

1/3 pages

NASA Grant NCC 2-41

Technology Research For Strapdown
Inertial Experiment and Digital
Flight Control and Guidance

FINAL TECHNICAL REPORT

(NASA-CR-176907) TECHNOLOGY RESEARCH FOR
STRAPDOWN INERTIAL EXPERIMENT AND DIGITAL
FLIGHT CONTROL AND GUIDANCE Final Technical
Report, 1 Oct. 1979 - 31 Dec. 1984
(University of Southern Colorado, Pueblo.)

N86-29800

Unclas

G3/04 43271

School of Applied Science
and Engineering Technology



UNIVERSITY OF SOUTHERN COLORADO

2200 North Bonforte Boulevard

Pueblo, Colorado 81001

NASA Grant NCC 2-41

Technology Research For Strapdown
Inertial Experiment and Digital
Flight Control and Guidance

FINAL TECHNICAL REPORT

NASA Grant NCC 2-41

Technology Research For Strapdown
Inertial Experiment and Digital
Flight Control and Guidance

FINAL TECHNICAL REPORT

Principal Investigator:

Ralph A. Carestia

Co-Investigator:

Don E. Cottrell

Research Assistants:

Arthur A. Beckman

Tim R. Johnson

Peter J. Holub

Richard W. Parker

Period Covered By Report:

October 1, 1979 to December 31, 1984

Institution:

University of Southern Colorado

2200 Bonforte Blvd.

Pueblo, Colorado 81001

NASA Technical Officer:

George Callas

TABLE OF CONTENTS

	<u>Page No.</u>
General Summary.....	1
Cumulative References.....	6
Appendices.....	7

NASA Grant NCC2-41
Technology Research for Strapdown Inertial Experiment and
Digital Flight and Guidance

FINAL REPORT

General Summary

The University of Southern Colorado research engineers were involved in: (1) Investigating strapdown inertial sensor systems and (2) Studying the system requirements of digital avionics systems in rotorcraft. This included feasibility examination of digital data acquisition techniques in aircraft and the use of digital systems in rotorcraft for the purpose of specifying design criteria for integrated guidance, navigation and control. Results from the research produced three literature publications and several working reports for NASA. A chronologically ordered cumulation of these publications is provided following this summary.

The main objectives of the University of Southern Colorado's cooperative agreement at NASA-Ames were to test a Strapdown Inertial Navigation System and to provide system criteria required to design reliable cockpit avionics systems for rotorcraft. The applied research conducted was divided into three areas: (1) To develop a digital data acquisition system and flight test a Strapdown Ring Laser Gyro Tetrad Inertial Navigation System in a VH-1B helicopter. (2) To develop, design and build an operational interface for the Rotorcraft Digital Advanced Avionics System (RODAAS) and (3) To integrate and test the navigational capabilities of a Differential Global Positioning Satellite System.

The first area of research involved a helicopter flight-test program to evaluate the performance of Honeywell's Tetrad - a strapdown, laser gyro, inertial navigation system. The results of 34 flights showed a mean final navigational velocity error of 5.06 knots, with a standard deviation of 3.84 knots; a corresponding mean final position error of 2.66 n.mi., with a standard deviation of 1.48 n.m.; and a modeled mean-position-error growth rate for the 34 tests of 1.96 knots, with a standard deviation of 1.09 knots. Tetrad's four-ring laser gyros provided reliable and accurate angular rate sensing during the test program and no sensor failures were detected during the evaluation.

The second area's objective of the cooperative agreement was to develop criteria suitable for investigating cockpit systems in rotorcraft. This criteria led to the development of two basic simulators.

The first was a standard simulator which could be used to obtain baseline information for studying pilot workload and interactions. Figure 1 is a block diagram of this system. The instrument panel used was typical of a UH-1H helicopter. The mathematical model for the UH-1H was implemented in the Dec 11/23 minicomputer, while all instrument controls were implemented in the microcomputer data adapter.

The second was an advanced simulator which integrated the RODAAS developed by Honeywell into this simulator. Unlike the standard system, the RODAAS integrates an advanced set of avionics functions into hardware. Figure 2 is the microcomputer system which was used to interface to the RODAAS instrumentation.

The second area also included surveying the aerospace industry to determine the level of use and impact of microcomputers and related components on avionics systems. It was determined that the usage of microcomputers and related components is very extensive. A report was written pertaining to this and is a part of the appendix.

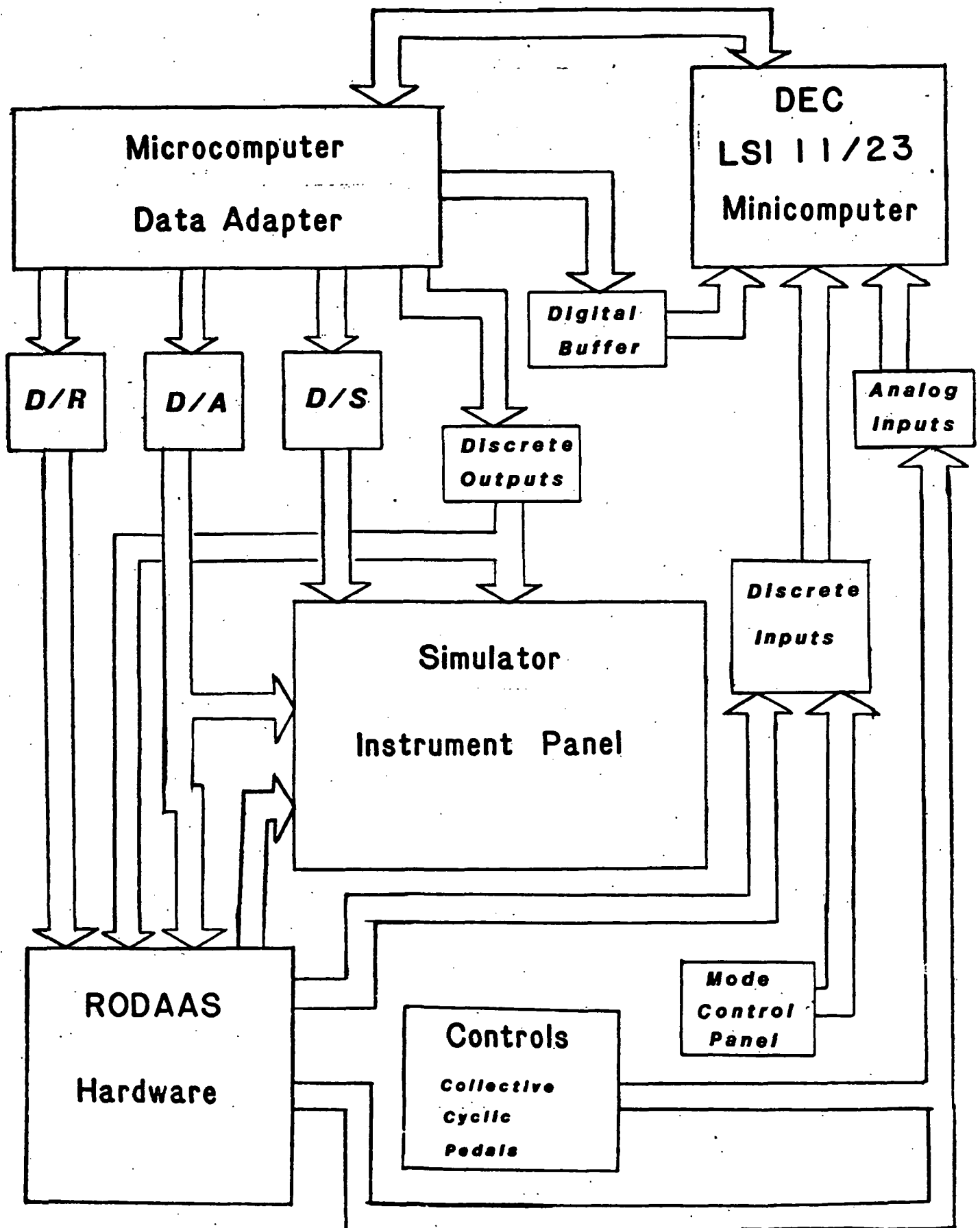
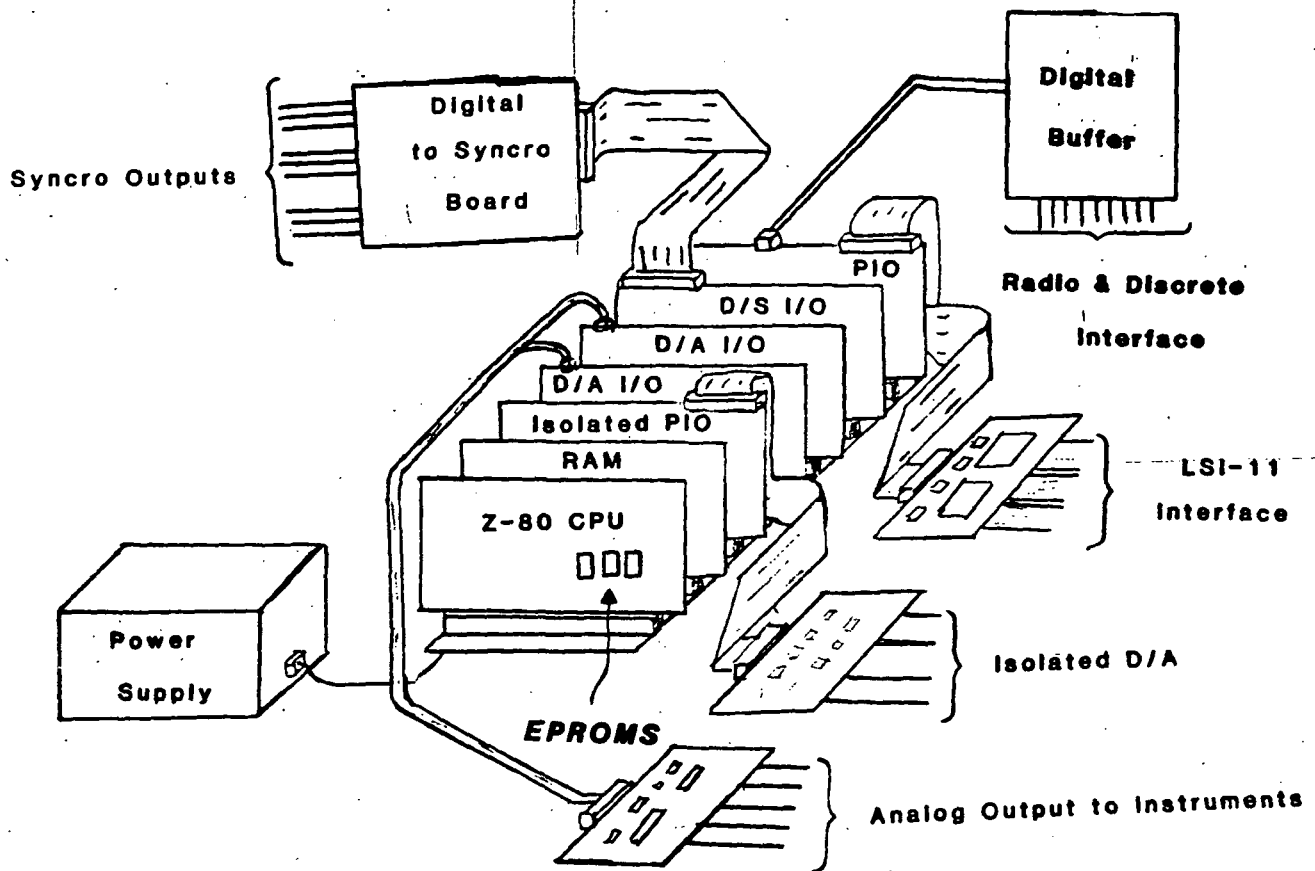
Figure 1 - Functional Diagram of System

FIGURE 2

Microcomputer System



The third area of investigation was to integrate and test the navigational capabilities of a Differential Global Positioning Satellite (Differential GPS) System, which is a prototype space-based radio navigation system developed by the Department of Defense. Work for NASA in support of the Differential GPS fell into the following areas: literature search, hardware installation in the SH3 helicopter and laboratory, technical visits to Magnavox to verify performance of a NASA interface processor (NIP) and final acceptance testing of the NIP, design and testing of software used in the NASA computer to gather data and send the data via telemetry to a ground station for recording, and support of the preliminary Differential GPS flight experiment. This report is accordingly embodied in the appendix.

Cumulative References

1. "Guidance and Flight Control Demonstration in a Helicopter Flight Environment Using a Laser-Gyro Inertial Navigation System", S.F. Schmidt, R.A. Carestia, R.J. Hruby and L.D. Corliss, National Aerospace Meeting, NASA Ames Research Center, Moffett Field, California, March 24-25, 1982.
2. "Performance of a Strapdown Ring Laser Gyro Tetrad Inertial Navigation System in a Helicopter Flight Environment:", R.A. Carestia, R.J. Hruby, and W.S. Bjorkman, National Aerospace Meeting, NASA Ames Research Center, Moffett Field, California, March 24-25, 1982.
3. "Flight Test Results of the Strapdown Ring Laser Gyro Tetrad Inertial Navigation System", R.A. Carestia, R.J. Hruby, and W.S. Bjorkman, NASA Technical Memorandum 84358, October, 1983.

The following consists of working papers written by university researchers during the grant reporting period:

<u>Contents</u>	<u>Title</u>
Appendix A	"A Survey of the Use and Trends of Microprocessors/Computers in the Avionics Industry", D. Cottrell, R. Carestia and P. Holub.
Appendix B	"Differential Global Positioning Satellite Project", A. Beckman.
Appendix C	"Report of Contract Work for University of Southern Colorado", T. Johnson.
Appendix D	"NASA Summer Project", R. Parker.

APPENDIX A

**A SURVEY OF THE USE
AND TRENDS OF
MICROPROCESSORS/COMPUTERS
IN THE AVIONICS INDUSTRY**

October 1984

**by Dr. D. Cottrell
R. A. Carestia
P. Holub**

A SURVEY OF THE USE AND TRENDS
OF MICROPROCESSORS/COMPUTERS
IN THE AVIONICS INDUSTRY

October 1984

by

Don Cottrell
Ralph Carestia
Peter Holub

ABSTRACT

In 1983, a nationwide survey was conducted of the avionics use of microprocessors and microcomputers, including their associated memories and peripheral devices. In this survey, it was determined that microprocessors and microcomputers have made a great impact upon the avionics industry. They are being used in numerous applications such as control, communications, guidance, navigation, and system monitoring. Based on this information and in light of the current state-of-the-art in computer hardware and software, this report suggests some philosophical design considerations which could influence hardware selection and attempts to predict trends in computer hardware applications for avionics systems.

TABLE OF CONTENTS

	Page
Introduction.....	1
Results of Survey.....	3
Section I - Microprocessors.....	4
Section II - Memory.....	12
Section III - I/O Ports, MPU vs. Non-MPU and IC's....	18
Section IV - Microprocessor Systems.....	19
Summary and Conclusions.....	23
Microprocessors.....	23
Memories.....	25
I/O Ports, IC's, and MPU's vs. Non-MPU's.....	26
Microprocessor Systems.....	27
References.....	29
Appendix (Survey Form).....	30

LIST OF FIGURES AND TABLES

Figure	Page
1. First Microprocessor Used.....	4
2. First Year of Microprocessor Use.....	5
3. MPU System Classification.....	6
4. MPU Data Size Bus.....	6
5. MPU Process Type.....	7
6. MPU Procurement.....	7
7. Assembled Module Type.....	8
8. MPU Applications.....	8
9. Microcomputer System Application.....	9
10. MPU Buying Decision Factors.....	10
11. Microprocessor Architecture.....	10
12. Advanced Architecture Use.....	11
13. Maximum System Memory Access Time.....	13
14. Memory System Requirements.....	14
15. Memory System Requirements.....	15
16. Memory Buying Decision Factors.....	16
17. External Data Storage.....	17
18. Key Discrete Devices.....	19
19. System Development Tools.....	19
20. High Level Languages.....	20
21. System Throughput.....	21
22. Maximum System Power.....	21
23. Redundancy Type.....	22

Table

1. I/O Ports Supported by the Microprocessor.....	18
2. % ICS Eliminated.....	18

INTRODUCTION

Many aircraft functions including control, navigation, guidance, communications, and system monitoring are being implemented reliably and accurately with the aid of the digital computer. The use of digital computer-based avionics equipment represents a potentially high cost for modern aircraft, thus a desirable goal for avionics systems is to find ways to minimize cost and still retain high system reliability. With the advent and increasing use of the microprocessor in recent years, this goal is being met.

Due to the advanced nature of the NASA/AMES flight research program, the selection of any avionics equipment must consider the types of systems which will be representative of the post 1985 time period. Selection is complicated by the fact that both digital systems technology, and the applications of microprocessors in these systems, are changing rapidly.

In late 1982, the University of Southern Colorado decided to conduct a nationwide survey of avionics engineers on their use of microprocessors and the associated peripheral devices, such as memory, in avionics systems. Since NASA had recently demonstrated the use of microprocessors in an advanced avionics system, the University and NASA concurred it would be desirable to know what level of microprocessor integration exists in the avionics industry. Hence, this survey report considers current state-of-the-art avionics computer hardware and microprocessors, suggests some philosophical design considerations which could influence computer hardware selection, and attempts to predict trends in computer hardware applications for avionics systems in the middle and late 1980's.

A questionnaire was developed by the University in late 1982 and early 1983. It was mailed and received by 168 avionics engineers identified as being directly involved with avionics system design in the spring of 1983 and again as a follow-up in the fall of 1983. To allow time for the maximum number of responses, analysis of the data did not begin until January of 1984. The number of questionnaires returned was 64, which constituted a 38% response.

This report should be valuable to the systems engineer in designing avionics systems because it includes detailed data on the types of microprocessors, memories, microprocessor applications, and programming languages presently used or anticipated. A copy of the survey form is included in the Appendix.

RESULTS OF SURVEY

The data, graphs, and tables are presented in this section according to the survey form given in the Appendix. There are twenty-eight questions arranged in the following outline form:

Section I: Microprocessors

- A. First type used
- B. When first used
- C. Plans for purchase
 - 1. Type (Standard, dedicated or custom)
 - 2. Bit size (4, 8, 12, 16, over 16)
 - 3. IC type (CMOS, NMOS, etc.)
- D. Procurement and purchasing
- E. Applications
- F. Architecture
 - 1. Performance
 - 2. Standard or advanced

Section II: Memory

- A. Type
 - 1. Semiconductor
 - 2. Bubble
- B. Access time
- C. Total memory required
 - 1. 1982
 - 2. 1983
- D. Purchasing Factors
 - 1. Speed
 - 2. Performance
 - 3. Price
 - 4. Others
- E. EPROM to mask, ROM capability
- F. External data storage

Section III: Miscellaneous

- A. I/O ports
- B. Critical discrete devices and IC's for MPU
- C. Does MPU system replace a current non-MPU based system?

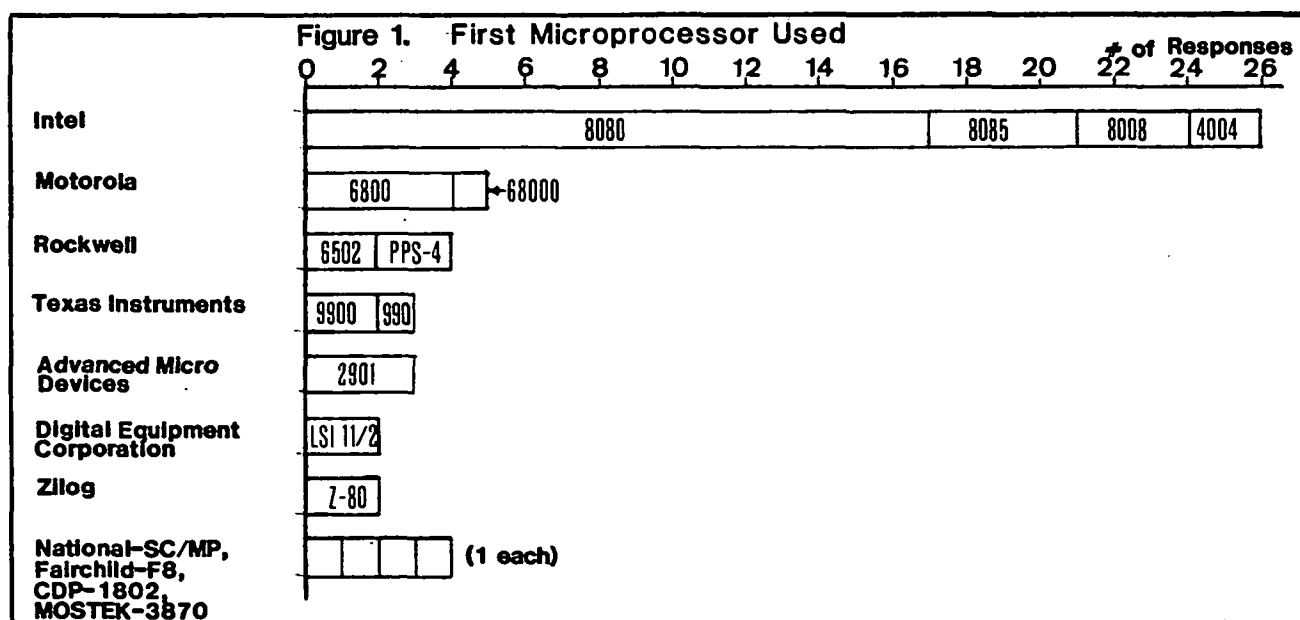
Section IV: Microprocessor Systems

- A. Will MPU system use system development tools and which ones?
- B. Importance of high level languages
- C. Throughput and maximum power of microcomputer system.
- D. Does MPU use "Built-In-Test or redundancy?
 1. What type?
 2. Time critical?
 3. Synchronous or asynchronous?
 4. Failure detection threshold?

SECTION I - MICROPROCESSORS

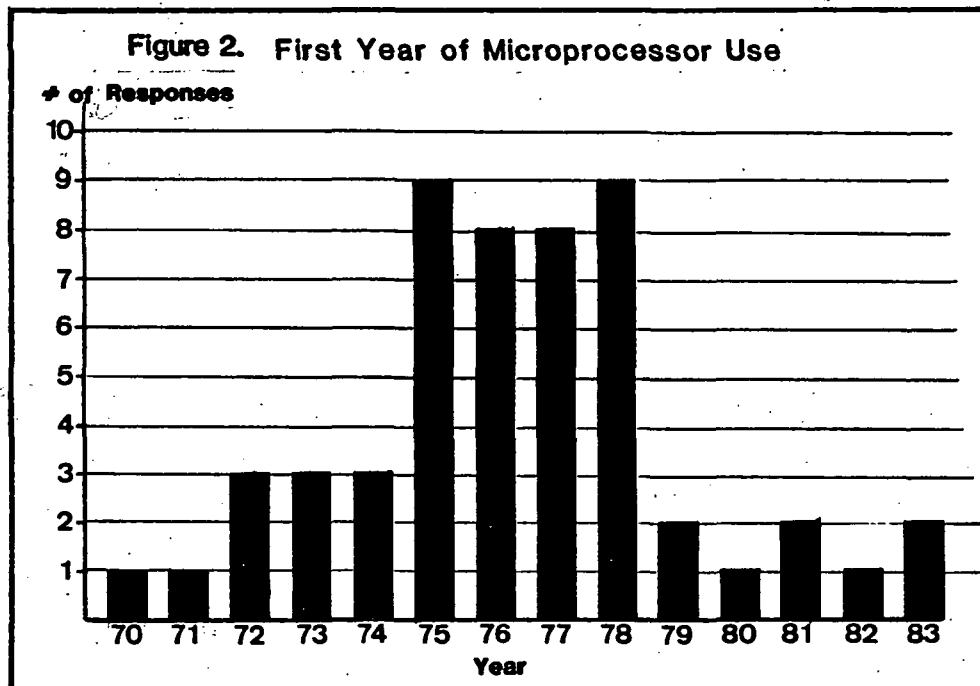
Q 1 - What was the first microprocessor you used?

The bar graph of Figure 1 shows the results. The greatest number of respondents (26) selected Intel's microprocessors with the Intel 8080 having the highest frequency of use in avionics possibly because it was one of the first available. Mororola's 6800 was second, with five responses. Other types of microprocessors used include those manufactured by Rockwell, Texas Instruments, Advanced Micro Devices, Digital Equipment Corporation, Zilog, National, Fairchild, and Mostek.



