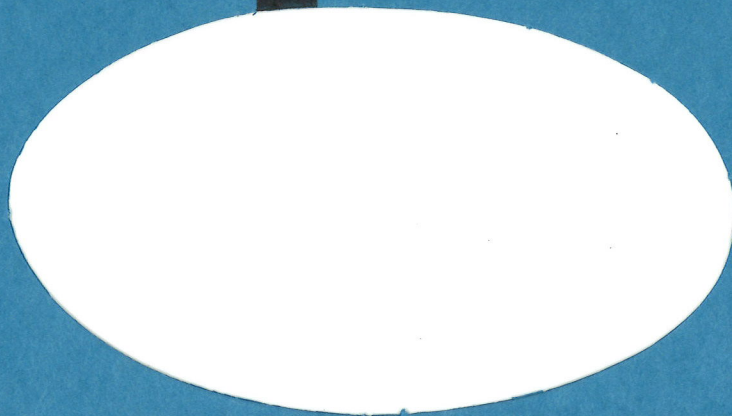


MASSACHUSETTS INSTITUTE OF TECHNOLOGY

# APOLLO

GUIDANCE, NAVIGATION  
AND CONTROL



**MIT**

**CHARLES STARK DRAPER  
LABORATORY**

CAMBRIDGE MASSACHUSETTS 02139

# APOLLO

## GUIDANCE, NAVIGATION AND CONTROL

Approved: Eldon C Hall Date: 2/10/71  
E.C. HALL, DIRECTOR, DIGITAL DEVELOPMENT  
APOLLO GUIDANCE AND NAVIGATION PROGRAM

Approved: David G Hoag Date: 10 Feb 71  
D.G. HOAG, DIRECTOR  
APOLLO GUIDANCE AND NAVIGATION PROGRAM

Approved: Ralph R Ragan Date: 10 Feb 71  
R.R. RAGAN, DEPUTY DIRECTOR  
CHARLES STARK DRAPER LABORATORY

E-2463

### APOLLO GUIDANCE COMPUTER IMPROVEMENT STUDY

by

Donald J. Bowler, et al

January 1970

# MIT

# CHARLES STARK DRAPER LABORATORY

CAMBRIDGE, MASSACHUSETTS, 02139

INDEXING DATA

DATE	OPR	#	T	PGM	SUBJECT	SIGNATOR	LOC
1-31-70	MIT	E-2463	R	APD	#	BOWLER	084-63

## ACKNOWLEDGEMENT

This report was prepared under DSR Project No. 55-38600, Task Statement No. 36, "AGC Improvement Program", sponsored by the Manned Spacecraft Center of the National Aeronautics and Space Administration through contract NAS9-4065.

This report is the compilation of the efforts of several people in the Digital Development Group.

Mr. Alan Harano	Sections 2.1, 2.5
Mr. Thomas Zulon	Section 2.2
Mr. John McKenna	Sections 3.1-3.3
Mr. Robert Scott	Section 3.4
Mr. William Weinstein	Sections 4.2-4.8
Mr. Albert Lee (Raytheon resident)	Sections 4.2, 4.6
Mr. Eldrid Atkins (Raytheon resident)	Sections 6.2, 7.6
Mr. Hugh Blair-Smith	Sections 7.2, 7.3
Mr. Brad Ross (CCA resident)	Section 7.2

Appreciation and credit are due to Mr. Eldon C. Hall, Director of the Digital Development Group, for his comments, interest, and guidance throughout this study.

The services of Mr. Wallis Bean and the Technical Publications Group are reflected throughout the report.

E-2463

APOLLO GUIDANCE COMPUTER  
IMPROVEMENT STUDY

ABSTRACT

This report studies several ways of increasing the Apollo Guidance Computer capabilities. The means include modifying a memory tray (Tray B) so as to include 16K of erasable, 65K of fixed, and an interface capability for expanding input/output, including a tape unit. An alternative to modifying Tray B was to use an Auxiliary Memory Unit developed by Raytheon Co. This latter approach is recommended.

The possibility of increasing throughput and programming flexibility by hardware and software changes was investigated. These investigations included software and hardware memory relocation schemes and development of a hardware dispatcher for the interpreter.

By: Donald J. Bowler, et al  
January 1970

## TABLE OF CONTENTS

Chapter		Page
1	INTRODUCTION . . . . .	1
	1.1 Objectives . . . . .	1
	1.2 AGC Growth Requirements . . . . .	2
2	ERASABLE MEMORY . . . . .	7
	2.1 Logical Design . . . . .	7
	2.1.1 Ground Rules . . . . .	7
	2.1.2 Address Structure . . . . .	8
	2.1.3 Control Logic . . . . .	11
	2.2 Mechanical Design of the 16K Tray B . . . . .	17
	2.2.1 Portafam Memory Component List . . . . .	17
	2.2.2 Integrated Circuit Availability . . . . .	26
3	BRAID MEMORY . . . . .	29
	3.1 Introduction . . . . .	29
	3.2 Organization . . . . .	30
	3.3 Electronics . . . . .	30
	3.3.1 Word-Line Selection . . . . .	33
	3.3.2 Sensing . . . . .	33
	3.4 Mechanical Design . . . . .	39
	3.4.1 First Sample Weave . . . . .	41
	3.4.2 Second Sample Weave . . . . .	42
	3.4.3 Terminations . . . . .	44
4	PROGRAM EVALUATION . . . . .	45
	4.1 Auxiliary Memory . . . . .	45
	4.2 Program Evaluation Sequence . . . . .	49
	4.3 AM-AGC Communication (Pinball Changes) . . . . .	50
	4.4 Erasable Addressing . . . . .	51
	4.5 Interpreter Changes . . . . .	52
	4.6 Mission Segmentation . . . . .	53
	4.6.1 By-Program . . . . .	53

Chapter		Page
	4.6.2 By-Bank . . . . .	57
4.7	Restart . . . . .	57
4.8	Conclusions. . . . .	58
5	COMPARISON BETWEEN THE MODIFIED TRAY B AND THE AUXILIARY MEMORY APPROACH . . . . .	59
5.1	Comparison . . . . .	59
5.2	Conclusion . . . . .	61
6	TEST CONNECTOR INTERFACES . . . . .	63
6.1	Introduction . . . . .	63
6.2	Electrical Interface Study . . . . .	66
7	SPECIAL STUDIES . . . . .	87
7.1	Erasable Versus Fixed Memory . . . . .	87
7.2	Speed-Up of the AGC . . . . .	88
	7.2.1 360 Simulation of the AGC Hardware Dispatcher . . . . .	88
	7.2.2 Summary . . . . .	91
7.3	Hardware Relocation . . . . .	93
7.4	Hardware-Software Facilities Necessary to Utilize the Raytheon Prototype Auxiliary Memory . . . . .	93
	7.4.1 Minimum Equipment Needed . . . . .	93
	7.4.2 Additional Software Highly Desirable . . . . .	96
7.5	Portafam Memory Tray-AGC Operation . . . . .	97
7.6	Auxiliary Memory Drawing List . . . . .	97
8	GLOSSARY OF TERMS . . . . .	101
8.1	Apollo Guidance Computer (AGC) . . . . .	101
8.2	AGC Addressing . . . . .	103
8.3	Assembler . . . . .	104
8.4	Assembler YULE . . . . .	104
8.5	Assembler (GAP) . . . . .	104
8.6	Auxiliary Memory . . . . .	104
8.7	BBCON . . . . .	105
8.8	BOTH BANK . . . . .	105
8.9	BRAID . . . . .	105
8.10	Buffer Box . . . . .	105
8.11	Channel . . . . .	105
8.12	CRS (Core-Rope Simulator) . . . . .	105
8.13	CMC (Command Module Computer) . . . . .	110
8.14	Digital Simulator . . . . .	110
8.15	EBANK . . . . .	110

Chapter		Page
8.16	Memory Cycle Time . . . . .	110
8.17	Erasable Memory . . . . .	110
8.18	FBANK . . . . .	110
8.19	FCADRE . . . . .	111
8.20	Fixed Memory . . . . .	111
8.21	Interpreter . . . . .	111
8.22	LGC (LM Guidance Computer) . . . . .	111
8.23	Multiprocessor . . . . .	111
8.24	PAC (Program Analyzer Console) . . . . .	111
8.25	Portafam . . . . .	111
8.26	Relocation . . . . .	112
8.27	Restart Monitor . . . . .	112
8.28	Restart . . . . .	112
8.29	Rope . . . . .	112
REFERENCES . . . . .		113

## CHAPTER 1

### INTRODUCTION

#### 1.1 Objectives

The objectives of this study were several; the most important was how best to increase the memory capacity of the Apollo Guidance Computer. The reasons behind this are developed in Chapter 1.

Essentially two ways have been studied. One way, adding an auxiliary memory, was studied by Raytheon (Ref. 1). Another, modifying the present memory tray and adding more erasable and fixed memory, is discussed in this report in Chapters 2 and 3.

An additional objective was to study what effect the increased memory could have on the mission programs and supporting software. This study was carried out utilizing the prototype auxiliary memory built by Raytheon Co. These results are contained in Chapter 4.

Chapter 5 discusses the relative merits of the external Auxiliary Memory versus the expanded E-memory approach.

The auxiliary memory utilized the computer test connector as an access point into the AGC and, because other groups have expressed an interest in utilizing this interface, Chapter 6 is devoted to a study of this interface.

Chapter 7 contains the results of a group of special studies, most of which are related to expanding the flexibility of the computer.

In order to enhance the readability of this report, a glossary has been put in Chapter 8 defining most of the terms encountered in the body of this report.



## 1.2 AGC Growth Requirements

Anyone who has designed a computer realizes that it is obsolete the day the design is released. The Apollo Computer is no exception: it has evolved as shown in Table 1-I. This table becomes more interesting when one remembers that back in 1962 one thought that 512 words of erasable and 4K words of fixed would be sufficient.

Table 1-I

	<u>E Memory</u>	<u>F Memory</u>	<u>Number of Interfaces*</u>
Block I	1K	24K	134
Block II	2K	36K	233

\*Not including test connector interface.

Expansion of the interface was still possible; in spite of the fact that the hardware design had been frozen before the functional specifications were completed. This was possible because some extra circuits had been provided where space was available and circuits initially unique to the LEM or CM were capable of being utilized by the other when the need arose.

Nevertheless, the need for memory did increase and was tight in both erasable and fixed. Figure 1-1 shows a memory map of erasable Bank 7 indicating which of its parts are being time-shared for Luminary 101 programs.

This shortage of E memory has resulted in very tight control and difficult programming. As the number of required functions increased, the speed of the AGC also became an additional constraint. There is a trade-off on speed versus storage. Several programs that would have been programmed in interpretive were programmed in basic, thus requiring more storage because the interpreter was too slow.

Also, some routines that would have been written in double-precision were written in single-precision to conserve time. These compromises, though adequate, would not otherwise have been done and did require additional analysis and time to find situations where one could accomplish them.

Thus, the lack of memory and the need for faster throughput did result in complicating and delaying the programming effort.

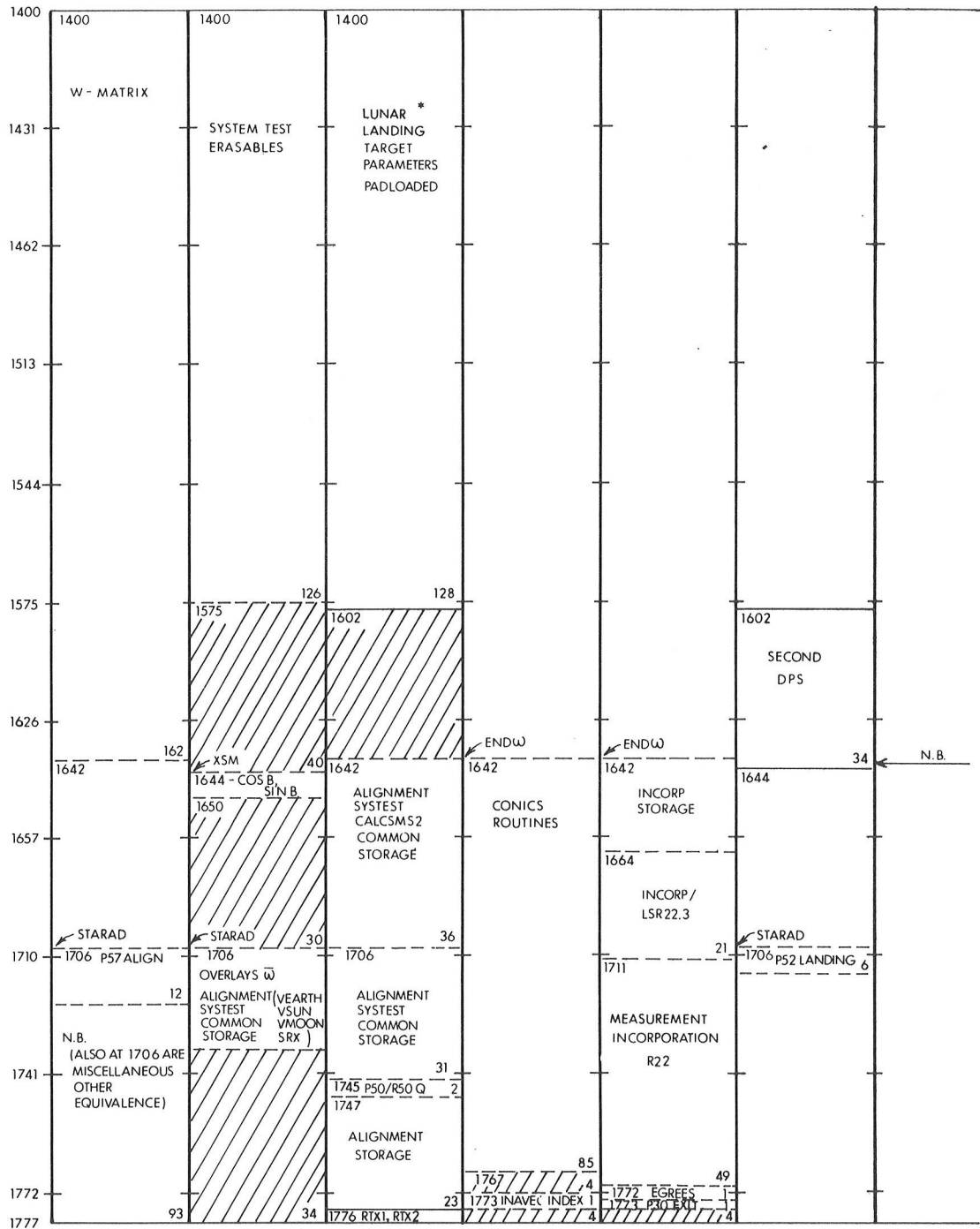


Figure 1-1 Luminary 101 Erasable Memory Map

Possible ways to speed up the AGC are discussed in Sections 7.2 and 7.3 but the major part of the report is concerned with increasing the memory capacity.

Figure 1-2 shows some comparisons in memory-module densities and indicates the reason for being able to consider increasing the memory capacity in the AGC and still remain within the same volume.

Figure 1-3 shows the resulting densities if Tray B were to have the erasable memory increased from 2K to 16K and if the fixed memory were increased from 36K to 65K.

Figure 1-4 shows some predicted speed and power improvements in the fixed-memory development. Since the "Easy Boat" and "New Boat" designs have sufficient output amplitude to be able to drive micrologic directly, they do not require special sense amplifiers. The New-Boat development is discussed in Chapter 3.

AGC ERASABLE (2k, 16 bit)	1, 117 bits/in <sup>3</sup>
SIRU (LOCKHEED) (4k, 32 bit)	840 bits/in <sup>3</sup>
AUX MEM (4k, 16 bit)	2, 024 bits/in <sup>3</sup>
PORTAFAM (16k, 16 bit)	5, 604 bits*/in <sup>3</sup>
AGC ROPE (6k, 16 bit)	2, 830 bits/in <sup>3</sup>
NEW BRAID	40, 000 bits/in <sup>3</sup>

\* INCLUDES SOME ELECTRONICS ON STACK

Fig. 1-2 Memory Module Densities

AGC ERASABLE (2K)	281 bits / in <sup>3</sup>	1
AGC FIXED (36K)	1, 391 bits / in <sup>3</sup>	5
AGC IMPROVED (16K) ERASABLE	1, 967 bits / in <sup>3</sup>	7
AGC IMPROVED (65K) FIXED	4, 100 bits / in <sup>3</sup>	14.6

Fig. 1-3 Memory Density (including Electronics)

	<u>APOLLO</u>	<u>EASY BOAT*</u>	<u>NEW BOAT</u>
BIT DENSITY (BITS/IN <sup>3</sup> )	1400	2300	4100
SPEED (μ SEC)	12	3	<0.5
POWER (WATTS)	20	12	<5

\*USED IN SIMFAM BRAIDS

Fig. 1-4 Fixed Memory Characteristics

## CHAPTER 2.

### ERASABLE MEMORY

As was indicated in the previous chapter, if one utilized the Portafam memory development, one would get an increase in the memory density by a factor of 7 and an increase in memory capacity by a factor of 8.

This chapter is essentially divided into two sections -- the logical design for the 16K erasable and the mechanical design.

#### 2.1 Logical Design

This section outlines a logical design for a 16K erasable memory, contained in tray B of the AGC.

##### 2.1.1 Ground Rules

The following ground rules were used in the design:

##### A. Addressing

1. All 16K can be addressed as erasable via use of an EEXT register.
2. The last 12K can also be addressed as fixed banks 44-57.
3. All 16K can be addressed, as erasable, via external connections.

B. The only access to EEXT is via the Both-Bank Register (address 0006). References to location 0003 will involve only the three bits of EBANK.

C. Whenever 'EBANK' 24 locations 1400-1577 are addressed, appropriate signals will also be transmitted to the external interface.

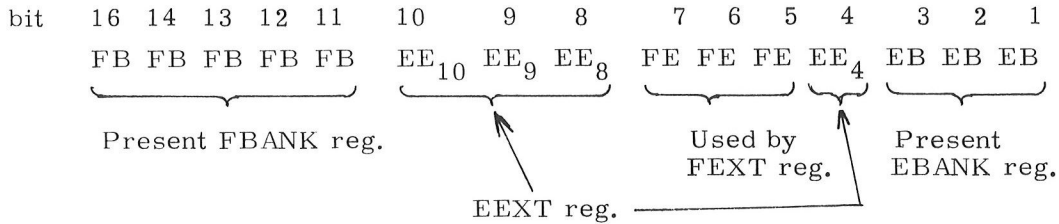
D. External cycle capability will be provided every memory cycle time (MCT) of the AGC.

### 2.1.2 Address Structure

The proposed addressing scheme is summarized in Table 2-I.

#### A. Erasable

The erasable address field is extended by a four-bit addition to the BB register (location 0006). Physically, this extension is located in Tray B, and is denoted EEXT for erasable extension. The bit configuration for location 0006 is:



Bits 1-3 are the contents of the EBANK register.

Bits 11-14, 16 are the contents of the FBANK register.

Bits 5-7 are used, in BBCON'S, for the FEXT register (chan. 7).

Bits 8-10, 4 are the contents of the EEXT register.

The "weighted values" for EEXT are as follows:

$$EE_4 \quad EE_{10} \quad EE_9 \quad EE_8$$

e.g., bit 4 is the most-significant bit and bit 8 is the least-significant bit. This arbitrary decision was made to keep a contiguous field for the three least-significant bits (the least-significant octal character is inserted into bits 8-10).

Erasable addressing under the new scheme is somewhat analogous to fixed addressing. Addresses 0-1377 can be accessed via addresses 0-1377 or EBANK 0-2, 1400-1777, similar to "fixed-fixed" addressing. For EBANK settings 0-3 the erasable address is uniquely specified, similar to FBANK 0-27. For EBANK 4-7 the EEXT register is consulted to determine the desired address, similar to FBANK 30-37 where FEXT is consulted.

Table 2-I. 16K Erasable Address Structure

<u>Equiv.</u> <u>EBANK</u>	<u>EEXT</u>	<u>EBANK</u> <u>Reg</u>	<u>S Reg</u>	14-bit <u>Eras.</u> <u>ADDR</u>	<u>FEXT</u>	<u>FBANK</u> <u>Reg</u>	<u>Equiv.</u> <u>FBANK</u>	
0-2	XX	X	0000-1377	0-1377				} Present 2K of Erasable
0-3	XX	0-3	1400-1777	0-1777				
4-7	00-01	4-7		2000-3777				
024-027	02			4000-5777				} 2K Addressable as Erasable only
034-037	03			6000-7777				
044-047	04			10000-11777	4	34	44	} 12K Addressable as Fixed or Erasable
054-057	05			12000-13777	↓	35	45	
064-067	06			14000-15777		36	46	
074-077	07			16000-17777	↓	37	47	
104-107	10			20000-21777	5	30	50	
114-117	11			22000-23777	↓	31	51	
124-127	12			24000-25777		32	52	
134-137	13			26000-27777		33	53	
144-147	14			30000-31777		34	54	
154-157	15			32000-33777		35	55	
164-167	16			34000-35777		36	56	
174-177	17			36000-37777	↓	37	57	

The reasons for using only EBANKS 4-7 to expand the memory field, as opposed to using 0-7, are:

- 1) Locations 0-1377 (EBANK 0-2, 1400-1777) can be accessed in two ways. It is not possible to infer the actual address components from signals present in Tray B.
- 2) Locations 0-7 (EBANK 0, 1400-1407) are "Special Central" locations (flip-flop registers). Since Tray A does not know that the EEXT register exists, addressing the first eight locations of any erasable bank ending in octal 0 will cause access to the flip-flop registers rather than to the erasable memory.

The advantages of this scheme are:

- 1) Capability to use all 256 words of every bank
- 2) Capability to address the first 1K of erasable regardless of the EEXT setting. This capability is of limited use for machine-code instructions since banks 0-2 can be addressed as 0-1377, but will probably simplify the Interpreter which generally uses EBANK X, 1400-1777.
- 3) Capability to address any other 1K segment without changing EEXT. Data common to many subroutines would be placed in the first 1K and data used by a particular subroutine would be placed in some other 1K segment.

The major disadvantage is that four bits are required for the EEXT register; therefore, a contiguous field is not available for EEXT in a BCON.

#### B. Fixed

The last 12K of the memory can be accessed as "fixed banks" 44-57, causing a read-restore cycle to be executed. There is no change in the operation of FEXT (channel 7) or the FBANK register (location 0004 or 0006). See Table 2-I for cross references between erasable and fixed addresses.



## C. External

An interface is provided for external read-write access to the entire 16K memory by using a 14-bit address register.

### 2.1.3 Control Logic

#### A. Timing

The E-Memory is assumed to have a full-cycle time (read/restore or clear/write) of  $\approx 2$  microseconds, and a split-cycle time (read half-cycle or write half-cycle) of  $< 1.5$  microseconds. Access time is  $\leq 1$  microsecond. Refer to Fig. 2-1 for the detail timing.

##### 1) AGC Erasable Cycles

Because the AGC reads from erasable at time 5 and writes at time 9, the memory will be exercised with a read half-cycle initiated by signal SETEK( T03), the contents of the memory will be gated to the G register by signal SBE ( $\approx T04$ ). All memory activity is then suspended until signal ZID ( $\approx T10$ ) which gates the contents of the G register to the memory buffer register (MBR). Approximately 0.25 microsecond later signal RSTKX ( $\approx T10$ .PHSE3) starts a write half-cycle, completing the AGC Erasable cycle.

Since signal SBE does not occur for special central addresses, no data are transferred from the memory to the G register. For address 6 the contents of the EEXT register are OR'ed into the G register by signal REY•MBRR(MBRR = Memory buffer register ready — generated by the memory sequence generator) which is approximately time equivalent to SBE; the contents of the G register are copied into EEXT by the leading edge of signal ZID.

##### 2) AGC Fixed Memory Cycles (Banks 44-57 only).

A "rope" cycle for fixed banks 44-57 will cause a read/restore cycle to begin at the leading edge of the RESETx pulse at time 5. The memory contents are gated to the G register by signal SBF ( $\approx T06$ .PHS4).

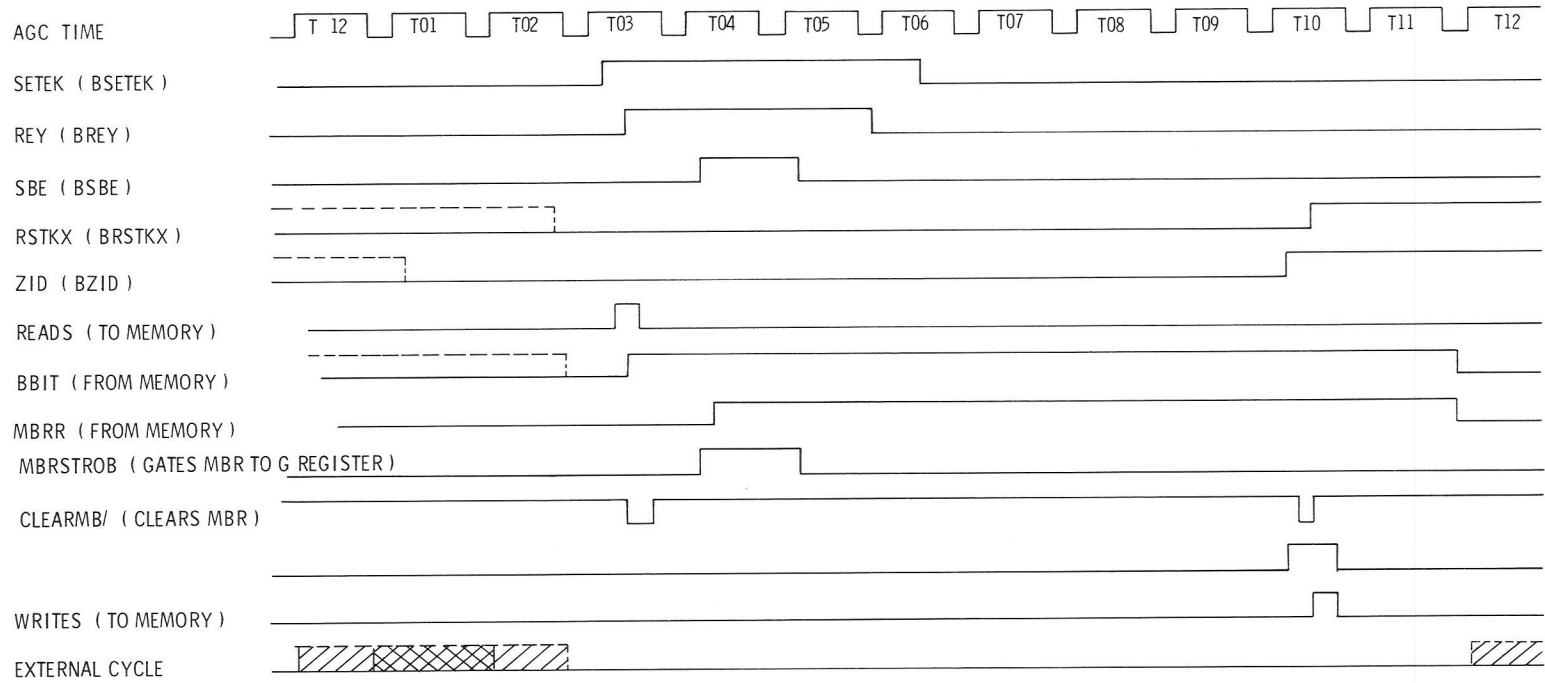


Fig. 2-1a AGC Erasable Memory Cycle

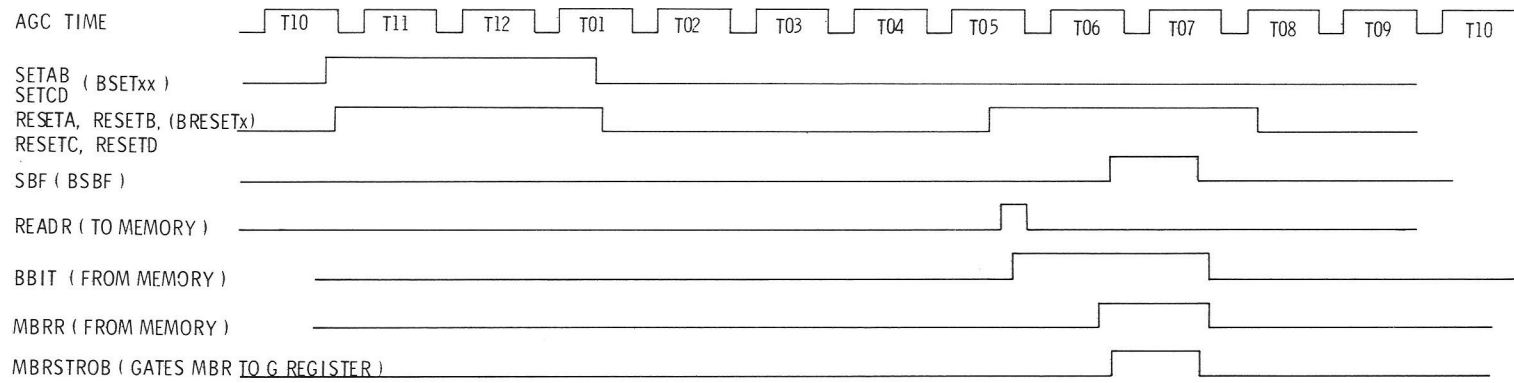


Fig. 2-1b AGC "Fixed" Memory Cycle

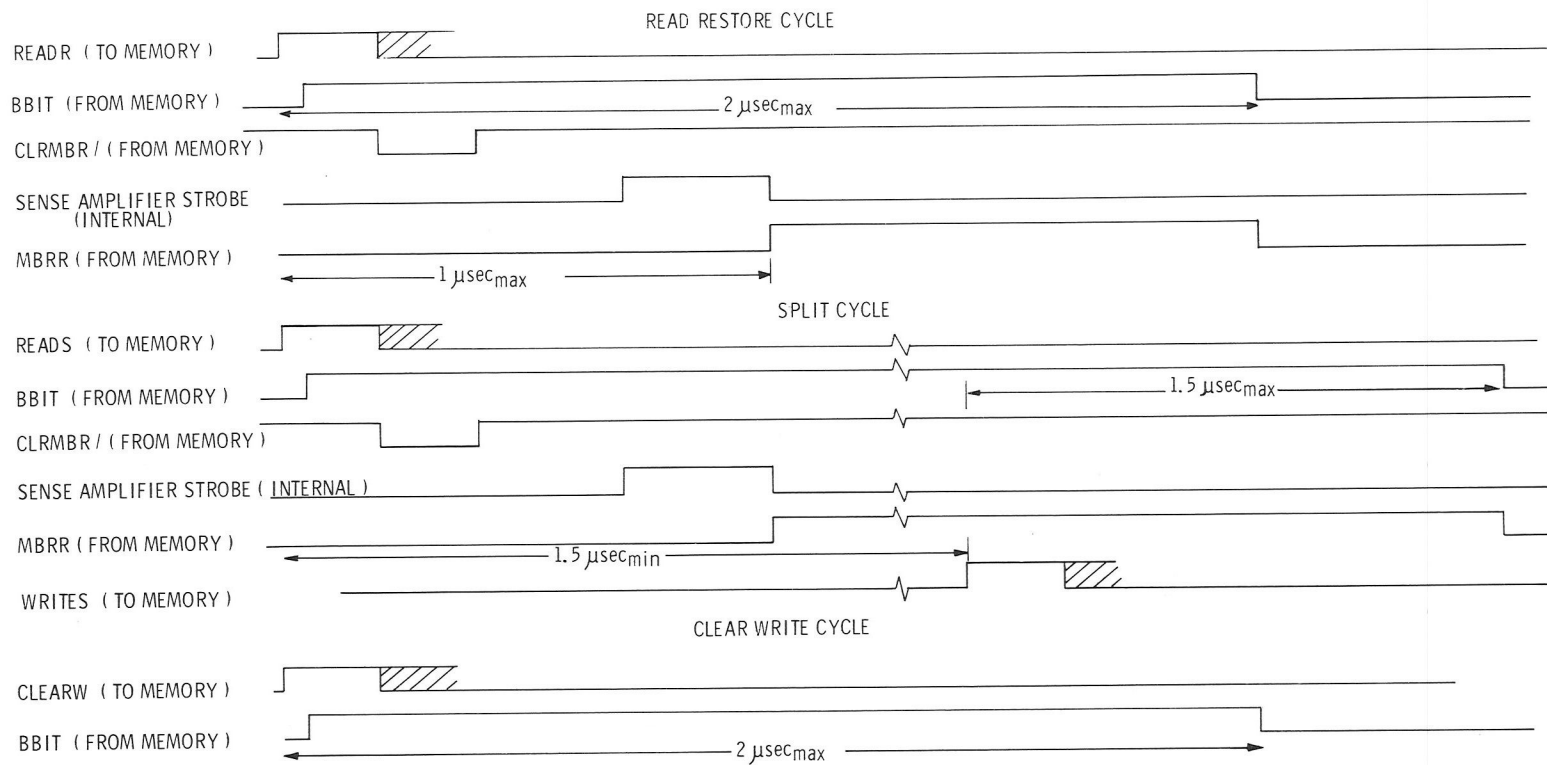


Fig. 2-1c Memory Interface Timing

### 3) External Memory Cycle Requests

If an external cycle request is present during time T12 of the AGC, an external cycle will begin immediately. Note that T12 occurs in every AGC memory cycle time (MCT) when the AGC is running, and occurs every 2 microseconds when the AGC is "stopped" via the Test Connector input MSTP. This feature guarantees availability of an external cycle every MCT and also allows external loading/reading when in the "stopped" state.

Incorporation of this feature will require a signal from tray A since T12 is not present in Tray B. Signal T12A (Net load -10) could be used or signal T12A/ could be generated utilizing an uncommitted "blue-nose" gate.

#### B. Memory Addressing and Buffering

There is no memory address register for the 16K E-Memory. All address signals will be present during the required times. These signals are generated using signals from Tray A, and the EEXT register in the case of AGC erasable cycles, or from the external address register (EXAD) mentioned below. The AGC signals are "mapped" into a "14-Bit Erasable Address" in accordance with Table 2-I.

A memory buffer register must be included to store data in preparation for transfer to the G register and to restore data during read-restore cycles.

#### C. External Interfacing

To economize on space, packaging, and components, all interfacing will be two-wire differential driver-receiver pairs using integrated drivers and receivers such as are now available. To minimize the overall component count (at the possible expense of added equipment in the external unit), serial transmission of data between the AGC and the external unit will be employed, substantially reducing the number of interface drivers and receivers needed as compared to parallel transmission.

Logic is provided in the proposed configuration to allow the incorporation of input-output expansion facilities with rather little additional equipment in the AGC Tray B. The I/O expansion scheme is based upon the dedication of 128 registers of erasable memory to I/O usage. The registers used are in EBANK 24, 1400-1577. The manner in which these registers are employed is to send a signal to the external unit (IOAD) when any of these registers is accessed. In this way, the external unit is notified when an output command is made and what its address is. On receipt of the IOAD signal the external equipment must generate 7 ADDRCLK pulses in order to strobe out the address. It must do this within 12 microseconds. The external unit must next do a read/restore cycle using this address. Figures 2-5, 2-6, and 2-8 contain the associated logic and timing.

