

LOAN DOCUMENT

		PHOTOGRAPH THIS SHEET						
DTIC ACCESSION NUMBER	LEVEL							
	NADC-76011-20 DOCUMENT IDENTIFICATION							
ACCESSION CODE NTIS GRAM DTIC TRAC UNANNOUNCED JUSTIFICATION BY DISTRIBUTION/ AVAILABILITY CODES <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 50%; text-align: left;">DISTRIBUTION</th> <th style="width: 50%; text-align: left;">AVAILABILITY AND/OR SPECIAL</th> </tr> <tr> <td style="height: 50px; vertical-align: bottom; font-size: 2em;">A-1</td> <td></td> </tr> </table>		DISTRIBUTION	AVAILABILITY AND/OR SPECIAL	A-1		DISTRIBUTION STATEMENT		
DISTRIBUTION	AVAILABILITY AND/OR SPECIAL							
A-1								
DISTRIBUTION STAMP								
		DATE ACCESSIONED						
DATE RECEIVED IN DTIC		DATE RETURNED						
19981223 042		REGISTERED OR CERTIFIED NUMBER						
PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC								

HANDLE WITH CARE

NADC-75-1000

ON-LINE ACOUSTIC ANALYSIS SYSTEM
(OLAAS)
PROCESSING & DISPLAY SYSTEM

N62269-77-C-0533

NADC
Tech. Info.

NO DISTRIBUTION
STATEMENT

BUNKER
RAMO

ELECTRONIC SYSTEMS DIVISION

8000282

OLAAS PROCESSING & DISPLAY SYSTEM

SOFTWARE DESIGN NOTEBOOK

TABLE OF CONTENTS

SECTION

FOREWORD

1.0 INTRODUCTION

1.1 OLAAS Software System Configuration

1.2 Software Features

1.3 Communication Interface

2.0 OPERATING INSTRUCTION

2.1 Introduction

2.2 OLAAS System Hardware Configuration

2.3 OLAAS System Operation

2.4 Bootstrap Loading Procedures

2.5 Coreload Generation Procedure

2.6 Fourier Analyzer Program (Keyboard Stack)

2.7 Disc Copy Procedure (MADIU Program)

3.0 OLAAS DISC SYSTEM

3.1 Introduction

3.2 OLAAS Fourier Disc Layout

3.3 OLAAS Fourier Disc Generation

4.0 REFERENCES

5.0 OLAAS SOFTWARE DOCUMENTATION

5.1 Software Overview

5.2 OLAAS Processing & Display System

5.3 OLAAS Processing & Display System Subroutines

5.4 OLAAS User Programs & Subroutines

5.5 OLAAS System/Coreload Transfer Files

5.6 OLAAS System/Coreload Command Files

5.7 OLAAS System/Coreload PCS Input Files

APPENDIX

A OLAAS Acceptance Test Plan

B FMIXX1 Program Listing

C Disc Directory For The RTE System Disc
(Lower Disc)

OLAAS PROCESSING AND DISPLAY SYSTEM

SOFTWARE DESIGN NOTEBOOK

FOREWORD

Bunker Ramo Corporation, Electronic Systems Division has been engaged in the development of the ON-LINE ACOUSTIC ANALYSIS SYSTEM (OLAAS). The OLAAS System is a computerized data analysis system for the ERAPS Sonobuoy Advance Development Model.

The OLAAS PROCESSING AND DISPLAY SYSTEM allows an operator to collect and analyze ERAPS data subject to user-selected parameters. The user/operator may select the number of beams analyzed, look directions, range and doppler window. The type of display plotting is also under the user/operator control.

This document is submitted in accordance with, and as a completion of the following data items:

Contract N62269-75-C-0418, Item 0005

- A004, Documentation of Computer Programs
- A005, Operation and Maintenance Handbooks
- A006, Detection Algorithm Description
- A007, System Performance Specification

Contract N62269-77-C-0533

- A002, Program Description Document

OLAAS PROCESSING AND DISPLAY SYSTEM

SOFTWARE DESIGN NOTEBOOK

1.0 INTRODUCTION

The OLAAS PROCESSING AND DISPLAY SYSTEM, Software Design Notebook, describes the documentation for On-Line Acoustic Analysis System (OLAAS System). This document consists of the programs and subroutines that make up the system, operating instructions for system operation and generation, and additional details pertaining to the overall system.

1.1 OLAAS SOFTWARE SYSTEM CONFIGURATION

The OLAAS Software System uses the Hewlett Packard 5451B FOURIER ANALYZER System hardware, including a HP7900 Disc Memory. Because of the size of the OLAAS System programs, and a need to maintain an initialized version of the analysis and data collection programs, it was necessary to segment the programs into separate coreloads. The HP2100S Computer has 32K of core memory, and the total system can accommodate 8 Fourier coreloads, using the HP7800A Disc Memory. The coreloads are organized as shown:

<u>Coreload</u>	<u>Function</u>
0	OLAAS Executive Program
1	OLAAS Analysis and System Executive Subroutine
2	OLAAS Raw Data Collection Subroutine
3	Copy of coreload 1 (used for processing)
4	Copy of coreload 2 (used for multi-ping recording)
5	OLAAS User Programs Analysis
6	Fourier Analyzer Graphics
7	OLAAS Plot Display Subroutines

The key to the system is the transferring of control from one coreload to another (core swapping or overlay).

1.2 SOFTWARE FEATURES

The major feature of the OLAAS System is that it is completely operator oriented, i.e.; the operator is prompted on the display screen with messages requesting inputs for each action to take place or parameter to be used. The system is versatile in that the operator may select any option available as desired. The one restriction in the system is that raw data has been collected and stored on the disc. Additional features are that the Hewlett Packard software is used to generate and change programs and create the OLAAS Fourier System Disc.

1.3 COMMUNICATION INTERFACE

The operator's communication with the OLAAS System is in either of two (2) methods. (1) Input through the CRT TERMINAL Keyboard is reply to prompt messages displayed on the screen. (2) operate the PH2100S Computer Operator Panel or the 5451A Fourier Analyzer Keyboard. Using the procedures in Section 2.0, Operating Instructions, will describe the use of these methods.

2.0 OPERATING INSTRUCTIONS

2.1 INTRODUCTION

Operation of the OLAAS System is through the CRT Keyboard and the control switches on the HP-2100 Computer front panel and the HP-5451B Fourier Analyzer front panel.

2.1.1 Conventions. Certain conventions apply to all keyboard entries, as follows:

(1) For numeric inputs, a permissible range is usually printed within parentheses: (0-100000). Any value outside this range will result in a repeated request to enter the number. The same will occur if an illegal character is entered.

(2) For numeric inputs, the current value is also printed within parentheses. This value will remain unchanged if RETURN is pressed (no characters entered).

(3) All numeric inputs are integers (i.e., no fractional part) unless a decimal point appears in the range of current value printouts. For integers, the decimal point must be omitted; for other numbers, the decimal point may be omitted if there is no fractional part (tenths, hundredths, etc.).

(4) For all numeric inputs, leading zeros, trailing zeros, and the sign, if positive, may be omitted.

(5) All inputs are completed by pressing RETURN. The use of the RETURN key is assumed and will not be repeated in the operating instructions.

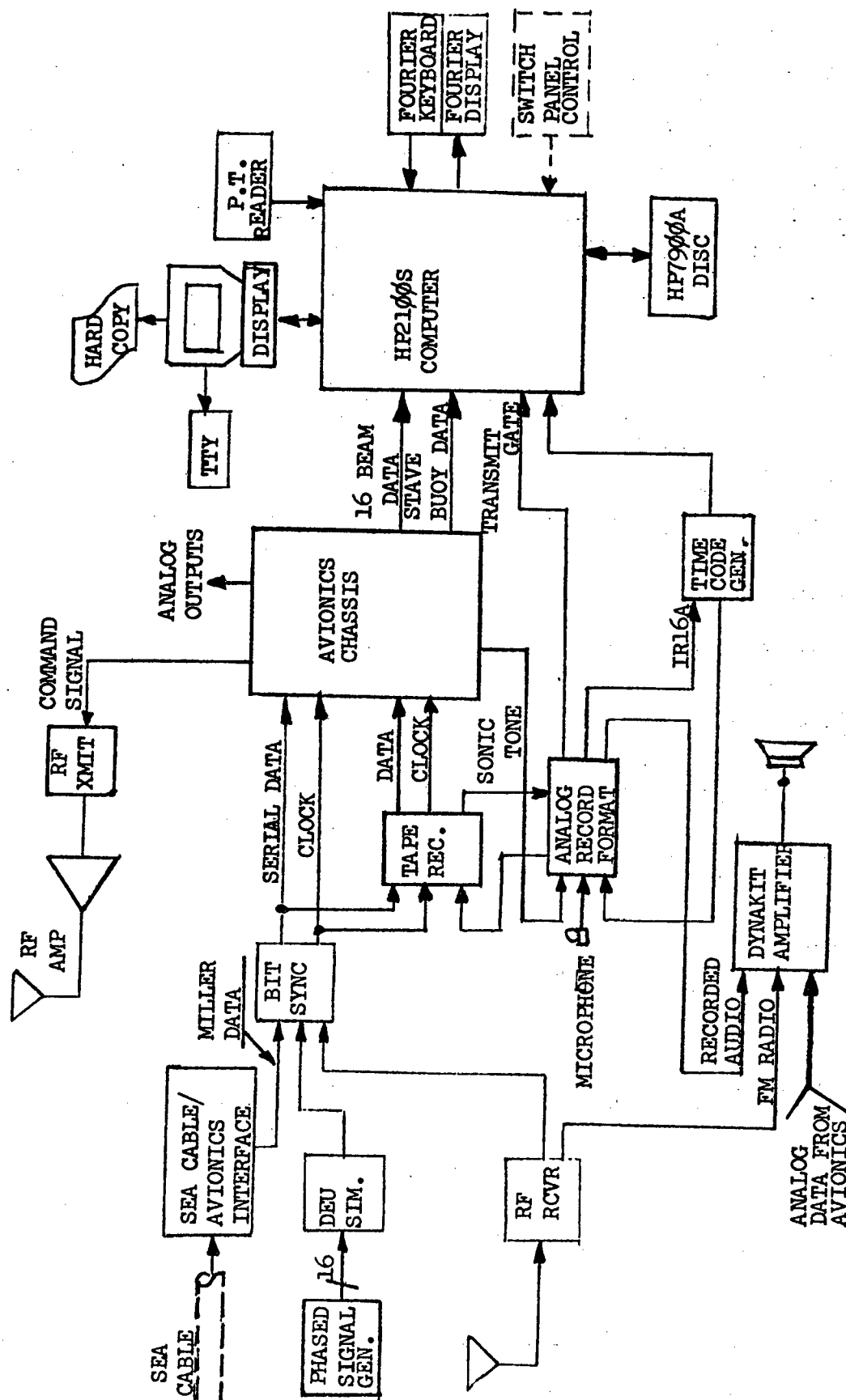
(6) An error recognized before pressing RETURN may be corrected by pressing RUBOUT, then making the proper entry.

(7) Entering 12345 in place of most numeric inputs will result in control being returned to the Executive.

(8) Entering 11111 in place of most numeric inputs when doing DATA RECORDING & ANALYSIS will result in control being returned to the start of processing and will save analyzed data.

2.2 OLAAS SYSTEM HARDWARE CONFIGURATION

The OLAAS Hardware Configuration is shown in the block diagram, Figure 2.2-1, and the ERAPS Van layout is shown in the block diagram, Figure 2.2-2. Turn-On and operation of the OLAAS System is described elsewhere.



OLAAS VAN SIGNAL FLOW

FIGURE: 2.2-1

2.3. OLAAS SYSTEM OPERATION PROCEDURE

2.3.1. Ensure that power is on at all applicable units in the van.

2.3.2. Ensure that the OLAAS SYSTEM Fourier Disc Cartridge is installed and that the disc is ready (DISC READY light is illuminated).

2.3.3. Execute OPERATION PROCEDURE: 2.4.1 - OLAAS System Loading Procedure.

2.3.4. The OLAAS EXECUTIVE program was entered by "bootstrap" loading the system and the following message was displayed:

```
***** OLAAS PROCESSING & DISPLAY SYSTEM *****  
EXECUTIVE PROGRAM
```

```
** ENTER DESIRED OPERATION **
```

```
D = DATA RECORDING & ANALYSIS  
F = USER PROGRAM ANALYZER  
G = FOURIER ANALYZER GRAPHICS
```

2.3.5. Enter the indicated letter for the desired operation and execute the procedure specified in the following paragraphs:

```
D = DATA RECORDING AND ANALYSIS      = 2.3.6  
F = USER PROGRAM ANALYSIS             = 2.3.7  
G = FOURIER ANALYZER GRAPHICS          = 2.3.8
```

2.3.6. Data Recording and Analysis Operation.

2.3.6.1. Enter the letter "D" for the DATA RECORDING ANALYSIS operation and the following message will be displayed:

```
***** OLAAS PROCESSING & DISPLAY SYSTEM *****  
EXECUTIVE PROGRAM
```

```
** ENTER DESIRED OPERATION **
```

```
D = DATA RECORDING & ANALYSIS  
F = USER PROGRAM ANALYZER  
G = FOURIER ANALYZER GRAPHICS
```

D

```
***** DATA RECORDING & ANALYSIS *****
```

```
SYSTEM INITIALIZATION
```

```
OPERATOR NOTE: SET SENSE SWITCH 15 ON TO CHANGE THE BUOY  
FREQUENCY, PULSE LENGTH, & CW NOTCH FILTER.
```

```
ENTER DATE
```

```
( )
```

2.3.6.2. If the operator sets Sense Switch 15 on the indicated question will appear in their proper position. These procedures assume the switch is off.

2.3.6.3. Enter the desired date and the following message will be displayed.

```
***** OLAAS DATA RECORDING & ANALYSIS *****
                25 JANUARY 1978
```

```
                ** ENTER DESIRED FUNCTION **
1 = ON-LINE DATA RECORDING & ANALYSIS
2 = OFF-LINE ANALYSIS & PLOT DISPLAY
3 = DATE ENTRY OR CHANGE                                ( 2)
```

2.3.6.4. Enter the indicated value for the desired function and execute the procedure specified by the following paragraphs:

```
1 = ON-LINE DATA RECORDING AND ANALYSIS      = 2.3.6.4.1
2 = OFF-LINE ANALYSIS AND DISPLAY              = 2.3.6.4.2
3 = DATE ENTRY OR CHANGE                      = 2.3.6.4.3
```

2.3.6.4.1 On-Line Data Recording and Analysis Function.

2.3.6.4.1.1 Enter the value "1" for the ON-LINE DATA RECORDING AND ANALYSIS function and the following message will be displayed:

```
***** OLAAS DATA RECORDING & ANALYSIS *****
                25 JANUARY 1978
```

```
                ** ENTER DESIRED FUNCTION **
1 = ON-LINE DATA RECORDING & ANALYSIS
2 = OFF-LINE ANALYSIS & PLOT DISPLAY
3 = DATE ENTRY OR CHANGE                                ( 2) 1
```

```
***** ON-LINE DATA RECORDING & ANALYSIS *****
```

```
                ** ANSWER ANALYSIS QUESTIONS **
SIGNAL MODE      CW = 0, FM UP = 1, OR FM DOWN = 2 ( 0) 0
```

2.3.6.4.1.1 Answer each question with the desired parameter required and the next question will appear. Figure 2.3.6.4-1 shows all possible "ON-LINE" questions that can be asked (Sense Switch 15 is on).

2.3.6.4.1.2 When the last question has been answered the raw data recording is done and then that data is analyzed and displayed. Figure 2.3.6.4-2 is an example of the analyzed display.

***** OLAAS DATA RECORDING & ANALYSIS *****
 25 JANUARY 1978

** ENTER DESIRED FUNCTION **

- 1 - ON-LINE DATA RECORDING & ANALYSIS
- 2 - OFF-LINE ANALYSIS & PLOT DISPLAY
- 3 - DATE ENTRY OR CHANGE

(2) 1

***** ON-LINE DATA RECORDING & ANALYSIS *****

** ANSWER ANALYSIS QUESTIONS **

FREQUENCY	F1 - 1 OR F2 - 2	(1)
PULSE LENGTH	SHORT - 2 OR LONG - 5	(2)
SIGNAL MODE	CW - 0, FM UP - 1, OR FM DOWN - 2	(0)
CW DOPPLER BAND	FULL-1, HALF-2, UP-3, OR DOWN-4	(2)
CW NOTCH FILTER	WIDTH IN FREQUENCY CELLS (0 TO 25)	(15)
START RANGE	IN KILOYARDS (5 TO 35)	(8)
STOP RANGE	IN KILOYARDS (MAXIMUM 40)	(24)
RANGE RESOLUTION	IN MULTIPLES OF 160 YARDS	(1)
BEARING	ANGLE IN DEGREES (CENTER BEAM)	(0)
BEAMS	NUMBER OF BEAMS	(3)
ANALYSIS THRESHOLD	PERCENT OF AVERAGE POWER (0-10000)	(100)

2 1 2

** RAW DATA RECORDING QUESTIONS **

START RANGE	IN KILOYARDS (5 TO 35)	(8)
STOP RANGE	IN KILOYARDS (MAXIMUM 40)	(24)
BEAMS	NUMBER OF BEAMS TO RECORD	(16)
NUMBER PINGS	NUMBER OF PINGS TO RECORD	(1)
CENTER BEAM	DESIRED CENTER BEAM DIRECTION	(0)
DATA SOURCE	1-ACTIVE ON-LINE / 0-TAPE RECORDER	(1)

1 2 0

FIGURE 2.3.6.4-1

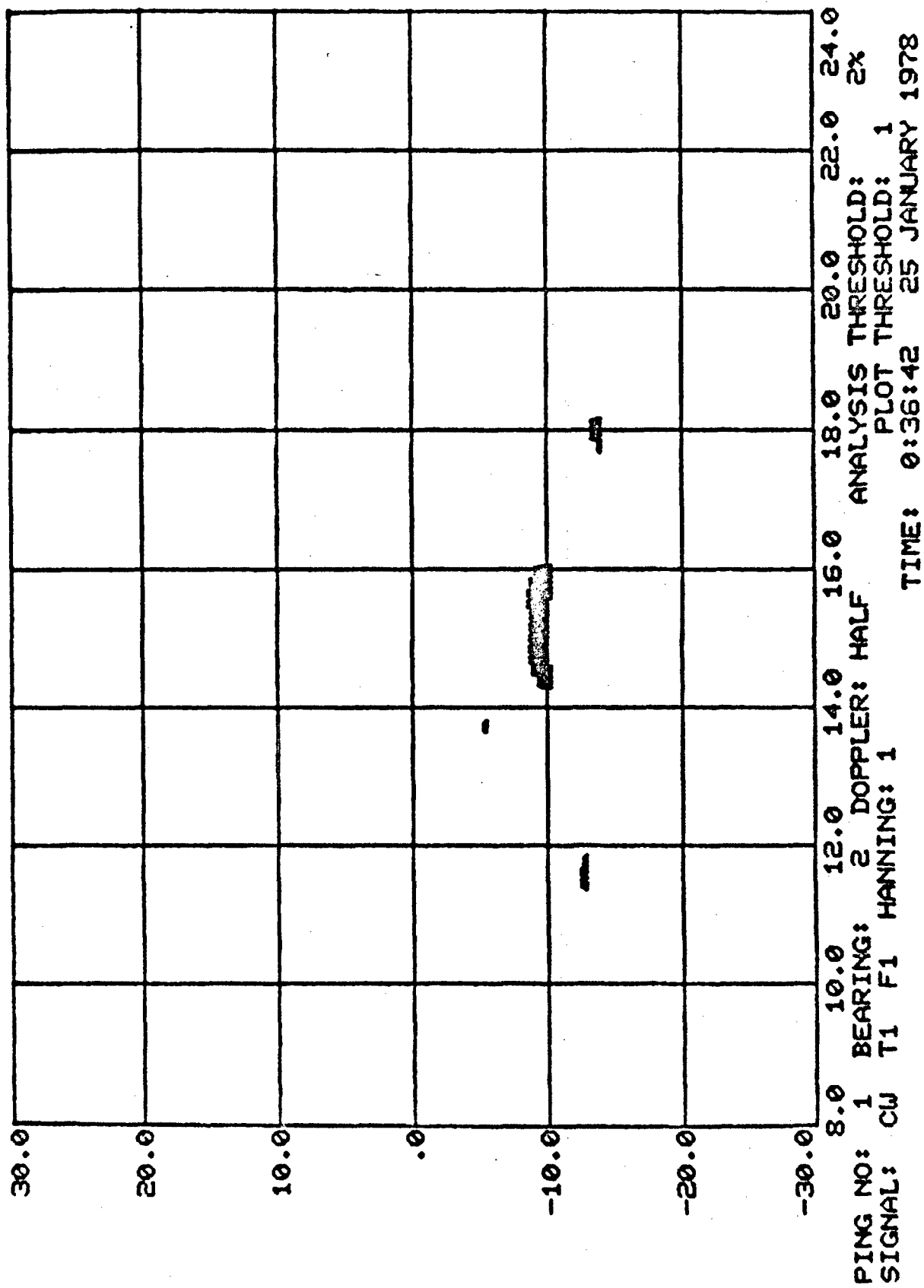


Figure 2.3.6.4-2

2.3.6.4.1.3 At the CRT TERMINAL keyboard press RESET PAGE and the following message will be displayed:

```
NEW THRESHOLD PRECENTAGE FOR PLOT (1 - 100)
(1% DISPLAYS ALL ANALYZED DATA, 100% = NONE)
TYPE 0 TO RETURN TO ANALYSIS PROGRAM
( 0)
```

2.3.6.4.1.4 The operator may desire to make a new plot using a higher "Plot Threshold", if so, enter the desired value (1 - 100) and a new plot will be displayed. This operation may be repeated as many times as desired. Enter a value of "0" to continue operation.

2.3.6.4.1.5 The RECORDED AND ANALYZED DATA STATUS TABLES will be displayed. Figure 2.3.6.4-3 is an example of the DATA STATUS TABLES. At the CRT TERMINAL keyboard press RESET PAGE and the following message will be displayed, the ON-LINE DATA RECORDING AND ANALYSIS is complete.

```
***** OLAAS DATA RECORDING & ANALYSIS *****
25 JANUARY 1978
```

```
    ** ENTER DESIRED FUNCTION **
1 = ON-LINE DATA RECORDING & ANALYSIS
2 = OFF-LINE ANALYSIS & PLOT DISPLAY
3 = DATE ENTRY OR CHANGE                      ( 1)
```

2.3.6.4.2 Off-Line Analysis and Plot Display.

2.3.6.4.2.1 Enter the value "2" for the OFF-LINE ANALYSIS AND PLOT DISPLAY function and the DATA STATUS TABLES will be displayed. Figure 2.3.6.4-3 is an example of the DATA STATUS TABLES.

2.3.6.4.2.2 At the CRT TERMINAL keyboard press RESET PAGE and the following message will be displayed:

```
***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
25 JANUARY 1978
```

```
    ** SELECT OFF-LINE OPTION **
1 = ANALYSIS OF PREVIOUSLY RECORDED DATA
2 = INDIVIDUAL BEAM DISPLAY
3 = MULTI-PING DISPLAY
4 = FOUR PART DISPLAY                      ( 1)
```

2.3.6.4.2.2 Enter the indicated value for the desired off-line option and execute the procedure specified by the following paragraphs:

***** OLAAS PROCESSING & DISPLAY SYSTEM *****
 25 JANUARY 1978

RECORDED DATA STATUS TABLE				
NUMBER OF PINGS RECORDED - 1				
PING RECORD WINDOW - 7 TO 39 SECONDS				
NUMBER OF BEAMS RECORDED - 16				
PING NUMBER	COMPASS (DEGREES)	FIRST BEAM	HEADING (DEGREES)	TIME
1	346	11	211	8: 1: 4

ANALYZED DATA STATUS TABLE				
PING NUMBER	CENTER BEAM ANGLE	NUMBER BEAMS PROCESSED	NUMBER OF EXCEEDENCES	AVERAGE EXCEEDENCE
1	31	1	366	25

FIGURE 2.3.6.4-3

1 = ANALYSIS OF PREVIOUSLY RECORDED DATA	= 2.3.6.4.2.2.1
2 = INDIVIDUAL BEAM DISPLAY	= 2.3.6.4.2.2.2
3 = MULTI-PING DISPLAY	= 2.3.6.4.2.2.3
4 = FOUR PART DISPLAY	= 2.3.6.4.2.2.4

2.3.6.4.2.2.1 Analysis of Previously Recorded Data Option.

2.3.6.4.2.2.1.1 Enter the value "1" for the ANALYSIS OF PREVIOUSLY RECORDED DATA option and the following message will be displayed:

```
***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
                25 JANUARY 1978
```

```
      ** SELECT OFF-LINE OPTION **
      1 = ANALYSIS OF PREVIOUSLY RECORDED DATA
      2 = INDIVIDUAL BEAM DISPLAY
      3 = MULTI-PING DISPLAY
      4 = FOUR PART DISPLAY                      (    1) 1
```

```
      ** ANSWER ANALYSIS QUESTIONS **
```

```
      FIRST PING          NUMBER FOR ANALYSIS (1 to 10)      (    1)
```

2.3.6.4.2.2.1.2 Answer each question with the desired parameter required and the next question will appear. Figure 2.3.6.4-4 shows all possible "analysis" questions that can be asked (Sense Switch 15 is on).

2.3.6.4.2.2.1.3 When the last question has been answered the data is analyzed and displayed. Figure 2.3.6.4-2 is an example of the analyzed display.

2.3.6.4.2.2.1.4 At the CRT TERMINAL keyboard press RESET PAGE and the following message will be displayed:

```
      NEW THRESHOLD PERCENTAGE FOR PLOT (1 - 100)
      (1% DISPLAYS ALL ANALYZED DATA, 100% = NONE)
      TYPE 0 TO RETURN TO ANALYSIS PROGRAM
      (    0)
```

2.3.6.4.2.2.1.5 The operator may desire to make a new plot using a higher "Plot Threshold", if so, enter the desired value (1 - 100) and a new plot will be displayed. This operation may be repeated as many times as desired.

2.3.6.4.2.2.1.6 The RECORDED AND ANALYZED DATA STATUS TABLES will be displayed. Figure 2.3.6.4-3 is an example of the DATA STATUS TABLES. At the CRT TERMINAL keyboard press RESET PAGE and the following message will be displayed, and the ANALYSIS OF PREVIOUSLY RECORDED DATA is complete.

***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
 25 JANUARY 1978

** SELECT OFF-LINE OPTION **

- 1 - ANALYSIS OF PREVIOUSLY RECORDED DATA
- 2 - INDIVIDUAL BEAM DISPLAY
- 3 - MULTI-PING DISPLAY
- 4 - FOUR PART DISPLAY

(1) 1

** ANSWER ANALYSIS QUESTIONS **

FIRST PING	NUMBER FOR ANALYSIS (1 TO 10)	(1)
LAST PING	NUMBER FOR ANALYSIS (1 TO 10)	(1)
FREQUENCY	F1 - 1 OR F2 - 2	(1)
PULSE LENGTH	SHORT - 2 OR LONG - 5	(2)
SIGNAL MODE	CW - 0, FM UP - 1, OR FM DOWN - 2	(0)
CW DOPPLER BAND	FULL-1, HALF-2, UP-3, OR DOWN-4	(2)
CW NOTCH FILTER	WIDTH IN FREQUENCY CELLS (0 TO 25)	(15)
START RANGE	IN KILOYARDS (5 TO 35)	(8)
STOP RANGE	IN KILOYARDS (MAXIMUM 40)	(24)
RANGE RESOLUTION	IN MULTIPLES OF 100 YARDS	(1)
BEARING	ANGLE IN DEGREES (CENTER BEAM)	(0)
BEAMS	NUMBER OF BEAMS	(3)
ANALYSIS THRESHOLD	PERCENT OF AVERAGE POWER (0-10000)	(100)

Figure 2.3.6.4-4

***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
25 JANUARY 1978

 ** SELECT OFF-LINE OPTION **
1 = ANALYSIS OF PREVIOUSLY RECORDED DATA
2 = INDIVIDUAL BEAM DISPLAY
3 = MULTI-PING DISPLAY
4 = FOUR PART DISPLAY (1)

2.3.6.4.2.2.2 Individual Beam Display Option.

2.3.6.4.2.2.2.1 Enter the value "2" for the INDIVIDUAL BEAM DISPLAY option and the following message will be displayed:

***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
25 JANUARY 1978

 ** SELECT OFF-LINE OPTION **
1 = ANALYSIS OF PREVIOUSLY RECORDED DATA
2 = INDIVIDUAL BEAM DISPLAY
3 = MULTI-PING DISPLAY
4 = FOUR PART DISPLAY (1) 2

 ** ANSWER INDIVIDUAL BEAM DISPLAY QUESTIONS **

DISPLAY PING PING NUMBER TO BE DISPLAYED (1)

2.3.6.4.2.2.2.2 Answer each question with the desired parameter required and the next question will appear. Figure 2.3.6.4-5 shows all possible "INDIVIDUAL BEAM" questions that can be asked (Sense Switch 15 is on).

2.3.6.4.2.2.2.3 When the last question has been answered the raw data recording is done and then that data is analyzed and displayed. Figure 2.3.6.4-6 is an example of the individual beam display.

2.3.6.4.2.2.2.4 At the CRT TERMINAL keyboard press RESET PAGE and the following message will be displayed:

NEW THRESHOLD PERCENTAGE FOR PLOT (1 - 100)
(1% DISPLAYS ALL ANALYZED DATA, 100% = NONE)
TYPE 0 TO RETURN TO ANALYSIS PROGRAM
(0)

2.3.6.2.2.2.5 The operator may desire to make a new plot using a higher "Plot Threshold", if so, enter the desired value (1 - 100) and a new plot will be displayed. This operation may be repeated as many times as desired. Enter a value of "0" to continue operation.

***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
 29 JANUARY 1978

** SELECT OFF-LINE OPTION **

- 1 - ANALYSIS OF PREVIOUSLY RECORDED DATA
- 2 - INDIVIDUAL BEAM DISPLAY
- 3 - MULTI-PING DISPLAY
- 4 - FOUR PART DISPLAY

(5) 2

** ANSWER INDIVIDUAL BEAM DISPLAY QUESTIONS **

DISPLAY PING
 DISPLAY TYPE
 START RANGE
 STOP RANGE
 BEARING
 BEAMS
 PLOT THRESHOLD

PING NUMBER TO BE DISPLAYED
 2 - INDIVIDUAL 5 - BEAMS OR'ED
 IN KILOYARDS (5 TO 35)
 IN KILOYARDS (MAXIMUM 40)
 ANGLE IN DEGREES (CENTER BEAM)
 NUMBER OF BEAMS
 1 TO 100 (X) (1- DISPLAY ALL)

(1)
 (2)
 (20)
 (28)
 (338)
 (1)
 (1)

***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
 29 JANUARY 1978

** SELECT OFF-LINE OPTION **

- 1 - ANALYSIS OF PREVIOUSLY RECORDED DATA
- 2 - INDIVIDUAL BEAM DISPLAY
- 3 - MULTI-PING DISPLAY
- 4 - FOUR PART DISPLAY

(2)

** ANSWER INDIVIDUAL BEAM DISPLAY QUESTIONS **

DISPLAY PING
 DISPLAY TYPE
 START RANGE
 STOP RANGE
 BEARING
 BEAMS
 PLOT THRESHOLD

PING NUMBER TO BE DISPLAYED
 2 - INDIVIDUAL 5 - BEAMS OR'ED
 IN KILOYARDS (5 TO 35)
 IN KILOYARDS (MAXIMUM 40)
 ANGLE IN DEGREES (CENTER BEAM)
 NUMBER OF BEAMS
 1 TO 100 (X) (1- DISPLAY ALL)

(1)
 (2)
 (20)
 (28)
 (338)
 (1)
 (1)

5

2.3.6.4.2.2.2.6 The RECORDED AND ANALYZED DATA STATUS TABLES will be displayed. Figure 2.3.6.4-3 is an example of the DATA STATUS TABLES. At the CRT TERMINAL keyboard press RESET PAGE and the following message will be displayed, and the INDIVIDUAL BEAM DISPLAY is complete.

***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
25 JANUARY 1978

** SELECT OFF-LINE OPTION **

- 1 = ANALYSIS OF PREVIOUSLY RECORDED DATA
- 2 = INDIVIDUAL BEAM DISPLAY
- 3 = MULTI-PING DISPLAY
- 4 = FOUR PART DISPLAY

(2)

2.3.6.4.2.2.2.7 The INDIVIDUAL BEAM DISPLAY contains an option for BEAMS OR'ED. To execute this option enter a value of "5" in answer to the DISPLAY TYPE question, all other operation are the same as paragraphs 2.3.6.4.2.2.2.1 through 2.3.6.4.2.2.2.6. Figures 2.3.6.4-7 and 2.3.6.4-8 are examples of the INDIVIDUAL BEAM DISPLAY question and BEAMS OR'ED display.

2.3.6.4.2.2.3 Multi-Ping Display Option.

2.3.6.4.2.2.3.1 Enter the value "3" for the MULTI-PING DISPLAY option and the following message will be displayed:

***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
25 JANUARY 1978

** SELECT OFF-LINE OPTION **

- 1 = ANALYSIS OF PREVIOUSLY RECORDED DATA
- 2 = INDIVIDUAL BEAM DISPLAY
- 3 = MULTI-PING DISPLAY
- 4 = FOUR PART DISPLAY

(2) 3

** ANSWER MULTI-PING DISPLAY QUESTIONS **

FIRST POINT NUMBER FOR ANALYSIS (1 TO 10) (1)

2.3.6.4.2.2.3.2 Answer each question with the desired parameter required and the next question will appear. Figure 2.3.6.4-9 shows all possible "MULTI-PING" questions that can be asked (Sense Switch 15 is on).

2.3.6.4.2.2.3.3 When the last question has been answered the data is displayed. Figure 2.3.6.4-10 is an example of the multi-ping display.

***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
 25 JANUARY 1978

** SELECT OFF-LINE OPTION **

- 1 - ANALYSIS OF PREVIOUSLY RECORDED DATA
- 2 - INDIVIDUAL BEAM DISPLAY
- 3 - MULTI-PING DISPLAY
- 4 - FOUR PART DISPLAY

(2) 2

** ANSWER INDIVIDUAL BEAM DISPLAY QUESTIONS **

DISPLAY PING
 DISPLAY TYPE
 BEARING
 BEAMS
 PLOT THRESHOLD

PING NUMBER TO BE DISPLAYED
 2 - INDIVIDUAL 5 - BEAMS OR'ED
 ANGLE IN DEGREES (CENTER BEAM)
 NUMBER OF BEAMS
 1 TO 100 (%) (1- DISPLAY ALL)

(1)
 (2)
 (2)
 (1)
 (1)

5

Figure 2.3.6.4-7

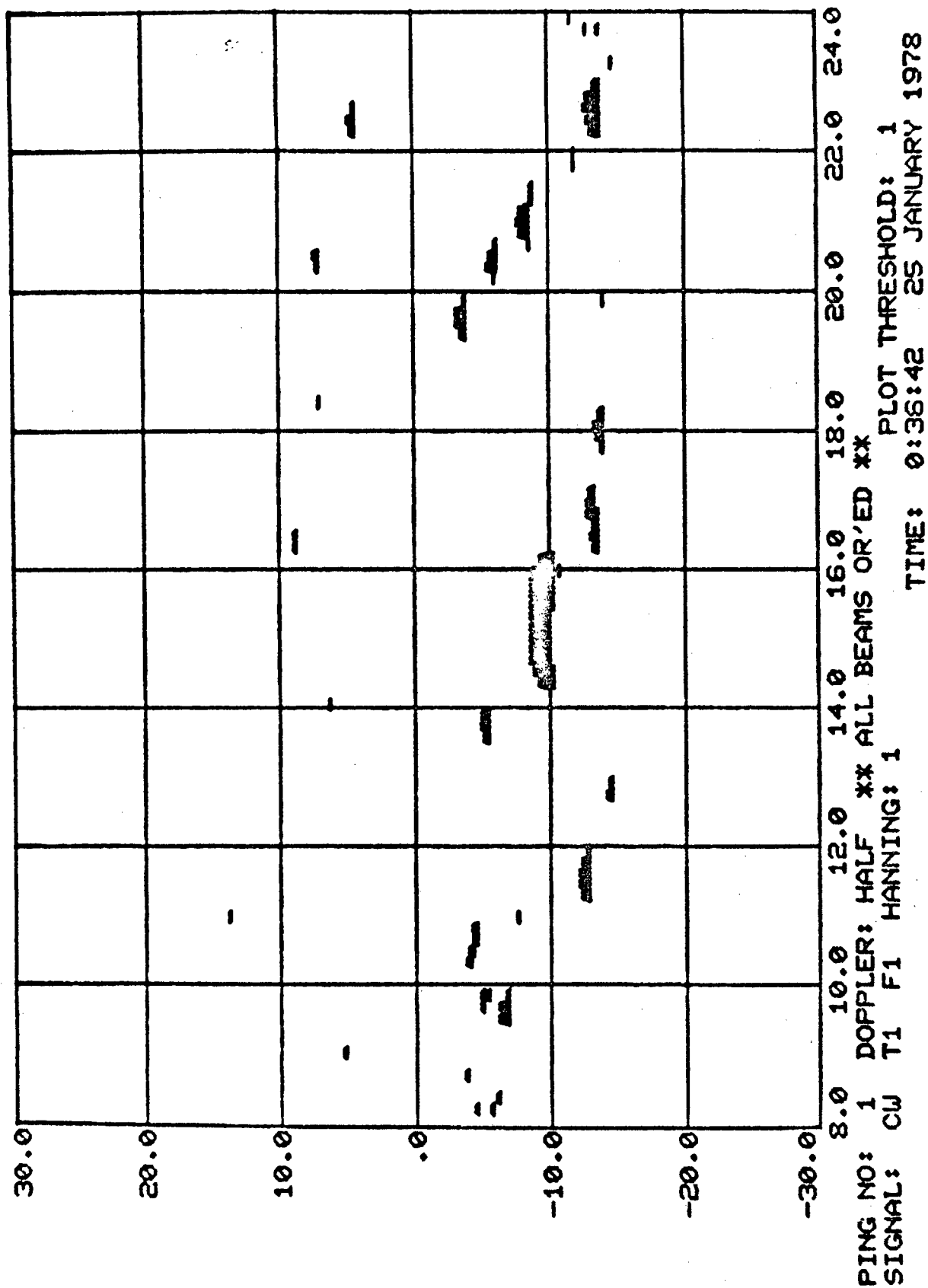


Figure 2.3.6.4-8

***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
 25 JANUARY 1978

** SELECT OFF-LINE OPTION **

- 1 - ANALYSIS OF PREVIOUSLY RECORDED DATA
- 2 - INDIVIDUAL BEAM DISPLAY
- 3 - MULTI-PING DISPLAY
- 4 - FOUR PART DISPLAY

(2) 3

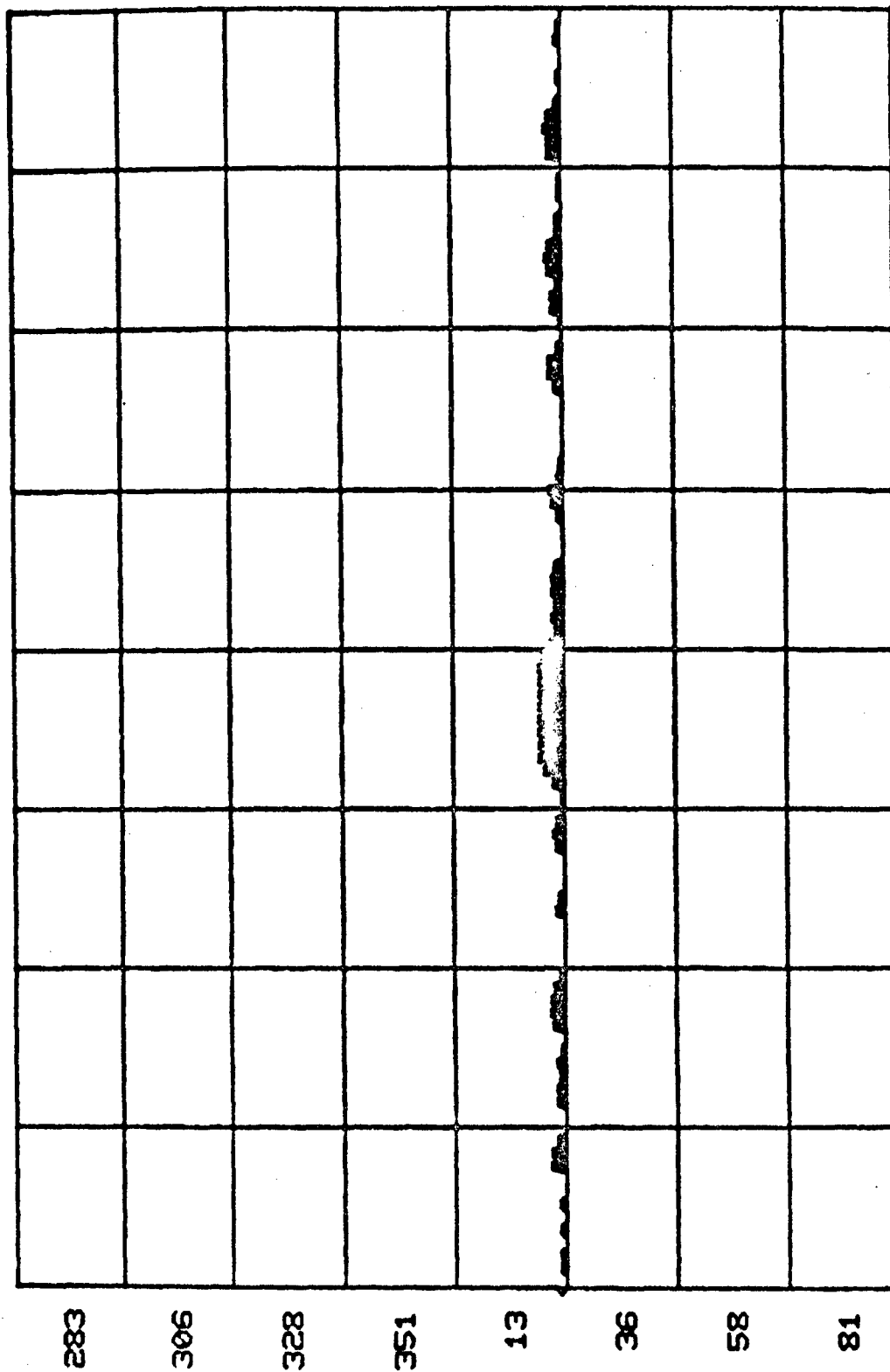
** ANSWER MULTI-PING DISPLAY QUESTIONS **

FIRST PING
 LAST PING
 START RANGE
 STOP RANGE
 BEARING
 PLOT THRESHOLD

NUMBER FOR ANALYSIS (1 TO 10)
 NUMBER FOR ANALYSIS (1 TO 10)
 IN KILOYARDS (5 TO 35)
 IN KILOYARDS (MAXIMUM 40)
 ANGLE IN DEGREES (CENTER BEAM)
 1 TO 100 (%) (1= DISPLAY ALL)

(1) 1
 (1) 1
 (8) 8
 (24) 24
 (2) 2
 (1) 1

Figure 2.3.6.4-9



8.0 10.0 12.0 14.0 16.0 18.0 20.0 22.0 24.0
 FIRST PING NO: 1 DOPPLER: HALF PLOT THRESHOLD: 1
 SIGNAL: CU T1 F1 HANNING: 1 TIME: 0:36:42 25 JANUARY 1978

Figure 2.3.6.4-10

2.3.6.4.2.2.3.4 At the CRT TERMINAL keyboard press RESET PAGE and the following message will be displayed:

```
ENTER NEW THRESHOLD PERCENTAGE FOR PLOT (1 - 100%)
(1% DISPLAYS ALL ANALYZED AMPLITUDES, 100%= NONE)
ENTER 0 TO RETURN TO ANALYSIS PROGRAM
( 0)
```

2.3.6.2.2.3.5 The operator may desire to make a new plot using a higher "Plot Threshold", if so, enter the desired value (1 - 100) and a new plot will be displayed. This operation may be repeated as many times as desired. Enter a value of "0" to continue operation.

2.3.6.4.2.2.3.6 The RECORDED AND ANALYZED DATA STATUS TABLES will be displayed. Figure 2.3.6.4-3 is an example of the DATA STATUS TABLES. At the CRT TERMINAL keyboard press RESET PAGE and the following message will be displayed, and the MULTI-PING DISPLAY is complete.

```
***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
25 JANUARY 1978
```

```
    ** SELECT OFF-LINE OPTION **
1 = ANALYSIS OF PREVIOUSLY RECORDED DATA
2 = INDIVIDUAL BEAM DISPLAY
3 = MULTI-PING DISPLAY
4 = FOUR PART DISPLAY ( 3)
```

2.3.6.4.2.2.4 Four Part Display Option.

2.3.6.4.2.2.4.1 Enter the value "4" for the FOUR PART DISPLAY option and the following message will be displayed:

```
***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
25 JANUARY 1978
```

```
    ** SELECT OFF-LINE OPTION **
1 = ANALYSIS OF PREVIOUSLY RECORDED DATA
2 = INDIVIDUAL BEAM DISPLAY
3 = MULTI-PING DISPLAY
4 = FOUR PART DISPLAY ( 3) 4
```

```
    ** ANSWER FOUR PART DISPLAY QUESTIONS **
```

```
DISPLAY PING      PING NUMBER TO BE DISPLAYED ( 1)
```

2.3.6.4.2.2.4.2 Answer each question with the desired parameter required and the next question will appear. Figure 2.3.6.4-11 shows all possible "ON-LINE" questions that can be asked (Sense Switch 15 is on).

***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
 25 JANUARY 1978

** SELECT OFF-LINE OPTION **

- 1 - ANALYSIS OF PREVIOUSLY RECORDED DATA
- 2 - INDIVIDUAL BEAM DISPLAY
- 3 - MULTI-PING DISPLAY
- 4 - FOUR PART DISPLAY

(3) 4

** ANSWER FOUR PART DISPLAY QUESTIONS **

DISPLAY PING	PING NUMBER TO BE DISPLAYED	(1) 1
DISPLAY BEARING	BEARING TO BE DISPLAYED (0 TO 359)	(2) 2
START RANGE	IN KILOYARDS (5 TO 35)	(8) 8
STOP RANGE	IN KILOYARDS (MAXIMUM 40)	(24) 24
PLOT THRESHOLD	1 TO 100 (X) (1- DISPLAY ALL)	(1) 1

Figure 2.3.6.4-11

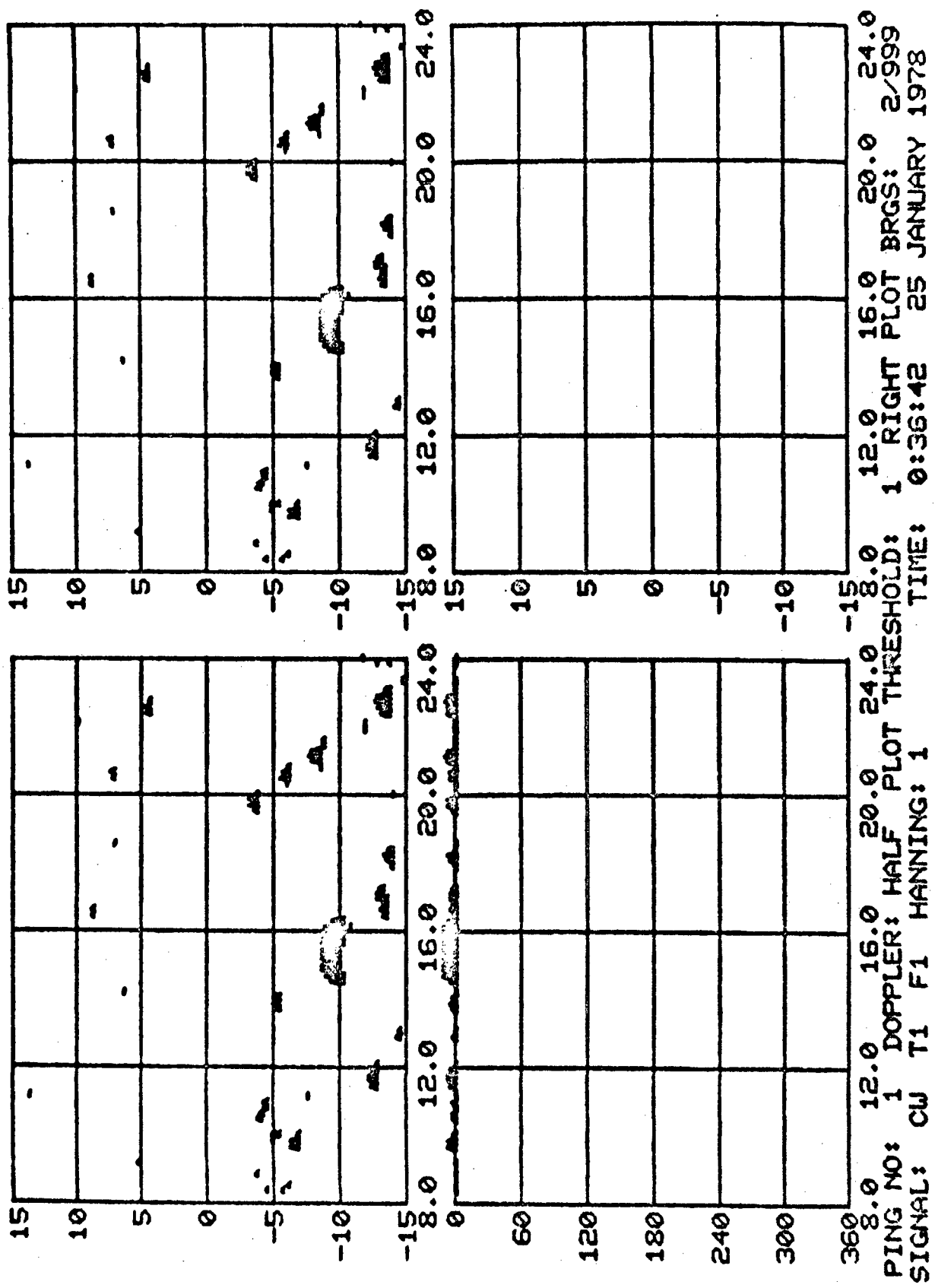


Figure 2.3.6.4-12

2.3.6.4.2.2.4.3 When the last question has been answered the data will be displayed. Figure 2.3.6.4-12 is an example of the FOUR PART DISPLAY.

2.3.6.4.2.2.4.4 At the CRT TERMINAL keyboard press RESET PAGE and the following message will be displayed:

```
NEW THRESHOLD PERCENTAGE FOR PLOT (1 - 100)
(1% DISPLAYS ALL ANALYZED DATA, 100% = NONE)
TYPE 0 TO RETURN TO ANALYSIS PROGRAM
( 0)
```

2.3.6.4.2.2.4.5 The operator may desire to make a new plot using a higher "Plot Threshold", if so, enter the desired value (1 - 100) and a new plot will be displayed. This operation may be repeated as many times as desired. Enter a value of "0" to continue operation.

2.3.6.4.2.2.4.6 The RECORDED AND ANALYZED DATA STATUS TABLES will be displayed. Figure 2.3.6.4-3 is an example of the DATA STATUS TABLES. At the CRT TERMINAL keyboard press RESET PAGE and the following message will be displayed, and the FOUR PART DISPLAY is complete.

```
***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY *****
                25 JANUARY 1978
```

```
                ** SELECT OFF-LINE OPTION **
1 = ANALYSIS OF PREVIOUSLY RECORDED DATA
2 = INDIVIDUAL BEAM DISPLAY
3 = MULTI-PING DISPLAY
4 = FOUR PART DISPLAY                                ( 4)
```

2.3.6.4.3 Date Entry or Change Function.

2.3.6.4.3.1 Enter the value "3" for the DATE ENTRY OR CHANGE function and the following message will be displayed:

```
***** OLAAS DATA RECORDING & ANALYSIS *****

                ** ENTER DESIRED FUNCTION **
1 = ON-LINE DATA RECORDING & ANALYSIS
2 = OFF-LINE ANALYSIS & PLOT DISPLAY
3 = DATE ENTRY OR CHANGE                                ( 2) 3

ENTER DATE
( )
```

If a date was entered in paragraph 2.3.6.3, then the following message will be displayed:

***** OLAAS DATA RECORDING & ANALYSIS *****
25 JANUARY 1978

** ENTER DESIRED FUNCTION **
1 = ON-LINE DATA RECORDING & ANALYSIS
2 = OFF-LINE ANALYSIS & PLOT DISPLAY
3 = DATE ENTRY OR CHANGE

(2) 3

ENTER DATE
(25 JANUARY 1978)

2.3.6.4.3.2 Enter the desired date and the following message will be displayed:

***** OLAAS DATA RECORDING & ANALYSIS *****
25 JANUARY 1978

** ENTER DESIRED FUNCTION **
1 = ON-LINE DATA RECORDING & ANALYSIS
2 = OFF-LINE ANALYSIS & PLOT DISPLAY
3 = DATE ENTRY OR CHANGE

(2)

Note that if nothing was entered for the date then the date field is cleared.

2.3.7 User Program Analysis Operation.

2.3.7.1 Enter the letter "F" for the USER PROGRAM ANALYSIS operation and the following message will be displayed:

***** OLAAS PROCESSING & DISPLAY SYSTEM *****
EXECUTIVE PROGRAM

** ENTER DESIRED OPERATION **
D = DATA RECORDING & ANALYSIS
F = USER PROGRAM ANALYZER
G = FOURIER ANALYZER GRAPHICS

F
MAX BLOCKS #/SIZE/SPACE
4/ 2048/ 8192

2.3.7.2 The operator may now call any of the available User Programs into execution. Figure 2.3.7-1 is an example of User Program Y7906 with the question asked in that program.

2.3.7.3 To return to the OLAAS EXECUTIVE PROGRAM, "re-boot" the system using OPERATING PROCEDURES: 2.4.1 - OLAAS System Loading Procedure.

***** OLAAS PROCESSING & DISPLAY SYSTEM *****
EXECUTIVE PROGRAM

** ENTER DESIRED OPERATION **

D - DATA RECORDING & ANALYSIS
F - USER PROGRAM ANALYZER
G - FOURIER ANALYZER GRAPHICS

F
MAX BLOCKS #/SIZE/SPACE
4/ 2048/ 8192

Y 7906

PLEASE ENTER THE TYPE OF ANALYSIS DESIRED:

FOR STAVE INPUT, ENTER 1

FOR BEAM INPUT, ENTER 2

FOR ANALOG INPUT, ENTER 3

(1)

1

ENTER THE STAVE NUMBER:

(5)

16

PLEASE ENTER THE BLOCK SIZE DESIRED:

(1024)

1024

ENTER THE NUMBER OF TIMES THAT YOU WANT ME TO

AVERAGE THE INPUT:

(10)

15

HOW MANY HANNINGS WITHIN EACH AVERAGING LOOP??

(1)

1

WORKING.....

2.3.8 Fourier Analyzer Graphics Operation.

2.3.8.1 Enter the letter "G" for the FOURIER ANALYZER GRAPHICS operation and the following message is all that is displayed:

```
***** OLAAS PROCESSING & DISPLAY SYSTEM *****  
EXECUTIVE PROGRAM
```

```
** ENTER DESIRED OPERATION **
```

```
D = DATA RECORDING & ANALYSIS  
F = USER PROGRAM ANALYZER  
G = FOURIER ANALYZER GRAPHICS
```

G

2.3.8.2 The operator will enter at the Fourier Analyzer keyboard a JUMP and Ø, then press RETURN. The contents of Fourier Data Block Ø will be displayed on the screen. Note: Fourier Data Block Ø must be loaded prior to calling the Fourier Analyzer Graphics.

2.3.8.2 To return to the OLAAS EXECUTIVE Program, "re-boot" the system using OPERATING PROCEDURE:

2.3.9 Error Halts. During execution of the data collection program, various error conditions will cause the program to halt. When the computer halts, the display shows the halt instruction. These halts and their meanings are as follows:

102002	Beam former completion (DMA) interrupt did not occur within 1.2 seconds after completion of disc write.
102003	Beam former completion (DMA) interrupt waiting when disc write completed.
102004	A Time Base Generator tick was missed (irrecoverable).
102005	Processing between transmit gate and start of recording took too long.
102006	Disc write length not a multiple of 128 words.
102007	Disc storage area exceeded.
102010	Beam former input length request not a multiple of 16 words.
102012	No disc space allowed.
102013	No 2 consecutive clock samples alike out of 4.
102030	Beam former interrupt occurred.

2.4 BOOTSTRAP LOADING PROCEDURES

The following BOOTSTRAP LOADING PROCEDURES are used to load the OLAAS PROCESSING AND DISPLAY SYSTEM, make changes to, and generate the system:

- 2.4.1 OLAAS System Loading.
- 2.4.2 Read Time Executive (RTE) System Loading.
- 2.4.3 Fourier (BCS) System Loading.
- 2.4.4 Paper Tape Loading.
- 2.4.5 Writing to Disc.
- 2.4.6 Reloading HP-2100 Computer System Loader.

OLAAS PROCESSING & DISPLAY SYSTEM

Page: 1 of 1

OPERATING PROCEDURE: 2.4.1 - OLAAS SYSTEM LOADING PROCEDURE

The OLAAS SYSTEM LOADING PROCEDURE will "Bootstrap", or load in the OLAAS SYSTEM and start execution.

STEP NO.	DESCRIPTION	OBSERVATION
1.	At the HP-2100S COMPUTER OPERATOR PANEL	
2.	Set P = 077750	DISPLAY REGISTER = 077750
3.	Press S and then press CLEAR DISPLAY	DISPLAY REGISTER = 000000
4.	Press EXTERNAL PRESET and INTERNAL PRESET	
5.	Press LOADER ENABLE	
6.	Press HALT/CYCLE	DISPLAY REGISTER = 030000
7.	Press MEMORY DATA and clear BIT 9 of the DISPLAY REGISTER if set	
8.	Press RUN twice	DISPLAY REGISTER = 000002
9.	Set P = 000002	
	Press RUN	The following message is displayed on the CRT Terminal screen:
		OLAAS PROCESSING & DISPLAY SYSTEM EXECUTIVE PROGRAM
		** ENTER DESIRED OPERATION ** D = DATA RECORDING & ANALYSIS F = USER PROGRAM ANALYZER G = FOURIER ANALYZER GRAPHICS
	THE OLAAS SYSTEM LOAD IS COMPLETE	

OPERATING PROCEDURE: 2.4.2 - REAL TIME EXECUTIVE (RTE) SYSTEM LOADING PROCEDURE

The REAL TIME EXECUTIVE (RTE) SYSTEM LOADING PROCEDURE will "Bootstrap", or load in, the RTE SYSTEM.

The RTE SYSTEM is used with the FILE MANAGEMENT (FMGR) program for program creation and changes, also generating Fourier coreloads.

STEP NO.	DESCRIPTION	OBSERVATION
1. 2. 3. 4. 5. 6. 7.	<u>At the HP-2100S COMPUTER OPERATOR PANEL</u> Set P = 077750 Press S and then press CLEAR DISPLAY Press EXTERNAL PRESET and INTERNAL PRESET Press LOADER ENABLE Press HALT/CYCLE Press MEMORY DATA and set BIT 9 of the DISPLAY REGISTER Press RUN twice	DISPLAY REGISTER = 077750 DISPLAY REGISTER = 000000 DISPLAY REGISTER = 031000 The following message is displayed on the CRT Terminal screen: SET TIME
8. 9.	<u>THE RTE SYSTEM LOAD IS COMPLETE</u> <u>At the CRT TERMINAL KEYBOARD</u> Press the SPACE BAR Key in: RU,FMGR and then press RETURN	The CRT TERMINAL screen displays: * The CRT TERMINAL screen displays: *RU,FMGR (Note: The "prompt" on the CRT TERMINAL screen for FMGR is a colon (:).

THE FILE MANAGEMENT PROGRAM IS
LOADED AND EXECUTING

OLAAS PROCESSING & DISPLAY SYSTEM

Page: 1 of 1

OPERATING PROCEDURE:

2.4.3 - FOURIER BASIC CONTROL SYSTEM (BCS) LOADING PROCEDURE

The FOURIER BASIC CONTROL SYSTEM (BCS) LOADING PROCEDURE will "Bootstrap", or load in, the FOURIER (BCS) SYSTEM and start execution at the selected Coreload starting cylinder on the Fourier Disc. The FOURIER (BCS) SYSTEM is used to start the system for a given coreload so that a Fourier Analyzer Program (Keyboard Stack) may be written or execution of the coreload.

STEP NO.	DESCRIPTION	OBSERVATION
1.	At the HP-2100S COMPUTER OPERATOR PANEL	
2.	Set P = 077750	DISPLAY REGISTER = 077750
3.	Press S and then press CLEAR DISPLAY	DISPLAY REGISTER = 000000
4.	Press EXTERNAL PRESET and INTERNAL PRESET	
5.	Press LOADER ENABLE	
6.	PRESS HALT/CYCLE	
7.	Press MEMORY DATA and clear BIT 9 of the DISPLAY REGISTER if set	DISPLAY REGISTER = 030000
8.	Press RUN	
	Press B and then set the DISPLAY REGISTER to the starting cylinder number (in octal)	DISPLAY REGISTER = Cylinder Number (in octal)
		Coreload 0 = 1 Coreload 1 = 7 Coreload 2 = 15 Coreload 3 = 23 Coreload 4 = 31 Coreload 5 = 37 Coreload 6 = 45 Coreload 7 = 53
9.	Press RUN	
10.	Set P = 000002	DISPLAY REGISTER = 000002
11.	Press RUN	
	THE FOURIER (BCS) SYSTEM LOAD IS COMPLETE	

OLAAS PROCESSING & DISPLAY SYSTEM

Page: 1 of 1

OPERATING PROCEDURE: 2.4.4 - PAPER TAPE LOADING PROCEDURE

The PAPER TAPE LOADING PROCEDURE will cause a paper tape to be read in and stored in core memory. The input program shall be in binary form, containing absolute addresses.

STEP NO.	DESCRIPTION	OBSERVATION
1.	<u>AT THE PAPER TAPE READER</u>	Photoreader is illuminated
2.	Turn on the PAPER TAPE READER	Ensure tape is not installed backwards.
3.	Press LOAD and place paper tape in reader	
4.	Press READ	
5.	<u>At The HP-2100S COMPUTER OPERATOR PANEL</u>	DISPLAY REGISTER = 077700
6.	Set P = 077700	
7.	Press EXTERNAL PRESET and INTERNAL PRESET	The tape should now read in, and stop when complete.
	Press LOADER ENABLE	DISPLAY REGISTER = 102077
	Press RUN	(Note: Halt code = 102055 is an address error and halt code = 102011 is a checksum error.)
	<u>THE LOAD IS COMPLETE</u>	

OPERATING PROCEDURE: 2.4.5 - WRITING TO DISC PROCEDURE

The WRITING TO DISC PROCEDURE is used for dumping core to a disc pack. In the OLAAS System this procedure is used to write the Fourier Analyzer Programs (Keyboard Stack) into the desired coreload so that the Keyboard Stack will be executed when the coreload is entered.

STEP NO.	DESCRIPTION	OBSERVATION
1.	At the HP-2100S COMPUTER OPERATOR PANEL Set P = 077600	DISPLAY REGISTER = 077600
2.	Press A and then press CLEAR DISPLAY	DISPLAY REGISTER = 000000
3.	Press B and then set the DISPLAY REGISTER to the starting cylinder number (in octal), also set BIT 15	DISPLAY REGISTER = 1000xxx xx = Cylinder Number (in octal) Coreload 0 = 01 Coreload 1 = 07 Coreload 2 = 15 Coreload 3 = 23 Coreload 4 = 31 Coreload 5 = 37 Coreload 6 = 45 Coreload 7 = 53
4.	Press EXTERNAL PRESET and INTERNAL PRESET	The Computer will HALT and the DISPLAY REGISTER = 102077.
5.	Press RUN	(Note: If the correct HALT is not obtained, repeat the procedure.)
<u>WRITING TO DISC IS COMPLETE</u>		

OPERATING PROCEDURE: 2.4.6 - RELOADING HP-2100S COMPUTER SYSTEM LOADER PROCEDURE

The RELOADING HP-2100S COMPUTER SYSTEM LOADER PROCEDURE will re-install the System Loader if it has been cleared out of core memory. The System Loader is located in upper core and should never be reached, however, if it is cleared and Procedures 2.4.1 or 2.4.2 can not be executed, then re-install the System Loader using this procedure

STEP NO.	DESCRIPTION	OBSERVATION
1.	At the HP-2100S COMPUTER OPERATOR PANEL	
2.	Press HALT/CYCLE	HALT/CYCLE is illuminated
3.	Set M = 000020	DISPLAY REGISTER = 000020
	Press MEMORY DATA and then enter the following numbers (in octal) press INCREMENT M after each number except the last number	DISPLAY REGISTER = Number Entered
	103710	
	102310	
	024021	
	102510	
	001127	
	103710	
	102310	
	024026	
	102410	
	170001	
	006004	
	024020 (last number)	
4.	After entering last number (024020), then set B = 077700	DISPLAY REGISTER = 077700
5.	Set P = 000020	DISPLAY REGISTER = 000020
6.	Press EXTERNAL PRESET and INTERNAL PRESET	
	Continue on Page 2	

OPERATING PROCEDURE: 2.4.6 CONTINUED - RELOADING HP-2100S COMPUTER SYSTEM LOADER PROCEDURE

STEP NO.	DESCRIPTION	OBSERVATION
7.	<u>At the PAPER TAPE READER</u>	Photoreader is illuminated
8.	Turn on the PAPER TAPE READER	Ensure tape is not installed backwards
9.	Press LOAD and place paper tape P/N 05451-90007 in reader	
	Press READ	
10.	<u>At the HP2100S COMPUTER OPERATOR PANEL</u>	LOADER ENABLE is illuminated
	Press LOADER ENABLE	The tape should now read in, and stop when complete.
11.	Press RUN	HALT/CYCLE is illuminated
12.	Press HALT/CYCLE	
	<u>NOTE: Since the bootstrap tape is for a 16K system, the following changes must be made.</u>	
13.	Press LOADER ENABLE	LOADER ENABLE is illuminated
14.	Set M = 077772	DISPLAY REGISTER = 077772
15.	Set MEMORY DATA = 100100	DISPLAY REGISTER = 100100
16.	Press LOADER ENABLE	LOADER ENABLE not illuminated
	<u>NOTE: If the following tape does not load in, repeat steps 1 -16.</u>	
	<u>At the PAPER TAPE READER</u>	
17.	Press LOAD and place paper tape "EMMETT IPDBL" in reader	Ensure tape is not installed backwards.
18.	Press READ	
19.	<u>At the HP-2100S COMPUTER OPERATOR PANEL</u>	
	Set P = 077700	DISPLAY REGISTER = 077700
20.	Press EXTERNAL PRESET and INTERNAL PRESET	
21.	Press LOADER ENABLE	LOADER ENABLE is illuminated
22.	Press RUN	The tape should now read in, and stop when complete. (Repeat steps 1-16 if tape does not load)
	Continue on Page 3	

OPERATING PROCEDURE: 2.4.6 CONTINUED - RELOADING HP-2100S COMPUTER SYSTEM LOADER PROCEDURE

STEP NO.	DESCRIPTION	OBSERVATION
23.	Set S = 001710	DISPLAY REGISTER = 001710
24.	Set P = 0000002	DISPLAY REGISTER = 0000002
25.	Press EXTERNAL PRESET and INTERNAL PRESET	
26.	Press LOADER ENABLE	LOADER ENABLE is illuminated
27.	Press RUN	The Computer will HALT with the DISPLAY REGISTER = 102077, which signifies FDBBL is installed.
28.	<u>At the PAPER TAPE READER</u> Press LOAD and place the paper tape "FBOOT", P/N 05451-1098382, in reader; then position the tape to the second program on the tape <u>At the HP-2100S COMPUTER OPERATOR PANEL</u>	Ensure tape is not installed backwards
29.	Set P = 077700	DISPLAY REGISTER = 077700
30.	Press EXTERNAL PRESET and INTERNAL PRESET	
31.	Press LOADER ENABLE	LOADER ENABLE is illuminated
32.	Press RUN	The tape should now read in, and stop before finishing the tape
33.	Set S = 177600	DISPLAY REGISTER = 177600
34.	Set P = 0000002	DISPLAY REGISTER = 0000002
35.	Press EXTERNAL PRESET and INTERNAL PRESET	
36.	Press RUN	The remaining tape should now read in, and stop when complete. DISPLAY REGISTER = 102076 which indicates a successful load.

THE SYSTEM SHOULD NOW BE OPERATIONAL

2.5 CORE LOAD GENERATION PROCEDURES

The Fourier disc is composed of eight core loads, as follows:

- 0 OLAAS Executive
- 1 OLAAS Data Collection, Analysis and Executive
- 2 OLAAS Raw Data Collection
- 3 Reserved for swapping CL 1
- 4 Reserved for swapping CL 2
- 5 OLAAS User Programs
- 6 Fourier Analyzer Graphics
- 7 OLAAS Plot Display

Each of these core loads is built using the Fourier Cross Loader (FXL) and the File Copy Program (COPYF), operating under RTE. For each core load there is a separate command file and a transfer file, both of which control FXL. The command file is read by FXL and controls its functioning. The transfer file starts the operation of FXL. All of these files may be edited with the Interactive Editor. The Fourier core loads require the file MPSYS, containing the BCS system, be present. To modify the BCS system, it is necessary to modify transfer file MPTR, FXL command file MPCMCS, and/or BCS generation command file MPPC, and transfer control to MPTR. Table 2.5-1 lists the files accessed in generating a replacement core load.

To illustrate the process of core load generation task, for example, the case of a modification to the executive. The steps would be as follows:

1. Boot RTE
2. RU, FMGR
3. RU, EDITR
4. Make necessary modifications to the executive program source file, MP0003. Leave edited file in MP0003 and logical source tracks by terminating EDITR with command ELR.
5. LG,2
6. RU,FTN4,2,99 (assemble new program)
7. PU,Y0003 (Purge old relocatable file)
8. SA,LG,#Y0003:::-1 (Save new relocatable file)
9. TR,MPTR0 (Create file containing #Y0003 and subroutines)
10. TR,MPDCP (Run FXL, creating absolute file DCPSYS)

11. RU,COPYF (Run File Copy program)
12. Answer COPYF questions as follows:
CORE SIZE: 32
CORELOAD NUMBER: 0 (for this example)
DISC LU NUMBER: 9
SYSTEM FILE NAME: DCPSYS
OVERLAY FILE USED: NO
PROCEED: YES

2.6 FOURIER ANALYZER PROGRAM (KEYBOARD STACK)

The Fourier Analyzer Program Keyboard Stack, is a sequence of commands that the Fourier Analyzer will perform automatically. Reference the Fourier Analyzer System - Operating Manual, 04541-90199 with supplements, Section VII for a complete description of this type of program.

In the OLAAS system, the USTRT subroutine transfers control to Label 0 (zero) in the keyboard program stack. The way that the analysis program, or any other program in any other core load, is started immediately, rather than requiring operator action, is that the keyboard program stack contains the program:

```
L    0
Y    0002 (for example)
```

This must be entered and stored on the disc whenever the core load in question is regenerated. To do so, follow this procedure:

1. Load the core load in question using OPERATING PROCEDURE:
2.4.3 - Fourier (BCS) System Loading.
2. At the Fourier Analyzer keyboard, press:
REPLACE, ENTER
LABEL 0, ENTER
USER PROG 0002, ENTER
END, ENTER
TERM, ENTER
3. Press L, ENTER (the program stack will be displayed).
4. At the computer, write the program stack to disc using OPERATING PROCEDURE: 2.4.5 - Writing to DISC.

OLAAS PROCESSING & DISPLAY SYSTEM

CORELOAD GENERATION FILE NAMES

CORELOADS:	CL 0	CL 1	CL 2	CL 5	CL 6	CL 7
MAIN PROGRAM SOURCE FILE:	MP003A	MP002A	Y0001	(Fourier System)	(Fourier System)	MP007A
MAIN PROGRAM OBJECT FILE:	#Y0003	Y10020	#Y0001	(Fourier System)	(Fourier System)	Y1007A
BCS GENERATION COMMAND FILE:	RTFC	N/A	RTFC	N/A	N/A	N/A
FXL COMMAND FILE:	RTCM	MPCM1	RTCM	MPCM5	MPCM6	MPCM7
FXL TRANSFER FILE:	MPTR0	MPTR1	MPTR2	MPTR5	MPTR6	MPTR7
SYSTEM FILE FOR "COPYF":	DCPSYS	MPSYS	DCPSYS	MPSYS	MPSYS	MPSYS
OVERLAY FILE:	N/A	RAF	N/A	RAF	RAF	RAF
SPECIAL GENERATION PROCEDURES:	TR,MPTR0	TR,MPTR1	TR,DCPTR	TR,MPTR5	TR,MPTR6	TR,MPTR7
	TR,RTTR		TR,RTTR			

BCS SYSTEM GENERATION: MPSYS

FXL TRANSFER FILE: MPTR

FXL COMMAND FILE: BR1CSC

BCS COMMAND FILE: RAFFC

SPECIAL GENERATION PROCEDURE: TR,MPTR

DISC FORMATTING PROCEDURE:

RU,FMEXL

Figure 2.5-1

2.7 DISC COPY - MADIU PROGRAM

DISC COPY uses the MADIU Program which is a Hewlett Packard Magnetic Tape/Disc Utility program for copying disc images. MADIU is fully described in the Magnetic Tape/Disc Utility Installation/Operation Note, 0541-90307. The OPERATING PROCEDURE: 2.7 - Disc Copy Procedure (MADIU Program) will execute the MADIU program in the OLAAS System environment.

The following messages is an example of a disc copy from the lower sub-channel, 0, to upper subchannel, 1.

```
(FOR 7900) TYPE "LO" "DU" "VE" "CO" OR "TE"
00
SOURCE SUBCHANNEL # ?
0
"DESTINATION SUBCHANNEL # ?"
1
```

OPERATING PROCEDURE: 2.7 - DISC COPY PROCEDURE (MADIU PROGRAM)

The DISC COPY PROCEDURE uses the MADIU Program, a HP Mag Tape/Disc Utility program for copying disc images. MADIU is fully described in the Magnetic Tape/Disc Utility Installation/Operation Note, 05451-90307, (see 4.0 References). The version used with the Fourier Bootstrap Software is configured with 4k SIO driver and no MAG TAPE capability so that it will fit on disc cylinder 0. Other operational characteristics are the same.

STEP NO.	DESCRIPTION	OBSERVATION
1.	At the PAPER TAPE READER Load the unconfigured MADIU binary paper tape using OPERATING PROCEDURE 2.4.4 - Paper Tape Loading	The MADIU Program is successfully loaded
2.	At the HP-2100S COMPUTER OPERATOR PANEL Set P = 000120	DISPLAY REGISTER = 000120
3.	Press EXTERNAL PRESET and INTERNAL PRESET	
4.	Press RUN	Answer the questions on the CRT Terminal screen
5.	At the CRT TERMINAL Answer the question of what operation to perform = CO (CO - Copy disc to disc (with auto verify))	The CRT TERMINAL screen displays: TYPE "LO" "DU" "VE" "CO" OR "TE"
6.	Press LF	Next question is displayed
7.	Answer the question of which subchannel contains the "source" to be copied. (NOTE: The FIXED FLATTER is 0 and the REMOVABLE FLATTER is 1)	The CRT TERMINAL screen displays: SOURCE SUBCHANNEL # ?
8.	Press LF	Next question is displayed
9.	Answer the question of which subchannel is to be the "destination" of the copy	The CRT TERMINAL screen displays: DESTINATION SUBCHANNEL # ?
10.	Turn the PROTECT/OVERRIDE switch to "override" (NOTE: The PROTECT/OVERRIDE switch is located on the 7900 DISC Unit behind the panel)	
11.	Press LF	DISPLAY REGISTER = Count up from 000001 to 000312
12.	Turn the PROTECT/OVERRIDE switch to "protect"	Any error displayed may be found in the MADIU Operation Note
	DISC TO DISC COPY IS COMPLETE	

3.0 OLAAS DISC STORAGE

3.1 INTRODUCTION

The OLAAS Disc Storage utilizes the Hewlett Packard 7900A Cartridge Disc Drive as the mass store device on the Fourier System. The HP7900A disc has an upper (removable cartridge) and lower (fixed) discs. The OLAAS system utilizes both discs. The lower (fixed) disc contains the RTE Software System, OLAAS source and object program files, and system generation files. See Appendix C for the RTE System Disc Directory. The upper (removable cartridge) disc contains the OLAAS Processing and Display System, referred to as the OLAAS Fourier Disc.

3.2 OLAAS FOURIER DISC LAYOUT

The OLAAS Fourier Disc layout is defined by the FMTXX1 program, that has been configured for this system, and is executed when the basic system (MPSYS) is generated. See Appendix B for the FMTXX1 program listing. Figure 3.2-1 is the OLAAS Fourier Disc configuration.

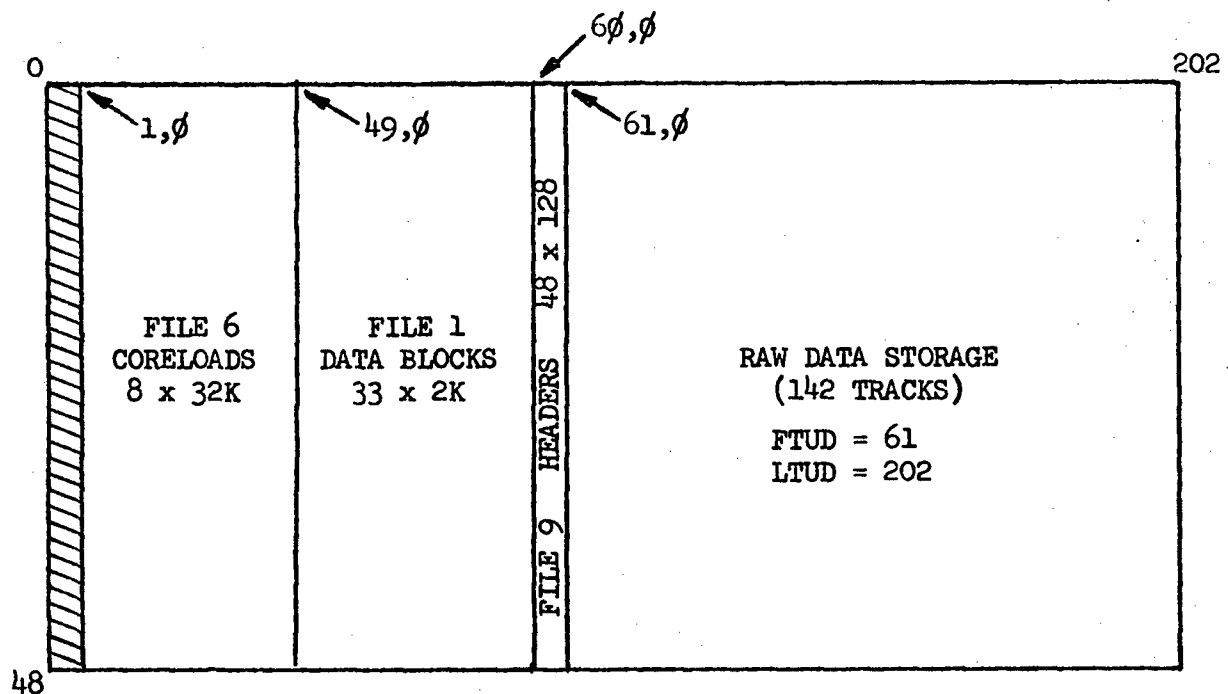


FIGURE 3.2-1 OLAAS Fourier Disc Configuration

3.2.1 FILE STRUCTURE

FILE 1 - DATA BLOCK

File 1, Disc Data Block, is set up for storage of data. Data block 0 is used for the system common data vector, ICOM. Each ping has reserved three (3) data blocks with the capacity for storing 1536 data points for each ping, with a maximum of ten (10) pings (data blocks 1-30) with 31 and 32 not used.

During analysis of raw data recorded on the disc, each amplitude is tested against a threshold. For each point which exceeds the analysis threshold, four integers are saved in the data blocks: 1) beam direction, 2) range index, 3) compressed amplitude and 4) doppler bin.

FILE 6 - SYSTEM CORELOAD

File 6, System Coreload, contains core images of the eight OLAAS coreloads. Using core swapping, the various coreloads are brought into the system for execution or returned to the disc. The OLAAS coreloads are as follows:

<u>Coreload</u>	<u>Function</u>
0	OLAAS Executive Program
1	OLAAS Analysis & System Executive Subroutine
2	OLAAS Raw Data Collection Subroutine
3	Copy of coreload 1 (used for processing)
4	Copy of coreload 2 (used for multi-ping recording)
5	OLAAS User Programs Analysis
6	Fourier Analyzer Graphics
7	OLAAS Plot Display Subroutine

FILE 9 - HEADERS

File 9, Headers, are associated with the data blocks and contain the data qualifiers.

3.2.2 RAW DATA STORAGE

Disc tracks 61 through 202, 142 tracks, are used for Raw Data storage (recording) of Buoy Data. The actual amount of data recorded is dependent upon number of beams, pings, and recording time.

3.3 OLAAS FOURIER DISC GENERATION

The generation of the OLAAS Fourier Disc is the result of creating the various coreloads. The system configuration is used when the coreloads are generated and Operating Procedure: 2.5 - Coreload Generation Procedure, when executed will create a coreload. Table 3.3-1 shows the files used to generate the coreloads.

OLAAS PROCESSING & DISPLAY SYSTEM

CORELOAD GENERATION FILE NAMES

CORELOADS:	CL 0	CL 1	CL 2	CL 5	CL 6	CL 7
MAIN PROGRAM SOURCE FILE:	MP003A	MP002A	Y0001	(Fourier System)	(Fourier System)	MP007A
MAIN PROGRAM OBJECT FILE:	#Y0003	Y10020	#Y0001	(Fourier System)	(Fourier System)	Y1007A
BCS GENERATION COMMAND FILE:	RTPC	N/A	RTPC	N/A	N/A	N/A
FXL COMMAND FILE:	RTCM	MPCM1	RTCM	MPCM5	MPCM6	MPCM7
FXL TRANSFER FILE:	MPTR0	MPTR1	MPTR2	MPTR5	MPTR6	MPTR7
SYSTEM FILE FOR "COPYF":	DCPSYS	MPSYS	DCPSYS	MPSYS	MPSYS	MPSYS
OVERLAY FILE:	N/A	RAF	N/A	RAF	RAF	RAF
SPECIAL GENERATION PROCEDURES:	TR,MPTR0	TR,MPTR1	TR,DCPTR	TR,MPTR5	TR,MPTR6	TR,MPTR7
	TR,RTTR		TR,RTTR			

BCS SYSTEM GENERATION: MPSYS

FXL TRANSFER FILE: MPTR

FXL COMMAND FILE: BR1CSC

BCS COMMAND FILE: RAFPC

SPECIAL GENERATION PROCEDURE: TR,MPTR

DISC FORMATTING PROCEDURE:

RU,FMFXL

4.0 REFERENCES

4.1 APPLICABLE DOCUMENTS

Following is a list of manufacturer's documents that describe the software and operator interface to the special purpose OLAAS PROCESSING & DISPLAY SYSTEM and the peripheral equipment. These documents contain supplementary information which is not duplicated in this manual.

HEWLETT-PACKARD COMPANY

92060-90012	RTE-II: A Guide for New Users
92001-93001	REE-II Programming and Operating Manual
92060-90014	RTE Interactive Editor Reference Manual
24998-90001	DOS-RTE Relocatable Library Reference Manual
5951-1321	HP Fortran IV reference Manual
92060-90005	RTE Assembler Reference Manual
92060-90010	RTE & BSM Pocket Guide
HP Fourier Analyzer System 5451B	
05451-90199	System Operating Manual
05451-90268	Supplement: March 1975 - System Operating Manual
02-5952-0651	Fourier Analyzer Training Manual (Application Note 140-0)
05451-90333	Magnetic Tape/Disc Option
05451-90307	Magnetic Tape/Disc Utility (MADIU)
05451-90396 & 05451-90434	Supplements: MADIU
05451-98393	Fourier Cross Loader FXL Operating & Software Note

HP PERIPHERALS

12539-90008	Time Base Generator
12566-90015	Microcircuit
12908-90011	Writeable Control Store
05451-90217	Data/Control Interface
12880-90001	Keyboard-Display Terminal
12531-90033	Buffer Teleprinter Interface
02752-90004	Teleprinter 2752A
12597-90022	Tape Reader
02748-90032	Tape Reader 2748B

TEKTRONIX, INC.

	Tektronix 4012 Computer Display Terminal User Instruction Manual
062-1428-01	Plot-10/MiniComputer - 4010 Series User Manual

5.0 OLAAS SOFTWARE DOCUMENTATION (Programs/Subroutines Descriptions, Flow Diagrams and listings)

5.1 SOFTWARE OVERVIEW

The OLAAS Software System uses the Hewlett Packard 5451B FOURIER ANALYZER System hardware, including a HP7900 Disc Memory. Because of the size of the OLAAS System programs, and a need to maintain an initialized version of the analysis and data collection programs, it was necessary to segment the programs into separate coreloads. The HP2100S Computer has 32K of core memory, and the total system can accommodate 8 Fourier coreloads, using the HP7900 Disc Memory. The coreloads are organized as shown:

<u>CORELOAD</u>	<u>FUNCTION</u>
0	OLAAS Execution Program
1	OLAAS Analysis and System Executive Subroutine
2	OLAAS Raw Data Collection Subroutine
3	Copy of coreload 1 (used for processing)
4	Copy of coreload 2 (used for multi-ping recording)
5	OLAAS User Programs Analysis
6	Fourier Analyzer Graphics
7	OLAAS Plot Display Subroutine

The key to the system is the transferring of control from one coreload to another (core swapping or overlay).

