CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

PHOENIX MISSION SIMULATION TEST

A project submitted in partial satisfaction of the requirements for the degree of Master of Science in Engineering.

by

Laurence Eugene Sellers

July, 1976

The project of Laurence Eugene Sellers is approved:

California State University, Northridge

June, 1976

ACKNOWLEDGMENT

The author wishes to thank many people for their suggestions and their specific help. My colleagues at the Pacific Missile Test Center, gave me general and specific advice. However, I want in particular to express my graditude to Dr. Ichiro Hashimoto, my graduate advisor and to Edwin I. Miyasaka, my technical leader. A personnel note of thanks is given to Roger Lucic who inspired me to do this project, and to Joyce Moody for her patience and timeliness. Most importantly, my love goes to Linda who gave her total support to me during this project.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYM AND SYMBOL LIST

ACRONYMS

A/D	ANALOG TO DIGITAL
ADPE	AUTOMATIC DATA PROCESSING EQUIPMENT
AOT	ACTIVE ON TARGET
AGC	AUTOMATIC GAIN CONTROL
CAS	CONTROL AND ANALYSIS SYSTEM
CRT	CATHODE RAY TUBE
D/A	DIGITAL TO ANALOG
DAS	DATA ACQUISITION SYSTEM
DVM	DIGITAL VOLTMETER
FFT	FAST FOURIER TRANSFORM
FY	FISCAL YEAR
GLAT	GOVERNMENT LOT ACCEPTANCE TEST
GMTS	GUIDED MISSILE TEST SET
HP	HEWLETT PACKARD
LIS	LABORATORY INTEGRATED SYSTEM
LTE	LAUNCH TO EJECT
MBAM	MAINBEAM AVOIDANCE MANEUVER
MECCA	MISSILE ENVIRONMENTAL COMPUTER CONTROL AND ANALYSIS
MOAT	MISSILE ON AIRCRAFT TEST
MST	MISSION SIMULATION TEST
MTP	MISSILE TEST PROGRAM
PATE	PRODUCTION ACCEPTANCE TEST AND EVALUATION

PMTC	PACIFIC MISSILE TEST CENTER
PRN	PROGRAM REFERENCE NUMBER
RF	RADIO FREQUENCY
RFMTG	RADIO FREQUENCY MOVING TARGET GENERATOR
RTN	RETURN
SAOT	SEMI-ACTIVE ON TARGET
TCR	TEST CONTROL ROOM
TTY	TELETYPEWRITER
UP	UMBILICAL PIN
VCLO	VOLTAGE CONTROLLED LOCAL OSCILLATOR

SYMBOLS

DC	DIRECT CURRENT
Hz	HERTZ
IPS	INCHES PER SECOND
K	KILO - (1000)
KHz	KILOHERTZ
mm	MILLIMETER
ms	MILLISECOND
mv	MILLIVOLT
usec	MICROSECONDS
V	VOLT
VDC	VOLTS DIRECT CURRENT

ABSTRACT

PHOENIX MISSION SIMULATION TEST

by

Laurence Eugene Sellers

Master of Science in Engineering

This Graduate Project Report describes the effort undertaken in forming and implementing the idea of a functional missile test composed of a simulated launch of a Navy PHOENIX air-to-air radar guided missile, and collecting continuous missile performance data. The missile test and data collection system is called a MST (Mission Simulation Test). The MST consists of a MTP (Missile Test Program) to control the missile, and an A/D (Analog to Digital) Driver program to collect the performance data.

The main goal of this project was to design the missile test and data collection system, without violating the hardware and software configurations of the AN/DSM-130(V) GMTS (Guided Missile Test Set); and to use the MST as part of the acceptance tests a missile must pass before final acceptance is approved by the Navy. This goal has been achieved.

CHAPTER I

INTRODUCTION

1.0 Background

A MST (Mission Simulation Test) for the AIM-54A PHOENIX missile is needed to improve the missile test capability of the MECCA (Missile Environmental Computer Control and Analysis) System. The MECCA system was developed by the Navy at the PMTC (Pacific Missile Test Center), Point Mugu, California, for use in support of missile PATE (Production Acceptance Test and Evaluation). Housed in the PMTC Environmental Simulation Laboratory, MECCA environmentally tests missiles in the laboratory, prior to final acceptance by the Navy.

During the production of the PHOENIX, selected missiles from a sample of each month's production lot is tested in the laboratory while others undergo ground checkout and captive flight tests on F-14A aircraft. These tests assess the quality of each production lot by determining compliance with reliability and performance specifications. Failures occurring in each test are considered in reaching the accept/reject decision for the entire lot.

The MECCA system provides automatic control of combined environmental conditions such as temperature, altitude, humidity, and vibration. This system simulates the conditions of shipboard handling, captive missile carry on an aircraft, and free flight of the missile. It has the unique feature of providing automatic functional test of an "operating" missile under environmental stress. The system is comprised of two primary subsystems; an environmental test, control and monitoring system, and a missile test, control and monitoring system. A diagram of the MECCA system is shown in Figure 1.

The present PHOENIX test set, the AN/DSM-130(V) GMTS (Guided Missile Test Set), which is the missile test, control and monitoring system, performs functional tests on the missile, but only in a sequential manner. The proper stimulus is sent to the missile via computer control, and the corresponding missile response is measured and recorded. The response measured is only from one small unit of the missile, not a "true" system level response. If one stimulus could be sent to the missile, and all the missile subsystem responses measured and recorded at one time, then the true system response would be known. It is precisely this feature that the MST possesses.

The MST consists of two computer programs. The first program is a MTP (Missile Test Program) that powers the missile up, applies the correct stimuli, and monitors the corresponding responses. The MTP simulates a typical launch mission. The second program is an A/D (Analog to Digital) driver to process PHOENIX performance data from the missile test program. The A/D driver operates a HP 5610A analog to digital converter. Missile responses from the MTP are recorded on a Bell and Howell, CEC/DATATAPE VR-3400 14 track magnetic tape unit.

Immediately after the MTP has completed the simulated launch, the tape recorded analog data are played back through the HP 5610A A/D converter. The missile responses are then "digitized" and outputted in an easy to read format. True parallel data processing takes place. A diagram of the MECCA system with the MST added capability is shown

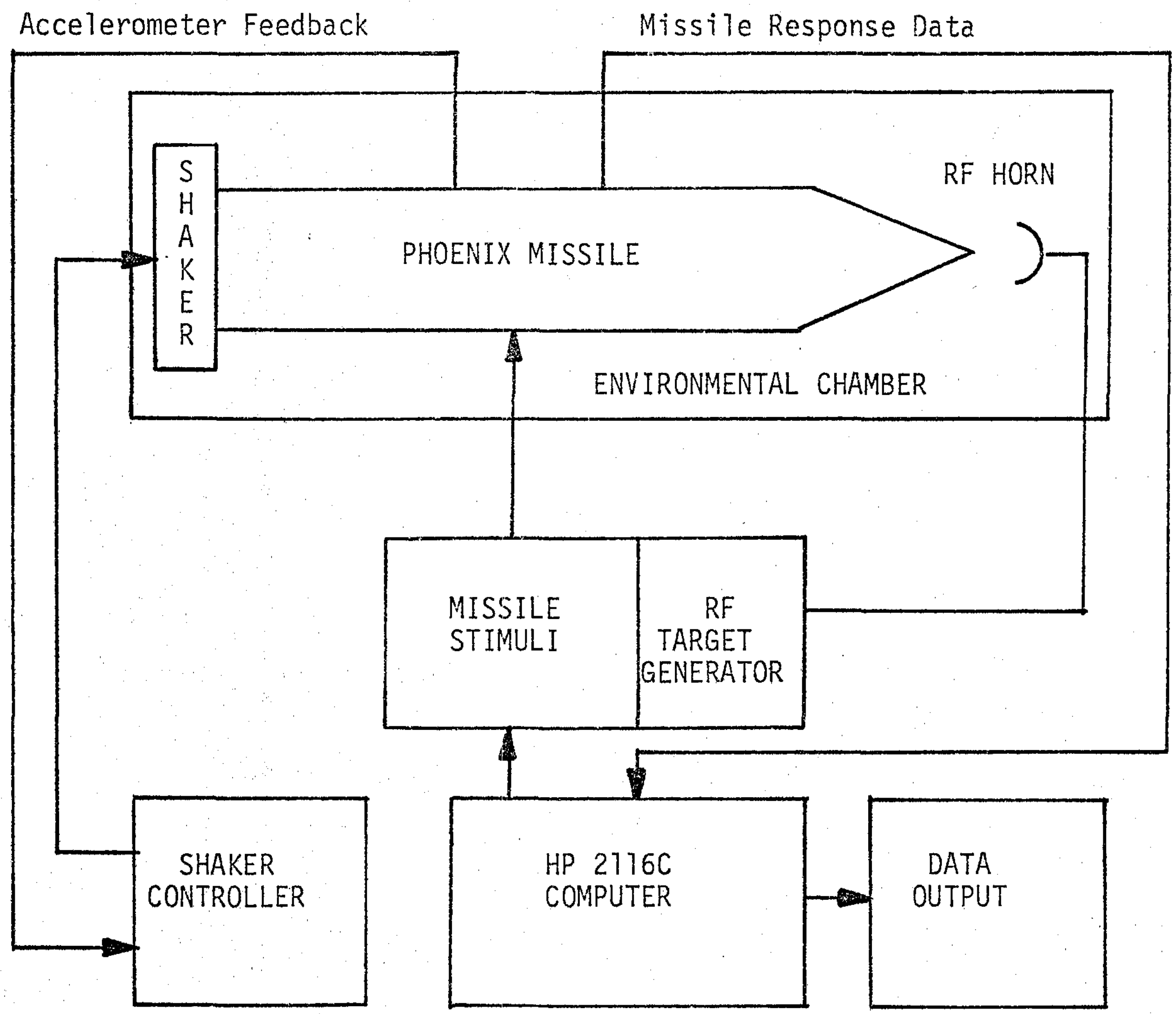


Figure 1. - MECCA System Diagram

in Figure 2. Figure 3 shows a block diagram of the MST.

The MST is part of a continuing process of improving the MECCA system. It is anticipated, in the future, that a RFMTG (Radio Frequency Moving Target Generator) will be incorporated into the MECCA system.

1.1 Problem Statement

The PHOENIX missile GMTS does not have the capability to output missile performance data in parallel. Present GLAT (Government Lot Acceptance Test) procedures require the evaluation of the performance of the missile as a complete operating system. The MST will give the added capability to continuously sample the subsystem performance of the missile, and simulatenously output subsystem performance data in analyzed form.

1.1.1 Constraints

In order to have an operating system developed and ready for operation for the FY-76 (Fiscal Year) procurement of PHOENIX missiles, the MST added capability had to meet the following constraints:

Time: A deadline of 30 June 1976 was established. This deadline allowed three to four months of MST validation and contractor negotiations before the MST would be used for fault isolation in missile lot acceptance.

Money: The MST job task was funded under the existing customer order number 6N3JC11C. $25000 dollars was made available for labor charges.

Manpower: The $25000 dollars created funding for approximately five-six man months of work effort.

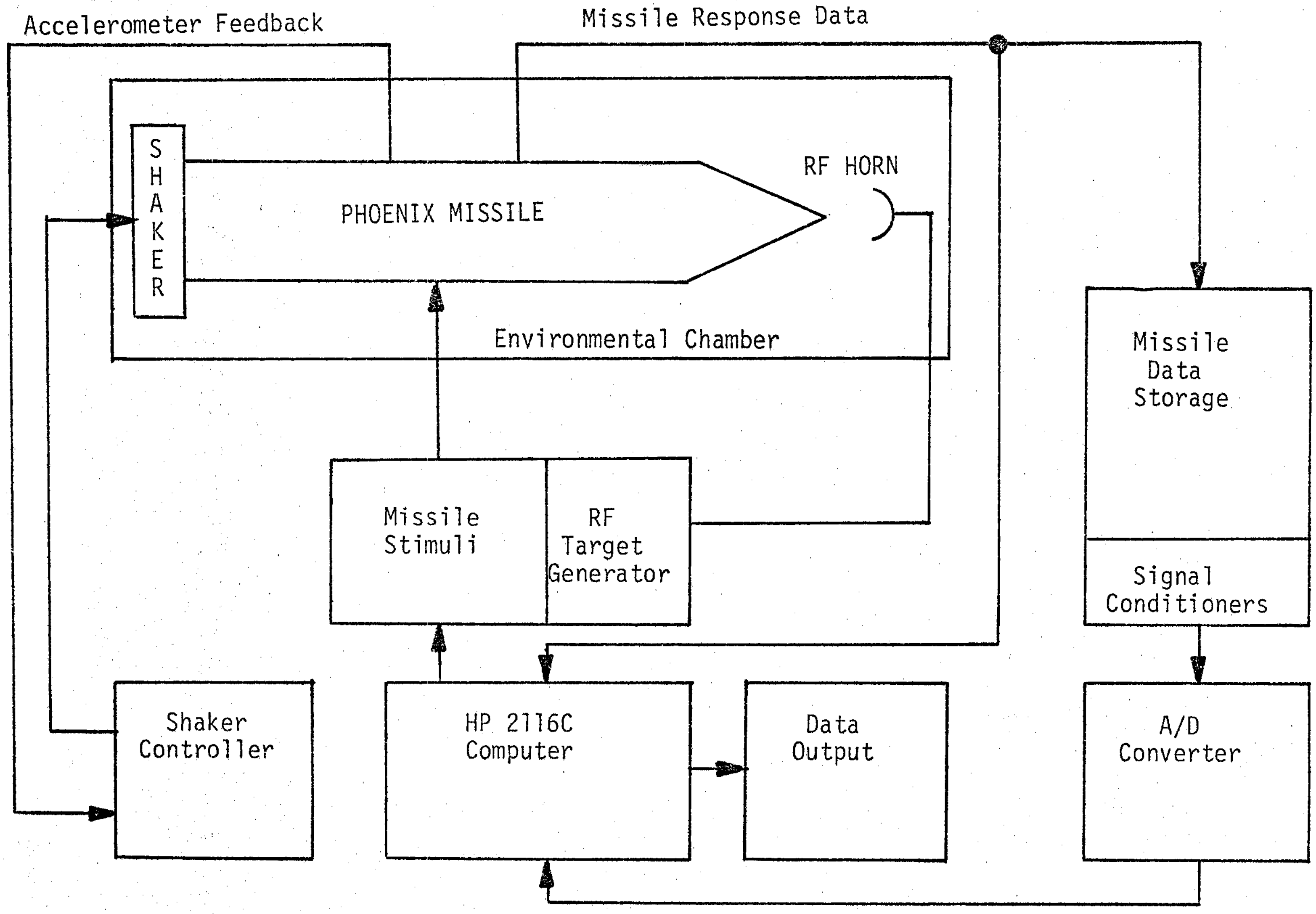


Figure 2. - MECCA System with MST

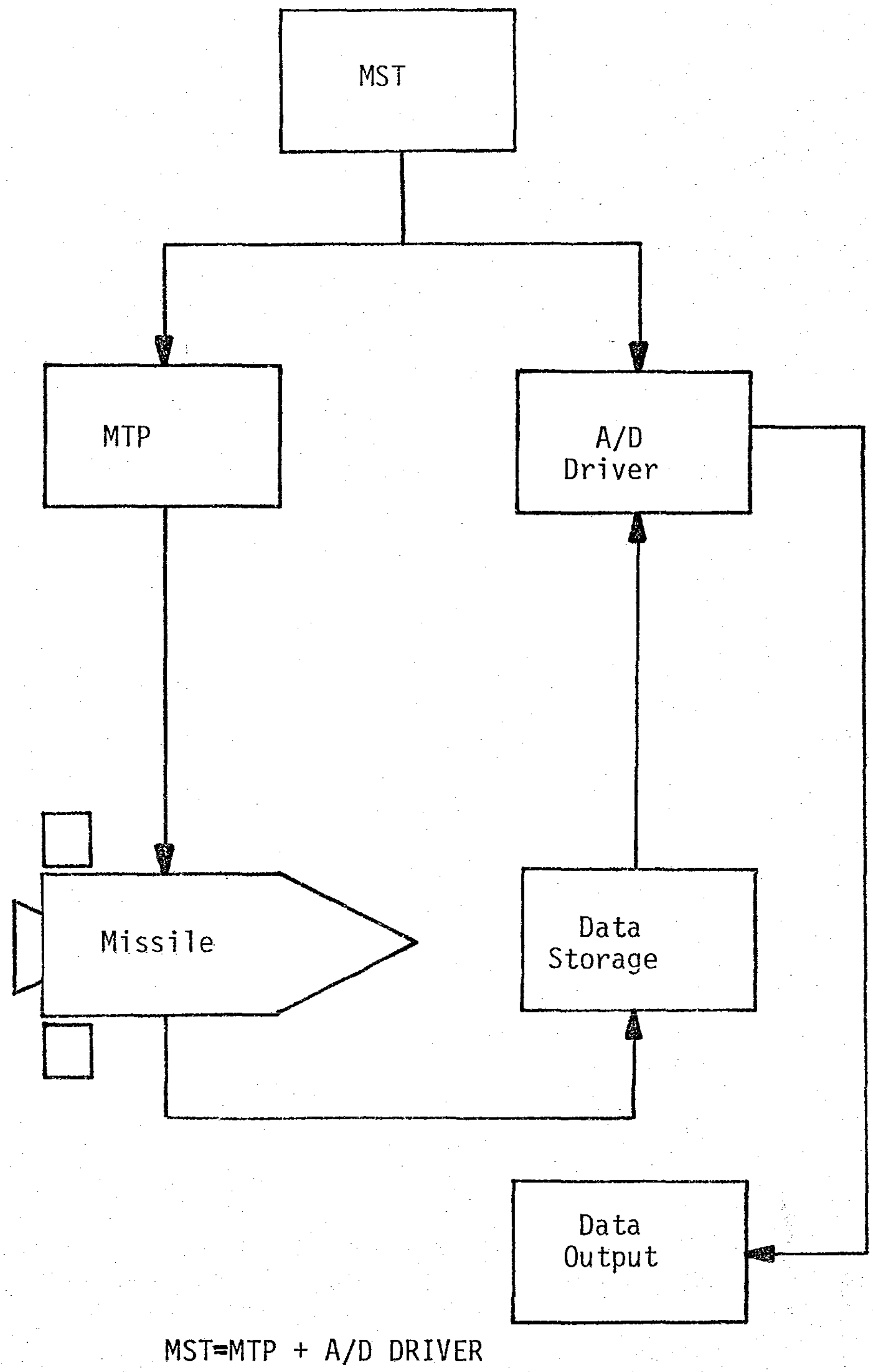


Figure 3. - Mission Simulation Test Block Diagram

AS - 4007 GLAT SPECIFICATION: The PHOENIX GLAT specification, that contains all the rules and regulations of missile lot acceptance testing, requires the AN/DSM-130(V) GMTS be the test set used for PHOENIX testing, and requires the current contractor developed software be the software used in missile testing.

