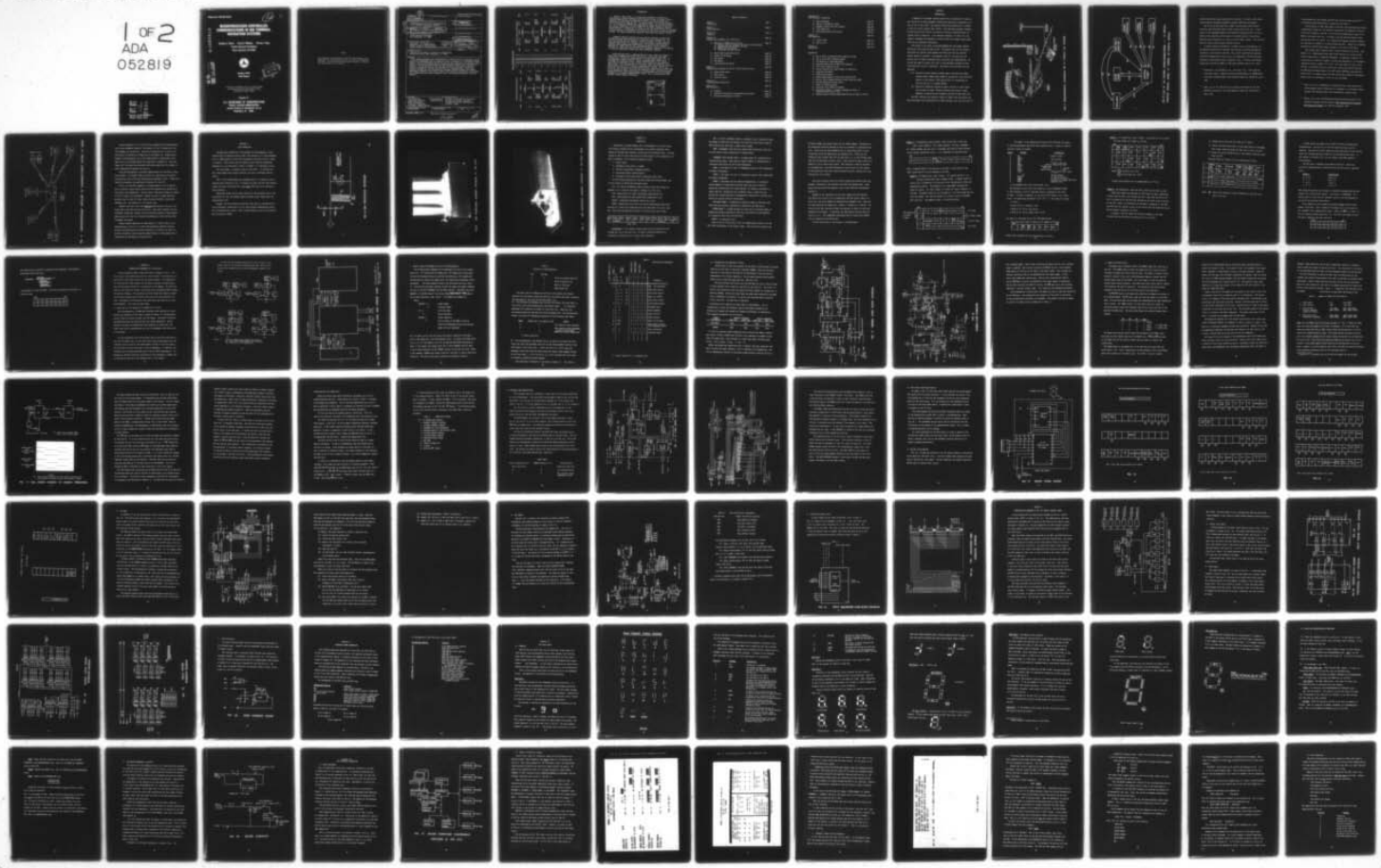
AD-A052 819 PURDUE UNIV LAFAYETTE IND SCHOOL OF ELECTRICAL ENGI--ETC F/G 17/2
MICROPROCESSOR-CONTROLLED COMMUNICATIONS IN AIR TERMINAL NAVIGA--ETC(U)
JAN 78 S E BELTER, C R WILLIAMS, S C BASS DOT-FA-74WA-3518

UNCLASSIFIED FAA-RD-78-25 NL

1 OF 2
ADA
052819



Report No. FAA-RD-78-25

(12)

MICROPROCESSOR-CONTROLLED COMMUNICATIONS IN AIR TERMINAL NAVIGATION SYSTEMS

Stephen E. Belter Craig R. Williams Steven C. Bass

Purdue Research Foundation

West Lafayette, IN 47907

AD A052819

AD No. DDC FILE COPY

DDC
APR 17 1978
F

January 1978

Final Report

Document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161.

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION

FEDERAL AVIATION ADMINISTRATION

Systems Research & Development Service

Washington, D.C. 20590

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FAA-RD-78-25		

4. Title and Subtitle
Microprocessor-Controlled Communications in Air Terminal Navigation Systems.

5. Report Date
January 1978

6. Performing Organization Code

7. Author(s)
S. E. Belter, C. R. Williams, S. C. Bass

8. Performing Organization Report No.

9. Performing Organization Name and Address
Purdue Research Foundation
West Lafayette, IN 47907

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
DOT-FA-74WA-3518

12. Sponsoring Agency Name and Address
Department of Transportation
Federal Aviation Administration
Systems Research and Development Service

13. Type of Report and Period Covered
Final rept.
1 Nov 75—31 Dec 77

14. Sponsoring Agency Code
ARD-741

15. Supplementary Notes
Prepared by the School of Electrical Engineering
Purdue University, West Lafayette, IN 47907

16. Abstract

This report documents the implementation of some old and new techniques to the problem of command-and-control signaling within a Category III instrument landing system. The old techniques include the use of tone signaling over balanced lines, along with isolation transformers and gas-discharge elements for lightning protection. The new methods include the application of microprocessor control devices to supervise, format, and interpret all communications. Additional features afforded by the microprocessor approach include automatic and manual maintenance logging at the control tower, more reliable transmission error detection, enhanced system status displays, and a self-contained operator training feature.

17. Key Words	18. Distribution Statement
Tone Signaling Lightning Protection Noise Suppression Signaling Techniques Instrument Landing System Microprocessor Communications	Document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	104	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

292000

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
	LENGTH			
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
	AREA			
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
	MASS (weight)			
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
	VOLUME			
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
	TEMPERATURE (exact)			
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
	LENGTH			
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
	AREA			
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
	MASS (weight)			
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
	VOLUME			
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
	TEMPERATURE (exact)			
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F

BIOGRAPHIES

Steven C. Bass, Ph.D., is an Associate Professor of Electrical Engineering at Purdue University where he specializes in Digital Filters, Computer-Aided Design, and Nonlinear Circuits. He has published and consulted in these areas (as well as digital design) for several outside firms including the Magnavox Co., the Admiral Corp., and the North Electric Co. He is secretary of the IEEE Committee on Signal Processing (Circuits and Systems Group) and is also a member of the IEEE Committee on Solid-State Circuits. In addition, he is the current editor of "Circuits & Systems," a light technical and newsletter publication of the IEEE Circuits and Systems Society. Dr. Bass received his doctorate from Purdue University in 1971.

Stephen E. Belter received the B.S. and M.S. degrees in electrical engineering from Purdue University in 1973 and 1974, respectively. He was also awarded the M.S. degree in Computer Science in 1976. His professional activities and interests are in the areas of microprocessors, computer-aided design, and broadcasting. He has done consulting for Magnavox Co., North Electric Co., and Admiral Corp. He is a member of Tau Beta Pi, IEEE, AES, and the Society of Broadcast Engineers. Mr. Belter is currently employed by Cyberlabs, West Lafayette, Indiana.

Craig Williams, Ph.D., received the B.S., M.S., and Ph.D. degrees in electrical engineering at Purdue University in 1971, 1973, and 1976 respectively. During the summer months from July 1968 to June 1971, he was employed at Naval Avionics Facility at Indianapolis (NAFI) as a draftsman and technical writer. After graduating with a B.S. degree in June 1971, he continued to work at NAFI as an electrical engineer until August 1973. This included work on a weather radar system and the analysis and testing of boards in a digital communications system. During the summer of 1973 he worked at the Naval Electronics Laboratories at San Diego as an Engineer-on-Loan. This work was concerned mostly with the design of hardware-software interfacing between a microprocessor and various system components in a digital communication system. He is currently with TRW Systems, Redondo Beach, California.

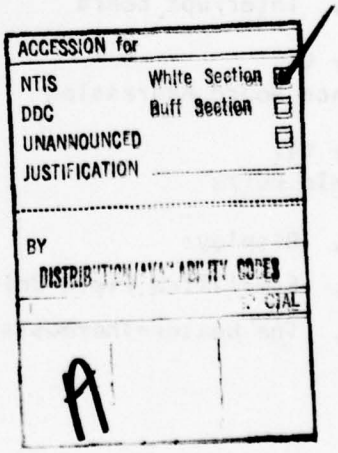


Table of Contents

Chapter I — Page 1
Introduction

Chapter II — Page 8
Line Protection

Chapter III — Page 12
Signaling

Chapter IV — Page 21
Communication Hardware at a Field Unit

A. Operation of MODEM and I/F Cards — Page 21
   1) Generic Types of Commands used by the Microprocessor
   2) Transmission and Reception of Data
   3) MODEM Card Description
   4) Interface Card Description

B. Front Panel Board Description — Page 43

C. Barrier Strip Boards — Page 45

D. A/D Boards — Page 49

E. D/A Boards — Page 53

F. Input Multiplexer Board — Page 56

Chapter V — Page 58
Communication Hardware at the Air Traffic Control Tower

A. Output Latch Board — Page 60

B. Relay Board — Page 60

C. Interrupt Board — Page 64

Chapter VI — Page 65
Interface Board Addressing

Chapter VII — Page 67
The Field Units

A. Displays — Page 67

B. Simplified Field Unit Troubleshooting Procedure — Page 75

C. The Heater-Thermostat Circuitry — Page 77

Chapter VIII
ATCT Operator Information

A. Basic Equipment Page 79
B. Monitor Information Frames Page 81
C. Commands Issued from the Keyboard Page 84
D. Error Reporting Page 89
E. Performance Rating Time Delays Page 93

Chapter IX
The Computers Page 94

A. Intel's 4040 Page 94
B. DEC's LSI-11 Page 100

Chapter X
Conclusions Page 102

Appendices

A. How to Power up the ATCT Tone Signaling System
B. How to do an ATCT "Software Reset"
C. The Meaning of a Watchdog Reset
D. Detailed Description of PDP-11/03 Software
E. Field Unit Program Listing
F. How to Identify and Insert PROM's in 4040 Units
G. Localizer D/A Board
H. AD7522 D/A Converter
I. Quick Change Boxes at Field Units and the ATCT
J. Barrier Strip Commons Mounted Inside Signaling Drawer
K. Bussing Required on Relay Cards
L. Theory of PROM Checksums
M. Isolating Faulty PDP-11/03 PROM's
N. Snow White Program - Assembly Language Listings of Program Patches; March, 1977
O. Communications System Changes Made During the Week of 3/6/77

CHAPTER I

INTRODUCTION

A commercial instrument landing system (ILS), as depicted in Figure 1, may involve up to seven equipment installations physically separated by as much as six to ten miles. The central controlling installation is placed at the air traffic control tower (ATCT). The remaining equipment elements, called "field units" (FU's), are located in shelters scattered about the landing field in question. In an advanced "Category III" (CAT III) ILS, Figure 2 indicates the placement of the 87 status and command signal paths that must be constantly maintained.

The design of the Texas Instruments/Thomson-CSF Instrument Landing System has its strong and weak points. The poorest part of the system is the status and control links between the field units and the control/monitor panels at the ATCT. As designed, a CAT III ILS uses over 100 buried wires or leased telephone lines to provide this communication. No attempt was made to protect the lines or the equipment attached to them from electrical noise or transients. The result can be divided into two symptoms:

(1) Electrical noise caused by nearby power lines and even runway landing lights induce false alarms on the monitor lines and false signals on the control lines causing the ILS transmitters to cycle on and off in a random fashion.

(2) Electrical transients caused by nearby (5 miles or less) lightning strikes are again induced on monitor and control lines, damaging or destroying the equipment attached to them (the ILS).

The FAA, realizing the potential danger to human lives and seeing the high maintenance costs associated with these problems, has, with the help of

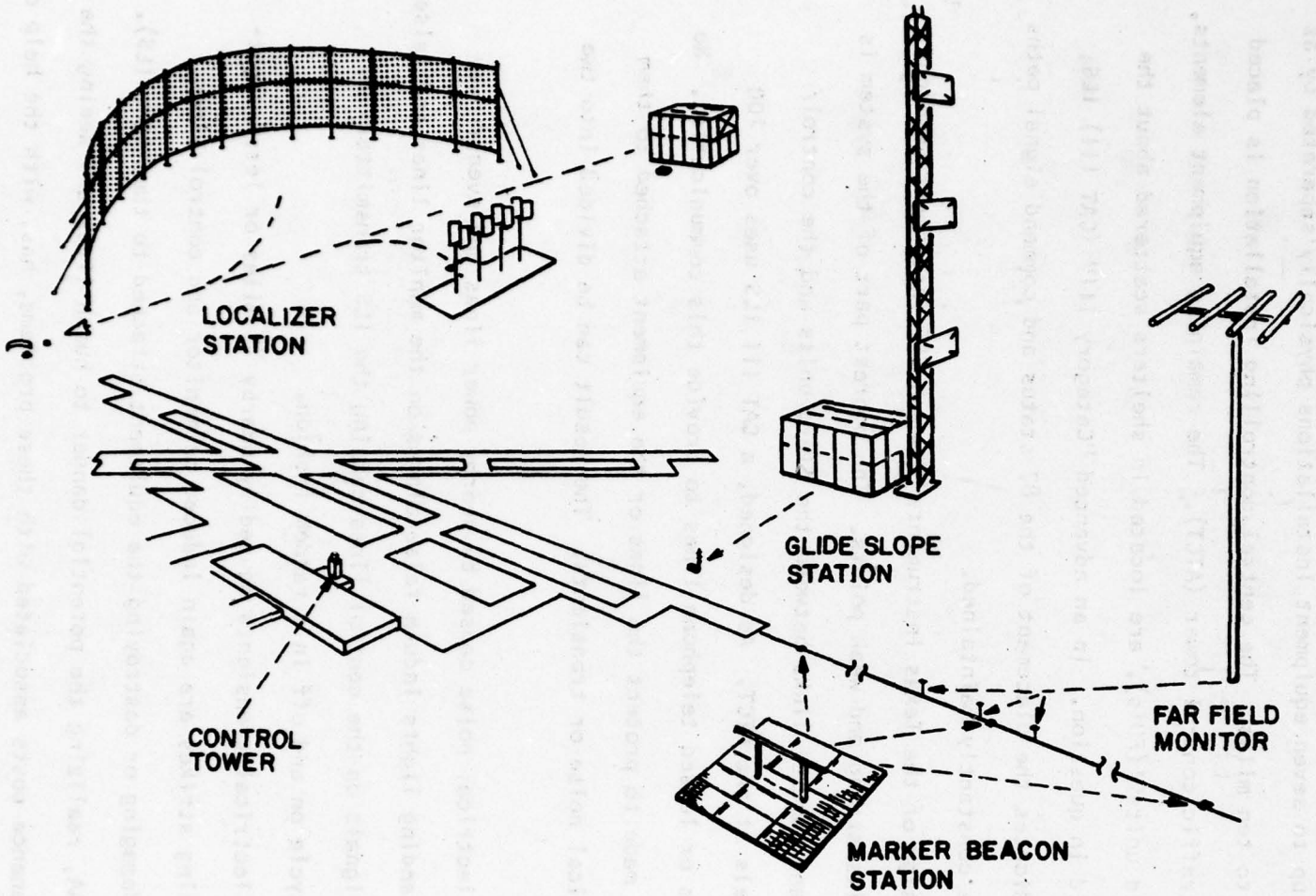


FIG. 1 GEOGRAPHICAL DIAGRAM OF A TYPICAL ILS SYSTEM

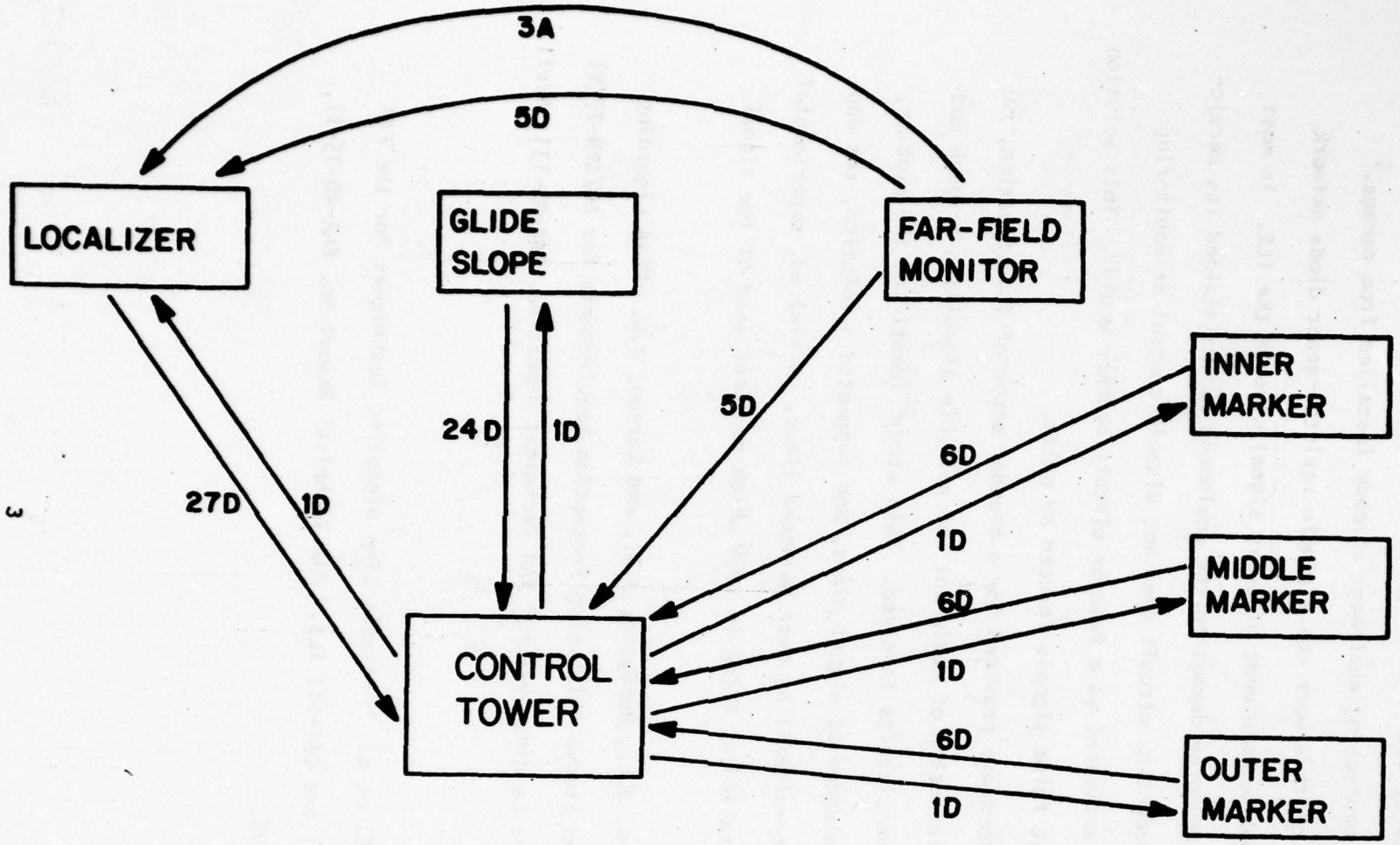


FIG. 2 CAT III ILS SYSTEM INTERCONNECTIONS --- "A" MEANS ANALOG SIGNAL, WHILE "D" REFERS TO DIGITAL SIGNAL

outside contractors, been searching for solutions. Initially, this effort concentrated on protecting equipment already installed from damage.[1]

The result of this work was a simple resistor-zener diode network which was applied to both ends of every signal wire in the ILS. In most cases, it has prevented damage to the equipment and maintained its servicability (the protection circuit does not disable control or monitoring functions when activated as a fuse or circuit breaker would). This solution does not prevent false signals induced by noise.

A second contract provided for a broader scope of investigation, requiring both protection of equipment and reliable signalling through modification of the circuits involved. This study[2] identified the problem, the use of unbalanced DC signal wires, and suggested a solution, the application of tone-signalling over balanced lines. A limited, experimental system was installed at NAFEC in 1975 which replaced some of the signal

[1] Huddleston, G.K., Nordgard, J.D., and Larson, R.W. "FAA Lightning Protection Study: Lightning Protection Requirements for AN/GRN-27(V) Instrument Landing System," FAA Technical Report No. FAA-RD-131, April 1974.

[2] Bass, S.C. et al, "Reliable Line Signaling Techniques for the FAA GRN-27(V) and CAT-III ILS," FAA Technical Report No. FAA-RD-75-11, March 1975.

wires between the Inner Marker and ATCT and the Glide Slope and ATCT.[3,4] A complete system installation is covered by this report.

During August of 1976, the authors installed a microprocessor-controlled communication system for the reliable transmission and reception of monitor and control signals at the CAT III ILS currently operating on runway 13 of NAFEC, at Atlantic City, NJ. In Figure 3, we see that the overall architecture was that of a "star": All communication between equipment elements was required to initiate or to pass through the ATCT. Signals not ultimately destined for the ATCT were simply relayed by the control tower gear.

Each signal path of Figure 3 was a single twisted pair of wires. (See Chapter II). Communication over these lines occurred as a serial stream of audio tone bursts of specified frequency. As detailed in Section III, a tone burst is 1/300 of a second in length and (depending on its frequency) represents either a logical "1" or "0". The logic level, in turn, could indicate the "on/off" value of a field unit status, or could be reserved for showing the up/down status of an ATCT cycle pushbutton, or could be a single "bit" in a numeric value of a DDM (differential depth of modulation) current level, as expressed in units of, say, microamps.

[3] Bass, S.C. et al, "Application of Balanced Lines, Tone Signaling and Microprocessor Control Techniques to a Category III Instrument Landing System," FAA Technical Report No. FAA-RD-76-24, February 1976.

[4] Better, S.E. et al, "Microprocessor-Controlled Communications in an Advanced Instrument Landing System," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-12:6, November 1976.

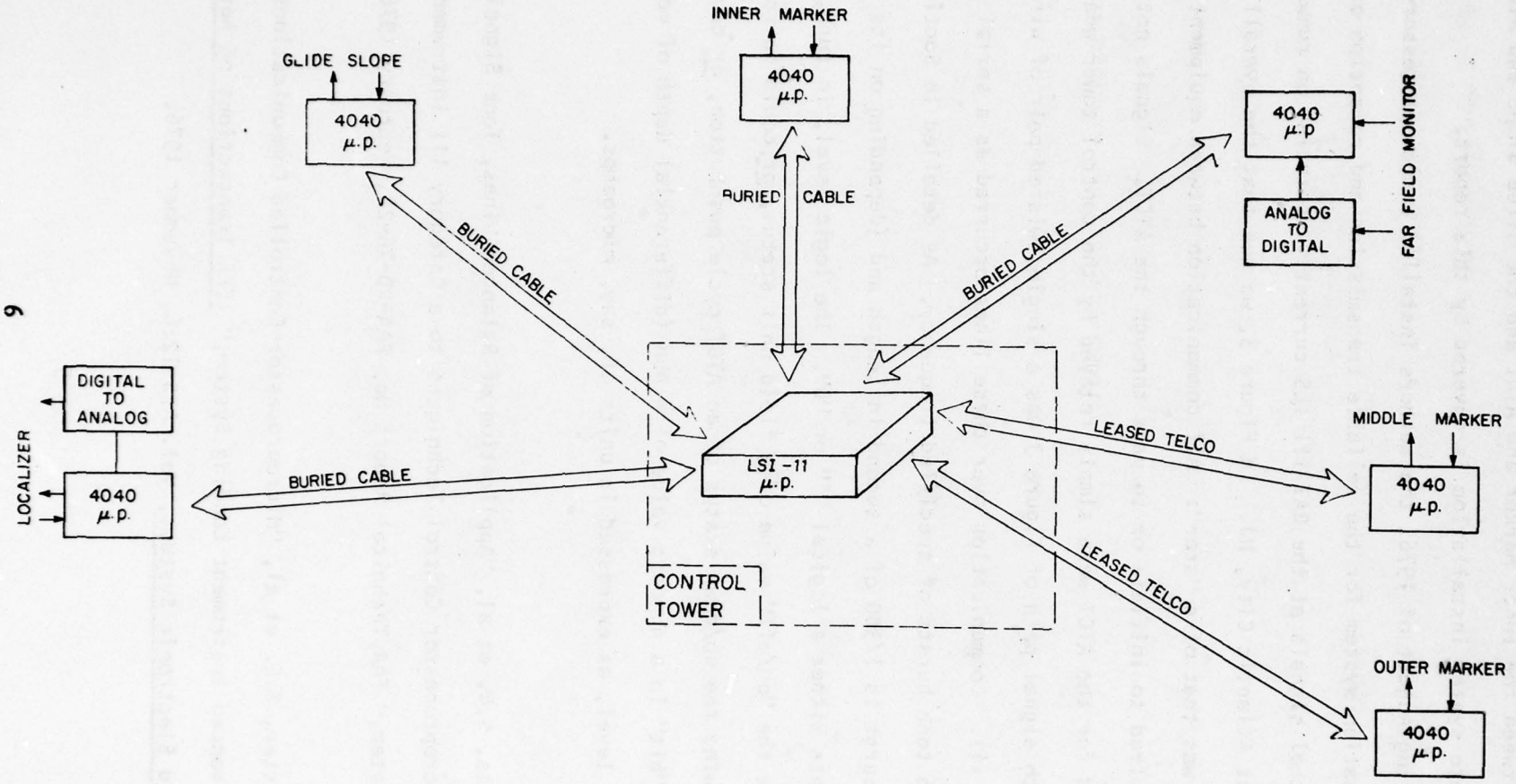


FIG. 3 MICROPROCESSORS INSTALLED IN COMMUNICATIONS SYSTEM AT NAFEC

Serial sequences of "1's" and "0's" are assembled by microprocessors: one at each equipment location. See Chapter III for a discussion of how this assembly is carried out. The actual translation of 1's and 0's to tone bursts is handled by a "modem card," see Chapter IV. Communication between a microprocessor (μ.p.) and a modem board is complicated, so we make use of an "interface board" also described in Chapter IV. Here the reader will find that the interface electronics takes care of a number of other communication problems for the μ.p.

Using microprocessors to oversee communications has provided at least one dividend: a "friendly" human interface for reporting and isolating problems that arise within the communications equipment itself. These error reporting procedures are described in Chapters VII and VIII.

The μ.p. at the ATCT "speaks" to a human operator via a TV monitor. Such a versatile output device allows the ATCT communication system to be (partially) self-documenting. Various "help" information frames are placed on the TV screen at the operator's request in order to teach operating personnel how to read the visual status displays provided, issue cycle commands, etc. See Chapter VIII for more on this.

Appendix sections D and E are <u>detailed</u> descriptions of the μ.p. programs used at the ATCT and the field units. When taken together with the thoroughly "commented" program listings provided elsewhere, these appendix sections give the user very extensive documentation of the software used throughout the system.

Though comments appearing in these pages will usually refer to μ.p. communications in ILS's, it is clear that equivalent benefits could be derived from applications of these techniques to virtually any other air terminal navigation system requiring communication of large quantities of information over distances of several miles.

CHAPTER II

LINE PROTECTION

Reliable tone signalling in the presence of electromagnetic interference (due to lightning storms, power line induction, arc lamp discharge, etc.) is made possible by the line-to-equipment interface circuitry shown in Figure 4. This circuit and the numerous noise induction mechanisms combated by it are described in detail in Chapter I's references 2 and 3.

The transformer is a Western Electric type 120 U. It provides not only large common mode voltage isolation, but also a "drainage reactor" effect.

The TII 317A three-element gas breakdown device is supplied by Telecommunications Industries, Inc. A voltage on either line wire that exceeds roughly 125 volts (to earth) will cause both line wires to be shorted to earth potential.

Backbiased diodes tied to power supplies on the equipment side of the transformer are the last defense against voltage surges induced onto the communication line.

However, the most effective protection from noise is provided by the use of balanced, twisted pairs for all long distance communication within the ILS communication system. Figs. 5-6 show photos of the line interface gear installed at NAFEC.

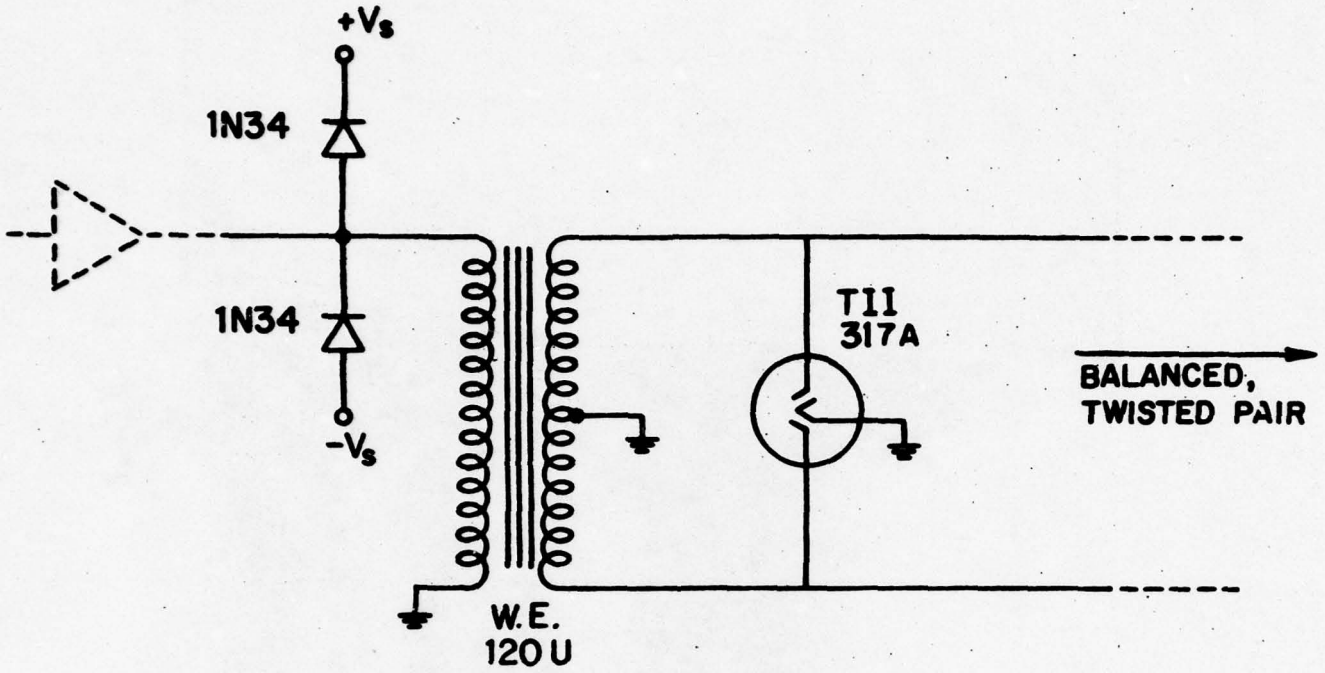


FIG. 4 LINE PROTECTION TECHNIQUE

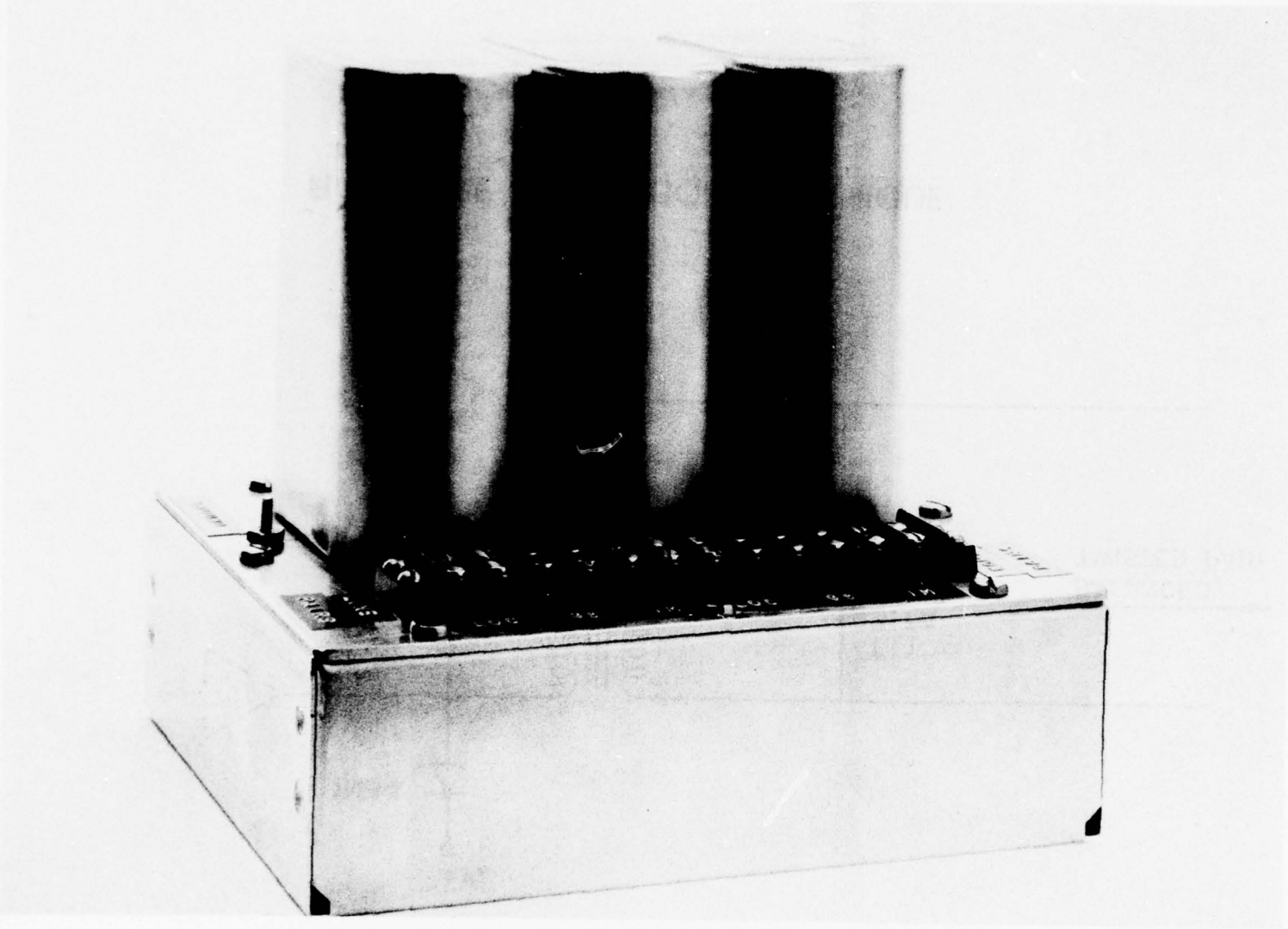


FIG.5 LINE PROTECTION ASSEMBLY INSTALLED AT THE ATCT

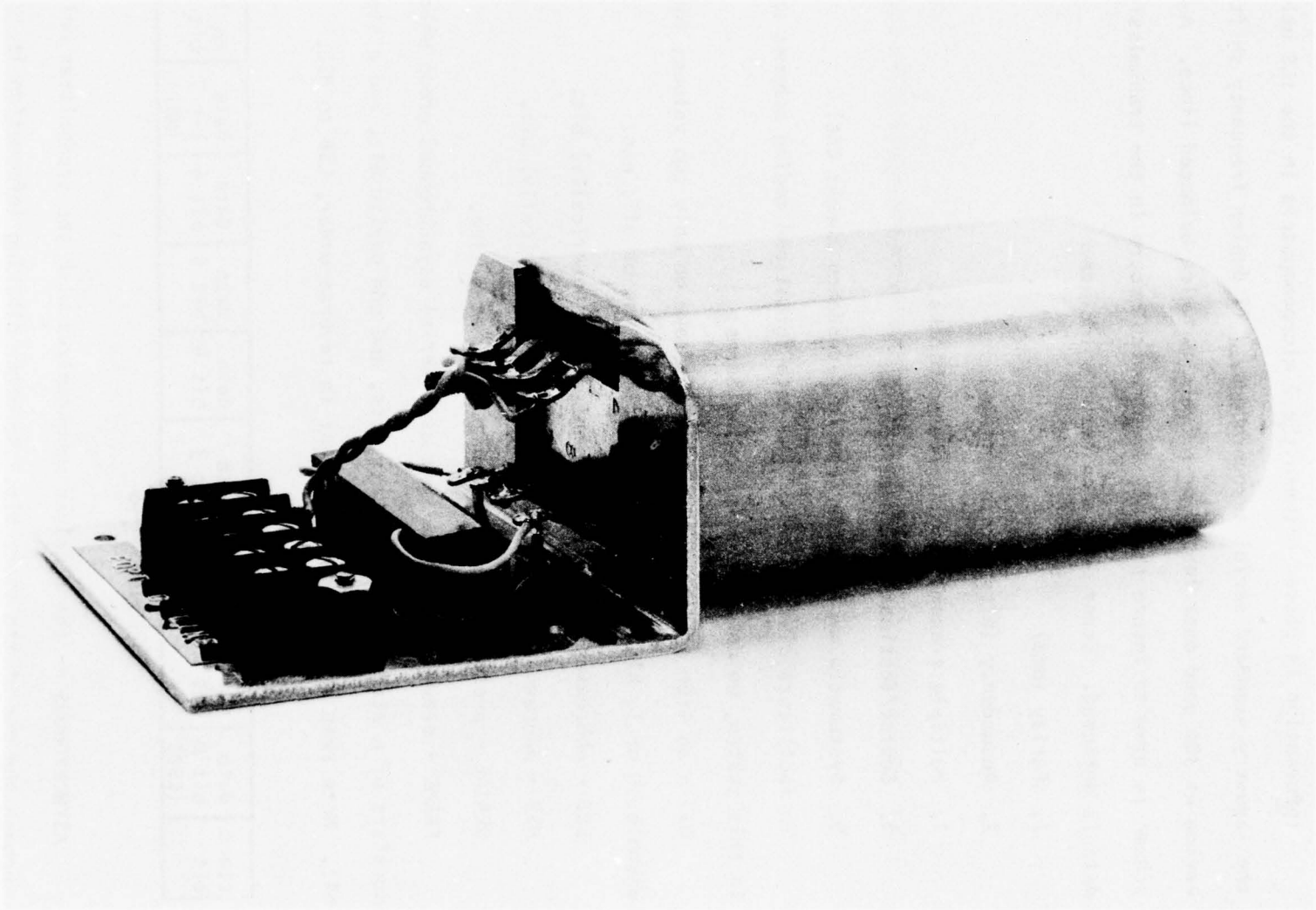


FIG. 6 LINE PROTECTION ASSEMBLY INSTALLED AT THE FIELD UNITS

CHAPTER III

SIGNALLING

Information is passed between the 7 microcomputers in the ILS using the industry standard serial asynchronous, full duplex frequency shift keying at 300 baud over leased or private two wire balanced lines. A coding scheme is used to insure that any error which occurs in the transmission of data is detected. Error detection methods include:

1. Parity (odd)
2. Redundancy (bits within a frame)
3. Multiple transmission (repeated blocks)
4. Source/destination identification
5. Systematic cyclic code (cyclic redundancy check: CRC)

To facilitate discussions of the data format and coding schemes used in this system, we define the following terms:

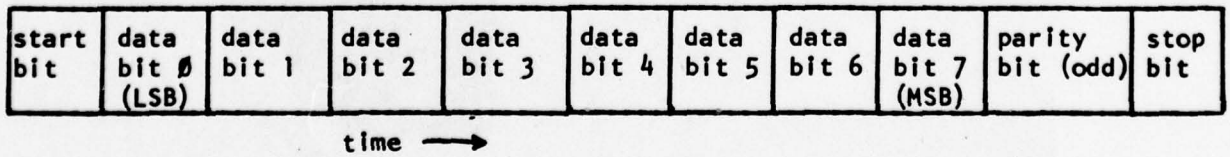
BIT - an item of information that can take on only two values; for example, 0 or 1, high or low, open or closed, on or off, etc.

LSB - abbreviation for least significant (numerically) bit.

MSB - abbreviation for most significant (numerically) bit.

SERIAL - presenting information one bit at a time.

FRAME - eleven bits to be sent as a serial asynchronous group which consists of a start bit, eight data bits, and odd parity bit, and a stop bit. Note that the data bits are sent in reverse order, LSB to MSB.



ASYNCHRONOUS -- the receiver synchronizes with the transmitter only through the use of the start bit. No other clocking information is available to synchronize the receiver and transmitter.

CRC - A cyclic redundancy check or systematic cyclic code which takes any number of data bits and creates nine check bits which can be used to define more errors than just a simple parity check.

BAUD - synonomous with "bits per second" when referring to the rate at which serial binary transmission is taking place.

FREQUENCY SHIFT KEYING (FSK) -- a common means for transmission of discrete binary data. Each possible signal element is represented by a sinusoidal oscillation at one of two frequencies.

MARK -- the higher of a pair of frequencies used for FSK transmission of binary information.

SPACE -- the lower of a pair of frequencies used for FSK transmission of binary information.

MODEM -- shortened term for modulator-demodulator, an electronic device capable of transcribing discrete input data into a form for convenient transmission over a data channel; for example, conversion to audio tones for transmission over telephone lines, a modem can also perform the inverse operation of demodulation conversion of received tone bursts to a serial string of binary data.

ORIGINATE MODEM -- a modem which transmits tones at 1070 hertz and 1270 hertz and receives information at 2025 hertz and 2225 hertz.

ANSWER MODEM -- a modem designed to work with an originate modem, it demodulates signals received at 1070 hertz and 1270 hertz and modulates at frequencies 2025 hertz and 2225 hertz.

NIBBLE -- a group of 4 bits

Each of the six field units in the ILS communicates directly with the PDP 11/03 minicomputer at the control tower. Each field unit contains one

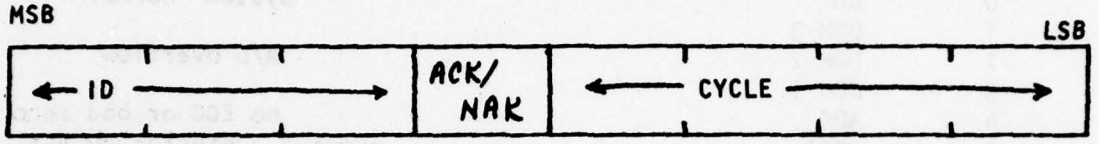
originate modem; the control tower has six answer modems. Information to be transmitted from one location of the ILS to another is translated into bits. These bits are then assembled into a single block which is then transmitted using the modems over the balanced line. There are four different block formats that will be used with 1, 4, 7, and 8 frames each. They will be described in detail later. Each frame will have a parity bit added to it as a simple error check. This checking is done by the ACIA on the modem card. Any block containing a parity error will be ignored by the particular field unit receiving that block and a request for a new transmission will be made.

Each block contains three bits which indicate the location where the message is destined or the location from which the message came. These three bits must match the expected value or the receiving microcomputer ignores it.

Because of the importance of the cycle command and the necessity that there be no errors in its transmission from the control tower to a field unit, the cycle command is assigned four redundant bits. There are two legal patterns for these bits: either a 1010(A) or a 0101(5). The receiving unit must see one of these two patterns in order for the cycle command to be accepted. In addition, the same pattern must be received twice in a row. This redundancy guarantees that no false cycle command will be accepted at the field unit.

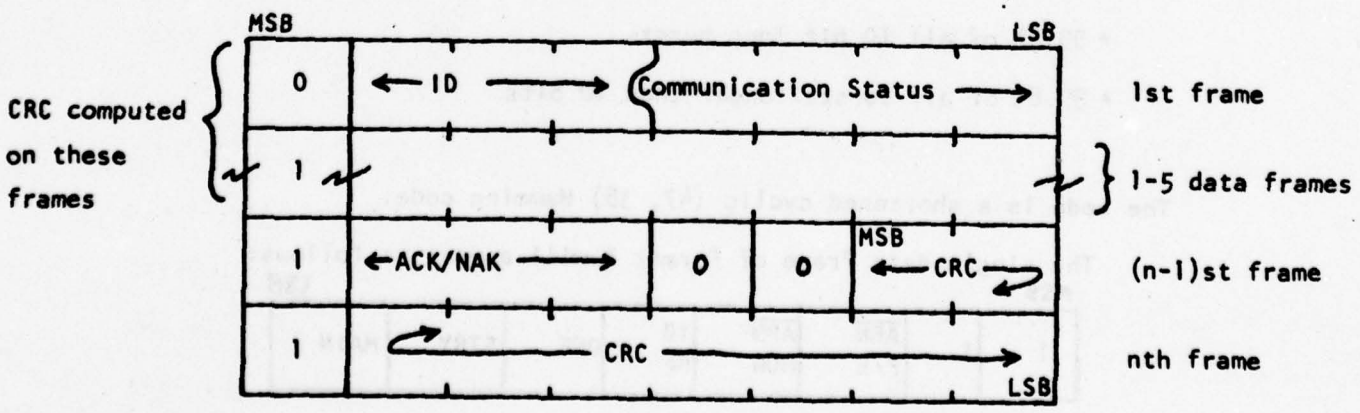
A brief description of the formats is included below.

FORMAT A (1 frame/block, used 5 places: ATCT → GS, ATCT → FFM, and ATCT → all markers.) This frame contains 3 ID bits, 1 ACK/NAK bit (1=ACK, 0=NAK), and four bits to represent the value of CYCLE (1010(A) =cycle pressed, 0101(5)=no cycle). It is organized as displayed below:



Note: although CYCLE is not required at the FFM, it must be one of the legal values V(A or 5) to be accepted at the FFM.

FORMAT B (4 frames/block, used 3 places: all markers → ATCT.) All multiple frame blocks have a common format. A header frame will consist of 0, followed by 3 ID bits and 4 bits indicating the field unit communication status. The leading 0 in a frame (MSB) indicates the start of a block. All other frames in a block will have a leading 1 in the MSB. The header frame will be followed by 1 to 6 data frames and two trailing frames. The trailing frames will contain 3 bits for ACK/NAK and 9 CRC bits. The general format is illustrated below:



The number in the Communication Status field indicates the status of the microprocessor-controlled tone signalling unit. Listed in order of priority--lowest to highest:

number	message	explanation
0	GO	system normal
1	DDM 3	
2	DDM 2	A/D overflow
3	DDM 1	
4	ADC	no EOC or bad zero
5	OVER	overrun detected by ACIA
6	FRAME	framing error " " "
7	CRC	CRC error " " program
8	PARITY	parity error " " ACIA
9	RESET	masks errors after reset
A	$\overline{\text{ID}}$	$\overline{\text{ID}}$ bits wrong
B	ID	ID bits wrong
C	NO DATA	no data from ATCT
D	TONE	no carrier from ATCT
E	MODEM	problems with modem self test
F	PROM	checksum error

In the ACK/NAK field, ACK = 010 and NAK = 101.

The systematic cyclic code (also known as a Cyclic Redundancy Check or CRC) produces a nine bit remainder which is appended to the block as shown. The CRC is computed on the first (n-2) frames of an n frame block. The generating polynomial is $x^9 + x^4 + 1$. This code will detect:

- 2 errors
- bursts 9 bits in length or less
- 99.6% of all 10 bit long bursts
- 99.8% of all bursts longer than 10 bits

The code is a shortened cyclic (47, 38) Hamming code.

The single data frame of Format B will appear as follows:

MSB							LSB
1	1	$\overline{\text{ABN}}$ P/E	$\overline{\text{ABN}}$ MON	ID RF	OFF	STBY	MAIN

Closed relay contacts will be represented by a "1" bit.

FORMAT C (7 frames/block, used 2 places: LOC → ATCT and GS → ATCT.)

The data frames will appear as follows:

MSB							LSB
1	$\overline{\text{ABN P/E}}$	$\overline{\text{ABN MON}}$	CAT 2	CAT 3	OFF	STBY	MAIN
1	SENS 2	CSE 2	NFM 1	CL 1	ID 1 (1)	SENS 1	CSE 1
1	CL 3	ID 3 (1)	SENS 3	CSE 3	NFM 2	CL 2	ID 2 (1)
1	TEMP	BAT	STBY CL	STBY ID (1)	STBY SENS	STBY CSE	NFM 3 (2)

Notes: (1.) Localizer only, GS=1

(2.) Glide slope only, LOC=1

Closed relay contacts will be represented by a "1" bit.

FORMAT D (8 frames/block, used from FFM to ATCT and from ATCT to LOC)

These data frames are organized to facilitate the easy transmission of data from the FFM to LOC through the ATCT. Data from the FFM to the ATCT is picked off at the ATCT and replaced by the CYCLE value to be sent to the LOC. Hence, no reformating of the data is necessary at the ATCT. The FFM field unit status is sent to the LOC so that A/D conversion errors may be properly treated at the D/A's.

In summary, the ATCT makes the following changes to the block received from the FFM before retransmitting it to the LOC:

1) Changes the ID bits from "6" (FFM) to "1" (LOC).

2) Inserts the CYCLE value (A,5) in the lower nibble of the second data frame, replacing values for BAT, CSE 3, CSE 2, and CSE 1.

3) Places the correct ACK/NAK value in the seventh frame and recomputes the CRC.

The data frames for Format D will be organized as follows:

1	$\overline{\text{CAT 3}}$ $\overline{\text{DIS}}$	not used	BYPASSED	POWER/ TEMP	MIS- MATCH	SHUT- DOWN	ALERT
1	DDM 1 SIGN	DDM 2 SIGN	DDM 3 SIGN	BAT ①	CSE $①^3$	CSE $①^2$	CSE $①^1$
1	(MSB) ← DDM 1 Magnitude → (LSB)						
1	← DDM 2 Magnitude →						
1	← DDM 3 Magnitude →						

Note: ① These bits replaced by CYCLE information at the ATCT to be sent to the LOC.

The DDM values are coded into an eight bit binary sign-magnitude representation. The sign bit is zero for positive values, one for negative. The seven bit binary magnitude, when multiplied by 8 millivolts, equals the voltage magnitude present at the A/D card analog input. If the voltage is divided by 51.1 kΩ, the result is the DDM reading in micro-amperes.

The CRC code is computed using modulo-2 division. Notice the similiarity between modulo-2 subtraction and the exclusive-or operation below:

Modulo-2	Exclusive-Or
0 - 0 = 0	0 ⊕ 0 = 0
0 - 1 = 1	0 ⊕ 1 = 1
1 - 0 = 1	1 ⊕ 0 = 1
1 - 1 = 0	1 ⊕ 1 = 0

When computing the CRC, the dividend is formed by concatenating the data frames into a single binary number. The divisor is 1000010001. The desired result is the nine bit remainder, which is the CRC appended to the end of the multiple frame blocks.