The core swapping uses the core dump (CDUMP) and core load (CLOAD) subroutines. The core dump (CDUMP) routine writes all of core into a specified coreload area of the disk. The core load (CLOAD) routine reads a specified coreload and transfers to an address given in the calling sequence. Because the coreload routine must be overlaid by the program it is reading, it uses NIBBL, the H.P. loader which resides in an area of core (77600-77677) which is not overlaid. For this reason, NIBBL must be configured into the system. CLOAD modifies NIBBL so that when the coreload is completed, it transfers to the address specified in the CLOAD calling sequence. For the data collection coreload, it is 2, and for any coreload containing the Fourier software, it is the address of USTRT, which must be the first program unit loaded after the BASIC System and must be in base page. (This address is usually 51.) USTRT transfers control to Label 0 (zero) in the keyboard program stack. The way that the analysis program, or any other program in any other coreload, is started immediately, rather than requiring operator action.

The OLAAS Software System is composed of four (4) major functions which direct the processing of data. These functions are executive control, analysis and system control, raw data collection, and detection plot display. Additional function, which are part of system, user program and graphic displays are called separately by the executive.

The software was written in a modular approach which serves to minimize the amount of core memory storage required for the whole system. Therefore, the major functions required various subroutines to accomplish the overall task. The major function with the necessary subroutines is used to make the various coreloads. This section contains all the programs and subroutines used by the system. Also included are the user programs, and the TRANSFER, COMMAND, and PCS ANSWER Files used to generate the system and/or various coreload.

5.1.1. BASIC OLAAS SYSTEM CONTROL LOGIC

The Basic OLAAS System Control Logic is shown in Figure 5.1-1. The following description is in support of that figure. For detail operations refer to the flow diagram for each program or subroutine. Note that control is based on the use of "Core Swapping" and the "Not-First-Time Switch" (NFTSW).

The OLAAS System is loaded and enters the EXECUTIVE program, Y0003, which displays the system operation message. The operator selects the desired operation by input of the correct letter. Control is transferred to the proper coreload. The available operation are: D (Data Recording and Analysis), F (User Program Analysis), and G (Fourier Analyzer Graphics).

(1) The operator selects a "D" and control is transferred to coreload 1 (Y1002). Initially the NFTSW=1 and the system is initialized. NFTSW is set to "5" and control is transferred to coreload 3 (Y1002), which is a copy of coreload 1, for processing. NFTSW=5 causes the operating mode questions to be asked.

(2) The operator selects "3" and the date may be changed or entered and control returns to ask the operating mode questions again.

(3) The operator selects "1" and the On-Line Analysis questions are asked NFTSW is set to "2" and control is transferred to coreload 2 (Y0001) for Raw Data Collection.

(4) The first time Y0001 is entered the recording questions are asked and then raw data is collected. A copy of coreload 2 is written into coreload 4 for multi-ping recording, and control is transferred to coreload 3 (Y1002).

(5) NFTSW=2 and the single ping analysis is done for the ping just recorded. NFTSW is set to "4" and control is transferred to coreload 7 (Y1007) for an analysis display.

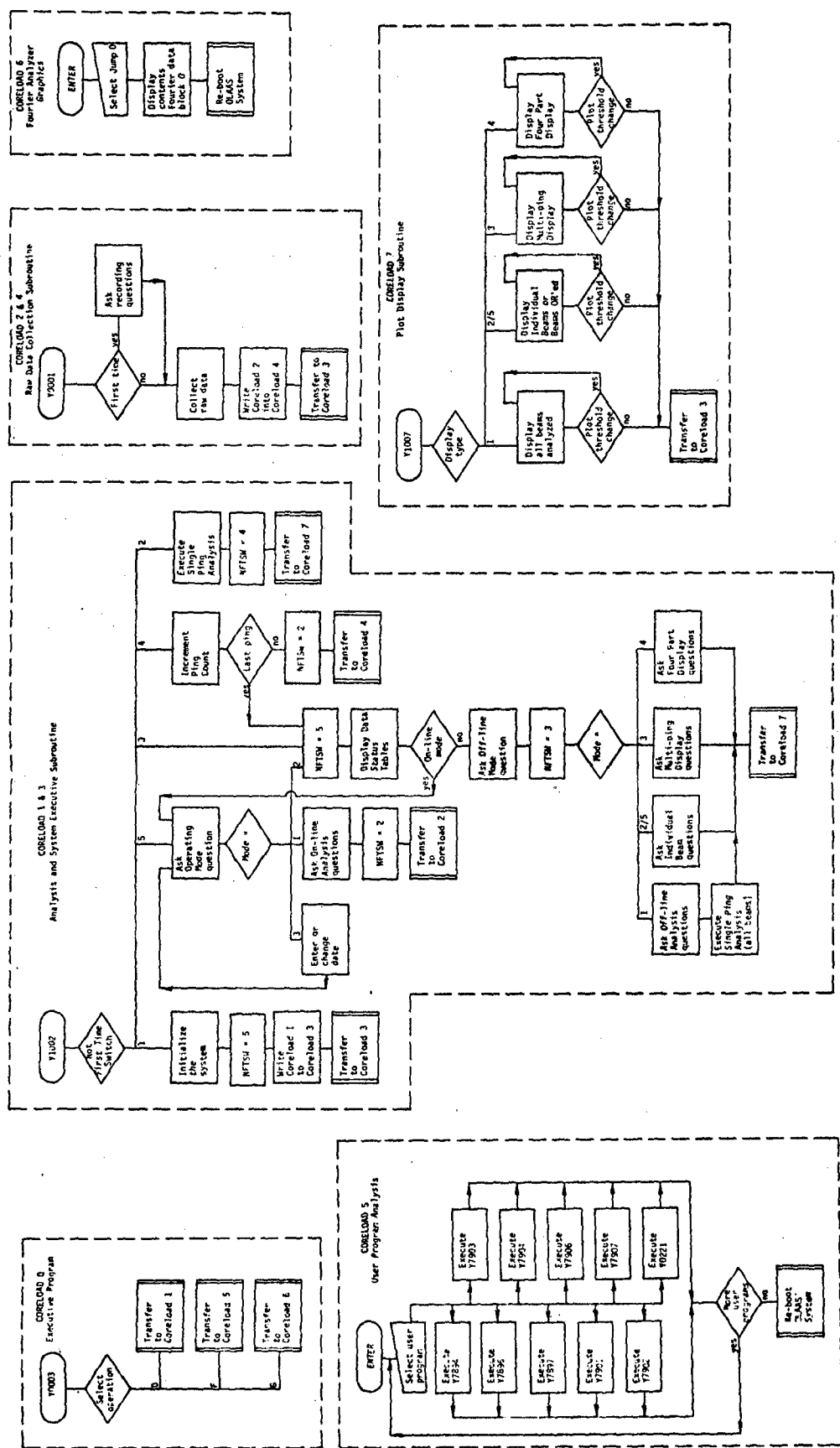


Figure 5.1-1. Basic OLAAS System Control Logic

(6) Test for the type of display is made and the plot is displayed. A check is made for a plot threshold change and if none control is transferred to coreload 3 (Y1002).

(7) NFTSW=4 and the ping count is incremented and checked for "last ping". If not last ping, the NFTSW is set to "2" and control is transferred to coreload 4 (Y0001) to record the next ping.

(8) This is not the first time Y0001 was entered, therefore the same parameter are used to collect raw data for the next ping. A new copy of coreload 2 is written into coreload 4 and control is transferred to coreload 3 (Y1002).

(9) NFTSW=2 repeats operation (5) and (6) above.

(10) NFTSW=4 and the ping count is incremented and checked for "last ping". If last ping, then NFTSW is set to "5" and the DATA STATUS TABLES are displayed. A check for ON-LINE mode is made and being set, control returns to ask the operating mode questions again.

(11) The operator selects "2" and the NFTSW is set to "5" and the DATA STATUS TABLES are displayed. A check for ON-LINE mode is made and being off, the OFF-LINE mode question is asked. The operator selects the desired off-line mode by entering the correct value. The available mode are: 1 (Analysis of Previously Recorded Data), 2 (Individual Beam Display), 3 (Multi-Ping Display), and 4 (Four Part Display).

(12) Note that the control in the "Off-Line" mode is basic the same for each selection. Therefore, this description is only for Analysis of Previously Recorded Data selection and the other selections shall reference this flow.

(13) The operator selects "1" and the NFTSW is set to "3". The Off-Line Analysis questions are asked and Single Ping Analysis is executed, for all beam requested. Control is transferred to coreload 7 (Y1007) for the analysis display.

(14) Test for the type of display is made and the plot is displayed. A check is made for a plot threshold change and if none, control is transferred to coreload 3 (Y1002).

(15) NFTSW=3 and the NFTSW is set to "5". The DATA STATUS TABLES are displayed and the ON-LINE mode is checked. Being Off-Line mode question is asked.

(16) With analyzed data available the operator may select any of the other three (3) off-line modes. The display questions will be asked and control is transferred to coreload 7 (Y1007) for the display. Operations (14) and (15) are repeated for each mode selection.

(17) The operator has two (2) options to restart the system. Enter a "12345" will cause control to return to coreload 0 (Y0003), Executive Program. Enter a "11111" will cause control to return to coreload 3 (Y1002) with NFTSW=5 and the operating mode question will be asked. Note that this option is only available when in coreload 3 (Y1003), Analysis and System Control Subroutines.

(18) The operator, when in the Executive Program (Y0003), may select User Program Analysis operation. Enter an "F" and control is passed to coreload 5, which contains the available user program.

(19) The operator enters the call for the user program selected and that program is executed. When execution is completed, the operator may select another user program or "re-boot" the OLAAS system to return control to the Executive Program (Y0003).

(20) The operator, when in the Executive Program (Y0003), may select Fourier Analyzer Graphics operation. Enter a "G" and control is passed to coreload 6, which contains the graphics package. Entering a "JUMP 0" command will cause the contents of Fourier Data Block 0 to be displayed. Re-boot the OLAAS system to return control to the Executive Program (Y0003).

5.1.2 SPECIAL SUBROUTINES

5.1.2.1 HEADR Subroutines. HEADR reads the header record from the disk. As there is a header record for each ping, the user specifies the ping number (first ping is ping 1) for which the header is to be read. Calling sequence is as follows:

CALL HEADR (IPING, IDATA)

where IPING is the ping number and IDATA is a 16 word array into which the header record is to be deposited. The content of the header record is as follows (all data items are integer format, with the exception of time, which is BCD):

Word	1	TIME	time code, hours, BCD
	2		time code, minutes and seconds, BCD
	3	ANGL1	heading of beam 1, degrees
	4		not used
	5	FSTB#	number of first beam recorded
	6	FBANG	heading of first beam recorded, degrees
	7	PING	ping number
	8	#BMS	number of beams recorded
	9	S	source (1=on line, 0=recorder)
	10	TOTPG	total number of pings (also known as M)
	11	K	delay before record, seconds
	12	N	length of record, seconds
	13		reserved
	14-16		not used

5.1.2.2. SOUND Velocity Calculation Program. SOUND converts temperature verses depth profile to a sound velocity versus depth profile. A standard salinity of 34PPM is assumed at all depths. The following is the operating instructions for the SOUND program:

STEP NO.	DESCRIPTION	OBSERVATION
1	Bootstrap load the RTE Software System and start the File Manager	RTE software system is loaded and file manager system is operating.
2	Enter and execute the following FMGR operator commands :LG,1↓ :MR,SOUND↓ :RU,LOADR,99↓ :RU,SOUND↓	Allocate number tracks to load and go track area. Transfer relocatable program file to LG tracks. Run loader program, to load program from LG area Run SOUND program.
3	Enter a string of depths in feet and temperatures in degrees F separated by a comma.	Each entry is displayed.
4	Enter a temperature of 0 to terminate the string.	The SOUND velocity table is displayed.
5	Terminate operation by entering a negative number for depth.	Operation returns to FMGR.

5.2 OLAAS PROCESSING AND DISPLAY SYSTEM

The OLAAS PROCESSING AND DISPLAY SYSTEM allows an operator to collect and analyze ERAPS Data subject to user-selected parameters. The user/operator may select the number of beams analyzed, look directions, range and doppler window. The type of display plotting is also under the user/operator control.

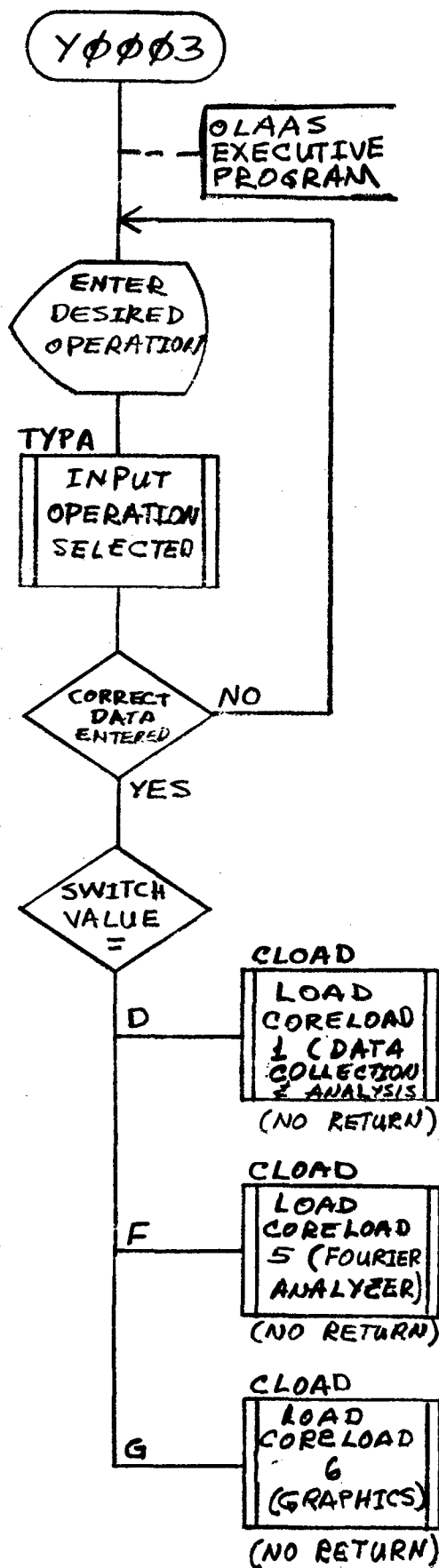
The following descriptions, flow diagrams, and listings are for the major functions of the system:

Y0003 - OLAAS Executive Program

Y1002 - OLAAS Analysis and System Executive Subroutine

Y0001 - OLAAS Raw Data Collection Subroutine

Y1007 - OLAAS Plot Display Subroutine

OLAAS EXECUTIVE PROGRAM (Y0003)
FLOW CHART

MP003A T=00004 IS ON CR00001 USING 00011 BLKS R-0020

```
0001 FTN
0002 PROGRAM Y0003
0003 *****
0004 *
0005 * TITLE: OLAAS EXECUTIVE PROGRAM
0006 *
0007 * MODULE: Y0003 SOURCE: MP003A VERSION: X01
0008 * OBJECT: #Y0003
0009 *
0010 * AUTHOR: R. A. FULLERTON DATE: 1976
0011 *
0012 * MODIFICATIONS:
0013 * 1. CHANGE THE EXECUTIVE QUESTION.
0014 * 12/09/77 W. BOWEN
0015 *
0016 * MODULE FUNCTION:
0017 * THE OLAAS EXECUTIVE PROGRAM IS THE INITIAL ENTRY
0018 * INTO THE OLAAS PROCESSING & DISPLAY SYSTEM, AND IS
0019 * IN CORELOAD 0. THE EXECUTIVE CALLS INTO EXECUTION
0020 * THE VARIOUS OPERATIONS OF THE SYSTEM (DATA COLLECTION,
0021 * USER PROGRAM ANALYZER, & FOURIER ANALYZER GRAPHICS).
0022 * NOTE: THIS IS A BCS PROGRAM & NO FOURIER ANALYZER
0023 * SOFTWARE IS LOADED.
0024 *
0025 * CALLING SEQUENCE:
0026 *
0027 * OLAAS SYSTEM BOOT LOAD PROCEDURE
0028 *
0029 * *****
0030 *
0031 *
0032 * > > > > DISPLAY THE EXECUTIVE QUESTIONS
```

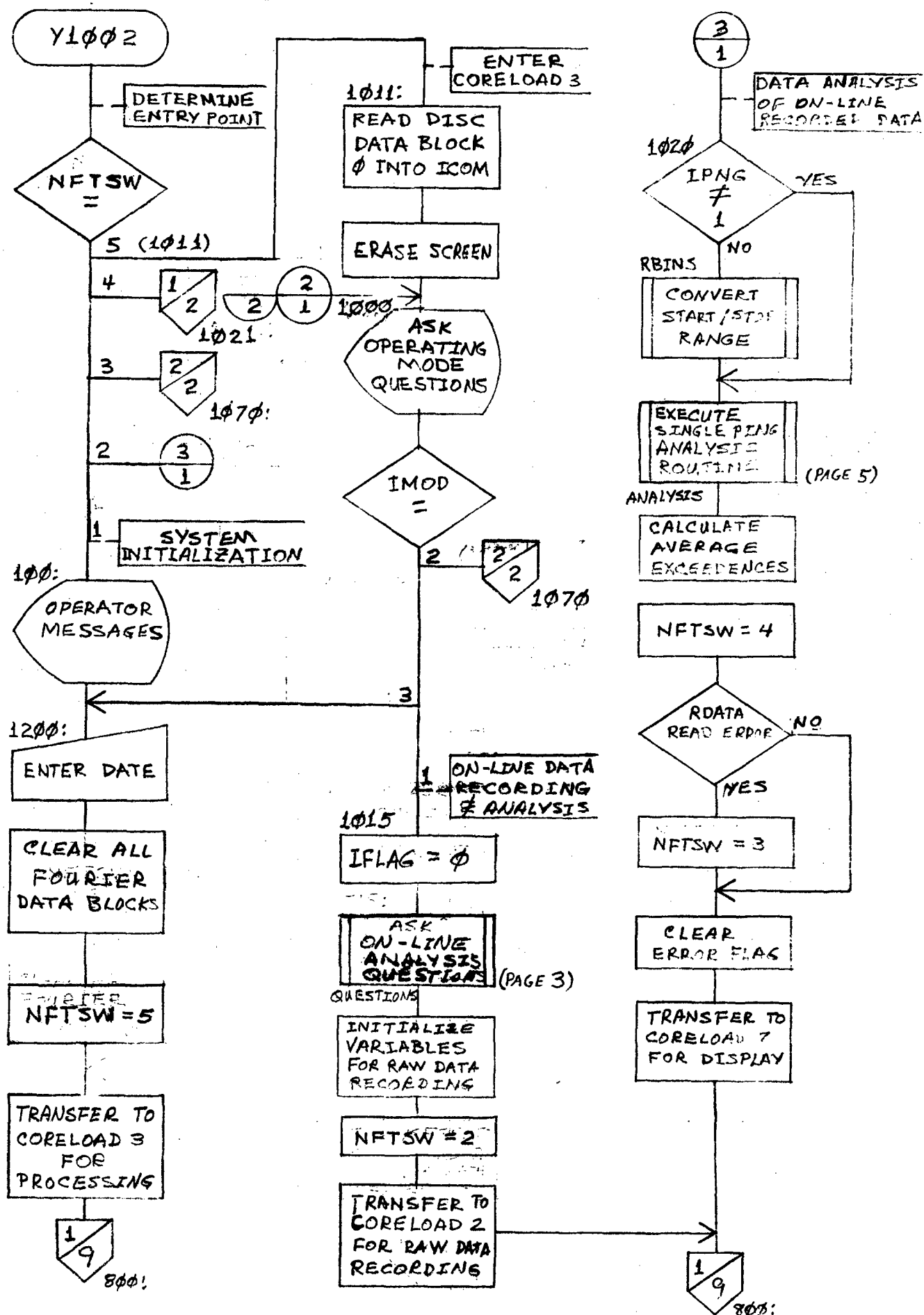


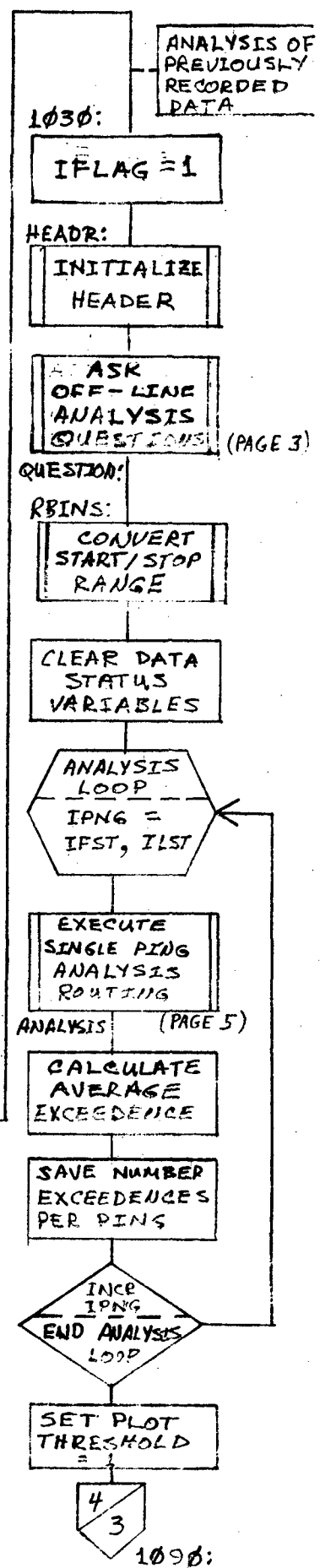
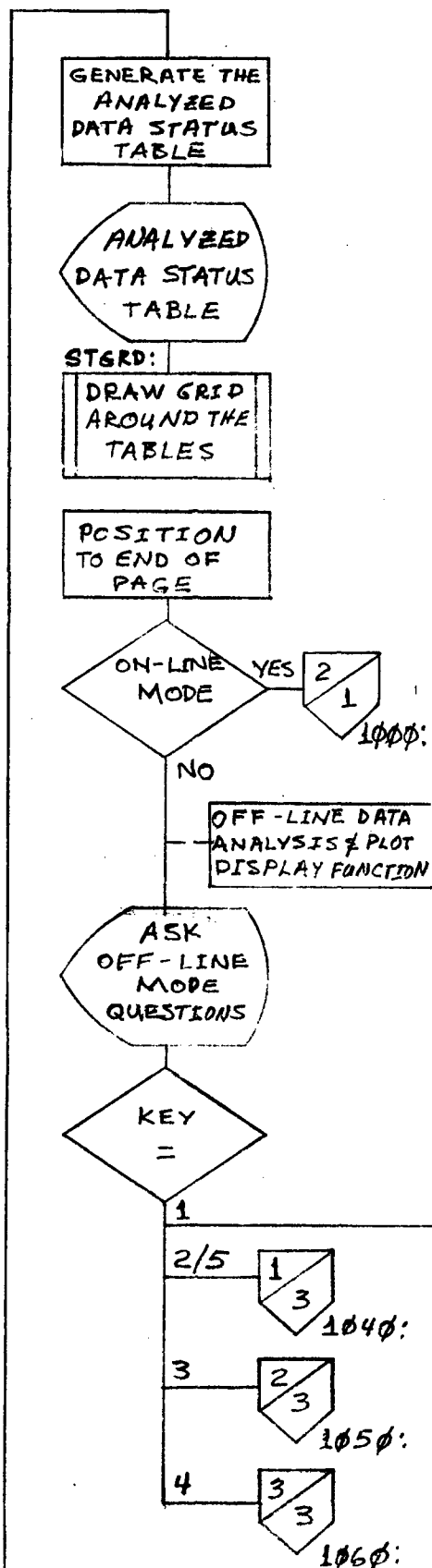
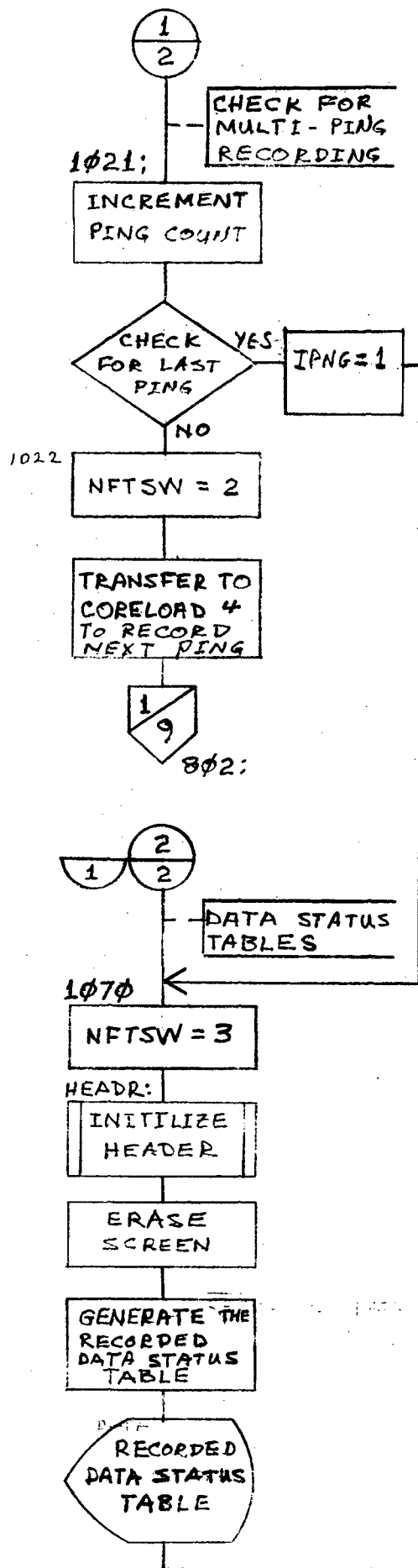
```

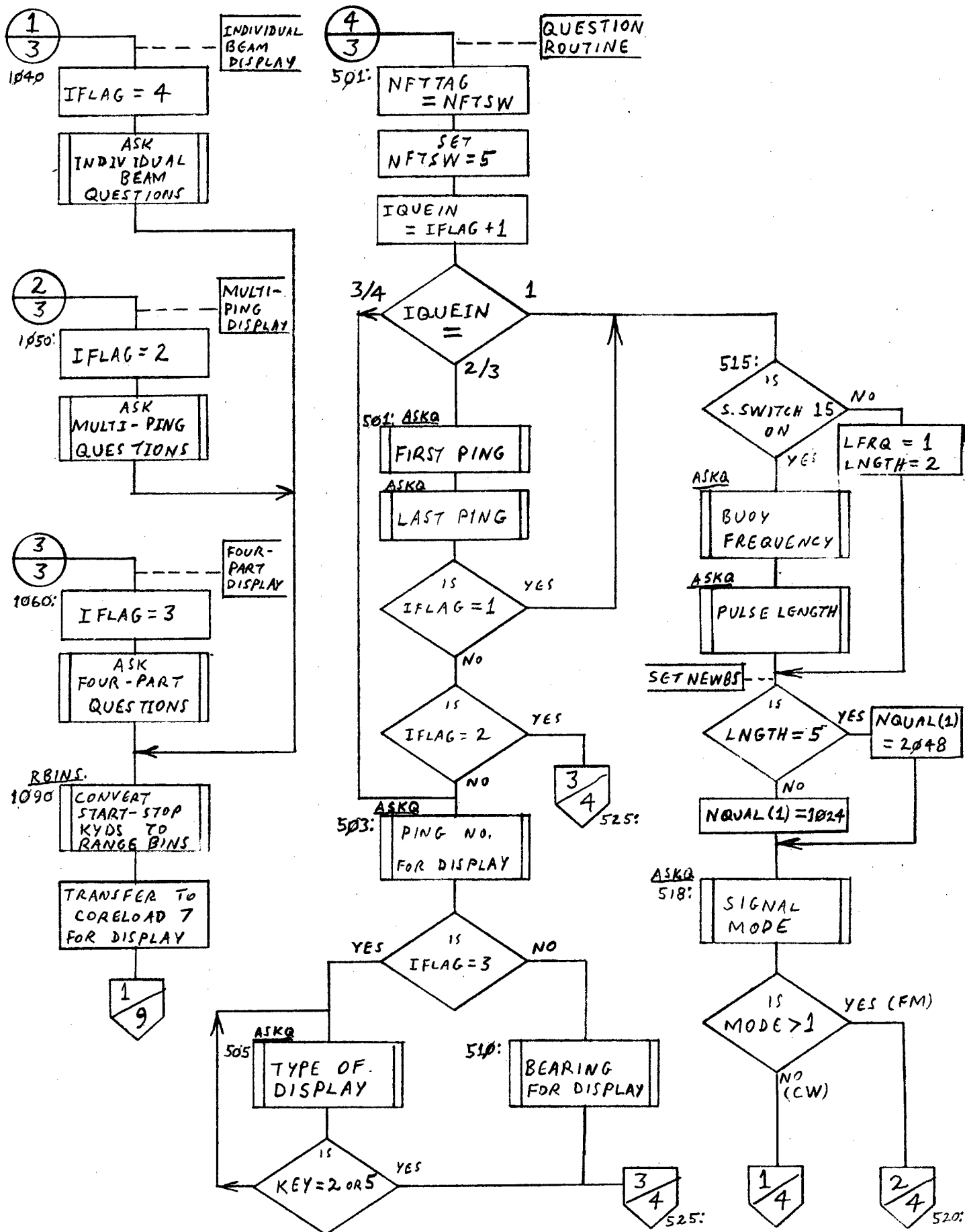
0033      10 WRITE (2,20)
0034      20 FORMAT(/"***** OLAAS PROCESSING & DISPLAY SYSTEM *****"/
0035      +15X,"EXECUTIVE PROGRAM"/9X,
0036      +"** ENTER DESIRED OPERATION **"/
0037      +9X,"D = DATA RECORDING & ANALYSIS"/
0038      +9X,"F = USER PROGRAM ANALYZER"/
0039      +9X,"G = FOURIER ANALYZER GRAPHICS")
0040
0041      C
0042      CALL TYP A (I,IFLG)
0043
0044      C > > > TRANSFER TO OPERATION SELECTED
0045
0046      IF (IFLG) 10,10,30
0047      C
0048      30 I=I+20000B
0049      IF (I .EQ. 2H D) CALL CLOAD (07B,51B)
0050      IF (I .EQ. 2H F) CALL CLOAD (37B,2)
0051      IF (I .EQ. 2H G) CALL CLOAD (45B,2)
0052      C
0053      GO TO 10
0054      C
0055      END
0056      END$

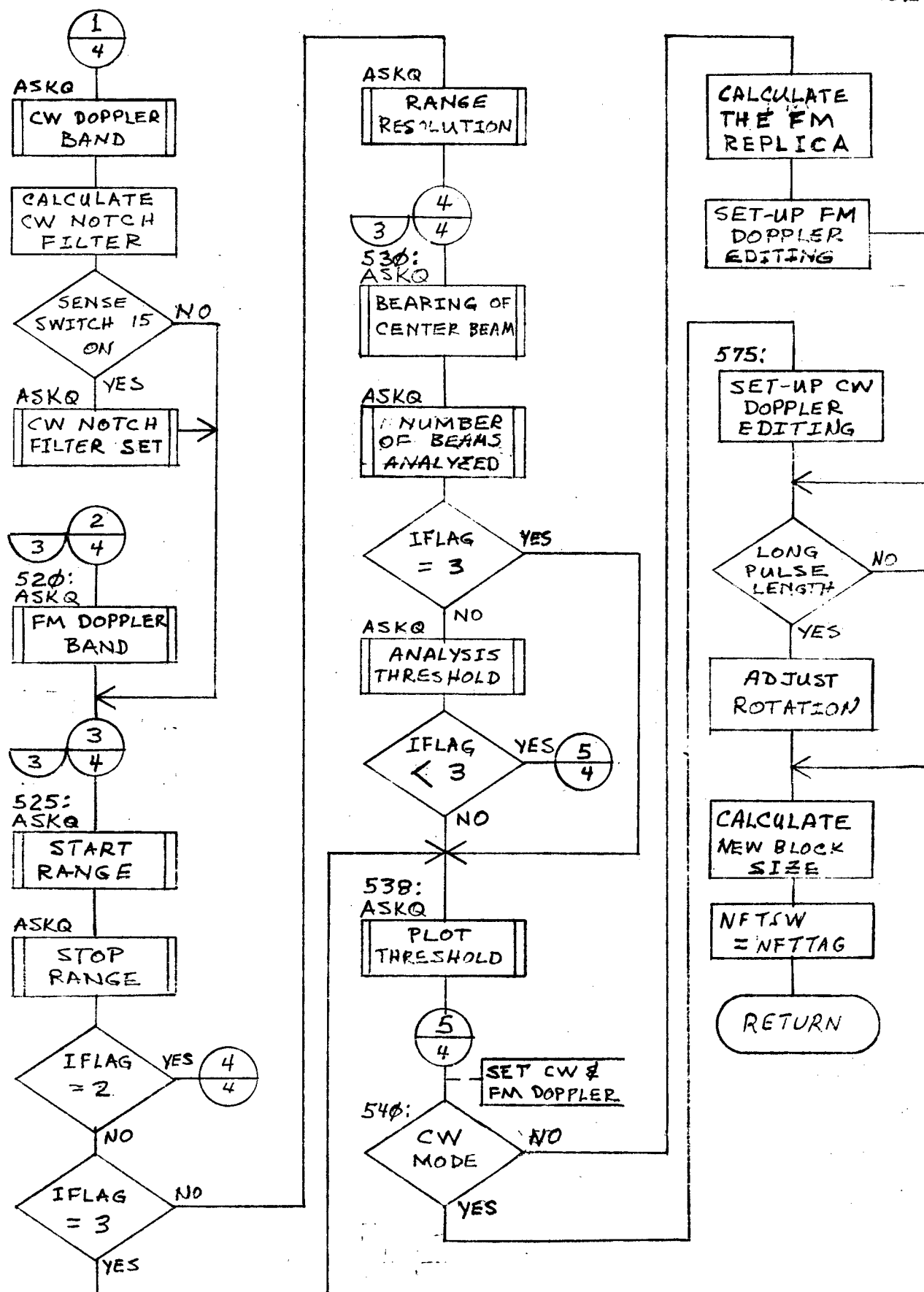
```

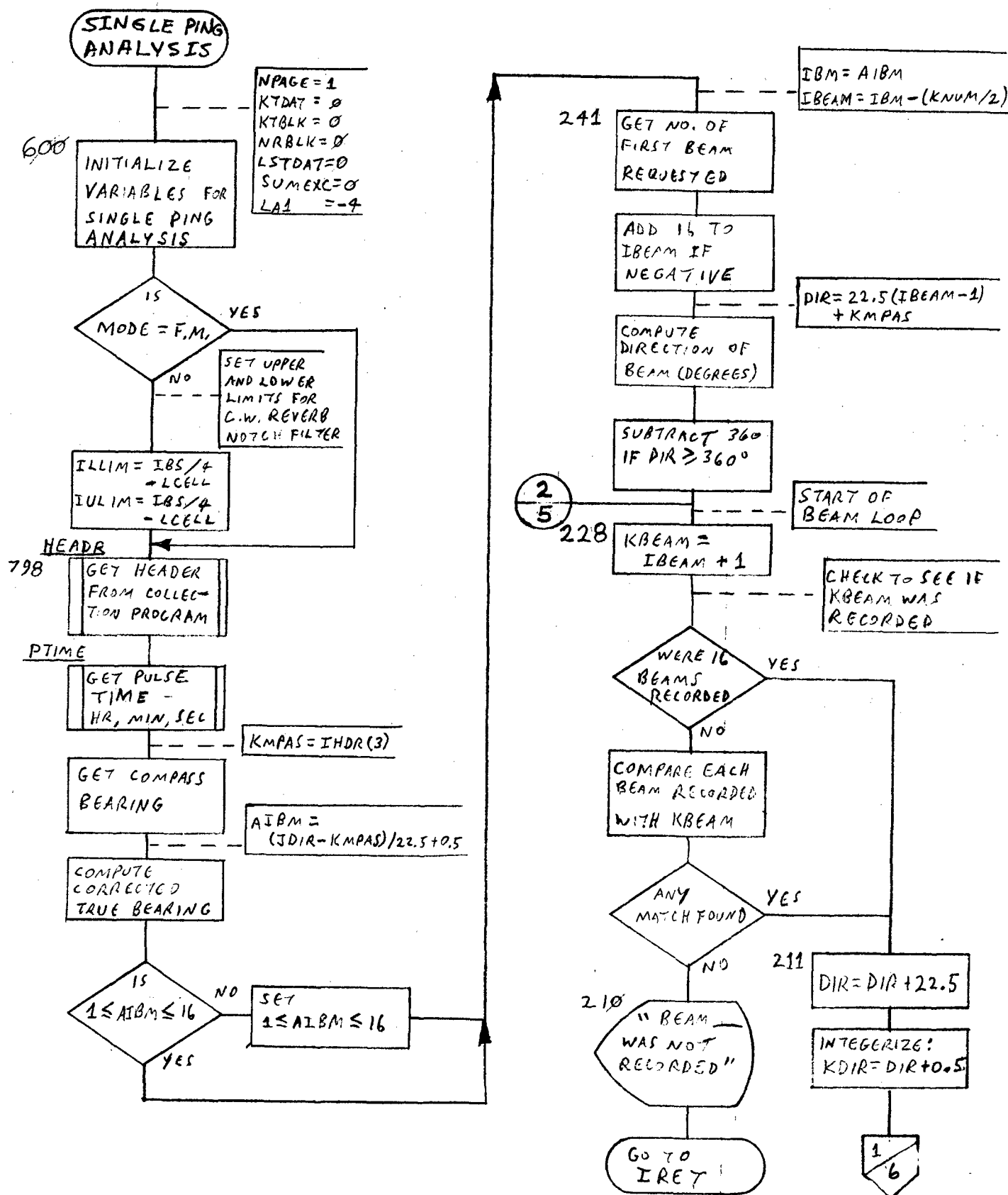
:

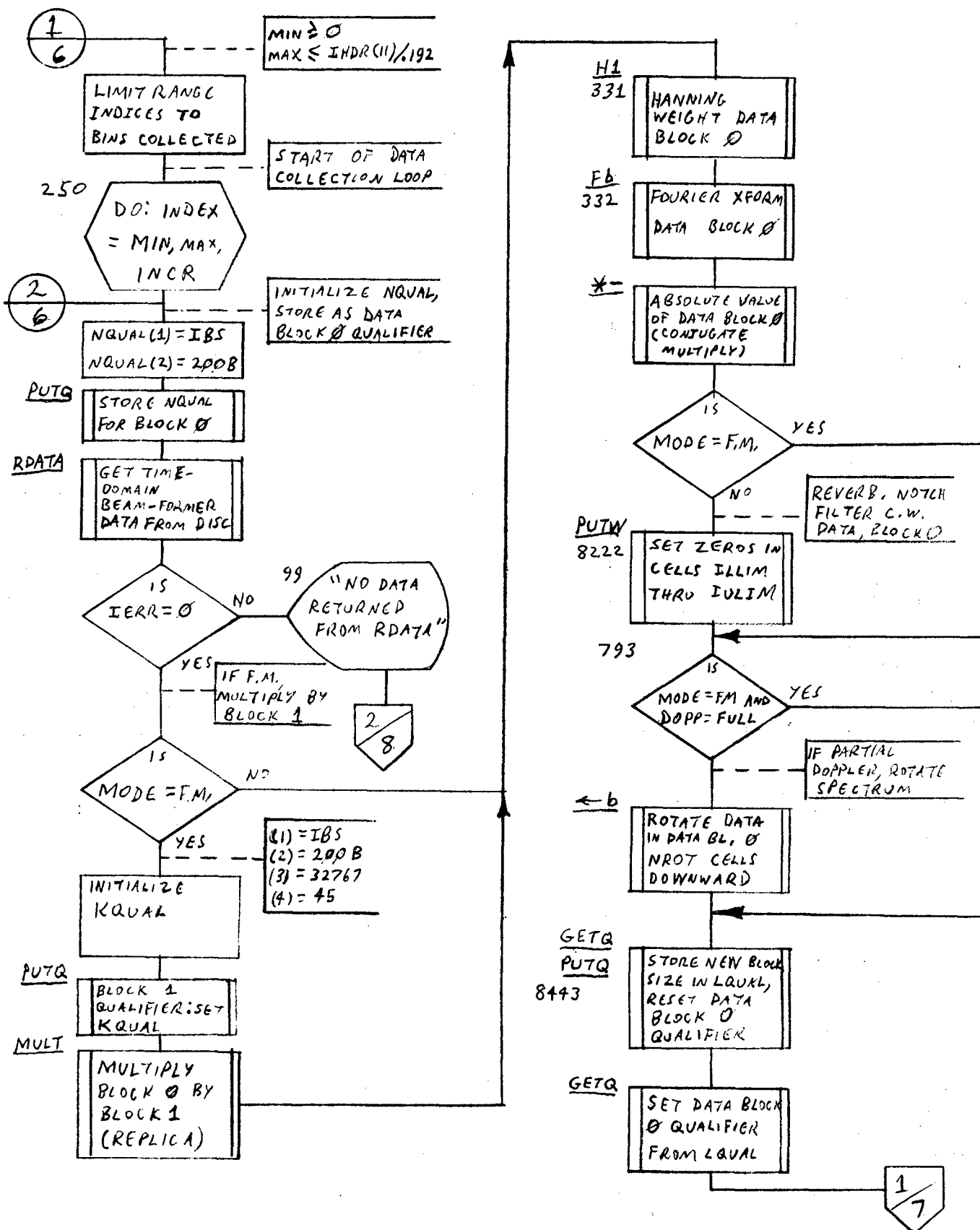


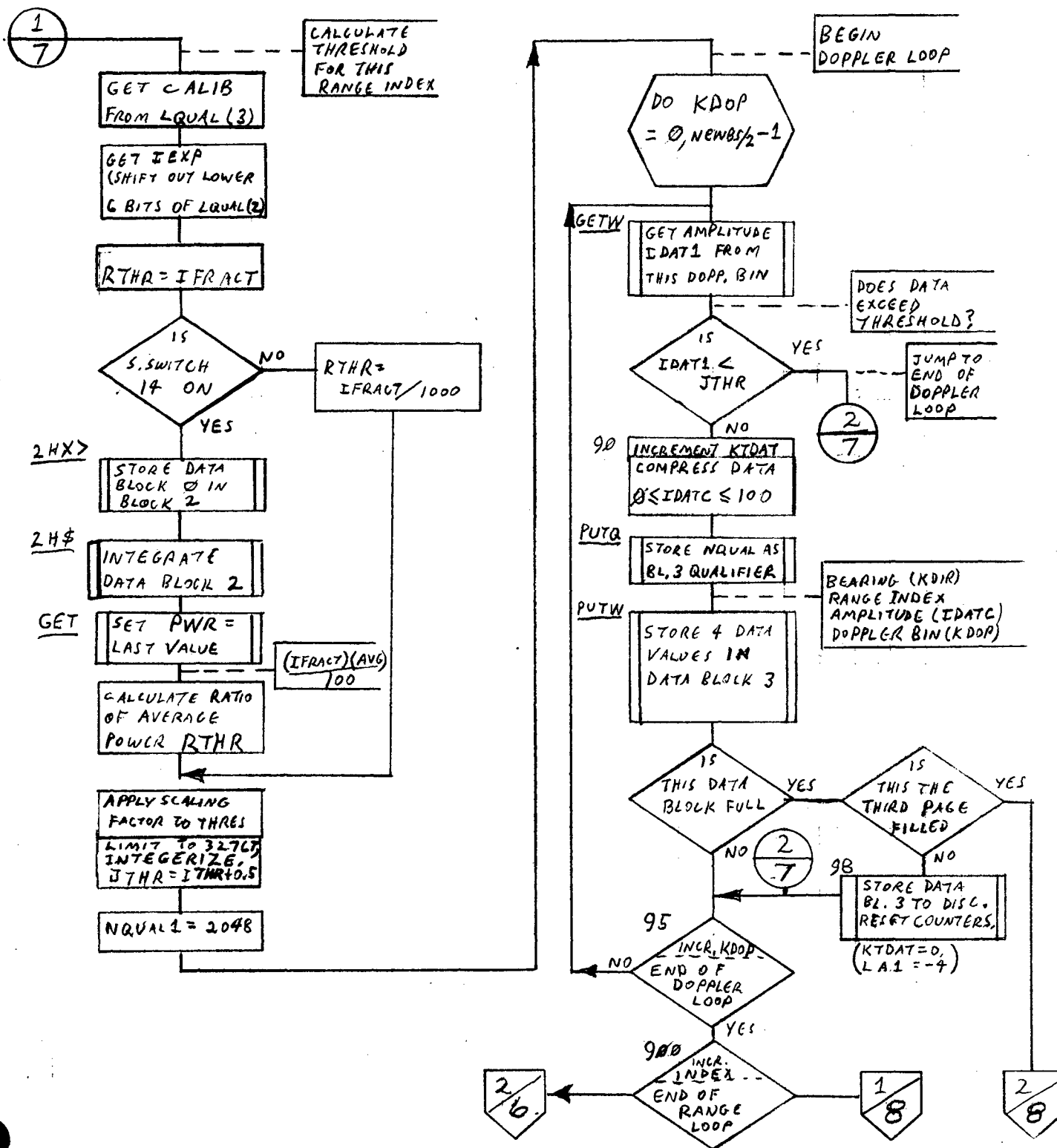






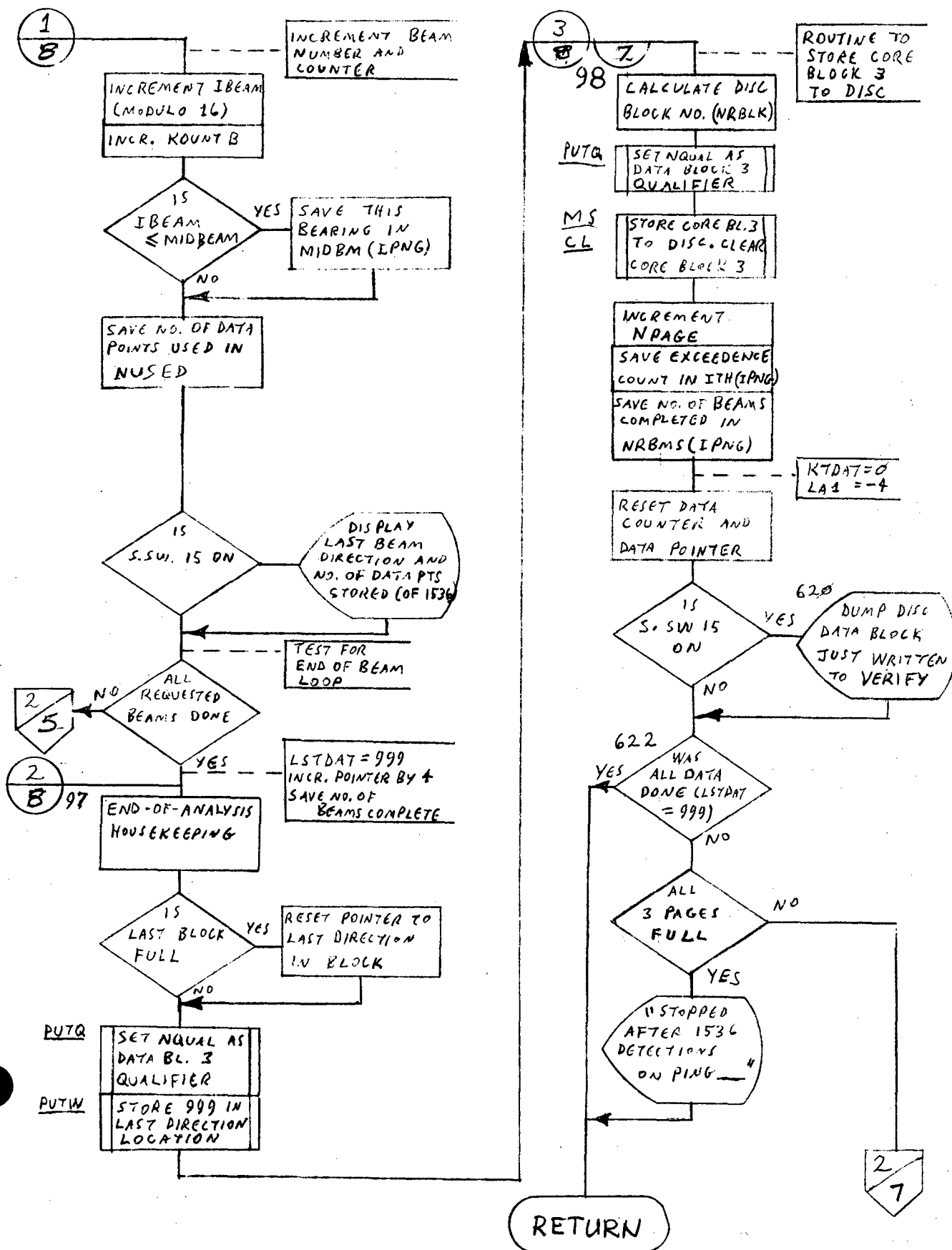






Y1002 - ANALYSIS & SYSTEM EXECUTIVE SUBROUTINE

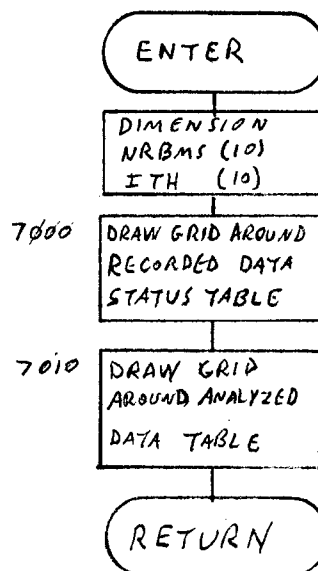
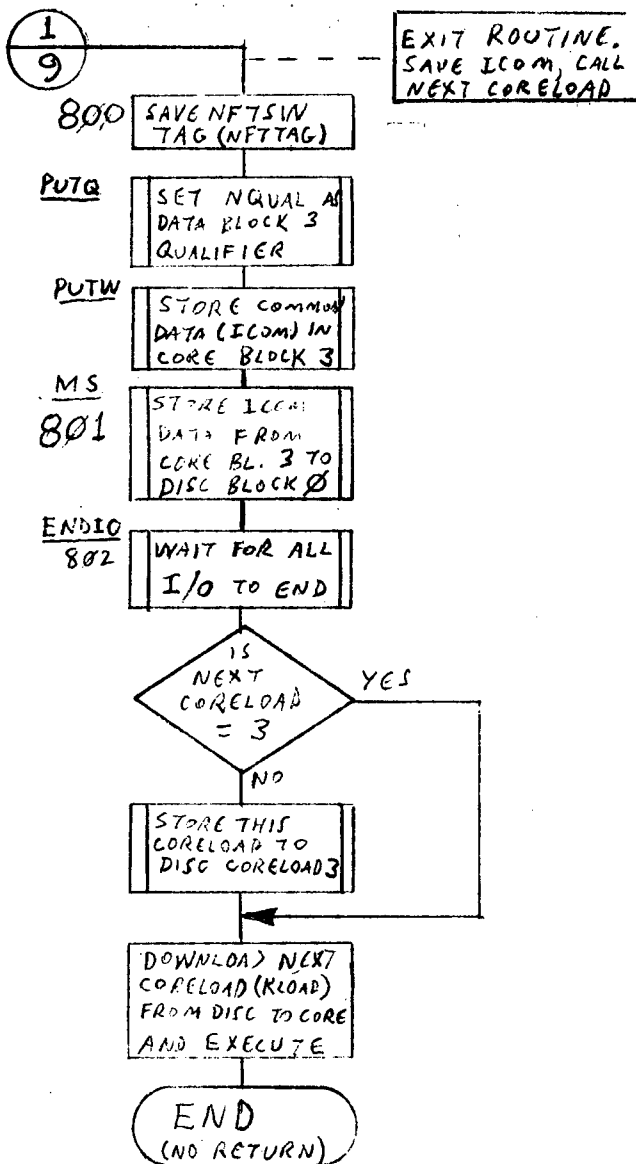
PAGE 8 OF 9



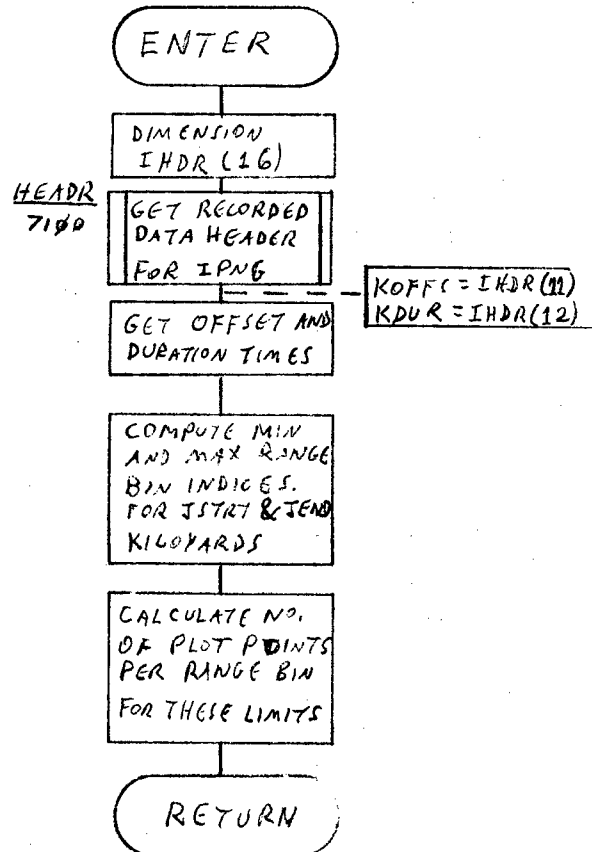
Y100Z - ANALYSIS & SYSTEM EXECUTIVE SUBROUTINE

PAGE 9 OF 9

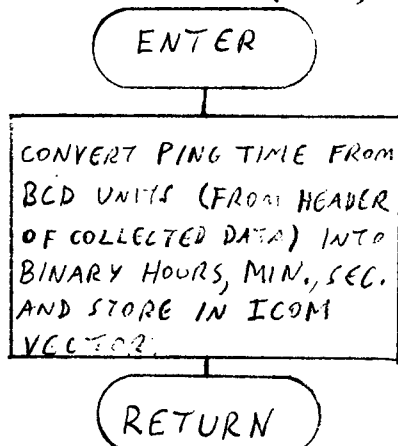
SUBROUTINE STGRD(NRCD, NRBM, ITH)



SUBROUTINE RBINS(IPNG, JSTRT, JEND, KOFFS, MIN, MAX, INCX)



SUBROUTINE PTIME(IPNG, ICOM)



```

0001 FTN4 SUBROUTINE Y1002                                     LATEST MODIFICATION: 2/6/78 (WPB)
0002 C *****
0003 C *****
0004 C *****
0005 C *****
0006 C *****
0007 C *****
0008 C *****
0009 C *****
0010 C *****
0011 C *****
0012 C *****
0013 C *****
0014 C *****
0015 C *****
0016 C *****
0017 C *****
0018 C *****
0019 C *****
0020 C *****
0021 C *****
0022 C *****
0023 C *****
0024 C *****
0025 C *****
0026 C *****
0027 C *****
0028 C *****
0029 C *****
0030 C *****
0031 C *****
0032 C *****

      TITLE: ANALYSIS AND SYSTEM EXECUTIVE SUBROUTINE

      MODULE: Y1002          SOURCE: MP002A          VERSION: X04
              OBJECT: Y10020

      AUTHOR: D. L. GOODMAN    DATE: 1976

      MODIFICATIONS:
        1: CHANGES FOR OLAAS SYSTEM PROBLEMS
           09/30/77 - B. TASHJIAN & W. BOLEN
        2: WRITES MULTI-PING DATA INTO 30 DATA BLOCKS.
           DISC BLOCK 0 IS USED AS COMMON AREA, STORED
           IN ICOM(128) IN FORTRAN.
           11/18/77 - C.R. MOOSE
        3: CHANGE USER-ENTERED QUESTIONS
           11/30/77 - W. BOLEN
        4: CHANGE TO CALCULATE THE ANALYSIS THRESHOLD
           1/24/78 - W. BOLEN

      MODULE FUNCTION:
      Y1002 IS THE EXECUTIVE SUBROUTINE FOR THE OLAAS
      ANALYSIS AND PERFORMS THE FOLLOWING FUNCTIONS:
      1. COLLECTS USER-ENTERED PARAMETERS FOR THE ANALYSIS.
      2. ON-LINE DATA RECORDING AND ANALYSIS.
      3. OFF-LINE DATA ANALYSIS.
      4. SINGLE PING AND BEAMFORMER DATA ANALYSIS.
      5. MULTI-PING ANALYSIS.
```

```

0033 * CALLING SEQUENCE: CALLED BY - Y0003 (OLAAS EXECUTIVE)
0034 *
0035 *
0036 * CALL CLOAD (07B,51B)
0037 *
0038 *****
0039 *****
0040 *****
0041 * COMMON AREA ASSIGNMENTS (ICOM AND/OR DISC DATA BLOCK 0)
0042 *****
0043 *****
0044 *
0045 * JDIU: DIVISOR OF ORIGINAL BLOCK FOR DOPPLER EDITING
0046 * IBND: SPECIFIC DOPPLER ANALYSIS BAND (1-JDIU BANDS)
0047 * NEUBS: BLOCK SIZE OF DOPPLER BAND
0048 * MIN: MINIMUM INDEX USED IN ANALYSIS BY RDATA PGM.
0049 * MAX: MAXIMUM INDEX USED IN ANALYSIS BY RDATA PGM.
0050 * IFRAC: ANALYSIS THRESHOLD PERCENTAGE SET BY USER
0051 * INCR: TIME INCREMENT IN MULTIPLES OF 0.192 SECONDS
0052 * KODE: HANNING EQUALS 1, NO HANNING EQUALS 0
0053 * KTYPE: RANGE, KILOYARDS EQUALS 1, TIME EQUALS 0
0054 * IPNG: PING NUMBER
0055 * KNUM: NUMBER OF BEAMS USED IN ANALYSIS
0056 * KOFFS: TIME OFFSET VALUE
0057 * MODE: 0 - CW PULSE, 1 - FM UP, 2 - FM DOWN
0058 * LGTH: EITHER 2 OR 5 SECONDS
0059 * KT1: TIME, HOURS
0060 * KT2: TIME, MINUTES
0061 * KT3: TIME, SECONDS
0062 * LFRQ: BUOY FREQUENCY, 1 - F1, 2 - F2
0063 * JDIU: DIVISOR OF ORIG. BLOCK (SAME AS WORD 0)
0064 * NTATE: DOPPLER AXIS ROTATION, 1 - YES, 0 - NO
0065 * NSHFT: FOR NTATE=1, NO. OF KNOTS TO SHIFT DOPPLER
0066 * SPARE
0067 * KEY: 0 - ON-LINE PLOT
          1 - ANALYSIS OF PREVIOUSLY RECORDED DATA

```

```

0068      2 - INDIVIDUAL BEAM DISPLAY
0069      3 - MULTI-PING DISPLAY
0070      4 - FOUR-PART DISPLAY
0071      (23-32) ITH1: NUMBER OF WORDS STORED FOR EACH PING
0072      SPARE
0073      (33)
0074      (34) MXPNGS: MAXIMUM NUMBER OF PINGS TO BE COLLECTED
0075      (35) IFST: FIRST PING NO. FOR ANALYSIS
0076      (36) ILST: LAST PING NO. FOR ANALYSIS
0077      (37) IDPLR: DOPPLER BAND (1-FULL, 2-HALF, 3-UP, 4-DOWN)
0078      (38) JSTRT: START RANGE (IN KYDS.)
0079      (39) JEND: END RANGE (IN KYDS.)
0080      (40) JDIR: DIRECTION OF BEARING OF CENTER BEAM OR
0081      (41) IMOD: MODE (ON-LINE / OFF-LINE ANALYSIS)
0082      SPARES
0083      (42-46)
0084      (47) INCRX: LENGTH OF ONE RANGE INCREMENT (IN PLOT POINTS)
0085      (48) KTHR: CURRENT PERCENTAGE FOR PLOT THRESHOLD (1-100)
0086      SPARE
0087      (49-50)
0088      (51-66) IHDR: HEADER FROM INPUT DATA TAPE
0089      SPARES
0090      (67-69)
0091      (70) NFFTAG: DUPLICATE OF NOT-FIRST-TIME SWITCH
0092      (71-80) IAUGEX: NUMBER OF EXCEEDENCES FOR ALL 10 PINGS
0093      (81-90) MIDBM: CENTER BEAM ANALYZED FOR ALL 10 PINGS
0094      (91-100) NRMS: NUMBER OF BEAMS ANALYZED FOR ALL 10 PINGS
0095      (101-109) IDATE: ALPHANUMERIC FIELD FOR DATE (18 CHAR.)
0096      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
0097      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
0098      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
0099      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
0100      * DEFINE ARRAYS, CONSTANTS, VARIABLES, & DATA
0101      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
0102      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
0103      DIMENSION ICOM(128),NQUAL(5),LQUAL(5),IHDR(16),MQUAL(5),
0104      +IOFF(10),IBCD(6),MASK(4),KQUAL(5),KLTRX(8),KLADR(8),ITH(10),
0105      +IAUGEX(10),MIDBM(10),NRMS(10),IDATE(9),IBUFD(9)

```

```

0103 EQUIVALENCE
0104 + (ICOM( 1), IBND), (ICOM( 2), NEWBS), (ICOM( 3), MIN),
0105 + (ICOM( 4), MAX), (ICOM( 5), IFRAC), (ICOM( 6), INCR),
0106 + (ICOM( 7), CODE), (ICOM( 8), KTYPE), (ICOM( 9), IPNG),
0107 + (ICOM( 10), KNUM), (ICOM( 11), KOFFS), (ICOM( 12), MODE),
0108 + (ICOM( 13), LENGH), (ICOM( 14), KT1), (ICOM( 15), KT2),
0109 + (ICOM( 16), KT3), (ICOM( 17), LFRQ), (ICOM( 18), JDIU),
0110 + (ICOM( 19), NTATE), (ICOM( 20), NSHFT),
0111 + (ICOM( 22), KEY), (ICOM( 23), ITH),
0112 + (ICOM( 34), MAXPNS), (ICOM( 35), IFST), (ICOM( 36), ILST),
0113 + (ICOM( 37), IDPLR), (ICOM( 38), JSTRT), (ICOM( 39), JEND),
0114 + (ICOM( 40), JDIR), (ICOM( 41), IMOD),
0115 + (ICOM( 47), INCRX), (ICOM( 48), KTHR),
0116 + (ICOM( 70), NFFTAG), (ICOM( 71), IAUGEX), (ICOM( 51), IHDR),
0117 + (ICOM( 91), NRBMS), (ICOM(101), IDATE),
0118 + (ICOM( 70), NFFTAG), (ICOM( 71), IAUGEX), (ICOM( 81), MIDBM),
0119 + (ICOM( 91), NRBMS), (ICOM(101), IDATE)
0120 DATA NFTSU/1, II/0, IMOD/2, IFST/1, ILST/1, LENGH/2/
0121 DATA MODE/1, IDPLR/1, IBND/2, CODE/1, KTYPE/0, JSTRT/8/
0122 DATA JEND/24, KMIN/10, KMAX/20, INCRT/1, JDIR/0, KNUM/3/
0123 DATA IFRAC/400, LFRQ/2, ISKIP/0, LSLP/0, NTATE/0, KEY/1/
0124 DATA NSHFT/14, MPNG1/1, MPNGKT/1, IDATE/9X2H, IPNG/1/
0125 DATA KLTRK/1B, 7B, 15B, 23B, 31B, 37B, 45B, 53B, IAUGEX/10X0/
0126 DATA KLADR/2B, 51B, 2B, 51B, 2B, 2B, 2B, 51B, MIDBM/10X999/
0127 DATA NQUAL/2048, 100B, 16383, 45, 0, NRBMS/10X0/
0128 DATA NQUAL/2048, 100B, 16383, 45, 0/
0129 DATA ITH/10X0, KTHR/1/
0130
0131 C *****
0132 C X SUBROUTINE ENTRY POINT
0133 C *****
0134
0135 C > > > DETERMINE ENTRY POINTS
0136 GO TO (100, 1020, 1070, 1021, 1000) NFTSU
0137

```

```

0138 C *****
0139 C * CORELOAD 1 ENTRY POINT (NFTSU = 1) *****
0140 C * SUBROUTINE INITIALIZATION *****
0141 C *****
0142 C *****
0143 C > > > DISPLAY OPERATOR'S MESSAGES *****
0144 C 100 WRITE (2,105) *****
0145 C 105 FORMAT (//4X,"***** DATA RECORDING & ANALYSIS *****"
0146 C +//13X,"SYSTEM INITIALIZATION"// "OPERATOR NOTE: SET SENSE"
0147 C + " SWITCH 15 (ON) TO CHANGE THE BUOY"/16X,"FREQUENCY, PULSE"
0148 C + " LENGTH, & CQ NOTCH FILTER.") *****
0149 C *****
0150 C > > > ENTER DESIRED DATE *****
0151 C 110 DO 115 I=1,9 *****
0152 C 115 IBUFD(I) = 2H *****
0153 C 120 WRITE (2,125) IDATE *****
0154 C 125 FORMAT (/"ENTER DATE"/(","9A2,"")) *****
0155 C CALL TPC (IBUFD,IFLG,18) *****
0156 C IF (IFLG) 120,140,130 *****
0157 C 130 DO 135 I=1,9 *****
0158 C 135 IDATE(I) = IBUFD(I) *****
0159 C 140 IMOD = 2 *****
0160 C IF (IMOD .EQ. 3) GO TO 145 *****
0161 C *****
0162 C >> DIAGNOSTICS >> SENSE SU. 3 (ON) GO TO CORELOAD 3 DIRECTLY *****
0163 C IF (ISSU(3) .LT. 0) NFTSU = 3 *****
0164 C IF (ISSU(3) .LT. 0) CALL CLOAD (23B,51B) *****
0165 C *****
0166 C > > > DISABLE MAXIMIZATION ROUTINE *****
0167 C CALL KYBD (2HY ,3010,1) *****
0168 C *****
0169 C > > > INITIALIZE FOURIER DATA BLOCKS *****
0170 C CALL KYBD (2HBS,2048) *****
0171 C CALL KYBD (2HCL,0) *****
0172 C CALL KYBD (2HCL,1) *****

```

```

0173 CALL KYBD (2HCL,2)
0174 CALL KYBD (2HCL,3)
0175
0176 C > > > TRANSFER TO CORELOAD 3 FOR PROCESSING
0177 145 NFTSU = 5
0178 KLOAD = 3
0179 GO TO 800
0180
0181 C ***** LEAVE CORELOAD 1 *****
0182
0183
0184 C *****
0185 C * CORELOAD 3 ENTRY POINT (NFTSU = 5)
0186 C * SUBROUTINE PROCESSING
0187 C *****
0188
0189 C > > > READ IN DISC DATA BLOCK 0 INTO ICOM
0190 1000 CALL PUTQ (3,NQUAL)
0191 CALL KYBD (2HMS,31,0)
0192 CALL KYBD (2HMS,11,3)
0193 DO 1005 J = 1, 128
0194 CALL GETJ (3, J, ICOM(J), II)
0195 1005 CONTINUE
0196
0197 C >> DIAGNOSTICS >> SENSE SJ. 10 (ON) WRITES ICOM AREA
0198 IF (ISSJ(10) .LT. 0) WRITE (1,9999) ICOM
0199 IF (ISSJ(10) .LT. 0) PAUSE 201
0200
0201 C > > > ERASE SCREEN
0202 CALL CHOUT(33B)
0203 CALL CHOUT(14B)
0204 CALL D750
0205
0206 C > > > ASK OPERATING MODE QUESTIONS
0207 1010 CALL ASKQ (IMOD,1,3,19,IDATE)

```



```

0208 IF (IMOD .EQ. 1) GO TO 1015
0209 IF (IMOD .EQ. 2) GO TO 1070
0210 IF (IMOD .EQ. 3) GO TO 110
0211
0212 C *****
0213 C * ON-LINE DATA RECORDING AND ANALYSIS FUNCTION *****
0214 C *****
0215 C *****
0216 C > > > ASK ON-LINE ANALYSIS QUESTIONS *****
0217 1015 IFLAG = 0
0218 WRITE(1,1300)
0219 1300 FORMAT(/"***** ON-LINE DATA RECORDING & ANALYSIS"
0220 + " *****//5X,"** ANSWER ANALYSIS QUESTIONS **")
0221 ASSIGN 1100 TO IRET
0222 GO TO 500
0223
0224 C > > > RE-ENTER FROM QUESTION SUBROUTINE *****
0225 1100 CONTINUE
0226
0227 C > > > INITIALIZE VARIABLES FOR RAW DATA RECORDING *****
0228 IPNG = 1
0229 IFST = 1
0230 DO 1302 I = 1,10
0231 1302 ITH(I) = 0
0232 IOFF(1) = 30
0233
0234 C > > > TRANSFER TO CORELOAD 2 FOR RAW DATA RECORDING *****
0235 NTSW = 2
0236 KLOAD = 2
0237 GO TO 800
0238
0239 C ***** LEAVE CORELOAD 3 *****
0240
0241
0242

```

```

0243 C *****
0244 C * ENTRY POINT ON RETURN FROM CORELOAD 2 (NFTSW = 2) *
0245 C * ON-LINE DATA ANALYSIS & DISPLAY *
0246 C *****
0247
0248 1020 CONTINUE
0249 IF (IPNG .NE. 1) GO TO 1102
0250
0251 C > > > CONVERT START/STOP RANGE
0252 CALL RBINS(IPNG, JSTRT, JEND, KOFFS, MIN, MAX, INCRX)
0253 1102 MQUAL(1) = 2048
0254
0255 C > > > TRANSFER TO SINGLE PING ANALYSIS
0256 ASSIGN 1105 TO IRET
0257 GO TO 600
0258
0259 C > > > RE-ENTER FROM SINGLE PING ANALYSIS SUBROUTINE
0260 1105 CONTINUE
0261 AUGEXC = SUMEXC / ITH(IPNG)
0262 IAUGEX(IPNG) = AUGEXC
0263 IF (IPNG .EQ. 1) WRITE(1, 9998)
0264 9998 FORMAT (" ")
0265
0266 C >> DIAGNOSTICS >> SENSE SW. 10 (ON)
0267 IF (ISSW(10) .LT. 0) WRITE(1, 9105)IPNG, ITH(IPNG), AUGEXC
0268 C9105 FORMAT(" PING NO.", I3, " HAS", I5, " EXCEEDENCES.",
0269 C + " THE AVERAGE VALUE IS", F10.1)
0270
0271 C > > > TRANSFER TO CORELOAD 7 FOR PLOT
0272 KTHR = 1
0273 NFTSW = 4
0274
0275 C > > > RECOVERY FROM BAD RDATA READ ERROR
0276 IF(IERR .NE. 0) NFTSW = 3
0277 IERR = 0

```

```
0278 C /~~~~~  
0279 KLOAD = 7  
0280 GO TO 800  
0281  
0282 C ***** LEAVE CORELOAD 3 *****  
0283  
0284  
0285 C *****  
0286 C * ENTRY POINT ON RETURN FROM CORELOAD 4 (NFTSU - 4) *  
0287 C * ON-LINE MULTI-PING RECORDING *  
0288 C *****  
0289  
0290  
0291 1021 IPNG = IPNG + 1  
0292  
0293 C > > > CHECK TO SEE IF LAST PING  
0294 CALL HEADR(1,IHDR)  
0295  
0296 C >> DIAGONSTICS >> SENSE SU. 10 (ON)  
0297 C IF(ISSU(10).LT. 0) WRITE(1, 9102) (ICOM(J), J = 51, 66)  
0298 C9102 FORMAT(' HDR ', 11I6)  
0299  
0300 IF(IPNG.LE. IHDR(10)) GO TO 1022  
0301 IPNG = 1  
0302 GO TO 1070  
0303  
0304 C > > > TRANSFER TO CORELOAD 4 TO RECORD THE NEXT PING  
0305 1022 NFTSU = 2  
0306 KLOAD = 4  
0307 GO TO 802  
0308  
0309 C ***** LEAVE CORELOAD 3 *****  
0310  
0311  
0312
```

```

0313 C *****
0314 C * ENTRY POINT TO DISPLAY THE DATA STATUS TABLES *
0315 C * (1) FOR OFF-LINE ANALYSIS & PLOT DISPLAY *
0316 C * (2) ON RETURN FROM CORELOAD 7 (NFTSW = 3) *
0317 C * OFF-LINE ANALYSIS & PLOT DISPLAY *
0318 C * (3) TO DISPLAY ON-LINE DATA STATUS (NFTSW = 5) *
0319 C *****
0320
0321 C > > > OLAAS DATA PARAMETER STATUS TABLES
0322 1070 NFTSW = 5
0323 CALL HEADR(1,IHDR)
0324 IEND = IHDR(11)+IHDR(12)
0325
0326 C > > > ERASE SCREEN
0327 CALL CHOUT(33B)
0328 CALL CHOUT(14B)
0329 CALL D750
0330
0331 C > > > RECORDED DATA STATUS TABLE GENERATION
0332 WRITE (2,1071) IDATE
0333 1071 FORMAT ('XXXXXXXXXX OLAAS PROCESSING & DISPLAY '
0334 + 'SYSTEM XXXXXXXXXXXX'/21X,9A2/)
0335
0336 WRITE (2,1072) IHDR(10),IHDR(11),IEND,IHDR(8)
0337 1072 FORMAT (16X,"RECORDED DATA STATUS TABLE"/7X,
0338 + "NUMBER OF PINGS RECORDED"//I3,/7X,
0339 + "PING RECORD WINDOW"//I3," TO",I3," SECONDS"/7X,
0340 + "NUMBER OF BEAMS RECORDED"//I3)
0341
0342 WRITE(1,1073)
0343 1073 FORMAT(7X,"PING",4X,"COMPASS",3X,"FIRST",3X,"HEADING",5X,
0344 + "TIME",/6X,"NUMBER",2X,"(DEGREES)",2X,"BEAM",3X,"(DEGREES)")
0345
0346 DO 1076 I=1,IHDR(10)
0347 CALL HEADR(I,IHDR)

```

```

0348 C > > > COMPUTE THE TIME OF PING
0349 CALL PTIME(I, ICOM)
0350
0351 1076 WRITE(1,1077) I,IHDR(3),IHDR(5),IHDR(6),KT1,KT2,KT3
0352 1077 FORMAT (8X,I2,6X,I4,6X,I2,6X,I4,5X,I2,":",I2,":",I2,":",I2)
0353
0354 C > > > ANALYZER DATA STATUS TABLE GENERATION
0355 1778 WRITE (2,1078)
0356 1078 FORMAT (/17X,"ANALYZED DATA STATUS TABLE"/
0357 +2X,"PING",3X,"CENTER BEAM",2X,"NUMBER BEAMS",3X,"NUMBER"
0358 + " OF",4X,"AVERAGE"/1X,"NUMBER",5X,"ANGLE",6X,"PROCESSED",
0359 +4X,"EXCEEDENCES",2X,"EXCEEDENCE")
0360
0361 DO 1080 I=1,10
0362 IF (NRBMS(I)).EQ. 0 .AND. ITH(I).EQ. 0) GO TO 1080
0363 WRITE (2,1079) I,MIDBM(I),NRBMS(I),ITH(I),
0364 +IAXGEX(I)
0365 1079 FORMAT (3X, I2, 8X, I3, 11X, I2, 10X, I4, 9X, I3)
0366 1080 CONTINUE
0367
0368 C > > > DRAW GRID AROUND THESE TABLES
0369 CALL STGRD (IHDR(10),NRBMS,ITH)
0370
0371 C > > > POSITION TO END OF PAGE
0372 1776 CALL TPLOT (0,0,0)
0373 CALL CHOUT (37B)
0374 WRITE (2,1777)
0375 1777 FORMAT (" ")
0376 IF (IMOD .EQ. 1) GO TO 1010
0377 GO TO 1210
0378
0379
0380
0381
0382

```

```

0383 C *****
0384 C * OFF-LINE DATA ANALYSIS & PLOT DISPLAY FUNCTION *
0385 C *****
0386
0387 C > > > CLEAR SCREEN AND DISPLAY ERROR MESSAGE
0388 1200 CALL CHOUT (33B)
0389 CALL CHOUT (14B)
0390 CALL D750
0391 WRITE (2,1205)
0392 1205 FORMAT (/'##### NO ANALYZED DATA AVAILABLE '
0393 + ' #####'/)
0394 KEY = 1
0395
0396 C > > > ASK OFF-LINE MODE QUESTIONS
0397 1210 CALL ASKQ (KEY,1,4,20,IDATE)
0398 NFTSU = 3
0399 IF (KEY.EQ. 1) GO TO 1030
0400 IF (ITH(IPNG).EQ. 0) GO TO 1200
0401
0402 C > > > TRANSFER TO THE SELECTED OFF-LINE MODE
0403 GO TO (1030,1040,1050,1060,1040) KEY
0404
0405 C *****
0406 C * ANALYSIS OF PREVIOUSLY RECORDED DATA (KEY = 1) *
0407 C *****
0408
0409 C > > > CHECK FOR HIGHEST PING REQUESTED FOR RECORDING
0410 1030 IFLAG = 1
0411 CALL HEADR (1,IHDR)
0412
0413 C >> DIAGNOSTICS >> SENSE SU. 3 (ON)
0414 C IF(ISSU(10).LT. 0) WRITE(1, 9102) (ICOM(J), J = 51, 66)
0415
0416 NHI = IHDR(10)
0417

```

```

0418 C > > > ASK OFF-LINE ANALYSIS QUESTIONS
0419 WRITE (2,1031)
0420 1031 FORMAT ('5X,'** ANSWER ANALYSIS QUESTIONS **'/)
0421 ASSIGN 1110 TO IRET
0422 GO TO 500
0423
0424 C > > > RE-ENTER FROM QUESTION ROUTINE
0425 1110 CONTINUE
0426
0427 C > > > CONVERT START/STOP RANGE
0428 CALL RBINS(IFST, JSTRT, JEND, KOFFS, MIN, MAX, INCRX)
0429
0430 C > > > CLEAR DATA STATUS VARIABLES
0431 DO 1303 I = IFST, ILST
0432   IAUGEX(I) = 0
0433   MIDBM(I) = 999
0434   NRBMS(I) = 0
0435   ITH(I) = 0
0436 1303
0437 C > > > OFF-LINE ANALYSIS LOOP
0438 DO 1038 IPNG=IFST, ILST
0439
0440 C > > > TRANSFER TO SINGLE PING ANALYSIS
0441 ASSIGN 1115 TO IRET
0442 GO TO 600
0443
0444 C > > > RE-ENTER FROM SINGLE PING ANALYSIS ROUTINE
0445 1115 CONTINUE
0446   AUGEXC = SUMEXC / ITH(IPNG)
0447   IAUGEX(IPNG) = AUGEXC
0448
0449 C > > > SAVE NUMBER OF EXCEEDANCES PER PING
0450   IDAT = ITH(IPNG)
0451 1038 CONTINUE
0452 C ***** END OFF-LINE ANALYSIS LOOP *****

```

```

0453 IPNG = IPNG - 1
0454 KTHR = 1
0455 GO TO 1090
0456
0457 C *****
0458 C * INDIVIDUAL BEAM DISPLAY (KEY = 2)
0459 C *****
0460
0461 C > > > ASK INDIVIDUAL BEAM DISPLAY QUESTIONS
0462 1040 IFLAG = 4
0463 WRITE (2,1041)
0464 1041 FORMAT (/5X,"XX ANSWER INDIVIDUAL BEAM DISPLAY "
0465 + "QUESTIONS XX"/)
0466 ASSIGN 1090 TO IRET
0467 GO TO 500
0468
0469 C *****
0470 C * MULTI-PING DISPLAY (KEY = 3)
0471 C *****
0472
0473 C > > > ASK MULTI-PING DISPLAY QUESTIONS
0474 1050 IFLAG = 2
0475 WRITE (2,1051)
0476 1051 FORMAT (/5X,"XX ANSWER MULTI-PING DISPLAY QUESTIONS XX"/)
0477 ASSIGN 1090 TO IRET
0478 GO TO 500
0479
0480 C *****
0481 C * FOUR PART DISPLAY (KEY = 4)
0482 C *****
0483
0484 C > > > ASK FOUR PART DISPLAY QUESTIONS
0485 1060 IFLAG = 3
0486 WRITE (2,1061)
0487 1061 FORMAT (/5X,"XX ANSWER FOUR PART DISPLAY QUESTIONS XX"/)

```



```

0488          ASSIGN 1090 TO IRET
0489          GO TO 500
0490
0491          C *****
0492          C * TRANSFER TO CORELOAD 7 FOR PLOT DISPLAY
0493          C *****
0494
0495          C > > > RE-ENTER FROM QUESTION ROUTINE
0496          C 1090 CONTINUE
0497
0498          C > > > CONVERT START/STOP RANGE, GET TIME OF IPNG
0499          C CALL RBINS(IPNG, JSTRT, JEND, KOFFS, MIN, MAX, INCRX)
0500          C CALL PTIME(IPNG, ICOM)
0501
0502          C > > > TRANSFER TO CORELOAD 7 FOR PLOT DISPLAYS
0503          C KLOAD = 7
0504          C GO TO 800
0505
0506          C ***** LEAVE CORELOAD 3 *****
0507
0508
0509          C *****
0510          C *
0511          C * QUESTION ROUTINE: COLLECTS USER-ENTERED PARAMETES FOR USE
0512          C * IN THE OLAS DATA ANALYSIS.
0513          C *
0514          C * NOTE: IFLAG VALUES: 0 - ON-LINE FUNCTION
0515          C * 1 - OFF-LINE ANALYSIS
0516          C * 2 - MULTI-PING DISPLAY
0517          C * 3 - FOUR PART DISPLAY
0518          C * 4 - INDIVIDUAL BEAN DISPLAY
0519          C *
0520          C *****
0521
0522          C > > > ENTER QUESTION ROUTINE & DETERMINE STARTING QUESTION

```

```

0523 500 NETTAG = NETSW
0524 NETSW = 5
0525 IQUEIN = IFLAG + 1
0526 GO TO (515, 501, 501, 503, 503) IQUEIN
0527
0528 C > > > FIRST PING
0529 501 CALL ASKQ (IFST,1,NHI,1)
0530
0531 C > > > LAST PING
0532 CALL ASKQ (ILST,IFST,NHI,2)
0533 IF (IFLAG .EQ. 1) GO TO 515
0534 IF (IFLAG .EQ. 2) GO TO 525
0535
0536 C > > > DISPLAY PING NUMBER (INDIVIDUAL BEAM OR FOUR PART)
0537 503 CALL ASKQ (IPNG,1,10,14)
0538 IF (IFLAG .EQ. 3) GO TO 510
0539
0540 C > > > DISPLAY TYPE
0541 505 CALL ASKQ (KEY,2,5,15)
0542 IF (KEY .EQ. 2 .OR. KEY .EQ. 5) GO TO 525
0543 WRITE (2,507)
0544 507 FORMAT ("XXXXXXXXX PARAMETER ERROR XXXXXXXXX",
0545 +5X,"RE-ENTER VALUE")
0546 GO TO 505
0547
0548 C > > > DISPLAY BEARING (FOUR PART DISPLAY)
0549 510 CALL ASKQ (JDIR,0,359,16)
0550 IF (IFLAG .EQ. 3) GO TO 525
0551
0552 C > > > BUOY FREQUENCY
0553 515 LFRQ = 1
0554 IF (ISSW(15) .LT. 0) CALL ASKQ (LFRQ,1,2,3)
0555
0556 C > > > PULSE LENGTH
0557 LENGTH = 2

```

```

0558 IBS = 1024
0559 IF (ISSW(15) .LT. 0) CALL ASKQ (LNGTH,2,5,4)
0560 IF (LNGTH .EQ. 5) IBS=2048
0561 NQUAL(1)=IBS
0562 NQUAL(2)=1008
0563 NQUAL(3)=16383
0564 NQUAL(4) = 45
0565 NQUAL(5) = 0
0566 CALL PUTQ(0,NQUAL)
0567
0568 C > > > SIGNAL MODE (CW, FM-UP, OR FM-DOWN)
0569 CALL ASKQ (MODE,0,2,5)
0570 IF (MODE .GE. 1) GO TO 520
0571
0572 C > > > CW DOPPLER BAND
0573 CALL ASKQ (IDPLR,1,4,6)
0574
0575 C > > > CW NOTCH FILTER SETTING
0576 LCELL = ((10.0 * IBS)/666.67) + 0.5
0577 IF (ISSW(15) .LT. 0) CALL ASKQ (LCELL,0,25,17)
0578 GO TO 525
0579
0580 C > > > FM DOPPLER BAND
0581 520 CALL ASKQ (IDPLR,1,2,7)
0582
0583 C > > > START RANGE
0584 525 CALL ASKQ (JSTRT,2,35,8)
0585
0586 C > > > STOP RANGE
0587 CALL ASKQ (JEND,JSTRT,40,9)
0588 IF (IFLAG .EQ. 2 .OR. IFLAG .EQ. 4) GO TO 530
0589 IF (IFLAG .EQ. 3) GO TO 538
0590
0591 C > > > RANGE RESOLUTION
0592 CALL ASKQ (INCRT,1,40,10)

```

```

0593 C > > > BEARING OF CENTER BEAM
0594 530 CALL ASKQ (JDIR,0,359,11)
0595 IF (IFLAG .EQ. 2) GO TO 538
0596
0597 C > > > NUMBER OF BEAMS ANALYZED
0598 CALL ASKQ (KNUM,1,16,12)
0599 IF (IFLAG .EQ. 4) GO TO 538
0600
0601 C > > > ANALYSIS THRESHOLD
0602 535 CALL ASKQ (IFRAC,0,10000,13)
0603 IF (IFLAG .LT. 3) GO TO 540
0604
0605 C > > > PLOT THRESHOLD
0606 538 CALL ASKQ (KTHR,1,100,18)
0607 IF (IFLAG .GE. 2) GO TO 595
0608
0609 C >>>>> SET-UP FOR CW & FM DOPPLER >>>>>>
0610 540 IF (MODE .EQ. 0) GO TO 575
0611
0612 C > > > CALCULATE THE FM REPLICA
0613 LSUP = 0
0614 IF (MODE .EQ. 2) LSUP = 1
0615 BJ = (IBS * 100.0)/(666.667 * LENGTH)
0616 IF (LFRQ .EQ. 1) GO TO 542
0617 IF (LSUP .EQ. 0) GO TO 545
0618 KSUP = 0
0619 GO TO 550
0620 545 KSUP = 1
0621 GO TO 550
0622 542 KSUP = LSUP
0623
0624 C > > > SET SCALE FACTORS FOR FM REPLICA
0625 550 KQUAL(1) = IBS
0626 KQUAL(2) = 100B
0627