Modifications: The GLAT specification also decrees that no modifications to the AN/DSM-130(V) can be made except those that are contractor/Navy approved. The GMTS must stay in contractor/Navy configuration control.

1.1.2 Assumptions

The following assumptions were made before the MST development began:

(1) The AN/DSM-130(V) GMTS will be used for missile testing.

(2) The MTP will be implemented only on the PHOENIX GMTS.

(3) The MST will be modular, so changes and additions will be easy to implement.

(4) The MST must be easy to operate, and use minimum of personnel.

(5) The task must be accomplished at minimum cost.

(6) The MST will be operated by trained PHOENIX personnel.

(7) The MST will have growth potential.

(8) Analog to digital conversion of data will be employed.

(9) Missile responses will be accessible via the missile umbilical.

(10) Missile signal characteristics (amplitude and frequency) will be previously determined.

1.1.3 Questions To Be Answered By This Report

General Design Questions

1. Can the existing PHOENIX test set be used for A/D conversion?

2. Does the PDP-11 computer have the capability of processing data from the HP 5610A A/D converter?

3. Can the HP 2116C and PDP-11 computers be used in combination to run missile tests and process missile response data?

4. What tape recording capabilities presently exist in the laboratory?

5. Will the MST allow real time data processing?

6. What computer language will be most useful for operator interplay?

7. How much computer memory will be available for data storage?

8. Will the HP 5610A A/D converter handle the expected data rate?

9. Will a signal multiplexer be required?

10. What signal conditioning will be necessary?

11. Will signals be patched through the GMTS cross-bar switch or through separate patches?

12. What signal conditioning capability presently exists in the laboratory?

Specific Design Questions

1. What type coaxial cable will be used?

2. What missile response signals will be monitored?

3. What sampling rate and amplitudes will limit signal processing?

4. What trigger source will be used?

1.2 Objective

The development of a MST will improve the MECCA test capability by using the MECCA AN/DSM-130(V) GMTS, and the PHOENIX missile to determine system response of the missile as it undergoes a typical launch mission. The MST will be designed to continuously collect performance data from all the major missile subsystems simultaneously. The missile functional performance will be evaluated for "system" performance vice the "one stimulus-one response" performance presently available.

1.3 Purpose

The MST will give quick assessment of missile performance for a typical launch mission. The MST will aid in locating missile failures as a result of GLAT or environmental testing. It will provide a "hands on" trouble-shooting ability of the missile that is not currently available.

CHAPTER 2

APPROACH TO PROBLEM

2.0 Define Instrumentation Requirements

A CAS (Control and Analysis System) was proposed as an addition to the MECCA system in 1972, to enhance missile functional test capability. The CAS was never funded by the Navy because of a cutback in developmental funds.

The existing GMTS has a serious drawback in that although several input missile stimuli can be computer controlled simultaneously, only one output parameter can be measured at a time. The present mechanical cross-bar switch provides switching between missile responses and the proper measuring device. The cross-bar switch can only switch to one measuring device at a time. Thus, events may occur too rapidly to be recorded by the existing system. Therefore, total missile performance is difficult to evaluate, and missile failures are difficult to trace to specific causes.

The proposed CAS would have allowed the recording of multiple outputs through multiplexing methods, data buffering, data filtering, and magnetic tape storage. The use of digital computers as data controllers permit easy modification and expansion. The system was specified to have up to sixty-four signals sampled at 250 Hz, or fewer signals sampled more frequently. Thus, total missile performance could be recorded, evaluated, and post-test analysis could be performed using the data recorded during the test.

The proposed system was to include, a 64 channel multiplexer, a minicomputer (CPU), and a magnetic tape unit. An additional CPU and a line printer were necessary if the system was to have additional flexibility for future system expansion and testing of a wider variety of missiles.

As an alternative to CAS the MST concept was proposed. The MST concept provides an alternative in building a data acquisition system, plus a missile test program to simulate a typical launch mission of the PHOENIX missile.

The proposed MST data acquisition system will include, as a minimum, a 16 channel analog to digital converter, a minicomputer (CPU), a 14 track FM (Frequency Modulation) analog magnetic tape unit, and a line printer.

In the testing of the PHOENIX missile it is necessary to record the signals listed in Table 1. This table lists the signal, and its approximate rate of change. To faithfully record these signals it is necessary to sample each signal at a minimum of five times the rate of change. The discrete, pulsed, and switched DC signals need not be sampled as frequent due to their discrete levels. Since the missile signals are essentially analog, provision will be needed for A/D conversion. The A/D converter should have 10-12 bit resolution, with a sampling rate of at least 500 Hz/channel.

Basically, the missile data requirements dictated the need for the following equipment:

(1) A computer (CPU) to receive digital data, process it, and output missile response information in a useful data format.

TABLE OF PHOENIX SIGNALS

10 Hz Signals

PIN #	SIGNAL	PIN #	SIGNAL
20	Seeker EL	121	R. Band Press
21	Seeker Az	127	Seeker Az Er.
30	Roll	139	Paccel Er.
34	Rear Mxr	140	Y Integ
39	Vrh	148	Vsg
48	VFC	157	RIG Demod
59	Vrg	167	Vsh
60	AGC	9	G-axis lim
66	RFC	11	4-axis lim
68	PFC	69	Dither
70	Roll Integ	114	VCLO
87	P	160	AVCO
103	A. P. Moat Out	141	Y Accel Er.
110	Seeker El. Er	180	Mult Test
119	P Lat Integ	196	AP Moat Cond.

200 Hz Signals

28	P Lat Accel	50	Y Lat Accel
31	Y	51	P
32	R		

SWITCHED DC

1	8 sec	81	Det
17	180 msec	96	TO MOAT
19	Pre Lau Ø	99	SD
27	Act Init	104	FTS
41	OT	111	XMTR PA
46	SE	116	Cen Chan
62	TL	131	AOT
79	FRO	136	Timer Zero Set
80	Rear Ø	184	Hyd Press
82	SAOT		

Pulsed 130 TDD

Discrete 63 ID

Table 1 - Missile Data Requirements

(2) An A/D converter to digitize the analog responses from the missile.

(3) A magnetic tape unit to store missile performance data.

(4) Signal conditioners to buffer, scale, and amplify the missile response data.

2.1 Definition of MST

2.1.1 Missile Test Program

The PHOENIX Mission Simulation Test is a simulated launch of a real PHOENIX missile. The test is computer controlled by a MTP which powers the missile up, supplies multiple stimuli to the missile, and records time related missile responses. Continuous missile responses will be recorded on a 14-track magnetic tape unit. (See A/D driver below).

The MTP will stimulate all major subsystems of the guidance and control sections of the missile. (Refer to Figure 4). The MTP will interrogate the guidance section to find out if correct velocity and angle tracking are taking place. The control section will be monitored for proper steering responses. The steering commands in the control section will be compared to the seekerhead angle tracking error in the guidance section. The missile steering commands should null out the angle tracking of the seekerhead. The control surfaces (flippers) will be monitored to determine if the steering signals are of sufficient magnitude and proper direction to "steer" the missile toward its intercept point.

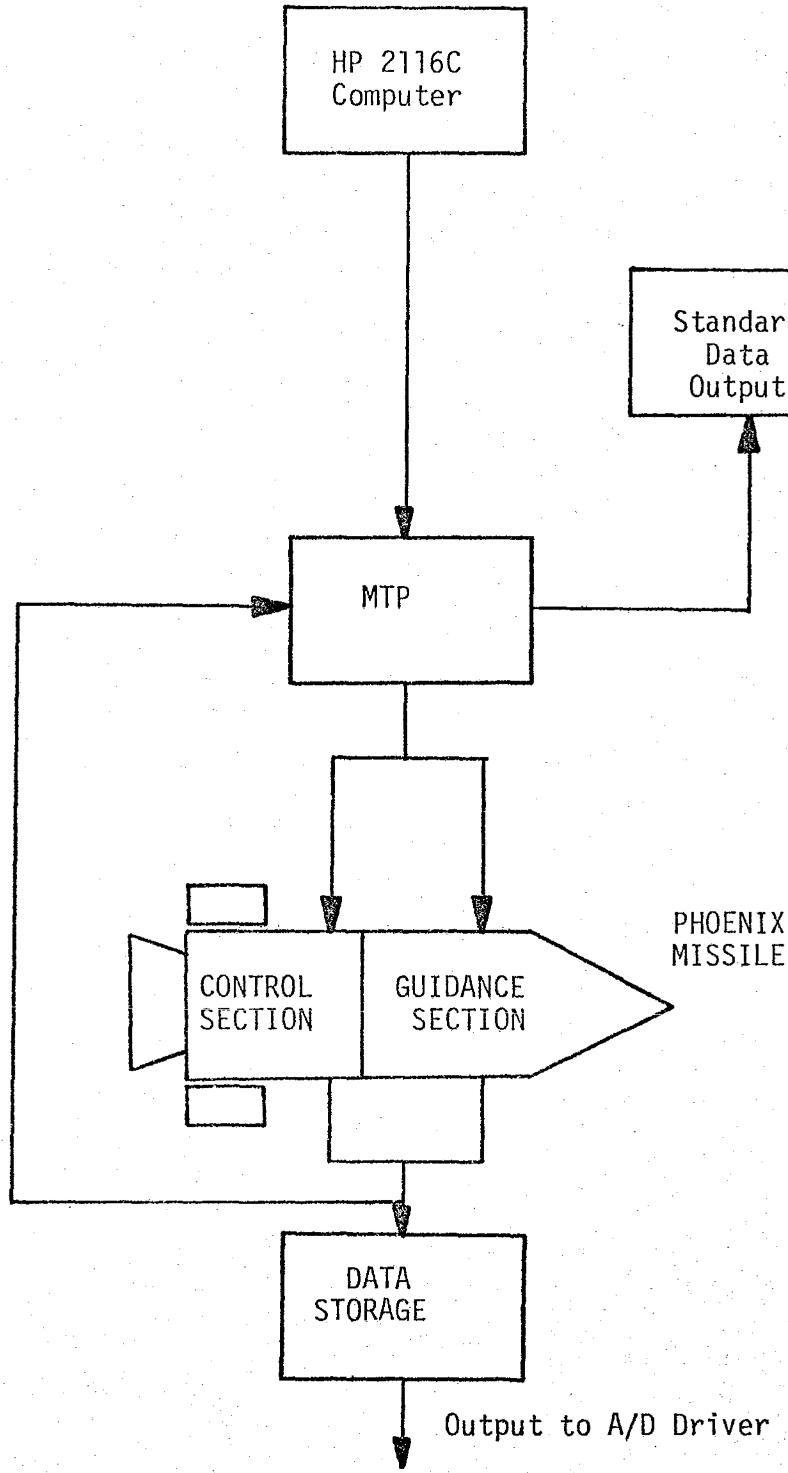


Figure 4. - Missile Test Program Integration

2.1.2 A/D Driver

The second part of the MST, is to process continuous performance data coming from the missile. The continuous data will be buffered, scaled, filtered, and recorded on a 14 track magnetic tape unit. The continuous data, after being recorded, will be played back through a data processing computer program call the A/D Driver. (See Figure 5).

The A/D driver will accept the analog data from the tape unit, digitize it, and store the digital data in a computer. The A/D driver will sample the analog data at a specified rate.

The driver will have the options of sampling on one, or up to eight channels (sixteen later). The A/D will be "triggered" on a voltage magnitude of the user's choice on all eight channels. The driver will also have the capability of saving data before the trigger actually occurs.

2.1.3 Data Processing

After the continuous performance data has been digitized and stored in a computer, data processing will begin. The data will be outputted in tabular form for analysis or it will be plotted.

An analysis of the data will also be possible. The digitized data will be compared to baseline missile response data. If at any time missile performance is not within baseline tolerances, the data will be flagged and a printout of the discrepancy will occur.

When a response is out of tolerance, the subsystem where the signal originates from will be identified. Special subsystem computer

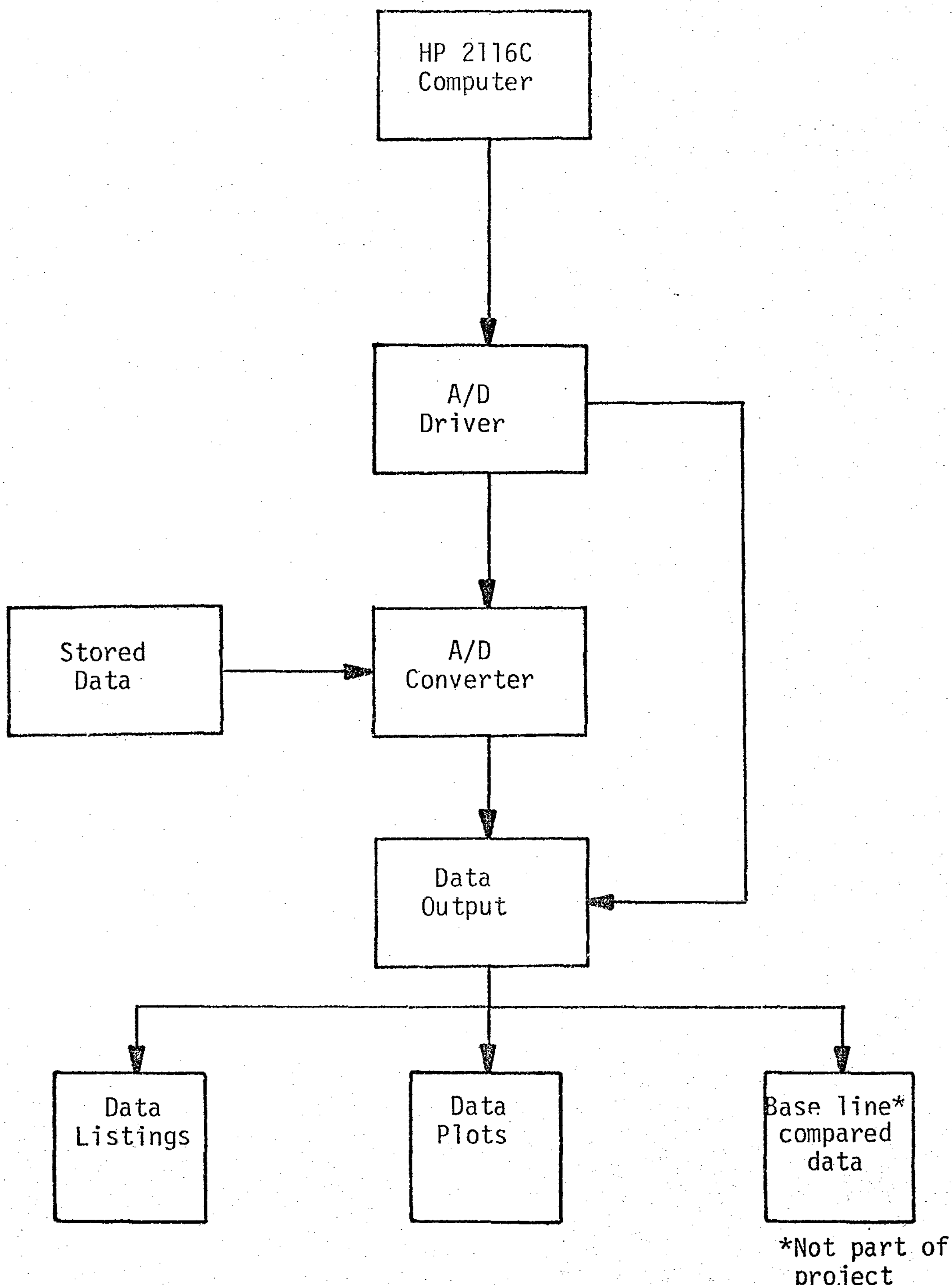


Figure 5. - A/D Driver Program Integration

programs will be called in and for a more detailed test of the subsystem.

