = VCC			100	μA
ICC	VCC Supply Current	T_A = +25°C		55		mA
		T_A = 0°C				
		T_A = −55°C				
IDD	VDD Supply Current	T_A = +25°C		55		mA
		T_A = 0°C				
		T_A = −55°C				
CO	Output Capacitance	f_C = 1.0MHz, Inputs = 0V		8	10	pF
CI	Input Capacitance			5	8	pF
CIO	I/O Capacitance			10	12	pF

SWITCHING CHARACTERISTICS over operating range (Note 1)

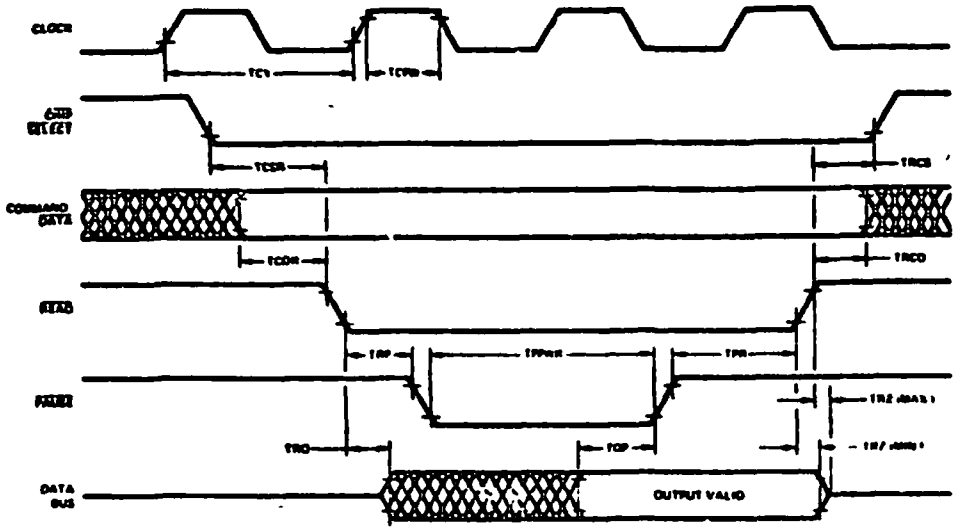
Parameters	Description		Am9511 Min.	Am9511 Max.	Am9511-4 Min.	Am9511-4 Max.	Units
[illegible]APW	$\overline{EACK}$ LOW Pulse Width		100		50		ns
[illegible]DR	C/$\overline{D}$ to $\overline{RD}$ LOW Set up Time		0		0		ns
[illegible]DW	C/$\overline{D}$ to $\overline{WR}$ LOW Set up Time		0		0		ns
TCPW	Clock Pulse Width		200		100		ns
TCSR	$\overline{CS}$ LOW to $\overline{RD}$ LOW Set up Time		50		25		ns
TCSW	$\overline{CS}$ LOW to $\overline{WR}$ LOW Set up Time		50		25		ns
TCY	Clock Period		480	5000	250	2500	ns
TDW	Data Bus Stable to $\overline{WR}$ HIGH Set up Time			200		100	ns
TEAE	$\overline{EACK}$ LOW to $\overline{END}$ HIGH Delay			200		100	ns
TEPW	$\overline{END}$ LOW Pulse Width (Note 2)		400		200		ns
TOP	Data Bus Output Valid to $\overline{PAUSE}$ HIGH Delay		0		0		ns
TPPWR	$\overline{PAUSE}$ LOW Pulse Width Read (Note 3)		1850		925		ns
TPPWW	$\overline{PAUSE}$ LOW Pulse Width Write (Note 3)		0		0		ns
TPR	$\overline{PAUSE}$ HIGH to $\overline{RD}$ HIGH Hold Time		0		0		ns
TPW	$\overline{PAUSE}$ HIGH to $\overline{WR}$ HIGH Hold Time		0		0		ns
TRCD	$\overline{RD}$ HIGH to C/$\overline{D}$ Hold Time		0		0		ns
TRCS	$\overline{RD}$ HIGH to $\overline{CS}$ HIGH Hold Time		0		0		ns
TRO	$\overline{RD}$ LOW to Data Bus ON Delay		50		25		ns
TRP	$\overline{RD}$ LOW to $\overline{PAUSE}$ LOW Delay (Note 4)			200		100	ns
TRZ	$\overline{RD}$ HIGH to Data Bus OFF Delay		50	200	25	100	ns
TSAPW	SVACK LOW Pulse Width		100		50		ns
TSAR	SVACK LOW to SVREQ LOW Delay			300		150	ns
TWCD	$\overline{WR}$ HIGH to C/$\overline{D}$ Hold Time		60		30		ns
TWCS	$\overline{WR}$ HIGH to $\overline{CS}$ HIGH Hold Time		60		30		ns
TWD	$\overline{WR}$ HIGH to Data Bus Hold Time		0		0		ns
TWI	Write Inactive Time	(Command)	2TCY	3TCY	2TCY	3TCY	ns
		(Data)	3TCY	4TCY	3TCY	4TCY	ns
[illegible]	$\overline{WR}$ LOW to $\overline{PAUSE}$ LOW Delay (Note 4)			200		100	ns

NOTES:

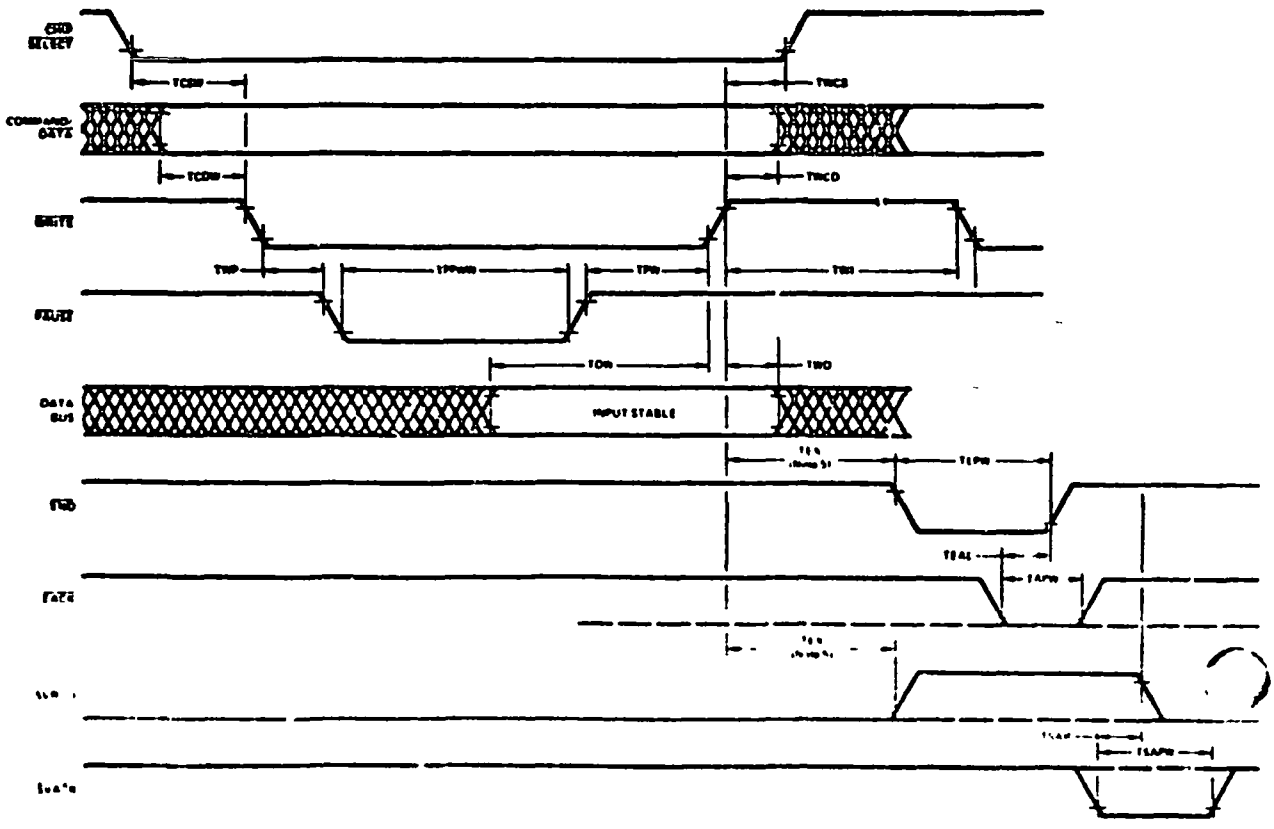
1. Typical values are for $T_A = 25°C$ and nominal supply voltage
2. $\overline{END}$ low pulse width is specified for $\overline{EACK}$ tied to VSS.
3. Based on stack access only. Variable, refer to functional description for details.
4. $\overline{PAUSE}$ is pulled low for both command and data operations
5. TEX is the execution time of the current command (see the Command Execution Times table).

SWITCHING WAVEFORMS

READ OPERATIONS



WRITE OPERATIONS



APPLICATION INFORMATION

The diagram in Figure 2 shows the interface connections for the Am9511 APU with operand transfers handled by an Am9517 DMA controller, and CPU coordination handled by Am9519 Interrupt Controller. The APU interrupts the CPU to indicate that a command has been completed. When the performance enhancements provided by the DMA and interrupt operations are not required, the APU interface can be simplified as shown in Figure 1. The Am9511 APU is designed with a general purpose 8-bit data bus and interface control so that it can be conveniently used with any general 8-bit processor.

In many systems it will be convenient to use the microcomputer system clock to drive the APU clock input. In the case of 8080A/9080A systems it would be the ϕ2TTL signal. Its cycle time will usually fall in the range of 250ns to 1000ns, depending on the system speed.

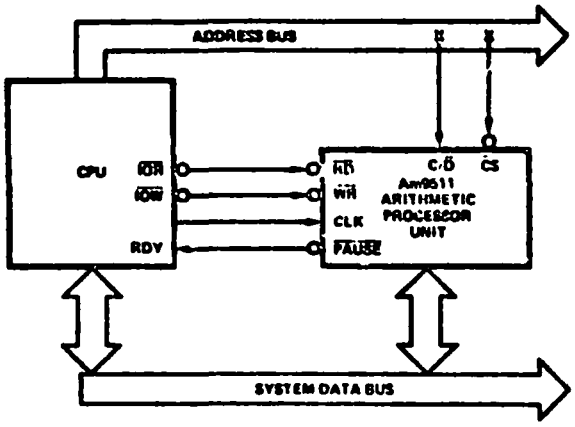


Figure 1. Am9511 Minimum Configuration Example.

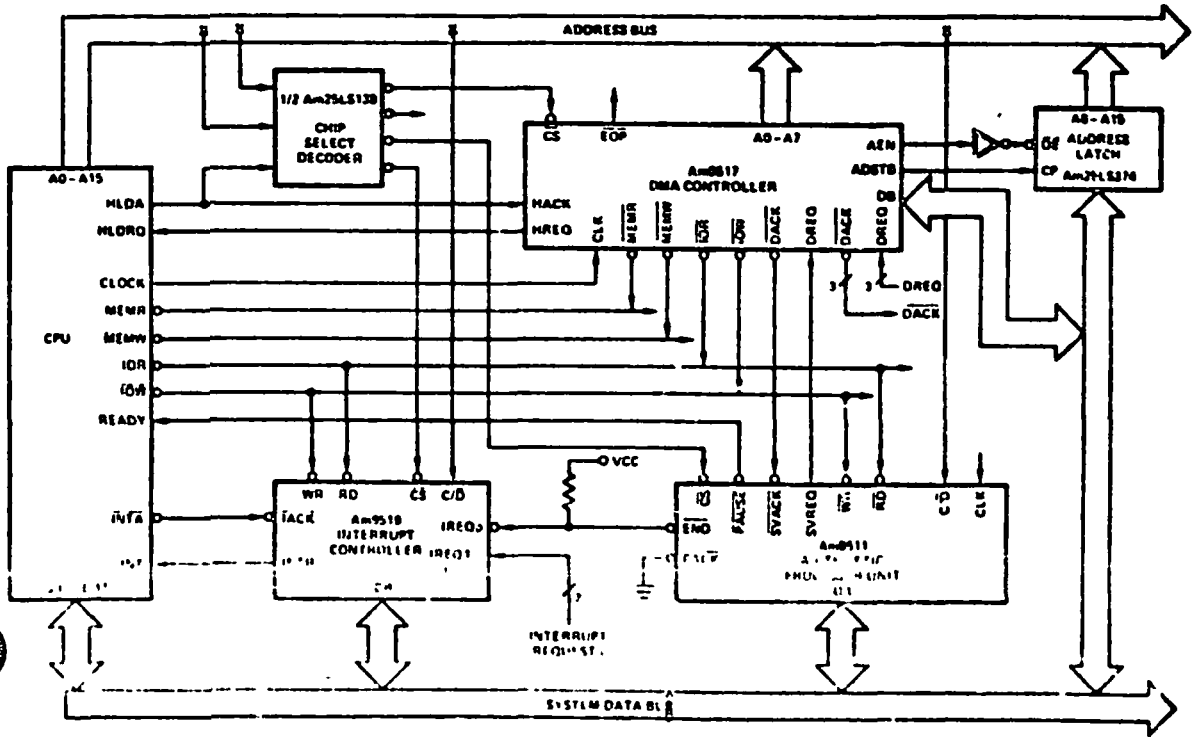


Figure 2 Am9511 High Performance Configuration Example

LSI-11

The following material describes the Digital Equipment Corporation LSI-11 microcomputer system. When packaged for an end-user, this unit is sold by the vendor as the PDP 11/03 minicomputer.

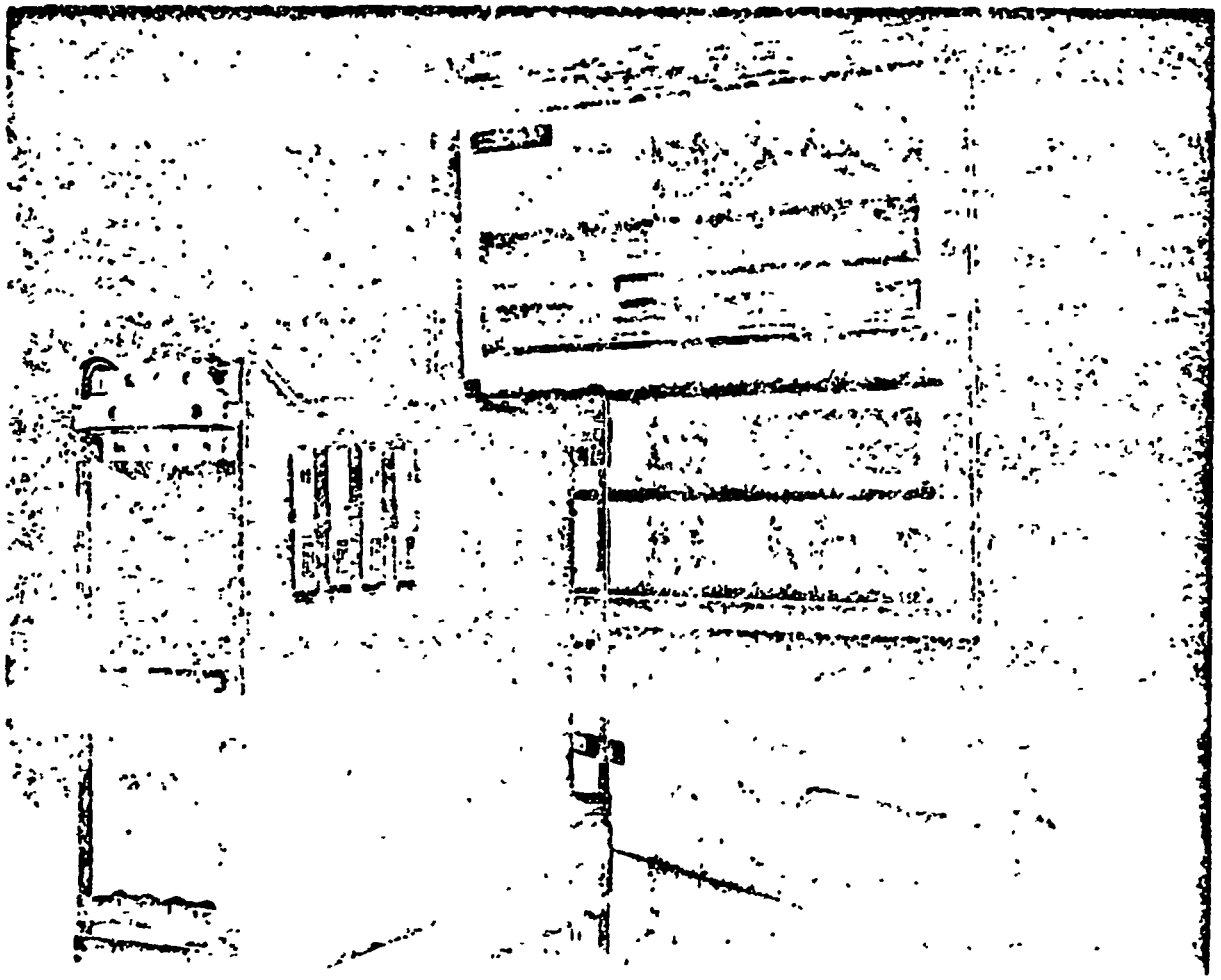
DIGITAL EQUIPMENT CORPORATION

OPTION BULLETIN

COMPONENTS GROUP

JANUARY 1977

LSI-11 Microcomputer System



INTRODUCTION

The use of large-scale integration (LSI) technology will takes new and existing product designs past a series of traditional barriers such as size, weight, packaging, reliability/maintainability and cost.

LSI technology enables DIGITAL to put an N-channel MOS central processor, 4096 (4K)-word random-access memory (RAM), vectored automatic priority interrupt logic, real-time clock input, power failure/autorestart logic, and buffered parallel 16-bit input/output port on one 8.5-by-10-inch printed circuit board
Added to these LSI-11 features are an enclosure and power supply (including lights, switches and fan) to produce the PDP-11/03—the LSI-11 in a box.

LSI-11 or the PDP-11/03 can be expanded by choosing from any of several memory and I/O modules—RAM, PROM/ROM or core memories, serial and parallel I/O interfaces, printed-circuit backplane/card guide assembly—all interfaced to the LSI-11 bus.

LSI modularity means you buy only what you need. You are free to concentrate your expertise and maximize your profitability by applying DIGITAL's microcomputer capability to a whole new range of control and processing opportunities.

Choosing either LSI-11 or 11/03 enables you to offer the newest products in the popular DIGITAL PDP-11 family of minicomputers, a family with unparalleled user acceptance. More than 30,000 PDP-11s are in use, the largest number in processing and control applications. Over four years of development and field experience are yours to profit from when you offer LSI microcomputers to your customers.

The wealth of tools available to help integrate the LSI-11 or PDP-11/03 into your product means that you can cut short your development cycle and get to market sooner.

Here's what is available

- A large, flexible instruction repertoire, including the 400-plus instructions of the basic PDP-11/40
- A simplified, application-oriented bus structure for maximum ease in handling I/O and memory operations.
- Off-the-shelf, plug-in expansion interfaces
- Off-the-shelf, plug-in core, RAM, and/or PROM/ROM expansion memories
- Resident firmware debugging techniques and ASCII console routines
- Operating system development on standard PDP-11/35, 11/40 or LSI-11
- Software and hardware training classes
- Complete documentation, including user's programming and maintenance manuals, configuration and installation guides
- The unmatched resources of the DECUS library for PDP-11 application programs

LSI-11 HARDWARE AND FIRMWARE

Microcomputer Module KD11-F

The 16-bit control processor functions are contained in four silicon gate N-channel metal oxide semiconductor (MOS), large-scale integration (LSI), integrated circuit chips. These chips provide all instructions, decoding, bus control, and arithmetic/logic unit (ALU) functions of the processor. The central processor contains eight general registers which can serve as accumulators, index registers, autoincrement/autodecrement registers, or stack pointers.

4096-by-16 read/write MOS semiconductor memory is contained on the microcomputer module. This memory is composed of LSI dynamic random-access memory (RAM) chips that require little operating power, provide fast access time, and are refreshed automatically by the processor's microcode, which is transparent to the user. A memory register on the KD11-F module addresses all on-board memory plus LSI-11 bus-compatible expansion memory up to 32K words or 64K bytes.

Multiplexed parallel I/O bus port-DMA operation. The LSI-11 bus is a high-speed, 38-line parallel bus containing data, address, control and synchronization lines. Sixteen lines are used for time multiplexing of data and addresses. All data and control lines are bidirectional, asynchronous, open-collector lines capable of providing a maximum parallel data transfer rate of 833K words per second under direct memory access operation.

Powerful PDP-11/40 basic instruction set. More than 400 powerful instructions make up the LSI-11's extensive basic instruction set. There are no separate memory, I/O or accumulator instructions. Thus the user can manipulate data in peripheral device registers as flexibly as in memory registers.

The basic operation code uses both single- and double-operand instructions for words or bytes, making it possible to perform such operations as adding, subtracting, or moving two operands in one step. This can reduce the number of instructions needed for many routines by as much as two-thirds. Much of the LSI-11's operating flexibility and processing power are derived from its wide variety of addressing techniques. Addressing can be direct, indirect, autoincrement, autodecrement, byte or word, indexing and stack-addressing. This flexibility means the LSI-11 can deal with data in the most efficient manner. The general registers can be used interchangeably as stack pointers, accumulators, and index registers. Address modification can be done directly in the general registers.

The KEV-11 Extended Instruction and floating-point instruction option provides fixed-point multiplication, division, and multiple shifting in single-precision arithmetic as well as floating-point addition, subtraction, multiplication and division.

Single-level, vectored, automatic priority interrupt provides for user-implementation of a priority-structured I/O interrupt system. Devices electrically closest to the microcomputer module receive highest priority, for either DMA or programmed I/O transfers. (DMA devices have a higher priority than programmed I/O devices.) This structure allows nesting of interrupts to as many levels as there are devices connected to the LSI-11 bus. Upon receipt of an interrupt grant, the device directs the processor to an interrupt vector location which contains the starting address of the device interrupt service routine and the new processor status word.

Real-time clock input signal line functions as an external interrupt line. When connected to a frequency source, it can serve as a real-time processor interrupt. A jumper on the microcomputer module enables or disables this highest priority interrupt function.

Asynchronous operation of all system modules permits each to function at its highest possible speed.

Power fail/auto restart provides jumper-selective restart through a power-up vector, a defined location, or an octal debugging technique (ODT) microcode.

Power failure is one of a series of errors and programming conditions which will cause the central processor to trap to a set of fixed locations.

Whenever dc power sequencing signals indicate an impending ac power loss, a microcoded power-fail sequence is initiated. The microcomputer traps to location 24 to execute a user's power-down routine. This will make possible an orderly system shut-down.

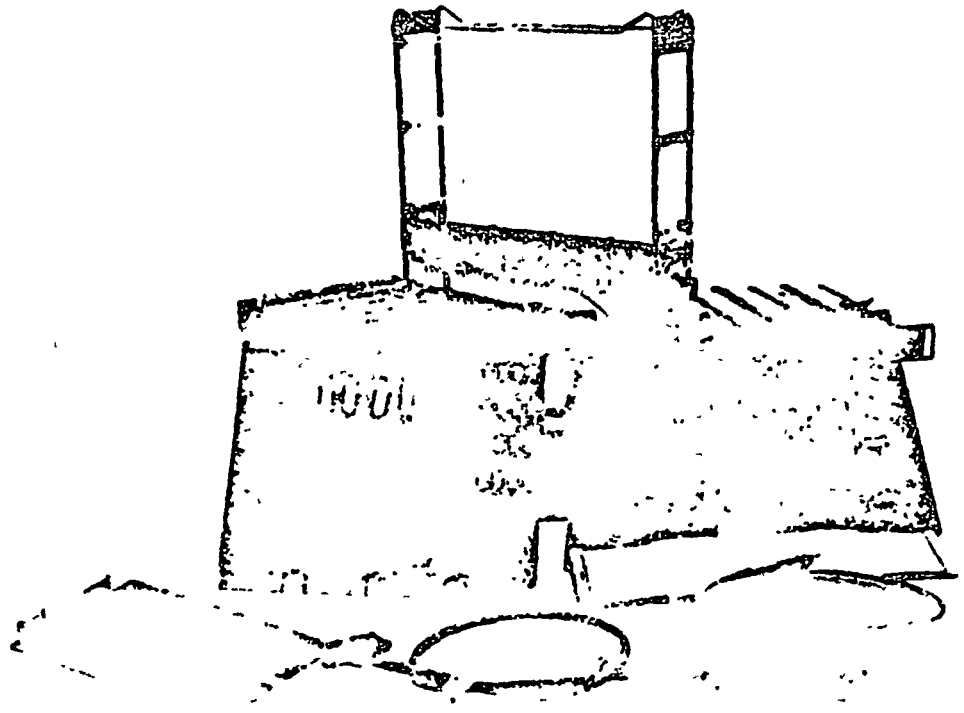
When power is restored, the processor can execute one of four jumper selected options.

1. The processor traps to location 24 and executes a user-defined power-up routine to restore the machine to its state prior to power failure.
2. Power-up to a defined location in memory.
3. Power-up to the ODT/console firmware routine (this assumes that an I/O interface that responds to the device address is present).
4. Power-up to a microcode bootstrap program (this assumes that the device corresponding to the bootstrap is present).

ODT/ASCII console routine/bootstrap all are resident in microcode to provide automatic entry into the debugging mode, replacement of conventional programmers' panel lights and switches with any terminal device generating standard ASCII codes, and the ability to automatically commence operation through resident bootstrap routines.

Word or byte processing provides very efficient handling of 8-bit characters without the need to rotate, swap, or mask.

8.5-by-10-inch board contains all of these features.



LSI-11 System Components

Expansion Memory Modules

4K dynamic random-access memory—MSV11-B is a dual-size (8 5-by-5-inch) read/write memory module utilizing dynamic MOS semiconductor memory devices. The module capacity is 4096 words of 16 bits, with memory-select circuitry for operation on 4K address boundaries. Dynamic memory refresh is performed automatically every 1 67 milliseconds by microcode on the microcomputer module.

16K dynamic MOS memory—MSV11-C is a quad-size (8 5-by-10 inch) read/write memory module with 16K words of 16 bits each. This module features 4K dynamic MOS technology, internal refresh, 4K bank memory addresses, and 750 nanosecond cycle time with 390 nanosecond access time. There are no special power requirements and memory contents can be protected in the event of a power loss by user-implemented battery back-up power source.