Q 2 - What year did you first use microprocessors?

This data is given in Figure 2 and indicates that most avionics specialists first began using the microprocessor between the years 1975-1978. This period coincides with the introduction of 8 bit microprocessors. Responses ranged from as early as 1970 to as late as 1983.



Q 3 - In 1983, have you purchased or do you plan to purchase MPU systems?

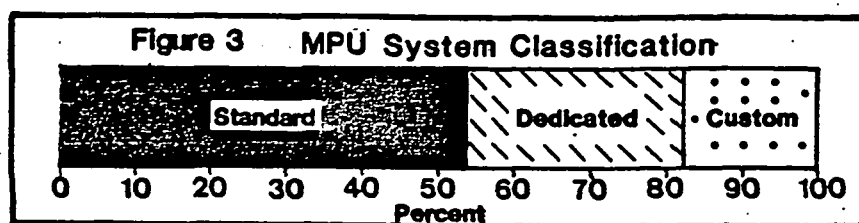
91% answered yes, 7% answered no, and 2% of the respondents did not answer the question.

Q 3A - If yes, will they be standard, dedicated, or custom?

Figure 3 shows the results:

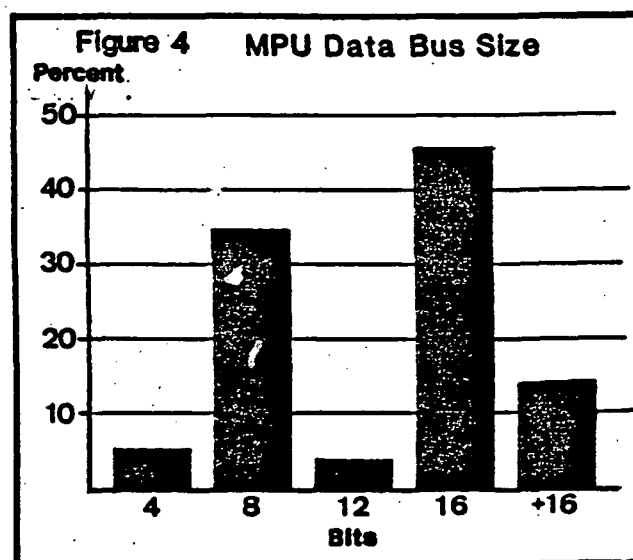
1. 55% are standard MPU's.
2. 29% are dedicated MPU's
3. 16% are custom MPU's.

It is likely that the majority use standard MPU's because of their flexibility for system design.



Q 3B - Have they been or will they be 4 bit, 8 bit, 12 bit, 16 bit, or over 16 bit?

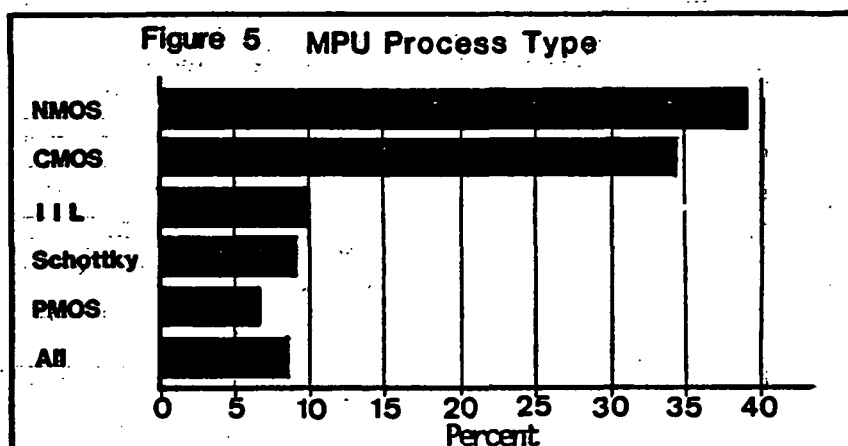
The results of this question are given in Figure 4. The 16 bit MPU is the most popular, followed by the 8 bit MPU. The over 16 bit MPU is at 13% of the total, which might indicate its growing popularity.



ORIGINAL PAGE IS
OF POOR QUALITY

Q 3C - Will the MPU's be IIL, Schottky, CMOS, NMOS, or PMOS?

As shown in Figure 5, NMOS and CMOS lead the field at 36% and 34% respectively; the others range from around 5% to 9%. NMOS devices have been popular and are in high production by manufacturers. They are also usually faster than PMOS MPU's. CMOS may be slower than NMOS or PMOS; they are low power devices, and are relatively unaffected by variations in power supply voltage (NMOS MPU's require well-stabilized power supplies). The IIL and Schottky devices are normally chosen where high speed is essential. Their high power requirements may be the reason they have limited use in avionics systems.



Q 4A - How will you procure your MPU'S?

As illustrated in Figure 6, this question breaks down as follows:

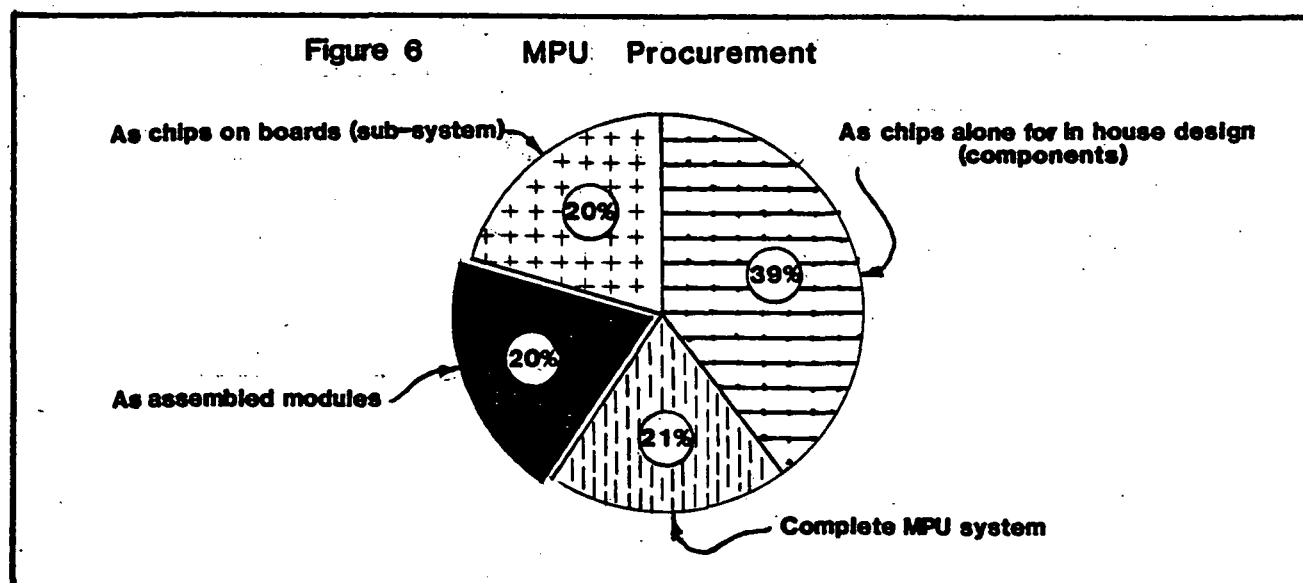
As chips alone for in-house design (components) - 39%

As chips on boards (sub systems) - 20%

As assembled modules - 20%

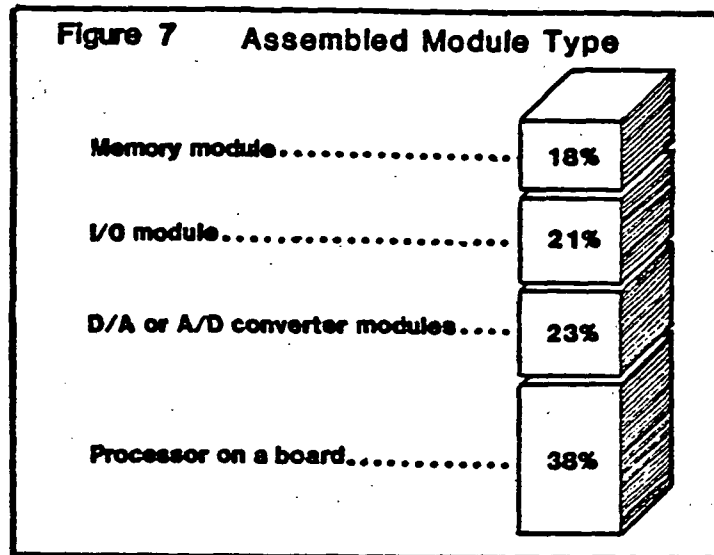
Complete MPU System
(includes memories and power supplies) - 21%

It appears that the greater percentage of respondents preferred to do their own in-house design using MPU chips. This corresponds with the results of Question 3A.



Q 4B - If assembled modules, which ones do you contemplate buying?

As shown by Figure 7, the largest percentage of designers, 38% contemplated buying the processor-on-a-board.



Q 5 - Does your MPU application relate to:

Digital System?

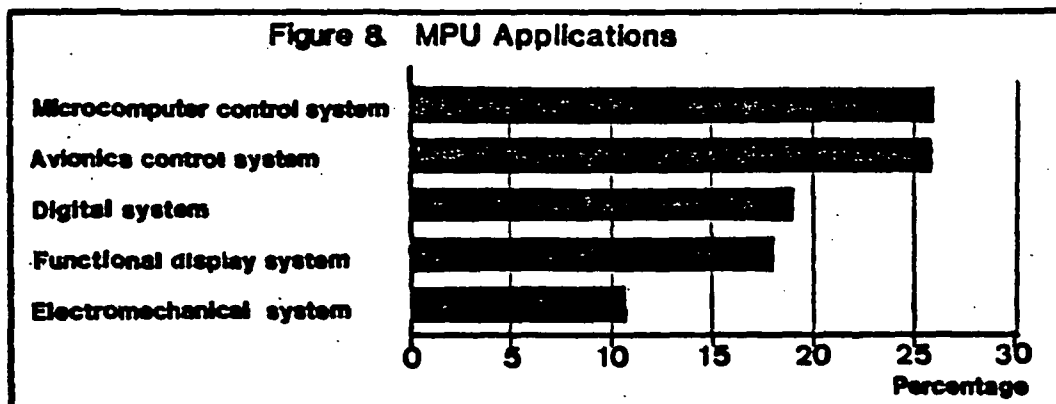
Electromechanical System?

Microcomputer Control System?

Avionics Computer System?

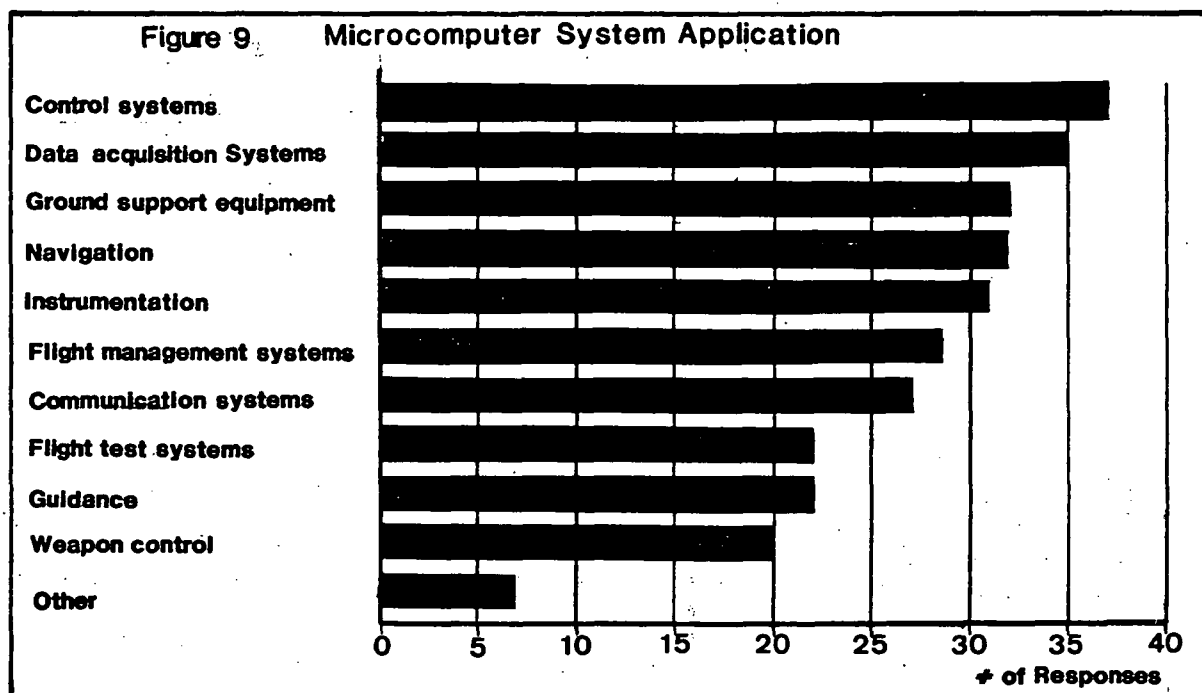
Functional Display System?

The results, given by Figure 8, indicate that microcomputer control systems and avionics control systems have the highest application of MPU's at 26% each.



Q 6 - In what applications do you use your microcomputer systems?

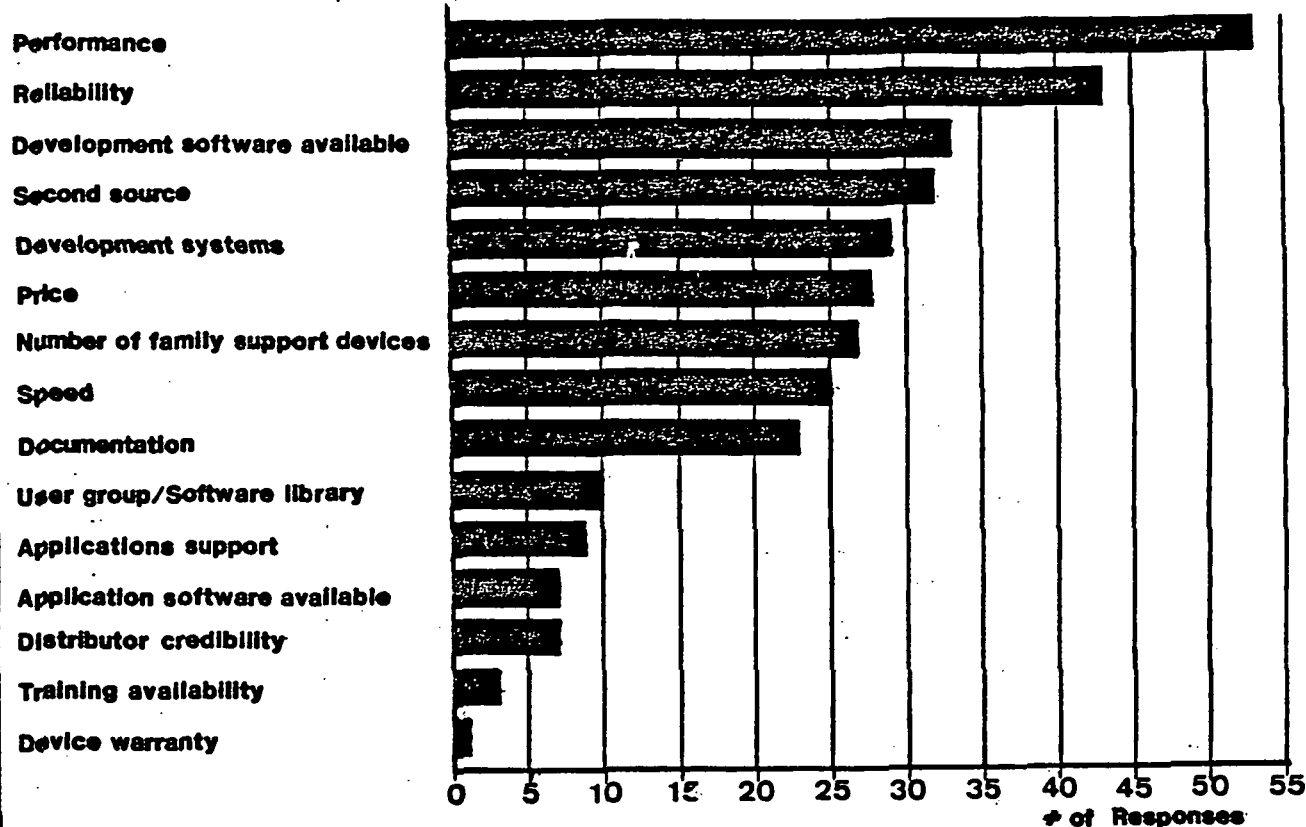
Figure 9 shows a fairly even distribution of applications. Control systems and data acquisition systems had the highest number of responses at 37 and 35 respectively. Most of the other applications were fairly closely grouped, ranging from 20-32 responses. Other applications had only 7 responses.



Q 7 - Which of the following factors do you feel are the most important in your company's MPU buying decisions?

As indicated by Figure 10, 15 different factors, such as price, performance, speed, etc., were considered. Performance was the factor most cited, followed by reliability and development software availability, as well as a second source. The other factors fell in a range of 1-28 responses.

Figure 10 MPU Buying Decision Factors

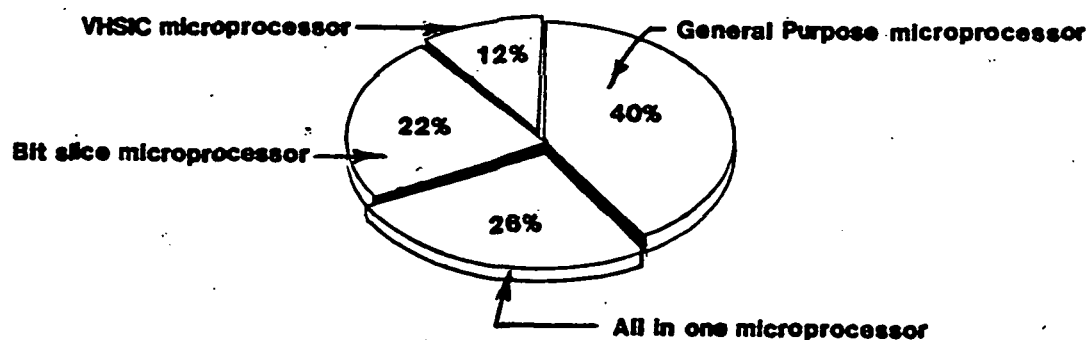


Q 8 - Is the microprocessor architecture a:
 general purpose microprocessor?
 all in one microprocessor?
 bit slice microprocessor?
 VHSIC microprocessor?

ORIGINAL PAGE IS
 OF POOR QUALITY

The pie chart of Figure 11 points to the general purpose microprocessor at 40% as the most popular architecture. The other less popular types of architectures range from 12% to 26%, but comprise a significant percentage of the total.

Figure 11 Microprocessor Architecture



Q 9 - Did the microprocessor perform as expected?

89% answered yes, 9% indicated no, and 2% gave no answer.

Q 10 - If not, tell why.

Generally, the MPU's did perform as expected; if they did not, then the major reason given was that the MPU was not fast enough:

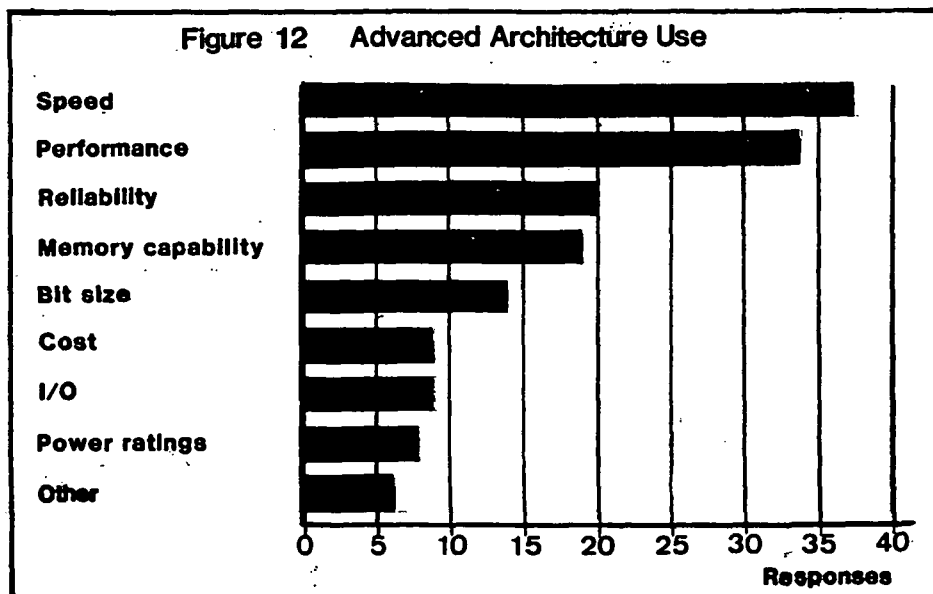
Not fast enough	- 67%
Limited I/O ports	- 17%
Limited interrupt capability	- 17%
Limited memory addressing capability	- 0%
Other	- 0%

Q 11A- Will you be using advanced architectures such as bit slice, Z8000, 68000, etc., in the future?

91% of all respondents said yes to this question, while 7% said no, and 2% gave no answer.

Q 11B- If yes, what general application and why?

41% of all respondents planned to use advanced architectures for avionics flight control and management systems; the rest did not indicate their application. The reasons for use of advanced architectures included speed, performance, reliability, memory capability, bit size, cost, I/O, power ratings, etc., in that order. See Figure 12.



Q 11C - If no, why not?

Only 3 respondents cited cost and 2 cited power ratings as reasons for not using advanced architectures. None of the respondents selected any of the other reasons listed in the survey.

SECTION II MEMORY

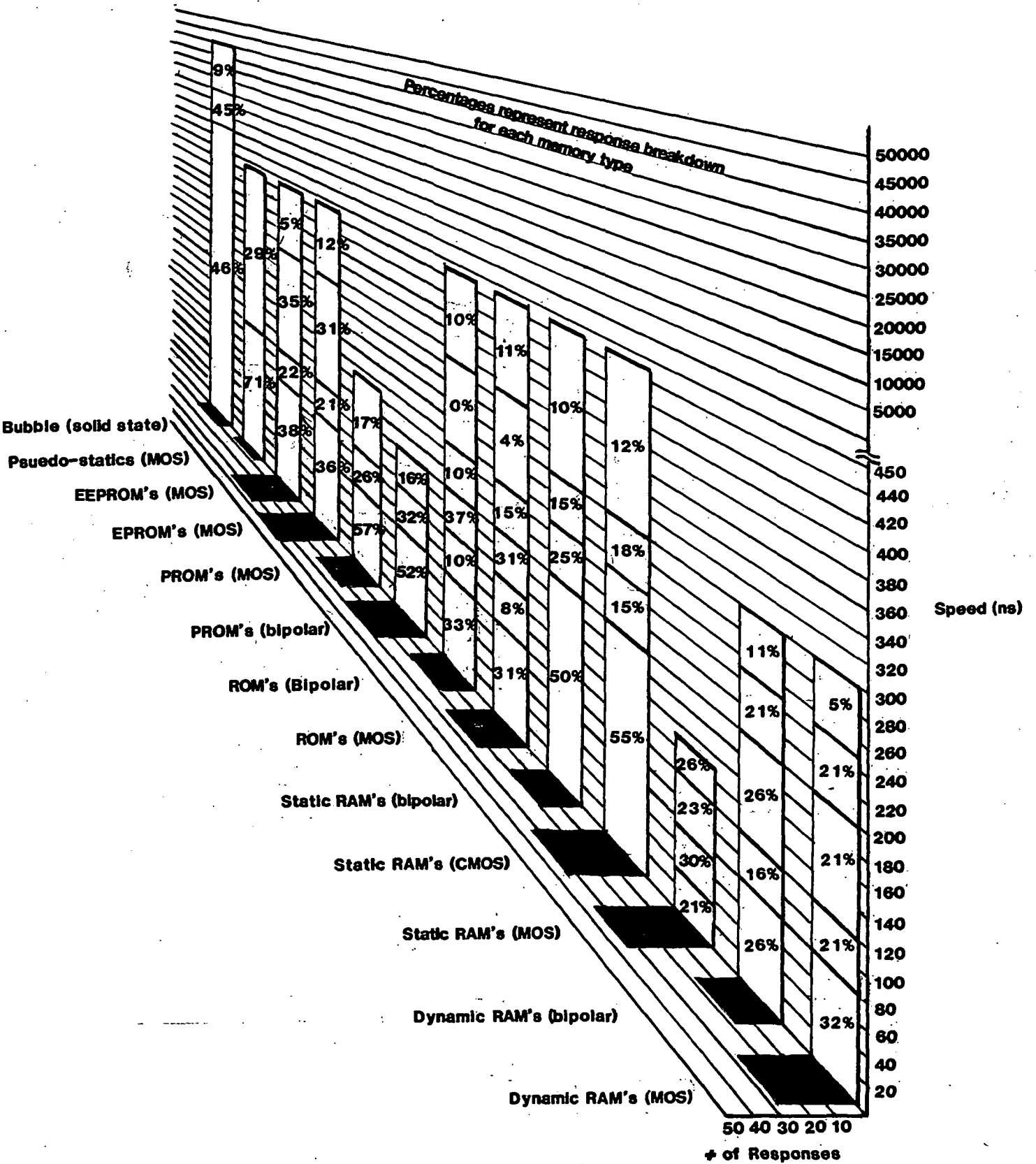
Q 12 - Have you used or do you plan to use semiconductor memories and/or bubble memories during 1982 or 1983?