The minimum data available after the MST has run will be:

(1) Standard GMTS printout (GO, NO-GO).

(2) Computer listings of digitized data.

(3) A statistical printout (MECCA Catalog Routines).

(4) Computer plots of digitized data.

(5) Special subsystem data listings.

CHAPTER 3

METHODOLOGY

3.0 General

Once the instrumentation requirements and the structure and content of the MST requirements were defined, an investigation of the instrumentation and computer systems that were available in the Environmental Simulation Laboratory was conducted. If an instrumentation system, that met the ideal requirements described in Chapter 2, could be built with existing hardware, a major milestone would be overcome. An investigation into the instrumentation applicability was performed on the below listed equipment.

3.1 Instrumentation Investigation

3.1.1 PDP-11 Minicomputer Processor

A PDP-11 Computer, manufactured by Digital Equipment Corporation, was procured by the Environmental Simulation Laboratory for the purpose of digitally controlling the numerous environmental chambers located within the laboratory.

The PDP-11 was purchased, along with a standard ASR 33 TTY (Teletypewriter) terminal and a CRT (Cathode Ray Tube) console. The PDP-11 was purchased without a software package. Absolute binary paper tapes, that contained the desired computer programs to run the environmental chambers, were to be read in through the TTY terminal. Absolute tapes were to be generated by a TYMSHARE computer program. All the PDP-11 was capable of, was to read absolute paper tapes, and output binary tape via the TTY terminal.

The advantages of using the PDP-11 computer were:

(1) Computer for data processing would not be the same as for controlling the missile. The HP 2116C would control the missile; the PDP-11 would process the missile performance data.

(2) Real time data processing would be possible.

(3) A general data processing system that could be applied to any test of any missile or missile component would be available.

(4) The PDP-11 processor was available, it was not currently being used in any active Navy program.

(5) An A/D converter manufactured by Datel Systems, Inc.was compatible with the PDP-11 and more than met the necessary data processing requirements stated in Chapter 2.

The disadvantages were:

(1) The PDP-11 has small data storage space. The PDP-11 has 12K core (4K in the processor plus 8K additional core). With less than 16K core, the computer programs would have to be written in assembler language.

(2) The available A/D converter (HP 5610A) was not compatiable with PDP-11 computer. Interface/buffer circuitry would have to be either purchased or designed; or another A/D bought.

(3) It is inconvenient to program the PDP-11 computer without having an operating software package to use with the PDP-11 processor. (No EDITOR, ASSEMBLER or LOADER software is available).

3.1.2 The 1923 Time Data System

The 1923 Time Data System was a system procured for real time data analysis of vibration frequency spectrums. The system contained a complete "analysis package" employing the FFT (Fast Fourier Transform) technique for spectrum analysis. The 1923 system was hardware limited to two analog channel inputs. No further attempt was made to make use of this system because of the two channel capability.

3.1.3 The HP 2116C Minicomputer Processor

The HP 2116C minicomputer, manufactured by Hewlett-Packard is part of the PHOENIX missile test set. As an available computer, the HP 2116C was also considered for applicability for developing a data acquisition system meeting the requirements of Chapter 2.

Since the computer was part of the existing PHOENIX test set, there were definite advantages for making use of it. The advantages were:

(1) Personnel were familiar with the operation and programming of the computer.

(2) The system had all necessary software and peripherals (BASIC, FORTRAN, Assembler, Magnetic Disk, Magnetic Tape, Paper Tape Input/Output Devices).

(3) The computer was convenient to use and was accessible.

The disadvantages were the following:

(1) There was limited core storage of missile performance data.

(2) No real time analysis (same computer would operate both the missile and data recovery system) was possible.

(3) Modifications would be allowed only if the integrity of the PHOENIX test set was not violated (GLAT specification requirement).

3.1.4 Combination HP 2116C and PDP-11 Processors

The concept of using both computers simultaneously was also considered. The HP 2116C would control the missile testing, and the PDP-11 would take the missile response signals directly from the A/D converter and store it. Immediately after the missile test had ended, the HP 2116C would receive the digital data from the PDP-11 and analyze it. The analyzed data would then be output to the HP 2767A line printer so the results could be viewed.

3.1.5 HP 5610A A/D Converter

The HP 5610A A/D converter is a general purpose converter, consisting of a 10 bit converter, a Sample and Hold amplifier, and a 16 channel Multiplexer available with 1, 8 and 16 channels. The A/D converter has the option of working in either REMOTE or LOCAL modes. The REMOTE mode puts the A/D under computer control. Each channel can be randomly or sequentially accessed. The conversion time is 10 usec. The analog input range is $\pm$ 10V full scale (15K ohms input impedance).

The advantages of using the HP 5610A were:

(1) The HP 5610A was compatible with the HP 2116C.

(2) The A/D converter was not currently being used in an active Navy program.

(3) The HP 5610A was easy to program.

(4) The 10 bit resolution met Chapter 2 data requirements.

(5) The HP 5610A was physically compatible for rack mount in HP 2116C equipment bay of the GMTS.

The disadvantages were:

(1) The HP 5610A was limited to 8 channel (16 channel expansion capability) operation.

(2) The HP 5610A was limited to ± 10V input range.

3.1.6 System 256-Data Acquisition System

The System 256 DAS (Data Acquisition System) manufactured by Datel Systems, Inc. is a complete computer input-output system for analog signals. The system uses modules to enable the user to solve any problem economically, while allowing future expansion flexibility.

The system itself is capable of handling up to 256 A/D channels and 64 D/A (Digital to Analog) channels. The system can be purchased with only 32 A/D channels and without any D/A capability. This system more than meets all of the missile data requirements stated in Chapter 2.

The advantages of using the System 256 were:

(1) The system more than met Chapter 2 requirements.

(2) Future expansion was easy.

(3) The system had Hybrid computer interfacing capability.

(4) The system did not have to have ADPE (Automatic Data Processing Equipment) approval.

The disadvantages were:

(1) The Datel system would have to be purchased. It would cost a minimum of $5000-6000 dollars.

(2) A software package would have to be developed or bought for the PDP-11 (CPU). The System 256 interfaces with PDP-11 computers.

3.2 Design of Experiments

An experiment was designed to obtain the frequency and amplitude of several missile performance parameters. The missile data characteristics listed in Table 1, (Chapter 2) were obtained from the PHOENIX LIS (Laboratory Integrated System). LIS is a Hybrid simulation of the PHOENIX missile using real missile hardware. The signal characteristics listed in Table 1 were a "best guess" estimate from the LIS laboratory. It was felt a test would help substantiate or invalidate the frequency measurements listed in Table 1.

A test was run on the PHOENIX GMTS to stimulate the guidance and control systems of the missile. The response signals listed in Table 2 were recorded on a strip chart recorder. Amplitude measurements were simultaneously taken by the GMTS.

MISSILE RESPONSE SIGNALS
FOR MST

MAX. AMPLITUDE	NAME	UP	RTN
± 18 VDC	Azimuth Steering Command (Vsh)	167	146
± 13.5 VDC	Azimuth Error (Vrh)	39	40
11 VDC	Prelaunch Phaselock	19	146
± 15 VDC	VCLO Control Voltage	114	40
± 10 VDC	Pitch Accelerometer Error	139	67
± 10 VDC	YAW Accelerometer Error	141	67
> 4.5 VDC	Transmitter-Oscillator MOAT Output	96	187
± 10 VDC	Roll Attitude	30	67
12 VDC	Timer Zero Set	137	67

Table 2 - Missile Response Signals for MST

Frequency calculations were then performed on the VCLO (Voltage Controlled Local Oscillator) control voltage of the missile. During velocity search and acquisition, the VCLO control voltage changes at the fastest rate of any of the missile response signals. Calculation of the VCLO control voltage search frequency yielded approximately a 20 Hz waveform. (See Appendix B).

The difference between the calculated value and the Table 1 value of the control voltage waveform freqeuncy was thought to be the difference between the actual VCLO search rate frequency, and what was thought to be a good engineering guess as to the "steady state" tracking rate frequency.

The 200 Hz signals listed in Table 1 were of even less frequency than the VCLO control voltage frequency. It was not determined why the Table 1 values did not agree with the findings of this experiment.

Considering the worst case, if there actually were analog signals of a 200 Hz frequency, the A/D converter has a maximum sampling rate of 3.5 KHz per channel (17 times more). The 200 Hz signals could still be digitized and reproduced with 17 samples per cycle.

3.2.1 Signal Degradation

After the coaxial cables had been cut, laid in cable trays, and appropriate connections made between the GMTS patch panel and the TCR (Test Control Room) patch panels, experiments were conducted to determine signal degradation because of the additional load on the missile from the cables.

Since typical missile response signals are of analog nature, (high signal level, low frequency) only two types of signal degradation were thought likely to occur: attenuation (because of loading effects), and additional noise.

A sinewave generator was used to determine signal attenuation from line capacitance and resistance of the cables. No observable attenuation occurred between 0-300 KHz. Above 300 KHz the sinewave generator did not maintain constant amplitude. No further attenuation measurements were performed since the frequency range of missile response signals was 0-200 Hz.

Signal measurements were made to determine the minimum signal that could be transmitted down the lines with a good signal to noise ratio. The signal was less than the 20 mv A/D resolution. Most missile responses occur in the 500 mv to 10 VDC range. Connecting the coaxial cables, tape recording equipment, and buffer amplifiers added no discernible noise or distortion. Overshoot of 1 VDC squarewaves (3 KHz) was less than 10%. In summary, signal degradation to expected missile response signals was negligible.

Several missile tests were run with the response data being directly fed to the A/D converter and digitized. Several more missile tests were run with the response signals digitized by the A/D converter after being recorded on a tape recorder. Data compilations of the two types data showed no difference between the two types of data.

3.2.2 Simple Assembly Programs

In order to perform the necessary analog to digital signal conversion of missile responses, the A/D driver computer program had to be written in assembler language. Assembler language is the "simplest" language for a computer to follow and thus the most efficient. The maximum sampling rate as determined from Chapter 2 data requirements would have to be implemented into a software program, in order to satisfactorily reproduce the missile's performance parameters accurately.

Two simple assembler language programs were written to learn how to write assembler language programs, and thus ultimately write the assembler A/D Driver. The first assembler program written was a simple program to punch onto paper tape, the numbers 1-256 decimal. The second program consisted of a FORTRAN program and an assembler program. This program allowed a driver for the paper tape punch to be called from FORTRAN. The A/D driver for the PHOENIX MST is called from FORTRAN, and this small program was a learning step toward writing the MST A/D driver. Appendix A contains listings of the assembler language listings.

3.3 Formulate Design

3.3.1 Choose Instrumentation System

Consolidating the hardware investigation findings (Chapter 3) and comparing them to the missile data requirements (Chapter 2), the following hardware options were open:

Computers: PDP-11 - Available

HP 2116C - Available

No option was considered for buying another computer because of funding constraints.

A/D Converters: HP 5610A - Available

Datel System 256 - Not available (must be used with PDP-11).

Applying the funding constraint stated in Chapter 1 a compromise was made to meet missile data requirements. Since virtually no funding was available for procurement of new equipment, the choice was limited to using the HP 5610A A/D. The HP A/D currently had only an eight channel multiplexer. If the MST proved to be a useful tool in fault isolation, funding would be made available to expand the HP 5610A to 16 channels. With sixteen channels, all the MOAT (Missile-On-Aircarft-Test) parameters could be continuously monitored during GLAT. MOAT uses 14 unique signals. Although the 16 channel capability was significantly less than the ideal 64, missile response could still be determined, but without any room for more detailed analysis or flexibility for other missile systems.

A comment must be stated about MST funding. The job task was funded for a five man-month period (approximately six calendar months). Coaxial cable, patch cords, miscellaneous connectors, and a patch panel, were purchased in order to "patch in" the MST A/D converter, and magnetic tape unit into the MECCA system. The money for the above hardware came from the MST general fund. Although the MST project was funded for labor charges only, approximately $500.00 was "sprung" loose to buy the necessary hardware. More money could have been made available for new equipment if sufficient need could be justified. The hardware investigation led to the conclusion that with 16 channel capability, most of the missile data requirements could be met. (See Chapter 4 Results and Conclusions).

3.3.2 Hardware and Equipment

The hardware necessary for eight channel data processing was the following:

HP 21566A	I/O Interface card
Coaxial Cable	2000 feet (Allowed for two spare cables)
Pitch cords (36")	25
Pin jacks	25
Patch panel	Designed and made at PMTC
Miscellaneous connectors	10
Miscellaneous hardware (Labeling, screws, nuts, insulation, ties, etc)	

The equipment necessary for eight channel data processing was the following:

HP 2116C Minicomputer

HP 5610A A/D Converter

VR-3400 Tape Recorder

Dynamic Filters and Differential amplifiers

Appendix C contains a complete list of all equipment that was used to implement the MST, and a complete list of equipment that was integrated into the MECCA system to be used as part of the MST.

3.3.3 Block Diagrams of the MST A/D Driver and MTP Programs

A block diagram of the MST A/D driver program was drawn to clarify the program structure of the A/D driver computer program. Figure 6 shows the structure of the A/D programs.

A block diagram of the MST Missile Test Program was also

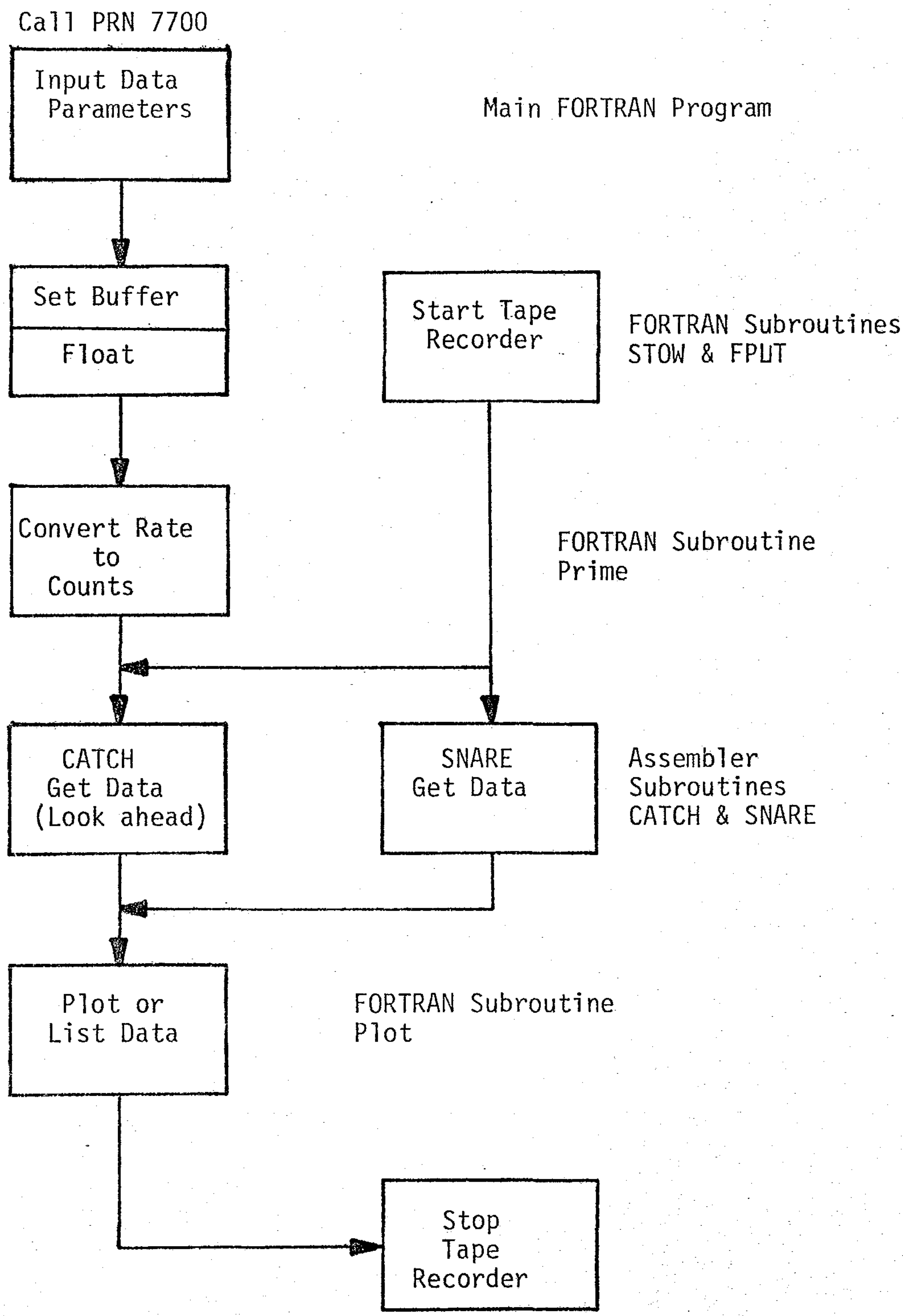


Figure 6. - Block Diagram of A/D Driver Program

drawn to clarify the structure of the MTP. Figure 7 shows the structure of the MTP.