This serves the function of the channel in the AGC, i.e., by proper implementation, these signals can be utilized as commands, etc.

#### Clear/Write Cycle

The external equipment can load information into the memory by loading register EXBR with data and EXAD with the address and generating a CYEXCW (cycle external clear/write sequence). When the AGC has completed this, it will generate a CYEXT signal.

#### Read/Restore Cycle

Similarly, the external equipment can read an AGC location by supplying an address in the EXAD register - generating a CYEXRR signal (cycle external read/restore signal). When the information is in the EXBR, a CYEXT signal is generated and the external equipment can shift it out.

The external interfacing will be via a connector on Tray B of the AGC.

Some 60 signals from Tray A are needed in the proposed E-Memory logic for addressing, timing, and data transfer. Additionally, 16 signals go from this logic back to Tray A for data transfer. All but a few of the signals are from unloaded expander gates or from expander gates loaded within the Tray B by rope-memory drive circuits. A few of the

signals are from normally connected nor-gates which are loaded by other gates in Tray A. These will be interfaced with expander gates in Tray B.

Figure 2-7 defines the interface between Tray A and the new Tray B.

Figure 2-1 shows the AGC memory-timing diagram and indicates that the 16K memory should have a 2- $\mu$ s cycle time.

Figure 2-2 is a block diagram showing how Figures 2-3 through 2-7 relate to the problem. Figure 2-3 implements the timing necessary to interface the memory with Tray A. Figure 2-4 shows the memory addressing and the extended erasable bank bits. Figures 2-5 and 2-6 define the interface with an external data adapter.

Figure 2-8 is a timing diagram for the external interface. It describes the three types of memory access - reading AGC, write AGC, and the reading of the external channel 24.

## 2.2 Mechanical Design of the 16K Tray B

The major part of this study is contained in a separate publication; Report E-2420, "Redesign of Tray B and Portafam Memory Modules To Provide the AGC with a 16K Erasable Memory". The results of this study indicate that it is feasible to modify the Tray-B and the Portafam-memory electronics to fit inside the present Tray-B envelope. A fair amount of redesign is required, especially in the Portafam power supply.

There is a significant number of nonflight-qualified components used in the Portafam electronics. The total number and types of components actually used in the Portafam memory are listed in E-2420. Part of the redesign would be to reduce the number of different types of components.

### 2.2.1 Portafam Memory Component List.

The following is a list of Portafam electronic components that do not have NASA drawing numbers (none are repeated). If found in more than one module, they are listed when first found.

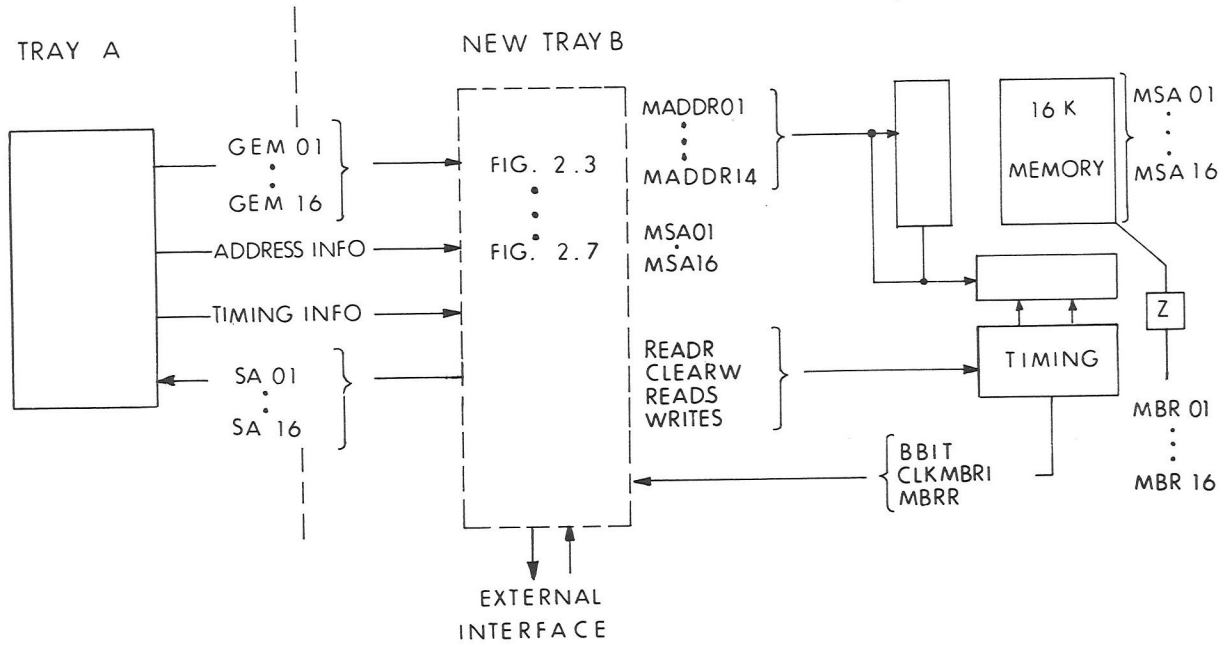


Fig. 2-2 Tray B Block Diagram



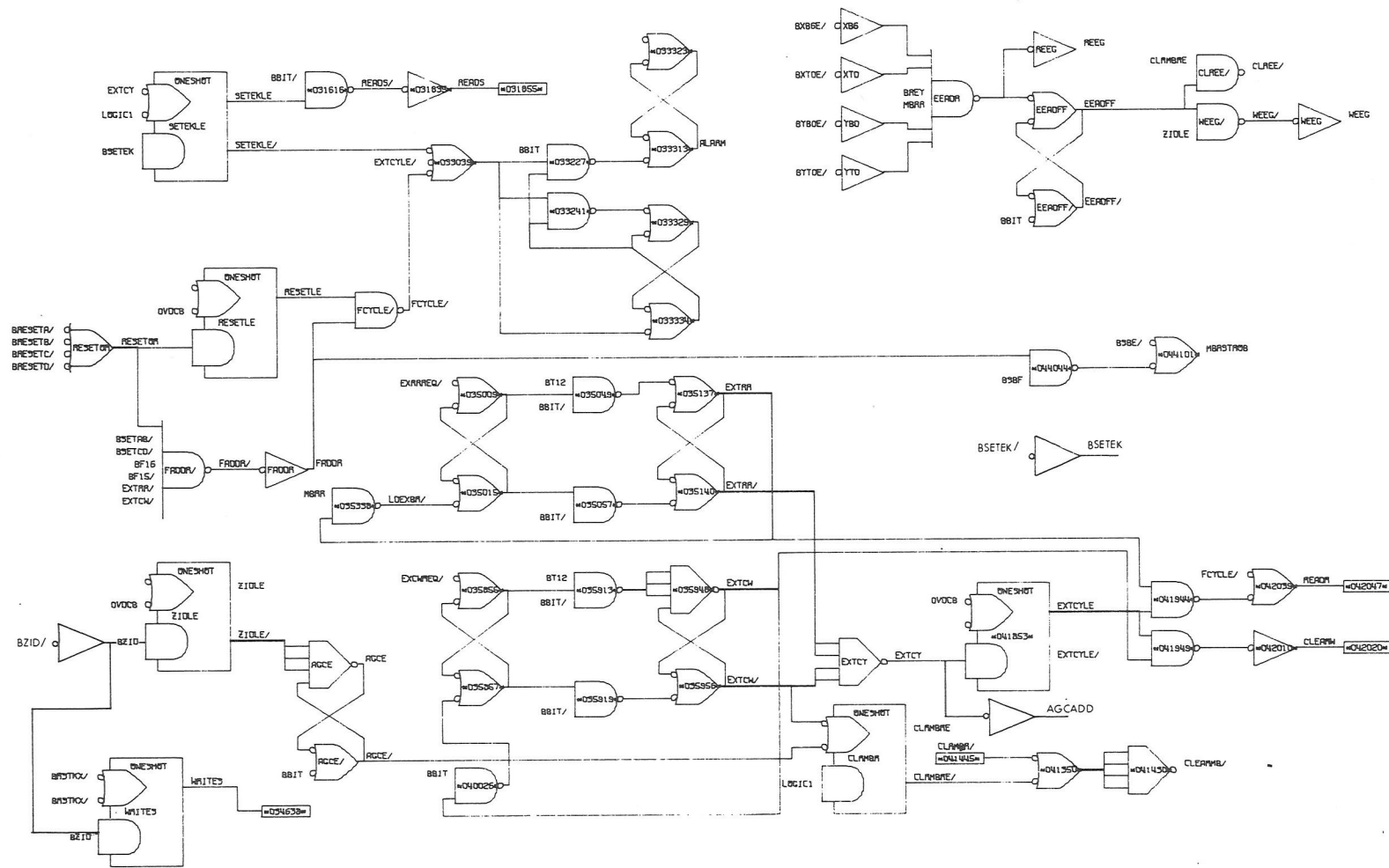


Fig. 2-3 Erasable Memory Control Pulses

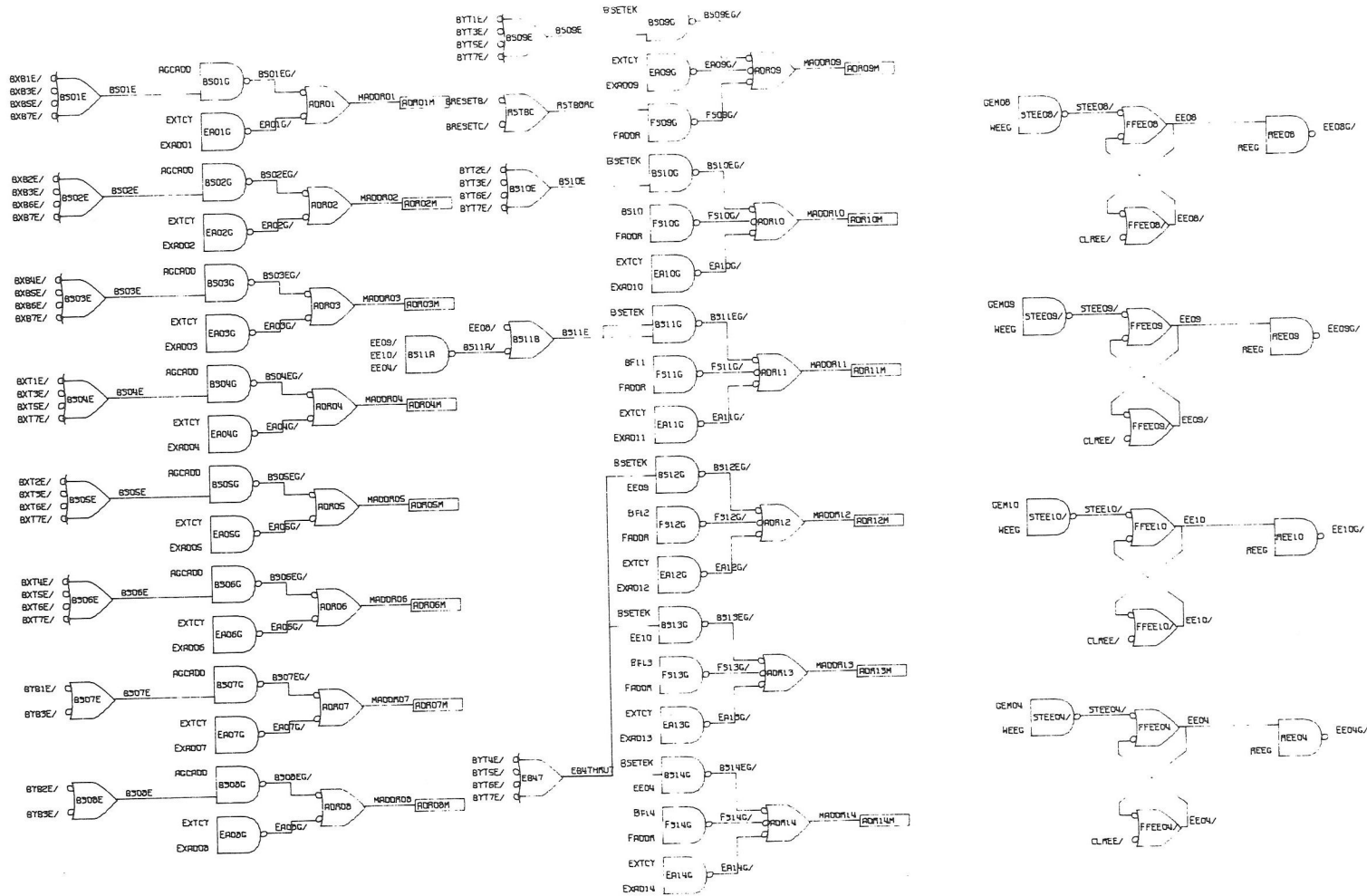


Fig. 2-4 Memory Addressing

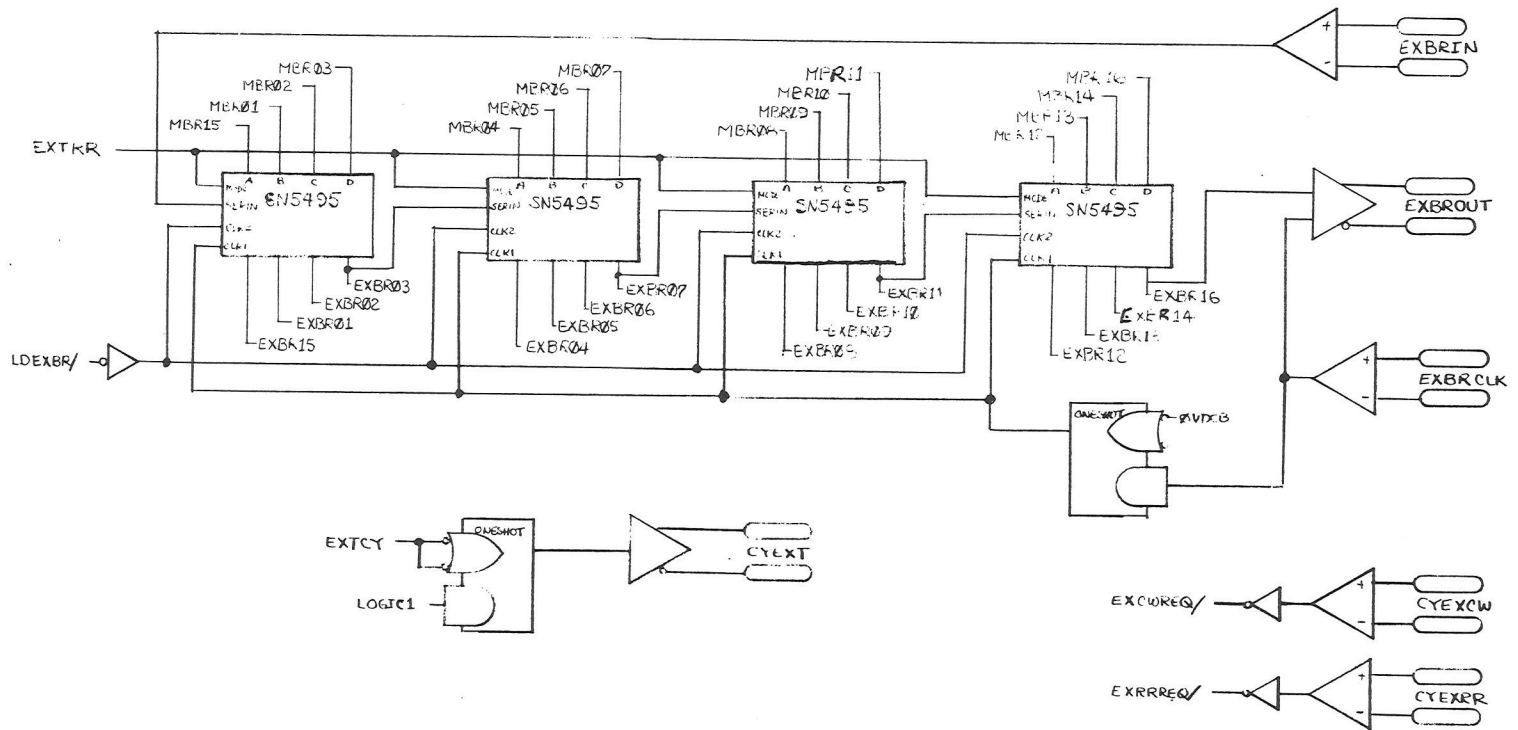


Fig. 2-5 External Buffer Register

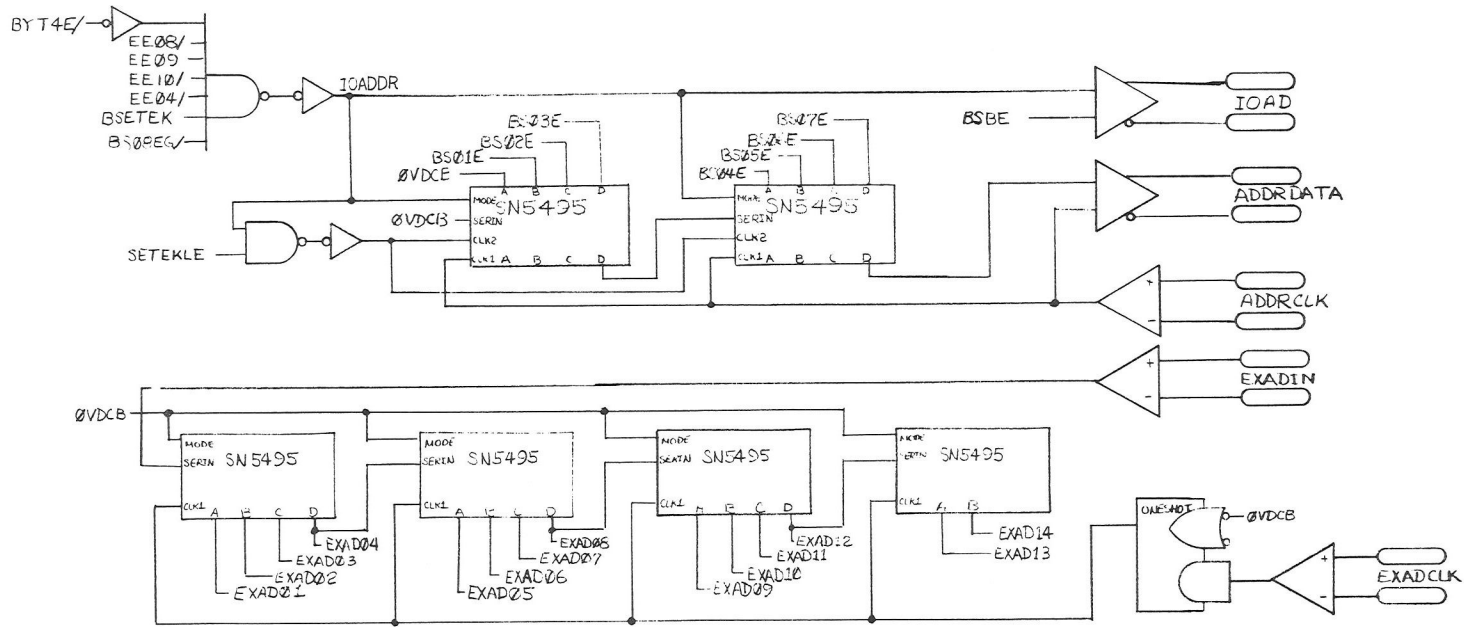
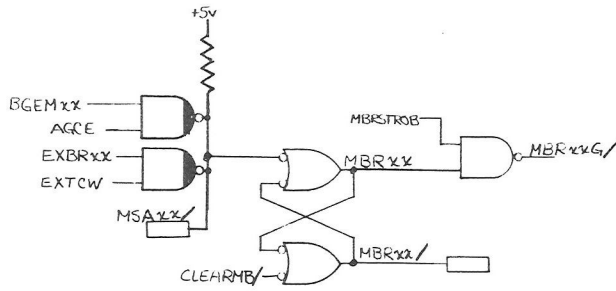
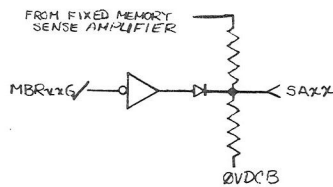


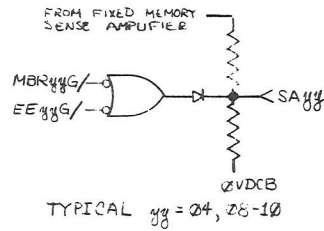
Fig. 2-6 External Address Interface



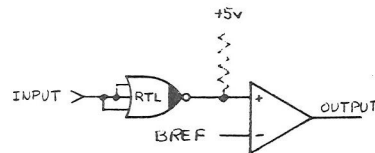
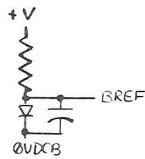
MEMORY BUFFER REGISTER  
TYPICAL xx = 01-16



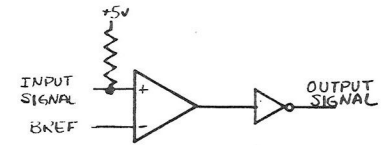
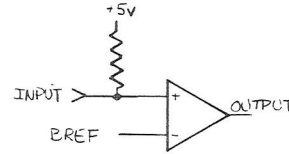
TYPICAL xx = 01-03, 05-07, 11-16  
NOTE: FOR xx = 15, OUTPUT IS SAP



TYPICAL yy = 04, 08-10

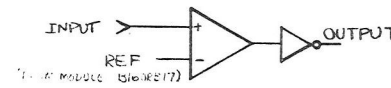


INPUT	OUTPUT
F11/	BF11
F12/	BF12
F13/	BF13
F14/	BF14
F15/	BF15/
F16/	BF16
S10/	BS10
T12A	BT12



GEM01	BGEM01
GEM02	BGEM02
GEM03	BGEM03
GEM04	BGEM04
GEM05	BGEM05
GEM06	BGEM06
GEM07	BGEM07
GEM08	BGEM08
GEM09	BGEM09
GEM10	BGEM10
GEM11	BGEM11
GEM12	BGEM12
GEM13	BGEM13
GEM14	BGEM14
GEM15	BGEM15
GEM16	BGEM16
RSTKX/	BRSTKX/
KEY	BKEY

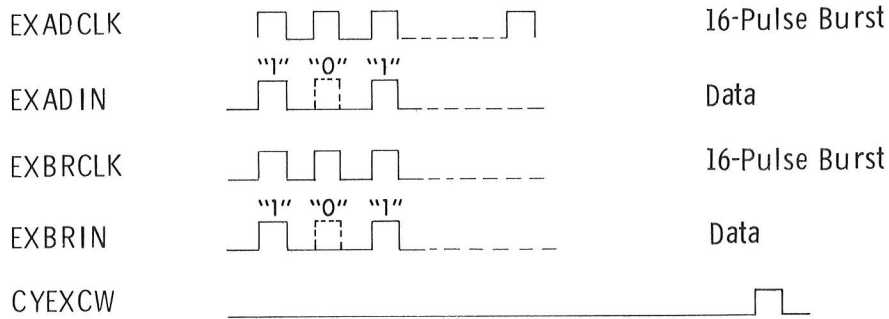
INPUT SIGNAL	OUTPUT SIG.
XB1E	BXB1E/
XB2E	BXB2E/
XB3E	BXB3E/
XB4E	BXB4E/
XB5E	BXB5E/
XB6E	BXB6E/
XB7E	BXB7E/
XT0E	BXT0E/
XT1E	BXT1E/
XT2E	BXT2E/
XT3E	BXT3E/
XT4E	BXT4E/
XT5E	BXT5E/
XT6E	BXT6E/
XT7E	BXT7E/
YB0E	BYB0E/
YB1E	BYB1E/
YB2E	BYB2E/
YB3E	BYB3E/
YT0E	BYT0E/
YT1E	BYT1E/
YT2E	BYT2E/
YT3E	BYT3E/
YT4E	BYT4E/
YT5E	BYT5E/
YT6E	BYT6E/
YT7E	BYT7E/
SBE	BSBE/
SETEK	BSETEK/
ZID	BZID/



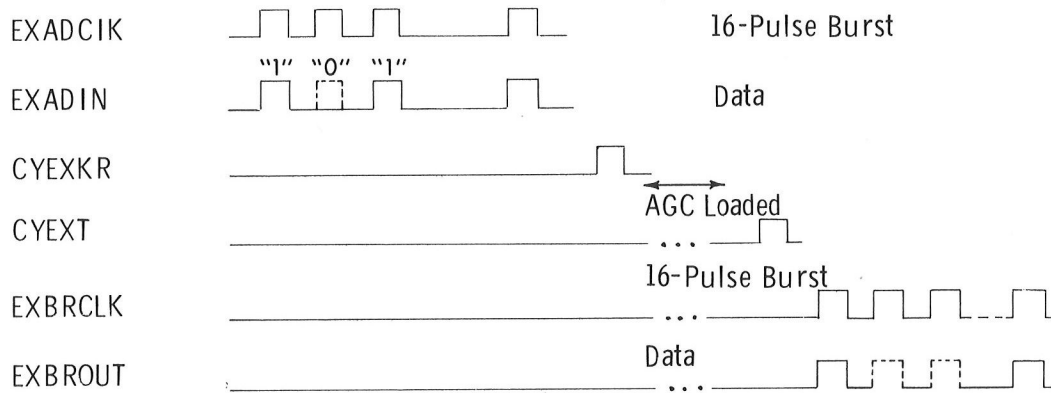
REF	INPUT	OUTPUT
40365A	SETAB	BSETAB/
40366A	SETCD	BSETCD/
40365A	RESETA	BRESETA/
40366A	RESETB	BRESETB/
40365A	RESETC	BRESETC/
40366A	RESETD	BRESETD/

Fig. 2-7 Tray A - Tray B Interface

EX LOAD OF AGC



EX READ OF AGC



EX READ OF BANK 24

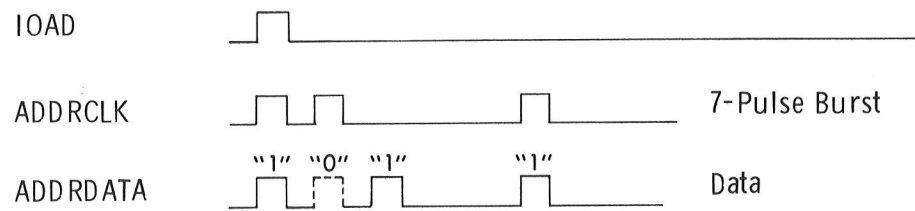


Fig. 2-8 External Interface Timing

#### ADDRESS SELECTOR

Decoder (flatpack) Fairchild 9301.  
Comp. transistor pair (flatpack) MOT SD2822F.  
Dual PNP (flatpack) MOT MD2904F.  
Diode Array (flatpack) Texas Inst. BC607.

#### SERVICE MODULE

Capacitor (Mallory) MTP685M060P1A.  
Capacitor (Mallory) MTP226M015P1A.  
Capacitor (Mallory) MTP106M030P1A.  
Capacitor (USCC) C22C561K  
Diodes 1N823A.

#### DIGIT DRIVER

Capacitor EMC-E05AT561K.

#### SEQUENCE GENERATOR CONTROL

Dual 4-input Nand buffer Sylvania SG130.  
Single 8-input Nand gate (HS) Sylvania SG260.  
And input J-K flip-flop Sylvania SF250.  
Quad 2-input Nand gate (HS) Sylvania SG220.  
Quad 2-input Nand gate T.I. SN5400.  
Triple 3-input Nand gate T.I. SN5410.  
Dual 4-input Nand buffer T.I. SN5440.  
Hex Inverter Fairchild 9016.  
Delay line Artronic type 22 A1211A.

#### COUNTER DECODER

Quad 2-input OR gate (high-speed) expandable Sylvania SG250.

#### HOLD

Quad 2-input Nand gate (high-speed) T.I. SN54H00.  
Dual 4-input Nand buffer (high-speed) T.I. SN54H40.  
Dual 4-input Nand gate (high-speed) T.I. SN54H20.

MAR II

Triple 3-input Nand gate T.I. SN54H10  
Triple 3-input And gate (high-speed) T.I. SN54H11.

MAR I

Dual J-K master-slave flip-flop T.I. SN5473.

MBR

Quad 2-input Nand gate (w. open coll.) T.I. SN5401.  
Quad 2-input Nor gate T.I. SN5402.

SENSE AMPLIFIER

Sense amplifier (open-collector output) T.I. SN7522.

### 2.2.2 Integrated Circuit Availability

This section indicates the availability in flat packs of certain integrated circuits (currently used in the memory section of Portafam) so that they could be considered for use in the redesign of Tray B.

The following is a listing of these integrated circuits. Column 1 assigns a number to these integrated circuits. Column 2 is the vendor number for these devices as used in Portafam. Column 3 is the vendor number for Mil Spec flatpack equivalent devices. Column 4 identifies their function.

For Portafam, 12 types of integrated circuits were supplied by Texas Instruments (IC-1 thru IC-12), 5 types by Sylvania (IC-13 thru IC-17), and 2 types by Fairchild (IC-18 and IC-19).



<u>Integrated Circuit</u>	<u>Vendor No. Pfm. IC</u>	<u>Vendor No. Mil Flatpack</u>	<u>Function</u>
IC-1	SN7400N	SN5400F	Quadruple 2-input positive Nand gate.
IC-2	SN7401N	SN5401F	Quadruple 2-input with open-collector output.
IC-3	SN7402N	SN5402F	Quadruple 2-input positive Nor gate.
IC-4	SN7410N	SN5410F	Triple 3-input positive Nand gate.
IC-5	SN7440N	SN5440F	Dual 4-input positive Nand buffer.
IC-6	SN74H00N	SN54H00F	Quadruple 2-input positive Nand gate (H).
IC-7	SN74H10N	SN54H10F	Triple 3-input positive Nand gate (H).
IC-8	SN74H11N	SN54H11F	Triple 3-input positive And gate (H).
IC-9	SN74H20N	SN54H20F	Dual 4-input positive Nand gate (H).
IC-10	SN74H40N	SN54H40F	Dual 4-input positive Nand buffer (H).
IC-11	SN7473N	SN5473F	Dual J-K master-slave flip-flop.
IC-12	SN7522N		Sense amplifier (open-collector output).
IC-13	SG130-03	SG130-02	Dual 4-input line driver.
IC-14	SG220-03	SG220-02	Quadruple 2-input Nand/Nor gate.
IC-15	SG250-03	SG250-02	Expandable Quad 2-input OR gate.
IC-16	SG263-03	SG260-02	Single 8-input Nand/Nor gate.
IC-17	SF253-03	SF250-02	And input J-K flip-flop.
IC-18	CCSL9301		MS1 One-of-ten decoder.
IC-19	TTOL9016	TTOL9016	Hex inverter.

## CHAPTER 3

### BRAID MEMORY

#### 3.1 Introduction

The Braid Memory<sup>(2)</sup> is a transformer-type read-only memory which has been under development at MIT for several years. Information is written permanently at manufacture by weaving a wire bundle ("braid") through an array of sensing transformers. The transformers are made from two-piece (U-I) linear ferrite cores after the wire bundle is formed and potted, allowing fabrication without actual threading of cores as must be done in core ropes. Manufacture may thus be accomplished more quickly; MIT uses a modified Jacquard loom handling 256 wires simultaneously to weave the braid and to assist in making wire terminations, and a million-bit Braid Memory may be woven and fully terminated in six regular (8-hour) working days.

A number of memories of the million-bit "Easy Boat" configuration have been built; this configuration exhibits a bit density of 2300 bits per cubic inch (including all electronics) with a full cycle time of 4 microseconds and a power consumption of 12 watts.

Earlier investigations<sup>(3)</sup> indicated that a high-performance, high-density Braid Memory could be built for the aerospace environment. Such a memory could exhibit a bit density at least twice that of previous braids, a speed perhaps ten times as great, and reduced power consumption, while employing mechanical refinements such as replaceable information planes.

A study has been conducted to determine the feasibility of producing such a high-performance Braid Memory in a configuration dubbed "New Boat". The study included verifying operation of the electronic and magnetic circuitry involved and demonstration of the mechanical feasibility of weaving a braid in the smaller dimensions required. Aspects of this study are described below.

### 3.2 Organization

The New Boat Braid Memory is a half-million-bit (524,288) module containing complete electronics, with input (address) and output registers. The first arrangement is for storage of 32,768 words of 16 bits for compatibility with the AGC; two modules could form a 65K fixed memory that would fit in the space now occupied by the rope modules and "baby tray" in a redesigned Tray B.

The information field consists of two independently woven and potted planes containing 512 wires each, and a 512-core transformer array. (See Fig. 3-1.) Each plane is driven by a 512-transistor array using 64 flat packages. The transistors are electrically arranged in a matrix with orthogonally based bases and emitters for simplified selection; the undriven common connection to the braid bundle is returned to a positive voltage (14 volts).

Selection of 16 sense positions, from the 512 used, is done by a high-speed analog multiplexer using N-channel J-FETs (Fig. 3-2). This scheme permits the use of low-impedance sense circuitry, which in turn allows reduced drive current while using sense windings of less than ten turns (for comparison, Easy Boat uses 60-turn bobbin-mounted windings).

Commercially available high-speed comparators are used as sense amplifiers, and output data are stored in a 16-bit register to avoid transmission of the narrow (approx. 50-ns) strobed outputs.

Reading a single word from storage involves selecting a single word line (see Section 3.3.1) and a single 16-bit sense group (see Section 3.3.2). A single current driver serves as a common source for all word lines.

### 3.3 Electronics

The New Boat electronics is arranged to provide improvements in speed and power consumption while permitting higher-density packaging in a configuration that might be easily flight qualified. The design is for a well-rounded system with speed, density, and power consumption all considered of equal importance; discrete circuitry has been minimized, and is used only where available integrated devices won't do the job.