The largest percentage by far, 84%, planned to use semiconductor memories, and only 14% planned to use bubble memories. Two percent of all those surveyed planned to use neither.

Q 13 - What speed (maximum access time in ns) is best suited for your system? (Check all that apply.)

For the various kinds of memories, such as dynamic RAM's, static RAM's, ROM's, PROM's, EPROM's, and bubble, the results are shown in Figure 13. For example, for CMOS static RAM's, 55% of the respondents indicated 20 to 200 ns speeds as being most suited to their system. Fifteen percent desired speeds of 200 to 250 ns, eighteen percent wanted speeds of 250 to 300 ns, and twelve percent felt 300 to 450 ns speeds would be best. The higher speeds of 20 ns to 200 ns are the most necessary in most situations, while the lower speeds ranging from 300 ns to 5000 ns are the least suited.

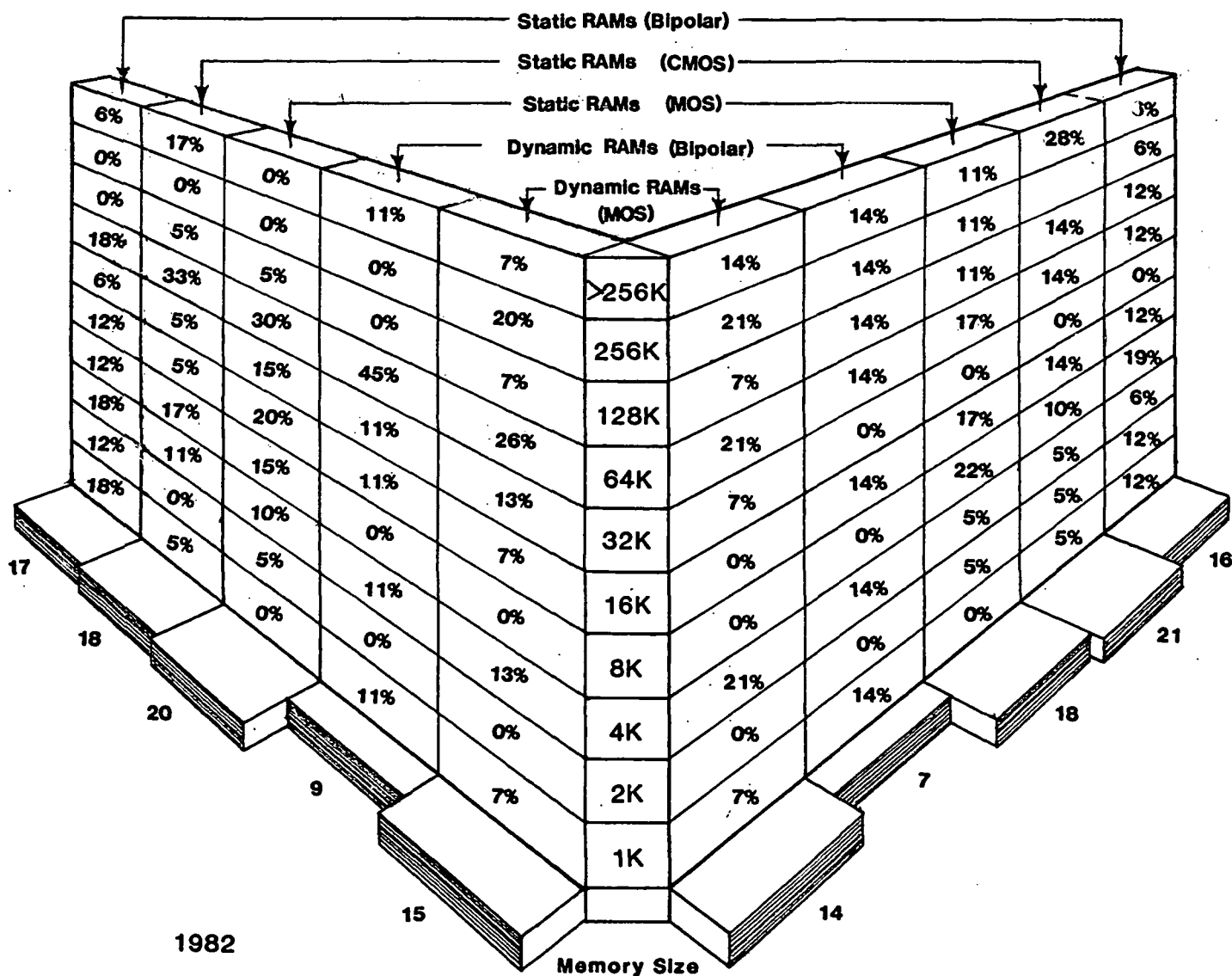
Figure 13. Maximum System Memory Access Time



Q 14 - How much total memory will your system require in 1982 and 1983?
(Bytes)

As shown by Figures 14 and 15, the 64K memory has the largest percentage of use in most categories, including Dynamic RAM's, Static RAM's, ROM's, PROM's, etc., in both 1982 and 1983. There is a slight trend in the usage of more memory (128K, 256K, and > 256K) between 1982 and 1983. EPROM's (MOS), and Static RAM's (MOS) had the greatest number of responses in 1982, whereas in 1983, EPROM's (MOS) and Static RAM's (CMOS) generated the largest number of responses.

Figure 14. Memory System Requirements

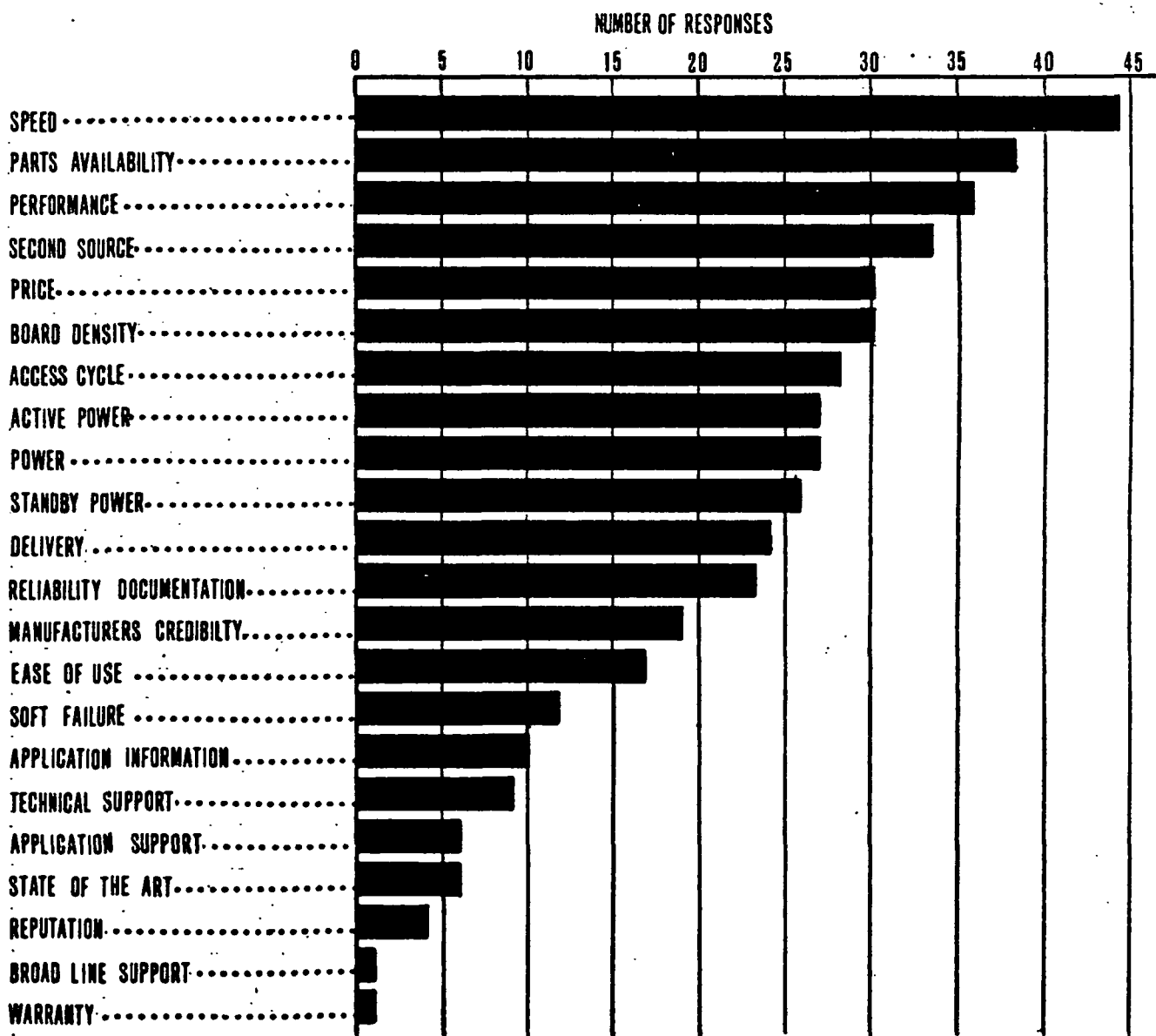


-Note: Percentages represent response breakdown for each memory type

Q 15 - Which of the following factors do you feel is important in your memory buying decisions? (Check all that apply.)

All factors are listed on the bar graph of Figure 16. The response distributions range from 1 to 44 with speed of the memory at 44 responses being the factor considered to be most important, followed by parts availability and performance, which are nearly equal. State-of-the-art, application support, warranty, broad line support, and reputation had the fewest responses and were thought to be the least important.

Figure 16. Memory Buying Decision Factors



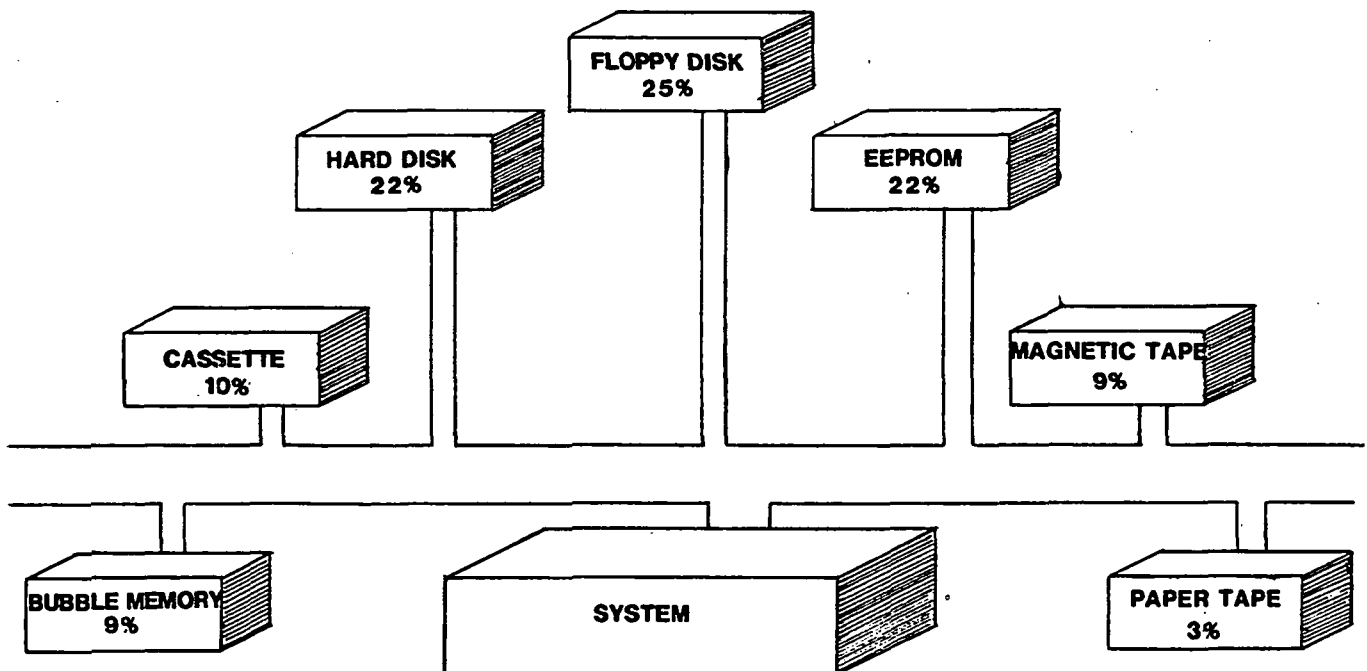
Q 16 - In your system, how important is EPROM to Mask ROM capability?

35% said they will not use EPROM without pin compatible mask ROM. However, a larger number, 47%, indicated compatibility was not important. Eighteen percent of the respondents gave no answer.

Q 17 - Does the system incorporate any external data storage?

The results of this question are presented in Figure 17; it shows that the floppy disk (25%) closely followed by the hard disk (22%) and EEPROM (22%) are the most often used for external data storage. These three comprise 69% of the total and cassettes (10%), paper tape (3%), magnetic tape (9%), and bubble memory (9%) comprise the remainder (31%).

Figure 17. External Data Storage



SECTION III - MISCELLANEOUS - I/O Ports, etc.

Q 18 - How many I/O ports does the microprocessor support?

The respondents indicated the following in Table 1:

No. of Ports	% of Total Respondents
0 - 10	55
10 - 100	27
Over 100	18

TABLE 1
I/O PORTS SUPPORTED BY THE MICROPROCESSOR

So, the majority of respondents used ten or fewer ports in their applications.

Q 19A - Does your MPU-based system replace a current non-MPU based system?

39% of those surveyed indicated yes and 61% said no.

Q 19B - If yes, how many discrete IC's or other semiconductor elements have been eliminated?

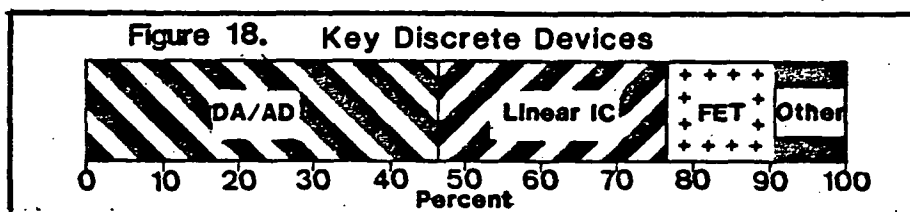
Table 2 indicates that of the respondents (13) who answered this question, 61% said they eliminated over 200 IC's, 32% said they eliminated 20-100 IC's, and 7% indicated only 0-10 IC's. The other categories had none.

No. of IC's Eliminated	% Respondents
0 - 10	7
10 - 20	0
20 - 100	32
100 - 200	0
Over 200	61

TABLE 2
% IC's ELIMINATED

Q 20 - What discrete devices or IC's are critical or key to your MPU systems?

As shown by Figure 18, the most critical is the DA/AD converter (46% of total), followed by linear IC's (30% of total), FET's (15%), and other (9%).



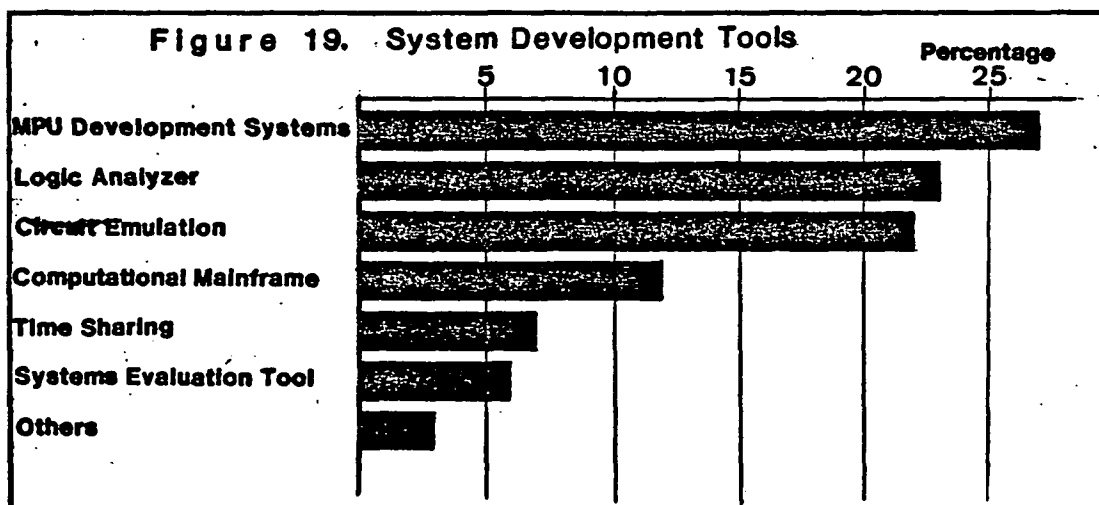
SECTION IV - MICROPROCESSOR SYSTEMS

Q 21A - Will you use system development tools that are marketed by micro-processor manufacturers?

By far, the majority (88%) said yes while only 12% said no.

Q 21B - What specific development tools do you/will you use?

Figure 19 shows that 27% of the respondents plan to use a MPU development system, 23% will use a logic analyzer, and 22% will utilize circuit emulation. Use of other tools, such as computerized main frame, time sharing, etc., ranged from 3% to 12% of the total.

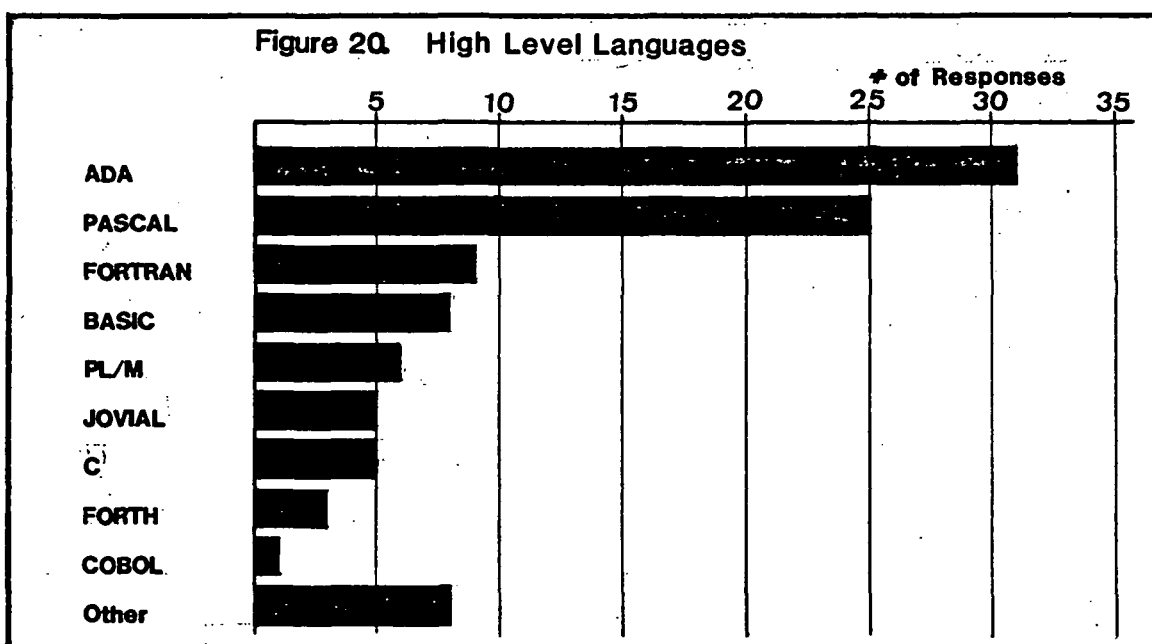


Q 22A - How important is availability of high level languages?

The majority, 66%, felt it was very important to have high level languages available, 26% said it was somewhat important, and only 8% indicated it was not important.

Q 22B - If a high level language is important, which do you favor?

As illustrated in Figure 20, ADA (the high level language suggested as a government standard) had the greater percentage of popularity, PASCAL was a strong second, FORTRAN and BASIC were nearly identical, while the others (JOVIAL, C, etc.) were less popular.



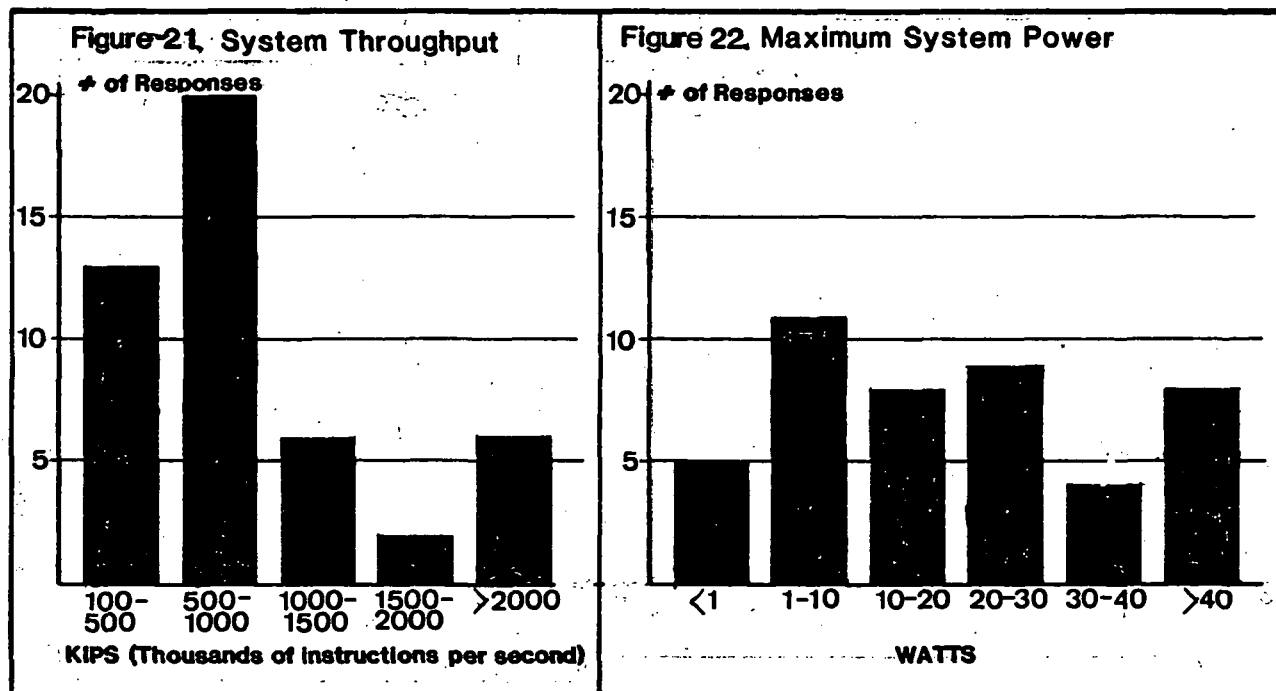
Q 23 - What is the throughput and the maximum power of your microcomputer systems? (Please specify the general application of the system.)

Of the 32 respondents answering this question, 19 or 59% were using their MPU systems for applications directly related to flight control systems. The majority of the remaining respondents' applications were indirectly related to avionics.

Figures 21 and 22 indicate the KIPS (thousands of instructions per second) and maximum power in watts for the various applications.

The largest percentage of KIPS was in the 500-1000 range at 41% of all responses, and the next highest percentage of KIPS was in the 100-500 range at 28% of the total responses.