```

```

0628 KQUAL(3) = 32767
0629 KQUAL(4) = 45
0630 CALL PUTQ(1,KQUAL)
0631 IF (KSWP) 560,560,555
0632 FS = 83.0 + (BW/2.0)
0633 SS = -100.0/LNGTH
0634 GO TO 565
0635 FS = 83.0 - (BW/2.0)
0636 SS = 100.0/LNGTH
0637 DT=1.0/666.66667
0638 DO 570 I = 0,(IBS-1)
0639 REPLI=cos(2.*3.14159*((FS*IXDT) + ((SS/2.0)*(IXDT)*X2))))
0640 J = REPLI * 32767.0
0641 CALL PUTW (1,I,J,0)
0642 CONTINUE
0643 CALL PUTQ (0,NQUAL)
0644
0645 C > > > DOPPLER EDITING FOR FM
0646 JDIU = 2
0647 NROT = 64
0648 IF (IDPLR.EQ. 2) JDIU = 4
0649 GO TO 580
0650
0651 C > > > DOPPLER EDITING FOR CW
0652 JDIU = 4
0653 IF (IDPLR.EQ. 1) JDIU = 2
0654 NROT = 128
0655 IF (IDPLR.EQ. 2) NROT=192
0656 IF (IDPLR.EQ. 3 .AND. LFRQ.EQ. 1) NROT=256
0657 IF (IDPLR.EQ. 4 .AND. LFRQ.EQ. 2) NROT=256
0658 IBLK = 0
0659 IF (LNGTH.EQ. 5) NROT = NROT*X2
0660 NEWBS = IBS/JDIU
0661
0662 C > > > SET-UP TO RETURN

```

```

0663 595 SUMEXC = 0.0
0664 NFTSW = NFTTAG
0665
0666 C > > > RETURN TO CALLER
0667 GO TO IRET
0668
0669 C ***** END QUESTION ROUTINE *****
0670
0671
0672 C *****
0673 C *
0674 C * ANALYSIS ROUTINE: ANALYZES BEAMFORMER DATA AS SPECIFIED *
0675 C * BY THE OLAAS DATA ANALYSIS PARAMETERS, *
0676 C * FOR A SINGLE PING. *
0677 C *
0678 C *****
0679
0680 C > > > ENTER ANALYSIS ROUTINE FOR SINGLE PING ANALYSIS
0681 600 CONTINUE
0682 NPAGE = 1
0683 KTDAT = 0
0684 KOUNTB = 0
0685 NRBLK = 0
0686 LSTDAT = 0
0687 LA1 = -4
0688 SUMEXC = 0.0
0689
0690 C > > > TEST FOR FM MODE
0691 IF (MODE .GE. 1) GO TO 798
0692
0693 C > > > COMPUTE PARAMETERS FOR THE CU REVERB NOTCH FILTER
0694 ILLIM = IBS / 4 - LCELL
0695 IULIM = IBS / 4 + LCELL
0696 II = 0
0697 798 CONTINUE

```

```

0698      CALL HEADR(IPNG,IHDR)
0699
0700      C > > > COMPUTE THE TIME
0701      CALL PTIME(IPNG, ICOM)
0702      KMPAS=IHDR(3)
0703
0704      C > > > CORRECTED DIRECTION MTRUE
0705      AIBM=((JDIR-KMPAS)/22.5)+.5
0706      IF(AIBM .GT. 16.)AIBM=AIBM-16.
0707      IF(AIBM.LT.1.)AIBM=AIBM+16.
0708      IBM=AIBM
0709      241 IBEAM = IBM - (KNUM/2)
0710      IF(IBEAM .LT. 0)IBEAM = IBEAM + 16
0711      KOUNT=1
0712
0713      C > > > COMPUTE DIRECTION OF BEAM
0714      DIR=(IBEAM * 22.5) + KMPAS -22.5
0715      IF(DIR .GE. 360.0)DIR=DIR-360.0
0716
0717      C *****
0718      C * START OF BEAM LOOP
0719      C *****
0720
0721      228 CONTINUE
0722      KBEAM=IBEAM+1
0723      IF(KBEAM .GT. 16)KBEAM = KBEAM - 16
0724
0725      C > > > CHECK IF BEAM WAS RECORDED
0726      IF(IHDR(8) .EQ. 16) GO TO 211
0727      DO 210 I = IHDR(5), (IHDR(8) + IHDR(5) - 1)
0728      KQ = I
0729      IF(KQ .GT. 16) KQ = KQ - 16
0730      IF(KBEAM .EQ. KQ ) GO TO 211
0731      210 CONTINUE
0732      WRITE(1, 207) KBEAM

```

```

0733 207 FORMAT(" BEAM ",I2," WAS NOT RECORDED")
0734
0735 C > > > RETURN TO CALLER
0736 C GO TO IRET
0737
0738 C ***** EXIT ANALYSIS ROUTINE *****
0739
0740 211 CONTINUE
0741 DIR=DIR+22.5
0742 KDIR=DIR+.5
0743 IF(KDIR .GE. 360)KDIR - KDIR - 360
0744 MIN = MAX0(MIN, 0)
0745 MAX = MIN0(MAX, IFIX(IHDR(12) / 0.192) )
0746
0747 C *****
0748 C * BEGIN DATA COLLECTION LOOP
0749 C *****
0750
0751 250 DO 900 INDEX=MIN,MAX,INCRT
0752
0753 C >> DIAGNOSTICS >> SENSE SU. 7 (ON)
0754 C IF(ISSU(7) .LT. 0) GO TO 97
0755
0756 C > > > SET SCALE FACTORS (10XX1,LINEAR,RECT,TIME)
0757 C NQUAL(1)=IBS
0758 C CALL PUTQ(0,NQUAL)
0759
0760 C > > > OBTAIN TIME-DOMAIN BEAMFORMER DATA
0761
0762 C >> DIAGNOSTICS >> SENSE SU. 9 (ON)
0763 C IF(ISSU(9) .LT. 0) WRITE(1, 991)IPNG,KBEAM,INDEX,IBS,IERR
0764 C
0765 C CALL RDATA(IPNG,KBEAM,INDEX,IBS,IERR)
0766
0767 C > > > COMPUTE VARIABLE THRESHOLD HERE

```



```

0768 330 IF(IERR .NE. 0) GO TO 99
0769
0770 C > > > IF F.M., MULTIPLY BY BLOCK 1
0771 CALL ENDIO
0772
0773 C >> DIAGNOSTICS >> SENSE SW. 8 (ON)
0774 C IF(ISSW(8) .LT. 0) WRITE(1, 991)KDIR,IPNG,KBEAM,INDEX,
0775 C +IBS,IERR
0776
0777 C > > > TEST FOR CW MODE
0778 C IF (MODE .EQ. 0) GO TO 331
0779
0780 C > > > SET DATA BLOCK 1 QUALIFIERS FOR FM
0781 KQUAL(1) = IBS
0782 KQUAL(2) = 100B
0783 KQUAL(3) = 32767
0784 KQUAL(4) = 45
0785 CALL PUTQ(1,KQUAL)
0786
0787 C > > > MULTIPLY DATA BLOCK 1 BY 0
0788 CALL KYBD(025040B,1)
0789
0790 C > > > HANN DATA BLOCK 0
0791 331 CALL KYBD(2H1,0)
0792
0793 C > > > TAKE FOURIER TRANSFORM OF DATA BLOCK 0
0794 332 CALL KYBD(2HF,0)
0795
0796 C > > > CONJUGATE MULTIPLY OF DATA BLOCK 0
0797 CALL KYBD(2HX-,0)
0798 CALL ENDIO
0799
0800 C > > > SKIP REVERBERATION FILTERING IF FM
0801 CALL ENDIO
0802

```

```

0803 C >> DIAGNOSTICS >> SENSE SW. 2 (ON)
0804 C IF(ISSW(2) .LT. 0) WRITE(1, 9793)IBS,LCELL,ILLIM,IULIM,NROT
0805 C9793 FORMAT("IBS,LCELL,ILLIM,IULIM,NROT",5X, 5I6)
0806
0807 C > > > TEST FOR FM MODES
0808 IF (MODE .GE. 1) GO TO 793
0809 DO 8222 L = ILLIM,IULIM
0810 CALL PUTW(0,L,0,II)
0811 793 CONTINUE
0812
0813 C > > > ROTATE SPECTRUM NROT CELLS TO THE LEFT
0814 IF (MODE .GE. 1 .AND. IDPLR .EQ. 1) GO TO 8443
0815
0816 C > > > SHIFT DATA BLOCK 0 FOR FM HALF DOPPLER
0817 CALL KYBD(057440B,0,NROT)
0818 CALL ENDIO
0819 8443 CALL GETQ(0, LQUAL)
0820 LQUAL(1)=NEWBS
0821 CALL PUTQ(0, LQUAL)
0822
0823 C > > > OBTAIN ABSOLUTE MAGNITUDE
0824 CALL ENDIO
0825
0826 C >> DIAGNOSTICS >> SENSE SW. 5 (ON)
0827 C IF(ISSW(5) .LT. 0) WRITE(1,991)KBEAM,INDEX,LQUAL
0828
0829 C > > > SEARCH CELLS IN BLOCK FOR THRESHOLD EXCEEDENCES
0830 CALL GETQ(0,LQUAL)
0831
0832 C >> DIAGNOSTICS >> SENSE SW. 10 (NOT ACTIVE)
0833 C IF(ISSW(10) .LT. 0) WRITE(1, 991)LQUAL
0834
0835 CALIB = 1.0 * LQUAL(3)
0836 IEXP = ISHFT(LQUAL(2),6)
0837

```

```

0838 C > > > CALCULATE THE ANALYSIS THRESHOLD
0839 RTHR = IFRAC
0840 IF (ISSU(14) .GE. 0) GO TO 9100
0841
0842 C > > > STORE FOURIER DATA BLOCK 0 IN NTEMP
0843 NTEMP = (4096 / NEUBS) + 1
0844 CALL KYBD (2HX,NTEMP)
0845
0846 C > > > INTEGRATE FOURIER DATA BLOCK 2
0847 CALL KYBD (2H$ ,NTEMP,1,((NEUBS/2)-1))
0848
0849 C > > > GET THE AVERAGE THRESHOLD
0850 CALL GET (NTEMP,((NEUBS/2)-1),PUR,DAT1)
0851 RTHR = (IFRAC * PUR / ((NEUBS/2)-2)) / 100.0
0852
0853 C >> DIAGNOSTICS >> SENSE SU. 6 (ON)
0854 NEWBLO = NEUBS/2
0855 IF (ISSU(6) .LT. 0) WRITE (2,9101) IFRAC,NEUBLO,PUR,RTHR
0856 C9101 FORMAT ('IFRAC',5X,'NEUBS/2',5X,'PUR',10X,'RTHR',
0857 C +/-,15,5X,15,5X,F8.6,5X,F8.6)
0858
0859 9100 IF (ISSU(14) .GE. 0) RTHR = RTHR * 1E-4
0860 THRES = (RTHR * 32767.0XX2)/(CALIB * 10.0XXIEXP)
0861
0862 C >> DIAGNOSTICS >> SENSE SU. 6 (NOT ACTIVE)
0863 C IF (ISSU(6) .LT. 0) WRITE(1, 9087)THRES,CALIB,RTHR,IEXP
0864 C9087 FORMAT(1X,3F15.5, 119)
0865
0866 IF(THRES .GT. 32767.) THRES = 32767.
0867 JTHR = THRES + 0.5
0868 NQUAL(1) = 2048
0869
0870 C *****
0871 C * BEGIN DOPPLER LOOP
0872 C *****

```

```

0873      DO 95 KDOP = 0, ((NEWBS/2)-1)
0874      CALL GETW(0, KDOP, IDAT1, IDAT2)
0875      FORMAT(2X, 10I6)
0876
0877
0878      C > > > TEST FOR THRESHOLD EXCEEDENCE
0879      IF(IDAT1 .LT. JTHR ) GO TO 95
0880      KTDAT = KTDAT + 1
0881      IDATC = IFIX(100.0 * (IDAT1 - JTHR) / IDAT1 + 0.5)
0882      SUMEXC = SUMEXC + 1.0 * IDATC
0883
0884      C > > > GET LOCATIONS IN BLOCK 3, STORE 3 DATA POINT VALUES
0885      LA1 = 4 * (KTDAT-1)
0886      CALL PUTW(3, NQUAL)
0887      CALL PUTW(3, LA1, KDIR, II)
0888      CALL PUTW(3, LA1+1, INDEX, II)
0889      CALL PUTW(3, LA1+2, IDATC, II)
0890      CALL PUTW(3, LA1+3, KDOP, II)
0891      ASSIGN 95 TO KRTN
0892      IF(KTDAT .GE. 512)      GO TO(98,98,97)NPAGE
0893      95      CONTINUE
0894
0895      C ##### END OF DOPPLER LOOP #####
0896
0897      900      CONTINUE
0898
0899      C ##### END OF DATA COLLECTION LOOP #####
0900      C >> DIAGONSTICS >> SENSE SW. 10 (ON)
0901      C IF(ISSW(10) .LT. 0) WRITE(1, 9900)
0902      C9900      FORMAT(" END OF RANGE INDEX LOOP")
0903
0904      IBEAM = IBEAM + 1
0905      IF(IBEAM .GE. 16) IBEAM = IBEAM - 16
0906      KOUNTB = KOUNTB + 1
0907      IF(IBEAM .LE. IBM+1) MIDBM(IPNG) = KDIR

```

```

0908      NUSED = ITH(IPNG) + KTDAT
0909
0910      C >> DIAGONSTICS >> SENSE SW. 3 (ON)
0911      IF(ISSJ(3) .LT. 0) WRITE(1, 9096)KDIR,NUSED,IPNG
0912      FORMAT(14," DEGREE BEAM COMPLETE. ",14," DATA POINTS USED
0913      + " (OF 1536) FOR PING",13)
0914
0915      IF(KOUNTB .LT. KNUM) GO TO 228
0916
0917      C ***** END OF BEAM LOOP *****
0918
0919      97 LSTDAT = 999
0920      LA1 = MIN0(2044, LA1+4)
0921      NRBMS(IPNG) = KOUNTB
0922      CALL PUTQ(3, NQUAL)
0923      CALL PUTW(3, LA1, 999,II)
0924
0925      C >> > STORE CORE DATA BLOCK 3 TO DISC. (PING 1 BEGINS
0926      C IN BLOCK 1).
0927      98 NRBLK = 3 * (IPNG - 1) + NPAGE
0928
0929      C >> DIAGONSTICS >> SENSE SW. 3 (ON)
0930      IF(ISSJ(3) .LT. 0) WRITE(1, 9098) NRBLK
0931      9098 FORMAT(" WRITING TO DISC DATA BLOCK ", I3)
0932
0933      CALL PUTQ(3, NQUAL)
0934      CALL KYBD(2HMS, 31, NRBLK)
0935      CALL KYBD(2HMS, 21, 3)
0936      CALL KYBD(2HCL, 3)
0937      NPAGE = NPAGE + 1
0938      ITH(IPNG) = ITH(IPNG) + KTDAT
0939      NRBMS(IPNG) = KOUNTB
0940      KTDAT = 0
0941      LA1 = -4
0942

```

```

0943 C >> DIAGNOSTICS >> SENSE SW. 10 (ON)
0944 C IF(ISSU(10).GE. 0) GO TO 622
0945 C >> > VERIFY DATA JUST WRITTEN ON DISC
0946 C WRITE(1, 9998)
0947 C WRITE(1, 987) NQUAL
0948 C987 FORMAT(' NQUAL =', 518)
0949 C CALL PUTQ(3,NQUAL)
0950 C CALL KYBD(2HMS, 31,NRBLK)
0951 C CALL KYBD(2HMS, 11,3)
0952 C DO 620 J = 0, 100, 4
0953 C CALL GETU(3, J, N1, II)
0954 C CALL GETU(3, J+1, N2, II)
0955 C CALL GETU(3, J+2, N3, II)
0956 C CALL GETU(3, J+3, N4, II)
0957 C WRITE(1, 9620) N1, N2, N3, N4, J
0958 C9620 FORMAT(1X, 5110) GO TO 621
0959 C IF(N1.EQ. 999) GO TO 621
0960 C620 CONTINUE
0961 C621 WRITE(1, 9998)
0962
0963 C >> > TEST FOR LAST DATA & RETURN TO CALLER
0964 C622 IF(LSTDAT.EQ. 999) GO TO IRET
0965
0966 C ##### EXIT ANALYSIS ROUTINE #####
0967
0968 C >> > TEST FOR 1536 DETECTIONS
0969 C CALL PUTU(3, 0, 999, II) GO TO KRTN
0970 C IF(NPAGE.LE. 3)
0971 C WRITE(1, 990) IPNG
0972 C990 FORMAT(/'STOPPED AFTER 1536 DETECTIONS ON PING ',I4)
0973
0974 C >> > RETURN TO CALLER
0975 C GO TO IRET
0976
0977 C ##### EXIT ANALYSIS ROUTINE #####

```

```

0978
0979 99 WRITE(1, 999)
0980 999 FORMAT(" NO DATA RETURNED FROM SUBROUTINE RDATA")
0981 GO TO 97
0982
0983 C ***** END ANALYSIS ROUTINE *****
0984
0985 C *****
0986 C * SAVE ICOM IN DISC BLOCK 0 AND EXIT
0987 C *
0988 C *****
0989
0990 C >> DIAGNOSTICS >> SENSE SW. 1 (ON)
0991 800 IF(ISSW(1) .LT. 0) PAUSE 2
0992 NFFTAG = NFTSU
0993 CALL PUTQ(3, NQUAL)
0994 DO 801 J = 1, 127
0995 CALL PUTJ(3, J, ICOM(J), II)
0996 801 CONTINUE
0997 CALL KYBD(2HMS, 31, 0)
0998 CALL KYBD(2HMS, 21, 3)
0999
1000 C >> DIAGNOSTICS >> SENSE SW. 10 (ON)
1001 IF(ISSW(10) .LT. 0) WRITE(1, 9801) KLOAD, NFTSU, NQUAL
1002 9801 FORMAT(/, " CORELOAD", I2, " NFTSU=", I2, " NQUAL ", I5)
1003 IF(ISSW(10) .LT. 0) WRITE(1, 9999) ICOM
1004 9999 FORMAT("03", I5)
1005
1006 C >> DIAGNOSTICS >> SENSE SW. 10 (ON)
1007 IF(ISSW(10) .LT. 0) WRITE(1, 9998)
1008 IF(ISSW(10) .LT. 0) PAUSE 2000
1009
1010 802 CALL ENDIO
1011 CALL CDUMP(23B)
1012 805 N = KLOAD + 1

```

```

1013 C > > > TRANSFER TO SELECTED CORELOAD
1014 CALL CLOAD(KLTRK(N), KLADR(N) )
1015
1016 C ***** TRANSFER TO SELECTED CORELOAD *****
1017 END
1018
1019 SUBROUTINE STGRD (NRCDS,NRBMS,ITH)
1020
1021 C
1022 C *****
1023 C *****
1024 C * STGRD SUBROUTINE DRAWS GRIDS AROUND THE DATA STATUS TABLES *
1025 C *****
1026 C *****
1027 C *****
1028 C *****
1029 C *****
1030 C *****
1031 C *****
1032 C *****
1033 C *****
1034 C *****
1035 C *****
1036 C *****
1037 C *****
1038 C *****
1039 C *****
1040 C *****
1041 C *****
1042 C *****
1043 C *****
1044 C *****
1045 C *****
1046 C *****
1047 C *****

```



```

1048 CALL TPLOT (1,326,IYHL)
1049 CALL TPLOT (0,434,632)
1050 CALL TPLOT (1,434,IYHL)
1051 CALL TPLOT (0,588,632)
1052 CALL TPLOT (1,588,IYHL)
1053 CALL CHOUT (378)
1054
1055 C > > > DRAW GRID FOR ANALYZED DATA STATUS TABLE
1056 7010 IYHLA - IYHL - 45
1057 IYHL - IYHL - 89
1058 CALL TPLOT (0,0,IYHLA)
1059 CALL TPLOT (1,840,IYHLA)
1060 CALL TPLOT (0,0,IYHL)
1061 CALL TPLOT (1,840,IYHL)
1062 DO 7015 I=1,10
1063 IF (NRBMS(I) .EQ. 0 .AND. ITH(I) .EQ. 0) GO TO 7015
1064 IYHL - IYHL - 22
1065 CALL TPLOT (0,0,IYHL)
1066 CALL TPLOT (1,840,IYHL)
1067 7015 CONTINUE
1068 CALL TPLOT (1,840,IYHLA+22)
1069 CALL TPLOT (1,0,IYHLA+22)
1070 CALL TPLOT (1,0,IYHL)
1071 CALL TPLOT (0,112,IYHLA)
1072 CALL TPLOT (1,112,IYHL)
1073 CALL TPLOT (0,294,IYHLA)
1074 CALL TPLOT (1,294,IYHL)
1075 CALL TPLOT (0,490,IYHLA)
1076 CALL TPLOT (1,490,IYHL)
1077 CALL TPLOT (0,672,IYHLA)
1078 CALL TPLOT (1,672,IYHL)
1079 CALL CHOUT (378)
1080 C > > > RETURN TO CALLER
1081 RETURN
1082 C ##### END OF STATUS TABLE GRID DRAWING SUBROUTINE #####

```

```

1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117

END

SUBROUTINE RBINS(IPNG, JSTRT, JEND, KOFFS, MIN, MAX, INCRX)

*****
* THIS ROUTINE CONVERTS START AND STOP RANGE IN KILOYARDS *
* TO MIN AND MAX RANGE INDICES FOR PLOTS. *
* START AND END TIMES (FROM END OF XMIT. PULSE) ARE OBTAINED *
* FROM CALLING HEADER DATA STORED DURING DATA COLLECTION. *
* INCRX IS THE LENGTH OF A RANGE BIN IN PLOT POINTS. *
*****

DIMENSION IHDR(16)

CALL HEADR(IPNG, IHDR)
KOFFS = IHDR(11)
MIN = IFIX( (JSTRT * 3000./2500. - KOFFS)/0.192 + 0.5)
MAX = IFIX( (JEND * 3000./2500. - KOFFS)/0.192 + 0.5)
INCRX = IFIX(896.0 / (MAX - MIN) + 0.5)

C >> DIAGONSTICS >> SENSE SW. 10 (ON)
C IF(ISSW(10) .LT. 0) WRITE(1, 7110)MIN, MAX, INCRX, IPNG,
C * JSTRT, JEND, KOFFS
C7110 FORMAT('MIN, MAX, INCRX =', 7I6)
C IF(ISSW(10) .LT. 0) PAUSE 10

C > > > RETURN TO CALLER
RETURN

C ##### END OF RANGE BINS CONVERT SUBROUTINE #####
END

SUBROUTINE PTIME(IPNG, ICOM)

```

```

1118 C *****
1119 C * THIS ROUTINE CONVERTS PING TIME FROM BCD UNITS (OBTAINED) *
1120 C * FROM THE HEADER WRITTEN DURING DATA COLLECTION) INTO *
1121 C * BINARY FORM. RESULTS ARE STORED IN PROPER ICOM LOCATIONS. *
1122 C *****
1123 C *****
1124
1125     DIMENSION ICOM(127), IHDR(16), IBCD(6), MASK(4)
1126
1127     DATA MASK/17B,360B,7400B,170000B/
1128
1129     CALL HEADR(IPNG, IHDR)
1130 C > > > HOURS
1131     IBCD(5) = IAND(MASK(1), IHDR(1) )
1132     IBCD(6) = IAND(MASK(2), IHDR(1) )
1133     ICOM(14) = IBCD(6) / 20B * 10 + IBCD(5)
1134 C > > > MINUTES AND SECONDS
1135     DO 10 I = 1, 4
1136     10 IBCD(I) = IAND(MASK(I), IHDR(2) )
1137
1138     ICOM(15) = IBCD(4) / 10000B * 10 + IBCD(3) / 400B
1139     ICOM(16) = IBCD(2) / 20B * 10 + IBCD(1)
1140
1141 C > > > RETURN TO CALLER
1142     RETURN
1143
1144 C ***** END OF PING TIME SUBROUTINE *****
1145
1146 C *****
1147 C ***** END OF ANALYSIS & SYSTEM EXECUTIVE SUBROUTINE *****
1148 C *****
1149 C *****
1150     END
1151     ENDS

```

OLAAS DATA COLLECTION PROGRAM - Y0001

1. FUNCTION. Y0001 collects, demultiplexes, and stores selected beam data on the disc.
2. CONSTRAINTS. None.
3. CALLING SEQUENCE. N/A
4. DESCRIPTION OF INPUT. See Figure 3 for description of Beam Former input and method. See Figure 4 for description of Buoy Data input and method. See Figure 5 for description of Time Code Generator input and method.
5. DESCRIPTION OF OUTPUT. All output is to the disc. See Figure 6 for format.
6. FILES USED. N/A
7. ERRORS AND RESTRICTIONS. The amount of data which can be written is limited by the Track Allocation Table. If an attempt is made to write beyond the allocated tracks, an irrecoverable error halt will occur.
8. COMPUTER OPERATOR INSTRUCTIONS. See general system operating instructions.
9. DESCRIPTION OF PROCESSING. Y0001 must keep track of whether the current execution is the first execution or not. When it is not the first execution, it has been read in from disc tracks to which it was written out following its previous execution, therefore, the states of all switches, counters, etc., have been saved. At (1), assuming this is the initial execution (ping 1), the not first time switch NFTSW will be zero. NFTSW will be set non-zero (not first time) and the operator parameters N, L, K, M, D, S are set to illegal values (-1). INPUT is called to obtain these parameters from the operator. The variables #BMS, TOTPG, and PING are initialized. At (2) the number of output blocks per ping is computed. This is the number of output buffer writes, consisting of 4 sectors per beam written. As the input data rate is 10667 Hz for 16 beams, the number of blocks expected during the ping can be computed from N, the amount of record time. The rounded, complemented result is stored in BLOCK and is used by the data retrieval routine to locate data on the disc. This number is rounded up to ensure that at least N seconds of data are recorded. The complemented result is stored in BLOCK and is used to time the recording process as well as used by the data retrieval routine to locate data on the disc.

At (3) the initialization of disc access pointers begins. One of the variables involved is head base, HEADB, which is the number of the lower head for the disc being used. This is either incremented or not according to which surface is being used. HEADB, TRAK, the current track, and MXTRK, the last track, are initialized for the lower disc. This means setting them to 2, FTLD, and LTLD, respectively. FTLD and LTLD are the first and last tracks for the lower disc as specified in the Track Allocation Table. MXTRK (from LTLD) is checked. If zero, no lower disc space has been allocated and the three variables are set to 0, FTUD, and LTUD, respectively, for the upper disc. If MXTRK is zero this time, no disc space at all has been allocated and an error exit is taken. LTUD is set to zero to indicate that the upper disc is currently in use. For either disc, current sector, SECT, is initialized to zero. All of these variables are calculated only once, on the first execution.

At (4) the logic flow from the not first time case rejoins us. A zero (NOP) is stored in cell 6. This is necessary because an interrupt request is pending from the core loading process, and will cause an interrupt to occur as soon as interrupts are enabled. There appears to be no way to handle this other than allowing it to occur and returning immediately to the interrupted program. The NOP does this. The block counter, BCTR, is set from BLOCK. SWAIT is called to wait for the transmit gate. After it occurs, the record delay, K, is multiplied by 10 to convert it to 0.1 second periods. This is necessary to increase the resolution of the delay. STDLY is called to initiate the counting of a period of K seconds. It returns immediately, leaving interrupts enabled and counting time base generator (TBG) interrupts as they occur. TDIN is called to input the time from the Time Code Generator and this data (2 words) is stored in the output buffer.

At (5) BDIN is called to input the 72 word Buoy Data block from the Avionics package. Of this, only the compass word, word 55, is used. The compass word is the heading of beam 1 in Gray code as a fraction of a revolution in bits 15-8. This is converted to weighted binary, then to degrees, then stored in ANGL1 and in the output buffer. The number of the first beam to be stored is then computed from the operator's desired direction, D, number of beams, L, and ANGL1. The equation is:

$$FSTB\# = \frac{D-ANGL1}{22.5} + 1 - \frac{L-1}{2}$$

The D-ANGL1 portion must be converted to positive degrees before proceeding. The result of the entire equation, a beam number, must be rounded before fixing. If the result is negative or zero, it is converted to a positive number by adding 16 (one revolution). Likewise, if the result is greater than 16, 16 is subtracted. The final result is stored in first beam number, FSTB#, and in the output buffer.

At (6) the angle of the first beam recorded is computed:

$$(FSTB\#-1) * 22.5 + ANGL1$$

The result is rounded, corrected by subtracting 360° if greater than 360° , and stored in the output buffer. The remaining header data is stored in the output buffer. The input buffer pointers are switched in preparation for Beam Former input.

At (7) DELAY is called to wait until the delay initiated by STDLY is over. BFIN is called to initiate input of 8192 words of Beam Former data. BFIN returns control immediately, so that the following processing takes place while input continues. Interrupts are turned off. WDISC is called to output the header record. Interrupts are turned back on and a delay of 1.2 seconds is begun. If this delay times out, this is an error condition, as the Beam Former completion interrupt should have occurred within this time.

When the Beam Former completion (DMA) interrupt occurs, control is transferred to location BFDUN, at (9). There the input buffer pointers are switched. BCTR is incremented and if not zero, BFIN is called again to start the next 8192 words of Beam Former input. If BCTR is zero, BFHLT is called to stop Beam Former input. In either case, remembering that BFIN returns control immediately, interrupts are turned off and demultiplexing (formatting) begins.

Formatting, at (10), is described much better by the flow chart than it can be in text, therefore the user is directed to the flow chart for details.

When formatting has been completed, there will be between 512 and 8192 words in the output buffer, depending upon the number of beams requested. At (11) the number of words in the buffer is computed ($L*512$) and that number of words is output by calling WDISC. A check is made on whether a Beam Former completion interrupt is waiting. If it is, the formatting and writing process took too long and consequently sync has been lost, so an error exit is taken. Otherwise, interrupts are turned on. BCTR is checked. If non-zero, control is transferred back to (8) to wait for Beam Former completion. If BCTR is zero, all requested data for this ping has been collected and the Beam Former has already been turned off. Ping number is incremented, the current core load is swapped out and the data analysis core load is read in. No check is made on whether PING has reached the limit TOTPG. It is the responsibility of the data analysis core load not to return control if all required data collection is finished.

Subroutines STDLY, TBGI, and DELAY work together. STDLY sets counter TCTR to the complement of the number of 0.1 second intervals passed to it in the calling sequence. It sets up the trap cell with a JSB through a base page

pointer to TBGI, starts the TBG at 0.1 second intervals, turns on interrupts, and returns. TBGI is entered when a TBG interrupt occurs. It turns off interrupts, saves the registers, and inputs the TBG status. If non-zero, a TBG tick has been missed, and an error exit is taken. Otherwise, TCTR is incremented. If TCTR is then zero, the desired count has been reached and the TBG is turned off with a CLC. Otherwise, it is re-enabled with a CLF. In either case, the registers are then restored, interrupts are turned on, and control is returned to the point of interruption. Subroutine DELAY is called to determine the state of TCTR. If zero upon entry, an error exit is taken, as the count was exhausted at an unknown time in the past. Otherwise, the program loops until TCTR is zero and then returns.

Subroutine WDISC writes data as a sequential access file on the disc. It assumes that TRAK, SECT, and HEADB are set to specify the starting location on the disc. HEADB will always show the lower head for the disc in use; SECT varies from 0-47 to cover both surfaces. WDISC sets the core address and count into WPTR and WCTR from the calling parameters. At WDIS1 it computes AA, the number of words to be written on the current track. This will be the minimum of WCTR or the number of words remaining on the track. WRITD is called and will write AA words starting at TRAK and SECT. Next, SECT is updated. Since all write requests must be a multiple of 128 words (one sector), all that is required is to add AA/128 to SECT. If AA/128 is not an integer, an error path is taken. If the result is 48, SECT is reset to zero and TRAK is incremented by one. SECT+AA/128 will not be greater than 48 because AA was chosen to write only to the end of the current track. If TRAK is updated, it is compared to MXTRK. If greater, switching to the upper disc is called for, and TRAK and MXTRK are initialized from FTUD and LTUD. If MXTRK is now zero, either the upper disc was already being used, or no space was allocated on the upper disc, and an error exit is taken. Otherwise, LTUD is set to zero to indicate upper disc in use, and HEADB is set to zero. At WDIS2, WPTR and WCTR are updated by the number of words written, AA. If WCTR is non-zero, the all data has not been written and control is transferred back to WDIS1 to update AA and write additional data. This continues until WCTR is zero and then control is returned to the caller. Upon return, TRAK, SECT, and HEADB are properly set to continue writing from the point at which writing was discontinued.

WRITD, called by WDISC, first converts SECT (range 0-47) and HEADB to HDSEC, the control word for head and sector which uses sector in the range 0-23. The conversion consists of using head and sector as is if sector is less than 24, and otherwise, head plus one and sector less 24. This is then shifted into the bit positions used by the disc controller. DMA is set up to write AA words starting at the location in WPTR. The remainder of the disc processing is as specified in the disc interface manual with the exception that DMA 7 is used. DMA 7 is used because DMA 6 is in use for input during disc output. If status following the write is bad, an error path is taken.

SWAIT waits for the transmit gate, or S interrupt. To do so, it disables interrupts, performs a set control - clear flag on the time code generator interface, to which the S interrupt is connected, and loops until flag is set on the time code generator interface. SWAIT then clears control on the interface, turns interrupts back on and returns.

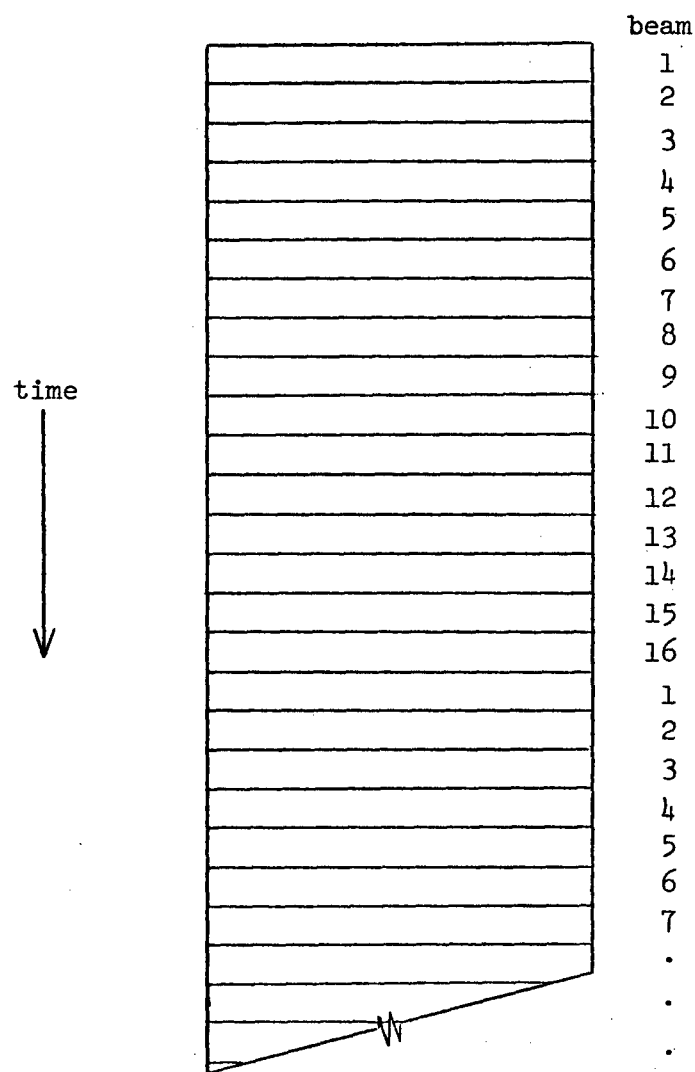
BFIN starts Beam Former input. It first verifies that the number of words called for in the calling sequence is a multiple of 16. If not, an error exit is taken. It sets up a JSB BFIT through a base page linkage in the DMA 6 interrupt trap cell, and an error halt in the Beam Former trap cell. The error halt is necessary to detect any delay in processing the DMA completion interrupt. DMA 6 is initialized for input of the number of words given in the calling sequence, the interface is enabled, DMA is started, and the Beam Former is started by outputting a 1 to the interface. This order is important, as otherwise the input does not dependably start with beam 1. Once input is started, control is returned to the caller. One of the calling parameters is the location to which control is to be transferred when input is complete. The immediate return to the caller is what allows the double buffering of input. As there are only about 93 microseconds between Beam Former words, and it is essential not to miss any, it would otherwise not be possible to do the formatting and output between BFIN calls. As it is, there is just enough time after a completion interrupt to start the next input request before the next word arrives.

BFIT processes the Beam Former DMA completion interrupts. It turns interrupts off, resets DMA 6, and branches to the completion address specified in the BFIN call. BFHLT is called from the main program to stop Beam Former input at the end of a ping. To do so, it outputs a zero to the interface, thus readying it for the synchronizing 1 on the next ping, clears control to prevent further interrupts, and returns.

BDIN inputs the 72 word Buoy Data record to the address specified in the calling sequence. The input method is the same as BFIN, but the word count is constant and control is not returned until input is complete. Therefore the use of interrupts is not required and all processing is performed with interrupts off and using the skip-on-flag-set method to detect completion. When done, a zero is output to the interface, control is cleared on the interface and DMA, interrupts are turned back on, and control is returned to the caller.

TDIN inputs the time from the Time Code Generator. As there is no signal which indicates when the output is not changing, the input method is to look for two consecutive identical samples. A brief analysis will show that if the equipment is functioning properly, two consecutive identical samples must occur within four samples. TDIN simply samples until identical samples are found, or the count of four is reached, at which point an error exit is

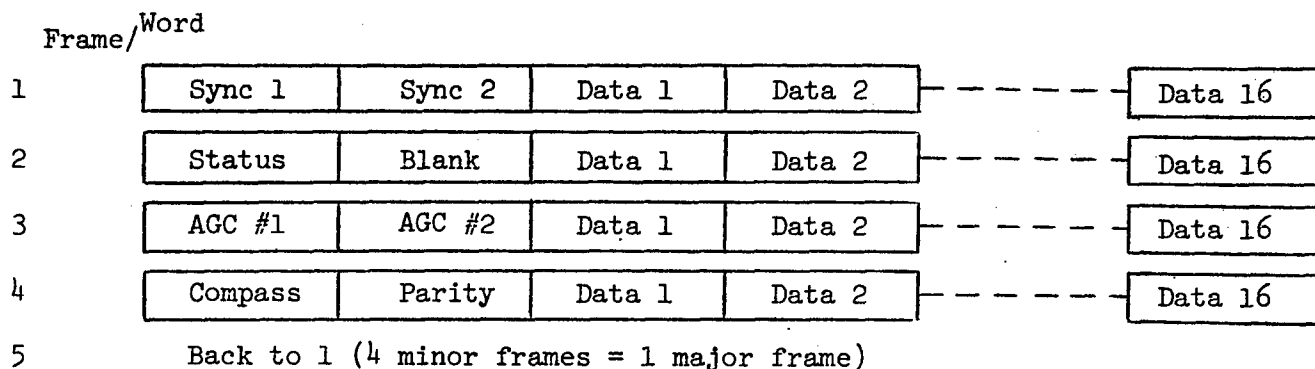
taken. Subroutine TIN performs the actual sampling. It outputs a 1 in bit zero, which gates hours into the input register, and reads this into the A register. It then outputs a 1 in bit 1, which gates minutes and seconds into the input register, and reads this into the B register. Both registers are complemented (as the inputs are inverted) and control is returned to TDIN with hours in A and minutes and seconds in B.



When the level of the Beam Former Interface Card output bit 0 changes from 0 to 1, the next input word will be beam 1. Input data rate is 10667 words per second.

Figure 3. Beam Former Input Format

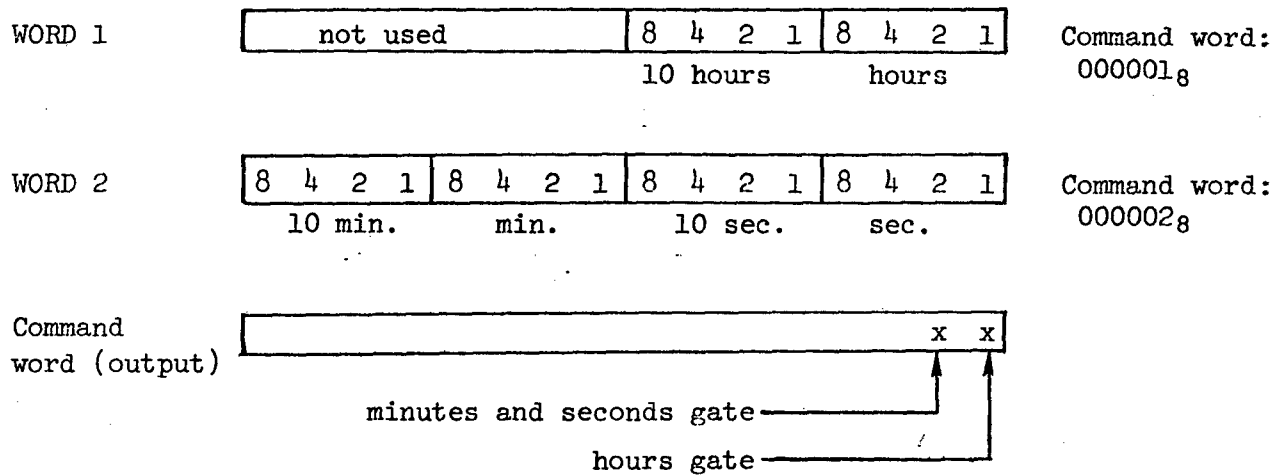
A data frame consists of 18 8-bit binary coded words, with the following format:



The third word in the frame is the sample from the reference stave. The fourth word is the stave 2 sample, and so on in serial order up to the 18th word with stave 16. Staves 2 through 16 follow in clockwise order from the reference stave on the array, when viewed from the top of the array.

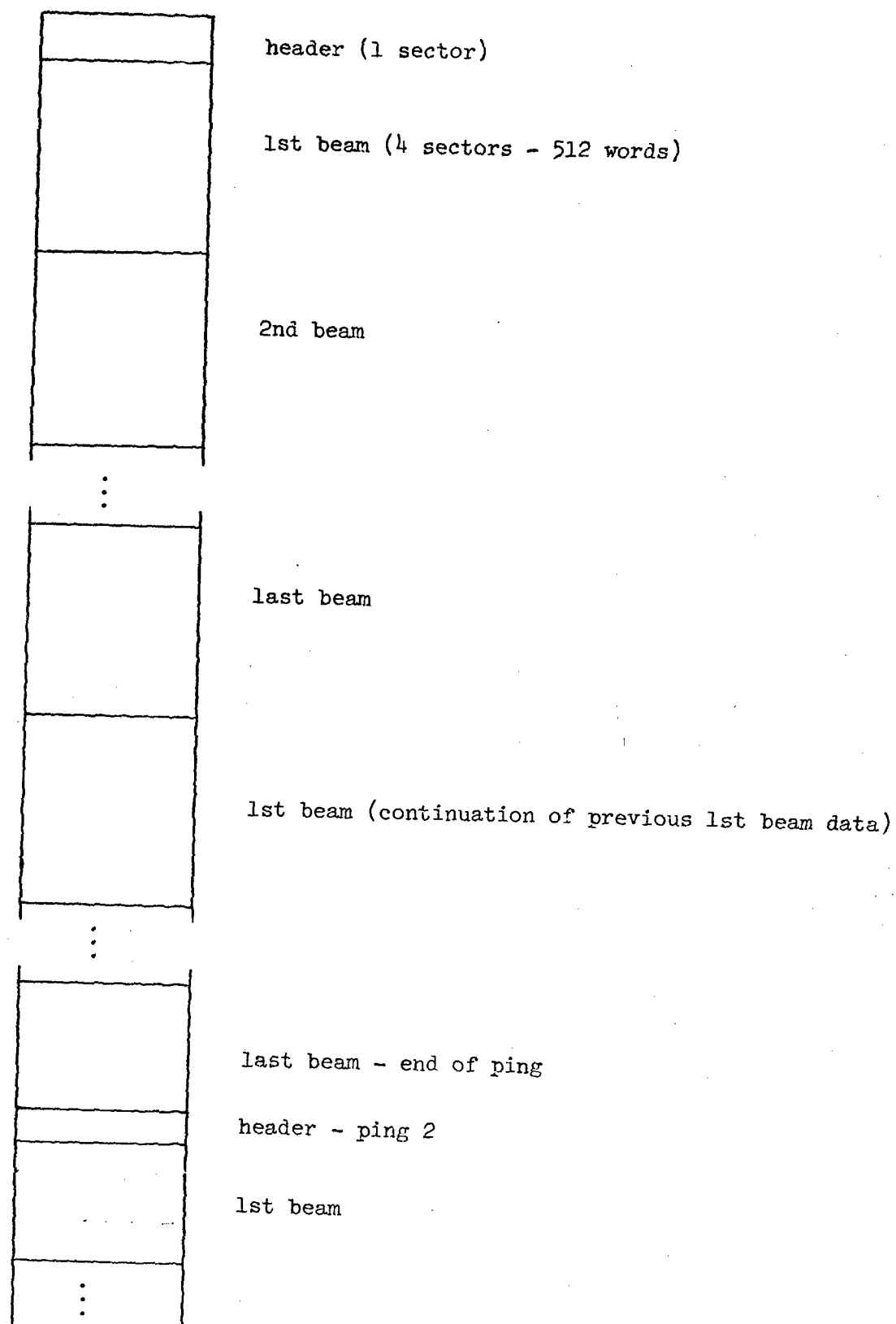
As received by the computer, each data word occupies the most significant 8 bits of the 16-bit word. Negative numbers are represented by their two's complement. Synchronization is achieved by output of a 1 to the Buoy Data interface card output register. When this is done the next input word will be word 1 of frame 1. Input is then continuous at 10667 words per second until stopped.

Figure 4. Buoy Data Input Format



NOTE: All inputs are inverted (one's complement)

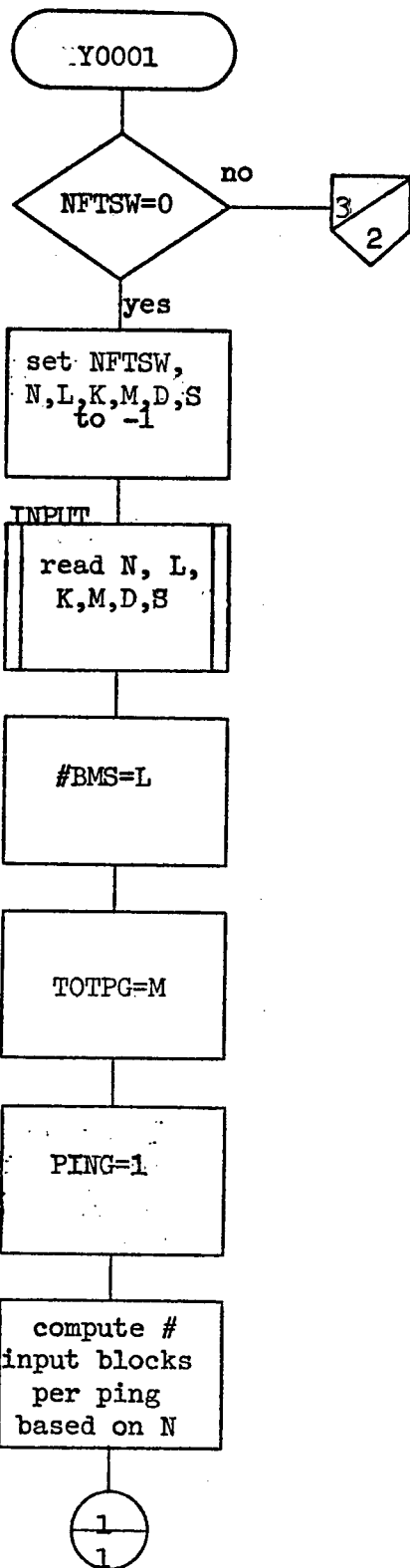
Figure 5. Time Code Generator Input Format



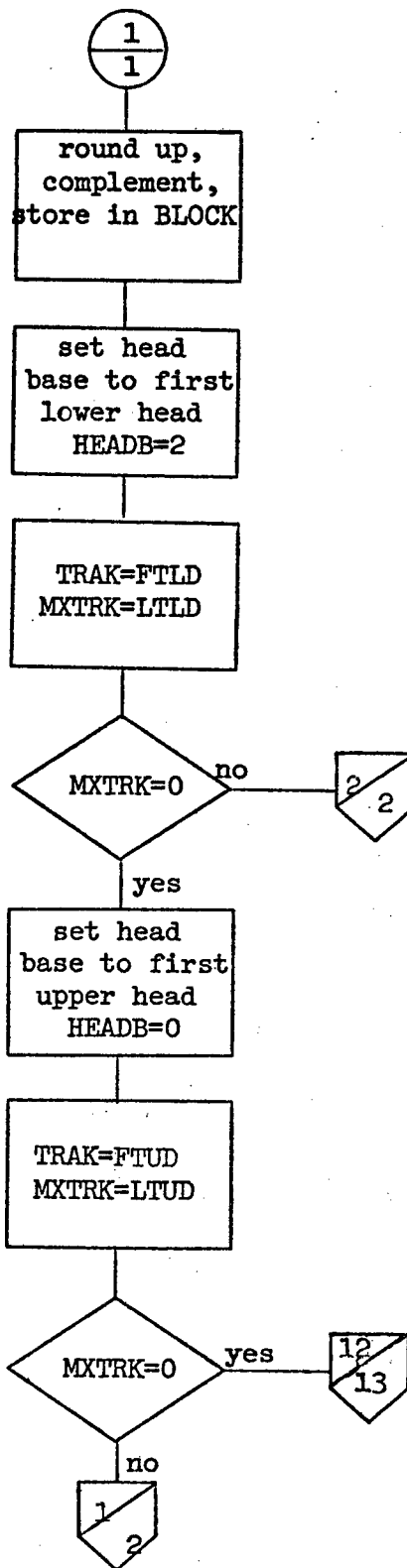
Actual amount of data recorded is dependent upon number of beams, pings, and recording time. Track boundaries occur every 48 128-word sectors.

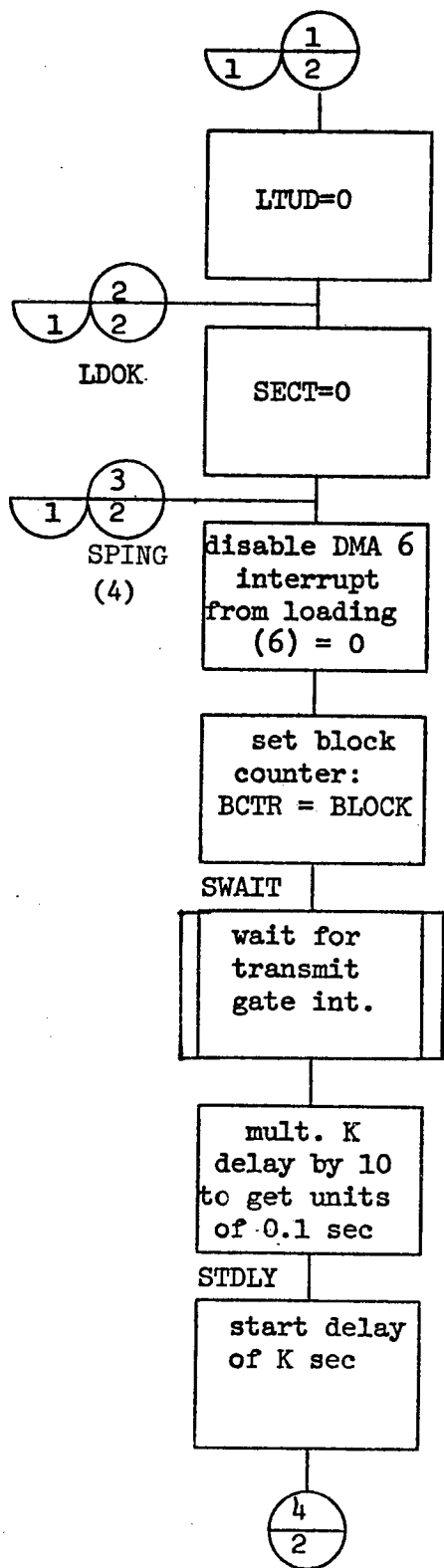
Figure 6. Disc Data Format

(1)

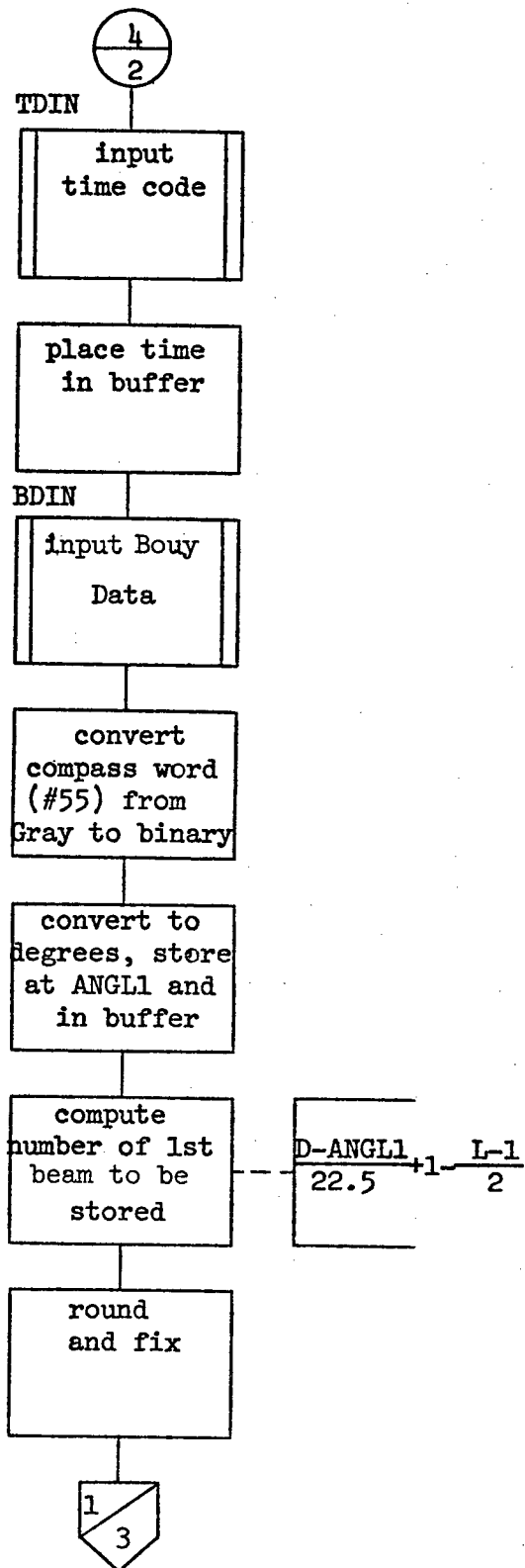


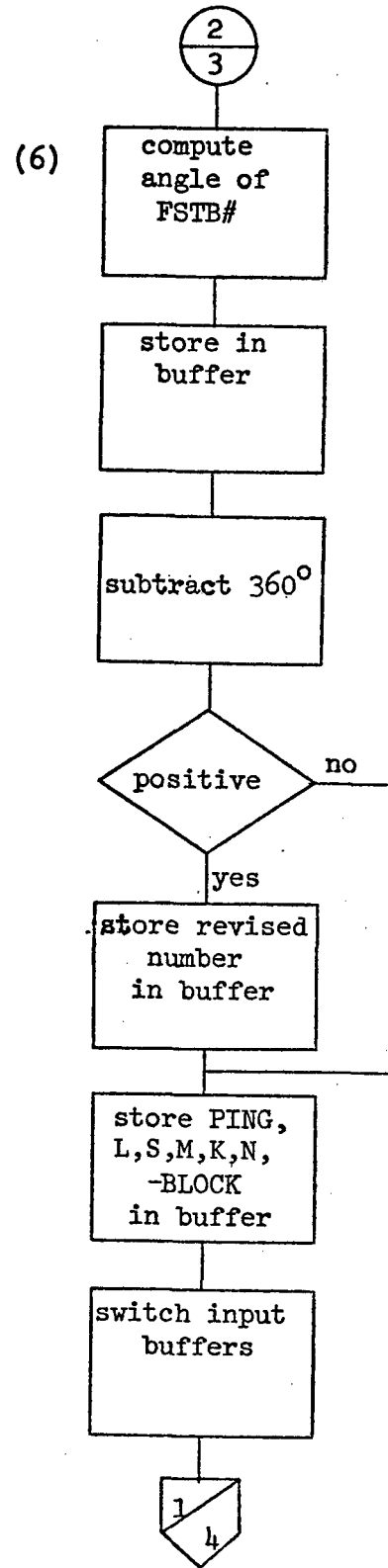
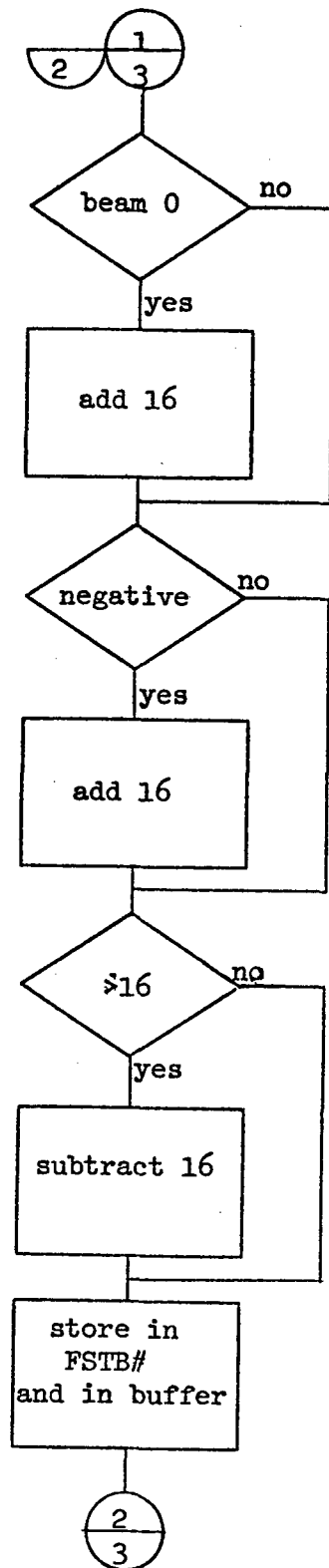
(3)



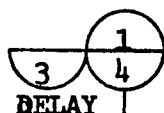


(5)

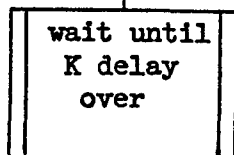




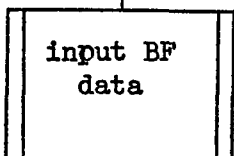
(7)



DELAY

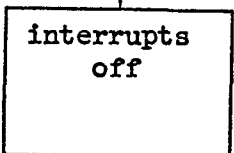


BFIN

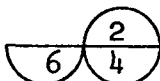
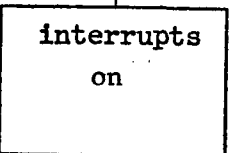
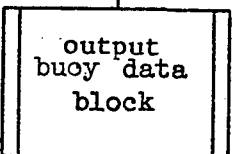


done

immediate exit

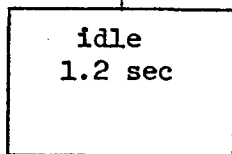


WDISC



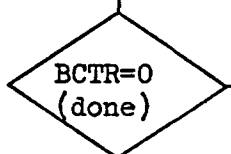
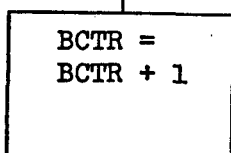
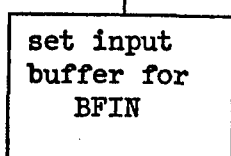
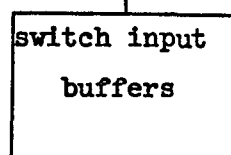
BFDLY

(8)



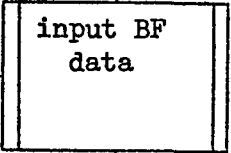
error,
too much
time

BFDUN
(8)



yes

BFIN

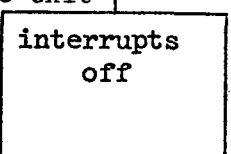


BFMOR

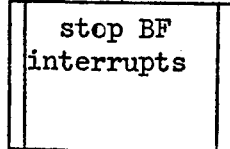
done

immediate exit

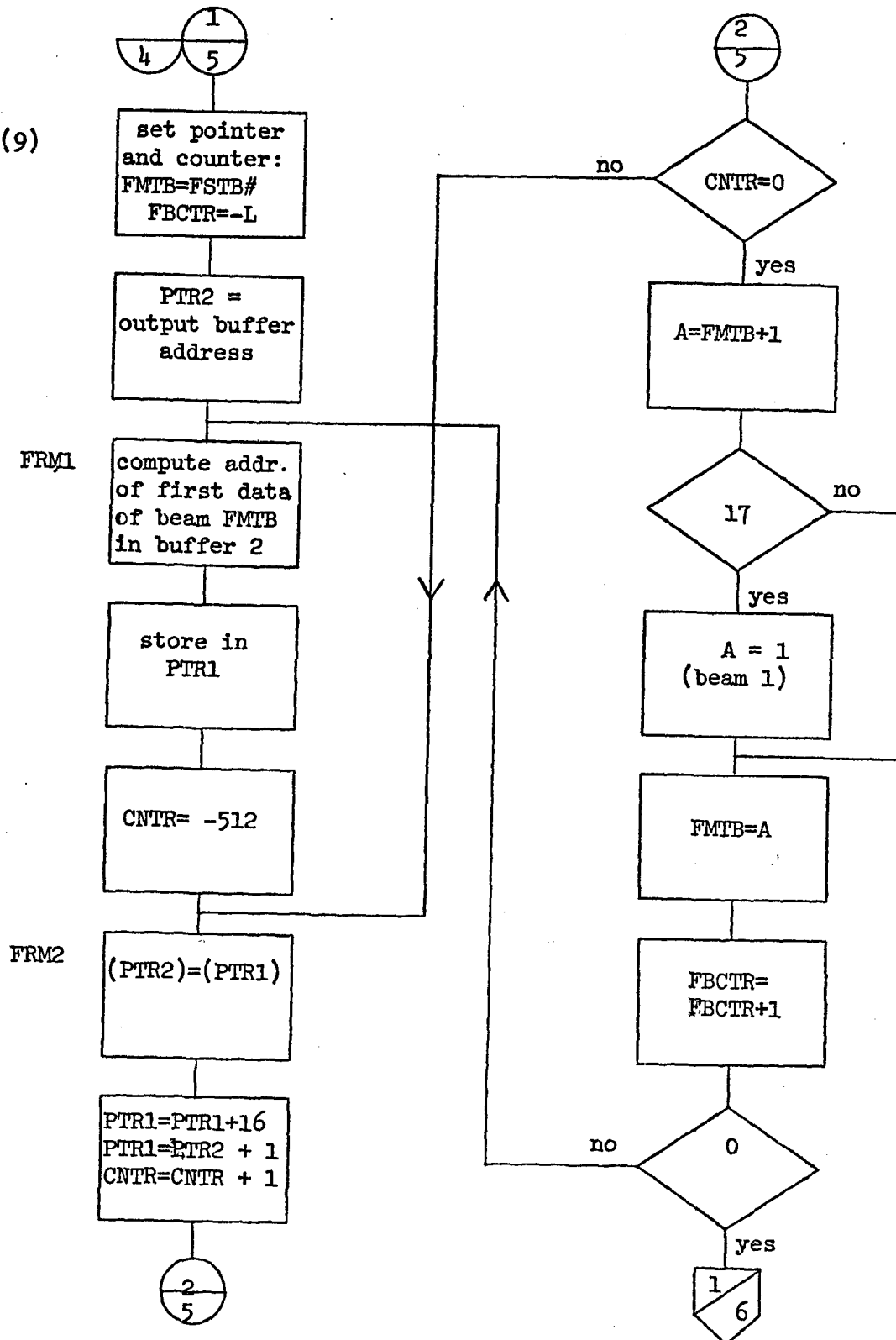
FRMAT



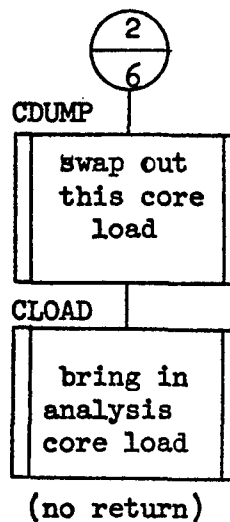
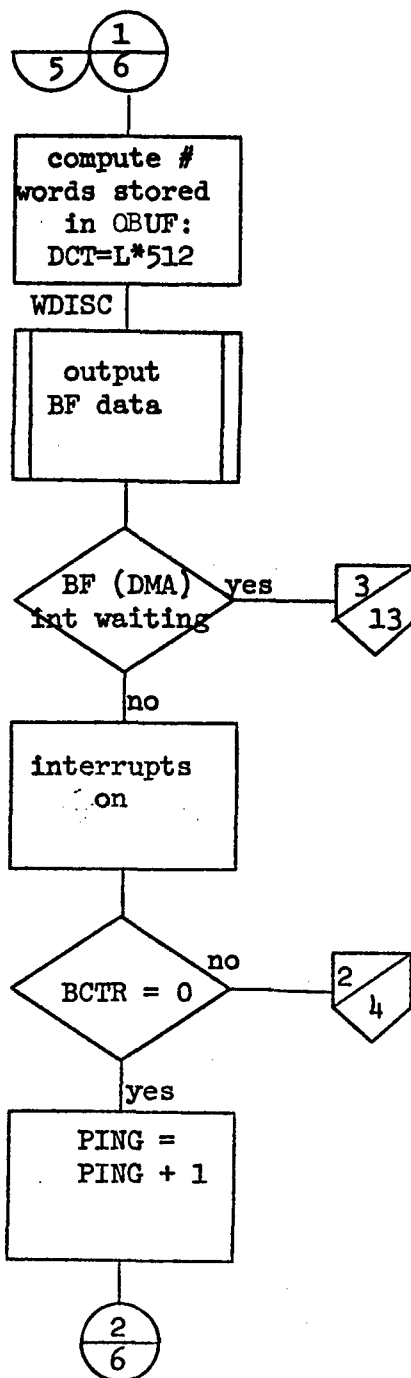
BFHLT

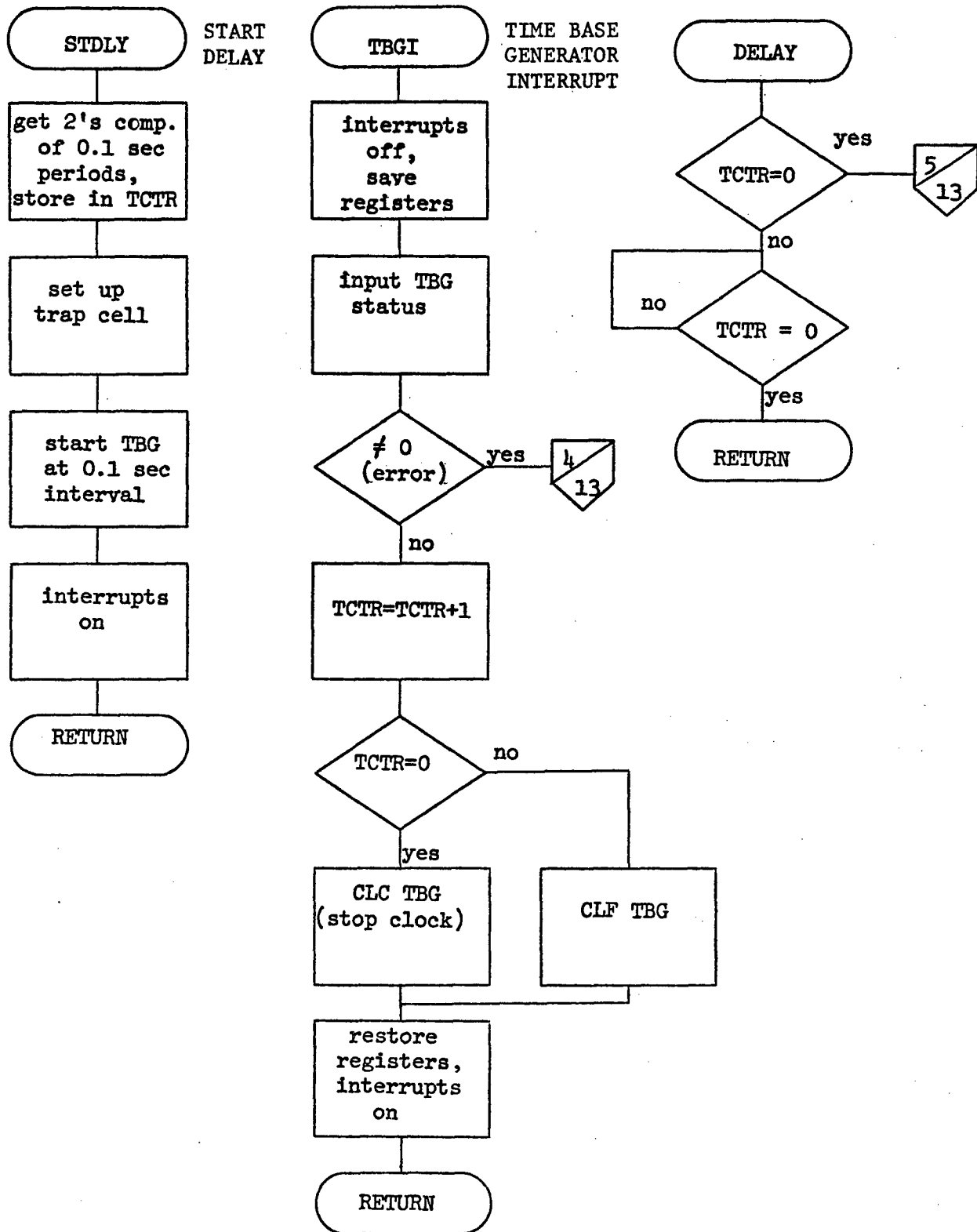


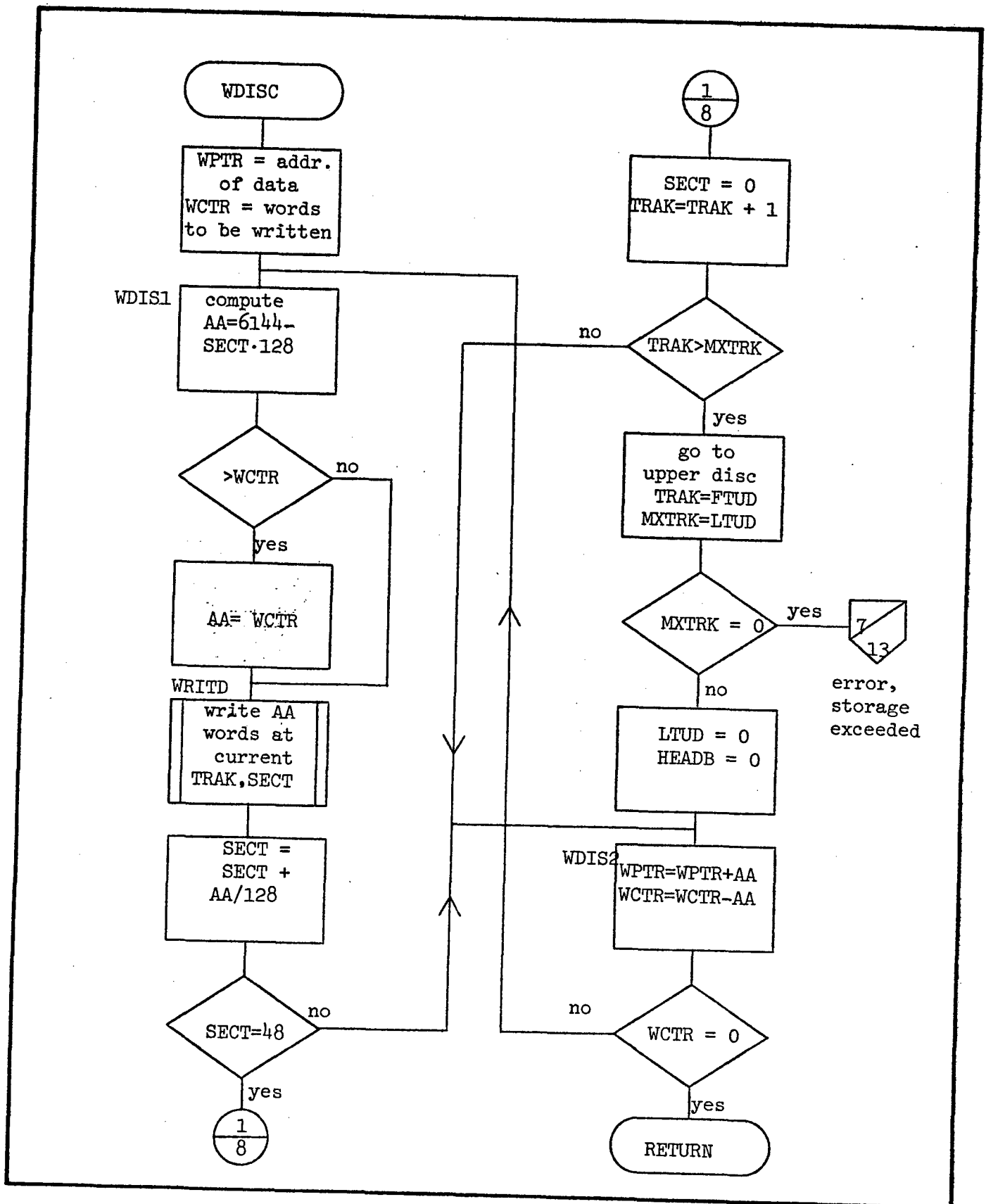
(9)

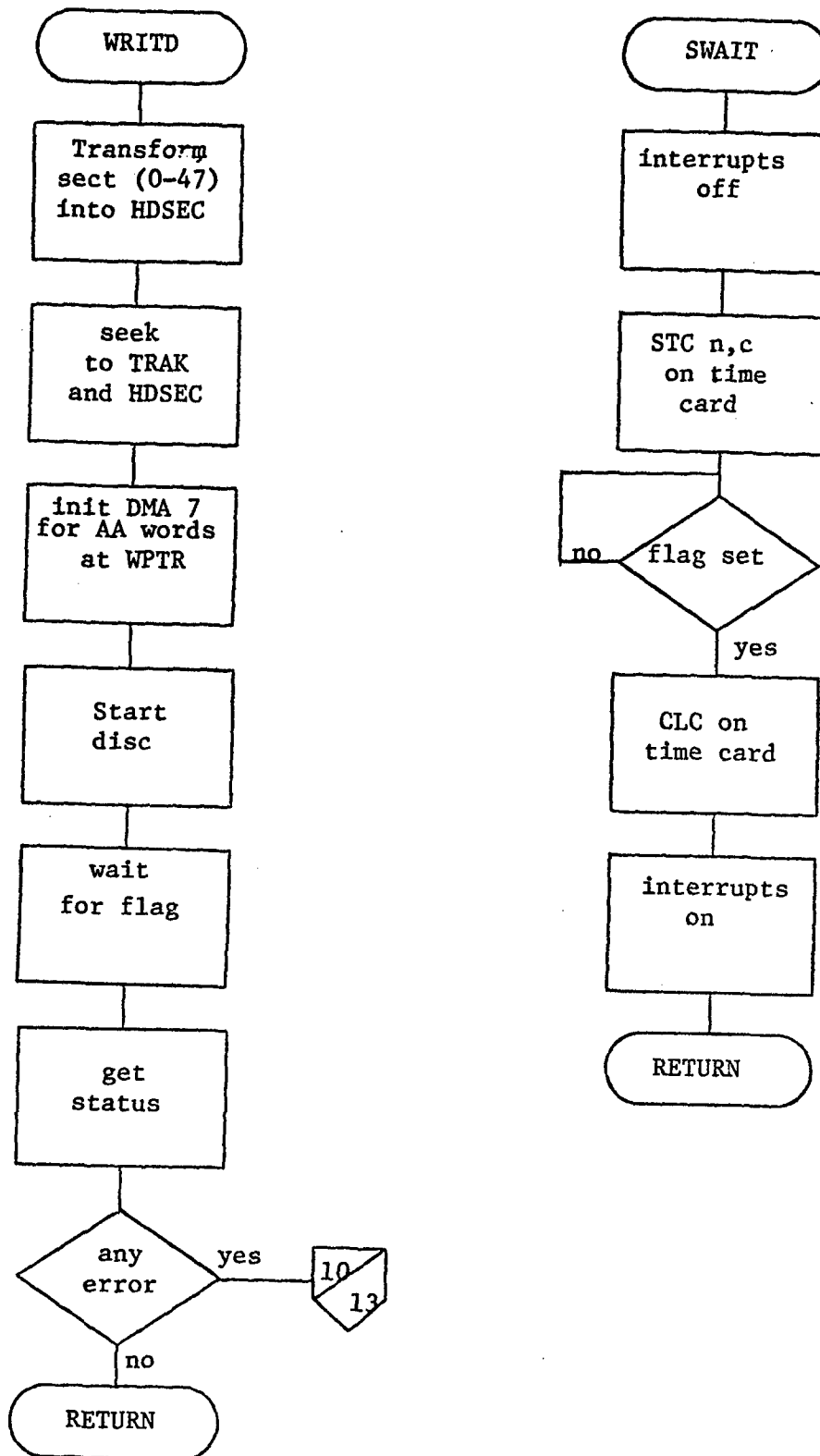


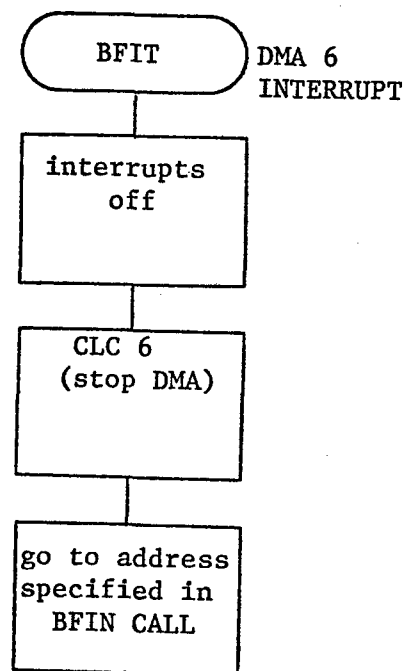
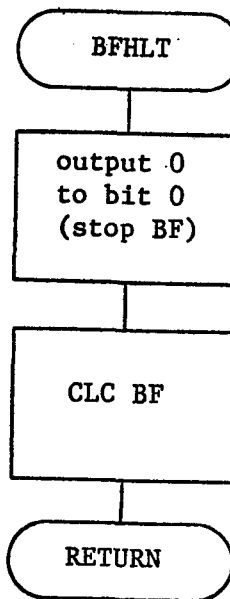
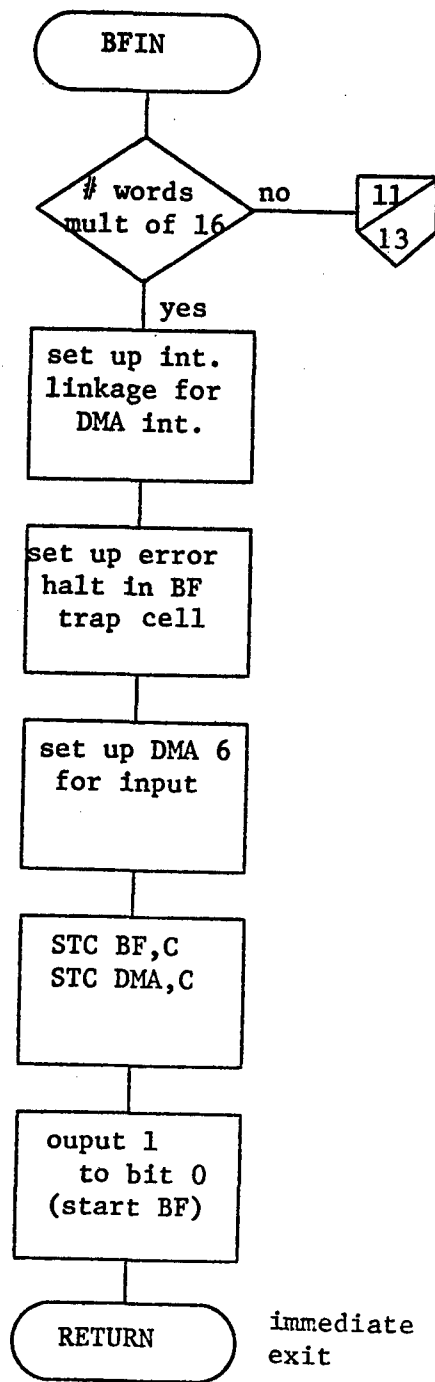
(10)

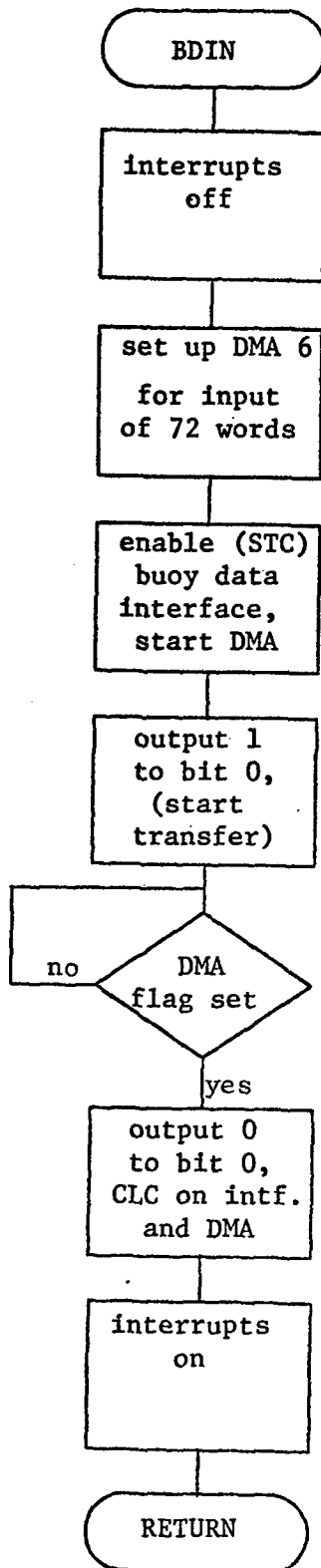


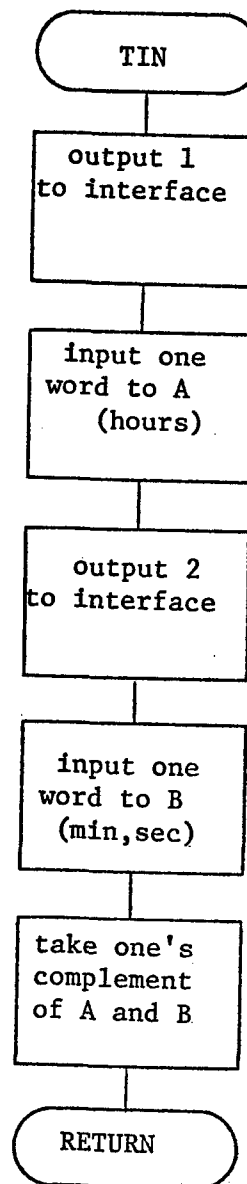
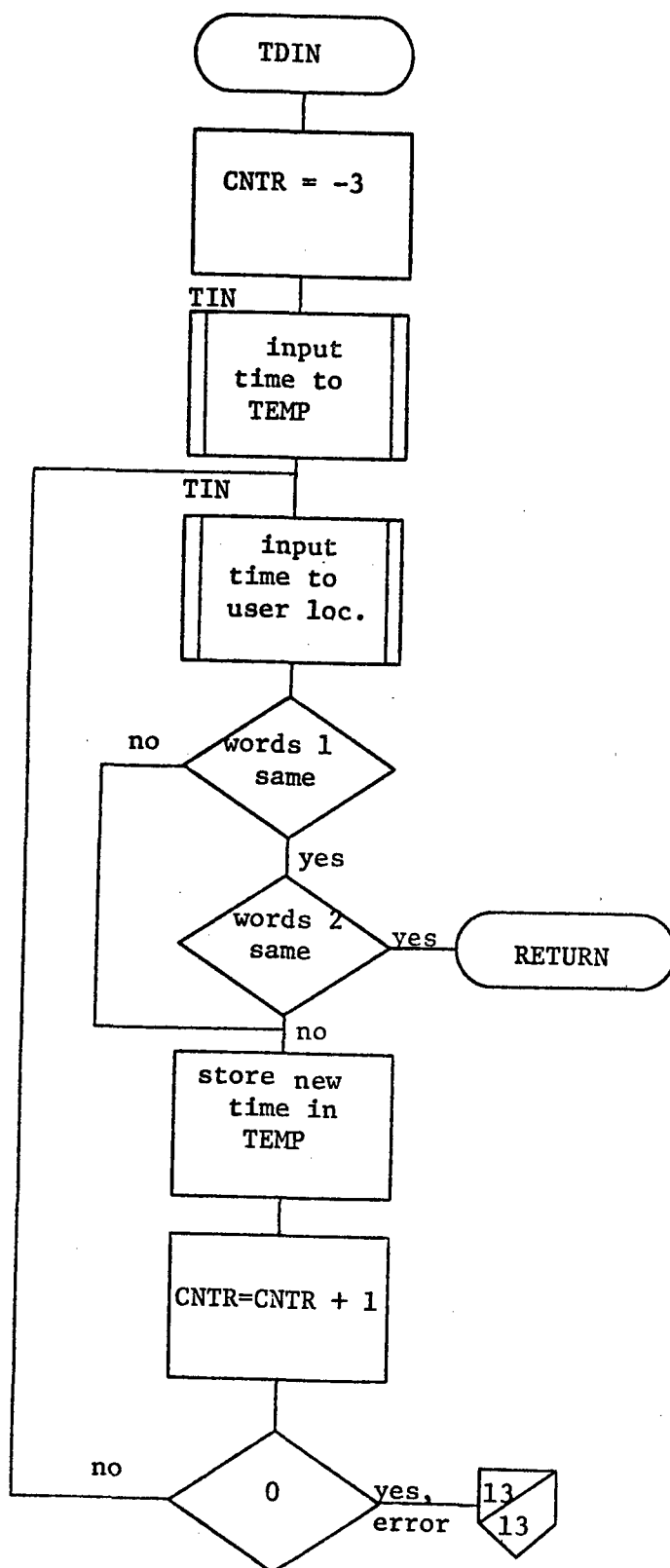