As an example, we will compute the normal message from the Inner Marker to the ATCT. This is Format B from above with ID bits equal to 011, ACK/NAK = 010, Communication Status = 0000, the $\overline{\text{ABN}}$'s = 1, Main = 1, and the remaining three status bits = 0. The first two frames on which the CRC is computed then look like this:

MSB							LSB
0	0	1	1	0	0	0	0
1	1	1	1	0	0	0	1

The long division required to compute the CRC remainder using module 2 subtraction looks like this:

```
                  0011000
              ------------
1000010001  ) 0011000011110001
              1000010001
              ----------
               0100011110001
                1000010001
                ----------
                0000101101001
```

The remainder is then 10110100l. The final two frames of the Format B block are now:

MSB LSB

1	0	1	0	0	0	1	0
1	1	1	0	1	0	0	1

CHAPTER IV

COMMUNICATION HARDWARE AT A FIELD UNIT

A block diagram of each of the field units is shown in Fig. 7. All of the field units communicate with the control tower in the same way, but they differ in how they process the ILS status signals. For example,the far field monitor (FFM) requires an A/D board to convert the DDM signals into digital words suitable for transmission by the modems. The localizer requires a D/A board to convert the digital words representing DDM voltages into analog voltages. The localizer and glide slope also require a board to multiplex incoming status signals from the ILS to the interface (I/F) card. Throughout this discussion, the words board and card will be used interchangably. They mean the same thing.

A. Description of the Operation of MODEM and I/F Cards

The interconnection of modem and interface cards required for transmission and reception of FSK tones is shown in Figure 8. Microprocessor control lines are loosely divided into two groups. One group initializes the modem board and one group controls the normal operations. Normal operation includes the transmission and reception of binary data, and other tasks such as communicating with the ILS equipment, and other cards in the system.

The microprocessor uses the interface card to write and read data to and from the modem card. At the field units, the microprocessor uses the interface card to write to the seven-segment display on the front panel; and read the data from the relays in the ILS equipment. In addition, the interface card provides watch-dog reset lines to the microprocessor. Although one interface card will service any of the locations, a modem card is required at each end of any balanced pair in the system.

At the field units one modem card and one I/F card are used. At the control tower, one I/F card and six modem cards are used. Both the interface card and the modem card may be used interchangeably anywhere in the system.

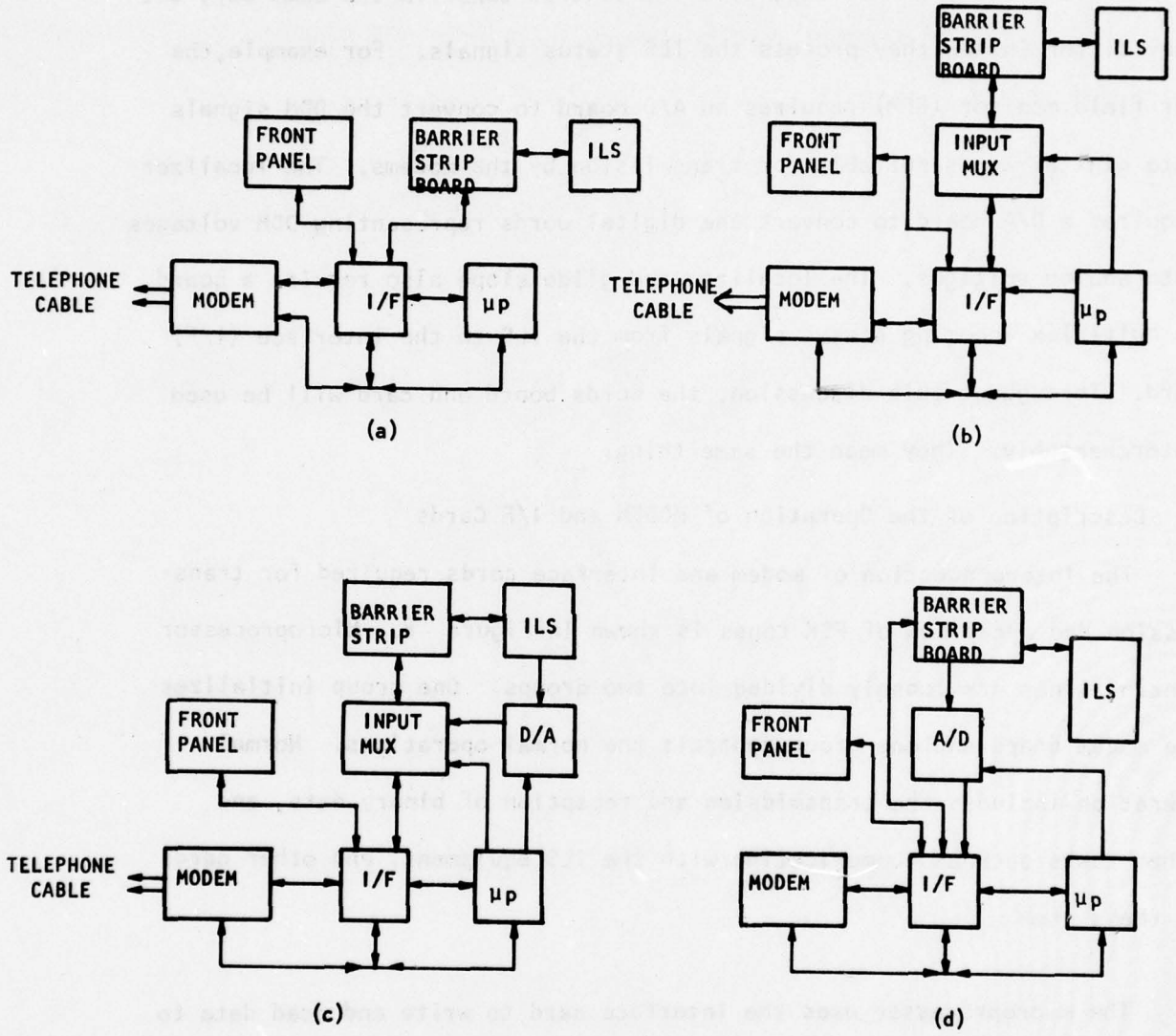


Figure 7 Field Unit Communication Hardware Block Diagrams: (a) Inner, outer, middle markers; (b) Glide Slope; (c) Localizer; (d) Far Field Monitor

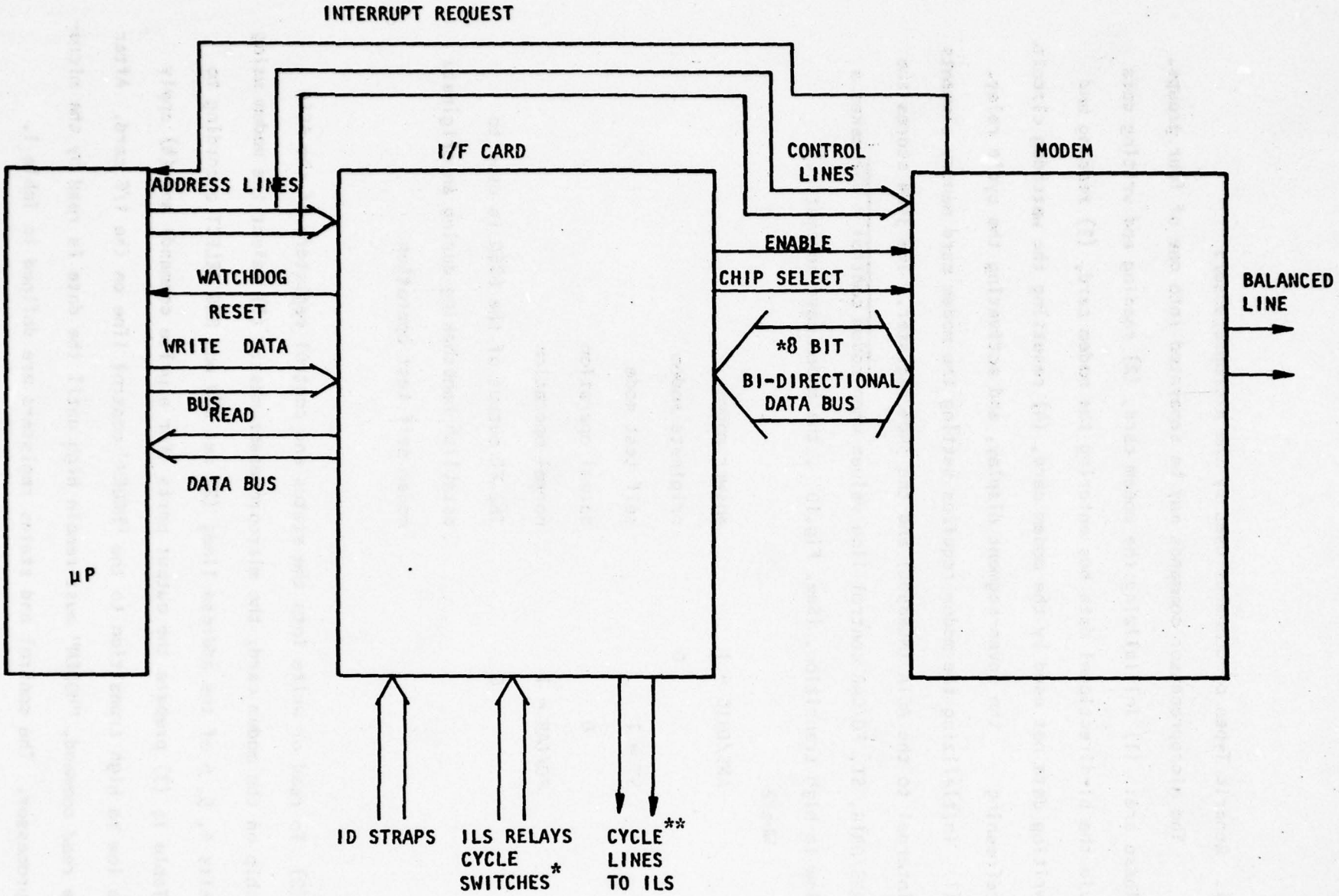


FIG. 8 INTERCONNECTION OF I/F CARD AND MODEM CARD

*FOR CONTROL TOWER I/F

**FOR FIELD UNITS USING THE CYCLE COMMAND

1. Generic Types of Commands used by the Microprocessor:

The microprocessor commands may be separated into one of four groups. These are: (1) initializing the modem card, (2) reading and writing data via the bi-directional data bus entering the modem card, (3) reading and writing data not used by the modem card, (4) resetting the watchdog circuit, refreshing the seven-segment display, and activating the cycle relay.

(1) Initializing the modem requires setting the modem card memory elements internal to the ACIA (MC6850L) and the 3404 register. The 3404 stores the ANS/ORIG, ST, FO/CAR control line values when $\overline{\text{modem control} \cdot \text{CS2}}$ makes a low to high transition. (See. Fig.10 , the modem card schematic.)

Where

ANS/ORIG =	1	answer modem
	0	originate modem
ST =	1	self test mode
	0	normal operation
FO/CAR =	1	normal operation
	0	The FO output of the 6860 is used to establish handshaking during an originate modem self test operation

(2) To read or write into the status and control registers of the ACIA chip on the modem card, the microprocessor must: (1) select the modem using bits 4, 5, 6 of the address line; (2) set R/W and REG SELECT according to Table 1; (3) prepare the output ports (for a write command) and (4) apply a low to high transition to the "MODEM" control line on the I/F card. After a read command, "MODEM" must remain high until the data is read by the microprocessor. The control and status registers are defined in Table 1.

Table 1

Selection of ACIA Registers

R/W	REG SEL	
0	0	Write to transmit data reg
0	1	Write to control reg
1	0	Read re data reg
1	1	Read status reg

The steps used for reading and writing to the receive and transmit registers are the same as those outlined for the control and status registers. The difference is the value of the REG SELECT line.

(3) The interface card handles data from the ILS relays, the front panel in field units, the cycle pushbuttons at the control tower, an internally generated 4 Hz clock, and ID straps for field unit use. These bits are multiplexed along with the data bits from the modem card. The microprocessor selects the source of information according to the following truth table.

MODEM	SELECT (bit 4 of address line)	SOURCE
0	0	ILS relays or cycle switches
0	1	4Hz clock/ID straps/thumbwheel switch on front panel
1	X	Data bits from the modem card

X = don't care (0,1)

(4) The microprocessor uses address bits 4, 5, and 6 to activate the cycle relay, and reset the watchdog, and write to the seven-segment display on the front panel at the field units. A "011" on bits 4, 5, 6 will reset the watchdog circuit and write the output data bits tothe seven-segment display on the front panel. A "111" on bits 4, 5, 6, will activate the cycle relay to simulate a pushbutton switch closing.

Some additional information is provided in Chapter VI. See Table 2.

TABLE 2

Interface Card Addresses

8-BIT ADDRESS LINE Bit 7 (MODEM)	Device Select 6 A_2 5 A_1	Select4	3 OUTPUT LATCH	2 MODEM CONTROL	1 REG SELECT	0 R/W	OPERATION
0	0		0	0	0	0	NOP (No operation)
	0 - 5	(1)	1	0	0	1	Read ACIA control
	0 - 5	(1)	1	0	0	0	Write ACIA control
	0 - 5	(1)	1	0	1	1	Read rec. buffer
	0 - 5	(1)	1	0	1	0	Write xmit. buffer
Z	0 - 5	(1)	1	1	X	X	Write modem control
	6		X	X	X	X	Watchdog reset
X	6		X	X	X	X	load 7-segment
X	7		X	X	X	X	Latch cycle FF
Z	1 - 3		0	0	0-1	0-1	Relay latches
ø	even		0	0	X	X	Read ILS switches
ø	odd		0	0	X	X	Read clock, straps, and thumbwheel switch
X	2		X	X	X	X	Start conversion
X	3		X	X	X	X	Load MUX select

X -- don't care (0,1) Z = constant or ø

2. Transmission and Reception of Data

Digital data is sent and received via the 8-bit, bi-directional tri-state data bus at the input of the ACIA, a Motorola MC6850. The bi-directional data bus is also used by the ACIA for initialization of the chip and for examination of its status registers. The tri-state feature allows several modem cards to share the same data bus. (See Fig. 10).

The ACIA converts the 8 data bits from the data bus into a serial stream of data complete with a start bit, parity bit, and stop bit. The serial 1's and 0's are converted to mark and space FSK binary signals by a modem chip, a Motorola MC6860. Similarly, binary mark and space data received by the modem chip is demodulated to digital 1's and 0's and converted back to parallel form by the ACIA. The data rate is 300 baud.

Transmission and reception of binary data is simultaneous. This is accomplished by using two different sets of mark/space frequencies and allowing only modems with compatible frequency assignments to communicate. The frequency assignments are as follows:

MODEM ASSIGNMENT	TRANSMIT FREQUENCY		RECEIVE FREQUENCY	
	MARK (HZ)	SPACE (HZ)	MARK (HZ)	SPACE (HZ)
ORIGINATE	1270	1070	2225	2025
ANSWER	2225	2025	1270	1070

In order to separate these frequencies and to service the modem, transmit and receive filters, buffers and limiters, and a duplexer are needed to complete the modem card. Also included is a "self test mode" switching capability. This is shown in Figs. 9 and 10.

Before two modems may communicate over a channel, they must establish that their carriers are being received. This is referred to as handshaking. The initial handshaking consists of the answer modem sending a 2225 Hz carrier to

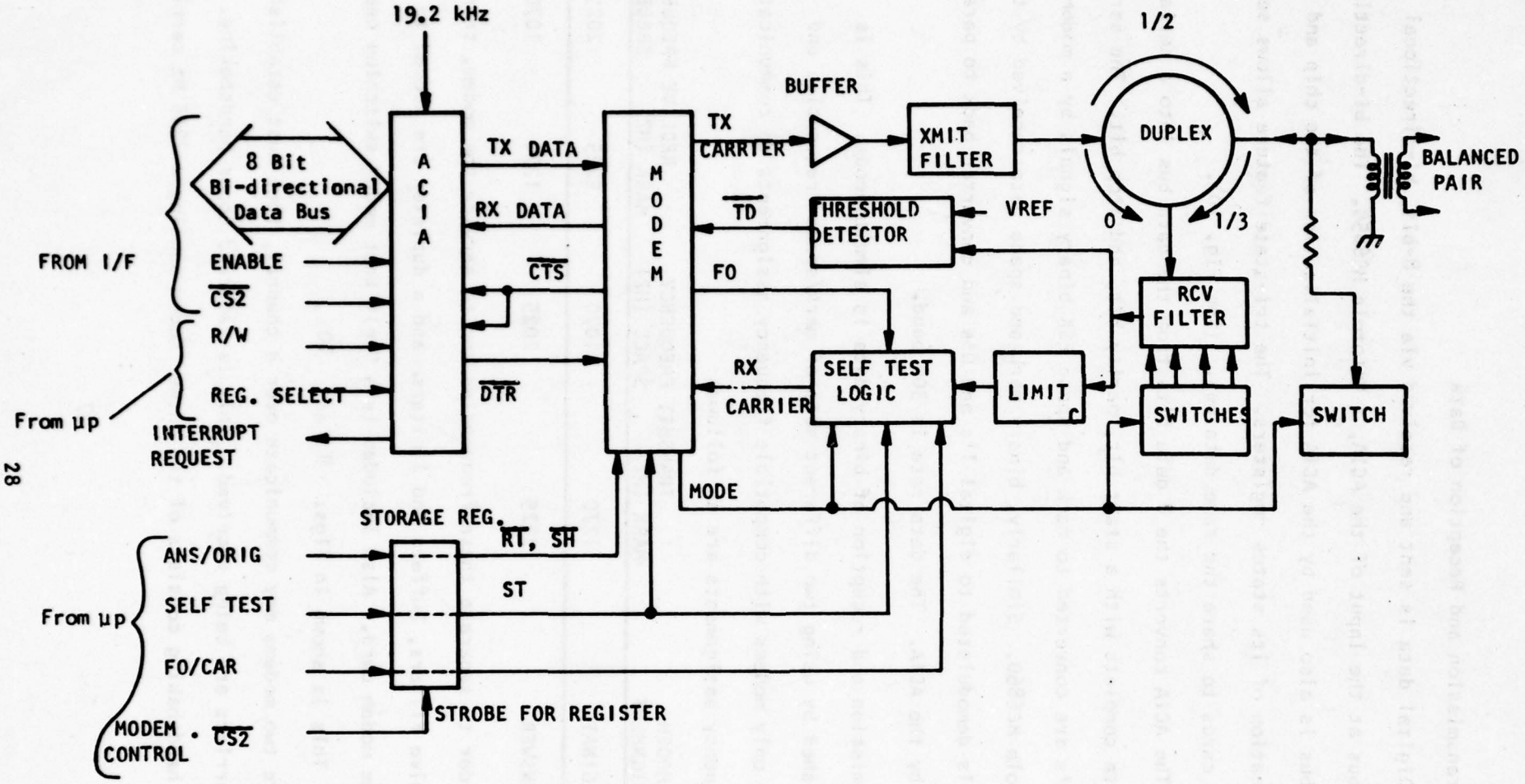


FIG. 9 MODEM CARD BLOCK DIAGRAM

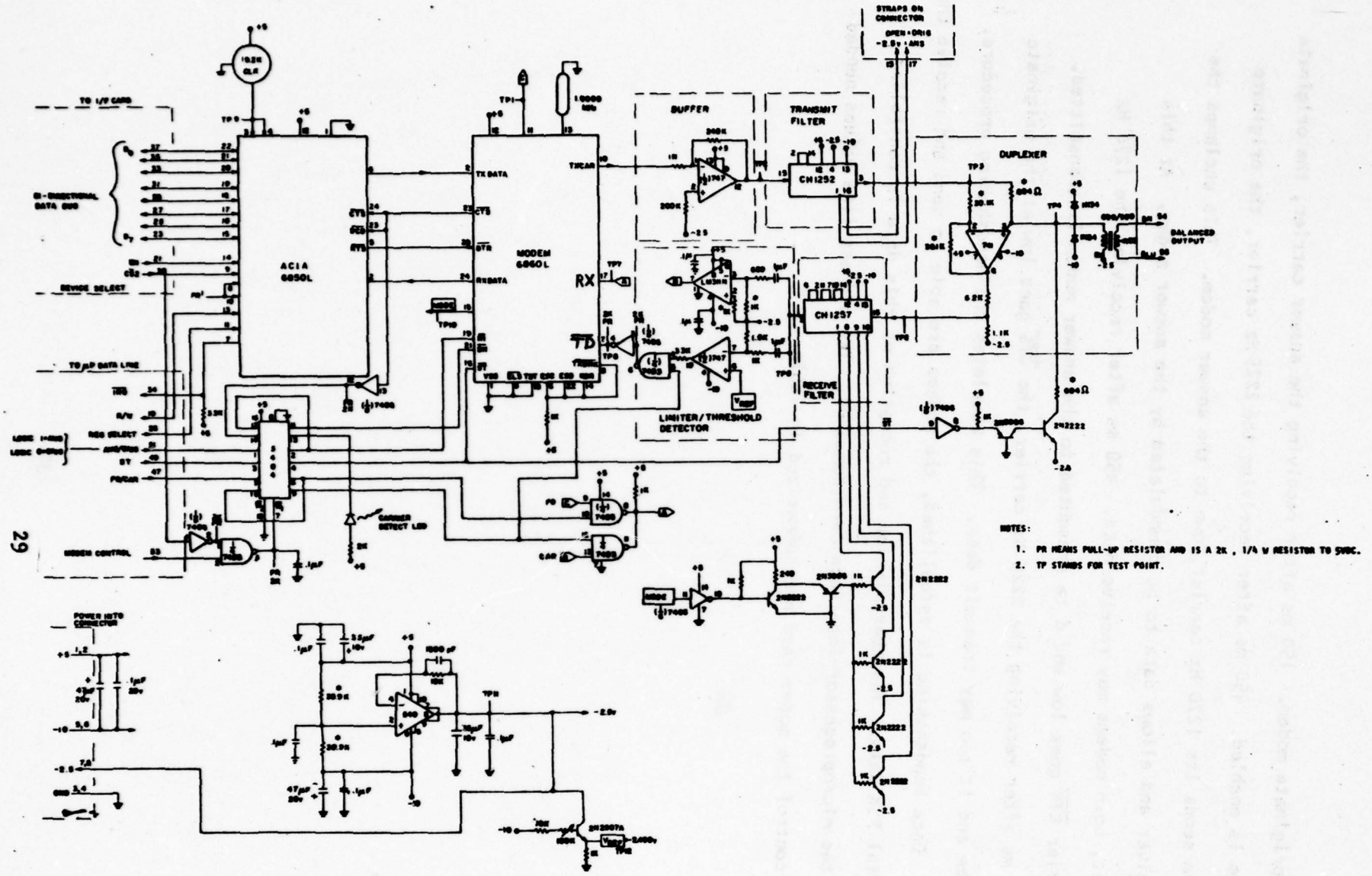


FIG. 10

MODEM CARD SCHEMATIC

the originate modem. 150 ms after receiving the answer carrier, the originate modem is enabled. 450 ms after receiving the 2225 Hz carrier, the originate modem sends its 1270 Hz carrier down to the answer modem. This unclamps the receiver and allows data to be demodulated by the answer modem. At this point, both modems may receive data. 450 ms after receiving the 1270 Hz carrier $\overline{\text{CTS}}$ goes low and d ta presented to the answer modem is transmitted. 750 ms after receiving the 2225 Hz carrier, the $\overline{\text{CTS}}$ goes low at the originate modem and it too may transmit data. This completes the handshaking procedure.

Once handshaking is established, the modems are able to send and receive the serial FSK data. The transmission and reception of this data is controlled by the microprocessor-interface card commands. The address line values needed to control the modem card are summarized in Table 2.

3. Modem Card Description:

The modem card is designed around the MC6860L modem chip (see Figs. 9 and 10). The MC6850L ACIA provides the modem chip with a parallel-serial interface to handle the flow of digital data. The buffer, transmit filter, duplexer, receive filter, and limiter/threshold detector process the FSK analog signals used by the modem. The -2.5v supply is used to create an analog signal ground reference. The INTEL 3404 latch stores control signals that establish the mode of operation. The remaining logic and discrete components are used during the self-test mode of the modem card.

The $\overline{RT}$, $\overline{SH}$ inputs on the modem chip establish the modem as an answer or an originate modem. The $\overline{ST}$ input, when low, puts the modem in a self-test mode. The self-test mode has the modem chip demodulating the same frequencies that it sends. The "MODE" output is a test point used as a signal for switching between answer and originate mode during a self-test and for debugging the modem card. The table below shows the relation between these signals.

$\overline{ST}$	$\overline{SH}$	$\overline{RT}$	MODE	
H	L	H	H	(orig)
H	H	L	L	(ans)
L	L	H	L	(orig)
L	H	L	H	(ans)

H = LOGIC HIGH
L = LOGIC LOW

The modem chip receives digital data from the ACIA at the TXDATA input and sends the ACIA digital data over the RXDATA line. $\overline{CTS}$ and $\overline{DTR}$ are used by the modem chip and the ACIA to signal they are ready to transmit and receive data.

The TXCAR output of the modem chip is the FSK tone to be sent down the buried cable. The tone is generated by an A/D converter from a sine-lookup table (both internal to the modem chip). The buffer is used to prevent

loading of the TXCAR output and to reduce the signal amplitude before it enters the transmit filter. The transmit filter is a bandpass filter whose center frequency is determined by straps on the edge connector. When the straps are left open, the originate frequencies are passed. When the straps are tied to -2.5VDC (analog ground), then the answer frequencies are passed. The filter passband has a typical gain of 10. The duplexer, under ideal loading, will reduce the transmit filter output by 1/2 which results in an output of -6dBm (600 ohms).

Signals received by the modem card are passed to the receive filter from the duplexer input with an effective gain of .3 (nominally). The receive filter is a bandpass filter whose center frequency (CF) is determined by the mode output and the ans/orig setting. During a self-test mode the CF shifts to that of the transmitter on the card. Normally the CF is that of the modem transmitter at the other end of the line. The sine-wave output of the receive filter is limited by the LM311 comparator. The square wave output of the LM311 is received by the modem chip's RX CAR input.

The modem chip measures the time between transitions on this square wave input and decides whether its frequency is that of a mark or a space. This decision is constantly updated with each new transition. Because of the type of demodulation employed, restrictions are placed on the duty cycle of the limited, filtered tone and on the slew rate of the comparator.

Two types of error are present in the demodulator used on the modem chip. These are phase jitter and bias distortion. Phase jitter occurs when a mark to space (or vice versa) transition occurs. According to when the transition takes place, the demodulator may erroneously decide mark after the space is

present. Bias distortion results when a demodulator decision is biased in favor of mark over space (or space over mark). This distortion is the result of the mark/space decision point existing somewhere between the two periods.*

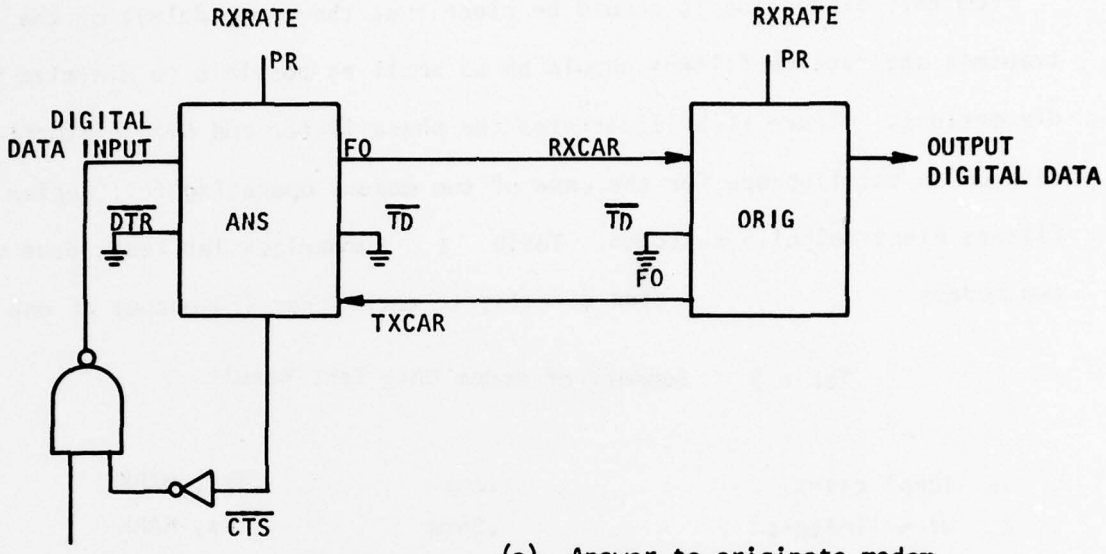
From this discussion it should be clear that the group delays of the transmit and receive filters should be as small as possible to minimize these distortions. Figure 11(b) illustrates the phase jitter and bias distortion seen on an oscilloscope for the case of two modems operating full duplex with filters electronically switched. Table 3 summarizes lab tests done using two modems tied directly to each other (FO output of one

Table 3 Summary of Modem Chip Test Results

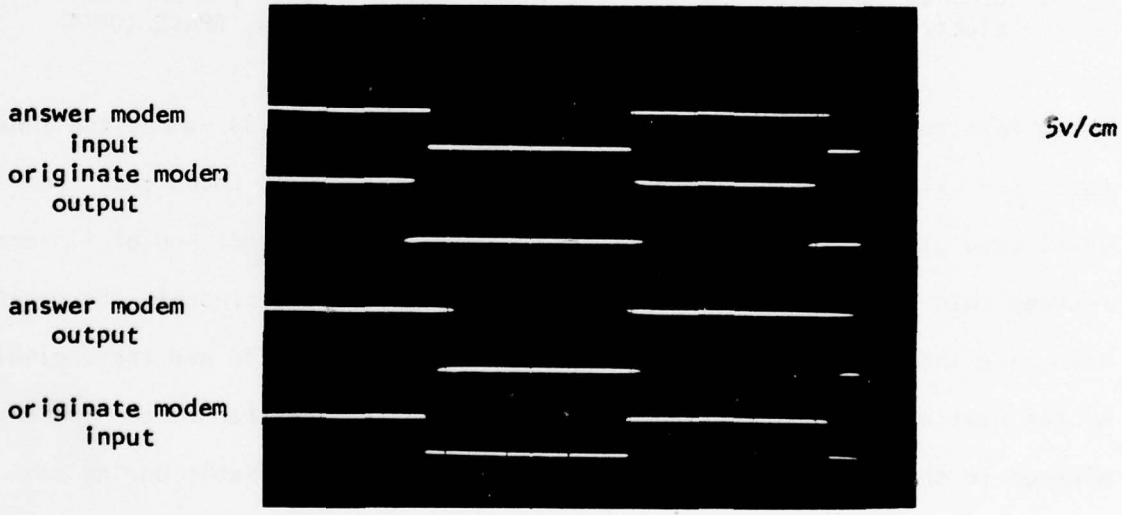
1.	Ideal case:	.2ms	.1ms, MARK
2.	With limiter:	.24ms	.1ms, MARK
3.	With filters, limiter, duplexer straps on filters	.4ms (ANS) .35ms (ORIG)	.6ms, MARK (ANS) .8ms, SPACE (ORIG)
4.	Same as (3) with electronic switching	.45ms (ANS) .35ms (ORIG)	.45ms, MARK (ANS) .8ms, SPACE (ORIG

modem into the RXCAR input of the other as shown in Fig. 11 (a)) as the ideal case, and using two modems with filters in between. It is found that the ideal case gives the ACIA a 1.45ms safety margin and the addition of filters reduces this safety margin to .9ms. This safety margin represents the worst case time interval between the bit center chosen by the ACIA and the beginning of the next bit. The jitter and distortion added by the channel are not considered in this safety margin since there was no data available during the tests. However, to protect against such errors, the ACIA has several different error detection mechanisms which will be discussed shortly.

*Actually it is nominally set to be half way between the two periods.



(a) Answer to originate modem test circuit (ideal case)



(b) Input-output waveforms for full duplex operation with electronically switched receive and transmit filters

FIG. 11 FULL DUPLEX MODEM TO MODEM OPERATION

The logic between the LM311 output and the RX CAR input is used for the self test of an originate modem. If handshaking has not been established, then the TXCAR output of an originate modem is not present. For this reason, the modem is first made to handshake with itself as an answer modem being self-tested, and then switched to an originate modem which is also self-testing. The carrier will now remain on and the self-test may continue. Note that no provision is made to switch an originate transmit filter to an answer filter during a self-test. Instead the FO output (during answer mode) of the 6860 is steered directly back into its own RX CAR input to establish handshaking. Once handshaking is established, then the originate carrier is sent through the transmit filter and self-testing is continued in a normal fashion.

The $\overline{TD}$ input to the modem chip is used to signal the presence or absence of the carrier. It is only required to go to a low level once every 30 ms. The duration of this low level may be as short as 1 µs. If $\overline{TD}$ should fail to go low after 30ms, then the $\overline{CTS}$ output of the modem goes high signaling the ACIA that loss of carrier has occurred. The reference level used to decide when the carrier is present is VREF. It is set to 50 mV with respect to the -2.5v analog ground, and it represents the signal level of a -30 dBm (600 ohm) signal entering the duplexer. The VREF level may be varied by adjusting the 100K pot in the base circuit of the 2N2907A. The circuit generating VREF is designed to track variations in the -2.5v supply.

The ACIA handles the interfacing of the modem chip with the 8 bit data bus. The 300 baud data rate used by the ACIA is established by the 19.2KHZ square wave oscillator. The 8 bits of data presented to the ACIA for transmission is formatted as was described in Chapter II. The ACIA adds the start bit (which is

always a mark) a parity bit, and a stop bit (which is always a space). This serial string is presented to the modem which converts the marks and spaces to FSK tones. Similarly, the ACIA receives serial data from the modem chip. When a mark to space transition is detected by the ACIA, it automatically waits 1/2 a bit duration from the transition time and starts sampling in 1 bit duration intervals. This will ideally result in sampling the center of each bit. Both the transmitter and the receiver are double buffered so that new data will be processed while old data is being sent or received.

A false start bit, that is a glitch that is too short to be a true start bit, is deleted by the ACIA. The ACIA will also sense framing errors caused by improper placement of the stop bit or start bit, and parity errors in the data sent. If frames are sent so quickly that a frame is lost before the microprocessor can read it, then the ACIA signals a receiver overrun error. Also the ACIA will indicate the status of $\overline{CTS}$ and $\overline{DTR}$, and the status of the transmitter data register and the receiver data register. This information is available to the microprocessor in the ACIA's status register. An interrupt request is used by the ACIA to indicate to the microprocessor the reception of a new frame or the loss of carrier. The microprocessor may program the ACIA to give an interrupt signal only under certain situations.

Troubleshooting the Modem Card:

There are twelve test points availabe on the modem card to aid in trouble-shooting the card. These points are listed in Table 4 and shown on the modem card schematic. Once it has been determined that a card is bad by replacing it with a spare, a procedure for testing the card is needed. Many problems may be diagnosed using the following procedure:

(1) Put the card on an extender board in the RC box. This will require switching one line back to DC signalling until the testing is over.

(2) Check the supplies for +5, -10, -2.5 VDC and check the 19.2KHz clock output. A bad +5 or -10 volts supply reading may indicate a shorted capacitor. A bad -2.5VDC reading indicates either the 540 operational amplifier is bad or a capacitor in that circuit is at fault. If the 19.2KHz oscillator doesn't work, remove the 6850 ACIA chip to see if it is loading down the oscillator. Replace the appropriate chip.

(3) Halt the PDP 11/03 in the self-test mode and look for a hand-shaking indication. If there is handshaking, then the problem may be difficult to locate. Trace through the analog section of the board to see if anything is obviously wrong. Also check voltages on the transistor switches to see if any of them are faulty. As a final suggestion, replace the ACIA chip.

If there is no handshaking then the problem should be more easily isolated. First check the LED to assure it is working properly. Then check $\overline{RI}$, $\overline{SH}$, $\overline{ST}$ and MODE on the 6860 modem chip for "0", "1", "0", and "1" respectively. If $\overline{RI}$, $\overline{SH}$, $\overline{ST}$ are wrong, then either the 3404 latch or a chip driving its $\overline{w_1}$ input is bad. If MODE is wrong, then the 6860 chip is bad. Also check $\overline{DTR}$ for a low.

If these signals are fine, then the problem lies in the modem chip or its analog circuitry. Look at FO (TP1) to see if the carrier square wave is present. If not, replace the 6860. If FO is present, then check its frequency for 2225Hz. Follow the TXCAR modem output around through the duplexer and back to the RX CAR, $\overline{TD}$ inputs. If the problem appears to be in the receive filter or duplexer, also check their transistor switches.

TABLE 4 MODEM TEST POINTS:

TP 1	FO SQUARE WAVE CARRIER
2	TRANSMIT CARRIER, SINUSOID
3	FILTERED TRANSMIT CARRIER
4	SECONDARY OF LINE TRANSFORMER
5	DUPLEXER OUTPUT TO THE RECEIVE FILTER
6	FILTERED RECEIVE CARRIER
7	LIMITED RECEIVE CARRIER
8	THRESHOLD DETECT SIGNAL
9	NOT USED
10	"MODE" OF 6860 CHIP
11	-2.5VDC
12	VREF FOR $\overline{TD}$ CIRCUIT

4. INTERFACE CARD DESCRIPTION:

The interface card is a collection of many small circuits each controlled by the microprocessor. The truth table and purpose of many of the circuits are described in the section on microprocessor commands or in the element data sheets. Fig. 12 is a block diagram of the I/F card and Fig. 13 is its schematic.

The I/F circuitry used in communicating with the modem card is the 3205 1-out-of-8 binary decoder, the 3216 bi-directional buffers, and their output circuitry, and the 74123 retriggerable one-shots.

The 3205 decoder will present a "0" on the output selected by inputs A0, A1, A2, and a "1" on all other outputs. Six of these outputs are for the $\overline{CS2}$ input of modem cards. The remaining outputs are used for activating the cycle relay and resetting the watchdog circuit.

The 3216 bi-directional buffers interface the input and outputs ports of the microprocessor with the bi-directional bus at the ACIA input. The R/W command determines whether information is read in or written out. The buffer outputs are automatically stored in the 74174 and 3404 latches when an ACIA output is read. These latches may be read by the microprocessor at a later time. The 74153's are used to select the inputs desired by the microprocessor. The following truth table defines their operation.

TRUTH TABLE

Select (Address bit 4)	MODEM (Address bit 7)	Data Selected
Don't care (0,1)	1	Modem card data bits
0	0	ILS relays/switches
1	0	4Hz clock/ID strap/ thumbwheel switch

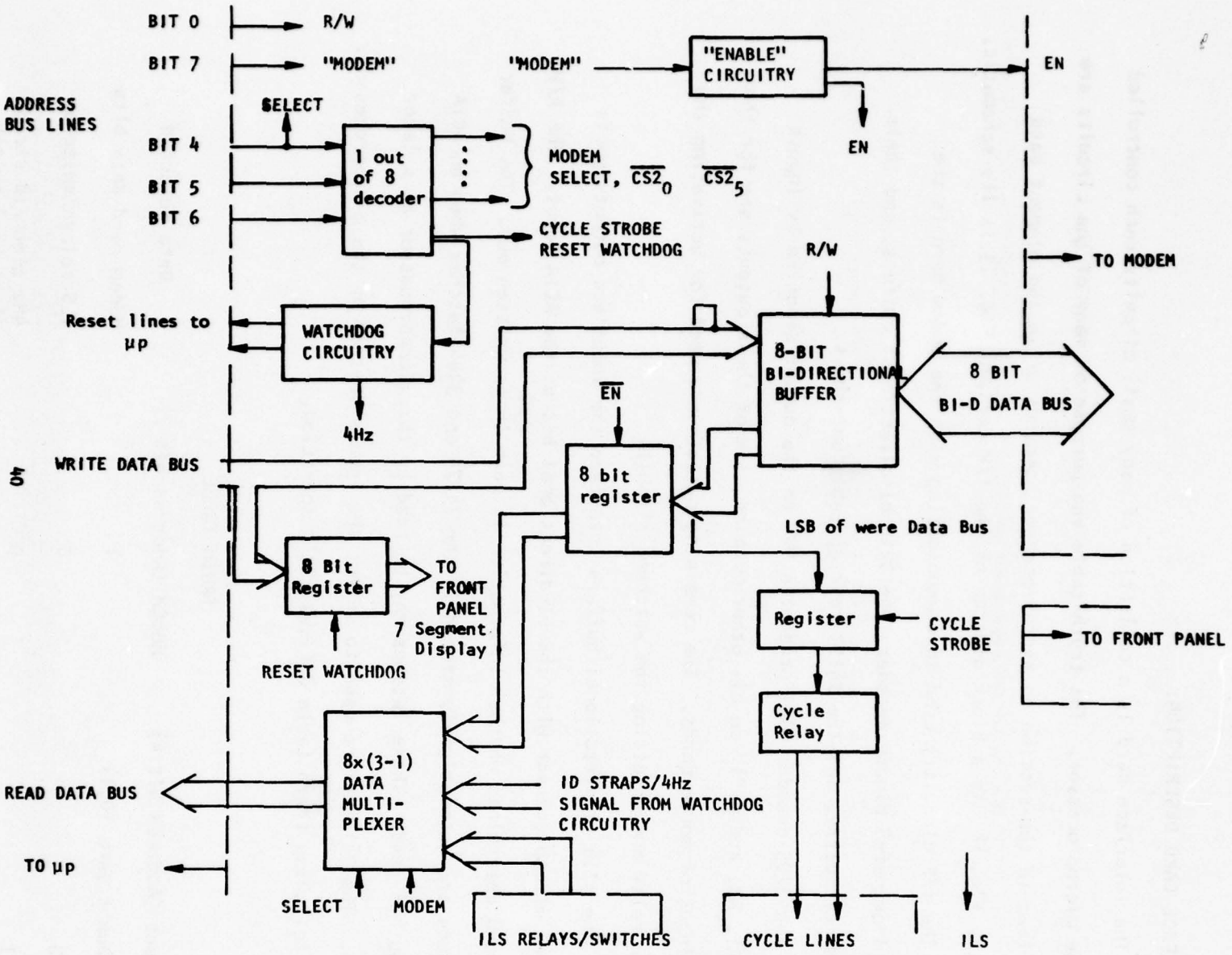


FIG. 12 I/F CARD BLOCK DIAGRAM

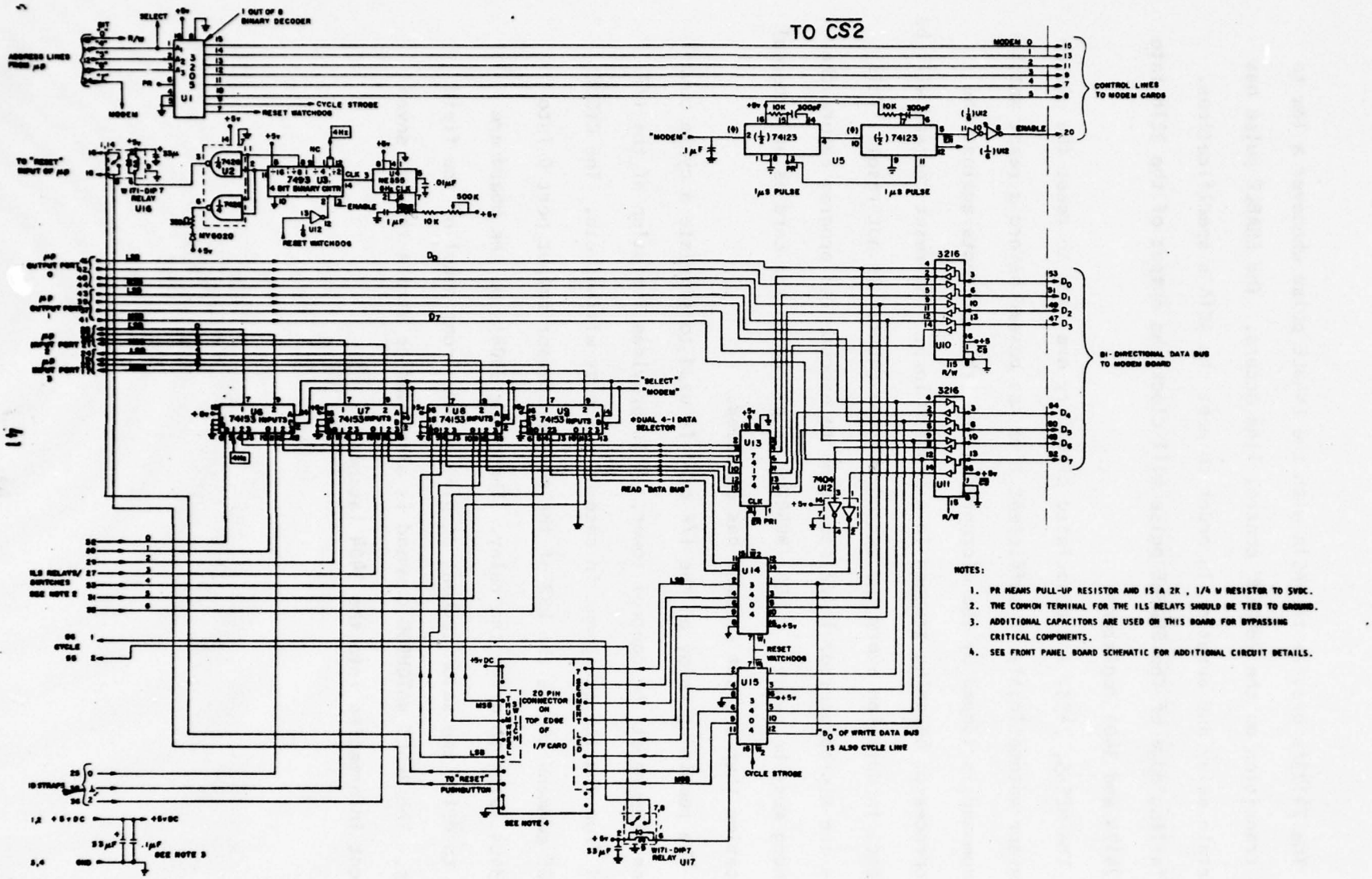


FIG. 13

INTERFACE CARD SCHEMATIC

The 74123's provide the ACIA with the ENABLE pulse whenever a low to high transition on the "MODEM" control line occurs. The ENABLE pulse has a certain delay and duration in order to meet the ACIA's specifications. The falling edge of the ENABLE pulse will clock the output of the 3216 into the 74174 and 3404 registers.

The NE555, 7493, and associated circuitry are used to reset the microprocessor automatically if sufficient time has passed before a reset watchdog command is issued by the microprocessor. This protects against the microprocessor becoming trapped in a program loop. The reset command will be 1/2 sec. in duration every two seconds if the counter is not reset. This gives the microprocessor 1 1/2 sec. time to execute its program before the watchdog must be reset. A LED (MV5020) is placed on the card as an external indication that a reset command has occurred.

The remaining relay on the I/F card is used to simulate a cycle push-button closure at the control tower. This provides isolation of the I/F board from the ILS equipment in case of a faulty wiring, etc. The CYCLE STROBE command clocks the LSB of the microprocessor output port 0 into the 3404 latch to drive the relay. The other 3404's on the board are used to drive the seven-segment display on the front panel of the field units. The RESET WATCHDOG command is also used to strobe the new seven segment information into the 3404 latches.

B. Front Panel Board Description:

As shown in Fig. 14, the front panel board contains the seven-segment LED display and its series resistors. It also provides the control line link between the I/F card and the thumbwheel switch and reset pushbutton switch on the front panel. Pull up resistors for the thumbwheel switch are also on the front panel board. A twenty-line ribbon cable is used to connect to the I/F card.

The seven-segment LED display provides information about the status of the communication system that is useful in troubleshooting. When possible an English message is spelled out using the alphabet shown in Fig. 29 . The thumbwheel switch may be used on conjunction with the LED display to test boards in the communication system. This is useful in determining which card to replace.

The reset pushbutton on the front panel is wired in parallel (via the 20-line ribbon cable) with the reset relay switch contacts and will force a software reset just as the watchdog circuitry would do if it weren't disabled periodically.

C. Barrier Strip Boards:

Fig. 15 - 18 show the location of the ILS status signals on the barrier strip boards for the field units. A 50-line ribbon cable connects the card cage to the barrier strip board. The most important ILS signals have LED's beside them to indicate their status.