Power requirements were distributed fairly evenly from less than 1 to greater than 40 watts with the 1-10 watt range having the highest value at 25% of all responses.



Q 24 - Does the microcomputer system have built-in-test fault?

72% of the respondents answered yes, while only 28% answered no. This apparently indicates that most designers prefer this feature.

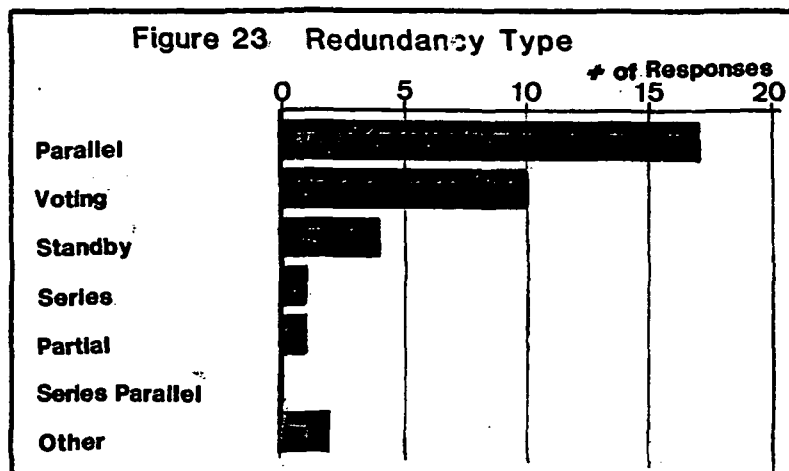
Q 25 - Do you use redundancy in the microcomputer systems you are designing?

The answer yes and no were approximately at the same percentage as follows: 53% said yes and 47% said no.

Q 26 - If yes, what types of redundancy do you use? (Please specify the general application of the system.)

As noted in Figure 23, 17 of those responding chose parallel redundancy. The second largest was voting redundancy with 10 responses.

[Most applications were listed for flight control systems and avionics.]



Q 27 - Are the microcomputers used in flight systems where failures are time critical?

74% said yes and 26% said no to this question.

Q 28 - In multiple microcomputer systems, are the microcomputers operated synchronously or asynchronously in your redundant application?

The answers to this question were distributed almost evenly: 48% indicated synchronous operation and 52% said the operation was asynchronous.

SUMMARY AND CONCLUSIONS

Microprocessors:

The results of this survey definitely support the assertion that microprocessors/microcomputers have a high frequency of use in avionics systems and have already replaced many of the minicomputers of the 1970's. This is not surprising since microprocessors have been available since the early 1970's. Due to their versatility, low cost, small size, and low power consumption they have been widely accepted by avionics engineers. In line with this, probably because they were the first to make microprocessors commercially available, Intel's microprocessor chips are the most often used in the avionics industry. The next most often used were Motorola's 6800 and 68000. Most avionics specialists started using the microprocessor between the years 1975-1978, but some started as early as 1970 and others as late as 1983.

The majority of the respondents (55%) purchased standard microprocessors and nearly everyone (91%) planned to buy microprocessor systems in 1983. There is also a trend to use microprocessors of 16 bits or more.

NMOS and CMOS (MPU) microprocessors are the most popular in avionics. The NMOS MPU device is usually faster than the PMOS MPU. The NMOS MPU's require well-stabilized power supplies, whereas the CMOS MPU is a low power device and relatively unaffected by variations in power supply voltage. The IIL and Schottky devices were normally chosen where high speed is essential, but require more power, which may explain why their use is limited in avionics systems.

The greater percentage of respondents (39%) preferred to do their own in-house design using MPU chips. Of those who would choose to purchase already assembled modules (20%), most of them would select the processor-on-a-board.

Applications of the MPU to avionics vary greatly, but the microcomputer control system and avionics computer system were generally the most prevalent. More specifically, control systems, data acquisition systems, navigation, ground support equipment, instrumentation, flight management systems, communication systems, guidance, flight test systems, etc., are the applications listed in order of highest use.

Performance and reliability appear to be the highest priorities in company MPU buying decisions. Secondary buying considerations include price, speed, documentation, availability of a second source, development systems, and software availability.

89% of the respondents indicated that their microprocessors typically performed as expected. If they did not, the reason most cited was that the MPU was not fast enough. With respect to MPU architecture, the general purpose MPU has the most popular usage, but the all-in-one MPU (which is a computer on a chip) and the bit slice MPU (MPU's connected in cascade to allow larger size words) together comprise the larger percentage. The VHSIC (very high speed IC) usage at this time is limited, although this will surely change with greater availability in the future.

Advanced architectures are certainly on the horizon for avionics with new chip designs such as the Z8000 and M68000 becoming available. Most

avionics specialists (91%) said they would be using these in applications relating to flight control and flight management systems.

Microprocessor manufacturers have been able to design a large variety of microprocessors and these permit many avionics applications. Also, the MPU is well-suited to the modular concept, which allows greater flexibility of system design. At present though, there is not enough standardization of most microprocessor units between manufacturers and more needs to be done.

In recent years, the cost distribution of computer system components, such as the CPU, memory, and peripheral devices, has changed drastically. In the early 1970's, the cost of minicomputer systems was in the \$10K-20K range with the CPU being the major expense. Later, system prices for computers based on LSI technology were in the \$1K-5K range and memory and peripheral devices had taken over as the major cost factors. Today that has been accelerated by the microprocessor, which is very inexpensive compared to memory and peripheral devices.

Memories:

Memories are a major portion of the cost of avionics computers, but there is a trend toward lower cost. Currently memories are basically of two types: (1) semiconductor storage and (2) magnetic storage. The semiconductor type is high speed and low capacity (although capacities have improved) when compared to magnetic storage which is low speed and high capacity. A typical semiconductor memory might have 64K bytes of storage capacity with a 200 ns access time, while a magnetic memory might have 1 megabyte of storage, but only 5000 ns access time (as in a bubble memory).

Speed was cited as very important in memory buying decisions. Probably because of this, semiconductor memory was chosen as most popular for avionics systems. 84% planned to use semiconductor memories, but only 14% planned to use bubble memories (2% planned to use neither). The higher speeds (maximum access times) of 20 ns to 200 ns are the most prevalent in avionics, while the lower speeds ranging from around 300 ns to 5000 ns are the least utilized.

The total memory that avionics systems require presently centers around the 64K capacity which had the largest usage, but there is a trend towards the larger memories of 128K, 256K, and greater.

If a system incorporates any external data storage, it would typically be a floppy disk, hard disk, or EEPROM. Cassettes, magnetic tape, and bubble memories are not typical for external data storage.

I/O Ports, IC's, and MPU's vs. Non-MPU's:

Slightly more than half of the designers used ten or fewer I/O ports in their applications, whereas around one-quarter used 10-100 I/O ports and less than 20% used over 100 I/O ports.

It was determined that 39% of all respondents indicated they replaced a current non-MPU based system with one that is MPU-based; 61% said they did not. If they had replaced the non-MPU based system, it was found that around two-thirds of the designers had eliminated over 100 discrete IC's; the other one-third eliminated 20-100 discrete IC's. This is a significant reduction which means fewer interconnections, thus increased system reliability. But this also increases the difficulty of error detection and

isolation. The discrete devices, or IC's, most critical to MPU systems were the DA/AD converter, linear IC's, FET's, etc., in that order.

Microprocessor Systems:

Strong trends in microprocessor systems for avionics indicate: (a) System development tools marketed by microprocessor manufacturers will have a great number of applications. The tools having the largest use will be MPU development systems, logic analyzers, and circuit emulators. (b) The availability of high level languages is quite important to at least two-thirds of the experts surveyed. The high level language that had the greatest popularity was ADA, followed by PASCAL, FORTRAN, and BASIC. It is not surprising that ADA was the most popular, since it is the one which has been adopted as the government standard. (c) Most MPU systems have built-in-testing which indicates a strong preference for it by most designers (nearly three-quarters of the total). (d) More than half of the respondents used redundancy in their microcomputer systems. When redundancy was used, the most popular method was parallel redundancy, followed by voting redundancy. (e) Failures in flight systems using microcomputers are typically time critical, according to nearly three-fourths of the respondents.

In conclusion, avionics systems incorporate microprocessors to a large extent in their designs and handle a wide range of applications, including flight management systems, flight control systems, navigation and guidance, data acquisition, communications, etc. There has been a reduction in the cost, size, weight, and power due to the microprocessor and VLSIC's. Use

of BIT (Built-In-Testing), redundancy and fewer interconnections has brought about greater reliability in airborne electronics systems.

The design philosophy of avionics systems has changed over the last decade. In the 1970's the concept was to use one large central computer to perform the desired functions. Now, however, most designers use the distributed techniques in which several or more microprocessors are used to perform the necessary functions. This idea has resulted in greater performance and reliability.

REFERENCES

1. Seacord, Charles L. and Vaughn, Darrel of Honeywell, Inc., "Computer Technology Forecast Study for General Aviation," NASA CR-37889, Contract No. NAS2-8971, June 20, 1976.
2. Blatt, Paul E. and Yechout, Thomas R., "Digital Flight Control Technology Roadmap," Technical Report AFFDL/FGL-TM-73-65, Wright Patterson AFB, Ohio, January, 1977.
3. Grgurich, John and Bradbury, Peter of Sperry Flight Systems, "Stoland Final Report," NASA CR-137972, Contract No. 2-6567, November, 1976.
4. Frost, Cecil R., "Military CPU's," Datamation, July 15, 1970.
5. Bursky, Dave and Barnes, Dave, "Choosing a μ p By Its Capabilities is a Growing Family Affair," Electronic Design, July 5, 1977.
6. Tsois, Steve, "IMP-16 Helps Small Planes Fly a Straight Course," Electronics, June 9, 1977.
7. Klass, Phillip J., "Civil Needs Spur Technological Advances," Aviation Week & Space Technology, August 16, 1976.
8. Money, S. A., Microprocessor Data Book, McGraw-Hill, 1982.
9. Lerner, E., "The Automated Cockpit," IEEE Spectrum, February, 1983.
10. Schmid, C. F., and Schmid, S. E., Handbook of Graphic Presentation, John Wiley & Sons, 1979.

APPENDIX

(1) What was the first microprocessor that you used? _____

(2) What year did you first use microprocessors? _____

(3) In 1983, have you purchased or do you plan to purchase MPU systems?

____ YES ____ NO

A. If yes, will they be:

____ Standard ____ Dedicated ____ Custom

B. Have they been or will they be:

____ 4 bit ____ 8 bit ____ 12 bit ____ 16 bit ____ Over 16 bit

____ All of these

C. Will they be:

____ IIL ____ Schottky ____ CMOS ____ NMOS ____ PMOS

____ All of these

(4) How will you procure your MPUs? (check all that apply)

____ As chips alone for in-house design (Components)

____ As chips on boards (Sub-System)

____ As assembled modules

____ Complete MPU system (includes memories and power supplies)

If assembled modules, which ones do you contemplate buying?

____ Processor-on-a-board

____ I/O module

____ Other _____

____ Memory module, size _____

____ D/A or A/D converter modules

(5) Does your MPU application relate to: (check all that apply)

____ Digital system (discretes)

____ Electromechanical system

____ Microcomputer control system

____ Avionics computer system

____ Functional display system

(6) In what applications do you use your microcomputer systems? (check all that apply)

____ Ground Support Equipment

____ Flight Management Systems

____ Flight Test Systems

____ Data Acquisition Systems

____ Communications Systems

____ Other: _____

____ Navigation

____ Guidance

____ Control Systems

____ Instrumentation

____ Weapon Control Systems

- (7) Which of the following factors do you feel are the most important in your company's MPU buying decisions?

<input type="checkbox"/> Price	<input type="checkbox"/> Reliability	<input type="checkbox"/> Development systems
<input type="checkbox"/> Performance	<input type="checkbox"/> Documentation	<input type="checkbox"/> Devel. software avail.
<input type="checkbox"/> Second src.	<input type="checkbox"/> Training avail.	<input type="checkbox"/> Applic. software avail.
<input type="checkbox"/> Distributor	<input type="checkbox"/> Applications	<input type="checkbox"/> Speed
<input type="checkbox"/> credibility	<input type="checkbox"/> support	<input type="checkbox"/> Number of family support
<input type="checkbox"/> Users group/Software library		<input type="checkbox"/> devices avail.
<input type="checkbox"/> Device warranty		

- (8) Is the microprocessor architecture a: (check all that apply)

☐ General purpose microprocessor
☐ All-in-one microprocessor
☐ Bit slice microprocessor
☐ VHSIC microprocessor (Very High Speed IC)

- (9) Did the microprocessor perform as expected?

☐ Yes ☐ No

- (10) If not, tell why?

☐ Not fast enough
☐ Limited I/O ports
☐ Limited interrupt capability
☐ Limited memory addressing capability
☐ Other _____

- (11) Will you be using advanced architectures such as bit-slice, Z8000, 68000, etc. in the future?

☐ Yes ☐ No

If yes, what general application, and why?

Application _____,

<input type="checkbox"/> Speed	<input type="checkbox"/> Power ratings
<input type="checkbox"/> Memory capability	<input type="checkbox"/> Performance
<input type="checkbox"/> Bit size	<input type="checkbox"/> Reliability
<input type="checkbox"/> Cost	<input type="checkbox"/> I/O
<input type="checkbox"/> Other _____	

If no, why not?

<input type="checkbox"/> Speed	<input type="checkbox"/> Power ratings
<input type="checkbox"/> Memory capability	<input type="checkbox"/> Performance
<input type="checkbox"/> Bit size	<input type="checkbox"/> Reliability
<input type="checkbox"/> Cost	<input type="checkbox"/> I/O
<input type="checkbox"/> Other _____	

- (15) Which of the following factors do you feel are important in your memory buying decisions? (Check all that apply)

<input type="checkbox"/> Speed	<input type="checkbox"/> Performance
<input type="checkbox"/> Parts availability	<input type="checkbox"/> Broad line support
<input type="checkbox"/> Ease of use	<input type="checkbox"/> Technical support
<input type="checkbox"/> Board density	<input type="checkbox"/> Access cycle
<input type="checkbox"/> Price	<input type="checkbox"/> Active power
<input type="checkbox"/> Reliability documentation	<input type="checkbox"/> Standby power
<input type="checkbox"/> State-of-the-art	<input type="checkbox"/> Reputation
<input type="checkbox"/> Second source	<input type="checkbox"/> Application information
<input type="checkbox"/> Application support	<input type="checkbox"/> Delivery
<input type="checkbox"/> Warranty	<input type="checkbox"/> Power
<input type="checkbox"/> Manufacturers credibility	<input type="checkbox"/> Soft failure

- (16) In your system, how important is EPROM to mask ROM compatibility?

☐ Will not use EPROM without pin compatible mask ROM
☐ Compatibility not important

- (17) Does the system incorporate any external data storage?

☐ Floppy disk ☐ Cassette ☐ Paper tape ☐ EEPROM
☐ Magnetic tape ☐ Hard disk ☐ Bubble memory

- (18) How many I/O ports does the microprocessor support? _____

- (19) Does your MPU-based system replace a current non-MPU based system?

☐ Yes ☐ No

If yes, how many discrete ICs or other semiconductor elements have been eliminated?

☐ 0-10 ☐ 10-20 ☐ 20-100 ☐ 100-200 ☐ Over 200

- (20) What discrete devices or ICs are critical or key to your MPU systems?

☐ Linear ICs ☐ FETs
☐ DA/ADs ☐ Other: _____

- (21) Will you use the system development tools that are marketed by microprocessor manufacturers?

☐ Yes ☐ No

What specific system development tools do you/will you use? (check all that apply)

<input type="checkbox"/> MPU development system	<input type="checkbox"/> Time sharing
<input type="checkbox"/> Computational mainframe	<input type="checkbox"/> Systems evaluation tool
<input type="checkbox"/> Logic analyzer	<input type="checkbox"/> Others: _____
<input type="checkbox"/> Circuit emulation	

(22) How important is availability of high-level languages?

☐ Very ☐ Somewhat ☐ Not important

If a high-level language is important, which do you favor?

☐ Basic ☐ Fortran ☐ Other: _____
☐ PL/M ☐ Pascal
☐ Cobol ☐ Ada

(23) What is the throughput and the maximum power of your microcomputer systems?
 (please specify the general application of the system)

Application 1 _____
 KIPS (thousands of instructions per sec)

☐ 100-500 ☐ 500-1000 ☐ 1000-1500 ☐ 1500-2000 ☐ > 2000

Watts

☐ <1 ☐ 1-10 ☐ 10-20 ☐ 20-30 ☐ 30-40 ☐ > 40

Application 2 _____
 KIPS (thousands of instructions per sec)

☐ 100-500 ☐ 500-1000 ☐ 1000-1500 ☐ 1500-2000 ☐ > 2000

Watts

☐ <1 ☐ 1-10 ☐ 10-20 ☐ 20-30 ☐ 30-40 ☐ > 40

Application 3 _____
 KIPS (thousands of instructions per sec)

☐ 100-500 ☐ 500-1000 ☐ 1000-1500 ☐ 1500-2000 ☐ > 2000

Watts

☐ <1 ☐ 1-10 ☐ 10-20 ☐ 20-30 ☐ 30-40 ☐ > 40

(24) Does the microcomputer system have Built-in-Test fault?

☐ Yes ☐ No

(25) Do you use redundancy in the microcomputer systems you are designing?

☐ Yes ☐ No

(26) If yes, what types of redundancy do you use? (please specify the general application of the system)

Application 1 _____

(26) Continued

<input type="checkbox"/> Parallel	<input type="checkbox"/> Series	<input type="checkbox"/> Series-parallel
<input type="checkbox"/> Standby	<input type="checkbox"/> Partial	<input type="checkbox"/> Voting
<input type="checkbox"/> Other _____		

Application 2 _____

<input type="checkbox"/> Parallel	<input type="checkbox"/> Series	<input type="checkbox"/> Series-parallel
<input type="checkbox"/> Standby	<input type="checkbox"/> Partial	<input type="checkbox"/> Voting
<input type="checkbox"/> Other _____		

Application 3 _____

<input type="checkbox"/> Parallel	<input type="checkbox"/> Series	<input type="checkbox"/> Series-parallel
<input type="checkbox"/> Standby	<input type="checkbox"/> Partial	<input type="checkbox"/> Voting
<input type="checkbox"/> Other _____		

(27) Are the microcomputers used in flight systems where failures are time critical?

☐ Yes ☐ No

(28) In multiple microcomputer systems are the microcomputers operated synchronously or asynchronously in your redundant application?

☐ Sync ☐ Async

Thank you!

OPTIONAL:

NAME: _____

ORGANIZATION: _____

ADDRESS: _____

TELEPHONE NO: _____

APPENDIX B

UNIVERSITY OF SOUTHERN COLORADO
DIFFERENTIAL GLOBAL POSITIONING SATELLITE PROJECT
FINAL REPORT
ARTHUR A. BECKMAN

INTRODUCTION

The NAVSTAR Global Positioning System (GPS) is a prototype space-based radionavigation system developed by the Department of Defense (DOD). The system is managed by the Joint Program Office (JPO) in the Air Force System Command (AFSC). When the system is fully deployed and operational in 1987 it will consist of 18 satellites in three planes of orbit.

Phase I of the deployment schedule verified that it is feasible to determine highly accurate user time and positioning by tracking four satellites placed in 10,900 nautical mile radius, 12-hour orbits. There are currently four functioning satellites in two circular orbital planes at 55 degree inclination with respect to the earth's equatorial plane, providing periodic three to four hour intervals of four satellite coverage over selected geographical areas.

Current DOD goals for NAVSTAR GPS accuracy call for 10 meter three-dimensional accuracy for military missions, and 100 meter accuracy for civil aviation. The satellites broadcast on two frequencies, L1 and L2. The signals on these two frequencies will consist of ephemeris data from the satellite data store and one of two pseudo-random codes. The coarse acquisition (C/A) code has a very short period and will be used by the military to acquire the much longer precision (P) code. The military will use the P code to navigate with, whereas civilian aviation will use the less accurate C/A code.

**ORIGINAL PAGE IS
OF POOR QUALITY**

The NAVSTAR GPS system consists of three segments. The satellite segment consists of the space-based satellites which broadcast ephemeris data and current satellite time. Each satellite has a highly accurate atomic clock on board. The ground segment consists of a number of ground-based stations to monitor satellite accuracy and status and another ground-based station to collect reports and transmit current ephemeris and correction data to each of the satellites. The final segment is the user segment which receives the satellite signals and uses them to navigate.

Four satellites are normally required for navigation. Ranges to the four satellites are determined by scaling the signal transit time with the speed of light. If the user clock could be maintained precisely with GPS system time, only three satellites would be required for navigation. The fourth satellite is required to resolve the fixed bias ambiguity between the imprecise user's clock and the GPS system time. User position is determined by simultaneous solution of four equations involving four unknowns- the three dimensional position estimates and user clock bias.

DIFFERENTIAL GPS

To prevent unfriendly nations from benefiting from the navigation capabilities of the GPS system, the DOD currently plans to corrupt the C/A code to degrade its' capabilities to no better than 100 meters (2 DRMS). This capability is sufficient for RNAV/IFR enroute, terminal, and non-precision approach and landing requirements but is not sufficient for RNAV/IFR Categories 1 or 2 precision approach and landing conditions.

ORIGINAL PAGE IS
OF POOR QUALITY

One of the major potential benefits of the NAVSTAR GPS system is the capability to provide enroute and landing capability in areas which are not covered by conventional navigation systems. These areas include large portions of Alaska and offshore oil drilling rigs.

The use of GPS in a differential mode has the capability to provide not only this remote navigation benefit but also allow category II and III approaches and landings if the FAA will modify its' regulations regarding helicopter approaches.

Differential navigation is currently in use with Omega, Loran-C, and the TRANSIT satellite system.

The differential technique significantly improves position accuracy by utilizing a surveyed ground site which receives navigation signals, applies corrections to the signals received, and broadcasts these corrections to any aircraft in the local area to apply to their solutions. The errors for all local users are normally highly correlated. The errors also tend to change more slowly than the time required to transmit them to nearby users. For Omega, Loran-C, and Transit, the technique provides improved position accuracy over distances up to a few hundred miles.

There are three basic differential GPS concepts:

1. Data link type.
2. Pseudolite type.
3. Translator type.

The data link type uses a surveyed GPS receiver which receives GPS signals, computes its' location, and computes the

corrections based on its' known location. These corrections are then transmitted to local users. It has the advantage of requiring little modification to airborne or ground GPS sets. The major disadvantage is that it requires a separate data link.

The pseudolite type also computes corrections in the same way as the data link type but instead of broadcasting just the corrections, it also generates its' own code and navigation message. It has a number of advantages but also has numerous disadvantages, mostly involving cost.

The translator type receives the GPS signal and retransmits them on an offset frequency. The airborne GPS set corrects for the errors in each signal by knowing the location of the transponder. The major advantage is the low cost of the ground equipment. The major disadvantage is that the cost of the airborne set is much higher.

The goal of the research being conducted at NASA Ames is to provide a data base of information regarding differential GPS. Major research prior to University of Southern Colorado involvement was an initial test flight series to establish the operation of the Z-set under the current satellite configuration, and modeling of the different concepts of differential GPS to determine the type which will be pursued in flight testing.

UNIVERSITY OF SOUTHERN COLORADO CONTRIBUTIONS.

Work for NASA in support of differential GPS fell into the following areas: literature search, hardware installation in the SH3 helicopter and laboratory, technical visits to Magnavox to verify performance of the NASA interface processor (NIP) and final acceptance testing of the NIP, design and testing of

ORIGINAL PAGE IS
OF POOR QUALITY

software used in the NASA computer to gather data and send the data via telemetry to a ground station for recording, and support of the preliminary differential GPS flight experiment.

The literature search involved a DIALOG based search for all published GPS related work. Pertinent articles were then read and analysed to insure that NASA has access to applicable previous research, and that research already performed will not be duplicated by NASA.

The helicopter used for NASA's GPS research is a SH3 Sea Stallion. In the previous experiment an antenna and pre-amp were mounted in tail of the aircraft and an antenna lead was run to the Magnavox Z-set receiver in the experimenter's rack. The Z-set displays its' output on a control display unit (CDU), also mounted on the rack. The Z-set receiver communicates with the CDU via a serial RS-232C link. In the previous experiment, an interface had been installed between the two devices to also transmit the data to a HP computer to record.