4K programmable read-only memory—MRV11-AA is a dual-size (8 5-by 5-inch) field programmable, read-only module utilizing either 256 x 4 or 512 x 4 fusible-link semiconductor devices. The module's maximum capacity is 2048 or 4096 words of 16 bits (depending upon which device is used), and is expandable in 256- or 512-word increments. This module is configured with 32 sockets for mounting memory IC devices of the user's choice. PROM chips can be supplied as an option. A pin-compatible masked ROM chip is available for volume applications so that the lowest possible cost can be achieved. Board-mounted jumpers enable selection of the module's address.

4K core memory module—MMV11-A is a quad-size (8.5-by-10-inch) core, read/write memory module containing 4096 words of 16 bits, with memory address selection circuitry for starting operation on any 4K boundary. Core memory provides non-volatile read/write storage for applications requiring protection against power losses.

Interfacing Modules

Serial line unit—DLV11 is a universal asynchronous receiver/transmitter serial interface module for use between the LSI-11 bus and serial devices. It is a dual-size (8.5-by-5-inch) module with the following features:

- Either optically isolated 20mA current loop or EIA interface.
- Selectable baud rates: 50, 75, 110, 134.5, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600
- Jumper-selectable stop bits and data bits
- LSI-11 bus interface and control logic for interrupt processing and vectored addressing of interrupt service routines.
- Interrupt priority determined by electrical position along the LSI-11 bus
- Control/status register (CSR) compatible with PDP-11 software routines. CSRs and receiver data buffer registers directly accessed via processor instructions.
- Plug, signal, and program compatible with PDP-11 DL 11-C

Parallel line unit—DRV11 is a general-purpose, 16-bit parallel interface between the LSI-11 bus and the user's peripheral device. It is a dual-size (8 5-by-5-inch) module with the following features:

- 16 diode-clamped data input lines
- 16 latched output lines
- 16-bit word or 8-bit byte data transfers.
- Complete device address decoding user-assigned
- LSI-11 bus interface and control logic for interrupt processing and vectored addressing of interrupt service routines.
- Interrupt priority determined by electrical position along the LSI-11 bus.
- Control/status registers (CSR) compatible with PDP-11 software routines. CSR and receiver data buffer registers directly accessed via processor instructions
- Plug, signal, and program compatible with PDP-11 DR11-C.
- Four control lines available to the peripheral device for output data ready, output data accepted, input data ready, and input data accepted logic operations
- Can be used with TTL or DTL logic-compatible devices
- Maximum data transfer rate of 90K words per second under program control
- Maximum drive capability of 25 feet of cable

Parallel line DMA—DRV11-B is a quad-size (8.5-by-10 inch) direct memory access interface module. It requires two device locations on the bus. The interface is programmed by the processor to move variable length blocks of 16-bit data words to or from specified locations in the system memory via the LSI-11 bus. Once programmed, no processor intervention is required to complete the data transfer. The DRV11-B is capable of transfer rates up to 250K, 16-bit word per second, and is capable of operating in burst modes and byte addressing.

LSI-11 bus foundation—DRV11-P is a versatile wire wrap module on a quad-size (8.5-by-10 inch) board. It requires two device locations on the bus. It contains a preassembled bus interface logic and can accommodate up to 60 14-pin ICs.
Because the bus interface logic is included, the module can be efficiently configured by the user to satisfy a variety of device interface logic applications.

Backplane/Card Guide Assembly H9270

The H9270 backplane/card guide assembly is a prewired LSI-11 structured backplane based on a standard DIGITAL four-by-four slot configuration. The H9270 has the following features:

- Designed to accept one microcomputer and up to six I/O and memory modules
- All LSI-11 bus data, control, and power connections are prewired on the printed circuit backplane to each module location
- Easily implemented, priority-structured I/O bus system based upon electrical position along the LSI-11 bus. Device priority levels established by a daisy-chained grant signal arrangement for interrupt and DMA requests. Placement of modules into the backplane automatically passes the bus grant signal to the next lower-priority device
- Backplane integral with card guide assembly
- Mounting capability in all planes
- Backplane size 11.15 by 2.8 by 11.0 inches

Power Supply H780

The H700 power supply provides the required dc voltage and current for the H9270, H9271, or equivalent backplane in the LSI-11 system.

The H780 features:

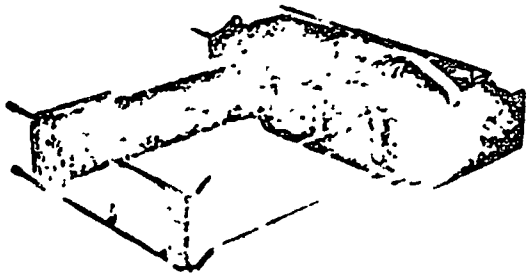
- +5V±3%, 18A (maximum) and +12V±3%, 3.5A (maximum), combined dc power must not exceed 120W.
- Overcurrent/short circuit protection—Output voltages return to normal after removal of overload or short. Current limited to approximately 1.2 times the required maximum rating.
- Overvoltage protection—+5V limited to +6.3V (approximately); +12V limited to +15V (approximately).
- Dual primary power configuration—Can be connected for nominal 115 V, 60 Hz or 230 V, 50 Hz input power
- Line Time Clock—A bus-compatible signal is generated by the power supply for the event (line time clock) interrupt input to the processor. This signal is either 50 or 60 Hz, depending upon primary power line frequency input to the power supply
- Power Fail/Automatic Restart—Fault detection and status circuits monitor ac and dc voltages and generate bus-compatible BPOK H and BDCOK H signals (respectively) to inform the LSI-11 system modules of power supply status. Automatic power signal sequencing is provided.
- Fans—Built-in fans provide cooling for the power supply and LSI-11 modules contained in an adjacent H9270 backplane.

Expanded Backplane DDV11-B

The DDV11-B without card guide is an expanded version of the standard LSI-11 backplane for use when additional LSI-11 option module space is required. A nine-by-four slot section of the backplane is LSI-11 bus structured and will accept one microcomputer, up to 15 option modules and one bus terminator module. An additional nine-by-two slot section of the backplane is provided with power connections.

Expander Box H909-C

The H909-C expander box provides a most convenient means for expanding the LSI-11 system. Each box includes the card guide and space for the DDV11-B and the power supply.

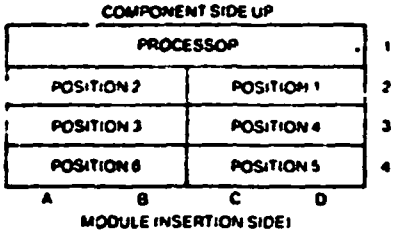


H780-h power supply attached to the H9270 backplane

LSI-11 BUS CONCEPT

The bus is a simple, fast, easy to use interface between LSI-11 modules. All LSI-11 modules connected to this common bidirectional bus structure receive the same interface signal lines. A typical system application in which the processor module, memory modules, and peripheral device interface modules are connected to the bus is shown in the diagram.

Bus data and control lines are bidirectional open-collector lines that are asserted low. The bus is composed of 16 data/address lines, 17 control/synchronization signal lines, and maintenance lines.



Control signal lines include two daisy-chained grant signals (four signal pins), which provides a priority-structured I/O system. The highest priority device is the module electrically located closest to the microcomputer module. Higher priority devices pass a grant signal to lower priority devices only when not requesting service. For example, Module A,' shown in Bus Priority Structure, is the highest-priority device, and is capable of interrupting processor operation (when interrupts are enabled) or executing DMA transfers at any time. Modules B and C have lower priorities, respectively, and can receive a grant signal when Module A is not requesting service. Similarly, Module C can receive a grant signal when both Modules A and B are not requesting service.

Both address and data words (or bytes) are multiplexed over the 16-bit bus. For example, during a programmed data transfer, the processor first asserts an address on the bus for a fixed time. After the address time has been completed, the processor executes the programmed input or output data transfer, the actual data transfer is asynchronous and requires a reply from the addressed 'device. Bus synchronization and control signals provide this function.

The processor module is capable of driving six device slots (double-height) along the bus, as supplied. Devices or memory can be installed in any location along this bus.

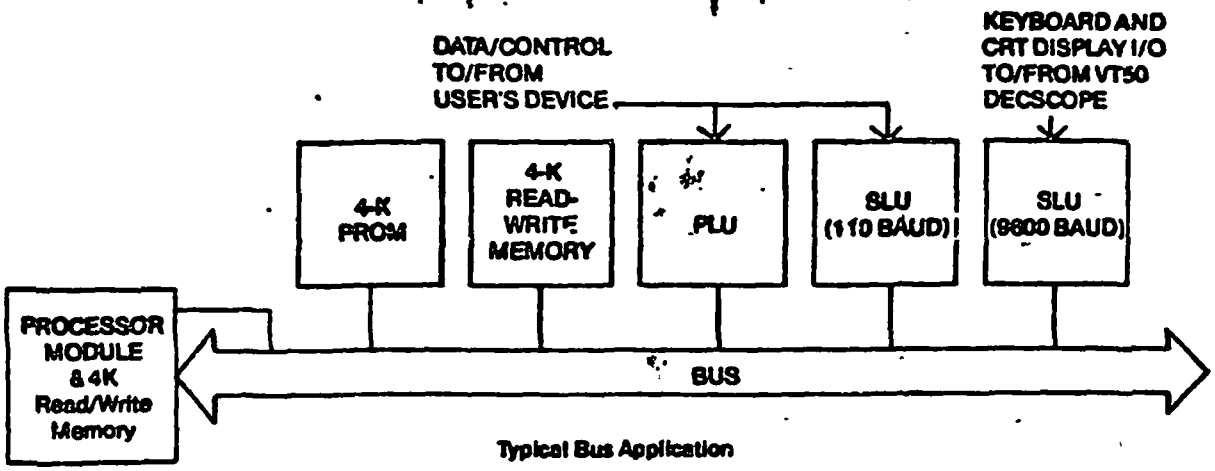
The processor's on-board memory address latch and address decoder addresses both the processor module's 4K RAM and generates bank-select signals on the bus.

The bus provides a vectored interrupt interface for any device. Hence, device polling is not required in interrupt processing routines. This results in a considerable savings in processing time when many devices requiring interrupt service are interfaced along the bus. When a device receives an interrupt grant signal, the processor inputs the device's interrupt vector. The vector points to two addresses which contain a new processor status word and the starting address of the interrupt service routine for the device.

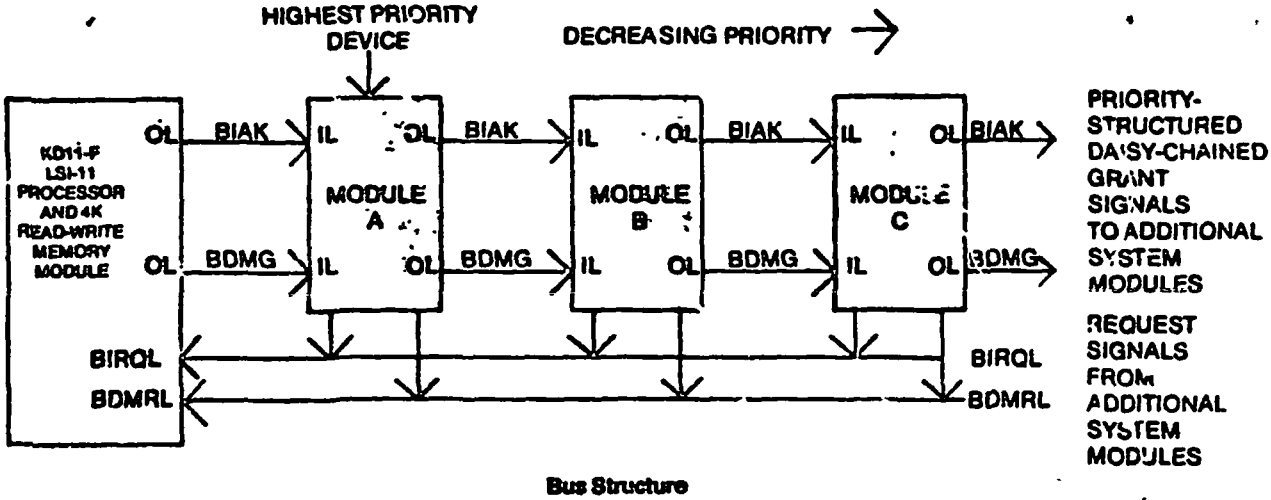
One input signal line functions as an external event interrupt line received on the processor module. This signal line can be connected to a frequency source, such as a line frequency, and used as a real-time interrupt. A jumper on the processor module enables or inhibits this function. When enabled, the device connected to this line has the highest interrupt priority external to the processor. Interrupt vector 100_8 is reserved for this function, and interrupt request via the external event line causes new PC and PS words to be loaded from locations 100_8 and 102_8.

MASS STORAGE FLOPPY DISK RXV11

The RXV11 is a dual drive floppy disk with interface. It has a capacity of 512K words with an average access time of 483 microseconds and provides unlimited off-line storage.



Typical Bus Application



Bus Structure

LSI-11 INSTRUCTION SET

SINGLE OPERAND

Mnemonic	Instruction
General	
CLR(B)	clear
COM(B)	complement
INC(B)	increment
DEC(B)	decrement
NEG(B)	negate
TST(B)	test
Shift & Rotate	
ASR(B)	arithmetic shift right
ASL(B)	arithmetic shift left
ROR(B)	rotate right
ROL(B)	rotate left
SWAB	swap bytes
Multiple Precision	
ADC(B)	add carry
SBC(B)	subtract carry
SXT	sign extend
PS WORD OPERATORS	
MFPS	move byte from PS
MTPS	move byte to PS

DOUBLE OPERAND

General	
MOV(B)	move
CMP(B)	compare
ADD	add
SUB	subtract
Logical	
BIT(B)	bit test
BIC(B)	bit clear
BIS(B)	bit set
XOR	exclusive or

CONDITION CODE

CLC	clear C
CLV	clear V
CLZ	clear Z
CLN	clear N
CCC	clear all CC bits
SEC	set C
SEV	set V
SEZ	set Z
SEN	set N
SCC	set all CC bits
NOP	no operation

PROGRAM CONTROL

Mnemonic	Instruction
Branch	
BR	branch (unconditional)
BNE	branch if not equal (to zero)
BEQ	branch if equal (to zero)
BPL	branch if plus
BMI	branch if minus
BVC	branch if overflow is clear
BVS	branch if overflow is set
BCC	branch if carry is clear
BCS	branch if carry is set
Signed Conditional Branch	
BGE	branch if greater than or equal (to zero)
BLT	branch if less than (zero)
BGT	branch if greater than (zero)
BLE	branch if less than or equal (to zero)
Unsigned Conditional Branch	
BHI	branch if higher
BLOS	branch if lower or same
BHIS	branch if higher or same
BLO	branch if lower
Jump & Subroutine	
JMP	jump
JSR	jump to subroutine
RTS	return from subroutine
MARK	mark
SOB	subtract one and branch (if=0)
Trap & Interrupt	
EMT	emulator trap
TRAP	trap
BPT	breakpoint trap
IOT	input/output trap
RTI	return from interrupt
RTT	return from interrupt

MISCELLANEOUS

HALT	halt
WAIT	wait for interrupt
RESET	reset external bus

FIXED POINT ARITHMETIC (EIS)

Mnemonic	Instruction
MUL	multiply
DIV	divide
ASH	shift arithmetically
ASCH	arithmetic shift combined

FLOATING POINT ARITHMETIC (FIS)

Mnemonic	Instruction
FADD	floating add
FSUB	floating subtract
FMUL	floating multiply
FDIV	floating divide

LSI-11 SOFTWARE

LSI-11 systems software includes a paper tape software operating package, a variety of operating systems, programming languages, diagnostic software, and special software options.

The LSI-11 paper tape software (QJV10-AB) is available as a basic utility package. It consists of an Editor, which allows the user to create and modify ASCII source files to be used as input to other system programs, an Assembler, which allows the user to translate assembly language programs into executable machine-coded programs; a Loader, which allows the user to input programs and data from various media into the machine, an On-line Debugging Technique (ODT) Package, which allows the user to debug assembled and linked programs, and an Input/Output Executive, which allows the user to control the flow of data to and from devices under program control

Real-Time Operating System RT-11

RT-11 is a floppy disk based, single-user, foreground/background system that can support a real-time job execution in the foreground and an interactive or batch program development job in the background. It is a high performance system which combines fast, on-line access with high level programming language capabilities and user-beneficial features such as stack processing and vectored interrupts.

Resource-Sharing Operating System RSX-11S

RSX-11S is an execute-only operating system designed to provide the most efficient resource-sharing environment for multiple real-time activities without a mass storage device. This operating system features multiprogramming, priority scheduling, contingency exist, and power-fail shut-down and auto-restart

RT-11 Programming Languages

MACRO-11, the assembler, provides full macro programming capabilities in systems with 8K of memory It has facilities for maintaining and using a macro library and performing conditional assembly

Multi-user BASIC is a fast incremental compiler developed by DIGITAL using a conversational programming language developed at Dartmouth. It provides on-line timeshared access to the LSI-11 Several users simultaneously can develop programs, enter and retrieve data, examine files, and communicate It is one of the easier programming aids to master, yet it offers extremely sophisticated techniques for complex manipulations and efficient problem-solving

FORTRAN-IV is an updated, improved version of a widely accepted scientific problem-solving language and compiler, contained in 8K words of memory It can be used to perform integer, real and double precision operations. Both program execution and compilation is much faster using this version Input and output can be accessed directly, and all RT-11 monitor functions are completely accessible through callable subroutines. Object programs are put out in run time format without any intermediate assembly

PROM Formatter QJV11-CB

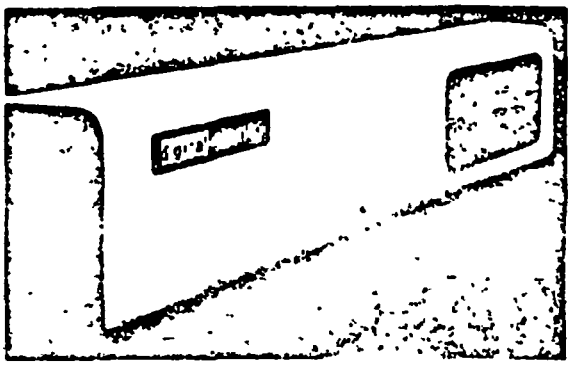
This software generates correctly formatted tape from which a PROM chip can be blasted.

LSI-11 Paper Tape Diagnostics ZJV01-RB

These tapes test the processor, exercise the memory, isolate problem modules and exercise the I/O devices.

DEVELOPMENT SYSTEMS

Development of applications-oriented software is a typical assignment for the PDP-11/03 and 11V03 systems among original equipment manufacturers building the LSI-11 board set into their products These systems, integrated with PDP-11 software are designed to offer powerful and flexible computation resources at a low price



PDP-11/03

The PDP-11/03 is formed by adding the following elements to the basic LSI-11 microcomputer module the power supply (including lights, switches and a dual fan assembly), standard rack mountable enclosure measuring 5¼ by 19 by 13 inches, H9270 backplane/card guide assembly The 11/03 is available in 4K RAM (115 and 230 volt) and 16K RAM and 4K Core (115 and 230 volt) versions. Each is expandable beyond the basic configuration by adding the options currently available as modules with the LSI-11

The PDP-11/03 is designed with a removable front panel (pop panel) By removing this panel, you expose the LSI modules and cables for easy removal or replacement from the front The power supply is located on the right side and has lights and switches attached so that when the front panel is removed, the lights and switches remain functional

PDP-11/03 Operating Specifications

Temperature	41°F to 122°F
Relative Humidity	10% to 95% (no condensation)
Input Voltage	
PDP-11/03-AA, BA	90-132 Vac, 115 Vac nominal, 47-63 Hz
PDP-11/03-AB, BB	180-264 Vac, 230 Vac nominal 47-63 Hz
Input Power	
PDP-11/03-AA, AB, BA, BB	210 watts max at full load, 190 watts typical at full load

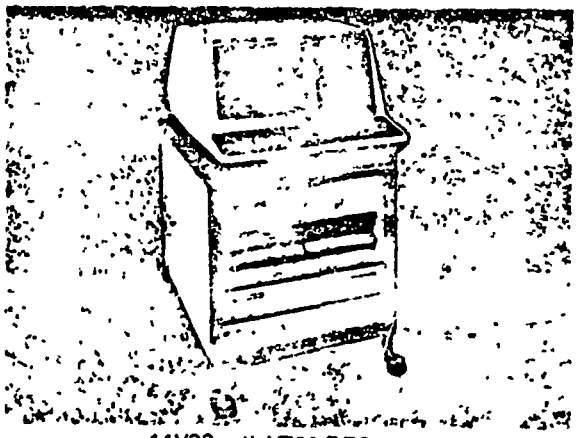
PDP-11V03

The PDP-11V03 consists of a PDP-11/03 with 8K of semiconductor read write memory dual drive floppy disk system terminator bootstrap module with DMA refresh a VT52 DECscope or LA36 DECwriter II input/output terminal and cursor control set with power distribution panel and DLV11 serial line unit

The 11V03 includes RT-11 which is a small, single-user foreground/background system that can support a real-time application job's execution in the foreground and an interactive or batch program development job in the background Programming languages supported under RT-11 are MACRO-11, Fortran IV and BASIC.

PDP-11V03 Operating Specifications

Temperature	
Operating	15 to 32°C (59 to 90° F)
Nonoperating	-35 to 60° C (-30 to 140° F)
(diskettes, nonoperating)	-35° to 52° C (-30 to 125° F)
Relative Humidity	
Operating	20 to 80%
Nonoperating	5% to 98%, noncondensing
(diskettes)	10 to 80%, noncondensing
Magnetic Field	less than 50 oersteds
Mechanical	
Cabinet size	26" H x 28" D x 21½" W
Weight	205 pounds
Electrical	
Input Voltage	
PDP-11V03-AA	100 to 127 Vac, 60 Hz, ±1 Hz, with VT52
PDP-11V03-EA	100 to 127 Vac, 60 Hz, ±1 Hz, with LA36
PDP-11V03-AD	200 to 254 Vac, 50 Hz, ±1 Hz, with VT52
PDP-11V03-ED	200 to 254 Vac, 50 Hz, ±1 Hz, with LA36
Input Power	940 watts max at full load

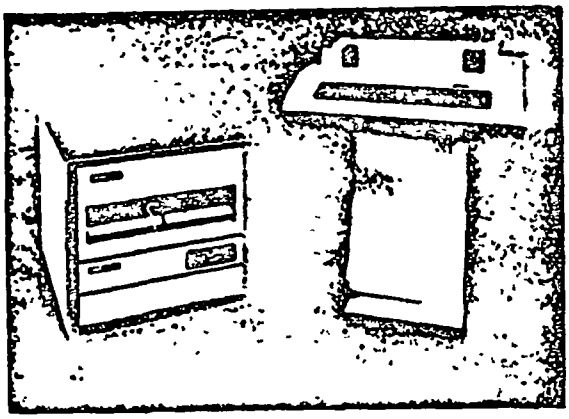


11V03 with VT52 DECscope

MICROCOMPUTER APPLICATIONS

Configuration flexibility, unparalleled software provisions, and a complete array of interface terminal and memory options make the LSI-11 an ideal addition to control communication and processing systems

The reliability small size, and cost-savings made possible by the LSI-11's large-scale integration architecture can justify its use in many new applications, including those summarized below



11V03 with LA36 DECwriter II.

In a Remittance Processing System:
The LSI-11 is both economical enough and powerful enough to be incorporated in a new line of single-user remittance processing stations Each station has its own LSI-11 with 30K words of memory This contrasts with traditional shared-processor systems that divide their computer power among a number of operator stations Due to their greater complexity, these older systems are often both I/O bound and computer bound The LSI-11 based stand-alone systems avoid this problem, yet cost less than a shared processor system with the same number of stations.

In an Ultrasonic Scanning System:
The LSI-11 serves as the central controller in an ultrasonic scanning system that uses high-frequency, low energy, pulsed sound waves to produce cross sectional views of the human anatomy Unlike X-ray, ultrasound can distinguish between different kinds of tissues, diagnose pregnancy and its abnormalities, and visualize a variety of conditions including tumors, certain opthalmic lesions, and other abnormalities

In Distributed Process Control:
LSI-11s are at the heart of a cost-effective distributed process control system that is helping plastic sheet and film manufacturers make better use of their increasingly expensive raw materials. LSI-11s control extruders to minimize set-up times and start-up scrap and to hold down variations in product thickness The controllers all report to a supervisory LSI-11, which provides the real-time operator information and control, and the centralized management reporting, usually associated with larger, more expensive systems

In a Voice-Input Terminal System:
The LSI-11 allowed an OEM to develop a less expensive voice input terminal that can reach a much wider market than previous terminals of this kind The terminal accepts the operator s spoken words identifies them from among his pre-recorded speech patterns and then relays the data in digital form to the host computer In industrial plants, for instance, the voice input terminal allows quality control inspectors to enter data while they look at and handle work pieces Since there is no time lag for manual data recording and subsequent keying, the host computer can provide the inspectors with real-time trend analysis results to help them spot problems early.

In Programmable Signs:
An LSI-11 controller is responsible for many of the programmable signs used in commercial and industrial applications One OEM's signs display alphanumeric, graphic, and high-contrast photographic information, even animated cartoons—in 16,000-point dot matrix formats. A single LSI-11 controller drives several signs. Another OEM builds an LSI-11 into a stand-alone system that has practical applications ranging from interactive graphic design and editing to classroom or other group visual presentations Both OEMs chose the LSI-11 because it could handle the large amount of processing their systems required In addition, the microcomputer had the proven hardware and the software development and debugging tools to make one-of-a-kind applications practical.

In Processing Color Film:
LSI-11s control a fully-computerized color film processing system for photofinishing laboratories. The system reduces photographic paper waste as much as 50% by minimizing operator errors, reducing set-up paper loss, and providing tighter quality control. The added throughput saves labor costs and lets photofinishers handle more volume without having to buy additional color printers. The system consists of from four to 25 LSI-11 controllers linked to a host PDP-11/35, which provides overall control and manages a central data base LSI-11s were ideal for this distributed control operation, according to the OEM, because their local computational power allows the central 11/35 minicomputer to handle up to 25 printers at once The LSI-11 s software and system compatibility with the 11/35, as well as its powerful set of 400 basic instructions, was a big advantage in developing application software

In Computer Numerical Control:
LSI-11s control a new generation of grinding machines that produce more uniform surfaces and require less operator intervention than competing machines The biggest advantage of the new grinding machines is that their LSI-11 controllers can be easily programmed for maximum productivity in performing a given application The programmer typically prepares the actual grinding program and stands by to watch the first workpiece as it's produced If the programmer sees a way to simplify, speed up, or in some other way improve the process, he can actually cycle backward and forward through the program, rewriting and testing as he goes The programmer can optimize the software and validate the results without ever having to leave the production floor Hardwired controllers or tape-fed controllers just don t permit this kind of flexibility

SERVICE AND WARRANTY

LSI-11 microcomputers are warranted for 90 days Return any defective unit during this period and it will be repaired or replaced without charge

Post-warranty service can be provided in a variety of ways, including complete repair through a local DIGITAL depot, a contract with DIGITAL's Field Service Group, or the purchaser's own service organization, with or without training at DIGITAL.