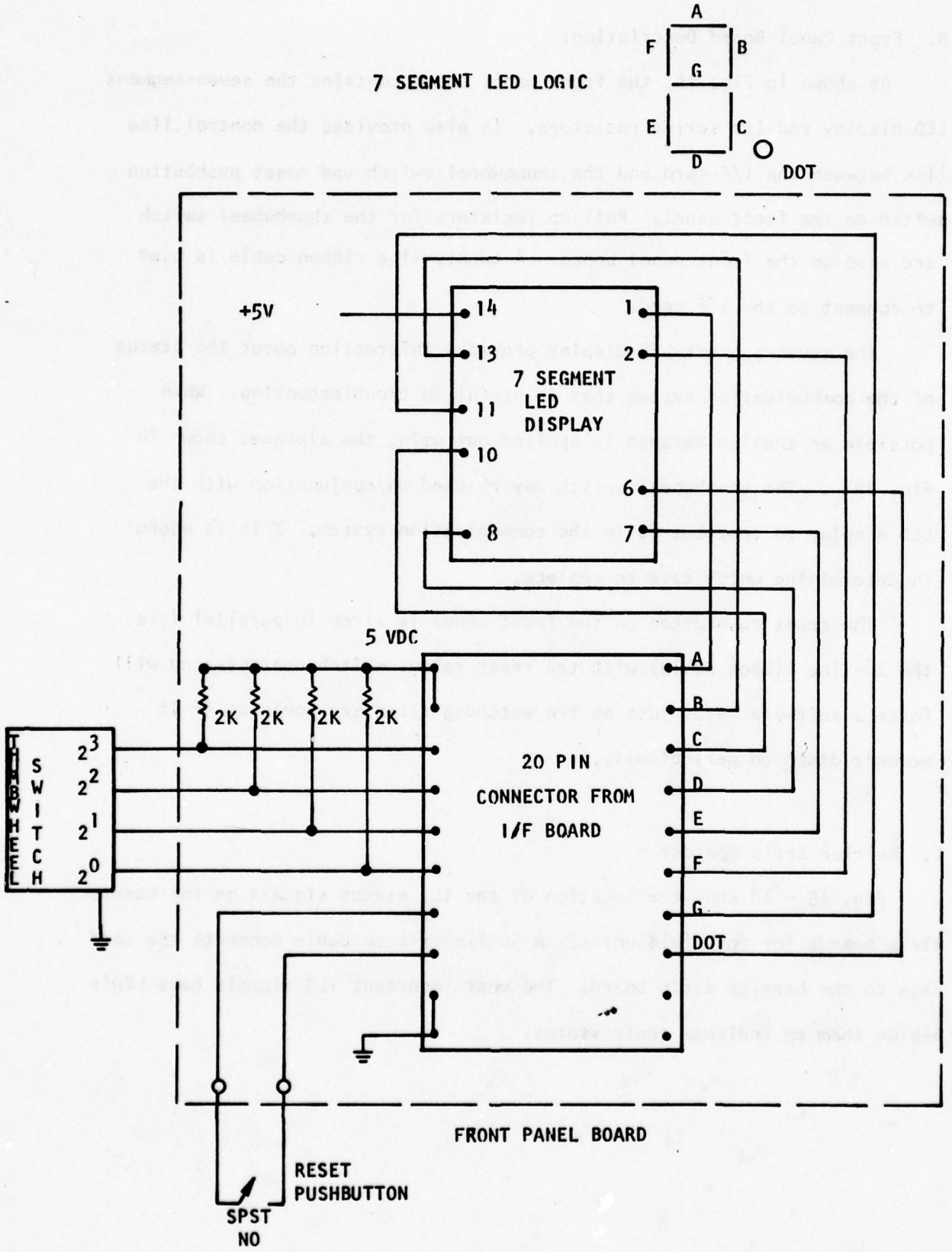
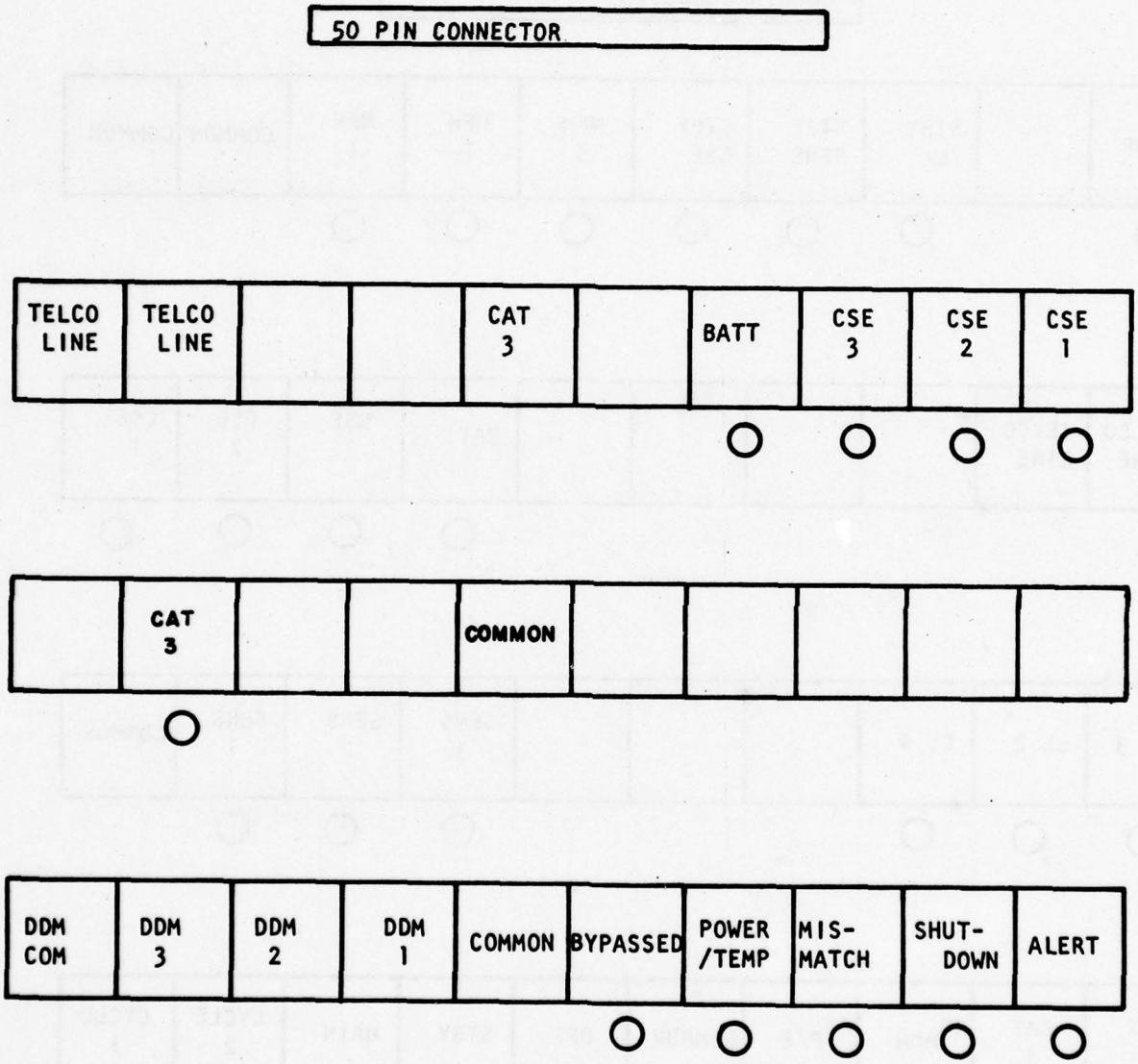


FIG. 14 FRONT PANEL BOARD

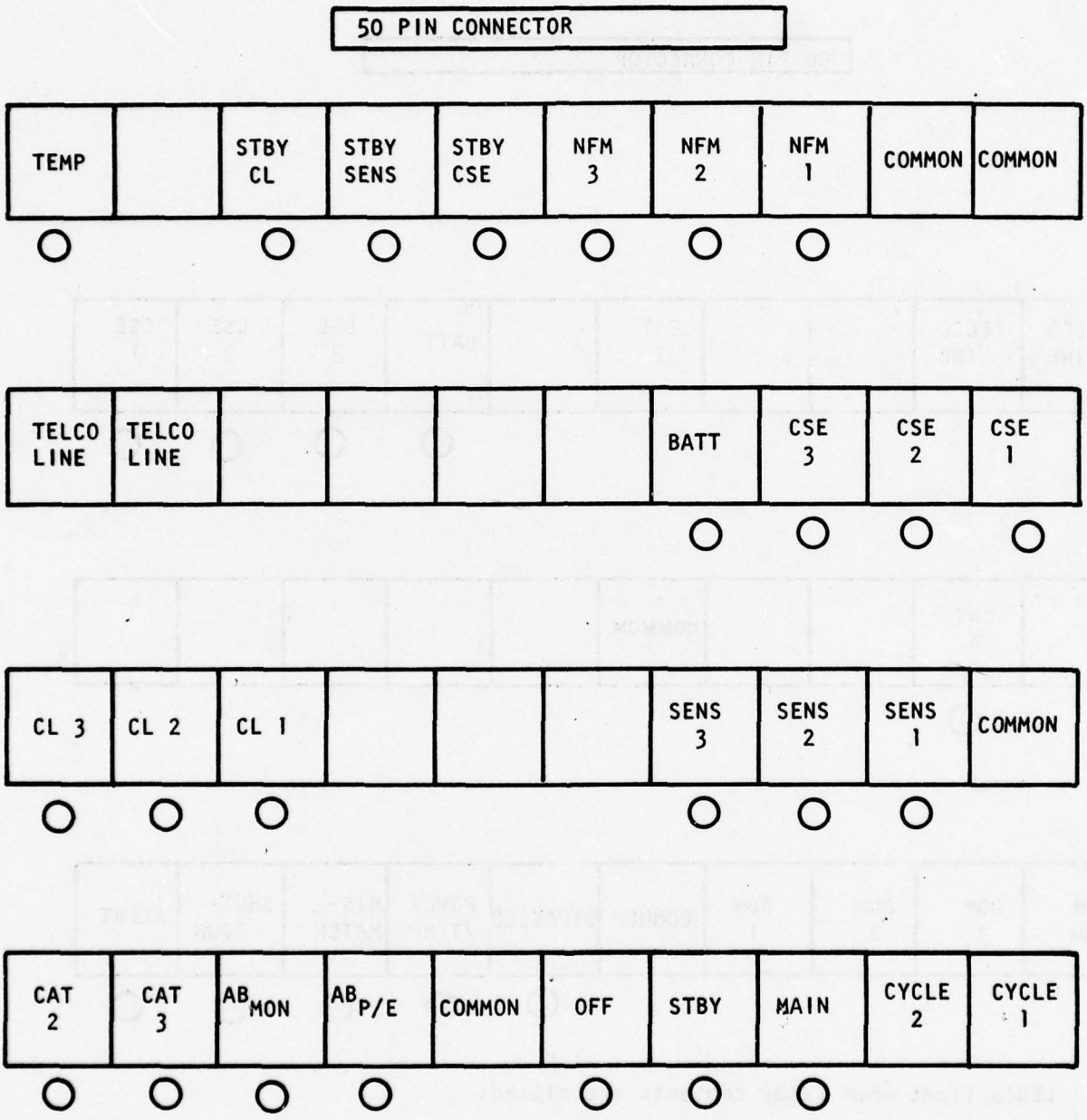
FAR-FIELD MONITOR BARRIER STRIP BOARD



LED's light when relay contacts are closed.

FIG. 15

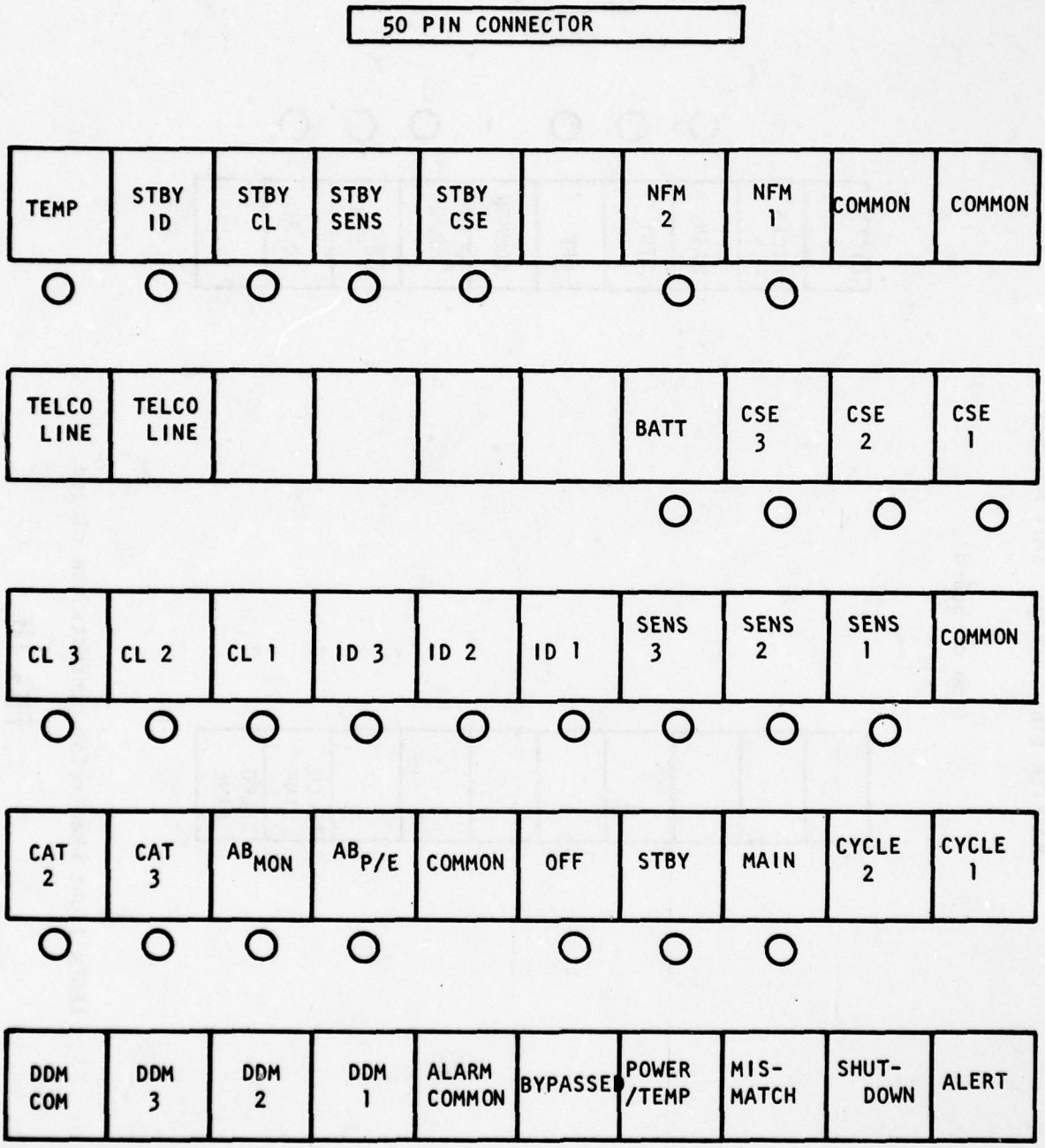
GLIDE SLOPE BARRIER STRIP BOARD



LED's light when relay contacts are closed.

FIG. 16

LOCALIZER BARRIER STRIP BOARD

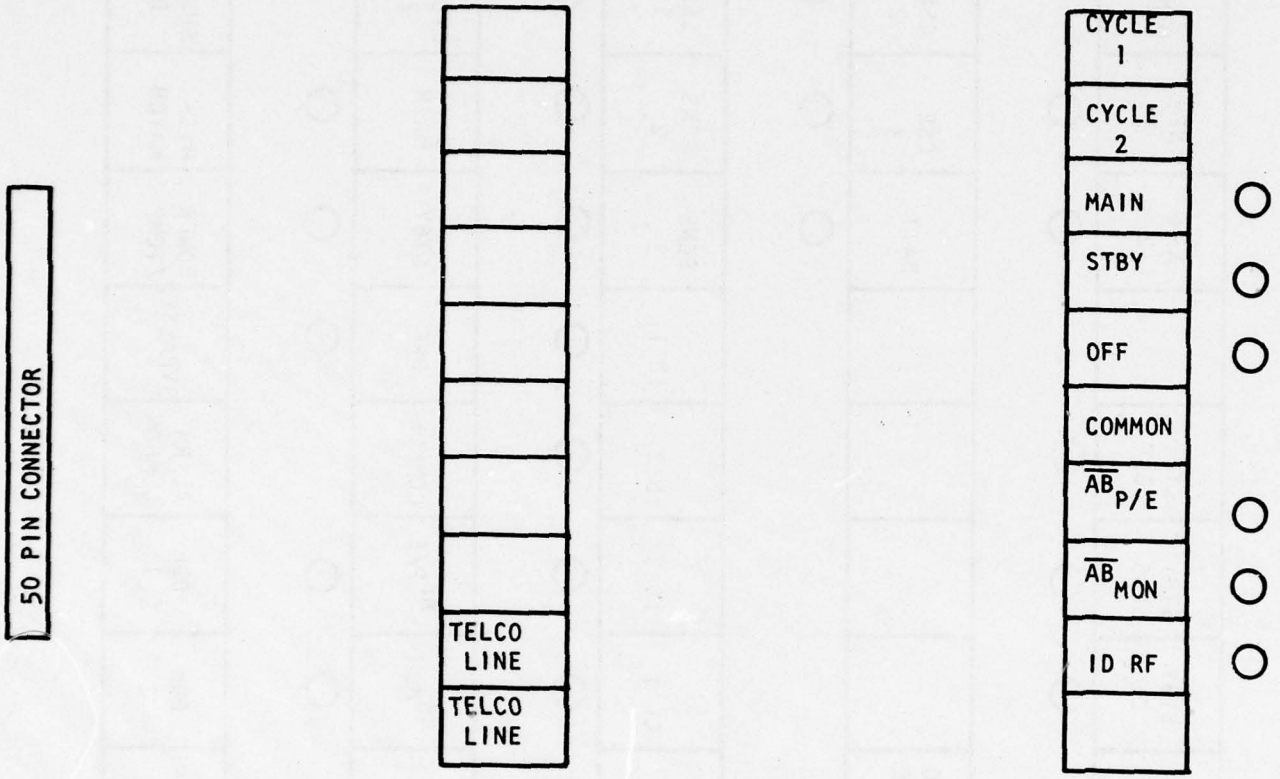


LED's light when relay contacts are closed.

FIG. 17

BARRIER STRIP BOARDS (All markers)

(top of board)



LED's light when relay contacts are closed.

FIG. 18

D. A/D CARD

A schematic of the A/D card used at the Far Field Monitor is shown in Fig. 19. This board serves two purposes: (1) to convert the selected DDM analog signal to an 8-bit digital word, and (2) to provide the μP. with a choice of reading either the 8-bit A/D output or the ILS relay outputs and A/D converter status signals.

The 3404 and AH5012 are used by the μ.P. to select the desired DDM signal. The AH5012 contains 4 FET analog switches which are open circuits when their gates are at logic 1 and are short circuits (150 ohms) when their gates are logic 0. The microprocessor may select the DDM signal to be converted by setting the proper levels on the 3404 inputs and applying a 1 to 0 transition to the $\overline{\text{Input select}}$ input pin on the 3404. If, for example, DDM 1 is to be converted, then a 1 is placed on A/D board pin 27, 0's on pins 25, 29, 23, and a 1 to 0 transition is applied to pin 21.

The DDM voltage is buffered by the LM308AN operational amplifier. The 100k pot in the LM308AN Feedback circuit is set so that a $\pm$1.024V level on the DDM input will result in a maximum or minimum value at the digital output. The microprocessor initiates a conversion by applying a 1 to 0 transition to A/D board pin 28. Note that the microprocessor may select DDM COMMON as an analog input. This allows the microprocessor to find the differences between the digital outputs that corresponds to an analog difference between a DDM signal and its common. This feature also protects against a drift in the zero volts level in the A/D converter chip and its input buffer.

The register select control line on the A/D board allows the μP to select the 8-bit digital output word when REG SELECT is low, or the relay

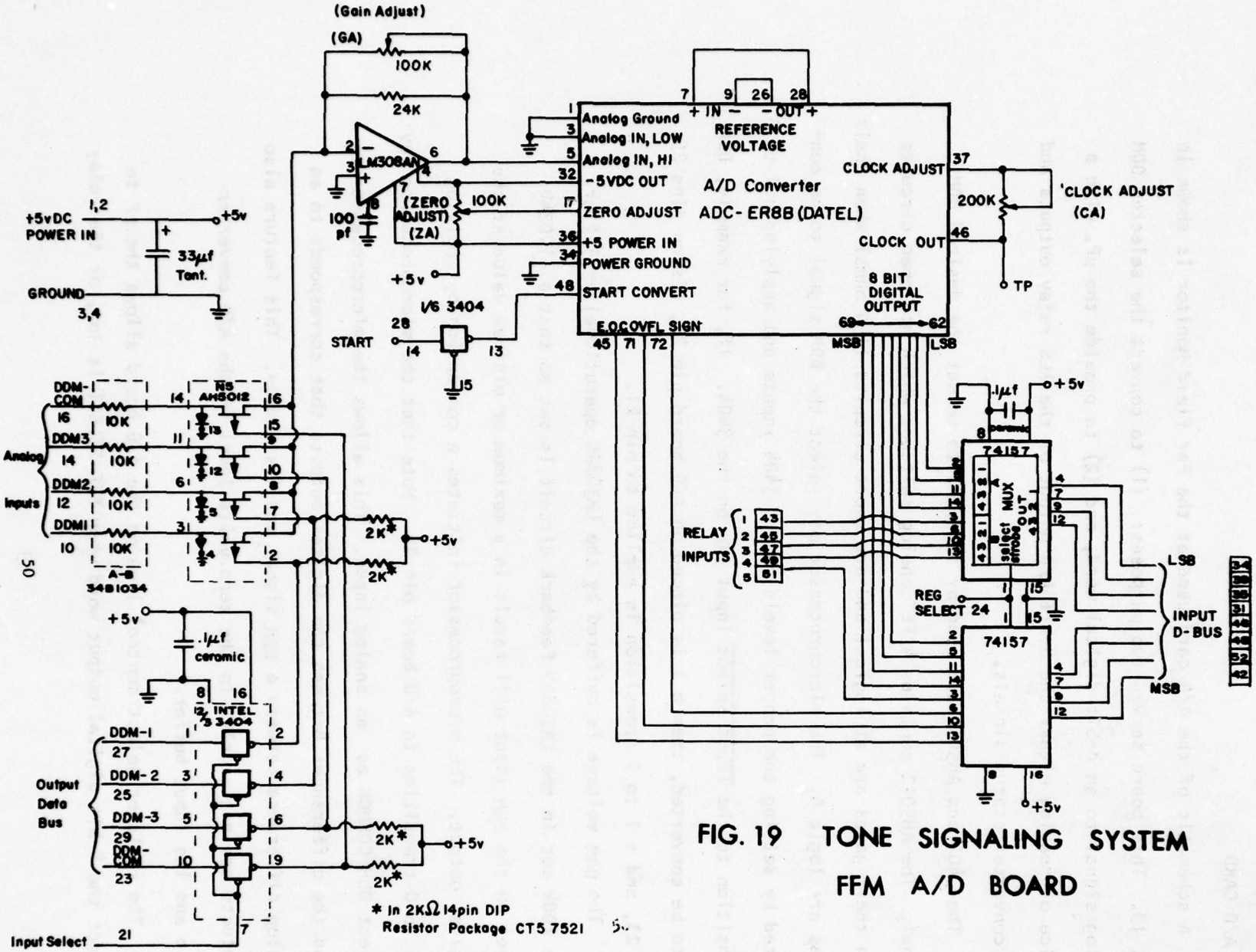


FIG. 19 TONE SIGNALING SYSTEM

FFM A/D BOARD

inputs and A/D chip status lines when REG SELECT is high. When the EOC output, pin 45 of the A/D chip, goes low, the microprocessor knows that the A/D conversion is complete. Pin 71 of the A/D chip signals an overflow has occured, and pin 72 is the sign of the digital output (0 for positive, 1 for negative).

In summary, the steps required to convert a DDM input are:

(1) select the desired analog input

(2) strobe the input select line

(3) monitor the EOC output for a high to low transition (REG SELECT is high)

(4) read the sign bit

(5) set REG SELECT low and read the 8-bit digital representation of the analog input.

These five steps must be performed twice. Once for the DDM common, and once for the DDM 1, 2, or 3 input. The difference is taken by the microprocessor to get the actual DDM level.

There are 3 pots on the A/D board to calibrate the A/D converter chip. The procedure for making these adjustments are:

(1) Allow five minutes warm up of the board.

(2) Adjust the 200k, clock adjust (CA), pot so that a frequency of 122.9 KHz is observed at TP-1.

(3) Select DDM-COM to be converted. Set the zero adjust (ZA) pot so that the magnitude of conversion is all zeroes, and the sign bit flickers between high and low levels.

(4) Now select DDM-1 with 1.024 volts applied to the DDM 1 terminal. Set the 100K gain adjust (GA) pot so that the magnitude of the conversion is all one's (full scale) and the sign bit is zero.

(5) Recheck zero adjustment. Adjust if necessary.

(6) Repeat (4), (5) for a -1.024 volt DDM 1 and a sign bit of 1 output.

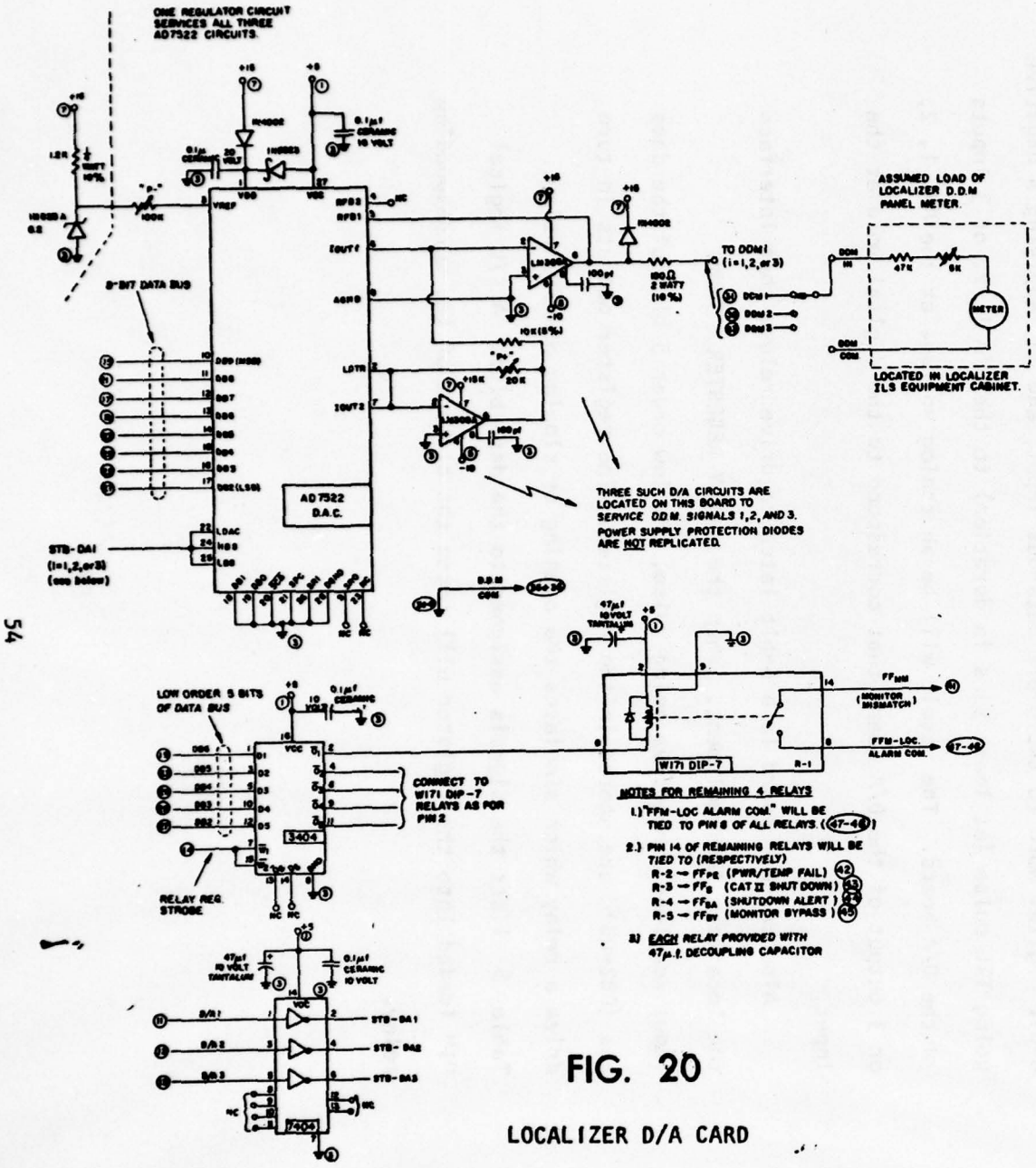
(7) Repeat (4) - (6) for DDM 2, DDM 3 and if necessary readjust the 100K gain adjust pot for an optimum value on all channels.

E. D/A CARDS

The D/A card is used in the localizer to convert digital DDM information into analog voltage to drive a meter in the ILS equipment. A schematic of the D/A converter is shown in Fig. 20 .

The D/A conversion is performed by the AD7522 chip. This chip is basically an R-2R ladder circuit with some additional holding registers to interface the digital source. A reference voltage and two operational amplifiers are added to complete the R-2R ladder circuit. Conversion of an 8-bit parallel digital word is accomplished by: (1) presenting the 8-bit digital word to the 8-bit data bus input, and (2) applying a negative going TTL pulse (at least 5 μs in duration) to the D/A 1, 2, or 3 inputs on the D/A board. The result will be an analog voltage at the DDM 1, 2, or 3 output of the D/A board that correspond to the digital word at the input.

Also on this board is a 6-bit latch to drive relays that interface the localizer ILS equipment. When the RELAY REGESTER STROBE input receives a negative going pulse, the low order 5 bits of the data bus (DB2-DB6) are stored in the register. The register outputs in turn drive a relay which simulates the opening or closing of a FFM relay. Table 5 lists the signals assigned to the data bits. A TTL logical "1" loaded into the register will cause the closure of the corresponding relay.



NOTES FOR REMAINING 4 RELAYS

1.) "FFM-LOC ALARM COM." WILL BE TIED TO PIN 8 OF ALL RELAYS. ((47-48))

2.) PIN 14 OF REMAINING RELAYS WILL BE TIED TO (RESPECTIVELY)
- R-2 — FF_{PS} (PWR/TEMP FAIL) (42)
- R-3 — FF_{S} (CAT II SHUT DOWN) (43)
- R-4 — FF_{SA} (SHUTDOWN ALERT) (44)
- R-5 — FF_{BY} (MONITOR BYPASS) (45)

3.) EACH RELAY PROVIDED WITH 47μ.f. DECOUPLING CAPACITOR

NOTES:

1.) CIRCLED NUMBERS REFER TO CARD EDGE CONNECTOR PIN NUMBERS.

2.) THE FOLLOWING CARD EDGE CONNECTOR PINS SHOULD BE WIRED TOGETHER:

1↔2 3↔4 5↔6 7↔8 35↔36 47↔48

3.) THE FOLLOWING CARD EDGE CONNECTOR PINS ARE NOT USED:

9	10	16	18
20	22	24	26
28	30	34	37
38	39	40	46
49	50	51	52
53	54	55	56

FIG. 20

LOCALIZER D/A CARD

Table 5 D/A Card Data Bus Assignments

Data Bus Bit	Signal from FFM to Localizer
DB6	FF_{MM} (monitor mismatch)
DB5	FF_{PE} (pwr./temp. fail)
DB4	FF_{S} (CAT II shutdown)
DB3	FF_{SA} (shutdown alert)
DB2	FF_{BY} (monitor bypass)

The calibration procedure for the D/A card is as follows:

(1) Load a "11111111" (all ones) into the DAC input register using the D/A 1, 2, or 3 control line as described above.

(2) Adjust potentiometer "P-" so that the output analog voltage ddm 1, 2, or 3 reads -1.016 volts.

(3) Load "00000000" (all zeroes) into the DAC input register.

(4) Adjust potentiometer "P+" so that the output voltage reads 1.024 volts.

(5) Load "10000000" into the DAC input and check to see that the analog output is within.003V of zero.

A similar procedure which uses the microprocessor and the thumbwheel switch for calibration is included in Chapter VII.

F. Input Multiplexer Card

A block diagram of the input multiplexer card is given in Fig. 21 along with its schematic in Fig. 22 . The card uses lines A, B, C to select one of 8 groups of a 4 bit nibble of data. The ENABLE line is not used. The card is wired for non-inverting operation. If the 4th group of data is needed, then lines ABC would be set to 110 respectively following the binary convention shown in Table 6.

Table 6

Select Line A	B	C	Select data group
0	0	0	0
1	0	0	1
0	1	0	2
1	1	0	3
0	0	1	4
1	0	1	5
0	1	1	6
1	1	1	7

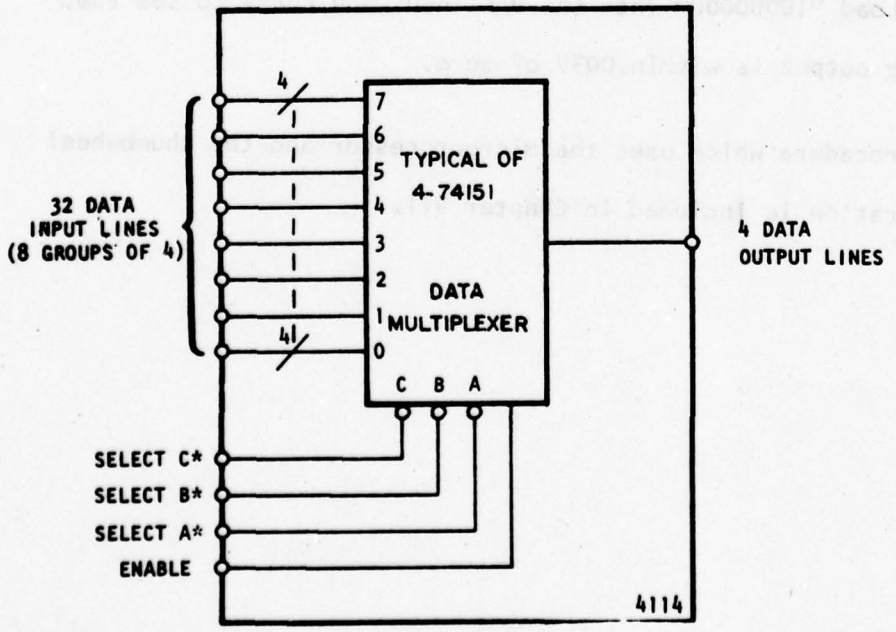
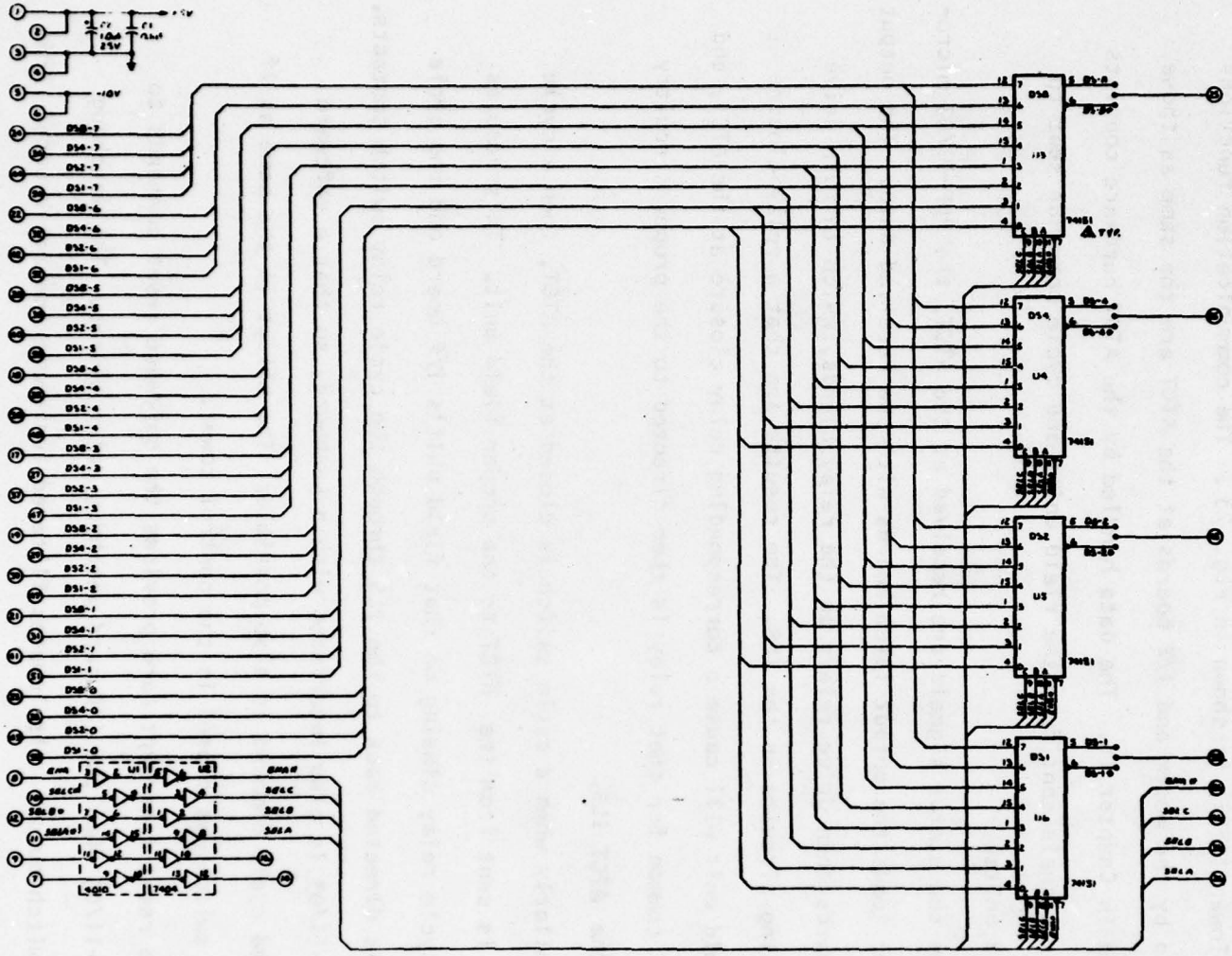


FIG. 21 INPUT MULTIPLEXER CARD BLOCK DIAGRAM



⚠ USE 74S51 FOR WIRE-OR OUTPUT CAPABILITY.

FIG. 22 INPUT MULTIPLEXER CARD SCHEMATIC DIAGRAM

CHAPTER V

COMMUNICATION HARDWARE AT THE AIR TRAFFIC CONTROL TOWER

A block diagram of the communication hardware at the Air Traffic Control Tower (ATCT) is shown in Fig. 23. The communication functions performed by the modem and I/F boards at the ATCT are the same as those discussed in Chapter IV. The data handled by the ATCT hardware consists of status signals sent from the field units and cycle commands sent to the field units.

When the status signals are received at the ATCT, the PDP-11/03 microprocessor loads the output latch boards with the received data. The output latch boards then drive relays on the relay boards, which in turn drive the sensing circuits in the ILS. The results are that a relay closure at a field unit will cause a corresponding relay closure at the ATCT, and the ATCT common for that relay is then directed to the proper circuitry within the ATCT ILS.

Similarly when a cycle switch is closed at the ATCT, then a cycle command is sent from the ATCT to the proper field unit. This results in the cycle relay closing on that field unit's I/F board and the cycle common is directed back to the ILS through the cycle relay switch contacts. The PDP-11/03 is also interfaced with a keyboard, so that a software initiated cycle command is also possible. The effect is the same as if a cycle switch was closed in the control tower.

The reset interrupt card provides the software reset commands to the PDP-11/03 when the I/F card watchdog relay closes. The watchdog relay switch output is lowpass filtered to remove contact bounce. This slowly rising signal is sensed by the schmidt trigger input of the one-shot on the interrupt card. The one-shot output is a $\overline{\text{TTL}}$ level pulse to the

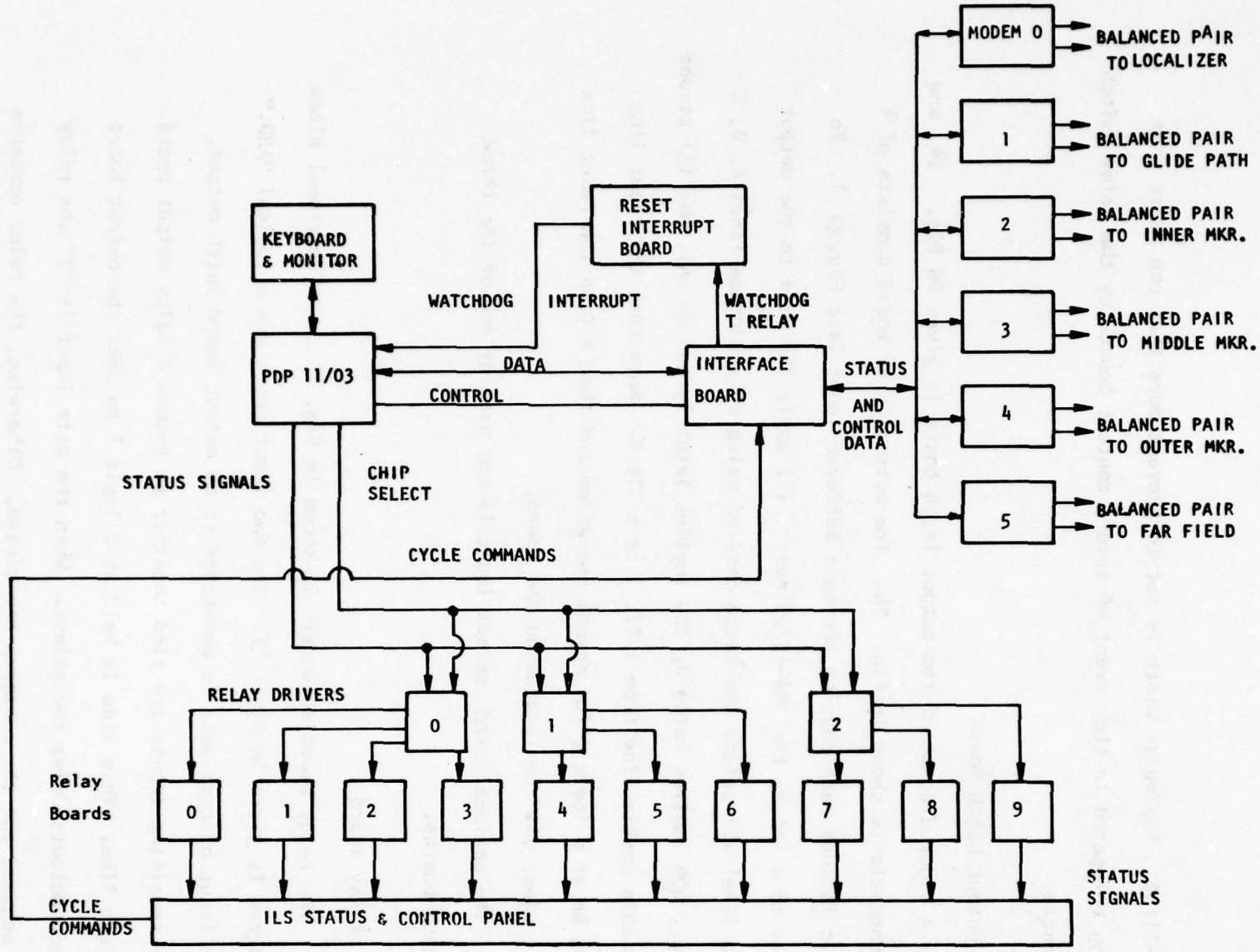


FIG. 23 ATCT BLOCK DIAGRAM

PDP -11/03. The pulse width is set to prevent more than one pulse from being triggered in the event of severe contact bounce by the relay switch contacts.

A. Output Latch Board

A block diagram of the output latch board is given in Fig. 24, and a schematic is shown in Fig. 26. The output latch board consists of 4 8 bit latches each sharing the same buffered input (see Fig. 24). To write to a latch, the PDP-11/03 must: (1) apply the data to the output data line; (2) select the latch desired using chip address lines A, B, C (e.g., 000 implies latch 0, 100 implies latch 1, and so on); and (3) strobe the chip enable line from a TTL 1 to a TTL 0. Note that the reset line must be at a logic 1 for normal operation and that a 0 on the reset line will clear all the latches on the board.

The optional card select logic is not used by any of the three output boards.

B. Relay Board

The relay board schematic is shown in Fig. 27 . A functional block diagram is given in Fig. 25. The two input gates are a logical "AND." One input of each gate is connected to an output board latch output. The remaining inputs are tied together to become a relay output board select line. This line is held at a logic 1 so that the output board latch outputs drive the relays. When the gate input is "0" the relay is engaged and the contacts are closed. Otherwise, the relay contacts are open.

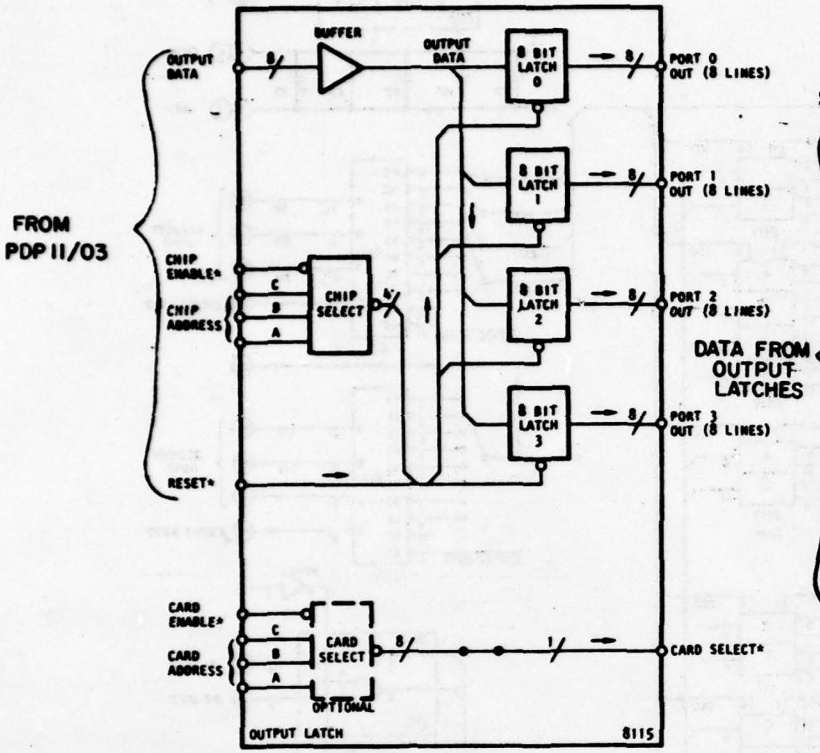


FIG. 24 OUTPUT LATCH BOARD FUNCTIONAL BLOCK DIAGRAM

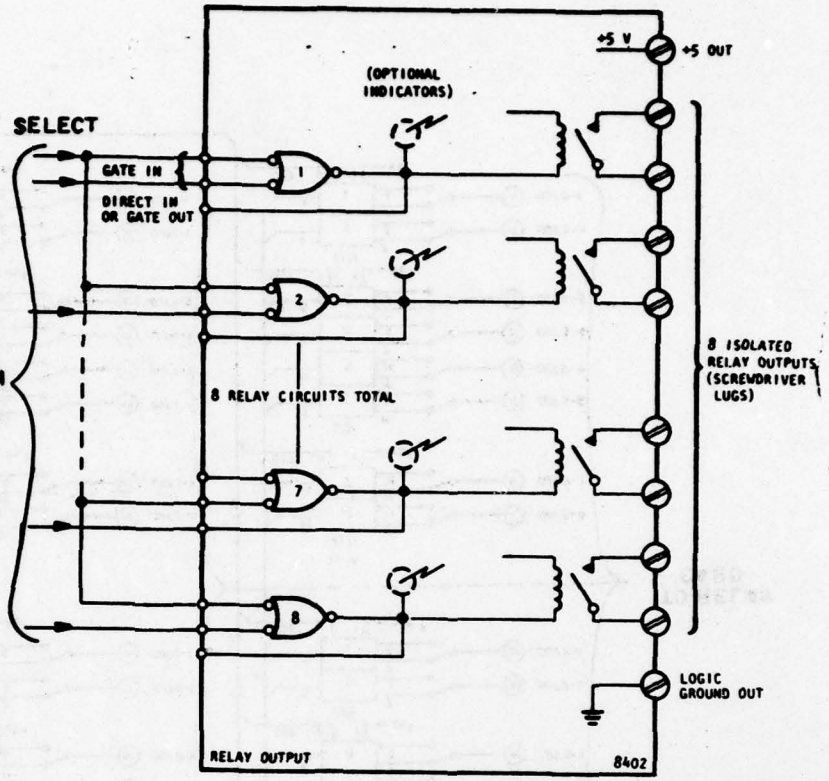


FIG. 25 RELAY BOARD BLOCK DIAGRAM

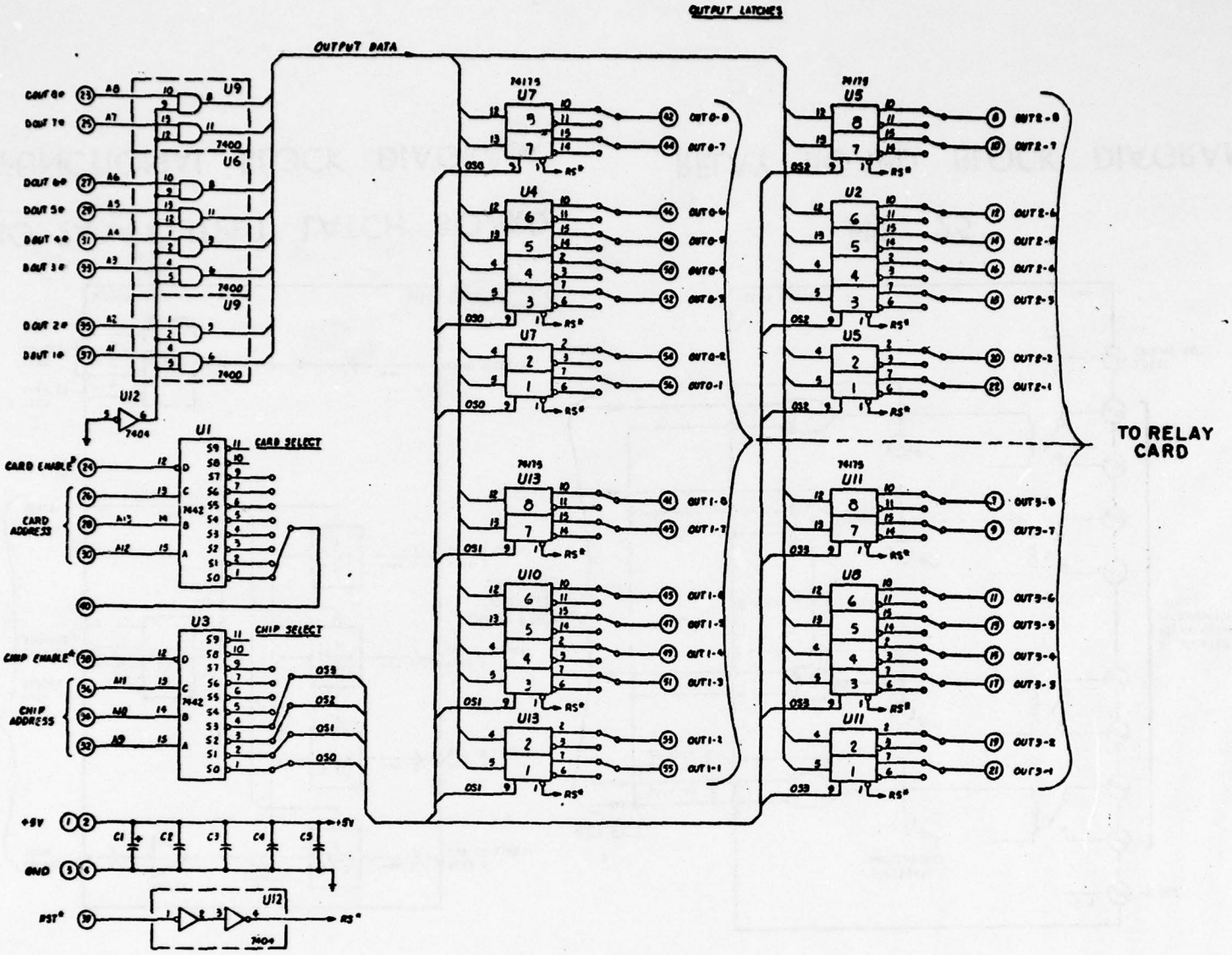


FIG. 26 OUTPUT LATCH BOARD SCHEMATIC DIAGRAM

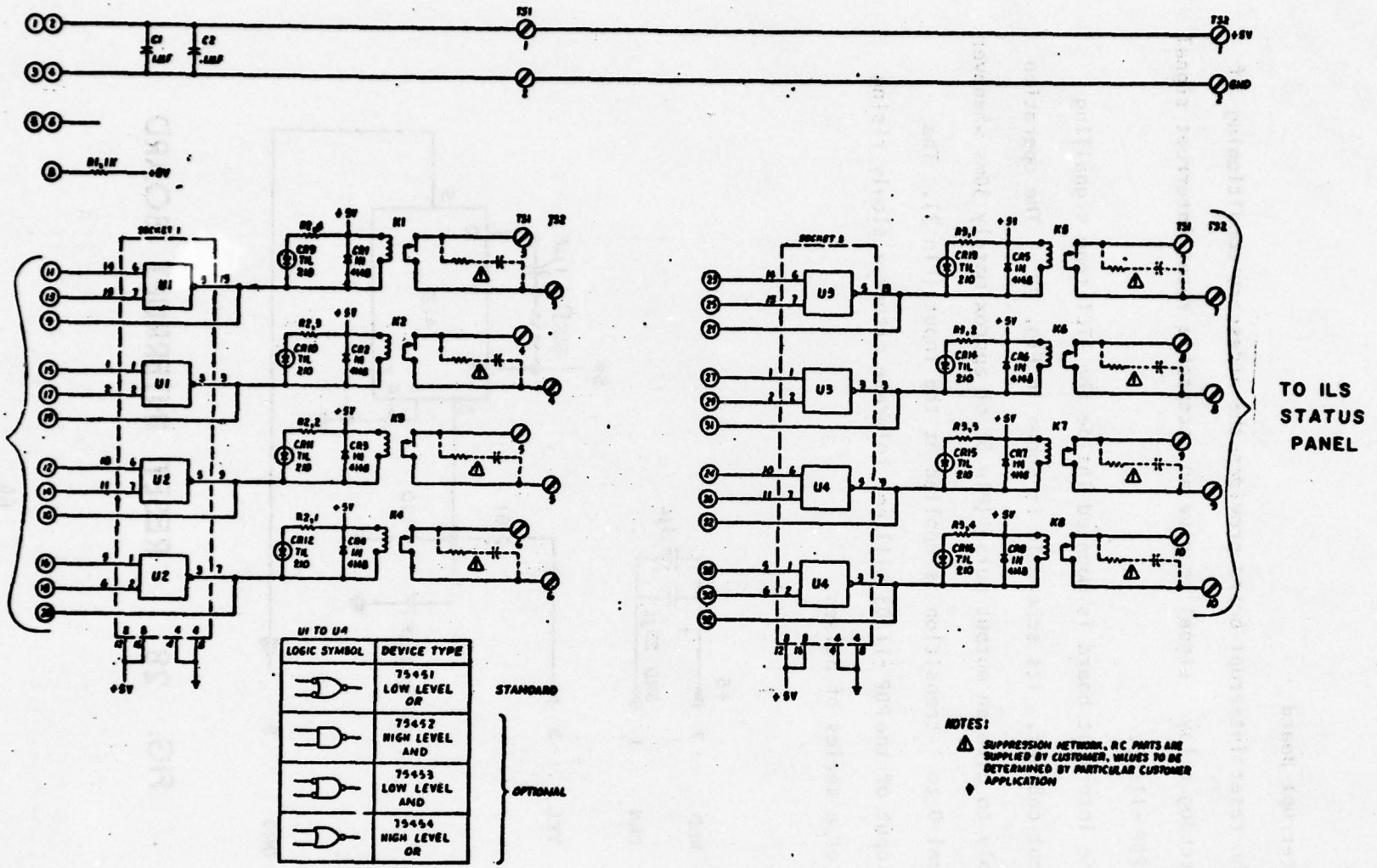


FIG. 27 RELAY BOARD SCHEMATIC DIAGRAM

C. Interrupt Board

The reset interrupt board provides the necessary conditioning of the watchdog relay signal to give an acceptable reset interrupt signal to the PDP -11/03.