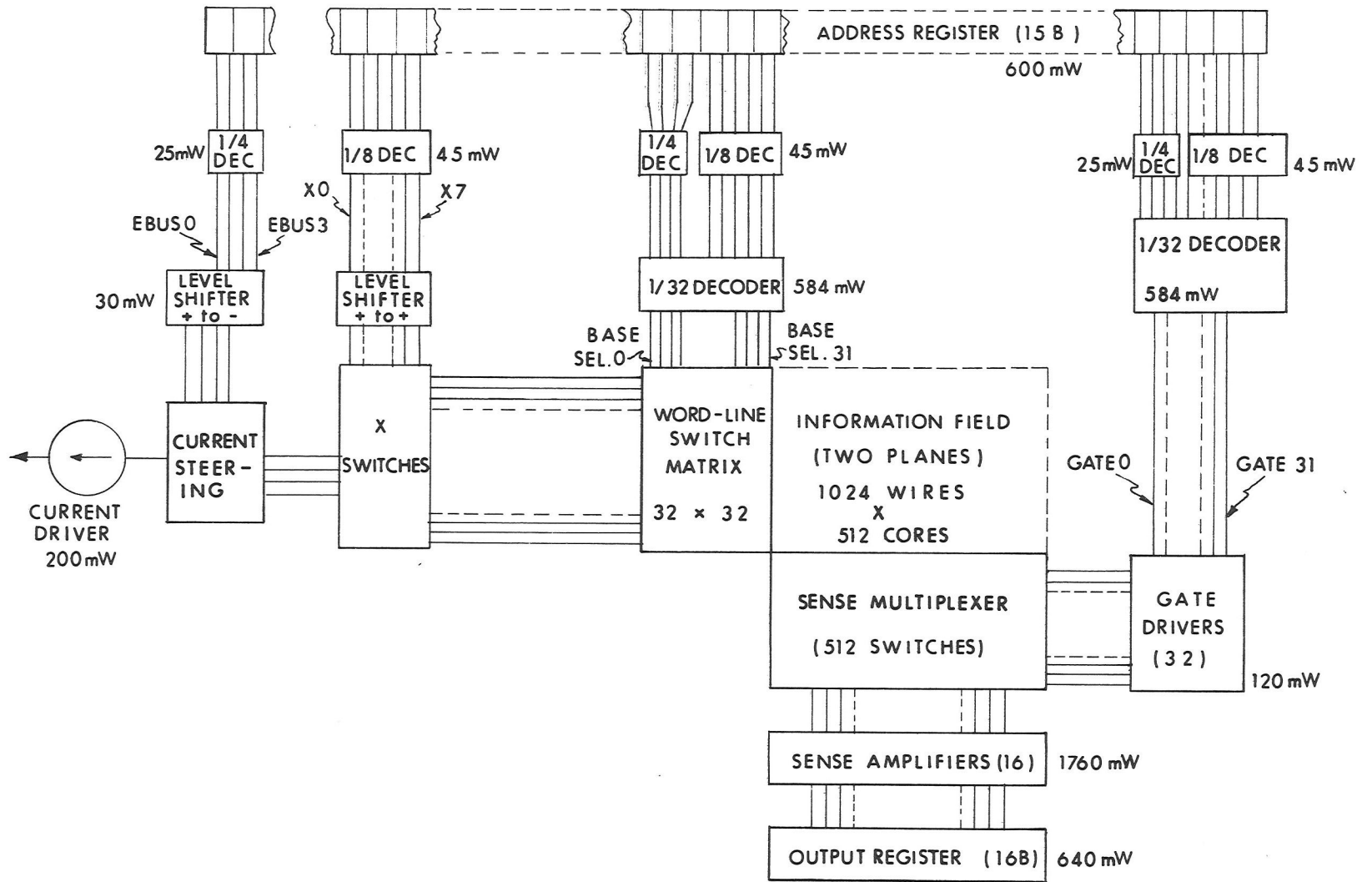


Fig. 3-1 New Boat Organization

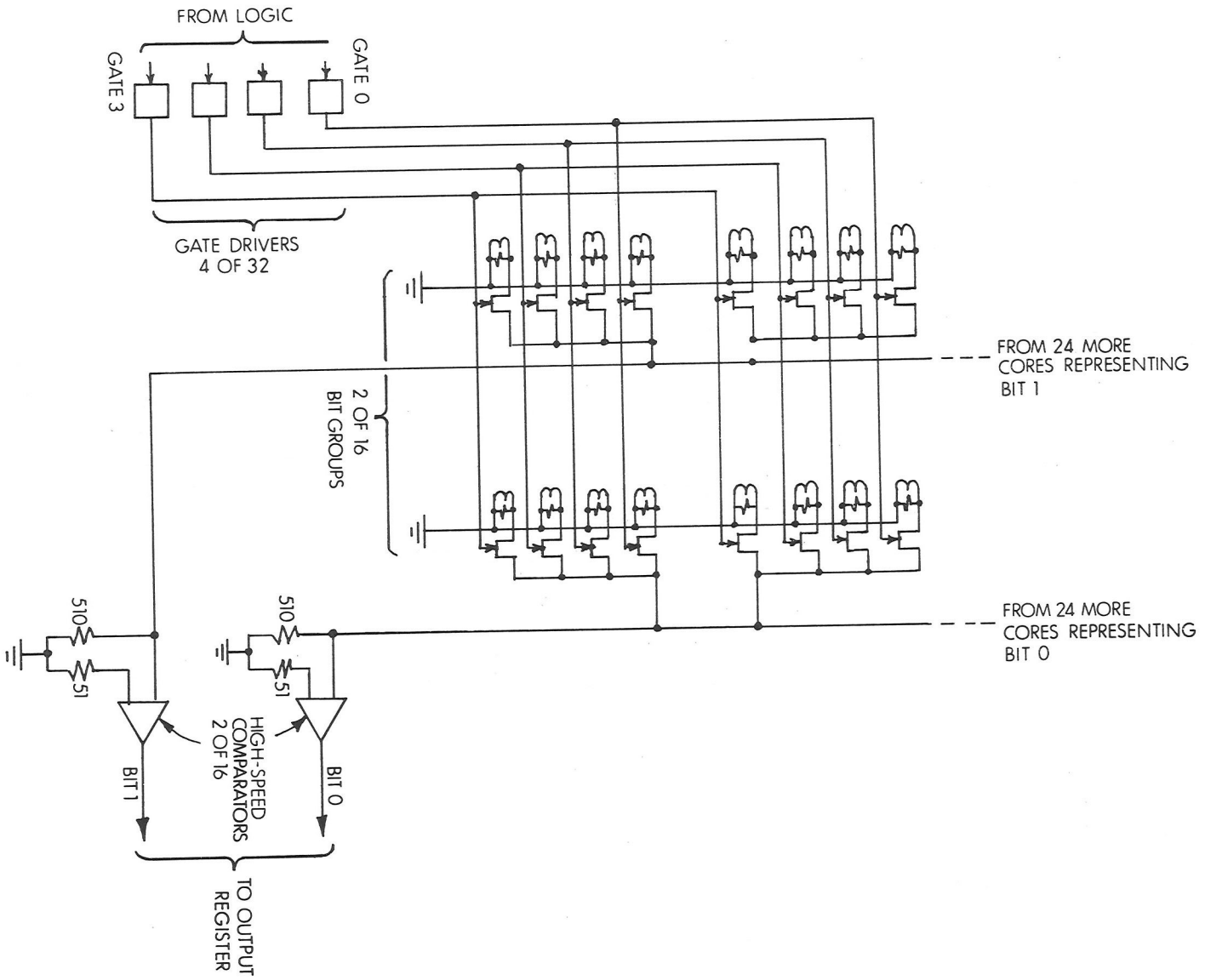


Fig. 3-2 Sense Multiplexer

### 3.3.1 Word-Line Selection

The basic device used in word-line selection is an eight-transistor flat pack with internal base and emitter busing (see Fig. 3-3). Sixty-four such packages drive a single plane, with a single collector used to drive each word line. This scheme provides greatly increased speed over the double-ended diode-per-line scheme at little cost. The speed improvement comes from not having to change bundle potentials during selection and hence not having to charge and discharge associated capacitances; in earlier high-capacity braids, most of the 3- to 5-microsecond cycle time is spent waiting for noise currents generated on selection to decay to an acceptable value.

The word-line selection circuitry is shown in reduced form in Fig. 3-4. One drive-matrix base bus is selected by applying a positive voltage from a 1-out-of-32 decoder (TTL); one X switch (four transistors) is enabled, and one of four current-steering switches is turned on. This establishes a path through a single word line for current from the current driver (Fig. 3-5). Note that the driver provides a reverse current to drive a special bias wire threading all cores; this is done to provide a positive core output for a logical "one" and a negative core output for a logical "zero".

Experiments indicate that New Boat will operate satisfactorily with the current driver set to 40 mA; this is a net current of + 20mA for a "one" and -20 mA for a zero.

### 3.3.2 Sensing

The New Boat braid uses cores of much smaller dimensions than those used earlier (see Fig. 3-6). Small cores are used for density considerations; previous studies<sup>(3)</sup> showed that, for a given minimum workable wire size and given fixed volume for electronics, there is a quantity of wires (hence core size) that results in optimum density. Preliminary calculations indicated that the number of wires was about a thousand, hence the choice of 1024 for New Boat. However, since a number of compromises were made for mechanical refinements, it cannot be said with certainty how close New Boat is to the optimum.

#### 3.3.2.1 Sense-Position Response

The magnetic material used for the cores is the same as that used previously, and the electrical equivalent for the core is about the

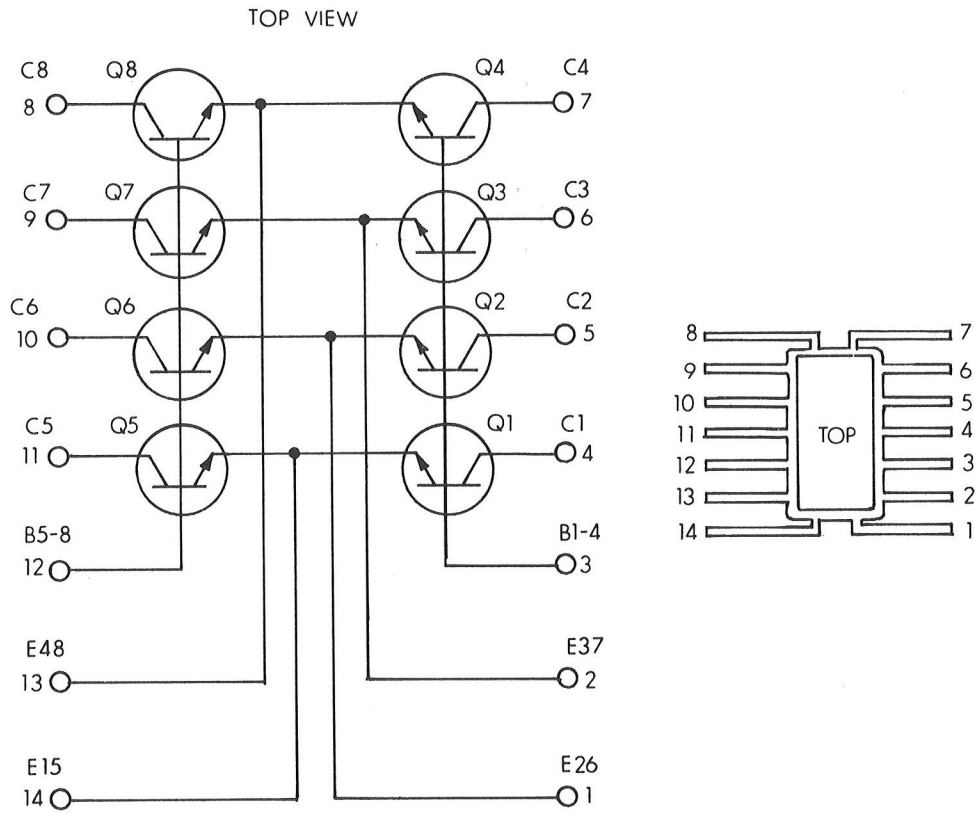


Fig. 3-3 Word-Line Switch Matrix Flat Pack

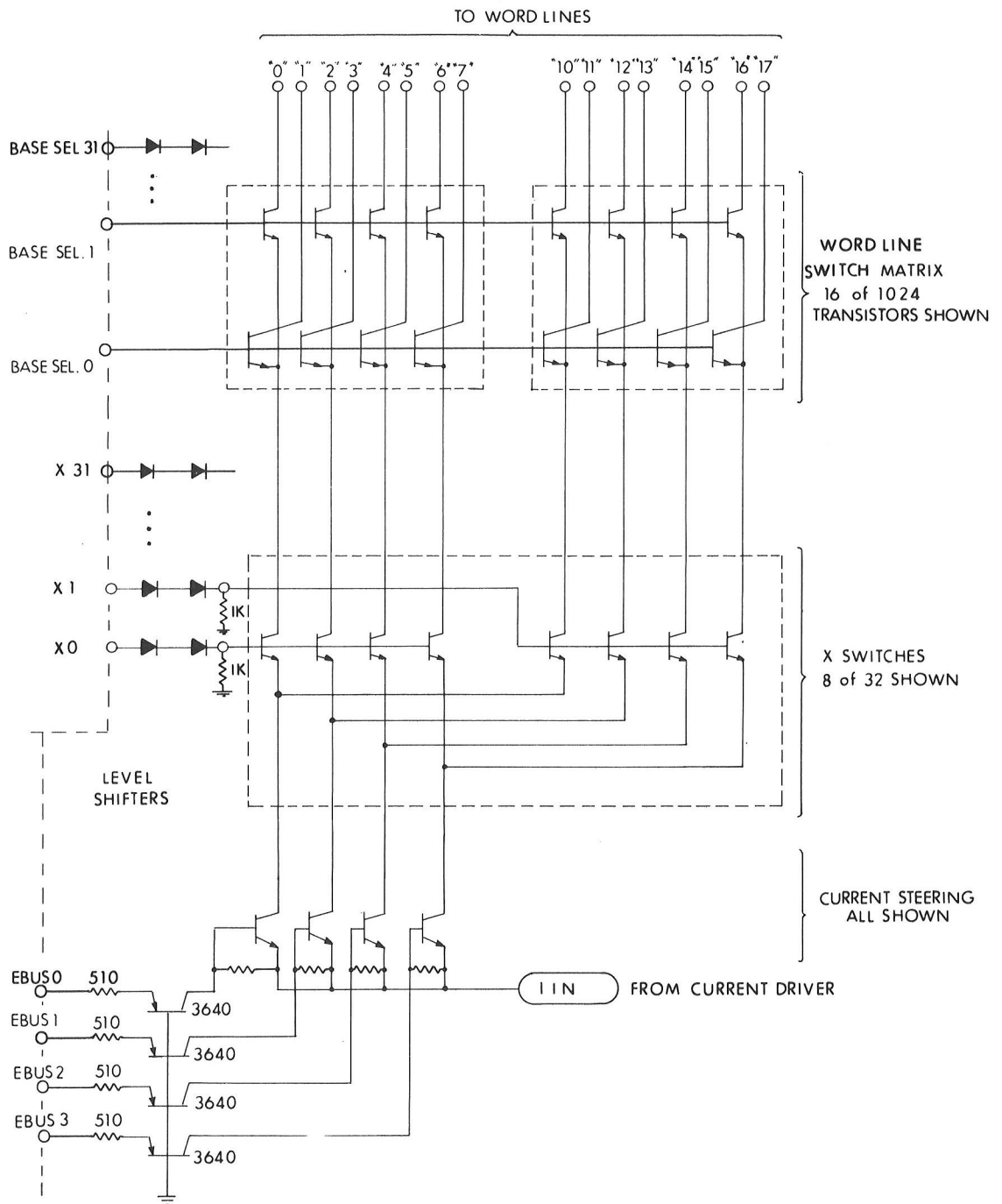


Fig. 3-4 Word-Line Selection Circuitry



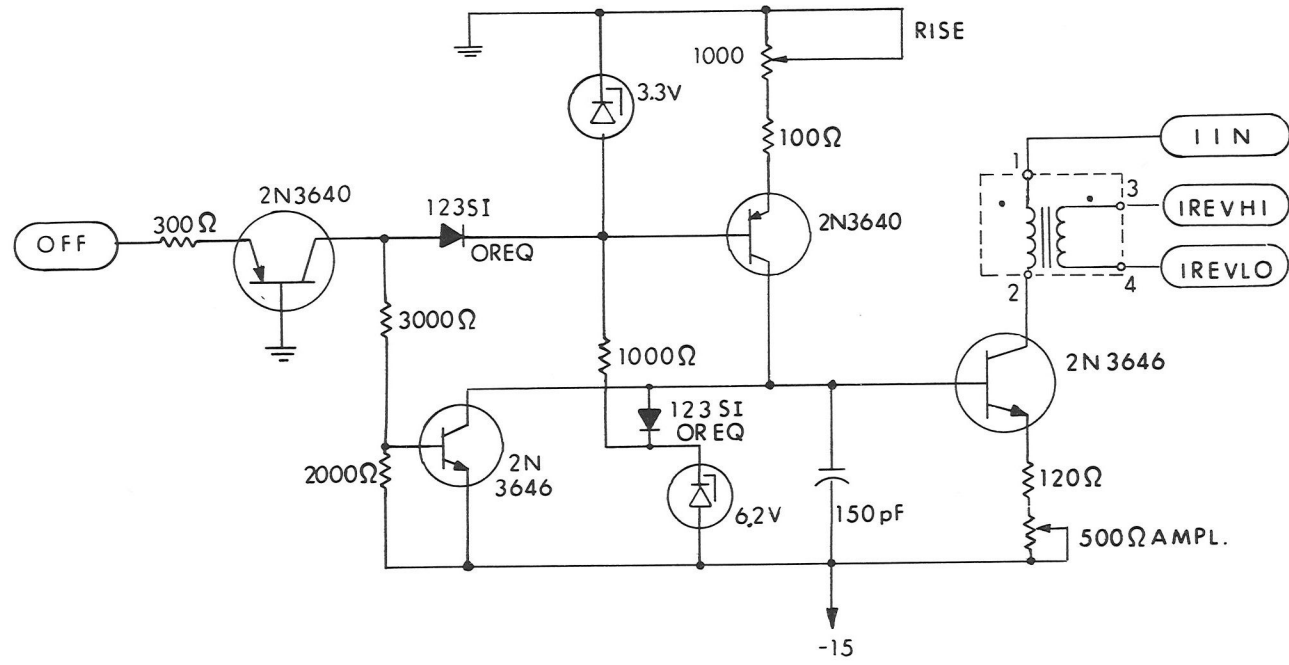
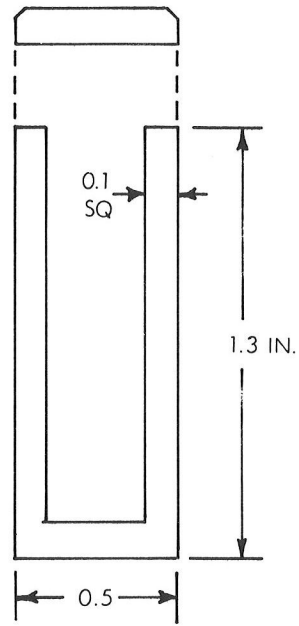
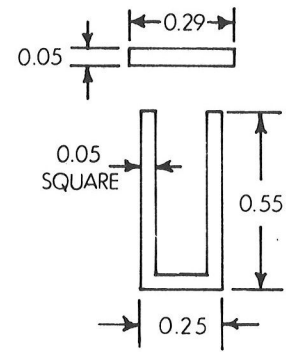


Fig. 3-5 Current Driver



(a) Easy Boat



(b) New Boat

Fig. 3-6 Transformer Cores

same with the core exhibiting a single-turn magnetizing inductance of 0.2 microhenry and core-loss equivalent resistance of about five ohms.

A simplified equivalent circuit appears in Fig. 3-7b.

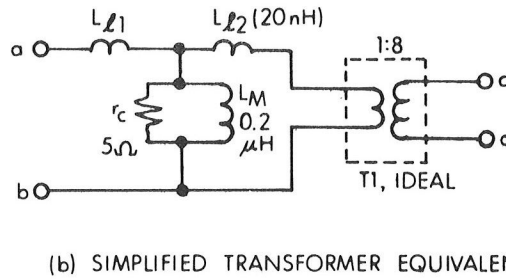
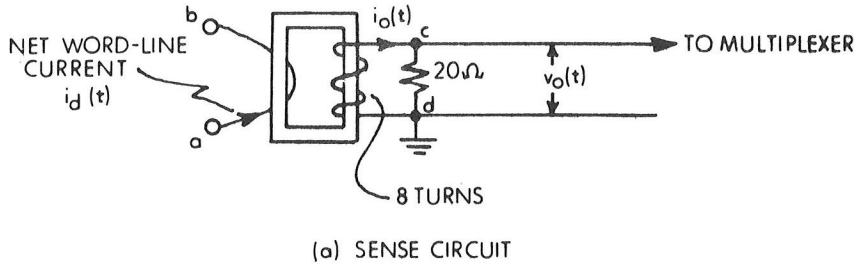


Fig. 3-7

For a drive-current step of magnitude  $I$ , the output current at time  $t$  is

$$i_o(t) = \frac{\frac{I}{N}}{1 + \frac{L_{l0}}{L_m}} \exp\left(-\frac{tR}{N^2 L_m \left(1 + \frac{L_{l0}}{L_m}\right)}\right)$$

where  $L_{l2} = N^2 L_{l0}$

$$I = \pm 20 \text{ mA}$$

$$i_o(t) = \pm 2.27 \text{ mA} - \frac{t}{0.7 \mu\text{sec}}$$

The output voltage is then

$$v_o(t) = (20\Omega) i_o(t) = \pm 45.4 \text{ mV} - \frac{t}{0.7 \mu\text{sec}}$$

The associated waveforms are shown in Fig. 3.8.

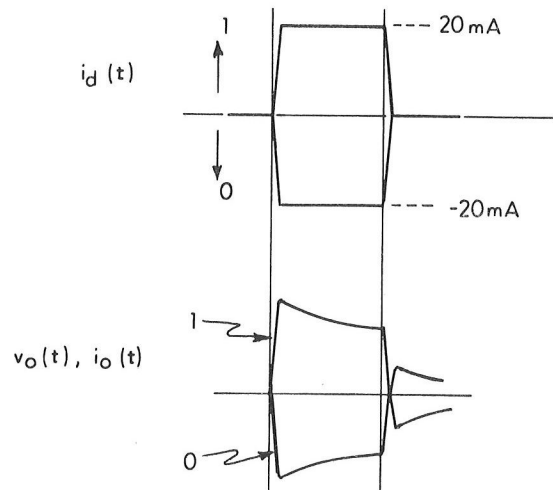


Fig. 3-8 V-I Relationship in Basic Sense Circuit

### 3.3.2.2 Sense Multiplexer

A high-speed analog multiplexer using N-channel junction FETs is used to steer core outputs to a single bank of sense amplifiers. Since the core outputs are positive for a "one" and negative for a "zero", thresholding need be done (the threshold is zero volts) and comparators may be used as sense amplifiers.

The single flatpack contains four FET switches (see Fig. 3-9). A single transistor exhibits a maximum "on" resistance of 45 ohms and a maximum "off" leakage current of 100 pA at room temperature;  $C_{dg}$  and  $C_{gs}$  are both less than 7 pf, and feedthrough can be kept within bounds by driving the gate with a voltage source.

A portion of the multiplexer is shown in Fig. 3-10. Push-pull drivers are used to switch the primarily capacitive gate lines while dissipating little standby power; these drivers are discrete because their integrated circuit counterparts are too slow.

## 3.4 Mechanical Design

To prove the feasibility of the New Boat mechanical design and to attempt to uncover any weaving or potting problems that might occur during fabrication, a fixture was built to attempt to weave and to pot a section of a braid plane. New

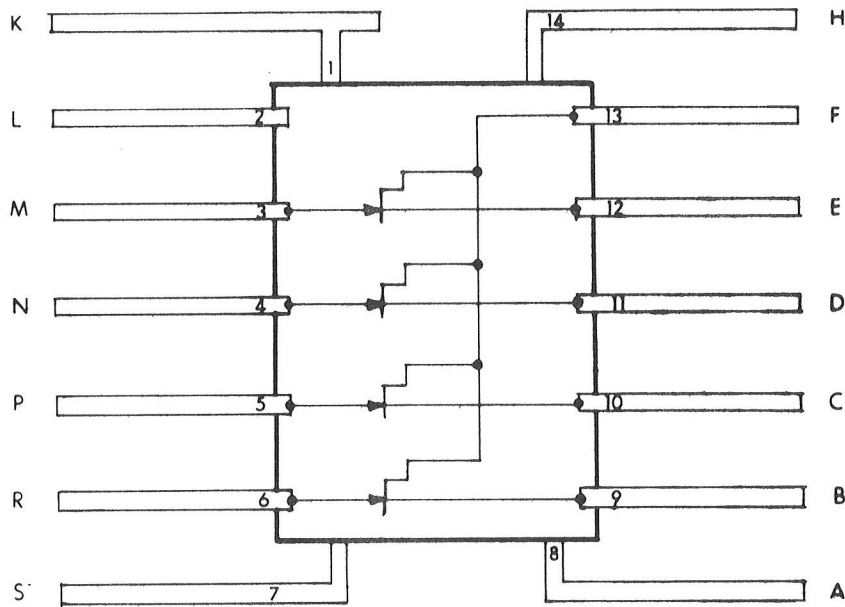


Fig. 3-9 FET Package

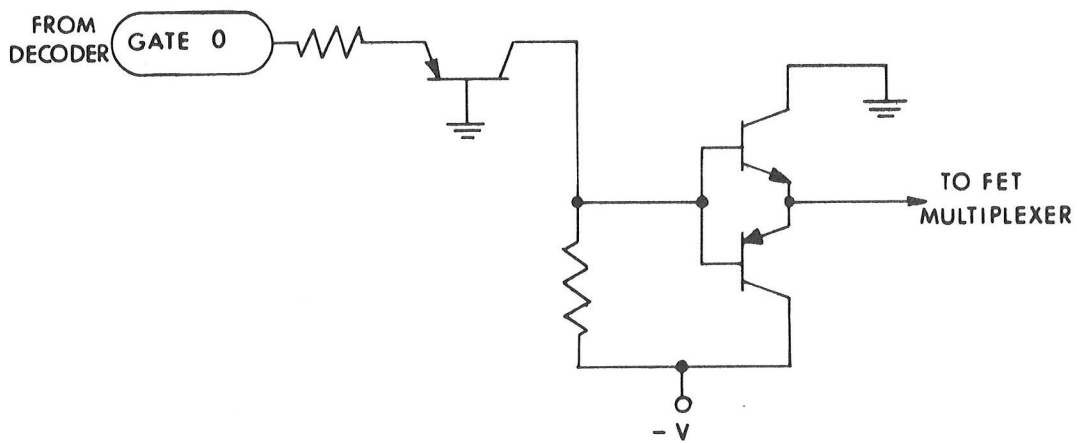


Fig. 3-10 Gate Driver

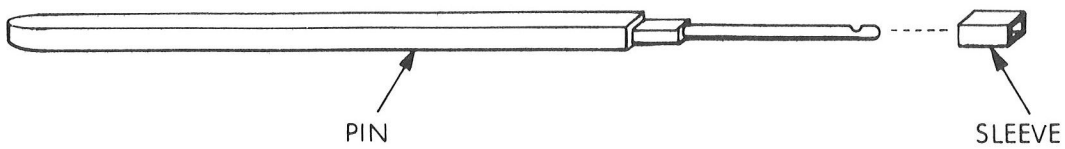


Fig. 3-11 Separator Pins

Boat cores are placed on 0.150" centers, and square holes are used for the cores, as this permits more space between the cores for wires to change from the "0" to "1" or "1" to "0" configuration; the braid itself is to be fully protected by laminates and by potted-in protective sleeves around the core legs. These were the main factors dictating the shape of the pins (temporary separators) to be used (see Fig. 3-11).

Since the potted holes were to be square, some means had to be devised to keep the pins in proper alignment during the weaving process. This was accomplished by hollowing out a portion of the base molding plate and by notching the ends of the pins that protruded into this area. The fixture is assembled by placing protective sleeves on the pins and then inserting each pin through a fiberglass bottom laminate and through round holes in the base molding plate. When all pins have been inserted, they are aligned by means of spacer bars; the fixture is then inverted and "CERRO" metal with a melting point of 140<sup>o</sup>F is melted into the cavity and allowed to flow around the notched pin tips. The metal, when melted, cooled, and hardened, holds all the pins in line and stationary. The braid frame is then fastened to the base molding plate. All parts are sprayed with mold release before assembly.

#### 3.4.1 First Sample Weave

Weaving was accomplished in basically the same manner as other braids. The temporary separators used to fit on top of the pins were square fiber-glass tubing 4" long. Wire handling was made somewhat more difficult because the separators are shorter than the 8" wire spread at the loom's comb. In addition, the closeness of the pins made it more difficult to separate the wires and insert a temporary separator on each pin. An attempt to spread the wires by hand in this area resulted in detrimental slackening of wires. It is felt that this problem might be eliminated by changes in the spool assembly of the loom.

Another problem occurred because no provision had been made to use turning pins on the end of each row of pins. (Turning pins are used to hold the wire bundle while the braid is turned 180<sup>o</sup> so that the next row may be woven.) Temporary separators were installed at each turning point to keep the wires in proper position for braiding and to help gain the slack necessary to pack the wires into the plane. Because of the small cross section of the pins, the tension on the wires caused the temporary separators to fan out (bend) in the area above the pins. This effect was most pronounced in the spacing area between each group of eight cores. Spacers were inserted in

each of these areas to alleviate this problem and to gain additional slack. After weaving, a second (top) fiber-glass laminate was placed on the top of the braid to protect the braid and to prepare for potting.

Potting was done with Stycast 1266 (the same material used in earlier braids) and no special problems arose; however, taking the mold apart proved to be quite difficult. After the potting had cured, the fixture was placed in the oven to post-cure at 200<sup>o</sup>F, and a pan was placed under the mold to catch the "CERRO" metal as it melted. The post-cure was completed and the "CERRO" metal dropped out as expected. At this point the pins should have been removable; however, removal of these pins proved to be a major problem. The potting compound had seeped between the pins and the mold baseplate making pin removal impossible without ruining the pins. Where the pins were hammered out, they bent or broke at the notch and, when they were finally removed, they took the sleeving out with them in 50% of the cases.

#### 3.4.2 Second Sample Weave

The first sample weave had demonstrated that weaving could indeed be done in the dimensions required. Improvements were made in the weaving fixtures, and a second sample weave was made.

The length of the pins was reduced to help cut down on the spreading or fanning-out problems, and the depth of the notch on the pin tip was reduced to add strength in this area. A new baseplate and a square-hole hold-down plate were made. (This latter was omitted in the first attempt.) All parts that were to be exposed to the potting compound were teflon-coated and all were coated with potting grease during assembly.

The construction of the braid progressed as before with the same basic difficulties (i.e., bending at the space between groups and the end-turning problems). Potting was no problem; following post cure the fixture was disassembled without much difficulty and with no breakage (see Fig. 3-12).

While the second attempt was oriented more towards solving the fixture and potting problems than towards the weaving problem, it is felt that with a few more basic aids the weaving could also be made easier. Special spacers should be made and installed between the weaving of each group of eight bits to help maintain the slack necessary in the wires. Since there are still no

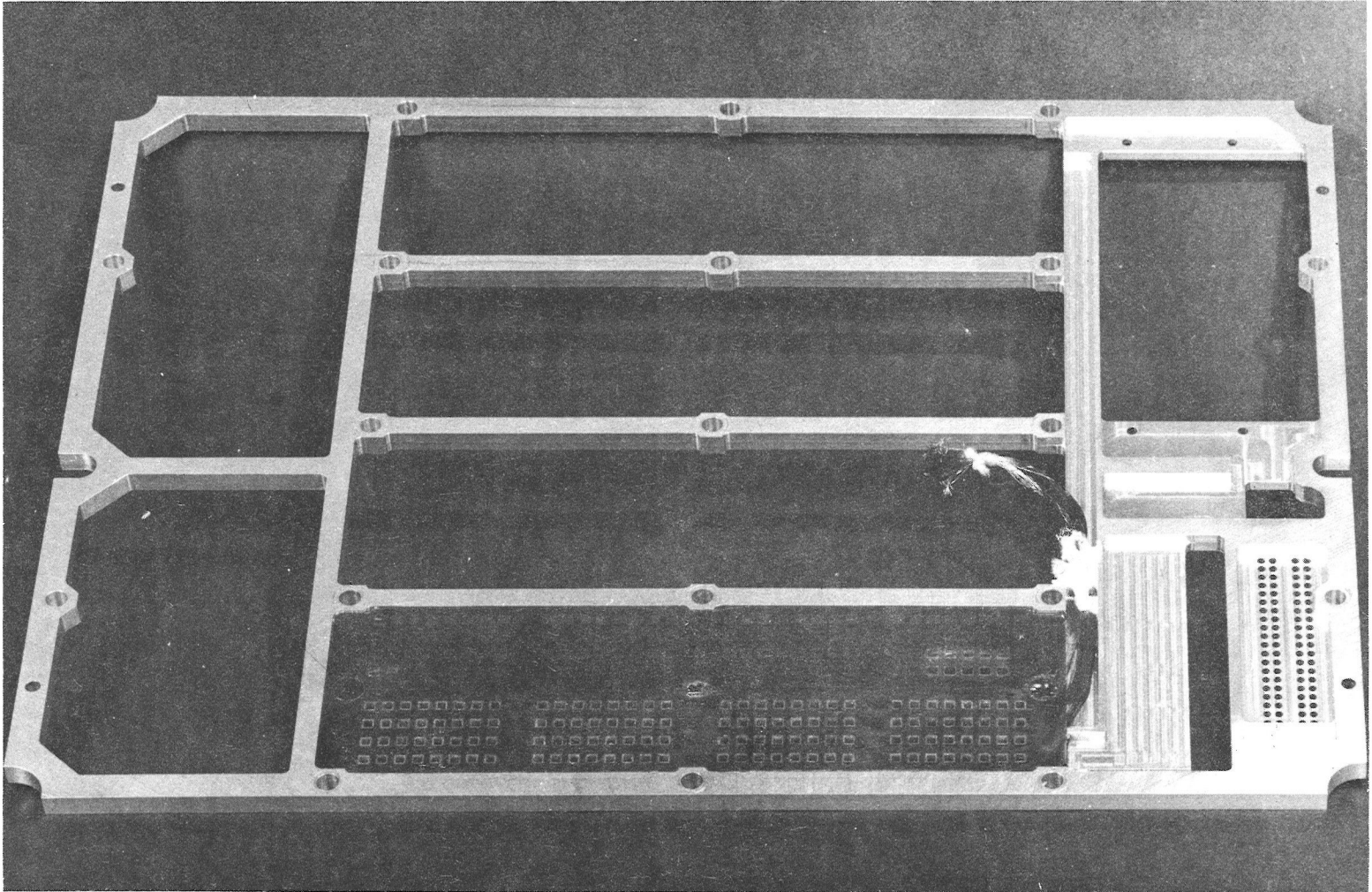


Fig. 3-12 Second Sample Weave



turning pins at the end of each row, a special rod, possibly round, could be inserted to keep all the wires in line when making a  $180^{\circ}$  turn.

Changes to the potting frame are recommended. The sharp edges at each of the assembly screw holes should be removed, and the groove at the ends where the wires run from one quarter-section to another should be cut deeper to accommodate the wires more readily. With these changes and perhaps some innovations in the loom (for tensioning), it is felt that weaving a braid of this density is indeed feasible.

### 3.4.3 Terminations

It should also be noted that weaving a braid of this density required the termination of wires be doubled in density, from 0.100" to 0.050"; with special tooling, a termination attempt was made using the present soldering method. The termination was successful but there are special handling problems that must also be solved since the overall assembly of the braid is still not defined at this time.

The effort on the Braid Memory was terminated at this point because of a redirection of effort based on a decision not to modify current Apollo equipment for post-Apollo applications.

## CHAPTER 4

### PROGRAM EVALUATION

#### 4.1 Auxiliary Memory

The primary tool used to evaluate the impact of the additional memory on mission program was the auxiliary memory. Figure 4-1 is a sketch showing how the AM and the AGC functionally tie together.