ORDERING INFORMATION

For further information on DIGITAL component products, contact:

Digital Equipment Corporation, Components Group, One Iron Way, Marlborough, Massachusetts 01752 Call 800-225-9480 toll-free from 8.30 AM to 5.30 PM our time.*

In Canada:
Digital Equipment of Canada Limited, PO Box 11500, Ottawa, Ontario, K2H 8K8 Call (613) 592-5111, ext. 154

In Europe:
Digital Equipment Corporation International (Europe), 81, route de l'Aire, C P 340, CH-1211 GENEVE 26 Call (022)42 79 50

*U.S only, Massachusetts residents please dial (617) 481-7400

digital

DIGITAL EQUIPMENT CORPORATION, Corporate Headquarters: Maynard, Massachusetts 01754, Telephone: (617) 897-5111—SALES AND SERVICE OFFICES: UNITED STATES—ALABAMA, Huntsville • ARIZONA, Phoenix and Tucson • CALIFORNIA, El Segundo, Los Angeles, Oakland, Ridgecrest, San Diego, San Francisco (Mountain View), Santa Ana, Santa Clara, Stanford, Sunnyvale and Woodland Hills • COLORADO, Englewood • CONNECTICUT, Fairfield and Meriden • DISTRICT OF COLUMBIA, Washington (Lanham, MD) • FLORIDA, Ft Lauderdale and Orlando • GEORGIA, Atlanta • HAWAII, Honolulu • ILLINOIS, Chicago (Rolling Meadows) • INDIANA, Indianapolis • IOWA, Bettendorf • KENTUCKY, Louisville • LOUISIANA, New Orleans (Metairie) • MARYLAND, Odenton • MASSACHUSETTS, Marlborough, Waltham and Westfield • MICHIGAN, Detroit (Farmington Hills) • MINNESOTA, Minneapolis • MISSOURI, Kansas City (Independence) and St Louis • NEW HAMPSHIRE, Manchester • NEW JERSEY, Cherry Hill, Fairfield, Metuchen and Princeton • NEW MEXICO, Albuquerque • NEW YORK, Albany, Buffalo (Cheektowaga) Long Island (Huntington Station), Manhattan, Rochester and Syracuse • NORTH CAROLINA, Durham/Chapel Hill • OHIO, Cleveland (Euclid), Columbus and Dayton • OKLAHOMA, Tulsa • OREGON, Eugene and Portland • PENNSYLVANIA, Allentown, Philadelphia (Bluebell) and Pittsburgh • SOUTH CAROLINA, Columbia • TENNESSEE, Knoxville and Nashville • TEXAS, Austin, Dallas and Houston • UTAH, Salt Lake City • VIRGINIA, Richmond • WASHINGTON, Bellevue • WISCONSIN, Milwaukee (Brookfield) • INTERNATIONAL—ARGENTINA, Buenos Aires • AUSTRALIA, Adelaide, Brisbane, Canberra, Melbourne, Perth and Sydney • AUSTRIA, Vienna • BELGIUM, Brussels • BOLIVIA, La Paz • BRAZIL, Rio de Janeiro and Sao Paulo • CANADA, Calgary, Edmonton, Halifax, London Montreal, Ottawa Toronto Vancouver and Winnipeg • CHILE Santiago • DENMARK, Copenhagen • FINLAND Helsinki • FRANCE, Lyon, Grenoble and Paris • GERMAN FEDERAL REPUBLIC, Cologne Frankfurt, Hamburg Hannover Munich Nuremburg, Stuttgart and West Berlin • HONG KONG • INDIA, Bombay • INDONESIA, Djakarta • IRELAND, Dublin • ITALY, Milan, Rome and Turin • IRAN, Tehran • JAPAN, Osaka and Tokyo • MALAYSIA Kuala Lumpur • MEXICO Mexico City • NETHERLANDS Utrecht • NEW ZEALAND, Auckland and Christchurch • NORWAY Oslo • PUERTO RICO Santurce • SINGAPORE • SWEDEN Gothenburg and Stockholm • SWITZERLAND Geneva and Zurich • UNITED KINGDOM, Birmingham, Bristol, Epsom, Edinburgh Leeds, Leicester, London, Manchester and Reading • VENEZUELA, Caracas •

TMS 9900

The TMS 9900 is a 16-bit NMOS microprocessor sold by Texas Instruments Incorporated. It is functionally equivalent to the SBP 9900, an I^2L version also manufactured by Texas Instruments Incorporated.

1. INTRODUCTION

1.1 DESCRIPTION

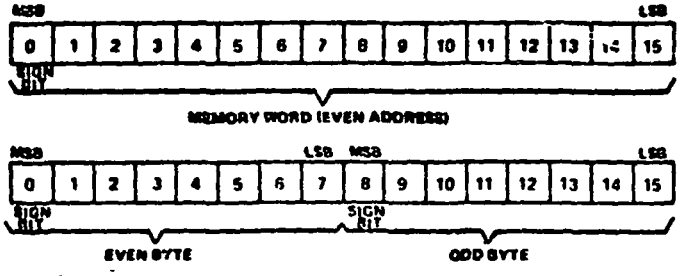
The TMS 9900 microprocessor is a single-chip 16-bit central processing unit (CPU) produced using N-channel silicon-gate MOS technology (see Figure 1). The instruction set of the TMS 9900 includes the capabilities offered by full minicomputers. The unique memory-to-memory architecture features multiple register files resident in memory which allow faster response to interrupts and increased programming flexibility. The separate bus structure simplifies the system design effort. Texas Instruments provides a compatible set of MOS and TTL memory and logic function circuits to be used with a TMS 9900 system. The system is fully supported by software and a complete prototyping system.

1.2 KEY FEATURES

- 16-Bit Instruction Word
- Full Minicomputer Instruction Set Capability Including Multiply and Divide
- Up to 65,536 Bytes of Memory
- 3.3-MHz Speed
- Advanced Memory-to-Memory Architecture
- Separate Memory, I/O, and Interrupt Bus Structures
- 16 General Registers
- 16 Prioritized Interrupts
- Programmed and DMA I/O Capability
- N-Channel Silicon-Gate Technology

2. ARCHITECTURE

The memory word of the TMS 9900 is 16 bits long. Each word is also defined as 2 bytes of 8 bits. The instruction set of the TMS 9900 allows both word and byte operands. Thus, all memory locations are on even address boundaries and byte instructions can address either the even or odd byte. The memory space is 65,536 bytes or 32,768 words. The word and byte formats are shown below.



2.1 REGISTERS AND MEMORY

The TMS 9900 employs an advanced memory-to-memory architecture. Blocks of memory designated as workspace replace internal hardware registers with program-data registers. The TMS 9900 memory map is shown in Figure 2. The first 32 words are used for interrupt trap vectors. The next contiguous block of 32 memory words is used by the extended operation (XOP) instruction for trap vectors. The last two memory words $FFFC_{16}$ and $FFFE_{16}$ are used for the trap vector of the LOAD signal. The remaining memory is then available for programs, data, and workspace registers. If desired, any of the special areas may also be used as general memory.

224 36

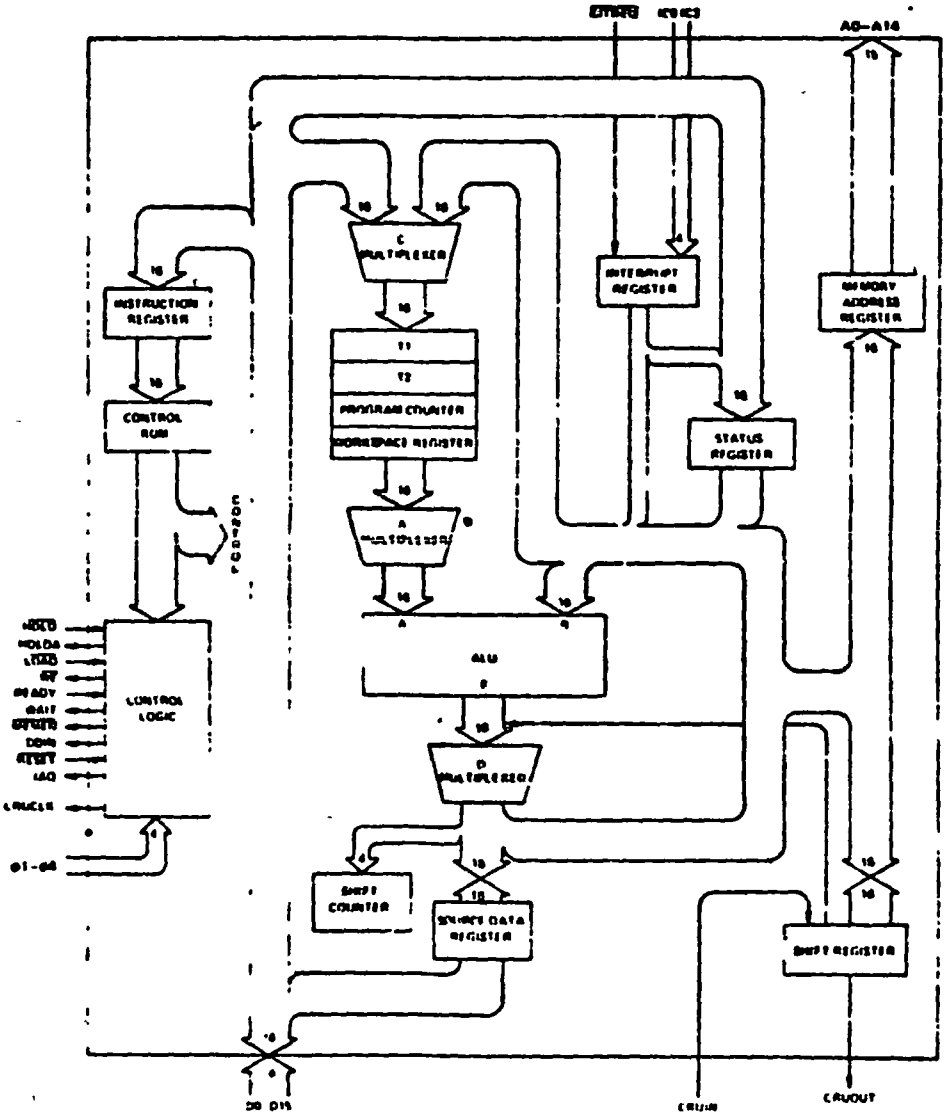


FIGURE 1 – ARCHITECTURE

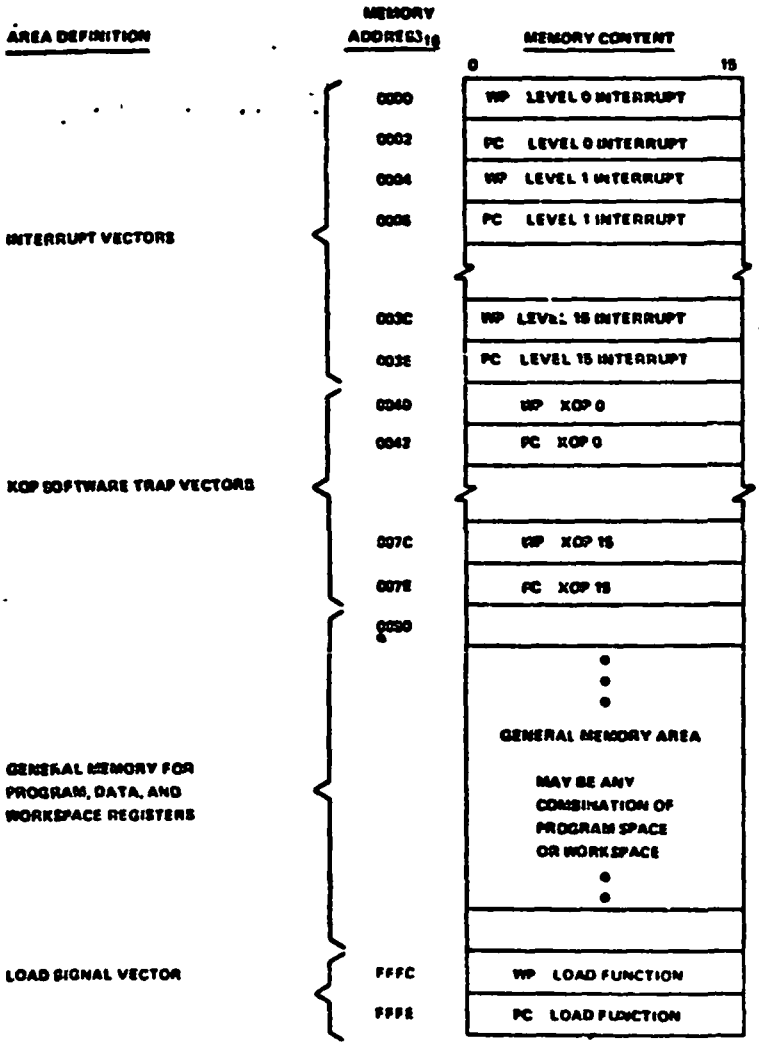
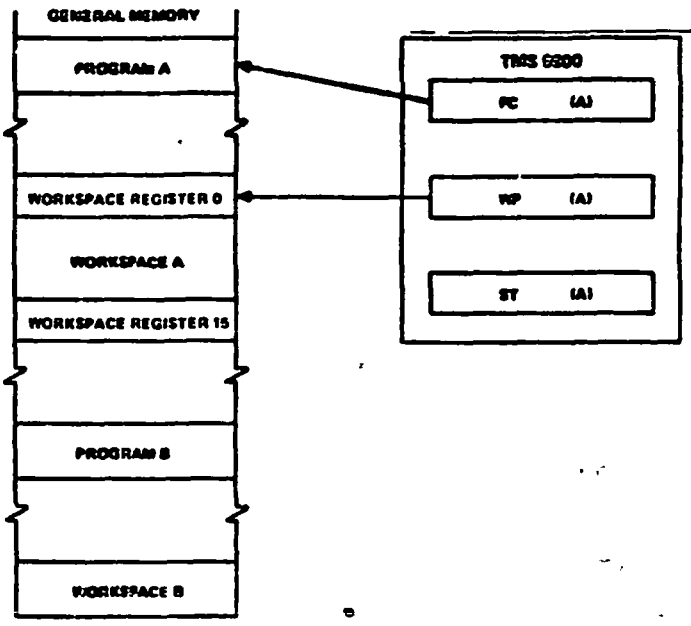


FIGURE 2 – MEMORY MAP

Three internal registers are accessible to the user. The program counter (PC) contains the address of the instruction following the current instruction being executed. This address is referenced by the processor to fetch the next instruction from memory and is then automatically incremented. The status register (ST) contains the present state of the processor and will be further defined in Section 3.4. The workspace pointer (WP) contains the address of the first word in the currently active set of workspace registers.

A workspace register file occupies 16 consecutive memory words in the general memory area (see Figure 2). Each workspace register may hold data or addresses and function as operand registers, accumulators, address registers, or

index registers. During instruction execution, the processor addresses any register in the workspace by adding the register number to the contents of the workspace pointer and initiating a memory request for the word. The relationship between the workspace pointer and its corresponding workspace is shown below.



The workspace concept is particularly valuable during operations that require a context switch, which is a change from one program environment to another (as in the case of an interrupt) or to a subroutine. Such an operation, using a conventional multi register arrangement, requires that at least part of the contents of the register file be stored and reloaded. A memory cycle is required to store or fetch each word. By exchanging the program counter, status register, and workspace pointer, the TMS 9900 accomplishes a complete context switch with only three store cycles and three fetch cycles. After the switch the workspace pointer contains the starting address of a new 16-word workspace in memory for use in the new routine. A corresponding time saving occurs when the original context is restored. Instructions in the TMS 9900 that result in a context switch include:

1. Branch and Load Workspace Pointer (BLWP)
2. Return from Subroutine (RTWP)
3. Extended Operation (XOP)

Device interrupts, $\overline{\text{RESET}}$ and $\overline{\text{LOAD}}$ also cause a context switch by forcing the processor to trap to a service subroutine.

2.2 INTERRUPTS

The TMS 9900 employs 16 interrupt levels with the highest priority level 0 and lowest level 15. Level 0 is reserved for the $\overline{\text{RESET}}$ function and all other levels may be used for external devices. The external levels may also be shared by several device interrupts, depending upon system requirements.

The TMS 9900 continuously compares the interrupt code (IC0 through IC3) with the interrupt mask contained in status register bits 12 through 15. When the level of the pending interrupt is less than or equal to the enabling mask level (higher or equal priority interrupt), the processor recognizes the interrupt and initiates a context switch following

completion of the currently executing instruction. The processor fetches the new context WP and PC from the interrupt vector locations. Then, the previous context WP, PC, and ST are stored in workspace registers 13, 14, and 15, respectively, of the new workspace. The TMS 9900 then forces the interrupt mask to a value that is one less than the level of the interrupt being serviced, except for level zero interrupt, which loads zero into the mask. This allows only interrupts of higher priority to interrupt a service routine. The processor also inhibits interrupts until the first instruction of the service routine has been executed to preserve program linkage should a higher priority interrupt occur. All interrupt requests should remain active until recognized by the processor in the device-service routine. The individual service routines must reset the interrupt requests before the routine is complete.

If a higher priority interrupt occurs, a second context switch occurs to service the higher priority interrupt. When that routine is complete, a return instruction (RTWP) restores the first service routine parameters to the processor to complete processing of the lower-priority interrupt. All interrupt subroutines should terminate with the return instruction to restore original program parameters. The interrupt-vector locations, device assignment, enabling-mask value, and the interrupt code are shown in Table 1.

TABLE 1
INTERRUPT LEVEL DATA

Interrupt Level	Vector Location (Memory Address In Hex)	Device Assignment	Interrupt Mask Values To Enable Respective Interrupts (ST12 thru ST15)	Interrupt Codes (IC0 thru IC3)
(Highest priority) 0	00	Reset	0 through F*	0000
1	04	External device	1 through F	0001
2	08		2 through F	0010
3	0C		3 through F	0011
4	10		4 through F	0100
5	14		5 through F	0101
6	18		6 through F	0110
7	1C		7 through F	0111
8	20		8 through F	1000
9	24		9 through F	1001
10	28		A through F	1010
11	2C		B through F	1011
12	30		C through F	1100
13	34		D through F	1101
14	38		E and F	1110
(Lowest priority) 15	3C	External device	F only	1111

*Level 0 can not be disabled.

The TMS 9900 interrupt interface utilizes standard TTL components as shown in Figure 3. Note that for eight or less external interrupts a single SN74148 is required and for one external interrupt $\overline{\text{INTREQ}}$ is used as the interrupt signal with a hard wired code (IC0 through IC3).

2.3 INPUT/OUTPUT

The TMS 9900 utilizes a versatile direct command-driven I/O interface designated as the communications-register unit (CRU). The CPU provides up to 4096 directly addressable input bits and 4096 directly addressable output bits. Both input and output bits can be addressed individually or in fields of from 1 to 16 bits. The TMS 9900 employs three dedicated I/O pins (CRUIN, CRUOUT, and CRUCLK) and 12 bits (A3 through A14) of the address bus to interface with the CRU system. The processor instructions that drive the CRU interface can set, reset, or test any bit in the CRU array or move between memory and CRU data fields.

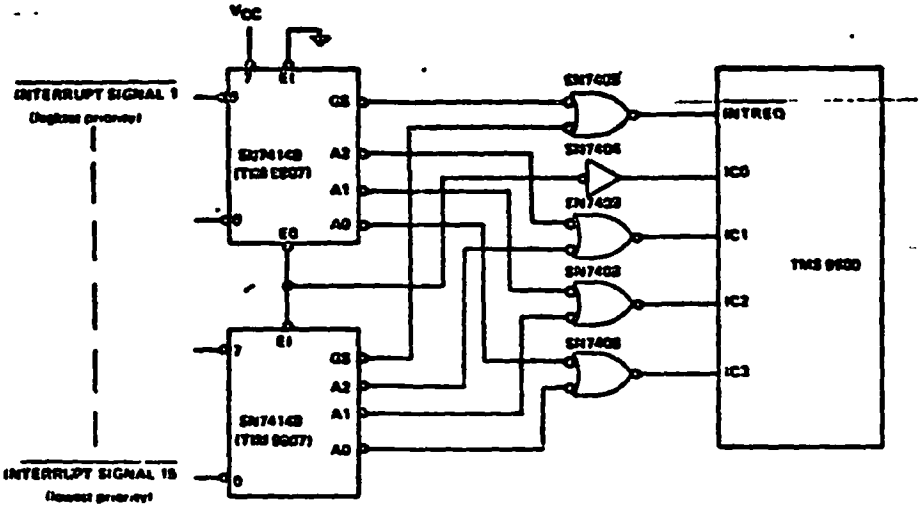


FIGURE 3 – TMS 9900 INTERRUPT INTERFACE

2.4 SINGLE BIT CRU OPERATIONS

The TMS 9900 performs three single-bit CRU functions: test bit (TB), set bit to one (SBO), and set bit to zero (SBZ). To identify the bit to be operated upon, the TMS 9900 develops a CRU-bit address and places it on the address bus, A3 to A14.

For the two output operations (SBO and SBZ), the processor also generates a CRUCLK pulse, indicating an output operation to the CRU device, and places bit 7 of the instruction word on the CRUOUT line to accomplish the specified operation (bit 7 is a one for SBO and a zero for SBZ). A test bit instruction transfers the addressed CRU bit from the CRUIN input line to bit 2 of the status register (EQUAL).

The TMS 9900 develops a CRU-bit address for the single-bit operations from the CRU-base address contained in workspace register 12 and the signed displacement count contained in bits 8 through 15 of the instruction. The displacement allows two's complement addressing from base minus 128 bits through base plus 127 bits. The base address from W12 is added to the signed displacement specified in the instruction and the result is loaded onto the address bus. Figure 4 illustrates the development of a single-bit CRU address.

2.5 MULTIPLE BIT CRU OPERATIONS

The TMS 9900 performs two multiple-bit CRU operations: store communications register (STCR) and load communications register (LDCR). Both operations perform a data transfer from the CRU to memory or from memory to CRU as illustrated in Figure 5. Although the figure illustrates a full 16-bit transfer operation, any number of bits from 1 through 16 may be involved. The LDCR instruction fetches a word from memory and right shifts it to serially transfer it on CRU output bits. If the LDCR involves eight or fewer bits, those bits come from the right justified field within the addressed byte of the memory word. If the LDCR involves nine or more bits, those bits come from the right justified field within the whole memory word. When transferred to the CRU interface, each successive bit receives an address that is sequentially greater than the address for the previous bit. This addressing mechanism results in an order reversal of the bits: that is, bit 15 of the memory word (or bit 7) becomes the lowest addressed bit in the CRU field and bit 0 becomes the highest addressed bit in the CRU field.

An STCR instruction transfers data from the CRU to memory. If the operation involves a byte or less transfer, the transferred data will be stored right justified in the memory byte with leading bits set to zero. If the operation involves nine or more bits, the transferred data is stored right justified in the memory word with leading bits set to zero.

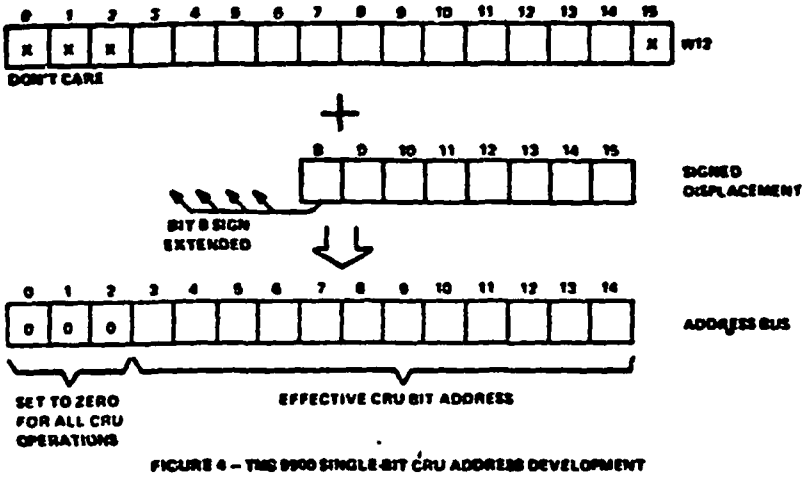


FIGURE 4 – TMS 9900 SINGLE-BIT CRU ADDRESS DEVELOPMENT

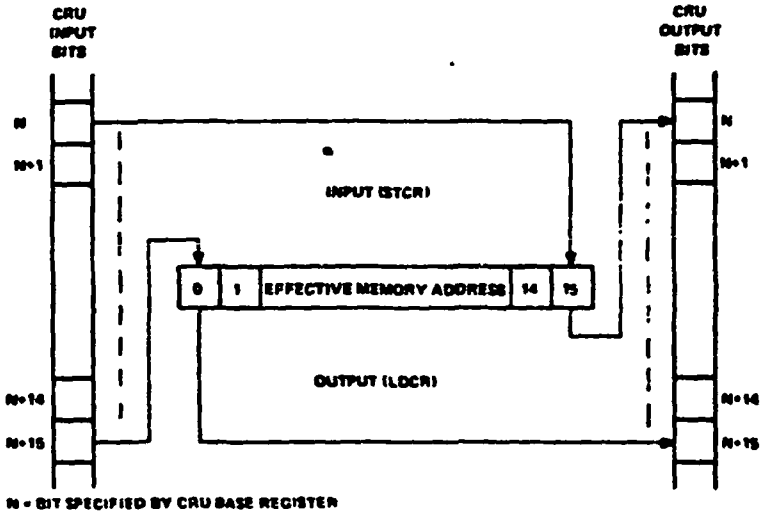


FIGURE 5 – TMS 9900 LDCR/STCR DATA TRANSFERS

When the input from the CRU device is complete, the first bit from the CRU is the least-significant bit position of the memory word or byte.

Figure 6 illustrates how to implement a 16-bit input and a 16-bit output register in the CRU interface. CRU addresses are decoded as needed to implement up to 256 such 16-bit interface registers. In system application however, only the exact number of interface bits needed to interface specific peripheral devices are implemented. It is not necessary to have a 16-bit interface register to interface an 8-bit device.