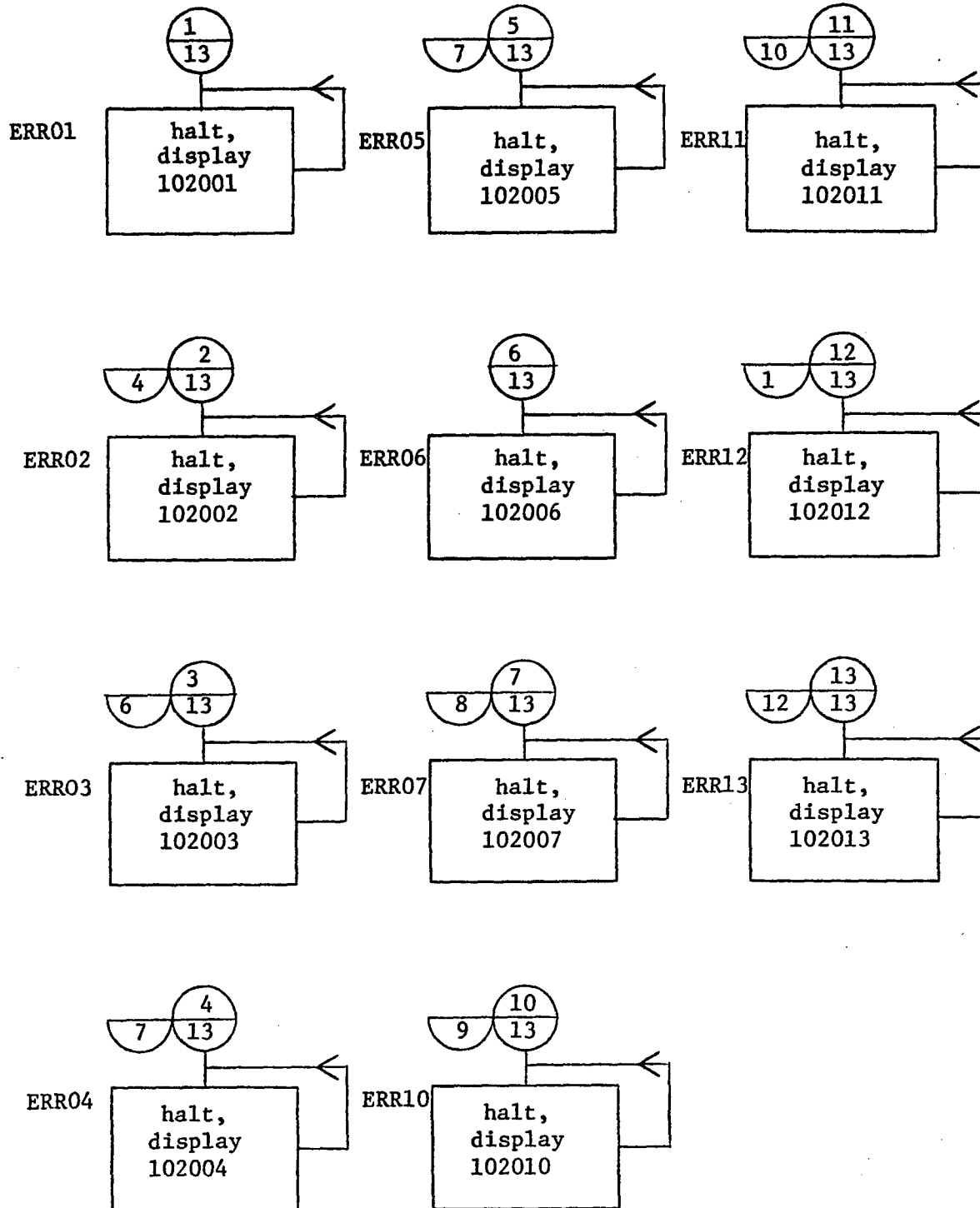












```

0001 ASMB,R,F,L
0002 NAM Y0001
0003 ENT Y0001,BFIT,TBGI
0004 EXT .ENTR,INPUT,LKTBG,LINK6,CDUMP,CLOAD
0005 EXT FTLD,LTLD,FTUD,LTUD
0006
0007*
0008*
0009*
0010*
0011*
0012*
0013*
0014*
0015*
0016*
0017*
0018*
0019*
0020*
0021*
0022*
0023*
0024*

*****
*
* TITLE: OLAAS RAW DATA COLLECTION SUBROUTINE
*
* MODULE: Y0001 SOURCE: Y0001 VERSION: X00
*
* OBJECT: #Y0001
*
* AUTHOR: R. TASHJIAN DATE: 1976
*
* MODULE FUNCTION: COLLECT BEAMFORMER RAW DATA AND STORE
* THAT DATA ON THE OLAAS FOURIER SYSTEM DISC.
*
* CALLING SEQUENCE: CALLED BY - Y1002 (ANALYSIS PGM.)
*
* CALL CLOAD (15B,2) (FOR CORELOAD 2)
* OR
* CALL CLOAD (31B,2) (FOR CORELOAD 4)
*
*****

```

[illegible]

PAGE 0003 #01 OLAAS DATA COLLECTION PROGRAM

0058	00033	063073R	LDA N	
0059	00034	105120	FLT	
0060	00035	105040	FMP	-F1.302083 COMPUTE # BLOCKS PER PING
0061	00036	061240R	FIX	
0062	00037	105100	INA	ROUND UP
0063	00040	002004	CMA, INA	
0064	00041	003004	STA BLOCK	
0065*	00042	073104R		
0066	00043	063242R	LDA -D2	INITIALIZE HEAD, TRACK, SECTOR
0067	00044	072656R	STA HEADB	HEAD 2 IS FIRST LOWER HEAD
0068	00045	062007X	LDA FTLD	
0069	00046	072653R	STA TRAK	
0070	00047	062010X	LDA LTLD	
0071	00050	072655R	STA MXTRK	
0072	00051	002002	SZA	TEST LTLD
0073	00052	026065R	JMP LDOK	NON-ZERO, LOWER DISC HAS SPACE
0074	00053	002400	CLA	ZERO, SET UP UPPER DISC
0075	00054	072656R	STA HEADB	HEAD 0 IS FIRST UPPER HEAD
0076	00055	062011X	LDA FTUD	
0077	00056	072653R	STA TRAK	
0078	00057	062012X	LDA LTUD	
0079	00060	072655R	STA MXTRK	
0080	00061	002003	SZA, RSS	
0081	00062	027063R	JMP ERR12	ERROR, NO DISC SPACE ALLOCATED
0082	00063	002400	CLA	
0083	00064	072012X	STA LTUD	LTUD=0 INDICATES NOW USING TOP

0084	00065	002400	LDOK	CLA			
0085	00066	072654R		STA	SECT		
0086*							
0087	00067	002400	SPING	CLA			DISABLE INT. FROM CORE LOADING
0088	00070	070006		STA	6		
0089*							
0090	00071	063104R		LDA	BLOCK		
0091	00072	073105R		STA	BCTR		
0092*							
0093	00073	017031R		JSB	SUWAIT		WAIT FOR S INTERRUPT
0094*							
0095	00074	063075R		LDA	K		
0096	00075	043243R		ADA	-D-1		SUBTRACT 1 FROM DELAY
0097	00076	100200		MPY	-D10		
	00077	061244R					
0098	00100	073106R		STA	TEMP		
0099	00101	016377R		JSB	STDLY		START DELAY OF K-1 SECONDS
0100	00102	000104R		DEF	X+2		
0101	00103	001106R		DEF	TEMP		
0102*							
0103	00104	016444R		JSB	DELAY		WAIT UNTIL K DELAY IS OVER
0104*							
0105	00105	063244R		LDA	-D10		
0106	00106	073106R		STA	TEMP		
0107	00107	016377R		JSB	STDLY		START DELAY OF 1 SECOND
0108	00110	000112R		DEF	X+2		
0109	00111	001106R		DEF	TEMP		
0110*							
0111	00112	016772R		JSB	TDIN		GET TIME FROM TIME INTF

PAGE 0004 *01 OLARS DATA COLLECTION PROGRAM

0112	00113	000115R	DEF X+2	ADDR IN DISC BUFFER FOR TIME
0113	00114	001240R	DEF OBUF	
0114*				
0115	00115	016736R	JSB BDIN	INPUT BUOY DATA (72 WORDS)
0116	00116	000120R	DEF X+2	
0117	00117	001130R	DEF DTA72	
0118	00120	063216R	LDA DTA72+54	GET COMPASS WORD
0119	00121	013245R	AND -B177400	
0120*				
0121	00122	070001	STA B	CONVERT GRAY TO BINARY
0122	00123	063246R	LDA -D-8	
0123	00124	073110R	STA CNTR	
0124	00125	002400	CLA	
0125	00126	002011	SLA, RSS	A ODD?
0126	00127	026133R	JMP EVEN	NO
0127	00130	006021	SSB, RSS	B NEGATIVE?
0128	00131	033247R	IOR -B100000	NO
0129	00132	026135R	JMP GB2	
0130	00133	006020	SSB	B NEGATIVE?
0131	00134	033247R	IOR -B100000	YES
0132	00135	001200	RAL	
0133	00136	005200	RBL	
0134	00137	037110R	ISZ CNTR	DONE
0135	00140	026126R	JMP GB1	NO, LOOP
0136	00141	105120	FLT	YES, RESULT IN A
0137*				
0138	00142	105040	FMP -F1.40625	CONVERT TO DEGREES
	00143	061250R		
0139	00144	105000	FAD -F0.5	
	00145	061252R		
0140	00146	105100	FIX	

STORE: THIS IS HEADING OF BEAM 1

0141	00147	073111R	STA ANGL1	
0142	00150	073242R	STA OBUF+2	
0143*				
0144	00151	063074R	LDA L	COMPUTE # OF 1ST BEAM TO
0145	00152	043243R	ADA -D-1	BE STORED
0146	00153	105120	FLT	
0147	00154	105060	FDU -F2.0	(L-1)/2
	00155	061254R		
0148	00156	104400	DST TEMP	
	00157	001106R		
0149	00160	063111R	LDA ANGL1	
0150	00161	003004	CMA,INA	
0151	00162	043077R	ADA D	
0152	00163	002020	SSA	NEGATIVE?
0153	00164	043256R	ADA -D360	YES, ADD 1 REV
0154	00165	105120	FLT	
0155	00166	105060	FDU -F22.5	
	00167	061257R		
0156	00170	105000	FAD -F1.0	(D-ANGL1)/22.5 + 1
	00171	061261R		
0157	00172	105020	FSB TEMP	-(L-1)/2
	00173	001106R		
0158	00174	105000	FAD -F0.5	ROUND
	00175	061252R		
0159	00176	105100	FIX	THIS IS FIRST BEAM NUMBER

0160	00177	002003	SZA, RSS	BEAM ZERO?
0161	00200	043263R	ADA -D16	YES, MAKE POSITIVE
0162	00201	002020	SSA	NEGATIVE?
0163	00202	043263R	ADA -D16	YES, MAKE POSITIVE
0164	00203	043264R	ADA -D-17	
0165	00204	002021	SSA, RSS	>16 ?
0166	00205	043265R	ADA -D-16	YES, SUBTRACT 1 REV
0167	00206	043266R	ADA -D17	RESTORE BEAM NUMBER
0168	00207	073112R	STA FSTB#	FIRST BEAM TO BE STORED
0169	00210	073244R	STA OBUF+4	STORE IN BUFFER
0170*				
0171	00211	043243R	ADA -D-1	GET BEAM NUMBER DIFFERENCE
0172	00212	100200	MPY -D45	X 22.5
0173	00213	061267R		
0174	00214	043111R	ADA ANGL1	+ ANGL1
0175	00215	043111R	ADA ANGL1	(X2 DUE TO SCALE FACTOR)
0176	00216	002004	INA	ROUND
0177	00217	000065	CLE, ERA	RESCALE
0178	00220	073245R	STA OBUF+5	POSITIVE ANGLE OF FSTB#
0179	00221	043270R	ADA -D-360	
0180	00222	002021	SSA, RSS	>360 ?
0181*	00223	073245R	STA OBUF+5	YES, STORE 1 REV LESS
0182	00224	063103R	LDA PING	
0183	00225	073246R	STA OBUF+6	PING # TO BUFFER
0184	00226	063074R	LDA L	
0185	00227	073247R	STA OBUF+7	# BEAMS
0186	00230	063100R	LDA S	
0187	00231	073250R	STA OBUF+8	SOURCE
0188	00232	063076R	LDA M	
0189	00233	073251R	STA OBUF+9	# PINGS
0190	00234	063075R	LDA K	
0191	00235	073252R	STA OBUF+10	DELAY
0192	00236	063073R	LDA N	
0193	00237	073253R	STA OBUF+11	TOTAL PULSE TIME

0194	00240	063104R	LDA BLOCK	# BLOCKS PER PING
0195	00241	003004	CMA, INA	
0196	00242	073254R	STA OBUF+12	
0197*				
0198	00243	063126R	LDA BUF1A	SWITCH BUFFERS
0199	00244	067127R	LDB BUF2A	
0200	00245	073127R	STA BUF2A	
0201	00246	077126R	STB BUF1A	
0202*				
0203	00247	016444R	JSB DELAY	WAIT UNTIL K DELAY OVER
0204*				
0205	00250	063126R	LDA BUF1A	
0206	00251	072254R	STA INBUF	
0207	00252	016666R	JSB BFIN	START INPUT OF BF DATA
0208	00253	000257R	DEF X+4	
0209	00254	000000	INBUF NOP	
0210	00255	020000	DEC 8192	
0211	00256	000275R	DEF BFDUN	INPUT BUFFER ADDRESS
0212*				
0213	00257	103100	CLF 0	
0214*				
				COMPLETION ADDRESS
				INTERRUPTS OFF

PAGE 0006 #01 OLAAS DATA COLLECTION PROGRAM

0215	00260	016462R	JSB WDISC	OUTPUT BUOY DATA BLOCK
0216	00261	000264R	DEF X+3	
0217	00262	001240R	DEF OBUF	
0218	00263	000200	DEC 128	

0219*				STF 0		INTERRUPTS ON
0220	00264	102100	BFDLY	LDA -D-3		ALLOW 1.2 SEC FOR INTERRUPT
0221	00265	063271R		STA CNTR		
0222	00266	073110R	DLY1	CLB		
0223	00267	006400	DLY2	ISZ B		0.3211264 SEC PER LOOP
0224	00270	034001		JMP DLY2		
0225	00271	026270R		ISZ CNTR		
0226	00272	037110R		JMP DLY1		
0227	00273	026267R		JMP ERR02		NO BF DONE INTERRUPT, ERROR
0228	00274	027043R				
0229*						
0230	00275	063126R	BFDUN	LDA BUF1A		SET INPUT BUFFER FOR BFIN
0231	00276	067127R		LDB BUF2A		INCR BCTR -07
0232	00277	073127R		STA BUF2A		NO
0233	00300	077126R		STB BUF1A		YES, STOP BF DATA
0234	00301	076310R		STB IBUFA		GO FORMAT FINAL DATA
0235	00302	037105R		ISZ BCTR		START INPUT OF BF DATA
0236	00303	026306R		JMP BFMOR		
0237	00304	016723R		JSB BFHLT		
0238	00305	026313R		JMP FRMAT		
0239	00306	016666R	BFMOR	JSB BFIN		
0240	00307	000313R		DEF X+4		
0241	00310	000000	IBUFA	NOP		
0242	00311	020000		DEC 8192		
0243	00312	000275R		DEF BFDUN		COMPLETION ADDRESS
0244*						
0245	00313	103100	FRMAT	CLF 0		INTERRUPTS OFF
0246	00314	063112R		LDA FSTB#		SELECT AND FORMAT DATA
0247	00315	073113R		STA FMTB		
0248	00316	063074R		LDA L		
0249	00317	003004		CMA, INA		
0250	00320	073114R		STA FBCTR		
0251	00321	063125R		LDA OBUFA		
0252	00322	073117R		STA PTR2		
0253	00323	063127R	FRM1	LDA BUF2A		

0254	00324	043113R	ADA FMTB
0255	00325	043243R	ADA -D-1
0256	00326	073116R	STA PTR1
0257	00327	063272R	LDA -D-512
0258	00330	073110R	STA CNTR
0259	00331	163116R	LDA PTR1, I
0260	00332	173117R	STA PTR2, I
0261	00333	063116R	LDA PTR1
0262	00334	043263R	ADA -D16
0263	00335	073116R	STA PTR1
0264	00336	037117R	ISZ PTR2
0265	00337	037110R	ISZ CNTR
0266	00340	026331R	JMP FRM2
0267	00341	063113R	LDA FMTB
0268	00342	002004	INA
0269	00343	053266R	CPA -D17
0270	00344	002404	CLA, INA

FRM2

INCREMENT BEAM
177
YES, SET TO 1

PAGE 0007 #01 OLAAS DATA COLLECTION PROGRAM

0271	00345	073113R	STA FMTB	ALL REQ'D BEAMS DONE?
0272	00346	037114R	ISZ FBCTR	NO
0273	00347	026323R	JMP FRM1	
0274*				
0275	00350	063074R	LDA L	YES, COMPUTE # WORDS JUST
0276	00351	100200	MPY -D512	STORED IN OBUF
	00352	061273R		
0277	00353	072357R	STA DCT	

ADDRESS	DATA	INSTR	COMMENT
0278*			
0279	00354 016462R	JSB WDISC	WRITE BF DATA ON DISC
0280	00355 000360R	DEF X+3	
0281	00356 001240R	DEF OBUF	
0282	00357 000000	NOP	(DISC WORD COUNT)
0283*			
0284	00360 102206	SFC 6	BF COMPL. INT. WAITING?
0285	00361 027045R	JMP ERR03	YES, ERROR
0286	00362 102100	STF 0	INTERRUPTS ON
0287*			
0288	00363 063105R	LDA BCTR	
0289	00364 002002	SZA	
0290	00365 026265R	JMP BFDLY	
0291	00366 037103R	ISZ PING	
0292*			
0293*			
0294*			
0295*			
0296*			
0297*			
0298	00367 016005X	JSB CDUMP	SWAP OUT THIS CORE LOAD
0299	00370 000372R	DEF X+2	
0300	00371 001124R	DEF CYLCO	
0301	00372 016006X	JSB CLOAD	BRING IN NEXT CORE LOAD
0302	00373 000376R	DEF X+3	
0303	00374 001123R	DEF CYLAN	
0304	00375 001121R	DEF XFER	
0305*			
0306*			
0307*			
0308	00376 000000	DTIM	
0309	00377 000000	STDLY	
0310	00400 016001X	JSB .ENTR	
0311	00401 000376R	DEF DTIM	
0312	00402 162376R	LDA DTIM,I	SET COUNT

0313	00403	003004	CMA, INA	
0314	00404	072454R	STA TCTR	
0315	00405	062003X	LDA LKTBG	SET UP TRAP CELL
0316	00406	070013	STA TBG	
0317	00407	063274R	LDA -D3	SET TBG TO 100 MS INTERVAL
0318	00410	102613	OTA TBG	
0319	00411	103713	STC TBG,C	
0320	00412	102100	STF 0	INTERRUPTS ON
0321	00413	126377R	JMP STDLY,I	RETURN
0322*				
0323	00414	000000	TBGI	TBG INTERRUPT PROCESSOR
0324	00415	103100	CLF 0	INTERRUPTS OFF
0325	00416	072455R	STA SAVEA	

PAGE 0008 #01 OLAAS DATA COLLECTION PROGRAM

0326	00417	076456R	STB SAVEB	
0327	00420	001520	ERA, ALS	
0328	00421	102201	SOC	
0329	00422	002004	INA	
0330	00423	072457R	STA SAVEO	
0331	00424	102513	LIA TBG	READ TBG STATUS
0332	00425	002002	SZA	
0333	00426	027047R	JMP ERR04	NON-ZERO: ERROR
0334	00427	036454R	ISZ TCTR	
0335	00430	026442R	JMP TNZRO	
0336	00431	106713	CLC TBG	TCTR=0, STOP CLOCK
0337	00432	062457R	TBGIX LDA SAVEO	RESTORE REGISTERS

0338	00433	103101	CLO		
0339	00434	000036	SLA,ELA		
0340	00435	102101	STO		
0341	00436	062455R	LDA SAVEA		
0342	00437	066456R	LDB SAVEB		
0343	00440	102100	STF 0		
0344	00441	126414R	JMP TBGI,I		
0345	00442	103113	CLF TBG		
0346	00443	026432R	JMP TBGIX		
0347*					
0348	00444	000000	DELAY		
0349	00445	062454R	LDA TCTR		
0350	00446	002003	SZA,RSS		
0351	00447	027051R	JMP ERR05		
0352	00450	062454R	LDA TCTR		
0353	00451	002002	SZA		
0354	00452	026450R	JMP DELA1		
0355	00453	126444R	JMP DELAY,I		
0356*					
0357	00454	000000	TCTR		
0358	00455	000000	SAVEA		
0359	00456	000000	SAVEB		
0360	00457	000000	SAVEO		
0361*					
0362*			DISC SUBROUTINES		
0363*					
0364	00460	000000	WPTR		
0365	00461	000000	WCTR		
0366	00462	000000	WDISC		
0367	00463	016001X	JSB .ENTR		
0368	00464	000460R	DEF WPTR		
0369*					
0370	00465	062654R	WDIS1		
0371	00466	100200	LDA SECT		
	00467	061275R	MPY -D128		

INTERRUPTS ON

RETURN

CLEAR FLAG TO WAIT FOR NEXT INT.

ERROR, TIME OVER AT ENTRY

LOOP UNTIL TCTR=0
RETURN

0372 00470 003004
 0373 00471 043276R
 0374 00472 073120R
 0375 00473 066461R
 0376 00474 003004
 0377 00475 042461R
 0378 00476 002020
 0379 00477 077120R
 0380 00500 016547R

CMA, INA
 ADA -D6144
 STA AA
 LDB WCTR
 CMA, INA
 ADA WCTR
 SSA
 STB AA
 JSB WRITD

AA - 6144 - SECTx128
 # WORDS REMAINING IN TRACK

AA .GT. WCTR?
 YES, SUBSTITUTE WCTR FOR AA
 WRITE AA WORDS

PAGE 0009 #01 OLAAS DATA COLLECTION PROGRAM

0381x
 0382 00501 006400
 0383 00502 063120R
 0384 00503 100400
 00504 061275R
 0385 00505 006002
 0386 00506 027053R
 0387 00507 042654R
 0388 00510 072654R
 0389x
 0390 00511 053277R
 0391 00512 002001
 0392 00513 026535R
 0393 00514 002400
 0394 00515 072654R
 0395 00516 036653R

CLB
 LDA AA
 DIV -D128
 SZB
 JMP ERR06
 ADA SECT
 STA SECT
 CPA -D48
 RSS
 JMP WDIS2
 CLA
 STA SECT
 ISZ TRAK

UPDATE SECTOR
 THIS ROUTINE USES AN ARTIFICIAL
 SECTOR IN RANGE 0-47

ERROR, LENGTH NOT MULT. OF 128
 SECT - SECT + AA/128

SECTOR NOT 48, SKIP TRACK UPDATE

0396	00517	062653R	LDA TRAK	TRAK .GT. MXTRK?
0397	00520	003004	CMA, INA	NO
0398	00521	042655R	ADA MXTRK	YES, GO TO SECOND DISC
0399	00522	002021	SSA, RSS	
0400	00523	026535R	JMP WDIS2	
0401	00524	062011X	LDA FTUD	
0402	00525	072653R	STA TRAK	
0403	00526	062012X	LDA LTUD	
0404	00527	072655R	STA MXTRK	
0405	00530	002003	SZA, RSS	
0406	00531	027055R	JMP ERR07	ERROR, STORAGE EXCEEDED
0407	00532	002400	CLA	
0408	00533	072012X	STA LTUD	LTUD=0 INDICATES NOW USING TOP
0409	00534	072656R	STA HEADB	HEAD BASE - 1ST UPPR DISC HEAD
0410*				UPDATE POINTER AND COUNTER
0411	00535	063120R	LDA AA	
0412	00536	042460R	ADA WPTR	
0413	00537	072460R	STA WPTR	
0414	00540	063120R	LDA AA	
0415	00541	003004	CMA, INA	
0416	00542	042461R	ADA WCTR	
0417	00543	072461R	STA WCTR	
0418	00544	002002	SZA	WCTR=0?
0419	00545	026465R	JMP WDIS1	NO, GO DO ANOTHER WRITE
0420	00546	126462R	JMP WDISC, I	RETURN
0421*				
0422	00547	000000	NOP	
0423	00550	062654R	LDA SECT	CONVERT SECT TO HEAD AND SECTOR
0424	00551	070001	STA B	
0425	00552	043300R	ADA -D-24	
0426	00553	002020	SSA	
0427	00554	026561R	JMP FRSTH	
0428	00555	070001	STA B	
0429	00556	062656R	LDA HEADB	
0430	00557	002004	INA	

0431	00560	002001	RSS
0432	00561	062656R	FRSTH LDA HEADB
0433	00562	001727	ALF,ALF
0434	00563	030001	IOR B
0435	00564	072657R	STA HDSEC

PAGE 0010 #01 OLAAS DATA COLLECTION PROGRAM

0436	00565	016615R	JSB SEEK
0437	00566	106717	CLC DISCD
0438	00567	106720	CLC DISCC
0439	00570	063301R	LDA -B120000 SET UP DMA
0440	00571	032662R	IOR SELDD
0441	00572	102607	OTA 7
0442	00573	106703	CLC 3
0443	00574	062460R	LDA WPTR
0444	00575	102603	OTA 3
0445	00576	102703	STC 3
0446	00577	063120R	LDA AA
0447	00600	003004	CMA,INA
0448	00601	102603	OTA 3
0449	00602	102717	STC DISCD
0450	00603	102117	STF DISCD
0451	00604	103707	STC 7,C
0452	00605	062660R	LDA WRCMD
0453	00606	102620	OTA DISCC
0454	00607	103720	STC DISCC,C
0455	00610	102320	SFS DISCC

WORDS TO WRITE

WAIT FOR COMPLETION

Address	Hex	Assembly	Comments
0456	00611	JMP X-1	
0457	00612	CLC 7	STOP DMA
0458	00613	JSB STATS	CHECK STATUS
0459	00614	JMP WRITD, I	RETURN
0460*			
0461	00615	NOP	
0462	00616	CLC DISCD	
0463	00617	CLC DISCC	
0464	00620	LDA TRAK	
0465	00621	OTA DISCD	
0466	00622	STC DISCD, C	
0467	00623	LDA SKCMD	
0468	00624	OTA DISCC	
0469	00625	STC DISCC, C	
0470	00626	SFS DISCD	
0471	00627	JMP X-1	
0472	00630	LDA HDSEC	
0473	00631	OTA DISCD	
0474	00632	STC DISCD, C	
0475	00633	SFS DISCC	
0476	00634	JMP X-1	
0477	00635	JSB STATS	
0478	00636	JMP SEEK, I	
0479*			
0480	00637	NOP	
0481	00640	CLC DISCD	
0482	00641	CLC DISCC	
0483	00642	CLA	STATUS COMMAND
0484	00643	OTA DISCC	
0485	00644	STC DISCC, C	
0486	00645	SFS DISCD	
0487	00646	JMP X-1	
0488	00647	LIA DISCD	
0489	00650	SLA	ERROR, DISC STATUS BAD
0490	00651	JMP ERR10	

0515	00676	027061R	JMP ERR11	REMAINDER NOT 0, ERROR
0516	00677	062004X	LDA LINK6	SET UP DMA INT. LINK
0517	00700	070006	STA 6	
0518	00701	063122R	LDA HLT30	SET UP HALT IN BF S.C.
0519	00702	070022	STA BF	
0520*				
0521	00703	063301R	LDA -B120000	SET UP DMA
0522	00704	032734R	IOR SELBF	
0523	00705	102606	OTA 6	
0524	00706	106702	CLC 2	
0525	00707	062663R	LDA BDATA	
0526	00710	033247R	IOR -B100000	
0527	00711	102602	OTA 2	
0528	00712	102702	STC 2	
0529	00713	062664R	LDA BFCT	
0530	00714	003004	CMA, INA	
0531	00715	102602	OTA 2	
0532	00716	103722	STC BF,C	ENABLE BEAM FORMER INTERRUPT
0533	00717	103706	STC 6,C	START DMA
0534	00720	002404	CLA, INA	
0535	00721	102622	OTA BF	START BEAM FORMER
0536	00722	126666R	JMP BFIN,I	RETURN
0537*				
0538	00723	000000	BFHLT NOP	STOP BEAM FORMER
0539	00724	002400	CLA	
0540	00725	102622	OTA BF	
0541	00726	106722	CLC BF	
0542	00727	126723R	JMP BFHLT,I	RETURN
0543*				
0544	00730	000000	BFIT	BEAM FORMER DMA COMPLETION PROCSR
0545	00731	103100	CLF 0	INTERRUPTS OFF
0546	00732	107706	CLC 6,C	STOP DMA

PAGE 0012 #01 OLAAS DATA COLLECTION PROGRAM

0547	00733	126665R	JMP COMPL,I	GO TO ADDR SPECIFIED IN BFIN CALL
0548*				
0549	00734	000022	SELBF DEF BF	
0550*				
0551*	BUOY DATA DRIVER			
0552*				
0553	00735	000000	BDLOC NOP	ADDRESS
0554	00736	000000	BDIN NOP	
0555	00737	016001X	JSB .ENTR	
0556	00740	000735R	DEF BDLOC	
0557	00741	103100	CLF 0	INTERRUPTS OFF - SHOULD TAKE 6 MS
0558	00742	063301R	LDA -B120000	SET UP DMA
0559	00743	032770R	IOR SELBD	
0560	00744	102606	OTA 6	
0561	00745	106702	CLC 2	
0562	00746	062735R	LDA BDLOC	
0563	00747	033247R	IOR -B100000	
0564	00750	102602	OTA 2	
0565	00751	102702	STC 2	
0566	00752	063302R	LDA -D-72	
0567	00753	102602	OTA 2	ENABLE BUOY DATA INTERFACE
0568	00754	103723	STC BD,C	START DMA
0569	00755	103706	STC 6,C	
0570	00756	002404	CLA,INA	START EXTERNAL DEVICE
0571	00757	102623	OTA BD	WAIT FOR COMPLETION
0572	00760	102306	SFS 6	
0573	00761	026760R	JMP X-1	
0574	00762	002400	CLA	STOP INTERFACE

0575	00763	102623	OTA BD		
0576	00764	106723	CLC BD		
0577	00765	106706	CLC 6		
0578	00766	102100	STF 0		STOP DMA
0579	00767	126736R	JMP BDIN,I		INTERRUPTS ON
0580	00770	000023	DEF BD		RETURN
0581*			SELBD		
0582*			CLOCK DRIVER		
0583*					
0584	00771	000000	TADD	NOP	
0585	00772	000000	TDIN	NOP	ADDRESS FOR TIME (2 WORDS)
0586	00773	016001X		JSB .ENTR	
0587	00774	000771R		DEF TADD	SET CNTR FOR MAX 4 READS
0588	00775	063271R		LDA -D-3	
0589	00776	073110R		STA CNTR	INPUT BOTH WORDS TO TEMP
0590	00777	017017R		JSB TIN	
0591	01000	104400		DST TEMP	
0592	01001	001106R		JSB TIN	INPUT BOTH WORDS TO CALLER'S LOC
0593	01002	017017R	TDIN1	DST TADD,I	
	01003	104400			
	01004	100771R		CPA TEMP	WORD 1 SAME AS PREVIOUS?
0594	01005	053106R		RSS	
0595	01006	002001		JMP TDIN2	NO
0596	01007	027012R		CPB TEMP+1	WORD 2 SAME AS PREVIOUS?
0597	01010	057107R		JMP TDIN,I	YES, RETURN
0598	01011	126772R		DST TEMP	NO, STORE NEW IN OLD
0599	01012	104400	TDIN2		
	01013	001106R			

PAGE 0013 #01 OLAAS DATA COLLECTION PROGRAM

0600	01014	037110R	ISZ CNTR	INCREMENT COUNTER
0601	01015	027002R	JMP TDIN1	LT 4 SAMPLES, TRY AGAIN
0602	01016	027065R	JMP ERR13	ERROR, NO 2 SAMPLES ALIKE IN 4
0603	01017	000000	NOP	INPUT SUBROUTINE
0604	01020	006404	CLB, INB	GATE HOURS
0605	01021	106624	OTB TIME	
0606	01022	102524	LIA TIME	
0607	01023	067242R	LDB -D2	
0608	01024	106624	OTB TIME	GATE MIN, SEC
0609	01025	106524	LIB TIME	
0610	01026	003000	CMA	
0611	01027	007000	CMB	
0612	01030	127017R	JMP TIN, I	RETURN TO TDIN
0613*				
0614*	S	INTERRUPT DRIVER (WAIT FOR INTERRUPT AND RETURN)		
0615*				
0616	01031	000000	SWAIT NOP	INTERRUPTS OFF
0617	01032	103100	CLF 0	
0618	01033	103724	STC TIME, C	WAIT FOR INTERRUPT
0619	01034	102324	SFS TIME	
0620	01035	027034R	JMP X-1	TURN OFF INTERFACE
0621	01036	106724	CLC TIME	INTERRUPTS ON
0622	01037	102100	STF 0	
0623	01040	127031R	JMP SWAIT, I	RETURN
0624*				
0625	01041	102001	ERR01 HLT 1	
0626	01042	027041R	JMP X-1	
0627	01043	102002	ERR02 HLT 2	
0628	01044	027043R	JMP X-1	
0629	01045	102003	ERR03 HLT 3	
0630	01046	027045R	JMP X-1	
0631	01047	102004	ERR04 HLT 4	

0632	01050	027047R	ERR05	JMP X-1
0633	01051	102005	HLT 5	
0634	01052	027051R	JMP X-1	
0635	01053	102006	HLT 6	
0636	01054	027053R	JMP X-1	
0637	01055	102007	HLT 7	
0638	01056	027055R	JMP X-1	
0639	01057	102010	HLT 10B	
0640	01060	027057R	JMP X-1	
0641	01061	102011	HLT 11B	
0642	01062	027061R	JMP X-1	
0643	01063	102012	HLT 12B	
0644	01064	027063R	JMP X-1	
0645	01065	102013	HLT 13B	
0646	01066	027065R	JMP X-1	
0647	01067	102014	HLT 14B	
0648	01070	027067R	JMP X-1	
0649	01071	102015	HLT 15B	
0650	01072	027071R	JMP X-1	
0651*				
0652	01073	000000	N	NOP
0653	01074	000000	L	NOP
0654	01075	000000	K	NOP
0655	01076	000000	M	NOP

PAGE 0014 #01 OLAAS DATA COLLECTION PROGRAM

0656 01077 000000 D NOP

0657	01100	000000	S	NOP	51	
0658	01101	000000	#BMS	NOP	30B	
0659	01102	000000	TOTPG	NOP	23	
0660	01103	000000	PING	NOP	31	
0661	01104	000000	BLOCK	NOP	OBLFA DEF OBUF	
0662	01105	000000	BCTR	NOP	BUF1A DEF BUF1	
0663	01106	000000	TEMP	BSS	2	BUF2A DEF BUF2
0664	01110	000000	CNTR	NOP	72	BSS 8192
0665	01111	000000	ANGL1	NOP	17B	BSS 8192
0666	01112	000000	FSTB#	NOP	20B	BSS 8192
0667	01113	000000	FMTB	NOP	13B	BSS 8192
0668	01114	000000	FBCTR	NOP	22B	DISC EQU 17B
0669	01115	000000	NFTSJ	NOP	23B	DISC EQU 20B
0670	01116	000000	PTR1	NOP	24B	TBG EQU 13B
0671	01117	000000	PTR2	NOP	0	BF EQU 22B
0672	01120	000000	AA	NOP	1	BD EQU 23B
0673	01121	000051	XFER	OCT		TIME EQU 24B
0674	01122	102030	HLT30	HLT		A EQU 0
0675	01123	000023	CYLAN	OCT		B EQU 1
0676	01124	000031	CYLC0	OCT		
0677	01125	001240R	OBLFA	DEF		
0678	01126	021240R	BUF1A	DEF		
0679	01127	041240R	BUF2A	DEF		
0680	01130	000000	DTA72	BSS		
0681	01240	000000	OBUF	BSS		
0682	21240	000000	BUF1	BSS		
0683	41240	000000	BUF2	BSS		
0684	00017		DISCD	EQU		
0685	00020		DISCC	EQU		
0686	00013		TBG	EQU		
0687	00022		BF	EQU		
0688	00023		BD	EQU		
0689	00024		TIME	EQU		
0690	00000		A	EQU		
0691	00001		B	EQU		

CYL NO. OF ANALYSIS SWAP AREA
CYL NO. OF COLLECTION SWAP AREA

(AVIONICS)

61240 051525
61241 052002
61242 000002
61243 177777
61244 000012
61245 177400
61246 177770
61247 100000
61250 055000
61251 000002
61252 040000
61253 000000
61254 040000
61255 000004
61256 000550
61257 055000
61260 000012
61261 040000
61262 000002
61263 000020

PAGE 0015 #01 OLAAS DATA COLLECTION PROGRAM

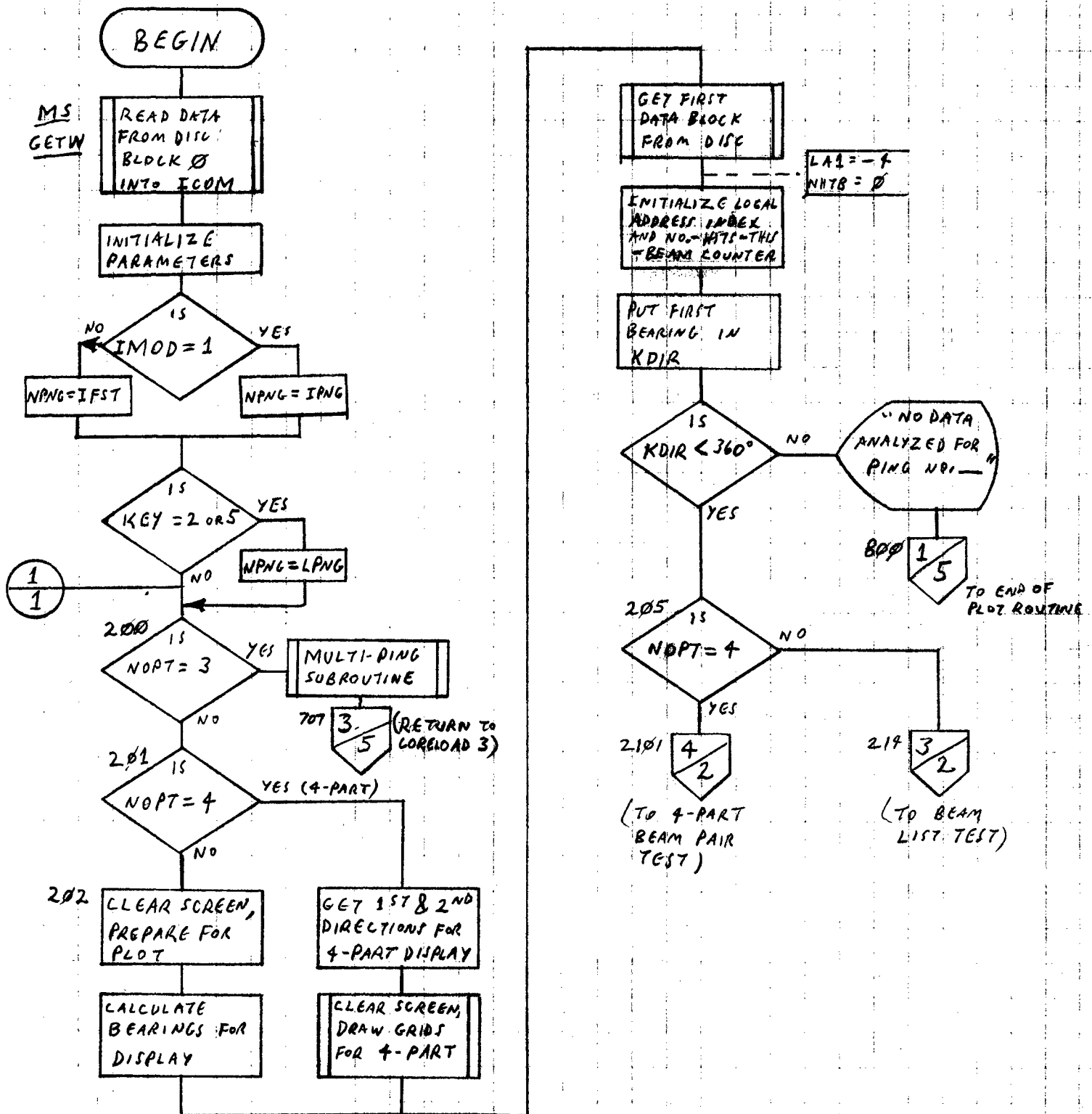
61264 177757
61265 177760
61266 000021
61267 000055
61270 177230

61271 177775
61272 177000
61273 001000
61274 000003
61275 000200
61276 014000
61277 000060
61300 177750
61301 120000
61302 177670

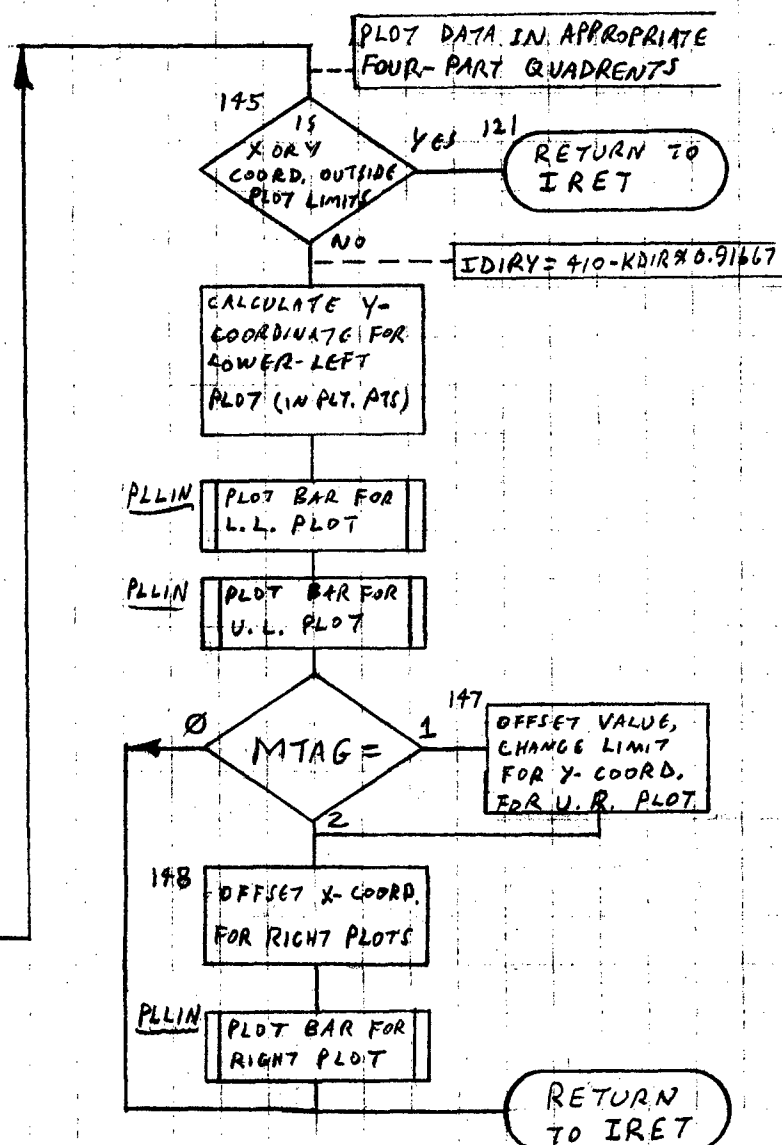
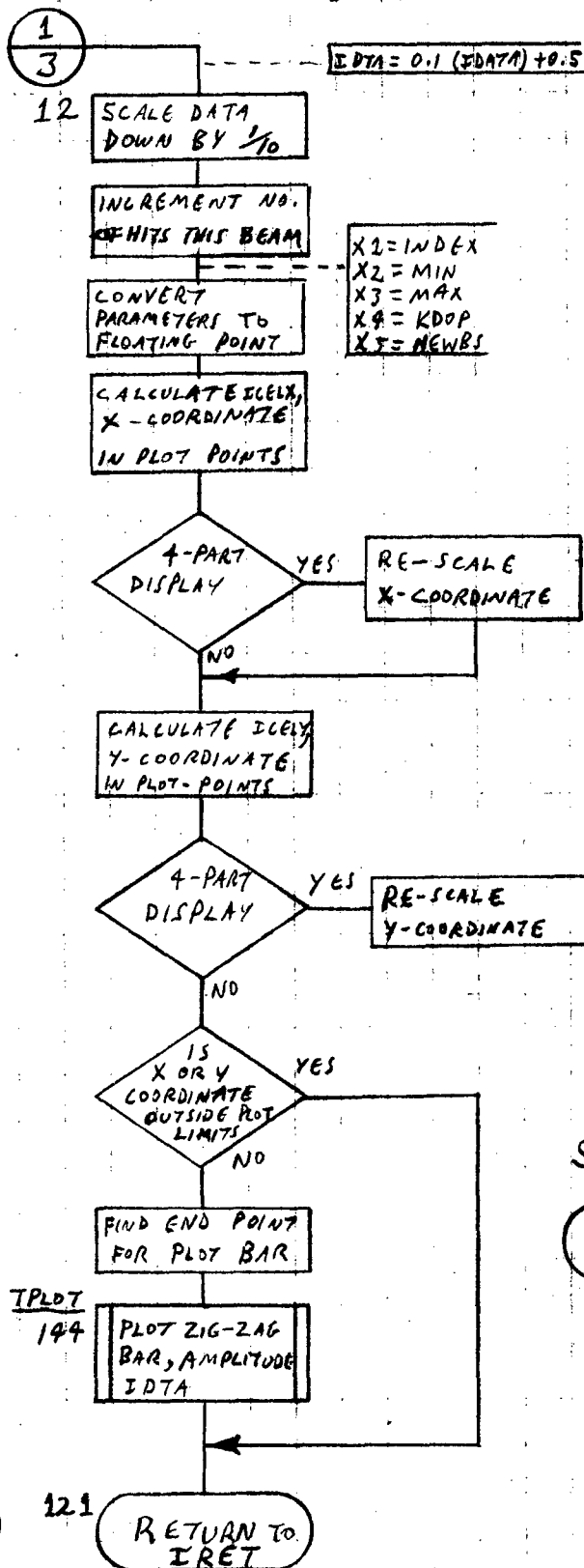
0692

END Y0001
** NO ERRORS *TOTAL *XRTE ASMB 750420**

Y1007 PLOT ROUTINE

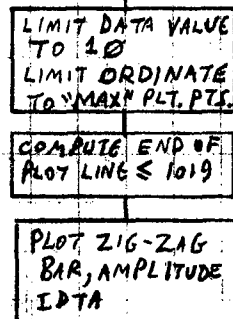


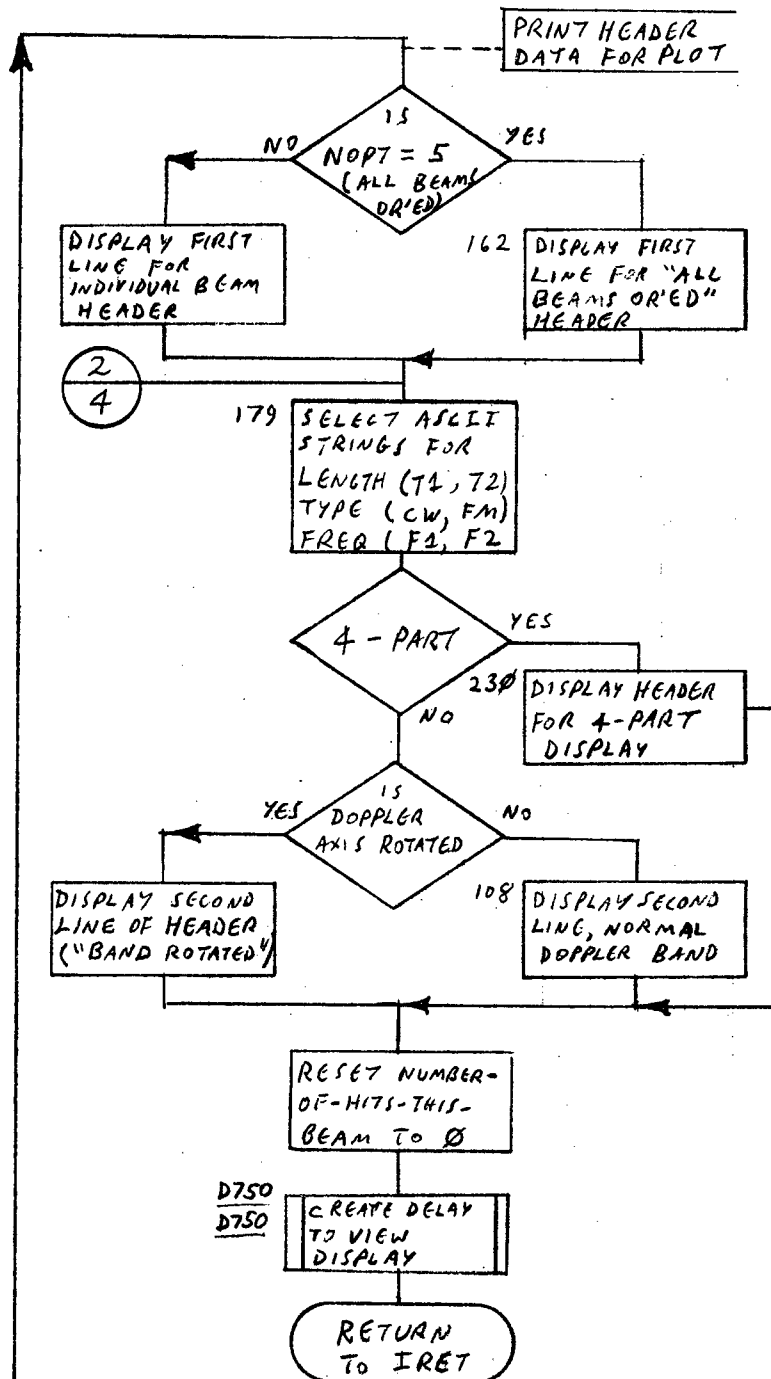
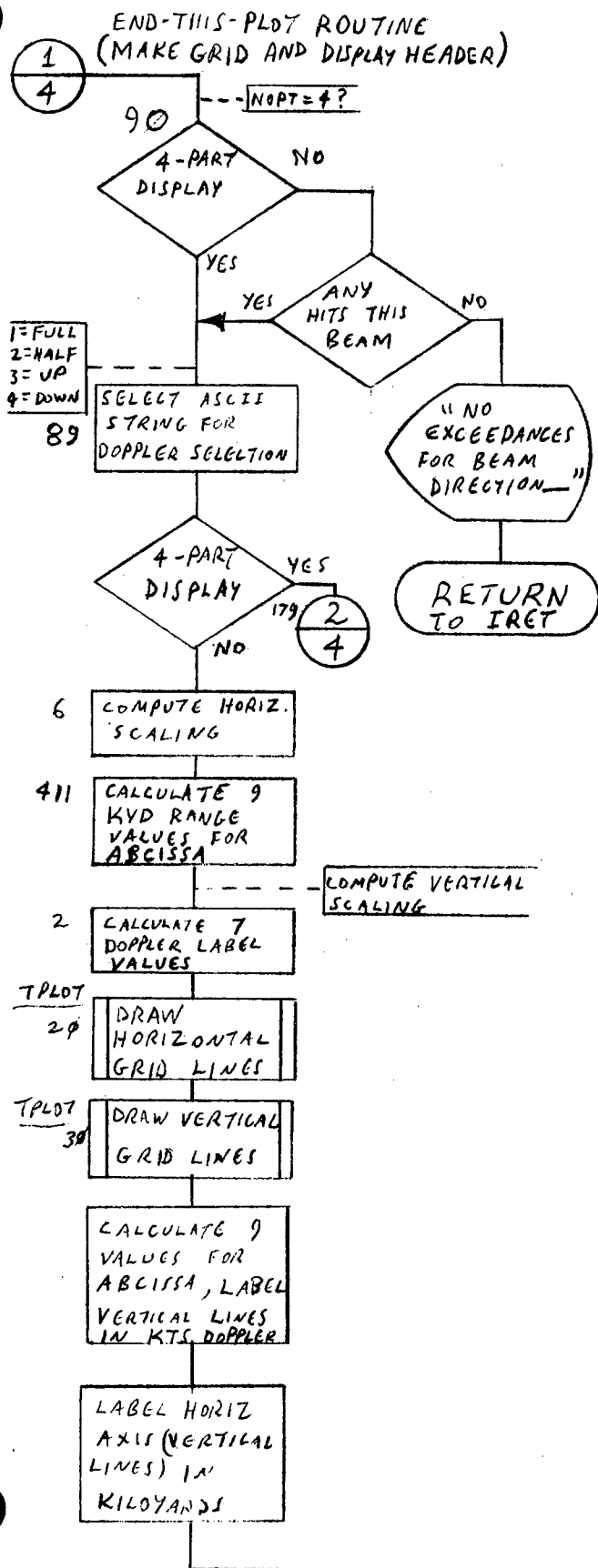
ROUTING TO PLOT A DATA POINT

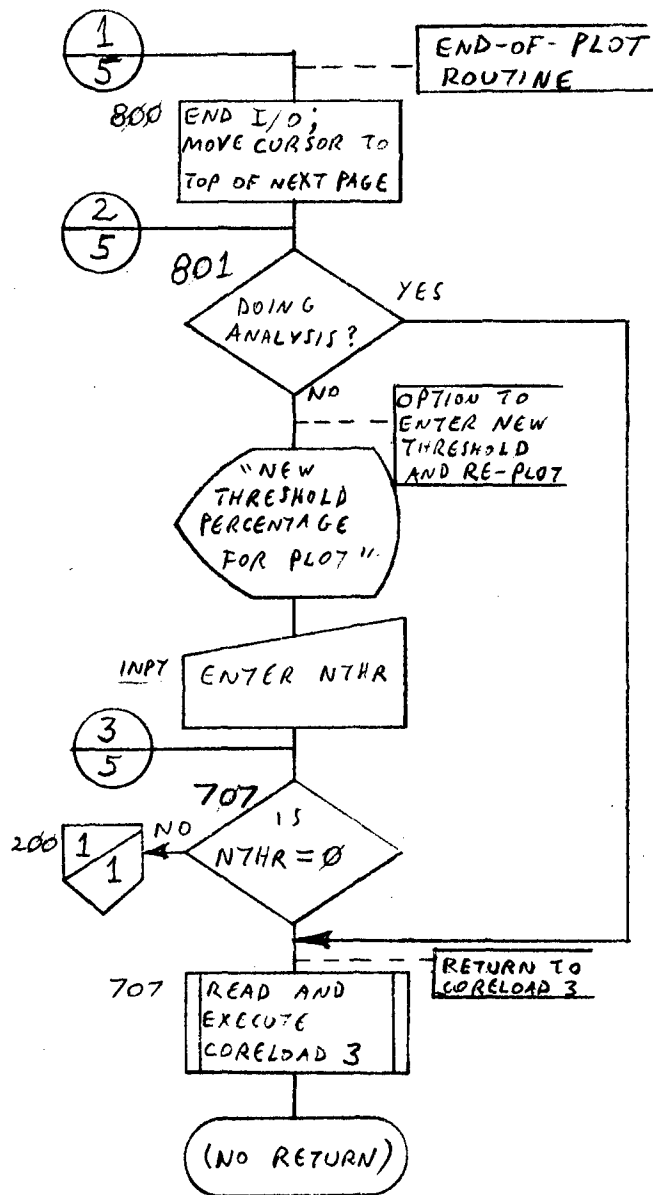


SUBROUTINE PLLIN (IX, INCRX, IY, IDTA, MAX)

ENTER

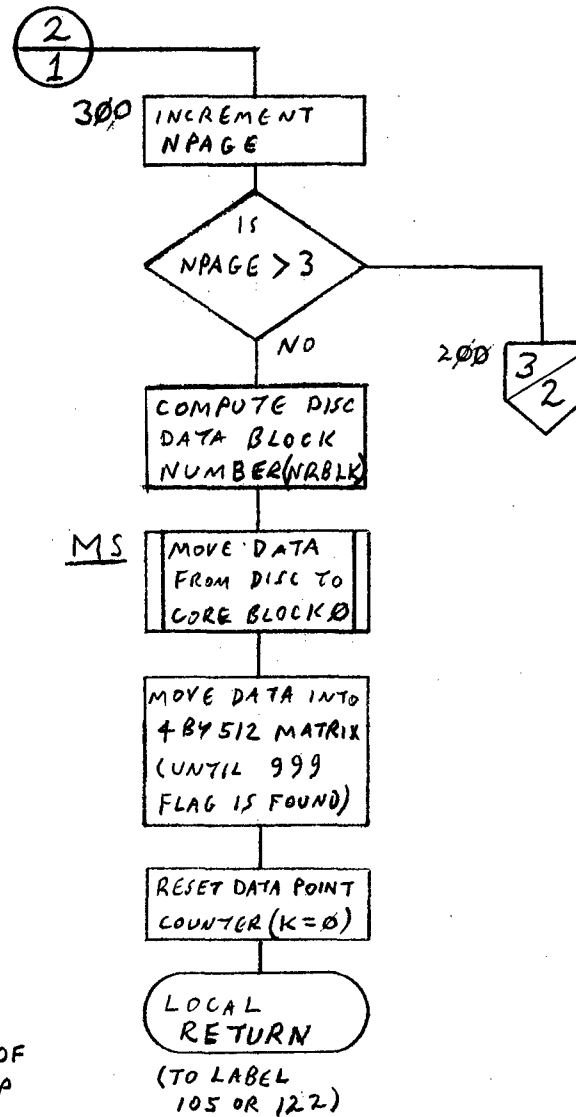
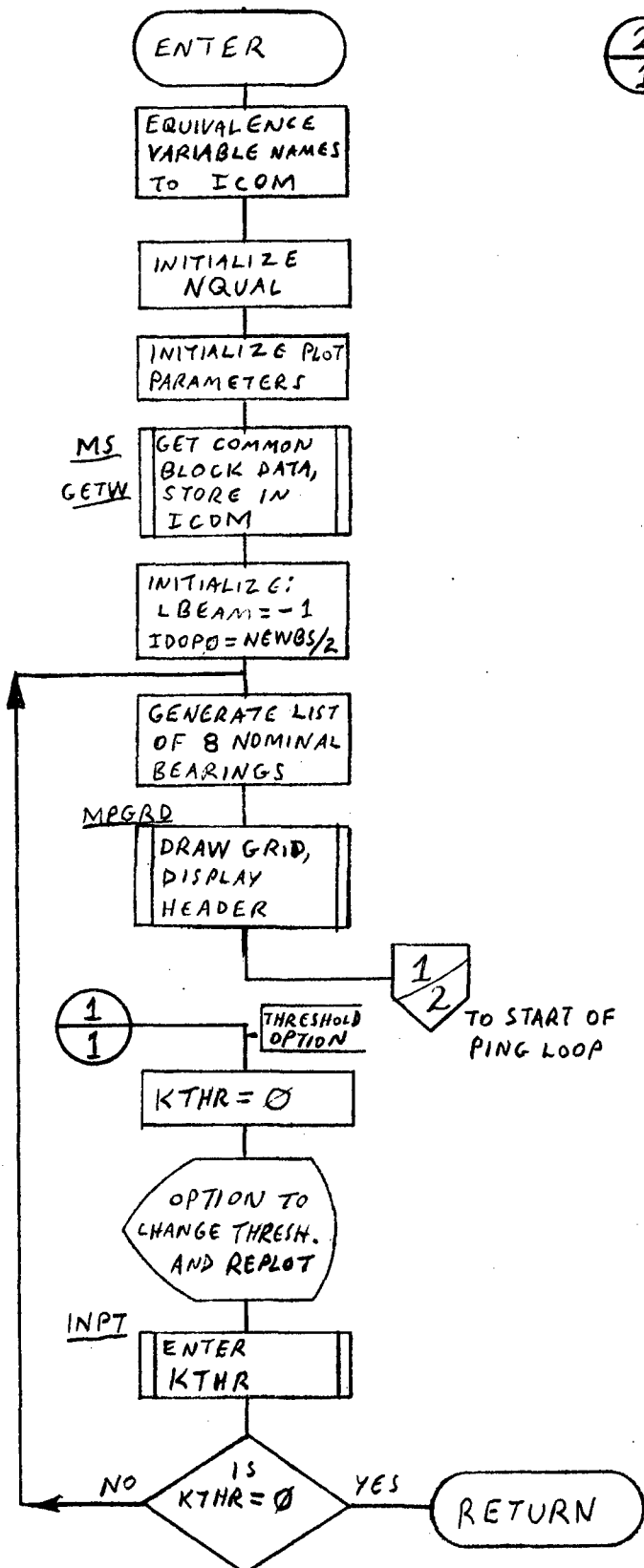


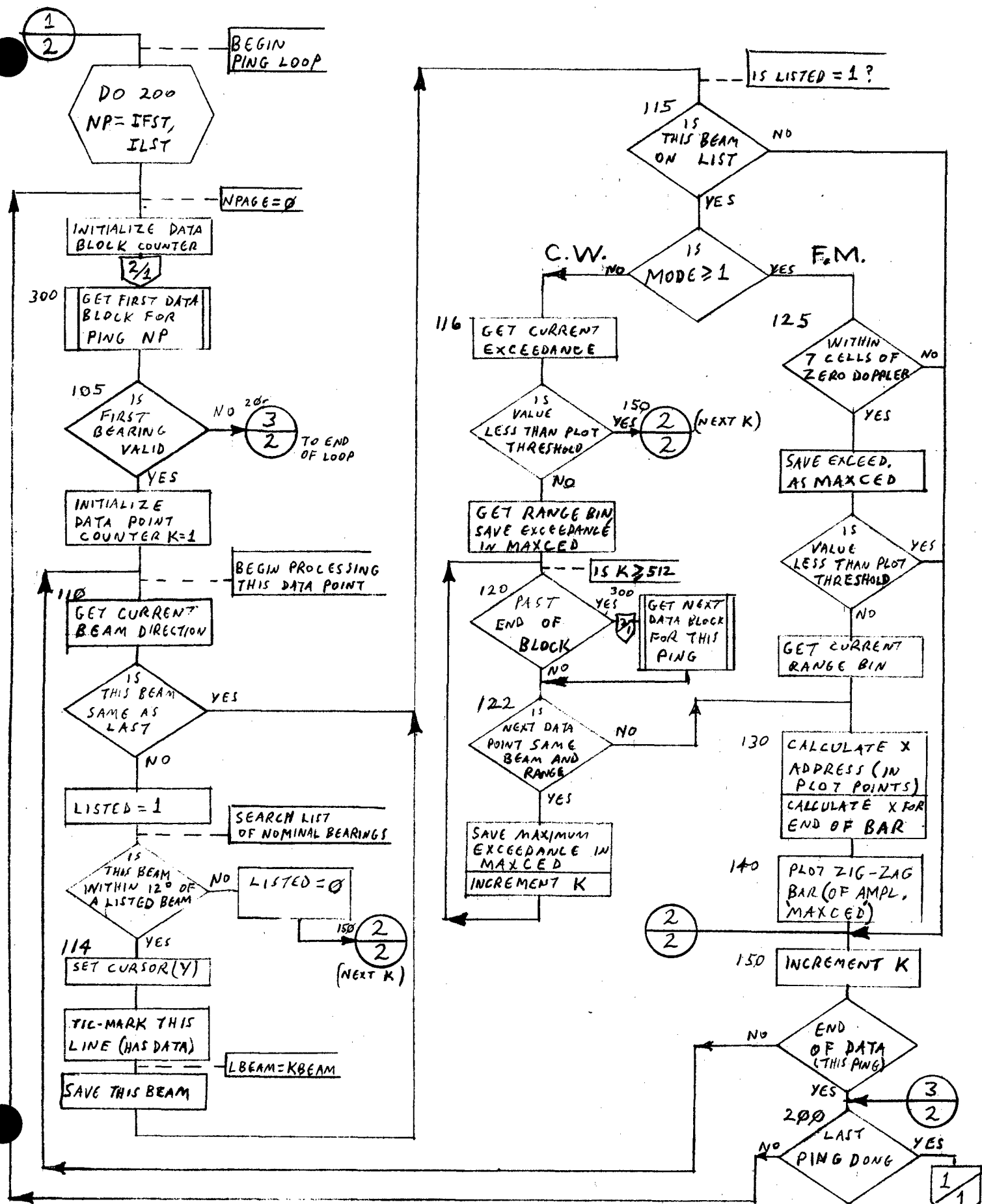




SUBROUTINE MPDSP

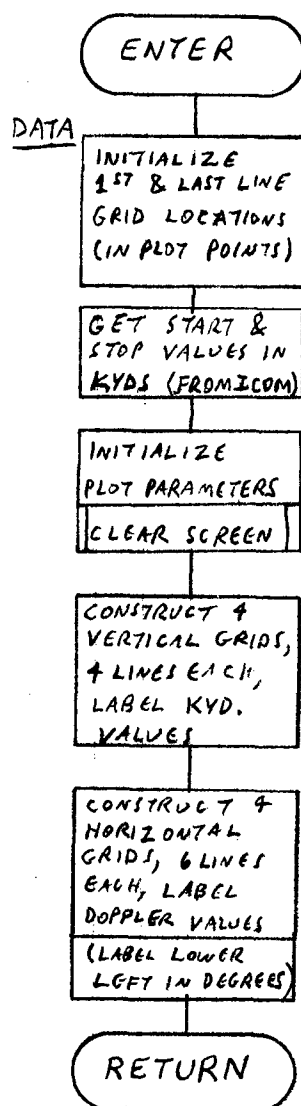
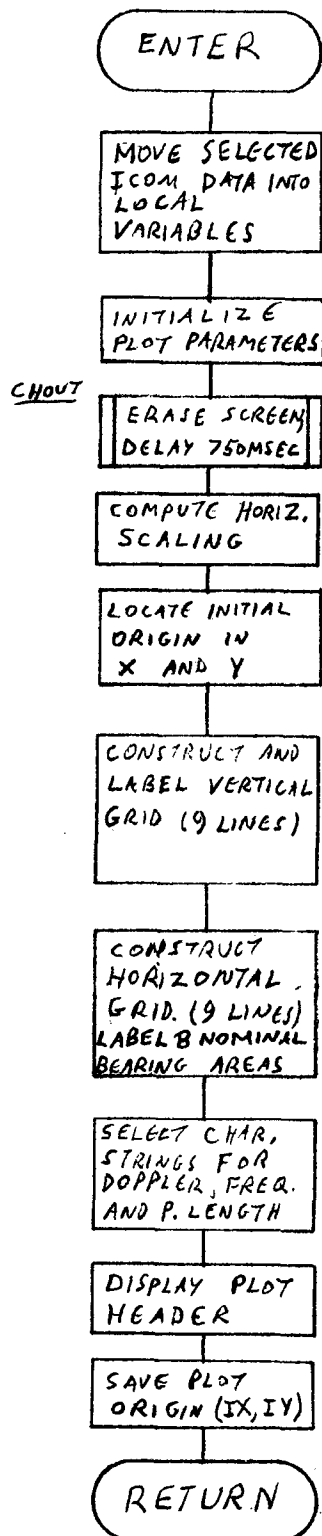
RETRIEVE-DATA-BLOCK-FROM-DISC ROUTINE





SUBROUTINE MPGRD (ICOM, LIST BM)

SUBROUTINE FPGRD (ICOM)



GRID CONSTRUCTION SUBROUTINES

(SHEET 1 OF 1)

MP0007A T-00004 IS ON CR00001 USING 00138 BLKS R-0796

```
0001 FTN4
0002 SUBROUTINE Y1007
0003 C
0004 C ***** LATEST MODIFICATION: 2/10/78 (WPB) *****
0005 C *
0006 C * TITLE: OLAAS PLOT DISPLAYS SUBROUTINES
0007 C *
0008 C * MODULE: Y1007 SOURCE: MP1007 VERSION: X02
0009 C * OBJECT: Y10070
0010 C *
0011 C * AUTHOR: R.A. FULLERTON 1976
0012 C *
0013 C * MODIFICATIONS:
0014 C * 1: ADDING MULTI-PING DISPLAY
0015 C * 11/30/77 C.R. MOOSE
0016 C * 2: CHANGE TO CALCULATE THE ANALYSIS THRESHOLD
0017 C * 1/21/78 W. BOWEN
0018 C *
0019 C * MODULE FUNCTION:
0020 C * THIS SUBROUTINE IS USED TO PLOT ERAPS DETECTION
0021 C * INFORMATION ON A TEKTRONIX 4012 DISPLAY. PLOT
0022 C * PARAMETERS ARE OBTAINED FROM THE ANALYSIS PART OF THE
0023 C * PROGRAM (CORELOAD 1/3).
0024 C *
0025 C * CALLING SEQUENCE: CALLED BY - Y1002 (ANALYSIS)
0026 C *
0027 C * CALL CLOAD (53B,51)
0028 C *
0029 C *****
0030 C *****
0031 C *****
0032 C *****
```

```

0033 *****
0034 * COMMON AREA ASSIGNMENTS (ICOM AND/OR DISC DATA BLOCK 0) *
0035 *****
0036 *
0037 * JDIU: DIVISOR OF ORIGINAL BLOCK FOR DOPPLER EDITING *
0038 * IBND: SPECIFIC DOPPLER ANALYSIS BAND (1-JDIU BANDS) *
0039 * NEWBS: BLOCK SIZE OF DOPPLER BAND *
0040 * MIN: MINIMUM INDEX USED IN ANALYSIS BY RDATA PGM. *
0041 * MAX: MAXIMUM INDEX USED IN ANALYSIS BY RDATA PGM. *
0042 * IFRAC: ANALYSIS THRESHOLD PERCENTAGE SET BY USER *
0043 * INCR: TIME INCREMENT IN MULTIPLES OF 0.192 SECONDS *
0044 * CODE: HANNING EQUALS 1, NO HANNING EQUALS 0 *
0045 * KTYPE: RANGE, KILOYARDS EQUALS 1, TIME EQUALS 0 *
0046 * IPNG: PING NUMBER *
0047 * KNUM: NUMBER OF BEAMS USED IN ANALYSIS *
0048 * KOFFS: TIME OFFSET VALUE *
0049 * MODE: 0 - CW PULSE, 1 - FM UP, 2 - FM DOWN *
0050 * LGTH: EITHER 2 OR 5 SECONDS *
0051 * KT1: TIME, HOURS *
0052 * KT2: TIME, MINUTES *
0053 * KT3: TIME, SECONDS *
0054 * LFRQ: BUOY FREQUENCY, 1 - F1, 2 - F2 *
0055 * JDIU: DIVISOR OF ORIG. BLOCK (SAME AS WORD 0) *
0056 * NTATE: DOPPLER AXIS ROTATION, 1 - YES, 0 - NO *
0057 * NSHFT: FOR NTATE=1, NO. OF KNOTS TO SHIFT DOPPLER *
0058 * SPARE *
0059 * KEY: 0 - ON-LINE PLOT *
0060 * 1 - ANALYSIS OF PREVIOUSLY RECORDED DATA *
0061 * 2 - INDIVIDUAL BEAM DISPLAY *
0062 * 3 - MULTI-PING DISPLAY *
0063 * 4 - FOUR-PART DISPLAY *
0064 * (23-32) ITH1: NUMBER OF WORDS STORED FOR EACH PING *
0065 * SPARE *
0066 * (33) *
0067 * (34) MPXNGS: MAXIMUM NUMBER OF PINGS TO BE COLLECTED *
      * (35) IFST: FIRST PING NO. FOR ANALYSIS *
*****

```

```

0068 C      (36)  ILST: LAST PING NO. FOR ANALYSIS
0069 C      (37)  IDPLR: DOPPLER BAND (1-FULL, 2-HALF, 3-UP, 4-DOWN)
0070 C      (38)  JSTRT: START RANGE (IN KYDS.)
0071 C      (39)  JEND: END RANGE (IN KYDS.)
0072 C      (40)  JDIR: DIRECTION OF BEARING OF CENTER BEAM OR
0073 C      (41)  IMOD: MODE (ON-LINE / OFF-LINE ANALYSIS)
0074 C      (42-46)  SPARES
0075 C      (47)  INCRX: LENGTH OF ONE RANGE INCREMENT (IN PLOT POINTS)
0076 C      (48)  KTHR: CURRENT PERCENTAGE FOR PLOT THRESHOLD (1-100)
0077 C      (49-50)  SPARE
0078 C      (51-66)  IHDR: HEADER FROM INPUT DATA TAPE
0079 C      (67-69)  SPARES
0080 C      (70)  NFFT: TAG: DUPLICATE OF NOT-FIRST-TIME SWITCH
0081 C      (71-80)  IAUGEX: NUMBER OF EXCEEDENCES FOR ALL 10 PINGS
0082 C      (81-90)  MIDBM: CENTER BEAM ANALYZED FOR ALL 10 PINGS
0083 C      (91-100)  NRBMS: NUMBER OF BEAMS ANALYZED FOR ALL 10 PINGS
0084 C      (101-109)  IDATE: ALPHANUMERIC FIELD FOR DATE (18 CHAR.)
0085 C      (110-119)  SPARES
0086 C      (120-129)  SPARES
0087 C      (130-139)  SPARES
0088 C      (140-149)  SPARES
0089 C      (150-159)  SPARES
0090 C      (160-169)  SPARES
0091 C      (170-179)  SPARES
0092 C      (180-189)  SPARES
0093 C      (190-199)  SPARES
0094 C      (200-209)  SPARES
0095 C      (210-219)  SPARES
0096 C      (220-229)  SPARES
0097 C      (230-239)  SPARES
0098 C      (240-249)  SPARES
0099 C      (250-259)  SPARES
0100 C      (260-269)  SPARES
0101 C      (270-279)  SPARES
0102 C      (280-289)  SPARES

```

```

DIMENSION NQUAL(5),HORZ(9),VERT(7),NBM(16),IBCD(6),LABEL(2),
+MASK(4),ICOM(127),IHDR(16),ITH(10),ITHNDX(10),LISTBM(16),
+IAUGEX(10),MIDBM(10),NRBMS(10),IDATE(9),OFFSET(4,2),
+PSCALE(2)

```

EQUIVALENCE

```

+(ICOM( 1), IBND), (ICOM( 2), NEUBS), (ICOM( 3), MIN),
+(ICOM( 4), MAX), (ICOM( 5), IFRAC), (ICOM( 6), INCR),
+(ICOM( 7), KODE), (ICOM( 8), KTYPE), (ICOM( 9), IPNG),
+(ICOM(10), KNUM), (ICOM(11), KOFFS), (ICOM(12), MODE),
+(ICOM(13), LENGTH), (ICOM(14), KT1), (ICOM(15), KT2),

```

```

0103 + (ICOM( 16), KT3), (ICOM( 17), LFRQ), (ICOM( 18), JDIU),
0104 + (ICOM( 19), NTATE), (ICOM( 20), NSHFT),
0105 + (ICOM( 22), KEY), (ICOM( 23), ITH),
0106 + (ICOM( 34), MXPNGS), (ICOM( 35), IFST), (ICOM( 36), ILST),
0107 + (ICOM( 37), IDPLR), (ICOM( 38), JSTRT), (ICOM( 39), JEND),
0108 + (ICOM( 40), JDIR), (ICOM( 41), IMOD),
0109 + (ICOM( 47), INCRX), (ICOM( 48), KTHR),
0110 + (ICOM( 70), NFFTAG), (ICOM( 71), IAGEX), (ICOM( 51), IHDR),
0111 + (ICOM( 91), NRBMS), (ICOM(101), IDATE)
0112
0113 DATA ITHNDX/1,2,3,4,5,6,7,8,9,10/
0114 DATA MASK/17B,360B,7400B,170000B/
0115 DATA NQJAL/2048,100B,16383,45,0/,LISTEN/16X0/,NOPT/2/
0116 DATA IERASE,IESCAP,IRESET,IDASH/14B,33B,37B,55B/
0117 DATA OFFSET/104.,274.,444.,104.,719.,577.,719.,435./
0118 DATA NDIR1, NDIR2, NDIR3, NDIR4/4X999/
0119 DATA PSCLAE/2.6592,2.216/,NDIR2,NDIR3/2X999/
0120
0121 C *****
0122 C * OLAAS PLOT DISPLAY SUBROUTINE INITIAL ENTRY POINT *
0123 C *****
0124 C *****
0125
0126 C >>>>> DIAGNOSTIC >> SENSE SWITCH 5 (ON) DUMP DATA BLOCK
0127 IF (ISSW(5) .LT. 0) CALL BRDB
0128
0129 C - - - - - READ DATA FROM DISC BLOCK 0 TO CORE BLOCK 0
0130 CALL KYBD(2HMS, 31, 0)
0131 CALL KYBD(2HMS, 11, 0)
0132
0133 C - - - - - TRANSFER DATA INTO COMMON VECTOR (ICOM)
0134 DO 106 J = 1, 127
0135 CALL GETW(0, J, ICOM(J), II)
0136 106 CONTINUE
0137

```



```

0138 C >>>>>> DIAGNOSTIC >> SENSE SWITCH 10 (ON) WRITES ICOM
0139 IF(ISSW(10) .LT. 0) WRITE(1, 9106) ICOM
0140 FORMAT('07', 10I7)
0141 IF(ISSW(10) .LT. 0) WRITE (1, 9998)
0142 FORMAT(' ')
0143 IF(ISSW(10) .LT. 0) PAUSE 15
0144
0145 C - - - - - INITIALIZE NEW PARAMETERS FOR THIS PLOT
0146 NSEP = 112
0147 IX = 1023 - 8 * NSEP
0148 IY = 780 - 6 * NSEP
0149
0150 IF (ISSW(14) .LT. 0) GO TO 107
0151 THRES = ICOM( 5) * 1E-6
0152 107 NTHR = KTHR
0153 NDIR = JDIR
0154 NBEAMS= KNUM
0155 NOPT = KEY
0156 NPNG = IPNG
0157 200 IF(NOPT .NE. 3) GO TO 201
0158
0159 C - - - - - TRANSFER TO MULTI-PING ROUTINE
0160 CALL MPDSP
0161 GO TO 707
0162
0163 201 IF(NOPT .NE. 4) GO TO 202
0164
0165 C - - - - - FOR 4-PART, GET FIRST AND SECOND NOMINAL DIRECTION
0166 NDIR1 = NDIR - 12
0167 IF(NDIR1 .LT. 0) NDIR1 = NDIR1 + 360
0168 NDIR2 = MOD(NDIR + 12, 360)
0169
0170 C >>>>>> DIAGNOSTIC >> SENSE SWITCH 9 (ON)
0171 IF(ISSW(9) .LT. 0) WRITE(1, 9108)NDIR1,NDIR2,NDIR
0172 9108 FORMAT('NDIR1&2 =',3I6)

```

```

0173 IF(ISSW(9) .LT. 0) PAUSE 24
0174
0175 C - - - - - DRAW GRID FOR FOUR-PART DISPLAY
0176 CALL FPGRD(ICOM)
0177 GO TO 2035
0178
0179 C - - - - - CLEAR SCREEN, PREPARE FOR PLOT
0180
0181 C >>>>> DIAGNOSTIC >> SENSE SWITCH 9 (ON)
0182 202 IF(ISSW(9) .LT. 0) PAUSE 1
0183
0184 CALL TPLOT(0,0,0)
0185 CALL CHOUT(IESCAP)
0186 CALL CHOUT(IERASE)
0187 CALL D750
0188
0189 C - - - - - CALCULATE BEARINGS OF DESIRED BEAMS FOR DISPLAY
0190 BRG = NDIR - 22.5*FIX((NBEAMS-1)/2.) - 11.25*MOD(NBEAMS-1,2)
0191 DO 203 N = 1, NBEAMS
0192 IF(BRG .GE. 360.) BRG = BRG - 360.
0193 IF(BRG .LT. 0.) BRG = BRG + 360.
0194 LISTBM(N) = IFIX(BRG + 0.5)
0195 203 BRG = BRG + 22.5
0196
0197 C >>>>> DIAGNOSTIC >> SENSE SWITCH 10 (ON)
0198 IF(ISSW(10) .LT. 0) WRITE(1, 9201) (LISTBM(I), I = 1, NBEAMS
0199 9201 FORMAT('BEAMS:',8I6)
0200
0201 C - - - - - GET FIRST DATA BLOCK FROM DISC,
0202 C SET LOCAL ADDRESS AND NUMBER OF HITS THIS BEAM TO 0.
0203 2035 KTBK = 1
0204 NRBLK = 3*(NPNG-1) + 1
0205 CALL PUTQ(0, NQUAL)
0206 CALL KYBD(2HMS, 31, NRBLK)
0207 CALL KYBD(2HMS, 11, 0)

```

```

0208 LA = 0
0209 NHTB = 0
0210
0211 C >>>>>> DIAGNOSTIC >> SENSE SWITCH 9 (ON)
0212 IF(ISSW(9) .LT. 0) WRITE(1, 9990)KTBLK,NRBLK,LA,LISTBM(1)
0213
0214 C - - - - - GET FIRST DIRECTION, GO TO BEAM LIST TEST
0215 204 CALL GETW(0, 0, KDIR, II)
0216 IF(KDIR .LE. 360) GO TO 205
0217 WRITE(1, 9214) NPNG, KDIR
0218 9214 FORMAT('NO DATA ANALYZED FOR PING NO.', I3, ' . BEARING -', I
0219 GO TO 800
0220 205 IF(NOPT .EQ. 4) GO TO 210
0221 GO TO 214
0222
0223 C - - - - - GET DIRECTION OF CURRENT DATA POINT. END OF DATA?
0224 210 CALL GETW(0, LA, KDIR, II)
0225
0226 C >>>>>> DIAGNOSTIC >> SENSE SWITCH 6 (ON)
0227 IF(ISSW(6) .LT. 0) WRITE(1, 9210)KDIR, LDIR, LA
0228 9210 FORMAT(50X, 'KDR', 3I5)
0229
0230 ASSIGN 801 TO IRET
0231 IF(KDIR .EQ. 999) GO TO 90
0232
0233 C - - - - - IF THIS BEAM IS SAME AS LAST, GO TEST AMPLITUDE
0234 IF(KDIR .EQ. LDIR) GO TO 220
0235
0236 C NEW BEAM - - - - - NEW BEAM - - - - - NEW BEAM - - - - -
0237
0238 C >>>>>> DIAGNOSTIC >> SENSE SWITCH 9 (ON)
0239 IF(ISSW(9) .LT. 0) WRITE(1, 9211)LDIR, KDIR
0240 9211 FORMAT('LAST/NEW DIR.', 2I5)
0241
0242

```

```

0243 C - - - - - IF FOUR-PART, IS THIS ONE OF TWO DESIRED BEAMS?
0244     IF(NOPT .NE. 4)
0245         LISTED = 1
0246         LDIR = KDIR
0247         MTAG = 0
0248         IDIF = IABS(KDIR - NDIR1)
0249         IF(IDIF .GT. 12 .AND. IDIF .LT. 348) GO TO 211
0250         MTAG = 1
0251         NDIR3 = KDIR
0252         GO TO 211
0253     211 IDIF = IABS(KDIR - NDIR2)
0254         IF(IDIF .GT. 11 .AND. IDIF .LT. 349) GO TO 221
0255         MTAG = 2
0256         NDIR4 = KDIR
0257         GO TO 221
0258
0259 C - - - - - IF PLOTTING INDIVIDUAL BEAMS, END PLOT FOR PREVIOUS BEAM
0260     212 ASSIGN 214 TO IRET
0261     IF(NOPT .LE. 2)
0262         > > > > > > > > > >
0263
0264 C - - - - - SAVE THIS DIRECTION. IS THIS BEAM ON THE LIST?
0265     214 LDIR = KDIR
0266         LISTED = 1
0267         DO 216 KTB = 1, NBEAMS
0268             IDIF = IABS(LISTBM(KTB) - LDIR)
0269             IF(IDIF .LE. 12 .OR. IDIF .GE. 348) GO TO 221
0270     216 CONTINUE
0271 C
0272         NOT IN BEAM LIST
0273         LISTED = 0
0274         GO TO 222
0275
0276 C - - - - - FOUND IN BEAM LIST? GET DATA VALUE, TEST THRESHOLD
0277     220 IF(LISTED .EQ. 0) GO TO 222
0278     221 CALL GETW(0, LA+2, IDATA, II)

```

[illegible]

```

0313 C %%%%%%%%%%%%%%
0314 C - -COMPRESS DATA VALUE FOR PLOTTING, INCR. NO. OF HITS THIS BEAM
0315 12 IDTA = IFIX(0.1 * IDATA + 0.5)
0316
0317 C >>>>>> DIAGNOSTIC >> SENSE SWITCH 1 (ON)
0318 IF(ISSW(1) .LT. 0) WRITE(2,989)KDIR,INDEX,IDTA,KDOP,MTAG
0319 989 FORMAT(1X, 5I9)
0320
0321 NHTB = NHTB + 1
0322
0323 C > > > CONVERT TO FLOATING POINT
0324 X1 = INDEX
0325 X2 = MIN
0326 X3 = MAX
0327 X4 = KDOP
0328 X5 = NEWBS
0329 ICELX = (INDEX - MIN)*896.0 / (MAX - MIN) + 127.5
0330 IF (NOPT .EQ. 4) ICELX = ICELX * 0.4911 + 1.6
0331
0332 C > > > DOPPLER SCALING - COMPUTE Y COORDINATE
0333 ICELY = OFFSET(IDPLR,LFRQ) + PSCALE(LFRQ) * X4
0334 IF (NOPT .EQ. 4) ICELY = ICELY * 0.9821 + 179.5
0335
0336 C > > > BRANCH TO PROPER LIMIT CHECK AND PLOT (1-BEAM OR 4-PART)
0337 IF (NOPT .EQ. 4) GO TO 145
0338 IF(ICELX .LT. IX .OR. ICELX .GT. 1019) GO TO 121
0339 IF(ICELY .LT. IY .OR. ICELY .GT. 780) GO TO 121
0340
0341 C *****
0342 C * PLOT THE DETECTIONS
0343 C *****
0344 C *****
0345 IF (IDTA .GT. 10) IDTA = 10
0346 M = 1
0347