The interrupt board is mounted inside the ATCT tone signalling equipment cabinet. Its schematic is shown in Fig. 28 . The operation is simply to give an output pulse (Pin 5) of approximately 30ms whenever a logical 0 to 1 transition is applied at the input (Pin 3). The reset input of the PDP -11/03 will not tolerate either a slowly rising signal or a series of pulses.

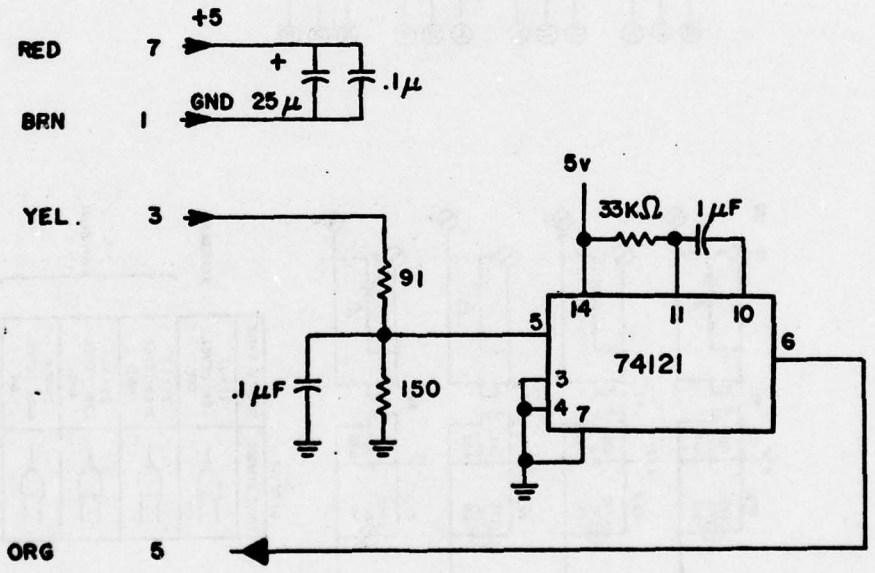


FIG. 28 RESET INTERRUPT BOARD

Chapter VI

INTERFACE BOARD ADDRESSING

The Interface Board was designed to allow either the PDP-11/03 or Pro-Log 4114 microcomputer card to control the remaining peripheral cards in the communications system. A detailed discussion of this card is contained in Chapter IV. The operation of the interface and other peripheral cards are governed by an 8 bit "address" from the computer at each location.

The function of each address is included below. The MSB of the address governs an edge-triggered monostable multivibrator, which triggers on the low to high transition. Hence, addresses 80 through FF hexadecimal assume the prior value of the MSB was zero.

As implemented at the ATCT using the PDP-11/03

Hexadecimal Address	Function
00	Read "CYCLE" buttons
0C*	Write modem control register (modem #0)
10,11,12,13,20,21,22,23, 30,31,32,33	Relay latch banks (3 cards, 4 banks each)
88*	Write ACIA control register (modem #0)
89*	Read ACIA status register (modem #0)
8A*	Write ACIA transmit buffer (modem #0)
8B*	Read ACIA receiver buffer (modem #0)
F0	Watchdog circuit reset

Addresses marked with an asterisk (*) should have the following number added to them for the other five modems:

10 for modem #1 20 for modem #2

30 for modem #3 40 for modem #4

50 for modem #5

As implemented at the field units using Intel 4040's

Hexidecimal Address	Function
0E	Write modem control register
18	Start A/D conversion
18	Load D/A 3
28	Load analog switch register
28	Load D/A 2
38	Load D/A 1
39	Read A/D magnitude bits
3B	Read A/D card inputs
48	Write D/A card relay register
49	Read interface card inputs
59	Read interface card clock, id straps, and thumbwheel switch
5F	No operation (safe rest state)
68	Write cycle relay register
78	Write 7 segment display
88	Write ACIA control register
89	Read ACIA status register
8A	Write ACIA transmit buffer
8B	Read ACIA receive buffer
F8	Reset watchdog circuit and write 7 segment display

Chapter VII

THE FIELD UNITS

A. DISPLAYS

Each of the six field units (at the localizer, glide slope, far-field monitor, and three markers) have a seven segment light emitting diode (LED) display and thumbwheel switch mounted on the front panel. These displays facilitate routine servicing of the communication system without test equipment. In most cases, problems may be identified by reading the English message flashed on the display by the microprocessor.

There is one display associated with each setting of the thumbwheel switch. The meaning of the displays are explained below:

POSITION 0

The normal setting for the thumbwheel switch is position 0. In this position, the microprocessor displays English messages describing the current status of the communication system. The most common message is "GO" which means that everything is working as designed: communication with the remote control is in progress and no transmission errors (caused by electrical noise or interference) are being encountered.

The message is spelled out sequentially by three characters on the display:

The first character, a dash or hyphen, indicates the start of a message. Each character appears on the display for approximately one second. The second character is a "G", and the third is an "O". The seven segment alphabet is shown in Fig. 29 . The decimal point or period is lit with

Seven-Segment Display Alphabet

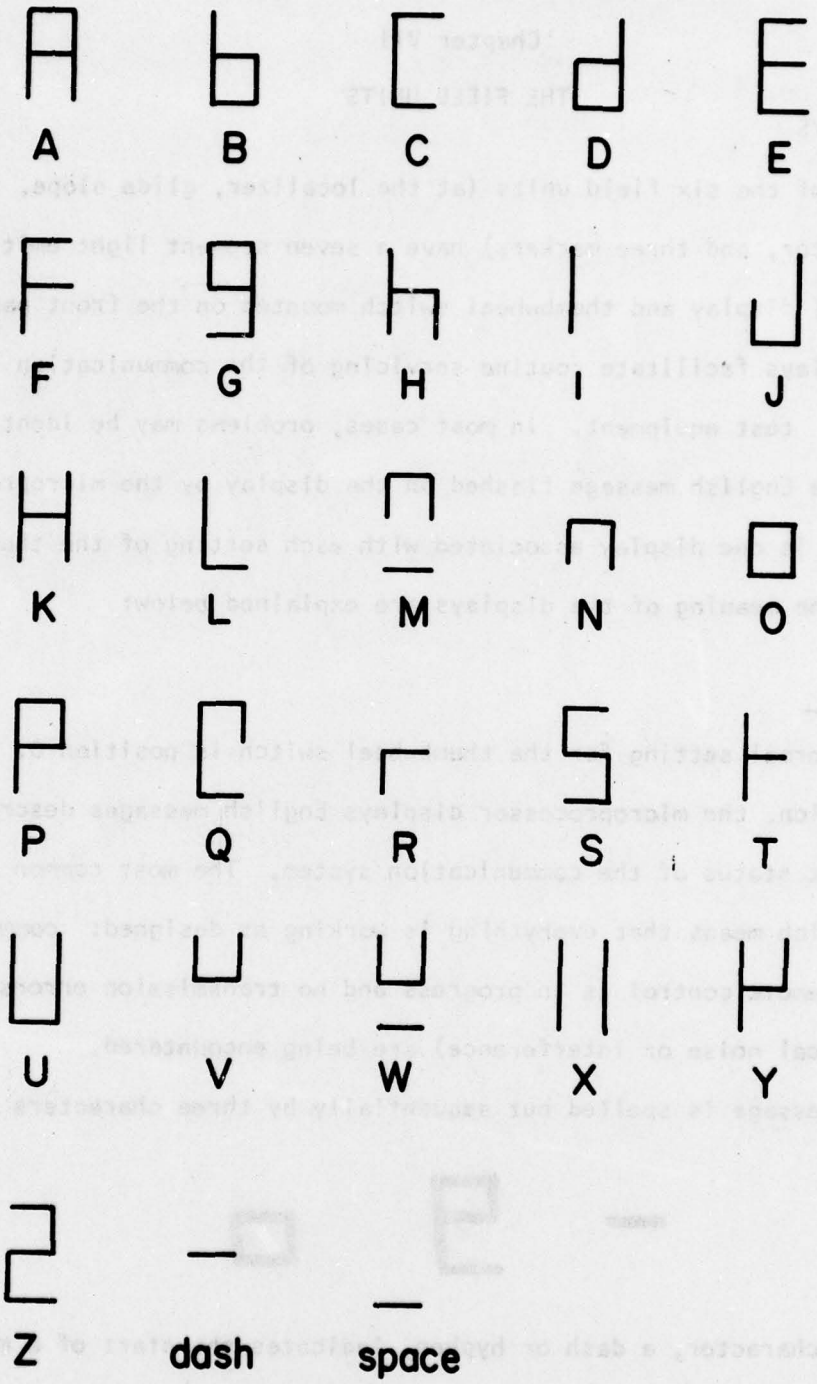


FIG.29

the last character of the message being displayed. This signifies the end of the message.

The remaining 14 messages which can be displayed in position 0 have the same general format: they begin with a hyphen and end with a period.

Each of the fifteen messages has an assigned priority, higher priority messages interrupt lower priority messages. A summary of the messages, their assigned priority, and a brief explanation is listed below:

Priority	Message	Explanation
1	GO	System go, no problems.
2	DDM3	Too large a voltage is being sensed at the FFM for DDM 3 (greater than $\pm$1 volt). Will appear only at the FFM and LOC.
3	DDM2	See explanation for DDM 3.
4	DDM1	See explanation for DDM 3.
5	ADC	Problems with the analog-to-digital converter are being detected at the FFM (zero calibration has drifted or no end-of- conversion signal is detected). Will appear only at the FFM and LOC.
6	OVER	An overrun error occured during the reception of data from the Remote Control.
7	FRAME	A framing error occured during the reception of data from the Remote Control.
8	CRC	A data error occured during the reception of data from the Remote Control (either a CRC error or other redundancy check).
9	PARITY	A parity error occured during the reception of data from the Remote Control.
10	RESET	The microprocessor was just reset, either by the front panel pushbutton or by the watchdog.
11	ID	The message received from the Remote Control should have been sent to a different field unit.

12	NO DATA	The carrier from the Remote Control is present, but no data has been received for the past 30 seconds.
13	TONE	No carrier is being received from the remote control.
14	MODEM	The modem card failed its self-test.
15	PROM	A checksum of the microprocessor's Programmable Read Only Memory failed.

POSITION 1

Moving the thumbwheel switch to position 1 will cause all eight LED's in the display to light as a lamp test.

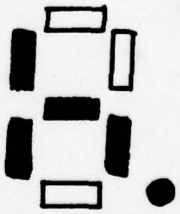
POSITION 2

Position 2 of the thumbwheel switch displays the last frame of information received from the Remote Control to be displayed. Each LED in the display represents 1 bit of the eight bit frame. When information is sent to the Localizer, this display will flicker as several frames are normally sent to the localizer in quick succession.

Typical displays (cycle button not pushed, no reception errors at the ATCT) are:

Glide Slope

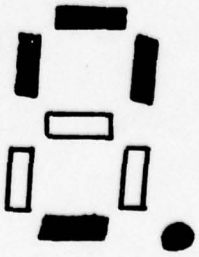
FFM



Inner Marker

Middle Marker

Outer Marker



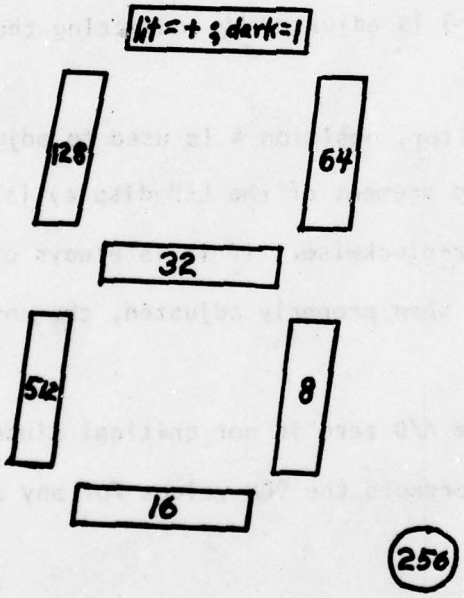
OM (cycle pushed)

Note that shaded segments above indicate segments which are <u>not</u> lit. Four bits are used to indicate the cycle button status (open, closed):

(open) ,

(closed) .

<u>POSITION 3</u>: DDM 1 (FFM & LOC)



millivolts

Add <u>dark</u> segments. Divide result by 51.1 (k ohms) to get micro-amps. Example: if the shaded areas below are dark then we get -(512 + 128) = -640 mV and -12.5 μA.

POSITION 4: (no meaning at GS, markers)

At the localizer, this position is used to adjust the D/A converters. The value loaded into the D/A's will be either plus full scale (+1.024 volts) or minus full scale (-1.016 volts). These two values will be loaded alternately every 15 seconds. To adjust the D/A's, connect a DVM to the DDM 1 output and adjust the potentiometer marked "1P&" so that the positive voltage is +1.024 volts, and adjust "1P-" so that the negative voltage is -1.016 volts. (See Fig. 20). These adjustments are interactive, so they should be repeated several times before continuing with DDM2.

DDM 2 is adjusted by connecting the DVM to DDM 2 and adjusting "2P&" and "2P-". Likewise, DDM 3 is adjusted by connecting the DVM to DDM2 and adjusting "3P&" and 3P-".

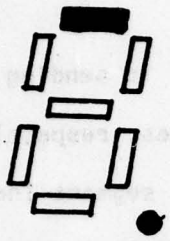
At the far field monitor, position 4 is used to adjust the zero of the A/D converter. If the top segment of the LED display is always on, adjust potentiometer "ZA" counter-clockwise. If it is always off, move the potentiometer clockwise. When properly adjusted, the entire display will blink on and off.

The adjustment of the A/D zero is not critical since the microprocessor automatically corrects the DDM values for any zero drift of the A/D.

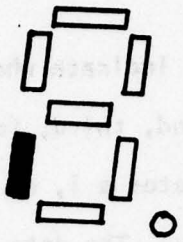
POSITION 5: At the markers, glide slope, and FFM, this position indicates the value of the cycle relay:*

---

*shaded segments indicate dark or unlit LED's.



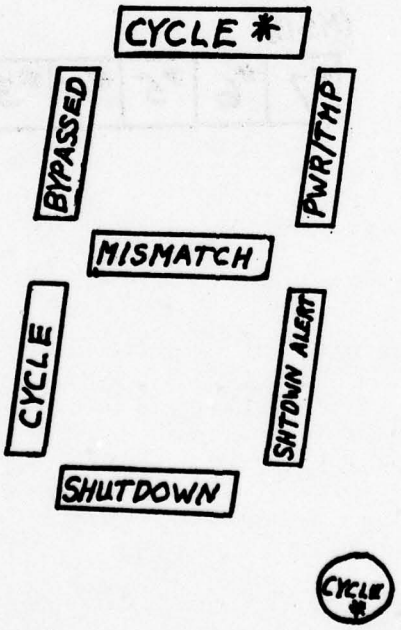
Relay open



Relay closed

Since the FFM has no transmitters, the display should always show relay open.

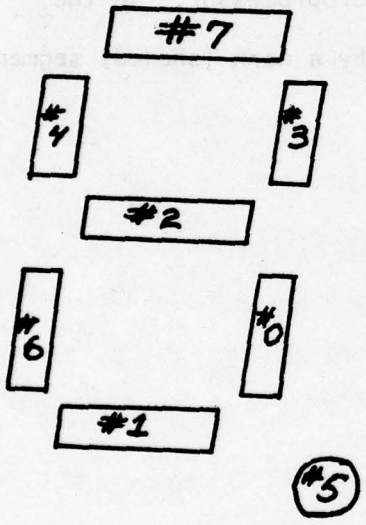
At the Localizer, this position also indicates the status of the five signals from the FFM as received by the microprocessor. In the following diagram, a closed relay is indicated by a dark (shaded) segment:

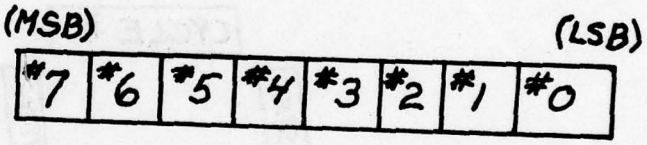


*Cycle relay closed if lit.

POSITIONS 6-9

These positions indicate what the microprocessor is sending to the ATCT in the second, third, fourth, and fifth frames, respectively. A dark segment indicates a 1, or closed relay. A lit segment indicates a 0, or an open relay. The data formats are explained in Chapter III. Each segment of the LED represents 1 bit in the frame as follows:





B. SIMPLIFIED TROUBLESHOOTING PROCEDURE

(1) Place the thumbwheel switch in position 0. If the display is dark, check the power source, switch, fuses, and power supply voltages. If all are okay, proceed to step (2).

(2) If the display is dark, flickers, makes no sense, or works "funny," try replacing the INTERFACE and MICROPROCESSOR cards. Note: the FFM display will usually flicker when first turned on. It will quit flickering within 2-3 minutes.

(3) If the message is not "GO":

DDM3, DDM2, DDM1, ADC: Check the FFM, DDM signals. If okay, try replacying the ADC card at the FFM, then the INTERFACE card.

OVER, FRAME: Try replacing the MODEM, INTERFACE, and MICROPROCESSOR cards, in that order. Also check the MODEM card at the ATCT.

CRC, PARITY: See OVER, FRAME above. Also check the TELCO line for excessive noise or insertion loss, poor balance.

RESET: Try replacing the MICROPROCESSOR and INTERFACE cards.

ID: See CRC, PARITY. Also check to be sure the field unit TELCO line is connected to the right pair of terminals in the ATCT and that two lines have not been switched.

NO DATA: Check the system at the ATCT to be sure the computer is running. Then try replacing the MODEM, INTERFACE, and MICROPROCESSOR cards. Then try the MODEM and INTERFACE cards at the ATCT.

TONE: Check the ATCT system and the TELCO line, then the MODEM, INTERFACE, and MICROPROCESSOR cards. Then try the MODEM and INTERFACE cards at the ATCT.

MODEM: Replace the MODEM card, then the INTERFACE and MICROPROCESSOR cards.

PROM: Replace the MICROPROCESSOR card.

IMPORTANT NOTE

Occasional occurances of these messages do not indicate a problem which needs maintenance!!

(4) If the message is "GO": Check the quick change plugs at the field unit and at the ATCT to be sure they are in the PURDUE/TONE socket.

(5) If plugs at QC boxes are right, isolate the problem using the LED's on the barrier strip boards, the front panel display, and the display at the ATCT. Signals that are correct on the barrier strip board, but wrong when read by the microprocessor may indicate a bad INTERFACE, MUX, ADC, or MICROPROCESSOR card.

C. THE HEATER-THERMOSTAT CIRCUITRY

The majority of the components used in the communications equipment are rated for operation between 0° and 70° Celcius. Since the termperature of the field units not located in heated shelters (FFM and the markers) could go below freezing, these units are equipped with electric heaters.

The heaters are simple units operating from 120 VAC. They maintain the temperature in the interior of the box between 26° and 36° C. A front panel lamp marked "HEATER ON" will light whenever the heater coil is actually operated. The AC power cord for the heater should only be plugged in during the winter when temperatures may fall below freezing. At other times, this heat would only add to the heat generated by the power supplies and electronics.

Should the temperature inside the box fall below freezing, a thermostat will disrupt power to the electronics to prevent malfunctions caused by the low temperature. This condition is signalled on the front panel by the extinguishing of the "TEMP NORMAL" lamp, while the "POWER" lamp remains on.

It is our opinion that when the heater is operational, the electronics will continue to operate with an outside temperature below -30°C. Furthermore, if the AC power fails, the electronics will continue to operate from several hours to indefinitely, depending on the exterior temperature. Although the heater will cease functioning when the AC power fails, the power supplies and electronics will continue to generate considerable heat on their own.

A schematic of the heater-thermostat is shown in Fig. 30 .

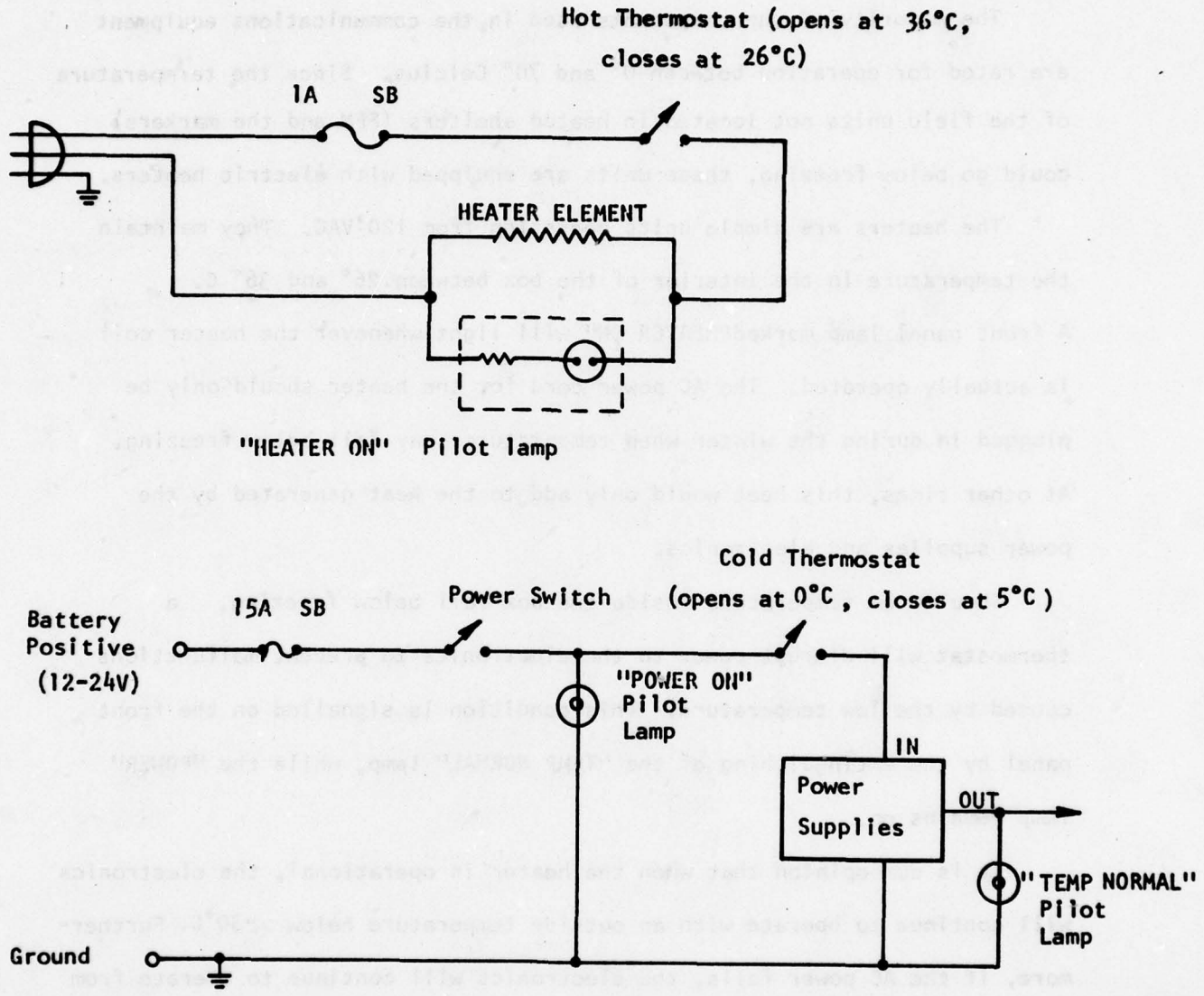


FIG. 30 HEATER SCHEMATIC

CHAPTER VIII
ATCT OPERATOR INFORMATION

A. Basic Equipment

Fig. 31 shows most of the major components installed at the ATCT to handle Mark III ILS communications. All operation sequencing and control is, of course, governed by the μp. (PDP-11/03), but note that the interface card is the center of most activity since the bulk of the μp. information is steered by this card. See Chapters IV and VI for discussions of the interface card.

The interrupt board holds "watchdog" circuitry as described in Chapter V. Should the μp. fail to refresh the watchdog pulse generator (located on the interface board), the interrupt board electronics will force an interrupt to be taken by the μp. The address of the watchdog interrupt service routine is stored in location 300_8.

The maintenance monitor, status, and remote control panels are all part of the original ILS equipment installed at the ATCT.

Each communications line to a distant field unit (FU) is driven by a modem card. See Chapter IV. Electronics on the modem card converts a serial stream of 1's and 0's to a sequence of tone bursts at a 300 baud (bits/sec) rate. Conversely, tone bursts simultaneously received from the far end of the line will be converted back to 1 and 0 signals by this same modem card.

The μp. directly drives a TV terminal as shown in Fig. 31. Information to a human operator is displayed on the TV monitor screen by the μp. The operator may in turn send commands to the μp. by typing appropriate command abbreviations on the terminal keyboard.

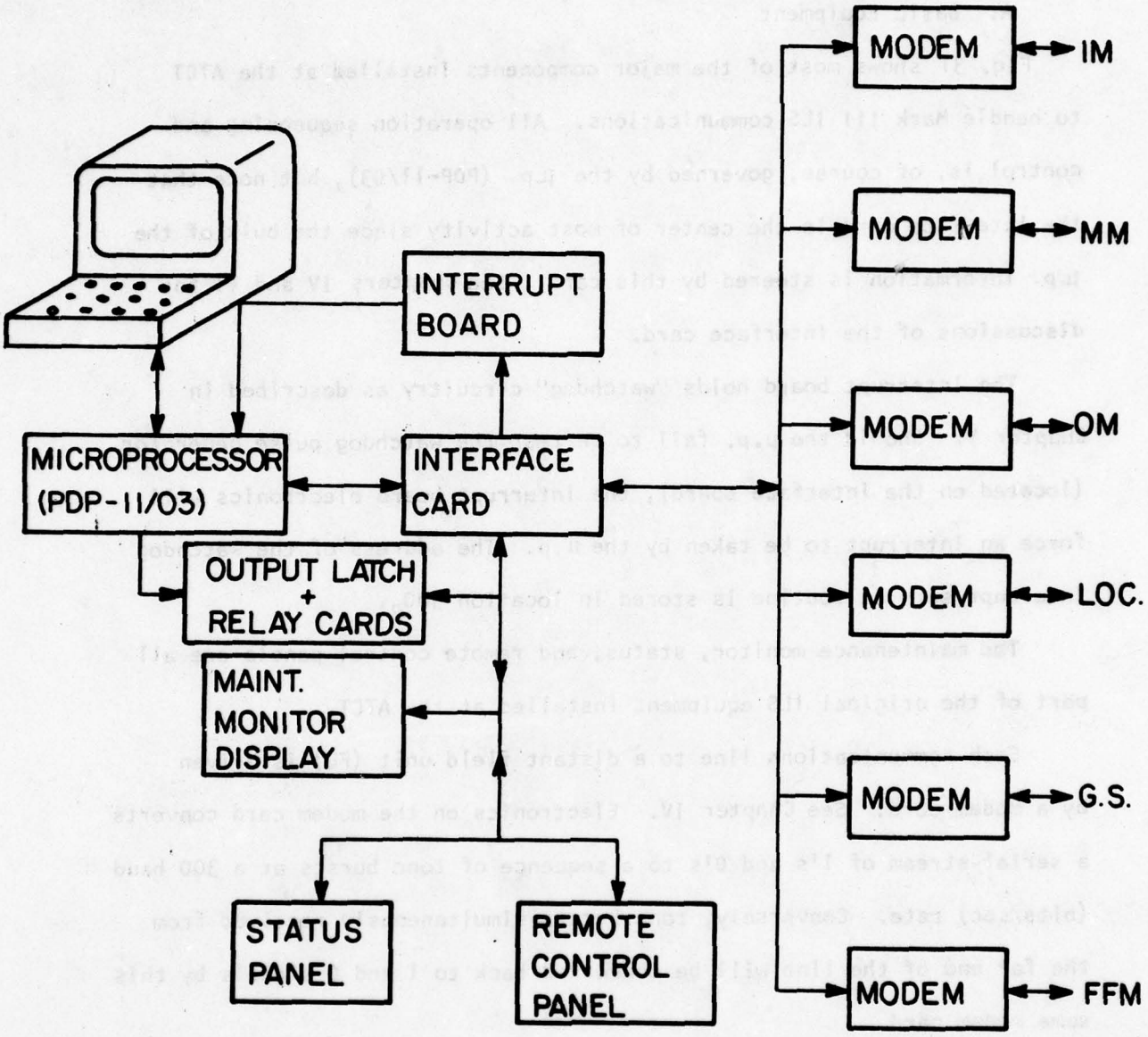


FIG. 31 MAJOR HARDWARE COMPONENTS INSTALLED AT THE ATCT

B. Monitor Information Frames

Several basic types of information frames can be displayed on the monitor screen. Most frequently the status frame will be desired (see Fig. 32.) This frame presents all the information (plus a few additional items) currently found on the status and remote control ILS panels. For each of the transmitting FU's this includes transmitter status (MAIN, STANDBY, or OFF), abnormal alarms (POWER/ENVIRONMENT and MONITOR), and performance indicators such as CAT III and CAT II.

Also, the two most recent log entries currently stored for each transmitting FU are displayed immediately below their status values. At the top of the screen appears a performance category rating of either CATEGORY 3, CATEGORY 2, SYSTEM DOWN, or LINK BREAK. The "CATEGORY" items are determined in a manner identical to that used in the ATCT remote control panel. "SYSTEM DOWN" signals that the ILS does not satisfy Category II rating criteria. "LINK BREAK" is used whenever the absence of either a received carrier or message has persisted for approximately three and one half minutes on any of the six communications lines.

Whenever a transmitter cycle is being requested (by either the keyboard or the remote control panel pushbuttons), a blinking CYCLE is placed at the far right of the status screen display line for that FU.

(The single status signal received from the FFM--CAT III--is not explicitly displayed on any monitor frame. However, its value is used by the μ.p. to determine the performance rating at the top of the status frame.)

The maintenance monitor (MM) frame (see Fig. 33) displays information received from the localizer, glide slope, and FFM FU's that is currently available on the MM status panel. To the left of the frame appear all

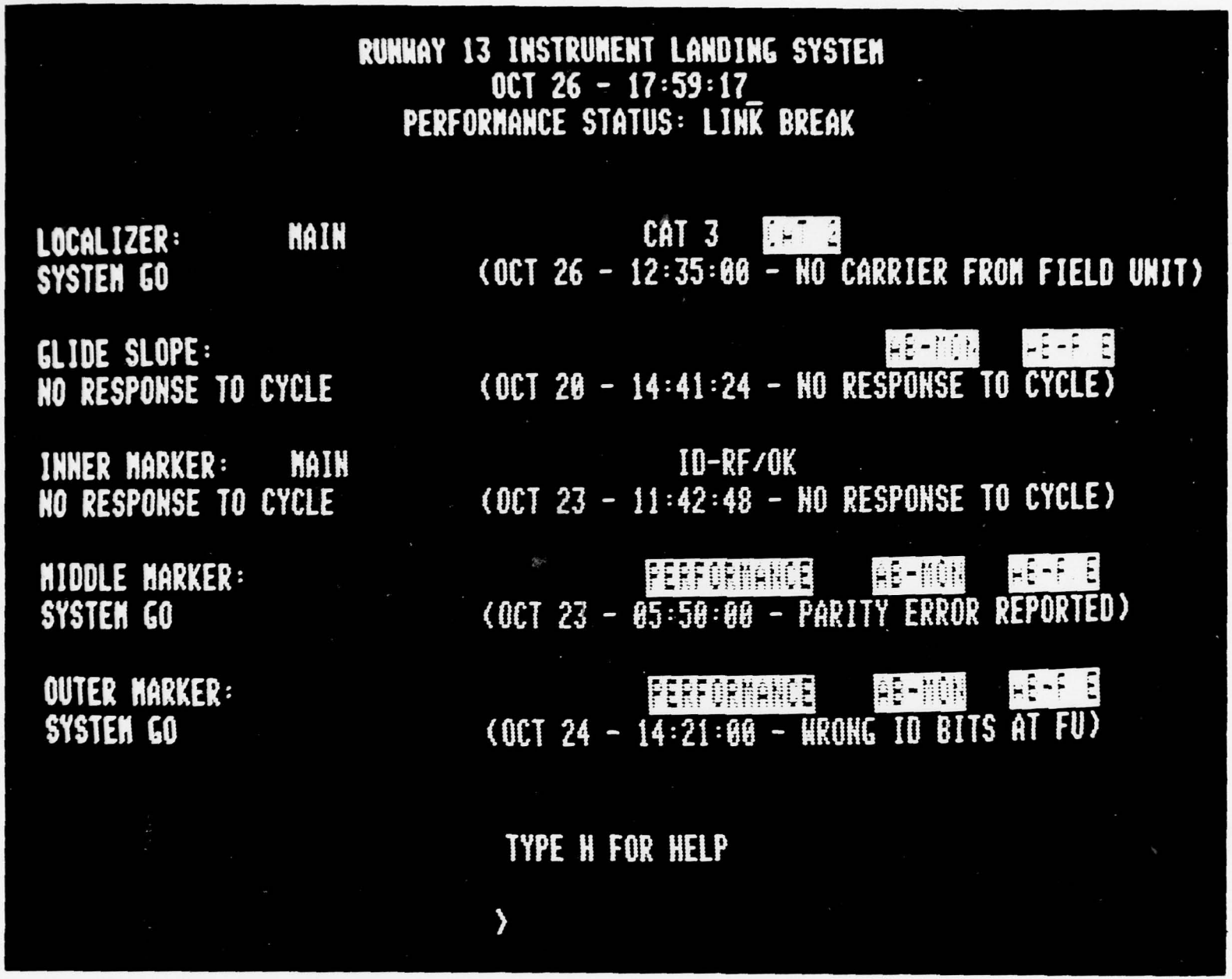
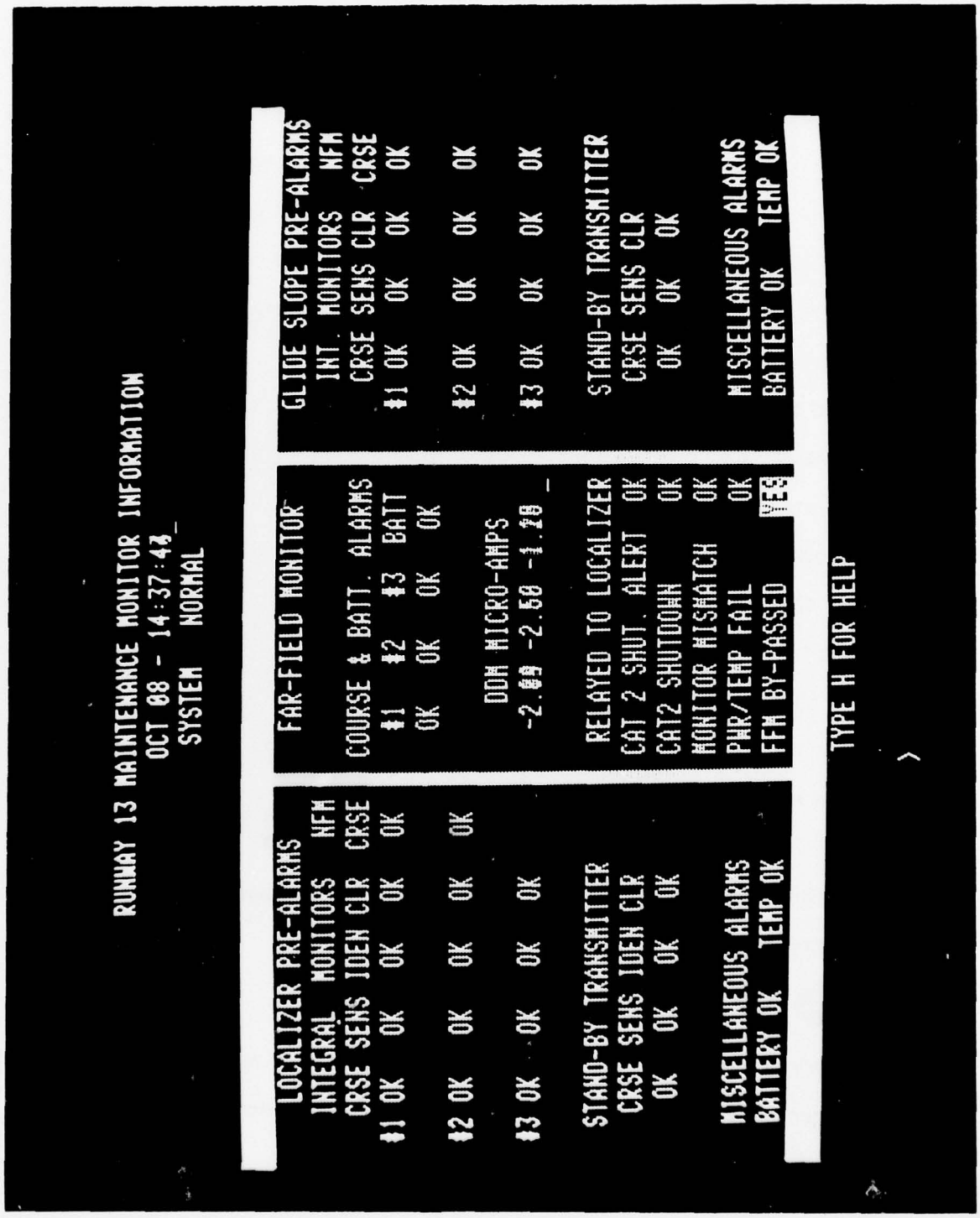


Fig. 32: The "status" information frame as displayed at the ATCT

Fig. 33: The "maintenance monitor" (MM) information frame



prealarm status being received from the localizer. On the right of this same frame is shown glide slope MM status values. At top center is displayed FFM MM status values.

Immediately below the FFM MM data appear items not displayed on the ILS panels: DDM values and the status of five alert and shutdown signals currently being relayed to the localizer from the FFM; see Fig. 2. All three DDM analog voltage values are converted to binary numbers at the FFM. These numbers are relayed through the ATCT to the localizer. However, they are simultaneously displayed on the ATCT monitor screen in decimal microamps.

At the top of the MM frame will appear "SYSTEM NORMAL" or "SYSTEM ABNORMAL." A NORMAL indication will appear only if all MM prealarms displayed on this screen are off or "OK."

Both the status and MM frames show the current date and time at the top of their displays.

A "log" frame may also be written on the monitor (see Fig. 34). Here, up to 20 lines of log entries are displayed for the operator's viewing. Log entries made automatically by the μ.p. are always one line in length. Entries made manually by a human operator may be of any length. As it appears on the screen, a log entry will show the date and time of its insertion into the log as well as its "origin": That is, to which FU the entry applies.

C. Commands Issued from the Keyboard

At any time the user may type "H" (for HELP). An information frame will then appear giving him a choice of four "self-documentation" frames that he may request be written on the screen.

Fig. 34: Making a manual entry to the PDP 11/03 log

```
BEGIN TYPING YOUR LOG ENTRY.  USE "RUBOUTS" TO
BACKSPACE AND CORRECT PREVIOUSLY TYPED CHARACTERS.
USE THE "RETURN" KEY TO STEP TO A NEW LINE IF                E
NEEDED.  TERMINATE YOUR ENTRY WITH A RIGHT BRACKET.          N
BE BRIEF BUT EXPLICIT.                                       D

OCT 26  00:03:00  KYBD  THIS IS A SAMPLE ENTRY FOR PHOTOGRAPHIC PURPOSES_


                                TYPE H FOR HELP

                                >
```

If he next types "?S," he will be shown an information frame covering basic aspects of the status and MM frames. In response to a "?L," the user will be instructed in using the four log commands honored by this μ.p.

Typing "?D," he will learn how to issue "date" and "time" commands. See below. Finally, should he make a "?C" keyboard response, the user will be told how to request the cycling of transmitters from his keyboard. Again, see below.

The form of a "date" command is

DATE MM/DD

followed by the depression of the " RETURN" key. (Characters above that are underlined are typed by the human operator; others are written automatically by the μ.p.) "MM" is the current numeric month (from one to 12) the user desires to enter. "DD" is the day number. When the RETURN key is pressed, the μ.p. will change its automatically-maintained date to that typed in. Then two automatic log entries will be made recording this date change. The "MM" and "DD" numbers will be checked for validity. No date change will be made if there is an obvious error in their values. Errors caught by the user while typing can be overcome by simply continuing to type the date: the μ.p. will ignore all but the last four numeric digits typed in.

To alter the 24-hour clock time being maintained by the μ.p., issue a "TIME" command from the keyboard:

TIME HH/MM

terminated with a "RETURN." "HH" is the current numeric hour (from zero to 23) while "MM" refers to the desired minute number (between zero and 59). At the depression of the RETURN key the μ.p. will change its maintained time to the value typed in. Two automatic log entries will then be made recording this time change. The "HH" and "MM" numbers will be

checked for obvious errors. Again, only the last four numerals typed in will be made use of by the μ.p.

Three types of log display commands may be issued from the keyboard:

LOG (return)

LOG MM/DD (return)

LOG ROLL (return)

The simple "LOG" command results in the last 20 (most recent) log lines being displayed on the monitor screen.

The "LOG MM/DD" command will display 20 log lines beginning with the first line having a date greater than or equal to the date typed in.

In response to the "LOG ROLL" command, an existing log display will be advanced by ten lines. Thus, the log may be stepped through easily beginning with any initial date.

To make a manual entry to the log, the human operator simply types "ENTER." The μ.p. responds by displaying instructions on how to make such an entry.

Cycle commands may be issued from the keyboard as well as the ILS panel pushbuttons. The general form of a keyboard cycle command is:

CYCLE {FU} {STATE} {PASSWORD}

Here, "FU" will actually be one of the following

LOCALIZER

GLIDE SLOPE

INNER MARKER

MIDDLE MARKER

OUTER MARKER

ALL

to indicate which FU is to receive a transmitter cycle command. Note that it is possible to affect all transmitting FU's using a single cycle command.

The "STATE" entry might not be used by the operator at all. If it is, it will be either MAIN or OFF. Thus, we may ask that one (or all) FU's be cycled automatically until each has reached a certain transmitter state.

Regardless of which cycle command type one issues, a special password (which is not given in this report) must be the last entry appearing in the command.

Examples of possible cycle commands are

CYCLE LOCALIZER (password)

This will send a single cycle command to the localizer FU. This is equivalent to pressing the status panel cycle pushbutton once.

CYCLE INNER MARKER OFF (password)

Here the inner marker FU will be issued cycle commands until its transmitter state reaches "OFF." Under normal (error-free) conditions, no more than one cycle command should be issued in response to such a command.

CYCLE ALL MAIN (password)

All transmitting FU's will receive cycle commands until every transmitter state reaches MAIN.

Keyboard cycle commands will be honored only if the status frame is currently being displayed. As a cycle command is issued (regardless of its source), a blinking "CYCLE" will be placed at the far right of the status line of the affected FU. If a FU fails to respond to a cycle, an automatic log entry ("NO RESPONSE TO CYCLE") indicating such is made by the μ.p.

D. Error Reporting

The seven microprocessors in this system will detect and report a number of erroneous conditions that may arise during normal communications. All such reports will ultimately be routed to the ATCT μ.p. where they will likely be displayed on the TV monitor status frame.

Among all the errors that can be detected at the ATCT itself, the following portion will be reported at each and every occurrence. Observe that the final four are not actually erroneous conditions.

ATCT Software initialize

No response to cycle

ATCT error during self-test

Date before date change

New date

Time before time change

New time

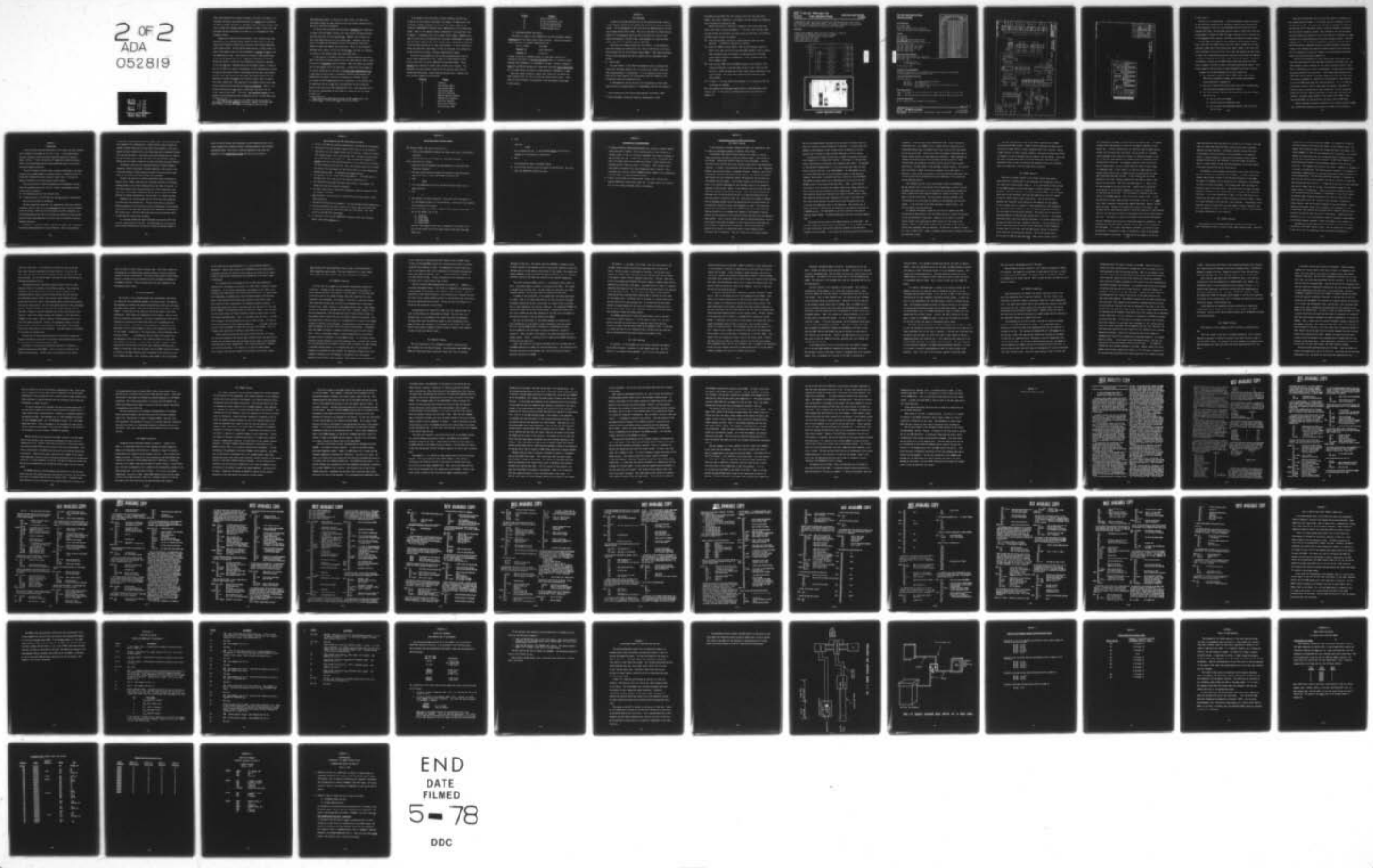
The remaining errors that may be detected at the ATCT end of each communications line are:

Priority	Message
0	System go
1	Receiver overrun
2	Framing error detected
3	Data error detected
4	Parity error detected
6	Wrong ID bits from FU
8	No data from field unit
9	No carrier from field unit
11	ATCT PROM checksum error

AD-A052 819 PURDUE UNIV LAFAYETTE IND SCHOOL OF ELECTRICAL ENGI--ETC F/G 17/2
MICROPROCESSOR-CONTROLLED COMMUNICATIONS IN AIR TERMINAL NAVIGA--ETC(U)
JAN 78 S E BELTER, C R WILLIAMS, S C BASS DOT-FA-74WA-3518

UNCLASSIFIED FAA-RD-78-25 NL

2 OF 2
ADA
052819



END
DATE
FILMED
5-78
DDC

Since each reported error causes an automatic log entry to be made, it is important to suppress the reporting process for chronic error conditions in order to prevent the ATCT μ.p. log memory space from being quickly filled with a single error message repeated hundreds of times.* Thus, error "debouncing" has been instituted at the ATCT μ.p. An explanation of this procedure follows.

Though error conditions may be detected in the system at any time, the ATCT μ.p. will consider placing reports of these errors in the log (and on the TV monitor screen) only at discrete time instants separated by 60 second periods. During each 60 second period, a running record is kept of the highest priority error encountered at each end of each of the six communication lines. (Thus there are 12 points in the system where errors can be detected by a μ.p.) If no error occurred at a line end during an interval, a priority zero ("system go") condition is assumed. If an error did occur, sometime during the 60 second interval, a count of one (1) is added to a counter corresponding to that communications line end. As long as this counter holds a value between one and three, the highest priority error seen during the most recent one minute interval will be reported in the ATCT log and on the TV monitor status frame. As soon as this count reaches four, the reporting of the highest priority error seen during the most recent 60 second period will be suppressed. That is, its further occurrence will not result in additional log entries (or TV screen reports) being made. Furthermore, ten continuous minutes of completely error-free operation must be observed at this line end before the

*The magnetic core reserved in the PDP-11/03 for log storage is sufficient for 800-900 automatic log entries. Manual log entries take approximately ten times the space of an automatic entry per line.

above-mentioned counter is returned to a zero value. Only when this "saturated" counter has been cleared to zero can further reports of this same error condition be accepted.