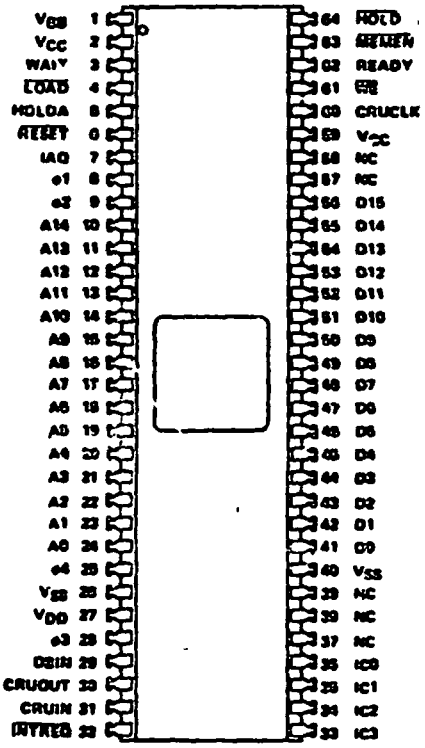
2.8 TMS 9900 PIN DESCRIPTION

Table 2 defines the TMS 9900 pin assignments and describes the function of each pin.

TABLE 2
TMS 9900 PIN ASSIGNMENTS AND FUNCTIONS

SIGNATURE	PIN	I/O	DESCRIPTION
			ADDRESS BUS
A0 (MSB)	24	OUT	A0 through A14 comprise the address bus. This 3-state bus provides the memory address vector to the external-memory system when MEMEN is active and I/O-port addresses and external-instruction addresses to the I/O system when MEMEN is inactive. The address bus assumes the high-impedance state when HOLDA is active
A1	23	OUT	
A2	22	OUT	
A3	21	OUT	
A4	20	OUT	
A5	19	OUT	
A6	18	OUT	
A7	17	OUT	
A8	16	OUT	
A9	15	OUT	
A10	14	OUT	
A11	13	OUT	
A12	12	OUT	
A13	11	OUT	
A14 (LSB)	10	OUT	
			DATA BUS
D0 (MSB)	41	I/O	D0 through D15 comprise the bidirectional 3-state data bus. The bus transfers memory data to (when writing) and from (when reading) the external-memory system when MEMEN is active. The data bus assumes the high-impedance state when HOLDA is active.
D1	42	I/O	
D2	43	I/O	
D3	44	I/O	
D4	45	I/O	
D5	46	I/O	
D6	47	I/O	
D7	48	I/O	
D8	49	I/O	
D9	50	I/O	
D10	51	I/O	
D11	52	I/O	
D12	53	I/O	
D13	54	I/O	
D14	55	I/O	
D15 (LSB)	56	I/O	
			POWER SUPPLIES
V_{BB}	1		Supply voltage (−5 V NOM)
V_{CC}	2,59		Supply voltage (5 V NOM). Pins 2 and 59 must be connected in parallel.
V_{DD}	27		Supply voltage (12 V NOM)
V_{SS}	26,40		Ground reference. Pins 26 and 40 must be connected in parallel.
			CLOCKS
ϕ1	8	IN	Phase-1 clock
ϕ2	9	IN	Phase-2 clock
ϕ3	28	IN	Phase-3 clock
ϕ4	25	IN	Phase-4 clock

TMS 9900 PIN ASSIGNMENTS



NC - No internal connection

2.9.3 CRU

CRU interface timing is shown in Figure 11. The timing for transferring two bits out and one bit in is shown. These transfers would occur during the execution of a CRU instruction. The other cycles of the instruction execution are not illustrated. To output a CRU bit, the CRU bit address is placed on the address bus A0 through A14 and the actual bit data on CRUOUT. During the second clock cycle a CRU pulse is supplied by CRUCLK. This process is repeated until the number of bits specified by the instruction are completed.

The CRU input operation is similar in that the bit address appears on A0 through A14. During the subsequent cycle the TMS 9900 accepts the bit input data as shown. No CRUCLK pulses occur during a CRU input operation.

3. TMS 9900 INSTRUCTION SET

3.1 DEFINITION

Each TMS 9900 instruction performs one of the following operations:

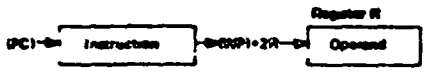
- Arithmetic, logical, comparison, or manipulation operations on data
- Loading or storage of internal registers (program counter, workspace pointer, or status)
- Data transfer between memory and external devices via the CRU
- Control functions

3.2 ADDRESSING MODES

TMS 9900 instructions contain a variety of available modes for addressing random-memory data (e.g., program parameters and flags), or formatted memory data (character strings, data lists, etc.). The following figures graphically describe the derivation of the effective address for each addressing mode. The applicability of addressing modes to particular instructions is described in Section 3.5 along with the description of the operations performed by the instruction. The symbols following the names of the addressing modes (R, *R, *R+, @ LABEL, or @ TABLE (R)) are the general forms used by TMS 9900 assemblers to select the addressing mode for register R.

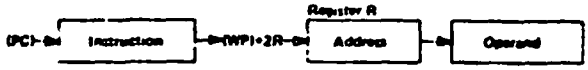
3.2.1 WORKSPACE REGISTER ADDRESSING R

Workspace Register R contains the operand.



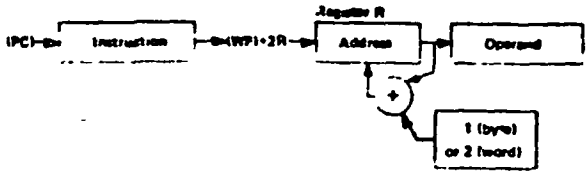
3.2.2 WORKSPACE REGISTER INDIRECT ADDRESSING *R

Workspace Register R contains the address of the operand.



3.2.3 WORKSPACE REGISTER INDIRECT AUTO INCREMENT ADDRESSING *R+

Workspace Register R contains the address of the operand. After acquiring the operand, the contents of workspace register R are incremented.

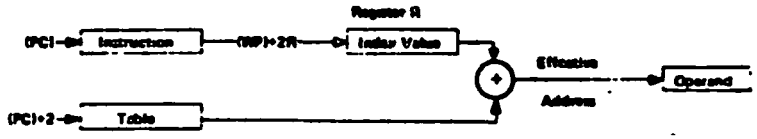


3.2.4 SYMBOLIC (DIRECT) ADDRESSING @ LABEL

The word following the instruction contains the address of the operand.

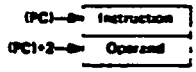
3.2.5 INDEXED ADDRESSING @ TABLE (R)

The word following the instruction contains the base address. Workspace register R contains the index value. The sum of the base address and the index value results in the effective address of the operand.



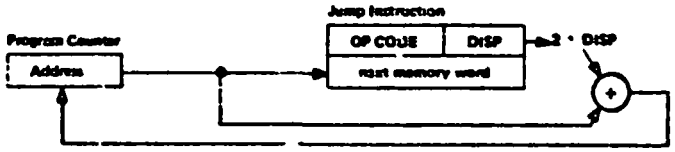
3.2.6 IMMEDIATE ADDRESSING

The word following the instruction contains the operand.



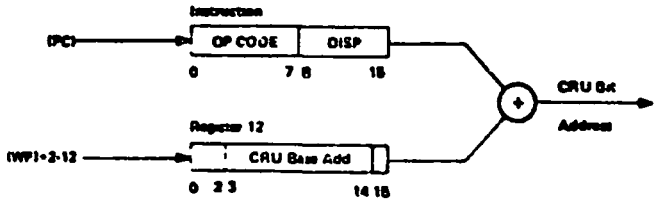
3.2.7 PROGRAM COUNTER RELATIVE ADDRESSING

The 8-bit signed displacement in the right byte (bits 8 through 15) of the instruction is multiplied by 2 and added to the updated contents of the program counter. The result is placed in the PC.



3.2.8 CRU RELATIVE ADDRESSING

The 8 bit signed displacement in the right byte of the instruction is added to the CRU base address (bits 3 through 14 of the workspace register 12). The result is the CRU address of the selected CRU bit.



3.3 TERMS AND DEFINITIONS

The following terms are used in describing the instructions of the TMS 9900

TERM	DEFINITION
B	Byte indicator (1=byte, 0 = word)
C	Bit count
D	Destination address register
DA	Destination address
IOP	Immediate operand
LSB(n)	Least significant (right most) bit of (n)
MSB(n)	Most significant (left most) bit of (n)
N	Don't care
PC	Program counter
Result	Result of operation performed by instruction
S	Source address register
SA	Source address
ST	Status register
STn	Bit n of status register
T_D	Destination address modifier
T_S	Source address modifier
W	Workspace register
WRn	Workspace register n
(n)	Contents of n
a → b	a is transferred to b
|n|	Absolute value of n
+	Arithmetic addition
–	Arithmetic subtraction
AND	Logical AND
OR	Logical OR
⊕	Logical exclusive OR
n̄	Logical complement of n

3.4 STATUS REGISTER

The status register contains the interrupt mask level and information pertaining to the instruction operation

0	1	2	3	4	5	6	7 8 9 10 11	12 13 14 15
ST0	ST1	ST2	ST3	ST4	ST5	ST6	not used (=0)	ST12 ST13 ST14 ST15
L	A	=	C	O	P	X		(interrupt Mask)

BIT	NAME	INSTRUCTION	CONDITION TO SET BIT TO 1
ST0	LOGICAL GREATER THAN	C CB	If MSB(SA) = 1 and MSB(DA) = 0 or if MSB(SA) = MSB(DA) and MSB of [(DA) – (SA)] = 1
		CI	If MSB(W) = 1 and MSB of IOP = 0 or if MSB(W) = MSB of IOP and MSB of [IOP – (W)] = 1
		ABS	If (SA) = 0
		All Others	If result = 0
ST1	ARITHMETIC GREATER THAN	C CB	If MSB(SA) = 0 and MSB(DA) = 1 or if MSB(SA) = MSB(DA) and MSB of [(DA) – (SA)] = 1
		CI	If MSB(W) = 0 and MSB of IOP = 1 or if MSB(W) = MSB of IOP and MSB of [IOP – (W)] = 1
		ABS	If MSB(SA) = 0 and (SA) ≠ 0
		All Others	If MSB of result = 0 and result ≠ 0

- Continued -

BIT	NAME	INSTRUCTION	CONDITION TO SET BIT TO 1
ST2	EQUAL	C, CB	If (SA) = (DA)
		CI	If (W) = IOP
		COC	If (SA) and $\overline{(DA)}$ = 0
		CZC	If (SA) and (DA) = 0
		TB	If CRUIN = 1
		ABS	If (SA) = 0
		All others	If result = 0
ST3	CARRY	A, AB ABS AI DEC, DECT, INC INCT, NEG S SB	If CARRY OUT = 1
		SLA SRA SRC SRL	If last bit shifted out = 1
ST4	OVERFLOW	A AB	If MSB(SA) = MSB(DA) and MSB of result ≠ MSB(DA)
		AI	If MSB(W) = MSB of IOP and MSB of result ≠ MSB(W)
		S SB	If MSB(SA) ≠ MSB(DA) and MSB of result = MSB(DA)
		DEC DECT	If MSB(SA) = 1 and MSB of result = 0
		INC INCT	If MSB(SA) = 0 and MSB of result = 1
		SLA	If MSB changes during shift
		DIV	If MSB(SA) = 0 and MSB(DA) = 1 or if MSB(SA) = MSB(DA) and MSB of [(DA) − (SA)] = 0
		ABS NEG	If (SA) = 8000_{16}
ST5	PARITY	CB MOVB	If (SA) has odd number of 1's
		LDCR, STCR	If 1 ≤ C ≤ 8 and (SA) has odd number of 1's
		AB SB SOCB SZCB	If result has odd number of 1's
ST6	XOP	XOP	If XOP instruction is executed
ST12-ST15	INTERRUPT MASK	LIMI	If corresponding bit of IOP is 1
		RTWP	If corresponding bit of WR15 is 1

3.5 INSTRUCTIONS

3.5.1 Dual Operand Instructions with Multiple Addressing Modes for Source and Destination Operand

General format

If B = 1 the operands are bytes and the operand addresses are byte addresses. If B = 0 the operands are words and the operand addresses are word addresses.

The addressing mode for each operand is determined by the T field of that operand.

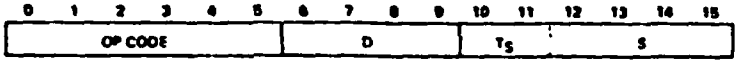
TS OR TD	S OR D	ADDRESSING MODE	NOTES
00	0 - 15	Workspace register	1
01	0 - 15	Workspace register indirect	
10	0	Symbolic	4
10	1 - 15	Indexed	2,4
11	0 - 15	Workspace register indirect auto-increment	3

NOTES: 1. When a workspace register is the operand of a byte instruction (bit 3 = 1) the left byte (bits 0 through 7) is the operand and the right byte (bits 8 through 15) is unchanged.
2. Workspace register 0 may not be used for indexing.
3. The workspace register is incremented by 1 for byte instructions (bit 3 = 1) and is incremented by 2 for word instructions (bit 3 = 0).
4. When TS = TD = 10 two words are required in addition to the instruction word. The first word is the source operand base address and the second word is the destination operand address.

MNEMONIC	OP CODE 0 1 2	B 3	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
A	1 0 1	0	Add	Yes	0-4	(SA)+(DA) → (DA)
AB	1 0 1	1	Add bytes	Yes	0-5	(SA)+(DA) → (DA)
C	1 0 0	0	Compare	No	0-2	Compare (SA) to (DA) and set appropriate status bits
CB	1 0 0	1	Compare bytes	No	0-2,5	Compare (SA) to (DA) and set appropriate status bits
S	0 1 1	0	Subtract	Yes	0-4	(DA) − (SA) → (DA)
SB	0 1 1	1	Subtract bytes	Yes	0-5	(DA) − (SA) → (DA)
SOC	1 1 1	0	Set ones corresponding	Yes	0-2	(DA) OR (SA) → (DA)
SOCB	1 1 1	1	Set ones corresponding bytes	Yes	0-2,5	(DA) OR (SA) → (DA)
SZC	0 1 0	0	Set zeroes corresponding	Yes	0-2	(DA) AND $\overline{(SA)}$ → (DA)
SZCB	0 1 0	1	Set zeroes corresponding bytes	Yes	0-2,5	(DA) AND $\overline{(SA)}$ → (DA)
MOV	1 1 0	0	Move	Yes	0-2	(SA) → (DA)
MOVB	1 1 0	1	Move bytes	Yes	0-2,5	(SA) → (DA)

3.5.2 Dual Operand Instructions with Multiple Addressing Modes for the Source Operand and Workspace Register Addressing for the Destination

General format



The addressing mode for the source operand is determined by the T_S field

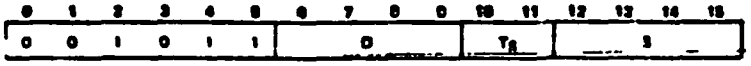
T_S	S	ADDRESSING MODE	NOTES
00	0, 1, ... 15	Workspace register	
01	0, 1, ... 15	Workspace register indirect	
10	0	Symbolic	
10	1, 2, ... 15	Indexed	1
11	0, 1, ... 15	Workspace register indirect auto increment	2

NOTES 1 Workspace register 0 may not be used for indexing
2 The workspace register is incremented by 2

MNEMONIC	OP CODE 0 1 2 3 4 5	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
COC	0 0 1 0 0 0	Compare ones corresponding	No	2	Test (D) to determine if 1's are in each bit position where 1's are in (SA). If so, set ST2
CZC	0 0 1 0 0 1	Compare zeros corresponding	No	2	Test (D) to determine if 0's are in each bit position where 1's are in (SA). If so, set ST2
XOR	0 0 1 0 1 0	Exclusive OR	Yes	0-2	(D) ⊙ (SA) → (D).
MPY	0 0 1 1 1 0	Multiply	No		Multiply unsigned (D) by unsigned (SA) and place unsigned 32-bit product in D (most significant) and D+1 (least significant). If WR15 is D, the next word in memory after WR15 will be used for the least significant half of the product.
DIV	0 0 1 1 1 1	Divide	No	4	If unsigned (SA) is less than or equal to unsigned (D), perform no operation and set ST4. Otherwise, divide unsigned (D) and (D+1) by unsigned (SA). Quotient → (D), remainder → (D+1). If D = 15, the next word in memory after WR15 will be used for the remainder.

3.5.3 Extended Operation (XOP) Instruction

General format



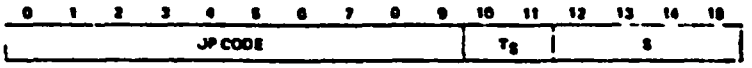
The T_S and S fields provide multiple mode addressing capability for the source operand. When the XOP is executed, ST6 is set and the following transfers occur:

$(40_{16} + 4D) \rightarrow (WP)$
$(42_{16} + 4D) \rightarrow (PC)$
SA → (new WR11)
(old WP) → (new WR13)
(old PC) → (new WR14)
(old ST) → (new WR15)

The TMS 9900 does not test interrupt requests ($\overline{INTREQ}$) upon completion of the XOP instruction.

3.5.4 Single Operand Instructions

General format



The T_S and S fields provide multiple mode addressing capability for the source operand.

MNEMONIC	OP CODE (0 1 2 3 4 5 6 7 8 9)	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
B	0 0 0 0 0 1 0 0 0 1	Branch	No	–	SA → (PC)
BL	0 0 0 0 0 1 1 0 1 0	Branch and link	No	–	(PC) → (WR11) SA → (PC)
BLWP	0 0 0 0 0 1 0 0 0 0	Branch and load workspace pointer	No	–	(SA) → (WP), (SA+2) → (PC), (old WP) → (new WR13) (old PC) → (new WR14), (old ST) → (new WR15) the interrupt input ($\overline{INTREQ}$) is not tested upon completion of the BLWP instruction
CLR	0 0 0 0 0 1 0 5 1 1	Clear operand	No	–	0 → (SA)
SETO	0 0 0 0 1 1 1 0 0	Set to ones	No	–	$FFFF_{16}$ → (SA)
INV	0 0 0 0 0 1 0 1 0 1	Invert	Yes	0-2	$\overline{(SA)}$ → (SA)
NEG	0 0 0 0 0 1 0 1 0 0	Negate	Yes	0-4	–(SA) → (SA)
ABS	0 0 0 0 0 1 1 1 0 1	Absolute value*	No	0-4	(SA) → (SA)
SWPB	0 0 0 0 0 1 1 0 1 1	Swap bytes	No	–	(SA) bits 0 thru 7 → (SA) bits 8 thru 15 (SA) bits 8 thru 15 → (SA) bits 0 thru 7
INC	0 0 0 0 0 1 0 1 1 0	Increment	Yes	0-4	(SA) + 1 → (SA)
INCT	0 0 0 0 0 1 0 1 1 1	Increment by two	Yes	0-4	(SA) + 2 → (SA)
DEC	0 0 0 0 0 1 1 0 0 0	Decrement	Yes	0-4	(SA) – 1 → (SA)
DECT	0 0 0 0 0 1 1 0 0 1	Decrement by two	Yes	0-4	(SA) – 2 → (SA)
X†	0 0 0 0 0 1 0 0 1 0	Execute	No	–	Execute the instruction at SA

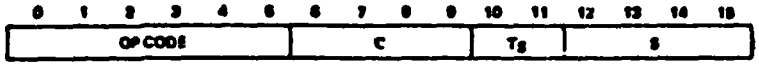
* operand is compared to zero for status bit

† additional memory words for the execute instruction are required to define the operands of the instruction located at SA. these words will be accessed from PC and the PC will be updated accordingly. The instruction acquisition signal (IAQ) will not be true when the TMS 9900 executes the instruction at SA. Status bits are affected in the normal manner for the instruction executed.

1978

3.5.5 CRU Multiple-Bit Instructions

General format



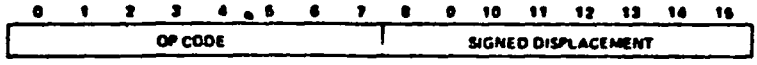
The C field specifies the number of bits to be transferred. If C = 0, 16 bits will be transferred. The CRU base register (WR12, bits 3 through 14) defines the starting CRU bit address. The bits are transferred serially and the CRU address is incremented with each bit transfer, although the contents of WR12 is not affected. T_S and S provide multiple mode addressing capability for the source operand. If 8 or fewer bits are transferred (C = 1 through 8), the source address is a byte address. If 9 or more bits are transferred (C = 0, 9 through 15), the source address is a word address. If the source is addressed in the workspace register indirect auto increment mode, the workspace register is incremented by 1 if C = 1 through 8, and is incremented by 2 otherwise.

MNEMONIC	OP CODE 0 1 2 3 4 5	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
LDCR	0 0 1 1 0 0	Load communication register	Yes	0-3,5[1]	Beginning with LSB of (SA), transfer the specified number of bits from (SA) to the CRU
STCR	0 0 1 1 0 1	Store communication register	Yes	0-3,5[1]	Beginning with LSB of (SA), transfer the specified number of bits from the CRU to (SA). Load unfilled bit positions with 0

[1] ST5 is affected only if 1 ≤ C ≤ 8.

3.5.6 CRU Single-Bit Instructions

General format

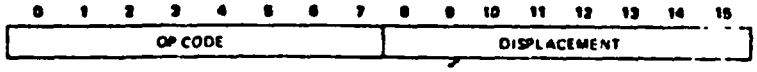


CRU relative addressing is used to address the selected CRU bit.

MNEMONIC	OP CODE 0 1 2 3 4 5 6 7	MEANING	STATUS BITS AFFECTED	DESCRIPTION
SBO	0 0 0 1 1 1 0 1	Set bit to one	-	Set the selected CRU output bit to 1
SBZ	0 0 0 1 1 1 1 0	Set bit to zero	-	Set the selected CRU output bit to 0
TB	0 0 0 1 1 1 1 1	Test bit	2	If the selected CRU input bit = 1, set ST2

3.5.7 Jump Instructions

General format

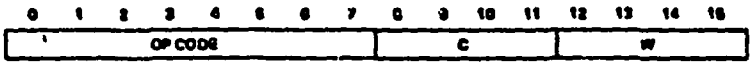


Jump instructions cause the PC to be loaded with the value selected by PC relative addressing if the bits of ST are at specified values. Otherwise, no operation occurs and the next instruction is executed since PC points to the next instruction. The displacement field is a word count to be added to PC. Thus, the jump instruction has a range of -128 to 127 words from memory word address following the jump instruction. No ST bits are affected by jump instruction.

MNEMONIC	OP CODE 0 1 2 3 4 5 6 7	MEANING	ST CONDITION TO LOAD PC
JEQ	0 0 0 1 0 0 1 1	Jump equal	ST2 = 1
JGT	0 0 0 1 0 1 0 1	Jump greater than	ST1 = 1
JH	0 0 0 1 1 0 1 1	Jump high	ST0 = 1 and ST2 = 0
JHE	0 0 0 1 0 1 0 C	Jump high or equal	ST0 = 1 or ST2 = 1
JL	0 0 0 1 1 0 1 0	Jump low	ST0 = 0 and ST2 = 0
JLE	0 0 0 1 0 0 1 0	Jump low or equal	ST0 = 0 or ST2 = 1
JLT	0 0 0 1 0 0 0 1	Jump less than	ST1 = 0 and ST2 = 0
JMP	0 0 0 1 0 0 0 0	Jump unconditional	unconditional
JNC	0 0 0 1 0 1 1 1	Jump no carry	ST3 = 0
JNE	0 0 0 1 0 1 1 0	Jump not equal	ST2 = 0
JNO	0 0 0 1 1 0 0 1	Jump no overflow	ST4 = 0
JOC	0 0 0 1 1 0 0 0	Jump on carry	ST3 = 1
JOP	0 0 0 1 1 1 0 0	Jump odd parity	ST5 = 1

3.5.8 Shift Instructions

General format:

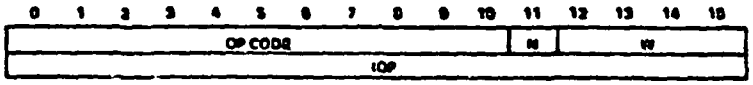


If C = 0, bits 12 through 15 of WR0 contain the shift count. If C = 0 and bits 12 through 15 of WR0 = 0, the shift count is 16.

MNEMONIC	OP CODE 0 1 2 3 4 5 6 7	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
SLA	0 0 0 0 1 0 1 0	Shift left arithmetic	Yes	0-4	Shift (W) left. Fill vacated bit positions with 0.
SRA	0 0 0 0 1 0 0 0	Shift right arithmetic	Yes	0-3	Shift (W) right. Fill vacated bit positions with original MSB of (W)
SRC	0 0 0 0 1 0 1 1	Shift right circular	Yes	0-3	Shift (W) right. Shift previous LSB into MSB.
SRL	0 0 0 0 1 0 0 1	Shift right logical	Yes	0-3	Shift (W) right. Fill vacated bit positions with 0's

3.5.9 Immediate Register Instructions

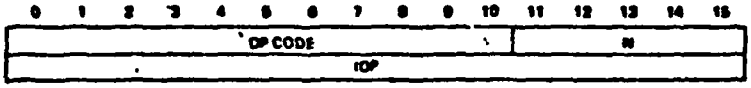
General format:



MNEMONIC	OP CODE 0 1 2 3 4 5 6 7 8 9 10	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
AI	0 0 0 0 0 0 1 0 0 0 1	Add immediate	Yes	0-4	(W) + IOP → (W)
ANDI	0 0 0 0 0 0 1 0 0 1 0	AND immediate	Yes	0-2	(W) AND IOP → (W)
CI	0 0 0 0 0 0 1 0 1 0 0	Compare immediate	Yes	0-2	Compare (W) to IOP and set appropriate status bits
LI	0 0 0 0 0 0 1 0 0 0 0	Load immediate	Yes	0-2	IOP → (W)
ORI	0 0 0 0 0 0 1 0 0 1 1	OR immediate	Yes	0-2	(W) OR IOP → (W)

3.5.10 Internal Register Load Immediate Instructions

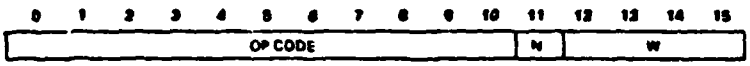
General format:



MNEMONIC	OP CODE (0 1 2 3 4 5 6 7 8 9 10)	MEANING	DESCRIPTION
LWPI	0 0 0 0 0 0 0 1 0 1 1 1	Load workspace pointer immediate	IOP → (WP) no ST bits affected
LIMI	0 0 0 0 0 0 0 1 1 0 0 0	Load interrupt mask	IOP, bits 12 thru 15 → ST12 thru ST15

3.5.11 Internal Register Store Instructions

General format

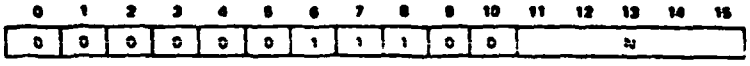


No ST bits are affected.