Essentially, this prototype unit has 8K of erasable memory (expandable to 16K) addressable as erasable banks 10-37 and 100-107. The 8K can also be addressed as fixed banks 60-67. One can write from the upper 4K onto the tape and can read from the tape to any of the 8K.

Some modifications were made to the prototype unit with regard to the Both Bank Structure. The original equipment permitted Both Bank bits 1-7 to be written into and read from Ebank bits 9-15; the equipment was modified so that Both Bank bits 1-4, 8-10 were read and written into instead of 1-7. This was done so that the Superbank bits (FEXT) could be put into bits 5-7 maintaining compatibility with the present APOLLO programs.

Figure 4-2 shows the prototype as packaged for testing at MIT/IL. The tape memory is shown at the top with the front panel opposite.

Two of the 4K memory modules are shown directly below it with gaps for two additional modules. The memory-drive electronics, the logic, and the power supply complete the rack.

Figure 4-3 shows the control panel and Fig. 6-1 shows the interface-module cabling and electronics. The interface module connects to the AGC test connector which is not used during a mission.

The AM receives commands by having the program load out-channels in the AM through the write lines. It then proceeds to execute these commands independently

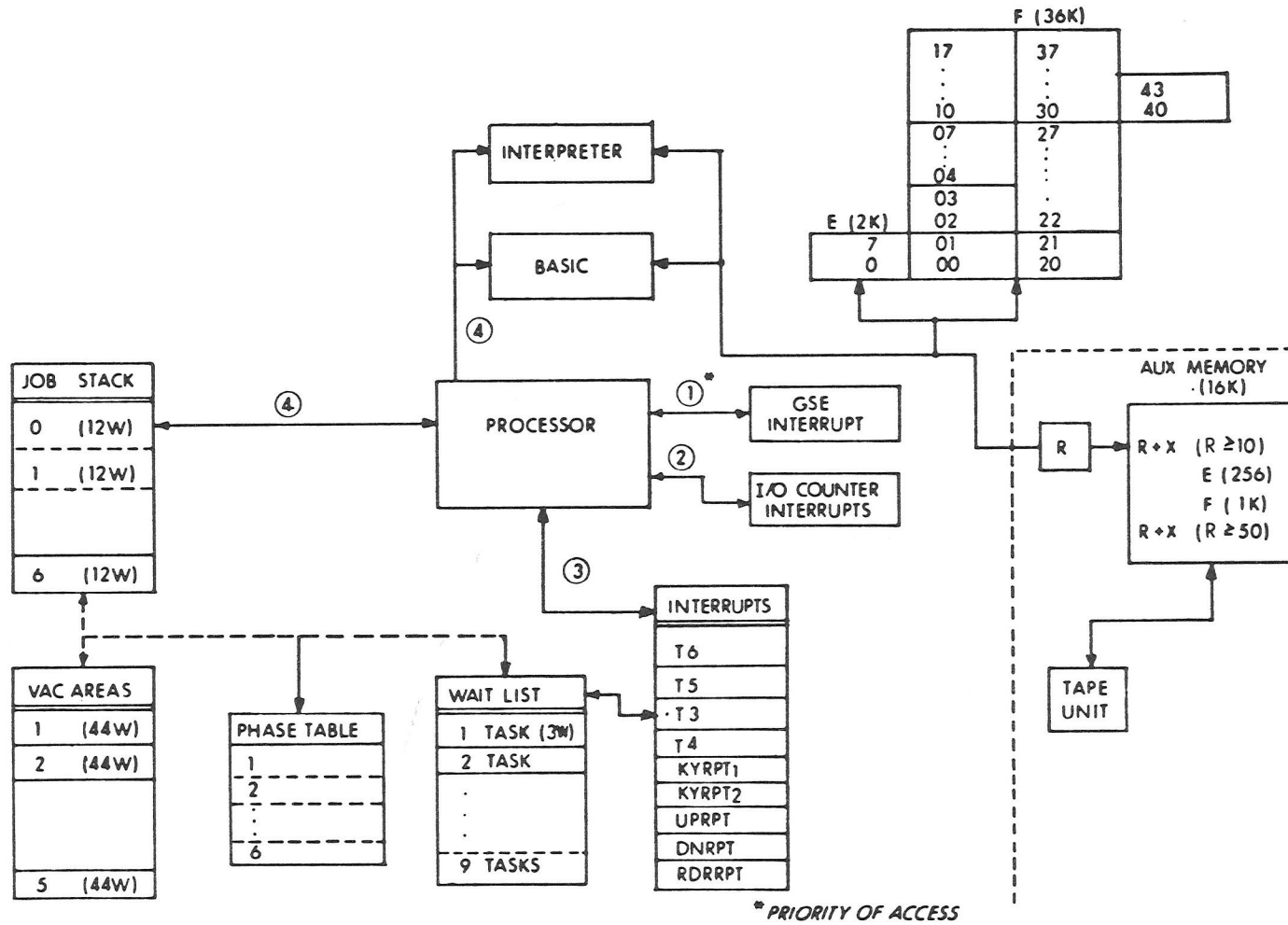


Fig. 4-1 AGC Auxiliary Memory

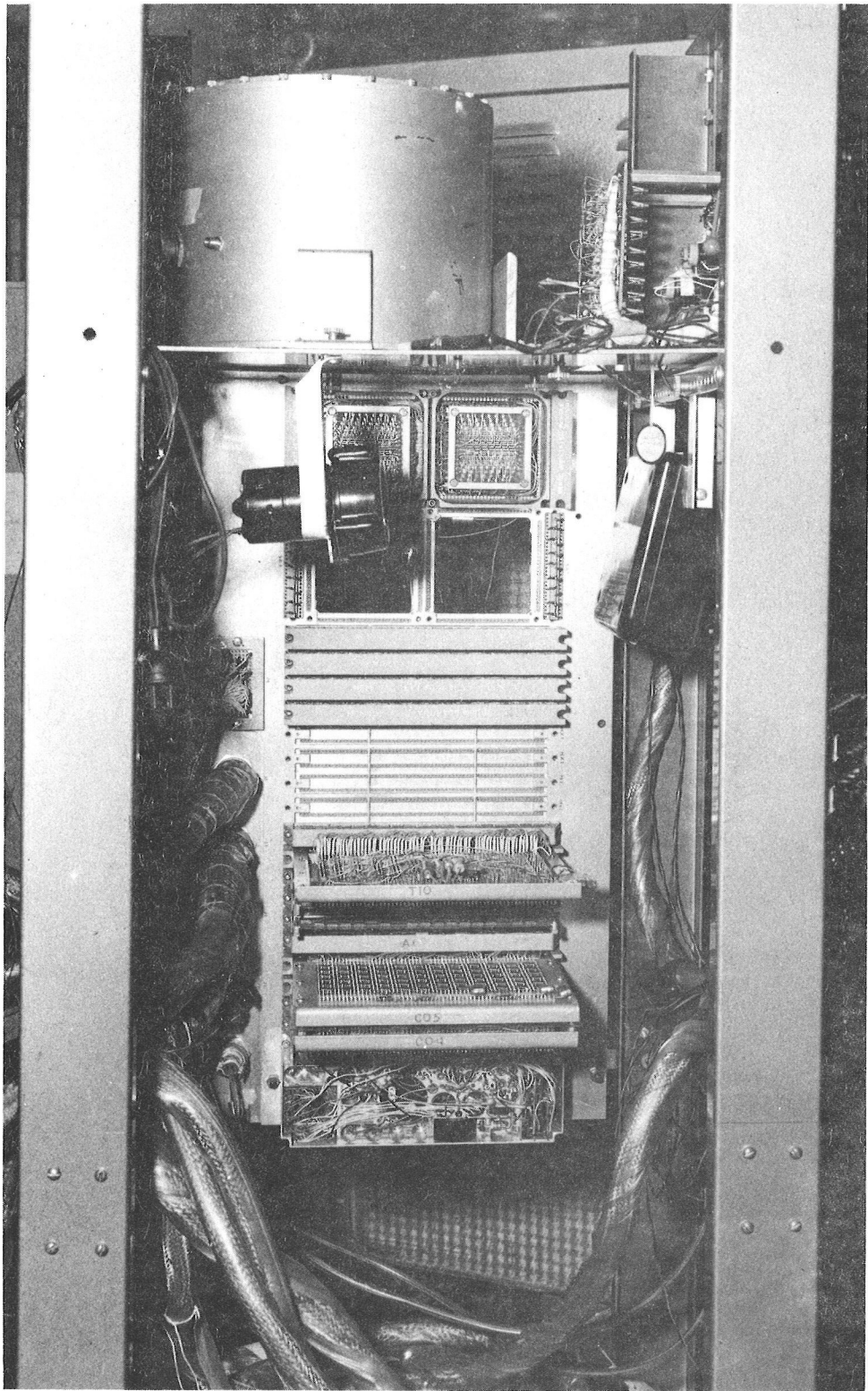


Fig. 4-2 Prototype Auxiliary Memory

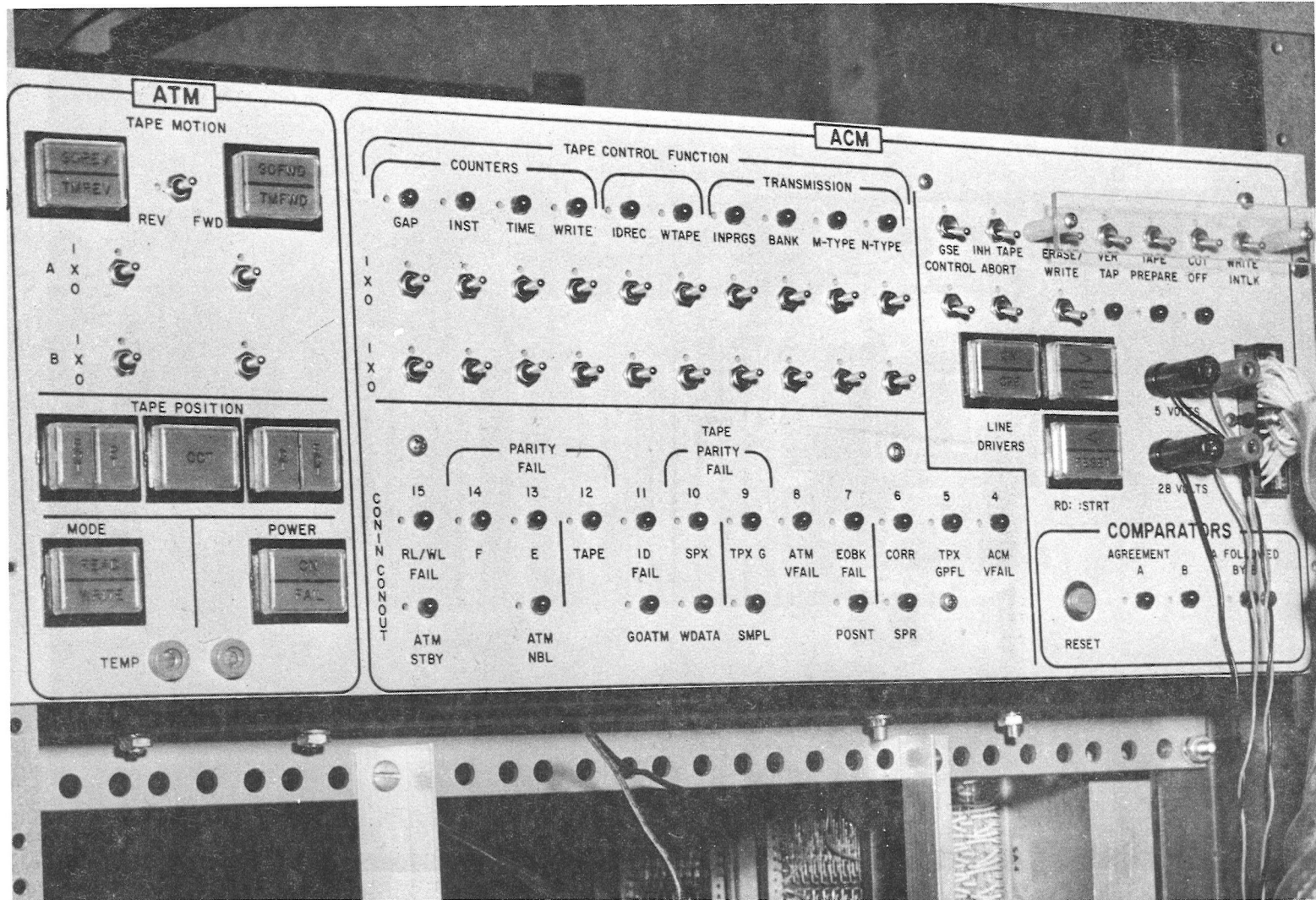


Fig. 4-3 Auxiliary Memory Control Panel

of the AGC. Whenever the AM needs to read memory or load memory during tape operations, it steals an AGC memory cycle. It does this by generating a MLDCH signal which essentially stalls the computer for one memory-cycle time.

The prototype equipment has worked quite well, the only problem noticed was a sensitivity to noise at the test connector interface which was felt to be due to the fact that the interface input resistors to the AGC were 1K ohms instead of 300 ohms as in the buffer box.

More detail descriptions of the Auxiliary Memory are contained in Ref. 1.

A mechanical change was made to the interface connector to allow the G/N test harness cables to get past the buffer-box electronics.

#### 4.2 Program Evaluation Sequence

The main study is contained in a separate publication, outlined as follows:

1. The AM, CRS, and AGC were tied together in the computer lab for operation with the Raytheon demonstration programs.
2. The AM was then operated in the systems test lab with a G&N system and a CRS. A demonstration program was run to transfer a program from the CRS to the AM and onto the AM tape and to verify it. This mainly demonstrated that the noise environment generated with all the equipment operating did not cause any problems. The demonstration program could not be used with a G&N since the interrupt locations read by the G&N system were utilized by the demonstration programs for other uses.
3. The AM was rewired for the Both-Bank modification. The drawings and many of the demonstration routines were changed also so as to work with the modification.
4. A K-start program (Lineages = AMDANCE) normally executed out of erasable bank, was operated out of fixed bank 65 of the AM. This program, used with Sundance Revision 306, provides for vertical erection and azimuth alignment, and permits the abort guidance system to measure the ASA alignment with the known and fixed reference base of the PGNCS.

It utilizes many routines in Sundance and appeared to be a fairly easy program to move.

In rewriting the program in bank 65, it had to be debugged while running the AM with the system. The first problems resulted in restarts that shut off the power to the AM, because the restart erased the channel bits which controlled the AM power. This motivated changes to the restart program and a special verb-noun combination (V94, N01-04) to make the turn-on, etc., of the AM easier. The complexity of the problems and the fact that the debugging took up system time pointed up the need for using the digital simulation as a debugging tool.

This program was run successfully out of the AM. The major problem was the using of a location called BUF2 for bank switching which was also used by the interpreter. When one used an interpretive routine in the Sundance program, the return location BUF2 was changed resulting in an incorrect return address which then resulted in a restart that cleared all the channel bits, thus turning power off the AM.

5. It was next decided to try to use P34 (TPI) in Luminary 69 normally operating out of bank 35 to operate out of bank 66 in the AM. The reason for using the terminal phase initiation (TPI) program was that it was a highly interpretive program and could be used not only for the segmentation study here but also for the software relocation and hardware dispatcher studies. This meant that only this program need be simulated. The problems encountered are discussed in detail in Section 4.6 below but, essentially, they pointed up how interactive and tightly written the AGC programs are.
6. It was then decided to move the whole bank up into the AM to eliminate many of the problems encountered in first moving part of a bank. At this point a massive revision of Luminary 69 was undertaken, incorporating the various display, interpreter, restart, and segmentation changes resulting in an assembly called ONEIF. This is reported in Section 4.3 below.

#### 4.3 AM-AGC Communication (Pinball Changes)

The first major step to control the AM from the DSKY was accomplished by means of a new extended verb: V94. V94 occupies about 500 locations and, since there was not a block of storage this large available in SUNDANCE, the lunar landing routines were deleted to make room. At this point it was first noticed that changing

blocks of the mission had repercussions in other parts of the program. Lunar Landing (LL) has multiple entry points (some of them not obvious from the coding), so the only way to determine all of the transfers into the LL routines is to delete them, assemble, and then trace back the cusses. Each of these transfers (which for our purpose will not be used) must now be made to point to some default routine; specifically, something which will make it obvious that an improper branch had been taken. In most cases this was done by a transfer to GOFLASH. The unexpected (garbage) display is usually sufficient to inform the operator that something is wrong. In all cases the coding changes were noted with a comment as to what the proper transfer used to be. This deletion of the Lunar Landing program is compatible with the idea that this program would probably be stored on tape and loaded into a section of memory sometime before it was needed.

V94 proper was placed in bank 31. The eight-line extended verb lead-in (TC TESTXACT) and a description of the verb options were put into the extended-verb log section (bank 43); the extended-verb transfer table was appropriately modified. V94 needs a noun to transfer certain information to the ATM, so N39 was created and defined in the erasable assignments section and in the Pinball Noun Tables. N39 is a three-component octal normal noun. There are also several temporary locations for V94 defined in EBANK 4.

V94 has four modes of operation selected by N12 (extended-verb option code).

- (1) Turn on the AM and recycle to the option-code display.
- (2) Load records from the tape into the ACM. Go with the ID of the first tape record in R1(N39) and the number of records to be loaded in R2(N39).
- (3) Write banks onto tape. Go with tape ID in R1, first bank to be loaded in R3.
- (4) Turn off the AM.

#### 4.4 Erasable Addressing

The ACM may be addressed as erasable as well as fixed. This provides up to 64 additional E-banks. The use of these E-banks is not as straightforward as one would like. The E-bank register in the AGC is only three bits long, so an additional four bits of bank information are necessary to enable one to reference all 72 E-banks. These four bits are built into the AM, and must be read and written through the AGC B-bank. A TS to E-bank will cause them to be zeroed. (The full E-bank appears as B-bank bits 10-8 and 4-1.) E-addressing in basic, though requiring some thought, is relatively simple. If the interpreter is to use the new E-banks, however, one can encounter some problems.



## 4.5 Interpreter Changes

The interpreter allows reference to only eight E-banks at any one time. A scheme for interpretive coding has been provided which:

- 1) Commits E-banks 0-3 to AGC E-banks 0-3
- 2) Allows E-banks 4-7 to reference AGC E-banks 4-7 or any one of the 16 groups of 4 AM E-banks. This relocation is effected by means of a register called ECODE associated with each job core set. ECODE is an index to one of the 17 variable E-bank groups. It is set by a new interpretive instruction called SEC (ECODE), and remains at a given value until it is changed or the job terminates.

What happens, then, is that all interpretive address constants (ECADRS) reference E-banks 0-7. When the decoding logic in the interpreter detects banks 4-7, it indexes on ECODE to fetch the E-bank.

The scheme necessitated changes to the interpreter, the executive, and the restart routines.

- (1) The five places where E-addresses are decoded were modified.
- (2) Coding for the SEC instruction was added and the instruction transfer table was modified.
- (3) The executive was modified to change the core sets to 13 locations (the 13th being ECODE) and the number of core sets cut from eight to seven. (The remaining five locations were defined as experimental temporaries). This was necessary for restart.
- (4) ENDOFJOB was modified to return ECODE to the nominal value (Bit 3). This allowed those interpretive programs previously written to run with the auxiliary memory.
- (5) RESTART was modified to initialize the core sets properly.

It was necessary to delete SETTRKR and RRLIMCHK from Fixed-Fixed in order to make room for the above changes. References to the deleted routines were bypassed in the Interrupt Lead-Ins, T4RUPT, and P20-P25. The Fixed-Fixed constant pool was reorganized to make the newly acquired locations contiguous.

To check out the modified interpreter, simulations were run on P34 (SUNDANCE 306) with both the old and new interpreter. The SEC instruction was not used (the modified interpreter is compatible with existing programs). The test is still valid,

however, because all references to E-banks 4-7 exercise the same decoding logic regardless of the value of ECODE. The cost of the software relocation is about a 3-percent increase in run time.

Figure 4-4 shows the results of two simulations that are superimposed. The full  $\square$  curve is the P34 program being run on the computer with the old interpreter. The dashed  $\square$  curve is the P34 program being run with the new interpreter. One can see that the old interpreter finished before the new one.

The simulation plot actually shows four plots. The (X) plot is the 100% activity line, the ( $\diamond$ ) plot shows what percentage of the time the computer is in interrupt, the ( $\square$ ) plot shows what percentage of the time the program is running, and the (+) plot shows the percentage of the time that the program has allotted to unpredicted interruptions. It is used as a simulation technique to see what safety margins one has in the program. The plots are calculated every two seconds.

It was also considered necessary to allow N01-N03 (machine address to be specified) to be able to reference the AM erasable. This can now be done by the use of an "extended ECADR" (seven bits of bank information followed by eight bits of address).

## 4.6 Mission Segmentation

### 4.6.1 By-Program

For purposes of trying to segment mission programs an effort was made to move P34 up to the ACM. This had many consequences, as was mentioned above, on the rest of the mission routines. It soon became obvious that simply moving a major program up to the ACM was hopeless.

P34-P35, P74-P75 is an assembly log section that is used to establish terminal phase initiation (TPI) maneuvers. This log section contains the control program, various subroutines, and constants. The control program and some subroutines (INTINT, VNPOOH, VN1645, S33-34.1) located in bank 30 were moved to AM bank 66. When these were assembled with Luminary 69, a number of cusses were caused by inter-Superbank references. In particular, the cussed references are FCADR, Constant definitions, and subroutine calls (Basic and Interpretive). Investigation revealed that, while the FCADR and Constants problems can be solved readily, fixing the subroutine calls required so many program changes that it was impractical.

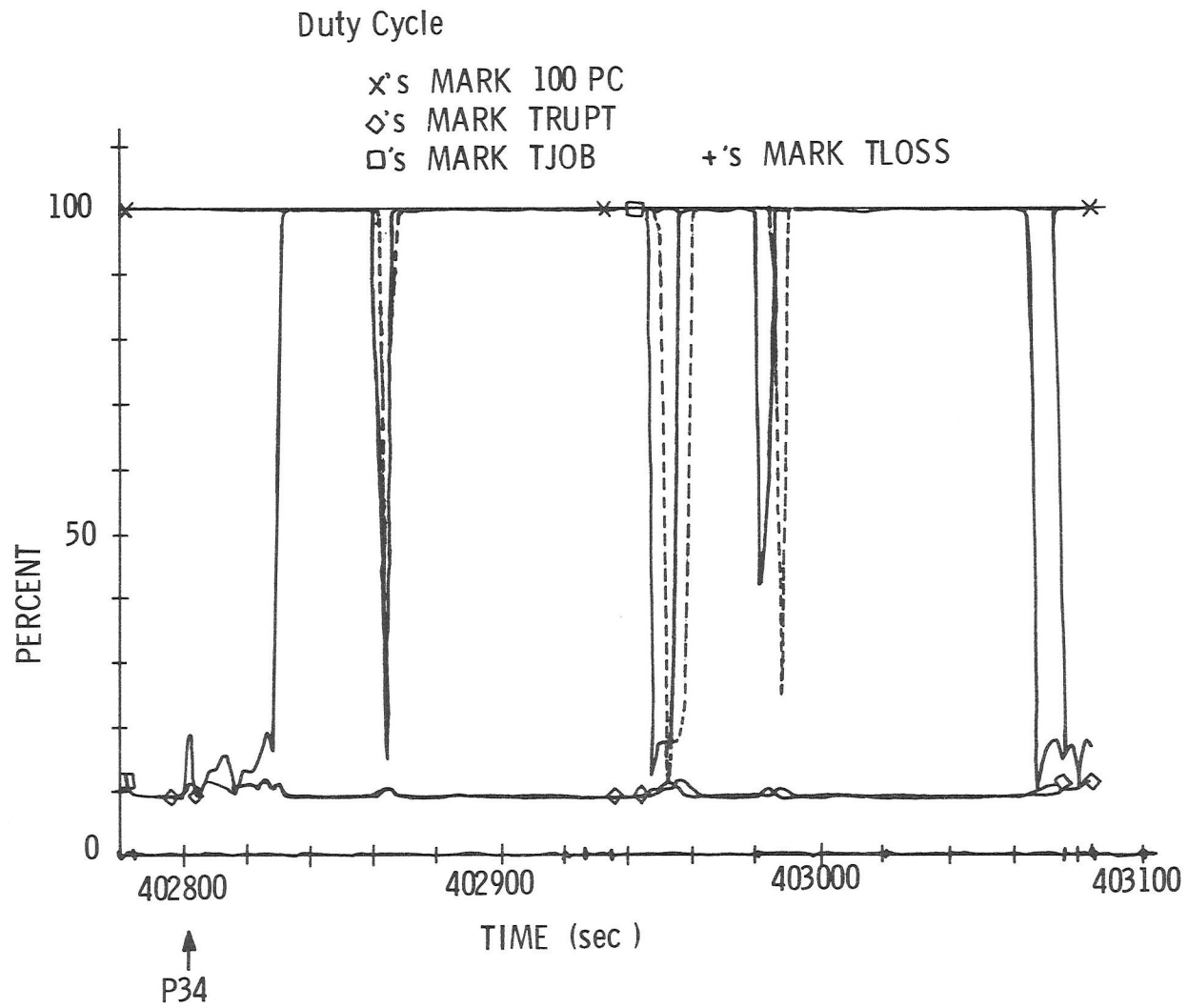


Fig. 4-4 Old and New Interpreter Simulations

The FCADR's (GAP assembler address constants) pointing to bank 66 are subroutine entry addresses within a transfer table located in switchable fixed. These were cussed because the bank declaration had set the Superbank register to the Superbank when the reference is to the external Superbank (see Fig. 4-5). The cusses may be removed by changing the bank declaration, but the transfer would be going to the Superbank if the Superbank register were not set properly. Accordingly, the FCADR's can be changed to point to an intermediate area in fixed memory where the Superbank register is set to the correct value. A transfer then will allow program control to be transferred to the proper Superbank.

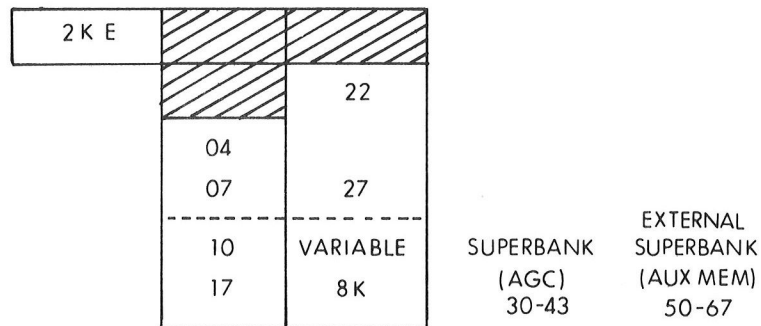


Fig. 4-5 Memory Map

In the intermediate area, 2CADRs are used to specify the transfer bank number and relative address for each reference to the external Superbank. Two lines of coding get this complete address and transfer to five words of common coding which set the registers and transfer control to the specified address. Since there are four such references, a total of 20 locations are needed.

It was suggested that 2CADRs be used to replace the FCADR's in the transfer table. Since 2CADR generates constants for bank registers and the address, it required two locations in the transfer table. Consequently, the program transfer logic which uses this table must take this into account through proper indexing. It was felt that changing this logic is undesirable.

There are five constants which were moved to bank 66 and referenced by bank 35. They may be redundantly defined in bank 35.

There are five basic language subroutines that are located in Superbanks different from the Superbank which contains programs using them. These subroutines

averaged only 7 instructions each. Consequently, rather than developing a calling sequence to take into account the Superbank settings, it was decided to be more expedient to duplicate these subroutines in the Superbank which uses them. It does not take many more memory locations and the bank settings need not be a source for concern when a restart or program interruption occurs.

The incompatibilities caused by moving P34 into bank 66 may be circumvented as outlined. This leaves interpretive subroutine calls. There were five interpretive subroutines relocated in bank 66 and used by programs in the Superbank. Since the five subroutines total 411 instructions, copying them back into Superbank is undesirable. To make these calls correctly, basic language coding should be used. Essentially, this coding obtains the particular subroutine complete address (2CADR), transfers control to a common area where the return complete address is saved and program transfer is made to the particular subroutine complete address. Upon completion of the subroutine functions, return is made through the saved subroutine return address.

For each call four locations are required to pre-specify the complete address and obtain that address. The common area contains 21 instructions and two temporary erasable storage locations. For each subroutine return, 10 instructions are required.

If there were no subroutine calls within these subroutines and if the number of calls were relatively small, the above scheme would be feasible and practical. However, this is not the case, because subroutines call other subroutines within this group. This results in destroying the saved calling-program return address. To prevent this, the subroutines must be altered, thereby requiring another level of coding and more erasable locations. In addition, each of the subroutines have more than one entry and each entry is called from many places. The program listing showed a total of 47 calls to these subroutines. This number required about 250 words of new coding which does not include those for protecting the calling-program return address. In addition, the 47 calls must be changed to reflect the new scheme. This scheme appears feasible, but the confidence factor is low because of the complexity in bank switching.

It was concluded that moving this part of the Mission program into the ACM for testing the ACM behavior with regard to mission programming is highly impractical. For the work involved, it seems more practical to move the whole bank.

#### 4.6.2 By-Bank

The most fruitful attack was in fact to redefine the entire Superbank 3 as Superbank 6. (This of course left bank 31 free for V94 and obviated the deletion of the Lunar Landing routines).

By changing 11 cards in the Relative Setloc log section, almost all references to Superbank 3 were changed to Superbank 6 - the assembler doing this automatically when creating address constants, etc. Only a few nonconventional situations had to be specially adjusted.

- (1) V37 in RESTART uses a single-precision address table for purposes of starting or restarting major modes. Since major mode programs exist only in Superbank 3, this is a workable scheme. But since Superbank 3 was now in Superbank 6, it was necessary to fix the coding to set the proper Superbank before transferring to programs with CADPS that pointed to Superbank 3.
- (2) SEUDOPOO references ENGINOF1 with a single-precision address, so Superbank had to be set and re-set around this call.

The eight banks that were moved were modified so that they could be written on tape. An ID was put in the first location and the next six were ignored because of the difficulty of writing into those ACM locations in a bank-to-bank transfer from the CRS (the ambiguity between the AGC special registers).

Verb 98 was written as a facility for effecting this bank-to-bank transfer (the only way of moving the assembled information into the ACM).

#### 4.7 Restart

A GOJAM causes bit 9 of channel 11 to be zeroed. This bit is a switch to supply power to the ACM line drivers. Once the line drivers go down, it is necessary to execute a complete turn-on sequence to get the ACM active again. This would not do if a program were being executed from the ACM. Since the line drivers do not go down until about 1/2 second after bit 9 is shut off, it is possible to get in and reset bit 9 before the drivers go down; in fact, this is what was done. The Restart program was modified so that the first thing it does is to check a flag (flagword 0, bit 11) to determine the status of the AM. If the AM were on, bit 9 is set, then the restart continues normally. The program as it now stands has not been tested in

detail, so it is very possible that bugs still remain. However, fixing them is now a debugging problem, not an organizational problem.

#### 4.8 Conclusions

Because of the limited size of the AGC memory, the mission programs are coded in a highly connected manner, often relying on knowledge that some segments of code will be in certain locations. Transfers of control often rely on the fact that unrelated routines will be assembled into the same bank.

The problems that were encountered were due mainly to moving sections of the existing mission program with the AM. Writing new programs, which will reside in the AM, and altering existing mission programs to be compatible with these would be less difficult.

## CHAPTER 5

### COMPARISON BETWEEN THE MODIFIED TRAY B AND THE AUXILIARY MEMORY APPROACH

#### 5.1 Comparison

If one were to say that the AGC had to have increased storage capacity but that additional space inside the spacecraft were not available, then the modified Tray B solution would be required. Similarly, if one were to say that a minimum impact on present subsystems were required, then the auxiliary memory approach would be required.

Since the possibility of such clear-cut alternatives is unlikely, additional considerations such as comparative costs have to be considered. This involves the associated GSE and the two configurations identified by Fig. 5-1 and 5-2. The two configurations are equivalent with respect to functional capability and electrical component count.

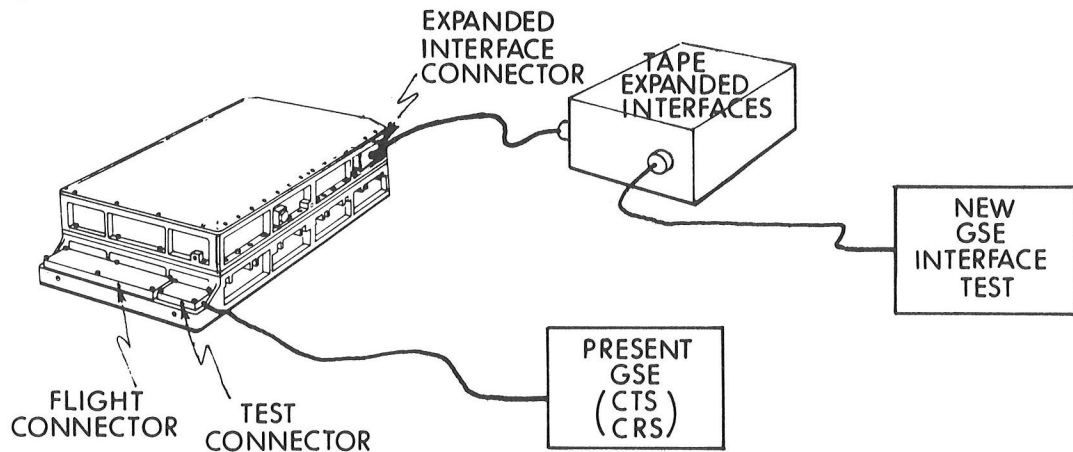


Fig. 5-1 Modified Tray B Configuration

Figure 5-1 indicates the test configuration for the Tray B modification, including the computer, the tape, and the interface subassembly with the associated test equipment. The present GSE cabling and buffer box are unchanged. New GSE and cabling is required for the tape and interface subassembly.