If during this ten minute waiting period a different error condition is found to be the highest priority error seen during a 60 second period, then this new error will be reported once. However, the 10 minutes of error-free operation will still be insisted upon. Meanwhile, this new error will become the "saturated" error for the purposes of deciding whether to make later reports and log entries. Thus, if the previously-saturated error returns, it will be reported once, and then will reassume its status as the saturated (and thus unreported) error.

If the counter is currently taking on a value between one and three, and if one minute of correct operation has just been observed, then a count of one will be subtracted from this counter. When the counter has reached zero in this manner, it will be held there. A check will then be made on the counter maintained for the other end of this same communication line. If they both are now at zero, a "system go" (priority zero) message will automatically be placed in the log to indicate the "recovery" from an error state of the FU communication line in question. If the counter for the remaining line end is not zero, then apparently an error condition exists at the other end of the communications line. This additional error must thus be "counted down" to zero before a "system go" will be issued for this F.U.*

*Note that error conditions involving the FFM communication line will appear only in the log (and not on the status frame).

Error numbers 8 and 9 are given a special status by the ATCT μ.p. As soon as an error counter "saturates" with either of these errors, the performance category displayed at the top of the status frame will be listed as "LINK BREAK" to indicate that a communication line is effectively broken. That is, for whatever reason, communication has ceased over the FU channel. In addition, the line of status values ("MAIN," "ABN/PE," etc.) shown for this FU will be replaced with the phrase "SYSTEM DOWN" to indicate that the status values are not to be trusted since a communication from this FU has not been received for at least three minutes. If the FU involved is either the localizer, glide slope, or FFM, all prealarms will be placed in an "ON" condition to further indicate communications trouble.

Errors are also detected by the six 4040 microprocessors at the FU ends of these communications line. These μ.p.'s send reports of their errors to the ATCT μ.p. to be "debounced" as described above. These <u>received</u> error reports will affect their own ATCT error counters. The additional six error counters required for this make up half of the 12 counters mentioned above. These remotely-reported errors, together with their priority numbers are given below:

Priority	Message
0	System go.
1	A/D overflow--DDM 3
2	A/D overflow--DDM 2
3	A/D overflow--DDM 1
4	A/DC error reported.
5	Receiver overrun reported
6	Framing error reported
7	Data error reported
8	Parity error reported
9	Field unit was reset

Priority	Message
11	Wrong ID bits at FU
12	RC data interruption noted
13	RC carrier reported lost
14	FU error during self-test
15	FU PROM checksum error

E. Performance Rating Time Delays

The ATCT μ.p. makes its own judgement of the ILS performance category independently of the remote control panel circuitry. Three of the status signals involved in a category determination are

CAT III DISABLE	(from FFM)
ABNORMAL P/E	(from localizer)
ABNORMAL P/E	(from G.S.)

A change in the CAT III DISABLE signal to a "bad news" state will be tolerated by the ATCT μ.p. for up to 70 seconds before a TV monitor screen downgrade from "CATEGORY 3" to "CATEGORY 2" is made. Further, a change in either of the ABN-P/E statuses to a "good news" condition must be maintained for 2 seconds before a TV monitor screen upgrade to "CATEGORY 3" is made.

Note that these "TV monitor screen" delay times will not affect the delays already built into the ILS status and remote control panels. Thus, there will be no effect on the operation of the performance rating lamps on these panels.

CHAPTER IX

THE COMPUTERS

At each of the seven locations of the tone-signalling system, there is a small computer controlling the sending and receiving of status and control signals. The microcomputers at the field units are manufactured by Pro-Log[1] and are based around Intel's 4040. The unit at the ATCT is a Digital Equipment[2] LSI-11 microcomputer which has the instruction set of one of their popular minicomputers, the PDP-11/40. When packaged in a case with power supply, the LSI-11 is sometimes called a PDP-11/03.

There were two programs written for this project. In the appendices where the programs are explained (D & E), they are referred to as the Snow White program (LSI-11) and the Dwarf program (4040). (The complete software system is then Snow-White and the Six Dwarfs, with our apologies to Walt Disney).

A. Intel's 4040

The Intel 4040 is a 4-bit PMOS microprocessor using a two-phase high level clock, two power supplies (+5, -10 volts), with a basic instruction time of approximately 11 microseconds. It is an improved version of the 4004 with 24 index registers, 60 instructions, interrupt capability, and a seven-level stack for subroutine calls.

As implemented on the Pro-Log card, the microcomputer includes 1280 eight-bit words of program memory on 5 C1702A PROM's, 80 four-bit nibbles of

[1] Pro-Log Corporation, 852 Airport Road, Monterey, California 93940

[2] Digital Equipment Corporation, Maynard, Massachusetts 01754

data memory on one 4002-1 RAM, and sixteen bits of TTL input and output (each). Additional information is included on the data sheets and schematics (next two pages and Figures 35 & 36).

Because this board is relatively inexpensive, the normal repair procedure would simply involve replacement. If the time, test equipment (DVM and good digital oscilloscope), and spare parts are available, the following procedure should isolate the problem:

(1) Check the two-phase, high-level, non-overlapping clock for proper operation. If defective, replace U12, the 4201.

(2) Remove all PROM's from the board. With the oscilloscope, check the 4040's 4 bus lines for activity and the 4289's address lines for square waves indicating that the processor is addressing each of the 4096 program memory locations in succession. If not, replace either the 4040 or 4289 or both.

(3) Place the test PROM (left with NAFEC personnel) into socket 0. This program periodically writes and reads to each of the 4 input and output ports. Check the operation of the strobes, input multiplexers, and output latches. The output ports should have low frequency square waves present.

(4) If steps 1 - 4 do not isolate the problem, it is a serious one and the card should be replaced.

Note the procedure outlined above applies only to a non-functional microcomputer card. A more general troubleshooting section was described in Chapter VII.

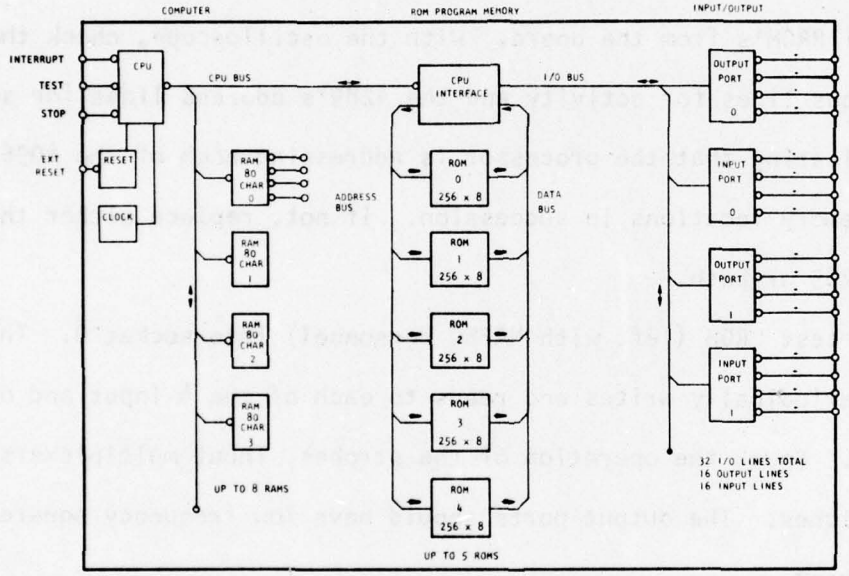
FIG. 35: PRO-LOG CPU CARD SPECIFICATIONS

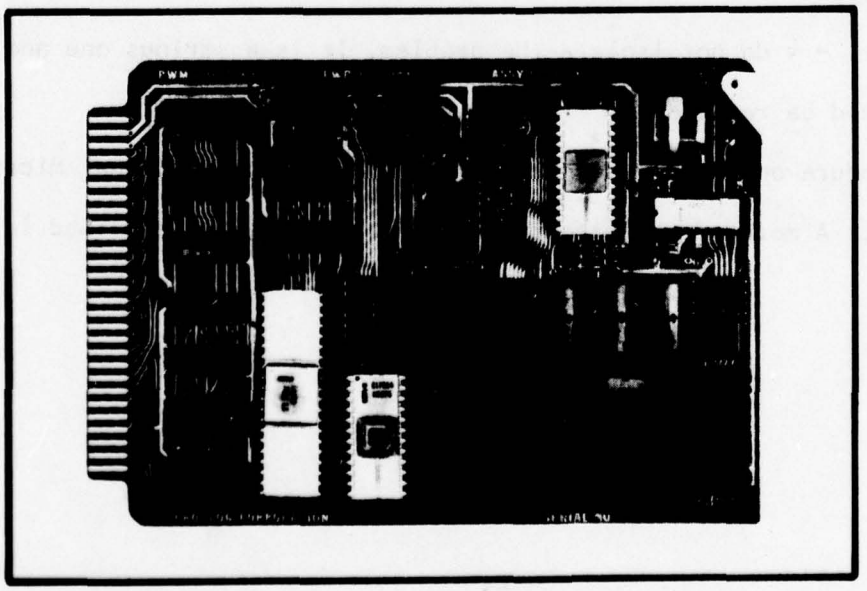
4040 EDGE CARD SYSTEMS
PLS-441A

A programmable logic system which implements the 4040 Microprocessor into a working system with CPU, ROM program memory, RAM register storage and I/O on a single card. The PLS-441 organization provides for reasonable program and I/O capacity to give the lowest cost approach to investigating PLS technology.

FEATURES

- Single card programmed logic system for protypes or production
- 1280 words of ROM program memory capacity (5 ROMs)
- 640 characters of RAM register storage (8 RAMs)
- Four output ports (16 lines)
- Four input ports (16 lines)
- One RAM output port (4 lines)





PLS-441 ONE-CARD SYSTEM

PLS-441 ONE-CARD SYSTEM SPECIFICATIONS

EDGE CONNECTOR PIN LIST					
PIN NUMBER				PIN NUMBER	
SIGNAL FLOW					SIGNAL FLOW
SIGNAL					SIGNAL
+5 VOLTS	IN	2	1	IN	+5 VOLTS
GROUND	IN	4	3	IN	GROUND
-10 VOLTS	IN	6	5	IN	-10 VOLTS
RO-8*	OUT	8	7	OUT	OUT 2-8*
RO-4*	OUT	10	9	OUT	OUT 2-2*
RO-2*	OUT	12	11	OUT	OUT 2-1*
RO-1*	OUT	14	13	OUT	OUT 2-4*
CLOCK ØIT	OUT	16	15	OUT	OUT 1-8*
TEST*	IN	18	17	OUT	OUT 1-2*
OUT 3-8*	OUT	20	19	OUT	OUT 1-1*
OUT 3-2*	OUT	22	21	OUT	OUT 1-4*
OUT 3-1*	OUT	24	23	OUT	OUT 0-8*
OUT 3-4*	OUT	26	25	OUT	OUT 0-2*
RST*	OUT	28	27	OUT	OUT 0-1*
EXT RESET*	IN	30	29	OUT	OUT 0-4*
IN 0-1*	IN	32	31	IN	IN 2-8*
IN 2-1*	IN	34	33	IN	IN 0-8*
IN 0-2*	IN	36	35	IN	IN 2-4*
IN 2-2*	IN	38	37	IN	IN 0-4*
IN 1-8*	IN	40	39	IN	STP*
IN 3-8*	IN	42	41	OUT	INTA*
IN 1-1*	IN	44	43	IN	INT*
IN 3-1*	IN	46	45		
IN 3-2*	IN	48	47		
IN 1-2*	IN	50	49		
IN 3-4*	IN	52	51	IN	NO
IN 1-4*	IN	54	53	OUT	STOP*
NC	IN	56	55	OUT	STPA*

Card Dimensions

4.5 inches high
6.5 inches long
0.48 inch maximum profile thickness
0.062 inch printed circuit board thickness

Includes

Card ejector
One 4040 CPU
One 4002 RAM and eight RAM sockets
~~One 1702A ROM~~ and five ROM sockets
Master power-on and external reset circuit
Crystal clock circuit
Four TTL output ports (16 lines)
Four TTL input ports (16 lines)
One MOS output port (4 lines)
CPU Test input (MOS)

4-BIT LOGIC PROCESSORS

Maximum Systems Capabilities

Eight 4002 RAMs (640 four bit characters)
Five 1302, 1602 or 1702 ROMs (1280 words of program memory)
20 output lines

16 TTL port lines
4 MOS RAM port lines

16 TTL input lines

Instruction Execution Capability

Capable of executing all of the 4040 CPU Instruction except for DCL and WPM
11.2 microseconds instruction execution time

Logic Levels Of External Connections

Low Level active

TTL Port: TTL compatibility and loading
MOS Input: TTL compatibility
MOS Output: Drive capability, one LPTTL or one TTL load with 12K pull-down to -VDD

Power Requirements

+VCC = +5 volts 5%@ 700 mA maximum fully loaded (30 mA per RAM, 35 mA per ROM)
GND = 0 volts
-VDD = -10 volts 5%@ 500 mA maximum fully loaded (30 mA per RAM, 35 mA per ROM)

Connector Requirements

56 pin, 28 position dual-readout on 0.125 centers

PRO-LOG CORPORATION 852 Airport Road • Monterev, California 93940
TWX: 910-360-7082
Phone: (408) 372-4593

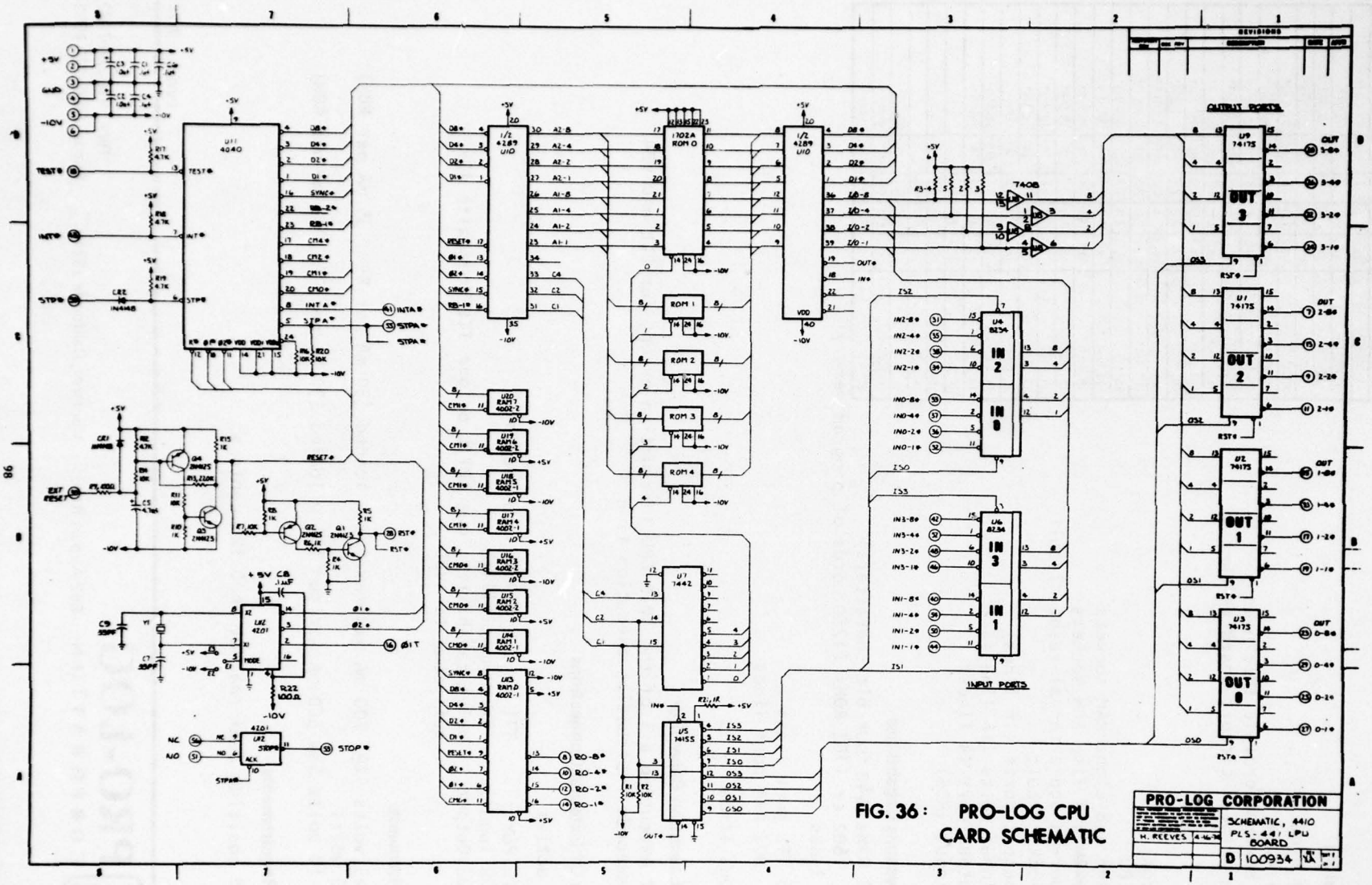
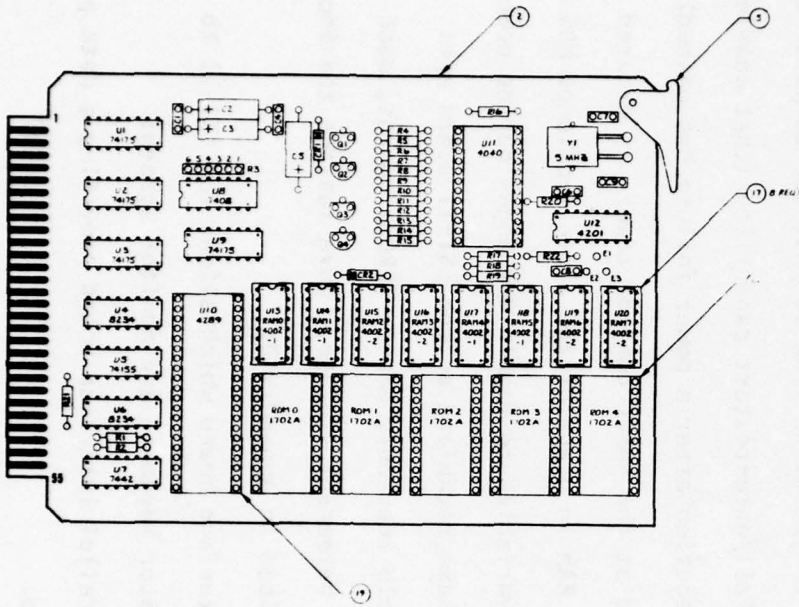


FIG. 36: PRO-LOG CPU CARD SCHEMATIC



ITEM	DESCRIPTION	REF. DESIGNATION
35	220K CC ¼W	R13
34	10K CC ¼W	R12,14,15,16,20
33	47K CC ¼W	R10,11,18,19
32	1K CC ¼W	R5,6,7,8,17,21
31	100Ω CC ¼W	R9, R22
30	22K NETWORK	R3
29	1N4148	CR1,CR2
28	2N4123	Q2,Q4
27	2N4123	Q1,Q3
25	33PF	C7, C9
24	0.1μF 50V	C1,C4,C6,C8
23	1.0μF 25V	C2,C3
22	4.7μF 20V	C5

PRO-LOG CORPORATION

ASSEMBLY, 4410

PLS-441 CPU BOARD

M. REEVES 4-16-75

D 100937 REV XA SHT 1 OF 1

FIG. 37 : PRO-LOG CPU CARD CHIP LAYOUT

B. DEC's LSI-11

The LSI-11 is a single board, 16-bit microcomputer capable of executing the PDP-11/40 instruction set (including floating point instructions). The microprocessor itself is a microprogrammed machine consisting of five N-channel MOS chips. The microcode stored on three of these chips allow the microprocessor to emulate the PDP-11/40 and to execute built-in routines for on-line debugging (ODT), operator interfacing, and bootstrap program loading.

On this same LSI-11 board is also available a 16-bit parallel input/output (I/O) bus (for communication with other boards plugged into the same backplane), 4096 words of MOS random access memory (RAM), a real-time clock input (ultimately tied to the 60 hz line so that the computer can keep track of time), and a power-fail/auto-restart capability (that enables the machine to continue program execution after a power failure has passed). The LSI-11 microcomputer installed at the NAFEC control tower was ordered with 4096 words of magnetic core RAM in place of the above mentioned MOS memory. This allows temporary data variables to remain unchanged during power failures.

Plugged into the same backplane as the LSI-11 board are:

1) two programmable read-only memory (PROM) boards, each holding 4096 words of permanent storage. Here resides the Snow-White program described in Appendix D;
2) one serial interface board which allows the LSI-11 to communicate with the operator keyboard and monitor screen;
3) one 16-bit parallel interface board providing a data path between the LSI-11 and:
    a) the six field unit modems,
    b) the ATCT cab cycle pushbuttons, and
    c) all ILS status and maintenance monitor lamps within the ATCT building.

When the PDP-11/03 HALT switch (on the front panel) is depressed, normal program execution ceases. In its place, the LSI-11 executes a built-in routine known as ODT. ODT prompts the operator for commands by displaying the character "@" on the monitor screen. Using the keyboard, the operator may now examine any memory location, load information into any core memory location, examine the contents of any of the eight LSI-11 registers, or begin exacution of a program beginning at any point throughout memory. Other ODT commands exist. All of them are documented in Section 7.3 of the "LSI-11, PDP-11/03 Processor Handbook" published by Digital Equipment Corporation. A copy of this handbook was supplied to NAFEC personnel by the contractor. ODT will necessarily be used, for example, to load and operate the nibble check sum routine described in Appendix M.

The LSI-11 microcomputer is a very sophisticated device whose repair should be undertaken by Digital Equipment Corporation servicemen only. The least expensive repair procedure for the serial interface, parallel interface, or PROM boards will likely be replacement. (A spare serial interface board was provided by the contractor.) The 64 PROM chips socketed on the PROM boards can be checked for program integrity by using the nibble check sum routine in Appendix M. Loading this routine manually, then executing it on all 64 PROM chips takes less than one man-hour. If it is found that the contents of one or more of these chips has changed, a new 3622 chip (512 by 4) may be ordered from an Intel distributor. He will program the memory chip with the contents prescribed on a paper tape provided by the customer. One paper tape, punched in the standard Intel "BNPF" format, was provided NAFEC (by the contractor) for each of the 64 PROM chips.

Digital Equipment Corporation schematics of all boards currently plugged into the PDP-11/03 backplane were passed on to NAFEC by the contractor.

CHAPTER X

CONCLUSIONS

At this writing, the system described in this report has been installed and in constant 24 hour/day use for over a year. It has demonstrated a practical solution to both the false signalling problem and lightning damage problem. In fact, the majority of suggestions concerning improvements to the system have centered around additional ways to use the computing and logging facilities.

The use of properly balanced lines, isolation transformers, and surge protectors has stopped damage to equipment caused by lightning strikes inducing high voltage and current pulses on signal lines. Of course, no amount of protection can protect against direct strikes.

There was one case of lightning damaging the microcomputer and associated ILS equipment during this period. There is considerable evidence in this case to indicate:

(1) The lightning entered on the 120 VAC lines.

(2) A surge arrestor on the 120 VAC lines was <u>not</u> properly installed and hence did not protect the equipment.

The tone signalling system with its comprehensive data error checking (redundancy, parity, CRC, etc.) has <u>eliminated</u> the false signalling problem in the old system. Where the old alarm system was either disabled or ignored because greater than 99% of the alarms were caused by noise, the new system virtually guarantees that an alarm is due to a condition requiring immediate attention.

Although it is much more complex than the old system, the microcomputer architecture demonstrated here is cost effective. This is due primarily

to the cost of the 100 plus buried and leased lines in the old system which are replaced by six balanced pairs. Cables buried at most airports are normally already being used with few spare wires available. The cost of copper and labor required to bury new cables have easily outpaced inflation over the past few years. Even worse, wires with DC continuity as required by the old system are no longer available from some telephone companies. (TELCO lines are usually leased for at least the middle and outer markers).

Maintenance costs are also decreased because of reduced damage by lightning. Despite the greatly increased complexity, the system is easy to maintain because of human engineering which allows testing and repair (down to the circuit card level) without test equipment.

Finally, the value of a reliable system here cannot be measured in dollars alone. Human lives depend on the proper operation of Instrument Landing Systems in all kinds of weather and at all times of the day. An ILS which produces false alarms distracts air traffic controllers from their job. An ILS whose transmitters may be cycling on and off randomly because of a nearby thunderstorm is failing when it is needed the most.

Suggestions for future systems can be divided into two categories: the field units and the ATCT unit. The use of an eight or twelve-bit CMOS microcomputer at the field units would eliminate most of the problems associated with the high power consumption and temperature limitations of the current units. The 4-bit PMOS units were the only practical choice in 1974 when this project was initiated.

In a complete system the Digital Equipment Corporation's PDP 11/03 is still a good choice for the ATCT. Two improvements at this location would include elimination of the control, status, and monitor panels in

favor of the CRT display and enhancements to the logging function to increase communication between personnel, decrease paperwork, and increase maintenance data. The current system was designed to help study the operation of the communication system, not the ILS as an entity.

Appendix A

How to Power-up the ATCT Tone-Signalling System

1. Plug in (115 VAC) the terminal electronics unit located behind the keyboard.
2. There are five slide switches located on the right end of the terminal electronics unit (as viewed from the rear of the rack). These are lettered "Sϕ" (Power switch), "S1", "S2", "S3" and "S4". All these switches should be in the "ON" position except "S1". These switches are arranged in a column on the end of the terminal electronics unit, with "Sϕ" at the bottom (of the column) and S4 at the top. Thus, all switches should be slid toward the front of the rack except for that switch which is in the "next-to-the-bottom" position (S1). It should be slid toward the rear.
3. The monitor (TV) unit itself should be plugged in. A switched socket is available for this on the end of the terminal electronics unit.
4. The "PULL-ON" switch toward the upper-right corner of the monitor (as viewed from the front) should be actuated.
5. The tone signalling unit (positioned immediately above the keyboard) should be plugged in (115 VAC).
6. All switches on the front panel of the PDP-11/03 should be placed in their lower positions.
7. The PDP-11/03 should now be plugged in. Its rear AC power switch should be on.
8. The tone signalling unit power switch (at its rear) should be placed "ON".
9. Assuming the terminal has had time to warm up, lift the "DC - ON - OFF" switch on the PDP-11/03 front panel.
10. If you wish to start up the "SNOW-WHITE" program, consult the "Software Reset" instructions, Appendix B.

Appendix B

How to do an ATCT "Software Reset"

The "Software Reset" (SR) can be carried out to:

1. Start up the SNOW-WHITE program from a dead start after a "disastrous" occurance.
2. Force a self-test of all modems for maintenance purposes.

A SR is carried out as follows:

1. Make sure the ATCT hardware has been powered up as described under "Power-up Procedure."
2. The most recent character typed on the terminal screen should have been a "@" sign. If not, type RETURN to get this "@."
3. Type

   46570;

4. Lift the ENABLE-HALT switch (on the PDP-11/03 front panel) to its upper position.
5. Type

   G

6. The computer will begin execution. During the first five seconds, or so, the modem self-tests will be carried out. During this time, nothing will be placed on the screen.
7. After this delay, the "status" frame will be written on the screen. If any of the modems tied to the

   a. localizer
   b. glide slope
   c. inner marker
   d. middle marker
   e. outer marker

   balanced lines fails the self test, a message will be placed at the left on the second line of the status display associated with that field unit.

8. Type

L

then the

RETURN

key to examine the log. If the FFM modem failed its self test, a message will be placed here indicating so.

9. Type

S

to re-display the status information frame.

10. Lift the "LTC" switch on the front panel of the PDP-11/03. This will start the SNOW-WHITE system time clock.

Appendix C

The Meaning of a Watchdog Reset

1. A 1/2 hertz oscillator, located on the interface card, provides a computer "Reset" signal once every 2 seconds. This resetting action is very similar to a "software reset" in that it "re-starts" SNOW-WHITE from scratch (although it does not erase the log). To avoid this resetting action, the computer must send a re-trigger pulse to a one-shot multivibrator (also located on the interface card) at least once every 1 1/2 seconds. (The retrigger pulse itself takes 1/2 second of the total 2 second oscillator interval.) Thus, a watchdog reset can occur should SNOWWHITE become trapped in an unintentional "infinite loop", or otherwise become confused.
2. A drop in the 115 VAC line voltage causes a "power fail" routine to be executed immediately before total power loss. At power return, this routine will, in turn, cause a watchdog reset to be executed.

Appendix D

Detailed Description of PDP-11/03 Software

The "MAIN" Programs

In the following, the phrase "sequence of code" will sometimes be used to refer to a contiguous sequence of computer instructions.

The complete "Snow White" Program consists of a number of routines. These routines may be broken up into two different types: sub-routines and co-routines. Subroutines should be familiar to any programmer. They may be called by any calling routine that desires to make use of that subroutine's functions. In theory, a co-routine may also be called by any other subprogram. However, the calling procedure is somewhat different. Normally, when calling a subroutine, one calls the routine at the start of its function and expects that function to run to completion before the subroutine returns control to the original calling program. A co-routine, however, may be called at any point in its function designated by the programmer and will not necessarily complete its entire task. Rather, it may execute a portion of that task and then return control to the calling program. Later, the calling program may once again call up the co-routine and ask it to continue executing at the very point where it previously left off. There may be any number of computer control exchanges between the calling program and the co-routine before the co-routine may finally be said to have completed its task. The instruction used to call a co-routine is "JSR PC, @(SP)+". The action of this rather odd looking instruction is to exchange the contents of the computer's program counter with the contents of the location currently pointed to by the computer's stack pointer. Thus, to be sure that we reach our desired destination, we must first place on the stack the address of the co-routine entry point where we wish to begin or resume co-routine execution. The co-routine may now execute for as long as it wishes after which it should execute another "co-routine call" instruction. This will return us to the calling program.

The calling program should now remove from the stack the address deposited there as a result of the co-routine call instruction. It should place this address in safe keeping in order to make use of the address at the next occasion when it wishes to return control once again to the co-routine.

The Snow White Program requires that a number of processes be serviced very frequently in a "round robin" fashion. Furthermore, it requires that some of these tasks be called much more frequently than others. The backbone of the Snow White Program, which is to say the program that does the round robin co-routine calling, is the "TASK MASTER." The TASK MASTER will call in a prescribed order each of the co-routines present in the Snow White Program. When control is returned from the co-routine to the TASK MASTER, the address at which control was relinquished will be stored by the TASK MASTER in one of five arrays called LEV1,LEV2,...,LEV5. Later when the co-routine is to be re-called, the address where we may resume execution of the co-routine may thus be found in one of these arrays. Those routines that should be called most frequently are stored in the LEV1 array. Those that should be called next most frequently are stored in the LEV2 array. And finally those routines that need service least frequently have their co-routine call addresses stored in the LEV5 array. The calling order of these subroutines is similar but not identical to the nesting of DO loops in a FORTRAN routine. The TASK MASTER is the first routine in the group of routines labeled "MAIN." The TASK MASTER begins with the instruction labeled TASK.

The second routine listed in the "MAIN" grouping is called SOFTX. This routine is used only during unusual circumstances. It is a software initialize or reset routine that sets certain sensitive variables in the Snow White Program to initial values. It is used as an entry point during the installation

procedure. It may be also called automatically when certain "disastrous" conditions arise. For example, due to a power failure or possibly an undiscovered software bug, the computer may "trap" to one of the trap vectors at the low end of core. Here it will pick up an address indicating where it should transfer control to under these trap conditions. These addresses are loaded with the entry point of this software initialize routine. After initializing a few variables, the SOFTX routine then calls the so-called DOG routine. The DOG routine is the watchdog interrupt routine. This routine also initializes a number of sensitive variables used by Snow White. In addition, it sets up the trap vectors in low core previously mentioned. Also it sets up the initial co-routine entry point addresses in the various LEV arrays mentioned earlier.

This watchdog routine is also called under unusual circumstances. On the interface card in the printed circuit board drawer, one will find the watchdog reset relay which if it is not reset by the Snow White Program will after a second and a half cause the computer's transfer of control to be made to the entry point DOG. Here the various variables required to be initialized are so initialized, the modems are reprogrammed, and also self-tested. If the self-test of any modem discovers a modem failure, this will be noted in the Snow White log.

An important difference, though not the only one, between the SOFTX routine and the DOG routine is that the SOFTX routine will cause the clearing of the log maintained by Snow White. However, an entry point to the DOG routine without first executing the SOFTX routine will not cause a log erasure. Rather, it will cause a single entry to be made to the log indicating that a watchdog reset was executed. No such entry is made in the case of a call to SOFTX first. Rather, a message indicating that a software initialize was performed is made.

The two final routines listed in the "MAIN" grouping are the PWRDN routine and the PWRUP routine. These two small subprograms are made use of during an AC power failure or power resumption after such a failure. During a power failure, and after a power resumption, the action of these two routines in tandem is to cause a program jump to be made to the DOG routine. Thus, when one sees a watchdog reset entry in the Snow White log, there exists the possibility that this was not caused by the watchdog relay on the interface card, but rather that a 110 volt power failure occurred.

The "MODEM" Routines

There are six modems located in the printed circuit board drawer. These handle all transmissions to and receptions from the Dwarves; that is the field units located around runway 13. It is the function of the routines in the MODEM grouping to drive these modems to handle receptions or transmissions. The first routine is a co-routine entitled MODRCV. This co-routine takes care of any receptions that have been made from a field unit (dwarf) to the air traffic control tower (Snow White). This co-routine first examines the "interrupt" lines coming into the computer from the modems. If any of these lines is a logical "1" after preprocessing, then apparently the particular modem assigned to that line requires some sort of service. Only two types of service can be requested. The normal type is the service requested whenever a frame has been received from a field unit. The second type, which occurs under only abnormal conditions, is a service requested with the loss of carrier from the respective field unit. This co-routine will check to determine which of the two service conditions is being requested. In either case it will first read the modem status register to determine if any error has occurred during reception. The errors possible are a loss of carrier (DCD) as mentioned above. Other errors include a parity

error detected by the modem, a framing error, or an overrun error. If anyone of these errors has occurred, an entry will be made by Snow White in the program log. If none of these errors has occurred, then the modem receive register will be read by this co-routine. The co-routine will increment a count assigned to this field unit receiver to determine whether or not this is the last frame expected in a complete block of information to be received from a field unit. For example, receptions from the far field monitor should consist not of a single frame but rather of eight frames in a complete block. If this frame is not the last frame expected in a complete block, then an exit is made from this co-routine after all such modem interrupts have been serviced. If this frame is the last frame expected in this block, further processing of the block of frames is now made by this co-routine. The first such check made is a check on the CRC code attached to the end of the block. A high level of confidence may be gained in the successful reception of a block of information if this CRC code can be verified. This verification is handled by the last routine in the MODEM grouping. The title of this routine is CRC. If the CRC code fails to check, then an entry will be made in the log. If it does check, then a successful reception has probably been made. However, further checks are made. The next is a check of the ID bits in this block. The localizer has an identification code of 1, the glide slope has an identification code of 2, all the way up to the far field monitor which has an identification code of 6. The computer is aware of which modem it has received information from and, therefore, which field unit should have made the transmission. The ID affixed to this block should, therefore, be the same as the ID associated with the modem. If it is not, then possibly crosstalk or crossing of wires has occurred. A successful ID check is followed by a check of the ACK/NAK bits also placed in this block. If these bits do not appear in one of two

prescribed patterns, then once again an error entry will be made in the log. After all these checks have been made, we may be sure that a successful reception has been completed. Information received from the field unit is now placed in various locations in the core memory of the computer. There is a special sequence of code used by the markers, another special sequence of code used by the localizer or the Glide Slope, and thirdly, there is a final section of code reserved exclusively for the distribution of the data received from the far field monitor.

The MODTRN co-routine handles transmissions to the various field units. That is, it drives the transmitters of the modems. It places transmission requests made by other routines in the Snow White Program in a job queue. This queue is unloaded by the co-routine, and its information is distributed into one of six other job queues. One of these lower level job queues is assigned to each of the field units. These lower level job queues are then examined to determine if any transmissions are to be made to the respective field unit. If so, that transmission is made. All the field units are to receive from the control tower a block consisting of only four frames. There is a single exception to this and that is the localizer. Transmissions to the localizer consist of blocks that are eight frames in length. Therefore, the end of the MODTRN co-routine is taken up with a small stretch of code reserved for making transmissions to the localizer.

The "STATUS" Routines

The routines in this grouping handle the display of the various ILS status information on the air traffic control tower terminal screen. The first

routine here is a co-routine entitled STACH1. Its function is to display the status values of the primary ILS signals on the terminal screen. For example, the transmitter state of any transmitting field unit whether it be "MAIN," "STANDBY," or "OFF" will be displayed by this co-routine. As a further example, the conditions of the various abnormal power or abnormal monitor statuses are also displayed by this routine. An additional function is the determination and display of the performance category of the ILS at any given time. This category can be one of four values. Category III, Category II, Link break, or System down. The determination of Category III or Category II performance is done identically to the description made in Texas Instruments ILS System Documentation. The Link break performance category is displayed at the top of the terminal screen whenever the loss of a carrier or no data has been received from any field unit. If this loss of carrier or no data condition persists for longer than four minutes, the performance category is down graded further to "System down." There are two built-in delays associated with the determination of a Category III performance. First of all, should the abnormal power condition associated with either the localizer or the glide slope switch to an "OK" condition after previously indicating an alarm situation, then the Snow White Program will wait two seconds before upgrading performance rating from a Category II to a Category III value. A second delay type concerns the so-called "CAT III Disable" condition transmitted to the control tower from the far field monitor. Should this alarm condition turn on, it will be allowed to remain on for seventy seconds before a performance category downgrade is made from Category III to Category II.

The STATLG co-routine takes care of the display of the two most recent log entries on the tower terminal screen. The most recent entry is placed to the left on the line immediately below the statuses displayed for a

particular field unit. To the right on this same line will be the next most recent log entry associated with that field unit. With this next most recent log entry will also be displayed the date and time at which the entry was made. All the entries made in the past for any field unit may be observed by examining the log.

The display of the 41 maintenance monitor signals as well as other statuses in the ILS is handled by the STACH2 co-routine. The information displayed on the terminal screen by this co-routine includes all 20 maintenance monitor signals currently being received from the localizer, all maintenance monitor signals (17) currently being received from the glide slope field unit, and all 4 maintenance monitor signals being received from the far field monitor. The far field monitor portion of this maintenance monitor terminal display will also show five relayed statuses that the Snow White Program is currently passing from the far field monitor to the localizer. These five relayed statuses are "Category II shut down alert", "Category II shut down", "Monitor mismatch", "Power/temperature fail", and "Far field monitor bypassed". Finally, the maintenance monitor far field monitor display shows the current values of the DDM signals being relayed from the far field monitor to the localizer. These DDM signals are displayed on the screen in microamp values.

The date and the time being maintained by the Snow White Program are displayed prominently at the top of the status frame mentioned above as well as at the top of the maintenance monitor frame.

The display of the DDM values mentioned previously is handled by a separate co-routine entitled DDMDSP. This is the last co-routine in the STATUS routine grouping. The DDM values are transmitted to the control

tower as eight bit binary values including sign. These binary numbers are representations of voltage values sensed by analog to digital converters located at the far field monitor. The control tower converts these voltage values to microamp values by performing various arithematic operations in the DDMDSP co-routine. These microamp values are then displayed on the maintenance monitor frame.

The "MISCIO" Routines

The routines in this grouping handle such miscellaneous input/output as reading the cycle pushbuttons located in the control tower, retriggering the watchdog, and sending the received status values to the local ILS status and remote control panels. The first co-routine in this grouping is entitled PSHBUT. It takes care of the reading of the current status of the cycle pushbuttons. These states are "debounced" by the software. This is to say that any pushbutton value whether it be open or depressed must exist for six consecutive co-routine calls before that value will be accepted by the Snow White software. As soon as a cycle pushbutton is recognized to be depressed, a pair of cycle commands is sent to the appropriate field unit via the modems. In addition, if the status frame is being displayed, then a blinking "CYCLE" is displayed at the far right of the status line corresponding to that field unit. This blinking "CYCLE" will exist for approximately five seconds. If during that five second interval a return transmission from the field unit has not been received indicating that a change in transmitter state has been made, then the PSHBUT routine will cause a log entry to be made indicating that no response by the field unit to the cycle command was made. This, of course, might happen if the ILS equipment

at the field unit were powered down or if it were otherwise operating improperly. Special code written into the MODTRN co-routine (described in a previous section) will cause the cycle relay at the field unit to remain closed as long as the corresponding cycle pushbutton in the control tower is depressed.

It is possible due to equipment failure or some other disastrous condition for the computer to be caught in a tight loop or otherwise perform improperly. If this occurs, then certainly the modems are not likely to be serviced, and communication to the field units will be lost. To guard against such conditions, a watchdog reset relay has been mounted on the interface card in the printed circuit board drawer located at the control tower. This relay will wait a second and a half between successive closures. If during that second and a half a reset pulse has not been sent from the Snow White computer to the watchdog relay, the relay will engage. The occurrence of this relay closure will cause the control tower to make an entry to the DOG routine mentioned above. That is, a watchdog reset will occur. An entry will be made in the log indicating that such a reset was made. In order to avoid the occurrence of this resetting operation, a retrigger pulse must be sent from the computer at least every one and a half seconds. This retrigger pulse generation is handled by the HITDOG co-routine.

Status values received from the various field units must, of course, not only be displayed on the control tower terminal screen, but must also be passed onto the status panel and the remote control panels in the equipment room and control tower, respectively. This relaying operation is handled by the TOILS co-routine. Its function is to pass the status values on to various relay drivers located in the printed circuit board drawer.

These drivers will cause associated relays to open or close according to their respective status values. The relay closures will, in turn, cause status panel and remote control panel light indicators to illuminate.

The "TRMOUT" Co-Routine

All text that is to appear on the terminal screen passes through this co-routine. A request to this routine to transfer text to the screen is made by making two 16-bit word entries in its job queue. The low order byte of the first word in the queue indicates under what conditions this text is actually to be displayed. For example, if the text is simply the current clock time, then we would wish the text to be displayed only if the screen currently held either a status or a maintenance monitor frame. Other frames that may be displayed do not show this clock time. Therefore, should a clock entry be encountered in the job queue when one of these alternate frames is being shown, then we would like the clock display request to be ignored. The high order byte of the first word in a job queue entry is a count of the number of bytes in the text to be displayed. The second word in the job queue entry holds the address of the first byte of the text to be displayed. Information which the TRMOUT co-routine is capable of displaying can be in one of three forms. First, a plain ASCII format is readily accepted by this co-routine. The second form of information is called MOD50 format. In this format, three characters may be contained in one 16-bit computer word. In normal ASCII format, only two characters may be stored per word. In order to achieve this 50% increase in character packing density, it is required that only a limited number of characters be available for such packing. The characters allowed are all the alphabetic characters, all the numerals (0 through 9), and a few special characters such as dollar sign and period. Text passed to this co-routine can

be of a mixed form containing both ASCII format as well as MOD50 format. In order to inform the TRMOUT routine of a switch from one format to another, the dollar sign character is used. The appearance of a dollar sign character in the midst of text to be transmitted to the terminal indicates the dollar sign should be ignored, and a switch from ASCII to MOD50 or vice versa should be made. Further characters will be interpreted in the new format until a second dollar sign is found.

The third format in which information may be accepted by TRMOUT is a so-called "log-packed" format. This format is triggered by the appearance of a right bracket in the stream of text passing to this co-routine. Upon encountering a right bracket character, this co-routine will interpret the next four bytes to be a special packed version of a current date, time, and source code to be associated with this data. The use for this third format is in displaying the system log on the screen.

The appearance of an "underline" symbol (_) will cause the next two bytes in the stream to be interpreted as follows. The first byte is a character that is to be passed to the terminal not once but repeatedly. The second byte after the underline symbol is to be interpreted as the number of times this character is to be placed on the screen. This repeat feature can be used for example to send a stream of spaces to the terminal in order to move the terminal cursor.

The "SERVICE" Routines

The four subroutines in this package are commonly called routines used throughout the Snow White Programs. The subroutines named PSHREG and POPREG are subroutines used to save and "unsave" the first six computer

registers on the stack. The former subroutine (PSHREG) is commonly called by other routines in the program in order to save these registers so that the registers may in turn be used by that portion of the program. The latter subroutine (POPREG) is used to perform the reverse operation, that is, registers previously stored on the stack are returned to the registers themselves.

The third subroutine BINDEC converts a non-negative binary number to ASCII. The binary number is placed in a variable called TODEC. This number is assumed to be less than 100. Two ASCII bytes are returned to this same variable location. The low order byte will contain the tens digit.

A number of the co-routines in this program have job queues associated with them. That is, any tasks that they are to carry out may be indicated by another routine placing one or more words in a queue of jobs to be done that is later examined by the routine in question. For example,the TRMOUT co-routine will place text on the terminal screen. Indications of where this text lies in memory and how many characters of the text are to be passed to the terminal are made in the job queue called JQTRMO.

The last two subroutines in the SERVICE package will automatically place information on a job queue or remove information from a job queue. The first routine, PUTQUE, places information on a designated job queue. The first address of the job queue to receive information is placed in register 5. Then, the words to be placed on the queue are pushed onto the computer stack. Finally, a subroutine call is made and the PUTQUE subroutine performs the job queue entry.

Later, the routine or co-routine associated with this job queue may use the TAKQUE subroutine to remove the next job queue entry. This subroutine will place the next entry on the computer stack, thus performing the reverse operation carried out by PUTQUE.

The form of a job queue is as follows: The first four words of the job queue contain counters and pointers associated with the words that follow. The first word in a job queue is two bytes. The low order byte contains an integer number indicating how many computer words must be placed in the queue in order to form a complete single entry. The high order byte of this first queue word indicates the current number of complete queue entries remaining to be serviced in the queue. The second word in a queue is a pointer indicating the location of the first word of the next queue entry to be serviced by the associated routine. The third word in the queue is also a pointer indicating the next available queue location open to receive the first word of a new queue entry. The fourth word in the queue is a pointer pointing to the last word in this queue storage area. Any further entries that would normally be placed beyond this last word will instead be placed back up at the top of the queue storage area (immediately after the initial four words) thus producing a queue that "wraps around".

The TAKQUE subroutine will upon exit cause either a zero or non-zero value to be placed in register 5. If this value is a zero, it is an indication to the calling routine that the queue was found to be empty and therefore no queue words have been placed on the computer stack. If register 5 is rather non-zero, it is an indication that a further entry was found and that its associated words have been placed on the computer stack.

The "TIME" Routines

The routines in this package carry out various functions that enable the Snow White Program to maintain its own calendar and time. The first routine is a co-routine entitled NEWTIM. It will or will not perform its

function according as the variable NEWSW is non-zero or zero, respectively. If this variable is non-zero, an immediate exit to the task master calling routine will be made. If this variable is rather non-zero, then this co-routine will examine the permanent second counter stored in location PSECND and convert this second count into an ASCII version for display by the terminal. Thus, each time the permanent second count is caused to advance by one, the action of this co-routine will have to be triggered. How this is done and by whom will be discussed shortly.

In order to perform the ASCII conversion, a special subroutine is made use of. This is the second routine in this package entitled HHMMSS. Data input to this second subroutine is in the form of an 18-bit second count that is stored in a two-word entry specified by the calling routine. The least significant 16 bits should previously have been placed in Register 4. The most significant two bits (that appear as the low order of a second word) should have been placed previously on the computer stack. The ASCII version of the resulting time is placed in a permanent ASCII storage area called PTIMAS. In addition, this routine will also convert the second count into a special log-packed version of the current time. This version is always stored in a variable PTIMLG. This log-packed version consists again of 18 bits. These bits are treated as three 6-bit fields that are concatenated. The low order 6 bits are the binary representation of the hour number. The middle 6 bits are treated as a binary count of the minute number. Finally, the upper 6 bits are taken as a binary count of the current second number. This log-packed time is part of the information that is tagged at the beginning of each log entry whether that entry be made automatically by the Snow White Program or whether that entry be a manually entered one.

Additional information that is stored at the beginning of each log entry includes the date at which the entry was made. Nine bits are required to store a log-packed date. The low order four bits are a binary count of the month number. The upper 5 bits are a binary count of the corresponding day number. Two routines in this package that treat this log-packed date will be discussed shortly.

The third routine in this grouping is called CLKINT. This routine is automatically called whenever a line time clock interrupt occurs. These interrupts appear 60 times each second in synchronization with the 110 volt line voltage. Thus, at each interrupt call, this subroutine must increment a count and take action at each 60th call. The action is first to add one to the permanent second counter held in PSECND. Then, the NEWSW switch is set to a non-zero value so that later when the TASK MASTER calls the NEWTIM subroutine, this new second count will be converted to an ASCII as well as a log-packed version. Next, this routine will determine whether or not a full 24 hours worth of seconds have passed. If not, an immediate return from interrupt is made. If 24 hours have passed, then the permanent second count is set to zero corresponding to midnight. Next, the 14 system time clocks used throughout the Snow White Program are decremented by 24 hours so that they may remain active even though they may be required to count beyond a full 24 hours into the next day. Finally, the INCSW switch is set to a non-zero value so that the INCDAT co-routine, described next, will advance the calendar date by one day.

If the INCSW switch is set to a non-zero value, the INCDAT co-routine will at the next TASK MASTER call advance the system calendar by one day The permanent version of this date is kept in log-packed form at the location PDATLG. Also, a permanent ASCII version of this date is maintained in

location PDATAS. This permanent storage area requires six bytes of computer memory. Given the log-packed version of the date, the MMDD subroutine converts this date to an ASCII version and stores it in this permanent location. This routine will be described shortly. The last operation carried out by the INCDAT routine is to display this ASCII date at the top of either the status or maintenance monitor frames. Thus, an entry is made into the TRMOUT job queue.

If a nine-bit log-packed date is placed in the location DATINN, then the NEWDAT co-routine will cause this date to become the new date to be used throughout the Snow White Program. First, an entry is made in the system log indicating the old date immediately preceding the date change. A second log entry is made immediately after the date change in order to indicate the new date. The difference between the dates of these two entries is an indication of how many days difference between the two dates have passed. The NEWDAT co-routine makes a call to the MMDD subroutine in order to convert this log-packed date into an ASCII version. Finally, the ASCII version of this new date is passed to the TRMOUT job queue in order that this date may be placed at the top of either the status or maintenance monitor frame.

The TIMASC subroutine converts an 18-bit log-packed time to ASCII. In order to call this subroutine into action, the calling routine must first place the 16 least significant bits of this time in register 3. The two most significant bits are held in the low order of a 16-bit word that is then placed in register 4 immediately before the subroutine call. This subroutine call makes use of the routine BINDEC described in the SERVICE routine grouping. The last subroutine in this particular package is the MMDD routine mentioned previously. Its job is to convert a nine-bit log-packed date to ASCII for later display on the terminal. Prior to a call to this routine, register 4 should be loaded

with the nine-bit log-packed version of the date.

Dates handled by various routines in this subroutine package are checked for validity. The number of allowed days in each month of the year is stored in a short array called DAYMON. The number of days in the month of February is allowed to be no larger than 28. Thus, at each leap year the date will be in error for one day.