```

```
0348 MY = ICELY  
0349 2144 IF (MY .GT. 780) GO TO 2147  
0350 CALL TPLOT (0,ICELX,MY)  
0351 CALL CHOUT (37B)  
0352 CALL CHOUT (55B)  
0353 MY = MY + 2  
0354 M = M + 1  
0355 IF (M .LE. IDTA) GO TO 2144  
0356 2147 CONTINUE  
0357 C >>>> DIAGNOSTIC >> SENSE SWITCH 9 (ON)  
0358 IF(ISSW(9).LT. 0) WRITE(1, 9121)ICELX,MY,LA,IDTA  
0359 9121 FORMAT("PLT",4I5)  
0360  
0361 C > > > RETURN TO CALLER  
0362 121 GO TO IRET  
0363  
0364 C > > > PLOT DETECTIONS ON 4-PART DISPLAY  
0365 145 IF (ICELX .LT. 60 .OR. ICELX .GT. 500) GO TO 121  
0366 IF (ICELY .LT.450 .OR. ICELY .GT. 780) GO TO 121  
0367 IF (IDTA .GT. 10) IDTA = 10  
0368 IDIRY = 410 - KDIR * 0.91667  
0369 CALL PLLIN (ICELX, INCRX, IDIRY, IDTA, 420)  
0370 CALL PLLIN (ICELX, INCRX, ICELY, IDTA, 780)  
0371 MAXHT=780  
0372 IF (MTAG.EQ.1) GO TO 148  
0373 IF (MTAG.EQ.2) GO TO 147  
0374 GO TO IRET  
0375  
0376 147 MAXHT=410  
0377 ICELY=ICELY-370  
0378 148 ICELX=ICELX+510  
0379 CALL PLLIN (ICELX, INCRX, ICELY, IDTA, MAXHT)  
0380 GO TO IRET  
0381  
0382 C < < < < < < < <  
C %%%%%%%%%XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```

0383 C > > > END PLOT. MAKE GRID AND PRINT HEADER.
0384 90 CALL CHOUT(IRESET)
0385 IF (NOPT.EQ. 4) GO TO 89
0386 IF(NHTB.LE. 0 .AND. LISTED.EQ. 1) WRITE(1, 9090) LDIR
0387 FORMAT(" NO EXCEEDENCES FOR BEAM DIRECTION", I4)
0388 IF(NHTB.LE. 0) GO TO IRET
0389
0390 C >>>>> DIAGNOSTIC >> SENSE SWITCH 9 (ON)
0391 IF(ISSU(9).LT. 0) WRITE(1, 9091)IX,IY,NSEP,KOFFS,KDIR,LA
0392 9091 FORMAT("GRD",6I6)
0393
0394 C >>>>> DIAGNOSTIC >> SENSE SWITCH 8 (ON)
0395 IF(ISSU(8).LT. 0) PAUSE 400
0396
0397 89 GO TO (91,92,93,94) IDPLR
0398 91 LABEL(1) = 2HFU
0399 LABEL(2) = 2HLL
0400 GO TO 95
0401 92 LABEL(1) = 2HHA
0402 LABEL(2) = 2HLF
0403 GO TO 95
0404 93 LABEL(1) = 2HUP
0405 LABEL(2) = 2H
0406 GO TO 95
0407 94 LABEL(1) = 2HDO
0408 LABEL(2) = 2HUN
0409 95 CONTINUE
0410
0411 C > > > COMPUTE HORIZONTAL SCALING
0412 IF (NOPT.EQ. 4) GO TO 179
0413 X7 = KOFFS
0414 6 DELTI = JEND - JSTRT
0415 DO 411 M= 1,9
0416 411 HORZ(M) = JSTRT + (M-1) * DELTI / 8
0417

```



```

0418 C > > > COMPUTE VERTICAL SCALING
0419 DO 2 I = 1,7
0420 J = I - 1
0421 2 VERT(I) = 30.0 - (J * 10.0)
0422
0423 C > > > CONSTRUCT GRID
0424 DO 20 J=1,7
0425 LY=IY + (J-1)*XNSEP
0426 CALL TPLOT(0,IX,LY)
0427 20 CALL TPLOT(1,1023,LY)
0428
0429 DO 30 M=1,9
0430 LX=IX + (M-1)*XNSEP
0431 CALL TPLOT(0,LX,IY)
0432 30 CALL TPLOT(1,LX,780 )
0433
0434 C > > > LABEL VERTICAL AXIS
0435 DO 19 L = 1,7
0436 J = L - 1
0437 LY = 772 - (J*XNSEP)
0438 IF(L .EQ. 1) LY = 780
0439 IF(L .EQ. 7) LY = LY + 12
0440 CALL TPLOT(0,55,LY)
0441 CALL CHOUT(IRESET)
0442 19 WRITE(2,22)VERT(L)
0443 22 FORMAT(F5.1)
0444
0445 C > > > LABEL HORIZONTAL AXIS
0446 CALL TPLOT(0,91,85)
0447 CALL CHOUT(IRESET)
0448 WRITE(2,406)(HORZ(I),I=1,9)
0449 406 FORMAT(7(F5.1, 3X),F5.1,1X,F5.1)
0450 CALL TPLOT(0,0,62)
0451 CALL CHOUT(IRESET)
0452

```

```

0453 C * * * * * PRINT HEADER DATA FOR PLOT * * * * *
0454 C CHECK FOR SINGLE OR MULTIPLE BEAM CASE (FOR HEADER INFO)
0455 KPNG = NPNG
0456 IF(NOPT.EQ. 5) GO TO 162
0457 37 WRITE (2,151) NPNG,LDIR,LABEL,IFRAC
0458 151 FORMAT ("PING NO:",I3,2X,"BEARING:",I4,2X,"DOPPLER: ",
0459 +2A2.5X,"ANALYSIS THRESHOLD: ",I5,"X")
0460 GO TO 179
0461
0462 C > > > MULTIPLE BEAM CASE
0463 162 WRITE (2,163) NPNG,LABEL
0464 163 FORMAT ("PING NO:",I3,2X,"DOPPLER: ",2A2.2X,"XX ALL"
0465 + " BEAMS OR'ED XX")
0466 179
0467 LEN = 2HT2
0468 LFRE = 2HF2
0469 LTYP = 2HCU
0470 IF(MODE.GT. 0) LTYP = 2HFM
0471 IF(LFRQ.EQ. 1) LFRE = 2HF1
0472 IF(LNGTH.EQ. 2) LEN = 2HT1
0473 IF(NOPT.EQ. 4) GO TO 230
0474 IF(NTATE.EQ. 0) GO TO 108
0475
0476 WRITE (2,181) LTYP,LEN,LFRE,KODE,NTHR,NSHIFT,KT1,KT2,
0477 +KT3,IDATE
0478 181 FORMAT ("SIGNAL:",2X,A2,2X,A2,2X,A2,2X,"HANNING:",I2,18X,
0479 + "PLOT THRESHOLD:",I3,/, "(BAND ROTATED -",I3," KNOTS)",I3X,
0480 + "TIME: ",I2,":",I2,":",I2,3X,9A2)
0481 GO TO 233
0482 108 WRITE (2,112) LTYP,LEN,LFRE,KODE,NTHR,KT1,KT2,KT3,IDATE
0483 112 FORMAT ("SIGNAL:",2X,A2,2X,A2,2X,A2,2X,"HANNING:",I2,18X,
0484 + "PLOT THRESHOLD:",I3,40X,"TIME: ",I2,":",I2,":",I2,3X,9A2)
0485 GO TO 233
0486
0487 C > > > FOUR-PART DISPLAY CASE
0488 230 CALL TPLOT(0,5,35)

```

```

0488 CALL CHOUT(IRESET)
0489 WRITE(1,9230) NPNG,LABEL,NTHR,NDIR3,NDIR4,LTYP,LEN,
0490 +LFRE,KODE,KT1,KT2,KT3,IDATE
0491 9230 FORMAT ("PING NO:",I3,2X,"DOPPLER: ",2A2,2X,"PLOT"
0492 + " THRESHOLD:",I3,2X,"RIGHT PLOT BRGS:",I4,"/",I3,
0493 + " SIGNAL:",2X,A2,2X,A2,2X,A2,2X,"HANNING:",I2,8X,
0494 + " TIME: ",I2,":",I2,":",I2,3X,9A2)
0495
0496 C > > > CREATE DELAY FOR VIEWING DISPLAYS
0497 233 CALL D750
0498 CALL D750
0499 NHTB = 0
0500
0501 C >>>>> DIAGNOSTIC >> SENSE SWITCH 7 (ON)
0502 IF(ISSU(7) .GE. 0) GO TO IRET
0503
0504 C * * * * * END OF PLOTS. RETURN TO CORELOAD 3
0505 800 CALL ENDIO
0506 CALL TPLOT(0, 0, 0)
0507 CALL CHOUT(IRESET)
0508 WRITE(1, 9998)
0509 801 CONTINUE
0510
0511 C >>>>> DIAGNOSTIC >> SENSE SWITCH 2 (ON)
0512 IF(ISSU(2) .LT. 0) PAUSE 4
0513
0514 C IF(IMOD .EQ. 1 .OR. NOPT .EQ. 1) GO TO 707
0515
0516 C - - - - - OPTIONS FOR PLOT OR NEW THRESHOLD - - -
0517 NTHR = 0
0518 WRITE(1, 9802)
0519 9802 FORMAT(" NEW THRESHOLD PERCENTAGE FOR PLOT (1 - 100)",
0520 * " (1% DISPLAYS ALL ANALYZED DATA, 100% = NONE)",
0521 * " TYPE 0 TO RETURN TO ANALYSIS PROGRAM")
0522

```

```

0523 CALL INPT(NTHR, 0, 100)
0524 IF(NTHR .EQ. 0) GO TO 707
0525 GO TO 200
0526
0527 C - - - - - EXIT PROGRAM. GO TO CORELOAD 3
0528 707 CALL ENDIO
0529 CALL CLOAD(23B,51B)
0530 157 WRITE(2,158)
0531 158 FORMAT('XXXXX PARAMETER ENTRY ERROR XXXXX')
0532 CALL ENDIO
0533 CALL CLOAD(23B,51B)
0534 FORMAT(1X, 10I7)
0535 C ##### END OF THE PLOT ROUTINE #####
0536 END
0537
0538 SUBROUTINE MPDSP
0539 C JAN. 26, 1978
0540 C #####
0541 C * OLARS MULTI-PING DISPLAY ROUTINE
0542 C #####
0543
0544 DIMENSION ICOM(127),LISTM(8),NQAL(5),MAT(4, 512),IDATE(9)
0545
0546 EQUIVALENCE (ICOM(2),NEWBS),(ICOM(3),MIN),(ICOM(4),MAX),
0547 +(ICOM( 6), INCR),(ICOM(11),KOFFS),(ICOM(12), MODE),
0548 +(ICOM(13),LNQTH),(ICOM(14), KT1),(ICOM(15), KT2),
0549 +(ICOM(16), KT3),(ICOM(17), LFRQ),(ICOM(22), KEY),
0550 +(ICOM(35), IFST),(ICOM(36), ILST),(ICOM(40), JDIR),
0551 +(ICOM(45), IX),(ICOM(46), IY),(ICOM(47), INCRX),
0552 +(ICOM(48), KTHR),(ICOM(51), IHDR),(ICOM(101),IDATE)
0553
0554 DATA NQAL/2048,1008,16383,45,0/
0555
0556 C >>>>>> DIAGNOSTIC >> SENSE SWITCH 2 (ON)
0557 IF(ISSW(2) .LT. 0) PAUSE 7

```

```

0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592

LBEAM = -1

C - - - - - INITIALIZE PLOT PARAMETERS
IRESET = 37B
IDASH = 55B

C - - - - - GET COMMON BLOCK DATA, STORE IN ICOM - - - - -
CALL PUTQ(0, NQUAL)
CALL KYBD(2HMS, 31, 0)
CALL KYBD(2HMS, 11, 0)
DO 10 I = 1, 127
  CALL GETJ(0, I, ICOM(I), II)
10 CONTINUE
IDOP0 = NEWBS / 4
CALL PTIME(IFST, ICOM)

C - - - - - GENERATE LIST OF DESIRED BEARINGS (NOMINAL) -
15 BRG = JDIR + 78.75
DO 20 J = 1, 8
  IF(BRG .LT. 0.) BRG = BRG + 360
  IF(BRG .GE. 360.) BRG = BRG - 360
  LISTB(J) = IFIX(BRG + 0.5)
20 BRG = BRG - 22.5

C >>>>> DIAGNOSTIC >> SENSE SWITCH 10 (ON)
IF(ISSW(10) .LT. 0) WRITE(2, 920) ICOM
920 FORMAT('MP ', 10I6)

C >>>>> DIAGNOSTIC >> SENSE SWITCH 4 (ON)
IF(ISSW(4) .LT. 0) WRITE(1, 921) LISTB, KTHR, IFST, ILST
921 FORMAT(20X, 8I4)

C >>>>> DIAGNOSTIC >> SENSE SWITCH 10 (ON)
IF(ISSW(10) .LT. 0) PAUSE 15

```

0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627

100 CALL MPGRD(ICOM, LISTBM)

C - - - - - BEGIN PING LOOP - - - -
DO 200 NP = IFST, ILST
NPAGE = 0

C - - - GET FIRST DATA BLOCK PAGE FOR THIS PING, SET LOCAL RETURN
ASSIGN 105 TO LRET

105 IF(MAT(1, 1) .GT. 360) GO TO 300
K = 1 GO TO 200

C - - - - - BEGIN PROCESSING THIS DATA POINT LINE
110 KBEAM = MAT(1, K)

C >>>>>> DIAGNOSTIC >> SENSE SWITCH 9 (ON)
IF(ISSW(9) .LT. 0) WRITE(1, 9320) (MAT(I,K), I = 1, 4)
IF(KBEAM .EQ. LBEAM) GO TO 115

C >>>>>> DIAGNOSTIC >> SENSE SWITCH 10 (ON)
IF(ISSW(10) .LT. 0) WRITE(1, 991) NP, NBLK, LBEAM, KBEAM
991 FORMAT(1X, 10I6)

C - - - - IS THIS BEAM CLOSE TO A BEAM IN THE LIST FOR PLOTTING? - -
LISTED = 1
DO 112 NDX = 1, 8
IDIF = IABS(LISTBM(NDX) - KBEAM)
IF(IDIF .LE. 12 .OR. IDIF .GE. 348) GO TO 114
112 CONTINUE
LISTED = 0 GO TO 150

C - - - - SET CURSOR Y LOCATION FOR THIS BEAM PING, MARK THIS LIN

[illegible]

```

0663 IF(ISSU(6) .LT. 0) WRITE(1, 9125) (MAT(I, K), I = 1, 4),
0664 *
0665 9125 FORMAT(2X, 6I8)
0666
0667 C IF(IABS(MAT(4, K) - IDOP0) .GT. 15) GO TO 150
0668
0669 C TEST TEST TEST ALL DOPPLERS OR'ED TEST TEST
0670
0671 C - - - - DATA IS WITHIN 15 CELLS OF ZERO DOPPLER! TEST THRESHO
0672 MAXCED = MAT(3, K)
0673
0674 C >>>>> DIAGNOSTIC >> SENSE SWITCH 6 (ON)
0675 IF(ISSU(6) .LT. 0) WRITE(1, 9125) KTHR, MAXCED, KRANGE, K
0676
0677 IF(MAXCED .LT. KTHR) GO TO 150
0678 KRANGE = MAT(2, K)
0679 KTEXC = 1
0680
0681 C - - - - - ((F.M. DE-SKEW ALGORITHM. ACCUMULATE
0682 C EXCEEDICES ON THIS DIAGONAL.)) * * *
0683 C MAXCED = KTEXC
0684 C IF(KTEXC .LT. 1) GO TO 150
0685
0686 C - - - - - CALCULATE AND PLOT COORDINATES FOR THIS POINT - - -
0687 130 ICELX = (KRANGE - MIN) * 890.0 / (MAX - MIN) + 127.5
0688
0689 C >>>>>> DIAGNOSTIC >> SENSE SWITCH 9 (ON)
0690 IF(ISSU(9) .LT. 0) WRITE(2, 9130) K, MAXCED, MIN, MAX, INCRX,
0691 * ICELX, LY, KBEAM, KRANGE
0692 9130 FORMAT(9I6)
0693
0694 IF(ICELX .LT. IX .OR. ICELX .GT. 1019) GO TO 150
0695 KUAL = MIN0(10, IFIX(0.1*MAXCED + 0.5) )
0696 MY = MIN0(780, LY + KUAL)
0697 IENDX = MIN0(1019, ICELX + INCRX-4)

```



```

0698      CALL TPLOT(0, ICELX, LY)
0699      DO 140 LX = ICELX, IENDX, 4
0700          CALL TPLOT(1, LX+2, MY)
0701          CALL TPLOT(1, LX+4, LY)
0702      CONTINUE
0703
0704      C --- -- INCREMENT DATA LINE INDEX, TEST FOR END OF DATA --
0705      150      K = K + 1
0706          ASSIGN 105 TO LRET
0707          IF(K .GT. 512)          GO TO 300
0708          IF(MAT(1, K) .LT. 360) GO TO 110
0709
0710      C --- -- END OF PING LOOP ---
0711          LBEAM = -1
0712      200      CONTINUE
0713
0714      C >>>>> DIAGNOSTIC >> SENSE SWITCH 2 (ON)
0715          IF(ISSU(2) .LT. 0)      PAUSE 200
0716
0717      CALL TPLOT(0,0,0)
0718      CALL CHOUT(IRESET)
0719
0720      C --- -- -- OPTION TO CHANGE THRESHOLD AND REPLOT
0721          KTHR = 0
0722          WRITE(1, 9200)
0723      9200      FORMAT(/" ENTER NEW THRESHOLD PERCENTAGE FOR PLOT (1 - 100X)
0724          * " (1% DISPLAYS ALL ANALYZED AMPLITUDES. 100X= NONE)"/
0725          * " ENTER 0 TO RETURN TO ANALYSIS PROGRAM")
0726          CALL INPT(KTHR, 0, 100)
0727          IF(KTHR .GT. 0)          GO TO 15
0728          RETURN
0729      C * * * * *
0730
0731      C --- -- -- GET DATA BLOCK FROM DISC ---
0732      300      NPAGE = NPAGE + 1

```

```

0733 IF(NPAGE.GT. 3) GO TO 200
0734 NRBLK = 3 * (NP - 1) + NPAGE
0735
0736 C >>>>>> DIAGNOSTIC >> SENSE SWITCH 10 (ON)
0737 IF(ISSW(10) .LT. 0) WRITE(1, 992)NRBLK
0738 992 FORMAT("READING BLOCK NO. ", I4, " FROM DISC")
0739
0740 CALL PUTQ(0, NQUAL)
0741 CALL KYBD(2HMS,31,NRBLK)
0742 CALL KYBD(2HMS,11,0)
0743
0744 C - - - - - MOVE DATA FROM CORE BLOCK 0 TO MATRIX (4 X 512)
0745 DO 310 L = 1, 2048
0746 CALL GETJ(0, L-1, MAT(L), II)
0747 IF(MAT(L) .NE. 999) GO TO 310
0748 IF(MOD(L-1,4) .EQ. 0) GO TO 320
0749 310 CONTINUE
0750 L = L - 1
0751
0752 C >>>>>> DIAGNOSTIC >> SENSE SWITCH 2 (ON)
0753 320 IF(ISSW(2) .LT. 0) PAUSE 320
0754
0755 C >>>>>> DIAGNOSTIC >> SENSE SWITCH 8 (ON)
0756 IF(ISSW(8) .LT. 0) WRITE(2, 9320) (MAT(J), J = 1, L)
0757 9320 FORMAT(10X, 4I8)
0758
0759 C - - - - - RESET LINE COUNTER, LOCAL RETURN TO 105 OR 122
0760 K = 0
0761 GO TO LRET
0762 C *****# END OF MULTI-PING DISPLAY ROUTINE *****#
0763 END
0764
0765 SUBROUTINE MPGRD(ICOM, LISTBM)
0766 DEC. 29, 1977
0767 C *****#

```

```

0768 C * MULTI-PING PLOT GRID GENERATION ROUTINE *
0769 C *****
0770 *****
0771 *****
0772 *****
0773 *****
0774 *****
0775 *****
0776 *****
0777 *****
0778 *****
0779 *****
0780 *****
0781 *****
0782 *****
0783 *****
0784 *****
0785 *****
0786 *****
0787 *****
0788 *****
0789 *****
0790 *****
0791 *****
0792 *****
0793 *****
0794 *****
0795 *****
0796 *****
0797 *****
0798 *****
0799 *****
0800 *****
0801 *****
0802 *****

      DIMENSION LISTBM(8), ICOM(127), LABEL(2), IDATE(9)

      IF (ISSU(14) .LT. 0) GO TO 208
      THRES = ICOM( 5) * 1E-6
      208  CODE = ICOM( 7)
           MODE = ICOM(12)
           LENGTH= ICOM(13)
           KT1 = ICOM(14)
           KT2 = ICOM(15)
           KT3 = ICOM(16)
           LFRQ = ICOM(17)
           IFST = ICOM(35)
           ILST = ICOM(36)
           IDPLR= ICOM(37)
           JSTRT= ICOM(38)
           JEND = ICOM(39)
           JDIR = ICOM(40)
           KTHR = ICOM(48)
           DO 10 I = 1, 9
           10 IDATE(I) = ICOM(I+100)

      C - - - - - INITIALIZE PLOT PARAMETERS / ERASE SCREEN. - - -
      IRESET = 37B
      IESCAP = 33B
      IERASE = 14B
      CALL CHOUT(IESCAP)
      CALL CHOUT(IERASE)
      CALL D750

      C - - - - - COMPUTE HORIZONTAL SCALING - - -
      DELTI = 1.0 * JEND - JSTRT

```

```

0803 C - - - - LOCATE INITIAL ORIGIN IN X AND Y - - -
0804 NSEPX = 112
0805 NSEPY = 80
0806 IX = 1023 - (8*NSEPX)
0807 IY = 780 - (8*NSEPY)
0808
0809 C - - - - CONSTRUCT VERTICAL GRID - - -
0810 NY = IY - 30
0811 DO 20 J = 0, 8
0812 LX = IX + J * NSEPX
0813 CALL TPLOT(0, LX, IY)
0814 CALL TPLOT(1, LX, 780)
0815
0816 C - - - - GET COORDINATES FOR NUMBERS, LABEL X-AXIS - -
0817 TYARDS = JSTRT + J * DELTI/8.0
0818 NX = LX - 50
0819 IF(J .GE. 8) NX = LX - 70
0820 CALL TPLOT(0, NX, NY)
0821 CALL CHOUT(IRESET)
0822 WRITE(2, 920) TYARDS
0823 920 FORMAT(F5.1)
0824 20 CONTINUE
0825
0826 C - - - - CONSTRUCT HORIZONTAL GRID
0827 NX = IX - 70
0828 DO 30 J = 1, 9
0829 LY = IY + (J-1) * NSEPY
0830 CALL TPLOT(0, IX, LY)
0831 CALL TPLOT(1, 1023, LY)
0832
0833 C - - - - GET COORD. FOR BEAM NUMBERS, LABEL Y-AXIS AREAS - - -
0834 IF(J .GT. 8) GO TO 30
0835 IF(LISTEM(J) .EQ. 0) GO TO 30
0836 NY = LY + 30
0837 CALL TPLOT(0, NX, NY)

```

```

0838      CALL CHOUT(IRESET)
0839      WRITE(2, 930) LISTBM(J)
0840      FORMAT(I4)
0841      30 CONTINUE
0842
0843      C --- -- -- WRITE HEADER DATA FOR PLOT --- --
0844      CALL TPLOT(0, 1, 80)
0845      CALL CHOUT(IRESET)
0846      LABEL(1) = 2HHA
0847      LABEL(2) = 2HLF
0848      IF(IDPLR.EQ. 1) LABEL(1) = 2HFU
0849      IF(IDPLR.EQ. 1) LABEL(2) = 2HLL
0850      WRITE(1, 940) IFST, LABEL(1), LABEL(2), KTHR
0851      FORMAT('FIRST PING NO:', I2, 5X, 'DOPPLER: ', 2A2, 6X,
0852      + 'PLOT THRESHOLD:', I3)
0853      LEN = 2HT2
0854      IF(LNGTH.GE. 1) LEN = 2HT1
0855      LFRE = 2HF2
0856      IF(LFRQ.EQ. 1) LFRE = 2HF1
0857      LTY = 2HCJ
0858      IF(MODE.GE. 1) LTY = 2HFM
0859      WRITE(1, 941) LTY, LEN, LFRE, KODE, KT1, KT2, KT3, IDATE
0860      FORMAT('SIGNAL:', 2X, A2, 2X, A2, 2X, A2, 2X, 'HANNING:', I2,
0861      + 9X, 'TIME: ', I2, ':', I2, ':', I2, 3X, 9A2)
0862
0863      C --- -- -- SAVE PLOT ORIGIN
0864      ICOM(45) = IX
0865      ICOM(46) = IY
0866
0867      C ##### END OF MULTI-PING PLOT GRID ROUTINE #####
0868      END
0869
0870      SUBROUTINE FPGRD(ICOM)
0871      C #####
0872      C * FOUR PART DISPLAY GRID GENERATION ROUTINE *

```

```

0873 C *****
0874 DIMENSION ICOM(127), IX(4), IY(4)
0875
0876
0877 C - - - - - INITIALIZE TABLE OF X AND Y LIMITS OF FOUR GRIDS
0878 DATA IX/60,500,570,1010/,IY/80,410,450,780/
0879 DATA ISEPX/110/,ISEPY/55/
0880
0881 JSTRT = ICOM(38)
0882 JEND = ICOM(39)
0883
0884 C - - - - - INITIALIZE PLOT PARAMETERS, ERASE SCREEN
0885 IRESET = 378
0886 IESCAP = 338
0887 IERASE = 148
0888 CALL CHOUT(IESCAP)
0889 CALL CHOUT(IERASE)
0890 CALL D750
0891
0892 C - - - - - CONSTRUCT VERTICAL GRID
0893 DO 20 I = 0, 4
0894 TYARDS = JSTRT + I * (JEND - JSTRT)/4.0
0895
0896 C >>>>> DIAGNOSTIC >> SENSE SWITCH 10 (ON)
0897 IF(ISSW(10) .LT. 0)WRITE(1, 910) TYARDS, JSTRT, JEND
0898 910 FORMAT(' TYARDS', F10.1, 2I6)
0899
0900 DO 20 J = 1, 3, 2
0901 LX = IX(J) + I * ISEPX
0902 CALL TPLOT(0, LX, IY(1) )
0903 CALL TPLOT(1, LX, IY(2) )
0904 CALL TPLOT(0, LX, IY(3) )
0905 CALL TPLOT(1, LX, IY(4) )
0906
0907 C - - - - - LABEL X AXIS

```

```

0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942

```

```

      MOVEH = 45
      IF(I .EQ. 4) MOVEH = 55
      CALL TPLOT(0, LX-MOVEH, IY(1) - 25)
      CALL CHOUT(IRESET)
      WRITE(1, 920) TYARDS
      CALL TPLOT(0, LX-MOVEH, IY(3) - 25)
      CALL CHOUT(IRESET)
      WRITE(1, 920) TYARDS
20  CONTINUE
920  FORMAT(F5.1)

C --- -- -- -- -- CONSTRUCT HORIZONTAL GRIDS
      DO 30 J = 0, 6
      YDOP = -15 + 5 * J
      YBRG = 360 - J*60
      DO 30 I = 1, 3, 2
      LY = IY(I) + J * ISEPY
      CALL TPLOT(0, IX(1), LY)
      CALL TPLOT(1, IX(2), LY)
      CALL TPLOT(0, IX(3), LY)
      CALL TPLOT(1, IX(4), LY)

C --- -- -- -- -- LABEL Y-AXIS
      CALL TPLOT(0, IX(3)-45, LY-8)
      CALL CHOUT(IRESET)
      WRITE(1, 930) YDOP
      CALL TPLOT(0, IX(1)-45, LY-8)
      CALL CHOUT(IRESET)
      IF(I .EQ. 1) WRITE(1, 930) YBRG
      IF(I .EQ. 3) WRITE(1, 930) YDOP
30  CONTINUE
930  FORMAT(I3)
C ##### END OF FOUR PART DISPLAY GRID GENERATION ROUTINE #####
      END

```

```

0943 SUBROUTINE PLLIN (IX, INCRX, IY, IDTA, MAX)
0944 *****
0945 * THIS SUBROUTINE PLOTS A VERTICAL BAR WITH BASE AT IX, IY. *
0946 * HEIGHT IS 2*IDTA. *
0947 *****
0948
0949 DATA IRESET/37B/
0950
0951 IF(IDTA .GT. 10) IDTA = 10
0952 MY = MIN0(MAX, IY + IDTA + IDTA)
0953 INCR = INCRX * 0.4991 + 0.5
0954 IENDX = MIN0(1019, IX + INCR - 4)
0955 CALL TPLOT(0, IX, IY)
0956 DO 15 LX = IX, IENDX, 4
0957 CALL TPLOT(1, LX+2, MY)
0958 CALL TPLOT(1, LX+4, IY)
0959 15 CONTINUE
0960 CALL CHOUT(IRESET)
0961 C ##### END OF THE LINE PLOT ROUTINE #####
0962 END
0963
0964 SUBROUTINE BRDB
0965 *****
0966 * DUMP DATA BLOCK SUBROUTINE *
0967 *****
0968
0969 DIMENSION IBUF(20), KQUAL(5), NQUAL(5)
0970
0971 DATA NQUAL/2048,100B,16383,45,0/
0972
0973 DATA NDBLK/0/,KBLK/0/,KOLS/10/,LINES/100/
0974 KTDAT = 0
0975 WRITE(1, 910)
0976 FORMAT(' ENTER DISC BLOCK NO. (-1 IF NONE)')
0977 910 CALL INPT(NDBLK, -1, 33)

```



```

0978 WRITE(1, 911)
0979 FORMAT(' ENTER CORE BLOCK NUMBER (0 - 3)')
0980 CALL INPT(KBLK, 0, 3)
0981 WRITE(1, 912)
0982 FORMAT(' NO. OF COLUMNS FOR PRINTOUT (1 - 10)')
0983 CALL INPT(KOLS, 1, 10)
0984 WRITE(1, 913)
0985 FORMAT(' ENTER NO. OF LINES FOR PRINTOUT')
0986 CALL INPT(LINES, 1, 2048)
0987 IF(NDBLK .LT. 0) GO TO 10
0988
0989 C - - - - - DISC BLOCK INTO CORE
0990 CALL GETQ(KBLK, KQUAL)
0991 CALL PUTQ(KBLK, NQUAL)
0992 CALL KYBD(ZHMS, 31, NDBLK)
0993 CALL KYBD(ZHMS, 11, KBLK)
0994 CALL ENDIO
0995
0996 C - - - - - WRITE DATA FROM CORE DATA BLOCK
0997 10 DO 30 L = 0, LINES-1
0998    DO 20 K = 0, KOLS -1
0999        KTDAT = KTDAT + 1
1000        NUORD = K + L * KOLS
1001        CALL GETQ(KBLK, NUORD, IBUF(K+1), 11)
1002    20 CONTINUE
1003        LK = L+1
1004        WRITE(1, 930)LK, (IBUF(I), I = 1, KOLS)
1005    30 CONTINUE
1006        FORMAT(1X, 11I6)
1007    930 CALL ENDIO
1008        CALL PUTQ(KBLK, KQUAL)
1009
1010 C > > > RETURN TO CALLER
1011 RETURN
1012 C ##### END OF DUMP DATA BLOCK SUBROUTINE #####

```

```

1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047

END

SUBROUTINE PTIME(IPNG, ICOM)

*****
* THE PTIME SUBROUTINE CONVERTS PING TIME FROM BCD UNITS *
* (OBTAINED FROM THE HEADER WRITTEN DURING DATA COLLECTION) *
* INTO BINARY FORM. RESULTS ARE STORED IN THE PROPER ICOM *
* LOCATIONS. *
*****
C *****

DIMENSION ICOM(127), MASK(4), IBCD(6), IHDR(16)

DATA MASK/17B, 360B, 7400B, 170000B/

C > > > GET PING TIME FROM HEADER
C CALL HEADR (IPNG, IHDR)

C > > > COMPUTE HOURS
IBCD(5) = IAND (MASK(1), IHDR(1))
IBCD(6) = IAND (MASK(2), IHDR(1))
ICOM(14) = IBCD(6) / 20B * 10 + IBCD(5)

C > > > COMPUTE MINUTES AND SECONDS
DO 10 I = 1,4
10 IBCD(I) = IAND (MASK(I), IHDR(2))

ICOM(15) = IBCD(4) / 10000B * 10 + IBCD(3) / 400B
ICOM(16) = IBCD(2) / 20B * 10 + IBCD(1)

C > > > RETURN TO CALLER
RETURN

C ***** END OF PING TIME SUBROUTINE *****

```

```
1048 C *****  
1049 C ***** END OF PLOT DISPLAY SUBROUTINE *****  
1050 C *****  
1051 C *****  
1052 C *****  
      END  
      ENDS
```

:

5.3 OLAAS PROCESSING & DISPLAY SYSTEM SUBROUTINES

The OLAAS System subroutines are used in conjunction with the major functional programs/subroutines as part of the modular concept used to design the system.

The following descriptions, flow diagrams, and listings are the subroutines used in the system:

- ASKQ Subroutine
- CDUMP Subroutine
- CLOAD Subroutine
- DMAL Subroutine
- D75 ϕ Subroutine
- INPT Subroutine
- INPUT Subroutine
- ISHFT Subroutine
- LINK6 Subroutine
- LKTBG Subroutine
- RDATA Subroutine
- TAT1 Subroutine
- TEK Subroutine
- TYPI Subroutine
- USTRT Subroutine

QUESTIONS/ANSWERS FOR Y1002 SUBROUTINE (ASKQ)

1. FUNCTION. The Question/Answer(ASKQ) subroutine is used to place the program operation parameter question on the screen with the previous parameter values given. User enters new parameters or uses the previous parameter values. Data entered is checked for range errors.

2. CONSTRAINTS. All input variables must be integers. This subroutine is used only with the Y1002.

3. CALLING SEQUENCE. Called by: Y1002.

Call ASKQ (IVAR, IMIN, IMAX, IQUE), IDATE

IVAR = Variable

IMIN = Minimum value allowed

IMAX = Maximum value allowed

IQUE = Question number

IDATE = Date

4. DESCRIPTION OF INPUT. Input is a typed value as requested by the displayed question.

5. DESCRIPTION OF OUTPUT. Output is the variables described in the calling sequence, in integer format. Printed output is the operator prompts.

6. FILES USED. N/A.

7. ERROR AND RESTRICTIONS. A message is output to the screen indicating an "out of range" value was entered. The subroutine then waits for a correct input value.

8. COMPUTER OPERATOR INSTRUCTIONS. When the ASKQ subroutine is called during normal system operation, the operator is prompted by a question appearing on the screen. The operator replies by entering a value through the keyboard and pressing a carriage return.

9. DESCRIPTION OF PROCESSING. ASKQ subroutine at (1) initializes the printer buffer length variable, IBUFL, to the number of characters to be output in each question. The print buffer area, IPBUF, is cleared each time the subroutine is entered.

At (2), determine the question entry point base on the question number parameter passed from the calling subroutine, Y1002. Each of the thirteen (13) questions are handled in the same manner.

i.e., A call to the Fortran IV library routine, CODE, causes an internal conversion in the Fortran Formatter and allows the screen's cursor to remain at the end of the displayed message. The requested question is set-up and control is transferred to (3).

At (3), the Print Question, PNTL, subroutine is called to output the question to the screen. Note that the last entered value or the initialized value of the selected question is displayed with the question.

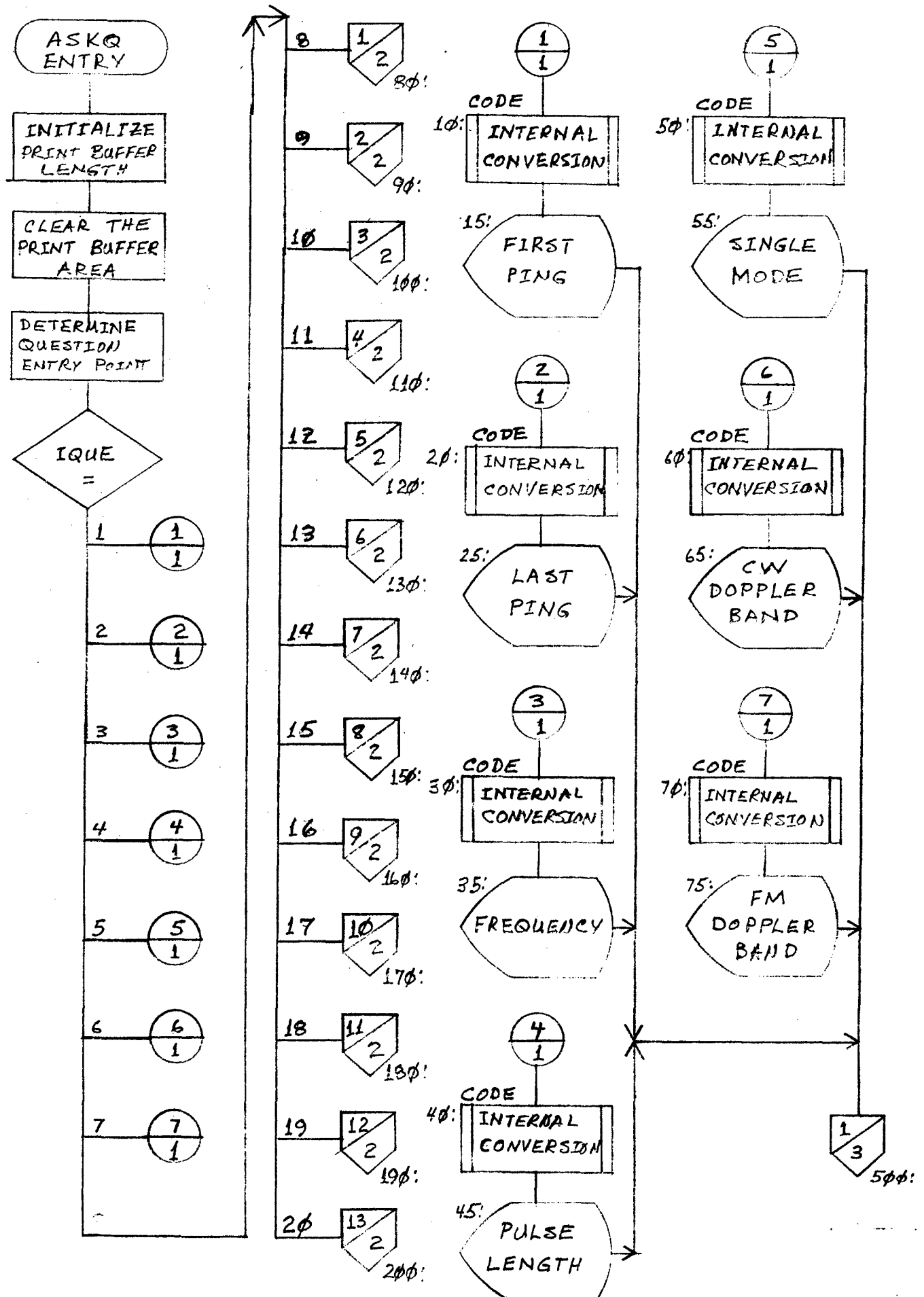
At (4), the keyboard input subroutine, TYPI, is called to allow the User Operator to enter an answer to the question. The three (3) possible answers are as follows:

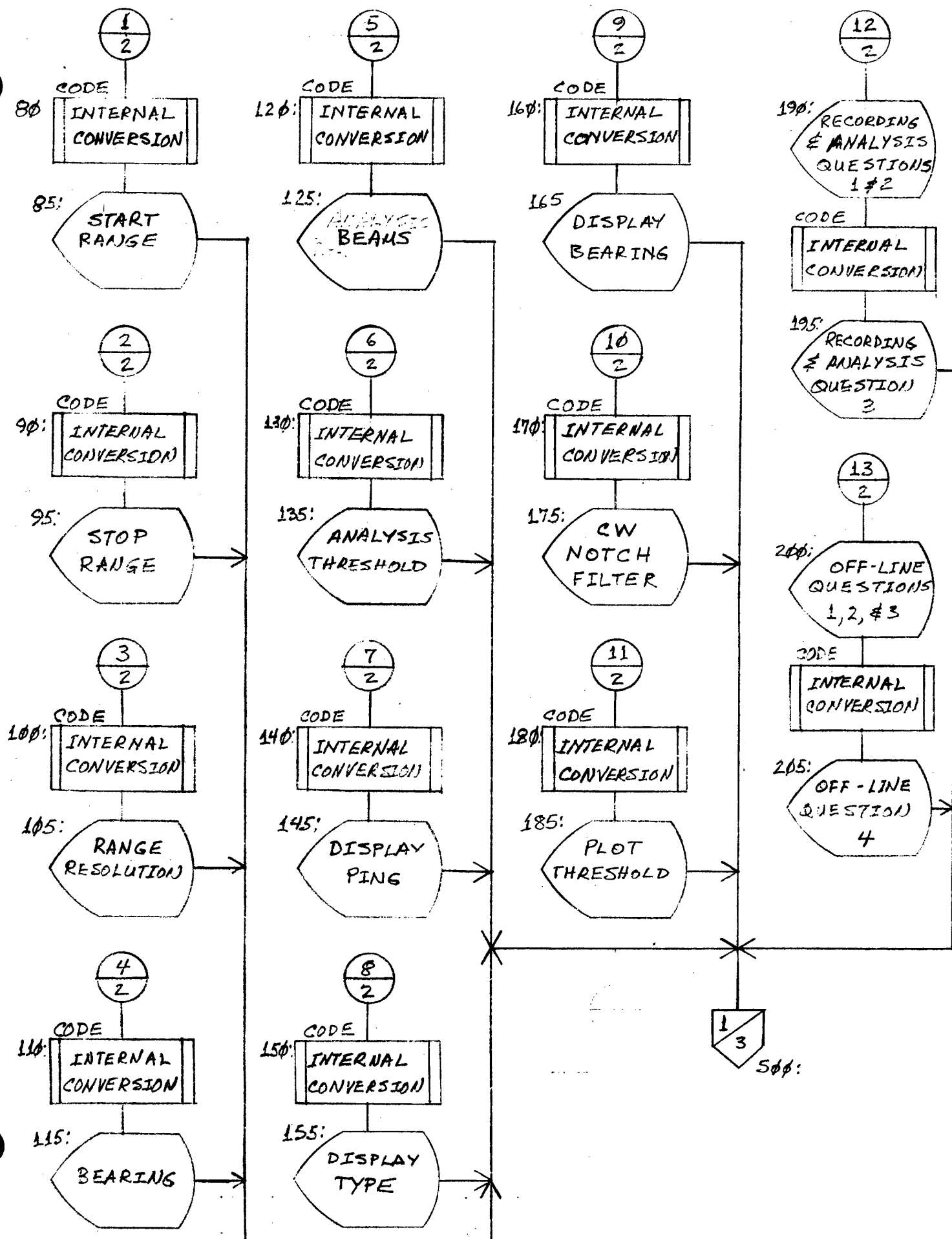
- A. Carriage Return = Use the same value now displayed on the screen (last entered value).
- B. 12345 = Abort the operation and return to Coreload Ø, OLAAS Executive Program.
- C. Enter new value = This is the new parameter that is to be used in the calling program.

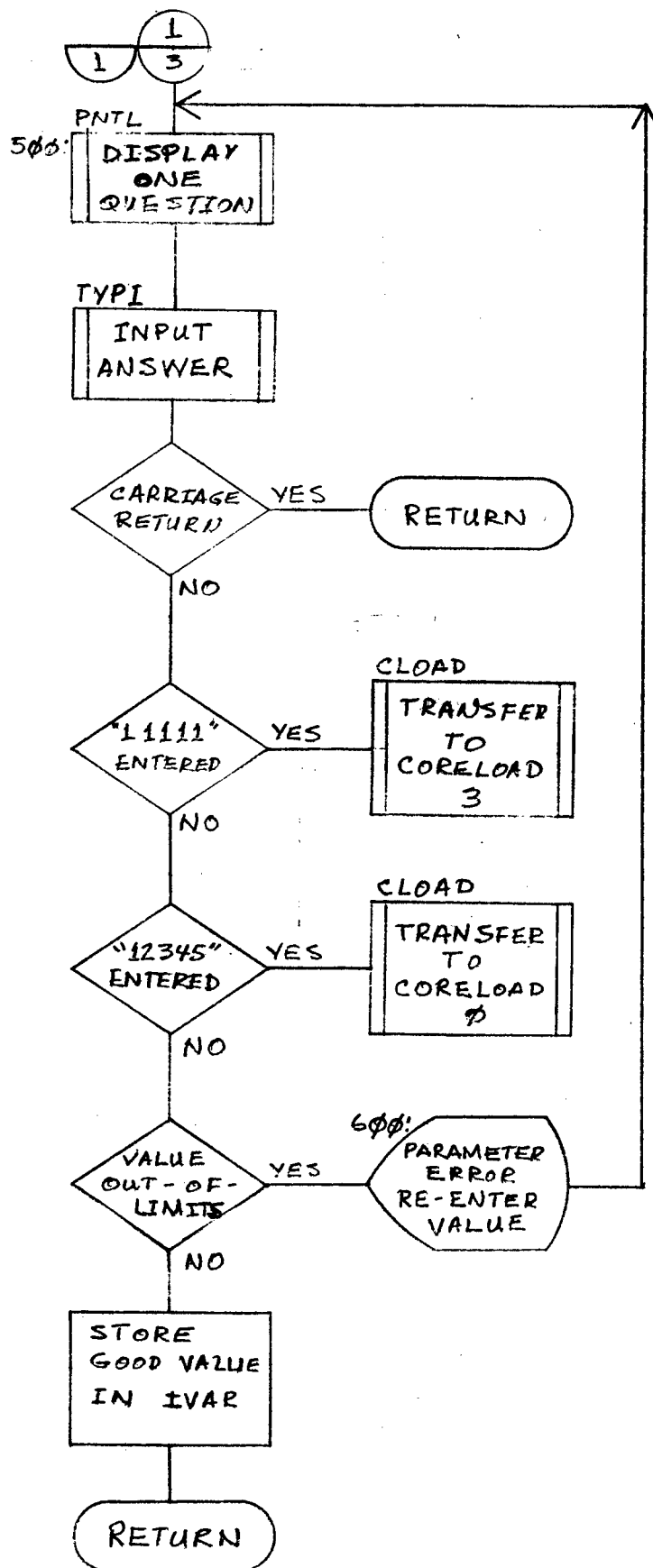
At (5), the new value entered is checked for an "out-of-range" error condition. If the value is out of range a message is displayed on the screen and control is passed back to output the question again and wait for the next input. The subroutine will repeat in the loop until a good value is entered.

At (6), the new value is loaded into the variable parameter, IVAR, and the ASKQ subroutine exits returning control back to the calling subroutine, Y1ØØ2.

ANALYSIS QUESTION/ANSWER SUBROUTINE (ASKQ)







ASKQ T=00004 IS ON CP00001 USING 00039 BLKS P=0160

FTN4

SUBROUTINE ASKQ (IVAR, IMIN, IMAX, IQUE, IDATE)

TITLE: Y1002 SUBROUTINE QUESTION/ANSWER PROGRAM

MODULE: ASKQ SOURCE: ASKQ VERSION: X01
OBJECT: #ASKQ

AUTHOR: W. BOWEN DATE: 25 JULY 1977

MODIFICATIONS:

1: ADD DISPLAY QUESTIONS
(INDIVIDUAL BEAM, MULTI-PING, & FOUR PART DISPLAY)
12/13/77 W. BOWEN

MODULE FUNCTION:

THE ASK QUESTION (ASKQ) SUBROUTINE IS USED TO PLACE
THE PROGRAM OPERATION PARAMETER QUESTIONS ON THE SCREEN
THE PREVIOUS PARAMETER VALUES GIVEN. USER ENTERS NEW
PARAMETERS OR USES THE PREVIOUS PARAMETER VALUES.
ENTRY DATA IS CHECKED FOR RANGE ERRORS.

CALLING SEQUENCE: CALLED BY - Y1002

CALL ASKQ (IVAR, IMIN, IMAX, IQUE)

INPUT PARAMETERS:

IVAR = VARIABLE
IMIN = MINIMUM VALUE ALLOWED
IMAX = MAXIMUM VALUE ALLOWED
IQUE = QUESTION NUMBER

0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032

```

0033 C * IDATE = DATE ENTERED
0034 C *
0035 C * OUTPUT PARAMETERS: IVAR = VARIABLE
0036 C *
0037 C *****
0038 C *****
0039 C *****
0040 C * DEFINE ARRAYS AND VARIABLES *
0041 C *****
0042 C *****
0043 C DIMENSION IPBUF(32), IDATE(9)
0044 C
0045 C IBUFL = 64
0046 C
0047 C *****
0048 C * INITIALIZE SUBROUTINE *
0049 C *****
0050 C *****
0051 C > > > CLEAR THE PRINT BUFFER AREA
0052 C DO 5 I=1,32
0053 C IPBUF(I) = 020040B
0054 C 5 CONTINUE
0055 C > > > DETERMINE QUESTION ENTRY POINT
0056 C GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,
0057 C +140,150,160,170,180,190,200) IQUE
0058 C
0059 C *****
0060 C * ANALYSIS & DISPLAY PARAMETER QUESTIONS *
0061 C *****
0062 C *****
0063 C > > > NUMBER OF PINGS
0064 C 10 CALL CODE
0065 C WRITE (IPBUF,15) IVAR
0066 C 15 FORMAT ("FIRST PING",9X, "NUMBER FOR ANALYSIS (1 TO 10)",
0067 C +7X, "(,15,")",2X)

```

```

0068      GO TO 500
0069
0070      C
0071      20 CALL CODE
0072      WRITE (IPBUF,25) IVAR
0073      25 FORMAT ("LAST PING",10X,"NUMBER FOR ANALYSIS (1 TO 10)",
0074      +7X,"(",15,")",2X)
0075      GO TO 500
0076      C > > > FREQUENCY (F1 OR F2)
0077      30 CALL CODE
0078      WRITE (IPBUF,35) IVAR
0079      35 FORMAT ("FREQUENCY",10X,"F1 = 1 OR F2 = 2",18X,
0080      + "(" ,15,")",2X)
0081      GO TO 500
0082      C > > > PULSE LENGTH (SHORT OR LONG)
0083      40 CALL CODE
0084      WRITE (IPBUF,45) IVAR
0085      45 FORMAT ("PULSE LENGTH",7X,"SHORT = 2 OR LONG = 5",13X,
0086      + "(" ,15,")",2X)
0087      GO TO 500
0088      C > > > SIGNAL MODE (CW, FM UP OR FM DOWN)
0089      50 CALL CODE
0090      WRITE (IPBUF,55) IVAR
0091      55 FORMAT ("SIGNAL MODE",8X,"CW = 0, FM UP = 1, OR FM DOWN"
0092      + " = 2",3X,"(",15,")",2X)
0093      GO TO 500
0094      C > > > CW DOPPLER BAND (FULL, HALF, UP, OR DOWN)
0095      60 CALL CODE
0096      WRITE (IPBUF,65) IVAR
0097      65 FORMAT ("CW DOPPLER BAND",4X,"FULL=1, HALF=2, UP=3, "
0098      + "OR DOWN=4",5X,"(",15,")",2X)
0099      GO TO 500
0100      C > > > FM DOPPLER BAND (FULL OR HALF)
0101      70 CALL CODE
0102      WRITE (IPBUF,75) IVAR
0103      75 FORMAT ("FM DOPPLER BAND",4X,"FULL = 1 OR HALF = 2",

```

```

0103      +14X, "(" ,15,")",2X)
0104      GO TO 500
0105      C > > > START RANGE
0106      80 CALL CODE
0107      WRITE (IPBUF,85) IVAR
0108      85 FORMAT ("START RANGE",8X,"IN MULTIPLES OF 10 KILOYARDS",
0109      +8X, "(" ,15,")",2X)
0110      GO TO 500
0111      C > > > STOP RANGE
0112      90 CALL CODE
0113      WRITE (IPBUF,95) IVAR
0114      95 FORMAT ("STOP RANGE",9X,"IN MULTIPLES OF 10 KILOYARDS",
0115      +8X, "(" ,15,")",2X)
0116      GO TO 500
0117      C > > > RANGE RESOLUTION
0118      100 CALL CODE
0119      WRITE (IPBUF,105) IVAR
0120      105 FORMAT ("RANGE RESOLUTION",3X,"IN MULTIPLES OF 160 YARDS",
0121      +11X, "(" ,15,")",2X)
0122      GO TO 500
0123      C > > > BEARING
0124      110 CALL CODE
0125      WRITE (IPBUF,115) IVAR
0126      115 FORMAT ("BEARING",12X,"ANGLE IN DEGREES (CENTER BEAM)",
0127      +6X, "(" ,15,")",2X)
0128      GO TO 500
0129      C > > > BEAMS
0130      120 CALL CODE
0131      WRITE (IPBUF,125) IVAR
0132      125 FORMAT ("BEAMS",14X,"NUMBER OF BEAMS",21X, "(" ,15,")",2X)
0133      GO TO 500
0134      C > > > THRESHOLD
0135      130 CALL CODE
0136      WRITE (IPBUF,135) IVAR
0137      135 FORMAT ("ANALYSIS THRESHOLD",1X,"0 TO 10000"

```

```

0138      + " (LEVEL IN MICRO-VOLTS)", 3X, "(", ", 15, ")", 2X)
0139      GO TO 500
0140      C > > > DISPLAY PING NUMBER (INDIVIDUAL BEAM DISPLAY)
0141      140 CALL CODE
0142      WRITE (IPBUF, 145) IVAR
0143      145 FORMAT ("DISPLAY PING", 7X, "PING NUMBER TO BE DISPLAYED",
0144      + 9X, "(", ", 15, ")", 2X)
0145      GO TO 500
0146      C > > > DISPLAY TYPE (INDIVIDUAL BEAM DISPLAY)
0147      150 CALL CODE
0148      WRITE (IPBUF, 155) IVAR
0149      155 FORMAT ("DISPLAY TYPE", 7X, "2 = INDIVIDUAL 5 = BEAMS ",
0150      + "OR'ED", 4X, "(", ", 15, ")", 2X)
0151      GO TO 500
0152      C > > > DISPLAY BEARING (FOUR PART DISPLAY)
0153      160 CALL CODE
0154      WRITE (IPBUF, 165) IVAR
0155      165 FORMAT ("DISPLAY BEARING", 4X, "BEARING TO BE DISPLAYED ",
0156      + "(0 TO 359)", 2X, "(", ", 15, ")", 2X)
0157      GO TO 500
0158      C > > > CW NOTCH FILTER
0159      170 CALL CODE
0160      WRITE (IPBUF, 175) IVAR
0161      175 FORMAT ("CW NOTCH FILTER", 4X, "WIDTH IN FREQUENCY CELLS"
0162      + " (0 TO 25)", 1X, "(", ", 15, ")", 2X)
0163      GO TO 500
0164      C > > > PLOT THRESHOLD
0165      180 CALL CODE
0166      WRITE (IPBUF, 185) IVAR
0167      185 FORMAT ("PLOT THRESHOLD", 5X, "1 TO 100 (%) (1= DISPLAY ALL)"
0168      + 6X, "(", ", 15, ")", 2X)
0169      GO TO 500
0170      C > > > RECORDING & ANALYSIS QUESTIONS
0171      190 WRITE (2, 193) IDATE
0172      193 FORMAT ("***** OLAAS DATA RECORDING & ANALYSIS "

```

```

0173 +*****"12X,9A2/5X,"** ENTER DESIRED FUNCTION **"/
0174 +"1 = ON-LINE DATA RECORDING & ANALYSIS"/
0175 +"2 = OFF-LINE ANALYSIS & PLOT DISPLAY"/
0176 +"3 = INITIALIZE CORELOAD 3 ONLY"/
0177 +"4 = OPERATOR: SET SWITCH 1 (ON) TO ASK THESE"
0178 +" THESE QUESTIONS:"/4X,"BUOY FREQUENCY, PULSE LENGTH,"
0179 +" STOP RANGE, & CW NOTCH FILTER")
0180 CALL CODE
0181 WRITE (IPBUF,195) IVAR
0182 195 FORMAT ("5 = DATE ENTRY",41X,"(",15,")",2X)
0183 GO TO 500
0184 C > > > OFF-LINE QUESTIONS
0185 200 WRITE (2,203) IDATE
0186 203 FORMAT ("***** OFF-LINE DATA ANALYSIS & PLOT DISPLAY"
0187 + " *****"/16X,9A2/5X,"** SELECT OFF-LINE OPTION **"/
0188 +"1 = ANALYSIS OF PREVIOUSLY RECORDED DATA"/
0189 +"2 = INDIVIDUAL BEAM DISPLAY"/
0190 +"3 = MULTI-PING DISPLAY")
0191 CALL CODE
0192 WRITE (IPBUF,205) IVAR
0193 205 FORMAT ("4 = FOUR PART DISPLAY",34X,"(",15,")",2X)
0194 GO TO 500
0195 C
0196 C *****
0197 C * CALL PRINT LINE (PNTL) SUBROUTINE TO PRINT QUESTIONS *
0198 C *****
0199 C *****
0200 500 CALL PNTL (IPBUF,IBUFL)
0201 C
0202 C *****
0203 C * CALL TO INPUT PARAMETERS FROM THE KEYBOARD *
0204 C *****
0205 C *****
0206 CALL TYPI (IN,I1)
0207 C > > > RETURN IF CARRIAGE RETURN ONLY

```

```

0208 IF (I1.EQ.0) RETURN
0209 C > > REPEAT THE LAST QUESTION ASKED (IN = 11111)
0210 C > > IF (IN.EQ.11111) GO TO 500
0211 C > > CHECK FOR RETURN TO EXECUTIVE (IN = 12345)
0212 C > > IF (IN.EQ.12345) CALL CLOAD (1,2)
0213 C > > CHECK IF DATA IS OUT OF RANGE (IMIN TO IMAX)
0214 C > > IF (I1.LT.0 .OR. IN.GT. IMAX .OR. IN.LT. IMIN)
0215 C > > +GO TO 600
0216 C
0217 C *****
0218 C * STORE GOOD VALUE IN IVAR AND RETURN *
0219 C *****
0220 C *****
0221 C IVAR = IN
0222 C RETURN
0223 C
0224 C *****
0225 C * ERROR DETECTED, GET NEW INPUT *
0226 C *****
0227 C *****
0228 C 600 WRITE (2,605)
0229 C 605 FORMAT ("*****X PARAMETER ERROR *****X RE-")
0230 C + "ENTER VALUE")
0231 C GO TO 500
0232 C
0233 C >>>>>> END OF SUBROUTINE ASKO >>>>>>
0234 C END
0235 C ENDS$

```

:

CORE DUMP ROUTINE - CDUMP

1. FUNCTION. CDUMP dumps all of core (locations 2-77577) onto six consecutive disc tracks specified in the calling sequence.
2. CONSTRAINTS. NIBBL must be in core in locations 77600-77677. It must be the unmodified HP version, as it is modified by CDUMP.
3. CALLING SEQUENCE. Called by: Y0001, Y0002

CALL CDUMP (ICYL)

where ICYL is the starting cylinder number of the desired coreload.

4. DESCRIPTION OF INPUT. None.
5. DESCRIPTION OF OUTPUT. Unknown.
6. FILES USED. N/A
7. ERRORS AND RESTRICTIONS. An irrecoverable error halt will occur if a disc transfer error or a disc status error occurs.
8. COMPUTER OPERATOR INSTRUCTIONS. N/A
9. DESCRIPTION OF PROCESSING. CDUMP uses NIBBL, the Disc Basic Binary Loader which reads or writes an entire coreload. However, NIBBL is a manually operated program and normally halts upon completion rather than return to a calling program. Therefore, CDUMP modifies NIBBL to cause it to return. Processing is as follows: CDUMP turns off interrupts, picks up CDUMP's return address, stores it in address 5, and stores a jump to location 5 in NIBBL where the completion halt usually is. As there are no unused locations in NIBBL in which to store an off-page pointer, a base page pointer must be used. Location 5 is a convenient location as it is reserved for parity error and memory protect interrupts, neither of which is used in this system. The cylinder number is picked up from the calling sequence, placed in the B register, bit 15 is set, and NIBBL is entered, with A=0. NIBBL will dump all of core up through address 77577 and return to the address in 5.

CDUMP

transfer
parameters

interrupts
off

A = 0
B = return
address

(5) = B

Store JMP 5
over halt
in NIBBL

B = cylinder
number from
calling
sequence

add bit 15

enter NIBBL
at 77600

```

0001
0003* THIS SUBROUTINE DUMPS THE 32X CORE LOAD TO THE CYLINDER
0004* SPECIFIED. IT WORKS BY REPLACING THE HALT IN NIBBL WITH A
0005* RETURN AND THEN JUMPING TO NIBBL. THE RETURN IS IN LOCATION
0006* 5 AND THEREFORE THE HALT ON PARITY ERROR IS DISABLED.
0007* NIBBL MUST BE IN CORE FOR THIS TO WORK. IT IS NORMALLY BROUGHT
0008* IN BY THE BOOT AT 77750.
0009*
0010*
0011*
0012*
0013*
0014*
0015*
0016*
0017*
0018*
0019*
0020*
0021*
0022*
0023*
0024*
0025*
0026*
0027*
0028*
0029*
0030*
0031*
0032*
0033*
0034*
0035*
0036*
0037*
0038*
0039*
0040*
0041*
0042*
0043*
0044*
0045*
0046*
0047*
0048*
0049*
0050*
0051*
0052*
0053*
0054*
0055*
0056*
0057*
0058*
0059*
0060*
0061*
0062*
0063*
0064*
0065*
0066*
0067*
0068*
0069*
0070*
0071*
0072*
0073*
0074*
0075*
0076*
0077*
0078*
0079*
0080*
0081*
0082*
0083*
0084*
0085*
0086*
0087*
0088*
0089*
0090*
0091*
0092*
0093*
0094*
0095*
0096*
0097*
0098*
0099*
0100*

```

```

0001
0003* THIS SUBROUTINE DUMPS THE 32X CORE LOAD TO THE CYLINDER
0004* SPECIFIED. IT WORKS BY REPLACING THE HALT IN NIBBL WITH A
0005* RETURN AND THEN JUMPING TO NIBBL. THE RETURN IS IN LOCATION
0006* 5 AND THEREFORE THE HALT ON PARITY ERROR IS DISABLED.
0007* NIBBL MUST BE IN CORE FOR THIS TO WORK. IT IS NORMALLY BROUGHT
0008* IN BY THE BOOT AT 77750.
0009*
0010*
0011*
0012*
0013*
0014*
0015*
0016*
0017*
0018*
0019*
0020*
0021*
0022*
0023*
0024*
0025*
0026*
0027*
0028*
0029*
0030*
0031*
0032*
0033*
0034*
0035*
0036*
0037*
0038*
0039*
0040*
0041*
0042*
0043*
0044*
0045*
0046*
0047*
0048*
0049*
0050*
0051*
0052*
0053*
0054*
0055*
0056*
0057*
0058*
0059*
0060*
0061*
0062*
0063*
0064*
0065*
0066*
0067*
0068*
0069*
0070*
0071*
0072*
0073*
0074*
0075*
0076*
0077*
0078*
0079*
0080*
0081*
0082*
0083*
0084*
0085*
0086*
0087*
0088*
0089*
0090*
0091*
0092*
0093*
0094*
0095*
0096*
0097*
0098*
0099*
0100*

```

```
0010* CALLING SEQUENCE: JSB CDUMP
0011* DEF *+2 DEF CYLNO
0012*
0013*
```

CALL CDUMP (CYLNO)
-OR-

0016	00000		NAM	CDUMP
0017			ENT	CDUMP
0018			EXT	.ENTR
0019	00000	CYLN0	NOP	
0020	00001	CDUMP	NOP	
0021	00002		JSB	.ENTR
0022	00003		DEF	CYLN0
0023	00004		CLF	0
0024	00005		CLA	
0025	00006		LDB	CDUMP
0026	00007		STB	5
0027	00010		LDB	JMP5
0028	00011		STB	HLTAD, I
0029	00012		LDB	CYLN0, I

```

0030 00013 046020R
0031 00014 126016R
0032 00015 124005
0033 00016 077600
0034 00017 077657
0035 00020 100000
0036

      ADB BIT15
      JMP NIBBL,I
      JMP 5,I
      JMP5 OCT 77600
      NIBBL OCT 77657
      HLTAD OCT 100000
      BIT15 OCT 100000
      END
** NO ERRORS *TOTAL **RTE ASMB 750420**

```

```

/ASMB: $END

```

CORE LOAD ROUTINE - CLOAD

1. FUNCTION. CLOAD loads all of core (locations 2-77577) from six consecutive disc tracks specified in the calling sequence, and transfers control to a location in the new core load specified in the calling sequence.

2. CONSTRAINTS. NIBBL must be in core in locations 77600-77677. It must be the unmodified HP version, as it is modified by CLOAD. The transfer address must be in base page.

3. CALLING SEQUENCE. Called by: Y0001, Y0002, Y0003

CALL CLOAD (ICYL, IXFER)

where ICYL is the starting cylinder number of the desired coreload
IXFER is the transfer address in the new coreload.

4. DESCRIPTION OF INPUT. Unknown.

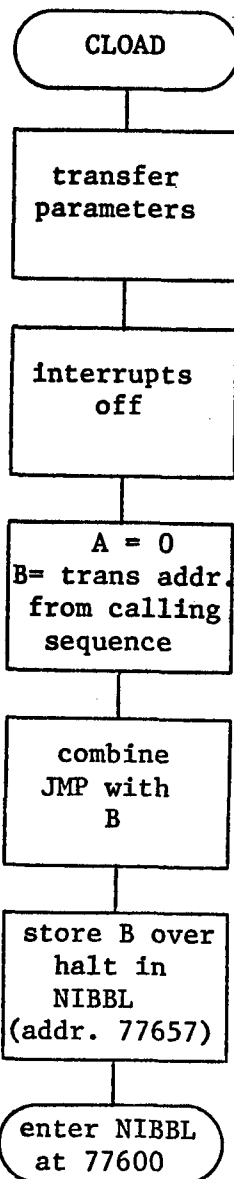
5. DESCRIPTION OF OUTPUT. None.

6. FILES USED. N/A

7. ERRORS AND RESTRICTIONS. An irrecoverable error will occur if a disc transfer error or a disc status error occurs.

8. COMPUTER OPERATOR INSTRUCTIONS. N/A

9. DESCRIPTION OF PROCESSING. CLOAD uses NIBBL, the Disc Basic Binary Loader which reads or writes an entire coreload. However, NIBBL is a manually operated program and normally halts upon completion rather than return to a calling program. Also, since all of core up to NIBBL will be overlaid by the new coreload, it would be impossible to return control to CLOAD. Therefore, CLOAD modifies NIBBL to return to an address specified in the CLOAD calling sequence. Processing is as follows: CLOAD turns off interrupts, combines the transfer address in the calling sequence with a JMP instruction, and stores this in NIBBL where the completion halt usually is. The cylinder number is picked up from the calling sequence, placed in the B register, and NIBBL is entered, with A=0. NIBBL will read the desired coreload and transfer control to the specified address.



PAGE 0002 #01 CORE LOAD SUBROUTINE

2001 ASMB,R,L
 0003* THIS SUBROUTINE BRINGS IN THE 32K CORE LOAD WHICH BEGINS
 0004* AT THE CYLINDER SPECIFIED AND STARTS EXECUTION AT ADDRESS 2.
 0005* IT WORKS BY REMOVING THE HALT IN NIBBL AND THEN JUMPING TO IT.
 0006* NIBBL MUST BE IN CORE FOR THIS TO WORK. IT IS NORMALLY BROUGHT
 0007* IN BY THE BOOT AT 77750.
 0008*

CALLING SEQUENCE:

JSB CLOAD
 DEF X+2
 DEF CYLNO
 DEF TRANS

-OR-

CALL CLOAD (CYLNO,TRANS)

NAM CLOAD
 ENT CLOAD
 EXT .ENTR

CYLNO
 TRANS
 CLOAD

JSB .ENTR
 DEF CYLNO
 CLF 0

CLA LDB JMP
 ADB TRANS,I
 STB HLTAD,I
 LDB CYLNO,I

00000

00000 000000
 00001 000000
 00002 000000
 00003 016001X
 00004 000000R
 00005 103100
 00006 002400
 00007 066014R
 00010 146001R
 00011 176016R
 00012 166000R

0001
 0003*
 0004*
 0005*
 0006*
 0007*
 0008*
 0009*
 0010*
 0011*
 0012*
 0013*
 0014*
 0015*
 0016
 0017
 0018
 0019
 0020
 0021
 0022
 0023
 0024
 0025
 0026
 0027
 0028
 0029

0030 00013 126015R JMP NIBBL,I
0031 00014 024000 JMP 0
0032 00015 077600 NIBBL OCT 77600
0033 00016 077657 HLTAD OCT 77657
0034 END
XX NO ERRORS *TOTAL **RTE ASMB 750420XX

/ASMB: SEND

DMA1

This assembly language program is used to transfer major frame ERAPS active data, under the control of Y7896. DMA is used in the data transfer from interface 23B.

FORTTRAN calling sequence:

CALL DMA1 (IDAT, KK)

where IDAT = 72 word array

KK =	{	1	Turn off interrupts, turn on DMA. Return to main program.
		2	Wait for DMA complete. Transfer 72 word major frame to IDAT. Return to main program.
		3	Wait for DMA complete, turn on interrupts then return to main program.

DMA1S T=00004 IS ON CR00001 USING 00003 BLKS R=00034

0001 ASMB,R,L
0002 HED DMA1
0003 NAM DMA1
0004 ENT DMA1
0005 EXT .ENTR

0006 *
0007 IDAT NOP
0008 KK NOP
0009 DMA1 NOP
0010 JSB .ENTR
0011 DEF IDAT
0012 LDA KK,I
0013 CPA =D2
0014 JMP KK2
0015 CPA =D3
0016 JMP KK3

*

0018 CLF 0
0019 LDA BUF1A
0020 JSB DMA1
0021 STC SC,C
0022 STC S,C
0023 CLA,INA
0024 OTR SC
0025 JMP DMA1,I

*

0026 *
0027 KK2 SFS 6
0028 JMP *-1
0029 CLC SC
0030 CLC 6
0031 LDA BUF2A
0032 LDB BUF1A

ENTRY - FORTRAN CALLABLE

KK=1, INTERRUPTS OFF
START INPUT
INIT DMA

START DMA

OUTPUT A 1 TO START
RETURN

KK=2, SWAP BUFFERS AND CONTINUE

SWAP BUFFERS

0033
 0034
 0035
 0036
 0037
 0038
 0039
 0040
 0041
 0042
 0043
 0044
 0045
 0046
 0047
 0048
 0049
 0050
 0051
 0052
 0053
 0054
 0055
 0056
 0057
 0058
 0059
 0060
 0061
 0062
 0063
 0064
 0065
 0066
 0067

STA BUF1A
 STB BUF2A
 JSB DMAI
 STC SC,C
 STC B,C
 LDB =D-72
 LDA BUF2A
 STA PTR1
 LDA IDAT
 STA PTR2
 LDA PTR1,I
 STA PTR2,I
 ISZ PTR1
 ISZ PTR2
 ISZ B
 JMP MUNXT
 JMP DMA1,I
 SFS B
 JMP *-1
 CLC SC
 CLC B
 CLA
 OTA SC
 STF 0
 JMP DMA1,I
 NOP
 STA B
 LDA SLCT2
 IOR =B120000
 OTA B
 CLC 2
 LDA B
 IOR =B100000

MUNXT

*

KK3

*

DMAI

START DMA
 MOVE LAST SET OF DATA

MOVE NEXT
 RETURN

KK=3. WAIT FOR DMA, STOP

INTERRUPTS ON
 RETURN

INIT DMA SUBROUTINE

```

0068      OTA 2
0069      STC 2
0070      LDA =D-72
0071      OTA 2
0072      JMP DMAI, I
0073
0074      *PTR1 NOP
0075      PTR2 NOP
0076      BUF1A DEF BUF1
0077      BUF2A DEF BUF2
0078      BUF1 BSS 72
0079      BUF2 BSS 72
0080      SLOC2 DEF SC
0081      SC EQU 23B
0082      A EQU 0
0083      B EQU 1
0084      END

```

(0-4)

D750S T-00004 IS ON CR00001 USING 00002 BLKS R-0016

0001	ASMB,R,L		
0002	NAM	D750	
0003	ENT	D750	
0004	D750	NOP	
0005		ISZ	D750
0006		LDA	K1
0007		STA	CNTR
0008	DLY1	LSL	16
0009		LSL	16
0010		LSL	16
0011		ISZ	CNTR
0012		JMP	DLY1
0013		JMP	D750,I
0014	K1	DEC	-29435
0015	CNTR	NOP	
0016		END	

:

USER DATA ENTRY PROGRAM - INPT

1. FUNCTION. INPT provides a means to enter new analysis parameters or to use previously stored values in the analysis program Y1002.
2. CONSTRAINTS. All input variables must be integers.
3. CALLING SEQUENCE.

Call INPT (IVAR,IMIN,IMAX)

where: IVAR is parameter value
IMIN is minimum permissible value
IMAX is maximum permissible value

4. DESCRIPTION OF INPUT. Input is a typed value entered by operator on keyboard, or a carriage return.
5. DESCRIPTION OF OUTPUT. Printed output in parentheses is previously stored value. Upon carriage return or entry of new value, value is passed to calling program.
6. FILES USED. N/A
7. ERRORS AND RESTRICTIONS. Error and range checking is provided by the subroutine.
8. COMPUTER OPERATOR INSTRUCTIONS. N/A.
9. DESCRIPTION OF PROCESSING. The purpose of the subroutine INPT is to read in analysis constraints, if within the proper range of values of IMIN and IMAX, or to use a previously entered value, if a carriage return is hit.

At (1), the previously entered value is written on the CRT screen in parentheses.

At (2), the subroutine TYPI is used to fetch information entered from the keyboard. Two arguments are passed, the new value IN, and a parameter I1. I1 is equal to zero if a carriage return has been hit, indicating the user elected to use the previous value, and positive, if the user entered a new value IN.

At (3), the condition of a carriage return being hit is checked, and a return to the calling program is made if it is true.

At (4), the new value entered, IN, is checked for equivalence to 12345. If so, a transfer is made to the executive program in coreload 0.

At (5), the newly entered value, IN, is checked to be within the range of IMIN and IMAX. If out of range, an error message is written and the program branches to (2) to accept a new entry. If no errors are detected, the value IVAR is set to IN, at (6). The subroutine then returns to the calling program.

INPTS T-00004 IS ON CR00001 USING 00004 BLKS R-00036

```
0001 FTN4,L
0002 SUBROUTINE INPT(IVAR,IMIN,IMAX)
0003 C
0004 C THIS ROUTINE IS USED TO BUFFER USER-ENTERED DATA
0005 C FOR USE IN THE OLAS PROGRAM. DATA IS CHECKED FOR
0006 C ERRORS IN ENTRY AND IF NO DATA IS ENTERED, THE PREVIOUS
0007 C VALUE OF THE PARAMETER WILL BE USED.
0008 C
0009 WRITE(2,100) IVAR
0010 100 FORMAT(" ",15," ")
0011 5 CALL TYPI(IN,I1)
0012 C
0013 C RETURN IF CARRIAGE RETURN ONLY
0014 C
0015 C IF(I1 .EQ.0) RETURN
0016 C
0017 C CHECK FOR RETURN TO EXECUTIVE (IN= 12345)
0018 C
0019 C IF(IN .EQ. 12345)CALL CLOAD(1,2)
0020 C
0021 C CHECK FOR BAD DATA
0022 C
0023 C IF(I1 .LT.0 .OR. IN .GT. IMAX .OR. IN .LT. IMIN) GO TO 30
0024 C
0025 C STORE NEW GOOD VALUE AS IVAR
0026 C
0027 C IVAR=IN
0028 C RETURN
0029 C
0030 C ERROR DETECTED, GET NEW INPUT
0031 C
0032 30 WRITE(2,150)
```


INPTA T-00004 IS ON CR00001 USING 00004 BLKS R-0036

```
0001 FTN4,L
0002 SUBROUTINE INPT(IVAR,IMIN,IMAX)
0003 C
0004 C THIS ROUTINE IS USED TO BUFFER USER-ENTERED DATA
0005 C FOR USE IN THE OLAAAS PROGRAM. DATA IS CHECKED FOR
0006 C ERRORS IN ENTRY AND IF NO DATA IS ENTERED, THE PREVIOUS
0007 C VALUE OF THE PARAMETER WILL BE USED.
0008 C
0009 WRITE(2,100) IVAR
0010 100 FORMAT(" ",IS,"")
0011 5 CALL TYPI(IN,I1)
0012 C
0013 C RETURN IF CARRIAGE RETURN ONLY
0014 C
0015 C IF(I1 .EQ.0) RETURN
0016 C CHECK FOR BAD DATA
0017 C
0018 C IF(I1 .LT.0 .OR. IN .GT. IMAX .OR. IN .LT. IMIN) GO TO 30
0019 C
0020 C STORE NEW GOOD VALUE AS IVAR
0021 C
0022 IVAR=IN
0023 RETURN
0024 C
0025 C ERROR DETECTED, GET NEW INPUT
0026 C
0027 30 WRITE(2,150)
0028 150 FORMAT("ERROR, REENTER VALUE")
0029 GO TO 5
0030 END
0031 ENDS
```

DATA COLLECTION PROGRAM OPERATOR INPUT ROUTINE - INPUT

1. FUNCTION. INPUT provides conversational input and error checking of operator parameters for the data collection program.
2. CONSTRAINTS. All input variables must be integers.
3. CALLING SEQUENCE. Called by: Y0001

```
JSB INPUT
DEF *+7
DEF N
DEF K
DEF L
DEF M
DEF D
DEF S
```

where N is pulse time (1-240)
K is delay (5-60)
L is number of beams (1-16)
M is number of pings (1-100)
D is desired direction (0-360)
S is source (0-1)

4. DESCRIPTION OF INPUT. Input is typed values in the order described in the calling sequence.
5. DESCRIPTION OF OUTPUT. Output is the variables described in the calling sequence, in integer format. Printed output is the operator prompts.
6. FILES USED. N/A
7. ERRORS AND RESTRICTIONS. None.
8. COMPUTER OPERATOR INSTRUCTIONS. See general system operating instructions.
9. DESCRIPTION OF PROCESSING. INPUT requests input of each variable by a message on the CRT. TYPI is called to input and convert the value from the keyboard. If TYPI returns an error flag, or no data is entered, or the value is out of range, the prompt is given again. Each variable is input in sequence by the same method.

MPINP

T=00004 IS ON CR00001 USING 00029 BLKS R=0160

FTN4

0001

0002

0003

0004

0005

0006

0007

0008

0009

0010

0011

0012

0013

0014

0015

0016

0017

0018

0019

0020

0021

0022

0023

0024

0025

0026

0027

0028

0029

0030

0031

0032

SUBROUTINE INPUT (N,K,L,M,D,S)

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

C * * * * *

TITLE: INPUT QUESTION/ANSWER SUBROUTINE FOR THE
OLAAS DATA COLLECTION SUBROUTINE

MODULE: INPUT SOURCE: MPINP VERSION: X01
OBJECT: #MPINP

AUTHOR: R. TASHJIAN DATE: 1976

MODIFICATIONS:

1: CHANGE THE QUESTION FORMATS.
12/14/77 W. BOWEN

MODULE FUNCTION:

THE INPUT QUESTION/ANSWER (INPUT) SUBROUTINE IS USED
TO PLACE THE DATA COLLECTION PARAMETER QUESTIONS ON THE
SCREEN. THE PREVIOUS PARAMETER VALUES ARE GIVEN. THE
USER ENTERS NEW PARAMETERS OR USES THE PREVIOUS
PARAMETER VALUES. ENTRY DATA IS CHECKED FOR RANGE
ERRORS.

CALLING SEQUENCE: CALLED BY - Y0001

CALL INPUT (N,K,L,M,D,S)

INPUT/OUTPUT PARAMETERS: N - PULSE TIME (AFTER K)
K - DELAY BEFORE RECORD
L - NUMBER BEAMS TO RECORD

```

0033 * M - TOTAL NUMBER OF PINGS *
0034 * D - DESIRED DIRECTION OF *
0035 * CENTER BEAM *
0036 * S - SOURCE (1 - ACTIVE ON-LINE *
0037 * 0 - TAPE RECORDER *
0038 * *
0039 *****
0040 *****
0041 *****
0042 * DEFINE ARRAYS AND VARIABLES *
0043 *****
0044 *****
0045 DIMENSION IPBUF(32)
0046
0047 INTEGER D,S
0048
0049 *****
0050 * INITIALIZE SUBROUTINE *
0051 *****
0052 *****
0053 C > > > CLEAR THE PRINT BUFFER AREA
0054 IBUFL =64
0055 DO 5 I=1,32
0056 IPBUF(I) = 020040B
0057 5 CONTINUE
0058 WRITE (2,8)
0059 8 FORMAT (/5X,"** RAW DATA RECORDING QUESTIONS **")
0060
0061 C > > > DETERMINE QUESTION ENTRY POINT
0062 IQUE = 0
0063 100 IQUE = IQUE + 1
0064 GO TO (10,20,30,40,50,60) IQUE
0065
0066 *****
0067 * DATA COLLECTION PARAMETER QUESTIONS *

```

```

0068 C *****
0069 C *****
0070 C > > > START RANGE
0071 10 IMIN = 5
0072 IMAX = 35
0073 MSTRT = 8
0074 CALL CODE
0075 WRITE (IPBUF,15) MSTRT
0076 15 FORMAT ("START RANGE",8X,"IN KILOYARDS (5 TO 35)"
0077 +13X,"(",15,")",2X)
0078 GO TO 500
0079
0080 C > > > STOP RANGE
0081 20 IMIN = MSTRT
0082 IMAX = 40
0083 MEND = 24
0084 CALL CODE
0085 WRITE (IPBUF,25) MEND
0086 25 FORMAT ("STOP RANGE",9X,"IN KILOYARDS (MAXIMUM 40)"
0087 +10X,"(",15,")",2X)
0088 GO TO 500
0089
0090 C > > > CONVERT TO CORRESPONDING TIME IN SECONDS
0091
0092 30 X = MSTRT * 1.2
0093 K = X + 0.5
0094 X = (MEND - MSTRT) * 1.2 + 3.0
0095 N = X + 0.5
0096
0097 C > > > NUMBER OF BEAMS TO RECORD
0098 IMIN = 1
0099 IMAX = 16
0100 L = 16
0101 CALL CODE
0102 WRITE (IPBUF,35) L

```

```

0103      35 FORMAT ("BEAMS",14X,"NUMBER OF BEAMS TO RECORD",
0104      +11X,"(",15,")",2X)
0105      GO TO 500
0106
0107      C > > > NUMBER OF PINGS TO RECORD
0108      40 IMIN = 1
0109      IMAX = 100
0110      M = 1
0111      CALL CODE
0112      WRITE (IPBUF,45) M
0113      45 FORMAT ("NUMBER PINGS",7X,"NUMBER OF PINGS TO RECORD",
0114      +11X,"(",15,")",2X)
0115      GO TO 500
0116
0117      C >>>>>> SKIP NEXT QUESTION IF ALL BEAMS ARE RECORDED >>>>>>
0118      50 IF (L.EQ. 16) GO TO 100
0119
0120      C > > > DESIRED DIRECTION OF CENTER BEAM
0121      IMIN = 0
0122      IMAX = 359
0123      D = 0
0124      CALL CODE
0125      WRITE (IPBUF,55) D
0126      55 FORMAT ("CENTER BEAM",8X,"DESIRED CENTER BEAM DIRECTION",
0127      +7X,"(",15,")",2X)
0128      GO TO 500
0129
0130      C > > > DATA SOURCE
0131      60 IMIN = 0
0132      IMAX = 1
0133      S = 1
0134      CALL CODE
0135      WRITE (IPBUF,65) S
0136      65 FORMAT ("DATA SOURCE",8X,"1-ACTIVE ON-LINE / 0-",
0137      + "TAPE RECORDER",2X,"(",15,")",2X)

```

```

0138      GO TO 500
0139
0140 *****
0141 * CALL PRINT LINE (PNTL) SUBROUTINE TO PRINT QUESTIONS *
0142 *****
0143
0144 500 CALL PNTL (IPBUF,IBUFL)
0145
0146 *****
0147 * CALL TO INPUT PARAMETERS FROM THE KEYBOARD *
0148 *****
0149
0150      CALL TYPI (IN,I1)
0151
0152      C > > > RETURN IF CARRIAGE RETURN ONLY
0153      IF (I1 .EQ. 0) GO TO 510
0154
0155      C > > > CHECK FOR RETURN TO EXECUTIVE (IN = 12345)
0156      IF (IN .EQ. 12345) CALL CLOAD (1,2)
0157
0158      C > > > CHECK IF DATA IS OUT OF RANGE (IMIN TO IMAX)
0159      IF (I1 .LT. 0 .OR. IN .GT. IMAX .OR. IN .LT. IMIN) GO TO 600
0160
0161 *****
0162 * STORE GOOD VALUE IN PARAMETER VARIABLES *
0163 *****
0164
0165      IF (IQUE .EQ. 1) MSTRT = IN
0166      IF (IQUE .EQ. 2) MEND = IN
0167      IF (IQUE .EQ. 3) L = IN
0168      IF (IQUE .EQ. 4) M = IN
0169      IF (IQUE .EQ. 5) D = IN
0170      IF (IQUE .EQ. 6) S = IN
0171      510 IF (IQUE .NE. 6) GO TO 100
0172

```


ISHFTS T-00003 IS ON CR00001 USING 00003 BLKS R-0000

```

0001 ASMB,R,L
0002 HED
0003 HED FUNCTION ISHFT(I,N)
0004 NAM ISHFT
0005 ENT ISHFT
0006 EXT .ENTR
0007 *
0008 * THIS FORTRAN FUNCTION SHIFTS I RIGHT N BITS WITH SIGN EXTEND.
0009 *
0010 I NOP
0011 N NOP
0012 ISHFT NOP
0013 JSB .ENTR
0014 DEF I
0015 LDA N,I
0016 AND -B17
0017 IOR -B101020 MERGE INTO ASR INSTRUCTION
0018 STA ASR
0019 LDB I,I
0020 ASR 16
0021 LDA B
0022 JMP ISHFT,I RETURN
0023 EQU 0
0024 EQU 1
0025 END

```

:

PAGE 0002 #01 TIME BASE GENERATOR INTERRUPT LINKAGE

0001		ASMB,R,L		
0003	00000	NAM LKTBG	R.E.T.	8-4-76
0004		ENT LKTBG		
0005		EXT TBGI		
0006	00000 016001X	LKTBG JSB TBGI		
0007		END		
**	NO ERRORS *TOTAL **RTE ASMB 750420**			

PAGE 0002 #01 BEAM FORMER (DMA) INTERRUPT LINKAGE

```
0001          ASMB,R,L
0003          000000      NAM LINK6      R.E.T. 8-4-76
0004          ENT LINK6
0005          EXT BFIT
0006* THIS PROVIDES A BASE PAGE LINK TO THE BEAM FORMER (DMA)
0007* COMPLETION PROCESSOR.
0008 00000 016001X LINK6 JSB BFIT
0009      END
** NO ERRORS *TOTAL **RTE ASMB 750420**
```

RETRIEVE DATA SUBROUTINE - RDATA

1. FUNCTION. RDATA retrieves data stored by Y0001 from the disc as specified by arguments in the call. A separate entry point, HEADR, retrieves the header data for a specified ping.
2. CONSTRAINTS. Although validity checks are made on ping and beam number and number of words, no check is made on the index specification, nor is any check made on whether the data was actually recorded.
3. CALLING SEQUENCE.