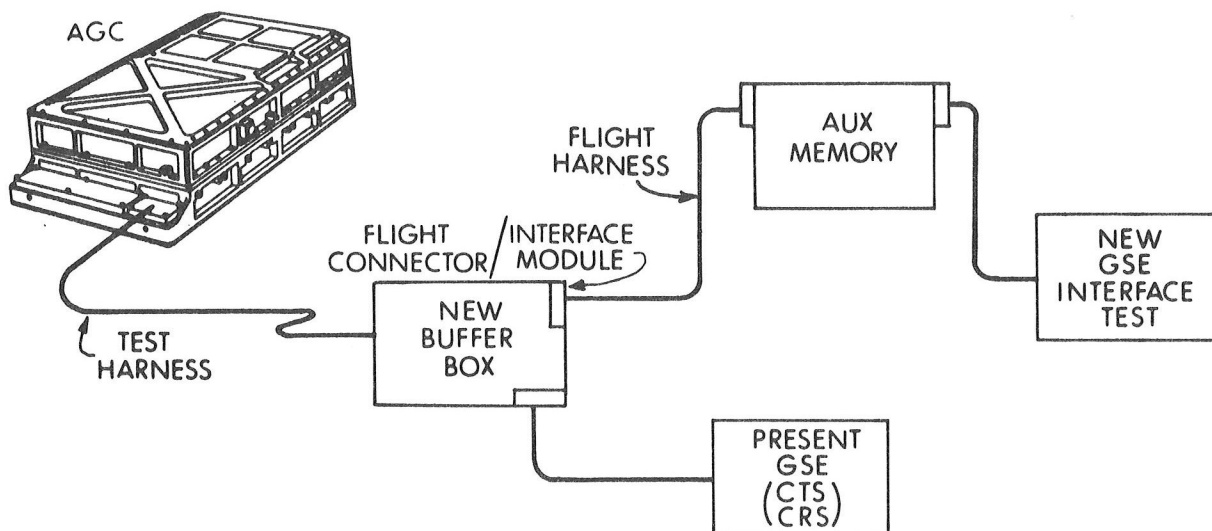


Fig. 5-2 Auxiliary Memory Configuration

Figure 5-2 indicates the test configuration for the AM approach including the AM with its flight harness and associated test equipment. A new buffer box and computer interface cable is required. The new GSE and cabling for interface testing is equivalent to the Tray B version.

The main cost advantage of the auxiliary memory approach over the modified Tray B approach is that of not being as tightly constrained in volume and thus not presenting as difficult an engineering and development problem. However, another cost advantage is the fact that the AM is fabricated and sold off as an independent assembly in place of the more complex Tray B version where a complete computer with its tape and interface assembly must be sold off.

The above considerations permit one to make up a comparative cost table.

#### COST COMPARISONS

	<u>AM</u>	<u>Tray B</u>	<u>Reason</u>
Development & documentation	<		Revisions more expensive than new design
New GSE		<	Additional buffer box
Factory Test	<		} Only have to sell off a single new assembly; not an AGC
Sell off & qualification	<		
Fabrication	<		Possible competitive sources

The main reason the cost would be less for the AM is that it would be a separate unit and not affect drawings or documentation already available on the AGC; the Tray-B approach effectively involves the sell-off and verification of the AGC all

over again as well as extensive documentation revisions. Aside from the cost factor, the present Tray B needs to utilize a Portafam-type (very-dense) memory which is still in the developmental stages. The auxiliary memory would not have to be a very compact memory, assuming one could find space elsewhere for it.

## 5.2 Conclusion

The relative cost picture along with the actual feasibility of accomplishing the job does make the auxiliary memory look best.

## CHAPTER 6

### TEST CONNECTOR INTERFACES

#### 6.1 Introduction

The test-connector interface is not normally used in flight. However, due to the fact that it does allow one access to the read and write bus along with various control pulses, it is a quite flexible interface.

The auxiliary memory utilizes this interface to augment the AGC memory and the interface capability of the AGC. The main problem in this utilization is that, when one turns on the power to the circuitry interfacing with the AGC, one has to be sure that the AGC is in a mode such that the AGC won't be adversely affected. The auxiliary memory applies the power to the interface circuits during a MRDCH command during which the AGC will not be affected by information on the bus. Similarly, during the shut-down process, the voltage is removed from the interface so as not to bother the AGC.

This section describes the present AGC-AM interface and signals. Noise problems experienced on the input were felt to be due in part to the use of 1K resistors instead of the 300-ohm resistors which are normally in the buffer box. Table 6-I contains a list of the interface signals. Figure 6-1 is a photograph of the test-connector interface electronics for the prototype equipment.

Section 6.2 describes the interface signal characteristics as measured on the prototype equipment.

The auxiliary memory actually utilizes 23 of the input signals and 55 of the output signals listed on the interface connectors. Since there are 33 input signals and 81 output signals plus the 11 special signals used by the test configuration by the Core Rope Simulator PACs and Computer Test Sets, it is recommended that there be both a flight harness and a test harness. The flight harness would be part of the test configuration. This would result in a minimum configuration in the wiring and circuits in the flight harness since only the isolation circuitry for 23 inputs

Table 6-I. List of Test-connector Input/Output Signals

INPUTS			
MDT01	} Data	*MTSCAI	} Commands
MDT02		MSTRT	
...		*MSTP	
MDT16		MRDCH	
*MONWBK		MLDCH	
MONPAR	*MLOAD	*MREAD	} Inhibits
*DOSCAL	*MNHSBF		
*DBLTST	*MNHRPT		
	*MNHNC		
	NHALGA		
	MAMU		
		WEFRC	
OUTPUTS			
MWL01	} Data	MST1	} Status
MWL02		MST2	
...		MST3	
...		*MBR1	
...		*MRB2	
MWL16		MGP/	
MSQ10		*MSP	
MSQ11		*MNISQ	
...		MIP	
...		*MINH2	
...		*MINKL	
MSQ14	} CMDS or Responses	*MTCSA/	
MSQEXT		MREQUIN	
MSQ16			
OUTCUM			
*MT01	} Timing or Gating Pulses	*ALGA	} Special
MT02		*STRT1	
...		*STRT2	
...		*SIGNY	
...		*CNTRL1	
...		*CNTRL2	
MT12		*MTH1	
*MRULOG		*MTL0	
MWFBG		BPLSW	
MWEBG		*4SW	
MWBBG		0VDCA	
MWSG		*MWATCH/	} Alarms
*MWZG		*MCTRAL/	
*MWQG		*MTCAL/	
*MWBG		*MRPTAL/	
*MWLG		*MSCDBL/	
*MWAG		*MWARNF/	
MWCH		*MSCALL/	
MWG		MOSCAL/	
MRSC		MVFAL/	
MRYG	*MPIPAL/		
*MRLG	MGOJAM		
MRGG	*MPAL/		
MONWT			
*MRAG			
MRCH			
*MON800			

\*Signals not used by the Auxiliary Memory

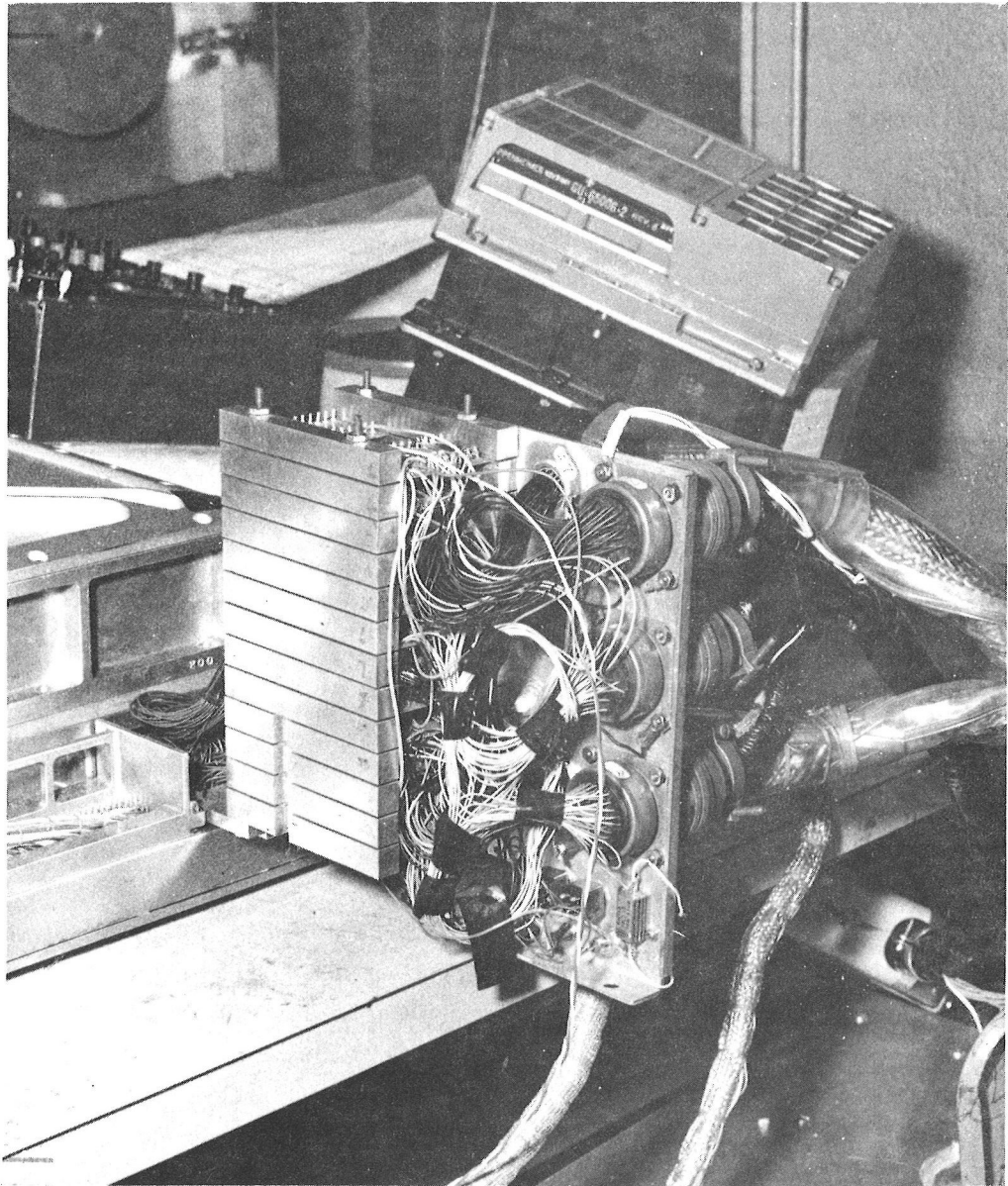


Fig. 6-1 Test Connector Interface Electronics

and 65 outputs are necessary in the interface module. Another configuration (the present AM configuration), which would permit the AM to interface directly with the computer during test, would require a larger flight interface module for the additional CTS interface and an additional cable connector. To provide the functions of the present restart monitor module, an additional 9 signals would be required in the flight harness and the logic of the monitor could be added to the AM.

## 6.2 Electrical Interface Study

This study is included to document the interface signal characteristics and to permit one to appreciate the sensitivity of the interface.

Figures 6-2 through 6-10 show the interface circuitry to which the photographic Figures 6-11 through 6-40 correspond. For example, photos of MDT02 and MDT10 signals were taken using both a voltage probe as well as a current probe. Test points that were inaccessible are noted on the photos. Data passing through the interface module (IM) for these two signals were photographed, first, with data originating in the CRS and, secondly, with data originating in the ACM (auxiliary core memory).

The photographs of all signals listed in this report were taken using a current probe when accessible, otherwise a voltage probe was used. These photos contain some interesting data.

The use of Signetics logic flatpacks (SE156J line drivers) for driving AGC signals into both the CRS and ACM is discussed next. These flatpacks are used in the IM and in the case of the CRS must drive a twisted pair approximately 17-1/2 feet long. The SE156J line drivers were intended for capacitive line driving applications in high-speed low-power digital systems. The individual twisted pair of the cable represents a characteristic impedance of 130 ohms. Due to the high-speed turn-off characteristics of the SE156J and the ultimate short circuit produced, some large currents are set up in the cable as shown in Fig. 6-20, 6-23, 6-26, 6-27, 6-30, and 6-32. A comparison with voltage waveforms at the same points will show the difference. A much shorter cable run to the ACM from the IM reveals somewhat the same effect only not as significant.

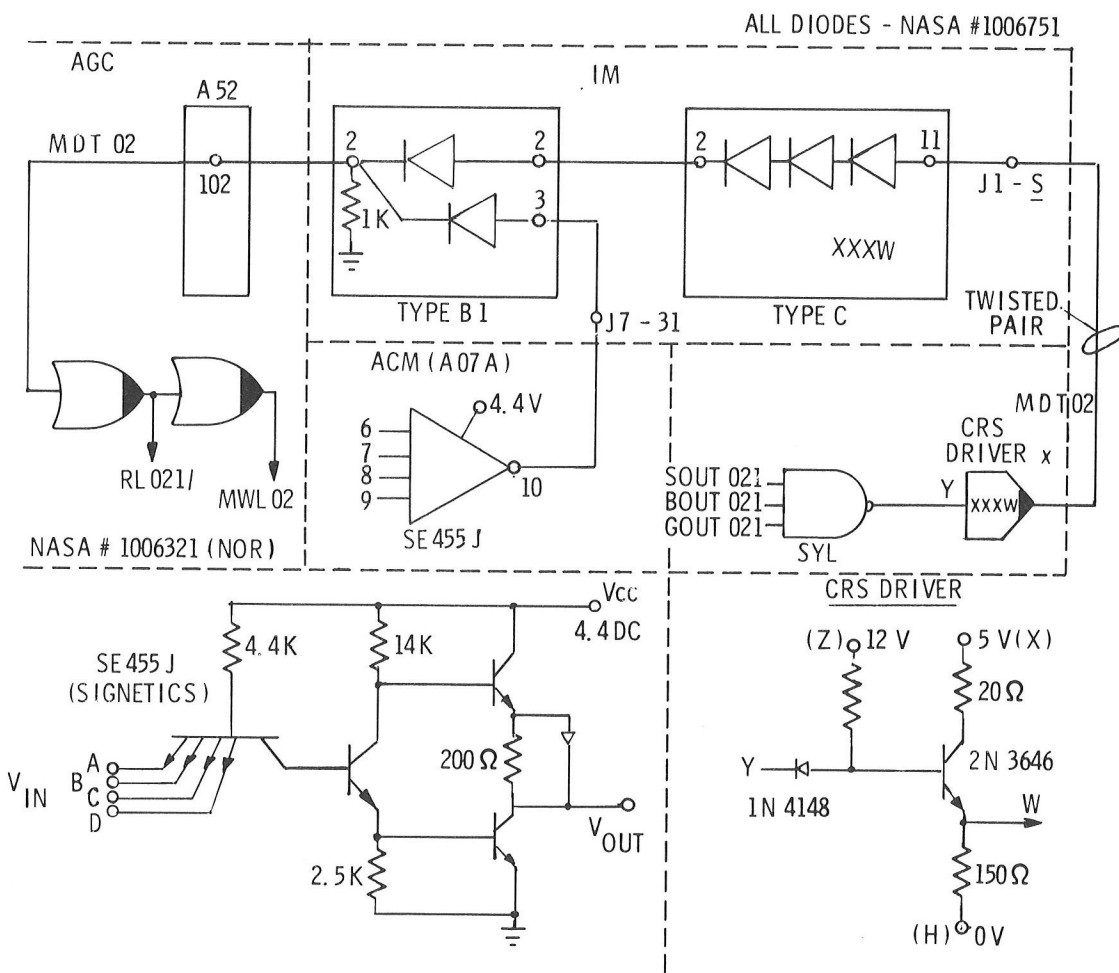


Fig. 6-2 MTD02 Input Interface

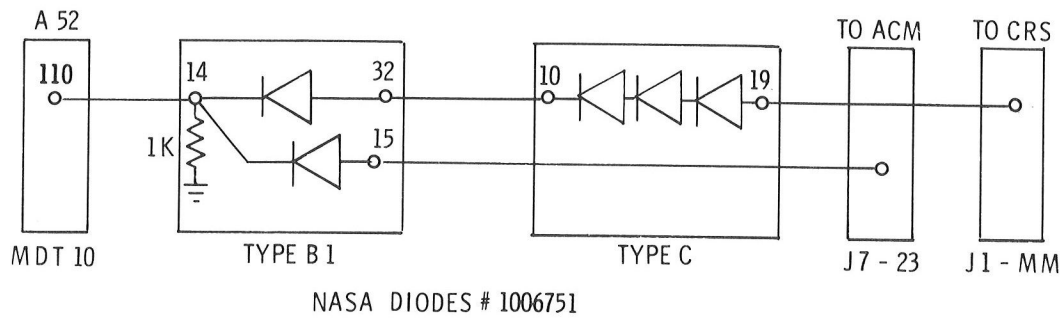


Fig. 6-3 MDT10 Input Interface

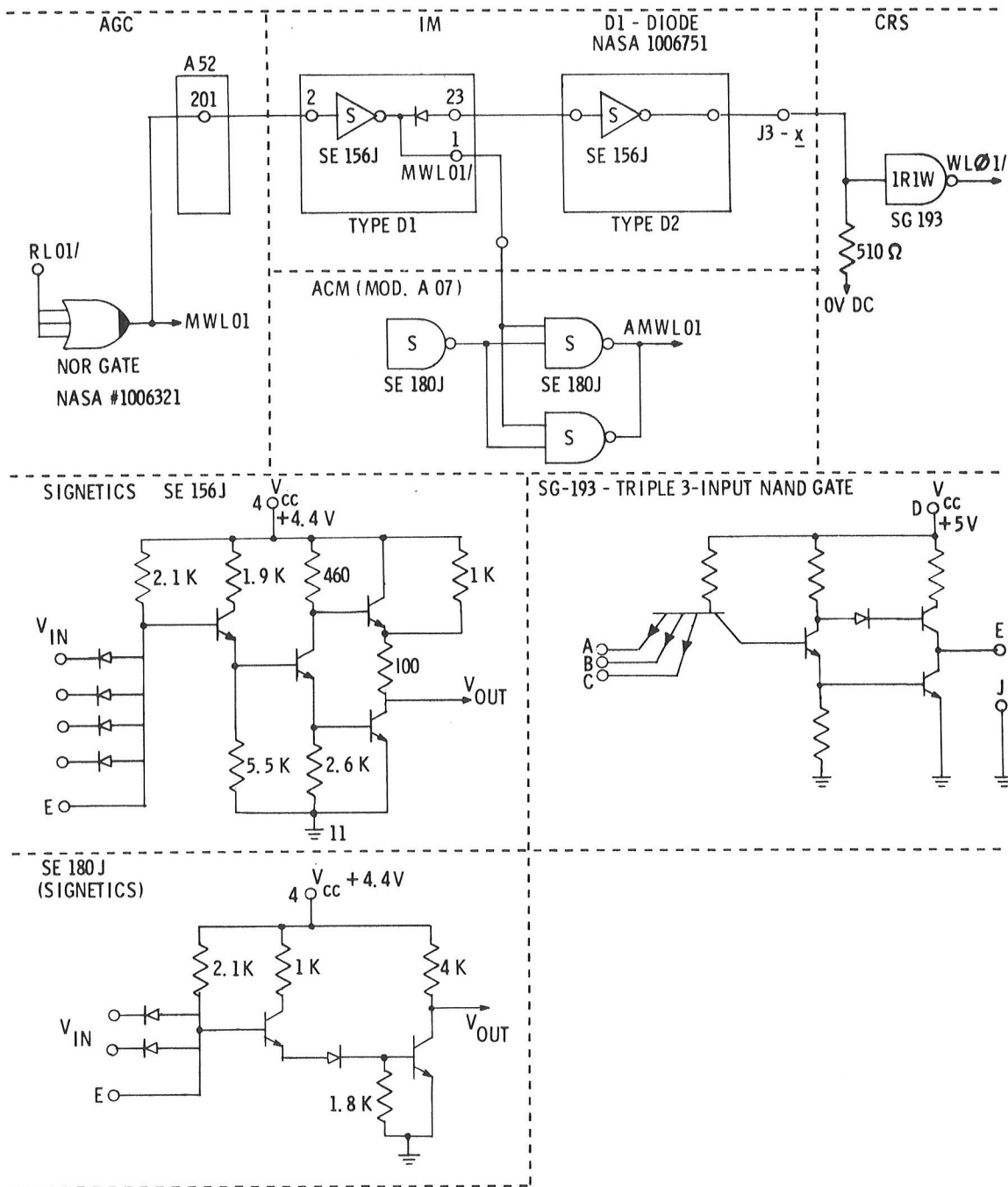


Fig. 6-4 MWL01 Output Interface



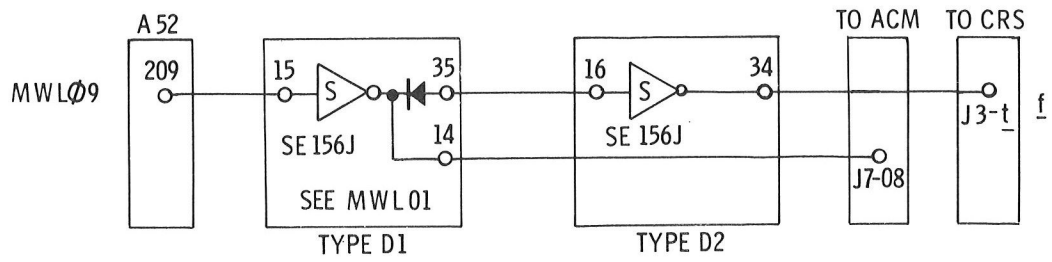


Fig. 6-5 MWL09 Output Interface

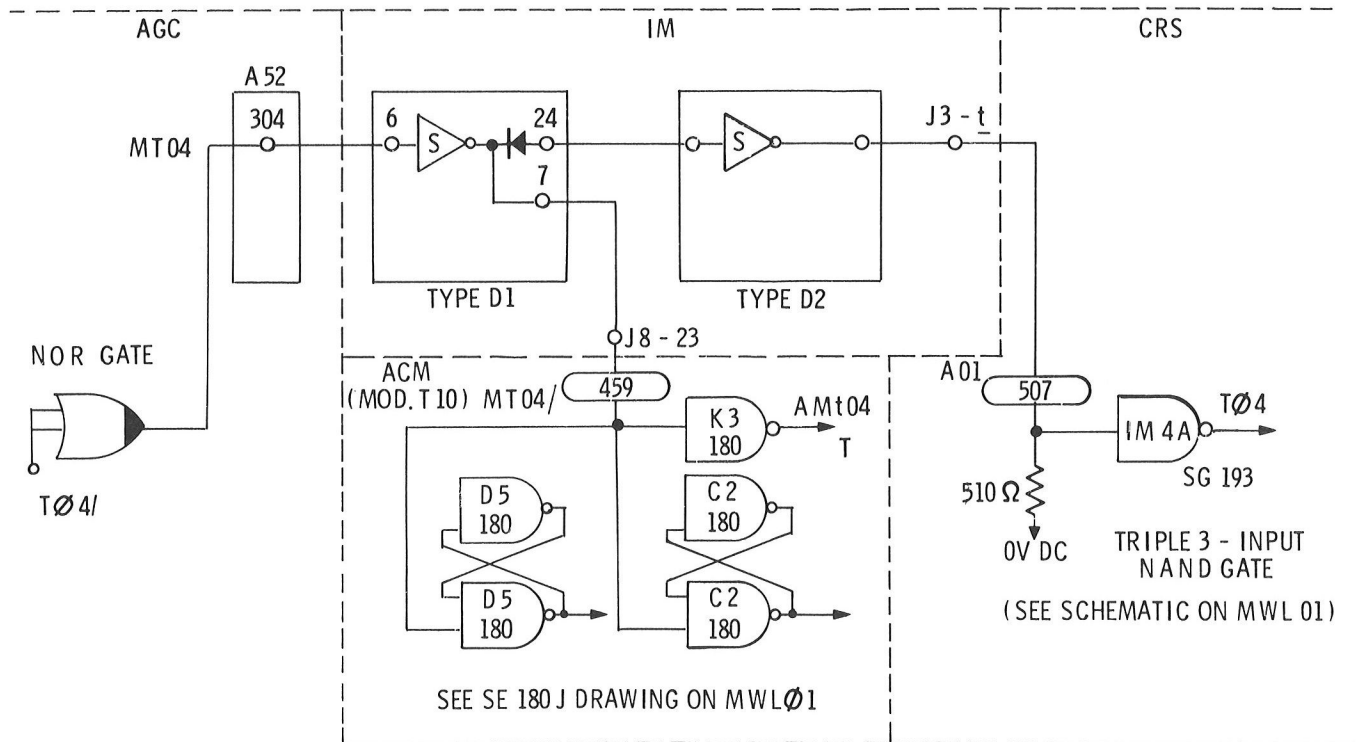


Fig. 6-6 MT04 Output Interface

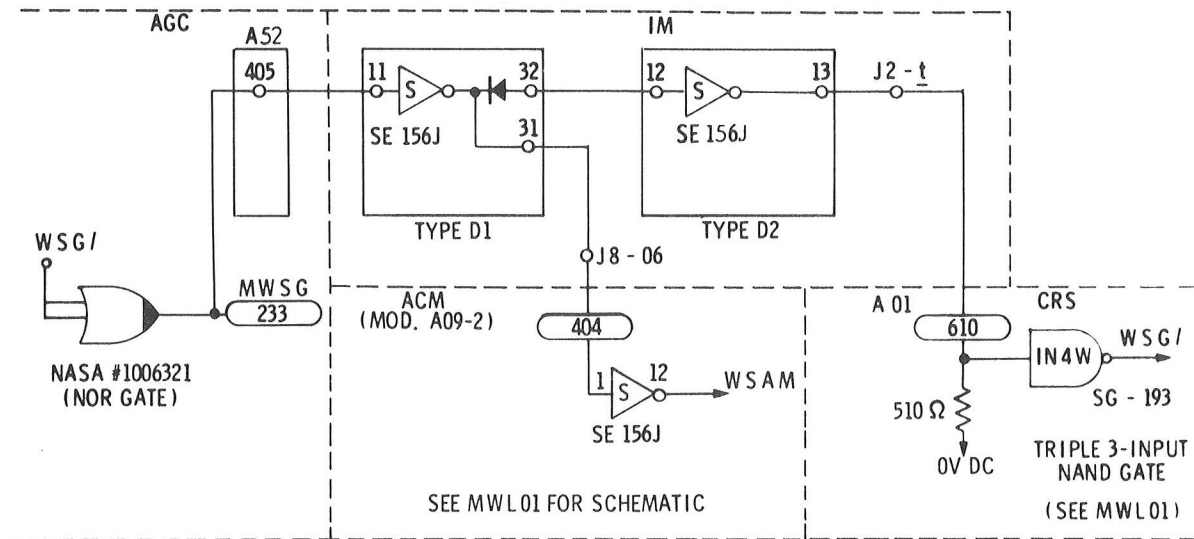


Fig. 6-7 MWSG Output Interface

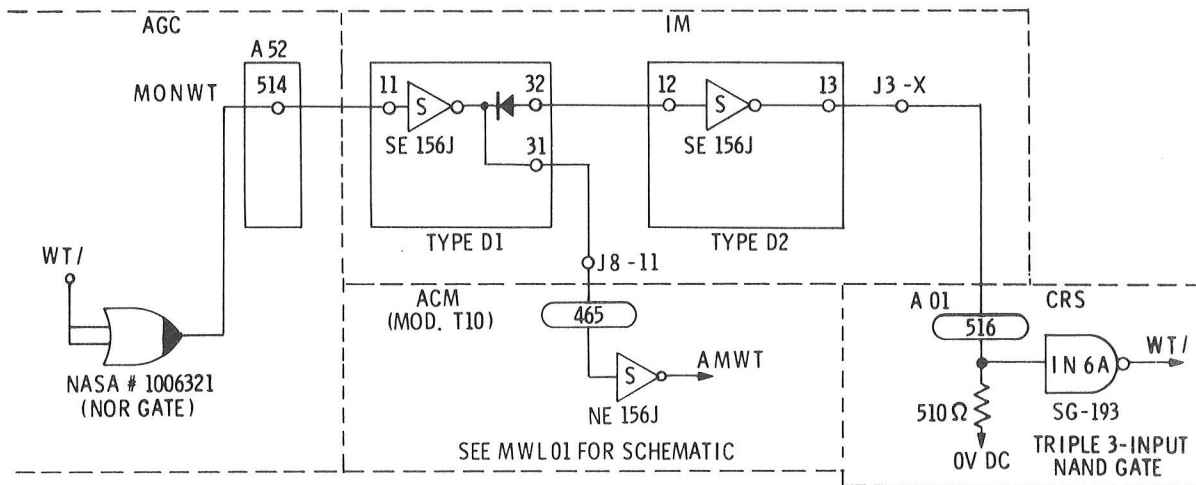


Fig. 6-8 MONWT Output Interface

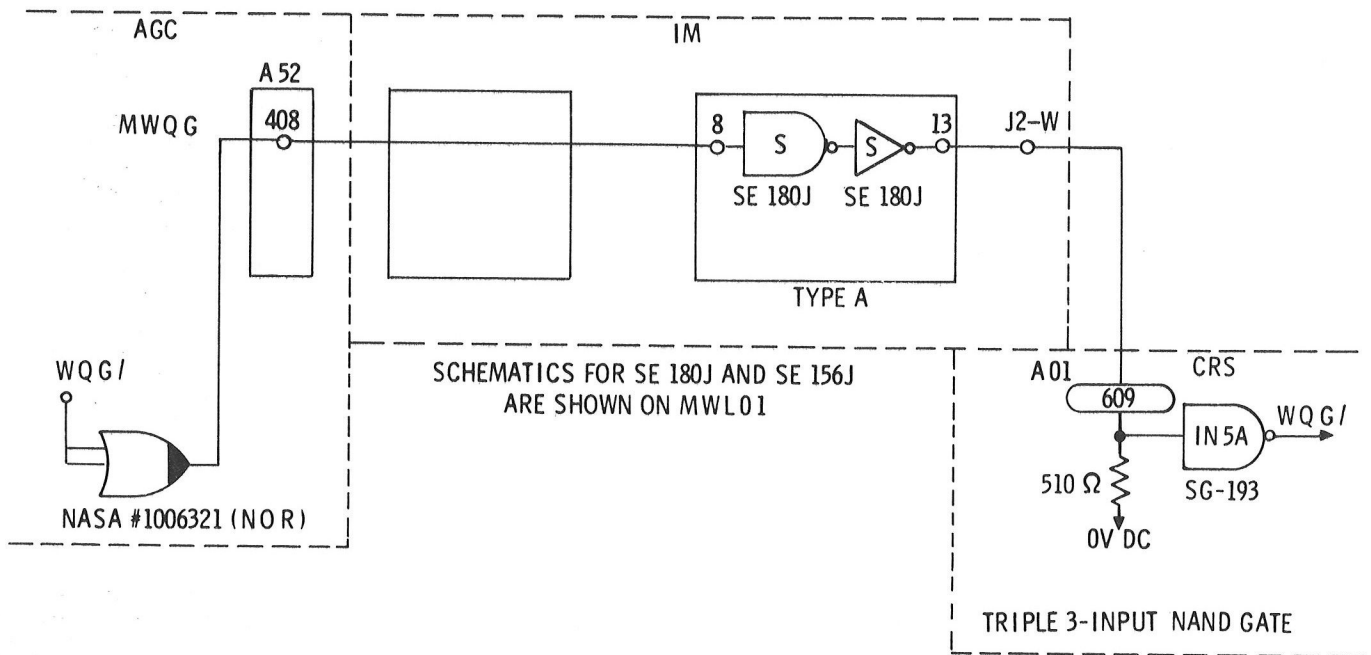


Fig. 6-9 MWQG Output Interface

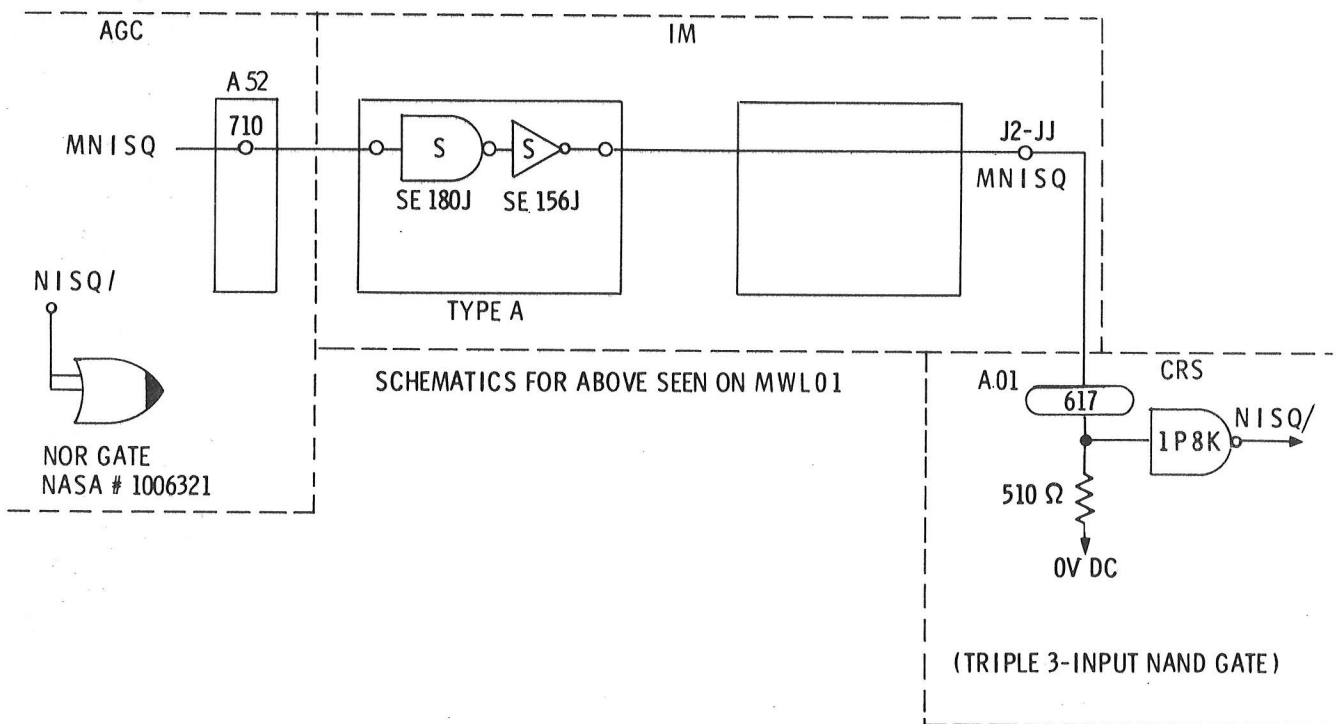


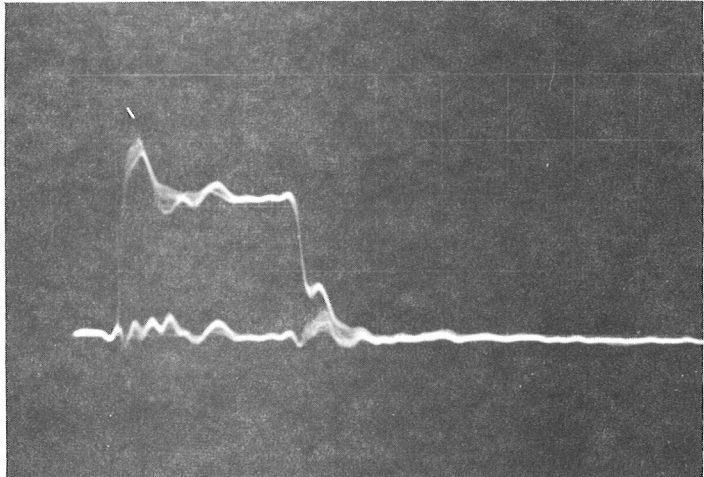
Fig. 6-10 MINSQ Output Interface

MDTO2  
DATA FROM CRS

VOLTAGE WAVEFORMS

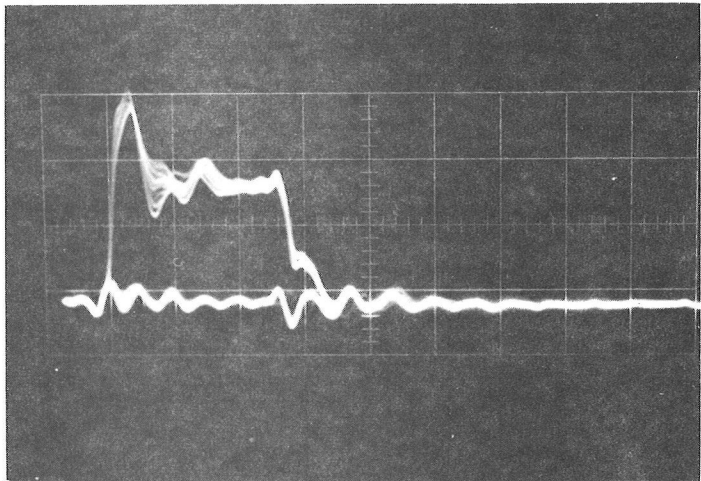
PIN 11 J1-S  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $2 \text{ V}/\text{cm}$   
SYNC - RGWAGC