The "REPORT" Co-Routines

Each communication link between the control tower and a dwarf (field unit) has associated with it two error families. The first family is an indication of the error status at the control tower end of the communication link. The second family keeps track of the current error status at the field unit end of the line. Errors occurring at the control tower line end are discovered by the Snow White Program itself. However, errors pertinent to the field unit line end are discovered by a dwarf program and are reported to the Snow White Program by a communication link.

The occurrence of an error at either line end will cause an automatic log entry to be made describing this error. It is conceivable that a chronic error condition would cause repeated entries that are identical to be made to the log. The log may hold between 900 and 1,000 automatic log entries. So, it would not take too long before the log would be completely filled with this repeating error. Therefore, errors are "debounced" using the following scheme. At the end of each 60-second interval, the REPORT co-routine examines the highest priority error occurring at each line end. If an error has occurred, this will cause an incrementing of a counter associated with that line end by one. Until this count reaches a total of three, each

successive error will cause a log entry to be made. Beyond the count of three, no further log entries will be made until ten full minutes of error-free operation at that line end has occurred. That is, the counter is set to a full count of ten and decremented by one at each minute during which no error occurred. Should an error occur during this ten-minute countdown, this count will be again saturated at a full count of ten, thus postponing the ten-minute countdown time interval. It is possible that during this countdown, a distinct error may occur of a higher priority. Should this happen, an automatic log entry will be made immediately. However, the error count will be saturated at a full value of ten just as if the older previous error had simply been repeated. The REPORT co-routine will then begin counting down this new error. Even though a full ten-minute countdown may have occurred at any particular line end, this does not necessarily imply that a full error recovery has been made. Since there are two line ends for each communication link, two counters must be cleared to zero before this error recovery can be considered to have taken place. An error recovery is indicated by a "SYSTEM GO" automatic log entry being made. If this entry is not made at the end of the 10-minute error-free interval, it must imply that a countdown is still taking place on the error counter associated with the opposite line end. Only when both line ends have been cleared of an error state will this "SYSTEM GO" error recovery message be displayed. Even errors occurring at the initial stages of a repeated error condition are treated with a countdown procedure. For example, should a single error occur during a 60-second interval, and then the immediately following 60-second interval be error-free, an immediate "SYSTEM GO" error recovery message will be made. However, should this error repeat itself during a second time interval, a full two-minute countdown will be insisted upon by the Snow White Program before the error recovery message

is made. Three errors occurring in three successive 60-second time intervals will require three full minutes of error-free countdown before a "SYSTEM GO" message is placed in the log. Beyond this count of three, the previously mentioned countdown from a full saturated value of ten is insisted upon.

Note from the usage above that the "SYSTEM GO" entry is not the only indication of error free operation of a communication link. Rather, the "SYSTEM GO" entry is to be considered an error recovery indicator. It is possible that there are no log entries currently existing pertaining to a particular communication link: not even a "SYSTEM GO" log entry. Though the comforting "SYSTEM GO" message does not appear in the log, this condition is the best of all possible ones since it indicates the detection of no errors during the memory of the software log.

Since there are six communication links serviced by the Snow White Program, there will be twelve error counters to be treated by the REPORT co-routine. Entries to the log are made by making use of the AUTOLG subroutine to be described below.

The "LOGSRV" Routines

The routines in this grouping are used to service or manipulate the log.

The first routine in the log is the AUTOLG subroutine. This is called whenever a log entry is desired to be made by any other section of code in the Snow White Program. Its purpose is to bring together as a stream of bits the log-packed date, time, and source code to be associated with this log entry.

A log entry requires four pieces of information. These are packed together as a single stream of bits that is stored in a sequence of four bytes. The first byte of a log entry will always be the right bracket character (135 code). The four bytes immediately following this list the date, time, and source code associated with this log entry. The low order nine bits are the log-packed date, the next 18 bits are the log-packed time, while the three bits immediately following this are the source code indicating to which communication line the error is to be associated. This basic information, together with the error number, is packed together and placed on the job queue of the AUTOLX co-routine. At the next call of this co-routine by the TASK MASTER, this routine will acutally perform the log entry. It will insert the "135" separator code, then the date, time, and source code packed binary data, followed by a sixth byte which will be a pointer to the MESADR or the TESADR tables. An entry in either of these two tables is itself a pointer referring to the start and the number of characters in a stream of ASCII data that actually represents the English representation of this error message. Thus, a total of only six bytes of computer storage is required for an automatic log entry. This since the actual ASCII text is not repeated for each occurrence of the error in the log, rather only a single-byte pointer is used. Manual log entries made from the terminal keyboard also have the same "prefix" data associated with them. However, the sixth byte which would ordinarily hold the automatic error message pointer instead holds the first character of the manual entry. These manual entry characters are taken byte by byte until the next right bracket (135 code) character appears.

The source codes mentioned above are as follows. Code one through six indicate the communication link to a field unit. One stands for the localizer communication link, two stands for the glide slope communication link, ... ,

while six refers to the far field monitor communication links. Source code seven is reserved for errors that refer not necessarily to any particular communication link but rather to the air traffic control tower program itself. Source code zero is used to indicate that the information that follows was originated by a manual log entry.

Of the 31 possible error messages that may be placed automatically in the log, four of these are of such a nature one would not expect them to repeat during normal system operation. Therefore, a countdown procedure is not applied to them. These messages are (1) "ATCT SOFTWARE INITIALIZE" (2) ATCT WATCHDOG RESET" (3) "NO RESPONSE TO CYCLE" (4) "ATCT ERROR DURING SELF TEST". Since a countdown is not initiated for these errors, the user should not be alarmed if one such error appears in the log and is not followed by an error recovery ("SYSTEM GO") message.

AUTOLX routine not only performs the actual log entry, but also keeps track of the two most recent log entries associated with each field unit communication link. Pointers referring to the start of the two most recent log entries for each field unit are stored in the array called RECLOG. This array is maintained by the AUTOLX routine as well as the RECLOG routine described below. These two most recent log entries will appear on the status frame immediately below the status information associated with the corresponding field unit. A single exception is the far field monitor whose most recent log entries can be observed only by displaying the most recent lines of the log itself.

The RUBLOG routine is called whenever the insertion of a new log entry would write over the oldest log entry now existing in the log. The log storage area starts in location 5204 and ends in location 17777. Information that would ordinarily like to be stored in any location beyond this last one, will

be "wrapped around" back to location 5204. Once a "wrap around" occurs, a possibility for writing on top of a much older log entry exists. These older log entries if they need to be deleted are deleted completely by the RUBLOG subroutine. This routine also deletes a log entry from the RECLOG array should this deleted entry happen to be one of the "most recent" two entries corresponding to any field unit.

The next subroutine in this package, entitled ONELIN, will advance a given log pointer ahead by one log line or one full log entry whichever comes first. The pointer is stored in location ONEPTR. Upon exit from this subroutine, this pointer will point to a line feed character (indicating the end of the line) or a right bracket character (indicating the end of a log entry). Thus, this routine may be used to advance line by line through the log.

The "PROMCK" Co-Routine

Though the entire Snow White Program is stored in fusable link PROM, it is conceivable that bits of this storage area might change with time or with integrated circuit failure. Therefore, a check is constantly being made of the integrity of this program store area. This is done by the PROMCK co-routine. At each call of this co-routine by the TASK MASTER, this routine will total (in a 16-bit sum) one-fourth of a PROM bank. A PROM bank consists of four PROM's that span 512 words of program storage. At each fourth call, the total accumulated so far is added to the corresponding checksum in the SUMS storage area. The total should be zero. If it is not, an automatic (debounced) log entry is made. These PROM checksum errors are the highest priority errors that can occur. They will, therefore, mask out in the log any other errors that occur during the same 60-second time interval.

The "COMAND" Routine

All commands requested from the terminal keyboard are initially decoded by this section of the program; and their actual execution, if not completely carried out by this portion of the program, is at least directed by various subroutines within this major code section. The first portion of this area is reserved for various tables to aid in command decoding. Each character of a command entry results in an echo being sent back to the terminal. This includes not only the character itself, but possibly one or both of two other types of information: Either additional characters to complete a word of which the keyboard entered character was the first character, and/or half intensity characters that prompt the user for the next character in the command stream. Which echo is to be sent back in response to a keyboard character is indicated in these tables. In addition, which routine is to be called in response to a completed command is also stored here. Finally, the number of characters allowed at each point in a command entry without causing a command abort condition is stored here for each command sequence possible. The actual command sequence decoding is carried out by the CMDINP routine. This is a co-routine called by the TASK MASTER. It forms something of the backbone of the entire COMAND routine grouping. By making use of the previously mentioned tables, the command decoder sends back appropriate echoes and, when finally required, causes a transfer of the computer control to the proper command execution sequence. An error condition uncovered at any point during a command entry will cause the command line at the bottom of the terminal screen to be erased completely. No action will be taken. Immediately following the CMDINP co-routine are the various sequences of code that handle the execution of each command.

The first of these is the EXLOG routine that carries out the execution of the LOG command. This command is requested from the keyboard of the Snow White Program to display the ten most recent lines of the log. This command execution code naturally makes use of the ONELIN subroutine to scan line by line through the log data til the last twenty lines are found. Two variables indicate starting and ending positions of the displayed log data on the screen. These are variables DSPFST (storing the first character to be displayed on the screen) and the variable DSPLST (referring to the last character of the log to be displayed on the screen). The last few instructions in this routine begin at the location entitled SHOWL. This stream of code actually carries out the display of the appropriate log lines on the terminal screen. It is called not only by the EXLOG code, but also other routines immediately below this to be mentioned shortly. Like all other sequences of code in the COMAND program, at the end of a command execution a return of control is made to the CMDINP calling program. From here, this co-routine will make a transfer of control back to the TASK MASTER.

The next routine here is also a log manipulation subprogram entitled EXLMMD. Its purpose is almost identical to that of the EXLOG routine mentioned immediately above. However, an additional entry is made from the keyboard immediately following the L identifier. This entry is a date. The idea is to display the first twenty log lines having an assigned date equal to or greater than this keyboard entered one. The CMDINP calling routine distinguishes between this subprogram and the one immediately preceeding it according as a switch (IN4ENT) is or is not set. This switch is set by the little IN4WDS subroutine. Presumably the setting of this switch was done during the entry of a date in the keyboard. If no date was entered immediately before

a carriage return, then apparently a date search is not desired and the EXLOG function is wanted. Otherwise, if a date was entered the EXLMMD task is carried out. After the entering of the keyboard date, this routine, like many other log manipulation routines in the Snow White package, does its log search line by line intermittantly with a return of control to the TASK MASTER. This repeated recall of the TASK MASTER is done to prevent the search of a very long log from causing that routine to hog the computer. If this particular routine cannot find a date greater than or equal to the date entered from the keyboard, it will halt at the end of the log and cause the display of merely the last twenty lines. Note that this command makes it possible to display the very top twenty entries in the log by simply tagging a date to the log mnemonic of January 1. Since no log entry can have a date previous to this, the EXLMMD subroutine will select these first twenty entries and display them. At this point, the user may issue "log roll" commands to scan further down the log.

The "log roll" function is carried by out by call to the EXLGRL routine. This routine advances the pointers stored in the DSPFST and the DSPLST variables by exactly ten lines. These updated variables are then passed to the SHOWL portion of the EXLOG routine for display. On the screen the result is that the log has been "rolled" forward by exactly 10 lines or half a screen's worth.

Occasionally, it is desired to make a manual entry to the log from the keyboard. This may be done with the "ENTER" command. After typing E, instructions are placed at the top of the screen to indicate to the user how to carry out his manual command entry. Then, the current date and time which will be associated with this manual entry are displayed. Finally, the screen cursor is moved to the first position where a keyboard entered

character will be placed. The user may now type in his desired entry. He may include carriage returns in this entry if the entry contains multiple lines. To correct a mistake typed previously on the same line, he may use the "RUBOUT" key of the keyboard. This causes the screen cursor to move to the left by one position. It also causes the character at this previous position to be erased. The user may now type this character again causing a correction in the log to be made. If he types successive rubouts, the results will be the wholesale erasing of all text up to but no further than the first character in the current line. This is to say that erasing beyond a line boundary is not possible. To terminate his entry and return to other screen manipulation routines, the user should type his last character as a right bracket. Note that this is to be distinguished from a right parenthesis. The user may note from extensive use that dollar sign and other special characters are not allowed. That is, he will observe that when they are typed in they are not echoed on the screen and indeed are not placed in his log entry. The reason for this is that the dollar sign and other special characters have very important meanings to the software. Thus, they cannot be used as part of general purpose text without causing unpredictable results. The instruction text that's placed at the beginning of a manual entry frame is stored in MOD50 format beginning in location $RTENT.

Whenever the user types S (status), a transfer of control is ultimately made to the EXSTAT code stream. This routine sees that the normal status panel display is placed on the screen. This display is the normal one that should be displayed on the screen unless a user is tending to the terminal and explicitly desires some other. The first action of this routine is to cause the fixed ASCII to be moved to the terminal. That is all titles and field unit names are placed in their fixed positions on the screen. In addition, the "type H for help" message, centered at the bottom of the screen,

is put in position. Also at this time the current date and time is placed on the screen.

Upon typing M (maintenance monitor), the user is given the display of all maintenance monitor signals within the instrument landing system. In addition, the values of five signals ordinarily relayed from the far field monitor to the localizer are also displayed. Finally, the three analog DDM signals are displayed in decimal form on this maintenance monitor display. Once again, the first step in this display, after first clearing the screen, is to place the fixed ASCII information. Then, as before, the date, time, and "type H ... " messages are passed to the terminal. The later display of the maintenance monitor statuses by the STACH2 co-routine is assured by the setting of the various RFSHSW switches associated with the maintenance monitor data. Further, the later decoding and display of the three DDM values is commanded by the setting of the FORCED variable. This variable triggers the action of the DDMDSP co-routine.

The date and time maintained by this software program is displayed not only in the center of the status and maintenance monitor frames, but is also used to tag each automatic and manual log entry. Thus it is important to maintain the system date and time with correct values. To enter a new time value, the T command is used. Its entry ultimately triggers the action of the EXTIME routine. This routine allows four or more digits to be keyed in from the keyboard. These are interpretted as an hour and a minute value, respectively. If fewer than four digits are keyed in, a leading zero will be presumed for the hour number. If more than four numeric digits are keyed in, only the last four will be used. This allows an error keying in the time value to be easily corrected. One simply types in the correct four numeric digits before issuing a final carriage return. This new time is placed in

the permanent second count variable called PSECND. Its ASCII version will be placed in the PTIMAS variable location. Also two automatic log entries will be made. One log entry indicating "time before time change" will be made and will have the old system time tagged onto it. The second automatic log entry "NEW TIME" will be tagged with the new time. Thus, time lapses between the old and new time are permanently recorded in the log.

The software system date may be corrected by use of the D command. Once again, after typing D the software program expects four numeric digits to follow, the first two being a month number and the second two being a day number. Errors in any of these four digits are easily recovered from by simply retyping the date. That is, the software examines only the last four numeric digits entered. The command is terminated with a carriage return keystroke. Other non-numeric characters are echoed back to the screen but are otherwise ignored. As in the time command, two automatic log entries are made. The first has the older date associated with it, while the second "new date" entry has this keyboard entered date associated with it.

The next command is a rather powerful one that allows cycle commands to be issued from the keyboard. The first field is simply the letter C which is echoed back on the screen as the word "CYCLE". The second field is one of six values: L for localizer, G for glide slope, I for inner marker, ..., A for "all." Thus it is possible to cycle any single transmitting field unit and in addition to cycle all five such field units in unison. The third field in this command may or may not be present. If it is, it is either an M for "MAIN" or the letter O for "OFF." That is, one may mention the desired destination state of the transmitting field unit if desired. If this third entry is not made, then a single cycle command will

be sent to the field unit advancing it by one state increment regardless of what the final destination turns out to be. The last field entered from the keyboard must always be present and is the correct four digit password. Mistakes may be corrected by simply typing the correct four digits once again. The command is terminated by a carriage return. The user will notice that the password is not echoed on the screen. The cycle command will not be honored unless the program verifies that the status display is currently on the screen. This is done so that the user may see firsthand the transmitter state changes that are effected by his cycle command. Cycle commands issued without displaying the status screen first are ignored. Cycle commands having the third field present in which a destination is mentioned cause up to three sets of cycle commands to be issued to any one field unit. Delays required by the instrument landing system gear are accounted for. In particular, this means that passage through the OFF state of any transmitter must require at least twenty seconds. Whenever a "non-off" state is entered, a wait of only five seconds is initiated. During the execution of the cycle command, no matter how complicated it is required to be, a blinking CYCLE is placed at the far right of this field unit's status line on the terminal screen. This blinking entry will be extinguished after the cycle command has been completed or an error is noted. The only possible error that may be encountered in this cycle command execution is the failure of a field unit to respond. Should this occur after the appropriate waiting time is made, an automatic log entry indicating "NO RESPONSE TO CYCLE" is made.

The keyboard entry of dates, times, and passwords are all handled by one subroutine entitled IN4NUM. It examines keyboard entered characters to determine whether or not they are numeric. If they are non-numeric, then the

keyboard entry is ignored, and a co-routine return is made. If the character is numeric, then it is stripped of its ASCII status and placed in the IN4WDS array. This is a four-byte array holding up to four numeric values. A pointer entitled NXTWD is used to point to the next open byte in this four-byte array.

Bytes may be extracted from this array in order by a subroutine call to the GETIN4 subroutine.

When typing H, the user is requesting help. The result is a transfer of control to the EXHELP subroutine, which places a number of lines of fixed ASCII information on the screen. The explanation in these lines gives the user a choice of four frames of detailed "help" information.

One of these frames is the information frame pertinent to the status and maintenance monitor displays. A display of this help frame is carried out by the EXQS subroutine. A second help frame gives the user a brief introduction to the various log manipulation commands. This help frame may be called up by a "?L" keyboard entry. The help frame giving the user an introduction to the date-time commands may be viewed by typing "?D." Finally, a "?C" keyboard entry calls the EXQC routine into action. This gives the user a rudimentary description of the cycle commands that may be issued from the keyboard. The two last subroutines in the COMAND code package are the TYPEH subroutine (which displays the "type H for help" message on the screen), and the EARCMD subroutine (which erases the command input line at the bottom of the screen).

Appendix E

Field Unit Program Listing

BEST AVAILABLE COPY

```
; ********************************************************
; *                PROGRAM FOR THE DWARVES               *
; ********************************************************
;
;       INTEL 4040 MICROPROCESSOR CONTROL PROGRAM FOR
;          THE MARK III INSTRUMENT LANDING SYSTEM
;       INSTALLED AT NAFEC IN ATLANTIC CITY, NEW JERSEY
;
;   THIS PROGRAM IS USED TO CONTROL THE REMOTE FIELD SITES OF
; THE RELIABLE COMMUNICATION SYSTEM INSTALLED FOR THE FAA BY THE
; CIRCUITS AND SYSTEMS GROUP OF THE SCHOOL OF ELECTRICAL ENGI-
; NEERING, PURDUE UNIVERSITY, WEST LAFAYETTE, INDIANA 47907.  THE
; MICROPROCESSOR IS USED IN CONJUNCTION WITH A STANDARD PRIVATE
; LINE MODEM UTILIZING FREQUENCY-SHIFT KEYING OVER ISOLATED,
; BALANCED LINES.  THE STATUS OF THE COMMUNICATIONS AND THE
; OPERATION OF THE SYSTEM MAY BE DETERMINED BY READING THE SEVEN
; SEGMENT LIGHT-EMITTING DIODE DISPLAY LOCATED ON THE FRONT PANEL
; OF THE FIELD UNITS.
;
;   THIS PROGRAM IS DESIGNED TO WORK AT ALL SIX OF THE REMOTE
; FIELD SITES OF THE ILS.  THIS MAKES THE MICROPROCESSOR CARDS
; WITH THEIR PROMS INTERCHANGABLE.  THERE ARE, HOWEVER, FOUR
; DISTINCT FUNCTIONS TO BE PERFORMED BY THE PROGRAM DEPENDING ON
; WHERE IT IS BEING EXECUTED.
;   1) AT THE THREE MARKERS, INNER (IM), MIDDLE (MM), AND
; OUTER (OM), THE PROGRAM READS, FORMATS AND SENDS SIX DIGITAL
; STATUS SIGNALS AND RECEIVES ONE DIGITAL COMMAND SIGNAL.
;   2) AT THE GLIDE SLOPE (GS) THE PROGRAM READS, FORMATS,
; AND SENDS SEVEN DIGITAL STATUS SIGNALS AND SEVENTEEN DIGITAL
; PRE-ALARM SIGNALS AND RECEIVES ONE DIGITAL COMMAND SIGNAL.
;   3) AT THE FAR-FIELD MONITOR (FFM) THE PROGRAM READS,
; FORMATS, AND SENDS SIX DIGITAL STATUS SIGNAL, FOUR DIGITAL
; PRE-ALARM SIGNALS, AND THREE ANALOG SIGNALS.  THE ANALOG
; SIGNALS ARE CONVERTED TO DIGITAL REPRESENTATION FOR SENDING
; VIA DIGITAL FSK.  THE PROGRAM CHECKS FOR ANY DRIFT IN THE
; ZERO OF THE A/D AND SUBTRACTS IT FROM THE DDM SIGNALS TO
; BE SENT.
;   4) AT THE LOCALIZER (LOC) THE PROGRAM READS, FORMATS,
; AND SENDS SEVEN DIGITAL STATUS SIGNALS AND TWENTY
; DIGITAL PRE-ALARM SIGNALS, AND RECEIVES ONE DIGITAL
; COMMAND SIGNAL, FIVE DIGITAL STATUS SIGNALS, AND THREE
; ANALOG SIGNALS.  PROVISION IS MADE AT THE LOCALIZER TO ALLOW
; EASY CALIBRATION OF THE D/A CONVERTER.
;   THE PROGRAM DETERMINES ITS LOCATION, AND HENCE WHAT IT IS
; SUPPOSED TO DO, FROM THREE BITS OF ONE INPUT PORT.  THESE
; THREE BITS ARE WIRED DIFFERENTLY (1 OR 0) AT EACH LOCATION,
; AND THEREFORE IDENTIFY THE CURRENT LOCATION OF THE MICRO-
; PROCESSOR CARD.  THESE BITS ARE USED IN ALL MESSAGES TO OR
; FROM THE ATCT TO PREVENT THE ACCIDENTAL CROSSING OF WIRES AND
; RESULTING CONFUSION OF STATUS AND COMMAND SIGNALS.
;   THE PROGRAM PROVIDES A SIMPLE WAY OF DETERMINING THE STATUS
; OF THE COMMUNICATIONS EQUIPMENT, AND ALSO AN EXTREMELY USEFUL
; TOOL FOR TROUBLESHOOTING - THE SEVEN SEGMENT FRONT PANEL
; DISPLAY WITH THUMBWHEEL SWITCH.  THE TECHNICIAN DESIRING
; INFORMATION USES THE SWITCH TO SPECIFY WHAT SPECIFIC INFOR-
; MATION IS DESIRED (SYSTEM STATUS, DATA BEING SENT, DATA
; BEING RECEIVED, LAMP TEST, ETC.), THE MICROPROCESSOR THEN
; RESPONDS BY DISPLAYING ENGLISH MESSAGES OR OTHER PATTERNS ON
; THE LEDS.
;   MOST OF THE COMMUNICATION PORTION OF THE PROGRAM IS INTER-
; RUPT DRIVEN.  THE ONLY DEVICE WHICH MAY INTERRUPT THE PROGRAM
; IS THE ACIA.  ALL RECEIVED DATA IS READ, CHECKED, AND INTER-
; PRETED BY THE INTERRUPT ROUTINE.  ANY ERROR IN THE RECEIVED
; DATA (OVERRUN, PARITY, FRAMING, CODING) IS SUFFICIENT CAUSE FOR
; THE WHOLE BLOCK TO BE IGNORED.  WITH THE EXCEPTION OF THE LOC,
; THE RECEIVED BLOCK CONSISTS OF ONLY ONE FRAME, HENCE THE
; CHECKING AND INTERPRETATION OCCURS IMMEDIATELY AFTER RECEIPT.
; AT THE LOCALIZER, HARDWARE ERRORS ARE CHECKED FOR AFTER EACH
; FRAME, BUT CODING ERRORS ARE NOT CHECKED AND INTERPRETATION
; DOES NOT BEGIN UNTIL AFTER THE WHOLE BLOCK HAS BEEN RECEIVED.
;   THE INITIATION OF A TRANSMISSION IS CONTROLLED BY THE MAIN
; PROGRAM, AND OCCURS WHENEVER ONE OF THE FOLLOWING CONDITIONS
; IS SATISFIED - 1) A NAK OR REQUEST FOR TRANSMISSION IS RE-
; CEIVED FROM THE ATCT, 2) THE VALUE OF ANY OF THE SENSED DATA
; (STATUS, PRE-ALARM, OR ANALOG SIGNALS) CHANGES, OR 3) APPROX-
; IMATELY FIFTEEN SECONDS HAVE ELAPSED SINCE THE LAST TRANS-
; MISSION.  AT THIS POINT THE MAIN PROGRAM FORMATS THE DATA TO
; BE SENT WITH HEADER AND TRAILER FRAMES.  IT THEN ENABLES
; TRANSMITTER INTERRUPTS, LEAVING THE ACTUAL TRANSMISSION OF
; FRAMES TO THE HARDWARE AND THE INTERRUPT ROUTINE.
;   DIGITAL DATA SENSED BY THE PROGRAM IS DEBOUNCED IN THE
; SOFTWARE (ALL DIGITAL INFORMATION IS READ FROM RELAY CONTACTS
; IN THE ILS) BY REQUIRING THAT THE VALUES REMAIN THE SAME FOR
; FOUR SUCCESSIVE SAMPLES.  ANALOG DATA IS READ UNDER SOFTWARE
; CONTROL.  THE PROGRAM CONTROLS THE ANALOG MULTIPLEXER WHICH
; SELECTS BETWEEN THE THREE DDM SIGNALS AND A ZERO REFERENCE,
; AND PROVIDES THE START CONVERSION SIGNAL.
;   THE PROGRAM ALSO PERFORMS DIAGNOSTICS ON THE HARDWARE AND
; SOFTWARE.  IF HANDSHAKING IS LOST, OR CANNOT BE ESTABLISHED
; WITH THE MODEM AT THE ATCT, THE MODEM AT THE FIELD UNIT IS
; SELF-CHECKED.  A COMPLETE CHECKSUM OF THE PROGRAM MEMORY IS
; COMPUTED EVERY HALF SECOND TO CHECK FOR FAULTY PROMS.  IN
; ADDITION, THE PROGRAM EXECUTES A SPECIAL SECTION OF CODE
; EVERY QUARTER OF A SECOND WHICH RESETS THE WATCHDOG TIMER.
; IF FOR SOME REASON OF SOFTWARE OR HARDWARE MISFUNCTION THIS
; RESET IS NOT PERFORMED, THE WATCHDOG CIRCUITRY AUTOMATICALLY
; RESETS THE MICROCOMPUTER.  THIS FEATURE ALLOWS THE SYSTEM TO
; RECOVER FROM *SOFT* PROBLEMS RAPIDLY (ABOUT 2 SECONDS) AND
; AUTOMATICALLY.
;   IN THE DISCUSSION BELOW, THE ABBREVIATION IR(*) REFERS
; TO ONE OR MORE INDEX REGISTERS, AND BITS IN A FOUR BIT
; REGISTER ARE REFERED TO BY THEIR BINARY VALUE (1, 2, 4, &
; 8).  MSB, LSB, MSN, AND LSN REFER TO MOST SIGNIFICANT BIT,
; LEAST SIGNIFICANT BIT, MOST SIGNIFICANT NIBBLE (4 BITS), AND
; LEAST SIGNIFICANT NIBBLE.  FOR INFORMATION ON THE HARDWARE
; ORGANIZATION AND INSTRUCTIONS, SEE THE INTEL CORPORATIONS
; BOOK - 4040 USERS MANUAL.
;   THE LOGICAL FLOW OF THE PROGRAM IS OUTLINED BELOW:  THE
; OPERATION OF THE PROGRAM IS CONTROLLED BY FOUR FLAGS WHICH ARE
; KEPT IN IR(10).  EACH BIT IS USED TO DETERMINE THE STATE OF THE
; PROGRAM AT ANY GIVEN INSTANT IN TIME.  THESE BITS AND THEIR
; MEANINGS ARE:
;   BIT 1 (LSB) - NEW DATA PRESENT FLAG.  WHEN EQUAL TO 1, THIS
; BIT SIGNIFIES THAT THE DATA HELD IN DATA RAM REGISTER 2 HAS
; CHANGED SINCE THE LAST TRANSMISSION TO THE REMOTE CONTROL, AND
; HENCE A NEW TRANSMISSION SHOULD BE MADE AS SOON AS FEASIBLE.
; SETTING THIS BIT IS ALSO USED TO FORCE A NEW TRANSMISSION WHEN
; A NAK IS RECEIVED, OR AFTER A RECEPTION ERROR.  THIS BIT IS
; RESET WHEN A NEW BLOCK OF INFORMATION IS FORMATTED TO BE SENT.
;   BIT 2 - OKAY TO UPDATE FLAG.  WHEN 1, THIS FLAG SIGNIFIES
```

BEST AVAILABLE COPY

```
; THAT THE PROGRAM MAY CHANGE THE DATA FRAMES HELD IN DATA RAM
; REGISTER 2.  BECAUSE THE CRC IS COMPUTED BEFORE A BLOCK IS
; SENT, IT IS IMPORTANT THAT THE INFORMATION HELD HERE NOT BE
; CHANGED UNTIL IT HAS BEEN SENT TO THE REMOTE CONTROL.  AFTER
; THE DATA FRAMES HAVE BEEN SENT, THIS FLAG IS SET TO ALLOW THE
; FORMING OF A NEW BLOCK OF INFORMATION.  THIS FLAG IS RESET WHEN
; THE BLOCK IS BEING FORMATTED TO BE SENT
;   BIT 4 - DATA BEING SENT FLAG.  THIS FLAG IS USED TO INDICATE
; THAT DATA IS CURRENTLY BEING SENT TO THE REMOTE CONTROL
; DURING AN INTERRUPT, IF THE TRANSMIT BUFFER IS EMPTY AND THE
; FLAG EQUALS 1, THE NEXT FRAME IS LOADED AND SENT.  AFTER THE
; LAST FRAME IS SENT BY THE INTERRUPT ROUTINE, IT CLEARS THIS BIT
; TO SIGNAL THE MAIN PROGRAM THAT THE BLOCK HAS BEEN SENT IN ITS
; ENTIRETY.  THIS BIT IS SET WHEN THE BLOCK IS BEING FORMATTED TO
; BE SENT
;   BIT 8 (MSB) - IGNORE FRAME FLAG.  THIS BIT IS RESET WHEN THE
; FIRST FRAME IN A BLOCK IS RECEIVED WITHOUT A HARDWARE ERROR
; (PARITY, OVERRUN, OR FRAMING).  IF AN ERROR IS DETECTED BY THE
; SOFTWARE IN THE FORMATTING (CRC OR DATA ERROR), OR A FRAME IS
; RECEIVED WITH A HARDWARE ERROR IN A MULTIPLE FRAME BLOCK, THIS
; FLAG IS SET TO INDICATE THAT THE INFORMATION IN THE BLOCK IS TO
; BE IGNORED
;   WHEN THE PROGRAM IS INITIALIZED AFTER A RESET OR AFTER
; BECOMING 'HUNG', THE CONTENTS OF IR(10) ARE SET TO HEXIDECIMAL
; B.  THIS FORCES THE PROGRAM TO WAIT FOR A NEW, COMPLETE BLOCK
; OF INFORMATION BEFORE UPDATING THE CYCLE RELAY OR OTHER
; OUTPUTS, INDICATES THAT NO DATA IS BEING SENT, ALLOWS THE
; 'SIDC' PORTION OF THE PROGRAM TO UPDATE THE INFORMATION TO BE
; SENT, AND INDICATES THAT A NEW BLOCK SHOULD BE SENT.  THE
; STRUCTURE OF THE PROGRAM IS SUCH THAT THIS FRAME WILL NOT BE
; SENT FOR AT LEAST 1/8-TH OF A SECOND, ALLOWING THE NEW DATA TO
; BE PLACED INTO DATA RAM REGISTER 2.  THE MNEUMONIC 'SIDC'
; STANDS FOR 'SCAN INPUTS, UPDATE THE DISPLAY, AND PERFORM A
; CHECKSUM ON PROGRAM MEMORY (PROM).'
;
;   MOST OF THE I/O DONE BY THE PROGRAM SIMULATES THE OPERATION
; OF AN EIGHT BIT BUS.  THERE IS AN EIGHT BIT ADDRESS BUS (OUTPUT
; PORTS 2-3), AN EIGHT BIT OUTPUT BUS (OUTPUT PORTS 0-1), AND TWO
; INPUT BUSES (MAIN IS INPUT PORTS 2-3, ALTERNATE IS INPUT PORTS
; 0-1).  MUCH OF THE DEVICE SELECTION LOGIC FOR THE SIMULATED
; EIGHT BIT MACHINE IS CONTAINED ON THE INTERFACE CARD.  THE
; DEVICE ADDRESSES WHICH SHOULD BE WRITTEN TO THE ADDRESS BUS AS
; LOADED INTO THE ACCUMULATOR ARE
;       DEVICE/OPERATION                ACCUMULATOR
;     NOP                                 A0
;     READ ACIA STATUS                    76
;     WRITE ACIA CONTROL                  77
;     READ ACIA BUFFER                    74
;     WRITE ACIA BUFFER                   75
;     WRITE MODEM CONTROL                 F1
;     LOAD DISPLAY, RESET WATCHDOG        07
;     LOAD DISPLAY ONLY                   87
;     LOAD CYCLE RELAY                    97
;     READ ILS SWITCHES/RELAYS            B6
;     READ CLOCK, ID, THUMBWHEEL          A6
;     START A/D CONVERSION                E7
;     LOAD ANALOG SWITCHES                D7
;     READ ADC RELAY INPUTS               C4  (ALT INPUT BUS)
;     READ ADC MAGNITUDE BITS             C6  (ALT INPUT BUS)
;     WRITE DAC RELAY REGISTER            B7
;     LOAD DAC 1                          C7
;     LOAD DAC 2                          D7
;     LOAD DAC 3                          E7
;   ADDRESSES WHOSE MSB IS ZERO ARE USED TO TRIGGER THE ONE-SHOT
; ON THE INTERFACE BOARD WHICH SIMULATES THE ENABLE PULSE
; REQUIRED BY THE ACIA AND WATCHDOG TIMER.  IN ADDITION TO THE
; DEVICES LISTED ABOVE, THERE IS ALSO AN INPUT MULTIPLEXER CARD
; USED AT THE LOCALIZER AND GLIDE SLOPE WHICH IS CONTROLLED BY
; OUTPUT PORT 0 AND READ FROM INPUT PORT 0.
;
;   THE ACIA IS PROGRAMMABLE AS TO MODE OF OPERATION AND THE
; TYPES OF INTERRUPTS.  IN ALL CASES, IT IS CONTROLLED BY THIS
; PROGRAM TO DIVIDE THE CLOCK FREQUENCY (19.2 KHZ) BY 64 TO GET
; 300 BAUD, AND TO SEND EIGHT DATA BITS, AN ODD PARITY BIT, AND
; ONE STOP BIT.  THE FOUR BYTES WRITTEN TO THE ACIA AND THEIR
; EFFECT ARE
;       OPERATION                     ACCUMULATOR
;     MASTER RESET                        E0
;     NO INTERRUPTS                       E1
;     ONLY RECEIVER INTERRUPTS            61
;     RECEIVER AND TRANSMITTER INTERRUPTS 41
;
;   EACH OF THE FIELD LOCATIONS HAS AN IDENTIFICATION NUMBER
; ASSOCIATED WITH IT.  THIS NUMBER APPEARS AS A PART OF EACH
; MESSAGE BETWEEN IT AND THE REMOTE CONTROL TOWER.  SINCE THE 4
; HERTZ CLOCK APPEARS AS THE LSB OF IR(14), THE ID BITS STORED
; IN IR(14) MAY TAKE ON TWO VALUES.  THE IDENTIFICATION NUMBERS
; ARE:
;       LOCATION          ID          IR(14)
;     LOCALIZER           1           2-3
;     GLIDE SLOPE         2           4-5
;     INNER MARKER        3           6-7
;     MIDDLE MARKER       4           8-9
;     OUTER MARKER        5           10-11
;     FAR-FIELD MONITOR   6           12-13
;
;   THE OPERATION OF THE MODEM IS CONTROLLED BY THE MODEM CONTROL
; REGISTER.  THIS GOVERNS THE TYPE OF MODEM (ANSWER/ORIGINATE),
; THE MODE (NORMAL/SELF-TEST), AND WHICH SIGNAL IS FED TO THE
; DEMODULATOR (RECEIVE FILTER/TRANSMITTER SQUARE WAVE).  THE
; THREE NIBBLES WRITTEN TO THE MODEM CONTROL REGISTER ARE:
;       OPERATION                     ACCUMULATOR
;     NORMAL ORIGINATE MODE               D
;     SELF-TEST ORIGINATE                 9
;     SELF-TEST ANSWER                    2
;
;
;                                   - STEPHEN E. BELTER
;                                   - MAY 31, 1976
;
;
        =0
;
;   THERE ARE TWO SPECIAL ADDRESSES IN PROGRAM MEMORY FOR THE
; 4040 - ADDRESS 0 WHERE THE PROCESSOR STARTS EXECUTION AFTER A
; RESET, AND ADDRESS 3 WHERE THE PROCESSOR JUMPS WHEN AN INTER-
; RUPT IS RECEIVED.  FRO-LOG CORPORATION RECOMMENDS PLACING A
; NOP AT ADDRESS 0 BECAUSE OF POSSIBLE TIMING PROBLEMS CAUSED
; DURING A RESET.  THE JUMP INSTRUCTION SKIPS TO THE BEGINNING
; OF THE MAIN PROGRAM, SKIPPING THE INTERRUPT ROUTINE AND SOME
```

BEST AVAILABLE COPY

```
; MISCELLANEOUS SERVICE ROUTINES.  THE LOCATION OF SUBROUTINES
; IN THIS PROGRAM WAS DONE TO AVOID DIVIDING A SUBPROGRAM
; BETWEEN TWO PROMS.  THE 4040 DOES NOT HANDLE A CONDITIONAL
; JUMP ACROSS PROM BOUNDARIES IN A UNIFORM MANNER, MAKING IT A
; VERY HAZARDOUS OPERATION IN PRACTICE.  THE RESULTING PROGRAM
; FLOW (EXTERIOR TO INDIVIDUAL ROUTINES) IS ESSENTIALLY RANDOM.
;
        NOP             ; RECOMMENDED BY PRO-LOG
        JUN INITIAL     ; JUMP TO THE INITIALIZATION ROUTINE
;
;   THIS IS THE INTERRUPT HANDLER.  IT IS DESIGNED TO SERVICE 3
; TYPES OF INTERRUPTS, RECEIVER BUFFER FULL, TRANSMIT BUFFER
; EMPTY, OR LOSS OF CARRIER.  THE THREE INTERRRUPTS ARE SERVICED
; IN THIS ORDER, WITH THE SUBROUTINE CHECKING FOR NEW INTERRUPTS
; BEFORE EXITING.
;   THIS IS A LEVEL 5 SUBROUTINE WHICH IMPLIES THAT THE MAIN
; ROUTINE MAY NOT BE EXECUTING MORE THAN A LEVEL 4 SUBROUTINE
; WHEN IT IS INTERRUPTED TO AVOID PROGRAM COUNTER STACK OVERFLOW!
;
;   THE INTERRUPT ROUTINE USES INDEX REGISTERS IN BANK 1 TO AVOID
; INTERFERING WITH THE MAIN LINE ROUTINE.  SINCE THE LOCALIZER
; DATA RECEIVING PROCEDURE USES THE VALUE OF IR(6) IN BANK 0, IT
; IS TRANSFERED TO BANK 1 IN THE PROCESS OF SAVING THE ACC.
; TO CONSERVE DATA MEMORY SPACE, THE CARRY BIT IS SAVED IN THE
; COMMAND REGISTER..
;
        XCH 6           ; SAVE THE ACC IN IR(6) OF BANK 0
        SB1             ; TRANSFER ORIGINAL IR(6) DATA TO BANK 1
        XCH 6           ; IR(6) OF BANK 0 NOW SAVED IN BANK 1
        RAL             ; SAVE THE CARRY IN BIT 2 OF THE COMMAND
        STC             ;   REGISTER BY USING RAM BANK 1 IF NO
        RAL             ;   CARRY, AND 'USING' BANKS 1 AND 2 IF
        DCL             ;   THE CARRY IS SET.
        JMS RDCNTL      ; READ THE CONTROL REGISTER OF THE ACIA
;
;  CHECK THE CONTROL REGISTER OF THE ACIA FOR THE TYPE OF
;  INTERRUPT.  THIS IS THE POINT WHERE THE ROUTINE LOOPS BACK IF
;  ANOTHER INTERRUPT OCCURS WHILE SERVICING THE FIRST ONE.
;
INTRUPT: LD 3           ; LOW NIBBLE OF THE CONTROL REGISTER
        RAR             ; MOVE RECEIVER BUFFER FLAG INTO CARRY
        JCN 2, XMIT1    ; JUMP IF NOT A RECEIVER INTERRUPT
        LD 2            ; NOW CHECK FOR RECEIVER ERRORS
        RAL             ; CHECK FOR A PARITY ERROR
        RAL
        JCN 2, FRAMERR  ; JUMP IF NO ERROR
        LDM 8           ; ERROR CODE FOR A PARITY ERROR
        JUN ACIAERR
FRAMERR: LD 2           ; CHECK FOR A FRAMING ERROR
        RAR             ; MOVE FRAMING ERROR FLAG INTO CARRY
        JCN 2, OVERRUN  ; JUMP IF NO ERROR DETECTED
        LDM 6           ; ERROR NUMBER FOR FRAMING ERROR
        JUN ACIAERR     ; REGISTER THE ERROR
OVERRUN: RAR            ; MOVE OVERRUN FLAG INTO THE CARRY
        JCN 2, READFRM  ; JUMP IF NO ERROR DETECTED
        LDM 5           ; ERROR CODE FOR OVERRUN ERROR
ACIAERR: JMS RCVERR     ; REGISTER A RECEIVER ERROR
        JMS RDFRAME     ; CLEAR THE ERROR
        JUN XMIT        ; LOOK FOR A TRANSMITTER INTERRUPT
READFRM: JMS RDFRAME    ; READ THE FRAME THAT WAS RECEIVED
;
;  AT THIS POINT SEPARATE HANDLING IS REQUIRED IF THIS IS THE
;  LOCALIZER.  THE LOCALIZER EXPECTS AN EIGHT FRAME BLOCK WHICH
;  INCLUDES A CRC CHECKSUM.  ALL OTHER FIELD UNITS GET A SINGLE
;  FRAME BLOCK WHICH SHARE A COMMON FORM.  NOW CHECK TO SEE IF
;  WE ARE EXECUTING AT THE LOCALIZER..
;
        CLC             ; CHECK TO SEE IF THIS IS THE LOCALIZER
        LDM C
        ADD 14          ; IR(14) WILL BE A 2 OR 3 IF LOC
        JCN 2, SINGLE   ; JUMP IF NOT THE LOCALIZER
;
;  PROCESS A FRAME RECEIVED AT THE LOCALIZER BY LOADING IT INTO
;  DATA RAM, THEN CHECKING TO SEE IF IT IS THE FINAL FRAME OF A
;  BLOCK.
;
        LD 2            ; IS THIS THE HEADER?
        RAL             ; CY=0 IMPLIES THIS IS THE HEADER FRAME
        FIM 4, 30       ; CONSTRUCT POINTER TO RAM IN IR(4-5)
        LD 6
        JCN 2, BODY     ; JUMP IF NOT THE HEADER
        JMS CLRIGNR     ; CLEAR THE IGNORE FRAME FLAG
BODY:   XCH 5           ; POINTER TO RAM IN IR(4-5) IS NOW VALID
        JMS WRITMEM     ; WRITE IR(2-3) INTO MEMORY AT IR(4-5)
        LD 5            ; PUT THE POINTER BACK IN IR(6)
        XCH 6
        ISZ 6, XMIT     ; JUMP IF NOT THE LAST FRAME OF THE BLOCK
;
;  WE HAVE RECEIVED A COMPLETE BLOCK OF INFORMATION AT THE
;  LOCALIZER.  COMPUTE THE CRC FOR THIS BLOCK, CHECK THE ACK/NAK
;  NIBBLE, THEN CHECK THE CRC.
;
        FIM 0, 30       ; POINTER TO THE HEADER
        LDM 6           ; PROCESS 12 NIBBLES
        JMS CRC         ; CRC IS LEFT IN IR(3,5,4)
        SBM             ; COMPARE THE ACK/NAK WITH AN ACK
        JCN 4, RCVDACK  ; JUMP IF WE RECEIVED AN ACK
        JMS NEWDATA     ; SET NEWDATA FLAG TO REQUEST X-MISSION
RCVDACK: JMS CHKCRC
;
;  IN THE PROCESS OF CHECKING THE CRC, WE HAVE CLEARED IR(3).  IF
;  THERE IS A CRC ERROR, WE WILL SIGNAL IT BY FAKING THE
;  RECEPTION OF A BAD CYCLE FIELD.  (SAVES 4 BYTES)
;
        LD 7            ; FLAG TO INDICATE CRC ERROR
        JCN C, CHKCYC
;
;  READ THE HEADER FRAME AND CHECK THE ID BITS.
;
        FIM 0, 30
        JMS READMEM     ; READ THE HEADER FRAME INTO IR(4-5)
        LD 4            ; MOVE ID BITS FROM IR(4) TO IR(2)
        XCH 2
        JMS CHECKID     ; CHECK THE ID BITS
;
;  IF THE IGNORE BLOCK FLAG IS NOW ON, JUMP AROUND THE LOADING
;  OF THE DDM SIGNALS AND DIGITAL FFM SIGNALS.
;
        LD 10
        RAL
```

BEST AVAILABLE COPY

```
        JCN 2, XMIT     ; IF ON, JUMP TO CHECK THE XMIT BUFFER
;
;  NOW CHECK THE FFM COMMUNICATIONS STATUS BITS FOR A DDM ERROR
;  WHICH MIGHT AFFECT US.  THEN LOAD THE FFM RELAYS AND DDM
;  DIGITAL-TO-ANALOG CONVERTERS.
;
        LDM B           ; IF ERROR IS 5 OR GREATER, THEN IT IS
        ADD 5           ;    NOT AN ADC ERROR
        JCN 2, NOADERR
        LD 5            ; IT IS AN ADC ERROR, REGISTER IT
        JMS ERROR
NOADERR FIM 0, 32       ; SECOND FRAME CONTAINS THE FFM RELAYS
        JMS READMEM     ; READ MEMORY INTO IR(4-5)
        JMS CMA54       ; INVERT TO GET PROPER VALUE TO WRITE
        FIM 2, B7       ; WRITE TO THE RELAYS ON THE DAC CARD
        JMS WRITE
        INC 1           ; POINTER TO THE DDM SIGNS
        SRC 1
        RDM             ; READ THE SIGNS INTO IR(7)
        RAL             ; TOSS OUT THE LEADING BIT
        XCH 7
        LDM 5           ; IF THE THUMBWHEEL SWITCH IS ON POSITION
        SUB 13          ;   4, DO NOT LOAD THE DDM VALUES INTO
        JCN 4, ETC      ;   THE D/A CONVERTERS
        INC 1           ; ADDRESS OF DDM 1 MAGNITUDE
        JMS WRDAC       ; CONVERT MAGNITUDE, LOAD DAC
        JMS WRDAC       ; CONVERT MAGNITUDE, LOAD DAC
        JMS WRDAC       ; CONVERT MAGNITUDE, LOAD DAC
;
;  NOW READ AND CHECK THE CYCLE COMMAND NIBBLE.  IF ONE OF THE
;  TWO LEGAL VALUES, DEBOUNCE IT AND SET THE RELAY.
;
ETC     FIM 0, 35       ; POINTER TO THE CYCLE NIBBLE
        SRC 1
        RDM
        XCH 3           ; PLACE THE CYCLE NIBBLE IN IR(3)
        JUN CHKCYC      ; JUMP TO THE CALL TO SUBROUTINE CYCLE
;
;  PROCESS A SINGLE FRAME BLOCK BY REGISTERING THE RECEIPT OF A
;  NAK, CHECKING THE ID BITS, THEN CHECKING AND PROCESSING THE
;  CYCLE FIELD.
;
SINGLE  JMS CLRIGNR     ; CLEAR THE IGNORE FRAME FLAG
        LD 2            ; CHECK FOR A NAK (LSB OF MSN)
        CLC             ; NEEDED FOR CHECKING THE ID BITS
        RAR             ; MOVE NAK BIT INTO THE ACC
        XCH 2           ; SAVE THE ID BITS
        JCN 2, ACK      ; JUMP IF ACK RECEIVED
        JMS NEWDATA     ; REQUEST A NEW TRANSMISSION
ACK     JMS CHECKID     ; NOW CHECK THE ID BITS
CHKCYC  JMS CYCLE
;
;  CHECK FOR A TRANSMIT INTERRUPT.  SINCE THE TRANSMIT INTERRUPT
;  MAY BE DISABLED, IF THE TRANSMIT BUFFER IS EMPTY, WE ALSO NEED
;  TO CHECK TO BE SURE THERE IS DATA TO BE SENT.
;
XMIT    JMS RDCNTL      ; GET THE LATEST SCOOP
        LD 3            ; XMIT BUFFER IS 2'S BIT OF LSN
        RAR
XMIT1   RAR             ; ENTRY IF NOT REQUEST INTERRUPT
        JCN 2, CHKDCD   ; JUMP IF TRANSMIT BUFFER IS FULL
        LD 10           ; CHECK TO BE SURE THERE IS DATA TO BE
        RAL             ;   SENT
        RAL
        JCN A, CHKDCD   ; JUMP IF WE ARE NOT SENDING DATA
;
;  LOAD THE TRANSMIT BUFFER WITH THE NEXT FRAME TO BE SENT.  IF
;  THIS IS THE LAST DATA FRAME, SET THE 'OKAY TO UPDATE' FLAG IN
;  IR(10).  IF THIS IS THE END OF THE BLOCK, CLEAR THE SENDING
;  DATA FLAG.
;
        FIM 0, 20       ; FORM A POINTER TO THE NEXT FRAME
        LD 9            ; TRANSMIT COUNTER
        XCH 1
        JMS READMEM     ; READ THE NEXT FRAME TO BE SENT
        JMS CMA54       ; COMPLEMENT FOR PRO-LOG'S NEGATIVE LOGIC
        FIM 2, 75       ; ADDRESS OF THE TRANSMITTER BUFFER
        JMS WRITE       ; SEND IT
        INC 9           ; UPDATE THE TRANSMIT COUNTER
        INC 9
        CLC             ; HAVE ALL THE DATA FRAMES BEEN SENT?
        LD 9
        SUB 8           ; NUMBER OF NIBBLES BEFORE TRAILER
        JCN A, CHKEND   ; JUMP IF NOT ALL DATA FRAMES SENT YET
        LDM 2           ; SET OKAY TO UPDATE FLAG
        JMS SETFLG
CHKEND  CLC
        LDM 4           ; HAVE ALL THE FRAMES IN THE BLOCK BEEN
        ADD 8           ;   SENT?
        CLC
        SUB 9
        JCN C, CHKDCD   ; JUMP IF NOT ALL SENT YET
        LDM B           ; CLEAR DATA BEING SENT FLAG
        JMS CLRFLG
        JMS RCVONLY     ; DISARM TRANSMIT INTERRUPTS
;
;  CHECK FOR A LOSS OF CARRIER INTERRUPT (DCD = DATA CARRIER
;  DETECT)
;
CHKDCD  JMS RDCNTL      ; READ THE CONTROL REGISTER
        LD 3            ; SHIFT DCD INTO THE CARRY
        RAL
        RAL
        JCN 2, FININT   ; JUMP IF CARRIER IS DETECTED
        LDM D
        JMS RCVERR      ; REGISTER DCD ERROR
        JMS RDFRAME     ; READ THE RECEIVER BUFFER TO CLEAR DCD
;
;  NOW CHECK FOR ANY NEW INTERRUPTS.  IF NONE ARE FOUND, THEN
;  RESTORE THE CARRY, ACC, AND IR(6) OF BANK 0.
;
FININT  JMS RDCNTL      ; READ THE CONTROL REGISTER OF THE ACIA
        LD 2            ; CHECK THE INTERRUPT REQUEST BIT
        RAL             ; PUT IT IN THE CARRY
        JCN A, INTRUPT  ; ANOTHER ONE HAS ARRIVED, SERVICE IT
        LCR             ; NO MORE INTERRUPTS, RESTORE THE CARRY
        RAR             ; SHIFT BIT 2 INTO THE CARRY
        RAR
        LD 6            ; RESTORE IR(6) OF BANK 0
        SBO             ; BACK TO BANK 0 TO GET THE ACC
```