3.4 Implement Design

3.4.1 Description of the MST A/D Driver

A flow diagram was constructed of the MST A/D driver program. The computer program consists of a main FORTRAN program, four FORTRAN subroutines, and two assembler subroutines. The driver is constructed so that the user can interplay with the computer console and obtain a great amount of information from the data that is digitized. Appendix A contains the A/D driver program listings.

The A/D driver is initiated by calling PRN (Program Reference Number) 7700. A title block is displayed on the CRT and the user must specify his data sampling requirements. The data is then automatically digitized and can be either listed or plotted on the appropriate output device. For operational instructions, refer to Appendix D.

After the MST A/D driver program was written and debugged, of complier errors, one channel of the A/D converter was connected to a signal generator. Since the A/D converter is programmed to sample at a rate specified by the user, it was necessary to make sure whatever sample rates and delay times were specified, are reproduced accurately by the computer. The computer reference oscillator was used as a clock. The sampling rate was converted to time between samples. The computer oscillator would "count down" so many milliseconds, and then take a sample. After again "counting down", the computer would take another sample, etc.

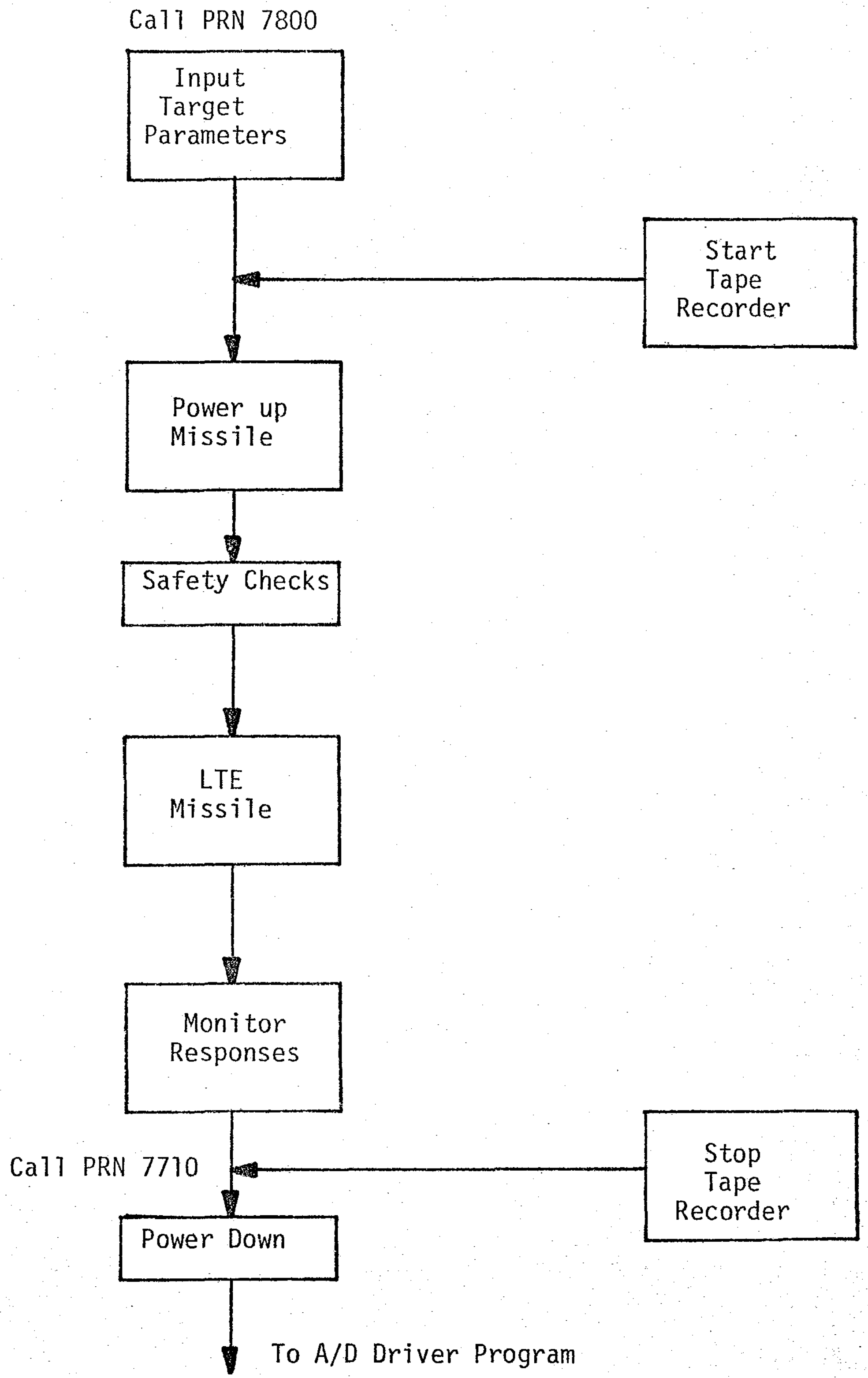


Figure 7. - Block Diagram of Missile Test Program

The A/D driver also has a delay time feature. The A/D will wait 0 to 30 seconds after the trigger occurs before sampling. The computer oscillator was used in the same manner, to count down until the specified time lasped, before taking samples of missile response data.

When fine tuning the sample time and delay time loops, it was discovered that the polarity of the DATA READY FLAG was opposite of what the converter required. Modifying the card to the proper logic level eliminated timing problems that were occurring in the timing loops. By using the reference oscillator, the timing loops in the A/D driver program have an accuracy of $\pm$ 1.6 usec. Appendix B contains calculations and procedures used to accurately determine timing parameters so the timing loops could be accurately programmed.

3.4.2 Description of the MTP

The MTP was written to provide maximum flexibilty for putting in target parameters i.e. (doppler frequency, signal strength, rear doppler, etc). All target and missile parameters necessary for launch are inputted via the computer console. This allows for a greater flexibility in providing launch parameters. Appendix A contains the program listing of the MTP.

After all the launch parameters have been set, the missile is powered up and a number of safety checks are made. (If any unsafe condition exists, the missile is powered down automatically.) The missile is executed through a LTE (Launch to Eject) cycle and is allowed to search for a target. The missile should track the target supplied via RF horns, and the missile response signals are recorded.

The most distinguishing feature of the MST is that the missile timer which controls all control system functions, is turned on. With the missile timer going, the missile guidance and control systems are allowed to run through the MBAM (Main Beam Avoidance Maneuver), mid-course guidance, and terminal guidance phases of flight.

The MST monitors the following missile parameters in a sequential manner:

Feedback Pots 1-4

Azimuth Steering Vsh

Elevation Steering Vsg

SAOT (Semi-Active-On-Target)

AOT (Active-On-Target)

And computes Feedback Pot rates 1-4.

The MST will monitor continuously any of the signals listed in Table 1. A "standard" set up for monitoring missile performance is composed of the following missile signals:

Detection Signal/Timer Zero Set (Trigger)

VCLO Control Voltage

Azimuth Steering Vsh

Elevation Steering Vsg

Autopilot MOAT Command

Pitch Accelerometer Error

YAW Accelerometer Error

Roll Attitude or AGC (Automatic Gain Control)

From the above signals, subsystem functions of the missile can be measured and determined to see how well the missile is working as a total system.

3.4.3 Data Analysis Programs

Once the missile performance data has been digitized, the data will be processed in at least two ways: Statistical calculations will be performed on each missile response to see if the response "matches up" with past responses (Discrete data points); and the entire signal will be compared to baseline responses to see if the missile performed correctly throughout the complete launch to intercept time.

The data comparison programs will not be written as part of this project. These programs are mentioned here to show how the missile data will be used. It is anticipated the programs will take the following format:

(1) A program to separate 8-16 channels of data.

(2) A "goodness of fit" test to baseline data.

(3) A timing correlation program to determine what subsystem in the missile was responsible for an unsatisfactory response.

3.5 Validating the MST

A simple procedure was created to validate the MTP and the A/D Driver programs. First, several missile tests would be run using a standard missile test. Second, several MTP's would be run, and compared to the standard missile test. If no difference in the performance data could be ascertained between the first and second set of tests, a third set of tests would be run and the missile responses recorded on the MST magnetic tape unit. The results from the second and third set of tests should be statistically the same if no degradation in missile response occurs. If there is no statistical difference in missile response, the MST would be considered valid.

Preliminary data indicates, no discernible difference in missile response using the A/D data conversion technique. Validating will continue to gather additional data using several missiles.

3.6 Future Development

The MECCA system is always being updated and its basic capability expanded. It is anticipated the MST described in this report will be expanded.

Two additions could be made to make the MST even more useful. It has already been mentioned that the data handling capability of the HP 5610A could be expanded from 8 channel to 16 channel merely by the addition of another 8 channel multiplexing card. Sixteen channel capability allows all fourteen MOAT signals to be monitored continuously and simultaneously as on the F-14A aircraft.

The second addition could be integration of a RFMTG into the MECCA system. The present PHOENIX GMTS provides a constant angle target. Providing a moving target both in velocity and angle will allow the missile to demonstrate its dynamic tracking ability better than it now can be monitored. By improving the basic capability of the MECCA system, GLAT can become more effective.

CHAPTER 4

RESULTS AND CONCLUSIONS

4.0 Results of Investigation

Although the funding constraint did not allow a great deal of choices between equipment, the following results were obtained from the investigation discussed in Chapter 3:

(1) Using the PDP-11 computer for data collection would be the most versatile.

(2) A minimum of an additional 24K of core memory for the PDP-11 would be necessary to give the same data storage as using the HP 2116C.

(3) Using the PDP-11 system would make real time data processing possible.

(4) Tape recording the missile response data makes it unnecessary to perform real time data processing.

(5) The Datel system 256 analog-digital system is ideally suitable to use with the PDP-11 computer.

(6) The system 256 meets all data processing requirements and has the capability to handle future expansion.

(7) Using the HP 2116C computer for data collection does not allow real time data processing.

(8) The HP 5610A A/D converter does not allow for future expansion of data processing requirements.

(9) Using the HP 2116C and associated software package made the programming of the MTP and the A/D driver time effective.

4.1 Results of Experiments

The results of the experiments conducted in Section 3.2 yielded the following:

(1) The coaxial cables did not "load down" the missile response signals.

(2) The missile response signals could be recorded and digitized with negligible degradation.

(3) The MTP and A/D driver programs worked as planned.

(4) The PHOENIX test set was given the continuous data monitoring capability without modification.

(5) The continuous performance data was useful.

4.2 Questions Answered by this Report

General Design Questions

(1) Can the existing PHOENIX test set be used for A/D conversion? - Yes. The PHOENIX test was successfully used to monitor continuous performance data.

(2) Does the PDP-11 computer have the capability of processing data from the HP 5610A A/D converter? - Essentially No. Without software and interface circuitry, the PDP-11 could not be used to process the digitized data from the HP 5610A.

(3) Can the HP 2116C and the PDP-11 computers be used in combination to run missile tests and process missile response data? - The same answer as in two above; without PDP-11 software, no attempt was made.

(4) What tape recording capabilities presently exist? - For analog data recording, there were two magnetic tape units. One 7 channel, and one 14 channel. The 14 channel VR 3400 was chosen so that with expansion to more than 8 A/D channels, tape recordings could be made for up to 14 channels.

(5) Will the MST allow real time data processing? - The configuration chosen does not allow real time data processing.

(6) What computer language will be most useful for operator interplay? - The MTP employs BASIC language. BASIC is easy to modify on line. The A/D driver program was written in both FORTRAN and in assembler language. The operator cannot modify the A/D driver on line. The A/D driver was purposely programmed this way so that the precise timing loops could not be changed.

(7) How much computer memory will be available? - The PDP-11 system had less than 4K of memory. The HP 2116C had more than 6K, and could be programmed using the disk to give over 12K memory for data storage.

(8) Will the HP 5610A A/D converter handle the expected data rate? - The A/D driver has a 28 KHz sampling rate, more than necessary.

(9) Will a signal multiplexer be required? - Yes. The A/D converter uses an 8 channel multiplexer.

(10) What signal conditioning will be necessary? - No amplification of the missile response data was necessary for tape recording. The data is fed directly to the tape unit. No filters are necessary. Some scaling is necessary if the missile responses rise above $\pm$ 10

VDC.

(11) What signal conditioning capabilities presently exist in the laboratory? - Complete buffering, amplification, scaling and filtering are available.

Specific Design Questions

(1) What type coaxial cable will be used? - Aircraft wire type II, AWG 20 conforms to MIL-C-7078A.

(2) What missile response signals will be monitored? - Essentially any of the 200 umbilical signals can be measured, but Table 2 lists the response signals that will normally be monitored.

(3) What sampling rate and amplitudes will limit signal processing? - The HP 5610A is programmed to sample up to a 28 KHz sampling rate. Some amplitude scaling will be necessary if the response signal level exceeds ± 10 VDC.

(4) What trigger source will be used? - Either the Timer Zero Set signal, or the velocity Detection Signal will be normally used for a trigger source. However, any of the missile response signals can be used for a trigger.

4.3 Conclusions

The following conclusions can be drawn from this project report:

(1) The concept of using the HP 2116C computer as a missile controller and a data recovery tool is possible.

(2) Continuous data monitoring of missile performance data does not degrade the missiles responses.

(3) The GLAT specification requirement that the PHOENIX test

set not be modified has been met.

(4) The contractor's software can be used for testing missiles, and continuous data monitoring can be used without changing the contractor's software.

(5) The MST is a useful tool for fault isolation of missile problems.

LIST OF REFERENCES

1. NAVAIR, 16-30 DSM 130 Technical Manual, PHOENIX Test Set, Guided Missile, AN/DSM-130(V) Volumes I, II, III. 1 January 1974

2. NAVAIR, 01-AIM-54-2-1 Techncial Manual, Functional Description, PHOENIX Guided Missile AIM-54A 1 February 1974 CONFIDENTIAL

3. Hewlett-Packard, Automatic Test System Model No. 9206B System Configuration Manual, Binders 1,5,6, 10 October 1971

4. Hewlett-Packard, A Pocket Guide To Hewlett-Packard Computers Reference No. 5950-8313.

5. Hughes Aircraft Company, System Engineering Analysis Report Mechanization AIM-54A Missile, Volume IV, PMS 610-1/2174. February 1972 CONFIDENTIAL

6. Hughes Aircraft Company, Field Test Specification Model AIM-54A Guided Missile, TS31051-663 Revision D. 14 March 1975 CONFIDENTIAL

7. Pacific Missile Test Center, Interface of Analog to Digital Converter with MECCA Facility ITP-73-50 Point Mugu, California. 1 November 1973

8. Dital Equipment Corporation, Digital PDP-11/20 Processor Handbook D-09-25 Los Angeles California. 1972

9. Datel Systems, Inc., Engineering Product Handbook

10. Datel Systems, Inc., System 256 Data Acquisition System Brochure

11. Schwartz, Mischa, Information Transmission, Modulation, and Noise. New York McGraw-Hill 1959

APPENDIX A

Computer listing of two simple assembler language programs.

Computer listing of the Missile Test Program.

Computer listing of the A/D Driver Program.