Fig. 6-11



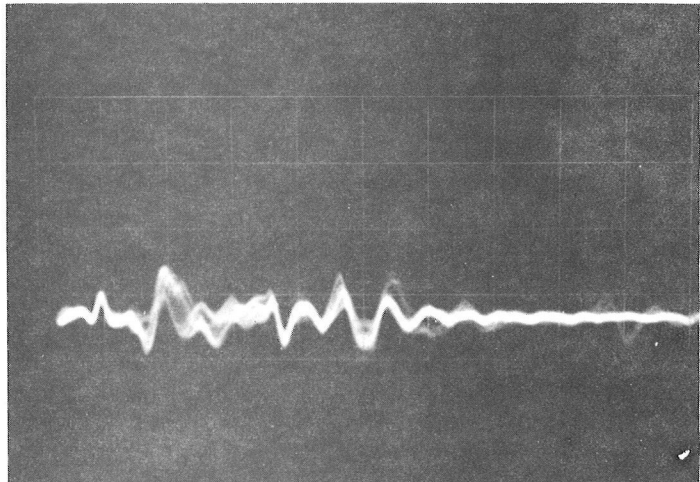
PIN 2 (102)  
HOR =  $0.2 \mu\text{ s}/\text{cm}$   
VER =  $1 \text{ V}/\text{cm}$   
SYNC - RGWAGC

Fig. 6-12



PIN 3 J7-31  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $1 \text{ V}/\text{cm}$   
(Check for leakage  
back to ACM)  
SYNC - RGWAGC

Fig. 6-13

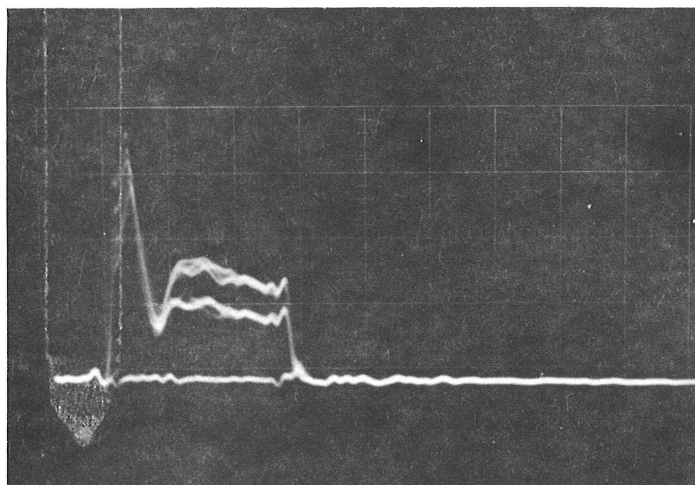


MDTO2  
DATA FROM CRS

CURRENT WAVEFORMS

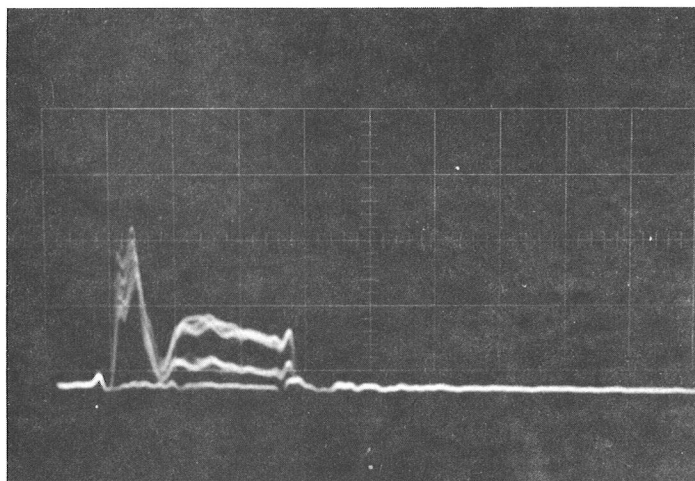
PIN 11 J1 - S  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $2 \text{ ma}/\text{cm}$   
SYNC = RGWAGC

Fig. 6-14



PIN 2 (102)  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $2 \text{ ma}/\text{cm}$   
SYNC = RGWAGC

Fig. 6-15



MDTO2

DATA FROM ACM

VOLTAGE (TOP) WAVEFORM

PIN 2 (102)

HOR =  $0.2 \mu\text{s}/\text{cm}$

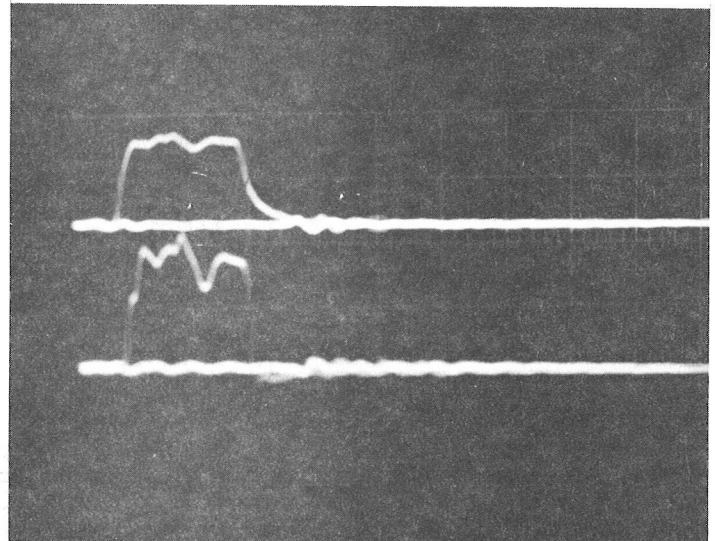
VER =  $2 \text{ V}/\text{cm}$

CURRENT (BOTTOM) WAVEFORM

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $2 \text{ ma}/\text{cm}$

Fig. 6-16



VOLTAGE WAVEFORM

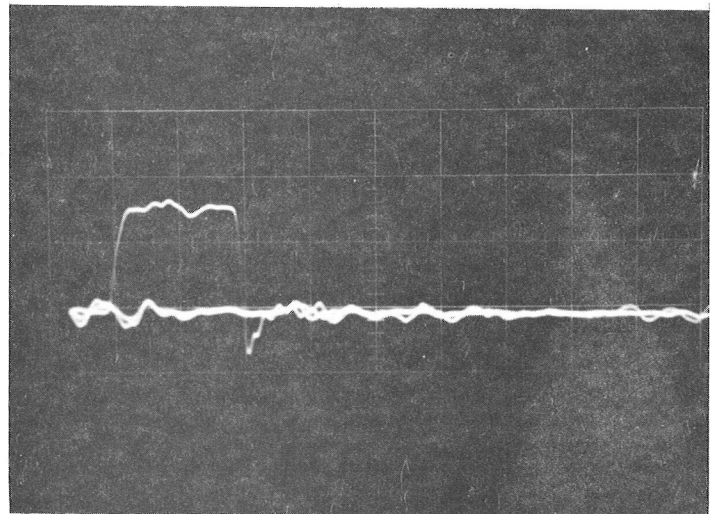
PIN 3 J7-31

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $2 \text{ V}/\text{cm}$

(Unable to get current  
Waveform - inaccessible)

Fig. 6-17

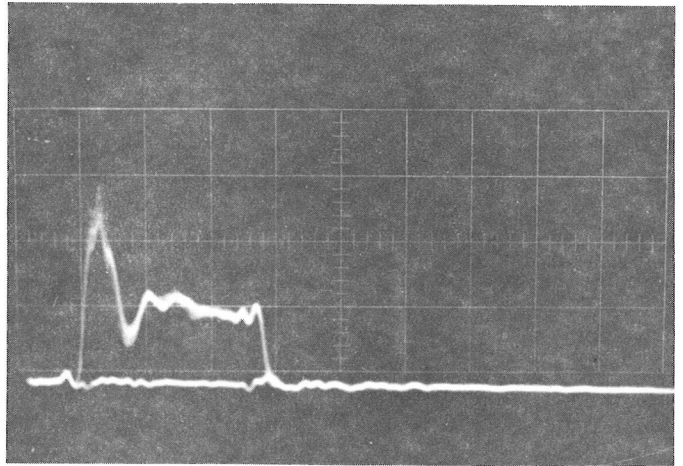


MDT10  
DATA FROM CRS

CURRENT WAVEFORMS

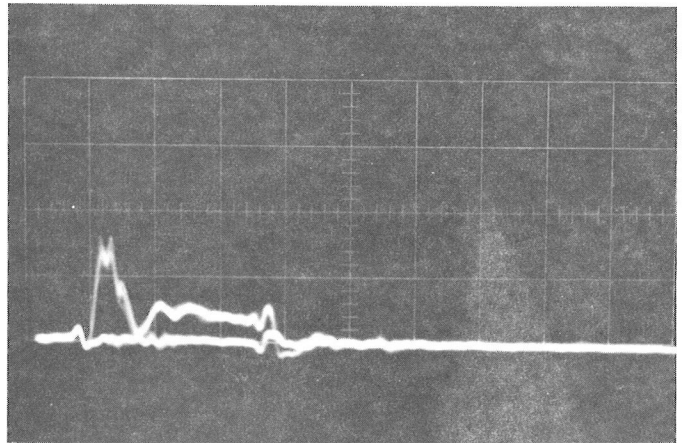
PIN 19 J1 - MM  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $2 \text{ ma}/\text{cm}$   
SYNC = RGWAGC

Fig. 6-18



PIN 14 (110)  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $2 \text{ ma}/\text{cm}$   
SYNC = RGWAGC

Fig. 6-19



MDT10

(DATA FROM ACM) (AMRL10)

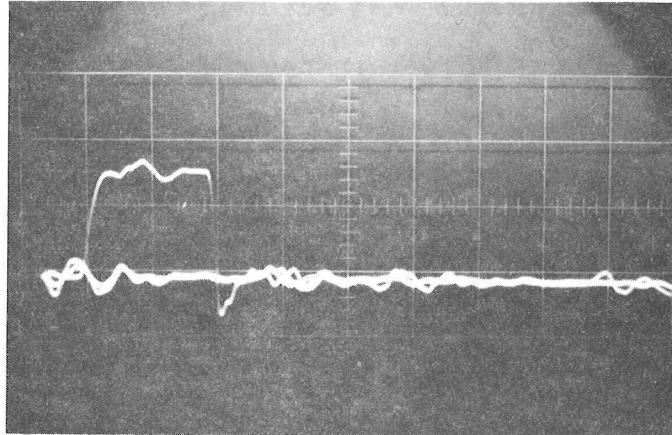
VOLTAGE WAVEFORM

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $2 \text{ V}/\text{cm}$

(Inaccessible with  
current probe)

Fig. 6-20



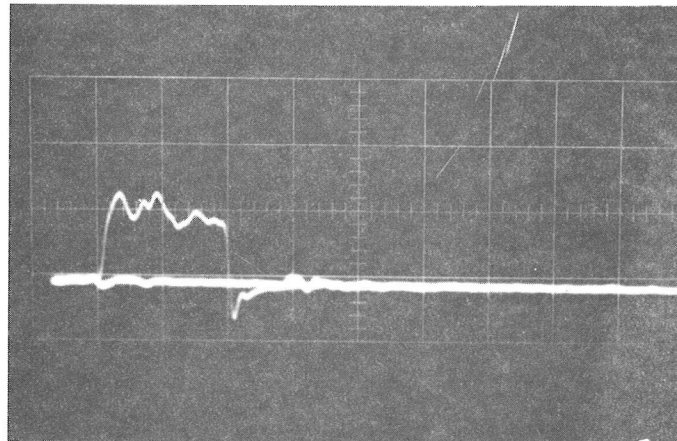
CURRENT WAVEFORM

PIN 14 (110)

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $2 \text{ ma}/\text{cm}$

Fig. 6-21



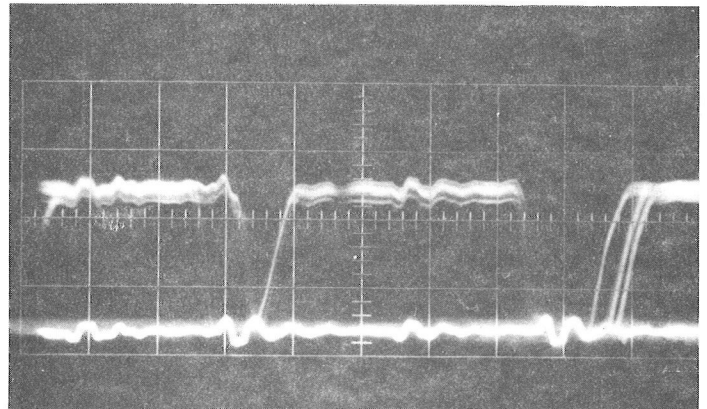


MWL01  
(DATA TO CRS)

VOLTAGE WAVEFORMS

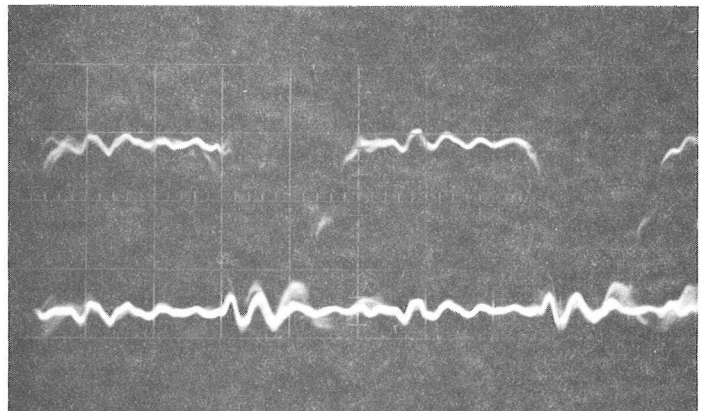
PIN 2 (201)  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $1 \text{ V}/\text{cm}$   
SYNC = WT

Fig. 6-22



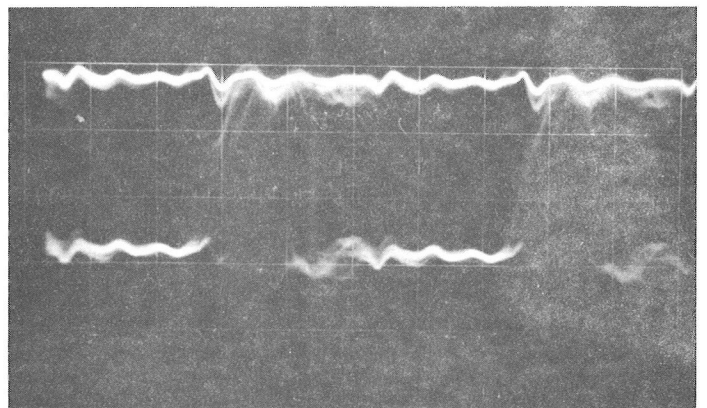
PIN 22 (J3 - x)  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $1 \text{ V}/\text{cm}$   
SYNC = WT

Fig. 6-23



PIN 1 (J7-16) IM ACM  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $2 \text{ V}/\text{cm}$   
SYNC = WT

Fig. 6-24



MWL09

(DATA TO CRS & ACM)

VOLTAGE WAVEFORMS

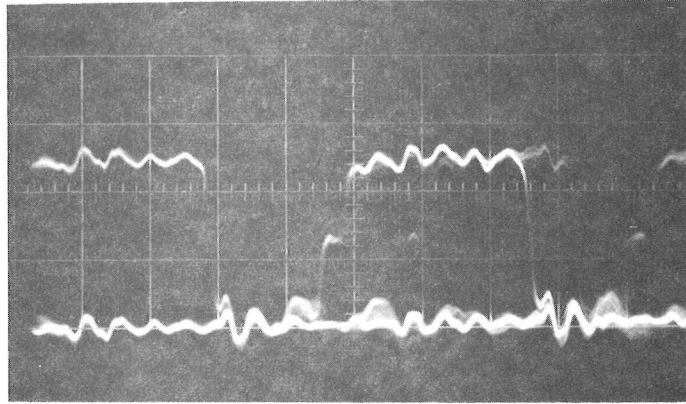
PIN 15 (209)

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $1 \text{ V}/\text{cm}$

SYNC = WT

Fig. 6-25

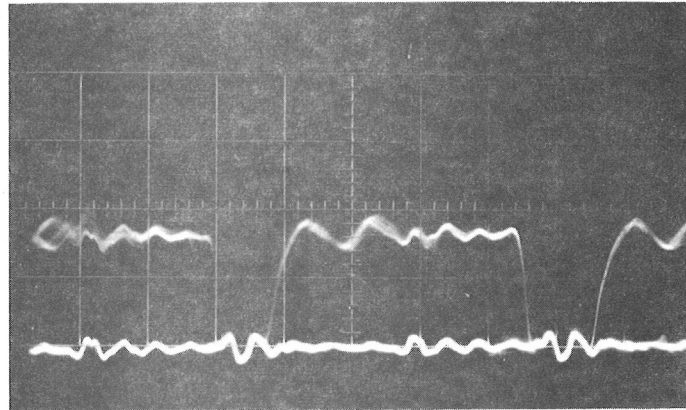


PIN 34 (J3-f)

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $2 \text{ V}/\text{cm}$

Fig. 6-26

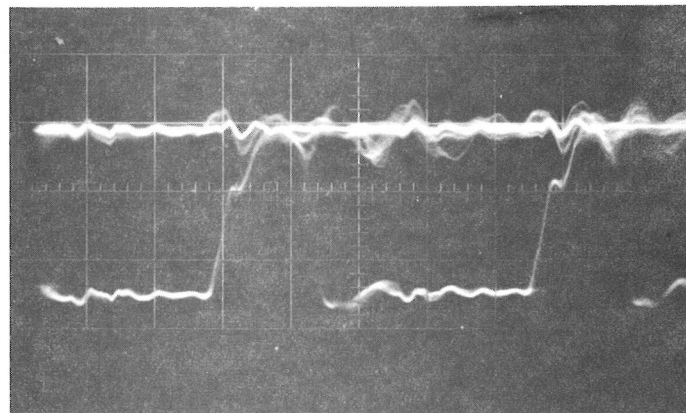


PIN 14 (J7-08) ACM

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $2 \text{ V}/\text{cm}$

Fig. 6-27

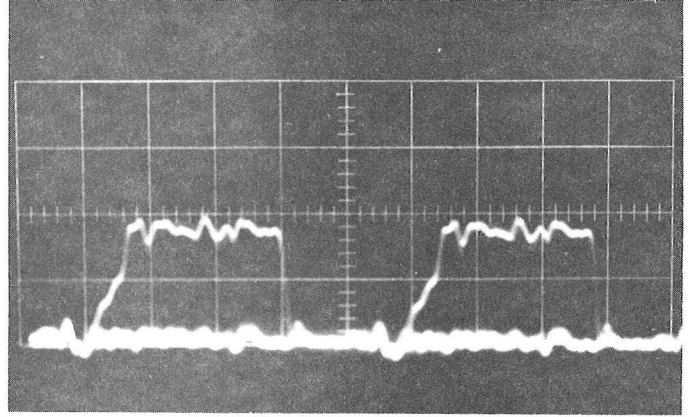


MT04

CURRENT WAVEFORM

PIN 6 (304)  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $1 \text{ ma}/\text{cm}$   
SYNC = WT

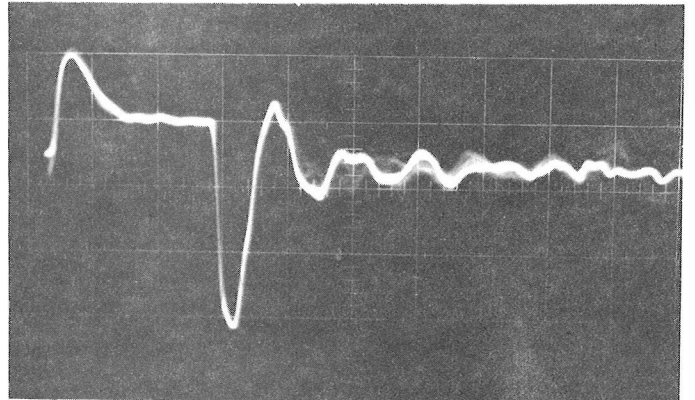
Fig. 6-28



CURRENT WAVEFORM

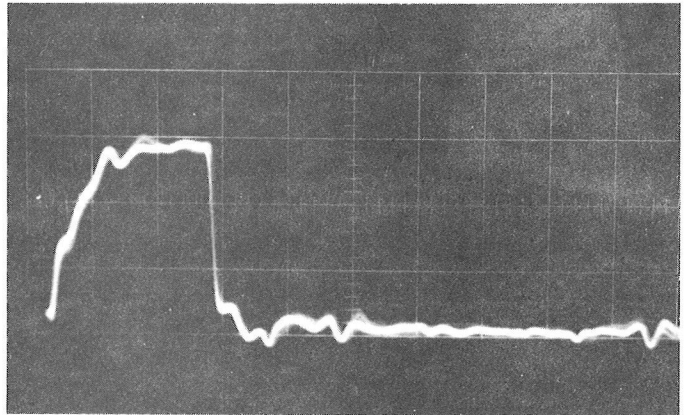
PIN 25 (J3-t) CRS  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $10 \text{ ma}/\text{cm}$   
SYNC = MT04

Fig. 6-29



VOLTAGE WAVEFORM

PIN 25 (J3-t) CRS  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $1 \text{ V}/\text{cm}$   
Fig. 6-29A



MT04

VOLTAGE WAVEFORM

PIN 7 (J8-23) ACM

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $2 \text{ V}/\text{cm}$

SYNC = MT04 (A01/507 CRS)

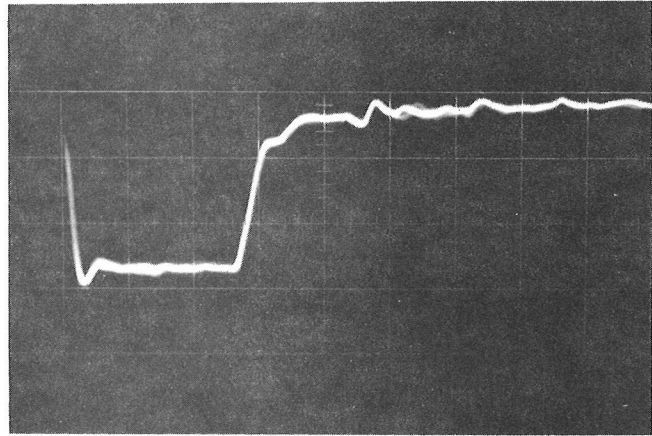


Fig. 6-30

MWSG

VOLTAGE WAVEFORM

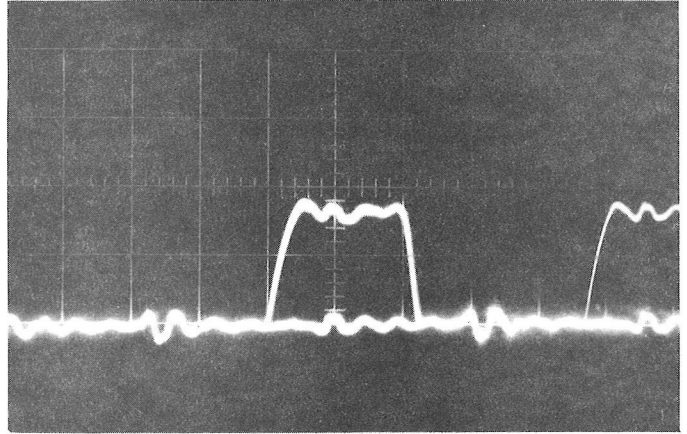
PIN 11 (405)

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $1 \text{ V}/\text{cm}$

SYNC = MWSG (A01/610 CRS)

Fig. 6-31



CURRENT WAVEFORM

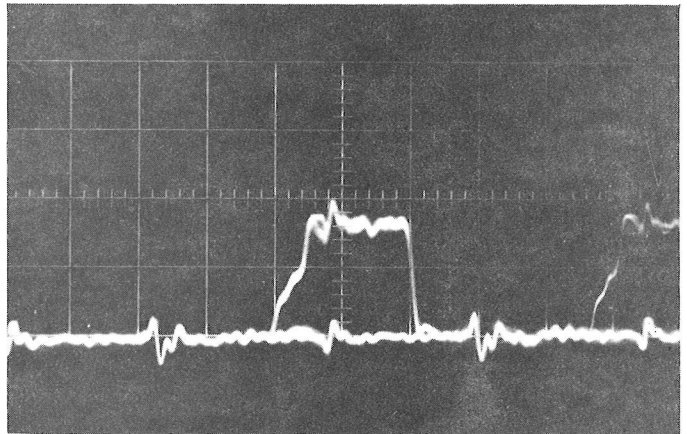
PIN 11 (405)

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $1 \text{ ma}/\text{cm}$

SYNC = MWSG

Fig. 6-31A



VOLTAGE WAVEFORM

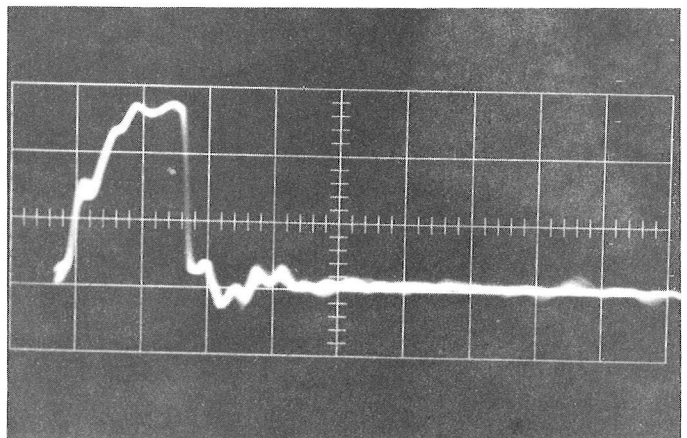
PIN 13 (J2-t)

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $1 \text{ V}/\text{cm}$

SYNC = MWSG

Fig. 6-32



MWSG

CURRENT WAVEFORM

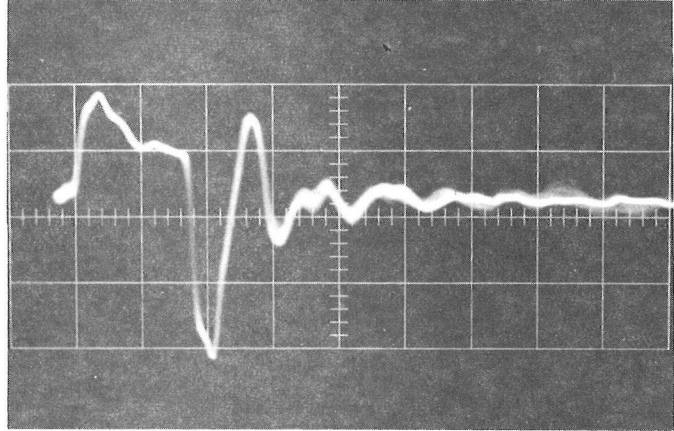
PIN 13 (J2-t)

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $10 \text{ ma}/\text{cm}$

SYNC = MWSG (A01/610 CRS)

Fig. 6-32A



VOLTAGE WAVEFORM

PIN 404 - Module A09 (ACM)

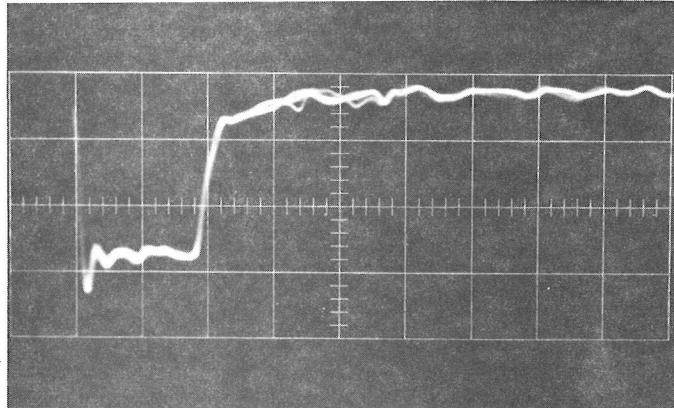
HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $2 \text{ V}/\text{cm}$

SYNC = MWSG

\*Pin 31 (J8-06) Inaccessible

Fig. 6-33



VOLTAGE WAVEFORM

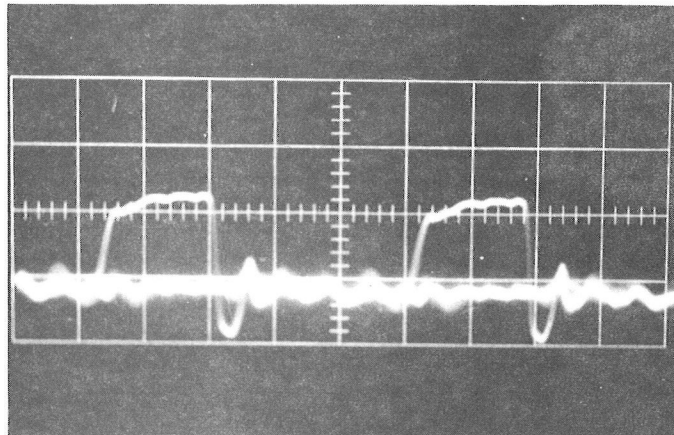
(A01/610) CRS

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $2 \text{ V}/\text{cm}$

SYNC = MWSG

Fig. 6-33A



MONWT

CURRENT WAVEFORM

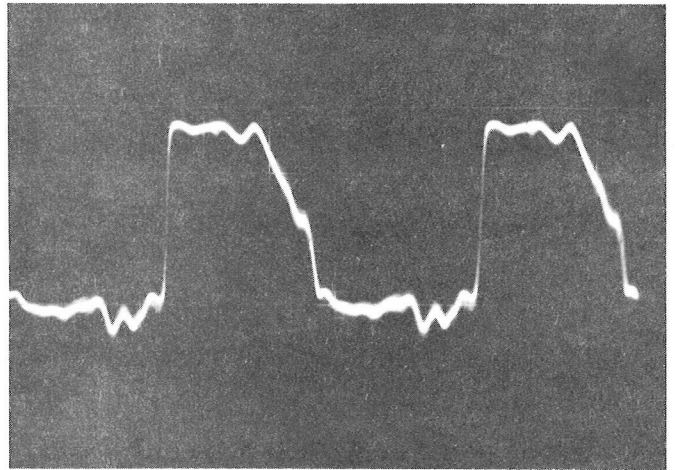
PIN 11 (514)

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $1 \text{ ma}/\text{cm}$

SYNC = MONWT

Fig. 34



CURRENT WAVEFORM

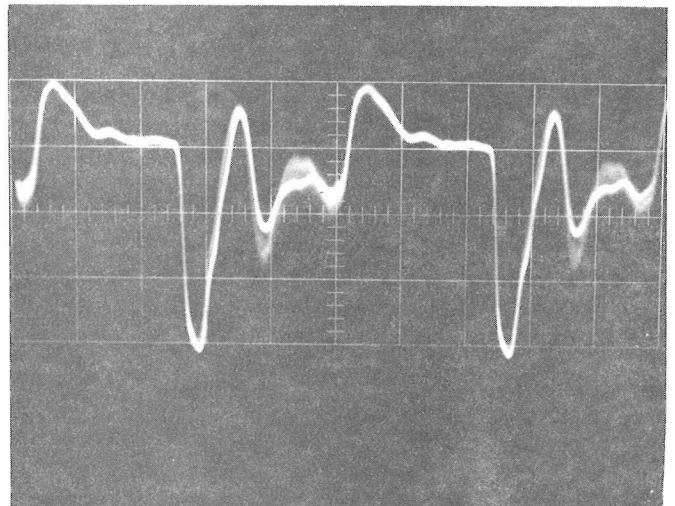
PIN 13 (J3-x)

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $10 \text{ ma}/\text{cm}$

SYNC = MONWT

Fig. 6-35



VOLTAGE WAVEFORM

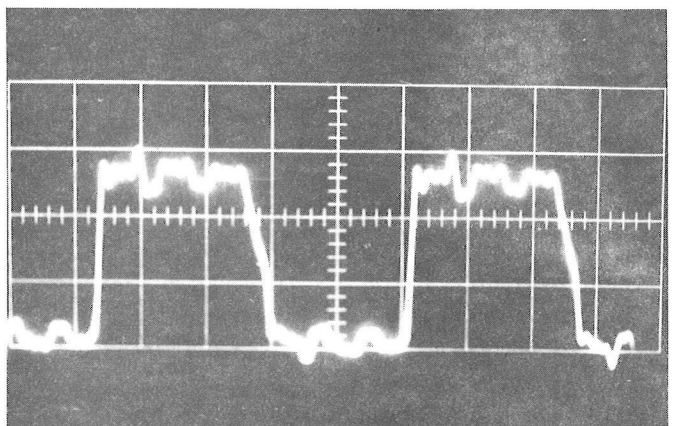
PIN 13 (J3-x)

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $1 \text{ V}/\text{cm}$

SYNC = MONWT

Fig. 6-35A



MONWT

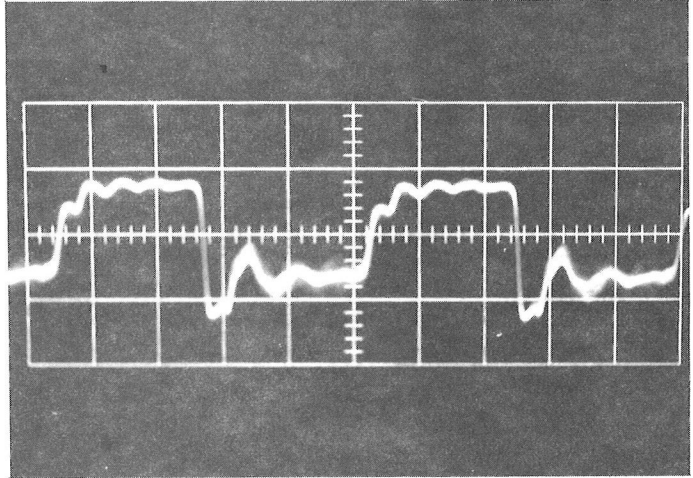
VOLTAGE WAVEFORM

Observed at A01/50 (CRS)  
from Twisted Pair at IM (J8-11)