BEST AVAILABLE COPY

```
        XCH 6           ; RESTORE THE ACC AND IR(6)
        BBS             ; RETURN FROM INTERRUPT
;
;   THIS SUBROUTINE IS USED TO LOAD AND COMPARE THE CRC COMPUTED
; THE CRC SUBROUTINE.  IT IS A LEVEL 6 SUBROUTINE WHICH USES
; IR(0-5,7) AND RETURNS IR(7) NON-ZERO IF THE CRC IS NOT EQUAL TO
; THE ONE ALREADY IN MEMORY.
;
CHKCRC: XCH 3           ; FIRST NIBBLE OF THE COMPUTED CRC
        JMS CRC2        ; SWAP AND COMPARE IT WITH MEMORY
        LD 4            ; SECOND NIBBLE OF THE CRC
        JMS CRC2
        LD 5            ; FINAL NIBBLE OF THE CRC
CRC2:   XCH 2           ; TEMPORARILY SAVE CRC IN IR(2)
        INC 1           ; POINT TO NEXT NIBBLE IN DATA RAM
        SRC 1
        RDM             ; SWAP CRC'S
        XCH 2
        WRM
        CLC             ; NOW COMPARE THEM
        SUB 2
        JCN 4, RETRNO   ; JUMP IF EQUAL
        INC 7           ; INCREMENT IR(7) - (INITIALLY ZERO)
RETRNO: BBL 0
;
;  THIS SUBROUTINE ALLOWS THE READING OF PROGRAM MEMORY BY THE
;  CHECKSUM PROCEDURE.
;
PROMO:  FIN 4           ; READ PROGRAM MEMORY
        JUN ADD23       ; ADD IT TO THE PARTIAL CHECKSUM
;
;   CHECK THE ID BITS PASSED THRU IR(2) AGAINST THE FIELD
; UNIT ID STORED IN IR(14).  THIS SUBROUTINE ASSUMES THE ID IS
; LOCATED IN THE LOWER THREE BITS, WITH THE MSB SET.  THIS IS
; A LEVEL 6 SUBROUTINE WHICH USES IR(0-2).
;
CHECKID: LD 14          ; LOAD THE ID AND CLOCK REGISTER
        CLC             ; REMOVE THE CLOCK BIT
        RAR
        CLC             ; NOW COMPARE THEM
        SUB 2
        JCN 4, RETRNO   ; RETURN IF THEY ARE EQUAL
        LDM 8           ; ERROR CODE FOR AN ID ERROR
;
;   THIS SUBROUTINE PROCESSES A RECEIVER ERROR, BY REGISTERING
; THE ERROR NUMBER PASSED THRU THE ACC, SETTING THE ACK/NAK
; REGISTER, IR(11), TO A NAK, AND SETTING THE IGNORE DATA/NEW
; DATA PRESENT BITS IN IR(10).  THIS IS A LEVEL 6 SUBROUTINE
; WHICH USES IR(0-1).
;
RCVERR: JMS ERROR       ; REGISTER THE ERROR
        XCH 11          ; SET THE ACK/NAK REGISTER TO NAK
        LDM 9           ; MASK FOR IGNORE DATA/NEW DATA PRESENT
;
;   THE FOLLOWING SUBROUTINE IS USED TO SET BITS IN IR(10).  THE
; BITS TO BE SET (AS A MASK) ARE PASSED IN THE ACC.  NO REGISTERS
; ARE DISTURBED  THIS IS A LEVEL 7 SUBROUTINE.
;
SETFLG: DIN             ; CRITICAL SECTION OF CODE SINCE IR(10)
        XCH 4           ;   IS BEING MODIFIED
        XCH 10
        OR4             ; NEW BITS ARE NOW SET, REPLACE IR(10)
        XCH 10
        XCH 4           ; RESTORE IR(4)
        EIN             ; END CRITICAL SECTION
RETCHK: BBL 0
;
;   THIS IS THE INITIALIZATION ROUTINE.  RESET AND PROGRAM THE
; ACIA, INITIALIZE THE DISPLAY REGISTERS, AND THE INDEX
; REGISTERS.  THIS IS WHERE THE PROGRAM BEGINS EXECUTION AFTER
; A POWER ON, MANUAL, OR WATCHDOG RESET.
;
INITIAL: FIM 4, E0      ; MASTER RESET THE ACIA
        JMS ACIA
        FIM 4, E1       ; DISABLE INTERRUPTS, 8 BITS, ODD PARITY
        JMS ACIA        ; WRITE TO THE ACIA
        LDM 9           ; ERROR CODE FOR A RESET
        DCL             ; USE RAM BANK 1
        JMS ERROR       ; SET THE DISPLAY REGISTERS
        FIM 2, A6       ; READ THE ID, CLOCK, AND THUMBWHEEL
        JMS READ
        LD 2            ; INITIALIZE THE THUMBWHEEL SWITCH VALUE
        XCH 13
        LD 3            ; SET THE ID BITS AND 4 HERTZ CLOCK
        XCH 14
        FIM 10, BD      ; SET IGNORE FRAME, UPDATE, NEW DATA, NAK
;
;   THIS IS THE MAIN PROGRAM LOOP.  THE PROGRAM RETURNS TO THIS
; POINT BEFORE IT SENDS A NEW BLOCK OF INFORMATION.  THE PROGRAM
; FIRST TRIES TO ESTABLISH HANDSHAKING WITH THE REMOTE CONTROL
; UNIT (SNOW WHITE) AS A NORMAL ORIGINATE MODEM.  IF THAT DOESN'T
; SUCCEED, THE ATTEMPT IS MADE TO FAKE HANDSHAKING THRU THE SELF-
; TEST FEATURE.  ASSUMING THAT HANDSHAKING IS ACCOMPLISHED
; SOMEHOW, THE PROGRAM CLEARS ANY PENDING DCD ERROR, AND WAITS
; FOR NEW DATA TO BE TRANSMITTED, OR FOR A 15 SECOND TIMER TO
; EXPIRE.  THE BLOCK IS THEN FORMED, FORMATTED, AND SENT.
;     SOME EXPLANATION OF THE MODEM IS PROBABLY IN ORDER HERE.
; THE MODEMS AT THE FIELD UNITS ARE *ORIGINATE* MODEMS, WHICH
; MEANS THEY WILL NOT TRANSMIT A CARRIER UNTIL THEY HAVE BEEN
; RECEIVING CARRIER FROM THE OTHER END OF THE LINE FOR ABOUT
; A HALF OF A SECOND.  HENCE, ON THE FIRST PASS THROUGH THE
; LOOP BELOW, THE MODEM IS PROGRAMMED TO BE A NORMAL ORIGINATE
; MODEM AND WE THEN WAIT UP TO 1.2 SECONDS FOR HANDSHAKING TO
; BE ESTABLISHED.  IF WE HAVE BEEN OPERATING FOR A WHILE
; (WE ARE NOT HERE BECAUSE OF A RESET WHICH EFFECTED THE
; MODEM), WE WILL PASS OUT OF THIS LOOP VERY QUICKLY (ABOUT
; 15 MILLISECONDS REQUIRED TO EXECUTE THE SUBROUTINE SIDC).
;    IF HANDSHAKING IS NOT ESTABLISHED WITHIN 1.2 SECONDS,
; SOMETHING IS WRONG.  IT COULD BE THE PHONE LINE, OUR
; RECEIVER, OR THE TRANSMITTER IN THE MODEM AT THE ATCT.  IN
; CASE IT IS THE LATTER, WE COULD STILL SEND MESSAGES TO THE
; TOWER (TELLING IT AMONG OTHER THINGS THAT WE ARE NOT HEARING
; ITS TRANSMITTER) IF WE COULD FAKE THE HANDSHAKING TO START
; OUR TRANSMITTER.  WE DO THIS BY SWITCHING TO AN ANSWER
; MODEM IN THE SELF-TEST MODE LONG ENOUGH TO START THE
; TRANSMITTER AND FOOL IT INTO THINKING IT HAS SUCEEDED IN
; HANDSHAKING WITH THE FAR END.  IF THIS FAILS, A PROBLEM
; EXISTS IN OUR MODEM AND WE SIGNAL ON THE SEVEN SEGMENT
; DISPLAY A MODEM ERROR.
;    ONCE WE HAVE FAKED THE HANDSHAKING IN THE ANSWER/SELF-
```

BEST AVAILABLE COPY

```
; TEST MODE, WE QUICKLY SWITCH TO ORIGINATE MODE, BUT STILL
; SELF-TESTING.  THIS PUTS OUR TRANSMITTER ON THE RIGHT
; FREQUENCY TO SEND INFORMATION TO THE ATCT.  SINCE THE MODEM
; IS STILL IN SELF-TEST MODE, WE WILL NOT GET A DCD (DATA
; CARRIER DETECT) ERROR.  NOTE THAT WE WILL NOT BE ABLE TO
; RECEIVE MESSAGES FROM THE ATCT UNTIL WE TRY ESTABLISHING
; HANDSHAKING AGAIN AFTER SENDING THE CURRENT BLOCK OF
; INFORMATION.
;
MAIN:   LDM D           ; SET MODEM TO NORMAL ORIGINATE MODE
MAIN1:  JMS MODEM       ; LOAD THE MODEM CONTROL REGISTER
        ADD 15          ; SET TIMER FOR 1.2 SEC FOR HANDSHAKING
        WR1             ; SAVE IN TRANSMITTER TIMER LOCATION
CTS:    JMS SIDC        ; SAMPLE INPUTS, ETC. ONCE WHILE WAITING
        JMS RDCNTL      ; READ THE ACIA STATUS REGISTER
        LD 3            ; CHECK THE CTS BIT
        RAL
        JCN 2, CTSOK    ; JUMP IF CLEAR-TO-SEND
        LD 0            ; LOAD THE TRANSMITTER TIMER
        SUB 15          ; COMPARE IT WITH THE CLOCK
        JCN C, CTS      ; LOOP BACK IF MORE TIME
        RD1             ; READ THE CURRENT MODEM CONTROL CONTENTS
        RAL             ; SET THE MSB TO FORM THE ERROR NUMBER
        LDM D           ; ERROR CODE IS D IF CURRENTLY ORGINATING
        JCN 2, MODERR
        LDM E           ; ERROR CODE IS E IF ANSWER SELF-TEST
MODERR: JMS ERROR       ; REGISTER DCD OR SELF-TEST ERROR
        LDM 2           ; SET MODEM TO ANSWER MODE, SELF-TEST
        JUN MAIN1       ; TRY IT AGAIN...
;
;  WE HAVE ESTABLISHED HANDSHAKING, EITHER WITH SNOW WHITE OR
;  O'RSELVES.  CHECK FOR A DCD ERROR, THEN SWITCH TO ORIGINATE/
;  SELF-TEST IF IN THE ANSWER MODE.
;
CTSOK:  RAL             ; SHIFT DCD INTO THE CARRY
        RD1             ; READ MODEM CONTROL WHILE SRC IS SET
        XCH 0           ; SAVE IT FOR LATER
        JCN 2, DCDOK    ; JUMP IF DCD IS NORMAL
        JMS RDFRAME     ; CLEAR THE DCD ERROR FLAG
DCDOK:  LD 0            ; LOAD CURRENT MODEM CONTROL CONTENTS
        RAL             ; CHECK FOR ANS/ORG
        JCN 2, ENRCVIN  ; JUMP IF ALREADY AN ORIGINATE MODEM
        LDM 9           ; SET ORIGINATE/SELF-TEST
        JMS MODEM
;
;  ENABLE THE RECEIVER INTERRUPT, SET THE 15 SECOND TIMER, AND
;  WAIT UNTIL THERE IS NEW DATA TO SEND.
;
ENRCVIN: JMS RCVONLY    ; ENABLE ACIA RECEIVER INTERRUPTS
        RD3             ; READ THE MINUTES CLOCK
        ADD 3           ; ADD 7 TO IT
        WR1             ; SET THE TRANSMIT TIMER LOCATION
OLDDATA: JMS SIDC       ; SCAN INPUTS, DISPLAY, AND CHECKSUM
        LD 10           ; GO IF NEW DATA IS PRESENT
        RAR
        JCN 2, GO       ; JUMP IF NEW DATA FLAG IS SET
        RD3             ; CHECK TO SEE IF 15 SECONDS HAVE ELAPSED
        SUB 0
        JCN C, OLDDATA  ; LOOP AGAIN IF TIME IS NOT UP
;
;  LOAD AND FORMAT A BLOCK OF DATA TO BE SENT IN DATA MEMORY
;  REGISTER 2.
;
GO:     FIM 4, 20       ; POINTER TO DATA RAM
        LD 14           ; THE SOURCE ID BITS ARE FIRST
        CLC
        RAR
        XCH 2
        CLB
        XCH 12          ; RESET COMMUNICATIONS STATUS
        XCH 3
        JMS WRITMEM     ; WRITE THE HEADER FRAME INTO MEMORY
        LDM 2           ; CALCULATE CRC COUNTER
        SUB 8
        DIN             ; CRITICAL SECTION OF CODE BEGINS HERE
        SB1             ; POP INTO BANK 1 TO USE IR(7)
        FIM 0, 20       ; STARTING POINTER FOR CRC
        JMS CRC         ; COMPUTE THE CRC FOR THIS BLOCK
        XCH 11          ; LOAD AND RESET THE ACK/NAK REGISTER
        WRM             ; PLACE IT INTO MEMORY
        JMS CHKCRC      ; LOAD THE CRC INTO MEMORY
        SB0             ; FINISHED WITH IR(7), BACK TO BANK 0
        CLB             ; SET POINTER TO FIRST FRAME TO BE SENT
        XCH 9
        LD 10           ; LEAVING THE IGNORE FRAME FLAG ALONE,
        RAL             ;   SET SENDING DATA, CLEAR OKAY TO
        LDM 8           ;   UPDATE AND NEW DATA PRESENT
        RAR
        XCH 10
        FIM 4, 41       ; ENABLE BOTH XMIT AND RECEIVE INTERRUPTS
        JMS ACIA        ; PERFORMS AN EIN BEFORE RETURNING
        ADD 15          ; SET MAXIMUM TIME TO SEND BLOCK
        WR1             ; SAVE IT IN THE TRANSMIT TIMER LOCATION
;
;  WAIT FOR THE BLOCK TO BE TRANSMITTED.  IF WE LOSE CARRIER OR
;  FOR SOME OTHER REASON THE BLOCK IS NOT SENT PROPERLY, THE
;  TIMER WILL RELEASE US FROM THIS LOOP.  NOTE THAT IF THIS
;  HAPPENS, IT WILL APPEAR THAT A WATCHDOG RESET OCCURED...
;
WAITXMT: JMS SIDC       ; SAMPLE INPUTS, ETC. WHILE WAITING
        LD 10           ; CHECK FOR DATA BEING SENT
        RAL
        RAL
        JCN A, MAIN     ; JUMP TO MAIN IF DONE SENDING
        RD1             ; CHECK THE TIMER
        CLC
        SUB 15
        JCN C, WAITXMT  ; LOOP IF TIME HAS NOT EXPIRED
        JUN INITIAL     ; PROBLEMS, START OVER AGAIN
;
;   THIS SUBROUTINE SUBTRACTS THE ZERO REFERENCE VALUE (IN 2'S
; COMPLEMENT) FROM THE DDM VALUE (ALSO 2'S COMPL ), CONVERTS IT
; TO SIGN-MAGNITUDE, FORMATS TO SEVEN MAGNITUDE BITS, AND STORES
; THE RESULT IN DATA RAM TO BE SENT.  IF THE NEW VALUE IS
; DIFFERENT FROM THE OLD VALUE, IT SETS THE NEW DATA PRESENT
; FLAG.  THIS IS A LEVEL 6 SUBROUTINE.
;   AFTER THE SUBROUTINE CALL TO READMEM, THE INDEX REGISTERS
; CONTAIN THE FOLLOWING INFORMATION:
;  IR(0) = 1
;  IR(1) = POINTER TO SECOND NIBBLE IN DATA RAM
```

BEST AVAILABLE COPY

```
;  IR(2) = LSN OF THE ZERO REFERENCE VALUE
;  IR(3) = MSN OF THE ZERO REFERENCE VALUE
;  IR(4) = LSN OF THE DDM VALUE
;  IR(5) = MSN OF THE DDM VALUE
;  IR(6) = LSB IS SIGN OF THE ZERO REFERENCE VALUE
;  IR(7) = MSB IS SIGN OF THE DDM VALUE
;
SUBO:   JMS READMEM     ; READ A DDM MAGNITUDE
        CLC             ; SUBTRACT LSN OF ZERO FROM THE DDM LSN
        LD 4
        SUB 2
        XCH 5           ; SAVE RESULTING LSN IN IR(5)
        CMC             ; SUBTRACT MSN OF ZERO FROM THE DDM MSN
        SUB 3
        XCH 4           ; SAVE RESULTING MSN IN IR(4)
        TCC             ; TO SUBTRACT SIGNS, FIRST SAVE BORROW
        XCH 7           ;   IN LSN OF IR(7)
        RAL             ; LOAD SIGN OF DDM INTO CARRY, SHIFT IN 0
        XCH 7           ; SIGN NIBBLE TO IR(7), BORROW TO ACC
        CMC             ; SUBTRACT SIGN OF DDM FROM REF. SIGN
        SUB 6           ; RESULT IS LSB IN ACC
        RAR             ; NEW SIGN IN THE CARRY, MAG IN IR(4-5)
        JCN A, SHIFT    ; JUMP IF THE RESULT IS POSITIVE
        JMS CMA54       ; CONVERT MAGNITUDE BEFORE FINISHING CY
        ISZ 5, SHIFT    ; INCREMENT IR(5-4)
        INC 4
SHIFT:  TCC             ; SIGN BACK TO ACC
        ADD 7           ; OR SIGN INTO IR(7)
        XCH 7           ; NEW SIGN SAFELY STORED IN IR(7)
        LD 4            ; LOAD MSN, SHIFT IN LEADING 1
        STC
        RAR
        XCH 5           ; SAVE THE MSN, LOAD THE LSN
        RAR
        XCH 4           ; SAVE THE LSN
        INC 0           ; CHANGE REGISTERS TO OLD MAGNITUDES
        SRC 1
        RDM             ; NOW SWAP THE NEW AND OLD LSN'S
        XCH 4
        WRM
        CLC             ; COMPARE THEM
        SUB 4
        JCN 4, SAMEDDM  ; JUMP IF NO CHANGE
        JMS NEWDATA
SAMEDDM LD 1            ; MOVE POINTER BACK TO THE OLD MSN
        DAC
        XCH 1
        SRC 1
        RDM             ; SWAP THE OLD AND NEW VALUES
        XCH 5
        WRM
        CLC             ; COMPARE THEM
        SUB 5
        JCN 4, RETRN1   ; JUMP IF THEY ARE THE SAME
NEWDATA LDM 1
        JUN SETFLG      ; SET NEW DATA PRESENT FLAG
;
;   THIS SUBROUTINE DEBOUNCES AN INPUT SIGNAL BY REQUIRING THAT
; IT HAVE THE SAME VALUE FOR FOUR SUCCESSIVE SAMPLES.  THE LATEST
; SAMPLE IS PASSED IN THE ACC.  A POINTER TO THE LOCATION IN DATA
; RAM OF THE LAST SAMPLE IS PASSED IN IR(0-1).  THE MAIN MEMORY
; CHARACTERS IN REGISTER 0 OF DATA RAM ARE USED FOR PREVIOUS
; SAMPLES, REGISTER 1 FOR THE DEBOUNCE COUNTERS, AND REGISTER 2
; FOR THE DEBOUNCED VALUES.  IR(2) IS USED FOR SCRATCH STORAGE,
; IR(1) IS INCREMENTED ON RETURN, IR(0) AND THE CARRY ARE LEFT AS
; 0.  THIS IS A LEVEL 6 SUBROUTINE WHICH USES IR(0-2).
;
DEBOUNC: SRC 1          ; SWAP ACC WITH CONTENTS OF MEMORY
        XCH 2
        RDM
        XCH 2
        WRM
        CLC             ; NOW COMPARE THE TWO SAMPLES
        SUB 2
        INC 0           ; MOVE THE POINTER TO DEBOUNCE COUNT
        SRC 1
        JCN C, DIFFER   ; JUMP IF THE SAMPLES ARE DIFFERENT
        RDM             ; THEY WERE THE SAME, INCREMENT THE COUNT
        IAC
        JCN C, RESET    ; JUMP TO STORE COUNT IF NOT NOW ZERO
        LD 10           ; CHECK TO SEE IF UPDATING IS FORBIDDEN
        RAR             ;   (2'S BIT OF IR(10))
        RAR
        JCN A, RESET1   ; IF 0, ILLEGAL TO CHANGE DEBOUNCED VALUE
        INC 0           ; NOT BLOCKED, MOVE POINTER TO DEBOUNCED
        SRC 1           ;   VALUE (REGISTER 2)
        CLC             ; COMPARE NEW AND OLD DEBOUNCED VALUES
        RDM             ; OLD DEBOUNCED VALUE
        SUB 2           ; NEW VALUE IS STORED IN IR(2)
        JCN 4, RESET1   ; THEY ARE THE SAME, DON'T BOTHER
        JMS NEWDATA     ; SET FLAG INDICATING NEW DATA PRESENT
        LD 2            ; WRITE NEW VALUE INTO MEMORY
        SKP
DIFFER: LDM C           ; RESET THE DEBOUNCE COUNT
RESET:  WRM             ; WRITE EITHER COUNT OR DEBOUNCED VALUE
RESET1: CLB             ; RESTORE IR(0) TO ZERO
        XCH 0
        INC 1           ; MOVE POINTER TO NEXT VALUE
RETRN1: BBL 0
;
;  THIS SUBROUTINE LOADS THE DIGITAL-TO-ANALOG CONVERTERS BY
;  READING THE SIGN-MAGNITUDE VALUE FROM MEMORY, THEN CONVERTING
;  TO OFFSET BINARY.  THIS IS A LEVEL 6 SUBROUTINE.
;
WRDAC:  INC 1           ; NEXT NIBBLE, PLEASE
        JMS READMEM     ; READ MEMORY POINTED TO BY IR(0-1)
        LD 7            ; GRAB THE SIGN BIT
        RAL
        XCH 7           ; SAVE THE REST OF THE SIGNS
        JCN 2, NEGATIV  ; JUMP AROUND 2'S COMPLEMENT IF NEGATIVE
        JMS CMA54       ; TAKE THE 2'S COMPLEMENT
        ISZ 5, NEGATIV
        INC 4
NEGATIV JMS CMA54       ; COMPENSATE FOR PRO-LOG'S NEGATIVE LOGIC
        INC 2           ; INCREMENT DEVICE ADDRESS POINTER
        JUN WRITE
;
;   THIS SUBROUTINE WRITES TO THE MODEM CONTROL REGISTER AND ALSO
; SAVES THE VALUE WRITTEN IN STATUS CHARACTER 1 OF REGISTER 2.
; THE FINAL SRC IS 3F.  THIS IS A LEVEL 6 SUBROUTINE WHICH USES
```

BEST AVAILABLE COPY

```
; IR(2-5).
;
MODEM:  FIM 2, 20       ; SAVE THE MODEM CONTROL CONTENTS FIRST
        SRC 3
        WR1
        XCH 5
        FIM 2, F1       ; MODEM CONTROL ADDRESS
        JUN WRITE       ; WRITE IT
;
;    THE FOLLOWING SUBROUTINE IS USED TO ALLOW THE READING OF
; PROGRAM MEMORY FOR PERFORMING A CHECKSUM ON THE PROMS.
;
PROM1:  FIN 4           ; READ PROGRAM MEMORY
        JUN ADD23
;
        .=200
;
;    THE FOLLOWING SUBROUTINE IS THE MAIN HOUSEKEEPING PORTION OF
; THIS PROGRAM.  IT SCANS AND DEBOUNCES THE INPUTS (DIGITAL AND
; ANALOG), REFRESHES AND UPDATES THE 7-SEGMENT DISPLAY, AND
; PERFORMS THE CHECKSUM OF PROGRAM MEMORY.  SIDC IS A MNEUMONIC
; FOR SCAN INPUTS, DISPLAY, AND CHECKSUM.
;    THIS IS A LEVEL 4 SUBROUTINE.
;
;
;  INDIRECT ADDRESSES FOR SUBROUTINES WHICH PERFORM VARIOUS
;  FUNCTIONS REQUIRED TO OPERATE THE ANALOG-TO-DIGITAL CONVERTER
;  SYSTEM FOR DIGITIZING THE DDM (DIFFERENCE IN DEPTH OF
;  MODULATION) SIGNALS.
;
        $STARTAD        ; SET MULTIPLEXER AND START CONVERSION
        $INCRADC        ; DUMMY STATE
        $WAITADC        ; WAIT FOR THE CONVERSION TO COMPLETE
        $READADC        ; READ AND PROCESS THE RESULTING VALUE
;
;  DETERMINE WHICH INPUTS TO SCAN BY USING THE VALUE OF THE ID
;  STRAPS ON THE INTERFACE CARD WHICH ARE STORED IN IR(14).
;
SIDC:   CLC             ; FIRST CHECK FOR THE LOC AND GLIDE SLOPE
        LDM A
        ADD 14          ; CARRY IS NOW ZERO IF LOC OR GS
        JCN A, SCANMUX  ; JUMP IF THIS IS THE LOC OR GLIDE SLOPE
        LDM 3           ; CLEAR THE CARRY IF THIS IS A MARKER
        ADD 14
        JCN A, SCANIFC  ; JUMP IF THIS IS A MARKER
;
;  THIS SUBROUTINE IS USED AT THE FAR-FIELD MONITOR TO READ THE
;  DIGITAL RELAY INPUTS (STATUS AND MAINTENANCE MONITOR SIGNALS)
;  AND TO OPERATE THE ANALOG-TO-DIGITAL CONVERTER WHICH READS
;  THE DDM SIGNALS.  THIS IS A LEVEL 4 SUBROUTINE.
;
SCANADC FIM 2, C4       ; SELECT ADC FLAGS AND RELAY INPUTS
        JMS READ01      ; READ THEM INTO IR(2-3)
        LD 2            ; LOAD THE RELAYS INTO ACC FOR DEBOUNCING
        FIM 0, 05       ; POINTER TO STORAGE FOR DEBOUNC
        JMS DEBOUNC     ; STORE AND DEBOUNCE THE DIGITAL INPUTS
;
;  IR(7) OF BANK 0 CONTAINS THE CURRENT STATE OF PROGRESS IN
;  SCANNING THE ANALOG INPUTS.  DEPENDING ON ITS VALUE, WE SHOULD
;  EITHER START A CONVERSION, WAIT FOR A CONVERSION TO FINISH,
;  OR READ AND PROCESS THE RESULTS OF A CONVERSION.
;
        LDM 3           ; MASK OFF LOWER TWO BITS OF THE STATE
        AN7             ;   NUMBER FOR INDIRECT ADDRESS
        XCH 1
        FIN 0           ; FETCH TASK ADDRESS
        JMS SUBJMP      ; FORGED INDIRECT SUBROUTINE JUMP
        LDM C           ; SIZE OF XMIT BLOCK LESS TRAILING FRAMES
        SKP             ; SCAN INTERFACE CARD INPUTS TOO
;
;   THIS SUBROUTINE SCANS THE RELAY CONTACTS ATTACHED TO THE
; INTERFACE CARD AND DEBOUNCES THEM.  IT IS A LEVEL 5 SUBROUTINE
; WHICH USES IR(0-3).
;
SCANIFC:LDM 4           ; NIBBLES IN BLOCK LESS TRAILING FRAMES
SCANI1: XCH 8           ; STORE THAT COUNT
        FIM 2, B6       ; ADDRESS OF INTERFACE INPUTS
        JMS READ        ; READ INPUTS INTO IR(2-3)
        FIM 0, 02       ; POINTER FOR DEBOUNCE
        LD 2            ; DEBOUNC THE FIRST NIBBLE
        RAL             ; BUT SET MSB FIRST
        STC
        RAR
        JMS DEBOUNC
        LD 3            ; DEBOUNC THE SECOND NIBBLE
        JMS DEBOUNC
        JUN CHECKS      ; JUMP TO THE CHECKSUM PROCEDURE
;
;   THIS SUBROUTINE USES THE INPUT MULTIPLEXOR CARD (FROM PRO-
; LOG) TO SCAN THE RELAY CONTACTS AT THE GLIDE SLOPE AND
; LOCALIZER.  MEMORY CHARACTERS 2 THRU 9 ARE READ FROM INPUTS 2
; THRU 7, THEN 0 AND 1 RESPECTIVELY.
;   THIS IS A LEVEL 5 SUBROUTINE USING IR(0-2).
;
SCANMUX:LDM A           ; NIBBLES IN BLOCK LESS TRAILING FRAMES
        XCH 8
        FIM 0, 02       ; POINT TO FIRST MEMORY CHARACTER
SCANM1: SRC 1           ; SELECTS RAM OUTPUT AND MUX INPUT PORTS
        LD 1
        WRR             ; SELECT THE MULTIPLEXOR INPUT PORT
        RDR             ; READ IT
        JMS DEBOUNC     ; STORE AND DEBOUNCE IT
        LDM A           ; CHECK FOR COMPLETION
        SUB 1           ; CARRY IS CLEARED BY DEBOUNC
        JCN C, SCANM1   ; LOOP THRU AGAIN IF NOT DONE
;
;   PERFORM A CHECKSUM ON ALL FIVE PROMS.  IN ORDER TO MINIMIZE
; THE IMPACT OF THE EXECUTION TIME OF A COMPLETE CHECKSUM, ONLY
; ONE-SIXTEENTH IS PERFORMED PER PASS.  THE ADDRESS POINTER AND
; PARTIAL SUM ARE STORED IN THE STATUS CHARACTERS 0, 2, AND 3 OF
; REGISTER 0 OF THE DATA RAM.  THIS ROUTINE TAKE APPROXIMATELY
; 12.8 MILLISECONDS TO EXECUTE.  THIS IS A LEVEL 6 ROUTINE WHICH
; USES IR(0-3).
;
CHECKS: FIM 0, 10       ; READ RAM AND RESTORE INDEX REGISTERS
        SRC 1
        RD0             ; RESTORE HIGH NIBBLE OF THE ADDRESS
        XCH 0
        RD2             ; RESTORE HIGH NIBBLE OF PARTIAL SUM
        XCH 2
```

BEST AVAILABLE COPY

```
        RD3             ; RESTORE LOW NIBBLE OF PARTIAL SUM
        XCH 3
AGAIN:  JMS PROM0       ; FETCH PROM 0 CONTENTS, ADD TO IR(2-3)
        JMS PROM1       ; PROM 1
        FIN 4           ; PROM 2
        JMS ADD23
        JMS PROM3       ; PROM 3
;
;  THE CONTENTS OF PROM 4 WERE FETCHED AFTER ADDING PROM 3'S
;  CONTENTS INTO THE PARTIAL SUM.  NOW CALL ADD23 TO ADD PROM
;  PROM 4'S CONTENTS.
;
        JMS ADD23       ; PROM 4
        ISZ 1, AGAIN    ; INCREMENT THE LSN OF THE ADDRESS
        ISZ 0, SAVCHK   ; INCREMENT THE MSN OF THE ADDRESS
        LD 2            ; COMPLETED WHOLE THING, CHECK RESULT
        JCN C, CHKERR   ; JUMP IF AN ERROR
        LD 3
        JCN 4, SAVCHK   ; JUMP IF OKAY
CHKERR: LDM F           ; ERROR, NON-ZERO CHECKSUM
        JMS ERROR       ; REGISTER THE ERROR
        FIM 4, 10       ; RESTORE THE SRC
        SRC 5
        FIM 2, 0        ; RESET THE CHECKSUM PARTIAL SUM
SAVCHK: LD 0            ; SAVE THE ADDRESS POINTER (MSN)
        WR0
        LD 2            ; SAVE THE HIGH NIBBLE OF THE PARTIAL SUM
        WR2
        LD 3            ; SAVE THE LOW NIBBLE OF THE PARTIAL SUM
        WR3
;
;   CALL THE DISPLAY DRIVER.
;
        JMS DISPLAY
        FIM 2, 87       ; ADDRESS OF 7-SEGMENT DISPLAY
        JMS WRITE       ; WRITE NEW CHARACTER
;
;   READ AND DEBOUNCE THE ID STRAPS, 4 HERTZ CLOCK, AND THE
; THUMBWHEEL SWITCH.  SINCE THE 4 HERTZ CLOCK SHOULD HAVE CLEAN
; EDGES, ONLY LIMITED DEBOUNCING IS PERFORMED ON IT.
;
        FIM 2, A6       ; ADDRESS OF THE INTERFACE INPUT PORT
        JMS READ        ; READ INTO IR(2-3), SRC IS 20 WHEN DONE
        RD2             ; SWAP THE THUMBWHEEL SWITCH SAMPLES
        XCH 2
        WR2
        CLC             ; NOW COMPARE THEM
        SUB 2
        JCN C, MOVING   ; JUMP IF THE SWITCH IS MOVING
        RD3             ; READ THE DEBOUNCE COUNT
        IAC             ; INCREMENT IT
        SKP             ; NOW STORE IT
MOVING: LDM 1           ; RESET THE DEBOUNCE COUNT
        WR3             ; STORE THE DEBOUNCE COUNT
        JCN C, CLOCK    ; JUMP IF NOT A NEW DEBOUNCED VALUE
        RD2             ; PUT NEW DEBOUNCED VALUE INTO IR(13)
        XCH 13
CLOCK:  LD 3            ; LOAD THE VALUE JUST READ FOR ID, CLOCK
        XCH 14          ; STORE IN IR(14)
        LD 3            ; CHECK THE CLOCK FOR THE PASSING OF 1/8
        RAR             ;    OF A SECOND.  IF PASSED, RESET THE
        JCN 2, TICK     ;    WATCHDOG AND UPDATE THE IR(15) CLOCK
        FIM 2, 07       ; RESET THE WATCHDOG, RE-WRITE THE 7-SEG
        JMS WRITE
        ISZ 15, TICK    ; CLOCK TICK, INCREMENT EIGHTHS
        FIM 2, 30       ; INCREMENT 2 SECOND CLOCK
        SRC 3
        RD3
        IAC
        WR3
        XCH 0           ; SAVE THE 2 SECOND COUNTER VALUE
        RD2             ; READ RECEIVER TIMER
        CLC             ; COMPARE THEM
        SUB 0
        JCN C, TICK     ; OKAY IF THEY ARE DIFFERENT
        RD2             ; ADVANCE TIMER BY 2 SECONDS
        IAC
        WR2
        LDM C           ; TIMES UP, REGISTER ERROR
        JMS RCVERR      ; REGISTER ERROR, SET NEW DATA FLAG
TICK:   FIM 0, 30       ; RETURN XMIT CLOCK IN IR(0)
        SRC 1
        RD1
        XCH 0
        BBL 0           ; CLOCK HAS TICKED, RETURN TO MAIN
;
;  THIS SUBROUTINE COMPUTES THE CYCLIC REDUNDANCY CHECK REMAINDER
;  FOR CONSECUTIVE NIBBLES IN DATA RAM, BEGINNING WITH THE
;  CHARACTER POINTED TO BY IR(0-1).  THE CRC WAS RESEARCHED BY
;  C. R. WILLIAMS AND IS A SHORTENED CYCLIC HAMMING CODE (47,38).
;  THE GENERATING POLYNOMIAL IS X^9 + X^4 + 1.  IT DETECTS:
;       . ALL SINGLE ERRORS
;       . ALL DOUBLE ERRORS
;       . ALL BURSTS OF LENGTH 9 OR LESS
;       . 99.6 PER CENT OF ALL BURSTS OF LENGTH 10
;       . 99.8 PER CENT OF ALL BURSTS OF LENGTH 11 OR GREATER
;  THE SUBROUTINE USES IR(0-7) AND LEAVES THE CRC REMAINDER IN
;  IR(3,4,5) IN THE FORMAT USED FOR MULTIPLE FRAME BLOCKS.  THIS
;  A LEVEL 7 SUBROUTINE.  THE NUMBER OF NIBBLES TO BE SCANNED IS
;  INDICATED BY THE NUMBER PASSED THRU THE ACC WHICH SHOULD EQUAL
;  THE 2'S COMPLEMENT OF (2-N).  THE CARRY IS CLEARED UPON
;  RETURN.
;
CRC:    XCH 7           ; SAVE THE COUNT IN IR(7), SAVE OLD IR(7)
;
;  INITIALIZE IR(4-5) WITH THE FIRST TWO CHARACTERS.  SET THE
;  CARRY TO ZERO TO START WITH UNCOMPLEMENTED SHIFT.
;
        JMS READMEM     ; READ THE FIRST TWO CHARACTERS
        INC 1           ; UPDATE THE POINTER TO NEXT CHARACTER
        CLC             ; INITIALIZE THE CARRY/LINK TO ZERO
;
;  SET COUNT FOR THE NUMBER OF BITS TO BE SHIFTED FROM IR(3)
;  (4 BITS) BEFORE READING THE NEXT CHARACTER.  THEN READ THE
;  CHARACTER AND STORE IT IN IR(3).  WATCH CAREFULLY...
;
CRCLOOP:FIM 2, C0       ; IR(2) IS BIT COUNTER
        SRC 1
        RDM             ; READ THE CHARACTER
;
```

BEST AVAILABLE COPY

```
;  IF THE CARRY IS NOW ZERO, LEFT SHIFT IR(3,5,4).  IF IT IS ONE,
;  DO THE SHIFT AND PERFORM THE EXCLUSIVE-OR (XOR) BY INVERTING
;  BITS 4 AND 0.
;
CRCLP2: JCN 2, CRCCOMP  ; CHECK THE CARRY
        RAL             ; LEFT SHIFT WITHOUT COMPLEMENTING
        XCH 3
        LD 5
        RAL
        JUN CRCLD5
CRCCOMP:RAL             ; LEFT SHIFT COMPLEMENTING BITS 4 AND 0
        XCH 3
        CMC
        LD 5
        RAL
        CMC
CRCLD5: XCH 5           ; FINISH BOTH SHIFTS
        LD 4
        RAL
        XCH 4
        LD 3            ; READY ACC FOR NEXT SHIFT
;
;  LOOP THRU THE FOUR BITS OF EACH CHARACTER AND THRU ALL OF THE
;  CHARACTERS TO BE USED.
;
        ISZ 2, CRCLP2   ; SHIFT THRU ALL OF IR(3)
        INC 1
        ISZ 7, CRCLOOP  ; READ THRU ALL OF THE CHARACTERS
;
;  THE CARRY AND IR(5-4) NOW CONTAIN THE REMAINDER.  SAVE THE
;  CARRY IN IR(3).
;
        TCC             ; MOVE THE CARRY TO THE ACC
        XCH 3           ; SAVE IT TEMPORARILY
        LD 4            ; PACK CARRY AND MSB OF IR(4) INTO IR(5)
        RAL
        LD 3
        RAL             ; CARRY AND MSB PACKED INTO ACC
        XCH 3
        LDM 8           ; SET MSB OF IR(4)
        OR4
        XCH 4
        SRC 1           ; (NEXT LOCATION IS THE ACK/NAK)
        BBL A           ; CODE FOR AN ACK
;
;   IN ORDER TO DECREASE THE POSSIBILITY THAT GARBAGE WILL BE
; LOADED INTO THE MODEM CONTROL REGISTER DURING A RESET, THE
; PROPER VALUE IS WRITTEN TO THE OUTPUT DATA BUS DURING ALL READS
; FOR A NORMAL ORIGINATE MODEM.  THIS IS A LEVEL 7 SUBROUTINE
; WHICH USES IR(2-3).
;
ADDRO:  CLB             ; SELECT PORT 0
        XCH 2           ; DON'T LOSE IR(2)
        SRC 3
        XCH 2           ; RESTORE IR(2)
        LDM D           ; PLACE NORMAL ORG ON LSN OF OUTPUT BUS
;
;   THIS SUBROUTINE IS USED TO ADDRESS THE INTERFACE CARD OR
; DEVICES CONNECTED TO IT.  THE ADDRESS TO BE WRITTEN IS PLACED
; IN IR(2-3) WHICH IS RETURNED UNDISTURBED.  THE FINAL SRC POINTS
; TO 3-IR(2).  IF THE MSB OF THE ADDRESS IS A ZERO (MODEM ADDRESS
; SEMAPHORE), A 'NOP' IS WRITTEN ONTO THE ADDRESS BUS TO
; INSURE A RISING EDGE (AFTER INVERSION) FOR PROPER OPERATION.
; SINCE THE WRITING OF AN ADDRESS MAY BE PART OF A CRITICAL
; SECTION OF CODE, INTERRUPTS ARE DISABLED AND LEFT DISABLED ON
; EXIT.  A 'NOP' IS LOADED INTO THE ACC ON RETURN TO FACILITATE
; DESELECTING A DEVICE IF A WRITE OPERATION IS BEING PERFORMED.
; THIS IS A LEVEL 7 SUBROUTINE.
;
ADDRESS:DIN             ; CRITICAL CODE BEGINS HERE
        WRR             ; SAVE A WORD
        LDM 2           ; LSN OF ADDRESS BUS IS PORT 2
        XCH 2
        SRC 3
        XCH 3           ; LSN IS NOW IN THE ACC
        WRR             ; WRITE LSN OF ADDRESS
        INC 2           ; MSN OF ADDRESS BUS IS PORT 3
        SRC 3
        XCH 3           ; REPLACE THE LSN OF THE ADDRESS
        XCH 2
        LD 2            ; LOAD THE MSN OF THE ADDRESS
        RAL             ; CHECK FOR MODEM ADDRESS
        STC
        RAR
        WRR
        LD 2            ; LOAD MSN OF ADDRESS AGAIN
        WRR             ; NOW WRITE IT TO THE ADDRESS BUS
        BBL A           ; FACILITATE DESELECT WHEN WRITING
;
;   THIS SUBROUTINE IS USED TO READ FROM THE INTERFACE CARD OR
; FROM DEVICES CONNECTED TO IT.  THE DEVICE WHOSE ADDRESS IS IN
; IR(2-3) IS READ, AND THE VALUE RETURNED IN IR(2-3).  THE ENTRY
; POINT *RDCNTL* IS USED TO READ THE CONTROL REGISTER OF THE
; ACIA.  THIS IS A LEVEL 6 SUBROUTINE WHICH USES IR(2-3).  THE
; FINAL SRC VALUE IS 20.
;
RDCNTL: FIM 2, 76       ; READ THE ACIA CONTROL REGISTER
READ:   JMS ADDRO       ; WRITE THE ADDRESS TO PORTS 2 AND 3
        RDR             ; READ THE MSN FROM PORT 3
        FIM 2, 20
        SRC 3
        XCH 2           ; STORE THE MSN
        RDR             ; READ THE LSN
        EIN             ; END CRITICAL SECTION (BEGAN IN ADDRESS)
        XCH 3
        BBL 0
;
;  THIS IS A PATCH TO THE SUBROUTINE ADD23
;
PATCH:  LD 4
        ADD 2
        XCH 2
        JUN PATCH1
;
        .=300
;
;   THIS SUBROUTINE IS THE 7-SEGMENT LED DISPLAY DRIVER.  THE
; DISPLAYS ARE DESIGNED TO ALLOW QUICK CHECKING AND SERVICING OF
; THE ILS COMMUNICATIONS WITHOUT THE USE OF ANY TEST EQUIPMENT.
; THE POSITION OF THE THUMBWHEEL SWITCH LOCATED ON THE FRONT
```

BEST AVAILABLE COPY

```
; PANEL INDICATES WHICH DISPLAY IS REQUESTED.  THE DIFFERENT
; DISPLAYS AVAILABLE ARE:
;    0 - DISPLAY THE COMMUNICATIONS EQUIPMENT STATUS (ENGLISH)
;    1 - LAMP TEST, LIGHT ALL THE SEGMENTS
;    2 - DISPLAY THE RECEIVED DATA
;    3 - VALUE OF DDM 1 (SIGN-MAGNITUDE)
;    4 - SET ANALOG ZERO FOR A/D AND D/A'S, VALUE DISPLAYED IS
;        THE SIGN AND 7 MSB'S AT THE FFM, AND THE VALUE BEING
;        LOADED AT THE LOC.
;    5 - THE RELAY OUTPUTS
;    6 - THE SECOND FRAME BEING SENT
;    7 - THE THIRD FRAME BEING SENT
;    8 - THE FOURTH FRAME BEING SENT
;    9 - THE FIFTH FRAME BEING SENT
; THIS IS A LEVEL 5 SUBROUTINE WHICH USES IR(0-5).  THE VALUE TO
; BE DISPLAYED IS RETURNED IN IR(4-5).
;
;
;  INDIRECT ADDRESSES FOR THE TEN POSSIBLE DISPLAYS.
;
        $NORMAL         ; NORMAL DISPLAY OF ENGLISH MESSAGES
        $RETRN3         ; LAMP TEST
        $READMEM        ; RECEIVED DATA
        $DDM1           ; DISPLAY DDM 1 VALUE BEING SENT
        $SETZERO        ; ALLOW SETTING OF ANALOG ZEROES
        $RELVALU        ; RELAY VALUES
        $DATA1          ; FRAME 2
        $DATA2          ; FRAME 3
        $DATA3          ; FRAME 4
        $DATA4          ; FRAME 5
;
;  FIRST SET THE CARRY BIT IF THIS IS THE LOCALIZER, THEN JUMP TO
;  THE PROPER DISPLAY PROCESSOR DEPENDING ON THE VALUE OF THE
;  DEBOUNCED THUMBWHEEL SWITCH.
;
DISPLAY CLC             ; IR(14) WILL BE 2 OR 3 IF THIS IS THE
        LDM C           ;   LOCALIZER, SO ADD 12 TO IR(14) TO SET
        ADD 14          ;   THE CARRY IF NOT THE LOCALIZER...
        LD 13           ; LOAD THE VALUE OF THE THUMBWHEEL SWITCH
        FIM 0, 0        ; FORM INDIRECT ADDRESS
        XCH 1
        FIN 2           ; LOAD ADDRESS TO DISPLAY PROCESS
        FIM 0, 3E       ; RECEIVED DATA POINTER
        FIM 4, 00       ; LAMP TEST
        JIN 3           ; JUMP TO IT
;
;  THE NORMAL DISPLAY CONSISTS OF CRIPTIC ENGLISH MESSAGES WHICH
;  DESCRIBE THE HIGHEST PRIORITY ERROR CONDITION (IF ANY) HAVING
;  OCCURED SINCE THE COMPLETION OF THE LAST MESSAGE.  IF NO
;  ERROR CONDITIONS HAVE OCCURED, THE MESSAGE IS '-GO.'.
;  EACH LETTER IN THE MESSAGE IS DISPLAYED FOR APPROXIMATELY ONE
;  SECOND.  LETTERS ARE SEPARATED BY BLANKING THE DISPLAY FOR AN
;  EIGHTH OF A SECOND.  MESSAGES ARE SEPARATED BY BLANKS.  THE
;  OCCURANCE OF AN ERROR OF HIGHER PRIORITY THAN THE ONE BEING
;  DISPLAYED WILL CAUSE THE NEW MESSAGE TO INTERRUPT THE OLD (SEE
;  THE SUBROUTINE *ERROR*).  THIS SUBROUTINE HAS FOUR POSSIBLE
;  STATES:  0 - BLANK DISPLAY, RESET POINTER TO '-GO.', 1 -
;  DISPLAY A DASH, 2 - BLANK THE DISPLAY BETWEEN CHARACTERS, AND
;  3 - DISPLAY A LETTER.  THE TRANSITIONS BETWEEN STATES OCCUR
;  AT THE END A PREDETERMINED INTERVAL, NORMALLY BY INCREASING
;  THE STATE NUMBER BY 1.  THE TRANSITION FROM STATE 3 IS TO
;  STATE 0 IF THE LAST LETTER IS BEING DISPLAYED, TO STATE 2
;  OTHERWISE.
;
NORMAL: FIM 0, 0
        SRC 1           ; DISPLAY POINTERS ARE IN REGISTER 0
        RD0             ; MSN OF THE DISPLAY POINTER
        XCH 0
        RD1             ; LSN OF THE DISPLAY POINTER
        XCH 1
        RD3             ; READ THE CURRENT STATE NUMBER
        CLC
        RAR             ; EVEN STATE IMPLIES BLANK DISPLAY
        JCN 2, NOTBLK   ; JUMP IF NOT BLANKED
        FIM 4, FF       ; BLANK THE DISPLAY FOR INTERLETTER GAP
        JCN C, INCSTA   ; STATE 0 IMPLIES RESET POINTER
        FIM 0, $MSG0    ; RESET POINTER TO 'ALLS FINE'
        JUN INCSTA
NOTBLK: JCN C, LETTER   ; STATE 1 IS DASH, 3 IS A LETTER
        FIM 4, FB       ; LOAD A DASH TO SIGNAL THE START
INCSTA: RD3             ; PREPARE NEXT STATE NUMBER
        IAC
        JUN NSTATE      ; CHECK CLOCK TO SEE IF TIME FOR NEXT
LETTER: FIN 4           ; LOAD CHARACTER INTO IR(4-5), SET ACC=0
        LDM 0
        OR4             ; IF THE DP IS LIT, THIS IS THE LAST CHAR
        CMA             ; ACC NOW ZERO IF DP IS DARK
        JCN C, LOAD0    ; JUMP IF DP LIT, NEXT STATE IS ZERO
        LDM 2           ; DP WAS DARK, NEXT STATE IS TWO
        ISZ 1, NSTATE   ; INCREMENT TENTATIVE POINTER
        INC 0
        SKP
LOAD0:  CLB             ; TENTATIVE NEXT STATE IS ZERO
NSTATE: XCH 2           ; TEMPORARILY SAVE NEXT STATE
        CLC
        RD2             ; COMPARE DISPLAY TIMER AND THE CLOCK
        SUB 15
        JCN C, RETRN3   ; IF NOT EQUAL, NOT TIME FOR NEXT STATE
        LD 0            ; MOVE TO NEXT STATE
        WR0             ; HIGH NIBBLE OF DISPLAY POINTER
        LD 1
        WR1             ; LOW NIBBLE OF DISPLAY POINTER
        LD 2
        WR3             ; NEXT STATE NUMBER
        RAR             ; COMPUTE DELAY FROM STATE NUMBER
        LDM 1           ;   (EVEN STATE=1 TICK, ODD STATE=9)
        RAR             ; ACC=8 FOR EVEN STATE, 0 FOR ODD; CY=1
        ADD 15          ; ADD IN THE CURRENT CLOCK VALUE
        WR2             ; SET DISPLAY TIMER
RETRN3: BBL 0
;
;  DISPLAY THE VALUE BEING SENT OR RECEIVED FOR DDM 1
;
DDM1:   CMC             ; SET IR(0) TO 2 IF FFM, 3 IF LOC
        LDM 1
        RAL
        FIM 0, 4        ; SET IR(1) TO 4
        XCH 0           ; NOW HAVE THE ADDRESS OF THE SIGN
        JMS READMEM     ; READ IT
        LD 4            ; PUT THE SIGN INTO THE CARRY
```