PAGE 0001

```
0001                        ASMB,R,B,L,T
ALM    R 000010
TFS    R 000011
BEAN   R 000001
**  NO ERRORS*
```

PAGE 0002 #01

```
0001                        ASMB,R,B,L,T
0002  00000                  NAM LARRY
0003  00000 002400           CLA
0004  00001 052011R         BEAN CPA TFS
0005  00002 026010R          JMP ALM
0006  00003 103616           OTA 16B,C
0007  00004 102716           STC 16B
0008  00005 026004R          JMP *-1
0009  00006 002004           INA
0010  00007 026001R          JMP BEAN
0011  00010 102000          ALM HLT
0012  00011 000400          TFS DEC 256
0013                         END
**  NO ERRORS*
```

```
FTN,L,B
      DO 1 I=1,256
    1 CALL PNCH(I)
      STOP
      END
      ENDS
```

```
 PAGE 0001


0001                      ASMB,R,B,L,T
BCK    R 000020
RTN    R 000021
PNCH   R 000000
**  NO ERRORS*
```

```
 PAGE 0002 #01


0001                      ASMB,R,B,L,T
0002   00000               NAM PNCH
0003                       ENT PNCH
0004   00000 000000       PNCH NOP
0005   00001 072020R       STA BCK
0006   00002 162000R       LDA PNCH,I
0007   00003 072021R       STA RTN
0008   00004 036000R       ISZ PNCH
0009   00005 162000R       LDA PNCH,I
0010   00006 160000        LDA 0,I
0011   00007 103100        CLF 0
0012   00010 103616        OTA 16B,C
0013   00011 102716        STC 16B
0014   00012 102316        SFS 16B
0015   00013 026012R       JMP *-1
0016   00014 103116        CLF 16B
0017   00015 102100        STF 0
0018   00016 062020R       LDA BCK
0019   00017 126021R       JMP RTN,I
0020   00020 000000       BCK NOP
0021   00021 000000       RTN NOP
0022                       END
**  NO ERRORS*
```

```
1  COM D[53],H[20],V[25],E[25],K[20],L[59],N[5],P[9]
2  REM MST(MISSION SIMULATION TEST) MISSILE TEST PROGRAM
3  REM MTP DESIGNED BY L.E. SELLERS
4  REM  PRN 7800
5  GOSUB 8700
20  SMODSW (0,D[51],E)
25  LET R=0
30  LET A=28
35  GOSUB 5045
90  READ P[1],P[3],P[4],P[5],P[7]
100  DATA 0,0,0,1,41
105  DATA 1,203,0,2,209,201
110  PRINT "FOR STT MODE INPUT (1), FOR SDM INPUT (0)"
115  PRINT
120  INPUT P[2]
130  PRINT "INPUT VELOCITY UPDATE DOPPLER, RANGE -10 TO 99.65"
140  PRINT "INPUT TARGET DOPPLER, RANGE -62 TO 193"
150  PRINT "INPUT REAR DOPPLER, RANGE -127 TO 128"
155  PRINT
160  INPUT D3,D9,D8
170  GOSUB 6000
215  LET D6=D9
220  LET E1=40
230  LET E2=1
240  LET E3=62
250  GOSUB 7000
255  LET D6=D8
260  LET E1=20
270  LET E2=17
280  LET E3=127
290  GOSUB 7000
700  PRINT " WHAT S/A CHANNEL DO YOU WISH? "
702  PRINT " S/A CHN. #1 -- TYPE IN 65 "
704  PRINT " S/A CHN. #2 -- TYPE IN 66 "
706  PRINT " S/A CHN. #3 -- TYPE IN 67 "
708  PRINT " S/A CHN. #4 -- TYPE IN 68 "
710  PRINT " S/A CHN. #5 -- TYPE IN 69 "
712  PRINT " S/A CHN. #6 -- TYPE IN 70 "
713  PRINT
714  INPUT P[6]
715  PRINT "CMD VEL="D3;"TARGET DOP="D9;"REAR DOP="D8
716  PRINT "P(1)="P[1];"P(2)="P[2];"P(3)="P[3];"P(4)="P[4]
717  PRINT "P(5)="P[5];"P(6)="P[6];"P(7)="P[7];"P(8)="P[8];"P(9)="
718  PRINT
720  IF P[6]=65 THEN 734
722  IF P[5]=66 THEN 738
724  IF P[6]=67 THEN 742
726  IF P[6]=68 THEN 746
728  IF P[6]=69 THEN 750
730  IF P[6]=70 THEN 754
732  IF P[6]<65 OR P[6]>70 THEN 120
734  GOSUB 3000
736  GOTO 756
738  GOSUB 3010
740  GOTO 756
742  GOSUB 3030
744  GOTO 756
746  GOSUB 3030
748  GOTO 756
750  GOSUB 3040
```

```
752  GOTO 756
754  GOSUB 3050
756  REM MSL ADD
758  DATA 2,150,152,0
760  REM A T BIAS
762  DATA 0,0,0,0,0
772  FOR I=1 TO 6
774  READ L[I]
776  NEXT I
778  FOR J=11 TO 19
779  READ L[J]
780  NEXT J
783  FOR I=1 TO 59
784  PRINT L[I];
786  NEXT I
790  PRINT
830  LET A6=99
835  PRINT "TYPE IN TARGET ATTN (0 TO 99)"
840  INPUT Q
845  LET A3=ABS(Q)
847  PRINT "Q= "Q,"A3= "A3
849  PRINT
850  GOSUB 5550
1055  SMODSW (1,9,E)
1060  SMODSW (1,69,E)
1075  LET R=1300
1080  GOSUB 8600
1085  PAUSE
1100  LINK 7810
3000  REM S/A CHANNEL SELECT
3001  LET L[7]=1
3002  LET L[8]=95
3003  L^T L[9]=0
3004  LET L[10]=0
3005  RETURN
3010  LET L[7]=2
3011  LET L[8]=95
3012  LET L[9]=128
3013  LET L[10]=0
3015  RETURN
3021  LET L[8]=95
3022  LET L[9]=149
3023  LET L[10]=0
3025  RETURN
3030  LET L[7]=3
3031  LET L[8]=95
3032  LET L[9]=128
3033  LET L[10]=149
3035  RETURN
3040  LET L[7]=2
3041  LET L[8]=95
3042  LET L[9]=168
3043  LET L[10]=0
3045  RETURN
3050  LET L[7]=3
3051  LET L[8]=95
3052  LET L[9]=128
3053  LET L[10]=168
3055  RETURN
5045  REM PWR ON/OFF TIME
```

```
5046  SCOM (0,S,A,E)
5047  CLKRD (D1,H1,M1,S1,E)
5048  LET S1=H1*3600+M1*60+INT(S1+.5)
5049  IF R=0 AND S1>S OR R=1 THEN 5051
5050  LET S1=S1+86400
5051  LET S=S1-S
5052  SCOM (1,S,A,E)
5053  RETURN
5550  REM TARGET ATTN
5551  IF A3<0 OR A3>99 THEN 835
5552  MODSW (2,170,E)
5553  LET X1=INT(A6/10)
5554  LET X2=INT(A3/10)
5555  LET X3=A6-(10*X1)
5556  LET X4=A3-(10*X2)
5557  IF X1=X2 THEN 5560
5558  IF (X1<X2 AND X1+5 >= X2) OR (X1>X2 AND X1-5 >= X2) THEN 556
5559  MODSW (1,187,E)
5560  IF X3=X4 THEN 5563
5561  IF (X3<X4 AND X3+5 >= X4) OR (X3>X4 AND X3-5 >= X4) THEN 556
5562  MODSW (1,171,E)
5563  WAIT (20)
5564  MODSW (1,177+X2,E)
5565  MODSW (1,161+X4,E)
5566  LET A6=A3
5567  RETURN
6000  LET D5=(D3+10.2)/.51
6005  LET N1=D5/36
6010  LET N2=D5-INT(N1)*36
6015  LET N3=N2/6
6020  LET N4=N2-INT(N3)*6
6025  FOR C=1 TO 3
6030  IF C>1 THEN 6045
6035  LET N6=INT(N4)
6040  GOTO 6065
6045  IF C>2 THEN 6060
6050  LET N6=INT(N3)
6055  GOTO 6065
6060  LET N6=INT(N1)
6065  IF N6>2.5 THEN 6080
6070  LET N[C]=3*(N6↑2)-5*N6+4
6075  GOTO 6085
6080  LET N[C]=-3*(N6↑2)+25*N6-47
6085  NEXT C
6090  LET P[8]=500+10*N[1]+N[2]
6095  LET P[9]=10000*N[3]+6524
6097  RETURN
7000  LET D7=D6+E3
7004  LET C=11
7006  LET D=1
7008  LET L[E1+1]=L[E1+11]=43
7010  FOR N=7 TO 0 STEP -1
7012  IF D7 >= 2↑N THEN 7020
7014  LET C=C+1
7016  LET L[E1+C]=N+E2
7018  GOTO 7026
7020  LET D=D+1
7022  LET L[E1+D]=N+E2
7024  LET D7=D7-2↑N
7026  NEXT N
```

```
5046  SCOM (0,S,A,E)
5047  CLKRD (D1,H1,M1,S1,E)
5048  LET S1=H1*3600+M1*60+INT(S1+.5)
5049  IF R=0 AND S1>S OR R=1 THEN 5051
5050  LET S1=S1+86400
5051  LET S=S1-S
5052  SCOM (1,S,A,E)
5053  RETURN
5550  REM TARGET ATTN
5551  IF A3<0 OR A3>99 THEN 835
5552  MODSW (2,170,E)
5553  LET X1=INT(A6/10)
5554  LET X2=INT(A3/10)
5555  LET X3=A6-(10*X1)
5556  LET X4=A3-(10*X2)
5557  IF X1=X2 THEN 5560
5558  IF (X1<X2 AND X1+5 >= X2) OR (X1>X2 AND X1-5 >= X2) THEN 5560
5559  MODSW (1,187,E)
5560  IF X3=X4 THEN 5563
5561  IF (X3<X4 AND X3+5 >= X4) OR (X3>X4 AND X3-5 >= X4) THEN 5563
5562  MODSW (1,171,E)
5563  WAIT (20)
5564  MODSW (1,177+X2,E)
5565  MODSW (1,161+X4,E)
5566  LET A6=A3
5567  RETURN
6000  LET D5=(D3+10,2)/.51
6005  LET N1=D5/36
6010  LET N2=D5-INT(N1)*36
6015  LET N3=N2/6
6020  LET N4=N2-INT(N3)*6
6025  FOR C=1 TO 3
6030  IF C>1 THEN 6045
6035  LET N6=INT(N4)
6040  GOTO 6065
6045  IF C>2 THEN 6060
6050  LET N6=INT(N3)
6055  GOTO 6065
6060  LET N6=INT(N1)
6065  IF N6>2.5 THEN 6080
6070  LET N[C]=3*(N6↑2)-5*N6+4
6075  GOTO 6085
6080  LET N[C]=-3*(N6↑2)+25*N6-47
6085  NEXT C
6090  LET P[8]=500+10*N[1]+N[2]
6095  LET P[9]=10000*N[3]+6524
6097  RETURN
7000  LET D7=D6+E3
7004  LET C=11
7006  LET D=1
7008  LET L[E1+1]=L[E1+11]=43
7010  FOR N=7 TO 0 STEP -1
7012  IF D7 >= 2↑N THEN 7020
7014  LET C=C+1
7016  LET L[E1+C]=N+E2
7018  GOTO 7026
7020  LET D=D+1
7022  LET L[E1+D]=N+E2
7024  LET D7=D7-2↑N
7026  NEXT N
```

```
7028  FOR N=E1+C+1 TO E1+19
7030  LET L[N]=0
7032  NEXT N
7034  FOR N=E1+D+1 TO E1+9
7036  LET L[N]=0
7038  NEXT N
7040  LET L[E1]=D
7042  LET L[E1+10]=C-10
7050  RETURN
8600  REM CHECK HPS THERM
8601  PRINT "LOOP UNTIL THERM >"R
8602  PRINT
8603  PRINT "TIME";"    THERM"
8604  MXBAR (76,144,E)
8605  DVMSU (1,1,1,E)
8606  LET T=0
8607  WAIT (200)
8608  DVMMU (1,V,1,E)
8609  IF V<.669 THEN 8611
8610  LET V=.669
8611  LET R1=(V*2500)/(.67-V)
8612  LET T=T+10
8613  PRINT T;R1
8614  IF R1>R THEN 8617
8615  WAIT (9900)
8616  GOTO 8608
8617  RETURN
8700  REM HEADER
8701  CRT (-1,0,0,E)
8702  DUPRINT (1)
8703  PRINT
8704  PRINT
8705  PRINT
8706  PRINT "                    ************************************"
8708  PRINT "                    *                                  *"
8710  PRINT "                    *                                  *"
8712  PRINT "                    * PHOENIX MISSION SIMULATION TEST  *"
8713  PRINT "                    *                                  *"
8715  PRINT "                    *       MISSILE TEST PROGRAM       *"
8717  PRINT "                    *                                  *"
8720  PRINT "                    *                                  *"
8722  PRINT "                    ************************************"
8724  PRINT
8726  PRINT
8748  DUPRINT (0)
8750  RETURN
8752  END
```

```
10  COM D[53],H[20],V[25],E[25],K[20],L[59],N[5],P[9]
15  REM PRN 7810
20  GOSUB 8800
30  GOSUB 7500
40  GOSUB 5670
42  PRINT "SAFETY CHECKS HAVE BEEN MADE"
45  PRINT
50  LET R=0
55  LET A=28
60  GOSUB 5045
100  LET L1=P[1]
110  LET L2=P[2]
120  LET L3=P[3]
130  LET L4=P[4]
140  LET L5=P[5]
150  LET L6=P[6]
160  LET L7=P[7]
170  LET L8=P[8]
180  LET L9=P[9]
1155  GOSUB 6200
1160  DATA 19,234,233,232,231,230,228,227,226,243,239,238,237,236
1165  DATA 246,253,252,251,250,249
1170  GOSUB 5200
1175  GOSUB 7220
1180  SMODSW (1,102,E)
1185  WAIT (20)
1190  SMODSW (1,47,E)
1195  WAIT (20)
1200  GOSUB 6200
1205  DATA 4,102,47,18,137
1210  SMODSW (0,K[1],E)
1225  CLKTI (1,0,E)
1230  DVSHIR (2,.4,20,E)
1250  DVSHIR (3,-.4,20,E)
1270  DVSHIR (1,-1.5,20,E)
1275  GOSUB 6200
1280  DATA 8,214,52,166,24,213,35,224,60
1285  SMODSW (1,K[1],E)
1287  MODSW (1,42,E)
1290  FOR I=1 TO 22
1295  READ L[I]
1300  NEXT I
1305  MODSW (1,43,E)
1310  MODSW (1,L[7],E)
1315  MODSW (0,L[1],E)
1320  MODSW (1,L[11],E)
1325  MODSW (0,L[18],E)
1326  SMODSW (1,61,E)
1330  DATA 5,39,38,37,35,33,3,40,36,34
1340  DATA 6,43,7,6,5,4,2,4,8,3,1,43
1350  DVMSU (1,10,0,E)
1355  MXBAR (89,67,E)
1357  LET F7=L9/10000
1358  LET F7=INT(F7)*10000
1360  LET L9=63354
1362  GOSUB 3000
1365  CLKTI (2,T,E)
1370  WAIT ((1.98-T)*1000)
1375  DVMMU (2,V[1],500,E[1])
1385  WAIT (450)
```

```
1390  MODSW (1,53,E)
1395  RG40M (1,L8,L9,1,E)
1400  MXBAR (107,-1,E)
1410  WAIT (895)
1415  DVMMU (2,V[3],500,E[3])
1425  WAIT (460)
1430  RG40M (1,L8,L9,1,E)
1435  MXBAR (167,146,E)
1440  WAIT (330)
1445  DVMMU (1,V[5],1,E[5])
1450  MXBAR (148,-1,E)
1460  DVMMU (1,V[6],1,E[6])
1465  MXBAR (122,67,E)
1475  WAIT (470)
1480  DVMMU (2,V[7],500,E[7])
1490  WAIT (155)
1495  RG40M (1,L8,L9,1,E)
1500  MXBAR (123,-1,E)
1510  WAIT (195)
1515  DVMMU (2,V[9],500,E[9])
1517  LET F7=L9/10000
1518  LET F7=INT(F7)*10000
1520  LET L9=66544
1522  GOSUB 3000
1525  GOSUB 6200
1530  DATA 8,52,166,9,35,24,15,213,214
1535  SMODSW (0,K(1),E)
1540  SMODSW (1,68,E)
1545  SMODSW (1,48,E)
1550  GOSUB 8400
1555  DVMSU (1,10,0,E)
1560  MXBAR (82,146,E)
1565  CLKTI (2,T,E)
1570  WAIT ((8.94-T)*1000)
1575  FOR C=11 TO 14
1580  RG40M (1,L8,L9,1,E)
1585  WAIT (320)
1590  DVMMU (1,V[C],1,E[C])
1595  IF C=14 THEN 1665
1600  WAIT (1635)
1605  NEXT C
1665  LET V[15]=5
1670  LET E[15]=2*SGN(E[11]+E[12]+E[13]+E[14])
1675  IF V[11]>1 OR V[12]>1 OR V[13]>1 OR V[14]>1 THEN 1700
1680  LET V[15]=0
1685  SMODSW (1,18,E)
1700  MODSW (0,183,E)
1710  MODSW (1,182,E)
1715  CLKTI (2,T,E)
1720  WAIT ((16.94-T)*1000)
1725  RG40M (1,L8,L9,1,E)
1730  WAIT (320)
1735  DVMMU (1,V[16],1,E[16])
1737  LET F7=L9/10000
1738  LET F7=INT(F7)*10000
1740  LET L9=66542
1742  GOSUB 3000
1745  WAIT (1640)
1750  RG40M (1,L8,L9,1,E)