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $2 \text{ V}/\text{cm}$

Fig. 6-36





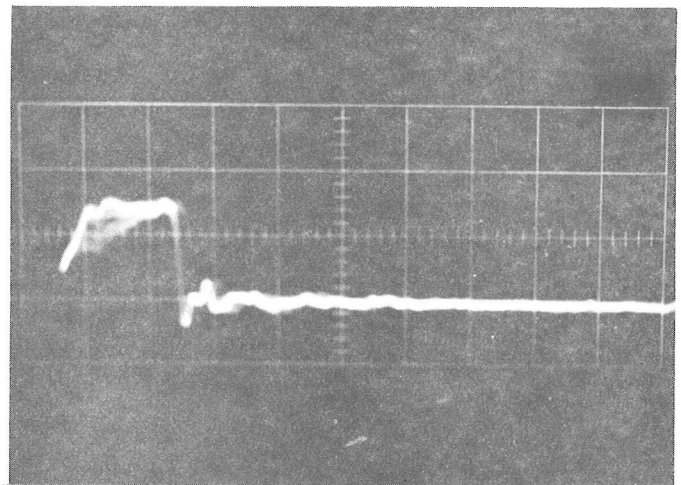
MWQG

CURRENT WAVEFORM

PIN 8 (408)  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $1 \text{ ma}/\text{cm}$   
SYNC = MWQG

Fig. 6-37

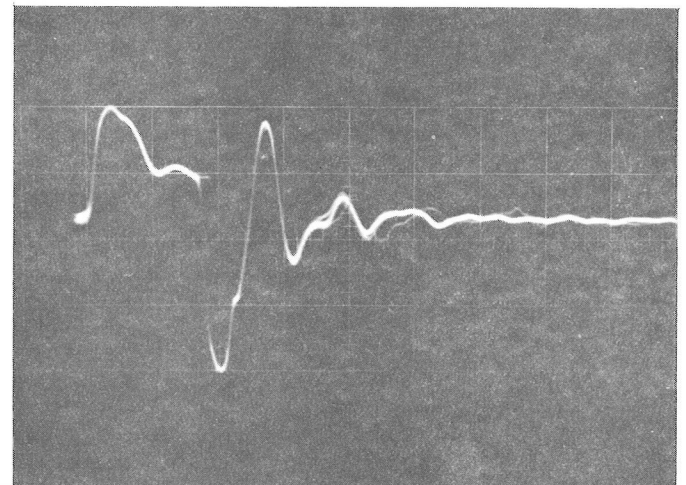
SYNC taken at (A01/609) CRS



CURRENT WAVEFORM

PIN 13 (J2-w)  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $10 \text{ ma}/\text{cm}$   
SYNC = MWQG

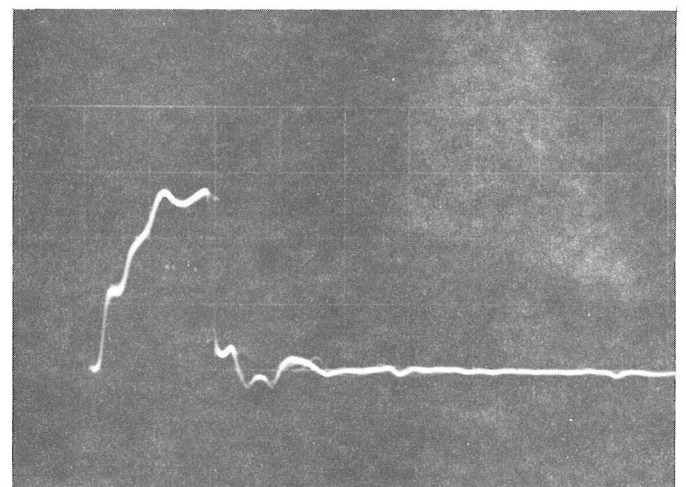
Fig. 6-38



VOLTAGE WAVEFORM

PIN 13 (J2-w)  
HOR =  $0.2 \mu\text{s}/\text{cm}$   
VER =  $1 \text{ V}/\text{cm}$   
SYNC = MWQG

Fig. 6-38A



MNISQ

CURRENT WAVEFORMS

PIN 5 (710)

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $1 \text{ ma}/\text{cm}$

SYNC = MNISQ (A01/617) CRS

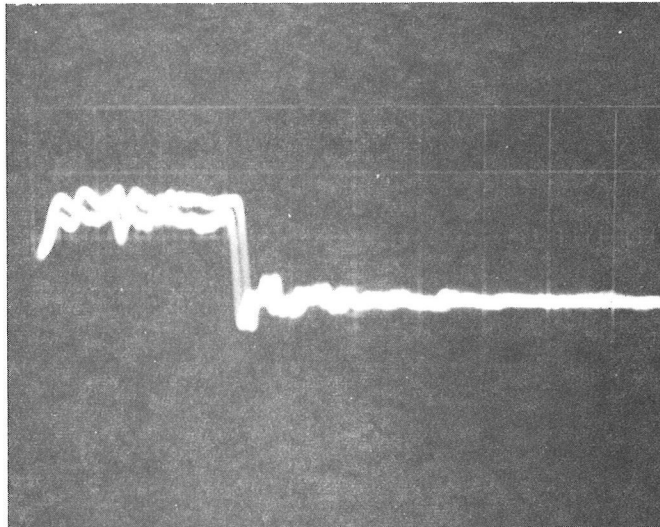


Fig. 6-39

PIN 12 (J2-JJ)

HOR =  $0.2 \mu\text{s}/\text{cm}$

VER =  $10 \text{ ma}/\text{cm}$

SYNC = MNISQ (A01/617) CRS

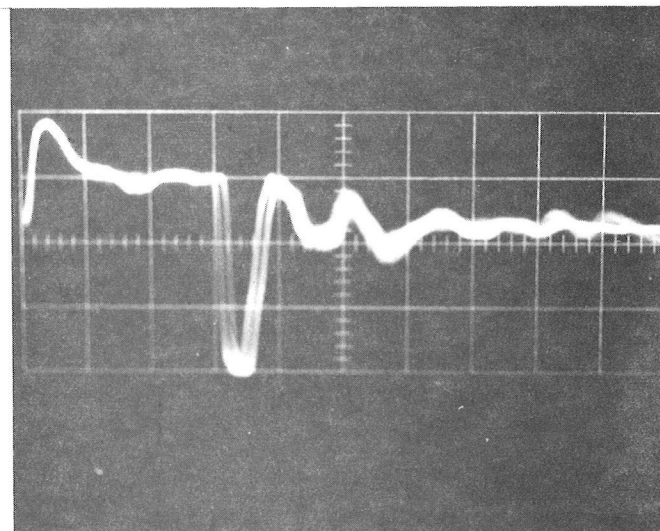


Fig. 6-40

## CHAPTER 7

### SPECIAL STUDIES

#### 7.1 Erasable Versus Fixed Memory

One can partition a computer into three parts.

1. Hardware.
2. Hard software (Braid or rope - fixed memory).
3. Soft software (erasable memory).

The three parts are all related and the relation is the degree of flexibility. Any program or sequence that is not going to change can be put into hardware or hard software. The choice between these two is just a question of which way is most convenient. However, the fact that one does not intend to change this program, or at most only a few times, should be the determining factor between deciding whether it should be put into soft software or hard software. In Apollo there was another factor in that the hard software (the ropes) were much more dense than the erasable memory, so a lot of program was put into the ropes when it might better have been put into erasable. The ratio of rope density to erasable is 5/1. That is, if one had decided to have only erasable memory in the computer with the same volume constraints, one would only have had 9K words of memory versus the 2K erasable and 36K fixed of the present design.

An additional factor one has to consider is that in a fixed memory the memory is not disturbed by electrical transients and one does not have to reload it if a transient occurs or if someone shuts down the equipment improperly. This fact also suggests that the restart or start up programs should be in a rope or a similar type of hardware.

The application for which the program is used and the type of associated equipment available also determine the partition into erasable or fixed memory. If the equipment is not easily reloadable because of operational constraints, etc., or if the time to reload or if the frequency is too high, then a fixed type of storage would be desirable. If the fixed program were fabricated at a reasonable cost, then it might be desirable to go to fixed storage and take the loss of the throw-away item - this obviously is related to how often one plans to do this.

In summary, there is no question but that there is a need for a fixed memory in a computer and that there are many routines that will be fairly stable, such as restart and turn-on procedure, interpreter, executive, conic subroutines, orbital integrations, star catalogues, launch and re-entry routines. However, in determining the trade-off between the fixed and erasable types of storage, one has to consider the advantages of the fixed memory over the erasable.

1. Fixed memory doesn't require loading.
2. It is insensitive to transients.
3. It is denser (more words/cubic in).
4. It is cheaper (Apollo experience, about a factor of two cheaper).
5. It requires less associated equipment.

## 7.2 Speed-Up of the AGC

Several ways of trying to speed up the computer were considered. The obvious one, to actually increase the clock frequency, would have impacted the GSE and Tray A as well as Tray B and for this reason was discarded. A second way that has been used by AGC programmers to speed up programs has been to program in basic rather than interpretive and to use single-precision versus double-precision routines, when one can take the associated degradation in storage or precision. Still another way is to precisely schedule jobs and tasks, but an accurate simulation is required for this type of effort.

The main method of trying to speed up the computer studied under this task was that of trying to speed up the dispatcher part of the interpreter. This could be done by designing a piece of hardware called the PRETBOX which would perform the job done by a program in the interpreter called the dispatcher. Figure 7-1 is a flow chart of the dispatcher showing what part of the program would be done by the PRETBOX hardware. Each word of the interpreter program has to be decoded into a particular operation on an operand. The dispatcher does the decoding and ends up placing the operand or data in the appropriate location and passing control to a subroutine that performs the operation.

### 7.2.1 360 Simulation of the AGC Hardware Dispatcher

A study was done that verified the logical design of the PRETBOX and also allowed one to evaluate how much time was saved by the hardware approach (versus the software).

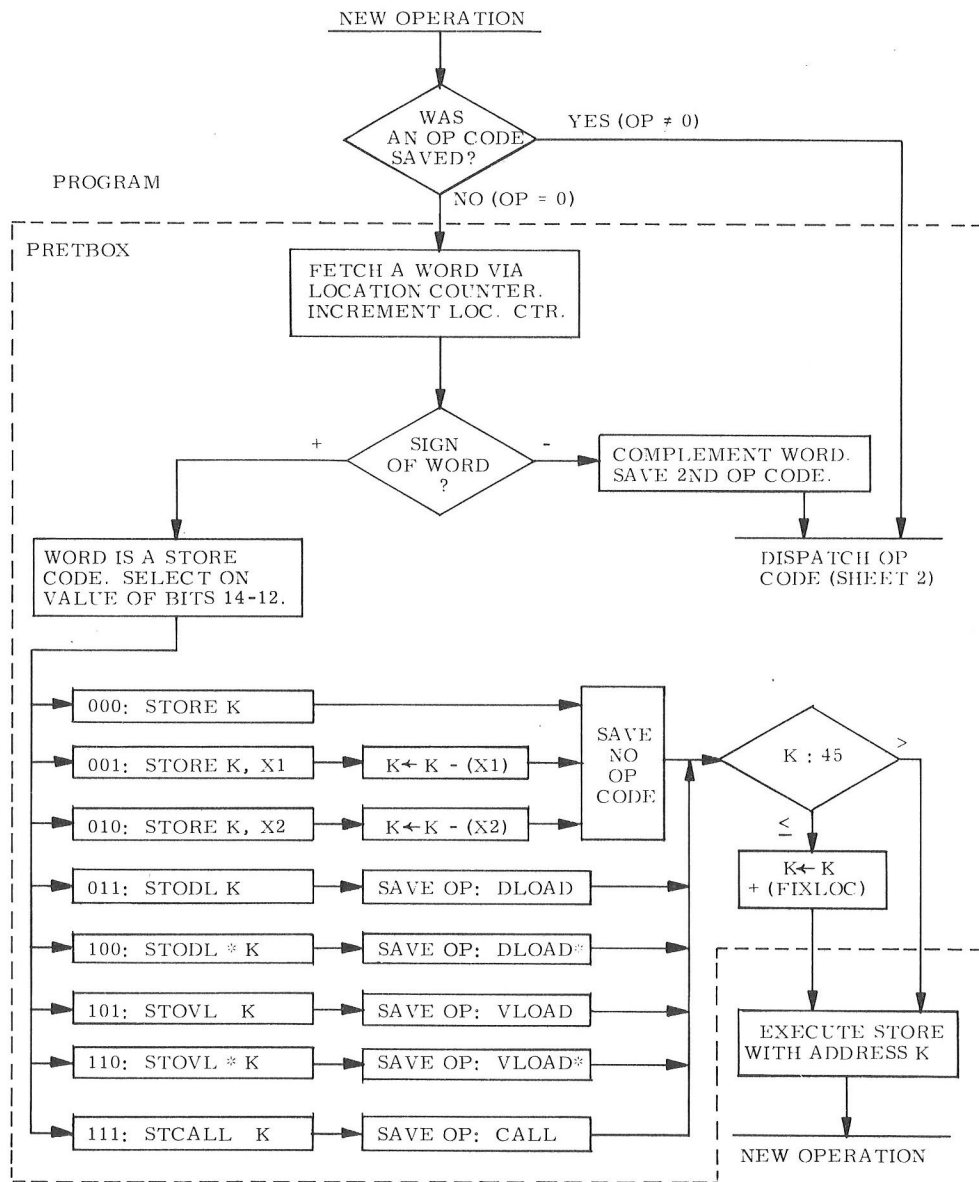


Fig. 7-1a Hardware Dispatcher Flow Chart

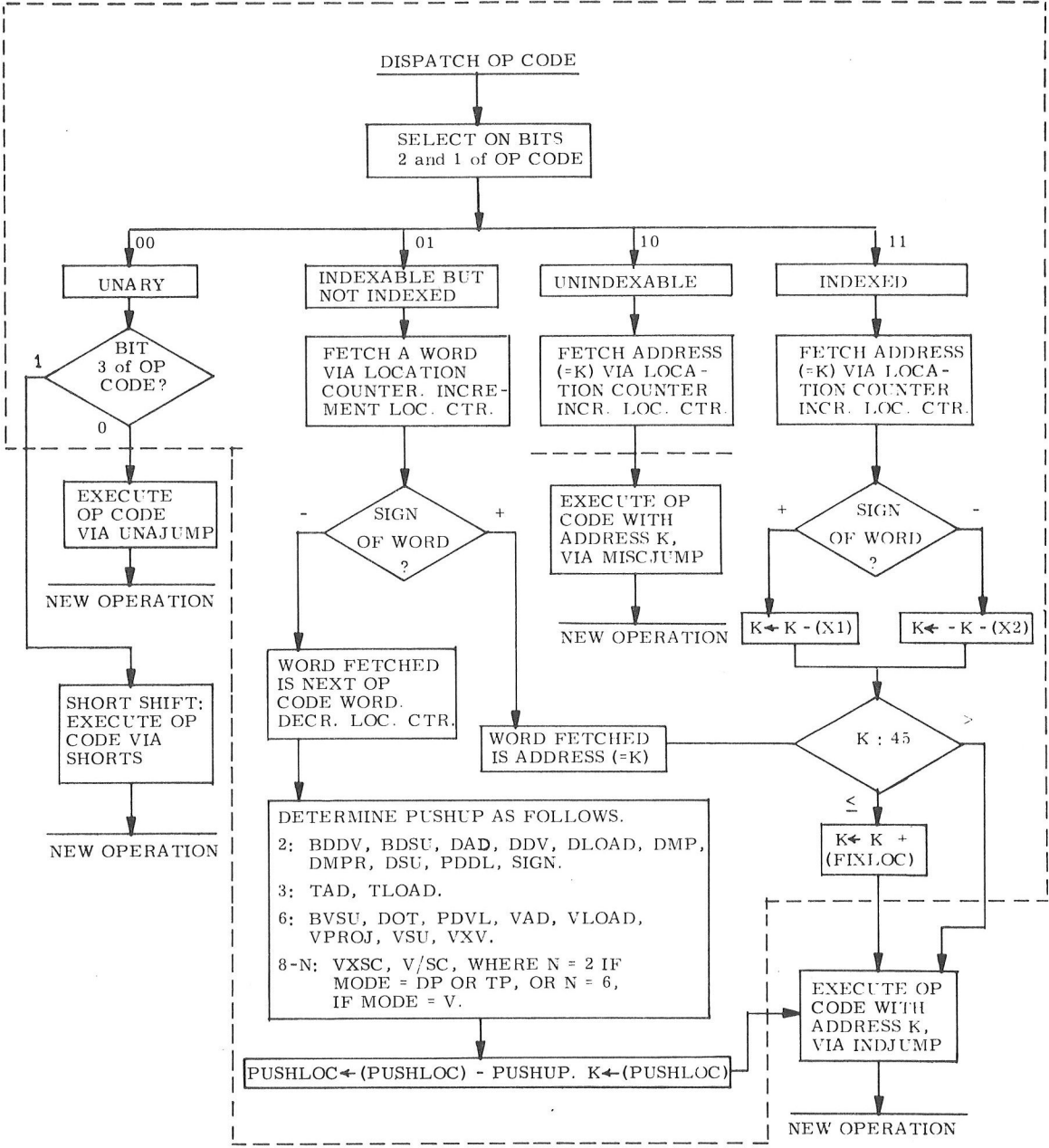


Fig. 7-1a (Cont.)

Since the time spent in various parts of the interpreter is a function of the AGC code being executed, it is desirable to simulate the action of the PRETBOX on the IBM 360 using an actual AGC program (P34) and real environment simulation. The PRETBOX simulator was coded in assembly language as an addition to a frozen version of the standard simulator. Unlike the main simulator, which operates on an instruction-by-instruction level, the PRETBOX was coded in a pulse-by-pulse fashion using 360 macros to define the exact operation of each defined pulse on all the registers and lines in the PRETBOX and the AGC proper.

The resultant 360 code was highly redundant and inefficient but offered the advantages of precise simulation and great flexibility. When changes became necessary in some phase of the PRETBOX design, it was only necessary to modify the macros involved and all affected areas were automatically updated in a consistent manner. Communication with the main simulator was established by reference to the three new AGC channels defined in the PRETBOX logic.

Debugging of the PRETBOX and its simulator began by using the AGC program SAYAH. This proved to be a much too simple-minded test which did not detect a number of existing problems, both in logic and programming. Debugging was continued using a simulation run of Luminary 69. These tests eventually simulated somewhat in excess of 250 AGC seconds of the run.

To evaluate the results, a plot of the duty cycle was made of an identifiable segment of the run which is saturated (Fig. 7-2). For comparison, the same run was reproduced with the software interpreter using the on-line simulator (Fig. 7-2 dashed curve). The selected area of the duty cycle plot was measured in both cases and indicated an improvement of 19% in running time for the PRETBOX.

### 7.2.2 Summary

The PRETBOX logic would be equivalent to 8 AGC logic modules or 1920 gates. This hardware eliminates about 200 words in the dispatcher part of the interpreter program (which is in fixed-fixed), i.e., frees 10% of fixed-fixed and increases the speed of the interpretive programs by 19%. The PRETBOX logic could be put into an auxiliary memory type of expansion but not in a modified Tray B because of a lack of access to the write lines.

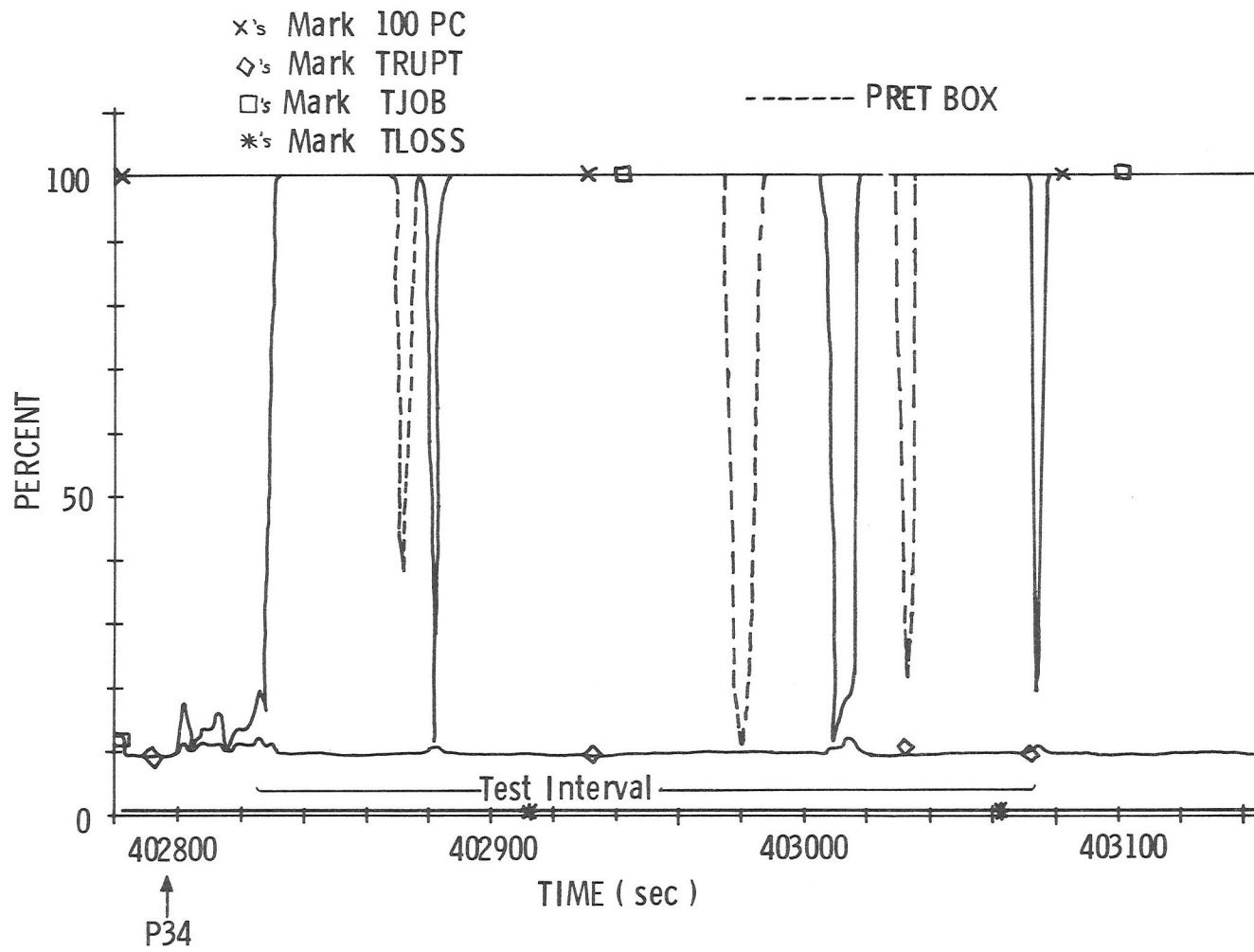


Fig. 7-2 Hardware Versus Software Dispatcher



### 7.3 Hardware Relocation

Hardware relocation was initially proposed as a method of expanding the interpretive erasable addressing capability (Ref. 2 p. 42). This method would have an advantage over the software approach in not slowing down the programs 3% and also allows the programs to be written in a relocatable manner. That is, one could have a resident system (operating system, monitor) which would include the interpreter, executive, restart, waitlist, etc. Segmented programs would then utilize the resident system. The segmented programs would be less efficient with respect to storage than the present Apollo programs but would allow one to have separate assemblies and reduce the possibility of programs interfering with each other. This approach would require a different assembler, digital simulator, and executive than presently exist in Apollo and, therefore, would not be a desirable addition to the Apollo program. It could be useful for post-Apollo work as it would allow various organizations to write their programs independently of one another and the close supervision and memory allocation committees now required would not be necessary. There is a trade-off between ease of programing and minimizing the number of storage locations used. The relocation scheme would use up much more storage.

The circuit shown in Fig. 7-3 is all that is required to implement the relocation features and would not require more than 7 MSI flatpacks. This circuit is designed for an auxiliary memory approach but a similar circuit could be made to work for the Tray-B approach. Essentially, fixed programs would be written starting at bank 50 and the executive would take them off the tape and relocate them in banks where there were inactive programs. The erasable part of the programs would also be relocated at the same time. The top half of Fig. 7-3 implements the fixed memory addressing, (banks 50-67) depicted in Table 7-I, and the bottom half implements the erasable addressing (banks 10-107).

### 7.4 Hardware-Software Facilities Necessary to Utilize the Raytheon Prototype Auxiliary Memory

#### 7.4.1 Minimum Equipment Needed

1. An AGC physically configured so that the interface module (Prudential Tower) can be attached to the test connector.
2. A PAC or CRS.
3. A program that provides certain utility routines which facilitate operating the AM.

TABLE 7-I

TABLE OF CORRESPONDENCE:  
PSEUDO-ADDRESSES, ACM E-BANKS, AND ACM FBANKS

SUPER-BANK 5			SUPER-BANK 6		
PSEUDO-ADDRESS	E-BANK	F-BANK	PSEUDO-ADDRESS	E-BANK	F-BANK
130000-130377	10	50	150000-150377	50	60
130400-130777	11		150400-150777	51	
131000-131377	12		151000-151377	52	
131400-131777	13		151400-151777	53	
132000-132377	14	51	152000-152377	54	61
132400-132777	15		152400-152777	55	
133000-133377	16		153000-153377	56	
133400-133777	17		153400-153777	57	
134000-134377	20	52	154000-154377	60	62
134400-134777	21		154400-154777	61	
135000-135377	22		155000-155377	62	
135400-135777	23		155400-155777	63	
136000-136377	24	53	156000-156377	64	63
136400-136777	25		156400-156777	65	
137000-137377	26		157000-157377	66	
137400-137777	27		157400-157777	67	
140000-140377	30	54	160000-160377	70	64
140400-140777	31		160400-160777	71	
141000-141377	32		161000-161377	72	
141400-141777	33		161400-161777	73	
142000-142377	34	55	162000-162377	74	65
142400-142777	35		162400-162777	75	
143000-143377	36		163000-163377	76	
143400-143777	37		163400-163777	77	
144000-144377	40	56	164000-164377	100*	66
144400-144777	41		164400-164777	101*	
145000-145377	42		165000-165377	102*	
145400-145777	43		165400-165777	103*	
146000-146377	44	57	166000-166377	104*	67
146400-146777	45		166400-166777	105*	
147000-147377	46		167000-167377	106*	
147400-147777	47		167400-167777	107*	

\*Not addressable. Reached through relocation only.

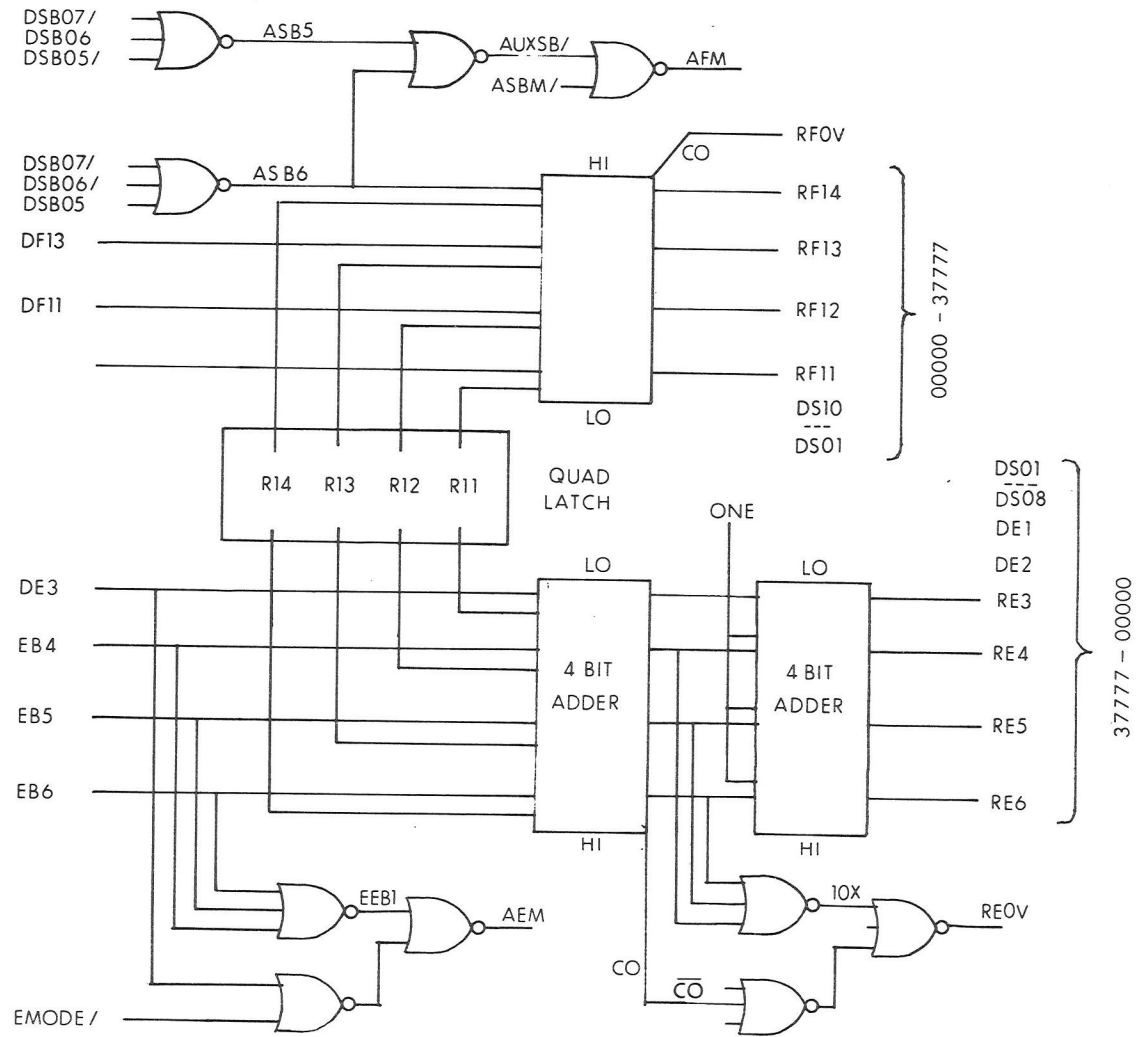


Fig. 7-3 Relocation Hardware

#### 7.4.2 Additional Software Highly Desirable

1. A GAP assembler.
2. A digital simulator.

If one just connected an AM to a G&N system with a regular mission program set of ropes in the computer, one could turn the AM on and off and read locations in the AM as fixed memory. One could also operate the tape unit - read onto the tape and load the AM core memory from the tape. All these operations would be time-consuming, requiring manual loading of various channel addresses. In a short time one would appreciate an appropriate Noun-Verb combination that would do these steps automatically.

If one connected the AM to an AGC and a PAC unit loaded with the Raytheon demonstration tape, one could perform most of the Raytheon demonstration programs, but would not be able to run the G&N system with the demonstration program in the AM.

MIT provided an assembly listing (RUINED REV 4) and a PAC tape which is a revision of Luminary 69. This assembly would allow the following.

1. AGC-AM to restart without turning off power to the AM.
2. Allow nouns 1-3 to reference erasables in the extended erasable banks.
3. Permit one, by the use of Verb 94 Noun 39, to turn the AM on or off, to read from tape to the AM and from the AM to the tape.
4. It will have changes in the interpreter that could allow software relocatable programs to be run.
5. It would have a routine which would allow one to transfer information in banks 44-47 of the PAC into the AM.
6. It would have parts of the lunar landing deleted and one less core set in the executive.

In general, if one wants to use the AM in conjunction with a G&N system, one should have a GAP assembler and a digital simulator. The former would allow one to integrate programs with those available in the mission (this is not always easy to do because the mission programs are very closely interrelated). Thus, if one wanted to have the downlink program store data in the auxiliary memory, one could put in the changes to the mission program and re-assemble it. Then one could call up a program in the AM (Verb 30) and process the data, using the DSKY as an output or any other available AGC interface (which have been suitably programmed). The digital simulator is quite useful, in fact, almost a necessity for debugging programs.

The basic problem in utilizing the AM is that the mission programs are very tightly written and, in order to utilize them effectively, one has to make significant changes in them. That is, in order to make them more generalized, there is a significant impact usually resulting in something being left out of the original mission program. One will have a better feeling for this type of problem after one reads Section 4.6 which illustrates this type of problem.

The auxiliary memory would be quite useful to study different ways of assembling mission programs, segmenting them, making them re-enterable, that is, studying the structures of how best to configure mission programs with regard to assembly, testing, and making use of the external storage. It does present limitations in that it is a specific piece of hardware which has its own constraints.

#### 7.5 Portafam Memory Tray-AGC Operation

One of the topics undertaken was to study the logic required to interface a Portafam memory tray with a single 16K stack to an AGC Tray A.

The Portafam memory cycle is basically a read/restore cycle but can be used as both fixed and erasable as described in Chapter 2. It is necessary to provide an address register to store the erasable read address in a manner similar to the K-cores in the AGC.

The address storage is shown in Fig. 7-4.

If one replaces the corresponding part (BSXX to MADRXX) of Fig. 2-4 with Fig. 7-4 and this together with the logic described in Fig. 2-3 and 2-7 (excluding the external interface logic), one has the logic necessary.

The Portafam memory would do two read/restore cycles for each AGC erasable cycle with the second cycle writing new information into the location.

#### 7.6 Auxiliary Memory Documentation and Operation

- A. General List of Wiring Drawings, Layouts, Schematics and Data on Subunits contained in the Raytheon documentation package for the Auxiliary Memory
  - 1. GGH-9 Wiring Diagram ACM, ATM, and AGC (system interconnection diagram for auxiliary and CRS, with corrections).
  - 2. HD-10 (a 13-page pamphlet of schematics of all signals passing through the IM (Interface Module) commonly referred to as the Prudential Tower).