MNEMONIC	OP CODE (0 1 2 3 4 5 6 7 8 9 10)	MEANING	DESCRIPTION
STST	0 0 0 0 0 0 1 0 1 1 0	Store status register	(ST) → (W)
STWP	0 0 0 0 0 0 1 0 1 0 1	Store workspace pointer	(WP) → (W)

3.5.12 Return Workspace Pointer (RTWP) Instruction

General format

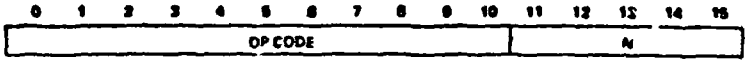


The RTWP instruction causes the following transfers to occur

(WR15) → (ST)
(WR14) → (PC)
(WR13) → (WP)

3.5.13 External Instructions

General format



External instructions cause the three most significant address lines (A0 through A2) to be set to the below described levels and the CRUCLK line to be pulsed, allowing external control functions to be initiated

MNEMONIC	OP CODE (0 1 2 3 4 5 6 7 8 9 10)	MEANING	STATUS BITS AFFECTED	DESCRIPTION	ADDRESS BUS A0	A1	A2
IDLE	0 0 0 0 0 0 1 1 0 1 0	Idle	-	Suspend TMS 9900 instruction execution until an interrupt LOAD or RESET occurs	L	H	L
RSET	0 0 0 0 0 0 1 1 0 1 1	Reset	12-15	0 → ST12 thru ST15	L	H	H
CRUF	0 0 0 0 0 0 1 1 1 1 0	User defined		---	H	H	L
CRUN	0 0 0 0 0 0 1 1 1 0 1	User defined		---	H	L	H
UNEX	0 0 0 0 0 0 1 1 1 1 1	User defined		---	H	H	H

Note

3.6 TMS 9900 INSTRUCTION EXECUTION TIMES

Instruction execution times for the TMS 9900 are a function of:

1) Clock cycle time, $t_{c(\phi)}$

2) Addressing mode used where operands have multiple addressing mode capability

3) Number of wait states required per memory access.

Table 3 lists the number of clock cycles and memory accesses required to execute each TMS 9900 instruction. For instructions with multiple addressing modes for either or both operands, the table lists the number of clock cycles and memory accesses with all operands addressed in the workspace-register mode. To determine the additional number of clock cycles and memory accesses required for modified addressing, add the appropriate values from the referenced tables. The total instruction-execution time for an instruction is

$$T = t_{c(\phi)} \quad (C + W \cdot M)$$

where

T = total instruction execution time;
$t_{c(\phi)}$ = clock cycle time,
C = number of clock cycles for instruction execution plus address modification,
W = number of required wait states per memory access for instruction execution plus address modification,
M = number of memory accesses.

TABLE 3
INSTRUCTION EXECUTION TIMES

INSTRUCTION	CLOCK CYCLES C	MEMORY ACCESS M	ADDRESS MODIFICATION† SOURCE	ADDRESS MODIFICATION† DEST
A	14	4	A	A
AB	14	4	B	B
ABS (MSB = 0)	12	2	A	-
(MSB = 1)	14	3	A	
AI	14	4		
ANDI	14	4	-	
B	8	2	A	
BL	12	3	A	
BLWP	26	6	A	-
C	14	3	A	A
CB	14	3	B	B
CI	14	3	-	
CKOF	12	1	-	-
CKON	12	1	-	-
CLR	10	3	A	
COC	14	3	A	
CZC	14	3	A	
DEC	10	3	A	
DECT	10	3	A	
DIV (ST4 is set)	16	3	A	
DIV (ST4 is reset)*	92-124	6	A	
IDLE	12	1		
INC	10	3	A	-
INCT	10	3	A	
INV	10	3	A	-
Jump (PC is changed)	10	1	-	
(PC is not changed)	8	1		
LDCR (C = 0)	52	3	A	
(1 ≤ C ≤ 8)	20+2C	3	B	
(9 ≤ C ≤ 15)	20+2C	3	A	
LI	12	3		
LIMI	16	2	-	
LREX	12	1		
RESET function	26	5		
LOAD function	22	5		
Interrupt context switch	22	5		
LWPI	10	2		
MOV	14	4	A	A
MOVB	14	4	B	B
MPY	52	5	A	-
NEG	12	3	A	-
ORI	14	4	-	-
RSET	12	1	-	-
RTWP	14	4	-	-
S	14	4	A	A
SB	14	4	B	B
SBO	12	2	-	-
SBZ	12	2	-	-
SETO	10	3	A	-
Shift (C ≠ 0)	12+2C	3	-	-
(C = 0, Bits 12-15 of WR0 = 0)	52	4	-	
(C = 0, Bits 12-15 of WR0 = N ≠ 0)	20+2N	4	-	-
SOC	14	4	A	A
SOCB	14	4	B	B
STCR (C = 0)	60	4	A	
(1 ≤ C ≤ 7)	42	4	B	
(C = 8)	44	4	B	
(9 ≤ C ≤ 15)	58	4	A	-
STST	8	2	-	-
STWP	8	2	-	-
SWPB	10	3	A	
SZC	14	4	A	A
SZCB	14	4	B	B
TB	12	2		
X**	8	2	A	
XOP	36	8	A	
XOR	14	4	A	
Undefined op codes: 0000-01FF, 0320-033F, 0C00-0FFF, 0780-07FF	6	1		

*Execution time is dependent upon the partial quotient after each clock cycle during execution.
**Execution time is added to the execution time of the instruction located at the source address minus 4 clock cycles and 1 memory access time.
†The letters A and B refer to the respective tables that follow.

ADDRESS MODIFICATION – TABLE A

ADDRESSING MODE	CLOCK CYCLES C	MEMORY ACCESSES M
WR (T_S or T_D = 00)	0	0
WR indirect (T_S or T_D = 01)	4	1
WR indirect auto-increment (T_S or T_D = 11)	8	2
Symbolic (T_S or T_D = 10, S or D = 0)	8	1
Indexed (T_S or T_D = 10, S or D = 0)	8	2

ADDRESS MODIFICATION – TABLE B

ADDRESSING MODE	CLOCK CYCLES C	MEMORY ACCESSES M
WR (T_S or T_D = 00)	0	0
WR indirect (T_S or T_D = 01)	4	1
WR indirect auto-increment (T_S or T_D = 11)	8	2
Symbolic (T_S or T_D = 10, S or D = 0)	8	1
Indexed (T_S or T_D = 10, S or D = 0)	8	2

As an example, the instruction MOVB is used in a system with $t_{c(0)} = 0.333\ \mu s$ and no wait states are required to access memory. Both operands are addressed in the workspace register mode.

$$T = t_{c(0)}\,(C + W \cdot M) = 0.333\,(14 + 0 \cdot 4)\ \mu s = 4.662\ \mu s$$

If two wait states per memory access were required, the execution time is

$$T = 0.333\,(14 + 2 \cdot 4)\ \mu s = 7.326\ \mu s.$$

If the source operand was addressed in the symbolic mode and two wait states were required

$$T = t_{c(0)}\,(C + W \cdot M)$$
$$C = 14 + 8 = 22$$
$$M = 4 + 1 = 5$$
$$T = 0.333\,(22 + 2 \cdot 5)\ \mu s = 10.656\ \mu s.$$

4. TMS 9900 ELECTRICAL AND MECHANICAL SPECIFICATIONS

4.1 ABSOLUTE MAXIMUM RATINGS OVER OPERATING FREE-AIR TEMPERATURE RANGE (UNLESS OTHERWISE NOTED)*

Supply voltage, V_{CC} (see Note 1)	−0.3 to 20 V
Supply voltage, V_{DD} (see Note 1)	−0.3 to 20 V
Supply voltage, V_{SS} (see Note 1)	−0.3 to 20 V
All input voltages (see Note 1)	−0.3 to 20 V
Output voltage (with respect to V_{SS})	−2 V to 7 V
Continuous power dissipation	1.2 W
Operating free-air temperature range	0 C to 70 C
Storage temperature range	−55 C to 150 C

*Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: Under absolute maximum ratings voltage values are with respect to the most negative supply V_{BB} (substrate) unless otherwise noted. Throughout the remainder of this section voltage values are with respect to V_{SS}.

Am2903

The following material describes the Am2903 four-bit bipolar microprocessor slice presently being sampled by Advanced Micro Devices. The device is an extension of Am2901A and should be in production later this year.

THE NEXT GENERATION FOUR-BIT BIPOLAR MICROPROCESSOR SLICE -- THE Am2903

Vernon Coleman
Michael W. Economidis
William J. Harmon Jr.

Advanced Micro Devices
901 Thompson Place
Sunnyvale, CA. 94086

INTRODUCTION

The Am2903 is the next generation bipolar microprocessor slice. The Am2903 performs all functions performed by the industry standard Am2901A and, in addition, provides a number of significant enhancements that are especially useful in arithmetic-oriented processors. Infinitely expandable memory and three-port three-address architecture are provided by the Am2903. In addition to a complete arithmetic and logic instruction set, the Am2903 provides a special set of instructions which facilitate the implementation of multiplication, division, normalization, and other previously time consuming operations.

OUTSTANDING FEATURES

Expandable Register File - Like the Am2901A, the Am2903 contains 16 internal working registers arranged in a two-address architecture. But the Am2903 includes the necessary "hooks" to expand the register file externally to any number of registers.

Built-in Multiplication Logic - Performing multiplication with the Am2901A requires a few external gates--these gates are contained on-chip in the Am2903. Three special instructions are used for unsigned multiplication, two's complement multiplication, and the last cycle of a two's complement multiplication.

Built-in Division Logic - The Am2903 contains all logic and interconnects for execution of a non-restoring, multiple-length division with correction of the quotient.

Built-in Normalization Logic - The Am2903 can simultaneously shift the Q Register and output in a working register. Thus, the mantissa and exponent of a floating point number can be developed using a single microcycle per shift. Status flags indicate when the operation is complete.

Built-in Parity Generation Circuitry - The Am2903 can supply parity across the entire ALU output for use in error detection and CRC code generation.

Built-in Sign Extension Circuitry - To facilitate operation on different length two's complement numbers, the Am2903 provides the capability to extend the sign at any slice boundary.

ARCHITECTURE OF THE Am2903

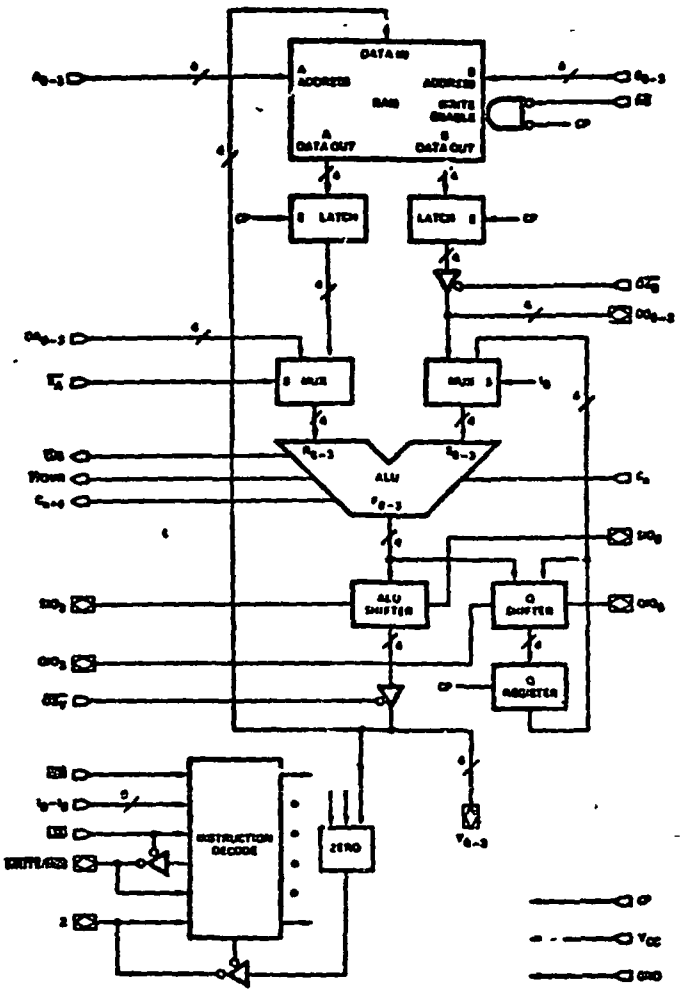
The Am2903 is a high-performance, cascadable, four-bit bipolar microprocessor slice designed for use in CPU's, peripheral controllers, microprogrammable machines, and numerous other applications. The microinstruction flexibility of the Am2903 allows the efficient emulation of almost any digital computing machine. The nine-bit microinstruction selects the ALU sources, function, and destination. The Am2903 is cascadable with full lookahead or ripple carry, has three-state outputs, and provides various ALU status flag outputs. Advanced Low-Power Schottky processing is used to fabricate this 48-pin LSI circuit.

All data paths within the device are four bits wide. As shown in the block diagram, the device consists of a 16-word by 4-bit, two-port RAM with latches on both output ports, a high-performance ALU and shifter, a multi-purpose Q Register with shifter input, and a nine-bit instruction decoder.

Two-Port RAM

Any two RAM words addressed at the A and B address ports can be read simultaneously at the respective RAM A and B output ports. Identical data appears at the two output ports when the same address is applied to both address

Am2903 BLOCK DIAGRAM



ports. The latches at the RAM output ports are transparent when the clock input, CP, is HIGH and they hold the RAM output data when CP is LOW. Under control of the $\overline{OE_B}$ three-state output enable, RAM data can be read directly at the Am2903 DB I/O port.

External data at the Am2903 Y I/O port can be written directly into the RAM, or ALU shifter output data can be enabled onto the Y I/O port and entered into the RAM. Data is written into the RAM at the B address when the write enable input, $\overline{WE}$, is LOW and the clock input, CP, is LOW.

Arithmetic Logic Unit

The Am2903 high-performance ALU can perform seven arithmetic and nine logic operations on two 4-bit operands. Multiplexers at the ALU inputs provide the capability to select various pairs of ALU source operands. The $\overline{E_A}$ input selects either the DA external data input or RAM output port A for use as one ALU operand and the $\overline{OE_B}$ and I_0 inputs select RAM output port B, DB external data input, or the Q Register content for use as the second ALU operand. Also, during some ALU operations, zeroes are forced at the ALU operand inputs. Thus, the Am2903 ALU can operate on data from two external sources, from an internal and external source, or from two internal sources. Table I shows all possible pairs of ALU source operands as a function of the $\overline{E_A}$, $\overline{OE_B}$, and I_0 inputs.

When instruction bits I_4, I_3, I_2, I_1,

ALU OPERAND SOURCES

$\overline{E}_A$	I_0	$\overline{OE}_B$	ALU OPERAND R	ALU OPERAND S
L	L	L	RAM Output A	RAM Output B
L	L	H	RAM Output A	DB_{0-3}
L	H	X	RAM Output A	Q Register
H	L	L	DA_{0-3}	RAM Output B
H	L	H	DA_{0-3}	DB_{0-3}
H	H	X	DA_{0-3}	Q Register

L = LOW H = HIGH X = Don't Care

TABLE I

and I_0 are LOW, the Am2903 executes special functions. Table 4 defines these special functions and the operation which the ALU performs for each. When the Am2903 executes instructions other than the nine special functions, the ALU operation is determined by instruction bits I_4, I_3, I_2, and I_1. Table 2 defines the ALU operation as a function of these four instruction bits.

Am2903's may be cascaded in either a ripple carry or lookahead carry fashion. When a number of Am2903's are cascaded, each slice must be programmed to be a most significant slice (MSS), intermediate slice (IS), or least significant slice (LSS) of

ALU FUNCTIONS

$I_4I_3I_2I_1$	Hex Code	ALU Functions
L L L L	0	I_0=L: Special Functions; I_0=H: F_i=HIGH
L L L H	1	F=S Minus R Minus 1 Plus C_n
L L H L	2	F=R Minus S Minus 1 Plus C_n
L L H H	3	F=R Plus S Plus C_n
L H L L	4	F=S Plus C_n
L H L H	5	F=$\overline{S}$ Plus C_n
L H H L	6	F=R Plus C_n
L H H H	7	F=$\overline{R}$ Plus C_n
H L L L	8	F_i=LOW
H L L H	9	F_i=$\overline{R}_i$ AND S_i
H L H L	A	F_i=$\overline{R}_i$ EXCLUSIVE OR S_i
H L H H	B	F_i=R_i EXCLUSIVE OR S_i
[illegible]	C	F_i=[illegible]
[illegible]	D	F_i=R_i NOR S_i
[illegible]	E	F_i=R_i NAND S_i
[illegible]	F	F_i=F_i OR S_i

L = LOW H = HIGH i = 0 to 3

TABLE 2

the array. The carry generate, $\overline{G}$, and carry propagate, $\overline{P}$, signals required for a lookahead carry scheme are generated by the Am2903 and are available as outputs of the least significant and intermediate slices.

The Am2903 also generates a carry-out signal, C_{n+4}, which is generally available as an output of each slice. Both the carry-in, C_n, and carry-out, C_{n+4}, signals are active HIGH. The ALU generates three other status outputs. These are sign, S; overflow, OVR; and zero, Z. The S output is generally the most significant (sign) bit of the ALU output and can be used to determine positive or negative results. The OVR output indicates that the arithmetic operation being performed exceeds the available two's complement number range. The S and OVR signals are available as outputs of the most significant slice. Thus, the multi-purpose $\overline{G}$/S and $\overline{P}$/OVR outputs indicate $\overline{G}$ and $\overline{P}$ at the least significant and intermediate slices, and sign and overflow at the most significant slice. Z is an open-collector input/output pin and can be wire OR'ed between slices. As an output, it generally indicates that the Y_{0-3} outputs are all LOW and can be used as a zero detect status flag. To some extent, the meaning of the C_{n+4}, $\overline{P}$/OVR, $\overline{G}$/S, and Z signals vary with the ALU function being performed. Refer to Table 5 for an exact definition of these four signals as a function of the Am2903 instruction.

ALU Shifter

Under instruction control, the ALU shifter passes the ALU output (F) non-shifted, shifts it up one bit position (2F), or shifts it down one bit position (F/2). Both arithmetic and logical shift operations are possible. An arithmetic shift operation shifts data around the most significant (sign) bit position of the most significant slice, and a logical shift operation shifts data through this bit position (see Figure A). SIO_0 and SIO_3 are bidirectional serial shift inputs/outputs. During a shift-up operation, SIO_0 is generally a serial shift input and SIO_3 a serial shift output. During a shift-down operation, SIO_3 is generally a serial shift input and SIO_0 a serial shift output.

To some extent, the meaning of the SIO_0 and SIO_3 signals is instruction

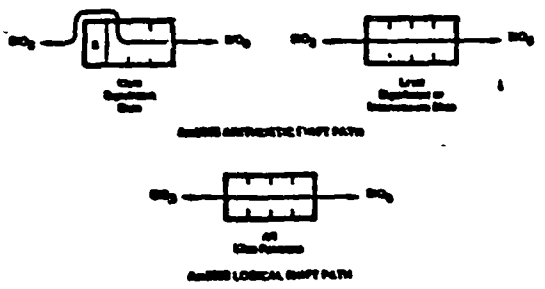


Figure A.

dependent. Refer to Tables 3 and 4 for an exact definition of these pins.

The ALU shifter also provides the capability to sign extend at slice boundaries. Under instruction control, the SIO_0 (sign) input can be extended through Y0, Y1, Y_2, Y_3 and propagated to the SIO_3 output.

A cascadable, five-bit parity generator/checker is designed into the Am2903 ALU shifter and provides ALU error detection capability. Parity for the F_0, F_1, F_2, F_3 ALU outputs and SIO_3 input is generated and, under instruction control, is made available at the SIO_0 output. Refer to the Am2903 applications section for a more detailed description of the Am2903 sign extension and parity generation/checking capability.

The instruction inputs determine the ALU shifter operation. Table 4 defines the special functions and the operation the ALU shifter performs for each. When the Am2903 executes instructions other than the nine special functions, the ALU shifter operation is determined by instruction bits $I_8I_7I_6I_5$. Table 3 defines the ALU shifter operation as a function of these four bits.

Q Register

The Q Register is an auxiliary four-bit register which is clocked on the LOW-to-HIGH transition of the CP input. It is intended primarily for use in multiplication and division operations; however, it can also be used as an accumulator or holding register for some applications. The ALU output, F, can be loaded into the Q Register, and/or the Q Register can be selected as the source for the ALU S operand. The shifter at the input to the Q Register provides the capability to shift the Q Register contents up one bit position (2Q) or down one bit

$I_8I_7I_6I_5$	Hex Code	ALU Shifter Function	SIO_0	SIO_3 Most Sig. Slice	SIO_3 Other Slices	$\overline{WRITE}$	Q Reg & Shifter Function	QIO_0	QIO_3
L L L L	0	Arith. F/2→Y	F_0	Input	Input	L	Hold	Z	Z
L L L H	1	Log. F/2→Y	F_0	Input	Input	L	Hold	Z	Z
L L H L	2	Arith. F/2→Y	F_0	Input	Input	L	Log. Q/2→Q	Q_0	Input
L L H H	3	Log. F/2→Y	F_0	Input	Input	L	Log. Q/2→Q	Q_0	Input
L H L L	4	F→Y	Parity	Input	Input	L	Hold	Z	Z
L H L H	5	F→Y	Parity	Input	Input	H	Log. Q/2→Q	Q_0	Input
L H H L	6	F→Y	Parity	Input	Input	H	F→Q	Z	Z
L H H H	7	F→Y	Parity	Input	Input	L	F→Q	Z	Z
H L L L	8	Arith. 2F→Y	Input	F_2	F_3	L	Hold	Z	Z
H L L H	9	Log. 2F→Y	Input	F_3	F_3	L	Hold	Z	Z
H L H L	A	Arith. 2F→Y	Input	F_2	F_3	L	Log. 2Q→Q	Input	Q_3
H L H H	B	Log. 2F→Y	Input	F_3	F_3	L	Log. 2Q→Q	Input	Q_3
H H L L	C	F→Y	Z	F_3	F_3	H	Hold	Z	Z
H H L H	D	F→Y	Z	F_3	F_3	H	Log. 2Q→Q	Input	Q_3
H H H L	E	SIO_0→Y0,Y1,Y2,Y3	Input	SIO_0	SIO_0	L	Hold	Z	Z
H H H H	F	F→Y	Z	F_3	F_3	L	Hold	Z	Z

Parity = $F_3 \forall F_2 \forall F_1 \forall F_0 \forall SIO_3$ ∀ = Exclusive OR

L = LOW H = HIGH Z = High Impedance

TABLE 3: ALU DESTINATION CONTROL FOR I_0 or I_1 or I_2 or I_3 or I_4 = HIGH, $\overline{CP}$ = LOW

I_8	I_7	I_6	I_5	Hex Code	Special Function	ALU Function	ALU Shifter Function	SIO_0	SIO_3 Most Sig. Slice	SIO_3 Other Slices	Q Reg & Shifter Function	QIO_0	QIO_3	$\overline{WRITE}$
L	L	X		0, 1	Unsigned Multiply	F=S+C_n if Z=L F=R+S+C_n if Z=H	Log. F/2→Y (Note 1)	F_0	Z	Input	Log. Q/2→Q	Q_0	Input	L
L	L	X		2,3	Two's Complement Multiply	F=S+C_n if Z=L F=R+S+C_n if Z=H	Log. F/2→Y (Note 2)	F_0	Z	Input	Log. Q/2→Q	Q_0	Input	L
L	H	L		4	Increment by One or Two	F=S+1+C_n	F→Y	Parity	Input	Input	Hold	Z	Z	L
L	H	H		5	Sign/Magnitude-Two's Complement	F=S+C_n if Z=L F=S+C_n if Z=H	F→Y (Note 3)	Parity	Input	Input	Hold	Z	Z	L
L	H	X		6, 7	Two's Complement Multiply, Correction	F=S+C_n if Z=L F=S-R-1+C_n if Z=H	Log. F/2→Y (Note 2)	F_0	Z	Input	Log. Q/2→Q	Q_0	Input	L
H	L			8, 9	Single Length Normalize	F=S+C_n	F→Y	Z	F_3	F_3	Log. 2Q→Q	Input	Q_3	L
H	L	X		A, B	Double Length Normalize and First Divide Op.	F=S+C_n	Log. 2F→Y	Input	$R_3 \forall F_3$	F_3	Log. 2Q→Q	Input	Q_3	L
H	H	L		C, D	Two's Complement Divide	F=S+R+C_n if Z=L F=S-R-1+C_n if Z=H	Log. 2F→Y	Input	$\overline{R_3 \forall F_3}$	F_3	Log. 2Q→Q	Input	Q_3	L
H	H	H		E, F	Two's Complement Divide, Correction and Remainder	F=S+F+C_n if Z=L F=S-R-1+C_n if Z=H	F→Y	Z	F_3	F_3	Log. 2Q→Q	Input	Q_3	L

L = LOW H = HIGH X = Don't Care Z = High Impedance $\forall$ = Exclusive OR Parity = $SIO_3 \forall F_3 \forall F_2 \forall F_1 \forall F_0$

NOTE 1: At the most significant slice only, the C_{n+4} signal is internally gated to the Y_3 output.

NOTE 2: At the most significant slice only, $F_3 \forall OVR$ is internally gated to the Y_3 output.

NOTE 3: At the most significant slice only, $S_3 \forall OVR$ is generated at the Y_3 output.

TABLE 4

SPECIAL FUNCTIONS: $I_0 = I_1 = I_2 = I_3 = I_4 = LOW$, $\overline{IEN} = LOW$

position (Q/2). Only logical shifts are performed. QIO_0 and QIO_3 are bidirectional shift serial inputs/outputs. During a Q Register shift-up operation, QIO_0 is a serial shift input and QIO_3 is a serial shift output. During a shift-down operation, QIO_3 is a serial shift input and QIO_0 is a serial shift output.

Double-length arithmetic and logical shifting capability is provided by the Am2903. The double-length shift is performed by connecting QIO_3 of the most significant slice to SIO_0 of the least significant slice, and executing an instruction which shifts both the ALU output and the Q Register.

The Q Register and shifter are controlled by instruction inputs. Table 3 defines the Am2903 special functions and the operations which the Q register and shifter perform for each. The Am2903 executes instructions other than the nine special functions. Q Register and shifter operation is controlled by instruction bits $I_8I_7I_6I_5$. Table 3 defines the Q Register and shifter operation as a function of these four bits.

Output Buffers

The DB and Y ports are bidirectional I/O ports driven by three-state output buffers with external output enable controls. The Y output buffers are enabled when the $\overline{OE_Y}$ input is LOW and are in the high-impedance state when $\overline{OE_Y}$ is HIGH. Likewise, the DB output buffers are enabled when the $\overline{OE_B}$ is LOW and in the high-impedance state when $\overline{OE_B}$ is HIGH.

Instruction Decoder

The Instruction Decoder generates required internal control signals as a function of the nine instruction inputs, I_{0-8}; the Instruction Enable input, $\overline{IEN}$; the $\overline{LSS}$ input; and the Shift-MSB input/output.