1755  WAIT (320)
```

```
1760  DVMMU (1,V[17],1,E[17])
1770  WAIT (1640)
1775  RG40M (1,L8,L9,1,E)
1780  DVMSU (1,10,1,E)
1785  MXBAR (131,-1,E)
1795  WAIT (1200)
1800  DVMMU (1,V[18],1,E[18])
1802  DVMSU (1,10,2,E)
1805  MODSW (2,170,E)
1807  WAIT (15)
1810  MODSW (1,183,E)
1815  MODSW (1,163,E)
1820  SMODSW (1,19,E)
1825  SMODSW (1,99,E)
1830  CLKTI (2,T,E)
1835  WAIT ((22.94-T)*1000)
1840  RG40M (1,L8,L9,1,E)
1855  WAIT (2000)
1860  DVMMU (1,V[19],1,E[19])
1862  MODSW (1,187,E)
1865  MODSW (0,183,E)
1870  MODSW (1,180,E)
1871  WAIT (8000)
1875  DVSHIR (2,6,20,E)
1880  DVSHIR (3,6,20,E)
1885  GOSUB 6200
1890  SMODSW (1,K[1],E)
1895  DATA 4,219,220,198,162
1900  MXBAR (148,146,E)
1910  WAIT (3000)
1911  LET V[20]=0
1912  FOR C=1 TO 50
1913  DVMMU (1,V,1,E[20])
1914  LET V[20]=V[20]+V
1915  WAIT (20)
1916  NEXT C
1917  LET V[20]=V[20]/50
1920  MXBAR (167,-1,E)
1925  WAIT (1000)
1926  LET V[21]=0
1927  FOR C=1 TO 50
1928  DVMMU (1,V,1,E[21])
1929  LET V[21]=V[21]+V
1930  WAIT (20)
1931  NEXT C
1932  LET V[21]=V[21]/50
1940  GOSUB 6200
1946  DATA 13,198,162,61,69,48,68,219,220,19,99
1948  DATA 60,224,18
1950  SMODSW (0,K[1],E)
1951  SMODSW (2,225,E)
1952  WAIT (15)
1953  SMODSW (1,H[1],E)
1955  SMODSW (1,18,E)
1956  SMODSW (1,15,E)
1957  SMODSW (1,137,E)
1958  MODSW (1,41,E)
1959  WAIT (25)
1960  MODSW (4,1,E)
1965  MODSW (1,D[27],E)
```

```
1970  FOR I=1 TO 3
1975  DVSHIR (I,0,20,E)
1980  NEXT I
1981  LET E[22]=E[2]-E[1]
1982  LET E[23]=E[4]-E[3]
1983  LET E[24]=E[8]-E[7]
1984  LET E[25]=E[10]-E[9]
1985  LET G=1
1990  FOR C=1 TO 25
1995  IF C>21 THEN 2030
2005  READ A,B
2010  GOTO 2050
2030  LET G=7
2035  READ A,B,D,E,F
2045  LET V[C]=(V[D]-V[E])/(F)
2050  DPROC (182,G,A,B,V[C],E[C])
2055  NEXT C
2060  DATA 2.6,-1.4,2.1,-1.9,3.7,-1.5,4.2,-1,-3.25,-7.25
2065  DATA -2.5,-5.5,.4,-4.4,1.5,-3.5,3.4,-1.4,3,-3,6,-1
2070  DATA 6,-1,6,-1,6,-1,6,4,6,4
2075  DATA 6,4,1,-1,5,3,3,2,3,2
2080  DATA -2.50000E-02,-1.825,1,2,-.5,1.825,2.50000E-02,3,4,-.5
2085  DATA -.67,-3.27,7,8,.5,3.27,.67,9,10,.5
2090  LINK 1002
3000  REM CHANGE L9 TO PROPER WORD
3005  LET F6=L9/10000
3010  LET F6=INT(F6)*10000
3015  LET L9=L9-F6
3020  LET L9=L9+F7
3025  RETURN
5045  REM PWR ON/OFF TIME
5046  SCOM (0,S,A,E)
5047  CLKRD (D1,H1,M1,S1,E)
5048  LET S1=H1*3600+M1*60+INT(S1+.5)
5049  IF R=0 AND S1>S OR R=1 THEN 5051
5050  LET S1=S1+86400
5051  LET S=S1-S
5052  SCOM (1,S,A,E)
5053  RETURN
5200  REM LTE SETUP
5202  MODSW (0,D[44],E)
5203  WAIT (15)
5204  MODSW (1,L6,E)
5206  SMODSW (0,18,E)
5207  SMODSW (2,225,E)
5208  WAIT (15)
5209  SMODSW (1,K[1],E)
5215  MODSW (2,202,E)
5216  WAIT (15)
5217  IF L[1]=0 THEN 5220
5218  MODSW (1,L[1],E)
5219  WAIT (15)
5220  MODSW (1,L[4],E)
5222  SMODSW (0,D[5],E)
5223  WAIT (15)
5224  SMODSW (1,L[7],E)
5226  MODSW (1,L[20],E)
5227  MODSW (0,L[30],E)
5229  MODSW (1,L[40],E)
5230  MODSW (0,L[50],E)
```

```
5231  MODSW (1,71,E)
5235  SMODSW (1,D[34],E)
5236  MODSW (1,41,E)
5237  WAIT (25)
5238  MODSW (0,41,E)
5239  PRINT "LTE INIT"
5240  CLKTI (1,0,E)
5241  WAIT (50)
5242  SMODSW (1,D[1],E)
5243  WAIT (30)
5244  IF L5=0 THEN 5249
5245  SMODSW (1,D[51],E)
5246  LET A=28
5247  LET R=1
5248  GOSUB 5045
5249  CLKTI (2,L[18],E)
5250  WAIT ((1.487-L[18])*1000)
5251  SMODSW (0,115,E)
5253  SMODSW (1,D[40],E)
5254  CLKTI (2,L[19],E)
5255  WAIT ((1.767-L[19])*1000)
5258  SMODSW (1,L[11],E)
5262  IF L3=0 THEN 5264
5263  SMODSW (1,175,E)
5264  WAIT (10)
5265  SMODSW (1,47,E)
5266  WAIT (10)
5267  SMODSW (0,D[9],E)
5268  WAIT (10)
5269  SMODSW (0,47,E)
5271  RG40M (1,0,L7,2,E)
5272  WAIT (15)
5273  SMODSW (0,D[34],E)
5275  RG40M (1,L8,L9,3,E)
5282  CLKTI (2,X,E)
5283  WAIT ((2.687-X)*1000)
5285  PRINT "LTE COMPLETE 3"
5287  SMODSW (0,100,E)
5288  WAIT (50)
5289  MODSW (0,71,E)
5291  SMODSW (0,D[40],E)
5292  SMODSW (0,95,E)
5293  WAIT (10)
5294  SMODSW (0,D[5],E)
5307  DVMSU (1,10,0,E)
5308  RETURN
5870  REM RLM PWR CHK
5871  MXBAR (54,35,E)
5872  DVMSU (1,100,1,E)
5873  WAIT (200)
5874  DVMMU (1,V,1,E)
5875  IF V>90 THEN 5880
5876  PRINT "POWER HAS NOT BEEN APPLIED"
5877  SCOM (1,1,49,E)
5878  SCOM (1,710,20,E)
5879  LINK 1002
5880  RETURN
6200  READ K[1]
6201  FOR I=2 TO K[1]+1
6202  READ K[I]
```

```
6203  NEXT I
6205  RETURN
7220  PRINT "HPS PWR-UP ROUTINE"
7221  DVMSU (1,1,1,E)
7222  MXBAR (243,243,E)
7223  WAIT (200)
7224  DVMMU (1,V1,1,E)
7226  DVMSU (1,100,1,E)
7227  MXBAR (199,163,E)
7235  SMODSW (1,182,E)
7236  LET R=1
7237  LET A=30
7238  GOSUB 5045
7242  WAIT (1000)
7243  DVMMU (1,V,1,E)
7244  LET L0=0
7245  IF ABS(V-27)>3 THEN 7258
7246  MXBAR (243,243,E)
7247  DVMSU (1,1,1,E)
7248  WAIT (200)
7249  DVMMU (1,V,1,E)
7250  LET A1=((15*V)/V1)-15
7251  LET L0=1
7252  IF A1>150 THEN 7258
7254  PRINT "HPS TURN ON COMP."
7256  RETURN
7258  GOSUB 8400
7259  IF L0=1 THEN 7270
7260  PRINT "HPS INPUT VOLTS OUT OF SPEC=";V;"VOLTS"
7261  SCOM (1,650,20,E)
7262  LET D[34]=V
7265  GOTO 7290
7270  PRINT "HPS CURRENT HIGH=";A1;"AMPS"
7275  SCOM (1,600,20,E)
7276  LET D[35]=A1
7290  SCOM (1,2,49,E)
7300  LINK 1002
7500  DVMSU (1,10,1,E)
7501  MXBAR (92,67,E)
7502  WAIT (200)
7503  DVMMU (1,V,1,E)
7504  IF ABS(V-12)>1 THEN 7510
7505  MXBAR (274,25,E)
7506  WAIT (200)
7507  DVMMU (1,V,1,E)
7508  IF ABS(V-2.24)>1 THEN 7510
7509  RETURN
7510  LINK 7503
8400  REM HPS TURN OFF
8401  SMODSW (0,182,E)
8402  LET R=0
8403  LET A=30
8404  GOSUB 5045
8405  DVMSU (1,100,1,E)
8406  MXBAR (199,163,E)
8407  WAIT (200)
8408  DVMMU (1,V,1,E)
8409  IF V<10 THEN 8413
8410  SCOM (1,680,20,E)
8411  SCOM (1,2,49,E)
```

```
8412  LINK 1002
8413  PRINT "HPS TURN OFF COMPLETE"
8414  RETURN
8800  REM HEADER
8805  DUPRINT (0)
8810  PRINT
8815  CLKRD (D,H,M,S,E)
8820  LET H=(H*100)+M
8825  PRINT "RUN DAY="D;" RUN TIME="H
8830  SCOM (1,H,174,E)
8835  DUPRINT (0)
8840  RETURN
8850  END
```

```
FTN,B,L
      PROGRAM MST/DRIVER
      DIMENSION BUF(6380)
C     NEED BUF(12760)
   90 FORMAT(A2, "←")
      WRITE(2,90)7455,7967
      DO 92 I=1,10
      WRITE(2,90)1799
   92 WRITE(2,90)7455,7967
   60 WRITE(2,260)
  260 FORMAT(20X,40H****************************************)
  265 WRITE(2,270)
  270 FORMAT(20X,1H*,38X,1H*)
  275 WRITE(2,280)
  280 FORMAT(20X,1H*,38X,1H*)
  285 WRITE(2,290)
  290 FORMAT(20X,1H*,4X,31HPHOENIX MISSION SIMULATION TEST,3X,1H*)
  295 WRITE(2,300)
  300 FORMAT(20X,1H*,38X,1H*)
  305 WRITE(2,310)
  310 FORMAT(20X,1H*,38X,1H*)
  315 WRITE(2,320)
  320 FORMAT(20X,1H*,8X,22HMISSILE DATA DIGITIZER,8X,1H*)
  325 WRITE(2,330)
  330 FORMAT(20X,1H*,38X,1H*)
  335 WRITE(2,340)
  340 FORMAT(20X,40H****************************************)
    1 WRITE(2,2)
    2 FORMAT(/"INPUT TOTAL SAMPLE TIME (TST) IN MS")
      READ(1,3)TST
    3 FORMAT(F10.0)
      WRITE(2,4)
    4 FORMAT("INPUT SAMPLING RATE (SR) SAMPLES/MILLISECOND")
      READ(1,3)SR
      LBUF=TST*SR+0.5
      IF (LBUF-6380)5,5,80
   80 WRITE(2,81)
   81 FORMAT(/"(TST)*(SR) CANNOT EXCEED 6380"/)
      GO TO 1
    5 WRITE(2,6)
    6 FORMAT("INPUT DELAY TIME (DT) IN MILLISECONDS")
      READ(1,3)DT
      IF(DT)7,12,12
    7 WRITE(2,8)
    8 FORMAT("INPUT TRIGGER CHANNEL NUMBER (ITCN) 0-7")
      READ(1,3)ITCN
    9 IF(ITCN)15,16,16
   12 WRITE(2,13)
   13 FORMAT("DO YOU WANT SEQUENTIAL OPERATION?")
   14 READ(1,37)IR
      IF(IR-89)21,23,21
   15 ITCN=0
      GO TO 30
   16 IDCN=ITCN
      GO TO 30
   21 WRITE(2,22)
```

```
 22 FORMAT("INPUT DATA CHANNEL NUMBER (IDCN), 0-7")
    READ(1,3)IDCN
    IF(IDCN)21,23,23
 23 WRITE(2,55)
 55 FORMAT("INPUT TRIGGER CHANNEL NUMBER (ITCN),0-7")
 24 READ(1,3)ITCN
    IF (ITCN)23,70,70
 70 IF(IR-89)30,25,30
 25 IDCN=-1
 30 WRITE(2,31)
 31 FORMAT("INPUT TRIGGER LEVEL (TVL) IN VOLTS")
    READ(1,3)TVL
 75 CALL STOW(BUF,LBUF,DT,TST,TVL,ITCN,IDCN,IE)
    IF(IE)100,32,100
100 IF(IE-2)200,203,101
101 IF(IE-4)206,209,209
200 WRITE(2,201)
201 FORMAT(/"A/D DATA INCORRECT.  IS A/D ON AND IN REMOTE?"/)
    GO TO 1
203 WRITE(2,204)
204 FORMAT(/"SAMPLE RATE (SR) IS TOO LARGE"/)
    GO TO 1
206 WRITE(2,207)
207 FORMAT(/"DELAY TIME (DT) MUST NOT EXCEED (TST)"/)
    GO TO 1
209 WRITE(2,210)
210 FORMAT(/"TRIGGER VOLTAGE LEVEL (TVL) RANGE EXCEEDED"/)
    GO TO 1
 32 WRITE(2,33)
 33 FORMAT("A/D SAMPLE COMPLETE")
 35 WRITE(2,36)
 36 FORMAT("DO YOU WANT A LISTING?")
    READ(1,37)IR
 37 FORMAT(A1)
    IF(IR-89)42,38,42
 38 WRITE(6,37)12
    WRITE(6,40)(BUF(I),I=1,LBUF)
 40 FORMAT(10F8.3)
 42 WRITE(2,44)
 44 FORMAT("DO YOU WANT A PRINTER PLOT?")
    READ(1,37)IR
    IF(IR-89)46,45,46
 45 CALL PLOT(BUF,LBUF)
 46 WRITE(2,47)
 47 FORMAT("DO YOU WANT TO TAKE ANOTHER SAMPLE?")
    READ (1,37)IR
    IF(IR-89)35,48,35
 48 WRITE(2,49)
 49 FORMAT("SAME SAMPLING PARAMETERS?")
    READ(1,37)IR
    IF(IR-89)1,75,1
    END
    END$
```

```
FTN,B,L
      SUBROUTINE STOW(IBUF,LBUF,DT,TST,TVL,ITCN,IDCN,IE)
C     REAL IBUF
      DIMENSION IBUF(2)
      CALL PRIME(IBUF,LBUF,DT,TST,TVL,ITCN,IDCN,IE)
      IF (IE)3,1,3
    1 J=LBUF
      K=J+J-1
      DO 2 I=1,LBUF
      X=IBUF(J)
      X=0.0012207*X
      CALL FPUT(IBUF(K),X)
      J=J-1
    2 K=K-2
    3 RETURN
      END
      END$
```

```
FTN,B,L
      SUBROUTINE PRIME(IBUF,LBUF,DT,TST,TVL,ITCN,IDCN,IE)
      DIMENSION IBUF(2)
      X=LBUF
      X=TST/X
      IWAIT=312.512*X-11.0
      IF(IWAIT)75,14,14
   75 IE=2
      RETURN
   14 IF(DT)16,20,21
   16 DL=-DT
      NB=DL/X+0.5
      IF(NB)80,80,17
   17 IF(LBUF-NB)80,80,18
   80 IE=3
      RETURN
   18 R=0
      GO TO 25
   20 NA=DT
      GO TO 22
   21 NA=DT-1.0
   22 R=1
   25 IE=0.1001*TVL
      IF(IE)85,26,85
   85 IE=4
      RETURN
   26 ITVL=819.2*TVL
      WRITE(2,30)
   30 FORMAT("START ANALOG DATA & PRESS RETURN")
      READ(1,31)W
   31 FORMAT(A1)
      IF(R)50,35,50
   35 CALL CATCH(IBUF,NB,LBUF,ITVL,ITCN,IWAIT)
   37 IF(IWAIT)10,38,40
   10 IE=1
      RETURN
   38 IF(R)40,40,55
   40 IWAIT=IWAIT + 1
      IS=1
      II=1
      J=IBUF(1)
   42 DO 49 I=1,NB
      II=II-IWAIT
      IF(II)46,46,47
   46 II=II+NB
   47 JJ=IBUF(II)
      IBUF(II)=J
      IF(II-IS)49,48,49
   48 IS=IS+1
      II=IS
      JJ=IBUF(II)
   49 J=JJ
      GO TO 55
   50 CALL SNARE(IBUF,NA,LBUF,ITCN,IDCN,ITVL,IWAIT)
      GO TO 37
   55 RETURN
```

```
END
ENDS
```

```
FTN,B,L
      SUBROUTINE FPUT(X,Y)
      X=Y
      RETURN
      END
      END$
```

```
FTN,B,L
      SUBROUTINE PLOT(BUF,LBUF)
      DIMENSION BUF(2)
    1 WRITE(2,2)
    2 FORMAT("STARTING DATA INDEX?")
      READ(1,3)IS
    3 FORMAT(I4)
      WRITE(2,4)LBUF
    4 FORMAT("LAST DATA INDEX, (MAX=",I4,")?")
      READ(1,3)IF
      BIG=0.0
      SML=0.0
      DO 12 I=IS,IF
      X=BUF(I)
      IF(BIG-X)9,10,10
    9 BIG=X
   10 IF(X+SML)11,12,12
   11 SML=-X
   12 CONTINUE
      IF(BIG-SML)13,14,14
   13 BIG=SML
   14 BIG=37.0/BIG
      WRITE(6,100)12
  100 FORMAT(A1)
      DO 35 I=IS,IF
      X=BUF(I)*BIG
      IR=0
      IF(X)120,15,130
  120 IR=X-0.5
      GO TO 140
  130 IR=X+0.5
  140 IF(IR)20,15,25
   15 WRITE(6,16)I
   16 FORMAT(I4,38X,"*")
      GO TO 35
   20 K=-1-IR
      IF(K)21,21,23
   21 WRITE(6,22)I
   22 FORMAT(I4,37X,"*I")
      GO TO 35
   23 J=38+IR
      WRITE(6,24)I,(32,L=1,J),42,(32,L=1,K),73
   24 FORMAT(I4,76A1)
      GO TO 35
   25 J=IR-1
      IF(J)26,26,28
   26 WRITE(6,27)I
   27 FORMAT(I4,38X,"I*")
      GO TO 35
   28 WRITE(6,24)I,(32,L=1,38),73,(32,L=1,J),42
   35 CONTINUE
      WRITE(2,41)
   41 FORMAT("PLOT IT AGAIN?")
      READ(1,45)IR
   45 FORMAT(A1)
      IF(IR-89)50,1,50
```

```
50 RETURN
   END
   END$
```