CALL RDATA (IRPING,IBEAM,INDXP,NWDS,ISTAT)

where IPING is the ping number (1-?)
PBEAM is the beam number (1-16)
INDXP is the index (0-?)
NWDS is the number of words to be read (1-4096)
ISTAT is the location for the status word

INDXP specifies the first word to be read. A change in INDXP of 1 corresponds to 128 words. Therefore INDXP=5 indicates that the reading is to start with word 512. ISTAT is set to zero if data is returned, non-zero if no data is returned. Data is stored in Data Block 0 (zero).

CALL HEADR (IPING,IDATA)

where IPING is the ping number
IDATA is a 16 word array where the header data is to be stored.

4. DESCRIPTION OF INPUT. Input is the information written on disc by the data collection program, Y0001. See Figure 6 in the Y0001 description for format.
5. DESCRIPTION OF OUTPUT. Output is the requested number of words of data for the specified ping. All beam data is placed in Data Block 0 (zero). This data is a series of 2's complement, fixed-point, single-word variables of like scaling. Output from subroutine HEADR is as follows. All data items are integer format, with the exception of time, which is BCD.

Word	1	TIME	time code, hours, BCD
	2		time code, minutes and seconds, BCD
	3	ANGL1	heading of beam 1, degrees

Word	4		not used
	5	FSTB#	number of first beam recorded
	6	FBANG	heading of first beam recorded, degrees
	7	PING	ping number
	8	#BMS	number of beams recorded
	9	S	source (1=on line, 0=recorder)
	10	TOTPG	total number of pings (also known as M)
	11	K	delay before record, seconds
	12	N	length of record, seconds
	13		reserved
	14-16		not used

6. FILES USED. N/A

7. ERRORS AND RESTRICTIONS. None.

8. COMPUTER OPERATOR INSTRUCTIONS. N/A

9. DESCRIPTION OF PROCESSING. RDATA first transfers parameters from the calling sequence and turns interrupts off. It sets the status parameter bad so that if no data is read, it will be set. If HEADR has never been called, certain parameters essential for locating the beam data will not have been calculated; INIT=0 indicates this. If INIT is zero, HEADR is called to read the ping 1 header into the local buffer and calculate the number of sectors per block (SPB) and per ping (SPP). INIT is set to 1.

At (2) the caller's ping number is checked for legality against the maximum ping number given in the header record. It should be noted that the header record being referenced at this point may be any header record, but maximum ping number is the same for all. Next, the requested ping number is checked against the number of the ping in the buffer. If different, HEADR is called again to read the requested ping header into the local buffer. It is always necessary to have the proper ping's header in core before reading beam data, as the number of the first beam (FSTB#) is needed in order to determine the position of the requested beam. The ping number check before reading the header avoids unnecessary disc accesses when many requests are made for the same ping.

At (3) the requested beam number is checked for being between FSTB# and FSTB#+#BMS-1. Some special processing is required due to the "wraparound" nature of the beam numbers; i.e., beam 1 is adjacent to beam 16.

Beginning at (4) the starting sector for the requested ping, beam, and index is computed. This also takes into account that there is a header record (1 sector) at the beginning of each ping. This point of the subroutine makes use of a special sector number, SSECT, which does not reset to zero at the beginning of each track. Therefore for track 0 sector 0, SSECT would be 0, and for track 1 sector 0, SSECT would be 48. SSECT is also relative and does not account for the actual first track on the disc. SSECT will later be converted to actual track and sector by RDISC. The computation of SSECT also includes computation of OFST, the offset within a block of 4 sectors of the same beam. This is necessary because unless the index specifies starting at a certain location the first disc read will not include a full 4 sectors. The counter CNTR is initialized to DTADO, which is within the Fourier system.

Finally at (5) the number of words to the end of the first block (RWDS) is calculated; this will be the number of words in the read request. If RWDS is greater than the number of words remaining to be read (-CNTR) the smaller number replaces RWDS. RDISC is called to read RWDS words directly into DTADO at the starting location given by PNTR. PNTR and CNTR are updated by the number of words read, RWDS. If CNTR is then still not zero, the special sector SSECT is updated to the beginning of the next block of the applicable beam. Since the next read will begin at the beginning of the 4 sector block, OFST is cleared. Control is then transferred back to (5) to read the next block. When CNTR finally reaches zero, the status indicator is set to zero (good), interrupts are turned on and control is returned to the caller.

The HEADR subroutine is called both by the user program to read the header into a user array, and by RDATA to read the header into the local buffer. As a result, the interrupt system status (on for user, off for RDATA) is unknown at entry. Therefore RDATA first determines this condition and saves it in INTST, then turns the interrupt system off. The data location from the calling sequence is set up in the RDISC calling sequence.

At (6) INIT is checked. If zero, there has been no header read into the local buffer, which is necessary to locate other data on the disc. In this case, SSECT is set to zero to specify the beginning of ping 1, which is the header, RDISC is called to read 16 words at SSECT into the local header buffer, and INIT is set to 1 to eliminate the reading of ping 1 header in the future.

At (7) the number of sectors per block

$$SPB = \#BMS * 4$$

and number of sectors per ping

$$SPP = SPB * BLOCK + 1$$

are computed. The 1 in the second equation accounts for the header at the beginning of each ping. The starting special sector (SSECT) for the requested ping, which is the header location for that ping, is computed and RDISC is called to read 16 words, the header, from that location into the user's buffer. Finally, HEADR restores the interrupt system condition and returns.

Subroutine RDISC gets its information on disc location from the special sector indicator, SSECT, as explained in the RDATA description. RDATA first converts this to a relative track and sector by dividing by 48. The quotient is stored in STRAK and the remainder is stored in SECTO. STRAK is relative to the first allocated track on the disc. SECTO has a range of 0-47, with the numbers 24-47 representing sectors 0-23 on the upper surface, as described for WDISC (a part of Y0001). The actual track (TRAK) is then computed based on STRAK and the Track Allocation Table entries: FTLD, LTLD, FTUD, and LTUD. If the final track number is greater than LTUD or there is no disc space allocated, an error exit is taken. Along with the track calculations, the head base (HEADB) is set up according to the disc in use. HEADB corresponds to the lower head for the disc in use. DISCR will use a head one greater if SECT specifies the upper surface.

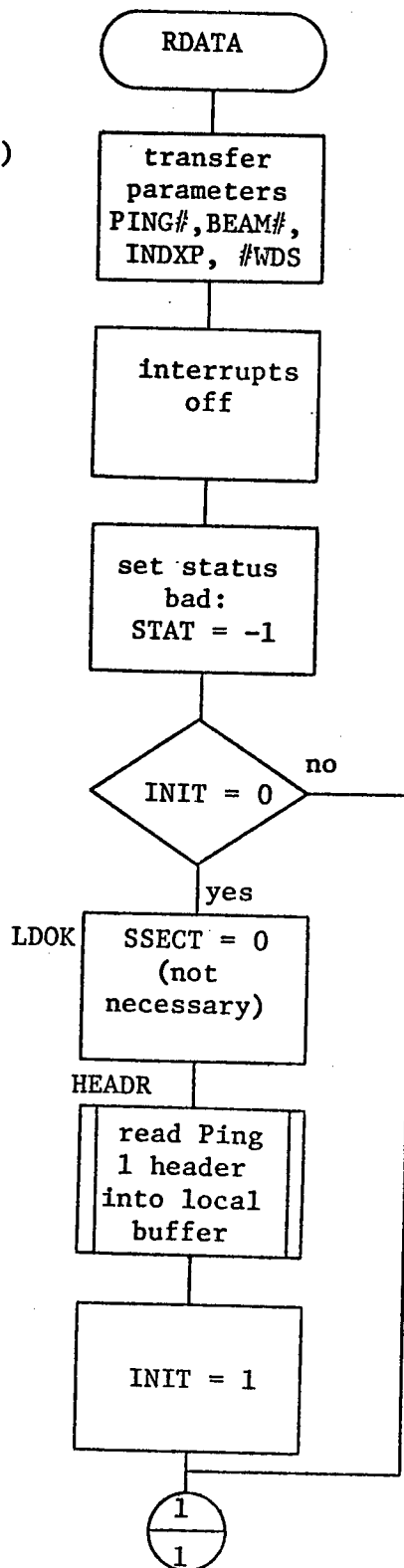
At (8) the number of words remaining on the track, starting at SECT, is calculated. This is significant because a disc access can not cross a track boundary; i.e., head motion commands must be explicitly given by the program. This is stored in AA. If RCTR is smaller, RCTR is stored in AA, and the access will not be to end of track. DISCR is called to perform the disc read, which will be AA words from TRAK and SECT on the disc specified by HEADB, into the location stored at RPTR.

At (9) SECT is updated. If the access just completed was to end of track, SECT will now be 48 and RDISC will reset SECT to zero and increment TRAK. If TRAK is greater than MXTRK, TRAK and MXTRK are set to the upper disc track parameters, FTUD and LTUD, respectively. If HEADB is zero, the disc in use was already the upper disc, so an error exit is taken. Otherwise, HEADB is set to zero.

At (10) the pointer and word counter, RPTR and RCTR are updated by the word counter, RPTR and RCTR are updated by the word count in the last access, AA. If RCTR is then zero, all disc access requested by RDISC has been completed, and it returns to the calling program. Otherwise, control branches back to (8) for another read.

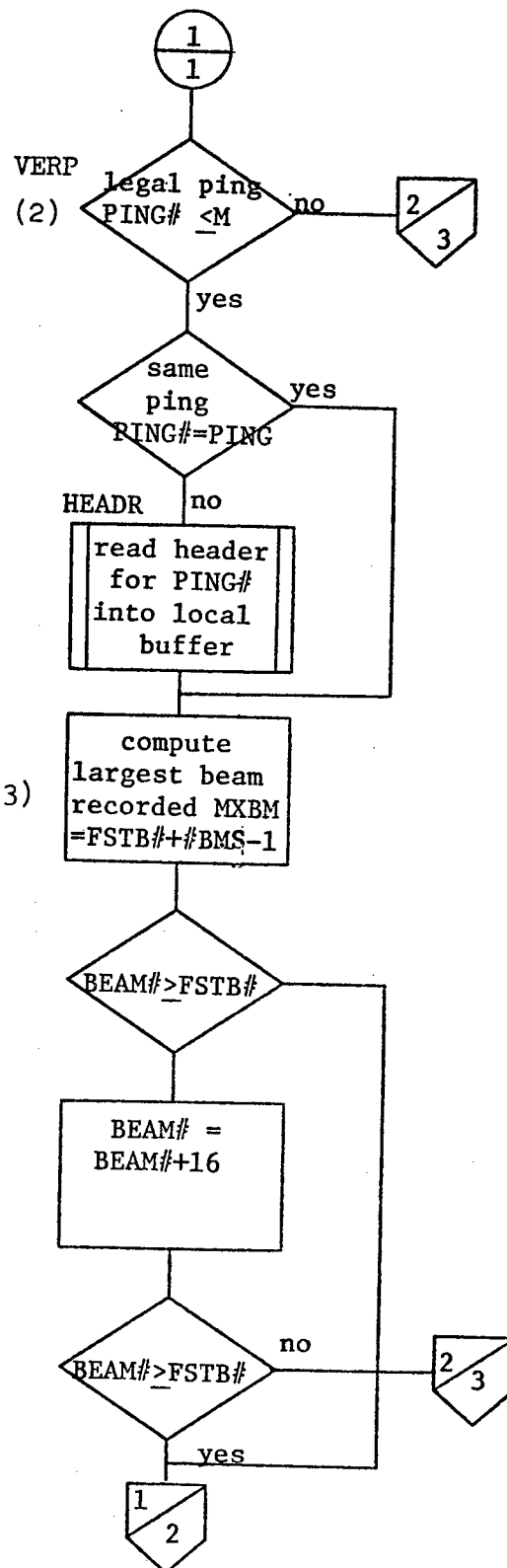
DISCR, called by RDISC, first converts SECT (range 0-47) and HEADB to HDSEC, the control word for head and sector which uses sector in the range 0-23. This is shifted to the bit positions required by the disc controller. DMA is set up to read AA words starting at the location in RPTR. The remainder of the processing is as specified in the disc interface manual. It should be noted that the order of commands for disc read is different from disc write, therefore there are some significant differences from WRITD. If status following the read is bad, an error path is taken.

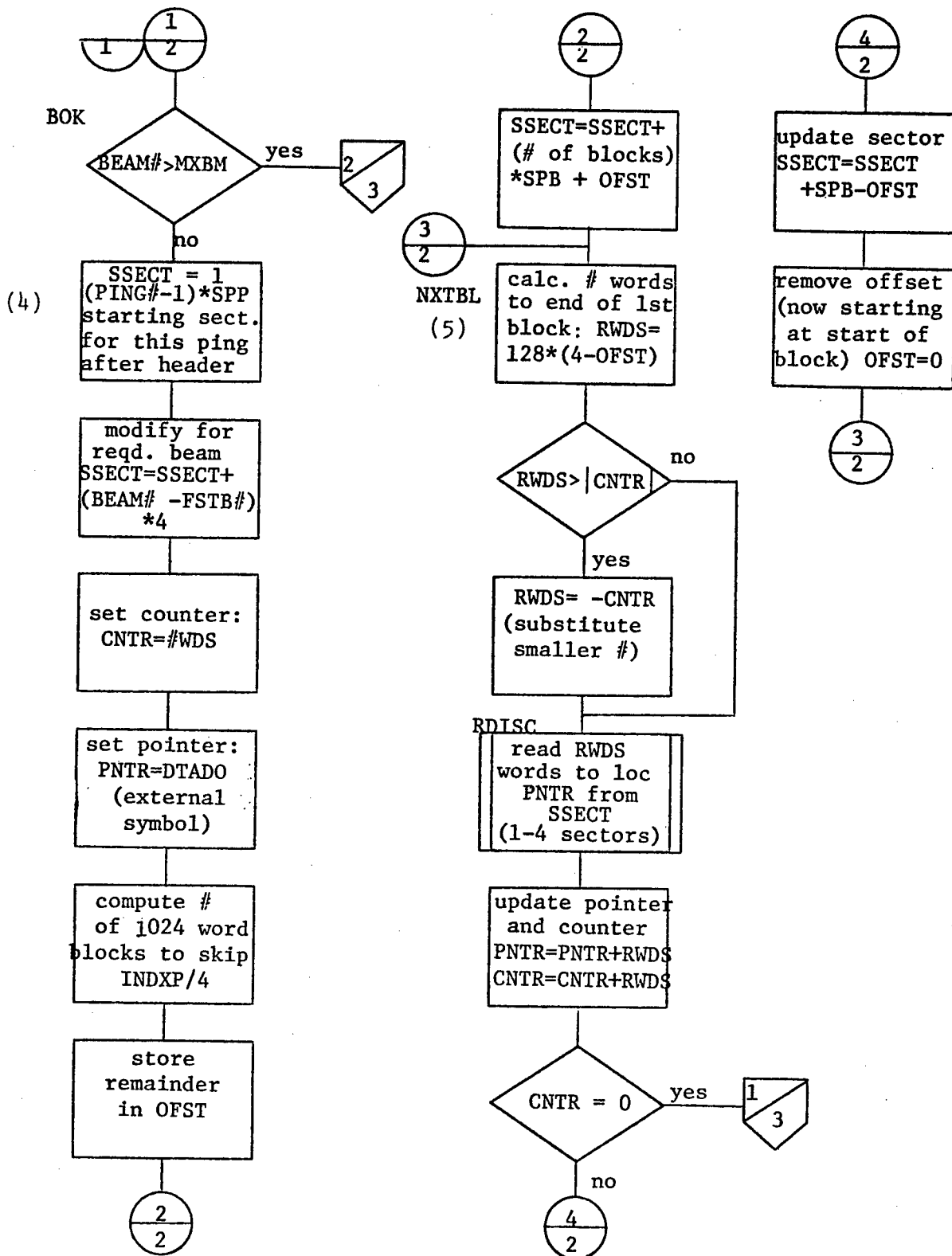
(1)

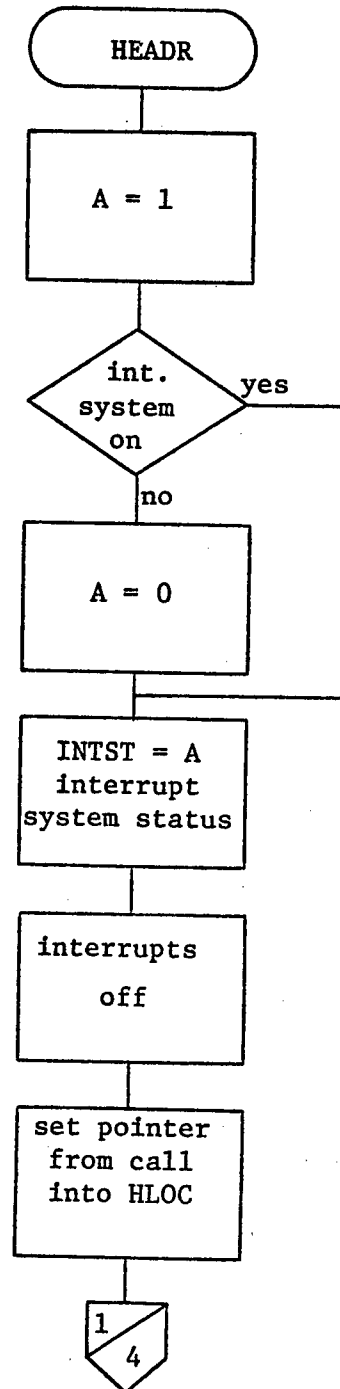
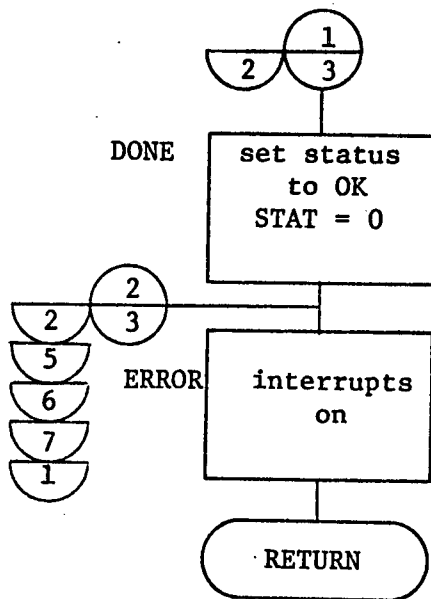


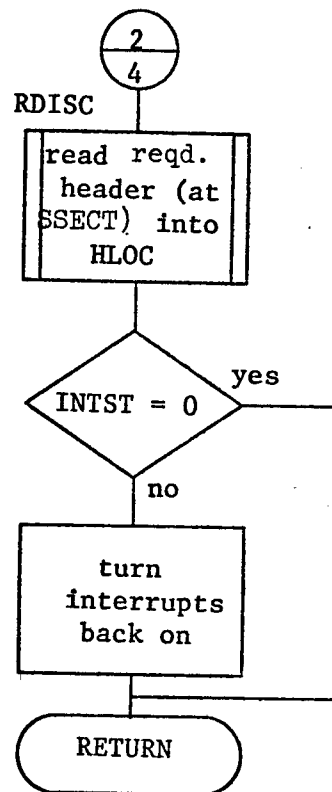
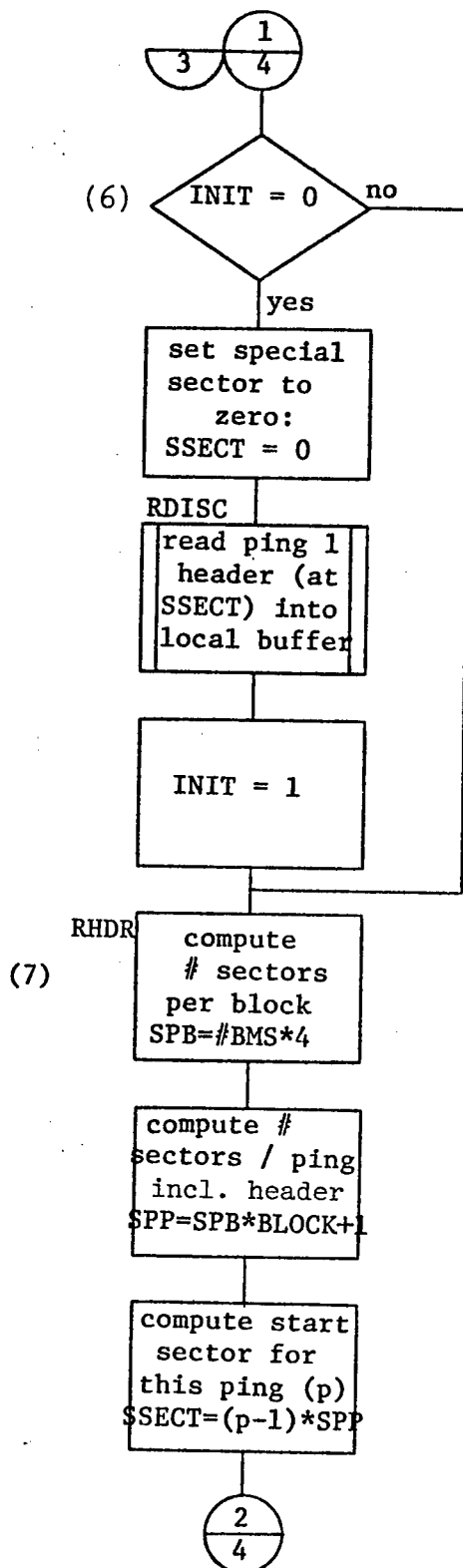
VERP

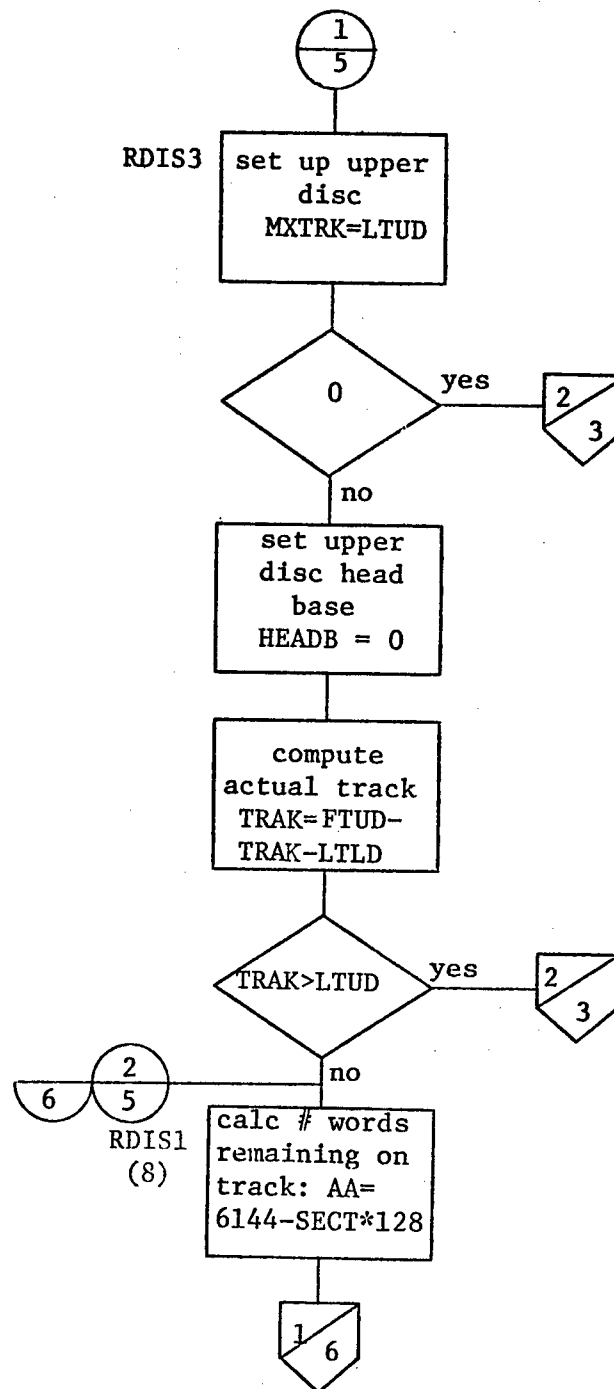
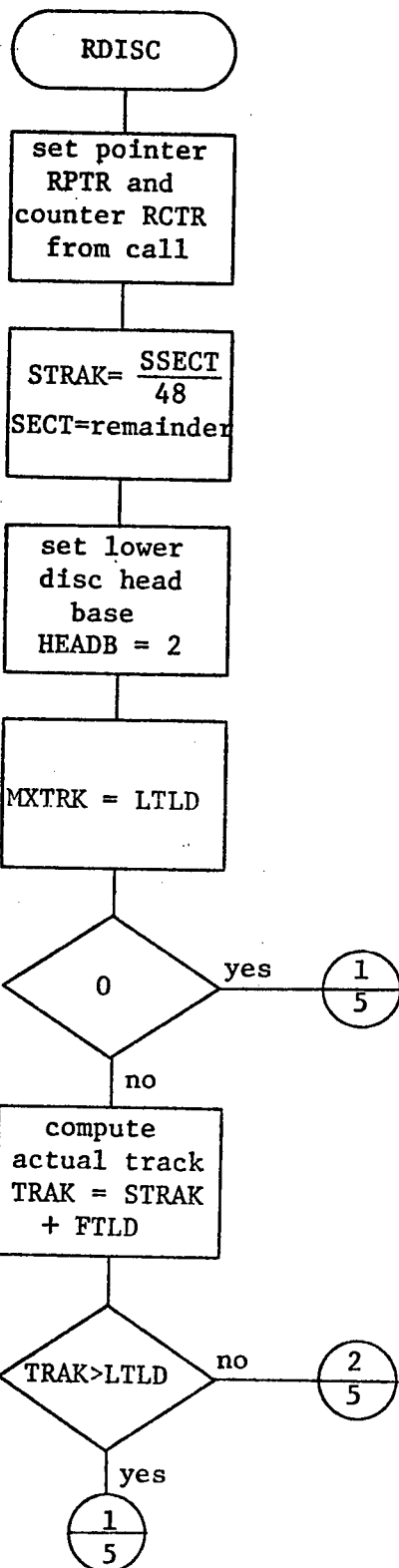
(2)

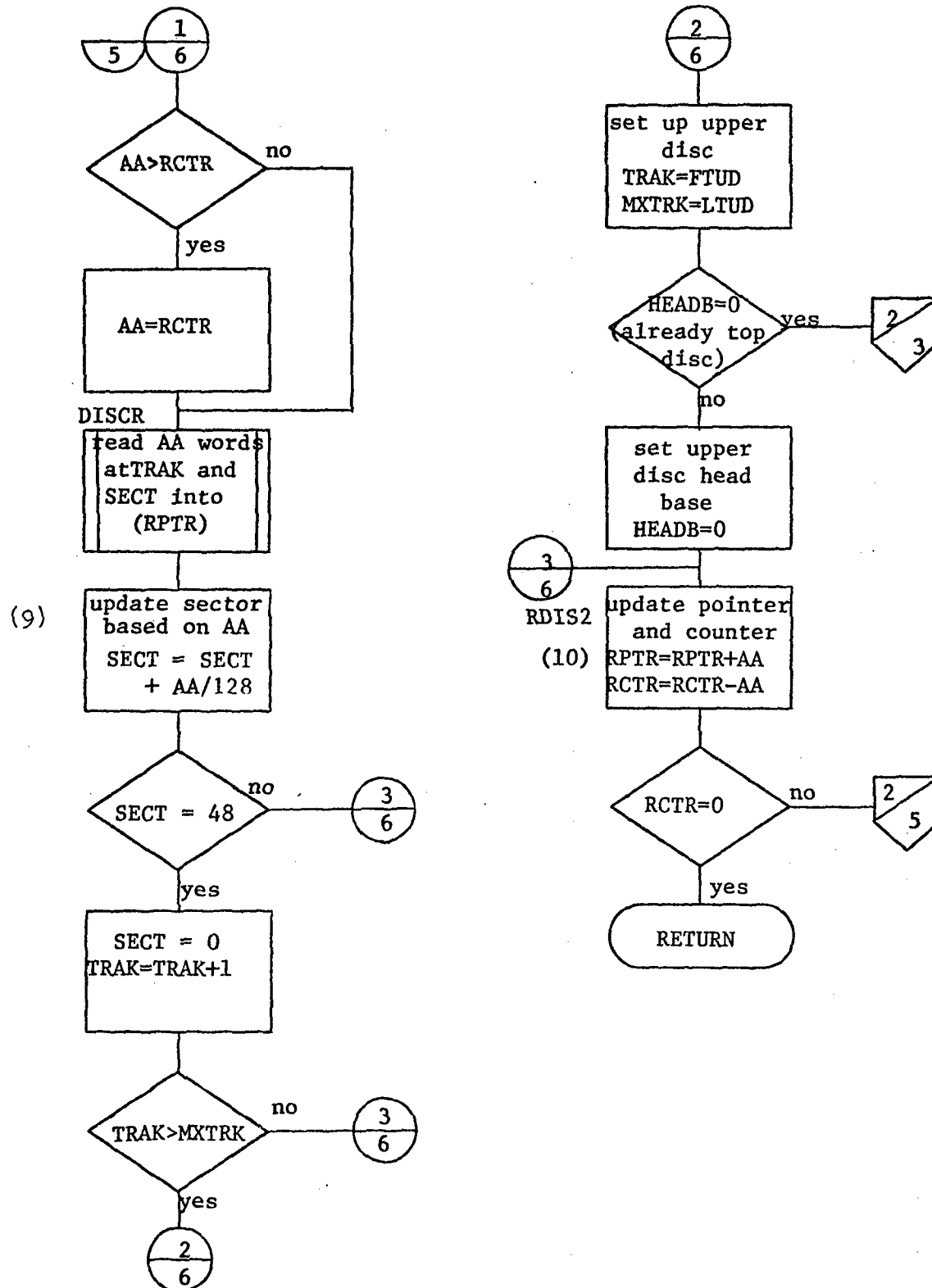


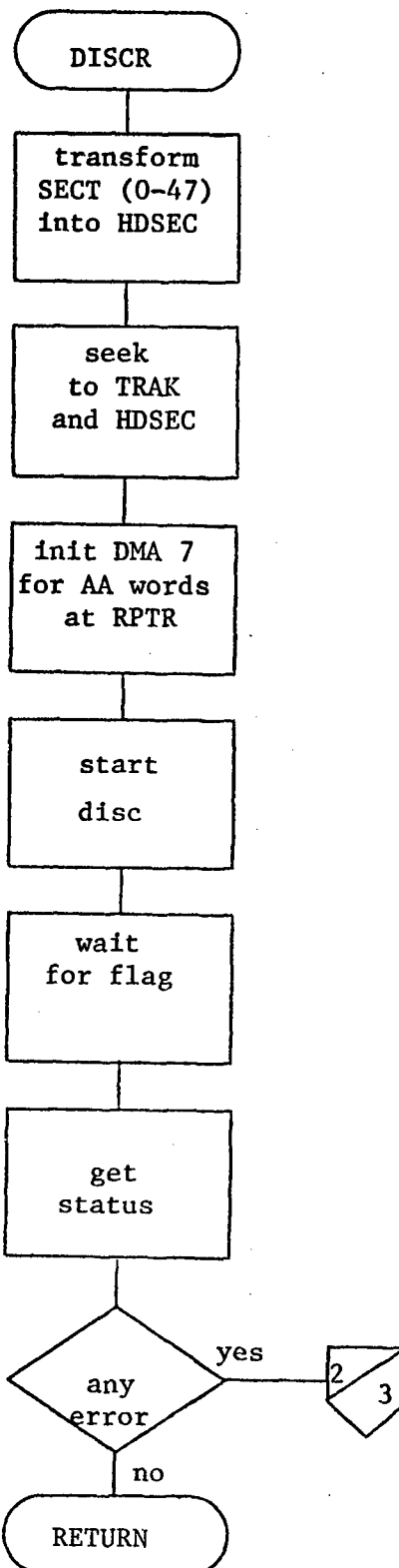












PAGE 0002 #01 RDATA SUBROUTINE

```
0001          ASMB,R,L      NAM RDATA
0003          00000      ENT RDATA,HEADR
0004          ENT .ENTR,DTAD0
0005          EXT FTLD,LTLD,FTUD,LTUD
0006
0007*
0008* THIS SUBROUTINE RETRIEVES DATA FROM DISC AS SPECIFIED
0009* BY ARGUMENTS IN CALL.
0010*
0011          00000 000000  PING# NOP
0012          00001 000000  BEAM# NOP
0013          00002 000000  INDXP NOP
0014          00003 000000  #WDS NOP
0015          00004 000000  STAT NOP
0016          00005 000000  RDATA NOP
0017          00006 016001X JSB .ENTR
0018          00007 000000R DEF PING#,I
0019          00010 162000R LDA PING#,I
0020          00011 072000R STA PING#,I
0021          00012 162001R LDA BEAM#,I
0022          00013 072001R STA BEAM#,I
0023          00014 162002R LDA INDXP,I
0024          00015 072002R STA INDXP,I
0025          00016 162003R LDA #WDS,I
```

PING (1-7)
BEAM (1-16)
INDEX (0-7, 1 IS 128 WORDS)
COUNT (1024 OR 2048)
STATUS (0 IF DATA RETURNED)

0026	00017	072003R	STA #WDS	
0027*				
0028	00020	103100	CLF 0	INTERRUPTS OFF
0029	00021	003400	CCA	
0030	00022	172004R	STA STAT,I	SET STATUS BAD
0031*				
0032	00023	062201R	LDA INIT	
0033	00024	002002	SZA	POINTERS INITIALIZED? (-1)
0034	00025	026036R	JMP VERP	YES, SKIP READ OF HEADER DATA
0035*				
0036	00026	002400	LDOK	
0037	00027	072203R	CLA SSECT	READ HEADER DATA
0038	00030	016234R	JSB HEADR	
0039	00031	000034R	DEF *+3	
0040	00032	000200R	DEF DI	
0041	00033	000212R	DEF LBUF+1	
0042	00034	002404	CLA, INA	SET INIT-1 TO SKIP HEADER READ
0043	00035	072201R	STA INIT	NEXT TIME
0044*				
0045	00036	062000R	VERP	
0046	00037	003004	LDA PING#	
0047	00040	042223R	CMA, INA	
0048	00041	002020	ADA M	
0049	00042	026176R	SSA	PING# .LE. NUMBER OF PINGS?
0050*			JMP ERROR	NO, ERROR
0051	00043	062000R	LDA PING#	
0052	00044	052220R	CPA PING	
0053	00045	026052R	JMP VERB	
0054	00046	016234R	JSB HEADR	PING SAME AS LAST, JUMP
0055	00047	000052R	DEF *+3	
0056	00050	000000R	DEF PING#	
0057	00051	000212R	DEF LBUF+1	
0058*				
0059	00052	062216R	VERB	
0060	00053	042221R	LDA FSTB#	VERIFY BEAM # WITHIN LIMITS
			ADA #BMS	

3061	00054	042553R	ADA -D-1	LARGEST BEAM # (SOMETIMES +16)
3062	00055	072205R	STA MXBM	
3063	00056	062216R	LDA FSTB#	
3064	00057	003004	CMA, INA	
3065	00060	042001R	ADA BEAM#	
3066	00061	002021	SSA, RSS	
3067	00062	026073R	JMP BOK	BEAM# .GE. FSTB#
3068	00063	062554R	LDA -D16	BEAM# .LT. FSTB#, ADD 16
3069	00064	042001R	ADA BEAM#	
3070	00065	072001R	STA BEAM#	AND TRY AGAIN
3071	00066	062216R	LDA FSTB#	
3072	00067	003004	CMA, INA	
3073	00070	042001R	ADA BEAM#	
3074	00071	002020	SSA	BEAM# .LT. FSTB# ?
3075	00072	026176R	JMP ERROR	YES, ERROR
3076*				CHECK FOR BEAM TOO LARGE
3077	00073	062001R	LDA BEAM#	
3078	00074	003004	CMA, INA	
3079	00075	042205R	ADA MXBM	
3080	00076	002020	SSA	BEAM# .GT. MXBM ?
3081	00077	026176R	JMP ERROR	YES, ERROR
3082*				STARTING SECTOR FOR THIS PING
3083	00100	062000R	LDA PING#	
3084	00101	042553R	ADA -D-1	
3085	00102	100200	MPY SPP	SKIP OVER HEADER DATA
	00103	000207R		
3086	00104	002004	INA	
3087	00105	072203R	STA SSECT	
3088*				
3089	00106	062216R	LDA FSTB#	
3090	00107	003004	CMA, INA	
3091	00110	042001R	ADA BEAM#	
3092	00111	100200	MPY -D4	
	00112	000555R		
3093	00113	042203R	ADA SSECT	STARTING SECTOR FOR DESIRED BEAM

Address	STA SSECT	Operation	Comments
0094	00114	072203R	
0095*			
0096	00115	062003R	
0097	00116	003004	
0098	00117	072204R	
0099	00120	062002X	
0100	00121	072152R	
0101*			
0102	00122	062002R	
0103	00123	006400	
0104	00124	100400	
	00125	000555R	
0105	00126	076210R	
0106	00127	100200	
	00130	000206R	
0107	00131	042210R	
0108	00132	042203R	
0109	00133	072203R	
0110*			
0111	00134	062210R	
0112	00135	003004	
0113	00136	042555R	
0114	00137	100200	
	00140	000556R	
0115	00141	072153R	
0116	00142	042204R	
0117	00143	003004	
0118	00144	066204R	
0119	00145	007004	
0120	00146	002020	
0121	00147	076153R	
0122	00150	016310R	
0123	00151	000154R	
0124	00152	000000	
0125	00153	000000	

0126	00154	062153R	LDA RWDS	UPDATE POINTER AND COUNTER
0127	00155	042152R	ADA PNTR	BY NUMBER OF WORDS READ
0128	00156	072152R	STA PNTR	
0129	00157	062153R	LDA RWDS	
0130	00160	042204R	ADA CNTR	
0131	00161	072204R	STA CNTR	
0132	00162	002003	SZA, RSS	COUNTER=0?
0133	00163	026174R	JMP, DONE	YES, DONE
0134	00164	062210R	LDA OFST	NO, REMOVE OFFSET FROM SSECT AND
0135	00165	003004	CMA, INA	UPDATE TO NEXT BLOCK
0136	00166	042203R	ADA SSECT	
0137	00167	042206R	ADA SPB	
0138	00170	072203R	STA SSECT	
0139	00171	002400	CLA	CLEAR OFFSET (NOW STARTING AT
0140	00172	072210R	STA OFST	START OF BLOCK)
0141	00173	026134R	JMP NXTBL	GO DO IT AGAIN
0142*				
0143	00174	002400	CLA	SET STATUS TO OK
0144	00175	172004R	STA STAT, I	
0145*				
0146	00176	102100	STF 0	INTERRUPTS ON
0147	00177	126005R	JMP RDATA, I	RETURN
0148	00176		EQU ERROR	
0149	00176		ERR12 EQU ERROR	
0150*				
0151	00200	000001	DEC 1	
0152	00201	000000	INIT NOP	
0153	00202	000000	INTST NOP	
0154	00203	000000	SSECT NOP	
0155	00204	000000	CNTR NOP	
0156	00205	000000	MXBM NOP	
0157	00206	000000	SPB NOP	
0158	00207	000000	SPP NOP	
0159	00210	000000	OFST NOP	
0160*				

3161* THE ORDER OF THE FOLLOWING 14 CARDS MUST NOT BE DISTURBED

PAGE 0003 #01 RDATA SUBROUTINE

0162	00211	000212R	LBUF	DEF *+1
0163	00212	0000000	TIME	BSS 2
0164	00214	0000000	ANGL1	NOP
0165	00215	0000000		NOP
0166	00216	0000000	FSTB#	NOP
0167	00217	0000000	FBANG	NOP
0168	00220	0000000	PING	NOP
0169	00221	0000000	#BMS	NOP
0170	00222	0000000	S	NOP
0171	00223	0000000	M	NOP
0172	00224	0000000	K	NOP
0173	00225	0000000	N	NOP
0174	00226	0000000	BLOCK	NOP
0175	00227	0000000		BSS 3

END OF SPECIAL BUFFER AREA

RRERETRIEVE HEADER SUBROUTINE

0176*				
0177*				
0178*				
0179	00232	0000000	HPNG#	NOP
0180	00233	0000000	HDATA	NOP
0181	00234	0000000	HEADR	NOP
0182	00235	016001X		JSB .ENTR
0183	00236	000232R		DEF HPNG#
0184	00237	002404		CLA, INA
0185	00240	102300		SFS 0

DETERMINE INT. SYS. STATUS

0186	00241	002400	CLA	INTST	INTST: 1=INT ON, 0=INT OFF
0187	00242	072202R	STA	0	INTERRUPTS OFF
0188	00243	103100	CLF		
0189	00244	062233R	LDA	HDATA	
0190	00245	072300R	STA	HLOC	
0191	00246	062201R	LDA	INIT	
0192	00247	002002	SZA		POINTERS INITIALIZED? (-1)
0193	00250	026261R	JMP	RHDR	YES, SKIP READ OF 1ST HEADER
0194	00251	002400	CLA		SET SPECIAL SECTOR POINTER TO
0195	00252	072203R	STA	SSECT	BEGINNING OF DATA
0196*					
0197	00253	016310R	JSB	RDISC	READ PART OF FIRST SECTOR
0198	00254	000257R	DEF	*+3	OF DATA AREA
0199	00255	000212R	DEF	LBUFF+1	
0200	00256	000020	DEC	16	
0201	00257	002404	CLA	INA	SET INIT=1 TO SKIP HEADER READ
0202	00260	072201R	STA	INIT	NEXT TIME
0203*					
0204	00261	062221R	LDA	#BMS	COMPUTE # SECTORS PER BLOCK
0205	00262	100200	MPY	=D4	AND PING
	00263	000555R			
0206	00264	072206R	STA	SPB	BEAMS X BLOCKS X 4 SECTORS/BEAM
0207	00265	100200	MPY	BLOCK	
	00266	000226R			
0208	00267	002004	INA		INCLUDE 1 SECTOR FOR HEADER
0209	00270	072207R	STA	SPP	
0210*					
0211	00271	162232R	LDA	HPNG#,I	
0212	00272	042553R	ADA	=D-1	STARTING SECTOR FOR THIS PING
0213	00273	100200	MPY	SPP	
	00274	000207R			
0214	00275	072203R	STA	SSECT	
0215*					
0216	00276	016310R	JSB	RDISC	READ HEADER, WHICH IS PART OF
0217	00277	000302R	DEF	*+3	FIRST SECTOR OF EACH PING

0218	00300	000000	HLOC	NOP		
0219	00301	000020		DEC 16		
0220	00302	062202R		LDA INTST		GET INT. SYS. STATUS AT ENTRY
0221	00303	002002		SZA 0		IF ZERO, DO NOT TURN ON
0222	00304	102100		STF 0		INTERRUPTS ON
0223	00305	126234R		JMP HEADR, I		RETURN
0224*						
0225*			DISC SUBROUTINES			
0226*						
0227	00306	000000	RPTR	NOP		
0228	00307	000000	RCRTR	NOP		
0229	00310	000000	RDISC	NOP		
0230	00311	016001X		JSB .ENR		
0231	00312	000306R		DEF RPTR		
0232*						
0233	00313	062203R		LDA SSECT		CONVERT SSECT TO TRACK AND SECTOR
0234	00314	006400		CLB		
0235	00315	100400		DIV -D48		
0236	00316	000557R				
0237	00317	072544R		STA STRAK		
0238	00320	076543R		STB SECT		SET LOWER DISC HEAD BASE
0239	00321	062560R		LDA -D2		
0240	00322	072546R		STA HEADB		
0241	00323	062544R		LDA STRAK		
0242	00324	066004X		LDB LTLD		ANY SPACE LOWER DISC?
0243	00325	076545R		STB MXTRK		
0244	00326	006003		SZB, RSS		
0245	00327	026337R		JMP RDIS3		NO, GO SET UP UPPER DISC
0246	00330	042003X		ADA FTLD		COMPUTE ACTUAL TRACK
0247	00331	072542R		STA TRAK		
0248	00332	003004		CMA, INA		
0249	00333	042004X		ADA LTLD		GREATER THAN LAST TRACK?
0250	00334	002021		SSA, RSS		NO, GO READ
0251	00335	026353R		JMP RDIS1		
0252	00336	003000		CMA		

0252	00337	066006X	RDIS3	LDB	LTUD	ANY SPACE UPPER DISC?
0253	00340	076545R		STB	MXTRK	
0254	00341	006003		SZB,	RSS	
0255	00342	026176R		JMP	ERROR	NO, ERROR
0256	00343	006400		CLB	HEADB	SET UPPER DISC HEAD BASE
0257	00344	076546R		STB	FTUD	
0258	00345	042005X		ADA	TRAK	COMPUTE ACTUAL TRACK
0259	00346	072542R		STA	INA	
0260	00347	003004		ADA	LTUD	
0261	00350	042006X		SSA	ERR07	TRACK GREATER THAN LAST TRACK?
0262	00351	002020		JMP		YES, ERROR
0263	00352	026176R				
0264*						
0265	00353	062543R	RDIS1	LDA	SECT	
0266	00354	100200		MPY	-D128	
0267	00355	000556R		CMA,	INA	
0268	00357	042561R		ADA	-D6144	AA - 6144 - SECT*128
0269	00360	072541R		STA	AA	# WORDS REMAINING IN TRACK
0270	00361	066307R		LDB	RCTR	
0271	00362	003004		CMA,	INA	
0272	00363	042307R		ADA	RCTR	AA .GT. RCTR?
0273	00364	002020		SSA	AA	YES, SUBSTITUTE RCTR FOR AA
0274	00365	076541R		STB	AA	READ AA WORDS
0275	00366	016433R		JSB	DISC	
0276*						
0277	00367	006400		CLB		UPDATE SECTOR
0278	00370	062541R		LDA	AA	THIS ROUTINE USES AN ARTIFICIAL
0279	00371	100400		DIV	-D128	SECTOR IN RANGE 0-47
0280	00372	000556R		ADA	SECT	
0281	00373	042543R		STA	SECT	SECT - SECT + AA/128
0282*						
0283	00375	052557R		CPA	-D48	
0284	00376	002001		RSS		

SECTOR NOT 48, SKIP TRACK UPDATE

0285	00377	026421R	JMP RDIS2	
0286	00400	002400	CLA SECT	
0287	00401	072543R	STA TRAK	
0288	00402	036542R	IS2 TRAK	
0289	00403	062542R	LDA TRAK	
0290	00404	003004	CMA, INA	
0291	00405	042545R	ADA MXTRK	
0292	00406	002021	SSA, RSS	
0293	00407	026421R	JMP RDIS2	
0294	00410	062005X	LDA FTUD	
0295	00411	072542R	STA TRAK	
0296	00412	062006X	LDA LTUD	
0297	00413	072545R	STA MXTRK	
0298	00414	062546R	LDA HEADB	
0299	00415	002003	SZA, RSS	
0300	00416	026176R	JMP ERROR	
0301	00417	002400	CLA HEADB	
0302	00420	072546R	STA HEADB	
0303*				
0304	00421	062541R	LDA AA	
0305	00422	042306R	ADA RPTR	
0306	00423	072306R	STA RPTR	
0307	00424	062541R	LDA AA	
0308	00425	003004	CMA, INA	
0309	00426	042307R	ADA RCTR	
0310	00427	072307R	STA RCTR	
0311	00430	002002	SZA	
0312	00431	026353R	JMP RDIS1	
0313	00432	126310R	JMP RDISC, I	
0314*				
0315	00433	000000	NOP	
0316	00434	062543R	LDA SECT	
0317	00435	070001	STA B	
0318	00436	042562R	ADA -D-24	
0319	00437	002020	SSA	

TRAK .GT. MXTRK?
NO
YES, GO TO SECOND DISC

ALREADY ON UPPER DISC?
ERROR, STORAGE EXCEEDED

HEAD BASE - 1ST UPPR DISC HEAD

UPDATE POINTER AND COUNTER

RCTR=0?
NO, GO DO ANOTHER READ
RETURN

CONVERT SECT TO HEAD AND SECTOR

0320 00440 026445R

JMP FRSTH

PAGE 0004 #01 RDATA SUBROUTINE

0321	00441	070001	STA B	
0322	00442	062546R	LDA HEADB	
0323	00443	002004	INA	
0324	00444	002001	RSS	
0325	00445	062546R	LDA HEADB	
0326	00446	001727	ALF, ALF	
0327	00447	030001	IOR B	
0328	00450	072547R	STA HDSEC	
0329	00451	016501R	JSB SEEK	
0330	00452	106717	CLC DISCD	
0331	00453	106720	CLC DISCC	
0332	00454	062563R	LDA -B120000	SET UP DMA
0333	00455	032552R	IOR SELDD	
0334	00456	102607	OTA 7	
0335	00457	106703	CLC 3	
0336	00460	062306R	LDA RPTR	
0337	00461	032564R	IOR -B100000	
0338	00462	102603	OTA 3	
0339	00463	102703	STC 3	
0340	00464	062541R	LDA AA	
0341	00465	003004	CMA, INA	
0342	00466	102603	OTA 3	
0343	00467	062550R	LDA RDCMD	
0344	00470	102620	OTA DISCC	

WORDS TO READ

3345	00471	103717	STC DISCD,C		
3346	00472	103707	STC 7,C		
3347	00473	103720	STC DISCC,C		
3348	00474	102320	SFS DISCC		
3349	00475	026474R	JMP *-1		
3350	00476	106707	CLC 7		
3351	00477	016523R	JSB STATS		
3352	00500	126433R	JMP DISCR,I		
3353*					
3354	00501	000000	NOP	SEEK	
3355	00502	106717	CLC DISCD		
3356	00503	106720	CLC DISCC		
3357	00504	062542R	LDA TRAK		
3358	00505	102617	OTA DISCD		
3359	00506	103717	STC DISCD,C		
3360	00507	062551R	LDA SKCMD		
3361	00510	102620	OTA DISCC		
3362	00511	103720	STC DISCC,C		
3363	00512	102317	SFS DISCD		
3364	00513	026512R	JMP *-1		
3365	00514	062547R	LDA HDSEC		
3366	00515	102617	OTA DISCD		
3367	00516	103717	STC DISCD,C		
3368	00517	102320	SFS DISCC		
3369	00520	026517R	JMP *-1		
3370	00521	016523R	JSB STATS		
3371	00522	126501R	JMP SEEK,I		
3372*					
3373	00523	000000	NOP	STATS	
3374	00524	106717	CLC DISCD		
3375	00525	106720	CLC DISCC		
3376	00526	002400	CLA		
3377	00527	102620	OTA DISCC		
3378	00530	103720	STC DISCC,C		
3379	00531	102317	SFS DISCD		

WAIT FOR COMPLETION

STOP DMA
CHECK STATUS
RETURN

STATUS COMMAND

0380	00532	026531R	JMP *-1	
0381	00533	102517	LIA DISCD	
0382	00534	000010	SLA	
0383	00535	026537R	JMP ERR10	ERROR, DISC STATUS BAD
0384	00536	126523R	JMP STATS,I	RETURN
0385*				
0386	00537	102010	ERR10	DISC STATUS BAD:
0387	00540	026537R	JMP *-1	IRRECOVERABLE ERROR
0388*				
0389	00541	000000	AA	
0390	00542	000000	TRAK	
0391	00543	000000	SECT	
0392	00544	000000	STRAK	
0393	00545	000000	MXTRK	
0394	00546	000000	HEADB	
0395	00547	000000	HDSEC	
0396	00550	020000	RDCMD	OCT 20000
0397	00551	030000	SKCMD	OCT 30000
0398	00552	000017	SELDD	DEF DISCD
0399	00017		DISCD	EQU 17B
0400	00020		DISCC	EQU 20B
0401	00000		A	EQU 0
0402	00001		B	EQU 1
	00553	177777		
	00554	000020		
	00555	000004		
	00556	000200		
	00557	000050		
	00560	000002		
	00561	014000		
	00562	177750		
	00563	120000		
	00564	100000		

2403
 ** NO ERRORS *TOTAL **RTE ASMB 750420**

END

TRACK ALLOCATION TABLE - TAT

1. FUNCTION. TAT is a non-executable program unit which provides the disc track availability information to all programs.
2. CONSTRAINTS. None.
3. CALLING SEQUENCE. Called by: N/A; used by: Y0001, RDATA.
4. DESCRIPTION OF INPUT. N/A
5. DESCRIPTION OF OUTPUT. N/A
6. FILES USED. N/A
7. ERRORS AND RESTRICTIONS. N/A
8. COMPUTER OPERATOR INSTRUCTIONS. N/A
9. DESCRIPTION OF PROCESSING. TAT has the following four entry points. Referencing them as externals provides the required information to the other program.

FTLD - first track lower disc

LTLD - last track lower disc

FTUD - first track upper disc

LTUD - last track upper disc

TAT1

T-00004 IS ON CR00001 USING 00003 BLKS R-0014

0001

ASMB,R,L

0002

HED TRACK ALLOCATION TABLE FOR Y0001 AND RDATA

0003

NAM TAT

0004

ENT FTLD,LTLD,FTUD,LTUD

0005

*

0006

0007

* WHENEVER THIS TABLE IS CHANGED, FILES #TAT1 AND #DCP MUST BE

0008

* UPDATED AND THE DATA COLLECTION AND ANALYSIS PROGRAMS MUST

0009

* BE REBUILT. IT IS IMPORTANT THAT BOTH SYSTEMS USE THE SAME

0010

* TAT.

0011

FTLD DEC 0

0012

LTLD DEC 0

0013

FTUD DEC 61

0014

LTUD DEC 202

0015

END

:

0033	SAVEX	BSS 1			
0034	PPSW	OCT 0		SAVE LOW ORDER X BITS	
0035		HED	CHOUT	POINT PLOT MODE SWITCH	
0036	*				
0037	CHOUT	ABS	IMFUR		
0038		JSB	LENTR	SET UP THE POINTERS	
0039		LDA	ARGAD,I	GET IDATA	
0040		JSB	LCHOT	AND OUTPUT IT	
0041		JMP	RETAD,I	RETURN	
0042		HED	CHIN		
0043		ABS	IMFUR	SET UP THE POINTERS	
0044	CHIN	JSB	LENTR	GET AN INPUT CHARACTER	
0045		JSB	LCHIN	STORE IT INTO IDATA	
0046		STA	ARGAD,I		
0047		JMP	RETAD,I	RETURN	
0048		HED	TPLOT		
0049	TPLOT	ABS	IMFUR	SET UP POINTERS	
0050		JSB	LENTR	GET THE FIRST ARGUMENT	
0051		LDA	ARGAD,I	ADVANCE ARGUMENT POINTER	
0052		ISZ	ARGAD	CHECK FOR CASE 1, I = 0	
0053		SZA,RSS			
0054		JMP	DARK		
0055		SSA		SKIP IF BRIGHT VECTOR	
0056		JMP	POINT	CASE 3, I < 0	
0057		JMP	POIBR	CASE 2, I > 0	
0058		HED	TPLOT	CASE 1	
0059	*				
0060	*				
0061	DARK	EQU *			
0062		LDA	GS	GET THE CODE FOR VECTOR MODE	
0063		JMP	POIFE	AND ENTER POINT PLOT	
0064		HED	TPLOT	CASE 3	
0065	*				
0066	*				
0067		POINT	EQU *		

0068	LDA GS	GET THE CODE FOR POINT PLOT
0069	STA PPSW	SET POINT PLOT SWITCH
0070	POIFE EQU *	
0071	JSB LCHOT	
0072	POIBR EQU *	AND OUTPUT IT
0073	LDA ARGAD,I	
0074	STA X	GET X
0075	ISZ ARGAD	AND SAVE IT LOCALLY
0076	LDA ARGAD,I	ADVANCE TO NEXT ARGUMENT
0077	STA Y	GET Y
0078	JSB PLOT	
0079	LDA SAVEX	NOW PLOT THIS POINT
0080	LDB PPSW	LOAD LOW ORDER X
0081	SZB	LOAD POINT PLOT SWITCH
0082	JSB LCHOT	SKIP IF NOT POINT PLOT MODE
0083	CLA	OUTPUT LOW ORDER X
0084	STA PPSW	
0085	JMP RETAD,I	RESET POINT PLOT SWITCH
0086	HED CURSI	RETURN
0087	ABS IMRUR	
0088	JSB LENTR	SET UP THE POINTERS
0089	LDA US	GET THE CODE FOR ALPHA MODE
0090	JSB LCHOT	AND OUTPUT IT
0091	LDA ESC	GET THE CODE TO ENABLE GRAPHIC INPUT
0092	JSB LCHOT	AND OUTPUT IT
0093	LDA SUB	
0094	JSB LCHOT	
0095	*	WAIT FOR THE FIRST INPUT
0096	JSB LCHIN	SAVE THE CHARACTER
0097	STA ARGAD,I	ADVANCE POINTER TO NEXT ARGUMENT
0098	ISZ ARGAD	
0099	*	
0100	JSB LCHIN	GET THE NEXT INPUT CHARACTER
0101	AND B37	MASK IT TO ONLY FIVE BITS
0102	ALF,RAL	POSITION FIVE BITS OVER

0103	STA X	AND SAVE THEM IN X
0104	JSB LCHIN	GET THE NEXT CHARACTER
0105	AND B37	AND MASK TO FIVE BITS
0106	IOR X	PLUG IN MSB
0107	STA ARGAD, I	AND RETURN X VALUE TO CALLER
0108	ISZ ARGAD	=> Y ARGUMENT
0109	JSB LCHIN	FETCH THE NEXT CHARACTER
0110	AND B37	AND MASK IT TO FIVE BITS
0111	ALF, RAL	MOVE FIVE PLACES TO THE LEFT
0112	STA Y	AND SAVE THEM IN Y
0113	JSB LCHIN	GET THE LAST CHARACTER
0114	AND B37	AND MASK IT OFF TO ONLY FIVE BITS
0115	IOR Y	PLUG IN MSB
0116	STA ARGAD, I	AND RETURN VALUE TO CALLER
0117	JMP RETAD, I	RETURN
0118	HED PLOT	
0119	* PLOT	
0120	ABS IMAJR	
0121	LDA Y	GET THE Y VALUE
0122	ALF, ALF	DISCARD THE RIGHTMOST
0123	ALF, RAR	FIVE BITS
0124	AND B37	MASK TO ONLY FIVE BITS
0125	IOR HOB	PLUG IN REQUIRED BITS
0126	JSB LCHOT	AND OUTPUT THE CHARACTER
0127	LDA Y	GET BACK THE Y VALUE
0128	AND B37	AND MASK TO ONLY FIVE BITS
0129	IOR LYB	PLUG IN REQUIRED BITS
0130	JSB LCHOT	AND OUTPUT THE CHARACTER
0131	LDA X	NOW GET THE X VALUE
0132	ALF, ALF	DISCARD THE RIGHTMOST
0133	ALF, RAR	FIVE BITS
0134	AND B37	
0135	IOR HOB	PLUG IN THE REQUIRED BITS
0136	JSB LCHOT	AND OUTPUT THE CHARACTER
0137	LDA X	GET BACK THE X VALUE

0138	AND B37	AND LEAVE ONLY LOW ORDER FIVE
0139	IOR LXB	PLUG IN THE REQUIRED BITS
0140	STA SALEX	SAVE LOW ORDER X BITS
0141	JSB LCHOT	AND OUTPUT THE CHARACTER
0142	JMP PLOT,I	RETURN
0143	HED LCHIN	
0144	ABS IMRUR	
0145	EQU *	
0146	SFS SC	WE ARE NOT ALLOWED TO
0147	JMP LCHIP	TOUCH IF BEING USED
0148	LDA *	TIME WASTER
0149	LDA *	
0150	SFS SC	CHECK ONCE MORE TO MAKE SURE
0151	JMP LCHIP	
0152	LDA TYRB	GET THE TTY INPUT BITS
0153	OTA SC	AND OUTPUT TO THE INTERFACE
0154	STC SC,C	START THE INPUT
0155	CLC SC	PREVENT AN INTERRUPT
0156	SFS SC	WAIT FOR SOME ACTION
0157	JMP *-1	
0158	LIA SC	GET THE CHARACTER
0159	AND B377	AND MASK IT OFF
0160	JMP LCHIN,I	RETURN
0161	HED LCHOT	
0162	ABS IMRUR	
0163	AND B377	MASK THE CHARACTER TO SIZE
0164	EQU *	
0165	SFS SC	WE ARE NOT ALLOWED TO
0166	JMP LCHOP	TOUCH IF BEING USED
0167	LDB *	TIME WASTER
0168	LDB *	
0169	SFS SC	CHECK ONCE MORE
0170	JMP LCHOP	
0171	LDB TYRB	GET THE TTY OUTPUT BITS
0172	OTB SC	AND TELL INTERFACE BOARD

0173	OTA SC	PUT THE CHARACTER OUT
0174	STC SC,C	START THE OUTPUT
0175	CLC SC	PREVENT AN INTERRUPT
0176	SFS SC	WAIT FOR THE OUTPUT
0177	JMP *-1	TO FINISH
0178	JMP LCHOT,I	RETURN
0179	HED LENTR	
0180	ABS IMPJR	
0181	LDA LENTR	A - ADDRESS OF OUR CALLER + 2
0182	ADA M2	SUBTRACT OFF THE TWO
0183	LDA A,I	FIND WHERE HE WAS CALLED FROM
0184	IOR BIT15	PLUG IN THE INDIRECT BIT
0185	STA RETAD	SAVE PTR TO RETURN ADDRESS
0186	INA	ADVANCE POINTER TO ARGUMENT LIST
0187	STA ARGAD	AND SAVE FOR THE ROUTINE
0188	JMP LENTR,I	RETURN
0189	END	
0190	**END-OF-TAPE	
0191	*	

:

KEYBOARD INPUT AND SPECIAL LINE PRINTING - TYPI, TYP A, TYP C, PNTL

1. FUNCTION. TYPI, TYP A, and TYP C perform keyboard input of integer strings and multiple ASCII characters, respectively, bypassing IOC and its requirement for line feed to be pressed and its lack of error checking. They also permit null entries. PNTL outputs a line of ASCII characters based on the contents of the printer buffer, with the buffer length given.

2. CONSTRAINTS. A maximum of 20 characters may be entered. No check is made for multiple + or - signs or values greater than can be held in 16 bits all of which are error conditions.

3. CALLING SEQUENCE. Called by: Y1002, Y0003, INPUT, ASKQ

CALL TYPI (IVAL,IFLG)

where IVAL is the location where the entered value will be stored
IFLG is the location where the error flag will be stored.

CALL TYP A (IVAL,IFLG)

where IVAL is the location where the ASCII character will be stored
IFLG is the location where the error flag will be stored
IFLG is set negative for error, zero for no input, and positive for good input.

TYP C - performs keyboard input of up to ten (10) ASCII characters.

CALL TYP C (AC,IFLGC)

AC: Location of buffer where the ASCII characters are stored.

IFLGC: Location where error flag is stored. (IFLGC values same as TYP A'S IFLGA values)

PNTL - prints one line of formatted data without doing a carriage return and line feed.

CALL PNTL (IPNTB,IBLG)

IPNTB: print buffer address

IBLG: buffer length

4. DESCRIPTION OF PROCESSING. Input is an ASCII string entered through the keyboard, followed by a carriage return. A DEL character deletes the entire line. A carriage return alone causes a return with IFLG set to zero. PNTL input is the contents of the printer buffer (IPNTB).

5. DESCRIPTION OF OUTPUT. Output is the value entered in integer format, or the single ASCII character (for TYPA). TYPC allows for ten ASCII characters to be output.

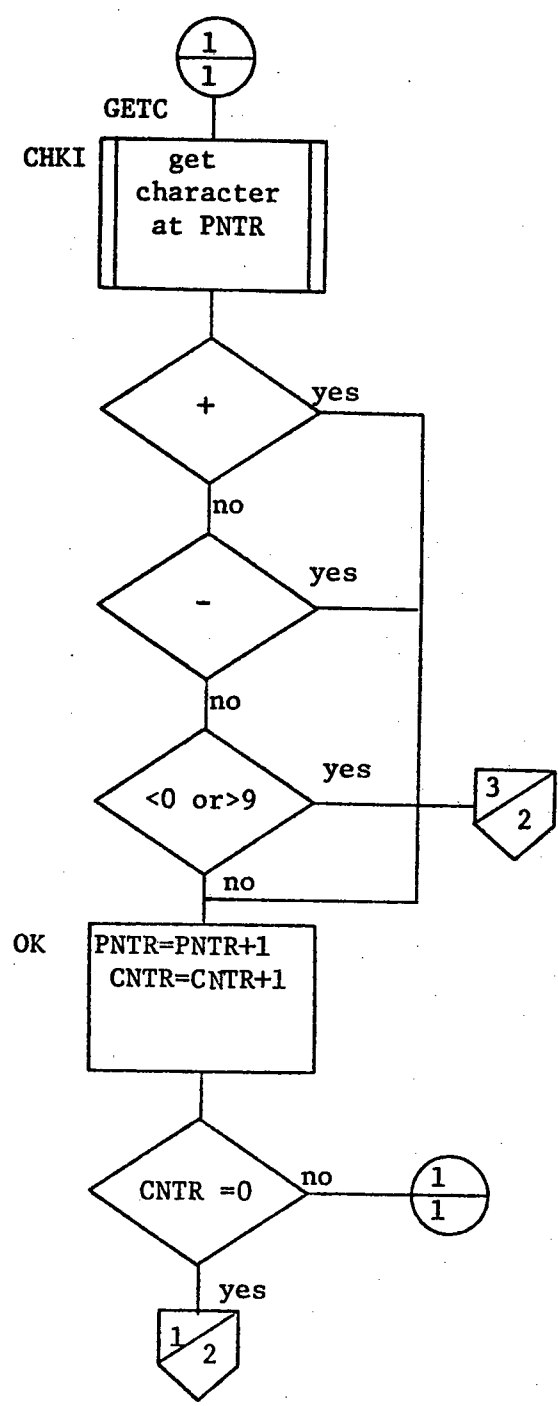
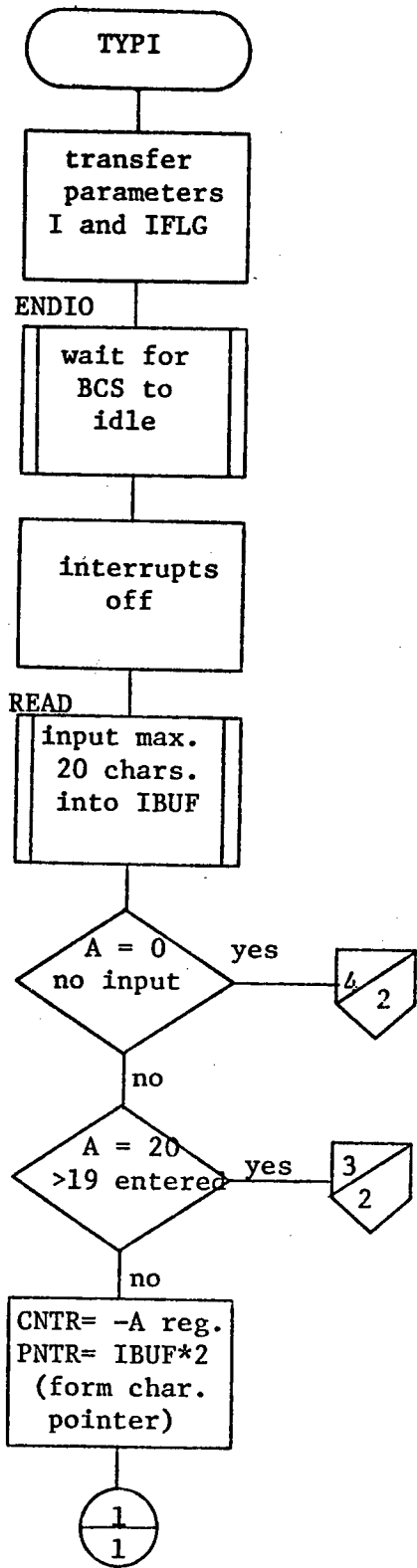
6. FILES USED. N/A

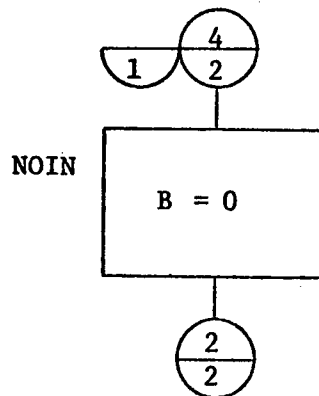
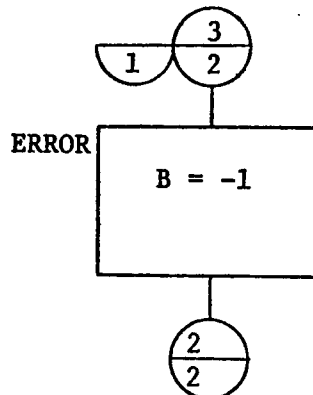
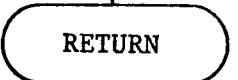
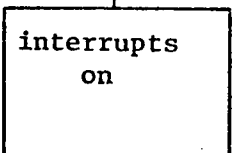
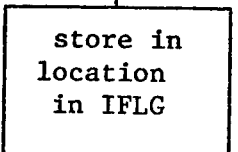
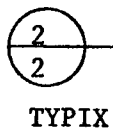
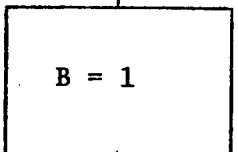
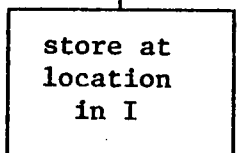
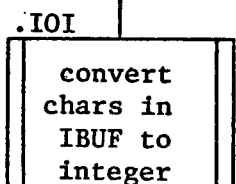
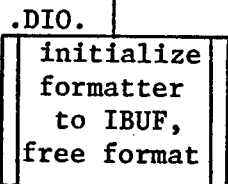
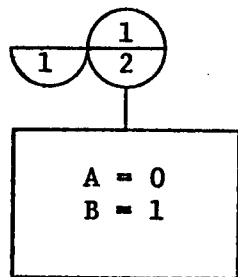
7. ERRORS AND RESTRICTIONS. None.

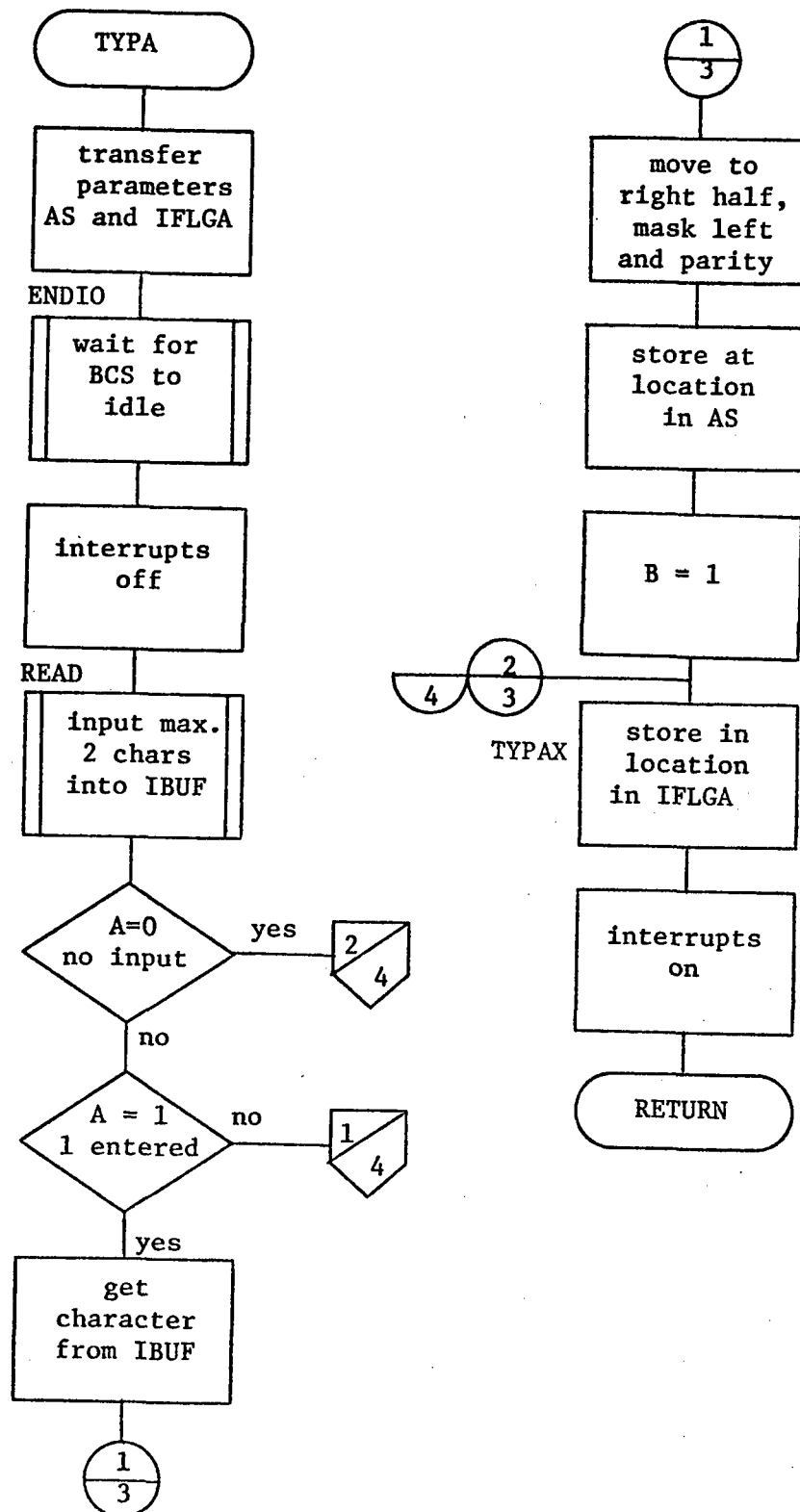
8. COMPUTER OPERATOR INSTRUCTIONS. N/A.

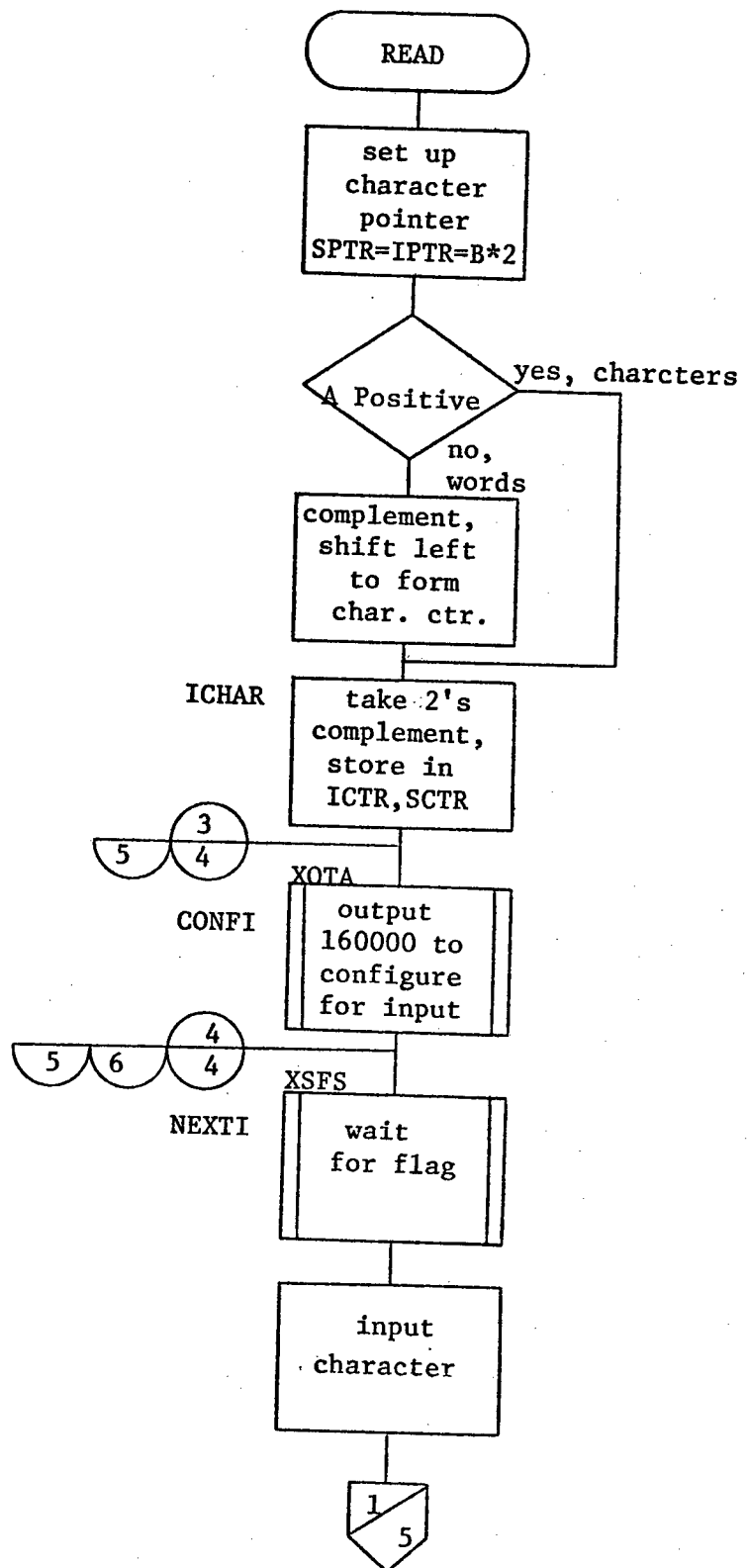
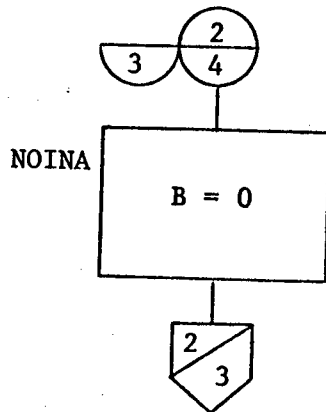
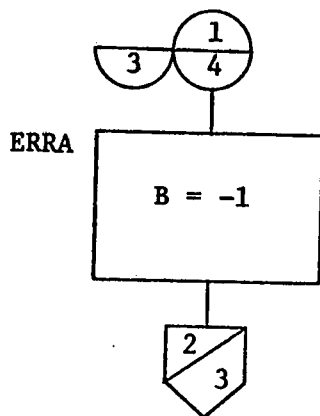
9. DESCRIPTION OF PROCESSING. TYPI, TYPA, and TYPC use their own I/O routines. To keep from interrupting BCS output, ENDIO is called at the beginning of each. The Formatter is called to convert the integer character string to binary in its internal conversion mode. However, in this mode there is no capability for supplying a character count. To overcome this shortcoming, a space is stored following the last character, as a delimiter. All other program details are shown in the flowchart, and the user is directed there for further information.

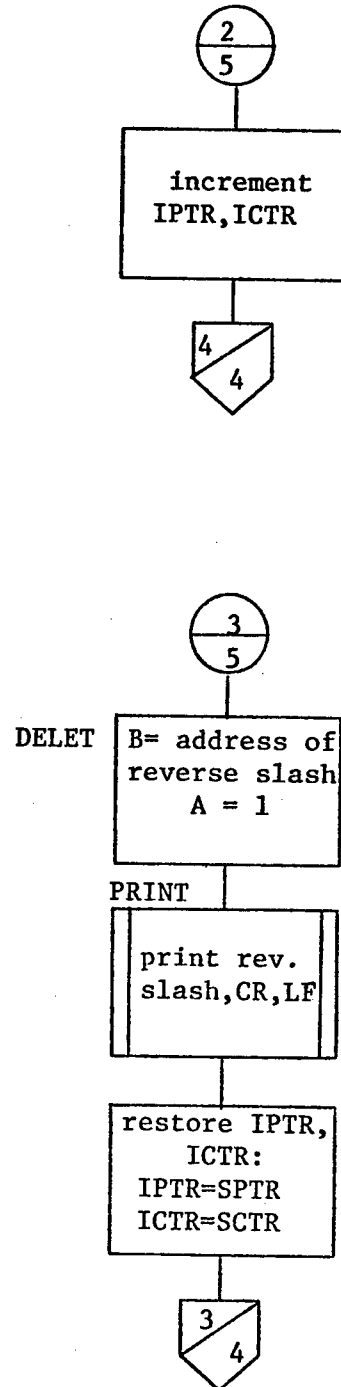
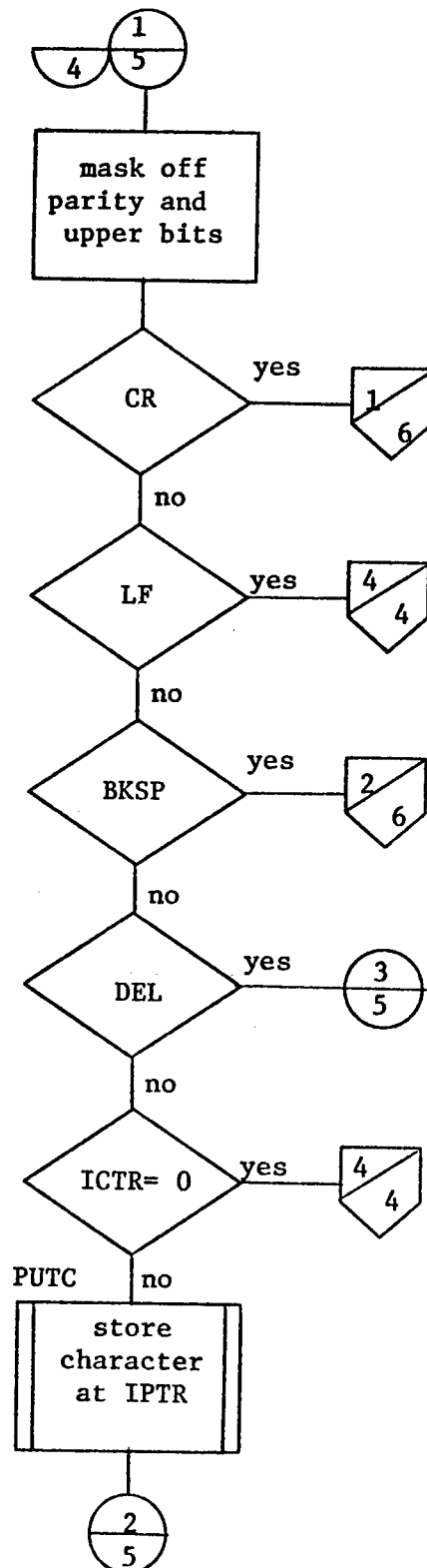
PNTL prints one line of formatted data without doing a carriage return and line feed. PNTL uses its own I/O routines. To keep from interrupting BCS output, ENDIO is called at the beginning of the routine. The pointer to the printer buffer is set into PNTR with the buffer length set into IBLG. The characters are then output. All other program details are shown in the flowchart and the user is directed there for further information.