The $\overline{WRITE}$ output is LOW when an instruction which writes data into

the RAM is being executed. Refer to Tables 3 and 4 for a definition of the $\overline{WRITE}$ output as a function of the Am2903 instruction inputs.

When $\overline{IEN}$ is HIGH, the $\overline{WRITE}$ output is forced HIGH and the Q Register and Sign Compare Flip-Flop contents are preserved. When $\overline{IEN}$ is LOW, the $\overline{WRITE}$ output is enabled and the Q Register and Sign Compare Flip-Flop can be written according to the Am2903 instruction. The Sign Compare Flip-Flop is an on-chip flip-flop which is used during an Am2903 divide operation (see Figure B).

Programming the Am2903 Slice Position

Tying the $\overline{LSS}$ input LOW programs the slice to operate as a least significant slice (LSS) and enables the $\overline{WRITE}$ output signal onto the $\overline{WRITE}/\overline{MSS}$ bidirectional I/O pin. When $\overline{LSS}$ is tied HIGH, the $\overline{WRITE}/\overline{MSS}$ pin becomes an input pin; tying the $\overline{WRITE}/\overline{MSS}$ pin HIGH programs the slice to operate as an intermediate slice (IS) and tying it LOW programs the slice to operate as a most significant slice (MSS).

(Hex) $I_8I_7I_6I_5$	(Hex) $I_4I_3I_2I_1$	I_0	G_i (i=0 to 3)	P_i (i=0 to 3)	C_{n+4}	F/OVR Most Sig. Slice	F/OVR Other Slices	Z/S Most Sig. Slice	Z/S Other Slices	Y Most Sig. Slice	Y Intermediate Slice	Y Least Sig. Slice
X	0	H	0	0	0	0	0	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	1	X	$\overline{R}_iAS_i$	$\overline{R}_iVS_i$	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	2	X	$R_i\overline{S}_i$	$R_i\overline{S}_i$	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	3	X	R_iAS_i	R_iVS_i	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	4	X	0	S_i	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	5	X	0	$\overline{S}_i$	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	6	X	0	R_i	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	7	X	0	$\overline{R}_i$	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	8	X	0	1	0	0	0	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	9	X	$\overline{R}_iAS_i$	1	0	0	0	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	A	X	$R_i\overline{S}_i$	R_iVS_i	0	0	0	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	B	X	$\overline{R}_iAS_i$	$\overline{R}_iVS_i$	0	0	0	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	C	X	$R_i\overline{S}_i$	1	0	0	0	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	D	X	$\overline{R}_i\overline{S}_i$	1	0	0	0	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	E	X	R_iAS_i	1	0	0	0	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
X	F	X	$\overline{R}_i\overline{S}_i$	1	0	0	0	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
0, 1	0	L	0 if Z=L R_iAS_i if Z=H	S_i if Z=L R_iVS_i if Z=H	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	Input	Input	Q_0
2, 3	0	L	0 if Z=L R_iAS_i if Z=H	S_i if Z=L R_iVS_i if Z=H	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	Input	Input	Q_0
4	0	L	See Note 1	See Note 2	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$	$\overline{Y}_0\overline{Y}_1\overline{Y}_2\overline{Y}_3$
5	0	L	0	S_i if Z=L S_i if Z=H	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3 if Z=L F_3VS_3 if Z=H	$\overline{G}$	S_3	Input	Input
6, 7	0	L	0 if Z=L $\overline{R}_i\overline{S}_i$ if Z=H	R_i if Z=L R_iVS_i if Z=H	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	Input	Input	Q_0
8, 9	0	L	0	S_i	See Note 3	$C_{n+3}VC_{n+4}$	$\overline{P}$	O_3	$\overline{G}$	$\overline{O}_0\overline{O}_1\overline{O}_2\overline{O}_3$	$\overline{O}_0\overline{O}_1\overline{O}_2\overline{O}_3$	$\overline{O}_0\overline{O}_1\overline{O}_2\overline{O}_3$
A, B	0	L	0	S_i	See Note 4	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	See Note 5	See Note 5	See Note 5
C, D	0	L	$R_i\overline{S}_i$ if Z=L $\overline{R}_i\overline{S}_i$ if Z=H	R_iVS_i if Z=L $\overline{R}_iVS_i$ if Z=H	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	Sign Compare FF output	Input	Input
E, F	0	L	$R_i\overline{S}_i$ if Z=L $\overline{R}_iAS_i$ if Z=H	R_iVS_i if Z=L $\overline{R}_iVS_i$ if Z=H	$GVPC_n$	$C_{n+3}VC_{n+4}$	$\overline{P}$	F_3	$\overline{G}$	Sign Compare FF output	Input	Input

L = LOW = 0 H = HIGH = 1 V = OR A = AND V = EXCLUSIVE OR

$P = P_3P_2P_1P_0$ $G = G_3VG_2P_3VG_1P_2P_3VG_0P_1P_2P_3$ $C_{n+3} = G_2VG_1P_2VG_0P_1P_2VC_nP_0P_1P_2$

NOTE 1: If $\overline{LSS}$ is LOW O_3 = S_3 and $O_{2,1,0}$ = 0
If $\overline{LSS}$ is HIGH, $O_{3,2,1,0}$ = $S_{3,2,1,0}$

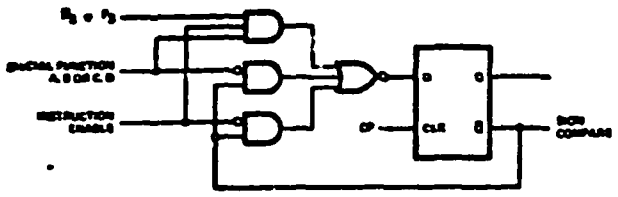
NOTE 2: If $\overline{LSS}$ is LOW P_3 = 1 and $P_{2,1,0}$ = $S_{2,1,0}$
If $\overline{LSS}$ is HIGH, $P_{3,2,1,0}$ = $S_{3,2,1,0}$

NOTE 3: At the most significant slice, C_{n+4} = G_3VC_n
At other slices, C_{n+4} = $GVPC_n$

NOTE 4: At the most significant slice, C_{n+4} = F_3VF_2
At other slices, C_{n+4} = $GVPC_n$

NOTE 5: $Z = \overline{O}_0\overline{O}_1\overline{O}_2\overline{O}_3F_0F_1F_2F_3$

TABLE 5: Am2903 STATUS OUTPUTS



The sign compare signal appears at the Z output of the most significant slice during special functions C, D and E, F. Refer to Table 5.

Figure 8. Sign Compare Flip-Flop

Am2903 SPECIAL FUNCTIONS

The Am2903 provides nine Special Functions which facilitate the implementation of the following operations:

- Single-and Double-Length Normalization
- Two's Complement Division
- Unsigned and Two's Complement Multiplication
- Conversion Between Two's Complement and Sign/Magnitude Representation
- Incrementation by One or Two

Table 4 defines these Special Functions.

The Single-Length and Double-Length Normalization functions can be used to adjust a single-precision or double-precision floating point number in order to bring its mantissa within a specified range.

Three Special Functions which can be used to perform a two's complement, non-restoring divide operation are provided by the Am2903. These functions provide both single- and double-precision divide operations and can be performed in "n" clock cycles, where "n" is the number of bits in the divisor.

The Unsigned Multiply Special Function and the two Two's Complement Multiply Special Functions can be used to multiply two "n" bit, unsigned or two's complement numbers, respectively, in "n" clock cycles. These functions utilize the conditional add and shift algorithm. During the last cycle of [illegible] complement multiplication, a conditional subtraction, rather than addition, is performed because the sign bit of the multiplier carries negative [illegible].

The Sign/Magnitude-Two's Complement Special Function can be used to convert number representation systems. A number expressed in Sign/Magnitude representation can be converted to the Two's Complement representation, and vice-versa, in one clock cycle.

The Increment by One or Two Special Function can be used to increment an unsigned or two's complement number by one or two. This is useful in 16-bit word, byte-addressable machines, where the word addresses are multiples of two.

Refer to Am2903 applications section for a more detailed description of these Special Functions.

PIN DEFINITIONS

A_{0-3} Four RAM address inputs which contain the address of the RAM word appearing at the RAM A output port.

B_{0-3} Four RAM address inputs which contain the address of the RAM word appearing at the RAM B output port and into which new data is written when the $\overline{WE}$ input and the CP input is LOW.

$\overline{WE}$ The RAM write enable input. If $\overline{WE}$ is LOW, data at the Y I/O port is written into the RAM when the CP input is LOW. When $\overline{WE}$ is HIGH, writing data into the RAM is inhibited.

DA_{0-3} A four-bit external data input which can be selected as one of the Am2903 ALU operand sources; DA_0 is the least significant bit.

$\overline{EA}$ A control input which, when HIGH, selects DA_{0-3} and, when LOW, selects RAM output A as the ALU R operand.

DB_{0-3} A four-bit external data input/output. Under control of the $\overline{OE_B}$ input, RAM output port B can be directly read on these lines, or input data on these lines can be selected as the ALU S operand.

$\overline{OE_B}$ A control input which, when LOW, enables RAM output B onto the DB_{0-3} lines and, when HIGH, disables the RAM output B tri-state buffers.

C_n The carry-in input to the Am2903 ALU.

I_{0-8} The nine instruction inputs used to select the Am2903 operation to be performed.

$\overline{IEN}$ The instruction enable input which, when LOW, enables the $\overline{WRITE}$ output and allows the Q Register and the Sign Compare flip-flop to be written. When $\overline{IEN}$ is HIGH, the $\overline{WRITE}$ output is forced HIGH and the Q Register and Sign Compare flip-flop are in the hold mode.

C_{n+4} This output generally indicates the carry-out of the Am2903 ALU. Refer to Table 5 for an exact definition of this pin.

$\overline{G}/S$ A multi-purpose pin which indicates the carry generate, $\overline{G}$, function at the least significant and intermediate slices, and generally indicates the sign, S, of the ALU result at the most significant slice. Refer to Table 5 for an exact definition of this pin.

$\overline{P}/OVR$ A multi-purpose pin which indicates the carry propagate, $\overline{P}$, function at the least significant and intermediate slices, and indicates the conventional two's complement overflow, OVR, signal at the most significant slice. Refer to Table 5 for an exact definition of this pin.

Z An open-collector input/output pin which, when HIGH, generally indicates the Y_{0-3} outputs are all LOW. For some Special Functions, Z is used as an input pin. Refer to Table 5 for an exact definition of this pin.

SIO_0, SIO_3 Bidirectional serial shift inputs/outputs for the ALU shifter. During a shift-up operation, SIO_0 is an input and SIO_3 an output. During a shift-down operation, SIO_3 is an input and SIO_0 is an output. Refer to Tables 3 and 4 for an exact definition of these pins.

QIO_0, QIO_3 Bidirectional serial shift inputs/outputs for the Q shifter which operate like SIO_0 and SIO_3. Refer to Tables 3 and 4 for an exact definition of these pins.

$\overline{LSS}$ An input pin which, when tied LOW, programs the chip to act as the least significant slice (LSS) of an Am2903 array and enables the $\overline{WRITE}$ output onto the $\overline{WRITE}/MSS$ pin. When $\overline{LSS}$ is tied HIGH, the chip is programmed to operate as either an intermediate or most significant slice and the $\overline{WRITE}$ output buffer is disabled.

$\overline{WRITE}/MSS$ When $\overline{LSS}$ is tied LOW, the $\overline{WRITE}$ output signal appears at this pin; the $\overline{WRITE}$ signal is LOW when an instruction which writes data into the RAM is being executed. When $\overline{LSS}$ is tied HIGH, $\overline{WRITE}/MSS$ is an input pin; tying it HIGH programs the chip to operate as an intermediate slice (IS) and tying it LOW programs the chip to operate as the most significant slice (MSS).

Y_{0-3} Four data inputs/outputs of the Am2903. Under control of the $\overline{OEy}$ input, the ALU shifter output data can be enabled onto these lines, or these lines can be used as data inputs when external data is written directly into the RAM.

$\overline{OEy}$ A control input which, when LOW, enables the ALU shifter output data onto the Y_{0-3} lines and, when HIGH, disables the Y_{0-3} three-state output buffers.

CP The clock input to the Am2903. The Q Register and Sign Compare flip-flop are clocked on the LOW-to-HIGH transition of the CP signal. When enabled by $\overline{WE}$, data is written in the RAM when CP is LOW.

Am2903 APPLICATIONS

The Am2903 is designed to be used in microprogrammed systems. Figure 1 illustrates a recommended architecture. The control and data inputs to the Am2903 normally will all come from registers clocked at the same time as the Am2903. The register inputs come from a ROM or PROM--the "microprogram store". This memory contains sequences of microinstructions which apply the proper control signals to the Am2903's and other circuits to execute the desired operation.

The address lines of the microprogram store are driven from the Am2910 Microprogram Sequencer. This device has facilities for storing an address,

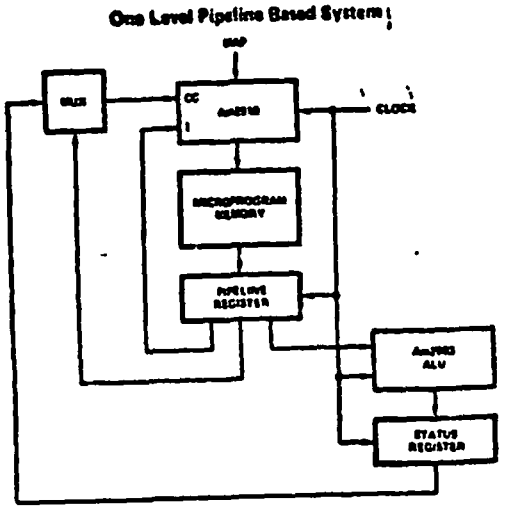


Figure 1. Typical Microprogram Architecture

incrementing an address, jumping to any address, and linking subroutines. The Am2910 is controlled by some of the bits coming from the microprogram store. Essentially, these bits are the "next instruction" control.

Note that with the microprogram register in between the microprogram memory store and the Am2903's, a microinstruction accessed on one cycle is executed on the next cycle. As one microinstruction is executed, the next microinstruction is being read from microprogram memory. In this configuration, system speed is improved because the execution time in the Am2903's occurs in parallel with the access time of the microprogram store. Without the "pipeline register", these two functions must occur serially.

Expansion of the Am2903

The Am2903 is a four-bit CPU slice. Any number of Am2903's can be interconnected to form CPU's of 8, 16, 32, or more bits, in four-bit increments. Figure 2 illustrates the interconnection of four Am2903's to form a 16-bit CPU, using ripple carry.

With the exception of the carry interconnection, all expansion schemes are the same. The QIO_3 and SIO_3 pins are bidirectional left/right shift lines at the MSB of the device. For all devices except the most significant, these lines are connected to the QIO_0 and SIO_0 pins of the adjacent more significant device. These connections allow the Q Registers of all Am2903's to be shifted left or right as a contiguous n-bit register, and also allow the ALU output data to be shifted left or right as a contiguous n-bit word prior to storage in the RAM. At the LSB and MSB of the CPU, the shift pins should be connected to a shift multiplexer which can be controlled by the microcode to select the appropriate input signals to the shift inputs.

Device 1 has been defined as the least significant slice (LSS) and its $\overline{LSS}$ pin has accordingly been grounded. The Write/Most Significant Slice ($\overline{WRITE}/\overline{MSS}$) pin of device 1 is now defined as being the Write output, which may now be used to drive the write enable ($\overline{WE}$) signal common to the four devices. Devices 2 and 3 are designated as intermediate slices and hence the $\overline{LSS}$ and $\overline{WRITE}/\overline{MSS}$ pins are tied HIGH. Device 4 is designated the most significant slice (MSS) with the $\overline{LSS}$ pin tied HIGH and the $\overline{WRITE}$/

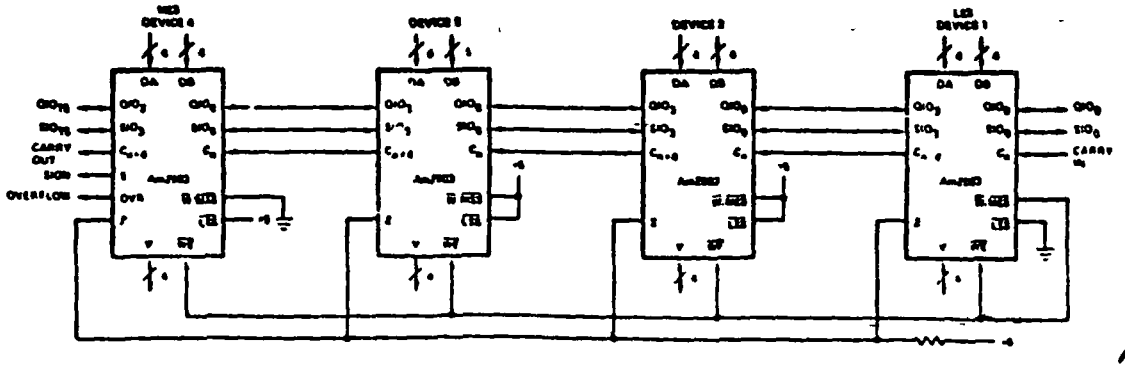


Figure 2 16-Bit CPU with Ripple Carry

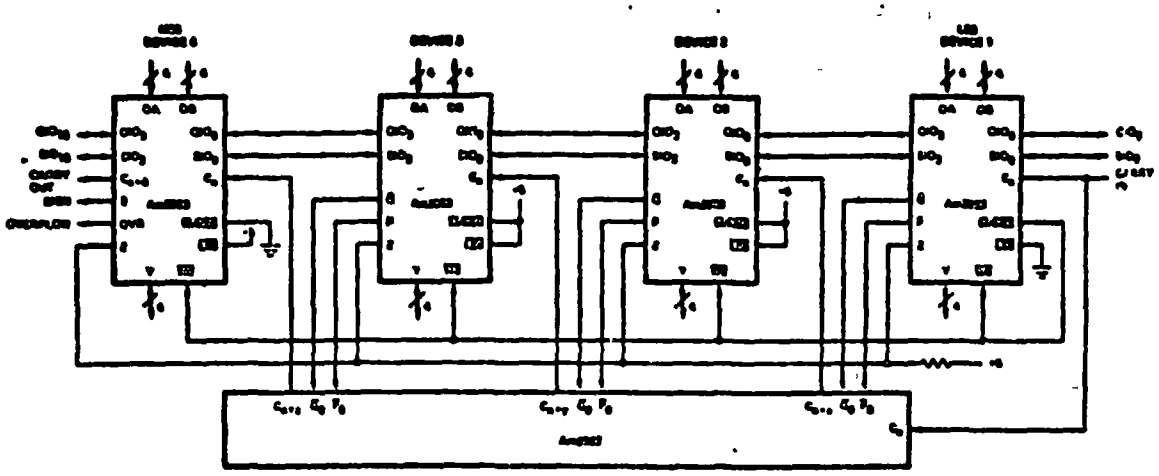


Figure 3. 16-Bit CPU with Carry Look Ahead

$\overline{MSS}$ pin held LOW. The open collector, bidirectional Z pins are tied together for detecting zero or for inter-chip communication for some special instruction. The Carry-Out (C_{n+4}) is connected to the Carry-In (C_n) of the next chip in the case of ripple carry. For a faster carry scheme, an Am2902 may be employed (as shown in Figure 3) such that the $\overline{G}$ and $\overline{P}$ outputs of the Am2903 are connected to the appropriate $\overline{G}$ and $\overline{P}$ inputs of the Am2902, while the C_{n+x}, C_{n+y}, and C_{n+z} outputs of the Am2902 are connected to the C_n input of the appropriate Am2903. Note that $\overline{G}$/S and $\overline{P}$/OVR pin functions are device dependent. The most significant slice outputs S and OVR while all other slices output $\overline{G}$ and $\overline{P}$.

The $\overline{IEN}$ pin of the Am2903 allows the option of conditional instruction execution. If $\overline{IEN}$ is LOW, all internal clocking is enabled, allowing the latches, RAM, and Q Register to function. If $\overline{IEN}$ is HIGH, the RAM and Q Register are disabled. The RAM is controlled by $\overline{IEN}$ if $\overline{WE}$ is connected to the $\overline{WRITE}$ output.

It would be appropriate at this point to mention that the Am2903 may be microcoded to work in either two- or three-address architecture modes. The two-address modes allow A+B→B while the three-address mode makes possible A+B→C. Implementation of a three-address architecture is made possible by varying the timing of $\overline{IEN}$ in relationship to the external clock and changing the B address as shown in Figure 4. This technique is discussed in more detail later in this paper.

Parity

The Am2903 computes parity on a chosen word when the instruction bits I_{5-8} have the values of 4_{16} to 7_{16} as shown in Table 3. The computed parity is the result of the exclusive OR of the individual ALU outputs and SIO_3. Parity output is found on SIO_0. Parity between devices may be cascaded by the interconnection of the SIO_0 and SIO_3 ports of the devices as shown in Figure 3. The equation for the parity output at SIO_0 port of the device 1 is given by $SIO_0 = F_{15} \oplus F_{14} \oplus F_{13} \oplus \ldots \oplus F_1 \oplus F_0 \oplus SIO_{15}$.

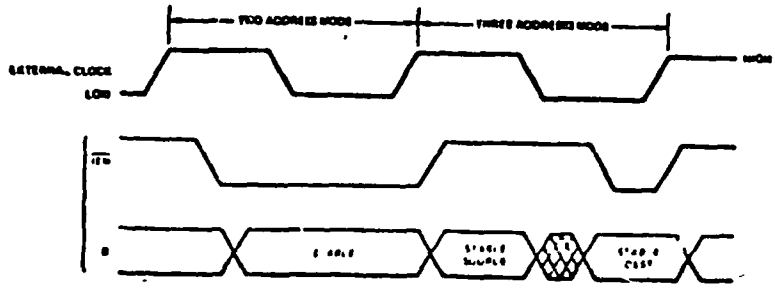


Figure 4. Relationship of $\overline{IEN}$ and Clock During Two and Three Modes

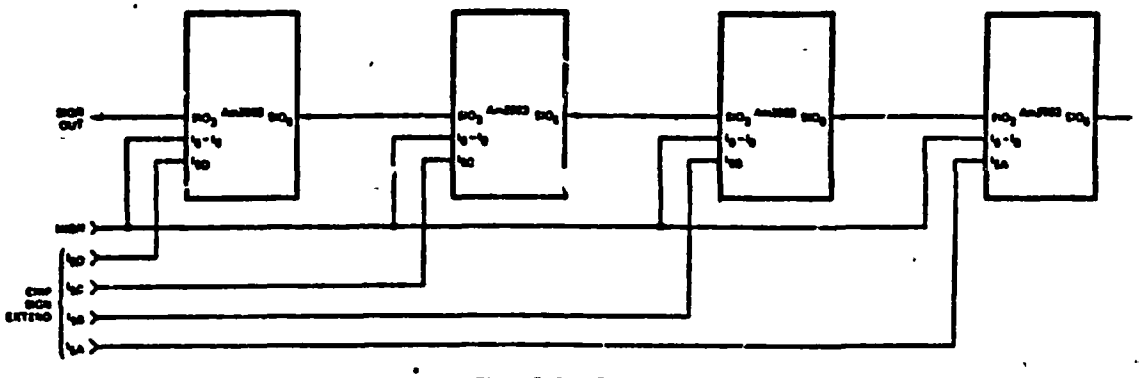


Figure 5. Sign Extend

Sign Extend

Sign extend across any number of Am2903 devices can be done in one microcycle. Referring again to the table of instructions (Table 3), the sign extend instruction (Hex instruction E) on I_{5-8} causes the sign present at the SIO_0 port of a device to be extended across the device and appear at the SIO_3 port and at the Y outputs. If the least significant bit of the instruction (bit I_5) is HIGH, Hex instruction F is present on I_{5-8}, commanding a shifter pass instruction. At this time, F_3 of the ALU is present on the SIO_3 output pin. It is then possible to control the extension of the sign across chip boundaries by controlling the state of I_5 when I_{6-8} are HIGH. Figure 5 outlines the Am2903 in sign extend mode. With I_{6-8} held HIGH, the individual chip sign extend is controlled by I_{5a-d}. If, for example, I_{5a} and I_{5b} are HIGH while I_{5c} and I_{5d} are LOW, the signal present at the boundaries of devices 2 and 3 (F_3 of device 2) will be extended across devices 3 and 4 to the SIO_3 pin of device 4. The output of the four devices will be available at their respective Y data ports. The next positive edge of the clock will load the Y outputs into the address selected by the B port. Hence, the results of the sign extension is stored in the RAM.

SPECIAL FUNCTIONS

When $I_{0-4}=0$, the Am2903 is in the Special Function mode. In this mode, both the source and destination are controlled by I_{5-8}. The Special Functions are in essence special microinstructions that are used to reduce the number of microcycles needed to execute certain functions in the Am2903.

Normalization, Single- and Double-Length

Normalization is used as a means of referencing a number to a fixed radix point. Normalization strips out all leading sign bits such that the two bits immediately adjacent to the radix point are of opposite polarity.

Normalization is commonly used in such operations as fixed-to-floating point conversion and division. The Am2903 provides for normalization by using the Single-Length and Double-Length Normalize commands. Figure 6a represents the Q Register of a 16-bit processor which contains a positive number. When the Single-Length Normalize command is applied, each positive edge of the clock will cause the bits to shift toward the most significant bit (bit 15) of the Q Register. Zeros are shifted in via the QIO_0 port. When the bits on either side of the radix point (bits 14 and 15) are of opposite value, the number is considered to be normalized as shown in Figure 6b. The event of normalization is externally indicated by a HIGH level on the C_{n+4} pin of the most significant slice.

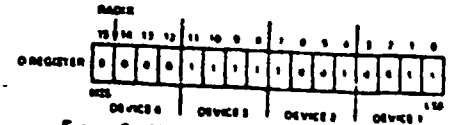


Figure 6a. Unnormalized Positive Number

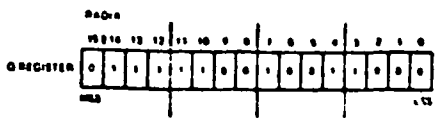


Figure 6b Normalized Positive Number

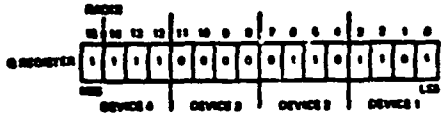


Figure 7a. Unnormalized Negative Single Length Number.

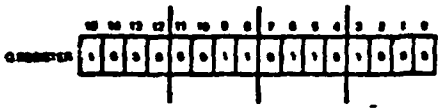


Figure 7b. Normalized Negative Single Length Number.