For the next set of experiments, a new interface was designed and built by Magnavox which connects directly to the Z-set's computer bus and transfers data via direct memory access (DMA), a method which allows the gathering of much more data. This interface includes a Digital Equipment Corporation (DEC) LSI computer to process requests for data from NASA's Norden PDP 11/34 computer, collect this data from the Z-set, do limited processing of this data, and then send it to the NASA computer using a serial RS-232C line.

Due to the volume of the data collected, the available on-

**ORIGINAL PAGE IS
OF POOR QUALITY**

board storage was deemed insufficient and all data is sent to a telemetry transmitter via a parallel port to be sent to the ground where it is recorded on magnetic tape. The Norden computer connects to a floppy disk drive for operating system and applications program storage. An HP terminal is used for operator input and to monitor data flow (See figure 1).

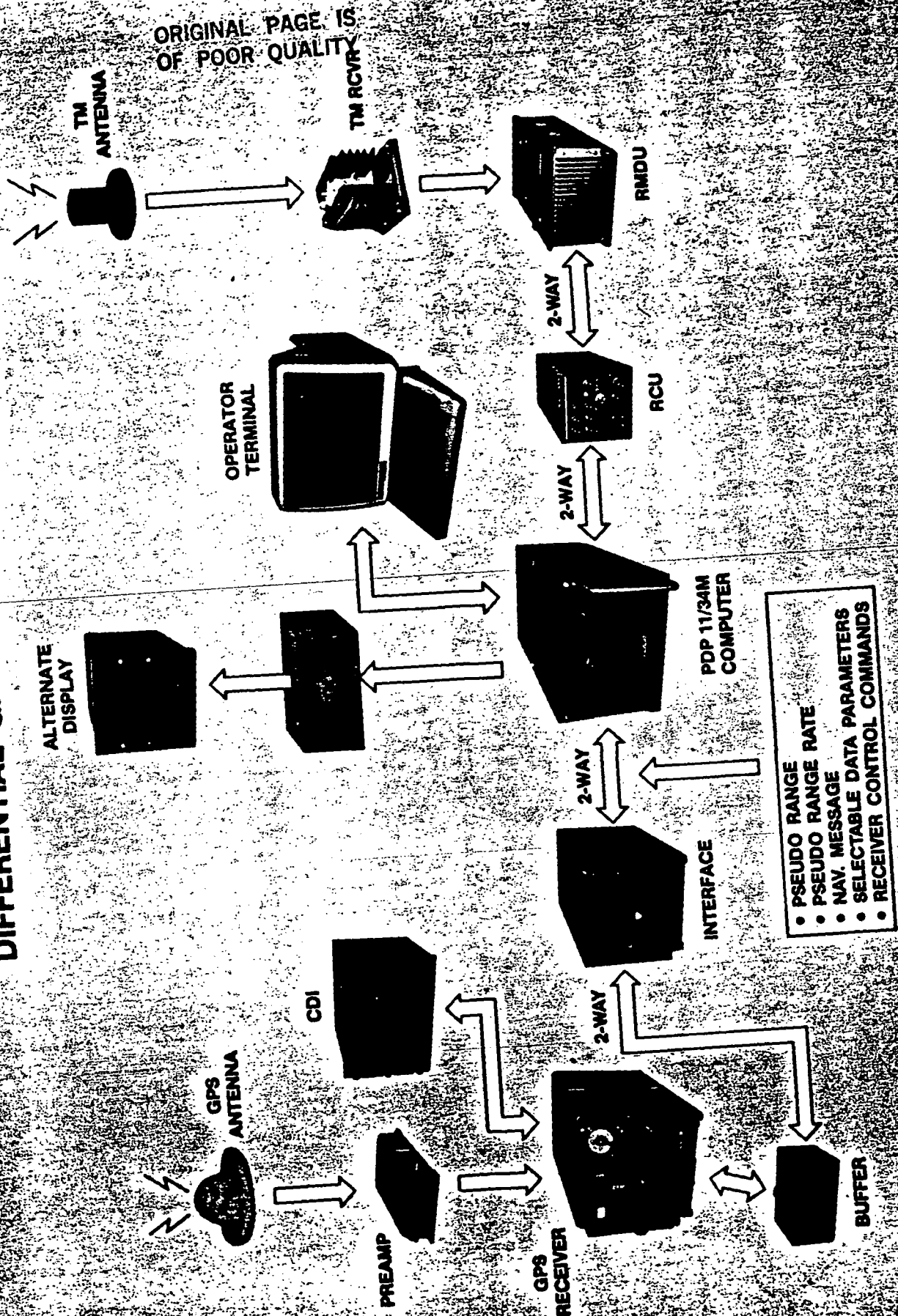
Equipment installation consisted of installing racks for the Z-set, NIP, Norden computer, and HP terminal, designing and building cables between the telemetry transmitter and Norden computer, and between the Norden computer and the HP terminal, and routing power to the various components. USC's input was to specify the layout for the racks, verify the design of the cabling, and test all cabling and overall operation of the system.

A second installation is in building 210. This laboratory installation is used for static operation to determine error characteristics, allow side-by-side comparisons with other sets, and allow testing of the various components. The antenna for the lab is on the building's roof as is the pre-amp. For previous work cabling had been installed for both a GPS antenna and a Loran antenna. The remaining equipment from the aircraft is installed in the lab itself.

To allow the operation of two sets, a second antenna was required. USC's role was to procure the cabling and connectors, coordinate the modifications to the roof antenna tower to accept a second antenna and pre-amp, coordinate the installation of the cabling, and test the system.

In order to extract information from the Magnavox Z-set

DIFFERENTIAL GPS AIRBORNE SYSTEM



ORIGINAL PAGE IS
OF POOR QUALITY

receiver for processing by the NASA computer, a contract was let to Magnavox's Advanced Products and Systems Company to build an interface processor which would receive requests for blocks of information from the Norden and then get this data from the receiver. Both the receiver and interface processor use DEC LSI computers with the Q-bus. The interface processor would communicate with the receiver via the bus and with the Norden using serial RS-232C.

Magnavox began work on the interface in January 1983 and was scheduled to finish in mid-October 1983. The author was assigned the task of monitoring progress, making liason trips to observe progress, and represent NASA during final acceptance testing.

Magnavox underestimated both the complexity of the project and the lead times for parts. By October 1983, the original software team had been replaced and a new start was scheduled for the software. The hardware was substantially complete in November and an attempt was made to write enough software to fly the originally scheduled experiments in December 1983. The author made two trips to Magnavox's location in late November to test the software. Unfortunately it was not possible to extract enough information to make the flights useful and the experiments were rescheduled for April 1984.

A meeting at NASA in December resolved the last of the conflicts with the design specifications and development then proceeded on the new schedule. Final acceptance testing at Magnavox's location in mid-March 1984 was successfully completed and the equipment was shipped to NASA.

ORIGINAL PAGE IS
OF POOR QUALITY

The purpose of the initial set of flight experiments was to provide a base of information to roughly define the actual performance capabilities of differential corrections. Since NASA's actual ground GPS station wasn't due to be finished until late 1985, Sierra Nevada Corporation was contracted to provide their services to locate a mobile second GPS set with data collection capabilities. This set would then be located on a surveyed site at the Navy's Crows Landing facility where the flight tests were scheduled to be conducted.

During a flight, both GPS sets would collect data at the same time. The airborne system would send the data to the ground station using telemetry from where it would be transmitted back to Ames for recording on magnetic tape. The ground system would send its' data via a high speed modem to Ames where it would be likewise recorded. A laser tracker at Crows would be utilized to provide a highly accurate record of where the aircraft was in space at all times during the flight. (See figure 2).

Subsequent to the flight the laser tracking data and data from the aircraft GPS set provide a record of the error experienced throughout the flight. The data from the ground based GPS set provides the errors experienced by that set by comparing the computed position and subtracting the known coordinates. This error data can then be applied to the airborne data to give a rough approximation of the benefit of differential GPS.

The author was responsible for all software running on the Norden computer on board the SH3. Three major routines were written to accomplish the tasks of communicating with the NIP and

PSEUDO-DIFFERENTIAL GPS FLIGHT EXPERIMENT

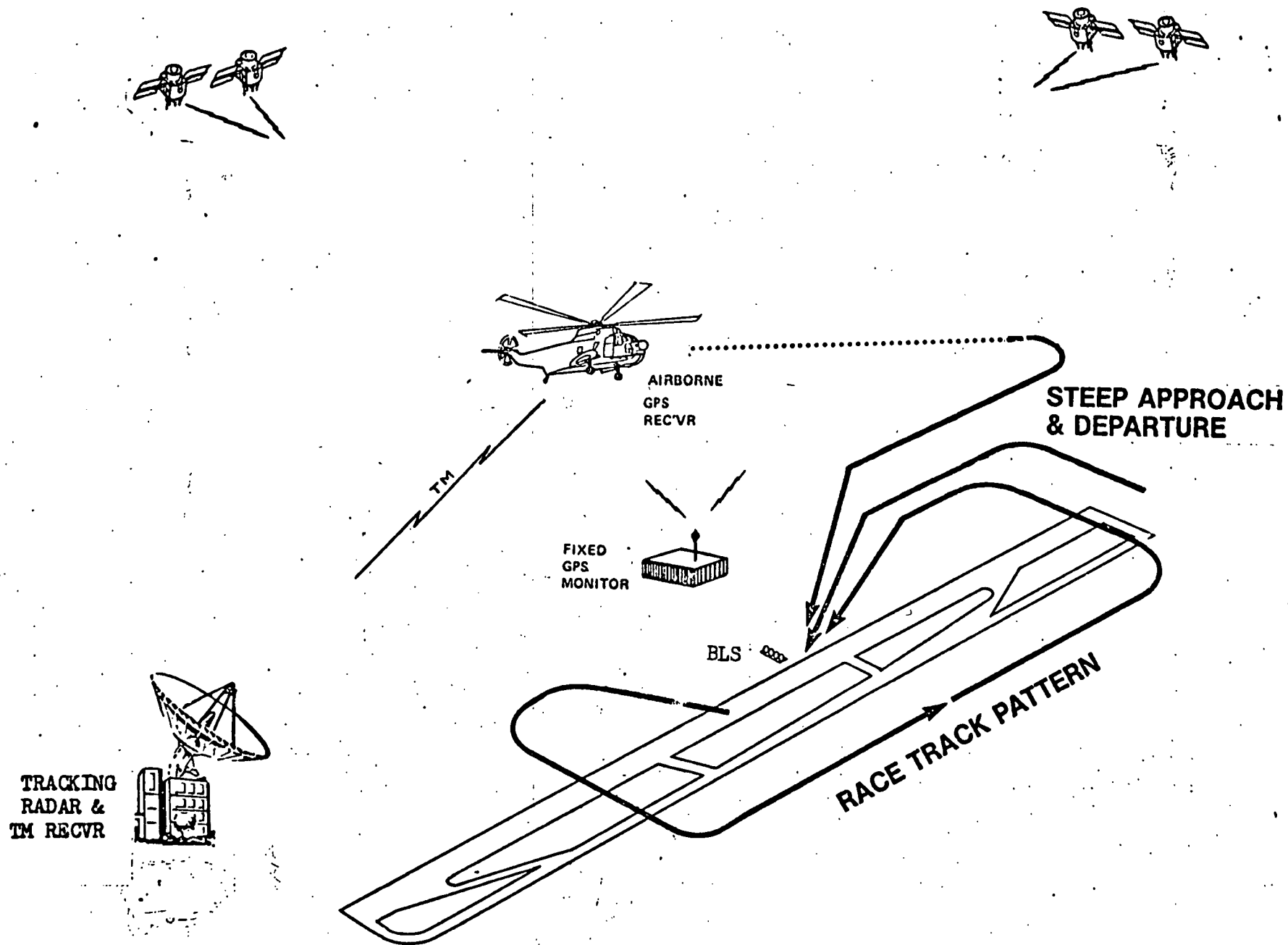


Figure 2

ORIGINAL PAGE IS
OF POOR QUALITY

the telemetry hardware. Two routines are actually collections of subroutines which handle all communications with a specified hardware device. The main routine handles interaction with the operator, prepares requests for transmission to the NIP, and buffers data for the telemetry link.

A Norden DZ-11 device is used to communicate with the NIP. It is a multiplexed eight port serial card. The DZ subroutines handle initialization, data transmission and receiving, and error handling. The transmission subroutine is called by non-time critical portions of the main routine and sends the message without returning to the calling routine until complete. The receiving subroutine on the other hand is called by a time critical portion of the main routine and must operate using interrupts. It uses double buffering to insure that no data is lost due to slow response time by the main routine. The incoming message's checksum's are verified to insure that a complete message is received.

A Norden DR device is used to communicate with the telemetry hardware. This is a 16-bit parallel port with one input port and one output port. The DR subroutines handle initialization of the DR card and output to the card. The output subroutine is called by a time critical part of the main routine and is therefore interrupt driven. It is single buffered and the main routine waits if it gets ahead of the output stream.

The main routine calls all of the initializations subroutines and then presents the operator with a menu of

choices. This menu can be recalled at any time by hitting a key on the keyboard. When the operator makes a choice the routine then prepares the appropriate messages to be sent and calls the subroutines. Several of the selections request repetitive data and once selected the software automatically collects the incoming data and sends it to the telemetry hardware with no further operator intervention. The routine constantly checks a series of flags from the subroutines when not in the menu mode. These flags inform the main routine of the status of the various buffers so that data can be copied at the appropriate time to prevent loss of data. Portions of the collected data are displayed on the operator console so that correct operation of the program can be verified.

The program was specifically written to allow for a wide variety of different configurations to allow for different flight experiments in the future. Only the messages to the NIP and the menu need be changed to request new data. For the initial flights a set of requests were prepared for ephemeris data and another set for range and pseudo-range data. The pseudo-range and range data were collected throughout the flight except when new ephemeris data was being collected.

Ephemeris data was collected just prior to beginning the experiment and whenever there was a satellite changeover during the flight (no more than one changeover). A communications link with the ground GPS station insured that they collected the ephemeris at the same time.

Support for the flight operations involved installing and testing the equipment prior to each flight, operating the

ORIGINAL PAGE IS
OF POOR QUALITY

equipment during the flight and then removal of those portions of the equipment required back in the lab after the flight.

Several flights were made to insure that enough data was collected. Each flight lasted at least an hour and involved approaches and take-offs as well as a racetrack pattern (see figures 3 and 4).

This initial set of flights were finished in early June.

The results appear to be useful and will be published in early 1985.

Appendices

- 1 - checklists for installation and operation of equipment
- 2 - software listings.

DIE GPS SH-3G FLIGHTS

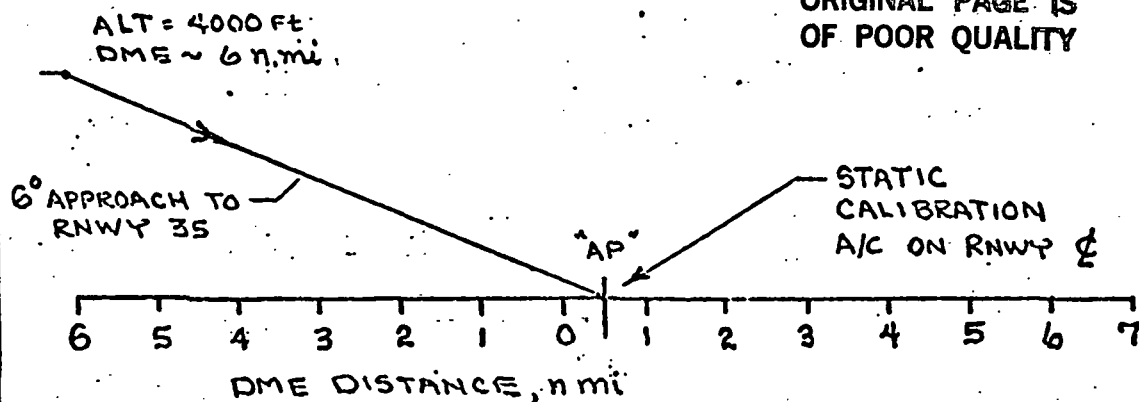
CROWS LANDING OPERATIONS

JUN-21 1984

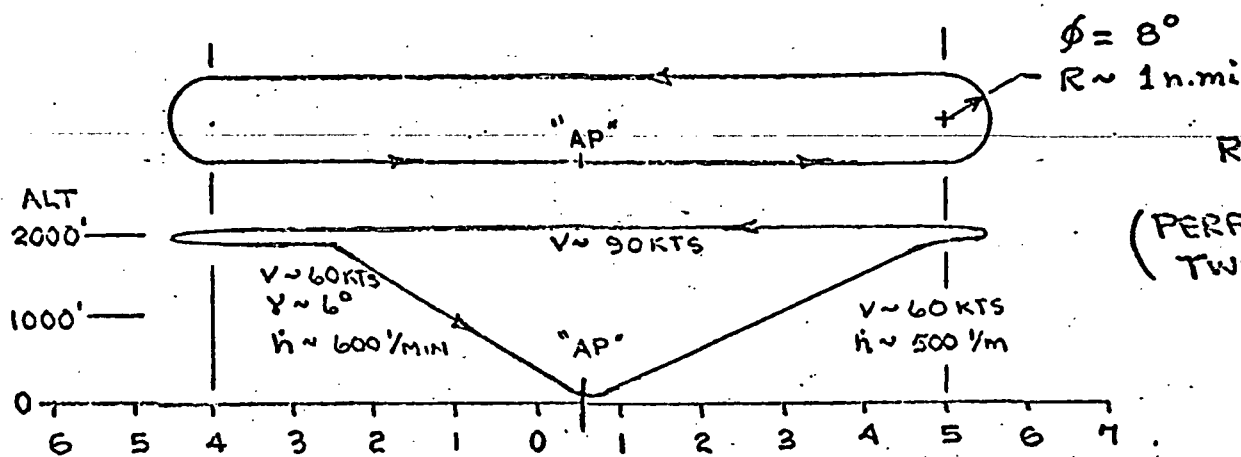
T.O. MOFFETT 16:00

T.O. CROWS 17:15

ORIGINAL PAGE IS
OF POOR QUALITY



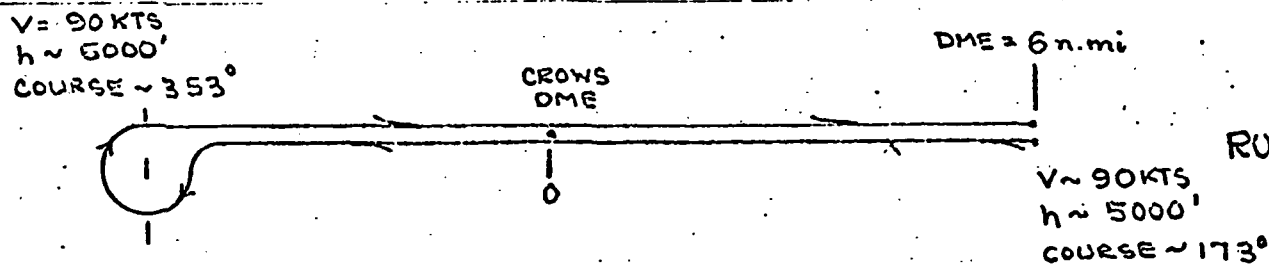
RUN A 17:15
17:30



RUN B

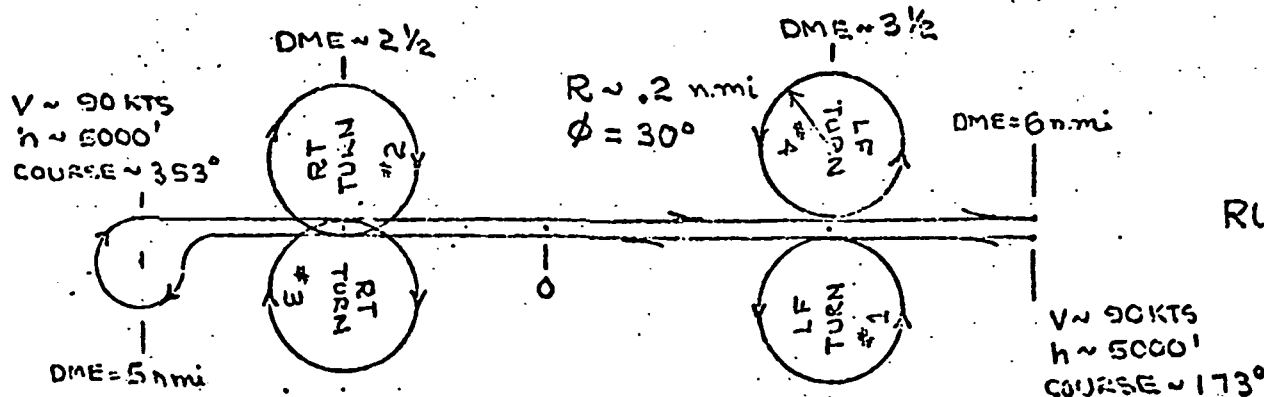
(PERFORM)
TWICE

17:30
18:00



RUN C

18:10
18:20



RUN D

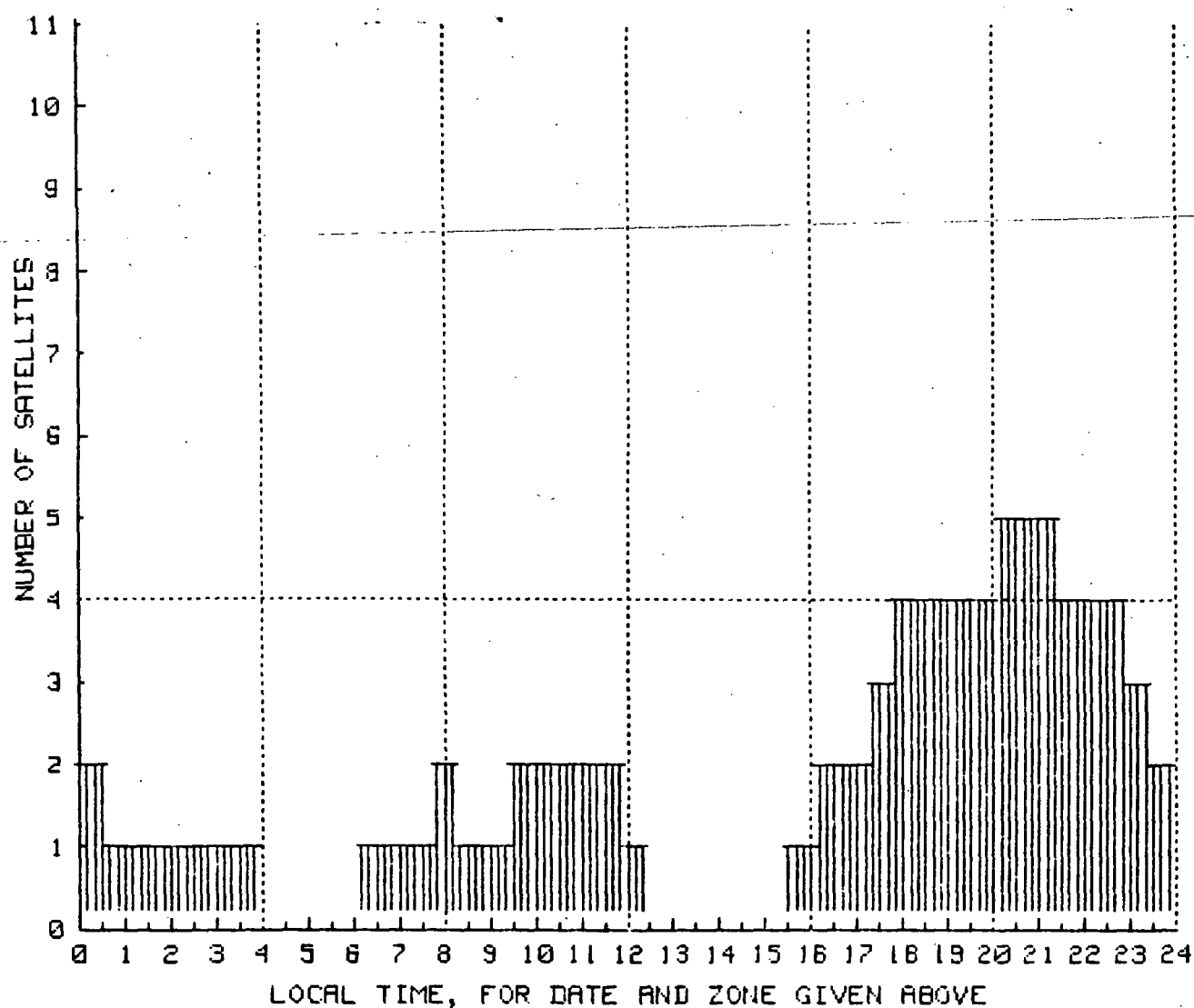
18:20
18:40

RETURN TO MOFFETT 19:00

ORIGINAL PAGE IS
OF POOR QUALITY

ALMANAC AGE IS 5 / 3 / 1984
DATA VALID FOR 5 / 24 / 1984
37.412 DEGREES LATITUDE
-121.1058 DEGREES LONGITUDE
5 METERS ABOVE WGS72 ELLIPSOID
SATELLITES
9 4 11 6 8

SATELLITE AVAILABILITY SUMMARY



GPS SH3 INSTALLATION
EQUIPMENT INSTALLATION

WARNING! INSURE THAT RACK POWER IS OFF.