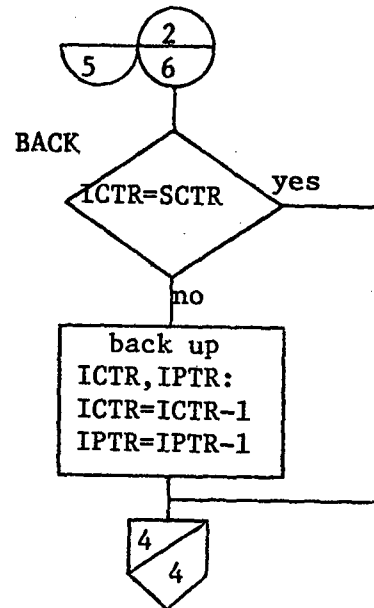
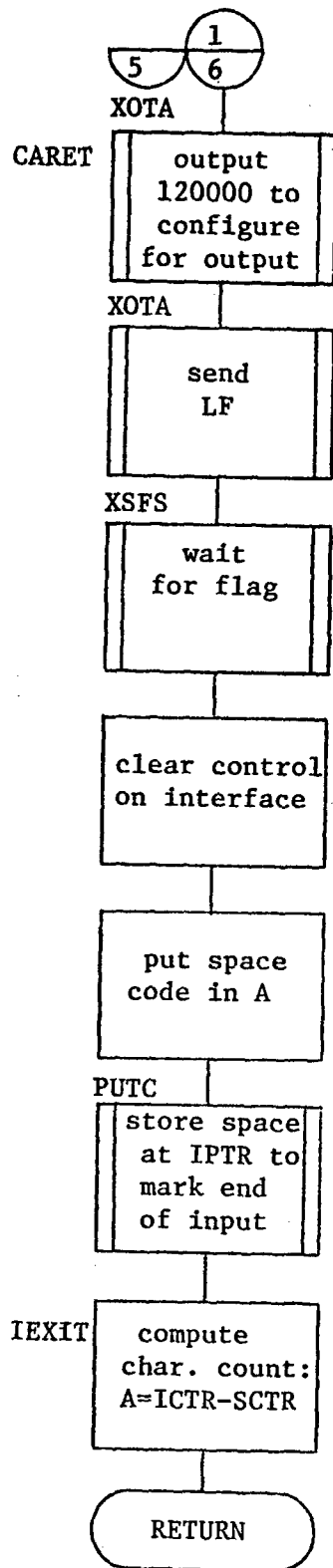


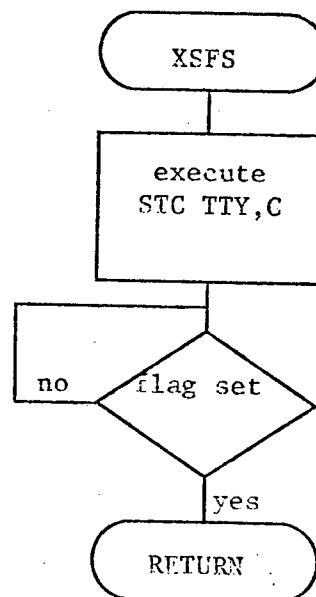
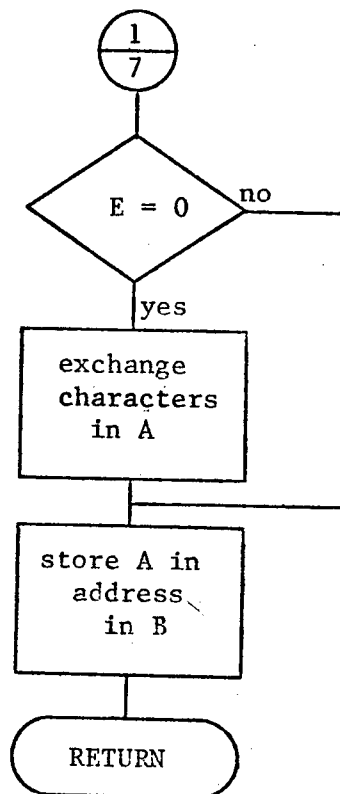
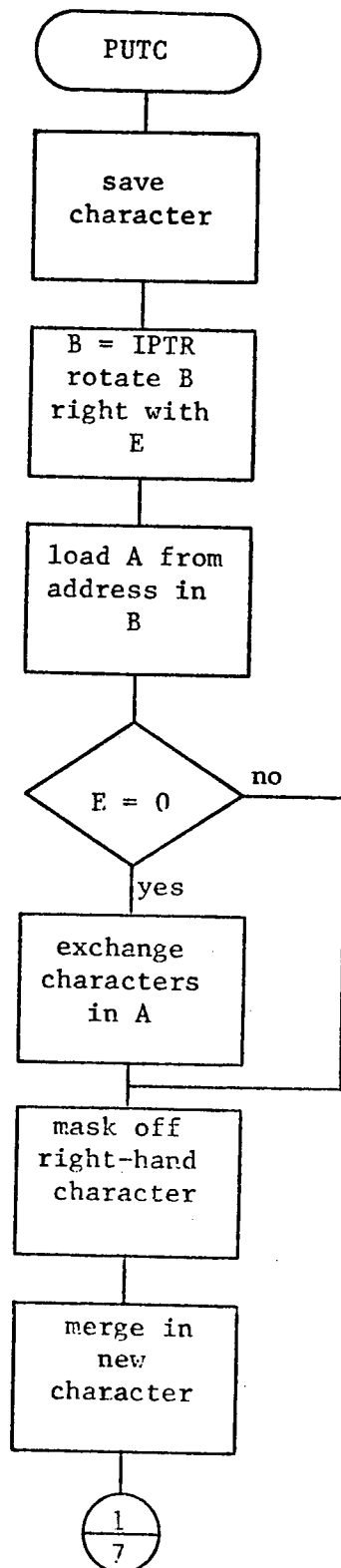


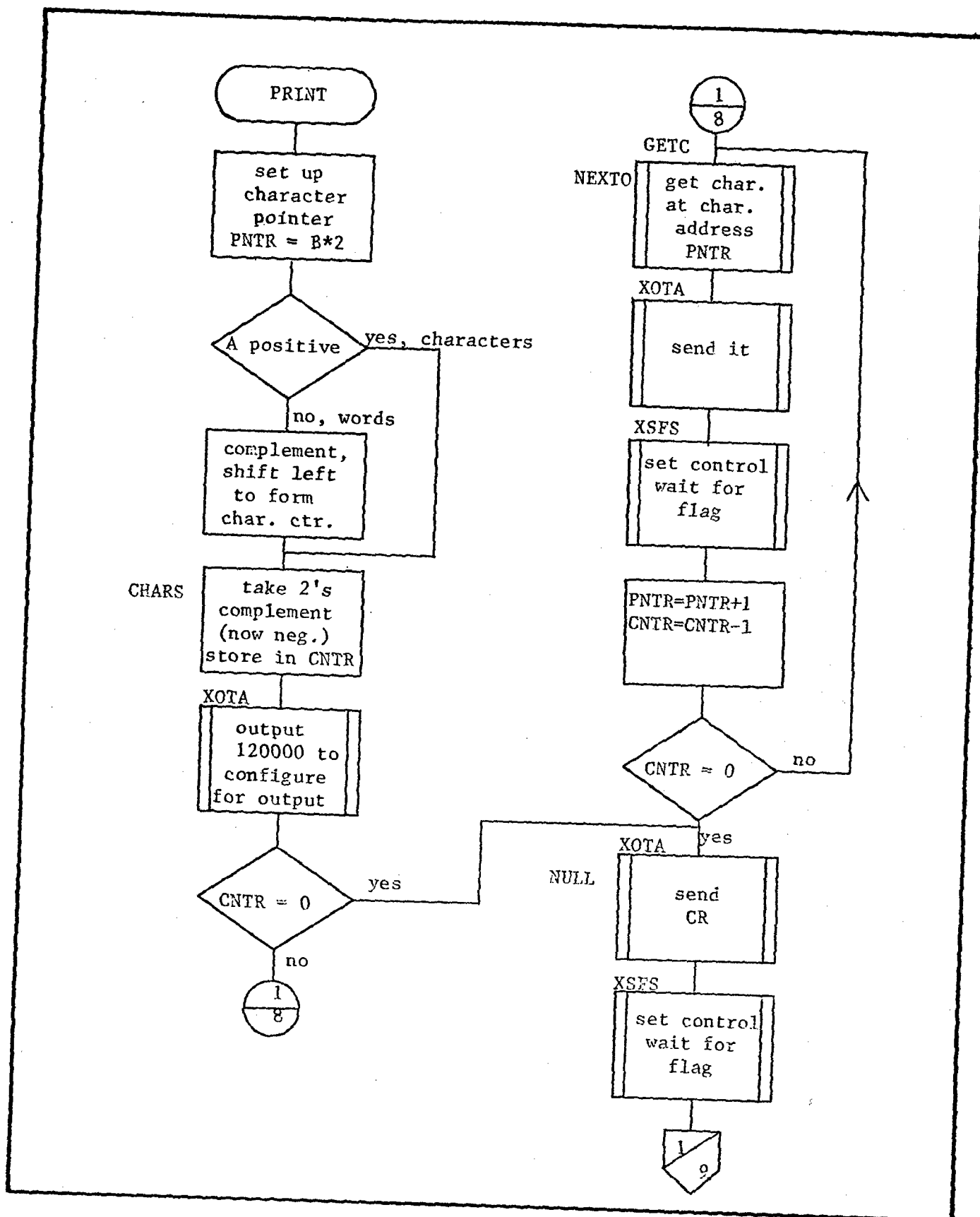


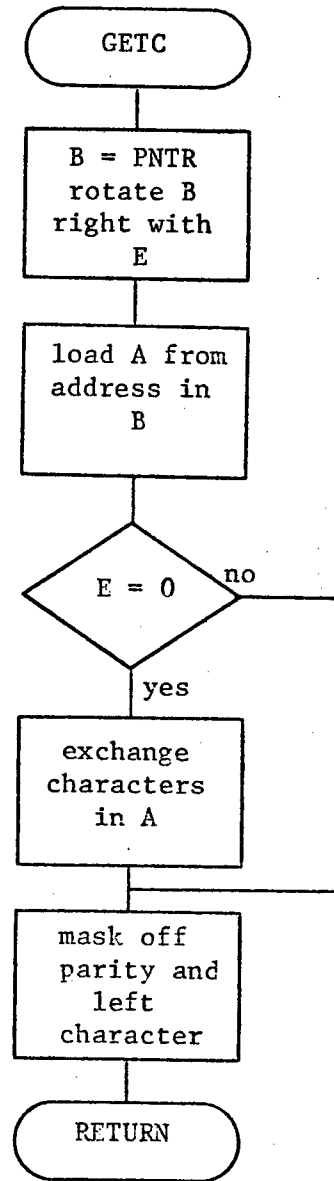
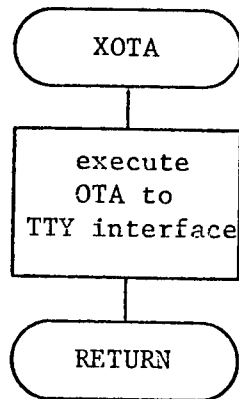
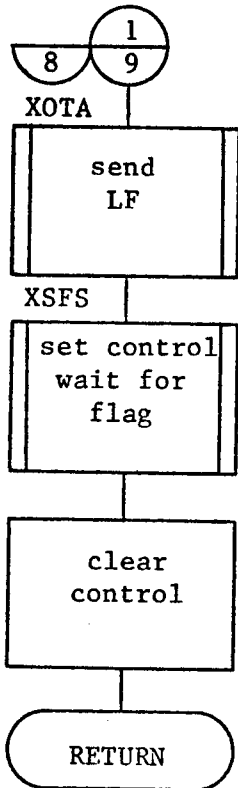


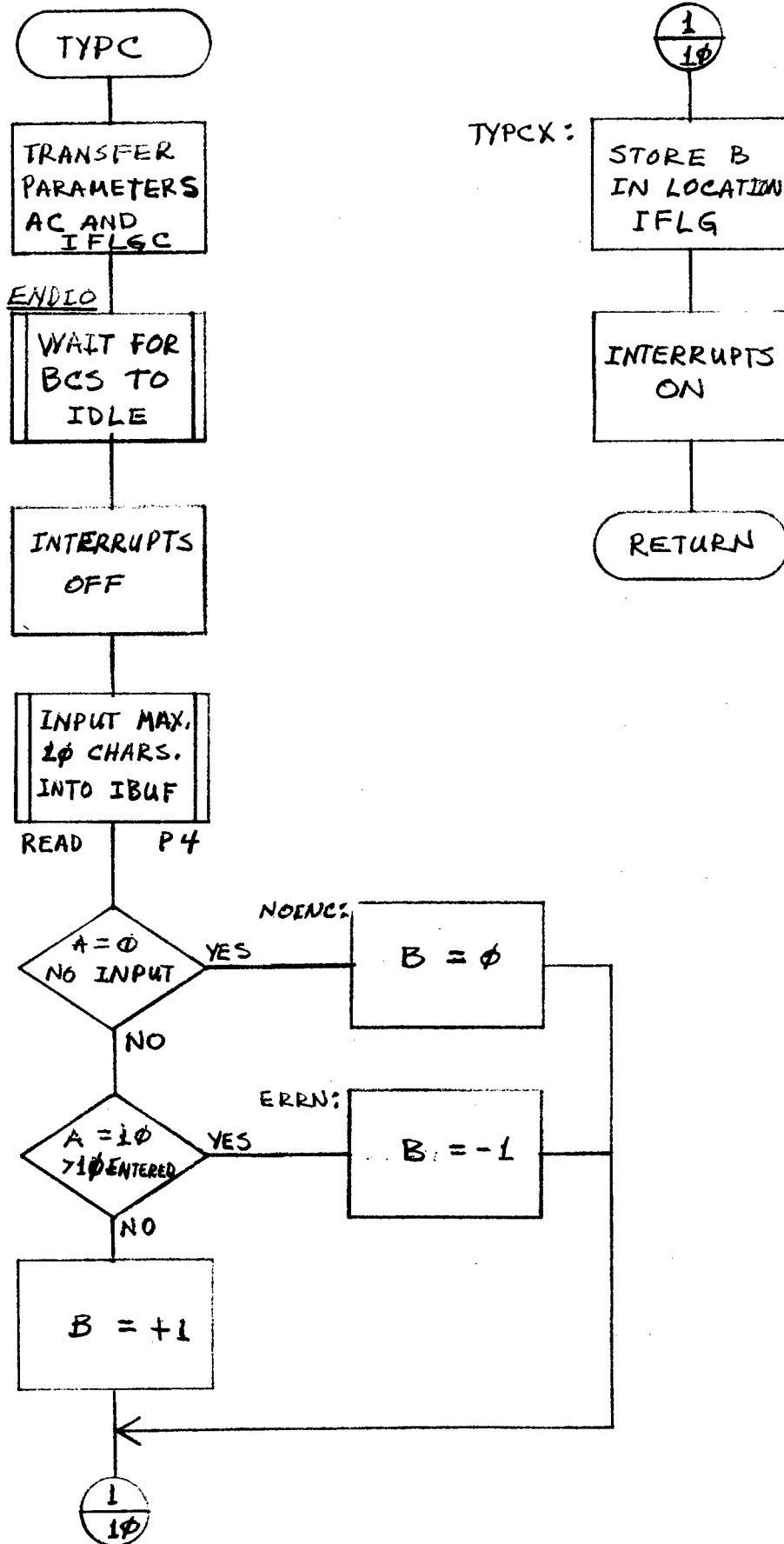


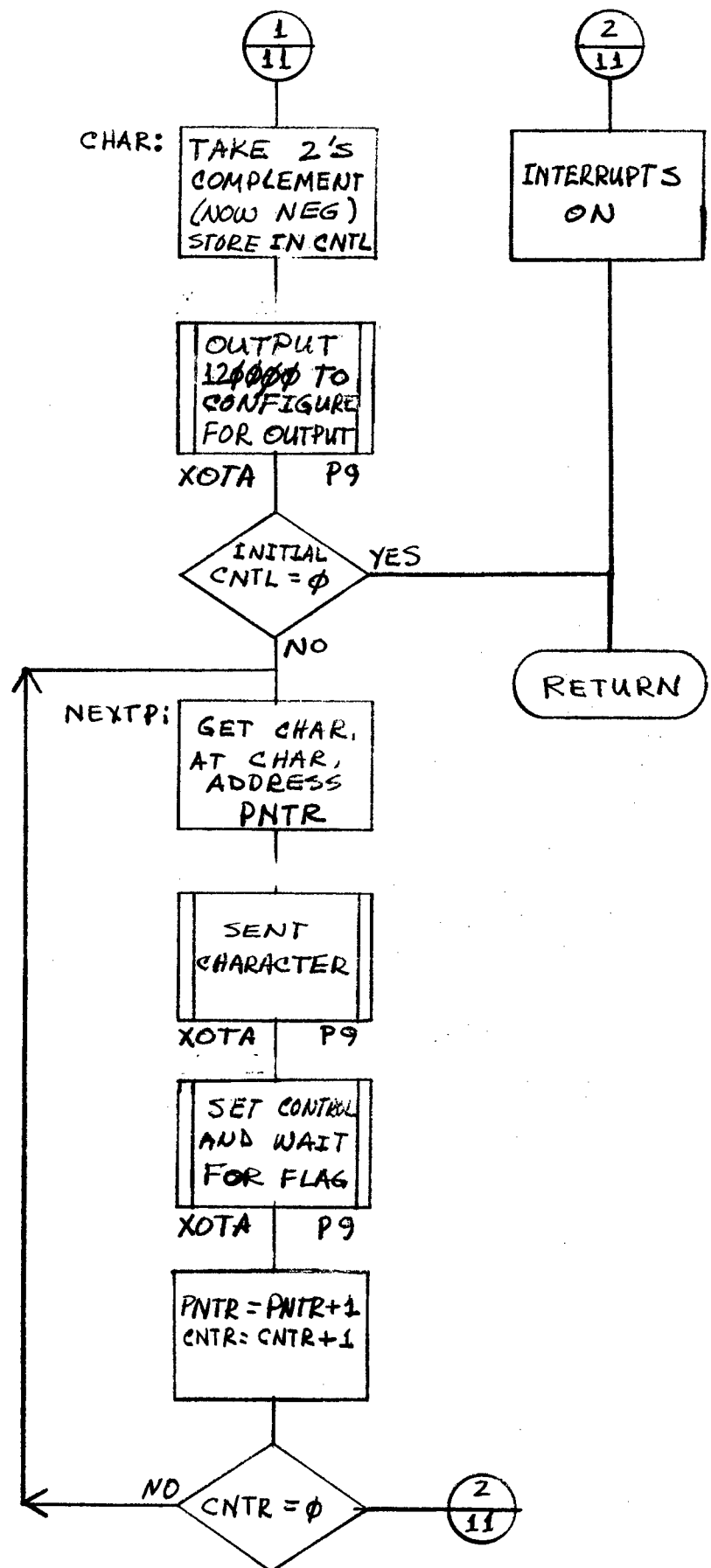
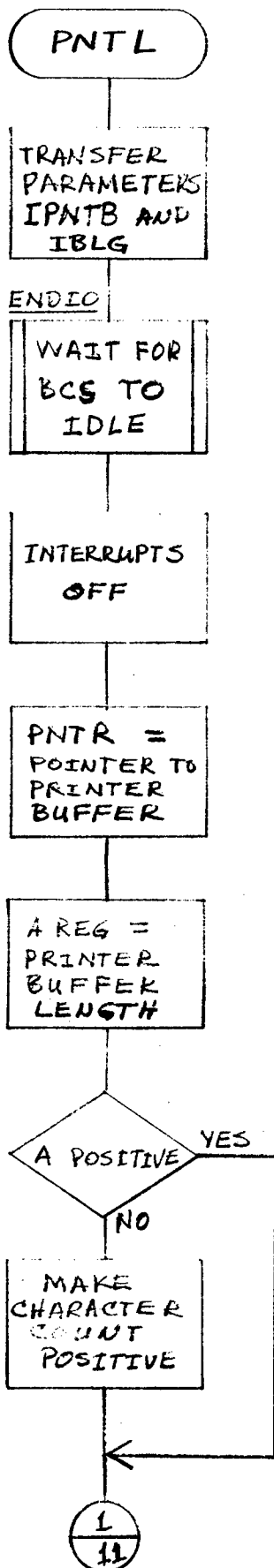













```

0001 ASMB,R,L
0003 NAM TYPI1
0004 ENT TYPI,TYPA,TYPC,PNTL
0005 EXT .ENTR,.DIO,.IOI,.ENDIO
0006*
0007* *****
0008*
0009* TITLE: KEYBOARD INPUT AND SPECIAL LINE PRINTING
0010*
0011* MODULE: TYPI SOURCE: TYPI1 VERSION: X00
0012* OBJECT: #TYPI1
0013*
0014* AUTHOR: R. TASHJIAN DATE: 1 DECEMBER 1976
0015*
0016* MODIFICATIONS:
0017* 1: ADD TYPC ROUTINE W. BOWEN
0018* 7/14/77
0019* 2: ADD PNTL ROUTINE W. BOWEN
0020* 7/28/77
0021* 3: CHANGE TYPC ROUTINE TO VAR. CHAR. COUNT
0022* 12/16/77 R. TASHJIAN
0023*
0024* MODULE FUNCTIONS:
0025*
0026* TYPI - PERFORMS KEYBOARD INPUT OF INTEGER STRINGS.
0027*
0028* CALL TYPI (I,I,IFLG)
0029*
*****

```

I: LOCATION WHERE ENTERED VALUE IS STORED.
IFLG: LOCATION WHERE ERROR FLAG IS STORED.
TYP A - PERFORMS KEYBOARD INPUT OF ASCII CHARACTERS
CALL TYP A (AS,IFLGA)
AS: LOCATION WHERE ASCII CHARACTERS ARE STORED.
IFLGA: LOCATION WHERE ERROR FLAG IS STORED.
IFLGA - NEGATIVE - ERROR
IFLGA - ZERO - NO INPUT
IFLGA - POSITIVE - GOOD INPUT
TYP C - PERFORMS KEYBOARD INPUT OF UP TO TEN (10) ASCII CHARACTERS.
CALL TYP C (AC,IFLGC,COUNT)
AC: LOCATION OF BUFFER WHERE THE ASCII CHARACTERS ARE STORED.
IFLGC: LOCATION WHERE ERROR FLAG IS STORED.
(IFLGC VALUES SAME AS TYP A'S IFLGA VALUES)
COUNT: NUMBER OF CHARACTERS ALLOWED
PNTL - PRINTS ONE LINE OF FORMATTED DATA WITHOUT DOING A CARRIAGE RETURN AND LINE FEED.

0030*
0031*
0032*
0033*
0034*
0035*
0036*
0037*
0038*
0039*
0040*
0041*
0042*
0043*
0044*
0045*
0046*
0047*
0048*
0049*
0050*
0051*
0052*
0053*
0054*
0055*
0056*
0057*

```
* * CALL PNTL (IPNTB,IBLG)
* *
* IANTB: PRINT BUFFER ADDRESS
* IBLG:  BUFFER LENGTH
* *****
0063* *****
0064* *****
0065* *****
0066* *****
0067* *****
0068* *****
0069* *****
0070 00000 000000 I NOP
0071 00001 000000 IF LG NOP
0072 00002 000000 TYPI NOP
0073 00003 016001X JSB .ENTR
0074 00004 000000R DEF I
0075 00005 016004X JSB ENDIO
0076 00006 000007R DEF X+1
0077 00007 103100 CLF 0
0078 00010 062467R LDA -D20
0079 00011 066124R LDB IBUF
0080 00012 016251R JSB READ
0081 00013 002003 SZA,RSS
0082 00014 026066R JMP NOIN
0083 00015 052467R CPA -D20
0084 00016 026064R JMP ERROR
0085 00017 003004 CMA,INA
0086 00020 072445R STA CNTR
0087 00021 066124R LDB IBUF
0088 00022 004066 CLE,ELB
0089 00023 076444R STB PNTR
*****
*****
***** TYP I ROUTINE *
*****
*****
WAIT UNTIL OTHER I/O DONE
ALLOW 20 CHARACTERS
NOTHING ENTERED
20 OR MORE ENTERED, ERROR
```


0115	00053	000125R	DEF IBUF+1	
0116	00054	000000	ABS 0	
0117	00055	000064R	DEF ERROR	
0118	00056	016003X	JSB .IOI.	
0119	00057	172000R	STA I,I	
0120	00060	006404	CLB,INB	SET IFLG = +1
0121*				
0122	00061	176001R	STB IFLG,I	
0123	00062	102100	STF 0	
0124	00063	126002R	JMP TYPI,I	
0125*				
0126	00064	007400	ERROR CCB	SET IFLG = -1
0127	00065	026061R	JMP TYPX	
0128*				
0129	00066	006400	NOIN CLB	
0130	00067	026061R	JMP TYPX	SET IFLG = 0
0131*				
0132*			*****	
0133*			* TYPX ROUTINE *	
0134*			*****	
0135*				
0136	00070	000000	AS NOP	
0137	00071	000000	IFLGA NOP	
0138	00072	000000	TYPX NOP	
0139	00073	016001X	JSB .ENTR	
0140	00074	000070R	DEF AS	
0141	00075	016004X	JSB ENDIO	
0142	00076	000077R	DEF *+1	
0143	00077	103100	CLF 0	
0144	00100	062477R	LDA -D2	
0145	00101	066124R	LDB IBUF	
0146	00102	016251R	JSB READ	
0147	00103	002003	SZA,RSS	
0148	00104	026122R	JMP NOINA	
0149	00105	052500R	CPA -D1	NOTHING ENTERED

WAIT UNTIL OTHER I/O DONE

0150	00106	002001	RSS		
0151	00107	026120R	JMP	ERRA	
0152	00110	062125R	LDA	IBUF+1	
0153	00111	001727	ALF,	ALF	
0154	00112	012501R	AND	=B177	
0155	00113	172070R	STA	AS,I	
0156	00114	006404	CLB,	INB	
0157*			SET	IFLG = +1	
0158	00115	176071R	TYPAX	STB	IFLGA,I
0159	00116	102100	STF	0	
0160	00117	126072R	JMP	TYPAX,I	
0161*			SET	IFLG = -1	
0162	00120	007400	ERRA	CCB	
0163	00121	026115R	JMP	TYPAX	
0164*			SET	IFLG = 0	
0165	00122	006400	NOINA	CLB	
0166	00123	026115R	JMP	TYPAX	
0167*					
0168	00124	000125R	IBUF	DEF	*+1
0169	00125	000000	BSS	21	

ERROR, MORE THAN 1 CHAR ENTERED
 PUT CHARACTER IN RIGHT SIDE
 SET IFLG = +1

SET IFLG = -1
 SET IFLG = 0

PAGE 0005 #01 SUBROUTINES TYPI, TYPAX, TYPY, AND PNTL

0170*	*****
0171*	* TYPY ROUTINE *
0172*	*****
0173*	*****
0174*	

0175	00152	000000	AC	NOP	
0176	00153	000000	IFLG	NOP	
0177	00154	000000	COUNT	NOP	
0178	00155	000000	TYPC	NOP	
0179	00156	016001X		JSB .ENTR	FACILITATE ARGUMENT TRANSFER
0180	00157	000152R		DEF AC	
0181	00160	016004X		JSB ENDIO	WAIT UNTIL OTHER I/O IS DONE
0182	00161	000162R		DEF X+1	
0183	00162	103100		CLF 0	CLEAR I/O FLAG
0184*					
0185	00163	162154R		LDA COUNT,I	ALLOW N ALPHANUMERIC CHARACTERS
0186	00164	066152R		LDB AC	
0187	00165	016251R		JSB READ	READ INPUT DATA
0188	00166	002003		SZA,RSS	
0189	00167	026203R		JMP NOINC	NOTHING ENTERED
0190	00170	166154R		LDB COUNT,I	
0191	00171	006004		~~INB~~	
0192	00172	050001		CPA B	
0193	00173	002001		~~RSS~~	
0194	00174	026201R		JMP ERRN	ERROR, MORE THAN 10 CHARACTERS ENTE
RED					
0195	00175	006404		CLB,INB	SET IFLG = +1
0196*					
0197	00176	176153R	TYPCX	STB IFLGC,I	
0198	00177	102100		STF 0	SET I/O FLAG
0199	00200	126155R		JMP TYPC,I	RETURN
0200*					
0201	00201	007400	ERRN	CCB	SET IFLG = -1
0202	00202	026176R		JMP TYPCX	
0203*					
0204	00203	006400	NOINC	CLB	SET IFLG = 0
0205	00204	026176R		JMP TYPCX	
0206*					
0207*					
0208*					

* PNTL ROUTINE *

0209*

0210*

0211 00205 000000 IPNTB NOP

0212 00206 000000 IBLG NOP

0213 00207 000000 PNTL NOP

0214 00210 016001X JSB .ENTR

0215 00211 000205R DEF IPNTB

0216 00212 016004X JSB ENDIO

0217 00213 000214R DEF X+1

0218 00214 103100 CLF 0

0219*

0220 00215 066205R LDB IPNTB

0221 00216 162206R LDA IBLG,I

0222*

0223 00217 004066 CLE,ELB

0224 00220 076444R STB PNTR

0225 00221 002021 SSA,RSS

FACILITATE ARGUMENT TRANSFER

WAIT UNTIL OTHER I/O IS DONE

CLEAR I/O FLAG

LOAD PRINT BUFFER ADDRESS POINTER

LOAD LENGTH OF PRINT BUFFER

CHANGE TO CHARACTER POINTER

SET-UP CHARACTER POINTER

IS BUFFER LENGTH POSITIVE ?

PAGE 0006 #01 SUBROUTINES TYPI, TYP A, TYPC, AND PNTL

0226

00222 026225R JMP CHAR

0227 00223 003004 CMA,INA

0228 00224 001200 RAL

0229*

0230 00225 003004 CHAR

0231 00226 072445R STA CNTR

YES, CHARACTERS

NEGATIVE WORDS, MAKE POSITIVE

X2 (MAKE CHARACTERS)

SAVE COMPLEMENTED CHARACTER COUNT

0232	00227	062454R	LDA B12K	CONFIGURE I/O CARD
0233	00230	016430R	JSB XOTA	
0234	00231	062445R	LDA CNTR	
0235	00232	002003	SZA,RSS	COUNT INITIALLY ZERO ?
0236	00233	126207R	JMP PNTL,I	YES, RETURN
0237*				
0238	00234	066444R	LDB PNTR	
0239	00235	004065	CLE,ERB	
0240	00236	160001	LDA B,I	
0241	00237	002041	SEZ,RSS	
0242	00240	001727	ALF,ALF	
0243	00241	012456R	AND B177	
0244*				
0245	00242	016430R	JSB XOTA	SENT A CHARACTER
0246	00243	016433R	JSB XSFS	SET CONTROL AND WAIT FOR FLAG
0247	00244	036444R	ISZ PNTR	INCREMENT CHARACTER POINTER
0248	00245	036445R	ISZ CNTR	INCREMENT CHARACTER COUNT, SKIP -
0				
0249	00246	026234R	JMP NEXTP	
0250	00247	106711	CLC SC	CLEAR INTRRUPTS
0251	00250	126207R	JMP PNTL,I	OUTPUT COMPLETE, RETURN
0252*				
0253*		*****		
0254*		* INTERNAL SUBROUTINES *		
0255*		*****		
0256*				
0257	00251	000000	READ NOP	CHANGE TO CHAR POINTER
0258	00252	004066	CLE,ELB	
0259	00253	076440R	STB IPTR	
0260	00254	076442R	STB SPTR	
0261	00255	002021	SSA,RSS	POST?
0262	00256	026261R	JMP ICHAR	YES, CHARACTERS
0263	00257	003004	CMA,INA	NEG WORDS, MAKE POS
0264	00260	001200	RAL	X2
0265*				

0266	00261	003004	ICHAR	CMA, INA	
0267	00262	072441R	STA	ICTR	
0268	00263	072443R	STA	SCTR	
0269*					
0270	00264	062453R	CONFI	LDA B16K	
0271	00265	016430R	JSB	XOTA	
0272*					
0273	00266	016433R	NEXTI	JSB XSFS	
0274	00267	102511	LIA	SC	
0275	00270	012456R	AND	B177	
0276	00271	052461R	CPA	CR	
0277	00272	026311R	JMP	CARET	
0278	00273	052462R	CPA	LF	
0279	00274	026266R	JMP	NEXTI	
0280	00275	052463R	CPA	BKSP	
0281	00276	026335R	JMP	BACK	

CONFIGURE INTERFACE

INPUT

CHECK FOR SPECIAL CHARACTERS

IGNORE LF

PAGE 0007 #01 SUBROUTINES TYPI, TYP A, TYP C, AND PNTL

0282	00277	052464R	CPA	DEL	
0283	00300	026325R	JMP	DELET	
0284	00301	066441R	LDB	ICTR	
0285	00302	006003	SZB, RSS		
0286	00303	026266R	JMP	NEXTI	
0287	00304	016347R	JSB	PUTC	
0288	00305	036440R	ISZ	IPTR	
0289	00306	036441R	ISZ	ICTR	
0290	00307	000000	NOP		

ICTR-0?

YES, DO NOT STORE

INCREMENT BUFFER POINTER

INCREMENT CHARACTER COUNT

0291	00310	026266R	JMP NEXTI	GO GET NEXT CHARACTER
0292*				
0293	00311	062454R	LDA B12K	CONFIGURE FOR OUTPUT
0294	00312	016430R	JSB XOTA	
0295	00313	062462R	LDA LF	SEND LF
0296	00314	016430R	JSB XOTA	
0297	00315	016433R	JSB XSFS	
0298	00316	106711	CLC SC	
0299	00317	062460R	LDA SP	STORE SPACE TO MARK END OF INPUT
0300	00320	016347R	JSB PUTC	
0301*				
0302	00321	062443R	IEXIT LDA SCTR	# CHARACTERS TO A
0303	00322	003004	CMA, INA	
0304	00323	042441R	ADA ICTR	
0305	00324	126251R	JMP READ, I	RETURN
0306*				
0307	00325	066465R	DELET LDB SLADD	PRINT REV SLASH, CR, LF
0308	00326	002404	CLA, INA	
0309	00327	016364R	JSB PRINT	
0310	00330	062442R	LDA SPTR	RESTORE INITIAL POINTER, COUNTER
0311	00331	072440R	STA IPTR	
0312	00332	062443R	LDA SCTR	
0313	00333	072441R	STA ICTR	
0314	00334	026264R	JMP CONFI	
0315*				
0316	00335	062441R	BACK LDA ICTR	COUNTER AT ORIGINAL VALUE?
0317	00336	052443R	CPA SCTR	YES, DO NOT MODIFY
0318	00337	026266R	JMP NEXTI	BACK UP COUNTER, POINTER
0319	00340	003400	CCA	NO
0320	00341	042441R	ADA ICTR	
0321	00342	072441R	STA ICTR	
0322	00343	003400	CCA	
0323	00344	042440R	ADA IPTR	
0324	00345	072440R	STA IPTR	
0325	00346	026266R	JMP NEXTI	

0351	00375	062454R	LDA B12K	CONFIGURE I/O CARD
0352	00376	016430R	JSB XOTA	
0353	00377	062445R	LDA CNTR	
0354	00400	002003	SZA,RSS	COUNT INITIALLY ZERO?
0355	00401	026410R	JMP NULL	YES, DO CR AND LF ONLY
0356*				
0357	00402	016420R	JSB GETC	
0358	00403	016430R	JSB XOTA	
0359	00404	016433R	JSB XSFS	
0360	00405	036444R	ISZ PNTR	
0361	00406	036445R	ISZ CNTR	
0362	00407	026402R	JMP NEXT0	
0363*				
0364	00410	062461R	LDA CR	DONE, DO CR
0365	00411	016430R	JSB XOTA	
0366	00412	016433R	JSB XSFS	
0367	00413	062462R	LDA LF	DO LF
0368	00414	016430R	JSB XOTA	
0369	00415	016433R	JSB XSFS	
0370	00416	106711	CLC SC	
0371	00417	126364R	JMP PRINT,I	RETURN
0372*				
0373	00420	000000	NOP	GET CHARACTER FROM PACKED BUFFER
0374	00421	066444R	LDB PNTR	
0375	00422	004065	CLE,ERB	
0376	00423	160001	LDA B,I	
0377	00424	002041	SEZ,RSS	
0378	00425	001727	ALF,ALF	
0379	00426	012456R	AND B177	
0380	00427	126420R	JMP GETC,I	
0381*				
0382	00430	000000	XOTA	EXECUTE OTA
0383	00431	102611	OTA SC	
0384	00432	126430R	JMP XOTA,I	
0385*				

0386	00433	000000	XSFS	NOP
0387	00434	103711	STC	SC, C
0388	00435	102311	SFS	SC
0389	00436	026435R	JMP	*-1
0390	00437	126433R	JMP	XSFS, I
0391*				
0392*				
0393*				

PAGE 0009 #01 SUBROUTINES TYPI, TYA, TYPC, AND PNTL

0394*	*****			
0395*	*****			
0396	00440	000000	IPTR	NOP
0397	00441	000000	ICTR	NOP
0398	00442	000000	SPTR	NOP
0399	00443	000000	SCTR	NOP
0400	00444	000000	PNTR	NOP
0401	00445	000000	CNTR	NOP
0402	00446	102300	SFS	SFS 0
0403	00447	000100	B100	OCT 100
0404	00450	000200	B200	OCT 200
0405	00451	001100	B1100	OCT 1100
0406	00452	003000	B3000	OCT 3000
0407	00453	160000	B16K	OCT 160000
0408	00454	120000	B12K	OCT 120000
0409	00455	177400	B1774	OCT 177400
0410	00456	000177	B177	OCT 177

0411	00457	000077	B77	OCT 77
0412	00460	000040	SP	OCT 40
0413	00461	000015	CR	OCT 15
0414	00462	000012	LF	OCT 12
0415	00463	000010	BKSP	OCT 10
0416	00464	000177	DEL	OCT 177
0417	00465	000466R	SLADD	DEF RSL
0418	00466	056000	RSL	OCT 56000
0419	00011		SC	EQU 11B
0420	00000		A	EQU 0
0421	00001		B	EQU 1
0422*				

00467	000024
00470	000053
00471	000055
00472	000056
00473	000057
00474	000072
00475	177720
00476	177766
00477	000002
00500	000001
00501	000177

0423
 ** NO ERRORS *TOTAL **RTE ASMB 750420**

END

```

0001 ASMB,R,L
0002 HED SUBROUTINES TYPI, TYPA, TYPC, AND PNTL
0003 NAM TYPI1
0004 ENT TYPI, TYPA, TYPC, PNTL
0005 EXT .ENTR, .DIO, .IOI, .END IO
0006 *****
0007 *****
0008 *****
0009 *****
0010 *****
0011 *****
0012 *****
0013 *****
0014 *****
0015 *****
0016 *****
0017 *****
0018 *****
0019 *****
0020 *****
0021 *****
0022 *****
0023 *****
0024 *****
0025 *****
0026 *****
0027 *****
0028 *****
0029 *****
0030 *****
0031 *****
0032 *****

      TITLE:  KEYBOARD INPUT AND SPECIAL LINE PRINTING

      MODULE:  TYPI      SOURCE:  TYPI1      VERSION:  X00
                OBJECT:  #TYPI1

      AUTHOR:  B. TASHJIAN      DATE:  1 DECEMBER 1976

      MODIFICATIONS:
      1:  ADD TYPC ROUTINE      7/14/77      W. BOWEN
      2:  ADD PNTL ROUTINE     7/28/77      W. BOWEN

      MODULE FUNCTIONS:

      TYPI - PERFORMS KEYBOARD INPUT OF INTEGER STRINGS.

              CALL TYPI (I,IFLG)

              I:  LOCATION WHERE ENTERED VALUE IS STORED.
              IFLG:  LOCATION WHERE ERROR FLAG IS STORED.

      TYPA - PERFORMS KEYBOARD INPUT OF ASCII CHARACTERS

```


0033 * * * * *
0034 * * * * *
0035 * * * * *
0036 * * * * *
0037 * * * * *
0038 * * * * *
0039 * * * * *
0040 * * * * *
0041 * * * * *
0042 * * * * *
0043 * * * * *
0044 * * * * *
0045 * * * * *
0046 * * * * *
0047 * * * * *
0048 * * * * *
0049 * * * * *
0050 * * * * *
0051 * * * * *
0052 * * * * *
0053 * * * * *
0054 * * * * *
0055 * * * * *
0056 * * * * *
0057 * * * * *
0058 * * * * *
0059 * * * * *
0060 * * * * *
0061 * * * * *
0062 * * * * *
0063 * * * * *
0064 * * * * *
0065 * * * * *
0066 * * * * *
0067 * * * * *

CALL TYPA (AS,IFLGA)
AS: LOCATION WHERE ASCII CHARACTERS ARE STORED.
IFLGA: LOCATION WHERE ERROR FLAG IS STORED.
IFLGA = NEGATIVE - ERROR
IFLGA = ZERO - NO INPUT
IFLGA = POSITIVE - GOOD INPUT
TYPC - PERFORMS KEYBOARD INPUT OF UP TO TEN (10) ASCII
CHARACTERS.
CALL TYPC (AC,IFLGC)
AC: LOCATION OF BUFFER WHERE THE ASCII
CHARACTERS ARE STORED.
IFLGC: LOCATION WHERE ERROR FLAG IS STORED.
(IFLGC VALUES SAME AS TYPA'S IFLGA VALUES)
PNTL - PRINTS ONE LINE OF FORMATTED DATA WITHOUT DOING
A CARRIAGE RETURN AND LINE FEED.
CALL PNTL (IPNTB,IBLG)
IPNTB: PRINT BUFFER ADDRESS
IBLG: BUFFER LENGTH

* TYPI ROUTINE *

NOP

0068	IFLG	NOP		
0069	TYPI	NOP		
0070		JSB	.ENTR	
0071		DEF	I	WAIT UNTIL OTHER I/O DONE
0072		JSB	ENDIO	
0073		DEF	X+1	
0074		CLF	0	ALLOW 20 CHARACTERS
0075		LDA	-D80	
0076		LDB	IBUF	
0077		JSB	READ	
0078		SZA	RSS	
0079		JMP	NOIN	NOTHING ENTERED
0080		CPA	-D80	
0081		JMP	ERROR	20 OR MORE ENTERED, ERROR
0082		CMA	INA	
0083		STA	CNTR	
0084		LDB	IBUF	
0085		CLE	ELB	
0086		STB	PNTR	
0087	*			
0088	CHKI	JSB	GETC	CHECK INPUT FOR LEGAL CHARS
0089		CPA	-B53	+ ?
0090		JMP	OK	
0091		CPA	-B55	- ?
0092		JMP	OK	
0093		CPA	-B56	. ?
0094		JMP	OK	
0095		CPA	-B57	/ ?
0096		JMP	OK	
0097		CPA	-B7E	= ?
0098		JMP	OK	
0099		ADA	-D-48	CHECK FOR RANGE 0-9
0100		SSA		
0101		JMP	ERROR	<0, ERROR
0102		ADA	-D-10	

0103		SSA, RSS	
0104		JMP ERROR	>9, ERROR
0105	* OK		
0106		ISZ PNTR	
0107		ISZ CNTR	ALL CHARS CHECKED?
0108		JMP CHKT	NO, LOOP
0109		CLA	
0110		CLB, INB	
0111		JSB .DIO.	
0112		DEF IBUF+1	
0113		ABS 0	
0114		DEF ERROR	
0115		JSB .IOI.	
0116		STA I, I	
0117		CLB, INB	SET IFLG = +1
0118	* TYPIX	STB IFLG, I	
0119		STF 0	
0120		JMP TYPI, I	
0121			
0122	* ERROR	CCB	SET IFLG = -1
0123		JMP TYPIX	
0124			
0125	* NOIN	CLB	SET IFLG = 0
0126		JMP TYPIX	
0127			
0128	*		*****
0129	*		* TYPA ROUTINE *
0130	*		*****
0131	*		
0132	*		
0133	AS	NOP	
0134	IFLGA	NOP	
0135	TYPA	NOP	
0136		JSB .ENTR	
0137		DEF AS	

0138	JSB ENDIO	WAIT UNTIL OTHER I/O DONE
0139	DEF *+1	
0140	CLF 0	
0141	LDA =D2	
0142	LDB IBUF	
0143	JSB READ	
0144	SZA,RSS	
0145	JMP NOINA	NOTHING ENTERED
0146	CPA =D1	
0147	RSS	
0148	JMP ERRA	ERROR, MORE THAN 1 CHAR ENTERED
0149	LDA IBUF+1	
0150	ALF,ALF	PUT CHARACTER IN RIGHT SIDE
0151	AND =B177	
0152	STA AS,I	
0153	CLB,INB	SET IFLG = +1
0154	* TYPAX STB IFLGA,I	
0155	STF 0	
0156	JMP TYRA,I	
0157		
0158	* ERRA CCB	SET IFLG = -1
0159	JMP TYRAX	
0160		
0161	* NOINA CLB	SET IFLG = 0
0162	JMP TYRAX	
0163		
0164	* IBUF	
0165	DEF *+1	
0166	BSS 21	
0167	*	*****
0168	*	* TYFC ROUTINE *
0169	*	*****
0170	*	
0171	* AC	
0172	NOP	

0173	IFLG	NOP				
0174	TYPC	NOP				
0175		JSB	.ENTR			FACILITATE ARGUMENT TRANSFER
0176		DEF	AC			
0177		JSB	ENDIO			WAIT UNTIL OTHER I/O IS DONE
0178		DEF	*+1			
0179		CLF	0			CLEAR I/O FLAG
0180	*					
0181		LDA	=D10			ALLOW 10 ALPHANUMERIC CHARACTERS
0182		LDB	AC			
0183		JSB	READ			READ INPUT DATA
0184		SZA	RSS			
0185		JMP	NOINC			NOTHING ENTERED
0186		CPA	=D11			
0187		RSS				
0188		JMP	ERRN			ERROR, MORE THAN 10 CHARACTERS ENTERED
0189		CLB	INB			SET IFLG = +1
0190	*					
0191		TYPCX	STB IFLG,I			
0192		STF	0			SET I/O FLAG
0193		JMP	TYFC,I			RETURN
0194	*					
0195		ERRN				SET IFLG = -1
0196		JMP	TYFCX			
0197	*					
0198		NOINC	CLB			SET IFLG = 0
0199		JMP	TYFCX			
0200	*					
0201	*					*****
0202	*					* PNTL ROUTINE *
0203	*					*****
0204	*					
0205		IPNTB	NOP			
0206		IBLG	NOP			
0207		PNTL	NOP			

0208	JSB .ENTR	FACILITATE ARGUMENT TRANSFER
0209	DEF IPNTB	
0210	JSB ENDIO	WAIT UNTIL OTHER I/O IS DONE
0211	DEF *+1	
0212	CLF 0	CLEAR I/O FLAG
0213	*	
0214	LDB IPNTB	LOAD PRINT BUFFER ADDRESS POINTER
0215	LDA IBLG,I	LOAD LENGTH OF PRINT BUFFER
0216	*	
0217	CLE,ELB	CHANGE TO CHARACTER POINTER
0218	STB PNTR	SET-UP CHARACTER POINTER
0219	SSA,RSS	IS BUFFER LENGTH POSITIVE ?
0220	JMP CHAR	YES, CHARACTERS
0221	CMA,INA	NEGATIVE WORDS, MAKE POSITIVE
0222	RAL	X2 (MAKE CHARACTERS)
0223	*	
0224	CMA,INA	SAVE COMPLEMENTED CHARACTER COUNT
0225	STA CNTR	
0226	LDA B12K	CONFIGURE I/O CARD
0227	JSB XOTA	
0228	LDA CNTR	COUNT INITIALLY ZERO ?
0229	SZA,RSS	YES, RETURN
0230	JMP PNTR,I	
0231	*	
0232	NEXTP	
0233	LDB PNTR	
0234	CLE,ERB	
0235	LDA B,I	
0236	SEZ,RSS	
0237	ALF,ALF	
0238	AND B177	
0239	*	
0240	JSB XOTA	SENT A CHARACTER
0241	JSB XSFS	SET CONTROL AND WAIT FOR FLAG
0242	ISZ PNTR	INCREMENT CHARACTER POINTER
	ISZ CNTR	INCREMENT CHARACTER COUNT, SKIP - 0

0243	JMP NEXTP		
0244	CLC SC	CLEAR INTERRUPTS	
0245	JMP PNTL,I	OUTPUT COMPLETE, RETURN	
0246			
0247	*****		
0248	* INTERNAL SUBROUTINES *		
0249	*****		
0250	*		
0251	READ		
0252	NOP		
0253	CLE,ELB	CHANGE TO CHAR POINTER	
0254	STB IPTR		
0255	STB SPTR		
0256	SSA,RSS	POST?	
0257	JMP ICHAR	YES, CHARACTERS	
0258	CMA,INA	NEG WORDS, MAKE POS	
0259	RAL	X2	
0260			
0261	* ICHAR		
0262	CMA,INA		
0263	STA ICTR		
0264	STA SCTR		
0265			
0266	* CONF I	CONFIGURE INTERFACE	
0267	LDA B16K		
0268	JSB XOTA		
0269			
0270	* NEXTI		
0271	JSB XSFS		
0272	LIA SC	INPUT	
0273	AND B17		
0274	CPA CR	CHECK FOR SPECIAL CHARACTERS	
0275	JMP CARET		
0276	CPA LF		
0277	JMP NEXTI	IGNORE LF	
0278	CPA BKSP		
0279	JMP BAQK		
0280	CPA DEL		
0281	JMP DELET		

0278	LDB ICTR		
0279	SZB, RSS		ICTR=0?
0280	JMP NEXTI		YES, DO NOT STORE
0281	JSB PUTC		
0282	ISZ IPTR		INCREMENT BUFFER POINTER
0283	ISZ ICTR		INCREMENT CHARACTER COUNT
0284	NOP		
0285	JMP NEXTI		GO GET NEXT CHARACTER
0286	*		
0287	CARET LDA B12X		CONFIGURE FOR OUTPUT
0288	JSB XOTA		
0289	LDA LF		SEND LF
0290	JSB XOTA		
0291	JSB XSFS		
0292	CLC SC		
0293	LDA SP		STORE SPACE TO MARK END OF INPUT
0294	JSB PUTC		
0295	*		
0296	IEXIT LDA SCTR		* CHARACTERS TO A
0297	CMA, INA		
0298	ADA ICTR		
0299	JMP READ, I		RETURN
0300	*		
0301	DELET LDB SLADD		PRINT REV SLASH, CR, LF
0302	CLA, INA		
0303	JSB PRINT		
0304	LDA SPTR		RESTORE INITIAL POINTER, COUNTER
0305	STA IPTR		
0306	LDA SCTR		
0307	STA ICTR		
0308	JMP CONF I		
0309	*		
0310	BACK LDA ICTR		COUNTER AT ORIGINAL VALUE?
0311	CPA SCTR		YES, DO NOT MODIFY
0312	JMP NEXTI		

BACK UP COUNTER, POINTER

NO

0313 CCA ICTR
0314 ADA ICTR
0315 STA ICTR
0316 CCA IPTR
0317 ADA IPTR
0318 STA IPTR
0319 JMP NEXTI

* PUTC

STORE CHARACTER

0320 NOP
0321 STA PRINT
0322 LDB IPTR
0323 CLE, ERB
0324 LDA B, I
0325 SEZ, RSS
0326 ALF, ALF
0327 AND B1774
0328 IOR PRINT
0329 SEZ, RSS
0330 ALF, ALF
0331 STA B, I
0332 JMP PUTC, I

* PRINT

CHANGE TO CHAR POINTER

0334 NOP
0335 CLE, ELB
0336 STB PNTR
0337 SSA, RSS
0338 JMP CHARS
0339 CMA, INA
0340 RAL

POS?
YES, CHARACTERS
NEG WORDS, MAKE POSITIVE
X2

* CHARS

CONFIGURE I/O CARD

0341 CMA, INA
0342 STA CNTR
0343 LDA B18K
0344 JSB X0TA
0345 LDA CNTR

0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347

0348	SZA, RSS	COUNT INITIALLY ZERO?
0349	JMP NULL	YES, DO CR AND LF ONLY
0350	* NEXT0	
0351	JSB GETC	
0352	JSB XOTA	
0353	JSB XSFS	
0354	ISZ PNTR	
0355	ISZ CNTR	
0356	JMP NEXT0	
0357	* NULL	
0358	LDA CR	DONE, DO CR
0359	JSB XOTA	
0360	JSB XSFS	
0361	LDA LF	DO LF
0362	JSB XOTA	
0363	JSB XSFS	
0364	CLC SC	
0365	JMP PRINT, I	RETURN
0366	* GETC	
0367	NOP	GET CHARACTER FROM PACKED BUFFER
0368	LDB PNTR	
0369	CLE, ERB	
0370	LDA B, I	
0371	SEZ, RSS	
0372	ALF, ALF	
0373	AND B177	
0374	JMP GETC, I	
0375	* XOTA	
0376	NOP	EXECUTE OTA
0377	OTA SC	
0378	JMP XOTA, I	
0379	* XSFS	
0380	NOP	SET CONTROL AND WAIT FOR FLAG
0381	STC SC, C	
0382	SFS SC	

```

0383      *
0384      *
0385      *
0386      *
0387      *
0388      *
0389      *
0390      *
0391      *
0392      *
0393      *
0394      *
0395      *
0396      *
0397      *
0398      *
0399      *
0400      *
0401      *
0402      *
0403      *
0404      *
0405      *
0406      *
0407      *
0408      *
0409      *
0410      *
0411      *
0412      *
0413      *
0414      *
0415      *
0416      *
0417      *

      JMP *-1
      JMP XSFS,I

      *****
      * LOCAL DATA *
      *****

      NOP
      NOP
      NOP
      NOP
      NOP
      NOP
      NOP
      SFS 0
      OCT 100
      OCT 200
      OCT 1100
      OCT 3000
      OCT 160000
      OCT 120000
      OCT 177400
      OCT 177
      OCT 77
      OCT 40
      OCT 15
      OCT 12
      OCT 10
      OCT 177
      DEF RSL
      OCT 56000
      EQU 11B
      EQU 0
      EQU 1

      IPTR
      ICTR
      SPTR
      SCTR
      PNTR
      CNTR
      SFS
      B100
      B200
      B1100
      B3000
      B16K
      B12K
      B1774
      B177
      B77
      SP
      CR
      LF
      BKSP
      DEL
      SLADD
      RSL
      SC
      A
      B
      *

      END

      BUFFER POINTER
      CHARACTER COUNTER
      START BUFFER POINTER
      TOTAL CHARACTER (AT START)

```

USER PROGRAM STARTUP ROUTINE - USTRT

1. FUNCTION. USTRT provides the capability of starting a keyboard program immediately when a Fourier system is started without the need for operator action. The keyboard program usually starts a user program.

2. CONSTRAINTS. None.

3. CALLING SEQUENCE. Called by: CLOAD

JMP USTRT

4. DESCRIPTION OF INPUT. None.

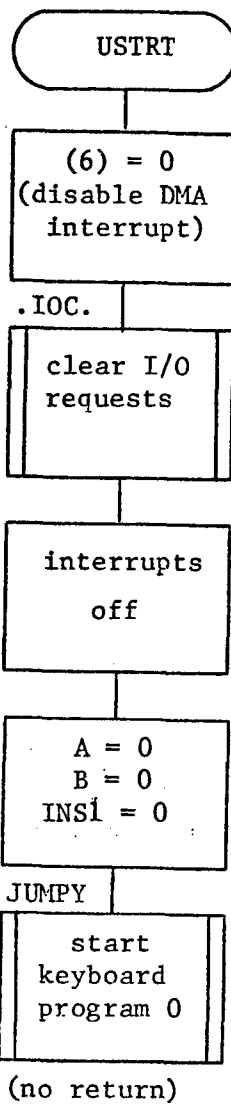
5. DESCRIPTION OF OUTPUT. None.

6. FILES USED. N/A

7. ERRORS AND RESTRICTIONS. None.

8. COMPUTER OPERATOR INSTRUCTIONS. N/A

9. DESCRIPTION OF PROCESSING. USTRT first sets address 6 to zero. This is necessary because USTRT is usually executed immediately after a core load, which leaves an interrupt request set in DMA 6. An intensive investigation disclosed no way of disabling this interrupt other than allowing it to occur and processing it. This is done by clearing address 6. Next, IOC is called to stop all I/O. Interrupts are disabled, A and B registers are cleared, location INSl is set to zero and subroutine JUMPY is called. JUMPY starts the keyboard program whose label is in INSl, in this case, 0.



USTRT T=00004 IS ON CR00001 USING 00002 BLKS R=0017

0001	ASMB,R,L				
0002	NAM	USTRT			
0003	ENT	USTRT			
0004	EXT	.IOC.,INS1,JUMPY,START			
0005	USTRT	CLA		DISABLE DMA	INTERRUPT HALT
0006	STA	6			
0007	JSB	.IOC.			
0008	OCT	0			
0009	CLC	0			
0010	CLB				
0011	CLA				
0012	STA	INS1		PUT JSB JUMP Y AT END OF RUNS	
0013	LDA	JUMP			
0014	STA	START			
0015	JMP	2		EXECUTE RUNS TO INIT WCS	
0016	JSB	JUMPY			
0017	END				

:

5.4 OLAAS USER PROGRAMS AND SUBROUTINES

The OLAAS System has various user programs and subroutines to accomplish certain users to aid in the analysis of the data. The following descriptions and listings are the programs and subroutines found in coreload 5:

- Y7891 - Subroutine
- Y7894 - User Program
- Y7896 - User Program
- Y7897 - User Program
- Y7898 - Subroutine
- Y7901 - User Program
- Y7902 - User Program
- Y7904 - User Program
- Y7906 - User Program
- Y7907 - User Program
- Y0221 - User Program
- BMDAT - Subroutine

Special User Program is not in coreload 5:

- SOUND - User Program

Y7894 USER PROGRAM

This program fills up Fourier Data Blocks with beamformer data. The user may select beams 1-16. The data is loaded on a major frame basis using the subroutine BMDAT. The user may enter up to $(8192/(\text{BLOCK SIZE}))$ beams to be stored in blocks $\emptyset$ to $(8192/\text{B.S.}) - 1$.

CALLING SEQUENCE:

CALLED BY: Y7906 User Program

Call Y7894 (L, JBUF)

L = 1 (Read in scaled beam data (4096 time domain words into block $\emptyset$)).

2 (Read in and sort data into blocks $\emptyset$ -15, 256 words each)

3 (Read, sort, and printout cell JBUF)

JBUF = (Printout cell)

3/18/77

YTS94S T-00004 IS ON CR00001 USING 00009 BLKS R-0090

```

0001 FTN4,L
0002 SUBROUTINE Y7894(L,JBUF)
0003 DIMENSION JBUF(6),ICHAN(16),IDAT(16),NQUAL(5)
0004 C
0005 C THIS ROUTINE FILLS UP FOURIER DATA BLOCKS WITH ACTIVE
0006 C BEAM DATA.
0007 C
0008 C JBUF(1) AND JBUF(2) ARE THE BEAMFORMER CHANNELS READ.
0009 C (ONLY 1 CHANNEL NEED BE READ IF DESIRED)
0010 C
0011 C DETERMINE BLOCK SIZE
0012 C
0013 CALL GETQ(0,NQUAL)
0014 IBS=NQUAL(1)
0015 C
0016 C DETERMINE THE NO. OF CHANNELS AND INITIALIZE THE ICHAN ARRAY,
0017 C WHICH IS THE ARRAY OF CHANNEL NUMBERS.
0018 C
0019 C SET UP DISCRETE CHANNEL TRANSFER.
0020 C
0021 C
0022 15 NC=L
0023 DO 22 I=1,NC
0024 ICHAN(I)=JBUF(I)
0025 C
0026 C CHECK TO SEE IF CHANNEL .LE. 16
0027 C
0028 IF(ICHAN(I).GE.17)GO TO 99
0029 CONTINUE
0030 22 C
0031 C CHECK FOR TOTAL DATA TRANSFER LESS THAN 8193 WORDS.
0032 C

```

0033	25	IF(IBS*NC.GE.8193)GO TO 99
0034	C	
0035	C	SET UP CALIBRATION.
0036	C	FIRST GET CURRENT VALUES.
0037	C	
0038	C	CALL GETQ(0,NQUAL)
0039	C	
0040	C	SET SCALE FACTOR TO 10**1, COORDINATES TO LINEAR,REC., TIME.
0041	C	
0042	C	NQUAL(2)=100B
0043	C	
0044	C	SET CALIBRATE WORD TO 16,383 TO CALIBRATE ERAPS DATA.
0045	C	
0046	C	NQUAL(3)=16383
0047	C	
0048	C	SET FREQUENCY CODE TO 1 KHZ FMAX.
0049	C	
0050	C	NQUAL(4)=45
0051	C	
0052	C	STORE AWAY IN ALL BLOCKS
0053	C	
0054		DO 60 I=1,NC
0055		IBLOK=I-1
0056	60	CALL PUTQ(IBLOK,NQUAL)
0057	C	
0058	C	SET UP FIRST 16 WORD ERAPS BEAM BLOCK TRANSFER..
0059	C	
0060	C	CALL BMDAT(IDAT,1)
0061	C	
0062	C	READ IN AND DISTRIBUTE IBS ERAPS BEAM FRAMES..
0063	C	
0064	C	DO 40 ICHNL=0,(IBS-1)
0065	C	
0066	C	GET 1 MINOR DATA FRAME
0067	C	

```

0068      CALL BMDAT(IDAT,2)
0069      C
0070      C DISTRIBUTE.
0071      C
0072      DO 30 IBLOK=0,(NC-1)
0073      IC=IBLOK+1
0074      IDATA=IDAT(ICHAN(IC))
0075      CALL PUTW(IBLOK,ICHL,IDATA,0)
0076      CONTINUE
0077      C
0078      C CLEAN UP.
0079      C
0080      CALL BMDAT(IDAT,3)
0081      RETURN
0082      C
0083      C ERROR RETURN.
0084      C
0085      99 WRITE(6,100)
0086      100 FORMAT("PARAMETER ERROR IN CALL TO Y7894")
0087      CALL WHAT
0088      RETURN
0089      END
0090      END$

```

:

BMDATS T=00004 IS ON CR0001 USING 00008 BLKS R=0084

0001	ASMB,R,L	HED BMDAT			
0002		NAM BMDAT			
0003		ENT BMDAT			
0004		EXT .ENTR			
0005					
0006	*				
0007	IDAT	NOP			
0008	KK	NOP			
0009	BMDAT	NOP			
0010		JSB .ENTR			
0011		DEF IDAT			
0012		LDA KK,I			
0013		CPA =D2			
0014		JMP KK2			
0015		CPA =D3			
0016		JMP KK3			
0017	*				
0018		CLF 0			
0019		LDA BUF1A			
0020		JSB DMAI			
0021		STC SC,C			
0022		STC 6,C			
0023		CLA,INA			
0024		OTA SC			
0025		JMP BMDAT,I			
0026	*				
0027	KK2	SFS 6			
0028		JMP *-1			
0029		CLC SC			
0030		CLC 6			
0031		LDA BUF2A			
0032		LDB BUF1A			

ENTRY - FORTRAN CALLABLE

KK-1, INTERRUPTS OFF
START INPUT
INIT DMA

START DMA

OUTPUT A 1 TO START
RETURN

KK-2, SWAP BUFFERS AND CONTINUE

SWAP BUFFERS

0033	STA BUF1A			
0034	STB BUF2A			
0035	JSB DMAI			
0036	STC SC,C			
0037	STC 6,C		START DMA	
0038	LDB -D-16		MOVE LAST SET OF DATA	
0039	LDA BUF2A			
0040	STA PTR1			
0041	LDA IDAT			
0042	STA PTR2			
0043	LDA PTR1,I			
0044	STA PTR2,I			
0045	ISZ PTR1			
0046	ISZ PTR2			
0047	ISZ B			
0048	JMP MUNXT		MOVE NEXT	
0049	JMP BMDAT,I		RETURN	
0050				
0051	SFS 6		KK-3, WAIT FOR DMA, STOP	
0052	JMP *-1			
0053	CLC SC			
0054	CLC 6			
0055	CLA SC			
0056	OTA SC			
0057	STF 0		INTERRUPTS ON	
0058	JMP BMDAT,I		RETURN	
0059				
0060	NOP		INIT DMA SUBROUTINE	
0061	STA B			
0062	LDA SLCT2			
0063	IOR -B120000			
0064	OTA 6			
0065	CLC 2			
0066	LDA B			
0067	IOR -B100000			

```

0068      OTA 2
0069      STC 2
0070      LDA -D-16
0071      OTA 2
0072      JMP DMAI,I
0073      *
0074      PTR1 NOP
0075      PTR2 NOP
0076      BUF1A DEF BUF1
0077      BUF2A DEF BUF2
0078      BUF1 BSS 16
0079      BUF2 BSS 16
0080      SLCT2 DEF SC
0081      SC EQU 22B
0082      A EQU 0
0083      B EQU 1
0084      END

```

```

:
```

Y7896

This program fills up Fourier Analyzer data blocks with active ERAPS stave data. The user may select stave numbers 1-16, header word 1 (17), and header word 2 (18). The data is loaded on a major frame basis using the subroutine DMAL. The user may enter up to (8192/Block size) channels to be stored in blocks 0 to (8192/B.S.) - 1.

Calling sequence:

Y7896 K1 K2 K3

K1 = First Channel

K2 = Second Channel

Y7896S T-00004 IS ON CR00001 USING 00011 BLKS R-0010

```
0001 FTN4,L
0002 SUBROUTINE Y7896 (NC,JBUF)
0003
0004 *****
0005 *
0006 * TITLE: LOAD A FOURIER DATA BLOCK
0007 *
0008 * MODULE: Y7896 SOURCE: Y7896S VERSION: X00
0009 *
0010 * AUTHOR: D. GOODMAN DATE: 23 AUGUST 1976
0011 *
0012 *
0013 * MODULE FUNCTION:
0014 * THE Y7896 SUBROUTINE LOADS A FOURIER DATA BLOCK WITH
0015 * ACTIVE ERAPS DATA. CHANNELS 1 -16 ARE DATA CHANNELS.
0016 * CHANNEL 17 IS THE FIRST HEADER WORD AND CHANNEL 18 IS
0017 * THE SECOND HEADER WORD.
0018 *
0019 * CALLING SEQUENCE: CALLED BY - Y7906
0020 *
0021 * CALL Y7896 (NC,JBUF)
0022 *
0023 * INPUT PARAMETERS: NC = NUMBER OF DATA BLOCK
0024 * JBUF = INPUT BUFFER
0025 *
0026 *****
0027
0028 DIMENSION JBUF(6),ICHAN(18),IDAT(72),NQUAL(5),ICHN(6)
0029
0030 DETERMINE BLOCK SIZE - STORE AS "IBS"
0031
0032 CALL GETQ(0,NQUAL)
```


0033	IBS=NQUAL(1)	
0034	CHECK B.S. - CHANNEL NO. PRODUCT	C
0035		C
0036		C
0037	IF (IBS * NC .GE. 8193) GO TO 95	
0038	IF (NC.EQ.0)GO TO 99	
0039	IF (NC.GT.6)GO TO 99	
0040		C
0041	REFORMAT KEYBOARD INPUT DATA	C
0042		C
0043	DO 26 I = 1,NC	
0044	IF(JBUF(I) .EQ. 17) GO TO 23	
0045	IF(JBUF(I) .EQ. 18) GO TO 24	
0046	ICHAN(I) = JBUF(I) + 2	
0047	GO TO 25	
0048	23 ICHAN(I) = 1	
0049	GO TO 25	
0050	24 ICHAN(I) = 2	
0051	25 IF(ICHAN(I) .GT. 18) GO TO 99	
0052	26 CONTINUE	
0053		C
0054	SET UP CALIBRATION	C
0055		C
0056	SET SCALE FACTOR TO 10**1,COORDINATES TO LINEAR,REC,TIME	C
0057		C
0058	NQUAL(2)=100B	
0059		C
0060	SET CALIBRATOR WORD TO 16383 TO CALIBRATE ERAPS DATA	C
0061		C
0062	NQUAL(3)=16383	
0063		C
0064	SET FREQUENCY CODE TO 1 KHZ FMAX	C
0065		C
0066	NQUAL(4)=45	
0067		C

```

0068 C STORE AWAY INTO ALL BLOCKS
0069 C
0070 DO 60 I=1,NC
0071 IBLOK=I-1
0072 60 CALL PUTQ(IBLOK,NQUAL)
0073 C
0074 C SET UP FIRST MAJOR FRAME TRANSFER
0075 C
0076 CALL DMA1(IDAT,1)
0077 C
0078 C READ IN AND DISTRIBUTE "IBS" MAJOR FRAMES
0079 C
0080 DO 75 KL = 1,NC
0081 75 ICHN(KL) = 0
0082 C
0083 C GET ONE MAJOR DATA FRAME
0084 C
0085 70 CALL DMA1(IDAT,2)
0086 C
0087 C DISTRIBUTE INTO BLOCKS
0088 C
0089 DO 80 IC = 1,NC
0090 IBLOK=IC-1
0091 DO 80 JJ = 1,4
0092 INDEX = ((JJ-1) * 18) + ICHN(IC)
0093 IDATA = IDAT(INDEX)
0094 ICHNL = ICHN(IC)
0095 CALL PUTW(IBLOK,ICHNL,IDATA,0)
0096 ICHN(IC) = ICHN(IC) +1
0097 IF(ICHN(NC) .GE. IBS) GO TO 81
0098 80 CONTINUE
0099 IF(ICHNL .LT. IBS) GO TO 70
0100 81 CALL DMA1(IDAT,3)
0101 C
0102 C TURN OFF DMA, TURN ON INTERRUPTS

```

```

C103 C
C104 C
C105 C
C106 C
C107 C
C108
C109
C110
C111
C112
C113
C114
C115
:

RETURN
ERROR RETURN
95 WRITE(6,101)
101 FORMAT(5X,"** ERROR, BLOCK SIZE TOO LARGE **")
RETURN
99 WRITE(6,100)
100 FORMAT(5X,"** ERROR IN PARAMETER ENTRY **")
RETURN
END
END$

```

Y7897 USER PROGRAM

This routine reads in 2048 words of multiplexed beam data, sorts it into 16 blocks of 128 words each, Hanns and FFT's each block, normalizes each block to channel 1, and prints the log magnitude and phase of each channel relative to channel 1.

CALLING SEQUENCE:

Y7897 JBUF(1)

- JBUF(1) = 1 - Read in scaled beam data
(2048 words to block 1)
- 2 - Reads in and sorts to block 0-15
(128 words each)
- 3 - FFT's the data

SUBROUTINES CALLED:

- Y7891 - Reads in scale Beam Data (2048 words to block 1)
- Y7898 - Sorts multiplexed Beam data.

Y7897S T=00004 IS ON CR00001 USING 00007 BLKS R=0054

```
00001 FTN4.L
00002 SUBROUTINE Y7897(I1,JBUF)
00003 DIMENSION JBUF(6),KBUF(1),NQUAL(5)
00004
00005 JBUF(1) = 1 TO READ IN SCALED BEAM DATA (2048 WDS TO BL 1)
00006 " 2 TO READ IN AND SORT (TO BL 0-15, 128 WDS EA)
00007 " 3 TO FFT THE DATA
00008
00009 THIS ROUTINE READS IN 2048 WORDS OF MULTIPLEXED BEAM DATA,
00010 SORTS IT INTO 16 BLOCKS OF 128 WORDS, HANNING AND FFTS
00011 EACH BLOCK, AND DISPLAYS THE RESULTS.
00012
00013 IGO=JBUF(1)
00014 BLOCKSIZE=2048 AND TIME DOMAIN
00015
00016 NQUAL(1)=2048
00017 NQUAL(2)=1008
00018 NQUAL(3)=16383
00019 NQUAL(4)=45
00020 NQUAL(5)=0
00021 CALL PUTQ(0,NQUAL)
00022
00023 READ IN DATA
00024
00025 I=1
00026 KBUF(1)=01
00027 CALL Y7891(I,KBUF)
00028 RETURN FOR READ IN ONLY
00029 IF(IGO .EQ.1)RETURN
00030
00031 STORE 1
00032 CALL KYBD(054076B,1)
```

```

0033 C SORT
0034 CALL Y7898(1,KBUF)
0035 IF(IGO.EQ.2)RETURN
0036 C BLOCKSIZE =128
0037 NQUAL(1)=128
0038 DO 7 I=0,16
0039 CALL PUTQ(I,NQUAL)
0040 C RETURN FOR READ-IN AND SORT ONLY
0041 C HANN AND FFT EACH BLOCK
0042 DO 10 I=0,16
0043 CALL KYBD(044061B,I)
0044 CALL KYBD(043040B,I)
0045 CALL KYBD(052114B,I)
0046 10 CHANGE BLOCK SIZE TO 2048
0047 CALL GETQ (0,NQUAL)
0048 NQUAL(1) = 2048
0049 CALL PUTQ (0,NQUAL)
0050 RETURN
0051 END
0052 END$

```

:

Y7891 USER SUBROUTINE

Reads in 2048 words of multiplexed Beam Data into data block 1.

CALLING SEQUENCE:

CALLED BY: Y7897 User Program

Y7891 (K1, K2)

Y7891S T=00004 IS ON CR00001 USING 00004 BLKS R=0040

0001	ASMB,R,L			
0002	NAM Y7891			
0003	ENT Y7891			
0004	EXT .ENTR,DTAD0			
0005	NPAR			
0006	N1			
0007	Y7891			
0008	JSB .ENTR			
0009	DEF NPAR			
0010	CLF 0		INT OFF	
0011	CLA			
0012	OTA SC2			
0013	LDA SLCT2		GET SELECT CODE	
0014	IOR -B120000			
0015	OTA 6		OUTPUT CW1	
0016	CLC 2			
0017	LDA DTAD0			
0018	IOR -B100000			
0019	OTA 2		OUTPUT CW2	
0020	STC 2			
0021	LDA -D-2048			
0022	OTA 2		OUTPUT CW3	
0023	STC SC2,C		START SC2	
0024	STC 6,C		START DMA	
0025	CLA,INA			
0026	OTA SC2		SET OUTPUT BIT 0 ON SC2	
0027	SFS 6		WAIT FOR DMA	
0028	JMP *-1			
0029	CLC SC2			
0030	CLC 6			
0031	CLA			
0032	OTA SC2		CLEAR OUTPUT BIT 0 ON SC2	


```

0033      INT ON
0034      JMP Y7891,I RETURN
0035      NOP
0036      NOP
0037      NOP
0038      DEF SC2
0039      EQU 22B
0040      END

```

Y7898 USER PROGRAM SUBROUTINE

This subroutine is used with Y7897 to sort 2048 multiplexed Beam words in block 1 and put sorted data into block 0. The data is sorted so that consecutive channels occupy adjacent 128 word cells in the 2048 data block.

CALLING SEQUENCE:

CALLED BY: Y7897 User Program

Y7898 (K1)

K1 - Dummy variable, not used in subroutine.

Y78988 T=00004 IS ON CR00001 USING 00003 BLKS R=0019

```
0001 FTN4,L
0002 SUBROUTINE Y7898(II,JBUF)
0003 DIMENSION JBUF(1)
0004 C
0005 C
0006 C THIS ROUTINE SORTS 2048 MULTIPLEXED BEAM WORDS
0007 C (IN BLOCK 1) AND PUTS SORTED DATA IN BLOCK 0.
0008 C K=WORD COUNTER FOR BLOCK 0
0009 K=0
0010 DO 20 I=0,15
0011 DO 20 J=0,127
0012 ICHNL=I+16*J
0013 CALL GETW(1,ICHNL,IDTAR,IDTAI)
0014 ICHNL=K
0015 CALL PUTW(0,ICHNL,IDTAR,IDTAI)
0016 K=K+1
0017 RETURN
0018 END
0019 END$
```

:

Y7901

This subroutine accesses disc-loaded Beamformer data as collected by the assembly language routine RDATA.

Calling sequence:

Y7901 K1 K2 K3

K1 = Ping number

K2 = Beam number

K3 = Index number (1 - 128 word; Index = 0.192 sec)

Y79015-T-00004 IS ON CRO0001 USING 02004 BLKS R=0027

0001 FTN4,L
0002 C
0003 C
0004 C
0005 C
0006 C
0007 C
0008 C
0009 C
0010 C
0011 C
0012 C
0013 C
0014 C
0015 C
0016 C
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032

* THIS SUBROUTINE ACCESSES DISC-LOADED
* BEAM-FORMER DATA AS COLLECTED BY THE
* ROUTINE RDATA.
* USER-ENTERED PARAMETERS-
* KBUF(1) = PING NUMBER
* KBUF(2) = BEAM NUMBER
* KBUF(3) = INDEX NUMBER

SUBROUTINE Y7901(KK,KBUF)
DIMENSION KBUF(3),NQUAL(5)
K1=KBUF(1)
K2=KBUF(2)
K3=KBUF(3)

SET SCALE FACTORS (10**1, LINEAR, RECT., TIME)

CALL GETQ(0, NQUAL)
K4= NQUAL(1)
NQUAL(2)=1000
NQUAL(3)=15333
NQUAL(4)=45
NQUAL(5)=0
CALL PUTQ(0,NQUAL)

10-4-76

0032 C
0034 .C
0035
0036
0037
0038
0039
0040
0041

READ IN DATA
CALL RDATA(K1,K2,K3,K4,K5)
IF(K5.EQ.0)RETURN
WRITE(2,100)
FORMAT(5X,"STATUS IS NON-ZERO, NO DATA RETURNED")
RETURN
END
ENDC

100

10-4-76

:

Y7902

The program prints out pertinent header data concerning Beamformer data recorded on the disc.

Calling sequence:

Y7902 K1

K1 = Ping number

Y7902S T=00004 IS ON CR00001 USING 00008 BLKS R=0058

FTN4,L
0001 C
0002 C
0003 C
0004 C
0005 C
0006 C
0007 C
0008 C
0009 C
0010 C
0011 C
0012 C
0013 C
0014 C
0015 C
0016 C
0017 C
0018 C
0019 C
0020 C
0021 C
0022 C
0023 C
0024 C
0025 C
0026 C
0027 C
0028 C
0029 C
0030 C
0031 C
0032 C

```
*****
* THIS PROGRAM READS IN HEADER DATA BASED ON
* THE BEAMFORMER INFORMATION RECORDED ON THE
* DISC. CALLING SEQUENCE : Y7902 IPNG
* (IPNG IS THE PARTICULAR PING NUMBER)
*****
*****
SUBROUTINE Y7902(L,JBUF)
INTEGER DATA
DIMENSION JBUF(1),DATA(16),MASK(4),IDTA(6)
MASK(1) = 17B
MASK(2) = 360B
MASK(3) = 7400B
MASK(4) = 170000B
CALL HEADR(1,DATA)
IF(JBUF(1).GT. DATA(10)) GO TO 700
IPING=JBUF(1)
CALL HEADR(IPING,DATA)
CHECK FOR LOCATION OF SOURCE
IF(DATA(9).EQ. 1) GO TO 70
WRITE(2,103)
103 FORMAT(5X,"SOURCE IS TAPE RECORDER")
GO TO 80
70 WRITE(2,104)
104 FORMAT(5X,"SOURCE IS ON-LINE")
80 CONTINUE
```

3-18-77


```

0033 C
0034 C
0035 C
0036     CALCULATE THE TIME
0037     NWORD = DATA(2)
0038     DO 20 I = 1,4
0039     MSK = MASK(I)
0040     20 IDTA(I) = IAND(MSK,NWORD)
0041     NWORD = DATA(1)
0042     DO 30 I = 1,2
0043     MSK = MASK(I)
0044     J = I + 4
0045     30 IDTA(J) = IAND(MSK,NWORD)
0046     KT1 = ((IDTA(6) / 208) * 10) + IDTA(5)
0047     KT2 = ((IDTA(4) / 10000B) * 10) + (IDTA(3) / 400B)
0048     KT3 = ((IDTA(2) / 208) * 10) + IDTA(1)
0049 C
0050 C
0051 C
0052     WRITE THE HEADER
0053     WRITE(2,100)KT1,KT2,KT3,DATA(3)
0054     100 FORMAT(5X,"TIME = ",I2," ",I2," ",I2,5X,"COMPASS HEADING = ",
0055     1I4," DEGREES")
0056     WRITE(2,101) DATA(5),DATA(6),DATA(8),DATA(7),DATA(10)
0057     101 FORMAT(5X,"FIRST BEAM = ",I3,2X,"AT",I4," DEGREES",/,5X,
0058     1"BEAMS RECORDED = ",I4,5X,"PING NUMBER",I4," OF",I4)
0059     WRITE(2,102) DATA(12),DATA(11)
0060     102 FORMAT(5X,"RECORDING TIME LENGTH",I3," SECONDS AFTER DELAY 0
0061     1I3," SECONDS")
0062     RETURN
0063     700 WRITE(2,701)JBUF(1)
0064     701 FORMAT("PING",I3," WAS NOT RECORDED")
0065     RETURN
0066     END
0067     END$

```

Y7904

This allows the user to print out the decimal value of each cell in a chosen block. The cells printed out are determined by answers the user gives to questions.

Calling sequence:

Y7904 (No additional parameters necessary)

3/24/77
Y7904S T-00004 IS ON CR00001 USING 00007 BLKS R-0046

```
0001 FTN4,L
0002 SUBROUTINE Y7904(L,JBUF)
0003 C
0004 C THIS ROUTINE IS USED TO EXAMINE THE CONTENTS OF A
0005 C FOURIER DATA BLOCK. USER-ENTERED PARAMETERS DIRECT THE
0006 C PROCESSING.
0007 C
0008 DIMENSION JBUF(2),KOUT(8),NQUAL(5)
0009 DATA IFCN/1/, IBS/1024/, IBLK/0/, IST/0/, KWRDS/150/
0010 WRITE(2,21)
0011 21 FORMAT("ENTER APPROPRIATE NUMBER",/,5X,"1 - WRITE EXISTING "
0012 1"FOURIER DATA BLOCK",/,5X,"2 - WRITE DISC-STORED DATA BLOCK"
0013 CALL INPT(IFCN,1,2)
0014 WRITE(2,22)
0015 22 FORMAT("ENTER BLOCK SIZE")
0016 CALL INPT(IBS,64,4096)
0017 CALL GETQ(0,NQUAL)
0018 NQUAL(1) = IBS
0019 CALL PUTQ(0,NQUAL)
0020 IF(IFCN.EQ. 2) GO TO 23
0021 WRITE(2,24)
0022 24 FORMAT("ENTER NUMBER OF BLOCK TO BE PRINTED OUT")
0023 KMAX = (8192/IBS) - 1
0024 CALL INPT(IBLK,0,KMAX)
0025 GO TO 29
0026 IBLK = 0
0027 29 WRITE(2,30)
0028 30 FORMAT("ENTER FIRST WORD TO BE PRINTED")
0029 CALL INPT(IST,0,IBS)
0030 WRITE(2,31)
0031 31 FORMAT("ENTER NUMBER OF CONSECUTIVE WORDS TO BE PRINTED")
0032 CALL INPT(KWRDS,1,IBS)
```

0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046

IF(IFCN.EQ. 1) GO TO 25
CALL KYBD(2HMS,31,0)
CALL KYBD(2HMS,11)
25 DO 26 K=IST,(KURDS+IST-1),8
DO 27 ICHNL=K,(K+8)
INDEX=ICHNL-K+1
CALL GETW(0,ICHNL,IDTA,INOP)
27 KOUT(INDEX)=IDTA
WRITE(2,28)K,(KOUT(I),I=1,8)
28 FORMAT(,"(",I4," ",8(1X,I6))
26 CONTINUE
32 RETURN
END
END\$

Y7906

This program allows the operator to select stave data, beam data, or analog input for analysis. This data is Fourier transformed, then averaged and Hanned a preset number of times (entered by the user). The resulting averaged power spectrum is then displayed.