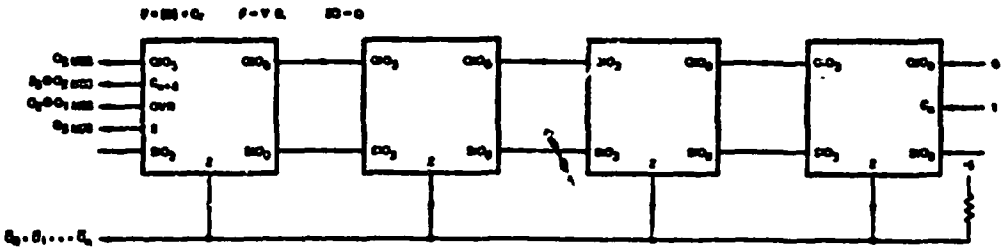


Figure 8. Single Length Normalize

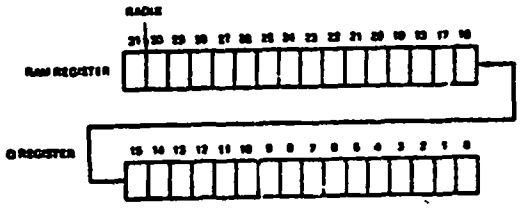


Figure 9. Double Length Word

There are also provisions made for a normalization indication via the OVR pin one microcycle before the same indication is available on the C_{n+4} pin. This is for use in applications that require a stage of register buffering of the normalization indication.

Since a number comprised of all zeros is not considered for normalization, the Am2903 indicates when such a condition arises. If the Q Register is zero and the Single-Length Normalization command is given, a HIGH level will be present on the Z line. An unnormalized negative number (Figure 7a) is normalized in the same manner as a positive number. The results of single-length normalization are shown in Figure 7b. The device interconnection for single-length normalization is outlined in Figure 8. During single-length normalization, shift counting may occur in an internal register.

Normalizing a double-length word can be done with the Double-Length Normalize command which assumes that a user-selected RAM Register contains the most significant portion of the word to be normalized while the Q Register holds the least significant half (Figure 9). The device interconnection for double-length normalization is shown in Figure 10. The C_{n+4}, OVR, S, and Z outputs of the most significant slice perform the same functions in double-length normalization as they did in single-length normalization except that C_{n+4}, OVR, and S are derived from the output of the ALU of the most significant slice in the case of double-length normalization, instead of the Q Register of the most significant slice as in single-length normalization. A high-level Z line in double-length normalization reveals that the outputs of the ALU and Q Register are both zero, hence indicating that the double-length word is zero.

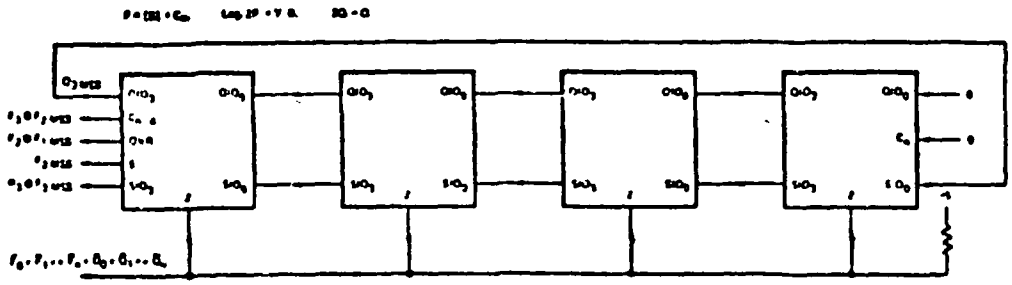


Figure 10 Double Length Normalize

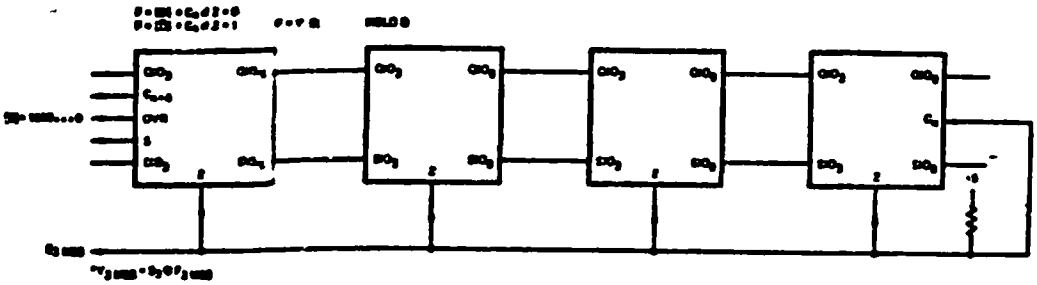


Figure 11. 2's Complement ⟷ Sign/Magnitude

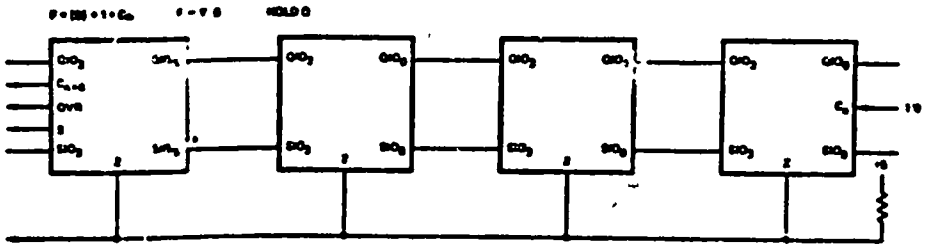


Figure 12. Increment by 2/1

When double-length normalization is being performed, shift counting is done either with an extra microcycle or with an external counter.

Sign Magnitude, Two's Complement Conversion

As part of the special instruction set, the Am2903 can convert between two's complement and Sign/Magnitude representations. Figure 11 illustrates the interconnection needed for sign magnitude/two's complement conversion. The C_n input of device 1 is connected to the Z pin. The sign bit (B_3 MSS) is brought out on the Z line and informs the other ALU's if the conversion is being performed on a negative or positive number. If the number attempted to be converted is the most negative number in two's complement [i.e., 100 . . . 00 (-2n)], an overflow indication will occur. This is because -2n is one greater than any number that can be represented in sign magnitude notation and hence an attempted conversion to sign magnitude from -2n will cause an overflow. When converting minus zero in sign magnitude notation (1 . . . 0) to two's complement notation, the correct result is obtained (0 . . . 0).

Increment by One or Two

Incrementation by One or Two is made possible by the Special Function of the same name. This command is quite useful in the case of byte addressable words. Referencing Figure 12, a word may be incremented by one if C_n is LOW or incremented by two if C_n is HIGH.

Unsigned Multiply

These Special Functions allow for easy implementation of unsigned multiplication. Figure 13 is the unsigned multiply

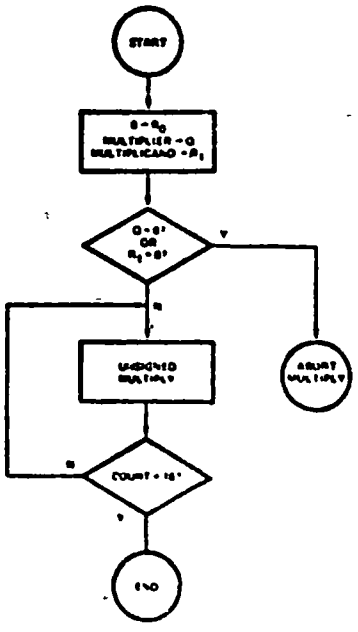
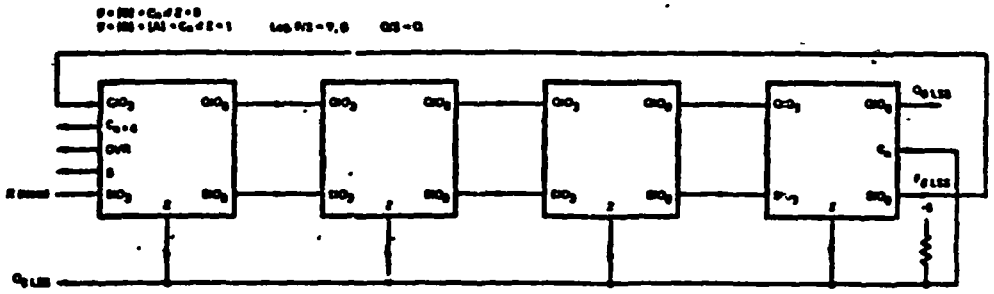


Figure 13 Unsigned Multiply Flowchart



Note For unsigned multiply, C_{n+4} MSS is internally shifted into position Y_3 MSS. For 2's complement multiply S ⊕ OVR is internally shifted into position Y_3 MSS.

Figure 14. Multiply

flow chart. The algorithm dictates that initially the RAM word addressed by Address port B be zero, the multiplier is in the Q Register, and the multiplicand in the register address by Address port A. If the Q Register or the register addressed by the A port is zero, the multiplication can be aborted. If both registers are non-zero, unsigned multiplication may then proceed.

When the Unsigned Multiply command is given, the Z pin of device 1 becomes an output while the Z pins of the remaining devices are specified as inputs as shown in Figure 14. The Z output of device 1 is the same state as the least significant bit of the Q Register during the Unsigned Multiply instruction; therefore, the Z output of device 1 informs the ALU's of all the slices, via their Z pins, to output the sum of the partial product (referenced by the B address port) plus the multiplicand (referenced by the A adddress port) if Z=1. If Z=0, the output of the ALU is simply the partial product (referenced by the B address port). Since C_n is held LOW, it is not a factor in the computation. Each positive-going edge of the clock will internally shift the ALU outputs toward the least significant bit and simultaneously store the shifted results in the register selected by the B address port, thus becoming the new partial sum. During the down shifting process, the C_{n+4} generated in device 4 is internally shifted into the Y_3 position of device 4. At this time, one bit of the multiplier will down shift out of the QIO ports of each device into the QIO_3 port of the next least significant slice. The partial product is shifted down between chips in a like manner, between the SIO_0 and SIO_3 ports, with SIO_0 of device 1 being connected to QIO_3 of device 4 for purposes of constructing a 32-bit long register to hold the 32-bit product. At the finish of the 16 x 16 multiply, the most significant 16 bits of the product will be found in the registers referenced by the B address lines while the least significant 16 bits are stored in the Q Register.

Two's Complement Multiplication

The algorithm for two's complement multiplication is illustrated by Figure 15. The multiplier and multiplicand are loaded and tested as they were during the unsigned multiply operation. The Two's Complement

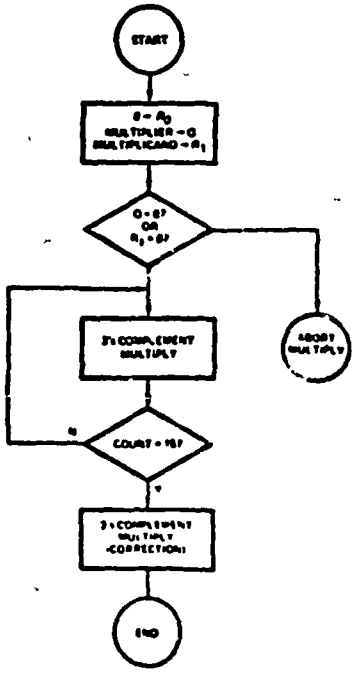
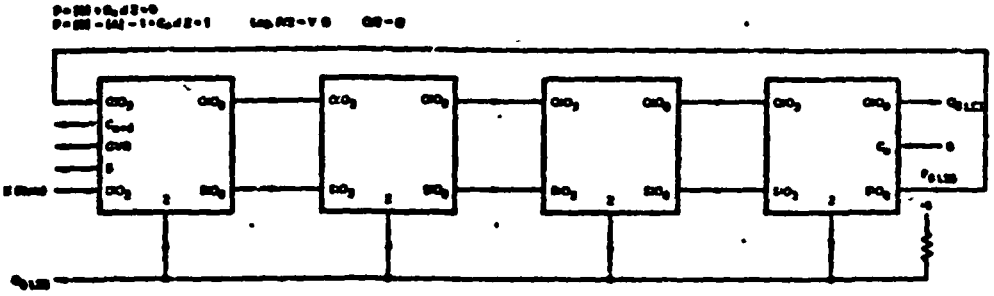


Figure 15 2's Complement 16 X 16 Multiply



Note: S ⊕ OVR is internally shifted into position Y_3 MSB.

Figure 16. 2's Complement Multiply, Correction

multiply Command is applied for 15 clock cycles in the case of a 16 x 16 multiply. During the down shifting process, the term S⊕OVR generated in device 4 is internally shifted into the Y_3 position of device 4. The data flow shown in Figure 14 is still valid. After 15 cycles, the sign bit of the multiplier is present at the Z output of device 1. At this time, the user must place the Two's Complement Multiply (Correction) command on the instruction lines. The interconnection for this instruction is shown in Figure 16. On the next positive edge of the clock, the Am2903 will adjust the partial product, if the sign of the multiplier is negative, by subtracting out the two's complement representation of the multiplicand. If the sign bit is positive, the partial product is not adjusted. At this point, two's complement multiplication is completed.

Two's Complement Division

The division process is accomplished using a non-restoring algorithm with round-off. The algorithm assumes that the absolute value of the divisor is greater than the absolute value of the MS half of the dividend and that the least significant bit of the double-length dividend is truncated. Referring to the flow chart outlined in Figure 17, the divisor is placed in a RAM Register referred to as R_0. The most significant half of the dividend is placed in R_1 while the least significant half is stored in the Q register. Next, the dividend is checked for zero. If zero, the divide can be aborted. If not zero, the most significant half of the dividend stored in R_1 is converted into its sign magnitude form and stored in R_2. If an overflow condition occurs during conversion, the most significant half of the double-length dividend is equal to or greater than the largest possible negative number and hence its resultant magnitude cannot be smaller than the magnitude of the divisor. The dividend must then be scaled. If the dividend is not too large, the next operation is to check the magnitude of the divisor. This is accomplished by doing a two's complement to sign magnitude conversion on the contents of R_0, the destination of the results being Register R_3. As in the case of the dividend, an overflow signifies the largest magnitude possible for a number. At this point the dividend is guaranteed to be larger than the divisor in absolute terms and the actual division process may begin. If, on the other hand, there is no overflow, the divisor is tested for zero. A zero divisor causes an abort. A non-zero divisor allows the next procedual step, which is the shifting out of the sign bits of the divisor and dividend. Next, the divisor in R_3 is subtracted from the most significant half of the dividend in R_2 and the $\overline{C_{n+4}}$ output is tested. A carry-out signifies that the divisor is less than or equal to the dividend and that either the divisor or dividend should be scaled and re-tested for relative magnitude. If there is no carry, the divisor is greater than the dividend and the first step in division may now take place.

The first step in division is to determine the sign of the quotient. This is done utilizing the Double-Length Normalize/First Divide Operation on the double-length dividend in the R_1 and Q Registers. R_1 is referenced by the "B" address port during this operation, while the divisor is addressed by the "A" address port. During this instruction, the sign of

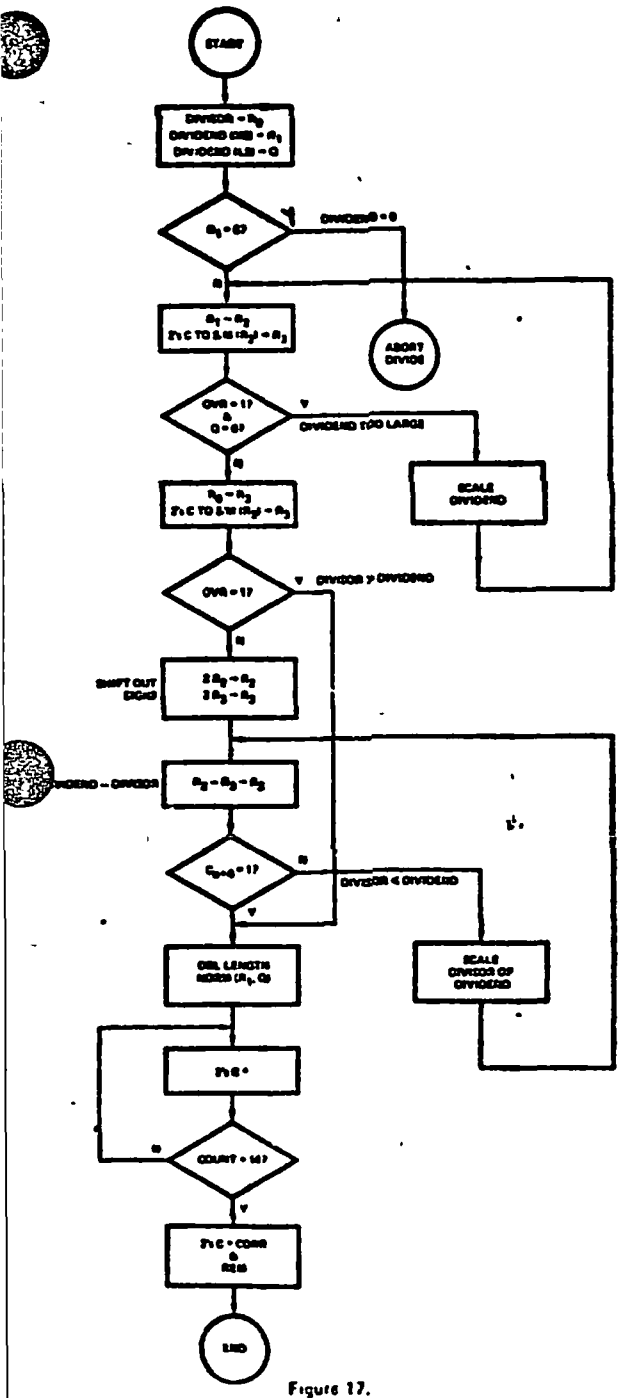


Figure 17.

The quotient will appear on SIO_3 of device 4, and may be stored in the Q register. Figure 18 illustrates the interconnection for the first divide operation. The next command is the Two's Complement Divide instruction which is presented to the Am2903 for 14 microcycles after which the Two's Complement Divide (correction and remainder) instruction is given for one microcycle. On each positive edge of the clock during a Two's Complement Divide instruction, the quotient will be shifted out of SIO_3 of the most significant slice which in turn may be connected to the QIO_0 port of the least significant slice such that the quotient is shifted into the Q Register as the least significant portion of the dividend which is being shifted out as pictured in Figure 19. When the Two's Complement Divide (Correction and Remainder) command is applied, a logical one should be applied to the QIO_0 port of the least significant slice for round-off (Figure 20). At this point, the remainder can be found in Register R which was defined by the "B" address port during the divide instructions. It is noteworthy that the Am2903 is not restricted to double-precision divide operations but can perform multi-precision divides.

Byte Swap

The multi-port architecture of the Am2903 allows for easy implementation of high- and low-order byte swapping. Figure 21 outlines a byte swap implementation utilizing two data ports. Initially, the lower order 8-bit byte is stored in devices 1 and 2, while the high-order byte is in devices 3 and 4. When the user wishes to exchange the two bytes, the register location of the desired word is placed on the B address port. When the byte swap line is brought LOW, the bytes to be swapped will be flowing from the DB ports of the Am2903 and are inverted by the Am25LS240A Three-state Buffers. The outputs of the three-state buffers are permuted such that the byte swap is achieved. The resultant inverted permuted data is presented to the DA ports of the Am2903 where it is re-loaded into the memories of the Am2903 on the next positive edge of CP using the source and function commands of $F=\overline{A}$ plus C_n (C_n=0) and the destination command $F \rightarrow Y,B$.

Memory Expansion

The Am2903 allows for a theoretically infinite memory expansion. Figure 22 pictures a 4-bit slice of a system which has 48 words of RAM and 1[illegible] words of ROM. RAM storage is provided by the Am2903 and the Am29705's. The

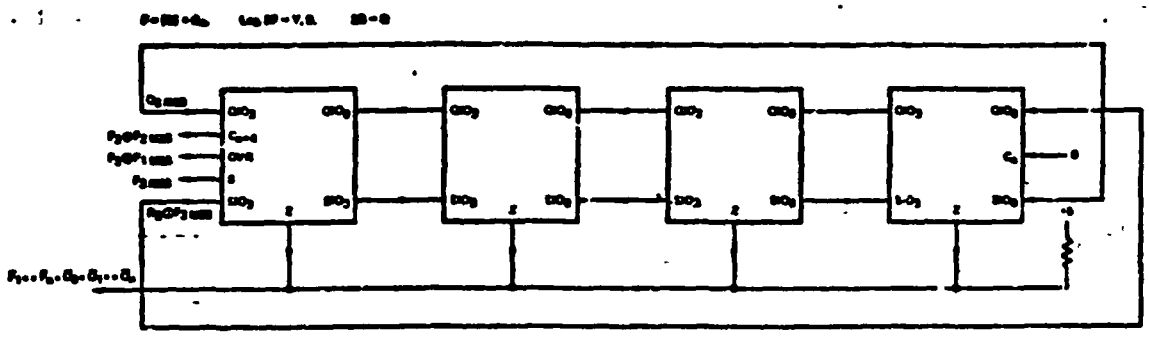


Figure 18. Double Length Normalize/First Divide Operation

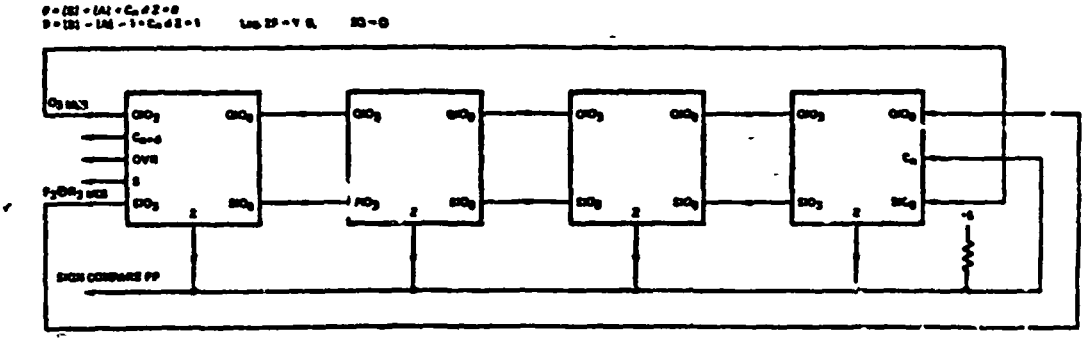


Figure 19. 2's Complement Divide

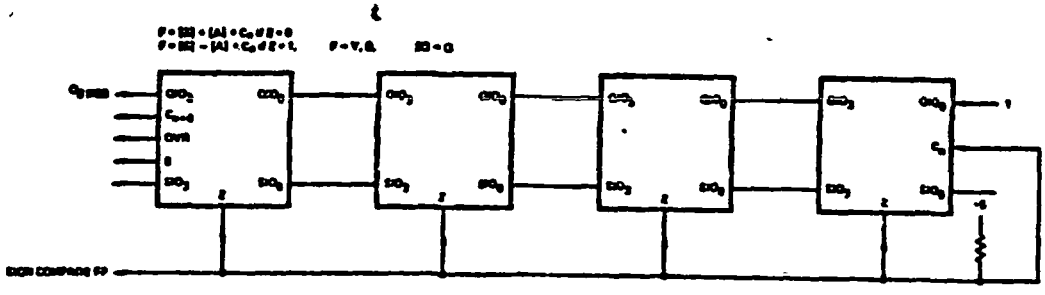


Figure 20. 2's Complement Divide Correction

Am29705 RAM is functionally identical to the Am2903 RAM. The Am29751 is used to store constants and masks and is addressable from address port A only. The system is organized around five data busses. Inter-bus communication may be done through the Am29705's or the Am29703. The memory addressing scheme specifies the data source for the R input of the ALU eminating from the register locations specified by address field A. A_{0-3} addresses 16 memory locations in each chip while address bits A_{4-6} are decoded and used for the output enable for the desired chip. The B address field is used to both select the S input of the ALU and to specify the register location where the result of the ALU operation is to be stored.

Bits B_{0-3} are for source register addressing in each chip. Bits B_4 and B_5 are used for chip output enable selection. B_{6-9} access the 16 destination addresses on each chip while bits B_{10} and B_{11} control the Write Enable of the desired chip. The source and destination register address are multiplexed such that when the clock is HIGH, the source register address is presented to the B address ports of the RAM's. The Instruction Enable ($\overline{IEN}$) is HIGH at this time. The data flows from the Y port or the internal

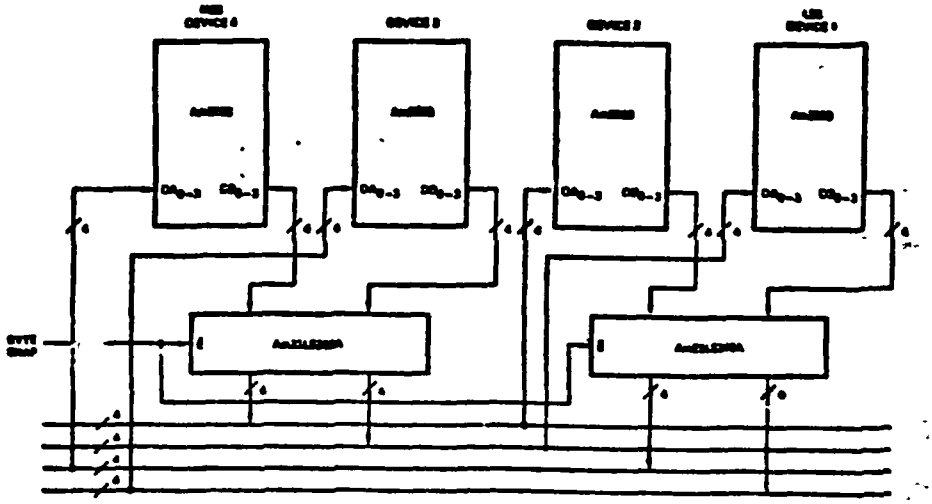


Figure 21. Byte Swap

B port, as selected by decoder whose inputs are AB_4 and AB_5. When the clock goes LOW, the data eminating from the selected Y outputs of the Am29705's and the RAM outputs of the Am2903 are latched and the destination address is now selected for use by the RAM address lines. When the destination address stablizes on the address lines, the $\overline{IEN}$ pin is brought LOW. The $\overline{WRITE}$ output of the Am2903 will now go LOW, enabling the decoder sourced by address bits AB10 and AB11. The selected decoder line will go LOW, allowing the desired memory location to be written into. To switch between two- and three-address architecture, the user simply makes the source and destination addresses the same; i.e., AB_{0-3}=AB_{6-9}. For two-address architecture, the MUX is removed from the circuit.