PAGE 0001

```
0001                    ASMB,R,B,L,T
AST    R 000145
BGN    R 000140
CMW    R 000144
CON    R 000120
END    R 000142
ERR    R 000124
EXT    R 000126
I*1    R 000146
L1     R 000056
L2     R 000101
LVL    R 000143
MID    R 000141
REP    R 000072
CATCH  R 000000
SAVA   R 000136
SAVB   R 000137
**  NO ERRORS*
```

PAGE 0002 #01

```
                          ASMB,R,B,L,T
        00000              NAM CATCH
                           ENT CATCH
*****************************************************************
*                                                               *
*             PHOENIX MISSION SIMULATION TEST                   *
*               A/D DRIVER SUBROUTINE CATCH                     *
*             NEGATIVE DELAY TIME(LOOK AHEAD)                   *
*                                                               *
*****************************************************************
*
*          A/D DRIVER THAT IS TUNED TO CONTROL
*          A/D SAMPLING RATE.  MAXIMUM SAMPLING
*          RATE IS APPROXIMATELY 28KHZ.  TRIG-
*          GER CHANNELS (0-7) SPECIFY ON WHICH
*          CHANNEL THE SAMPLING IS PERFORMED.
*          OPERATOR SUPPLIES THE TRIGGER
*          CHANNEL VIA THE KEYBOARD.  THE
*          CALL FROM FORTRAN IS AS FOLLOWS:
*
*  CALL CATCH(IBUF,NB,LBUF,ITVL,ITCN,IWAIT)
*          WHERE THE PARAMETERS ARE DEFINED AS
*          FOLLOWS:
*
*  IBUF  = THE NAME OF AN INTEGER BUFFER TO
*            STORE THE A/D DATA IN.
*  NB    = THE NUMBER OF SAMPLES TO STORE BEFORE
*            THE TRIGGER OCCURS.(LOOK AHEAD)
*  LBUF  = LENGTH OF THE BUFFER IBUF.
*  ITVL  = A TRIGGER LEVEL THAT CAUSES
*            THE DRIVER TO TAKE DATA BEYOND
*            THE FIRST "NB" DATA POINTS.
*            THIS TRIGGER IS TRIPPED IF THE
*            DATA EXCEEDS THE MAGNITUDE OF
*            "ITVL" IN THE SAME SENSE AS THE
*            SIGN OF "ITVL".  (I.E.  A SIG-
*            NAL OF -5.0V WILL TRIGGER A
*            "ITVL" OF -4.9V AND A SIGNAL
*            OF 5.5V WILL TRIGGER A "ITVL" OF
*            5.4V ETC.)
*  ITCN  = THE TRIGGER CHANNEL NUMBER (0-7)
*  IWAIT = A WAIT COUNT TO CONTROL THE SAMPLING
*            RATE.  THE LARGER "IWAIT" IS THE
*            SLOWER THE SAMPLING RATE.
*  IWAIT = THIS PARAMETER IS ALSO A RETURNED
*            PARAMETER, THE VALUE BEING THE OFFSET
*            OF THE FIRST "NB" POINTS AS TAKEN
*            CYCLICALLY IN THE FIRST "NB" ELEMENTS
*            OF "IBUF".  THE REST OF THE DATA
*            POINTS ARE STORED IN "IBUF" STARTING
*            WITH THE "NB +1" ELEMENT OF "IBUF".
*            "IWAIT" WILL BE RETURNED WITH A -1
*            VALUE IF THERE  IS AN A/D ERROR.
*            ALL NORMAL OFFSET VALUES WILL BE
*            POSITIVE OR ZERO.
```

PAGE 0003 #01

```
00000 000000  CATCH NOP          RETURN ADDRESS STORED IN CATCH
00001 072136R  STA SAVA          SAVE VALUE IN A REGISTER
00002 076137R  STB SAVB          SAVE VALUE IN B REGISTER
00003 066000R  LDB CATCH         GET ADDRESS
00004 160001   LDA 1,I           OF RETURN ADDRESS
00005 072000R  STA CATCH         RETURN ADDRESS STORED HERE
00006 006004   INB               BUMP B BY ONE
00007 160001   LDA 1,I           GET ADDRESS OF IBUF
00010 002021   SSA,RSS           SKIP NEXT STATE. IF A ISN'T POS.
00011 026014R  JMP *+3           JUMP 3 STATEMENTS
00012 012147R  AND =B77777       MAKE IBUF POS. IF NOT ALREADY
00013 160000   LDA 0,I           LOAD A WITH IBUF
00014 072140R  STA BGN           STORE IBUF IN BEGIN
00015 006004   INB               BUMP B
00016 160001   LDA 1,I           GET ADDRESS OF (NB) 2ND PARAMETER
00017 160000   LDA 0,I           LOAD A WITH NB
00020 042140R  ADA BGN           ADD A TO BEGIN
00021 072141R  STA MID           STORE A IN MIDDLE
00022 006004   INB               BUMP B
00023 160001   LDA 1,I           GET ADDRESS OF LBUF (3)
00024 160000   LDA 0,I           LOAD A WITH LBUF
00025 042140R  ADA BGN           ADD TOP OF IBUF TO LENGTH = END
00026 072142R  STA END           STORE IN END
00027 006004   INB
00030 062150R  LDA =B2020        LOAD A WITH BINARY SSA
00031 072072R  STA REP           STORE SSA INSTRUCTION IN REPEAT
00032 160001   LDA 1,I           GET ADDRESS OF ITVL (4)
00033 160000   LDA 0,I           LOAD A W/ITVL
00034 003025   CMA,SSA,INA,RSS   COMP TRIG LVL SKPS IF TRIG IS POS
00035 036072R  ISZ REP           CHANGES SIGN OF SKIP
00036 072143R  STA LVL           STORES COMP. TRIGGER LEVEL
00037 006004   INB
00040 160001   LDA 1,I           GET ADDRESS OF ITCN (5)
00041 160000   LDA 0,I           LOAD A W/ITCN
00042 072144R  STA CMW           STORE A IN COMMON WORD
00043 006004   INB
00044 160001   LDA 1,I           GET ADDRESS OF IWAIT
00045 072145R  STA AST           STORE IWAIT ADDRESS
00046 160000   LDA 0,I           LOAD A W/IWAIT
00047 003000   CMA               COMPLEMENT IWAIT
00050 072146R  STA IW1           STORE COMPLEMENT
00051 103100   CLF 0             DISABLES INTERRUPT SYSTEM
00052 106722   CLC 22B           CLEAR CONTROL OF CH 22
00053 062144R  LDA CMW           LOAD CMW IN A
00054 102622   OTA 22B           OUTPUT TRIGGER CH TO A/D
00055 066140R  LDB BGN           LOAD B W/TOP OF IBUF
00056 103722  L1 STC 22B,C       SET CONTROL - STARTS A/D
00057 062151R  LDA =B-1          FINE
00060 002006   INA,SZA           TUNING
00061 026060R  JMP  *-1          LOOP
00062 062146R  LDA IW1           WAIT
00063 002006   INA,SZA           SAMPLE
00064 026063R  JMP *-1           TIME
00065 102322   SFS 22B           CHECK FOR FLAG
00066 026124R  JMP ERR           JUMP TO ERROR IF NO FLAG SET
```

PAGE 0004 #01

```
00067 102522   LIA 22B        LOAD A W/ FIRST A/D SAMPLE
00070 170001   STA 1,I        STORE DATA IN IBUF
00071 042143R  ADA LVL        CHECK FOR TRIGGER
00072 000000  REP NOP         DUMMY-LOADED W/SSA OR SSA,RSS
00073 026120R  JMP CON        JUMP TO CONTINUE
00074 076143R  STB LVL        TRIG.EXCEED.STORE PTR IN LEVEL
00075 066141R  LDB MID        SET PTR TO REMAINDER OF IBUF
00076 062151R  LDA =B-1       FINE TUNING TO KEEP L1 AND L2
00077 002006   INA,SZA        THE SAME LENGTH - THIS TIGHT
00100 026077R  JMP *-1        LOOP ADD 52, COMP.CYCLES
00101 103722  L2 STC 22B,C    SET CONTROL
00102 062152R  LDA =B-4       FINE TUNING LOOP
00103 002006   INA,SZA        ADDS (9) COMPUTER
00104 026103R  JMP *-1        CYCLES
00105 062146R  LDA IN1        WAIT
00106 002006   INA,SZA        SAMPLE
00107 026106R  JMP *-1        TIME
00110 102322   SFS 22B        SKIP IF FLAG NOT SET
00111 026124R  JMP ERR        JUMP ERROR
00112 102522   LIA 22B        READ IN FIRST DATA AFTER TRIG.
00113 170001   STA 1,I        STORE A IN IBUF (MID TO END)
00114 006004   INB
00115 056142R  CPB END        IF B EXCEEDS END
00116 026126R  JMP EXT        JUMP TO EXIT
00117 026101R  JMP L2         IF NOT CONTINUE SAMPLING
00120 006004  CON INB         2ND HALF OF LOOK AHEAD LOOP
00121 056141R  CPB MID        COMPARE PTR W/MID
00122 066140R  LDB BGN        RESET POINTER
00123 026056R  JMP L1         CONTINUE DATA SAMPLING
00124 002400  ERR CCA         ERROR - CLEARS A AND MAKES (-1)
00125 026131R  JMP *+4        JUMP 4 STATEMENT
00126 062140R EXT LDA BGN     EXIT LOAD A W/TOP OF IBUF
00127 003004   CMA,INA        MAKE TOP IBUF NEGATIVE
00130 042143R  ADA LVL        GIVES OFFSET OF B4 DATA
00131 102100   STF 0          ENABLE INTERRUPT SYSTEM
00132 172145R  STA AST,I      RETURN IWAIT
00133 062136R  LDA SAVA       RESTORE A
00134 066137R  LDB SAVB       RESTORE B
00135 126000R  JMP CATCH,I    RETURN TO PREVIOUS SUBROUTINE
00136 000000  SAVA NOP
00137 000000  SAVB NOP
00140 000000  BGN NOP
00141 000000  MID NOP
00142 000000  END NOP
00143 000000  LVL NOP
00144 000000  CMW NOP
00145 000000  AST NOP
00146 000000  IW1 NOP
00147 077777
00150 002020
00151 177777
00152 177774
               END
**  NO ERRORS*
```

PAGE 0001

```
0001                    ASMB,R,B,L,T          ASSEMBLY HEADER
BGN    R 000151
CON    R 000064
DLA    R 000152
END    R 000153
ERR    R 000137
EXT    R 000141
IM     R 000127
IW1    R 000160
L2     R 000120
LVL    R 000156
REP    R 000074
ASAV   R 000157
CMW1   R 000155
CMW2   R 000154
SAVA   R 000147
SAVB   R 000150
SNARE  R 000000
**  NO ERRORS*
```

PAGE 0002 #01

```
                 ASMB,R,B,L,T          ASSEMBLY HEADER
      00000       NAM SNARE            NAME OF PROGRAM-SNARE
                  ENT SNARE            ENTRY POINT
*********************************************************
*                                                       *
*         PHOENIX MISSION SIMULATION TEST               *
*           A/D DRIVER SUBROUTINE SNARE                 *
*               POSITIVE DELAY TIME                     *
*                                                       *
*********************************************************
*
*            A/D DRIVER THAT IS TUNED TO CONTROL A/D
*            SAMPLING RATE.  MAXIMUM SAMPLING RATE
*            IS APPROXIMATELY 28KHZ.  TRIGGER AND
*            DATA CHANNELS (0-7) SPECIFY ON WHICH
*            CHANNEL THE SAMPLING IS PERFORMED.
*            A SEQUENTIAL MODE WHERE THE A/D SAM-
*            PLES EACH CHANNEL SEQUENTIALLY START-
*            ING WITH CHANNEL (0) IS ALSO PROVIDED.
*            THE OPERATOR SPECIFIES THESE PARAMETERS
*            VIA THE KEYBOARD.  THE CALL FROM FOR-
*            TRAN IS AS FOLLOWS:
*
*   CALL SNARE(IBUF,NA,LBUF,ITCN,IDCN,ITVL,IWAIT)
*            WHERE THE PARAMETERS ARE DEFINED AS
*            FOLLOWS:
*
*   IBUF   = THE NAME OF AN INTEGER BUFFER TO STORE
*            THE A/D DATA IN.
*   NA     = THE DELAY TIME AFTER THE TRIGGER LEVEL
*            HAS OCCURRED.  THE A/D WILL NOT PERFORM
*            ANY DATA CONVERSIONS UNTIL AFTER THIS
*            DELAY TIME HAS PASSED.
*   LBUF   = THE LENGTH OF THE INTEGER BUFFER "IBUF".
*   ITCN   = THE TRIGGER CHANNEL NUMBER (0-7).
*   IDCN   = THE DATA CHANNEL NUMBER (0-7), OR THE
*            SEQUENTIAL OPERATION MODE OF THE A/D.
*   ITVL   = A TRIGGER LEVEL THAT CAUSES THE
*            DRIVER TO TAKE DATA AFTER THE TRIGGER
*            LEVEL HAS BEEN EXCEEDED, AND THE DELAY
*            TIME HAS PASSED.  THIS TRIGGER IS TRIP-
*            PED IF THE DATA EXCEEDS THE MAGNITUDE
*            OF "ITVL" IN THE SAME SENSE AS THE
*            SIGN OF "ITVL".  (I.E.  A SIGNAL OF
*            -5.0V WILL TRIGGER A "ITVL" OF -4.9V
*            AND A SIGNAL OF 5.5V WILL TRIGGER A
*             "ITVL" OF 5.4V ETC.)  MAXIMUM TRIGGER
*            LEVELS ARE -10V TO +10V.
*   IWAIT  = A WAIT COUNT TO CONTROL THE SAMPLING RATE.
*            THE LARGER "IWAIT" IS THE SLOWER THE
*            SAMPLING RATE.
*   IWAIT  = THIS PARAMETER IS ALSO A RETURNED PARA-
*            METER, THE VALUE BEING A ZERO OFFSET OF
*            DATA FROM THE START OF "IBUF".  IWAIT
*            WILL ALSO BE RETURNED WITH THE VALUE
```