1. Insure that pre-amp has been installed in tail of SH3 by aircraft maintenance technician.
2. Insure that RMDU is connected properly (coordinate with aircraft manager).
3. Secure HP 2382A terminal to rack using straps and screws (baseplate). Connect DB-25 connector to rear and plug power cord into power strip at rear of rack.
4. Install CDU in cutout. Connect cable marked "CDU" to J-1 on back of CDU.
5. Insure that jumper cable is in place between J-1 of CDU and J-3 of Z-Set (old interface used to connect to this break).
6. Install Norden in full ATR rack. Connect power cable to J-4. Connect cable to terminal to J-1. Connect cable from NIP to J-6. Connect UNIBUS cable to J-5. Verify that all pins are solid on cable to RMDU and connect to J-3 (one pin is held in place by glue).
7. Install Miltope floppies and controller in their rack (if flying) or hook up on top of rack. Connect UNIBUS cable to J-1 of controller. Plug in power cords to power strip at rear of the rack.
8. Install Z-set in right hand 3/4 ATR rack. Connect antenna lead to J-4. Connect power cable to J-2. Connect CDU cable (from CDU J-1) to J-3.
9. Install NIP in left hand 3/4 ATR rack. Connect power cable to J-2. Do not plug in. Connect cable from J-6 of Norden to J-3. Connect thick cable from Z-set to J-1.
10. Have wiring inspected and tied down.

To remove equipment, reverse procedures. Note that all cables that connect to the NIP are also used in the lab.

Tools required: Flat head and phillips screwdrivers and wire cutters (to free cabling from tie-downs).

**GPS SH3 INSTALLATION
PRE-FLIGHT INSTRUCTIONS**

WARNING! - THESE STEPS TO BE ACCOMPLISHED IN SEQUENCE.

1. Insure that power is available to SH3 and that experimenters station power switch (to right of righthand pilot) is on.
2. Insure that all equipment is turned off. Turn brightness switch on CDU fully counter-clockwise (click).
3. Flip two right hand switches up on small orange box at upper left of rack. Verify that green lights come on.
4. Push in CB-18, 21, 22, 19, 7, 8, 10, and 12 (GPS, rack and RMDU power). Push in CB-4 and 5 for AIMS altimeter (if used) and CB-3 if radar tracking will be utilized.
5. Apply power to terminal (push switch in rear).
6. Apply power to disk controller and drives.
7. Push in CB-16 to turn on computer.
8. Turn brightness switch on CDU clockwise to adjust brightness of display.
9. Plug NIP power cord into power strip at the rear of the rack.
10. Pre-flight Z-set per page 5-12 of Z-set user's manual.
11. Insert aircraft floppy disk in DY0:
12. Hold down shift key on terminal and hit break key.
13. Hit caps key.
14. Type L<sp>l71000<cr> followed by S<cr>. System should boot.
15. After system boots type INIT VM:<cr> followed by Y<cr>. The next 6 instructions load virtual memory with a minimal system and the program.
16. Type COPY<cr> *.SYV<cr> VM:*.SYS<cr>.
17. Type COPY<cr> RT11VM.SYS<cr> VM:RT11SJ.SYS<cr>.
18. Type COPY<cr> SWAP.SYS,NORINT.SAV<cr> VM:<cr>.
19. Type COPY<cr> STARTV.COM<cr> VM:STARTS.COM<cr>.
20. Type COPY/BOOT<sp>VM:RT11SJ<sp>VM:<cr>.
21. Type BOOT<sp>VM:<cr>. What boots is a minimal system running on a virtual disk (in core memory). The only usable

command is R (not RU or RUN).

22. Type R<sp>NORINT<cr> to verify program operation.

23. When NORINT comes up type <sp> to begin.

24. Type 2<cr> to send a test pattern to the ground station. Verify with the ground station that they are receiving the pattern.

25. Type <sp> to get menu back. Type 6<cr> to exit program.

26. Verify that the almanac has been gathered successfully. Page 4-10 of Z-set User's Manual.

27. Remove the aircraft disk.

28. Pull CB-16 out. Computer must be turned off before peripherals to prevent core memory from being trashed.

29. Turn off the floppy drives and the terminal.

30. Unplug the NIP.

31. Turn brightness knob on CDU fully counter-clockwise (click).

32. Pull CB-12, 10, 8, 7, 18, 19, 21, and 22. Pull any other circuit breakers still in.

33. Flip switches at orange box down.

34. Remove UNIBUS cable from front of Norden. Take UNIBUS terminator off of the floppy drive controller and screw into J-5 on Norden.

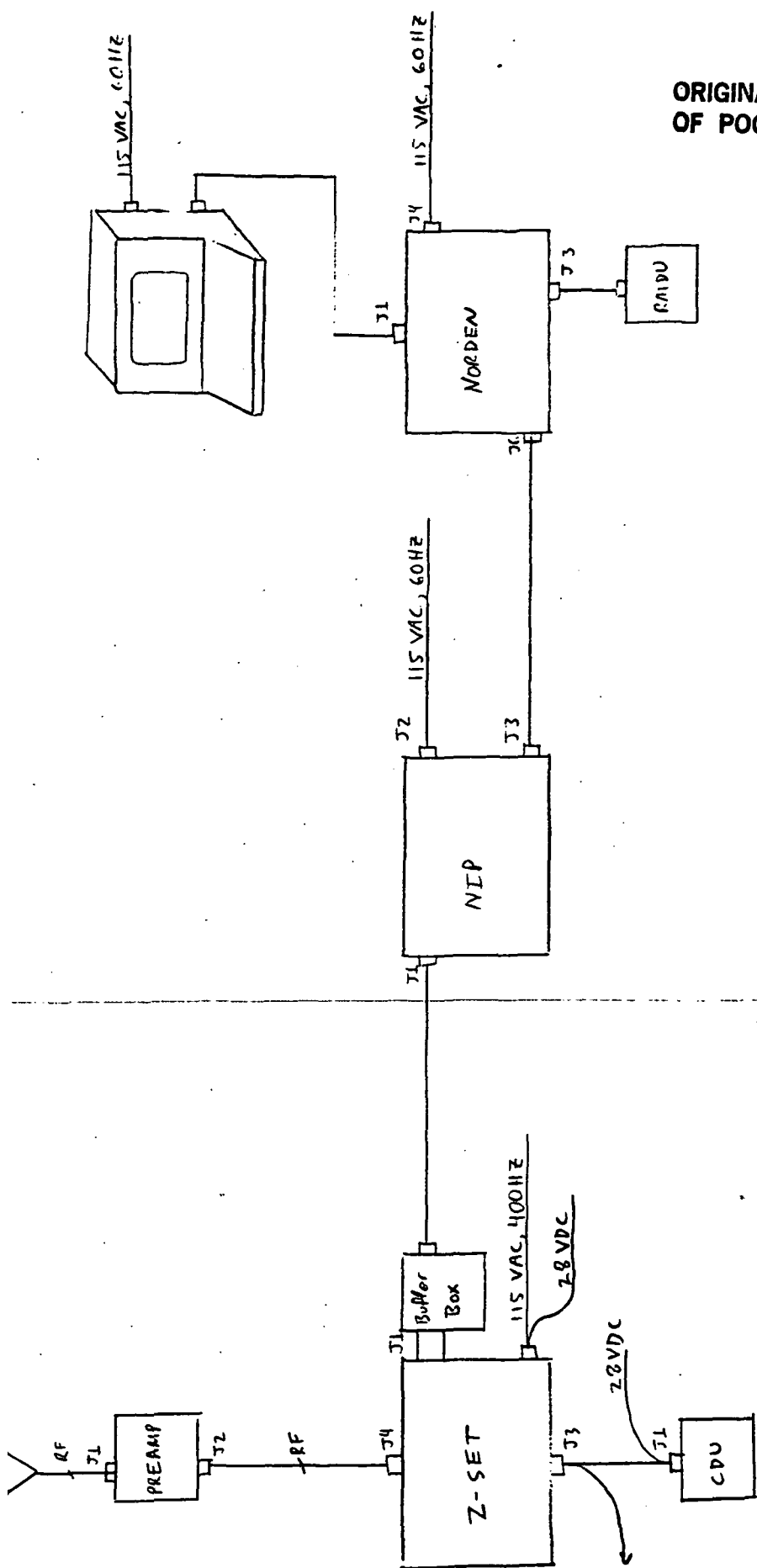
35. Remove floppy drive from aircraft if required.

**GPS SH3 INSTALLATION
FLIGHT OPERATIONS**

WARNING! - THESE STEPS TO BE ACCOMPLISHED IN SEQUENCE.

1. Wait until engines have started and pilots give go ahead for removing seatbelts.
2. Flip two right hand switches up on small orange box at upper left of rack. Verify that green lights come on.
3. Push in CB-18, 21, 22, 19, 7, 8, 10, and 12. Push in CB-4 and 5 for AIMS altimeter (if used) and CB-3 if radar tracking will be utilized.
4. Plug NIP power cord into power strip at rear of the rack.
5. Apply power to terminal (push switch in rear).
6. Push in CB-16.
7. Turn brightness switch on CDU clockwise to adjust brightness of display.
8. See page 5-12 of Z-set user's manual to prepare for take off.
9. Hold down shift key on terminal and hit break key.
10. Hit caps key.
11. Type L<sp>0<cr> followed by S<cr> to boot system in memory.
12. After system boots type R<sp>NORINT<cr>.
13. When NORINT comes up, type <sp> to begin.
14. Type l<cr><sp>l<cr> to get accurate status of Z-set.
15. Verify that status of Z-set is healthy and that the set is navigating (see Z-set user's manual page 5-12).
16. Type <sp>5<cr> to begin data collection.
17. Next three instructions gather ephemeris. Type <sp>3<cr>, wait for 5 seconds, then type <sp>4<cr>. Block 34 should be collected immediately.
18. Turn selector switch on CDU to EPE position, push 5A key, wait until A appears in upper left on display, then push CLR key. "A" should be cleared from display. Z-set re-gathers ephemeris.
19. Wait for all blocks to be collected. Completion indicated when Z-set is again navigating on four satellites.
20. Type <sp>5<cr> to restart data collection.

ORIGINAL PAGE IS
OF POOR QUALITY



Differential GPS Airborne Equipment

**GPS SH3 INSTALLATION
SHUTDOWN INSTRUCTIONS**

WARNING! - THESE STEPS TO BE ACCOMPLISHED IN SEQUENCE.

1. Type (sp)6(cr) to exit program.
2. Pull CB-16 out.
3. Turn off the terminal.
4. Unplug the NIP.
5. Turn knob on CDU fully counter-clockwise (click).
6. Pull CB-12, 10, 8, 7, 18, 19, 21, and 22. Pull any other circuit breakers still in.
7. Flip switches at orange box down.
8. See equipment installation instructions for SH3 for removal of equipment.

**GPS LAB INSTALLATION
EQUIPMENT INSTALLATION**

1. Insure that all breakers are out on power distribution box.
2. Connect CDU (J-1) to merged cable coming from J-3 of Z-Set.
3. Connect AIMS altimeter (if used) to cables coming from J-3 of Z-Set and power distribution box.
4. Connect cable from CDU to J-3 of Z-Set.
5. Connect J-2 of Z-Set to cable coming from power distribution box.
6. Connect J-1 of Z-Set to buffer box. To simplify installation, thick cable leading to NIP should already be connected to buffer box. Connect cable to J-1 of NIP.
7. Connect antenna lead to J-4 of Z-Set (marked as GPS #1).
8. Connect ground wire from 28 volt power supply to one of the lower righthand screws on the face of the Z-Set.
9. Insure that the 28 volt leads from the power distribution box are connected to the 28 volt power supply. The two ground wires which parallel the antenna wires from the crawl space should be connected to the 28 volt power supply.
10. Loran antenna cable is being used to power preamp on roof and should be connected to 15 volt power supply. Antenna has dual banana plug attached. Notch on plug is ground side.
11. Connect cable (bring over from aircraft) between J-3 of NIP and J-6 of Norden.
12. Connect console cable from breakout panel on rack to J-1 of Norden. Connect terminal to panel (J1B). Cable should have wires 2 & 3 straight through (verify with ohm meter).
13. Connect UNIBUS cable to MILTOPE floppies or extender box (See separate instructions for extender box) and J-5 of Norden.
14. Connect power cable to J-2 on NIP. Do not plug into power.

GPS LAB INSTALLATION
POWER ON

WARNING! - THESE STEPS TO BE ACCOMPLISHED IN SEQUENCE.

1. Fill out GPS log.
2. 115 VAC, 400HZ power cord connected to power distribution box.
3. Power cords to disk drives, terminal, and power supplies plugged in (60HZ).
4. GPS brightness control on CDU full counter-clockwise (click).
5. CB-1 on power distribution box in. Then CB-2 in.
6. CB-3 (AIMS PUR), and CB-4 (AIMS VIB.) in if altimeter used.
7. Turn on power to antenna pre-amp. Verify approx. 0.5 amp current and 15 volts.
8. Turn on +28 volt power supply to CDU (two switches).
WARNING! Do not touch the 28 VDC rheostat adjustment. It is preset on 80 for proper operation.
9. Turn GPS brightness control on CDU clockwise to adjust display brightness.
10. Plug in NIP to 115 VAC, 60HZ.
11. Follow GPS operations checklist for proper system operations.
12. Apply power to terminal.
13. Apply power to disk drives.
14. Apply power to Norden computer.
15. Follow software operation instructions.

GPS LAB INSTALLATION
POWER OFF

WARNING! - THESE STEPS TO BE ACCOMPLISHED IN SEQUENCE.

1. GPS brightness switch full counter-clockwise (click).
2. CB-1 out.
3. CB-2, CB-3, CB-4 out.
4. +28 VDC output power switch (right) off.
5. +28 VDC power switch (left) off.
6. Turn off power to antenna pre-amp.
7. Exit any software running on Norden.
8. Unplug NIP.
9. Turn off Norden.
10. Turn off disk drives.
11. Turn off terminal.

**EXTENDER BOX
CONNECTING TO NORDEN**

1. Unload heads of RL02s (load switches out).
2. Turn power switch on commercial PDP 11/34A to OFF.
3. Pull extender box drawer out and open top.
4. Remove UNIBUS cable leading to SIGMA 9 interface (left rear cable). Note orientation of connector.
5. Remove terminator card (card in back with purple top) from SIGMA 9 interface and place in slot emptied by UNIBUS cable in extender box.
6. If additional memory board is needed (for RSX-11M), verify proper memory address using attached sheet (NORDEN has 64K words internally). Place memory board between main UNIBUS cable and RX211 card after removing grant jumper.
7. Close extender box and push it back into place.
8. Connect UNIBUS cable from NORDEN to connector behind extender box.
9. Pull top off of commercial computer and remove UNIBUS cable.
NOTE LOCATION OF NOTCH!
10. Insert UNIBUS cable into slot next to where the cable from the NORDEN screws into extender box. Note that the tab which prevents the cable from being inserted backwards is broken. Insure that notch is over broken portion.
11. Apply power to NORDEN and terminal.
12. Apply power to disks and extender box by flipping silver switch above and right of RX02s up.
13. Push load buttons on RL02s in.

Reverse procedure to reinstall commercial computer. Remember grant jumper after removing memory.

ORIGINAL PAGE IS
OF POOR QUALITY

0001 PROGRAM NCPRINT

INTERFACE SOFTWARE BETWEEN NIP AND SMDU TO RUN ON NORDEN

CALLS TO: DZINIT, DZOUT, DZSTOP, IAND, INITRD, NIPSTA, INITRD,
RXINT, PRINIT, PRINT, DROUT, DPSTOP

CALLS FROM: NONE

PARAMETERS: NONE

COMMONS: NONE

ENVIRONMENT: NORDEN POP-11/34A RT-11 FORTRAN IV

WRITTEN BY: ARTHUR A. BECKMAN, UNIV. OF SO. COLORADO, 1984

RESTRICTIONS: NONE

REFERENCES: USER IS EXPECTED TO BE FAMILIAR WITH THE Z-SET
AND NIP USER MANUALS.

NOTE: NORMAL COMPILATION RESULTS IN A VERSION WHICH WORKS
WITHOUT THE DR CARD OUTPUT. TO PRODUCE THE AIRBORNE
VERSION USE THE DEBUG SWITCH TO INCLUDE STATEMENTS
WITH D IN FRONT OF THEM.

MODIFICATIONS:

DATA CONSTANTS

INTEGER * 2 MSLLOC ILOCATION OF LENGTH WORD IN MESSAGE ARRAY

INTEGER * 2 RSTLOC ILOCATION OF RESPONSE MESSAGE STATUS BYTE

INTEGER * 2 RSPLOC ILOCATION OF RESPONSE TYPE BYTE

DATA MSLLOC, RSTLOC, RSPLOC/2,5,2/

INTEGER * 2 OUTTYP(16) IARRAY TO MATCH RESPONSE TYPE TO OUTPUT

DATA OUTTYP/1,7,2,5,3,7,4,4,4,2,4,4,4,4,6,2,2/


```

C RST RESTARTS THE NIP (WARM START)
00033 DATA RSTMSG/"252,10,5,"107,0/
C ENQ SENDS A REPETITIVE REQUEST FOR STATUS
00034 DATA ENQMSG/"252,5,6,"113,R","256/
C 832 REQUESTS THAT BLOCK 32 BE SENT REPETITIVELY
00035 DATA B32MSG/"252,4,7,"113,32,R","216/
C CAN CANCELS ALL REPETITIVE MESSAGE REQUESTS IN NIP
00036 DATA CANMSG/"252,2,6,"115,"377,1/

C DATA VARIABLES
C
00037 LOGICAL # 1 BUFFB(500),BUFFA(500) !BYTE MESSAGE BUFFERS
00038 LOGICAL # 1 OUTBU1(500) !BYTE OUTPUT BUFFER
00039 INTEGER # 2 BUFWA(250),BUFWB(250) !WORD MESSAGE BUFFERS
00040 INTEGER # 2 OUTBUW(250) !WORD OUTPUT BUFFER
00041 EQUIVALENCE (BUFFB,BUFWB),(BUFFA,BUFWA)
00042 EQUIVALENCE (OUTBU1,OUTBUW)
00043 INTEGER # 2 ERRSAV,ERR04,ERR05 !ERROR FLAGS FOR BUFFER A AND B
                                AND TEMPORARY STORAGE
                                !REQUEST TYPE VARIABLE
                                !BUFFER FILLED FLAGS
                                !FLAG TO CONTROL COMMAND ENTRY
                                !FLAG TO CONTROL OUTER LOOP
                                !FLAG TO INDICATE DR COMPLETION STATUS
                                !LENGTH OF MESSAGE VARIABLES
                                !TEMPORARY VARIABLES
                                !TEMPORARY VARIABLES ARE FOR
                                !TIME (TOO), LATITUDE AND LONGITUDE
                                !LAT8,LON8), ALTITUDE (ALT4), SPEED (SPD4), ESTIMATED
                                !POSITION-ERROR (EPE4), GROUND TRACK (TRK4), AND LOGICAL
                                !SATELLITE IN USE (LSAT).
C OTHER VARIABLES AND EQUIVALENCE STATEMENTS SERVE TO LOAD THE
C DATA FROM THE MESSAGES INTO THE MAIN VARIABLES.
00052 INTEGER # 2 TIME(2),TIME1,TIME2
00053 REAL # 4 TOO
00054 INTEGER # 2 LSAT
00055 INTEGER # 2 DAY,HOUR,MIN,SEC
00056 INTEGER # 2 LAT(4),LAT1,LAT2,LAT3,LAT4,ALT1,ALT2,TRK1,TRK2
00057 INTEGER # 2 LON(4),LON1,LON2,LON3,LON4,SPD1,SPD2,EPE1,EPE2
00058 INTEGER # 2 ALT(2),TRK(2),SPD(2),EPE(2)

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

0059 REAL # 8 LAT8,LON8
0060 REAL # 4 ALT4,SPD4,TRK4,EPE4
0061 INTEGER # 2 LATD,LATM,LATS,LOND,LONM,LONS
0062 EQUIVALENCE (TIME1,TIME(1),TOD),(TIME2,TIME(2))
0063 EQUIVALENCE (LAT1,LAT(1),LAT8),(LAT2,LAT(2)),(LAT3,LAT(3))
0064 EQUIVALENCE (LAT4,LAT(4)),(LON1,LON(1),LON8),(LON2,LON(2))
0065 EQUIVALENCE (LON3,LON(3)),(LON4,LON(4))
0066 EQUIVALENCE (ALT1,ALT(1),ALT4),(ALT2,ALT(2))
0067 EQUIVALENCE (TRK1,TRK(1),TRK4),(TRK2,TRK(2))
0068 EQUIVALENCE (SPD1,SPD(1),SPD4),(SPD2,SPD(2))
0069 EQUIVALENCE (EPE1,EPE(1),EPE4),(EPE2,EPE(2))

```

C PROGRAM START

```

0070 TYPE #,
0071 CALL DZINIT
0072 CALL DRINIT
0073 CALL NIPSTA
0074 CALL IPOKE ("44,"50100.0R,IPEEK("44))
0075 FLAG = 1
0076 MORE = .TRUE.
0077 CALL DZOUT (RSTMSG,RSTMSL)
0078 TYPE #,"A RESTART REQUEST HAS JUST BEEN SENT TO THE NIP."
0079 CALL ISLEEP (0.0,1.0)
0080 CALL INITRD(BUFFA(1),DONEA,ERRDA,BUFFB(1),DONEB,ERRDB)
0081 TYPE #,"HIT ANY KEY TO BEGIN."
0082 DO WHILE (MORE)
0083 IF(.NOT.(MORE))GO TO 220
0084 IF (DONEA)
0085 IF(.NOT.(DONEA))GO TO 130
0086 THEN
0087 BUFFER A IS FILLED AND NEEDS TO BE COPIED
0088 ERRSAB=ERRDA
0089 DLENB = BUFWA(M3LLOC)
0090 IF (FLAG.LE.0) GO TO 110
0091 DO 120 I = 1,DLENB
0092 OUTRUB(I) = BUFFA(1)
0093 CONTINUE
0094 DONEA = 0
0095 ERRDA = 0
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835
0836
0837
0838
0839
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0860
0861
0862
0863
0864
0865
0866
0867
0868
0869
0870
0871
0872
0873
0874
0875
0876
0877
0878
0879
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0950
0951
0952
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
0972
0973
0974
0975
0976
0977
0978
0979
0980
0981
0982
0983
0984
0985
0986
0987
0988
0989
0990
0991
0992
0993
0994
0995
0996
0997
0998
0999
1000

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

FORTRAN-IV      V02.6      Mon 24-Sep-84 13:42:28      PAGE 005

0092      GO TO 500
0093      GO TO 160
      ELSE
0094      IF (DONEB)
      IF (.NOT.(DONEB)) GO TO 170
      THEN
0096      BUFFER 0 IS FILLED AND NEEDS TO BE COPIED
      PRESERVE SPACE
0097      CLEAR = BUFFER(CHLLOC)
      IF (FLAG.LS.0) GO TO 140
      GO TO 150
0099      OUTBUF(1)=BUFFER(1)
      CONTINUE
0100      DONEB = 0
0101      ERRORB = 0
0102      GO TO 500
0103      GO TO 210
0104      ELSE
      NO MESSAGE AWAITING ATTENTION, CHECK KEYBOARD INPUT
      IF (ITTNR()) .GE. 0)
      IF (.NOT.(ITTNR().GE.0)) GO TO 210
      THEN
0107      READ = 1
      REPEAT
0108      TYPE 1000,
0109      FORMAT (' RECOGNIZED COMMANDS: ',/,
      1) ENQUIRE',/,
      2) TEST PATTERN',/,
      3) CANCEL REQUESTS',/,
      4) SPHERIS COLLECTION',/,
      5) BLOCKS 32 & 34',/,
      6) EXIT PROGRAM',/,
      7) RESTART NIP')
      CALL GTLIN(LINE,LPRM)
      DECODE(1,1010,LINE,ERR=180) REQ
      FORMAT (I2)
      CASE REQ OF
0113      GO TO (300,310,320,330,340,350,360),REQ
0114      GO TO 370
      DONEB:

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0115 300      CALL DZOUT (ENQMSG,ENQMSL)
0116          READ = 0
0117          GO TO 380
C
0118 310      DOIST:
0119          CALL DZOUT(84MSG,B4MSL)
0120          READ = 0
0121          GO TO 380
C
0121 320      DOCAN:
0122          CALL DZOUT (CANMSG,CANMSL)
0123          READ = 0
0124          GO TO 380
C
0124 330      DOEPH:
0125          CALL DZOUT(DFB34M,DFB34L)
0126          CALL DZOUT(86MSG,86MSL)
0127          CALL DZOUT(87MSG,87MSL)
0128          CALL DZOUT(88MSG,88MSL)
0129          CALL DZOUT(89MSG,89MSL)
0130          CALL DZOUT(834MSG,834MSL)
0131          READ = 0
0132          GO TO 380
C
0132 340      DOBLO:
0133          CALL DZOUT(DFB33M,DFB33L)
0134          CALL DZOUT(322MSG,832MSL)
0135          CALL DZOUT(333MSG,833MSL)
0136          READ = 0
0137          GO TO 380
C
0137 350      DOEXE:
0138          MORE=.FALSE.
0139          READ = 0
0140          GO TO 380
C
0140 360      DORES:
0141          CALL DZSTOP
0142          CALL NIPSTA
0143          CALL DZOUT (RSTMSG,RSTMSL)
0144          TYPE #,'RESTART SENT, WAITING....'
0145          CALL ISLEEP (0,0,1,0)
0146          CALL DZINIT
0147          CALL INITRD(BUFFA(1),DONEA,ERROA,
0148          BUFFB(1),DONEB,ERROB)
0149          TYPE #,'HIT ANY KEY TO BEGIN'
```

ORIGINAL PAGE IS
OF POOR QUALITY

0148 READ = 0

0149 GO TO 390

ELSE:

0150 GO TO 390

ENDCASE

UNTIL READ = 0

IF (READ.EQ.0) GO TO 210

GO TO 180

ENDIF

ENDIF

ENDIF

ENDWHILE

GO TO 100

0154 210 RXENAB=.FALSE.