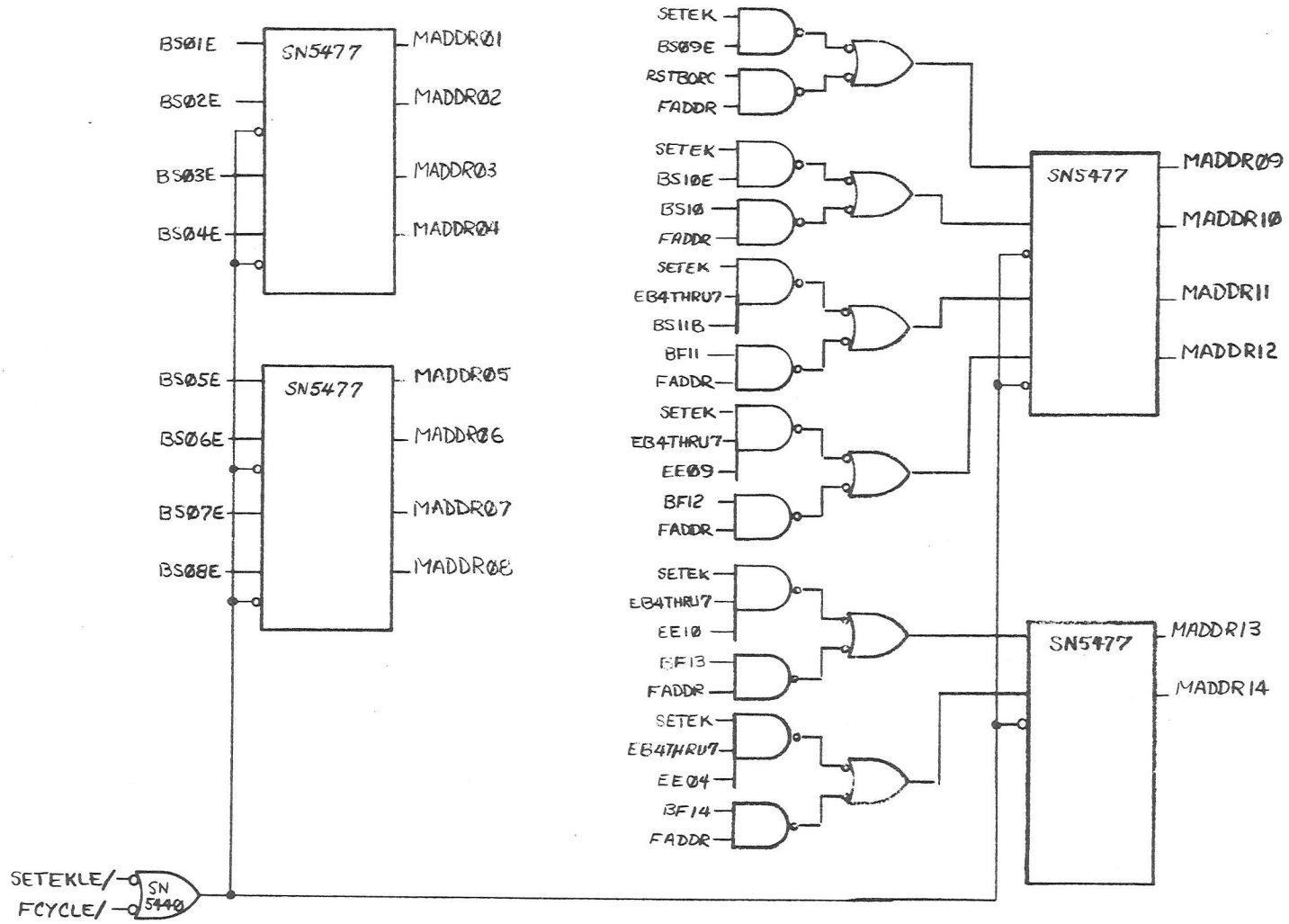


Fig. 7-4 Address Storage: Full Cycle Memory

3. The IM subassembly drawings are included in GGH-24 (Type A), GGH-14 (Type B), GGH-27 (Type C) and GGH-2 (Type D).
4. GGH-22 (mechanical layout of the IM).
5. Schematics of the ACM (Auxiliary Core Memory) modules A01-A11 (these modules are divided up into 1/2 modules and the schematics are arbitrarily indicated as A or B). Also C04=A04, C05=A05, T10=A10. Schematics are also included for the two core memories and the associated memory circuits (sense amps, etc.). Signetics logic flatpacks are used in all logic modules.

NOTE: MATRIX signal-layer-logic half-module prints are available on all modules A02-A11 except A04, A05, and A10. These modules were built on breadboards and the layout is available on page 1-8 Fig. 1-2 of Volume I of the Final Report. All logic can be identified by the matrix prints or the breadboard sideplate layout.

6. The ACM wirelist will furnish signal name, module type (MOD #), module name (1/2 MOD #), and pin numbers.
7. Volumes I and II of the Final report will give detailed information about the auxiliary core memory and the ATM (auxiliary tape memory).

#### B. General Hookup and Operation of the Auxiliary Memory

Connecting up the Auxiliary Memory can be accomplished by following the wiring diagram for the ACM, ATM, and AGC (GGH-9).

Three power supplies must be furnished as indicated on the drawing. DO NOT exceed the voltages indicated because of logic limitations.

The required 10V furnished by the CRS to the IM is zenered to 4.7V in the IM. If the 10V is exceeded, the Dalohm resistors on the IM will overheat. If using the PAC, a fourth power supply may be required. This power supply should be substituted for the 12-15V being supplied by the PAC to the IM (10V).

## CHAPTER 8

### GLOSSARY OF TERMS

#### 8.1 Apollo Guidance Computer (AGC)

This is a 16-bit (15 + parity) general-purpose, single-address, parallel computer. A diagram showing its primary data flow paths is shown in Fig. 8-1.

It has 2040 words of erasable-core memory storage, 8 words of flip-flop storage, and 36,864 words of fixed memory (Rope).

It has 34 basic instructions, 134 interpreter instructions, 13 involuntary instructions, and 5 test instructions. There are 10 levels of hardware interrupt, 19 input counters, 10 output counters, 6 input channels, and 9 output channels.

It has a total of 213 interface circuits besides power and test inputs.

It is operated by means of one or two identical display units (DSKY) which have a numeric keyboard. The DSKY can display 3 words at one time, provide major program, noun, verb, status, and caution information.

Its basic memory cycle time is  $11.7 \mu\text{s}$ . The add time is  $23.4 \mu\text{s}$ , multiply time  $46.8 \mu\text{s}$ , and double-precision add time  $35.1 \mu\text{s}$ .

The computer weighs 70 lb, is less than one cubic foot, and dissipates 70 watts at 28-volt input (15 watts in standby mode).

The DSKY weighs approximately 18 lb and has a volume of 0.2 cubic foot.

The computer has an operating system composed of an executive and waitlist. It is capable of program restart, detecting and recovering from power loss, failure detection, self-check, and alarm initiation.



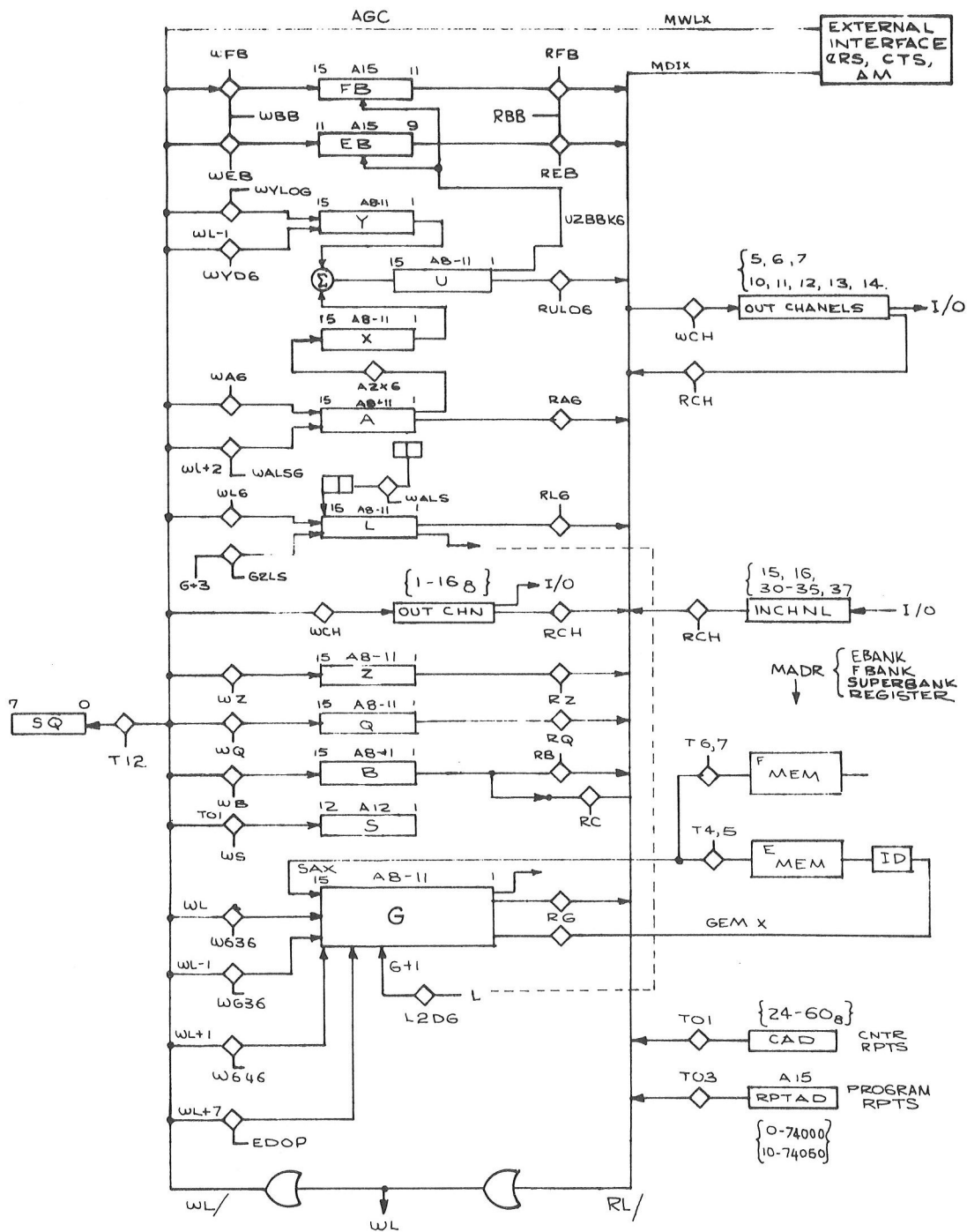


Fig. 8-1 Data Flow Diagram

## 8.2 AGC Addressing

The AGC is essentially divided into two types of memory. Erasable and Fixed.

- a. Erasable memory consists of 2,048 words, the first 8 of which are in flip-flops. The 2,048 words are broken up into 8 E-banks of 256 words each. An address in the range of 1400 to 1777 automatically requires the E-bank (11-9) bits to define its location. Addresses 0000-1377 do not require an E-bank bit and are used for transferring between different E-banks. These addresses are also equivalent to banks 0-2. An ECADR = {0000XXXYYYYYYYY} is used by the programmer in constructing an erasable-bank address. The X bits are written into the E-bank and 1400 is added to the Y bits, giving an address between 1400 and 1777. Thus, an ECADR contains the ingredient for a complete erasable address.
- b. Fixed Memory consists of 36,864 words partitioned into 36 banks of 1024 words each. An address in the range 2000-3777 automatically requires the Fbank bits (15-11). The F-bank bits can be considered as defining banks 00-37. However, if the fixed bank is in the range 30-37, the Superbank is required. There are 3 bits defining Superbank addresses 0-7. The convention is that the Superbank only defines banks 30,40. The 2048 fixed addresses, banks 2 and 3, are addressable also as 4000-7777. These latter addresses do not utilize the fixed or Superbank bits in locating the addressed register and are used for bank switching routines. An FCADR = XXXXXYYYYYYYYYYY is used by the programmer in constructing a fixed-bank address. The X bits are loaded into the F-bank and 2000 is added to the Y bits, giving an address between 2000 and 3777. To set the fixed bank and erasable bank bits and the Superbank bits, the programmer can use BBCON = XXXXX---YYY-EEE. If one writes Both-Bank, the Xs go into the fixed bank register and the Es go into the E-bank register. Similarly, if one reads the Both Bank register, the bank bits go into the respective positions on the write lines. The Ys are used for writing into the Superbank and reading out of it. Thus, this type of word takes care of all the bank settings.
- c. PSEUDO Address. The YULE and GAP assemblers relate each AGC address to a particular octal pseudo address. For erasable banks 0-7 the pseudo address corresponds directly to the ECADR. PSEUDO = (EEEEAAAAAAAA), where the Es represent the E-bank bits and the As represent the lower 8 bits of the 1400-1777 address.

For the fixed-bank bits the fixed-fixed locations 4000-7777 map 1-1. The 2000-3777 addresses map as follows:

PSEUDO = SSSFFFAAAAAAAAA + 4000, where S stands for the Superbank setting, F for the three lower bits of the F-bank, and A for the lowest 10 bits of the 2000 + address. Actually, the F's and A's comprise the low 13 bits of the FCADR. As can be seen, the fixed-fixed addresses (4000-7777) have two representations in the PSEUDO addressing just as in the AGC addressing. This is not true of the unswitched erasable (0000-1377).

To go from the PSEUDO address to the AGC bank address one just performs the inverse operations indicated above.

### 8.3 Assembler

This is a program that provides conversion from a program in a source language (programming language composed of symbols) to a program in machine language (object, target program).

In the process of doing this it assigns computer-memory addresses to the instruction and operands and provides a program listing and diagnostics.

### 8.4 Assembler YULE

This is an assembler for Apollo computer programs which operate on a Honeywell 1800 computer complex. It also provides wiring information for the fixed memory rope along with the normal assembler outputs.

### 8.5 Assembler (GAP)

This is an assembler for Apollo computer program which operated on an IBM 360-75 complex. It also provides wiring information for the fixed-memory rope along with the normal assembler outputs. A LUMINARY assembly listing has better than 1700 pages of printout.

### 8.6 Auxiliary Memory

This is usually used in reference to the Raytheon prototype unit used in this study. It consists of 8192 words of erasable capable of being extended to 16K and

an associated tape unit capable of storing simplex or triplex words. It interfaces with the AGC through the AGC test connector. The 16K of memory is capable of being addressed as fixed or erasable. It has fixed-bank 50-67 and erasable-bank addresses 10-107. It is controlled by the AGC utilizing AGC channel addresses 20-27. A data-flow schematic is shown in Fig. 8-2.

#### 8.7 BBCON

See AGC addressing above.

#### 8.8 BOTH BANK

See AGC addressing above.

#### 8.9 BRAID

This is a nonalterable type of memory which uses linear cores to couple between the word line and the output bit. If the word line does not loop the core, there is no output on the bit line.

#### 8.10 Buffer Box

This is a piece of hardware that is attached to the AGC test connector during factory and subsystem test. It is an interface adapter unit providing matching and noise rejection between the GSE and the computer test-interface micrologic.

#### 8.11 Channel

The AGC interface discrettes or status signals are combined into words whose address is designated by a channel number. There are input and output channels. There are special instructions in the AGC for performing operations on the channel contents.

#### 8.12 CRS (Core-Rope Simulator)

This is a piece of equipment that is capable of interfacing with the AGC by means of a test-connector buffer box. Different versions have from 36-40K of erasable memory capable of being addressed as AGC fixed. (Fig. 8-3)

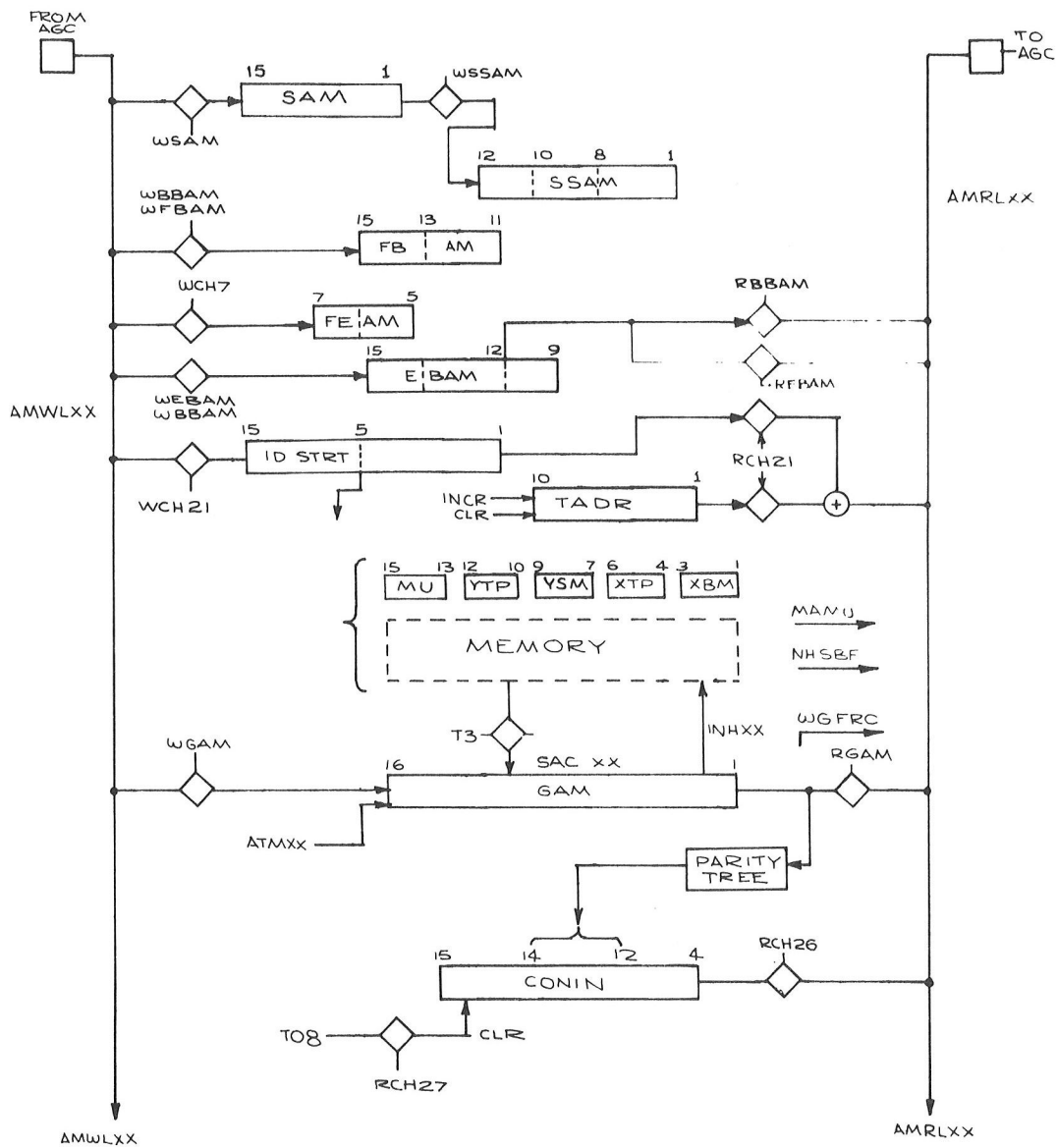


Fig. 8-2 Auxiliary Memory Data Flow Diagram

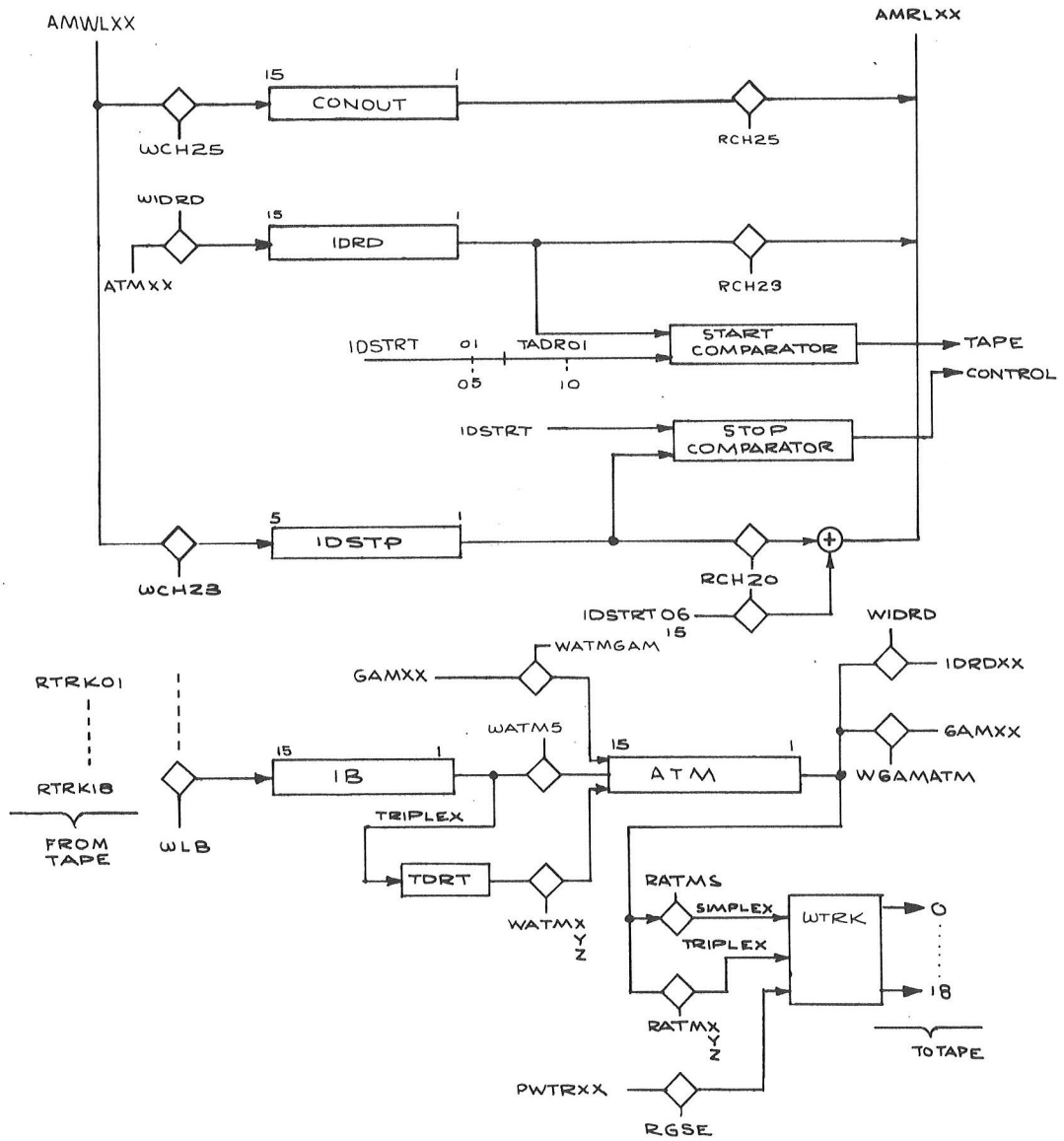


Fig. 8-2 (cont)

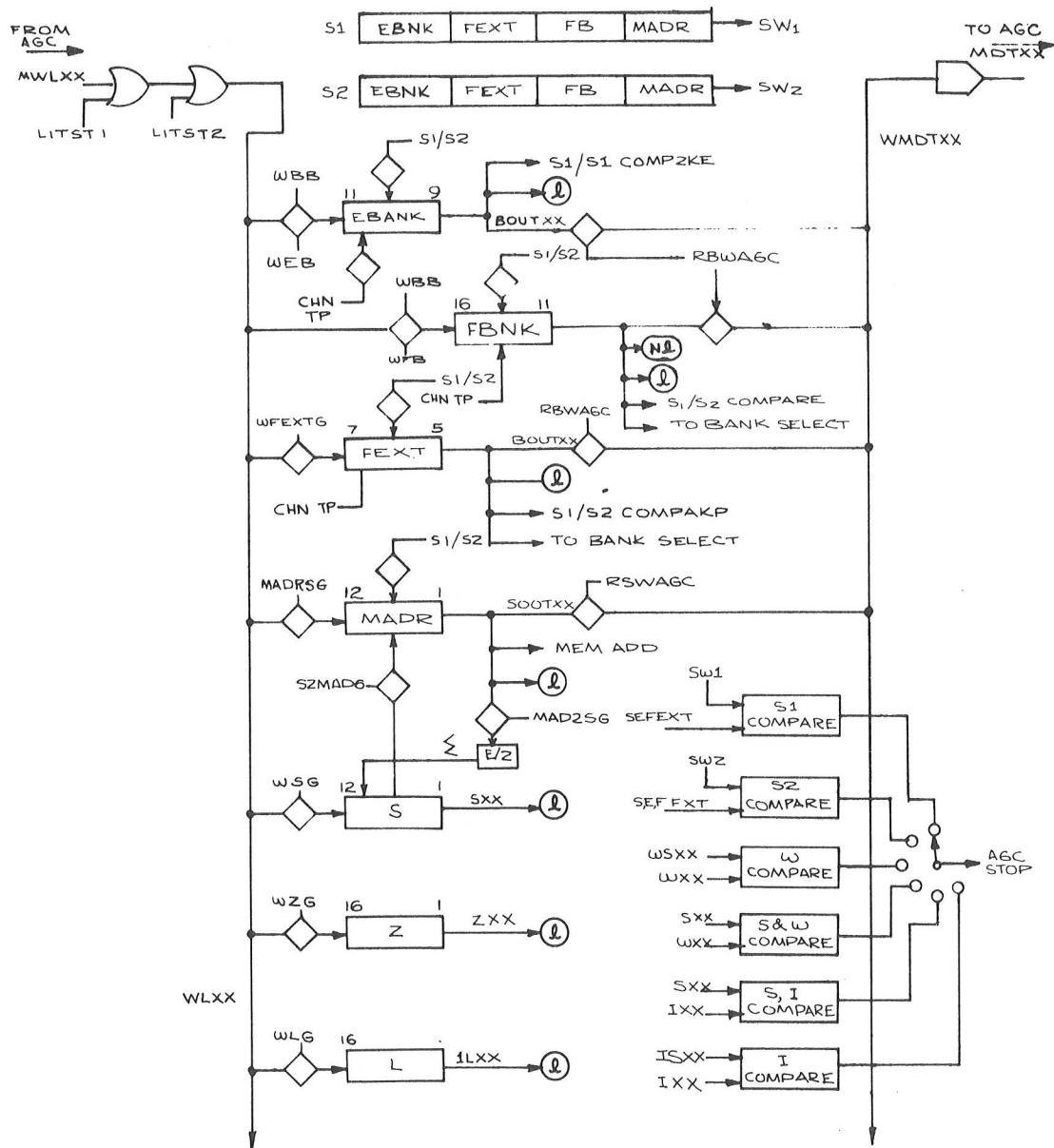
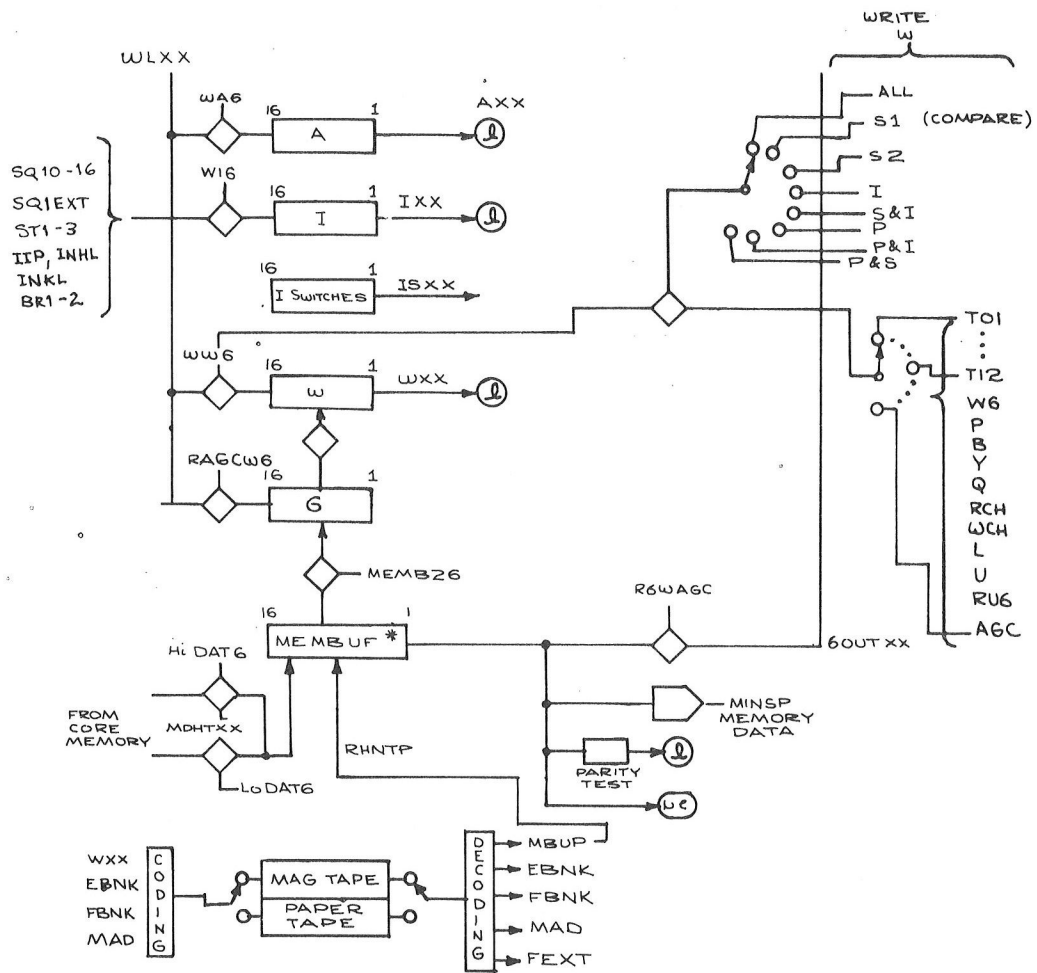


Fig. 8-3 Core Rope Simulator Data Flow Diagram



\* Old CRS's have no MBUF only G register  
 also old CRS's have 36K instead of 40K of memory

\*\* Magnetic or Paper Tape

Fig. 8-3 (cont)



The memory can be loaded by either paper tape or magnetic tape and one can also load the 2K of AGC erasable. A data-flow diagram of the CRS is shown in Fig. 8-3. The CRS is used to check out fixed programs before they are built into ropes and for monitoring the AGC operation. It is capable of monitoring several registers in the AGC; the "CRS" is sometimes called the "Monitor".

#### 8.13 CMC (Command Module Computer)

Usually one calls the Apollo Guidance Computer the AGC but since the only difference between the command-module computer and the LM Guidance Computer is the program in fixed and erasable memories, in order to distinguish between the two we have the names CMC and LGC.

#### 8.14 Digital Simulator

This is a software program operating on the IBM 360 that accepts AGC programs (CM or LM) and executes them on a simulated AGC instruction-by-instruction, simulating the flight environment inputs, etc. It provides either graphical or printed diagnostics and the results of the interaction of the program with the environment.

#### 8.15 EBANK

See AGC addressing above.

#### 8.16 Memory Cycle Time

This is the amount of time it takes one to do a fixed-memory access or an erasable-memory read-write operation. It is  $11.7 \mu s$ .

#### 8.17 Erasable Memory

2048 locations in the AGC capable of being modified and used for storage.

#### 8.18 FBANK

See AGC addressing above.

#### 8.19 FCADRE

See AGC addressing above.

#### 8.20 Fixed Memory

This refers to the 36,864 words of memory storage in the AGC that can not be modified. One can read the memory location but not write into it.

#### 8.21 Interpreter

This is a program that translates source-language expressions into machine language and immediately executes them.

#### 8.22 LGC (LM Guidance Computer)

This is the guidance computer in the Lunar Module. It differs from the CMC only by the program in its 6 rope modules.

#### 8.23 Multiprocessor

This term is usually used in referring to a computer that has two or more arithmetic units capable of simultaneously and independent operation.

#### 8.24 PAC (Program Analyzer Console)

This is a core-rope simulator built by Raytheon for field use. It has a paper tape and keyboard input. (FR-65-336A Program Analyzer Console, April '66, Raytheon Co.)

#### 8.25 Portafam

This is a piece of equipment that interfaces with the AGC through the rope-module connectors. It has 48K words of erasable memory, 36K of which can be used to simulate the fixed-rope memory and 12K which can be used for program tracing. It has a magnetic tape unit (IBM compatible) which is used for loading and reading the 36K erasable memory and for recording the 12K trace memory.

## 8.26 Relocation

This word usually refers to a program which is written so that a base address can be added to all its memory references without affecting its operation. This effectively allows one to place the program anywhere in memory. The bank structure makes the AGC a natural for this type of operation.

## 8.27 Restart Monitor

This is a module designed to be attached to the test connector interface. It stores failure indications that cause restart in channel 77 and is not reset by the restart process.

## 8.28 Restart

This is a program in fixed memory that allows the computer to recover from certain types of intermittent failures or transients. It does this by requiring the programmers to partition the programs into restartable segments.

## 8.29 Rope

This term is used in reference to the nonalterable memory presently in the AGC. Each core corresponds to 12 AGC words. A square-loop core is threaded by several wires that are used to set the core in the "1" state. When the core is reset to the "0" state, it will read out a "1" onto a particular bit line if the wire associated with that bit threads the core; otherwise no signal will be coupled onto the line corresponding to a zero. Each core is capable of storing 12 AGC words (= 12 x 16 wires) and there are 512 cores in each rope module.

## REFERENCES

1. "Auxiliary Memory for Apollo Guidance Computer", Report R68-4125, Contract NAS9-5994, Raytheon Company S&ID, 11 March 1968.
2. R. Alonso and W. Aldrich, "The 'Braid' Transformer Memory", Report R-498, Instrumentation Laboratory, MIT, Cambridge, Massachusetts, 1965.
3. MIT Report R-600, "Control, Guidance and Navigation for Advanced Manned Missions", Vol. II, "Computer Subsystems", Chapter 4, "Multiprocessor Computer Subsystems", J. McKenna et al, January 1968.

E-2463

DISTRIBUTION LIST

R. Ragan	J.H. Laning
D. Hoag	P. Mimno
R. Millard	F. Glick
R. Battin	E.C. Hall
N. Sears	A. Hopkins
R. Crisp	D. Hanley
A. Laats	D. Bowler (30)
G. Cherry	A. Green
G. Stubbs	H. Blair-Smith
J. Harper	M. Johnston
J. Vittek	R. Filene
T. Brand	J. McKenna
M. Hamilton	W. Weinstein
J. Kernan	J. Allen
D. Fraser	J. Martin
J. Nevins	R. Howie
T. Fitzgibbon	J. Leavitt
J. O'Connor	K. Grigs
J. Gilmore	G. Jones
R. O'Donnell	R. Scott
T. Lawton (MSC)	G. Cross
G. Reasor (MSC)	R. Kendall
M. Adams	J. Marino
P. Felleman	R. Weatherbee
G. Silver	Apollo Library (2)
G. Edmonds	MIT/CSDL Library (10)
R. Lones	

External:

NASA/RASPO (1)

Raytheon (6)

R. Zagrodnick

A. Lee (2)

E. Atkins (2)

T. Rogers

ACED (2)

J. Weber

C. Benes

MSC:

National Aeronautics and Space Administration (65&1R)

Manned Spacecraft Center

Houston, Texas 77058

ATTN: Apollo Document Control Group (BM 86) (18&1R)

M. Holley (2)

T. Gibson (1)

J. Williams (5)

I. Burtzloff (5)

G. Xenakis (5)

K. Cox (5)

C. McCullough (5)

T. Chambers (1)

G. Marlowe (1)

C. Frasier (1)

C. Levy (1)

W. Young (1)

E. Lee (1)

S. House (1)

E. Chevers (10)

H. Howard (1)

P. Sollack (1)

KSC: (1R)

National Aeronautics and Space Administration

J. F. Kennedy Space Center

J. F. Kennedy Space Center, Florida 32899

ATTN: Technical Document Control Office