BEST AVAILABLE COPY

```
        RAL
        RAL
        INC 1           ; IR(0-1) NOW POINTS TO THE MAGNITUDE
        JMS READMEM     ; READ THE MAGNITUDE
        CMC
        RAR             ; ACC = 8 IF SIGN WAS POSITIVE
        ADD 4           ; COMPLEMENT MSB OF IR(4) IF POSITIVE
        XCH 4           ; DONE, NOW DISPLAY IT
        BBL 0
;
;  ALLOW THE SETTING OF THE ANALOG ZERO.  AT THE FAR-FIELD
;  MONITOR, DISPLAY THE ZERO VALUE BEING READ SO THAT THE ZERO
;  MAY BE ADJUSTED.  AT THE LOCALIZER, ALTERNATELY PLACE PLUS
;  AND MINUS FULL SCALE ON THE D/A'S TO ALLOW SETTING OF THEIR
;  GAIN.
;
SETZERO: JCN F, SETDAC  ; JUMP IF THIS IS THE LOCALIZER
        FIM 0, 1C       ; READ THE LAST REFERENCE VALUE
        JMS READMEM     ; READ THE MAGNITUDE OF THE REFERENCE
        INC 1           ; NEXT NIBBLE IS THE SIGN BIT
        SRC 1
        RDM             ; THE SIGN BIT IS THE LSB
        RAR             ; RIGHT SHIFT THE RESULT
        LD 5            ; RIGHT SHIFT THE SIGN INTO THE MSN
        RAR
        XCH 4           ; SAVE THE MSN, LOAD THE LSN
        RAR             ; RIGHT SHIFT THE LSN
        XCH 5           ; SAVE THE RESULTING LSN
        BBL 0           ; VALUE DISPLAYED IS 2'S COMPLEMENT
SETDAC: SRC 1           ; ALLOW SETTING OF DAC POTENTIOMETERS
        RD3             ; READ THE 2 SECOND COUNTER/CLOCK
        RAL             ; PUT THE MSB INTO THE CARRY
        JCN A, STDAC1   ; IF 0-14 SECONDS, LEAVE IR(4-5)=00
        FIM 4, FF       ; IF 16-30 SECONDS, SET IR(4-5)=FF
STDAC1: FIM 2, C7       ; ADDRESS OF DAC 1
        JMS WRITE       ; WRITE DATA TO OUTPUT BUS, THEN ADDRESS
        JMS LOADDAC     ; WRITE THE SAME TO DAC 2
LOADDAC: INC 2          ; WRITE IT AGAIN TO DAC 3
        JUN WRITE
;
;  DISPLAY THE VALUES BEING WRITTEN TO THE RELAYS
;
RELVALU: FIM 0, 32      ; READ THE LOCALIZER RELAYS
        JMS READMEM
        LD 4            ; ADD THE LSB OF THIS NIBBLE TO THE CYCLE
        RAR
        RD0
        RAL
        XCH 4           ; PUT THE CYCLE INTO THE MSN
        BBL 0
;
;  DISPLAY THE SECOND FRAME TO BE SENT
;
DATA1:  LDM 2
        SKP
;
;  DISPLAY THE THIRD FRAME TO BE SENT
;
DATA2:  LDM 4
        SKP
;
;  DISPLAY THE FOURTH FRAME TO BE SENT
;
DATA3:  LDM 6
        SKP
;
;  DISPLAY THE FIFTH FRAME TO BE SENT
;
DATA4:  LDM 8
        FIM 0, 20
        XCH 1
;
;   THIS SUBROUTINE READS THE TWO CONSECUTIVE LOCATIONS IN DATA
; RAM POINTED TO BY IR(0-1) AND LEAVES THE RESULT IN IR(4-5).
; THIS IS A LEVEL 7 SUBROUTINE.
;
READMEM: SRC 1          ; POINT TO FIRST LOCATION
        RDM             ; READ THE FIRST NIBBLE
        XCH 4           ; STORE IT IN IR(4)
        INC 1           ; MOVE POINTER
        SRC 1
        RDM             ; READ THE SECOND CHARACTER FROM RAM
        XCH 5           ; SAVE IT IN IR(5)
        BBL 0
;
;  SEVEN-SEGMENT DISPLAY ENGLISH MESSAGE TEXTS
;
MSG0:   #60             ; GO
        #98
MSG1:   #B0             ; DDM 3
        #B0
        #65
        #50
MSG2:   #B0             ; DDM 2
        #B0
        #65
        #11
MSG3:   #B0             ; DDM 1
        #B0
        #65
        #D6
MSG4:   #22             ; ADC,
        #B0
        #0D
MSG5:   #B8             ; OVER
        #E3
        #29
        #9B
MSG6:   #68             ; STOP
        #AB
        #B8
        #03
MSG7:   #2D             ; CRC
        #BB
        #0D
MSG8:   #23             ; PARITY
        #22
        #BB
        #BF
        #AB
```

BEST AVAILABLE COPY

```
        #83
MSG9:   #BB             ; RESET
        #29
        #68
        #29
        #8B
MSGB:   #BF             ; ID
        #90
MSGC:   #BA             ; NO DATA
        #B3
        #FF
        #B0
        #22
        #AB
        #02
MSGD:   #AB             ; TONE
        #B5
        #BA
        #09
MSGE:   #65             ; MODEM
        #B3
        #B0
        #29
        #45
MSGF:   #23             ; PROM
        #BB
        #B8
        #45
;
;   THIS SUBROUTINE READS A FRAME FROM THE ACIA, AND STORES IT
; BOTH IN IR(2-3) AND LOCATIONS 3E-3F IN DATA RAM.  THE
; SUBROUTINE READ WAS NOT USED TO AVOID STACK OVERFLOW WHILE
; PROCESSING AN INTERRUPT.  THIS IS A LEVEL 6 SUBROUTINE WHICH
; USES IR(2-5)
;
RDFRAME FIM 2, 74       ; ADDRESS OF THE ACIA'S RECEIVER BUFFER
        JMS ADDR0       ; SELECT THE ACIA RECEIVER BUFFER
        RD3             ; SET THE RECEIVER TIMER FOR 30 SECONDS
        DAC
        WR2
        RDR             ; READ THE MSN OF THE ACIA BUFFER
        CMA             ; COMPLEMENT TO NEGATE NEGATIVE LOGIC
        XCH 2           ; SAVE IT
        FIM 4, 2E       ; SELECT INPUT PORT 2
        SRC 5
        RDR             ; READ AND SAVE THE MSN OF THE BUFFER
        CMA
        XCH 3
        INC 4           ; MOVE THE POINTER TO 3E
;
;   THIS SUBROUTINE WRITES THE CONTENTS OF IR(2-3) INTO THE
; CONSECUTIVE LOCATIONS IN DATA RAM POINTED TO BY IR(4-5).  THIS
; IS A LEVEL 7 SUBROUTINE WHICH USES IR(2-5).  ALL THE REGISTERS
; ARE LEFT UNDISTURBED EXCEPT IR(5) WHICH IS INCREMENTED.
;
WRITMEM SRC 5           ; SELECT FIRST NIBBLE IN DATA RAM
        LD 2            ; WRITE THE MSN
        WRM
        INC 5           ; INCREMENT IR(4-5)
        SRC 5
        LD 3            ; WRITE THE LSN
        WRM
        BBL 0
;
;  THIS SUBROUTINE COMPLEMENTS IR(4-5).  IT IS A LEVEL 7 ROUTINE.
;
CMA54:  LD 5
        CMA
        XCH 5
        LD 4
        CMA
        XCH 4
        BBL 0
;
;   SUBROUTINE FOR READING PROGRAM MEMORY TO PERFORM A CHECKSUM.
;
PROM3:  FIN 4           ; READ PROGRAM MEMORY
        JUN ADD23       ; ADD IT TO THE CHECKSUM
;
        .=400
;
;  INDIRECT POINTERS TO THE MESSAGE TEXTS
;
        $MSG0
        $MSG1
        $MSG2
        $MSG3
        $MSG4
        $MSG5
        $MSG6
        $MSG7
        $MSG8
        $MSG9
        NOP             ; CHECKSUM FOR ALL FIVE PROMS
        $MSGB
        $MSGC
        $MSGD
        $MSGE
        $MSGF
;
;   THIS SUBROUTINE IS USED TO SET THE ERROR REGISTER, IR(12),
; FOR SENDING TO THE REMOTE CONTROL, AND TO START A NEW DISPLAY.
; THE ERROR NUMBER IS PASSED THRU THE ACC.  THIS IS A LEVEL 7
; SUBROUTINE WHICH USES IR(0-1).
;
ERROR:  XCH 1           ; LOAD IR(0-1) WITH (0,ERROR NUMBER)
        CLB
        XCH 0           ; IR(0-1) NOW POINTS TO INDIRECT ADDRESS
        LD 12           ; COMPARE NEW ERROR # WITH OLD IN IR(12)
        SUB 1           ; NEW ERROR NUMBER
        JCN 2, ERROR1   ; A BORROW IMPLIES NEW # IS HIGHER
        LD 1            ; LOAD NEW ERROR NUMBER INTO IR(12)
        XCH 12
ERROR1: SRC 1           ; DISPLAY POINTERS ARE IN RAM REGISTER 0
        FIN 0           ; FETCH MESSAGE TEXT ADDRESS
        RD1             ; COMPARE IT WITH THE CURRENT ONE
        CLC             ;   BY SUBTRACTING THE NEW FROM THE OLD
        SUB 1
        CMC
        RD0
```

BEST AVAILABLE COPY

```
        SUB 0
        JCN 2, RETRNAK  ; BORROW IMPLIES NEW HAS HIGHER PRIORITY
        LD 1            ; LOAD THE NEW DISPLAY POINTER
        WR1             ; LSN OF THE DISPLAY POINTER
        LD 0
        WR0             ; MSN OF THE DISPLAY POINTER
        LDM 8           ; LOAD TIMER WITH ONE SECOND
        ADD 15
        WR2
        LDM 1           ; LOAD THE STATE (DISPLAY DASH)
        WR3
RETRNAK: BBL D          ; LOAD ACC WITH THE CODE FOR A NAK
;
;   THIS SUBROUTINE CONTROLS THE OPERATION OF THE A/D CONVERTER,
; READS THE CONVERTER OUTPUT, AND PREPARES IT FOR TRANSMISSION.
; THERE ARE 3 DDM SIGNALS WHICH ARE READ AND A REFERENCE VALUE
; WHICH INDICATES THE CONVERTED VALUE OF ZERO VOLTS.  THIS
; REFERENCE IS SUBTRACTED FROM THE DDM VALUES TO MAKE THE
; READINGS LESS SENSITIVE TO THE DRIFT OF THE A/D CONVERTER.  THE ;
; A/D FLAGS HELD IN IR(3) WHEN THIS SUBROUTINE IS CALLED HAVE THE ;
; FOLLOWING MEANING:
;   BIT 1 = 0
;   BIT 2 = 0 IF RESULT IS NEGATIVE, 1 IF POSITIVE
;   BIT 4 = 0 IF OVERFLOW OCCURED DURING CONVERSION
;   BIT 8 (MSB) = 1 IF THE CONVERSION IS COMPLETE (EOC)
;
SUBJMP: LD 3            ; LOAD THE FLAG NIBBLE
        RAL             ; LEAVE THE EOC FLAG IN THE CARRY
        JIN 1           ; JUMP TO THE PROPER SERVICE ROUTINE
;
;  SELECT AN ANALOG INPUT USING THE ANALOG SWITCH ON THE ADC
;  CARD, THEN SIGNAL THE ADC TO START CONVERSION.  THE PATTERN
;  WRITTEN TO THE ANALOG SWITCH CORRESPONDS TO A ZERO BEING
;  SHIFTED RIGHT TO LEFT THRU A FIELD OF ONES.  IN OTHER WORDS
;  STATE 0 WILL WRITE '1110', STATE 4 WILL WRITE '1101', STATE 8
;  WILL WRITE '1011', AND STATE C WILL WRITE '0111'.  IR(7) HOLDS
;  THE CURRENT STATE NUMBER, IR(6) IS USED TO TIME THE AMOUNT OF
;  TIME USED IN CONVERSION.  THIS IS A LEVEL 5 SUBROUTINE.
;
STARTAD: LD 7           ; LOAD THE STATE NUMBER (0,4,8,C)
        RAR             ; THE CARRY (EOC) IS ASSUMED EQUAL TO 1
        XCH 5           ; IR(5) NOW EQUALS (8,10,12,14)
        LDM F           ; CARRY IS NOW EQUAL TO ZERO
STARTO  RAR             ; ROTATE ACC-CY UNTIL 0 IS PLACED RIGHT
        INC 5           ; INCREMENT IR(5) BY TWO
        ISZ 5, STARTO
        XCH 5           ; STORE FOR USE BY THE WRITE SUBROUTINE
        FIM 2, D7       ; ADDRESS OF ANALOG SWITCH
        JMS WRITE       ; WRITE USING THE INTERFACE CARD
        XCH 6           ; LOAD THE CONVERSION TIMER
        INC 7           ; SET NEXT STATE
        JUN LOADDAC     ; START THE CONVERSION
;
;  WAIT FOR THE ANALOG-TO-DIGITAL CONVERTER TO FINISH CONVERSION.
;  IF THE END OF CONVERSION SIGNAL IS NOT RECEIVED IN A
;  REASONABLE AMOUNT OF TIME (STORED IN IR(6)), SIGNAL AN ADC
;  ERROR AND ADVANCE TO THE NEXT STATE ANYWAY.  THIS IS A LEVEL
;  6 SUBROUTINE.
;
WAITADC JCN 2, INCRADC  ; INCREMENT THE CURRENT STATE IF DONE
        ISZ 6, RETADC   ; INCREMENT TIMER
        JMS ADCERR      ; TIMED-OUT, SIGNAL ADC ERROR
INCRADC: INC 7          ; GO TO NEXT STATE
RETADC: BBL 0
;
;  READ THE RESULT OF THE CONVERSION, CONVERT IT TO TWO'S
;  COMPLEMENT, AND STORE IT IN DATA RAM.  WHEN READ, THE VALUE
;  OF THE CONVERSION IS REPRESENTED IN SIGN-MAGNITUDE FORMAT AS
;  FOLLOWS:  1 = +, 0 = -, MAGNITUDE BITS COMPLEMENTED.  THIS IS
;  A LEVEL 5 SUBROUTINE.
;
READADC: XCH 6          ; SAVE THE OVERFLOW AND SIGN IN IR(6)
        FIM 2, C6       ; READ THE MAGNITUDE BITS
        JMS READ01      ; READ FROM PORTS 0-1 INTO IR(2-3)
        LD 6            ; PUT THE OVERFLOW FLAG IN THE CARRY
        RAL
        XCH 6           ; SAVE THE SIGN IN IR(6) AGAIN
        JCN 2, ADCOK    ; JUMP IF NO OVERFLOW

;  THE FOLLOWING 5 STATEMENTS WERE REMOVED FROM THE PROGRAM
;  TO DELETE THE DDM1, DDM2, AND DDM3 MESSAGES WHICH OCCURED
;  ON OVERFLIGHTS AND WERE FILLING THE 'LOG' AT THE CONTROL
;  TOWER.  THE STATEMENTS WERE REPLACED BY NOP'S.
;
;       LD 7            ; COMPUTE THE ERROR NUMBER (WHICH DDM)
;       CMA
;       RAR
;       RAR
;       JMS ERROR       ; DDM 1 = 3, DDM 2 = 2, DDM 3 = 1
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        FIM 2, F0       ; PUT ALMOST FULL SCALE IN IR(2-3)
;
;  AT THIS POINT, THE SIGN IS IN THE MSB OF IR(6), (1=+, 0=-),
;  THE COMPLEMENTED MAGNITUDE MSN IS IN IR(3), AND THE
;  COMPLEMENTED LSN IS IN IR(2).  NOW COMPUTE THE NORMAL 2'S
;  COMPLEMENT FORM FROM THIS SIGN-MAGNITUDE DATA AND SAVE IN
;  DATA RAM REGISTER 1.
;
ADCOK:  LD 6            ; RESTORE THE ACC (CONTAINS SIGN BIT)
        RAL             ; MOVE THE SIGN INTO THE CARRY
        CMC             ; CARRY NOW HOLDS CORRECT SIGN (0 = +)
        JCN A, PLUS     ; JUMP IF A POSITIVE NUMBER
        ISZ 2, STORE    ; INCREMENT IR(2-3),CY
        ISZ 3, STORE
        CMC
        JUN STORE       ; NOW STORE THE RESULT
PLUS:   LD 3            ; COMPLEMENT IR(2-3)
        CMA
        XCH 3           ; FINISHED WITH THE MSN
        LD 2            ; NOW COMPLEMENT THE LSN
        CMA
        XCH 2
STORE:  FIM 4, 14       ; POINTER TO THE SIGN NIBBLE IN DATA RAM
        SRC 5
        RDM             ; READ IT
```

BEST AVAILABLE COPY

```
        RAL             ; SHIFT IN THE LATEST SIGN
        WRM             ; STORE IT
        XCH 6           ; SAVE IT FOR POSSIBLE USE LATER
        LD 7            ; COMPUTE THE MAGNITUDE DESTINATION
        CLC             ;   DDM 1 = 16-17, DDM 2 = 18-19,
        RAR             ;   DDM 3 = 1A-1B, ZERO = 1C-1D
        ADD 5
        XCH 5
        JMS WRITMEM     ; STORE IR(2-3) IN DATA RAM
        ISZ 7, RETADC   ; IF THIS IS NOT THE ZERO REFERENCE, JUMP
;
;  ALL 3 DDM SIGNALS AND THE ZERO REFERENCE HAVE BEEN READ AND
;  STORED IN DATA RAM.  NOW SUBTRACT EACH OF THE DDM SIGNALS FROM
;  THE ZERO REFERENCE VALUE, CONVERT THEM BACK TO SIGN-MAGNITUDE,
;  AND LOAD THEM INTO DATA REGISTER 2 TO BE SENT.
;
        INC 5           ; SAVE REFERENCE SIGN IN LOCATION 1E
        SRC 5
        LD 6
        WRM
        OR4             ; SET WHAT WILL BECOME MSB OF SIGNS
        XCH 7           ; LOAD THE DDM SIGNS INTO IR(7)
        LD 10           ; NOW CHECK FOR OK TO UPDATE DATA REG. 2
        RAR
        RAR
        JCN A, CHECKO   ; JUMP IF NOT ALRIGHT
        FIM 0, 16       ; POINTER TO DDM 1
        JMS SUBO        ; SUBTRACT FROM ZERO, CONVERT, AND STORE
        FIM 0, 18       ; POINTER TO DDM 2
        JMS SUBO
        FIM 0, 1A       ; POINTER TO DDM 3
        JMS SUBO
        FIM 0, 24       ; NOW STORE THE SIGN NIBBLE TO BE SENT
        SRC 1
        RDM             ; SWAP THE OLD AND NEW VALUES
        XCH 7
        WRM             ; NOW WRITE THE NEW VALUE
        CLC             ; COMPARE THE OLD AND THE NEW
        SUB 7
        JCN 4, CHECKO   ; JUMP IF THEY ARE THE SAME
        JMS NEWDATA     ; SET THE NEW DATA PRESENT FLAG
CHECKO: CLB             ; RESET THE ADC STATE REGISTER, IR(7)
        XCH 7
        LDM 1           ; MASK OFF THE SIGN BIT OF IR(6)
        AN6             ; WE REQUIRE THE 4 MAGNITUDE MSB'S OF THE
        ADD 3           ;   REFERENCE VALUE TO BE ZERO...
        JCN 4, RETADC   ; RETURN IF ZERO IS CLOSE ENOUGH
ADCERR: LDM 4
        JUN ERROR
;
;   THIS SUBROUTINE IS USED TO CHECK THE CYCLE COMMAND NIBBLE FOR
; VALID DATA.  IF THE NIBBLE CONTAINS A VALID VALUE AND THE
; IGNORE DATA FLAG IS OFF, THE VALUE IS DEBOUNCED AND THEN
; WRITTEN TO THE CYCLE RELAY.  THIS IS A LEVEL 6 SUBROUTINE WHICH
; USES IR(0-5).
;
CYCLE:  LD 10           ; CHECK THE IGNORE DATA FLAG
        RAL             ; SHIFT IT INTO THE CARRY BIT
        JCN 2, RETCYC   ; SKIP THE REST IF IT IS ON
        CLC             ; COMPARE THE CYCLE WITH 'CYCLE'
        LDM A           ; LEGAL VALUE FOR CYCLE COMMAND
        SUB 3
        JCN 4, LEGALC   ; JUMP IF IT IS A LEGAL VALUE
        CLC             ; COMPARE IT WITH 'NO CYCLE'
        LDM 5           ; LEGAL VALUE FOR NO CYCLE COMMAND
        SUB 3
        JCN 4, LEGAL
DATAERR: LDM 7          ; ERROR CODE FOR DATA ERROR
        JUN RCVERR      ; PROCESS IT AND RETURN
;
;  A LEGAL VALUE HAS BEEN RECEIVED IN A LEGAL BLOCK OF DATA, NOW
;  DEBOUNCE IT AND WRITE THE DEBOUNCED VALUE TO THE CYCLE RELAY.
;  TO BE DEBOUNCED, THE CYCLE COMMAND MUST APPEAR TWICE IN A ROW.
;
LEGALC: JMS NEWDATA     ; FORCE SENDING OF A NEW BLOCK
LEGAL:  LDM 3           ; LAST CYCLE SAMPLE IS IN REGISTER 3
        XCH 2
        SRC 3
        RDO             ; SWAP THE TWO MOST RECENT SAMPLES
        XCH 5
        LD 3
        WRO
        CLC             ; NOW COMPARE THEM
        SUB 5
        JCN C, RETC1    ; IF NOT EQUAL, FORCE CONFIRMING BLOCK
        FIM 2, 9F       ; RELAY ADDRESS
;
;   THIS SUBROUTINE IS USED TO WRITE TO DEVICES ON OR CONNECTED
; TO THE INTERFACE CARD.  THE CONTENTS OF IR(4-5) ARE WRITTEN TO
; THE DEVICE WHOSE ADDRESS IS IN IR(2-3).  THE CONTENTS OF THESE
; REGISTERS ARE NOT CHANGED.  THIS IS A LEVEL 6 SUBROUTINE.  THE
; FINAL SRC VALUE IS 3-IR(2).
;
WRITE:  LDM 0           ; PORT 0 GETS THE LSN OF DATA
        XCH 4
        SRC 5
        XCH 5           ; LSN OF DATA IS NOW IN THE ACC
        DIN             ; CRITICAL SECTION OF CODE BEGINS HERE
        WRR
        INC 4
        SRC 5           ; MSN OF DATA GOES TO PORT 1
        XCH 5           ; PUT THE LSN BACK IN IR(5), MSN IN IR(4)
        XCH 4
        LD 4            ; WRITE THE MSN (DONE BY ADDRESS)
        JMS ADDRESS     ; WRITE THE ADDRESS TO PORTS 2 AND 3
        WRR             ; DESELECT DEVICE BY WRITING A 'NOP'
        EIN             ; END CRITICAL SECTION OF CODE
RETCYC: BBL A           ; TIMER VALUE FOR A-TO-D CONVERTER
RETC1:  JUN NEWDATA     ; FORCE NEW BLOCK BY SENDING ONE
;
;   THIS SUBROUTINE IS USED TO READ FROM THE ADC BOARD, USING
; THE ADDRESSING FEATURES OF THE INTERFACE CARD.  IT IS A
; DIFFERENT ROUTINE BECAUSE THE INPUT IS TO BE READ FROM IR(0-1)
; INSTEAD OF IR(2-3).  BECAUSE THE MODEM ADDRESS IS NOT USED
; WITH THIS SUBROUTINE, THERE IS NO NEED TO WRITE TO THE OUTPUT
; DATA BUS BEFORE SETTING THE ADDRESS.  THIS SUBROUTINE EXPECTS
; THE ADDRESS TO BE READ IN IR(2-3), AND RETURNS THE VALUE READ
; IN IR(3) [MSN] AND IR(2) [LSN].  THIS IS A LEVEL 6 SUBROUTINE.
;
READ01: JMS ADDR0       ; SET THE ADDRESS BUS
```

BEST AVAILABLE COPY

```
        FIM 2, 0        ; POINTER TO INPUT PORT FOR LSN
        SRC 3
        RDR             ; READ THE LSN
        INC 2
        SRC 3           ; POINTER TO INPUT PORT FOR MSN
        XCH 2           ; SAVE THE LSN
        RDR             ; READ THE MSN
        EIN             ; END OF CRITICAL SECTION
        XCH 3           ; SAVE THE MSN
        BBL 0
;
;  CLEAR THE IGNORE FRAME BIT FROM THE FLAGS IN IR(10).
;
CLRIGNR:LDM 7
;
;   THE FOLLOWING SUBROUTINE IS USED TO CLEAR BITS IN IR(10).  A
; MASK OF THE BITS TO BE LEFT UNDISTURBED IS PASSED THRU THE ACC.
; NO REGISTERS ARE DISTURBED.  THIS IS A LEVEL 7 SUBROUTINE.
;
CLRFLG: DIN             ; CRITICAL SECTION OF CODE SINCE IR(10)
        XCH 6           ;   IS BEING MODIFIED
        XCH 10
        AN6             ; BITS ARE NOW CLEARED, REPLACE IR(10)
        XCH 10
        XCH 6           ; RESTORE IR(6)
        EIN             ; END CRITICAL SECTION
        BBL 0
;
;   THIS SUBROUTINE WRITES THE CONTENTS OF IR(3-4) TO THE ACIA
; CONTROL REGISTER.  THIS IS A LEVEL 6 ROUTINE WHICH USES IR(2-5)
;
RCVONLY:FIM 4, 61       ; RECEIVER INTERRUPTS ONLY
ACIA:   FIM 2, 77       ; ADDRESS OF THE ACIA
        JUN WRITE       ; WRITE IT
;
;   THIS SUBROUTINE IS USED BY THE CHECKSUM PROCEDURE TO ADD THE
; CONTENTS OF IR(4-5) TO THE CURRENT VALUE OF THE CHECKSUM IN
; IR(2-3).  IT IS A LEVEL 7 SUBROUTINE.
;
ADD23:  CLC             ; ADD THE LSN'S, IR(3) & IR(5)
        LD 5
        ADD 3
        XCH 3           ; PLACE THE RESULT IN IR(3)
        JUN PATCH       ; ADD THE MSN'S, IR(2) & IR(4)
;
;   THE FOLLOWING *FIN* INSTRUCTION HAS NO DIRECT BEARING ON THE
; SUBROUTINE ADD23.  IT IS USED BY THE CHECKSUM ROUTINE IN A
; BACKWARDS WAY TO PERFORM THE CHECKSUM ON PROM 4.
;
PATCH1: FIN 4           ; FETCH FROM PROGRAM MEMORY
        BBL D           ; RETURN MASK FOR FINDING DECIMAL POINT
;
        END
;
```

APPENDIX F

How to Identify and Insert PROM's in 4040 Units

The microprocessor cards in the field units have five programmable read only memories which hold the program for the microprocessor. These PROM's are Intel type C1702A. Each of these units is labeled with a letter and a number pencilled on the white ceramic substrate. These letters and numbers are used to identify the individual PROM's and their contents.

There is a single program which works in all of the field units. This program is divided into five parts, each part is held in a single PROM, on each of the microprocessor cards. The PROM's containing the first fifth of the program have a number 0 on them. Likewise, the PROM's containing the second fifth of the program have a number 1 on them and so on through the last fifth of the program which always contains the number 4 on top of the PROM. The letters are used to distinguish one PROM from another, but they have no relationship to what is contained on the PROM. Hence, one microprocessor card might contain PROM's A0, A1, A2, A3, and A4. Another card might have PROM's E0, E1, F2, H3, and D4. Both cards will work properly and contain the proper program assuming the proper PROM number is inserted in the proper socket.

If one holds the microprocessor card component side up, with the gold-plated fingers on the left and the nylon card ejector in the upper righthand corner, the five PROM sockets are in the lower edge of the card. The five PROM's should be placed in the sockets from left to right PROM 0, PROM 1, PROM 2, PROM 3, and PROM 4. Pin 1 of the PROM's which is marked by a dot or a bump in the corner. Pin 1 should always be placed in the upper lefthand corner of the socket. Be sure that all 24 pins fit into the sockets and are not bent out of the way.

The PROM's have been inserted correctly into the microprocessor card if when plugged into the field unit and turned on the message RESET appears or in fact any message except PROM. If no messages appear or if the PROM message appears, check to be sure that all the PROM's are correctly inserted into their sockets and that the 0, 1, 2, 3, and 4 PROM's have been inserted from left to right on the bottom of the card. The PROM error message on the seven-segment display indicates that either one of the PROM's is defective or that an incorrect PROM has been inserted in one of the sockets. See Appendix L for further information.

APPENDIX G

LOCALIZER D/A BOARD

BOARD EDGE CONNECTOR PIN ASSIGNMENTS

PIN(S)	ASSIGNMENT
1 & 2	5 volt supply (VCC). A conservative estimate of maximum current draw is 165 ma.
3 & 4	Ground. Reference for all power supplies (+15, +5, -10) as well as DDM signals.
5 & 6	-10 volt supply. A conservative estimate of maximum current draw is 5 ma.
7 & 8	+15 volt supply. A conservative estimate of maximum current draw is 10 ma.
9 & 10	Not used.
11	D/A 1. A negative going TTL pulse, of duration not less than 5 µs., will load the "DAC register" of the digital-to-analog converter servicing DDM1. Thus, immediately after the leading (falling) edge of this pulse, the DDM1 analog output voltage (pin 31) will likely change to a new value. Loading of this input: Sink 1.6 ma, and source 40 µa.
12	D/A 2. See comments for pin 11.
13	D/A 3. See comments for pin 11.
14	Relay register strobe. A negative going TTL pulse, of duration not less than 15 ns., will load the relay register with the five low-order bits of the data bus. If data bus leads are labeled DB2 (LSB) through DB9 (MSB), then the following data bus assignments are true:

DB	FFM SIGNAL
4	FF_{MM} (Monitor mismatch)
5	FF_{PE} (Pwr./Temp. Fail)
3	FF_{S} (CAT II shutdown)
2	FF_{SA} (Shutdown alert)
6	FF_{BY} (Monitor bypass)

A TTL logical "1" loaded into a register bit will cause the closure of its corresponding relay contacts. Loading of this input: Sink 1.5 ma., and source 20 µa.

PIN(S)	ASSIGNMENT
15	DB9. Most significant bit of 8-bit data bus. A TTL "1" here would result in a "1" being placed in the MSB of a D/A register. Loading of this input: 3 μa source and sink.
16	Not used.
17	DB7. See comments for pin 15.
18	Not used.
19	DB6. A TTL "1" here would result in a "1" being placed in a D/A register or the closure of the FF_{BY} (monitor bypassed) relay. Loading of this input: Sink 0.25 ma., and source 13 μa.
20	Not used.
21	DB8. See comments for pin 15.
22	Not used.
23	DB5. See comments for pin 19. Services D/A registers as well as FF_{PE} (Pwr./Temp. Fail) relay.
24	Not used.
25	DB3. See comments for pin 19. Services D/A registers as well as FF_{S} (CAT II shutdown) relay.
26	Not used.
27	DB2. Least significant bit of 8-bit data bus. See comments for pin 19. Services D/A registers as well as FF_{SA} (Shutdown alert) relay.
28	Not used.
29	DB4. See comments for pin 19. Services D/A registers as well as FF_{MM} (Monitor mismatch) relay.
30	Not used.
31	Analog voltage output for DDM1. To be fed to localizer ILS. Output is short circuit protected. External voltages between -10 and +30 volts may be accidentally applied to this terminal without circuit damage. Output is designed to drive meter circuitry having a DC resistance of at least 47 K.Ω.
32	DDM2 Analog output voltage. See comments for pin 31.
33	DDM 3 Analog output voltage. See comments for pin 31.
34	Not used.

PIN(S)	ASSIGNMENT
35 & 36	DDM COM. Reference line for all three DDM analog outputs. Is to be passed to localizer ILS. Is tied (on this board) to power supply ground (pins 3 & 4).
37 – 40	Not used.
41	Relay contact servicing FF_{BY} (Monitor bypass) signal. Is to be fed to localizer ILS. Contact current between this pin and "relay common" (pins 47 & 48) cannot exceed 0.5 amp. Voltage between contacts cannot exceed 100 VDC.
42	Relay contact servicing FF_{PE} (Pwr./Temp. fail) signal. See comments for pin 41.
43	Relay contact servicing FF_{MM} (Monitor mismatch) signal. See comments for pin 41.
44	Relay contact servicing FF_{S} (CAT II shutdown) signal. See comments for pin 41.
45	Relay contact servicing FF_{SA} (Shutdown alert) signal. See comments for pin 41.
46	Not used.
47 & 48	FFM COM. All relays have a single contact tied to this pin. Is to be fed to localizer ILS.
49 – 56	Not used.

APPENDIX H

AD7522 D/A CONVERTER

CODE MEANING AND POT ADJUSTMENTS

The conversion scheme carried out by the AD7522, and its associated operational amplifier circuitry, is as according to the following table. Here we assume that both associated potentiometers have been pre-adjusted as described below.

DATA BUS WORD DB9 ---- DB2	DDM OUTPUT
11111111	-1.016 volts (=-1.024+0.008)
11111110	-1.008 volts
⋮	
10000001	-0.008 volts
10000000	0.0 volts
01111111	+0.008 volts
⋮	
00000001	1.016 volts
00000000	1.024 volts

Thus, conversion to this code from sign-plus-magnitude format could be carried out as follows.

1. If the S + M word is negative (MSB = "1"), its value may be fed to the AD7522 directly.

2. If the S + M word is non-negative (MSB = "0"), subtract its value from 10000000 using 8-bit arithmetic. The result may then be passed to the AD7522. Example:

```
 10000000
-00010111      (S + M format)
 01101001      (feed to AD7522)
```

Note that S + M format contains two representations for zero: 00000000 and 10000000. Thus, one code has been wasted. This turns out to mean that full-scale positive at the DDM output will be 1.016 volts, even though the 1.024 volt state is available from the AD7522.

To set the gain (and indirectly the zero position) of an AD7522 section, carry out the following procedure:

1. Load the DAC register with 11111111 (all ones). Then, using a digital voltmeter tied to the DDM output, adjust potentiometer "P-" until this output reads -1.016 volts.

2. Load the DAC register with 00000000 (all zeros). Then adjust potentiometer "P+" to obtain 1.024 volts at the DDM output.

The DAC register may now be loaded with 10000000. The DDM output should be within 2-3 millivolts of zero.

When loading the DDM output, bear in mind that this output has a 100 ohm output resistance.

APPENDIX I

QUICK-CHANGE BOXES AT FIELD UNITS AND THE ATCT

The quick-change boxes enable the ILS signalling system to be operated either under the Purdue tone-signalling system or under the original DC-signalling system. The basic philosophy of the system is shown in Fig. I-1 . The quick-change block physically switches all lines used in either signalling system. This includes maintenance monitor signals because they, too, use a power supply within the ILS as one of their buried cable lines. Therefore, these lines must also be switched to insure complete isolation of the ILS when operating under the Purdue tone system.

Figure I-2 shows the quick-change box set-up at a field unit location. The Telco box lines are tied to the male connector shown in the figure. The ILS equipment and the Purdue equipment each have lines going to their respective female connectors. Physically reswitching the male connector to the proper female connector will complete the desired signalling system circuit and completely isolate the other signalling system from transients entering through the Telco lines.

The setup at the ATCT is similar to the one at a field unit. There are six connections to switch at the ATCT quick change box to duplicate the switching done at the field units. This is accomplished with a male connector and two female connectors per field unit so that a field unit may be operated on either tone or DC signalling independent of any other field unit.

The maintenance monitor signals and DDM signals are hard-wired to the tone-signalling system and cannot be easily changed back to the DC system. This decision was made with the approval of representatives of the FAA, NAFEC, and Eastern Region to simplify installation and maintenance.

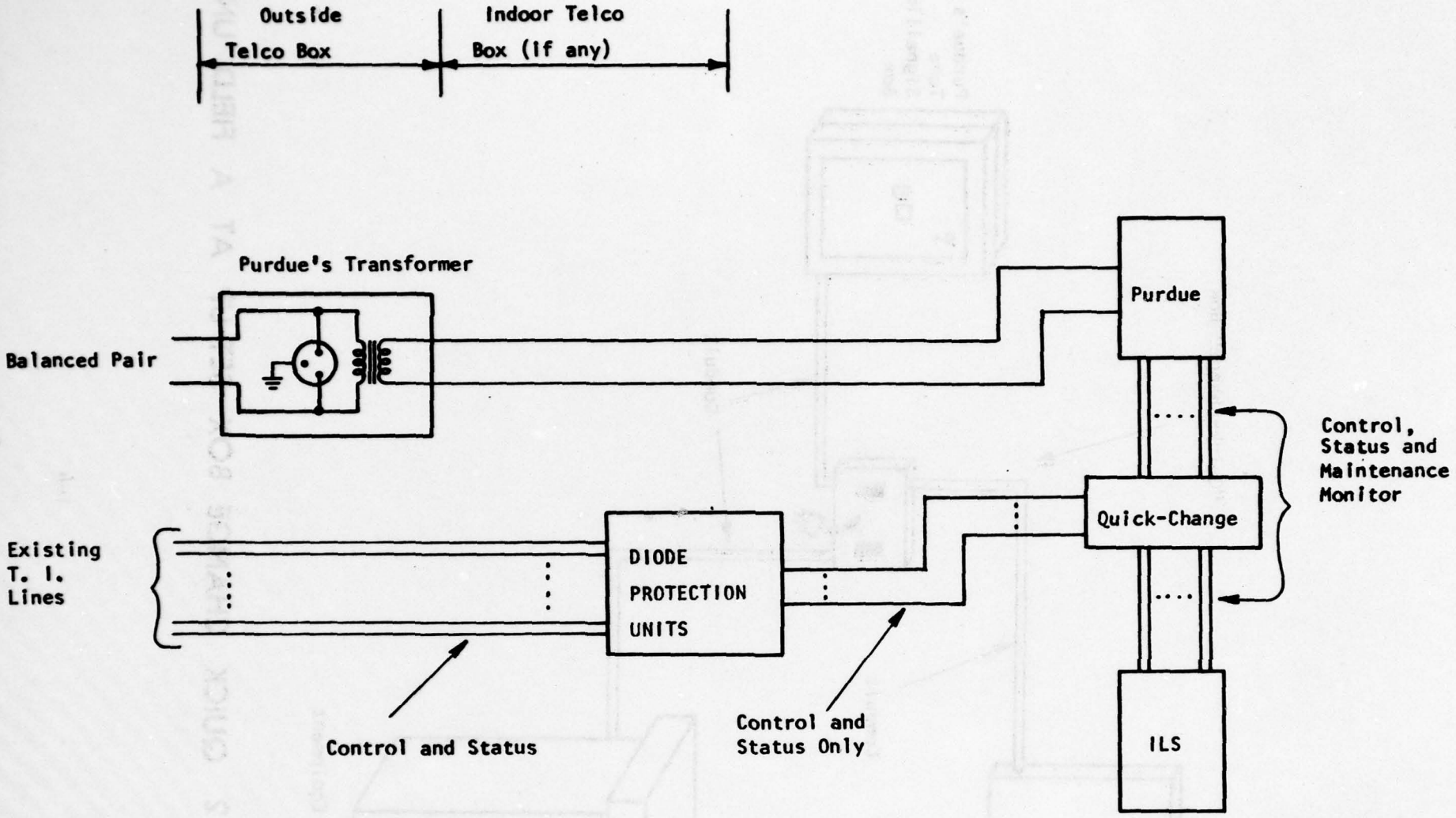


FIG. I- 1 QUICK CHANGE BOX FUNCTION

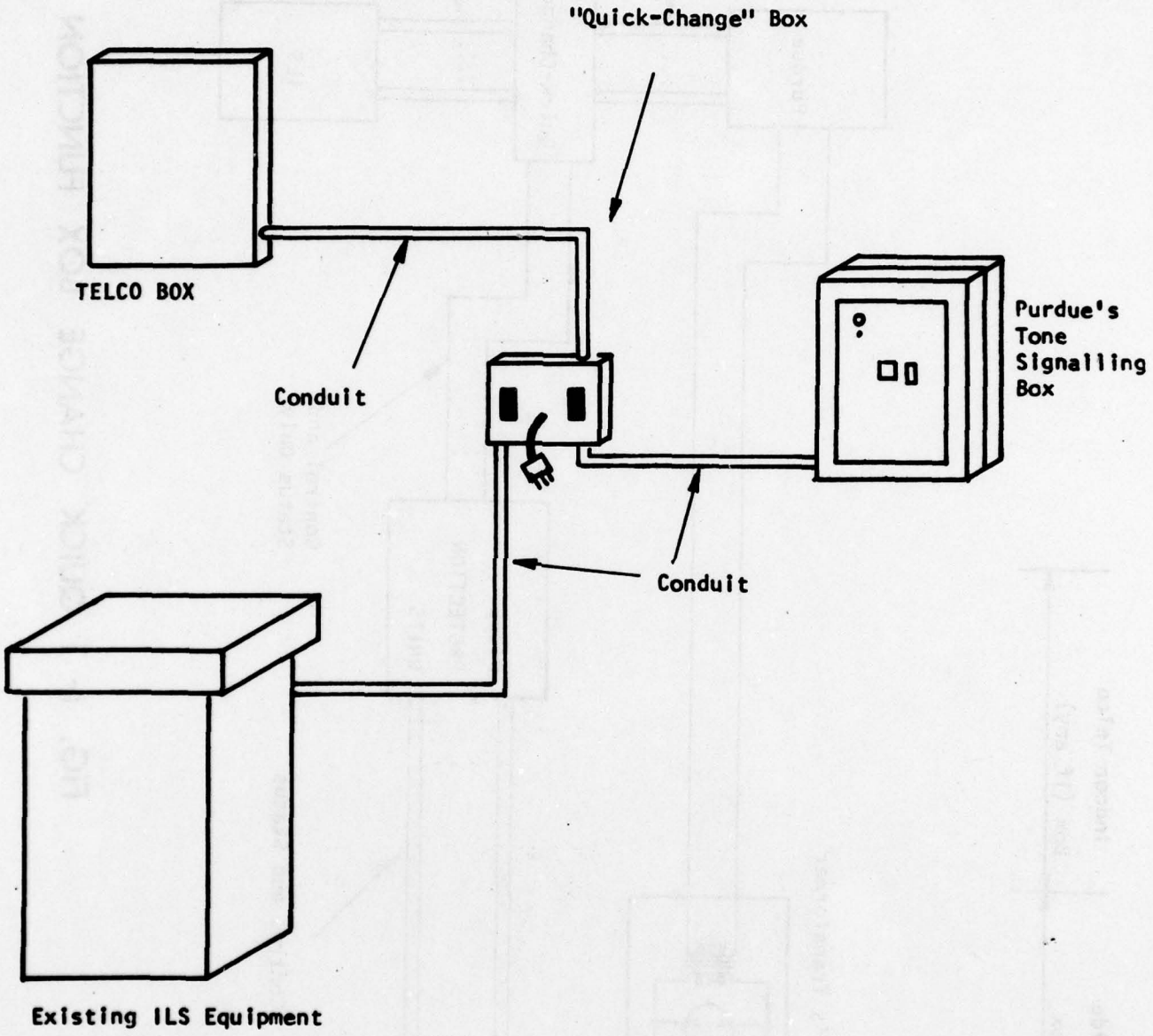


FIG. I-2 QUICK CHANGE BOX SET-UP AT A FIELD UNIT

Appendix J

Barrier Strip Commons Mounted Inside Signaling Drawer

Terminals 1-2 are tied together and provide a status signal common for various Purdue relay cards:

R0-TS2 (3-10)
R1-TS2 (3-10)
R2-TS2 (3-10)
R3-TS2 (3 only)
R4-TS2 (3-10)

Terminal 3 and 4 provide localizer maintenance monitor common to the following:

R5-TS2 (3-10)
R6-TS2 (3-10)
R3-TS2 (4 only)
R7-TS2 (8-10)

Terminal 5 provides glide slope maintenance monitor common to:

R7-TS2 (3 only)
R8-TS2 (3-10)
R9-TS2 (3-10)

Terminal 6 provides FFM maintenance monitor common to:

R7-TS2 (4-7)

Appendix K

Bussing Required on Relay Cards

RELAY CARD NO.	TERMINALS TO BE BUSSED TOGETHER ON TERMINAL STRIP 2 (TS2)
RØ	3 through 10
R1	3 through 10
R2	3 through 10
R3	none
R4	3 through 10
R5	3 through 10
R6	3 through 10
R7	4 through 7
	8 through 10
R8	3 through 10
R9	3 through 10

APPENDIX L

Theory of PROM Checksums

The programs for all seven computers in the tone signaling system are held in programmable read only memories. These PROM's will normally hold their contents intact for many years, regardless of whether or not power is applied at all times. It is possible, however, due to electrical failure for the contents to change. The effect of a change in memory could be varied. It might have no effect, it might change the contents of one of the display messages, or it could cause the program to quit working altogether. Both the microprocessors and the field units in the minicomputer in the control tower check the program memories to be sure that that contents have not changed.

This check is done using a conventional error detection technique known as a checksum. The particular computer involved will periodically add the contents of all the memory locations. The resulting sum, referred to as a checksum, should always be equal to the same number. If it is not, the computer knows that the program memory has changed in some way and signals the error on its appropriate screen.

At the field units, the microprocessor card which gives a PROM error should be replaced with one of the spare cards. The particular PROM which has changed can be detected by switching PROM's from a working microprocessor card. The control tower program will identify which bank of PROM's is in error. In either case, the offending PROM's should be returned to Purdue for replacement.

APPENDIX M

NIBBLE CHECK SUM ROUTINE

To Isolate Faulty PDP-11/03 PROMS

Instructions for Usage

After manually loading the check sum routine starting at location 200, begin execution at location 200. At each program halt, examine the contents of register R3, comparing its least significant four bits (as expressed in hexadecimal) with the reference check sums listed below. The first program halt yields the sum for PROM bank 20000, nibble 0. The second stop will load R3 with the bank 20000-nibble 1 sum. Successive program halts yield check sums for the following nibbles:

Halt Number	Bank Number	Nibble Number
1	20000	0
2	20000	1
3	20000	2
4	20000	3
5	22000	0
6	22000	1

Here, nibble zero refers to the least significant four bits of a 16-bit computer word. Nibble 3 refers to the most significant four bits. At each program stop, the bank number of the next check sum may be found in register R4. The number of the next nibble to be checked appears in register R0.

Program Listing (Nibble check sum routine)

Location	Code	Symbolic Address	Source	Code
000200	012704		MOV	#20000, R4
202	020000			
204	005000		CLR	R0
206	005003	TOP:	CLR	R3
210	012705		MOV	#1000, R5
212	001000			
214	012401	LOOPA:	MOV	(R4)+, R1
216	010002		MOV	R0, R2
220	005702	LOOPB:	TST	R2
222	001406		BEQ	DONESH
224	006201		ASR	R1
226	006201		ASR	R1
230	006201		ASR	R1
232	006201		ASR	R1
234	005302		DEC	R2
236	000770		BR	LOOPB
240	060103	DONESH:	ADD	R1, R3
242	005305		DEC	R5
244	001363		BNE	LOOPA
246	022700		CMP	#3, R0
250	000003			
252	001407		BEQ	IS3
254	042703		BIC	#177760, R3
256	177760			
260	005200		INC	R0
262	162704		SUB	#2000, R4
264	002000			
266	000000		HALT	
270	000746		BR	TOP
272	005000	IS3:	CLR	R0
274	042703		BIC	#177760, R3
276	177760			
300	000000		HALT	
302	000741		BR	TOP

Nibble Check Sum Reference Table

Bank Address	Nibble 3 Bits 15-12	Nibble 2 Bits 11-8	Nibble 1 Bits 7-4	Nibble 0 Bits 3-0
20000	6	3	1	A
22000	A	E	9	C
24000	D	A	9	D
26000	2	D	6	2
30000	8	1	7	3
32000	A	7	2	F
34000	8	A	D	E
36000	5	8	2	8
40000	4	C	7	B
42000	0	F	0	9
44000	7	A	0	A
46000	7	E	A	1
50000	8	7	0	8
52000	F	0	7	B
54000	8	0	6	5
56000	4	8	D	8

APPENDIX N

SNOW WHITE PROGRAM

Assembly Language Listings of

Program Patches
March, 1977

```
057606      MOVB        #1, RECEVE (R0)
            MOV         R0, R1
            ASL         R1
            JMP         @ #51434

057510      MOV         @ #4254, @ #5200
            MOV         @ #4212, @ #4254
            MOV         #135, R4
            JMP         @ #24136
;           DISABLE     AUTOLX AT THIS POINT

057534      MOV         @ #5200, @ #4254
            DEC         LOGBOT
            JMP         @ #24640

057552      CMPB        ERRTWR-1(R2), #1
            BNE         .+6
            JMP         @ #54150
            CMPB        ERRTWR-1(R2), #13
            BEQ         .+6
            JMP         @ #54260
            JMP         @ #54150
```

APPENDIX O

MICROPROCESSOR
CONTROLLED ILS COMMUNICATIONS SYSTEM
CHANGES MADE DURING THE WEEK OF
March 6, 1977

1. Whenever the ATCT u.p. (PDP-11/03) is halted, or powered down, an immediate indication will be made on both the SP, MM, and RC (tower cab) panels. This is done by illuminating all "abnormal" indicators, and extinguishing all "MAIN", "STANDBY", and "OFF" lamps. Of course, this will result in the sounding of SONALERT's at both the SP and RC panels.

2. Whenever either of these two errors occurs at the ATCT:

   a) NO CARRIER FROM FIELD UNIT

   b) NO DATA FROM FIELD UNIT

   an indication of an existing serious condition will be placed on the SP and RC panels. This is done by illuminating both "abnormal" indicators, and extinguishing the "MAIN", "STANDBY", and "OFF" lamps <u>for the transmitting field unit in question</u>.

   In the case of the FFM (which is <u>not</u> a transmitting FU) the alarm indication is made first by illuminating all four FFM-MM lamps, and second, by turning on the four abnormal alarms that are relayed to the localizer ("CAT II SHUTDOWN ALERT", "CAT II SHUTDOWN", "MONITOR MISMATCH", and "POWER-TEMPERATURE FAIL"). Upon receiving these <u>forced</u> alarms, the localizer will, of course, shut down.