PAGE 0003 #01

```
*             -1 WHEN THE A/D IS NOT OPERATING
*             NORMALLY.
     00000 000000  SNARE NOP             DUMMY STORAGE FOR RETURN ADDRES
     00001 072147R  STA SAVA             SAVE A REGISTER CONTENT
     00002 076150R  STB SAVB             SAVE B REGISTER CONTENT
     00003 066000R  LDB SNARE            LOAD B W/ADDRESS OF RTN ADDRESS
     00004 160001   LDA 1,I              LOAD A W/RETURN ADRESS
     00005 072000R  STA SNARE            STORE RTN ADDRESS IN SNARE
     00006 006004   INB                  INCREMENT B REGISTER BY 1
     00007 160001   LDA 1,I              LOAD A W/ADDRESS OF IBUF
     00010 002021   SSA,RSS              SKIP IF A IS NOT POSITIVE
     00011 026014R  JMP *+3              JUMP THREE INSTRUCTIONS
     00012 012161R  AND =B77777          MAKE IBUF POSITIVE
     00013 160000   LDA 0,I              LOAD A WITH IBUF
     00014 072151R  STA BGN              STORE IBUF IN BEGIN
     00015 006004   INB
     00016 160001   LDA 1,I              GET ADDRESS OF DELAY TIME
     00017 160000   LDA 0,I              LOAD A WITH DELAY TIME
     00020 002002   SZA                  SKIP IF A IS ZERO
     00021 003000   CMA                  COMPLEMENT DELAY TIME
     00022 072152R  STA DLA              STORE DELAY TIME IN DELAY
     00023 006004   INB
     00024 160001   LDA 1,I              GET ADDRESS OF LBUF
     00025 160000   LDA 0,I              LOAD A WITH LBUF
     00026 042151R  ADA BGN              ADD LBUF TO IBUF
     00027 072153R  STA END              STORE LBUF IN END
     00030 006004   INB
     00031 160001   LDA 1,I              GET ADDRESS OF ITCN
     00032 160000   LDA 0,I              LOAD A WITH TRIGGER CHNL NO.
     00033 072155R  STA CMW1             STORE ITCN IN COMM. WORD 1
     00034 006004   INB
     00035 160001   LDA 1,I              GET ADDRESS OF IDCN
     00036 160000   LDA 0,I              LOAD A WITH IDCN
     00037 002020   SSA                  SETS UP FOR SEQUENTIAL MODE
     00040 062162R  LDA =B40000          SEQ. BIT 14
     00041 072154R  STA CMW2             STORE 2ND COMMAND WORD
     00042 006004   INB
     00043 062163R  LDA =B2020           BINARY OF SSA INSTRUCTION
     00044 072074R  STA REP              REPEAT HAS SSA INSTRUCTION
     00045 160001   LDA 1,I              GET ADDRESS OF ITVL
     00046 160000   LDA 0,I              LOAD A WITH ITVL
     00047 003025   CMA,SSA,INA,RSS      SKIPS IF TRIG. IS POSITIVE
     00050 036074R  ISZ REP              SETS UP SSA,RSS INSTRUCTION
     00051 072156R  STA LVL              STORE COMP.TRIGGER LEVEL
     00052 006004   INB
     00053 160001   LDA 1,I              GET ADDRESS OF IWAIT
     00054 072157R  STA ASAV             SAVE ADDRESS OF IWAIT
     00055 160000   LDA 0,I              LOAD A WITH IWAIT
     00056 003000   CMA                  COMPLEMENT IWAIT
     00057 072160R  STA IW1              STORE IWAIT
     00060 103100   CLF 0                DISABLES INTERRUPT SYSTEM
     00061 062155R  LDA CMW1             LOAD A WITH TRIG. CHNL NO.
     00062 106722   CLC 22B              CLEAR CONTROL CH 22
     00063 102622   OTA 22B              OUTPUT TRIG. CHNL TO A/D
     00064 103722  CON STC 22B,C         SET CONTROL(ENCODE)
```

PAGE 0004 #01

```
00065 062164R  LDA =B-3          WAIT FOR FLAG
00066 002006   INA,SZA
00067 026066R  JMP *-1
00070 102322   SFS 22B           SKIP IF FLAG SET
00071 026137R  JMP ERR           JUMP TO ERROR IF NO FLAG
00072 102522   LIA 22B           LOAD INTO A FIRST DATA SAMPLE
00073 042156R  ADA LVL           CHECK TO SEE IF TRIG. EXCEEDED
00074 000000  REP NOP            DUMMY LOADED W/SSA OR SSA,RSS
00075 026064R  JMP CON           NO TRIG. JUMP TO CONTINUE
00076 062152R  LDA DLA           LOAD A WITH DELAY TIME
00077 002003   SZA,RSS           SKIP IF DELAY IS NOT ZERO
00100 026106R  JMP *+6
00101 066165R  LDB =B-454        ONE MS INNER LOOP TIME
00102 006005   INB,SZB
00103 026102R  JMP *-1
00104 002006   INA,SZA
00105 026101R  JMP *-4           END OF DELAY LOOP
00106 000000   NOP               FINE TUNING FOR DELAY LOOP
00107 066151R  LDB BGN           SET POINTER TO TOP OF IBUF
00110 062154R  LDA CHN2          LOAD A WITH DATA CHNL NO.
00111 106722   CLC 22B           CLEAR CONTROL
00112 102622   OTA 22B           OUTPUT DATA CHNL TO A/D
00113 103722   STC 22B,C         BEGIN FIRST DATA LOOP
00114 062166R  LDA =B-2          WAIT FOR FLAG
00115 002006   INA,SZA
00116 026115R  JMP *-1
00117 026127R  JMP IM            JUMP TO 2ND HALF OF LOOP
00120 103722  L2 STC 22B,C       BEGIN NORMAL SAMPLING
00121 056153R  CPB END           COMPARE PTR TO END OF IBUF
00122 026141R  JMP EXT           JUMP IF BUFFER FILLED
00123 062160R  LDA IW1           SAMPLE
00124 002006   INA,SZA           WAIT
00125 026124R  JMP *-1           TIME
00126 006004   INB
00127 102322  IM SFS 22B         SKIP IF FLAG SET
00130 026137R  JMP ERR           IF NO FLAG JUMP TO ERROR
00131 102522   LIA 22B           LOAD A WITH MORE DATA
00132 170001   STA 1,I           STORE DATA WHERE PTR POINTS
00133 062167R  LDA =B-4          FINE
00134 002006   INA,SZA           TUNING
00135 026134R  JMP *-1           LOOP
00136 026120R  JMP L2            JUMP TO NORMAL SAMPLING
00137 003400  ERR CCA            ERROR EXIT (A=-1)
00140 026142R  JMP *+2
00141 002400  EXT CLA            EXIT FROM SNARE (A=0)
00142 102100   STF 0             ENABLES INTERRUPT SYSTEM
00143 172157R  STA ASAV,I        PASSES ZERO OFFSET OF (-1)
00144 062147R  LDA SAVA          RESTORE A REGISTER
00145 066150R  LDB SAVB          RESTORE B REGISTER
00146 126000R  JMP SNARE,I       RETURN TO PREVIOUS SUBR.
00147 000000  SAVA NOP
00150 000000  SAVB NOP
00151 000000  BGN NOP
00152 000000  DLA NOP
00153 000000  END NOP
```

```
 PAGE 0005 #01

0166   00154 000000  CMW2 NOP
0167   00155 000000  CMW1 NOP
0168   00156 000000  LVL NOP
0169   00157 000000  ASAV NOP
0170   00160 000000  IW1 NOP
       00161 077777
       00162 040000
       00163 002020
       00164 177775
       00165 177324
       00166 177776
       00167 177774
0171                   END
**  NO ERRORS*
```

APPENDIX B

Calculations for determining, the sample time, and delay times of the A/D driver; and calcualtions of the maximum missile response signal frequency are included in the following appendix.

CALCULATIONS FOR SAMPLE TIME

The below procedure was used to calculate minimum and maximum sampling times for the A/D driver program. IWAIT is the wait count between data samples. Define:

CT = Computer cycle time in increments of 1.6 usec.

IWAIT = Number of computer cycles to count between data samples.

OH = Overhead, the number of fixed computer cycles in a loop.

SR = Data sampling rate (samples/millisecond).

TST = Total sample time (milliseconds).

X = Time between samples (milliseconds/sample) the reciprocal of SR.

An HP 5327B counter/timer was used to measure the time interval between successive ENCODE or Set Control pulses going from the HP 2116C computer to the HP 5610A A/D converter. A small assembler program, hand toggled into the computer, was used to measure the time interval data. The following data was collected:

IWAIT (OCTAL)	COMP IW1 (OCTAL)	FREQ (SR) Hz	TIME INTERVAL (X)usec
0	-1	31250	31.5 - 31.6
1	-2	30991	34.7 - 34.8
4	-5	22321	44.3 - 44.4
63	-64	4280.7	233.1 - 233.2
2047	-2048	151.93	6582.0 - 6582.1
32766	-32767	9.53	104879.0

The A/D driver assembler language program has the following format for implementing IWAIT:

OPERATION	COMPUTER CYCLES
L1 STC 22B, C	1
LDA IW1	2
INA, SZA	IWAIT +1
JMP * - 1	IWAIT
SFS 22B	1
JMP ERR	0
JMP L1	1

The SAMPLE TIME is computed according to the following formula.

SAMPLE TIME = (2 IWAIT + OVERHEAD) (CYCLE TIME), or

$$\text{IWAIT} = \frac{1}{2}\left(\frac{\text{SAMPLE TIME}}{\text{CYCLE TIME}} - \text{OVERHEAD}\right) \quad (1)$$

Since the data gives us the time interval or SAMPLE TIME, and the IWAIT parameter was given, the computer CYCLE TIME was calculated. The OVERHEAD term will cancel out since it was constant for all measurements.

$$\text{CYCLE TIME} = \frac{\text{SAMPLE TIME}}{2\ \text{IWAIT} + \text{OVERHEAD}} \quad (2)$$

$$\text{CYCLE TIME} = \frac{104879-31.5}{2\ (32766)-0} = 1.59993\ \text{usec} \quad (3)$$

and, from (1)

$$\text{IWAIT} = \frac{\text{SAMPLE TIME}}{2\ (1.59993)} - \frac{\text{OVERHEAD}}{2}$$

Thus IWAIT = 312.512 X -B

X = SAMPLE TIME

B = OH/2

The OVERHEAD in the A/D driver program is 22 computer cycles, thus

B = OH/2 = 22/2 = 11

or finally,

$$IWAIT = 312.512\ X - 11.0 \qquad (4)$$

The above expression for IWAIT is contained in the PRIME subroutine of the A/D driver program.

Below are some sample times calculated from (4) above:

IWAIT	SAMPLE TIME (X) ms	SAMPLING RATE (SR) KHz
0	.035	28.41 (MAX)
1	.038	26.04
5	.051	19.53
10	.067	14.88
100	0.35	2.82
2000	6.45	.155
10000	32.3	.031
32766	104.8	.00954 (MIN)

CALCULATIONS FOR DELAY TIME

The below procedure was used to calculate the variable parameter (X) for the delay time loop. Define:

CT = Computer cycle time in increments of 1.6 usec.

DLA = Complemented delay time in assembler program.

DT = Delay time inputted from keyboard.

N = Number of times each computer operation is used in the delay time loop.

OH = Overhead, the number of fixed computer cycles in a loop.

X = Inner loop parameter for counting to one millisecond.

The A/D driver has the following format for implementing the delay time:

OPERATION	COMPUTER CYCLES
.	
.	
.	
LDA DLA	0 (outer loop)
LDB = B-X	2 (N+1) (inner loop)
INB, SZB	X (N+1)
JMP, * -1	(X-1) (N+1)
INA, SZA	(N+1)
JMP * -4	(N+1) -1
.	
.	
.	

For N = 0, the inner loop is used one time, or a minimum. Set this minimum loop time equal to one millisecond. If the minimum delay time is set to one millisecond increments, then the parameter X will be the necessary inner loop cycles to equal one millisecond.

The parameter X can be calculated using the following formula:

$$DT = [2(N+1)+X(N+1)+(X-1)(N+1)+(N+1)+(N+1)-1+OVERHEAD](CT)$$

LET = $DT = DT_{min}$ = 1ms = 1000 usec

N = 0 (min times through loop)

OH = Whatever the overhead is. In the A/D Driver OH = 23;

so,

$$DT_{min} = [(N+1)(2X+3)+OH-1](CT) \quad (2)$$

or,

$$1000 = (2X+3+23-1)(CT)$$

$$1000 = (2X+25)(CT)$$

$$X = \frac{1}{2}\frac{1000}{CT} - 25 = \frac{1}{2}\frac{1000}{1.6} - 25$$

$$= \frac{(625-25)}{2} = 300_{10}$$

or,

$$X = 454_8 \quad (3)$$

Thus, if the inner loop is cycled 454_8 times, the delay time will always be in multiples of one millisecond. If the X parameter had not been an integer, the overhead would have had to been increased or decreased by one. The A/D driver uses integer numbers. It was precisely the overhead factor that allowed all of the seven timing loops in the A/D driver to be matched to within ± one computer cycle time.

CALCULATION OF VCLO CONTROL VOLTAGE

VELOCITY SEARCH FREQUENCY

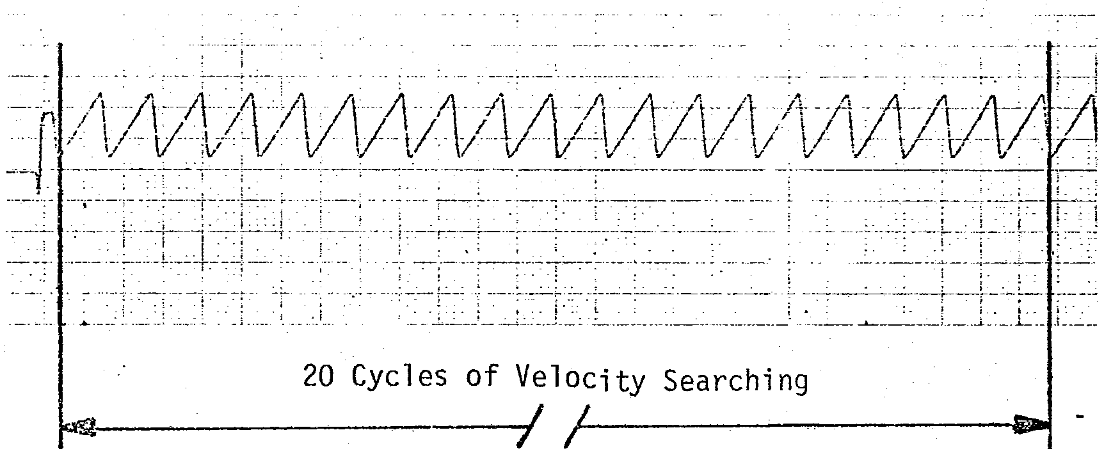
A brush recording was made of the VCLO control voltage waveform while the VCLO was velocity searching. The brush recording was run at maximum speed of 125 mm/sec.

The frequency was calculated by the following method. First, a reference point was chosen on the VCLO waveform. From this reference point, 30 cycles of the waveform were counted out. The thirty cycles took 190 mm to complete, thus

$$\text{VCLO Control Voltage} = \left(\frac{30 \text{ Cycles}}{190 \text{ mm}}\right)\left(\frac{125 \text{ mm}}{\text{sec}}\right)$$

$$= 19.74 \approx 20 \text{ Hz}$$

Refer to Figure 8, for a pictorial view of the actual VCLO control voltage waveform.



Brush Recorder Speed: 125 mm/sec
Length of 30 cycles of waveform: 190 mm
(only 20 cycles shown)

$$\text{VCLO Control Voltage Frequency} = \left(\frac{30\ \text{cycles}}{190\ \text{mm}}\right)\left(\frac{125\ \text{mm}}{\text{sec}}\right)$$

$$= 19.74 \approx 20\ \text{Hz}$$

Figure 8. - VCLO Control Voltage Waveform

APPENDIX C

List of equipment used in developing the MST.

List of equipment incorporated into the MECCA system as part of the MST.

The following is a list of equipment used in developing the MST:

HP 2116C Computer

HP 5327B Counter/Timer

HP 3480B DVM

HP 5610A A/D Converter

HP 200CD Widerange Oscillator

Tektronix RM561A Oscilloscope

Tektronix 7613 Oscilloscope

Tektronix TM 503 Signal Generator

Dynamic 7525 Differential DC Amplifiers

Dynamic 6364A/RF Electronic Filters

Bell/Howell VR-3400 14 Channel Tape Unit

Brush Mark 260 Analog Pen Recorder

The following is a list of equipment incorporated into the MECCA system as part of the MST:

HP 2116C Computer

HP 5610A A/D Converter

Bell/Howell VR-3400 14 Channel Tape unit

Dynamic 7525 Differential DC Amplifiers

Dynamic 6364A/RF Electrical Filters

APPENDIX D

Operational instructions for using the MST.

OPERATIONAL INSTRUCTIONS

To use the MST computer programs follow the procedure listed below:

(1) Load the MST disk cartridge into the disk drive unit located under the tape unit in the MECCA computer equipment bay.

(2) Reboot computer following rebooting procedures listed in the appropriate NAVAIR Manual.

(3) Run a complete long self test on the AN/DSM-130(V) GMTS.

(4) If the self test results are good, turn on the HP 5610A A/D converter located under the disk drive unit.

(5) Patch the VR-3400 tape unit to the MST PHOENIX patch panel. Set the tape unit to a $\pm$ 10V range scale. Turn power on.

(6) Call up PRN 7800 from the computer console. Start the VR-3400 tape unit (15 IPS). Input the necessary information asked.

(7) The MTP will now run to completion completely automatic.

(8) After the MTP was been terminated, stop the VR-3400 tape unit, rewind the tape.

(9) Patch the VR-3400 to the A/D patch panel.

(10) Call PRN 7700 from the computer console. Input the necessary information. Start the VR-3400 tape unit when display console instructs to do so.

(11) The missile response data that was recorded on the VR-3400 will now be digitized and stored in the HP 2116C computer.

(12) Once the digitization is complete, a listing or a plot can be obtained.

(13) The recorded missile response data can be sampled as many times as is necessary.

(14) Once all data processing is completed, rewind the VR-3400 tape unit and remove power.

(15) If additional special missile tests are required call PRN 3 from the computer console, SCRATCH, and feed in the special missile programs via the paper tape reader. Follow instructions in each special missile program.

(16) After all testing is complete call PRN 7710, to remove power from the missile.