0155 220 CALL DZSTOP

0156 CALL DOSTOP

CALL EXIT

0157

----- SUBROUTINE OUTPUT (CALLED BY GO TO STATEMENT TO SAVE TIME)

OUTPUTS DATA TO CRT AND TELEMETRY. TYPES ERROR MESSAGES TO CRT.

CALLS TO: OROUT

CALLS FROM: NOPRINT

PARAMETERS: NONE

COMMONS: NONE

ENVIRONMENT: NORDEN POP 11/34A RT-11 FORTRAN

WRITTEN BY: ARTHUR A. BECKMAN, UNIV. OF SO. COLORADO, 1984

RESTRICTIONS: JUMPED TO BY GO TO STATEMENT TO SAVE TIME.

0158 500 IF (.NOT.(CEPRSAV.GT.0)) GO TO 550

THEN

ERROR PRESENT, IDENTIFY AND HANDLE.

ORIGINAL PAGE IS
OF POOR QUALITY

EVEN THOUGH THE ERROR FLAG IS SET THE BUFFER IS RETURNED ONLY WHEN

C FINALLY FILLED WITH GOOD DATA.

C IF (.NOT.(ERRSAV.EQ.1)) GO TO 510

C THEN

C TYPE *, 'SYNC'

C GO TO 550

C ELSE

C IF (.NOT.(ERRSAV.EQ.2)) GO TO 520

C THEN

C TYPE *, 'HEADER'

C GO TO 550

C ELSE

C IF (.NOT.(ERRSAV.EQ.4)) GO TO 530

C THEN

C TYPE *, 'TRAILER'

C GO TO 550

C ELSE

C IF (.NOT.(ERRSAV.EQ.8)) GO TO 540

C THEN

C TYPE *, 'COLLIS'

C GO TO 550

C ELSE

C TYPE *, 'MULT FR'

C ENDOF

C 0177 550 RSPNSE=OUTSUB(RSPLOC)

C 0178 550 OLENH=OLENB/2

C IF (LAND(OLENB,1).EQ.1)

C IF (.NOT.(LAND(OLENB,1).EQ.1)) GO TO 560

C THEN

C 0181 550 OUTSUB(OLENB+1)=0

C 0182 550 OLENH=OLENH+1

C ENDOF

C IF ((RSPNSE.GE.1).AND.(RSPNSE.LE.16))

C 0183 560 IF (.NOT.(RSPNSE.GE.1).AND.(RSPNSE.LE.16)) GO TO 870

C THEN

C 0185 560 TYPE 2000, OUTSUB(RSTLOC),

C (CMD(I,RSPNSE),I=1,CMDL(RSPNSE))

C 0186 2000 FORMAT('STATUS:',04,2X,3A,':,A)

C 0187 560 J=OUTTYP(RSPNSE)

C CASE J OF

ORIGINAL PAGE IS
OF POOR QUALITY

```
0188      GO TO (700,710,770,780,790,830,850),J
0189      GO TO 860
      CCK:
0190      TYPE 2010,(OUTBU8(6+I),I=1,OLENB-7)
0191      FORMAT(' ',80A)
0192      GO TO 360
      C
      ACK,CON,NAK,RST:
0193      RSPNSE=OUTBUW(4)
      IF (RSPNSE.LE.10)
0194      IF(.NOT.(RSPNSE.LE.10)) GO TO 750
      THEN
0196      TYPE 2020,(CMD(I,RSPNSE),I=1,CMDL(RSPNSE))
0197      FORMAT(' ',3A,,:,A)
      IF (COLENB.GT.9)
0198      IF(.NOT.(COLENB.GT.9))GO TO 730
      THEN
      IF (RSPNSE.EQ.10)
0200      IF(.NOT.(RSPNSE.EQ.10)) GO TO 720
      THEN
      TYPE 2030,OUTBUW(5)
0203      FORMAT(' ',6)
0204      GO TO 730
      ELSE
0205      TYPE 2040,OUTBUW(5)
0206      FORMAT(' ',12)
      ENDIF
      ENQ:
0207      GO TO 760
      ELSE
0208      TYPE *,'???'
      ENDIF
      TYPE #,'
0209      TYPE #,'
0210      GO TO 360
      ENQ:
0211      TYPE 2050,OUTBUW(4),OUTBUW(5),(OUTBU8(10+I),I=1,15)
0212      FORMAT(X,/,2sts:',07,10X,'Nipsts:',07,/,
      'Evmsk:',5(X,03),/, 'Evmskr:',5(X,03),/,
      'Setid:',12,3X,'Setvar:',13,.,',12,3X,
      'Nipver:',13,.,',12,/)
      GO TO 360
0213
```


Mon 24-Sep-84 16:42:28

V02.6

FORTRAN IV

```

C
0214 780 REZ0,REZ1,RIP,SEZ0,SEZ1,SIP:
      TYPE 2060,(OUTBUW(3+I),I=1,DLENW-4)
0215 2060 FORMAT(' ',06,' ',/,S(X,8(06,:',','),/))
0216 GO TO 360
      DAT:
0217 790 CONTINUE
      CALL OROUT (OUTSUB(1),FLAG)
      IF (OUTBUW(4).EQ.32)
        IF (.NOT.(OUTBUW(4).EQ.32)) GO TO 810
      THEN
        TIME1=OUTBUW(9)
        TIME2=OUTBUW(10)
        TOD=TOD+OUTBUW(11)/10.
        DAY=TOD/86400.
        TOD=TOD-(DAY*86400.)
        DAY=DAY+1
        HOUR=TOD/3500
        TOD=TOD-(HOUR*3500)
        MIN=TOD/60
        SEC=TOD-(MIN*60)
        DO 800 I=1,4
          LAT(I)=OUTBUW(13+I)
          LON(I)=OUTBUW(17+I)
        CONTINUE
        LATD=LAT8
        LAT8=ABS(60.*(LAT8-LATD))
        LATM=LAT3
        LAT5=60.*(LAT8-LATM)
        LONQ=LON2
        LON8=ABS(60.*(LON8-LONQ))
        LONM=LON8
        LONS=60.*(LON8-LONM)
        ALT1=OUTBUW(22)
        ALT2=OUTBUW(23)
        ALT4=3.2808*ALT4
        SPD1=OUTBUW(24)
        SPD2=OUTBUW(25)
        SPD4=3.6*SPD4
        TRK1=OUTBUW(26)
        TRK2=OUTBUW(27)

```

ORIGINAL PAGE IS
OF POOR QUALITY

FORTRAN IV

V02.6

Mon 24-Sep-84 16:42:28

PAGE 012

0236

GO TO 860

-CAN, UNDEFINED:

0237

GO TO 860

-ELSE:

GO TO 860

END CASE

0238

GO TO 880

-ELSE:

TYPE #, 'ILLEGAL RESPONSE CODE'

ENDIF

0240

GO TO 160

0241

END

ORIGINAL PAGE IS
OF POOR QUALITY

FORTRAN IV Storage Map for Program Unit NGRINT

Local Variables, .PSECT \$DATA, Size = 003630 (972. words)

Name	Type	Offset	Eqv	Name	Type	Offset	Eqv
ALT1	I#2	003444	Eqv	ALT2	I#2	003446	Eqv
B32MSL	I#2	003476		B33MSL	I#2	003504	
B4MSL	I#2	003520		B6MSL	I#2	003510	
B8MSL	I#2	003514		B9MSL	I#2	003516	
DAY	I#2	003566		DFE33L	I#2	003500	
DONEA	L#1	003544		DONEB	L#1	003545	
EPE1	I#2	003460	Eqv	EPE2	I#2	003462	Eqv
ERR0A	I#2	003536		ERR0B	I#2	003540	
FLAG	L#1	003550		HOUR	I#2	003570	
J	I#2	003562		LATD	I#2	003576	
LATS	I#2	003602		LAT1	I#2	003424	Eqv
LAT3	I#2	003430	Eqv	LAT4	I#2	003432	Eqv
LOND	I#2	003604		LONM	I#2	003606	
LON1	I#2	003434	Eqv	LON2	I#2	003436	Eqv
LON4	I#2	003442	Eqv	LON8	R#8	003434	Eqv
MIN	I#2	003572		MORE	L#1	003547	
OLENB	I#2	003552		OLENW	I#2	003554	
REQ	I#2	003542		RSPL0C	I#2	003470	
RSTLOG	I#2	003466		RSTMSL	I#2	003472	
SEC	I#2	003574		SPD1	I#2	003454	Eqv
SPD4	R#4	003454	Eqv	TIME1	I#2	003420	Eqv
TOD	R#4	003420	Eqv	TRK1	I#2	003450	Eqv
TRK4	R#4	003450	Eqv				

Local and COMMON Arrays:

Name	Type	Section	Offset	Size	Dimensions
ALT	I#2	\$DATA	003444	000004	(2)
BUFFA	L#1	\$DATA	001450	000764	(250)
BUFFB	L#1	\$DATA	000464	000764	(250)
BUFFA	I#2	\$DATA	001450	000764	(250)
BUFFB	I#2	\$DATA	000464	000764	(250)
B32MSG	L#1	\$DATA	000337	000007	(4)
B33MSG	L#1	\$DATA	000410	000007	(4)
B34MSG	L#1	\$DATA	000417	000006	(3)
B4MSG	L#1	\$DATA	000425	000007	(4)
B6MSG	L#1	\$DATA	003425	000006	(3)

ORIGINAL PAGE IS
OF POOR QUALITY

FORTRAN-IV Storage Map for Program Unit NPRINT

B7MSG-L#1	\$DATA	000433	000004	(3.)	(6)
B8MSG-L#1	\$DATA	000441	000005	(3.)	(6)
B9MSG-L#1	\$DATA	000447	000006	(3.)	(6)
CANMSG-L#1	\$DATA	000346	000006	(3.)	(6)
CMD-L#1	\$DATA	000100	000100	(32.)	(4,16)
CMDL-I#2	\$DATA	000040	000040	(16.)	(16)
CMR-R#4	\$DATA	000100	000100	(32.)	(16)
DF833M-L#1	\$DATA	000372	000016	(7.)	(14)
DF834M-L#1	\$DATA	000354	000016	(7.)	(14)
ENQMSG-L#1	\$DATA	000331	000006	(3.)	(6)
EPE-I#2	\$DATA	003460	000004	(2.)	(2)
LAT-I#2	\$DATA	003424	000010	(4.)	(4)
LINE-L#1	\$DATA	000204	000100	(4.)	(8)
LON-I#2	\$DATA	003432	000010	(4.)	(4)
LPRUM-L#1	\$DATA	000200	000004	(2.)	(4)
OUT808-L#1	\$DATA	002434	000704	(250.)	(510)
OUT80W-I#2	\$DATA	002434	000704	(250.)	(250)
OUTTP-I#2	\$DATA	000000	000040	(16.)	(16)
RSTMSG-L#1	\$DATA	000324	000006	(3.)	(5)
SPD-I#2	\$DATA	003454	000004	(2.)	(2)
TIME-I#2	\$DATA	003420	000004	(2.)	(2)
TRK-I#2	\$DATA	003450	000004	(2.)	(2)

ORIGINAL PAGE IS
OF POOR QUALITY

Subroutines, Functions, Statement and Processor-Defined Functions:

Name	Type	Name	Type	Name	Type
DZINIL-R#4	DZOUT	R#4	DZSTOP	EXIT	R#4
IAND-I#2	INITRO	I#2	IPEEK	IPOKE	I#2
ITINR-I#2	NIPSTA	I#2		ISLEEP	I#2

TITLE DZGPS DZ CARD INTERFACE SOFTWARE
IDENT /01/

AUTHOR: AAB ON 5/30/74

CHANGES:

MODULE FUNCTION
THESE ROUTINES HANDLE ALL COMMUNICATIONS WITH THE DZ-11 BOARD.
DZINIT INITIALIZES THE DZ AND SETS UP THE INTERRUPT VECTOR.
INITRD SETS UP PARAMETERS FOR DZ READ OPERATIONS.
RXINT IS THE INTERRUPT DRIVEN INPUT ROUTINE ENABLED BY INITRD.
DZOUT HANDLES OUTPUT TO THE DZ. DZSTOP TURNS OFF RXINT.
IAND LOGICALLY ANDS TWO VALUES. NIPSTA PUTS NIP INTO DDT AND THEN
CAUSES A COLD START (JUMP TO ADDRESS ZERO).

THIS CODE IS POSITION INDEPENDENT AND WILL RUN UNDER THE FB MONITOR.

PAGE
NLIST SEX
BREAK PAGE FOR PREFACE
SUPPRESS BIN EXTENSION

LOCAL SYMBOL DEFINITIONS:

LNKCSR = 160100
LNKTCR = LNKCSR + 4
LNKRD = LNKCSR + 2
LNKLPR = LNKCSR + 2
LNKVEC = 400
TLM = 252
MSLOFF = 2
HDSIZ = 6
PRO = 0
PR7 = 340
ASCZER = 60
G = 107
MESSAGE START CONSTANT
MESSAGE OFFSET IN ARRAY
MESSAGE HEADER SIZE
PRIORITY ZERO
PRIORITY SEVEN
ZERO IN ASCII
LETTER G IN ASCII

LOCAL MACROS:

CHANGES:

ORIGINAL PAGE IS
OF POOR QUALITY

存存存存存

: REGISTER SAVE AREA

R1SAV: .WORD 0
R2SAV: .WORD 0
R3SAV: .WORD 0
R4SAV: .WORD 0

: INTERRUPT ENABLE FLAG

: BUFFER A ADDRESS

: DONE FLAG ADDRESS FOR A

: ERROR FLAG ADDRESS FOR A

: BUFFER B ADDRESS

: DONE FLAG ADDRESS FOR B

: ERROR FLAG ADDRESS FOR B

: CHECKSUM

: BYTE COUNTER

: BUFFER A ADDRESS SAVE AREA

: BUFFER B ADDRESS SAVE AREA

: MESSAGE LENGTH STORAGE

: BUFFER A LENGTH

: BUFFER B LENGTH

: BUFFER IN USE FLAG

: GOOD BUFFER FLAG

: EMT ARG BLOCK

: LIST FOR DEVICE

: CLEAR CSR

: AND TRANSMIT CONTROL

: LNKCSR + 4

: LNKCSR

: RESTORE PREVIOUS VECTOR

: RESTORE PREVIOUS INTERRUPT ADDRESS

: END-OF-LIST

: OFFSET VALUE FOR RELOCATION

: OFFVAL: .WORD 0

: TIME: .WORD 0,0

: EMTBLK: .WORD 0,0

: ASCII MESSAGE AREA

VEGMSG: .ASCII /??-RX-DRIVER-W-VECTOR ALREADY PROTECTED ???

ERRMSG: .ASCII /WAIT ERROR.../K15>K12>K200>

: EVEN

: ****

: FUNCTION DETAILS:

: INPUTS:

: JZINIT HAS NO INPUTS.

: INITRD RECEIVES FROM THE CALLING FORTRAN ROUTINE:

: BUFA, BUFB: ADDRESSES OF DATA BUFFERS.

: DONA, DONB: ADDRESSES OF FLAGS TO CALLING

: ROUTINE THAT BUFFERS FILLED.

: ERRA, ERRE: ADDRESSES OF FLAGS TO CALLING

: ROUTINE OF ERROR CONDITIONS.

CALLING ROUTINE.
 DZSTOP HAS NO INPUTS.
 IAND RECEIVES TWO VALUES FROM THE CALLING FORTRAN
 ROUTINE TO BE LOGICALLY AND'ED.
 NIPSTA HAS NO INPUTS.

OUTPUTS:

DZINIT HAS NO OUTPUT.
 INITRD HAS NO OUTPUT.
 RXINT RETURNS THE ONE FLAG AND ERROR
 FLAGS.
 THE ERROR FLAGS CONTAIN THE FOLLOWING:
 BIT 0- SYNCHRONIZATION ERROR, MISSED DATA
 BIT 1- ERROR IN HEADER CHECKSUM
 BIT 2- ERROR IN TRAILER CHECKSUM
 BIT 3- BUFFER NOT READ IN TIME. OVERWRITTEN.
 BIT 4- MULTIPLE ERRORS.
 DZOUT SENDS DATA TO THE DZ-11.
 DZSTOP HAS NO OUTPUTS.
 IAND RETURNS ONE OR ZERO IN R0.
 NIPSTA HAS NO OUTPUT.

SIDE EFFECTS: DZINIT, RXINT, DZOUT, IAND, NIPSTA AND DZSTOP HAVE
 NO SIDE EFFECTS.

INITRD CHANGES ALL OF THE PARAMETERS SENT BY THE
 CALLING ROUTINE.

 START CODE HERE

PAGE

CALL .IIVOUT, GDEVIC, PROTECT, .IRINT, .IIVAT, .DZUT
 IAND, NIPSTA, INITRD, RXINT, DZINIT, DZOUT, DZSTOP

DZINIT INITIALIZES THE DZ CARD AND SETS UP THE VECTORS.

DZINIT: MOV PC, R0 ; PROCESS FOR POSITION INDEPENDENCE

OFFSETS: MOV R0, OFRVAL ; SAVE OFFSET VALUE

MOV #20, @LNKCSR ; CLEAR THE DZ11

LO: BIT #20, @LNKCSR ; CHECK THAT THE DZ11 CLEARED

BNE 10\$

MOV #17030, @LNKLP3 ; SET UP PORT PARAMETERS:

; 9600 BAUD, NO PARITY, ONE STOP-BIT,

; EIGHT BITS

BIS #40, @LNKCSR ; ENABLE PORT 0 RECEIVER INTERRUPT

ORIGINAL PAGE IS
 OF POOR QUALITY


```

00000000      303
00000001      MOV     AG,B4LNKCSR + 6 : SEND ASCII G TO NIP
00000002          :                               : NIP COLD STARTS
00000003          :                               : RESTORE REGISTERS
00000004      MOV     ROSAV,R0
00000005      MOV     RISAV,R1
00000006      RTS     PC
00000007
00000008      ; *****
00000009      ; INITRD COPIES ADDRESSES AND DATA SENT BY THE CALLING ROUTINE,
00000010      ; SETS UP THE LOCAL VARIABLES, AND ENABLES INTERRUPTS.
00000011      ;
00000012      ; *****
00000013      ; INITRD:
00000014      MOV     2(CR5),DUEA           : SAVE PARAMETER ADDRESSES, BUFFER A
00000015      MOV     4(CR5),DONEA         : DONE FLAG FOR BUFFER A
00000016      MOV     5(CR5),ERRA       : ERROR FLAG FOR BUFFER A
00000017      MOV     10(CR5),EUFEB       : BUFFER B
00000018      MOV     12(CR5),DONEB       : DONE FLAG FOR BUFFER B
00000019      MOV     14(CR5),ERRBB        : ERROR FLAG FOR BUFFER B
00000020      CLR     3(DONA)              : CLEAR VARIABLES
00000021      CLR     3(ERBA)
00000022      CLR     10(ENDB)
00000023      CLR     12(EFEB)
00000024      CLR     14(ENDB)
00000025      CLR     16(ENDB)
00000026      CLR     18(ENDB)
00000027      CLR     20(ENDB)
00000028      CLR     22(ENDB)
00000029      CLR     24(ENDB)
00000030      CLR     26(ENDB)
00000031      CLR     28(ENDB)
00000032      CLR     30(ENDB)
00000033      CLR     32(ENDB)
00000034      CLR     34(ENDB)
00000035      CLR     36(ENDB)
00000036      CLR     38(ENDB)
00000037      CLR     40(ENDB)
00000038      CLR     42(ENDB)
00000039      CLR     44(ENDB)
00000040      CLR     46(ENDB)
00000041      CLR     48(ENDB)
00000042      CLR     50(ENDB)
00000043      CLR     52(ENDB)
00000044      CLR     54(ENDB)
00000045      CLR     56(ENDB)
00000046      CLR     58(ENDB)
00000047      CLR     60(ENDB)
00000048      CLR     62(ENDB)
00000049      CLR     64(ENDB)
00000050      CLR     66(ENDB)
00000051      CLR     68(ENDB)
00000052      CLR     70(ENDB)
00000053      CLR     72(ENDB)
00000054      CLR     74(ENDB)
00000055      CLR     76(ENDB)
00000056      CLR     78(ENDB)
00000057      CLR     80(ENDB)
00000058      CLR     82(ENDB)
00000059      CLR     84(ENDB)
00000060      CLR     86(ENDB)
00000061      CLR     88(ENDB)
00000062      CLR     90(ENDB)
00000063      CLR     92(ENDB)
00000064      CLR     94(ENDB)
00000065      CLR     96(ENDB)
00000066      CLR     98(ENDB)
00000067      CLR     100(ENDB)
00000068      CLR     102(ENDB)
00000069      CLR     104(ENDB)
00000070      CLR     106(ENDB)
00000071      CLR     108(ENDB)
00000072      CLR     110(ENDB)
00000073      CLR     112(ENDB)
00000074      CLR     114(ENDB)
00000075      CLR     116(ENDB)
00000076      CLR     118(ENDB)
00000077      CLR     120(ENDB)
00000078      CLR     122(ENDB)
00000079      CLR     124(ENDB)
00000080      CLR     126(ENDB)
00000081      CLR     128(ENDB)
00000082      CLR     130(ENDB)
00000083      CLR     132(ENDB)
00000084      CLR     134(ENDB)
00000085      CLR     136(ENDB)
00000086      CLR     138(ENDB)
00000087      CLR     140(ENDB)
00000088      CLR     142(ENDB)
00000089      CLR     144(ENDB)
00000090      CLR     146(ENDB)
00000091      CLR     148(ENDB)
00000092      CLR     150(ENDB)
00000093      CLR     152(ENDB)
00000094      CLR     154(ENDB)
00000095      CLR     156(ENDB)
00000096      CLR     158(ENDB)
00000097      CLR     160(ENDB)
00000098      CLR     162(ENDB)
00000099      CLR     164(ENDB)
00000100      CLR     166(ENDB)
00000101      CLR     168(ENDB)
00000102      CLR     170(ENDB)
00000103      CLR     172(ENDB)
00000104      CLR     174(ENDB)
00000105      CLR     176(ENDB)
00000106      CLR     178(ENDB)
00000107      CLR     180(ENDB)
00000108      CLR     182(ENDB)
00000109      CLR     184(ENDB)
00000110      CLR     186(ENDB)
00000111      CLR     188(ENDB)
00000112      CLR     190(ENDB)
00000113      CLR     192(ENDB)
00000114      CLR     194(ENDB)
00000115      CLR     196(ENDB)
00000116      CLR     198(ENDB)
00000117      CLR     200(ENDB)
00000118      CLR     202(ENDB)
00000119      CLR     204(ENDB)
00000120      CLR     206(ENDB)
00000121      CLR     208(ENDB)
00000122      CLR     210(ENDB)
00000123      CLR     212(ENDB)
00000124      CLR     214(ENDB)
00000125      CLR     216(ENDB)
00000126      CLR     218(ENDB)
00000127      CLR     220(ENDB)
00000128      CLR     222(ENDB)
00000129      CLR     224(ENDB)
00000130      CLR     226(ENDB)
00000131      CLR     228(ENDB)
00000132      CLR     230(ENDB)
00000133      CLR     232(ENDB)
00000134      CLR     234(ENDB)
00000135      CLR     236(ENDB)
00000136      CLR     238(ENDB)
00000137      CLR     240(ENDB)
00000138      CLR     242(ENDB)
00000139      CLR     244(ENDB)
00000140      CLR     246(ENDB)
00000141      CLR     248(ENDB)
00000142      CLR     250(ENDB)
00000143      CLR     252(ENDB)
00000144      CLR     254(ENDB)
00000145      CLR     256(ENDB)
00000146      CLR     258(ENDB)
00000147      CLR     260(ENDB)
00000148      CLR     262(ENDB)
00000149      CLR     264(ENDB)
00000150      CLR     266(ENDB)
00000151      CLR     268(ENDB)
00000152      CLR     270(ENDB)
00000153      CLR     272(ENDB)
00000154      CLR     274(ENDB)
00000155      CLR     276(ENDB)
00000156      CLR     278(ENDB)
00000157      CLR     280(ENDB)
00000158      CLR     282(ENDB)
00000159      CLR     284(ENDB)
00000160      CLR     286(ENDB)
00000161      CLR     288(ENDB)
00000162      CLR     290(ENDB)
00000163      CLR     292(ENDB)
00000164      CLR     294(ENDB)
00000165      CLR     296(ENDB)
00000166      CLR     298(ENDB)
00000167      CLR     300(ENDB)
00000168      CLR     302(ENDB)
00000169      CLR     304(ENDB)
00000170      CLR     306(ENDB)
00000171      CLR     308(ENDB)
00000172      CLR     310(ENDB)
00000173      CLR     312(ENDB)
00000174      CLR     314(ENDB)
00000175      CLR     316(ENDB)
00000176      CLR     318(ENDB)
00000177      CLR     320(ENDB)
00000178      CLR     322(ENDB)
00000179      CLR     324(ENDB)
00000180      CLR     326(ENDB)
00000181      CLR     328(ENDB)
00000182      CLR     330(ENDB)
00000183      CLR     332(ENDB)
00000184      CLR     334(ENDB)
00000185      CLR     336(ENDB)
00000186      CLR     338(ENDB)
00000187      CLR     340(ENDB)
00000188      CLR     342(ENDB)
00000189      CLR     344(ENDB)
00000190      CLR     346(ENDB)
00000191      CLR     348(ENDB)
00000192      CLR     350(ENDB)
00000193      CLR     352(ENDB)
00000194      CLR     354(ENDB)
00000195      CLR     356(ENDB)
00000196      CLR     358(ENDB)
00000197      CLR     360(ENDB)
00000198      CLR     362(ENDB)
00000199      CLR     364(ENDB)
00000200      CLR     366(ENDB)
00000201      CLR     368(ENDB)
00000202      CLR     370(ENDB)
00000203      CLR     372(ENDB)
00000204      CLR     374(ENDB)
00000205      CLR     376(ENDB)
00000206      CLR     378(
```

```

#FILLB--,R3
JMP (R3)
FILL: FILL : BRANCH IF BUFFER B
: BUFA,ERRA,MSLA,SAVBUA : LOAD BUFFER A
: BUF8,ERRB,MSLB,SAVBUB : LOAD BUFFER B
: ABUF : CHECK BUFFER IN USE
: CHNGB : BRANCH IF BUFFER B
: GOODBU : CHECK FOR ERRORS
: CLRVAR : DON'T SWITCH BUFFERS IF FOUND
: ABUF : SWITCH BUFFERS
: GOODBU : RESET BUFFER FULL FLAG
: INCB : SET DONE FLAG FOR BUFFER A
: TSTB : CHECK TO VERIFY THAT BUFFER B CLEARED
: BEQ : KEEP GOING IF NO COLLISION
: BIS : COLLISION ERROR
BR
CHNGB: : CHECK FOR ERRORS
: GOODBU : DON'T SWITCH BUFFERS IF FOUND
: CLRVAR : SWITCH BUFFERS
: ABUF : RESET BUFFER FULL FLAG
: INCB : SET DONE FLAG FOR BUFFER B
: TSTB : CHECK TO VERIFY THAT BUFFER A CLEARED
: BEQ : KEEP GOING IF NO COLLISION
: BIS : COLLISION ERROR
CLRVAR: TST
BEQ
MOV
BR
10$:
20$:
MSL
CKSM
I
R0SAV,R0
R1SAV,R1
R2SAV,R2
R3SAV,R3
R4SAV,R4
RTI
: ****
:
: DZOUT OUTPUTS A STRING TO THE DZ OUTPUT BUFFER (AND SO TO THE NIP)
:
: ****
DZOUT:
10$:
MOV 2(R5),R1 : STORE BUFFER'S ADDRESS
MOV 34(R5),R2 : # OF BYTES TO SEND
T #100000,3#LNKCSR : READY TO SEND?
BEQ 10$ : NO; THEN LOOP

```

ORIGINAL PAGE IS
OF POOR QUALITY

; DZSTOP-TURNS OFF THE INTERRUPT ROUTINE AND THE DZ CARD

; DZSTOP: DEG-ENA6 ; TURN OFF INTERRUPT SERVICE ROUTINE
; BIC #100, @LNKCSR ; TURN HARWARE INTERRUPTS OFF
; RTS PC

END

..TITLE DRGPs CONTROL DR CARD
IDENT 701/

AAB ON 4/13/84

AUTHOR:

CHANGES:

MODULE FUNCTION

THIS SET OF ROUTINES CONTROLS COMMUNICATIONS TO THE PARALLEL PORT
CONNECTED TO THE RMOU

CODE IS POSITION INDEPENDENT AND CAN BE USED WITH EB MONITOR

OR DETAILS:

DRCR = 167770
DROUBUF 167772
DROINBUF 167774

PRIORITY 5 (USER SELECTABLE ON BOARD)
VECTORS 300 (REQUEST A) OR 304 (REQUEST B)
BIT 6 IN CSR SETS IF A TO ALLOW INTERRUPTS WHEN PEQA IS SET BY
DEVICE

PAGE

NCIST BFX

SEEK PAGE FOR PREFACE
SUPPRESS BIN EXTENSION

LOCAL SYMBOL DEFINITIONS:

DRVEC = 300

DRCR = 167770

DROUBUF = DRCR + 2

ORMSLOFF = 2

PRO = 0

PR7 = 340

ZERO = 0

OROVER = 0

LENGTH OFFSET IN BUFFER
NO PRIORITY
MAXIMUM PRIORITY
OVERHEAD

LOCAL MACROS:

ORIGINAL PAGE IS
OF POOR QUALITY

```

R0SAVDR: WORD 0      ; REGISTER SAVE AREA
R1SAVDR: WORD 0
R2SAVDR: WORD 0
OFFVAL: WORD 0
USEBUF: WORD 0      ; BUFFER IN USE FLAG
DRBUE: WORD 0      ; OUTPUT BUFFER ADDRESS
DRENAB: WORD 0      ; INTERRUPT ENABLE FLAG
DRLN: WORD 0      ; LENGTH OF BUFFER
ORDONE: WORD 0      ; ADDRESS OF DONE FLAG
DAREA: BLKW 3      ; EMT ARG BLOCK
DRLIST: WORD DRCSR  ; LIST FOR DEVICE
WORD 0      ; CLEAR CSR
DRVSAR: WORD 0,0    ; RESTORE PREVIOUS VECTOR
WORD 0,0      ; RESTORE OLD INTERRUPT ADDRESS
WORD 0      ; END-OF-TABLE

```

``` ; ***** ; FUNCTION DETAILS: ```

```

; INPUTS:
;
; CALLING ROUTINE SENDS BUFFER ADDRESS AND DONE FLAG
; ADDRESS TO DROUT. DRSTOP AND DRINIT RECEIVE NO
; INPUTS. DRINT IS INTERRUPT DRIVEN.

```

```

; OUTPUTS:
;
; DROUT RETURNS DONE FLAG TO FORTRAN ROUTINE.
; DRINIT OUTPUTS CONTROL WORD TO DR DEVICE.
; DRSTOP HAS NO OUTPUTS.
; DRINT OUTPUTS EITHER ZERO OR DATA FROM THE BUFFER
; RECEIVED FROM THE FORTRAN ROUTINE TO THE DR.

```

```

; SIDE EFFECTS: DONE FLAG CHANGED.

```

```

; *****

```

```

; START CODE HERE

```

```

.PAGE
.MCALL .DEVICE,.PROTECT
.GLOBL DRINIT, DROUT, DRINT, DRSTOP

```

```

; *****

```

```

; DRINIT INITIALIZES THE DR BOARD AND SETS UP THE VECTOR

```

```

; *****

```

```

DRINIT:
        MOV     R0,ROSAVDR
        MOV     PC,R0

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

MOV #XVEC, DRVSAB
MOV @DRVEC, DRVSAB + 2
MOV #ORVEC + 2, DRVSAB + 4
MOV @DRVEC + 2, DRVSAB + 6
;DEVICE #DRAREA, #DRLIST
MOV #DRINT-OFFSET, #DRPVEC
;
ADD OFFVAL, @DRVEC
MOV #PR7, @DRVEC + 2
MOV #100, @DRCSR
INC DRENAB
MOV RDSAVDR, RD
RTS PC

```

; ****

```

; DRQUT HANDLES SET UP TO OUTPUT A BUFFER WHEN CALLED BY THE FORTRAN
; ROUTINE. DRINT WILL OUTPUT THE BUFFER.
;
; ****

```

DRQUT:

```

MOV R1, RISAVDR
MOV 2(R5), DRBUF
MOV 4(R5), DRDOONE
CLR R8
MOV DRBUF, R1
ADD #DRMSLOFF, R1
MOV (R1), DRLEN
ADD #DROVER, DRLEN
BIT #1, DRLEN
BEQ 10$
MOV DRBUF, R1
ADD DRLEN, R1
MOV #ZERO, (R1)
INC DRLEN
10$: ASR DRLEN
MOV RISAVDR, R1
INC USEBUF
RTS PC

```

; ****

```

; DRSTOP DISABLES DRINT AND LOADS THE DR OUTPUT BUFFER WITH ZERO
;
; ****

```

DRSTOP:

```

DRENAB
MOV #ZERO, @DRROUTINE

```

```

; LOAD INTERRUPT ROUTINE ADDRESS
; AT INTERRUPT VECTOR

```

```

; SET PSW FOR PRIORITY 7
; ENABLE INTERRUPTS
; ENABLE INTERRUPT ROUTINE

```

```

; SAVE REGISTER
; SAVE ADDRESSES OF BUFFER
; DONE FLAG
; CLEAR DONE FLAG
; FETCH LENGTH WORD TO DRLEN

```

```

; CORRECT FOR OVERHEAD
; CHECK IF ODC # OF BYTES
; BRANCH IF NOT
; ZERO LAST BYTE

```

```

; COUNT LAST BYTE
; CHANGE BYTE COUNT TO WORD COUNT
; RESTORE REGISTER
; SET USE BUFFER FLAG

```

```

; DISABLE INTERRUPT ROUTINE
; SEND ZEROS TO RMOU

```


PRINT OUTPUTS ONE WORD TO THE OR OUTPUT BUFFER AT EACH INTERRUPT.
 WHEN NO DATA IS AVAILABLE, A ZERO IS OUTPUT TO SEPARATE MESSAGES.
 MESSAGE FORMAT IS THE SAME AS DESCRIBED IN THE NIP USER'S MANUAL.

```

: *****
PRINT:
MOV R2,R2SAVDR      : SAVE REGISTERS
TSTB ORENAB         : VERIFY ROUTINE ENABLED
ONE 10$             : BRANCH IF TRUE
BR OREND
MOV PC,R2           : OTHERWISE EXIT
ADD #OREND-.,R2
JMP (R2)
10$: TST USERBUF
ONE 30$             : CHECK FOR BUFFER IN USE
MOV #ZERO,#DROUTBUF : BRANCH IF BUFFER PRESENT
BR OREND            : OTHERWISE OUTPUT ZERO,
MOV PC,R2           : AND EXIT
ADD #OREND-.,R2
JMP (R2)
30$: MOV DRBUF,#DROUTBUF : OUTPUT NEXT WORD
INC DRBUF           : INCREMENT ADDRESS TO NEXT WORD
INC DRBUF
DEC ORLEN           : DECREMENT COUNTER
TST ORLEN           : TEST IF LAST WORD SENT
BEQ 40$             : BRANCH IF BUFFER FINISHED
BR OREND
MOV PC,R2           : OTHERWISE EXIT
ADD #OREND-.,R2
JMP (R2)
40$: INCB DRBDONE : SET DONE FLAG
CLR USERBUF        : CLEAR USE BUFFER FLAG
OREND: MOV R2SAVDR,R2 : RESTORE REGISTERS
RTI
SEND

```

ORIGINAL PAGE IS
 OF POOR QUALITY

APPENDIX C

REPORT OF CONTRACT WORK
FOR UNIVERSITY OF SOUTHERN COLORADO
JULY 1983 - AUGUST 1983
TIM R. JOHNSON

My main task during this two-month period was to install the RSX11M operating system on the Helicopter Group Lab's LSI 11/23 computer.

The LSI 11/23 computer system is connected to various peripherals (see diagram) including a Ramtek-9400 color graphic system, voice input and output devices, digitizer tablet, and touch panel. The system is used as a general purpose graphics workstation for design of interactive color displays and concepts. The computer had been running under the RT-11 operating system, but we decided that the RSX11M operating system (multi-user, multi-tasking) would make better use of the hardware resources for the following reasons:

- 1) timesharing - two or more people could be developing, editing, compiling, or doing other tasks at the same time.
- 2) bigger program space - RSX11M makes better use of the 128 kwords of memory present on the system, allowing programs up to 32 kwords in size (larger if overlaid).
- 3) multi-programming - more than one process can run simultaneously, allowing more overall throughput, better real-time response for those programs that need it, and new programming options such as inter-task communication and parallelism.
- 4) improved security and organization - files and programs can be logically grouped into separate directories, and files can be protected against accidental or unauthorized deletion or alteration.

The work of converting to RSX11M consisted mainly of writing, modifying, and installing new system software drivers for the various peripherals, testing these, and writing new I/O routines in the graphics support library to use the new drivers. In this way, many RT-11 programs that were written for the color graphics system and other devices would

need very few changes made to them in order to run under the new RSX11M operating system. In addition, some new features offered by the RSX11M drivers could be used by all programs, present and future, in a simple, general way.

The following drivers were installed:

- a) serial drivers for the touchpanel and Voterm -

I decided to use the RSX11M terminal driver (TTDRV) to communicate with our two serial devices, the touchpanel and Voterm voice recognition unit. The use of the system terminal driver allows a separate interrupt-driven 'typesahead' buffer for each device, which can then be queried by a program when it wants input from the device. The buffer can be checked to see if any characters are in it, and the program can specify whether to wait until some are (hang mode) or return to the program and continue (nohang mode). In this way, both touchpanel and Voterm can be used as dual, either/or input devices, which provides flexibility in programming and built-in backup should one or the other device be unavailable.

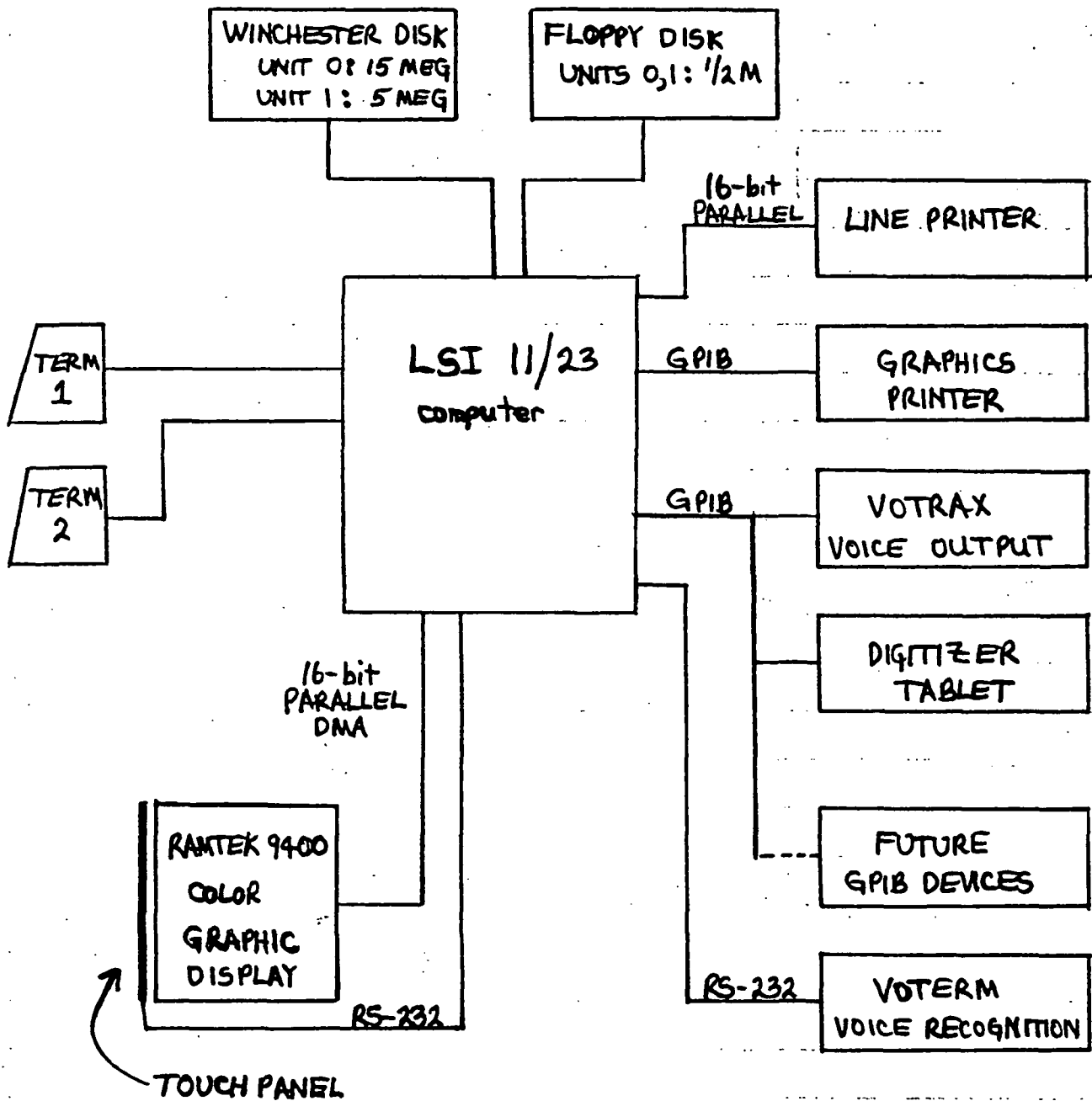
- b) graphics printer - The Hewlett-Packard graphics printer was connected to a dedicated IBV11 interface and cable (general purpose interface bus, IEEE-488 standard), and I modified the RSX11M standard line printer driver (LPDRV) to run the HP printer. The main changes were that the new HP driver has to address the HP printer as a GPIB device on startup as well as when the printer goes offline, since the printer will become un-addressed if powered off during printing (to clear a paper jam, for instance). Also, the new HP driver must pass all characters to the printer, since the driver will also be used to print from the Ramtek screen. This will involve sending binary data to the printer (with the printer in graphics mode) so tabs and other control characters must cause no special action on the part of the new HP driver.
- c) Ramtek driver - A device driver for the Ramtek-9400 was obtained and installed, and new I/O routines for the Ramtek graphic support library (Ramlib) were written to use the new driver.

The driver uses direct-memory access to communicate with the Ramtek device, so I/O to the Ramtek is faster now than under RT-11, which used programmed I/O.

- d) GPIB driver (for Votrax and digitizer tablet) - A new driver, IBDRV, was obtained and installed to communicate with the Votrax voice synthesizer and the HP digitizer tablet, both devices on the GPIB (IEEE-488) bus. Future devices can be simply plugged onto the external cable for this bus, set to a unique GPIB bus address, and programmed through the IBDRV driver. Two new I/O routines, IBREAD and IBWRITE, were written and included in the Ramtek support library to communicate with GPIB devices. In addition, the routine IBSETMODE allows the IB: driver to be set up in four different read and write modes, as well as specifying the end-of-data character for a read of a GPIB device.

The conversion of old RT-11 graphic programs to the new RSX11M system was fairly straightforward. Some procedure names had to be changed, (ie from Writedriver, Readdriver to Ibwrite, Ibread), and the touchpanel and Voterm routines have a new parameter 'handmode', a boolean, to specify whether to wait for input or return immediately if there is none. In addition, a new routine, Flushbuffer, was added to the support library. This routine flushes the touchpanel or Voterm 'typeahead' buffer, since there will be some cases where a program does not want any previous input from the device but only input that occurs after a specific request for input from that device. The usual case when this happens is during successive menu box selection using the touchpanel, where two or three 'hits' for the same menu box will be stored in the typeahead buffer, but all those after the first hit are redundant and possibly wrong if taken to be the input for the next series of menu box tests on a new display frame, for instance.

SYSTEM CONFIGURATION



APPENDIX D

NASA
SUMMER
PROJECT

June-August, 1983

by
Richard W. Parker

to
Dr. Don Cottrell

During the summer of 1983 at NASA Ames Research Center, located at Moffett Field in California, I performed various technical and development tasks. The tasks that were performed and required of me were to assist Ralph Carestia in the development of a few boards used on the project and to do technical tasks on projects previously started.

I assembled two Intel SDK-86 single board computers and trouble-shot another one to be used for training purposes by Ralph. The SDK-86's were very enlightening to assemble, because they gave me an understanding of a 16 bit computer system and some familiarity with 8086 assembly language.

The boards that were assembled and developed were couple port I/O boards, a 16 port DAC board, a digital to syncro board, a digital to analog discrete board, and an analog to digital discrete board. With regard to the later two boards, I was instrumental in their design. The digital to analog changed a digital word from a S-100 interface to 28vdc and 0vdc for the simulator use, the analog to digital changed the 28vdc input to a logic 5vdc level used in constructing words utilized by the computer.

The digital to analog board used a DG162 for a switching device for operating a transistor, used as a current sinking device, switching developing the 28vdc used as an input to the simulator.

The analog to digital board used a comparator (LM 311) to change a 28vdc input to a logic 5vdc level. I used a debounce circuit to eliminate unwanted spikes caused by the switches bouncing when the operator used them. A storage register (D flip/flop) is used to latch the information until the computer requests the data. A signal from the computer resets the register after it receives the data to enable sampling of the word again. The signal used for the reset function was the Data Transmitted line.

The summer employment was very enjoyable and a good experience for students to have. The expense was a little rough if the students have some financial responsibility elsewhere. But I enjoyed the experience and thank the university for selecting me to participate in the program and I hope the program will be able to continue.