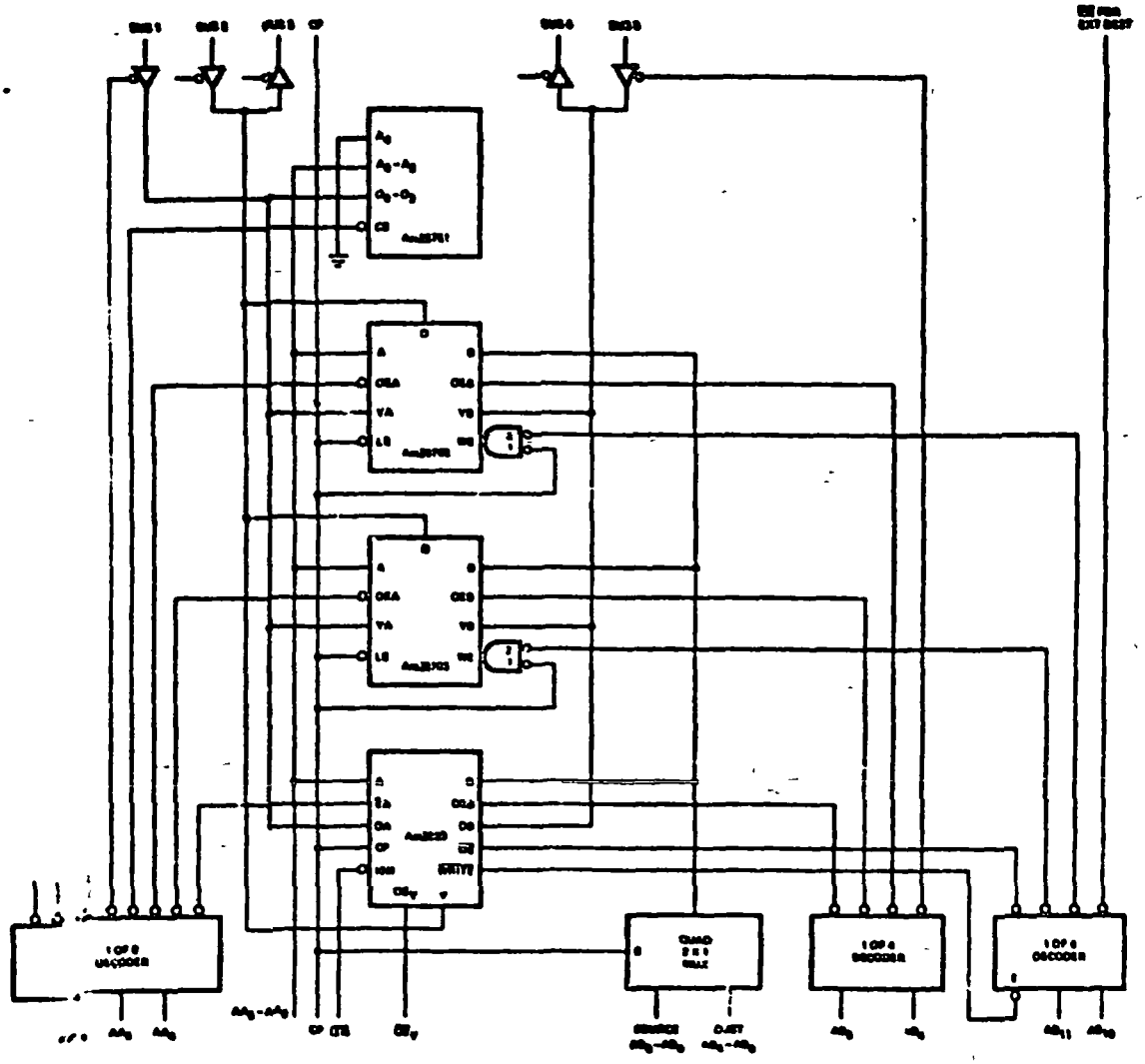


Figure 22. Expanded Memory

Rockwell International ALU

The following material describes the radiation hardened CMOS/SOS ALU developed by Rockwell International.

ROCKWELL INTERNATIONAL

Radiation Hardened CMOS/SOS/LSI

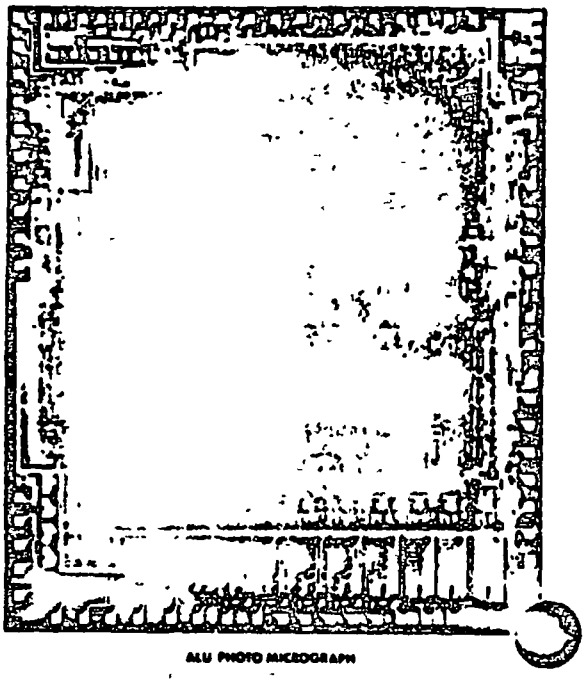
ARITHMETIC AND LOGIC UNIT (ALU)

SPECIAL FEATURES

- 8 BIT SLICE
- EXPANDABLE TO 32 BITS
- ADDER/SUBTRACTOR
- LOGIC FUNCTIONS
- INCREMENT/DECREMENT
- 32 BIT CARRY/BORROW
- LEFT/RIGHT SHIFTS
- 64 PIN CERAMIC FLAT PACKAGE
- FULLY STATIC OPERATION
- RADIATION HARDENED

TYPICAL ELECTRICAL CHARACTERISTICS

- 4 MHz CLOCK RATE
- GATE-PROTECTED INPUTS
- FULL DIELECTRIC ISOLATION
- -55°C TO +125°C OPERATION
- RADIATION HARDNESS COMPATIBLE WITH SPACE AND MISSILE SYSTEM REQUIREMENTS
- +10 TO +12 VDC BIAS RANGE
- LESS THAN 300 mW POWER DISSIPATION AT 4 MHz



ALU PHOTO MICROGRAPH

APPLICATIONS

ARITHMETIC AND LOGIC FUNCTIONS FOR 8, 16, 24 OR 32 BIT PROCESSORS

GENERAL DESCRIPTION

THE ALU PERFORMS ALL THE STANDARD ARITHMETIC AND LOGIC FUNCTIONS NORMALLY ASSOCIATED WITH COMPUTER ARITHMETIC. IT IS DESIGNED TO FUNCTION AS THE KEY ELEMENT OF A CENTRAL PROCESSOR UNIT (CPU). THIS DESIGN IS IMPLEMENTED AS AN EIGHT-BIT SLICE WITH ALL THE NECESSARY INPUTS AND OUTPUTS TO ALLOW CASCADING TO FORM A 32-BIT UNIT. THE MAJOR COMPONENTS OF THE DEVICE ARE IDENTIFIED IN FIGURE 1.

ONE OF THE FEATURES OF THE OVERALL CPU IS THE TRI-STATE BIDIRECTIONAL BUS, WHICH IS THE MAIN DATA PATH. THIS DATA BUS REQUIRES THE ALU CHIP TO HAVE A BIDIRECTIONAL PORT (PORT B) WITH TRI STATE DRIVERS. THE ARCHITECTURE REQUIRES THAT THE ALU HAVE TWO ADDITIONAL PORTS TO COMMUNICATE WITH OTHER DEVICES. ONE OF THESE PORTS IS STRICTLY AN INPUT PORT TO PROVIDE THE ALU WITH OPERANDS (PORT B2). THE SECOND PORT IS ANOTHER TRI-STATE BIDIRECTIONAL DATA PORT (PORT A). TRI-STATE DRIVERS ARE USED TO PROVIDE MAXIMUM SYSTEM FLEXIBILITY AND ALSO TO ALLOW A CONTROL PIN TO BE USED FOR INPUT AND OUTPUT.

SINCE PORT B1 IS OFTEN USED FOR TRANSPORTING DATA UNRELATED TO THE COMPUTATION BEING PERFORMED IN THE ALU, THE ALU MUST HAVE INTER-CHIP SHIFT PATHS THAT ARE INDEPENDENT OF PORT B1. THE REQUIREMENTS FOR DIVISION, AND FOR THE TWO-BIT AT-A TIME MULTIPLY ALGORITHMS, DICTATE FOUR SHIFT INPUTS AND FOUR SHIFT OUTPUTS FOR EACH ALU CHIP.

TRANSMISSION GATE MULTIPLEXERS (MUX'S) ARE PROVIDED ON ALL INPUT AND OUTPUT PORTS. MULTIPLEXERS ON THE INPUT ALLOW FOR SELECTION OF INPUTS AND ALSO PROVIDE SHIFT PATHS FOR IMPLEMENTING DIVISION AND TWO-BIT-AT-A-TIME MULTIPLY ALGORITHMS. SHIFT INFORMATION IS OBTAINED FROM A HIGHER ORDER DEVICE (FOR FUNCTIONS SUCH AS MULTIPLY) OR FROM A LOWER ORDER DEVICE (FOR FUNCTIONS SUCH AS DIVIDE). MULTIPLEXERS ON THE OUTPUT NODES ALLOW VARIOUS INTERNAL PATHS TO BE SELECTED FOR PROVIDING OUTPUTS.

THE MULTIPLEXERS FEED INTO CLOCKED INPUT BUFFER REGISTERS (MECHANIZED WITH TRANSMISSION GATES). THESE CLOCKED REGISTERS ALONG WITH LOGIC-CONTROLLED REGISTERS, ARE USED EXTENSIVELY IN BOTH THE CONTROL AND THE ARITHMETIC SECTIONS. THESE LATCHES ALLOW INTERNAL SYNCHRONIZATION OF CONTROLS AND OPERANDS AS WELL AS SYSTEM SYNCHRONIZATION. TIMING FOR THE CLOCKED REGISTERS IS PROVIDED BY A SINGLE-LOGIC LEVEL CLOCK INPUT.

FIVE INPUT SIGNALS ARE DECODED TO GENERATE THE NECESSARY CONTROL SIGNALS. DECODING OF THESE ARITHMETIC AND LOGIC FUNCTION CONTROL LINES TAKES PLACE BEFORE THE BUFFER REGISTERS ARE OPENED, THEREFORE, THEIR DECODE TIME IS ELIMINATED FROM THE CRITICAL TIME DELAY PATH. TABLE I LISTS THE ARITHMETIC AND LOGICAL OPERATIONS THAT THE ALU CAN PERFORM.

THE CARRY STRUCTURE FOR THE DEVICE IS IMPLEMENTED IN THE FOLLOWING MANNER. THE INTERNAL CHIP CARRY STRUCTURE IS IMPLEMENTED USING A PARALLEL LOOK-AHEAD APPROACH FOR MAXIMUM SPEED. HOWEVER, THE INTER CHIP CARRY STRUCTURE USES A PASS THROUGH APPROACH WHICH MINIMIZES THE NUMBER OF REQUIRED PINS.

[illegible] AT A [illegible] THE PROCEDURES. NUCLEAR DISCHARGE [illegible] SINCE MOS DEVICE CHARACTERISTICS [illegible] TO ALSO CAUSE DAMAGE HAS BEEN [illegible] PROCESSING TECHNIQUES.

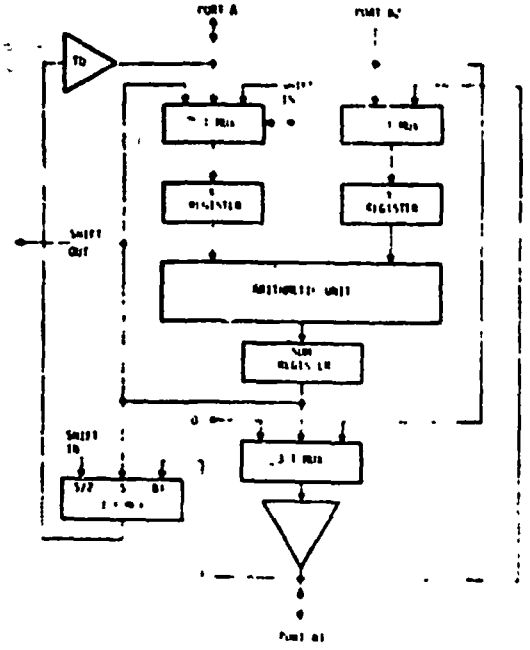


Figure 1 Block Diagram

ORIGINAL PAGE IS OF POOR QUALITY

Table 1. ALU Arithmetic and Logical Functions

LOGICAL AND	$S = X \cdot Y$
LOGICAL OR	$S = X + Y$
LOGICAL EXCLUSIVE OR	$S = X \odot Y$
ZERO	$S = 0$
ADD	$S = X + Y$
ADD AND INCREMENT	$S = X + Y + 1$
SUBTRACT	$S = X - Y$
SUBTRACT AND DECREMENT	$S = X - Y - 1$
DIVIDE LOGIC	$S = X \div Y$
PASS Y	$S = Y$
INCREMENT Y	$S = Y + 1$
2's COMPLEMENT Y	$S = 0 - Y$
1's COMPLEMENT Y	$S = 0 - Y - 1 = \bar{Y}$
PASS X	$S = X$
INCREMENT X BY 1	$S = X + 1$
INCREMENT X BY 2	$S = X + 2$
DECREMENT X BY 1	$S = X - 1$
DECREMENT X BY 2	$S = X - 2$
ALL 1's	$S = 1111\,1111_2$
REMOVE EXPONENT BIAS	$S = X - 200_8$
EXPONENT UNDERFLOW CHECK	$S = X - 040_8$

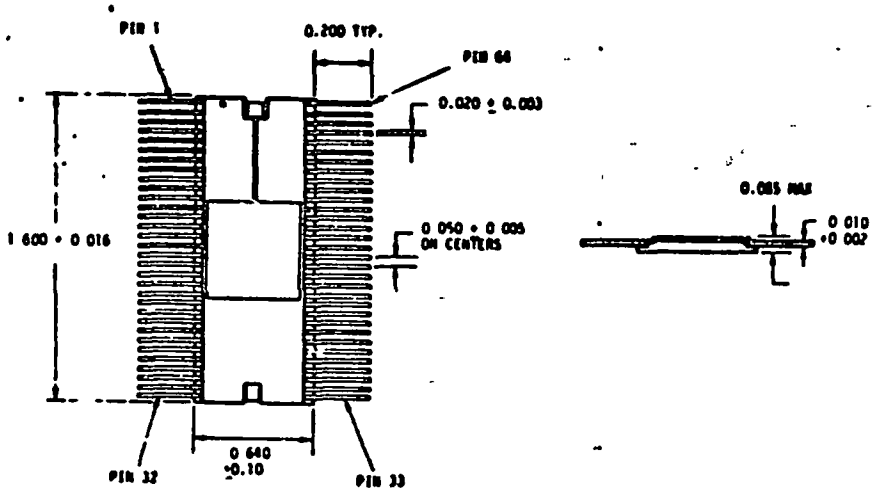


Figure 2. Package Dimensions

Table 2 Pin Connections

PIN	TERM	PIN	TERM	PIN	TERM	PIN	TERM
1	COUT	17	B22	33	ZIN	49	FC0
2	NC	18	B12	34	LO	50	FC2
3	NC	19	B23	35	S5	51	FC3
4	E2	20	B13	36	S6	52	S5
5	RA0	21	B24	37	S7	53	VDD
6	RA1	22	B14	38	E1	54	OVOUT
7	RA2	23	B25	39	VDD	55	NC
8	RA3	24	B15	40	VSS	56	XMUX1
9	RA4	25	B16	41	LC	57	XMUX2
10	RA5	26	B17	42	YMUX2	58	XMUX4
11	RA6	27	B26	43	YMUX1	59	CBOUT
12	RA7	28	B27	44	CLOCK	60	B7
13	B20	29	DMUX	45	VSS	61	B6
14	B10	30	SMUX1	46	IDI	62	B5
15	B21	31	SMUX2	47	R(OS)	63	S0
16	B11	32	CIN	48	FC1	64	ZOUT

FOR FURTHER INFORMATION CONTACT

E J STEPHENS
ROCKWELL INTERNATIONAL
AUTONETICS GROUP
3370 MIRALOMA AVE . MAIL STATION GA 30
ANAHEIM, CA 92803

TELEPHONE (714) 632-3157

Pub. No. P77-1026/201
New Oct. 1977 (200)
Printed in USA

MICROCOMPUTER COMPONENT VENDORS

The vendors whose addresses are listed below have local offices of representatives in most major cities throughout the United States. Inquiries or requests for more information should be directed to the local representative for quickest response.

Advanced Micro Devices, Inc.
901 Thompson Place
Sunnyvale, California 94086
(408) 732-2400

Digital Equipment Corporation
Components Group
One Iron Way
Marlborough, Massachusetts 01752
(800) 225-9480

Rockwell International
E. J. Stephens
Autonetics Group
3370 Miraloma Avenue, Mail Station GA 30
Anaheim, California 92803
(714) 632-3357

Texas Instruments Incorporated
MS366 - P. O. Box 5012
Dallas, Texas 75222
(214) 238-6805

APPENDIX II

TIMING AND SIZING ESTIMATES OF COMMON ROUTINES AND ALGORITHMS USED IN NSSC-I ANALYSIS

COMMON ROUTINES	SIZE (18 BIT WORDS)	SIZE (18 BIT WORDS)
Double [1] Add	13	5
Double Subtract	16	5
Double Multiply	24	5
Double Absolute Value	11	3
Double Matrix Multiply	65	8
Double Sine/Cosine	202	3
Double Square Root	126	3
Double Arctangent	105	3
Double Arcsine	(uses Arctangent)	21
Double Divide	100[2]	5

NOTES:

1 Double = Double Precision, 36 bits

2 Estimate

EXECUTION TIMES FOR MATHEMATICAL ROUTINES[3]

	Single Precision	Double Precision
Add	.016 ms.	.063 ms.
Subtract	.018 ms.	.083 ms.
Multiply	.053 ms.	.233 ms.*
Divide	.085 ms.*	2.500 ms.**
Sine and Cosine	.375 ms.	1.600 ms.
Square Root	.54 - .84 ms.***	3.53 - 4.92 ms.***

Matrix Multiplication (m x n) · (n x p):

Single precision = .055mp + .105mnp ms.

Double Precision = .095mp + .310mnp ms.

* The result is truncated, not rounded off
** Guess--the routine has not been coded. This is an iterative procedure and execution times will vary
***Iterative procedure--execution times will vary according to the number of iterations.

Arctangent Algorithm - This algorithm was described by J. S. Walther on the class of cordic algorithms. The "pseudo" code that defines this algorithm follows:

NOTE:

3 Refer to CSC, 1976, pg. 32

CYCLES				
	DBLATAN	PROC	X, Y, Z	
4		LDA	ZERO	;INITIAL SHIFT
6		STA	I	
4	LOOP:	LDA	Y	
4		LDE	Y + 1	
5[4]		DSM	I	;SHIFT I PLACES
6		STA	X2	
6		STE	X2 + 1	
4		LDA	X	;Y2 2^{-1}X
4		LDE	X + 1	
5[4]		DSM	I	
6		STA	Y2	
6		STE	Y2 + 1	
4		LDA	Y2 + 1	;CHANGE SIGN
6		NEG		
6		STA	Y2 + 1	
4		LDA	Y2	
6		CMP		
4		ADC		
6		STA	Y2	
4		LDA	ALPHA	;Z2 =
6		STA	Z2	
4		LDA	Y	
3		TAP		;TEST Y 0
4		BRC	L1	;CF POS, ADD
62		DBLSUB	X, X, X2	;ELSE SUB
62		DBLSUB	Y, Y, Y2	;Y = Y - Y2
62		DBL SUB	Z, Z, Z2	;Z = Z - Z2
4		BRU	INCR	
50	L1:	DBLADD	X, X, X2	;X = X + X2
50		DBLADD	Y, Y, Y2	;Y = Y + Y2
50		DBLADD	Z, Z, Z2	;Z = Z + Z2

```
CYCLES

4          INCR:    LDA     I
4                   SUB     ONE          ;NEG TO RT. SHIFT
6                   STA     I
4                   TAL     LIMIT
4                   BR      LOOP
```

INPUTS: $X = A$; $Y = B$, $Z = 0$

OUTPUTS: $X = (A^2 + B^2)^{\frac{1}{2}}$; $Y = 0$, $Z = Z + \tan^{-1}\frac{(Y)}{(X)}$

TIMING ESTIMATE = 15.894 ms

Due to the nature of the algorithm, the precision of the result is $\log_2 N$ bits less; where N is the number of bits of precision of temporary results.

NOTES:

4. plus one cycle per bit shift

Arcsine Algorithm - The arcsine was calculated using the arctangent plus the following relationshipL

ARCSINE (N) = ARCTAN (f (N))

where $f(N) = N\,(1-N^2)^{-\frac{1}{2}}$

Due to the limitation of the arctangent, the result here is 30 bit of precision.

TIMING EST:MATE = 23.04 ms

APPENDIX III

PERTINENT REFERENCE MATERIALS

The following pages list source material which was reviewed during this study and which may be useful in future investigations.

OADS Sensor and Performance Study Related Information

[APL,1] "Interface Control Document for GPSPAC RECEIVER/PROCESSOR ASSEMBLY," Applied Physics Laboratory, Silver Spring, Maryland, April, 1977.

[Birnbaun,1] "Time Requirements in the Navstar Global Positioning System," A. J. VanDierendonck, General Dynamics Electronics Division, and M. Birnbaum, Space and Missile Systems Organization, presented at the 30th Annual Frequency Control Symposium, June, 1976.

[Clevinger,1] Correspondence from R. L. Clevinger of BBRC to K. Villyard of Martin Marietta regarding SST Star Tracker Accuracy Requirement.

[Goddard,1] "Goddard Trajectory Determination System (GTDS) Users Guide," Goddard Space Flight Center, Greenbelt, Maryland July, 1975.

[Headley,1] "On-Orbit Calibration Procedure and Alogrithum for DRIRU-II IMU and BBRC-SST Star Trackers," by P. Headly, Martin Marietta Internal Report, April, 1978.

[Martin Marietta,1] "DRIRU-II Gyro Misalignment Calibration Method and Simulation Result," Martin Marietta Internal Memorandum by P. Headley, 30 August, 1977.

[Martin Marietta,2] "SDG-5 Gyro PSD Test Data"

[Martin Marietta,3] Gilbert, L.E., and Mahajan D.T., "On-board Landmark Navigation and Attitude Reference Parallel Processor System," Martin Marietta Aerospace, Presented at the Flight Mechanics/Estimation Theory Symposium at the Goddard Space Flight Center, Ocotober 1977.

[Satin,1] "Satellite Navigation with the Global Position Satellite System (GPSS)," A. Satin, Martin Marietta Memorandum, 7 September 1977.

[Teledyne,1] "Standard High Performance Inertial Reference Unit DRIRU-II Technical Description," Volume 1, Teledyne Systems Company, February 1977.

[Yong,1] "Attitude Determination Algorithm of a Strapdown Inertial Reference Gyro System Using Teledyne's DRIRU-II Units," K. Yong, Martin Marietta Memorandum, 5 May 1977.

[Yong,2] "Attitude Determination Algorithm of a Star Tracker System Using BBRC-SST Star Tracker," K. Yong, Martin Marietta Memorandum, 7 June 1977.

[Yong,3] "Attitude Determination Simulation Results of Using Teledyne's DRIRU-II Strapdown System Under Maneuvering Envrionment," Martin Marietta Interna' Memorandum, K. Yong, 24 August 1977.

Microprocessor Related Information

Advanced Micro Devices, Inc.: The AM2900 Family Data Book. Advanced Micro Devices, Inc., Sunnyvale, California, 1976

Advanced Micro Devices, Inc.: AM9511 Arithmetic Processing Unit, Preliminary Information. Advanced Micro Devices, Inc., Sunnyvale, California, 1977

P. J. Besser, et. al.: Development of a High Capacity Bubble Domain Memory Element and Related Epitaxial Garnet Materials for Application in Spacecraft Data Recorders. Item 1, Development of a High Capacity Memory Element. NASA-CR-144983, 1977

T. T. Chen, O. D. Bohning, et. al.: Investigation of System Irtegration Methods for Bubble Domain Flight Recorders. NASA-CR-132643, 1975

Vernon Coleman, et. al: The Next Generation Four-Bit Bipolar Microprocessor Slice - The AM2903. Advanced Micro Devices, Sunnyvale, California, 1977

Digital Equipment Corporation: PDP 11/03 Processor Handbook. Digital Equipment Corporation, Maynard Massachusettes, 1975

Eric R. Caren: "Magnetic Bubble Memory Devices and Applications." Computer Design, Feb. 1978, pgs 164-168

Albert G. Ferris and Edward P. Greene: "A Proposed Concept for Improved NASA Mission Data Management Operations," NASA X-533-76-81, October 1976

Intel Corporation: 8080 Microcomputer Systems Users Manual. Intel Corporation, Santa Clara, California 1975

Interdata, Inc.: Model 8/16 Processor, Product Description. Interdata, Inc., Oceanport, New Jersey, 1977

JPL: Report on Phase II of the Microprocessor Seminar Held at Caltech, April 1977. JPL Report 77-39

J. Egil Juliussen, et. al.: "Bubbles, Appearing First as Microprocessor Mass Storage." Electronics, August 4, 1977, pgs. 81-86

John F. Mason: "Bubble Memories Are Going Military in Air and Space Applications," Electronic Design, pgs. 34-36

Ware Myers: "Current Developments in Magnetic Bubble Technology." Computer, Aug. 1977, pgs. 73-82

R. M. Orndorff and D.T. Butcher: High-Performance Analog and Digital Custom CMOS/SOS LSI Circuitry. Rockwell International, Anaheim, California, 1977

Rockwell International: Radiation Hardened CMOS/SOS/LSI Arithmetic and Logic Unit, Preliminary Device Description. Rockwell International Autonetics Group, Anaheim, California, 1977

Texas Instruments, Inc.: TMS 9916 Bubble Memory Controller. Texas Instruments, Inc., Dallas, Texas, 1977

Texas Instruments, Inc.: Magnetic Bubble Memories and System Interface Circuits. Texas Instruments, Inc., Dallas, Texas, 1977

Texas Instruments, Inc.: Bipolar Microcomputer Components Data Book. Texas Instruments, Inc., Dallas, Texas, 1977

Texas Instruments, Inc.: TMS 9900 Microprocessor Data Manual. Texas Instruments, Inc., Dallas Texas, 1976

Larry W[illegible]ler: "Tests Show Spotty LSI Record." Electronics, Feb. 2, 1978, pgs. 78-79

NSSC-I Related Information

Computer Science Corporation: Preliminary Study of On-Board Attitude Control for the MultiMission Modular Spacecraft, NASA contract NAS5-11999 Task Assignment 558 Report, February 1977

J. S. Waltner: A Unified Algorithm for Elementary Functions. Hewlett-[illegible]