Calling sequence:

Y7906 (No additional parameters necessary)

8/24/77

PROCESS T-00004 IS ON CR00100 USING 00005 BLKS R-0018

```
0001 FTN4
0002 SUBROUTINE Y7906(L,JBUF)
0003 *****
0004 *****
0005 *****
0006 *****
0007 *****
0008 *****
0009 *****
0010 *****
0011 *****
0012 *****
0013 *****
0014 *****
0015 *****
0016 *****
0017 *****
0018 *****
0019 *****
0020 *****
0021 *****
0022 *****
0023 *****
0024 *****
0025 *****
0026 *****
0027 *****
0028 *****
0029 *****
0030 *****
0031 *****
0032 *****

      THIS ROUTINE ALLOWS THE OPERATOR TO ENTER
      VARIOUS PARAMETERS IN ORDER TO REDUCE THE
      NUMBER OF KEYBOARD ENTRIES NEEDED FOR AVERAGING
      DEU CHANNEL DATA. THE USER ENTERED PARAMETERS
      ARE--
          STAVE,BEAM OR ANALOG INPUT
          CHANNEL NO. FOR STAVE AND BEAM ANALYSIS

      OTHER PARAMETERS ENTERED LATER ARE AS FOLLOWS:
          NUMBER OF AVERAGES REQUIRED
          BLOCK SIZE
          NUMBER OF HANNINGS NEEDED

      *****

      DIMENSION JBUF(1),NQUAL(5),LBUF(6),NBUF(6)
      DATA IBEL/000007B/,IAN/1/,IAV/10/,IHAN/1/,ISTU/5/,
      CIBEAM/13/,IBS/1024/

      SET BLOCK SCALE FACTORS

      CALL GETQ(0,NQUAL)
      NQUAL(1) = 1024
      NQUAL(2) = 1008
      NQUAL(3) = 16383
      NQUAL(4) = 45
      CALL PUTQ(0,NQUAL)
```

```

0033 C
0034 C
0035 C
0036
0037 WRITE(1,5)
0038 5 FORMAT('PLEASE ENTER THE TYPE OF ANALYSIS DESIRED: ',/,
0039 1*'FOR STAVE INPUT, ENTER 1 ',/, 'FOR BEAM INPUT, ENTER 2 ',
0040 2*', 'FOR ANALOG INPUT, ENTER 3 ')
0041 CALL INPT(IAN,1,3)
0042 IF(IAN.EQ.3)GO TO 10
0043 IF(IAN.EQ.2)GO TO 8
0044 6 WRITE(1,7)
0045 7 FORMAT('ENTER THE STAVE NUMBER: ')
0046 CALL INPT(ISTV,1,16)
0047 LBUF(1)=ISTV
0048 GO TO 14
0049 8 WRITE(1,9)
0050 9 FORMAT('ENTER THE BEAM NUMBER: ')
0051 CALL INPT(IBEAM,1,16)
0052 NBUF(1)=IBEAM
0053 GO TO 14
0054 10 WRITE(1,11)
0055 11 FORMAT('PLEASE SET THE SAMPLE RATE AND THE INPUT LEVEL ',/,
0056 C'CONTROLS ON THE ANALOG TO DIGITAL CONVERTER. ')
0057 14 WRITE(1,15)
0058 15 FORMAT('PLEASE ENTER THE BLOCK SIZE DESIRED: ')
0059 16 N=6
0060 CALL INPT(IFS,64,4096)
0061 17 IF(N.EQ.13)GO TO 18
0062 IHH=(2**N)
0063 IF(IFS.EQ.IHH)GO TO 22
0064 N=N+1
0065 GO TO 17
0066 18 WRITE (1,19)
0067 19 FORMAT('YOU STUPID TURKEY!!!',/, 'TRY AGAIN!!!')
DO 20 JK=1,10,1

```

```

0068 20 WRITE(1,21)IBEL
0069 21 FORMAT(1A2)
0070 CONTINUE
0071 GO TO 16
0072 22 CALL GETQ(0,NQUAL)
0073 NQUAL(1) = IBS
0074 CALL PUTQ(0,NQUAL)
0075
0076 C
0077 C
0078 C
0079 C
0080 C
0081 C
0082 C
0083 C
0084 C
0085 C
0086 C
0087 C
0088 C
0089 C
0090 C
0091 C
0092 C
0093 C
0094 C
0095 C
0096 C
0097 C
0098 C
0099 C
0100 C
0101 C
0102 C

      GET NUMBER OF AVERAGES AND NUMBER OF HANNINGS DESIRED

      WRITE(1,23)
      23 FORMAT('ENTER THE NUMBER OF TIMES THAT YOU WANT ME TO ',/,
        C'AVERAGE THE INPUT: ')
      CALL INPT(IAU,1,10000)
      WRITE(1,24)
      24 FORMAT('HOW MANY HANNINGS WITHIN EACH AVERAGING LOOP??')
      CALL INPT(IHAN,0,10000)
      WRITE(1,200)
      200 FORMAT('WORKING.....')

      CLEAR OUT THE BLOCKS

      CALL KYBD(041514B,0)
      CALL KYBD(041514B,1)

      START OF AVERAGING LOOP

      DO 45 I=1,IAU,1
      IF (IAN.EQ.3)GO TO 25
      IF (IAN.EQ.2)GO TO 245
      CALL Y7896(1,LBUF)
      GO TO 27
      245 CALL Y7894(1,NBUF)
      GO TO 27

```


3103	C		
3104	C	CALL KEYBOARD ENTRIES	
3105			
3106		25 CALL KYBD(051101B,0)	
3107		27 IF(IHAN.EQ.0)GO TO 29	
3108		DO 28 J=1,IHAN,1	
3109		28 CALL KYBD(044061B,0)	
3110		CONTINUE	
3111		29 CALL KYBD(043040B,0)	
3112		CALL KYBD(2HX-,0)	
3113		CALL KYBD(2HA+,1)	
3114	C		
3115	C	STORE RUNNING TOTAL IN BLOCK ONE	
3116			
3117		CALL KYBD(2HX>,1)	
3118		45 CONTINUE	
3119	C		
3120	C	LOAD TOTAL INTO BLOCK ZERO	
3121			
3122	C	CALL KYBD(2HX<,1)	
3123	C	DIVIDE TOTAL BY NUMBER OF AVERAGES	
3124	C		
3125		CALL KYBD(2H: ,0,IAU)	
3126	C		
3127	C	TAKE LOG MAGNITUDE OF BLOCK ZERO	
3128	C		
3129		CALL KYBD(052114B,0)	
3130		WRITE(1,46)	
3131		46 FORMAT("DONE!!!")	
3132		WRITE(1,47)IBEL	
3133		47 FORMAT(1A2)	
3134	C		
3135	C	DISPLAY BLOCK ZERO	
3136	C		
3137		CALL KYBD(042040B,0)	

RETURN
50 END
END\$

3138
3139
0140

:

Y7907 USER PROGRAM

This subroutine tests channel response for all 16 DEU channels. However, all channel inputs must be tied to the same signal source. The user entered parameters are as follows:

F1 or F2 DEU

Test vessel pressure (PSI)

Data Channel block number (Use Y7896 or Y7906 along with Y0221 to determine)

CALLING SEQUENCE:

Y7907 (stand alone routine)

SUBROUTINES CALLED:

Y7896 & INPT

Y7907S T-00004 IS ON CR00001 USING 00020 BLKS R-0115

```
0001 FTN4,L
0002 SUBROUTINE Y7907(L,JBUF)
0003 DIMENSION JBUF(5),NQVAL(5)
0004 DATA DELT/8.33333333333E-5/,IBEL/000007B/,IFR/2/,IFRE1/0000/,
0005 1IPRES/14/,IBLK/000/,IREF/0/,INUM/15/,NQVAL(1)/1024/,
0006 2IDA/103/,IDB/257/,IDC/411/,IDD/408/,IDE/254/,IDF/101/
0007
0008 *****
0009 *
0010 * THIS ROUTINE TESTS CHANNEL RESPONSE FOR
0011 * ALL 16 DEU CHANNELS. HOWEVER, ALL CHANNEL
0012 * INPUTS MUST BE TIED TO THE SAME SIGNAL
0013 * SOURCE. THE USER ENTERED PARAMETERS ARE:
0014 * F1 OR F2 DEU
0015 * INPUT FREQUENCY
0016 * TEST VESSEL PRESSURE (PSI)
0017 * DATA CHANNEL BLOCK NUMBER
0018 * (USE Y7896 OR Y7906 ALONG WITH Y0221
0019 * TO DETERMINE)
0020 *
0021 *****
0022 *
0023 CALL PUTQ(0,NQVAL)
0024 WRITE(1,10)
0025 10 FORMAT("WHAT IS THE CENTER FREQUENCY FOR THE DEU UNDER TEST?
0026 1"FOR AN F1 DEU,ENTER 1",/,"FOR AN F2 DEU,ENTER 2".)
0027 CALL INPT(IFR,1,2)
0028 IF(IFR.EQ.1)IFRE=((27960+40)/7)
0029 IF(IFR.EQ.2)IFRE=((1596+4)*3)
0030 WRITE(1,11)
0031 11 FORMAT("PLEASE ENTER THE SIGNAL FREQUENCY: ")
0032 CALL INPT(IFRE1,3700,5100)
```

```

0033 WRITE(1,145)
0034 145 FORMAT("ENTER THE PRESSURE LEVEL FOR THIS TEST (PSI):")
0035 CALL INPT(IPRES,0,10000)
0036 IFRE3=IFRE1-IFRE
0037 IF(IFR.EQ.2)GO TO 147
0038 IFRE2=166+IFRE3
0039 GO TO 148
0040 IFRE2=166-IFRE3
0041 IF(IFR.EQ.2)GO TO 110
0042 IF(IFRE1.EQ.(IFRE-100))IBLK=IDA
0043 IF(IFRE1.EQ.IFRE)IBLK=IDB
0044 IF(IFRE1.EQ.(IFRE+100))IBLK=IDC
0045 GO TO 112
0046 IF(IFRE1.EQ.(IFRE-100))IBLK=IDD
0047 IF(IFRE1.EQ.IFRE)IBLK=IDE
0048 IF(IFRE1.EQ.(IFRE+100))IBLK=IDF
0049 112 IF(IFRE1.NE.(IFRE-100)).AND.IFRE1.NE.IFRE.AND.IFRE1.NE.
0050 1(IFRE+100))GO TO 114
0051 GO TO 215
0052 114 WRITE(1,12)
0053 12 FORMAT("NOW ENTER THE CHANNEL BLOCK NUMBER FOR THIS ",/,
0054 1"FREQUENCY:")
0055 CALL INPT(IBLK,0,500)
0056 215 WRITE(1,115)
0057 115 FORMAT("DO YOU WISH TO SPECIFY THE REFERENCE CHANNEL ?",/,
0058 1"(1 OR 0)")
0059 CALL INPT(IREF,0,1)
0060 IF(IREF.EQ.1)GO TO 120
0061 INUM=1
0062 GO TO 125
0063 120 WRITE(1,121)
0064 121 FORMAT("ENTER THE REFERENCE CHANNEL NUMBER:")
0065 CALL INPT(INUM,1,16)
0066 DO 75 JK=1,16,1
0067 JBUF(1)=JK

```

```

0068 JBUF(2)=INUM
0069 CALL Y7896(2,JBUF)
0070 IF(JK.EQ.1)GO TO 16
0071 GO TO 18
0072 16 CALL CHOUT(33B)
0073 CALL CHOUT(14B)
0074 WRITE(1,17)
0075 17 FORMAT(//,25X,"DEU CHANNEL RESPONSE")
0076 IF(IFRE1.LT.IFRE)GO TO 175
0077 IF(IFRE1.EQ.IFRE)GO TO 177
0078 GO TO 179
0079 175 WRITE(1,176)IFR,IFRE3,IPRES
0080 176 FORMAT(//,10X,"FREQUENCY=F",I1,I4,10X,"PRESSURE=",I4," PSI")
0081 GO TO 200
0082 177 WRITE(1,178)IFR,IPRES
0083 178 FORMAT(//,10X,"FREQUENCY=F",I1,15X,"PRESSURE= ",I5," PSI")
0084 GO TO 200
0085 179 WRITE(1,180)IFR,IFRE3,IPRES
0086 180 FORMAT(//,10X,"FREQUENCY=F",I1,"+",I3,10X,"PRESSURE= ",I5,
0087 1" PSI")
0088 200 WRITE(1,201)
0089 201 FORMAT(2X,"*****")
0090 1"*****"
0091 24X,"AMPLITUDE DIFFERENCE",2X,"PHASE DIFFERENCE",5X,
0092 3"COMMENTS",/)
0093 18 CALL KYBD(044061B,0)
0094 CALL KYBD(044061B,1)
0095 CALL KYBD(043040B,0,1)
0096 CALL KYBD(035040B,1)
0097 CALL KYBD(052120B)
0098 CALL GET(0,IBLK,DATAR,DATAI)
0099 CALL KYBD(052114B)
0100 DATA=20.*(ALOGT(DATAR))
0101 29 JKL=JK-INUM
0102 32 AJKL=JKL

```

```

0103      FREQ2=IFREQ2
0104      SHFT=360.0XFREQ2*AJKL*DELT
0105      PHAS=DATAI-SHFT
0106      PHAS1=ABS(PHAS)
0107      IF (PHAS1.GT.10.)GO TO 60
0108      DATA1=ABS(DATA)
0109      IF (DATA1.GT.1.0)GO TO 60
0110      IF (JK.EQ.INUM)GO TO 48
0111      GO TO 50
0112      48 WRITE(1,49)JK,DATA,PHAS
0113      49 FORMAT(6X,I2,13X,F5.2," DB",11X,F6.2," DEGREES",5X,
0114      1** REF **")
0115      GO TO 75
0116      50 WRITE(1,51)JK,DATA,PHAS
0117      51 FORMAT(6X,I2,13X,F5.2," DB",11X,F6.2," DEGREES")
0118      GO TO 75
0119      60 WRITE(1,61)JK,DATA,PHAS,IBEL
0120      61 FORMAT (6X,I2,13X,F5.2," DB",11X,F6.2," DEGREES",5X,
0121      1*X CHECK *,A2)
0122      75 CONTINUE
0123      CALL TPLOT(0,25,125)
0124      CALL TPLOT(1,25,610)
0125      CALL TPLOT(1,1000,610)
0126      CALL TPLOT(1,1000,125)
0127      CALL TPLOT(1,25,125)
0128      CALL TPLOT(0,200,125)
0129      CALL TPLOT(1,200,610)
0130      CALL TPLOT(0,510,610)
0131      CALL TPLOT(1,510,125)
0132      CALL TPLOT(0,800,125)
0133      CALL TPLOT(1,800,610)
0134      CALL TPLOT(0,25,548)
0135      CALL TPLOT(1,1000,548)
0136      99 WRITE(1,100)IBEL
0137      100 FORMAT(A2)

```

0138
0139

END
END\$

:

HEWLETT-PACKARD FOURIER ANALYZER USER'S GROUP

CONTRIBUTED PROGRAM DISCLAIMER

☒ INITIAL SUBMISSION

☐ REVISION (PROGRAM NO.)

CLASSIFICATION CODE (a list of classification codes is included on next page)

y

02	2	1

↑ Classification

"KROSS"

TITLE DATA BLOCK CURSOR ROUTINE

CONTRIBUTOR'S NAME DEAN JOHNSON

ORGANIZATION HEWLETT-PACKARD
SANTA CLARA DIVISION

ADDRESS 5301 STEVENS CREEK BLVD.
SANTA CLARA, CALIF. 95050

TELEPHONE (408) 246-4300 EXT-2566

Name and Organization to appear in catalog? ☒ Yes ☐ No

May a user contact you directly? ☒ Yes ☐ No

MATERIAL SUBMITTED

☒ Documentation ☒ Source Program ☐ Test Case

☒ Other LIBRARY

ACKNOWLEDGEMENT AND AGREEMENT:

To the best of my knowledge, this contributed program is free of any proprietary information belonging to any person or organization. I am making this program information available to the HEWLETT-PACKARD Fourier Analyzer User's Group. I hereby agree that Hewlett-Packard may reproduce, publish, and use it, and authorize others to do so without obligations or liability of any kind.

(Signature)

(Date)

HEWLETT-PACKARD FOURIER ANALYZER USER'S GROUP

CONTRIBUTED PROGRAM DOCUMENTATION FORM

1.0 IDENTIFICATION

10637-80221

1.1 Program Title

DATA BLOCK CURSOR ROUTINE

1.2 User Program Command Number

Y0221

1.3 Program Language

☐ FORTRAN II

☒ Assembly Language

☐ FORTRAN IV

2.0 PROGRAM DESCRIPTION

When called, Y0221 places a cursor across any frequency domain data block displayed (~~time domain data is ignored~~). The cursor can be moved left or right, and information - channel #, block #, frequency, and amplitude - can be printed out by pressing display register buttons. The program works on linear or log, rectangular or polar, and on partial displays. If Y0221 is called a second time, the program ends.

The display register controls are:

- BIT 7 = FAST RIGHT MOVEMENT
- " 8 = STEP RIGHT
- " 9 = PRINT INFO
- " 10 = HIGH SPEED DISPLAY W/OUT CURSOR
- " 11 = ZOOM
- " 12 = STEP LEFT
- " 13 = FAST LEFT MOVEMENT

Time Data

CHANNEL NUMBER OF BLOCK nnnn

TIME = nnnn * 10** ± nn SEC

AMPLITUDE = nnnn * 10** ± nnn

Freq. Data

← Same

FREQUENCY = nnnn * 10** ± nn HZ

← Same

CONTRIBUTED PROGRAM DOCUMENTATION FORM

3.0 ENVIRONMENT DESCRIPTION

3.1 Supported Software Requirements

☒ 5451B Software Release ____

☐ 5451__/71__ Software Release ____

☐ 5451__/70__ Software Release ____

3.2 Non-Supported Software Requirements

Name and order number of other software packages called by the
contributed program:

NONE

3.3 Storage Requirements

FORTRAN, Assembly Language (to be completed by User's Group)

(2270)₁₀ without external subprograms

(. 442)₁₀ with external subprograms

(93)₁₀ Basepage

3.4 Hardware Requirements

5451B FOURIER SYSTEM

CONTRIBUTED PROGRAM DOCUMENTATION FORM

4.0 USER PROCEDURES

4.1 Tape Identification

Account for all physical tapes included in this package.

Tape Number	Contents Programs, tests, data, etc., in order from outside to inside of tape	Control Statement (if any)
1	Y0221 SOURCE TAPE KROSS TTY1 HASSL BCD2A B2BCD INT.D	ASMB,R,B,L, T,C
1	Y0221 LIBRARY TAPE KROSS TTY1 HASSL BCD2A B2BCD INT.D	

NOTE: No control statement should be on Source Tape if Assembly Language

CONTRIBUTED PROGRAM DOCUMENTATION FORM

4.0 USER PROCEDURES (Cont.)

4.2 Load and Run Instructions

- 1) FORM AN OVERLAY, ENTERING "Ø221" AS A NUMBER
- 2) FORCE LOAD "YØ221 LIBRARY"
- 3) LIBRARY LOAD "BCS RELOCATABLE LIBRARY"
- 4) END OVERLAY GENERATION
- 5) LOAD OVERLAY OVER SYSTEM AND RUN
- 6) CALL Y221 TO START

4.3 DIAGNOSTIC MESSAGES OR ADDITIONAL EXITS

TT WHAT? UNDEFINED DATA TYPE
 (BAD COORDINATE CODE)

FC WHAT? CHANNEL # TOO LARGE

CONTRIBUTED PROGRAM DOCUMENTATION

5.0 SUBPROGRAM INFORMATION

5.1 Entry Points

Y0221

5.2 Externals Called

☒ User Program Utility Library ☐ FORTRAN II Library

☐ Fourier Routines

☐ FORTRAN IV Library

☒ Others

FLOATING POINT BCS RELOCATABLE
LIBRARY (24250-60001)

CONTRIBUTED PROGRAM DOCUMENTATION FORM

6.0 SPECIAL CONSIDERATIONS

NONE

7.0 EXAMPLE INPUT/OUTPUT (Test Case)

NONE

8.0 LITERATURE REFERENCE

NONE

HEWLETT-PACKARD FOURIER ANALYZER USER'S GROUP

CONTRIBUTED PROGRAM DISCLAIMER

☒ INITIAL SUBMISSION

☐ REVISION (PROGRAM NO.)

CLASSIFICATION CODE (a list of classification codes is included on next page)

Y 02 2 1

↑ Classification

"KROSS"

TITLE DATA BLOCK CURSOR ROUTINE

CONTRIBUTOR'S NAME DEAN JOHNSON

ORGANIZATION HEWLETT-PACKARD
SANTA CLARA DIVISION

ADDRESS 5301 STEVENS CREEK BLVD.
SANTA CLARA, CALIF. 95050

TELEPHONE (408) 246-4300 EXT-2566

Name and Organization to appear in catalog? ☒ Yes ☐ No

May a user contact you directly? ☒ Yes ☐ No

MATERIAL SUBMITTED

☒ Documentation ☒ Source Program ☐ Test Case

☒ Other LIBRARY

ACKNOWLEDGEMENT AND AGREEMENT:

To the best of my knowledge, this contributed program is free of any proprietary information belonging to any person or organization. I am making this program information available to the HEWLETT-PACKARD Fourier Analyzer User's Group. I hereby agree that Hewlett-Packard may reproduce, publish, and use it, and authorize others to do so without obligations or liability of any kind.

(Signature)

(Date)

HEWLETT-PACKARD FOURIER ANALYZER USER'S GROUP

CONTRIBUTED PROGRAM DOCUMENTATION FORM

1.0 IDENTIFICATION

10637-80221

1.1 Program Title

DATA BLOCK CURSOR ROUTINE

1.2 User Program Command Number

Y0221

1.3 Program Language

☐ FORTRAN II

☒ Assembly Language

☐ FORTRAN IV

2.0 PROGRAM DESCRIPTION

When called, Y0221 places a cursor across any frequency domain data block displayed (~~time-domain data is ignored~~). The cursor can be moved left or right, and information - channel #, block #, frequency, and amplitude - can be printed out by pressing display register buttons. The program works on linear or log, rectangular or polar, and on partial displays. If Y0221 is called a second time, the program ends.

The display register controls are:

BIT 7 = FAST RIGHT MOVEMENT

" 8 = STEP RIGHT

" 9 = PRINT INFO

" 10 = HIGH SPEED DISPLAY W/OUT CURSOR

" 11 = ZOOM

" 12 = STEP LEFT

" 13 = FAST LEFT MOVEMENT

Time Data

CHANNEL # OF BLOCK nnnn

TIME = nnnn * 10** ± nn SEC

AMPLITUDE = nnnn * 10** ± nnn

Freq. Data

← Same

FREQUENCY = nnnn * 10** ± nn HZ

← Same

CONTRIBUTED PROGRAM DOCUMENTATION FORM

3.0 ENVIRONMENT DESCRIPTION

3.1 Supported Software Requirements

☒ 5451B Software Release ____

☐ 5451__/71__ Software Release ____

☐ 5451__/70__ Software Release ____

3.2 Non-Supported Software Requirements

Name and order number of other software packages called by the
contributed program:

NONE

3.3 Storage Requirements

FORTTRAN, Assembly Language (to be completed by User's Group)

(2270)₁₀ without external subprograms

(442)₁₀ with external subprograms

(93)₁₀ Basepage

3.4 Hardware Requirements

5451B FOURIER SYSTEM

CONTRIBUTED PROGRAM DOCUMENTATION FORM

4.0 USER PROCEDURES

4.1 Tape Identification

Account for all physical tapes included in this package.

Tape Number	Contents Programs, tests, data, etc., in order from outside to inside of tape	Control Statement (if any)
1	Y0221 SOURCE TAPE KROSS TTY1 HASSL BCD2A B2BCD INT.D	ASMB,R,B,L, T,C
1	Y0221 LIBRARY TAPE KROSS TTY1 HASSL BCD2A B2BCD INT.D	

NOTE: No control statement should be on Source Tape if Assembly Language

CONTRIBUTED PROGRAM DOCUMENTATION FORM

4.0 USER PROCEDURES (Cont.)

4.2 Load and Run Instructions

- 1) FORM AN OVERLAY, ENTERING "Ø221" AS A NUMBER
- 2) FORCE LOAD "YØ221 LIBRARY"
- 3) LIBRARY LOAD "BCS RELOCATABLE LIBRARY"
- 4) END OVERLAY GENERATION
- 5) LOAD OVERLAY OVER SYSTEM AND RUN
- 6) CALL Y221 TO START

4.3 DIAGNOSTIC MESSAGES OR ADDITIONAL EXITS

TT WHAT? UNDEFINED DATA TYPE
 (BAD COORDINATE CODE)

FC WHAT? CHANNEL # TOO LARGE

CONTRIBUTED PROGRAM DOCUMENTATION

5.0 SUBPROGRAM INFORMATION

5.1 Entry Points

YØ221

5.2 Externals Called

☒ User Program Utility Library ☐ FORTRAN II Library

☐ Fourier Routines

☐ FORTRAN IV Library

☒ Others

FLOATING POINT BCS RELOCATABLE
LIBRARY (24250-60001)

SOUND Velocity Calculation Program. SOUND converts temperature versus depth profile to a sound velocity versus depth profile. A standard salinity of 34PPM is assumed at all depths. The following is the operating instructions for the SOUND program:

STEP NO.	DESCRIPTION	OBSERVATION
1	Bootstrap load the RTE Software System and start the File Manager	RTE software system is loaded and file management system is operating.
2	Enter and execute the following FMGR operator commands :LG,1+ :MR,SOUND+ :RU,LOADR,99+ :RU,SOUND+	Allocate number tracks to load and go track area. Transfer relocatable program file to LG tracks. Run loader program to load program from LG area. Run SOUND program
3	Enter a string of depths in feet and temperatures in degrees F separated by a comma.	Each entry is displayed.
4	Enter a temperature of 0 to terminate the string.	The SOUND velocity table is displayed.
5	Terminate operation by entering a negative number for depth.	Operation returns to FMGR.

CUPROF T-00004 IS ON CR00001 USING 00010 BLKS R=0056

```
0001 FTN4,L
0002 PROGRAM SOUND
0003 C
0004 C
0005 C*****
0006 C*
0007 C* TITLE: SOUND VELOCITY CALCULATION
0008 C*
0009 C* MODULE: SOUND SOURCE: SUPROF VERSION: X00
0010 C*
0011 C* AUTHOR: J. P. BRETT DATE: 12 OCT 1977
0012 C*
0013 C* MODULE FUNCTION:
0014 C*
0015 C* THIS PROGRAM CONVERTS TEMPERATURE VS DEPTH PROFILE
0016 C* TO A SOUND VELOCITY VS DEPTH PROFILE. A STANDARD
0017 C* SALINITY OF 34PPM IS ASSUMED AT ALL DEPTHS.
0018 C*
0019 C*
0020 C* INPUT PARAMETERS: DEPTH = DEPTH
0021 C* TEMP = TEMPERATURE
0022 C*
0023 C* NOTE: A MAXIMUM OF 100 INPUT ENTRIES ARE PERMITTED
0024 C*
0025 C* OUTPUT PARAMETERS: SU = SOUND VELOCITY
0026 C*
0027 C*****
0028 C*
0029 C*
0030 C*
0031 DIMENSION TEMP(100),DEPTH(100),SU(100)
0032 WRITE(1,33)
```

```

0033 33      FORMAT("UPROF - VELOCITY PROFILE PROGRAM",/, "THIS PROGRAM"
0034      * " CONVERTS THE TEMPERATURE VS DEPTH",/, "PROFILE INTO THE"
0035      * " EQUIVALENT SOUND VELOCITY VS DEPTH",/, "PROFILE ON A"
0036      * " POINT-BY-POINT BASIS INTERACTIVELY.",/, "TO TERMINATE LOOP"
0037      * ", ENTER A NEGATIVE NUMBER FOR DEPTH.",/)
0038 9        WRITE(1,10)
0039 10       FORMAT(// "ENTER A STRING OF DEPTHS IN FEET AND"/
0040      1 "TEMPERATURES IN DEGREES F SEPARATED BY A COMMA."/
0041      2 "ENTER A TEMPERATURE OF 0 TO TERMINATE THE STRING.")
0042      I=0
0043      WRITE(1,35)
0044      FORMAT(// "DEPTH, TEMP")
0045      I=I+1
0046      READ(1,*) DEPTH(I), TEMP(I)
0047      FORMAT(2F10.2)
0048      IF (DEPTH(1).LT.0.) GOTO 14
0049      IF (TEMP(I).NEQ.0.) GO TO 50
0050      I=I-1
0051      DO 70 N=1, I
0052      TEMP(N)=(TEMP(N)-32.)*5.0/9.0
0053      DEPTH(N)=DEPTH(N)*0.3048
0054      SAL = 34.0
0055      PRES = 1.04 + 0.102506 *(1 + .00528 * SIN(.5846853)**2)*
0056      1DEPTH(N)+2.524E-7*DEPTH(N)**2
0057      VT=4.5721*TEMP(N)-4.4532E-2*TEMP(N)**2-2.6045E-4*TEMP(N)**
0058      $3+7.9851E-6*TEMP(N)**4
0059      UP = 1.60272E-1*PRES + 1.0268E-5 * PRES**2 + 3.5216E-9 *
0060      $PRES**3 - 3.3603E-12 * PRES**4
0061      US = 1.39799 *(SAL - 35) + 1.69202E-3*(SAL-35)**2
0062      USTP=(SAL-35)*(-1.1244E-2*TEMP(N)+7.7711E-7*TEMP(N)**2+
0063      $7.7016E-5*PRES-1.2943E-7*PRES**2+3.158E-8*PRES*TEMP(N)+
0064      $1.579E-9*PRES*TEMP(N)**2)+PRES*(-1.8607E-4*TEMP(N)+
0065      $7.4812E-6*TEMP(N)**2+4.5283E-8*TEMP(N)**3)+PRES**2*
0066      $(-2.5294E-7*TEMP(N)+1.8563E-9*TEMP(N)**2)+PRES**3*
0067      $(-1.9646E-10*TEMP(N))

```


0068		VF = 1449.14 + VT + VP + US + USTP
0069	70	SU(N)=VF*3.280833
0070		DO 80 N=1,I
0071		DEPTH(N)=DEPTH(N)*3.280833
0072		TEMP(N)=TEMP(N)*9.0/5.0+32.0
0073	80	CONTINUE
0074		WRITE(1,13)
0075	13	FORMAT(20X,"TEMPERATURE DEPTH SOUND VELOCITY",/)
0076		WRITE(1,12)(TEMP(N),DEPTH(N),SU(N),N=1,I)
0077	12	FORMAT(23X,F4.1,7X,F7.1,7X,F6.1)
0078		GO TO 9
0079	14	END
0080		END\$

:

5.5 OLAAS SYSTEM/CORELOAD TRANSFER FILES

A powerful feature of the RTE System is its ability to run from "Transfer Files". A transfer file is an ASCII file containing a sequence of File Manager directives. Using transfer files, the basic system and the various coreloads are generated. The following are the Transfer Files used in the OLAAS System:

- MPTR - Basic System File
- MPTRØ - Coreload Ø File
- RTTR - Basic System for Coreloads Ø & 2 File
- MPTR1 - Coreload 1 File
- MPTR2 - Coreload 2 File
- MPTR5 - Coreload 5 File
- MPTR6 - Coreload 6 File
- MPTR7 - Coreload 7 File

MPTR T-00004 IS ON CR00001 USING 00002 BLKS R-0013

0001 :RU,CSORT,BR,1C,SC,,1
0002 :PU,PCSJOB
0003 :PU,MPSYS
0004 :PU,MPSNST
0005 :PU,SOGJOB
0006 :PK,-2
0007 :DU,RAFPC,PCSINP
0008 :DU,CSORTC,CMMAND
0009 :RP,FXL2
0010 :RU,FXL
0011 :RP,,FXL2
0012 :TR

MPTR TRANSFER FILE CREATES MPSYS, THE BASIC BCS AND
FOURIER SYSTEMS (32K, MASS STORE), USING COMMAND FILE,
BR1CSC, AND PCS INPUT FILE, RAFFC.
COMMAND FILE, BR1CSC, IS AN EXPANDED VERSION OF RAFCM.

0013
0014
0015
0016

:

MPTR0 T=00004 IS ON CR00001 USING 00001 BLKS R=0008

0001 :PU,#DCP
0002 :ST,#Y0003,#DCP
0003 :DU,#CDUMP,#DCP,,2,1
0004 :DU,#CLOAD,#DCP,,2,1
0005 :DU,#TYPI1,#DCP,,2,1
0006 :TR,RTTR
0007 :TR

MPTR0 TRANSFER FILE CREATES #DCP, WHICH CONTAINS THE
RELOCATABLE INPUTS FOR GENERATION OF CORELOAD 0, THE OLAAS
EXECUTIVE PROGRAM.

0008
0009
0010

:

RTTR T=00004 1. ON CP00001 USING 00002 BLKS R-0011

0001 :PU,PCSJOB
0002 :PU,DOPSYS
0003 :PK,-2
0004 :DU,RTPC,PCSINP
0005 :DU,RTCM,CMMAND
0006 :RP,FXL2
0007 :RU,FXL
0008 :RP,,FXL2
0009 :TR

RTTR TRANSFER FILE CREATES DOPSYS, THE BASIC BCS SYSTEM
USED IN THE GENERATION OF CORELOAD 0 & 2, USING #DCP AND
THE PCS INPUT FILE, RTPC.

:

MPTR1 T=00004 IS ON CR00001 USING 00001 BLKS R=0004

0001 :PU,RAF
0002 :PK,-2
0003 :DU,MPCM1,CMMAND
0004 :RU,FXL3
0005 :TR

MPTR1 TRANSFER FILE GENERATES CORELOAD 1, ANALYSIS &
SYSTEM EXECUTIVE SUBROUTINE, USING COMMAND FILE, MPCM1.

:

MPTR2 T=00004 IS ON CR00001 USING 00002 BLKS R-0012

0001	:	PU,	#DCP
0002	:	ST,	#Y0001,#DCP
0003	:	DU,	#LKTBG,#DCP,,2,1
0004	:	DU,	#LINKS,#DCP,,2,1
0005	:	DU,	#MPINP,#DCP,,2,1
0006	:	DU,	#CDUMP,#DCP,,2,1
0007	:	DU,	#CLOAD,#DCP,,2,1
0008	:	DU,	INPTO,#DCP,,2,1
0009	:	DU,	#TYPI1,#DCP,,2,1
0010	:	DU,	#TAT1,#DCP,,2,1
0011	:	TR,	RTTR
0012	:	TR	

MPTR2 TRANSFER FILE CREATES #DCP, WHICH CONTAINS THE
RELOCATABLE INPUTS FOR GENERATION OF CORELOAD 2, THE
OLAAS RAW DATA COLLECTION SUBROUTINE.

:

MPTRS T=00004 IS ON CR00001 USING 00001 BLKS R-0004

0001
0002
0003
0004
0005
0006
0007

:PU,RAF
:PK,-2
:DU,MPCMS,CMMAND
:RU,FXL3
:TR

MPTRS TRANSFER FILE GENERATES CORELOAD 5, OLAAS USER
PROGRAM, USING THE COMMAND FILE, MPCMS.

:

MPTR6 T=00004 IS ON CR00001 USING 00001 BLKS R=0005

0001 :PU,RAF
0002 :PK,-2
0003 :DU,MPCM6,CMMAND
0004 :RU,FXL3
0005 :TR

MPTR6 TRANSFER FILE GENERATES CORELOAD 6, FOURIER
ANALYZER GRAPHICS, USING COMMAND FILE, MPCM6.

0006
0007

:

MPTR7 T=00004 IS ON CR00001 USING 00001 BLKS R=0005

0001 :PU,RAF
0002 :PK,-2
0003 :DU,MPCM7,CMMAND
0004 :RU,FXL3
0005 :TR

MPTR7 TRANSFER FILE GENERATES CORELOAD 7, OLAS PLOT
DISPLAY SUBROUTINE, USING COMMAND FILE, MPCM7.

:

5.6 OLAAS SYSTEM/CORELOAD COMMAND FILES

A Command File is an ASCII file containing a sequence of FXL commands. FXL, the Fourier Cross Loader, is a package which operates in the RTE Software System to generate automatically the Fourier System, SOG, BCS and Overlays, as well as the listings that accompany them. The information generated is stored in RTE files for later output. Included with FXL is the program COPYF (Copy Files), which can be used to store RTE-generated systems and overlays directly into Fourier File 6 on the Fourier disc. This transfers software from RTE files to the Fourier Analyzer disc. The following are the Command Files used in the OLAAS System:

BR1CSC	- Basic System File
MPCM1	- Coreload 1 File
MPCM5	- Coreload 5 File
MPCM6	- Coreload 6 File
MPCM7	- Coreload 7 File

ERICSC T=00003 IS ON CR00001 USING 00014 BLKS R=0000

0001 **
0002 .

BRICSC COMMAND FILE IS FOR THE BUNKER RAMO 5451B
WHICH CONTAINS PROVISION FOR THE FOLLOWING:
VARIABLE PARAMETERS

7900A (I0-17)

THE DUMMY DRIVERS IN THE SYSTEM ARE:

D.06, D.10, D.53, D.56

THE DESIGN OF THIS SYSTEM HAS BEEN DONE WITH FLEXIBILITY OF
USE IN MIND. USING THIS SYSTEM AS A BASE, MOST OF THE
STANDARD OPTINS CAN BE IMPLEMENTED BY MAKING APPROPRIATE
OVERLAYS.

THIS COMMAND FILE GOES WITH THE PCS ANSWER FILE, RAFFC.
GENERATED ARE PCSJOB (BCS SYSTEM TAPE) AND MPSYS (THE BASIC
FOURIER CORELOAD). THE SNAPSHOT FILE IS MPSNST.

THE TRANSFER FILE IS MPTR.

U. BOWEN 11/3/77

PCS OPERATIONS

A,PCSJOB

DBOOT

F,A4501D

Y,xx

IOC

F,B3000C

Y,IOC

REST OF PCS LIB

F,B3000C

Y,D.00

Y,LOADR

0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032

0033	END OF PCS	
0034	##	
0035	A,MPSYS	
0036	F,#USTRT	
0037	B,**	
0038	*****	
0039	BCS OPERATIONS	
0040		CSORT+
0041	FLIB1 LIBRARY TO BASEPAGE	
0042	F,B3052B	
0043	B,INS	
0044	F,B3052B	
0045	Y,CONST	
0046	FLIB2 LIBRARY	
0047	F,B3053D	
0048	Y,RUN5	
0049	F,B3053D	
0050	Y,WCS36	
0051		CSORT-
0052		CSORT+
0053	F,B3053D	
0054	Y,CNTRL	
0055	F,B3053D	
0056	Y,XEQ	
0057	F,B3053D	
0058	Y,WHAT	
0059	F,B3053D	
0060	Y,INTCH	
0061	F,B3053D	
0062	Y,ADCPG	
0063	F,B3053D	
0064	Y,CIO	
0065	F,B3053D	
0066	Y,CONV	
0067	F,B3053D	

0069	Y, EXEC
0069	F, B3053D
0070	Y, CHKN
0071	F, B3053D
0072	Y, WRBUF
0073	F, B3053D
0074	Y, CPLXM
0075	F, B3053D
0076	Y, LOGA
0077	F, B3053D
0078	Y, PROPH
0079	F, B3053D
0080	Y, SKIP
0081	F, B3053D
0082	Y, YCOM
0083	F, B3053D
0084	Y, YXCOM
0085	F, B3053D
0086	Y, EDEL
0087	F, B3053D
0088	Y, BUF0
0089	F, B3053D
0090	Y, JUMPY
0091	F, B3053D
0092	Y, HIST
0093	F, B3053D
0094	Y, CLEAR
0095	F, B3053D
0096	Y, .MPY.
0097	F, B3053D
0098	Y, .DIU.
0099	F, B3053D
0100	Y, DPSHR
0101	F, B3053D
0102	Y, INTEG

0103	F, B3053D
0104	Y, DTAD0
0105	F, B3053D
0106	Y, SPEC
0107	F, B3053D
0108	Y, FOUR.
0109	F, B3053D
0110	Y, RANGE
0111	F, B3053D
0112	Y, MULT.
0113	F, B3053D
0114	Y, ADD.
0115	F, B3053D
0116	Y, BDIV
0117	FLIB3 LIBRARY (NOT LATEST VERSION, WHICH IS D)
0118	F, B3054C
0119	Y, DIS45
0120	F, B3054C
0121	Y, BLANK
0122	F, B3054C
0123	Y, UCS16
0124	F, B3054C
0125	Y, KEYPH
0126	F, B3054C
0127	Y, PRINT
0128	F, B3054C
0129	Y, RDBUF
0130	F, B3054C
0131	Y, KREAD
0132	F, B3054C
0133	Y, ROTAT
0134	F, B3054C
0135	Y, DISP.
0136	F, B3054C
0137	Y, SURIN

0138	F,B3054C
0139	Y,BUF IN
0140	F,B3054C
0141	Y,LOCAT
0142	F,B3054C
0143	Y,XSTOR
0144	F,B3054C
0145	Y,BLOK
0146	F,B3054C
0147	Y,HANN
0148	F,B3054C
0149	Y,MAXC
0150	F,B3054C
0151	Y,MAXIM
0152	F,B3054C
0153	Y,COOR
0154	F,B3054C
0155	Y,BDIF
0156	F,B3054C
0157	Y,LECTR
0158	F,B3054C
0159	Y,DPREC
0160	F,B3054C
0161	Y,IF.
0162	F,B3054C
0163	Y,ADSCl
0164	F,B3054C
0165	Y,A%COM
0166	F,B3054C
0167	Y,WCS20
0168	F,B3054C
0169	Y,WCS24
0170	FLIB4 LIBRARY
0171	F,B3055C
0172	Y,WCS22

DISC LIBRARY (NOT LATEST VERSION, WHICH IS G)

0173 F,B3049F
 0174 Y,D.34A
 0175 F,B3049F
 0176 Y,MASS
 0177 F,B3049F
 0178 Y,.ENTR
 0179 F,B3049F
 0180 Y,.EAU.
 0181 FMTXX (DISC FORMAT)
 0182 F,\$FMTX1
 0183 L,FMTXX
 0184 ? LIBRARY
 0185 F,A3060A
 0186 L,D.06
 0187 F,A3060A
 0188 L,D.10
 0189 F,A3060A
 0190 L,D.53
 0191 F,A3060A
 0192 L,D.56
 0193 F,A3060A
 0194 L,I.10
 0195
 0196

CSORT-

REMAT

M,72000
 F,B3025C
 Y,xx
 R,xx

LIST UNDEFINED SYMBOLS

END OF SYSTEM

SAVE SNAPSHOT

C,MPSNST

0173
 0174
 0175
 0176
 0177
 0178
 0179
 0180
 0181
 0182
 0183
 0184
 0185
 0186
 0187
 0188
 0189
 0190
 0191
 0192
 0193
 0194
 0195
 0196
 0197
 0198
 0199
 0200
 0201
 0202
 0203
 0204
 0205
 0206
 0207

0208 SOG
0209 A, SOGJOB
0210 M, 2010
0211 F, B0009C
0212 Y, **
0213 \$\$
0214 ::
0215 END
:

MPCMI T=00004 I 4 CR00001 USING 00004 BLKS R=0040

0001 300
0002 H,RAF
0003 MPCMI COMMAND FILE IS FOR GENERATING OVERLAYS FOR THE
0004 BASIC FOURIER SYSTEM (MPSYS), USING THE SNAPSHOT FILE
0005 "MPSNST", FOR CORELOAD 1. THE TRANSFER FILE IS MPTR1.
0006
0007
0008
0009

OPEN OVERLAY AND SNAPSHOT FILES

T,MPSNST
ENTER USER PROGRAM NUMBERS

0,1002
END OF USER PROGRAM LIST

0E
OPEN USER PROGRAMS

F,Y10020

L,**

F,#RDATA

L,**

F,#TAT1

L,**

F,INPT0

L,**

F,#ASKQ

L,**

F,#TYP11

L,**

F,#CDUMP

L,**

F,#CLOAD

L,**

F,ISHFT0

0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032

0033	L,XX	
0034	F,D7500	
0035	L,XX	
0036	F,#DISC	
0037	L,XX	
0038	F,#LINKD	
0039	L,XX	
0040		END OF RELOCATIBLE USER PROGRAM ENTRY
0041		LOAD LIBRARIES
0042		
0043		TEKPLLOT ROUTINES
0044	F,TEKOC	
0045	L,XX	
0046	F,ERASO	
0047	L,XX	
0048		USER PROGRAM LIBRARY
0049	F,B3018C	
0050	L,XX	
0051		BCS FORTRAN IV LIBRARY
0052	F,F4145B	
0053	L,XX	
0054		BCS FLOATING POINT LIBRARY
0055	F,F4250B	
0056	L,XX	
0057		LIST UNDEFINED SYMBOLS
0058	U,XX	
0059		END
0060	E,01	
0061	::	
0062	\$\$	

:

MPCMS T=00004 IS 00000001 USING 00006 BLKS R=0092

0001
0002 H.BAF
0003
0004 MPCM5 COMMAND FILE IS FOR GENERATING OVERLAYS FOR THE
0005 BASIC FOURIER SYSTEM (MPSYS), USING THE SNAPSHOT FILE
0006 "MPSNST", FOR CORELOAD 5. THE TRANSFER FILE IS MPTRS.
0007
0008

OPEN OVERLAY AND SNAPSHOT FILES

T,MPSNST
ENTER USER PROGRAM NUMBERS

0012 e,7891
0013 e,7894
0014 e,7896
0015 e,7897
0016 e,7898
0017 e,7899
0018 e,7901
0019 e,7902
0020 e,7904
0021 e,7906
0022 e,7907

CURSOR PROGRAM

e,0221
END OF USER PROGRAM LIST

eE

OPEN USER PROGRAMS

0027
0028 F,Y78910
0029 L,**
0030 F,Y78940
0031 L,**
0032 F,Y78960

0033	L,XX	
0034	F,Y78970	
0035	L,XX	
0036	F,Y78980	
0037	L,XX	
0038	F,Y78990	
0039	L,XX	
0040	F,Y79010	
0041	L,XX	
0042	F,Y79020	
0043	L,XX	
0044	F,Y79040	
0045	L,XX	
0046	F,Y79060	
0047	L,XX	
0048	F,Y79070	
0049	L,XX	
0050	F,Y02210	
0051	L,XX	
0052		OPEN SUBROUTINES CALLED BY USER PROGRAMS
0053	F,BMDATO	
0054	L,XX	
0055	F,DMA10	
0056	L,XX	
0057	F,\$RDATA	
0058	L,XX	
0059	F,INPTB	
0060	L,XX	
0061	F,\$TYPI	
0062	L,XX	
0063	F,\$TAT1	
0064	L,XX	
0065	F,D7500	
0066	L,XX	
0067	F,\$RSTAU	

```

0068 Y,xx
0069     END OF RELOCATIBLE USER PROGRAM ENTRY
0070
0071     LOAD LIBRARIES
0072
0073     SCREEN ERASE - MODIFIED TEKTRONIX ROUTINE
0074 F,TEKOC
0075 L,xx
0076 F,ERASO
0077 L,xx
0078     USER PROGRAM LIBRARY
0079 F,B3018C
0080 L,xx
0081     BCS FORTRAN IV LIBRARY
0082 F,F4145B
0083 L,xx
0084     BCS FLOATING POINT LIBRARY
0085 F,F4250B
0086 L,xx
0087     LIST UNDEFINED SYMBOLS
0088 U,xx
0089     END
0090 E,01
0091 ::
0092 $$

```

```

:
```

MPCMG T=00004 IS ON CP00001 USING 00004 BLKS R=0046

```
0001 00
0002 A,RAF
0003 MPCMG COMMAND FILE IS FOR GENERATING OVERLAYS FOR THE
0004 BASIC FOURIER SYSTEM (MPSYS), USING THE SNAPSHOT FILE
0005 "MPSNST", FOR CORELOAD 6. THE TRANSFER FILE IS MPTR6.
0006
0007
0008 OPEN OVERLAY AND SNAPSHOT FILES
0009 T,MPSNST
0010 ENTER USER PROGRAM NUMBERS
0011 0,5804
0012 0,5805
0013 0,5809
0014 0,5810
0015 0,5814
0016 0,5815
0017 0,5816
0018 0,5817
0019 0,5821
0020 END OF USER PROGRAM LIST
0021 0E
0022
0023 LOAD LIBRARIES
0024
0025 GRAPHICS LIBRARY
0026 F,A3015I
0027 L,**
0028 USER PROGRAM LIBRARY
0029 F,B3018C
0030 L,**
0031 BCS FLOATING POINT LIBRARY
0032 F,F4250B
```


0033 L,**
0034 GRAPHICS DRIVER D.06
0035 F,A4503A
0036 Y,**
0037 LIST UNDEFINED SYMBOLS
0038 U,**
0039 END
0040 E,01
0041 ::
0042 \$\$

:

```

MPCMT  T=00004 IS ON CR00001 USING 00004 BLKS R=0054
0001  @
0002  A,RAF
0003
0004      MPCMT7 COMMAND FILE IS FOR GENERATING OVERLAYS FOR THE
0005      BASIC FOURIER SYSTEM (BASYS), USING THE SNAPSHOT FILE
0006      "MPSNST", FOR CORELOAD 7.  THE TRANSFER FILE IS MPTR7.
0007
0008
0009      OPEN OVERLAY AND SNAPSHOT FILES
0010      T,MPSNST
0011      ENTER USER PROGRAM NUMBERS
0012      @,1007
0013      @,7909
0014      END OF USER PROGRAM LIST
0015      @E
0016      OPEN USER PROGRAMS
0017      F,Y10070
0018      L,XX
0019      F,Y79090
0020      L,XX
0021      F,$RDATA
0022      L,XX
0023      F,$TAT1
0024      L,XX
0025      F,INPT0
0026      L,XX
0027      F,$TYPI1
0028      L,XX
0029      F,$CDUMP
0030      L,XX
0031      F,$CLOAD
0032      L,XX

```

0033	F, #DISC	
0034	L, **	
0035	F, #LINKD	
0036	L, **	
0037	F, D7500	
0038	L, **	
0039		END OF RELOCATIBLE USER PROGRAM ENTRY
0040		LOAD LIBRARIES
0041		
0042		TEKPLLOT ROUTINES
0043	F, TEKOC	
0044	L, **	
0045	F, ERASO	
0046	L, **	
0047		USER PROGRAM LIBRARY
0048	F, B3018C	
0049	L, **	
0050		BCS FORTRAN IV LIBRARY
0051	F, F4145B	
0052	L, **	
0053		BCS FLOATING POINT LIBRARY
0054	F, F4250B	
0055	L, **	
0056		LIST UNDEFINED SYMBOLS
0057	U, **	
0058		END
0059	E, 01	
0060	:::	
0061	\$\$	

:

5.7 OLAAS SYSTEM/CORELOAD PCS INPUT FILES

The information required by PCS concerning the system I/O configuration is passed on to FXL in an RTE PCS INPUT FILE. A PCS Input File is an ASCII file containing the sequence of all answers required by PCS. The following are the PCS Input files used in the OLAAS System:

RAFPC	-	Basic System File
RTPC	-	Coreload ϕ & 2 File

PHFPC T-00003 IS ON CR00001 USING 00003 BLKS R-0000

0001	30	
0002	77577	
0003	0,0	
0004	10,D.00	
0005	11,D.00	
0006	12,D.00	
0007	23,D.00	
0008	14,D.54	
0009	15,D.41	
0010	16,D.43	
0011	17,D.34	
0012	17,D.34,D,U16	
0013	17,D.34,D,U17	
0014	17,D.34,D,U20	
0015	17,D.34,D,U21	
0016	17,D.34,D,U22	
0017	17,D.34,D,U23	
0018	17,D.34,D,U24	
0019	17,D.34,D,U25	
0020	17,D.34,D,U26	
0021	14,D.55	
0022	13,D.53	
0023	21,D.56	
0024	11,D.06	
0025	24,D.10	
0026	/E	
0027	10	
0028	10	
0029	7	
0030	10	
0031	7	
0032	10	

0033	6.7
0034	4.PFSR
0035	6.106706
0036	7.106707
0037	10.1.00
0038	11.1.00
0039	12.1.00
0040	14.103114
0041	15.1.41
0042	16.1.43
0043	17.102017
0044	20.C.34
0045	21.000000
0046	22.1.00
0047	24.1.10
0048	/E

RAFFPC, THE PCS INPUT FILE USED TO CREATE THE BASIC
FOURIER SYSTEM, MPYS. RAFFPC IS USED WITH THE
TRANSFER FILE, MPTR, AND COMMAND FILE, BRICSC.

:

RTPC 00004 IS ON CR00001 USING 00002 BLKS R-0031

0001	30
0002	77577
0003	0.0
0004	10.D.00
0005	11.D.00
0006	12.D.00
0007	23.D.00
0008	/E
0009	10
0010	10
0011	7
0012	10
0013	7
0014	10
0015	6.7
0016	4,102004
0017	6,106706
0018	7,106707
0019	10,1.00
0020	11,1.00
0021	12,1.00
0022	14,103114
0023	15,102015
0024	16,102016
0025	17,102017
0026	20,102020
0027	21,000000
0028	22,102022
0029	23,102023
0030	24,102024
0031	/E
0032	

RTPC, PCS INPUT FILE, USED TO CREATE THE BASIC

0033
0034

FOURIER SYSTEM, DCPSYS. RTPC IS USED WITH THE
TRANSFER FILES, MPTR0, MPTR2, AND RTTR.

:

APPENDIX
A

OIAAS ACCEPTANCE TEST PLAN

OLAAS ACCEPTANCE TEST PLAN

1.0 SCOPE

This report details the proposed Acceptance Test Plan for the On-Line Acoustic Analysis System (OLAAS). The objective of the test is to verify proper operation of OLAAS Processing and Display System as set forth in the OLAAS specification.

2.0 OVERVIEW

Bunker Ramo has been engaged in the development of the OLAAS system, which is the computerized data analysis system for the ERAPS Sonobuoy Advanced Development Model. The system is housed in a portable van, and includes the interface to hardware developed under the ERAPS contract as well as the incorporation of hardware and interfaces procured under OLAAS. The development includes a specially written software package, Processing and Display System for the automatic detection analysis of ERAPS data.

3.0 TEST SPECIFICATIONS

3.1 Test Methods

The Acceptance Test will be made in the OLAAS Van, under the observation and with the interaction of NAVAIRDEVCON (NADC) personnel. Bunker Ramo personnel will operate the equipment and NADC personnel will be asked to submit arbitrary analysis parameters.

3.2 Test Procedures

The Acceptance Tests shall be made in accordance with the detailed procedures described in Section 4.0.

4.0 DETAILED TEST PROCEDURE

4.1 OLAAS Processing and Display System Software Tests

The OLAAS Processing and Display System software tests will be conducted as follows:

1. Bunker Ramo personnel will set-up the equipment for the tests.
2. NADC personnel will be asked to supply:
 - a. Date and time for CW and FM pings.
 - b. Changes to the analysis parameters for program verification.
3. System operation shall be according to the OLAAS Processing and Display System Operating Procedures.
4. Raw data (CW or FM) shall be input to the system from the tape recorder using tapes recorded during sea testing.
5. Hard copy prints shall be made of the detection plots, recording and analysis parameters.

4.2 Single Ping Recording and Analysis.

1. Using the OLAAS SYSTEM OPERATION PROCEDURE, 2.3, record and analyze a single CW ping using the parameters at time of recording. Verify:
 - a. Data is correct for the selected ping.
 - b. Date appears on the hard copy prints.
 - c. The CW Notch Filtering
2. Using the OLAAS SYSTEM OPERATION PROCEDURE, 2.3, re-analyze the data recorded in step 1 changing only the CW Notch Filter. Verify:
 - a. CW Notch Filtering has changed based on the value entered and as noted in step 1.

3. Using the OLAAS SYSTEM OPERATION PROCEDURE, 2.3, request a FOUR PART DISPLAY. Verify:
 - a. Data is correct according to step 1.
4. Using the OLAAS SYSTEM OPERATION PROCEDURE, 2.3, record and analyze a single FM ping using the parameters at time of recording. Verify:
 - a. Data is correct for the selected ping.
 - b. Date appears on the hard copy prints.
5. Repeat step 3 using an FM ping.
6. Using the OLAAS SYSTEM OPERATION PROCEDURE, 2.3, re-analyze the data recorded in step 4 changing selected analysis parameters. Verify:
 - a. Data is correct according to step 4.

4.3 Multi-Ping Recording and Analysis

1. Using the OLAAS SYSTEM OPERATION PROCEDURE, 2.3, record and analyze four (4) CW pings using the parameters at time of recording. Verify:
 - a. Data is correct for each of the selected pings.
2. Using the OLAAS SYSTEM OPERATION PROCEDURE, 2.3, request a MULTI-PING DISPLAY. Verify:
 - a. Data is correct according to step 1.
3. Using the OLAAS SYSTEM OPERATION PROCEDURE, 2.3, request a FOUR PART DISPLAY. Verify:
 - a. Data is correct according to step 1 and the ping selected.

4. Using the OLAAS SYSTEM OPERATION PROCEDURE, 2.3, re-analyze the data for a selected ping recorded in step 1 changing selected analysis parameters. Verify:
5. Repeat steps 1 through 4 using FM pings.

5.0 TEST SCHEDULE

The OLAAS Acceptance Test shall be made after 30 January 1978 at a time convenient to NADC. The test requires one (1) day to complete.

APPENDIX

B

FMIXX1 PROGRAM LISTING

FMTNXX1 T=00004 IS ON CR00001 USING 00052 BLKS R=0409

0001 ASMB

0002 HED A-05451-94194-1 REV.A NAM FMTXX ERS & LISTING
0003 NAM FMTXX

0004 *

0005 * PURPOSE:

0006 * THIS MODULE COMPRISES TABLES AND SUBROUTINES
0007 * WHICH DEFINE THE 7970 MAGTAPE/7900 DISC STRUCTURE
0008 * TO THE 5451 MASS-KEYSTROKE PROCESSOR.

0009 *

0010 * ENTRY POINTS:

0011 *

0012 * 1. FMTXX

0013 * THIS 5 WORD/ENTRY TABLE DEFINES THE NUMBER OF
0014 * RECORDS, AN EQUIPMENT TABLE ENTRY, AND A CORE
0015 * BUFFER FOR EACH OF THE 10 MASS-KEYSTROKE FILES.
0016 * THE EGT INFORMATION IS SETUP BY GLOBAL SUBROUTINE
0017 * MS00. THE RELATION BETWEEN THE LOGICAL RECORD
0018 * SIZE AND THE PHYSICAL RECORD SIZE IS ESTABLISHED
0019 * BY THE MASS SUBROUTINE.

0020 *

0021 * 2. BOOT7

0022 * THIS SUBROUTINE CAUSES THE HIGHCORE-RESIDENT MAG-
0023 * TAPE OR DISC BOOT TO LOCATE AND LOAD AND TRANSFER
0024 * CONTROL TO ANY CORE-IMAGE CONTAINED ON THE SYSTEM
0025 * FILE (IE FILE 6).

0026 *

0027 * 3. FMTDD

0028 * THIS 8 WORD/ENTRY TABLE DEFINES THE PHYSICAL PAR-
0029 * TITIONING OF THE 7900 DISC INTO FILES. THE TABLE
0030 * IS ONLY USED BY D.34, ITS FORMAT IS DEFINED IN
0031 * DETAIL THERE. NOTE THAT MAXIMUM TRANSFER SPEED
0032 * FOR THE 7900/7901 DISC NOT OBTAINED BY TRANSFERRING
0033 * DATA FOR A FULL DISC REVOLUTION THEN ACCESSING THE
0034 * NEXT TRACK, BUT RATHER BY TRANSFERRING FOR ABOUT
0035 * 2/3 OF A REVOLUTION THEN POSITIONING DURING THE
0036 * LAST 1/3.

0037 *


```

0066 *      BSS 1      BITS6-0-EQT UNIT REFERENCE NUMBER
0069 *      BSS 1      NON-INDIRECT DEF TO CORRESPONDING EQT ENTRY?
0070 *      DEC N      PHYSICAL RECORD LENGTH
0071 *      DEC M      NUMBER OF RECORDS
0072 *      DEF DEF    DEF TO DEF TO CORE BUFFER USED BY MASS FOR THIS FI
0073 *
0074 *      THE FILES MAY EXIST ON 7970 MAGTAPE (USING D.33A DRIVER) OR ON
0075 *      7900/7901 MOVING-HEAD DISC (USING D.34A DRIVER). ANY OTHER MED
0076 *      (EG 3030 MAGTAPE, CORE, CASSETTE, ETC) REQUIRES A SPECIAL DRIVE
0077 *      PATTERNED AFTER D.33/D.34, A FILE-POSITION TABLE PATTERNED AFTE
0078 *      FMT33/FMT34, CODING CHANGES TO MASS (RELATING TO 33/34, THE IND
0079 *      FILE, AND IF DESIRED, THE SYSTEM FILE), AND PERHAPS AN AUX TABL
0080 *      LIKE D.34'S DIFS TABLE OR D.33'S EQT CONVENTIONS (SEE BELOW).
0081 *      *****
0082 *

```

AUX D.34 INFO, DIFS TABLE DEFINES PHYSICAL DISC STRUCTURE

EXAMPLE:

```

0083 *      DEC 16      LOGICAL FILE NUMBER ARBITRARILY CHOSEN (MATCHES EQT
0084 *      DEC 50      NUMBER OF RECORDS IN FILE
0085 *      DEC 1       DISC SUB-UNIT NUMBER (1 FOR FIRST REMOVABLE DISC)
0086 *      DEC 30      STARTING CYLINDER
0087 *      DEC 0       STARTING SECTOR (48 SECTORS/CYL, 128 WORDS EACH)
0088 *      DEC 8       SECTORS/SECTION (SECTORS USED EACH REC ON ONE CYL)
0089 *      DEC 2       SECTIONS/RECORD (DIFFERENT CYLINDERS EACH RECORD)
0090 *      DEC 20      RECORDS/ROW (# OF RECORDS USING SAME SECTION)
0091 *
0092 *
0093 *
0094 *
0095 *      THE FIRST RECORD OF THIS FILE WOULD APPEAR IN SECTORS 0-7 OF
0096 *      CYLINDERS 30 & 31 FOR A MAX REC SIZE = 128X8X2 = 2048 WORDS.
0097 *      RECORDS 0-19 WOULD GO IN SECTORS 0-7 OF CYLS 30-69.
0098 *      RECORDS 20-39 WOULD GO IN SECTORS 8-15 OF CYLS 30-69.
0099 *      THE FINAL 10 RECORDS WOULD GO IN SECTORS 16-23 OF CYLS 30-49
0100 *
0101 *      *****
0102 *

```

ADDRESS	ENTRY	SKP	TABLE WORDS/ENTRY	TABLE DEF TO ENTRY	DUMMY (USED BY MASS,99 ONLY)
0103	ENTRY	DEC 5			
0104		DEF *+1			
0105	*				
0106		BSS 2			
0107		DEC 0,0			
0108		DEF 0			
0109	*				
0110		EXT DTAD0			
0111		BSS 2			
0112		DEC 2048,33			
0113		DEF DTAD0			
0114	*				
0115		EXT DTAD0			
0116		BSS 2			
0117		DEC 4096,0			
0118		DEF DTAD0			
0119	*				
0120		EXT STOR0			
0121		BSS 2			
0122		DEC 336,0			
0123		DEF STOR0			
0124	*				
0125		EXT BUF0			
0126		BSS 2			
0127		DEC 36,0			
0128		DEF BUF0			
0129	*				
0130		EXT .LBLK			
0131		BSS 2			
0132		DEC 21,0			
0133		DEF .LBLK			
0134	*				
0135		BSS 2			
0136	SYS	OCT 77700			
0137					

PROGRAM STACK

ASCII TEXT

INDEX

SYSTEM
CORESIZE - LENGTH OF NON-PROTECT

0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172

DEC 3
DEF 2

BSS 2
DEC 512,0
DEF 0

BSS 2
DEC 128,0
DEF 0

BSS 2
DEC 128,48
DEF 0

STA RECS
JSB MS00

CLC 0
STF 6
STF 7

LDA .BOOT
RSS
LDA A,I

COMMON

USER8

DATA BLOCK HEADER

* BOOT7, THIS PROGRAM IS USED TO LOAD A PROGRAM FROM THE SYSTEM
* AND TRANSFER CONTROL TO IT. DEPENDING ON THE DEVICE ON WHICH T
* THE SYSTEM FILES ARE STORED, BOOT7 IS REALLY S7900 OR S7970.

CORE ABS 16384 CORESIZE, SET BY BOOT7
SA OCT 100002 STARTING ADDRESS, SYS FILE CORELOADS
TABL7 DEF BOOT7 DEF TO BOOT'S TABLE
DACFL NOP

ENTRY: LDA,JSB. (NO RETURN)

MAKE SURE FMTXX IS SETUP

KILL DMA IN CASE ITS BUSY

CALC MAX BOOTABLE ADDRESS
(--DEF BOOT) AND STORE IT
IN XPROG FOR BOOT

0173	RAL,CLE,SLA,ERA
0174	JMP *-2
0175	CMA,INA
0176	STA XPROG
0177	CMA,INA
* 0178	
0179	IOR -B7700
0180	AND -B77700
0181	STA CORE
0182	LDB A
0183	LDA DUMPT
0184	DST DUMPT
* 0185	
0186	LDA CORE
0187	CLB
0188	DIV -D6144
0189	INA
0190	STA N7900
* 0191	
* 0192	
0193	LDB SYS+1
0194	INB
0195	LDA B,I
0196	ALF,ALF
0197	AND -B77
0198	CPA -B33
0199	JMP S7970
0200	CPA -B34
0201	JMP S7900
* 0202	
0203	S0000 HLT 7B
0204	JMP *-1
* 0205	
0206	S7900 BSS 0
0207	LDA RECS

```

02008 MPY N7900
02009 AND -B777
02010 ADA P7900
02011 JMP GOGO
02012 *
02013 S7970 BSS 0
02014 LDA RECS
02015 MPY N7970
02016 ALS,ALS
02017 ALS
02018 AND -B7770
02019 ADA N7970
02020 ADA -D-1
02021 ADA F7970
02022 SZA,RSS
02023 LDA SA
02024 GOGO LDB SA
02025 DST TABL7,I
02026 JMP SPROG
02027 *
02028 *CHARACTERISTICS OF 7970 SYSTEM FILE
02029 N7970 DEC 1 -CORESIZE/RECORDSIZE
02030 F7970 OCT 0 PHYSICAL FILENUMBER
02031 *CHARACTERISTICS OF 7900 OR 7901 SYSTEM FILE
02032 N7900 DEC 2 -CORESIZE/RECORDSIZE(SET BY BOOT?)
02033 P7900 OCT 1 PLATTER (10-9),CYLINDER (8-0)
02034 *
02035 RECS OCT 0 REC NUMBER IN SYS FILE
02036 .BOOT DEF BOOT
02037 *XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*
02038 * AUX D.34 INFO, DIFS TABLE DEFINES PHYSICAL DISC STRUCTURE,
02039 * THE BCS EGT TABLE 'SUBUNIT' FIELD IS A POINTER TO THE DIFS
02040 * 'LOGICAL FILE NUMBER'.
02041 *XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*
02042 *

```

```

      SETUP A=0,PLATTER,CYLINDER
      THEN LOAD AND GO.

```

```

      SETUP A=0,FILE,REC,NRECS-1

```

```

      LOAD 1ST RECORD ON THE TAPE ?
      YES, MAKE MBOOT HAPPY
      AND B=1,STARTINGADDRESS
      THEN STORE IN BOOT TABLE
      AND JUMP (NEVER RETURN).

```

```

      *XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*
      * AUX D.34 INFO, DIFS TABLE DEFINES PHYSICAL DISC STRUCTURE,
      * THE BCS EGT TABLE 'SUBUNIT' FIELD IS A POINTER TO THE DIFS
      * 'LOGICAL FILE NUMBER'.
      *XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*

```

0243	FMTDD	DEC 8			
0244	DIFS	DEF	*+1		
0245	*				
0246	*				
0247	*				
0248		LST			*SUB-UNIT-0 FILE
0249		DEC 0			-LOGICAL FILE NUMBER
0250		DEC 203			-NBR OF RECORDS IN FILE
0251		DEC 0			-SUB-UNIT
0252		DEC 0			-STARTING CYLINDER
0253		DEC 0			-STARTING SECTOR
0254		DEC 48			-SECTORS/SECTION
0255		DEC 1			-SECTIONS/RECORD
0256		DEC 203			-RECORDS/ROW
0257	*				
0258	*				
0259	*				
0260		DEC 1			*SUB-UNIT-1 FILE
0261		DEC 203			
0262		DEC 1			
0263		DEC 0			
0264		DEC 0			
0265		DEC 48			6144 WORDS
0266		DEC 1			
0267		DEC 203			
0268	*				
0269	*				
0270	*				
0271		OCT 16			FILE 1 DATA BLOCKS
0272		DEC 33			LOGICAL UNIT - 16B
0273		DEC 1			# OF DATA BLOCKS
0274		DEC 49			REMOVABLE DISC
0275		DEC 0			
0276		DEC 16			2048 WORDS/BLOCK
0277		DEC 1			

DEC 11	FILE 1	DATA BLOCKS PART 2 (SCRATCH)
0275	*	
0276	*	
0280	*	
0281	*	
0282	**	
0283	**	# OF DATA BLOCKS
0284	**	FIXED DISC
0285	**	
0286	**	
0287	**	4096 WORDS/BLOCK
0288	**	
0289	**	
0290	*	
0291	*	FILE 2
0292	*	ADC THROUGHPUT
0293	**	
0294	**	# OF THROUGHPUT BLOCKS
0295	**	FIXED DISC
0296	**	
0297	**	
0298	**	4096 WORDS/BLOCK
0299	**	
0300	**	
0301	*	
0302	*	FILE 3
0303	*	PROGRAM STACKS
0304	**	
0305	**	# OF PROGRAM STACKS
0306	**	REMOVABLE DISC
0307	**	
0308	**	
0309	**	384 WORDS/STACK
0310	**	
0311	**	
0312	*	

0313	FILE 4	ASCII TEXT	
0314			
0315	OCT 21		
0316	DEC 336	# OF ASCII RECORDS	
0317	DEC 1	REMOVABLE DISC	
0318	DEC 153		
0319	DEC 34		
0320	DEC 1	128 WORDS/RECORD	
0321	DEC 1		
0322	DEC 24		
0323	FILE 5	INDEX BLOCKS	
0324			
0325	OCT 22		
0326	DEC 208	# OF INDEX BLOCKS	
0327	DEC 1	REMOVABLE DISC	
0328	DEC 49		
0329	DEC 46		
0330	DEC 1	128 WORDS/BLOCK	
0331	DEC 1		
0332	DEC 104		
0333	FILE 6	CORELOADS	
0334			
0335	OCT 23	LOGICAL UNIT - 23B	
0336	DEC 8	# OF SYSTEM CORELOADS	
0337	DEC 1	REMOVABLE DISC	
0338	DEC 1	START CYLINDER	
0339	DEC 0	START SECTOR	
0340	DEC 48	32K/CORELOAD	
0341	DEC 6	SECTIONS/RECORD	
0342	DEC 8	RECORDS/ROW	
0343	FILE 7	COMMON	
0344			
0345			
0346			
0347			

0348	**	OCT 24	# OF COMMON RECORDS
0349	**	DEC 258	REMOVABLE DISC
0350	**	DEC 1	
0351	**	DEC 67	
0352	**	DEC 34	
0353	**	DEC 4	512 WORDS/RECORD
0354	**	DEC 1	
0355	**	DEC 86	
0356	*		
0357	*	FILE 8	USER FILE
0358	*		
0359	**	OCT 25	
0360	**	DEC 1200	# OF USER RECORDS IN PART 1
0361	**	DEC 1	REMOVABLE DISC
0362	**	DEC 177	
0363	**	DEC 0	
0364	**	DEC 1	128 WORDS/RECORD
0365	**	DEC 1	
0366	**	DEC 25	
0367	*		
0368	*	FILE 9	HEADERS FOR DATA BLOCKS
0369	*		
0370		OCT 26	
0371		DEC 48	# OF DATA BLOCK HEADERS
0372		DEC 1	REMOVABLE DISC
0373		DEC 60	
0374		DEC 0	
0375		DEC 1	
0376		DEC 1	128 WORDS/HEADER
0377		DEC 1	
0378	*		
0379	*	FILE 9	HEADERS FOR DATA BLOCKS PART 2 (SCRATCH)
0380	*		
0381	**	OCT 26	
0382	**	DEC 192	# OF SCRATCH DATA HEADERS

```

0383      **      DEC 0      FIXED DISC
0384      **      DEC 191
0385      **      DEC 32
0386      **      DEC 1      128 WORDS/HEADER
0387      **      DEC 1
0388      **      DEC 12
0389      *
0390      *      DEC -1      *END OF DISC FILE
0391      *      *****
0392      *      AUX D.33 INFO, DEFINES PHYSICAL MAGTAPE STRUCTURE.
0393      *      STRUCTURE IS CONTAINED IN THE 'DMA' AND 'SUBUNIT' FIELDS OF THE
0394      *      APPROPRIATE BCS EGT ENTRIES. THE PCS-TIME TYPINS ARE OF THE FO
0395      *
0396      *      CC,D.33C,DJ,U00$
0397      *
0398      *      WHERE CC IS THE (LOWER NUMBERED) SELECT CODE OF THE MAGTAPE, AN
0399      *      $ SPECIFIES THE DAISY-CHAIN UNIT CONTAINING THE FILE, 00 IS THE
0400      *      PHYSICAL FILE NUMBER (0,1,2,...) OF THE FILE, AND [DJ] IS SPECI
0401      *      IF THE FILE IS FORMATTED (IE, REWRITABLE).
0402      *      *****
0403      FMT33 BSS 0
0404      FMT34 DEC 1
0405      DEF FMTBB
0406      FMTBB BSS 21
0407      FORMT NOP
0408      JMP FORMT,I
0409      UNS
0410      END

```

DUMMY MAGTAPE FORMT

:

APPENDIX

C

DISC DIRECTORY (LOWER DISC) FOR RTE SYSTEM DISC

DL..1
 CR=00001
 ILAB=SYSTEM NXTR=0194 NXSEC=018 #SEC/TR=096 LAST TR- 0202 #DR TR=01

NAME	TYPE	#BLKS/LU	SCODE	TRACK	SEC	OPEN TO
B3014C	00005	00009	00000	0060	000	
B3022D	00005	00079	00000	0060	018	
B3023F	00005	00084	00000	0061	080	
B3025B	00005	00023	00000	0063	056	
A4500A	00005	00001	00000	0064	006	
F0005B	00005	00004	00000	0064	008	
F0017C	00005	00007	00000	0064	016	
A3060A	00003	00045	00000	0064	030	
F4250B	00005	00082	00000	0065	024	
B3018C	00005	00021	00000	0066	092	
RAFFPC	00003	00003	00000	0067	038	
A4501D	00005	00003	00000	0067	044	
B3052B	00005	00004	00000	0067	050	
B3025C	00005	00023	00000	0067	058	
B3055C	00005	00007	00000	0068	008	
B3050B	00005	00005	00000	0068	022	
B0009C	00005	00008	00000	0068	032	
OURTR	00003	00001	00000	0068	048	
Y7892S	00003	00002	00000	0068	050	
OURTRT	00004	00001	00000	0068	054	
F4145B	00005	00107	00000	0068	056	
Y78920	00005	00003	00000	0070	078	
Y7893S	00003	00003	00000	0070	084	
Y78930	00005	00006	00000	0070	090	
EOF	00004	00001	00000	0071	006	
ERDATS	00004	00008	00000	0071	008	
ERDAT0	00005	00002	00000	0071	024	
NIBELS	00003	00013	00000	0071	028	
Y78950	00005	00005	00000	0071	054	

A3016F	00005	00031	00000	0071	064
A3015I	00005	00075	00000	0072	030
A4503A	00005	00008	00000	0073	084
LINK6	00004	00002	00000	0074	004
LKTBG	00004	00002	00000	0074	008
#LKTBG	00005	00001	00000	0074	012
#LINK6	00005	00001	00000	0074	014
RTTR	00004	00002	00000	0074	016
RTPC	00004	00002	00000	0074	020
&DDISC	00004	00002	00000	0074	024
BEAM1	00004	00015	00000	0074	028
Y7898X	00003	00002	00000	0074	058
BEAM3	00004	00014	00000	0074	062
Y0004	00004	00002	00000	0074	090
TAT	00004	00003	00000	0074	094
DRPTR	00004	00001	00000	0075	004
DMA1S	00004	00008	00000	0075	006
Y7891Q	00004	00006	00000	0075	022
DMA10	00005	00002	00000	0075	034
Y78960	00005	00006	00000	0075	038
Y7899S	00003	00004	00000	0075	050
Y78990	00005	00002	00000	0075	058
RTCM	00004	00004	00000	0075	062
Y92000	00005	00033	00000	0075	070
PIN	00004	00004	00000	0076	040
Y7900S	00003	00002	00000	0076	048
Y79000	00005	00002	00000	0076	052
Y7891X	00004	00004	00000	0076	056
D1200	00003	00018	00000	0076	064
Y7897T	00004	00011	00000	0077	004
DO1	00005	00008	00000	0077	026
TYP1	00004	00022	00000	0077	042
FMTXX	00003	00052	00000	0077	086
Y0005	00004	00043	00000	0078	094
B3000C	00005	00054	00000	0079	084

B3053D	00005	00088	00000	0081	000
B3054C	00005	00074	00000	0082	080
B3049F	00005	00056	00000	0084	036
Y9200	00005	00033	00000	0085	052
OURCM	00003	00003	00000	0086	022
Y02210	00005	00029	00000	0086	028
Y7895S	00003	00006	00000	0086	086
BEAM	00003	00014	00000	0087	002
Y7897X	00003	00010	00000	0087	030
DOUG	00004	00008	00000	0087	050
STPAR	00004	00002	00000	0087	066
#STPAR	00005	00002	00000	0087	070
#Y0004	00005	00003	00000	0087	074
Y79030	00005	00003	00000	0087	080
Y79010	00005	00003	00000	0087	086
RAFCM	00003	00007	00000	0087	092
NIBBL	00003	00013	00000	0088	010
CLOAD	00004	00004	00000	0088	036
CDUMP	00004	00004	00000	0088	044
#CLOAD	00005	00001	00000	0088	052
#CDUMP	00005	00001	00000	0088	054
#TEST	00005	00005	00000	0088	056
D750S	00004	00002	00000	0088	066
ERASO	00005	00001	00000	0088	070
#Y7905	00005	00003	00000	0088	072
OURC1	00004	00004	00000	0088	078
OURTR1	00004	00001	00000	0088	086
EXTR	00004	00002	00000	0088	088
Y0003	00004	00004	00000	0088	092
Y7905S	00004	00003	00000	0089	004
Y79050	00005	00001	00000	0089	010
RDATA	00004	00041	00000	0089	012
ARR10	00005	00013	00000	0089	094
BMDATS	00004	00008	00000	0090	024
Y7894S	00004	00009	00000	0090	040

Y78940	00005	00004	00000	0090	0058
BMDATO	00005	00002	00000	0090	0066
Y1003S	00004	00002	00000	0090	0070
TEK00	00005	00004	00000	0090	0074
INPTS	00004	00004	00000	0090	0082
Y10030	00005	00002	00000	0090	0090
#RSTAU	00005	00005	00000	0090	0094
RSTAU	00004	00029	00000	0091	0008
#Y0005	00005	00007	00000	0091	0066
Y7909S	00004	00005	00000	0091	0080
OURC7	00004	00004	00000	0091	0090
OURTR7	00004	00001	00000	0092	0002
Y8891S	00004	00004	00000	0092	0004
Y0006	00004	00035	00000	0092	0012
CHARO	00005	00002	00000	0092	0082
DMA2S	00004	00008	00000	0092	0086
DMA20	00005	00002	00000	0093	0006
INPTA	00004	00004	00000	0093	0010
INPTB	00005	00002	00000	0093	0018
INPTO	00005	00002	00000	0093	0022
Y7906A	00004	00014	00000	0093	0026
Y79090	00005	00003	00000	0093	0054
OURTR3	00004	00001	00000	0093	0060
Y7910	00005	00006	00000	0093	0062
Y7907R	00004	00018	00000	0093	0074
Y79100	00005	00006	00000	0094	0014
Y7908S	00004	00003	00000	0094	0026
Y79080	00005	00002	00000	0094	0032
Y0003S	00004	00004	00000	0094	0036
ISHFTS	00003	00003	00000	0094	0044
ISHFT0	00005	00001	00000	0094	0050
Y79020	00005	00008	00000	0094	0052
Y1002T	00004	00075	00000	0094	0068
Y7906R	00004	00015	00000	0096	0026
Y7904S	00004	00007	00000	0096	0056

OUDEU	00004	00006	00000	0096	070	+001
OURDEU	00004	00001	00000	0096	082	+002
Y7912S	00004	00004	00000	0096	084	+003
Y79120	00005	00004	00000	0096	092	+004
Y79UPS	00004	00007	00000	0097	004	+005
UELOCP	00005	00008	00000	0097	018	+006
Y7907A	00004	00021	00000	0097	034	+007
ROMU	00004	00003	00000	0097	076	+008
ROMVO	00005	00003	00000	0097	082	+009
UELOCO	00005	00010	00000	0097	088	
UTESTS	00004	00011	00000	0098	012	
FXL	00006	00054	00000	0098	034	
FXL2	00006	00052	00000	0099	046	
FXL3	00006	00055	00000	0100	054	
COPYF	00006	00022	00000	0101	068	
BEAMF	00006	00045	00000	0102	016	
PATRN	00006	00037	00000	0103	010	
CALC1	00006	00026	00000	0103	084	
#LINKD	00005	00001	00000	0104	040	
BCSJ0B	00007	00024	00000	0104	042	
BCSJ0B	00007	00024	00000	0104	090	
BCSJ0B	00007	00024	00000	0105	042	
BCSJ0B	00007	00024	00000	0105	090	
BCSJ0B	00007	00024	00000	0106	042	
BCSJ0B	00007	00024	00000	0106	090	
BCSJ0B	00007	00024	00000	0107	042	
BCSJ0B	00007	00024	00000	0107	090	
BCSJ0B	00007	00024	00000	0108	042	
BCSJ0B	00007	00024	00000	0108	090	
RTGEN	00003	00003	00000	0109	042	
BTEST	00006	00097	00000	0109	048	
Y7907	00004	00002	00000	0111	050	
#TYPI	00005	00004	00000	0111	054	
Y1111S	00004	00066	00000	0111	062	
%DDISC	00005	00002	00000	0113	002	

DDISC	00006	00025	00000	0113	006	
INPUT	00004	00006	00000	0113	056	
#INPUT	00005	00005	00000	0113	068	
*CSORT	00005	00072	00000	0113	078	
Y7911S	00004	00041	00000	0115	030	
OUCSC1	00004	00006	00000	0116	016	
OUCST1	00004	00001	00000	0116	028	
OUCSC5	00004	00006	00000	0116	030	
OUCST5	00004	00001	00000	0116	042	
OUC5	00004	00006	00000	0116	044	
Y7901S	00004	00006	00000	0116	056	
Y7903S	00003	00004	00000	0116	068	
CHOUT	00003	00028	00000	0116	076	
ARR1	00004	00014	00000	0117	036	
TEKS	00003	00056	00000	0117	064	
Y7907S	00004	00020	00000	0118	080	
Y7910S	00004	00007	00000	0119	024	
Y7902S	00004	00009	00000	0119	038	
VELOC5	00004	00008	00000	0119	056	
PLMOD1	00004	00047	00000	0119	072	
PCSINP	00003	00002	00000	0120	070	
CMMAND	00004	00004	00000	0120	074	
BTESTS	00003	00010	00000	0120	082	
CSORT	00006	00092	00000	0121	006	+001
CMMAND	00004	00004	00000	0122	094	+002
CMMAND	00004	00004	00000	0123	006	
FMTXX1	00004	00052	00000	0123	014	
#FMTX1	00005	00004	00000	0124	022	
FMTXX2	00004	00052	00000	0124	030	
#FMTX2	00005	00004	00000	0125	038	
TAT1	00004	00003	00000	0125	046	
TAT2	00004	00003	00000	0125	052	
#TAT2	00005	00001	00000	0125	058	
BR1TR1	00004	00002	00000	0125	060	
PCSINP	00003	00002	00000	0125	064	+001

#RDATA	00005	00006	00000	0125	068	
BR1CSC	00003	00014	00000	0125	080	+003
CMMAND	00004	00004	00000	0126	012	+004
CMMAND	00004	00004	00000	0126	020	
RAFMP	00003	00007	00000	0126	028	
OURC6	00004	00006	00000	0126	042	
OURTR6	00004	00001	00000	0126	054	
UPROF	00005	00010	00000	0126	056	
Y7913S	00004	00015	00000	0126	076	
Y79070	00005	00014	00000	0127	010	
Y7906	00005	00010	00000	0127	038	
Y79130	00005	00014	00000	0127	058	
SOUND	00005	00014	00000	0127	086	
Y79060	00005	00009	00000	0128	018	
#TAT1	00005	00001	00000	0128	036	
Y79040	00005	00006	00000	0128	038	
#TAT0	00005	00001	00000	0128	050	
Y10020	00005	00049	00000	0128	052	
DISC	00003	00024	00000	0129	054	
#Y0010	00005	00004	00000	0130	006	
USTRT	00004	00002	00000	0130	014	
#USTRT	00005	00001	00000	0130	018	
MPTR	00004	00002	00000	0130	020	
#FPGRD	00005	00008	00000	0130	024	
OLCMCS	00003	00014	00000	0130	040	
RDTR	00004	00002	00000	0130	068	
#DISC	00005	00004	00000	0130	072	
MPTR7	00004	00001	00000	0130	080	
BSNST2	00003	00024	00000	0130	082	
BSNST2	00003	00024	00000	0131	034	+001
MPCM7	00004	00004	00000	0131	082	
MPCM6	00004	00004	00000	0131	090	
MPCM5	00004	00006	00000	0132	002	
OURTR5	00004	00001	00000	0132	014	
MPTR5	00004	00001	00000	0132	016	

NPTRG	00004	00001	00000	0132	018	
#TAT	00005	00001	00000	0132	020	
RNFTTR	00004	00002	00000	0132	022	
NPTR1	00004	00001	00000	0132	026	
MPCM1	00004	00004	00000	0132	028	
RDCPTR	00004	00002	00000	0132	036	
RDCPCM	00004	00004	00000	0132	040	
Y0011	00003	00034	00000	0132	048	
ANMOD5	00004	00083	00000	0133	020	
PLMOD5	00004	00047	00000	0134	090	
SUPROF	00004	00016	00000	0135	088	
Y7896S	00004	00015	00000	0136	024	
PGMHED	00004	00007	00000	0136	054	
Y7906S	00004	00017	00000	0136	068	
ANMOD7	00004	00089	00000	0137	006	
MP003A	00004	00011	00000	0138	088	
MP1007	00004	00110	00000	0139	014	
D7500	00005	00001	00000	0141	042	
DCPTR	00004	00002	00000	0141	044	
NPTR2	00004	00002	00000	0141	048	
TYPI1S	00004	00050	00000	0141	052	
#TYPI1	00005	00006	00000	0142	056	
NPTR0	00004	00001	00000	0142	068	
MP007A	00004	00138	00000	0142	070	
MPTR1C	00004	00001	00000	0145	058	
MPCM1C	00004	00007	00000	0145	060	
Y1007A	00005	00077	00000	0145	074	
TEKOC	00005	00003	00000	0147	036	
MPSYS	00007	00024	00000	0147	042	
MPSYS	00007	00024	00000	0147	090	+001
MPSYS	00007	00024	00000	0148	042	+002
MPSYS	00007	00024	00000	0148	090	+003
MPSYS	00007	00024	00000	0149	042	+004
MPSYS	00007	00024	00000	0149	090	+005
MPSNST	00003	00024	00000	0150	042	

MPSNST	00003	00024	00000	0150 090	+001
ASKQ	00004	00039	00000	0151 042	
MP007C	00004	00131	00000	0152 024	
RTST	00004	00150	00000	0154 094	
Y7897S	00004	00007	00000	0158 010	
Y7891S	00004	00004	00000	0158 024	
Y7898S	00004	00003	00000	0158 032	
Y78910	00005	00001	00000	0158 038	
Y78980	00005	00002	00000	0158 040	
OLCM1	00004	00004	00000	0158 044	
OLTR1	00004	00001	00000	0158 052	
Y78970	00005	00004	00000	0158 054	
ASKQ1	00004	00037	00000	0158 062	
OLTR	00004	00002	00000	0159 040	
CSORTC	00004	00025	00000	0159 044	
BASYS	00007	00024	00000	0159 094	+001
BASYS	00007	00024	00000	0160 046	+002
BASYS	00007	00024	00000	0160 094	+003
BASYS	00007	00024	00000	0161 046	+004
BASYS	00007	00024	00000	0161 094	+005
BASYS	00007	00024	00000	0162 046	
BASNST	00003	00024	00000	0162 094	+001
BASNST	00003	00024	00000	0163 046	
SOGJOB	00007	00024	00000	0163 094	
#ASKQ1	00005	00016	00000	0164 046	
OLTR2	00004	00002	00000	0164 078	
OLCM7	00004	00004	00000	0164 082	
OLTR7	00004	00001	00000	0164 090	
OL007A	00004	00055	00000	0164 092	
OLTR0	00004	00002	00000	0166 010	
OL007B	00004	00056	00000	0166 014	
MP007B	00004	00130	00000	0167 030	
OL002A	00004	00092	00000	0170 002	
#Y0003	00005	00003	00000	0171 090	
#ASKQ	00005	00015	00000	0172 000	

#Y0001	00005	00010	00000	0172 030	
Y10070	00005	00080	00000	0172 050	
#MPINF	00005	00009	00000	0174 018	
MPINF	00004	00029	00000	0174 036	
Y0001	00004	00069	00000	0174 094	
#DCP	00005	00010	00000	0176 040	+001
#DCP	00005	00010	00000	0176 060	+002
#DCP	00005	00010	00000	0176 080	
PCSJOB	00007	00100	00000	0177 004	
DCPSYS	00007	00024	00000	0179 012	+001
DCPSYS	00007	00024	00000	0179 060	+002
DCPSYS	00007	00024	00000	0180 012	
MP002B	00004	00165	00000	0180 060	
Y10020	00005	00060	00000	0184 006	
RAF	00007	00024	00000	0185 030	+001
RAF	00007	00024	00000	0185 078	+002
RAF	00007	00024	00000	0186 030	+003
RAF	00007	00024	00000	0186 078	
MP002A	00004	00165	00000	0187 030	
MP002C	00004	00165	00000	0190 072	

: