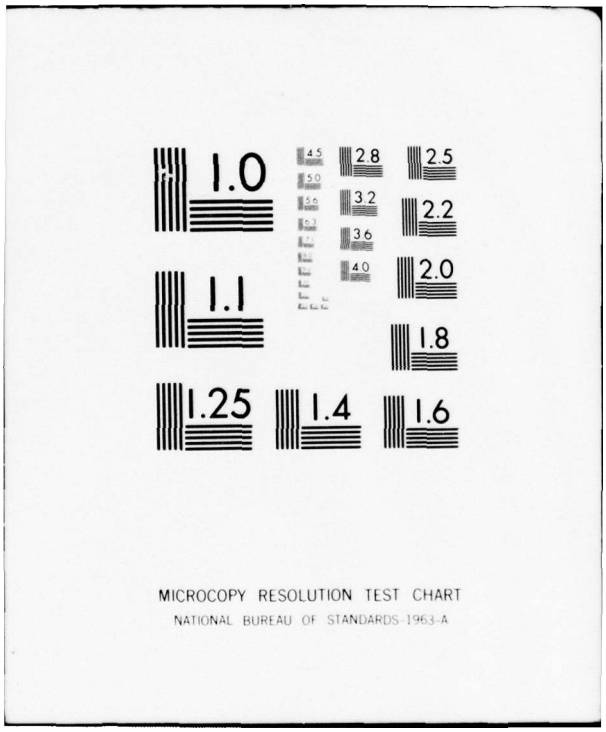
AD-A038 875 AIR FORCE WEAPONS LAB KIRTLAND AFB N MEX F/G 17/7
COMPUTER-PROGRAM FOR FAA/ATCRBS TAPE READER, PROGRAM JASPER I.(U)
MAR 77 M WETHERBE, W J MOULDS
UNCLASSIFIED AFWL-TR-76-219 NL

| OF |
AD
A038875

END

DATE
FILMED
5-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Code 23
9.5

AFWL-TR-76-219

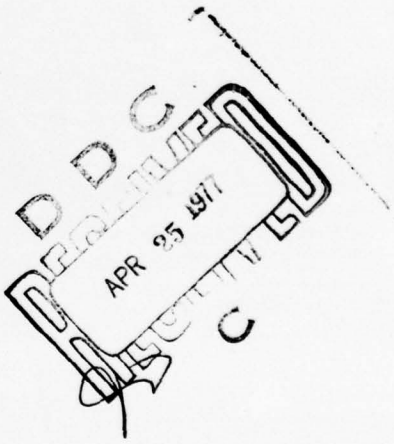
AFWL-TR-
76-219

AD A038875

COMPUTER PROGRAM FOR FAA/ATCRBS TAPE READER, PROGRAM JASPER I

March 1977

Final Report



Approved for public release; distribution unlimited.

BEST AVAILABLE COPY

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117

DDC FILE COPY
ADJ NO.

AFWL-TR-76-219

This final report was prepared by the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico, under Job Order 99910013. Mr. William J. Moulds (SUE) was the Laboratory Project Officer-in-Charge.

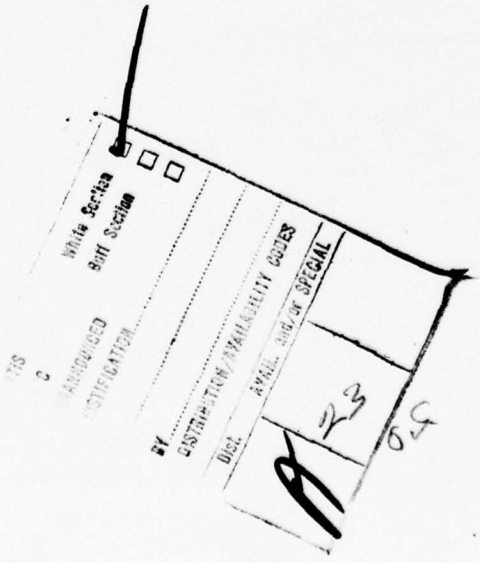
When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

WILLIAM J. MOULDS
Chief, Engineering Branch
Project Officer

GUSTAV J. FREYER
Colonel, USAF
Commander, AFWL



DO NOT RETURN THIS COPY. RETAIN OR DESTROY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFWL-TR-76-219	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): COMPUTER PROGRAM FOR FAA/ATCRBS TAPE READER, PROGRAM JASPER I.	5. TYPE OF REPORT & PERIOD COVERED: Final Report, Oct 74 - May 76
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Mead, William J. / Wetherbe, W. J. / Moulds	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Air Force Weapons Laboratory (SUE) Kirtland Air Force Base, NM 87117	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62601F/99910013
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE: March 1977
	13. NUMBER OF PAGES: 72
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 72p.	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

DDC AVAILABILITY NOTICE APR 25 1977 C

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

False Targets
False Replies
Interrogator Beacon
Reflecting Surfaces
Target Reply
Secondary Radar Beacon
Air Traffic Control Radar Beacon System (ATCRBS)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report presents an AFWL modified computer code, JASPER, which combines two computer codes originally developed by MIT Lincoln Laboratories. JASPER automatically identifies reflecting surfaces which cause false targets in the Air Traffic Control Radar Beacon System (ATCRBS), and determines their positions and orientation. The code is presently capable of processing data extracted from ARTS-III terminal area processing systems. An explanation of the code is provided and a sample problem is included so that the user may

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

013 150

20. ABSTRACT

become familiar with the input data and the flexibility of the output list. A later version of JASPER in "close to ANSI" Fortran has been written. This version can be easily adapted to use on most computers. Data unpacking is done without the use of SHIFT or MASKING expressions. JASPER is written in Fortran version 3.4.4 (SCOPE) for the CDC 7600 computer.

PREFACE

This research was performed under Program Element 62601F.

Inclusive dates of research were October 1974 through 1 May 1976. This report was submitted on 15 May 1976 by the Air Force Weapons Laboratory Project Officer, Mr. W. J. Moulds (SUE).

The authors wish to acknowledge Mr. A. G. Cameron of Lincoln Laboratories for the guidance and help given during the writing of Program JASPER.

This technical report has been reviewed and is approved.

CONTENTS

Section		Page
I	INTRODUCTION	5
	1. Purpose	5
	2. Background Information	5
	3. General Description of Program	6
	4. General Description of FAA Tape	6
II	DETAILED DESCRIPTION OF FAA TAPE	7
III	DETAILED DESCRIPTION OF PROGRAM	17
	1. Card Input	17
	2. Printer Output	17
IV	DESCRIPTION OF MAJOR SUBROUTINES	20
	1. Subroutine GEN	20
	2. Subroutine JADE	20
	3. Subroutine RUBY	20
	4. Subroutine SET	20
	APPENDIX A - COMMON BLOCK VARIABLES	23
	APPENDIX B - GUIDE FOR READING FLOW CHART 1	27
	APPENDIX C - GUIDE FOR READING FLOW CHART 2	30
	APPENDIX D - GUIDE FOR READING FLOW CHART 3	33
	APPENDIX E - JASPER PROGRAM LISTING	35
	APPENDIX F - ILLUSTRATIVE RUNS	59
	APPENDIX G - GLOSSARY OF PRINCIPAL VARIABLE NAMES USED FOR OUTPUT	66
	APPENDIX H - DETERMINATION OF SYSTEM REPEATABILITY	68

ILLUSTRATIONS

Figure		Page
1	Message Type 1	9
2	Message Type 2	10
3	Message Type 4	10
4	Message Type 6	11
5	CTST Word	12
6	RGAZT Word	12
7	CORDT Word	13
8	XYDOT Word (Active Track)	13
9	TIMET Word	14
10	ABEAT Word	14
11	RBEAT Word	15
B1	Flow Chart 1 - JASPER	28
C1	Flow Chart 2 - Subroutine GEN	31
D1	Flow Chart 3 - Subroutine JADE	34
H1	Beacon Half-Angle Definition	69

TABLES

Number		Page
1	Message Definition (Code, Word Count, Type, Name)	8
2	Subroutines for Program JASPER	21

SECTION I

INTRODUCTION

1. PURPOSE

This report provides the Air Force with an analytical tool with which current and potential false target reflectors can be evaluated or determined. Furthermore, an explanation of the computer code is provided and a sample problem is included so that the user may become familiar with the input data and the flexibility of the output list.

The program decodes and lists data from the Federal Aviation Administration (FAA) Air Traffic Control Radar Beacon System's (ATCRBS) data extraction tapes. These data are screened from a very large volume of data recorded on these tapes and the processed data are related to false target reflectors. These reflectors are identified by azimuth, range and orientation.

2. BACKGROUND INFORMATION

JASPER is an AFWL modified code combining features from two computer codes supplied by Lincoln Laboratories, Program PT and Program LOGAN.

PT selects all radar target reports and sends them to a subroutine named REFLECT.[1] REFLECT examines all of these reports and upon finding a possible false target, lists a "questionable data group" along with a computed position of the reflector.

LOGAN does not use the subroutine REFLECT but does decode and list data contained in a table called "Central Track Store". It decodes and lists "Target Replies" (hits). LOGAN also decodes and lists the target reports used by PT. Target reports are generated on the FAA tapes when replies from a given target persist for a preset number of times. This number is usually set at six. Central Track Store supplies detailed information related to a given target such as position, velocity, etc.

---

(1) Private communication with Mr. A. G. Cameron, Lincoln Laboratories, May 1974.

3. GENERAL DESCRIPTION OF PROGRAM

JASPER uses REFLECT taken from PT and also decodes The Central Track Store as done by LUGAN. Combining these features with other features unique to JASPER resulted in a single program adapted to the type of analysis being done by the AFWL at Kirtland Air Force Base.

For our analysis it was desired to have the latest track store data for a given target and an output of all target reports immediately proceeding and following any "questionable data group" detected by REFLECT. JASPER does this by maintaining in memory the latest one hundred target reports and the latest track store data for a designated target. When REFLECT detects a "questionable data group" it lists all of this information and sets a flag which causes JASPER to list following target reports until the current scan is completed.

Program flow logic and the method of data unpacking and decoding was redesigned. The new design was adapted to the CDC 6600 computer and resulted in a significant saving in program run time.

4. GENERAL DESCRIPTION OF FAA TAPE

This tape is written with odd parity (bin mode). Data are packed into 30-bit words. Records consist of 200 30-bit words. All data are defined in terms of a 30-bit word structure.

The first word of each record is always a record number. Records are numbered consecutively starting at zero. If an error occurs in recording, the record is repeated until a good recording is made. Repeated records carry the same record number.

The second word of a record is always the first word of a message* and the last five bits of the first word of a message are always the message code (MC). For each legal MC there is a message word count and definition of format. If a given message will not fit in the remaining space of a record it will be made the first message of the next record. An MC of zero indicates there are no remaining messages in a given record. An illegal MC means that data sync has been lost.

---

* See table 1 and figures 1 through 11 for message definition and word format. Narrative description follows in Section II.

SECTION II

DETAILED DESCRIPTION OF FAA TAPE

Table 1 shows the word count, message type and message name associated with each valid message code.

JASPER decodes only messages with codes 1 through 6. The word count is required on all messages to maintain data synchronization.

This report makes reference to message type 1, message type 2, message type 4, and message type 6. Message type 2 appears with an MC of either 2 or 3, message type 4 appears with an MC of either 4 or 5.

Message type 1 contains the antenna azimuth and is followed by reply words if a target is observed by ATCRBS. The reply words are identified with a 1 in bit position 29 (see formats for the azimuth word and the reply word).

JASPER uses the azimuth word to update the scan count and monitors bit 29 of following words in order to pass through all reply words which might follow.

Message type 2 is passed on to the subroutine, REFLECT.

Message type 4 is used to update the sector time clock.

When a message type 6 appears, word 1 of parameter card 4 is checked. If this word reads 7777 the track store table is updated. Otherwise the update is made only if the beacon code form message type 6 matches word 1 of parameter card 4.

Note that the word count of message type 6 was originally 17. Because of continual updating of ATCRBS by the FAA, this word count was changed to 19 and currently is 20. Depending upon date of extraction tape, the word count will have been revised accordingly. Therefore, tapes dated before August 1974 use 17, tapes dated between August 1974 and 24 February 1975, use 19, and tapes dated after 24 February 1975, use 20 (see figure 6). The word count on message type 19 is currently 13 but has been 12. The word count of message type 15 is listed as a variable. The count is extracted from the front end of the message.

Word formats used by JASPER are shown following table 1, in figures 1 through 11.

Table 1.

MESSAGE DEFINITION (CODE, WORD COUNT, TYPE, NAME)

MC	WRDS	TYPE	NAME
1	1 + N REPLY WRDS	1	BEACON REPLIES
2	2	2	TARGET REPORTS SUBSYSTEM 1
3	2	2	TARGET REPORTS SUBSYSTEM 2
4	1	4	SECTOR TIME SUBSYSTEM 1
5	1	4	SECTOR TIME SUBSYSTEM 2
6	17/19/20	6	TRACKING
7	11	7	KEYBOARD ENTRIES
8	7	8	DISPLAY OUTPUT
9	4	9	AUTO ACQUIRE
10	10	10	AUTO TERMINATE
11	4	11	AUTO ASSIGN CODE
12	4	12	AUTO TAB COAST
13	1	13	AUTO HANDOFF COAST
14	1	14	DATA LOSS
15	VARIABLE	15	FLIGHT PLANS
16	6	16	BCN INDENT.
17	1	17	MEMORY DUMP - NO STOP
18	1	18	MEMORY DUMP - AND STOP
19	12/13	19	TRIAL TRACKING

MC = Message Code
WRDS = Word Code
TYPE = Message Type
NAME = Message Name

AZIMUTH WORD

29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		S	M	V	D	N	0	A	C	0	0	0	AZIMUTH LSB = 1 ACP												MC = 1				

S = Subsystem

M = Minimum Range Limit on

V = Overload indicator

D = DAS alarm

N = Sensor alarm

A = Mode A Replies follow

C = Mode C Replies follow

REPLY WORD

29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	G	S	If Mode A / If Mode C					BCN Code / Alt (Gray Code)							E	R	X	0	0	RANGE					LSB = 1/16 NM			

G = Garble

S = Special Position Indicator

X = Illegal if Mode C

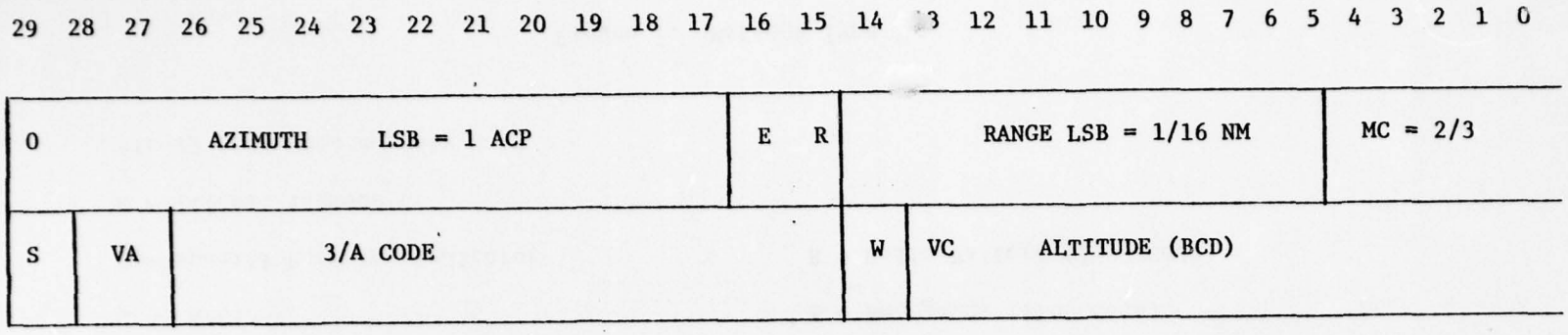
Bit 29 indicates a reply word

E = Emergency (7700 Code)

R = Radio Failure (7600 Code)

Figure 1. Message Type 1.

TARGET REPORT



S = Special Position Indicator

VA = Mode A Validity

VC = Mode C Validity

W = Strong Target if set to 1

Figure 2. Message Type 2.

SECTOR TIME

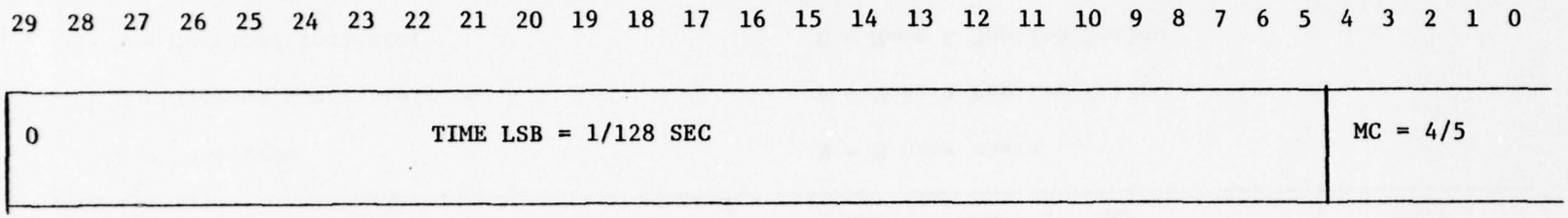


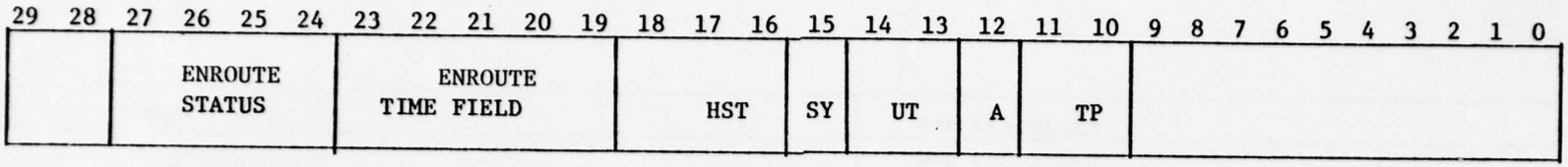
Figure 3. Message Type 4.

TRACK STORE DATA

	29 28 27 26 25 24	23 22 21 20 19 18	17 16 15 14 13 12	11 10 9 8 7 6	5 4 3 2 1 0
1	TRACK NO.		POSITION NO.		MC = 6
2	TIME LSB = 1/1024 SEC				
3	ACID 1	ACID 2	ACID 3	ACID 4	ACID 5
4	ACID 6	ACID 7	ALT	ALT	ALT
5			CTST (ADDED AUG 74)		
6			RGAZT		
7			CORDT		
8			XYDOT		
9			TIMET		
10			ABEAT		
11			TRACT (ADDED 8 JAN 75		
12			CFLGT		
13			LGCT		
14			RBEAT		
15			CTSST		
16			DBIT (ADDED AUG 74)		
17	BIN PARAMETER			BIN PARAMETER	
18	BIN PARAMETER			BIN PARAMETER	
19	BIN PARAMETER			BIN PARAMETER	
20	BIN PARAMETER			BIN PARAMETER	

Figure 4. Message Type 6

FORMATS OF SELECTED WORDS FROM MESSAGE TYPE 6

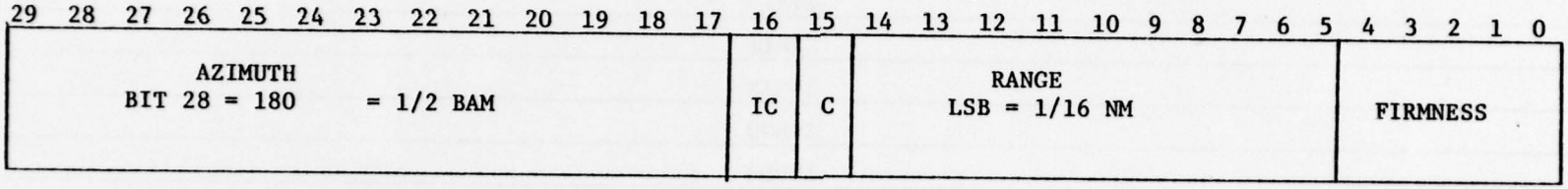


HST = Handoff Status Bits

SY = Subsystem Bit, SY = 0 for Subsystem 1, SY = 1 for Subsystem 2

UT, A and TP define the type and status of the track file

Figure 5. CTST Word.



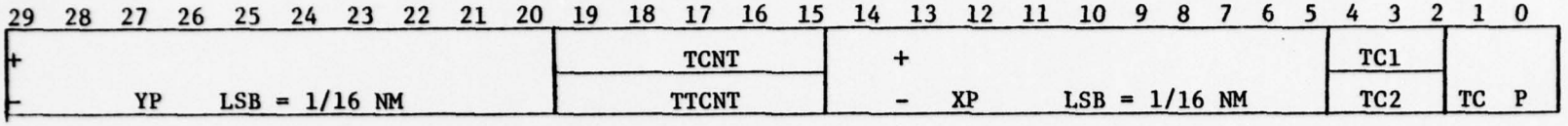
IC = Inhibit Flag, IC = 1 for inhibit correlation

IC = 0 for do not inhibit

C = Correlation Flag C = 1 means track correlated this scan

C = 0 means track did not correlate this scan

Figure 6. RGAZT Word.



YP = Predicted Y Coordinate

TCNT = Number of Coast Scans

TTCNT = Number Turning Track Correlations

XP = Predicted X Coordinate

TC1 = Track Class, Straightline

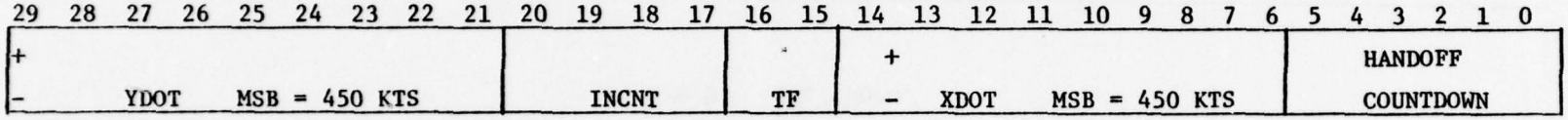
TC2 = Track Class, Turning

TC = A shared flag for initial and turning correlations

P = The predict bit, P = 1, track has been predicted this scan

P = 0, track has not been predicted this scan

Figure 7. CORDT Word.



YDOT = Y Component of Velocity in NM/SEC

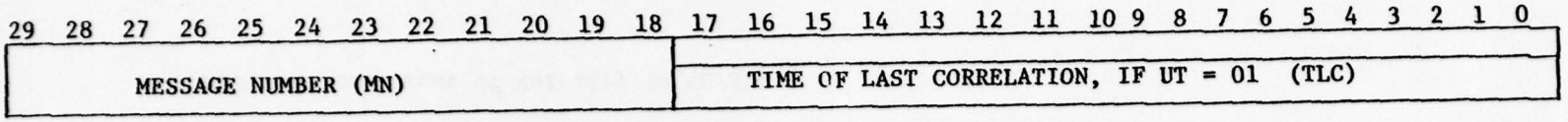
INCNT = Number of Successive Successful Correlations

TF = Flag, TF = 01, Track was processed in first scan

TF = 00, track was not processed in first scan

XDOT = X Component of Velocity in NM/SEC

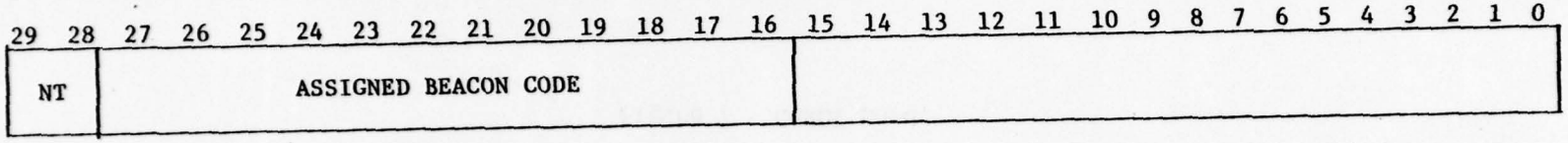
Figure 8. XYDOT Word (Active Track).



MN = Number of the last TI or TA Message

TLC = Time of last successful correlation in seconds

Figure 9. TIMET Word.

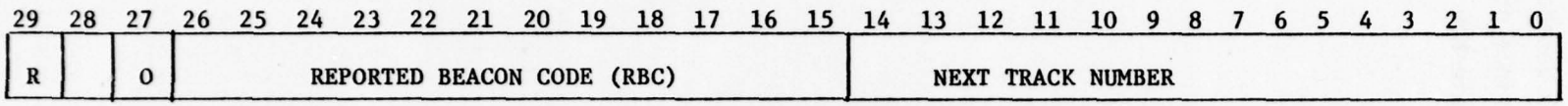


NT = Beacon code status, NT = 00, track has ABC

NT = 01, track has TABC (TENTATIVE ABC)

NT = 10, track has no ABC

Figure 10. ABEAT Word.



R = RBC status bit, R = 1, track has no RBC on this scan

R = 0, track has an RBC on this scan

Figure 11. RBEAT Word.

JASPER includes a subroutine which will decode and list reply words from message type 1. It is not in current use since printer output would be excessive. Comment cards within the code indicate how this subroutine can be made active.

SECTION III

DETAILED DESCRIPTION OF PROGRAM

1. CARD INPUT

The first cards read are for comments or documentation to be listed on printer output. Up to nine cards may be used. The last card must have a X in Column 1.

The comments cards are followed by four parameter cards.

CARD 1 FORMAT 3I5 NPSS, JPSS, LRCRD

The program skips NPSS records and then processes JPSS records from the data input tape.

LCRD = 1, for 100 word input records (original FAA tape)

LRCRD = 5, for 500 word input records (reformated FAA tape)

CARD 2 FORMAT 19I3

This card contains the message word count for each of the message codes, 1 through 14 and 16 through 19. Use 99 in field number 15.

CARD 3 FORMAT 2I5

These two words are required since some tapes combine data from two control towers under subsystem 1 and sybsystem 2. Enter 2 and 4 for subsystem 1. Enter 3 and 5 for subsystem 2.

CARD 4 FORMAT 2I5

Word 1 is a four digit beacon code. The latest track store information on that code will be included with the questionable group printout. Word 2 is a maximum target altitude in feet.

2. PRINTER OUTPUT

The first page of output is a description of card input. The next page is a list of card input. This is followed by data derived from the FAA tape.

Whenever a beacon code appears in the tape data for the first time it is listed with the label ADDITIONAL CODE.

Whenever signals from a false target appear, a printout of data is initiated. A description of these data follow.

The first line will contain the label, QUESTIONABLE GROUP CODE =, followed by a beacon code and a second label, LAST SECTOR TIME, followed by the current reading of the sector time clock.

Below this line are printed the scan numbers, the azimuth values (degrees, magnetic heading) and the range values (nautical miles) from the group of target reports in question. This is followed by a computed location, including orientation, of the reflector which caused the false target. If there is a target split off the reflector, both the split and the false target are printed.

Selected data from the latest track store table are the next information printed. If the user asks for a track store monitor on a particular beacon code, only data belonging to that code will be stored. If a dummy code of 7777 is specified, all data will be stored. In this case the data printed may belong to any beacon code in the area.

Column headings for these data have meaning as shown below.

UT	Used to define the type and status of the track file.
A	Used to define the type and status of the track file.
TP	Used to define the type and status of the track file.
HR	Hours
MIN	Minutes
SEC	Seconds
ACID	Aircraft Identification
ALT	Aircraft Altitude
ABCD	Assigned Beacon Code
RBCD	Reported Beacon Code
AZM	Azimuth (degrees)
RGN	Range (nautical miles)

XP	Predicted X coordinate (nautical miles)
YP	Predicted Y coordinate (nautical miles)
XD	X component of velocity (nautical miles/second)
YD	Y component of velocity (nautical miles/second)
VEL	Velocity (nautical miles/second)
HDNG	Heading (degrees)

The final printout initiated by the appearance of a false target is a sequential listing of the current scan and following scan target reports. A three line gap in the listing shows the end of the current scan. The first target report of the following scan plus one is also printed.

Column headings for these data have meaning as shown below.

H	Hour
M	Minute
SEC	Seconds
RNG	Range (nautical miles)
AZM	Azimuth (degrees)
ALT	Aircraft altitude
BCD	Beacon code
VA	Mode A validity
VC	Mode C validity
S	Special position indicator
F	Flag F = 1, when target report precedes or includes the questionable group F = 5, when target report follows the questionable group.
RCRD	Current record count from the FAA data tape.

SECTION IV

DESCRIPTION OF MAJOR SUBROUTINES*

1. SUBROUTINE GEN

   a. Entry ZAP

A call to ZAP processes one record of data from buffer JB. It makes calls to ZAP1, ZAP2, ZAP4 and ZAP6. These are all entries in subroutine JADE, and are primarily used for unpacking and processing data transferred through buffer JX. ZAP1 is used for message type 1 data, ZAP2 for message type 2 data, ZAP4 for message type 4 and ZAP6 for message type 6 data. A very important function of ZAP is branching or routing of program action. The branching logic is indexed by the variable MC extracted directly from the data. MC is the message code value.

2. SUBROUTINE JADE

Used for unpacking and processing data transferred through buffer JX. See above.

3. SUBROUTINE RUBY

   a. Entry TRSPLT

Used by ZAP6 for unpacking words following the fourth word of message type 6 data. Special routing is used depending upon the value input from parameter card 2, word 6, JMC (6). This is the word count for message type 6 and has changed from 17 to 19 and currently is 20. The special routing makes possible the correct processing of data from any of the three periods referred to in Section II.

4. SUBROUTINE SET

   a. Entry LOAD

This entry clears buffer JA (100), JB (200), JX (50) and reads the next data record from tape to buffer JA (100).

LEN is set to the number of words read, KNT (record count) is updated and LAST is set to 0 for normal return, set to 1 for an end of file return.

   b. Entry SPLIT

This entry converts 60-bit words from JA to 30-bit words and stores them in buffer JB.

*See Table 2.

c. Entry PRNTA

Currently not used, if called PRNTA will make an octal listing of data in buffer JA.

d. Entry PRNTB

PRNTB makes an octal listing of the contents of buffer JB. It is called when data synchronization has been lost.

Table 2.

SUBROUTINES FOR PROGRAM JASPER

A.	MAIN
B.	SET
C.	GEN
D.	JADE
E.	RUBY
F.	JADER
G.	REFLECT

e. Flow charts for this program are shown in figures B1, C1, and D1. Guides for reading these flow charts are in appendixes B, C, and D.

f. The use of common block variables is explained in appendix A.

APPENDIX A

COMMON BLOCK VARIABLES

1. COMMON/BLKI/

JA(100), JB(200), LEN, LAST, MC, JX(50), JP, KNT, KX

JA(100)	- Data from input tape
JB(200)	- Input data converted to 30-bit words
LEN	- Input record length
LAST	- Flag, set at 0 for normal return set at 1 for end of file return
MC	- Message code extracted from data
JX(50)	- Used for unpacked message words
JP	- Used for indexing JB(200), indicates the *first word of a message*
KNT	- Record count
KX	- Count of target replies in a given type 1 message (first reply in JX(2), last reply in JX(KX)

The above common block appears in the main program, and in the following subroutines:

SET

GEN

JADE

RUBY

2. COMMON/BLK2/ SCNDS, LHR, LMN, FSCS, JAFC(5), ASEC, KAFC(4), FAFC(8)

SCNDS - Latest sector time in seconds extracted from message Type 4/5

LHR, LMN, FSCS - The above time in hours, minutes and seconds (see Entry ZAP4)

JAFC(5), ASEC, KAFC(4), FAFC(8) are used to store the latest selected track store data, message type 6 (selected by user option with Beacon Code ID)

These data are defined as follows:

JAFC(1)	- UT
JAFC(2)	- A
JAFC(3)	- TP
JAFC(4)	- Hour
JAFC(5)	- Minutes
ASEC	- Seconds
KAFC(1)	- Aircraft ID
KAFC(2)	- Altitude
KAFC(3)	- Assigned Beacon Code
KAFC(4)	- Reported Beacon Code
FAFC(1)	- Azimuth
FAFC(2)	- Range
FAFC(3)	- A/C position X
FAFC(4)	- A/C position Y
FAFC(5)	- X Dot
FAFC(6)	- Y Dot
FAFC(7)	- Velocity
FAFC(8)	- Heading

The above common block appears in the main program, and in the following subroutines:

GEN

JADE

REFLECT

3. *COMMON/BLK3/NHR(100), NMN(100), FSC(100), FRNG(100), FAZ(100), NALT(100), NCD(100), NST(100), NVC(100), NVA(100), NSP(100), KAP, LAP, NKNT(100)*

All buffers dimensioned 100 are used for storing the latest 100 target reports (message type 2) with related sector time (NHR, NMN, FSC) and input record count (NKNT).

These data are defined as follows:

NHR(100)	- Hours
NMN(100)	- Minutes
FSC(100)	- Seconds
FRNG(100)	- Range
FAZ(100)	- Azimuth
NALT(100)	- Altitude
NCD(100)	- Beacon Code
NST(100)	- Strong Target Flag
NVC(100)	- Mode C validity
NVA(100)	- Mode A validity
NSP(100)	- Special position indicator
NKNT(100)	- Input record count
KAP	- This is an end around index showing position 1 of the last 100 target reports.
LAP	- This is a flag, when set to 1 entry ZAP will list target reports until a change in scan count occurs. Reflect sets lap at 1 following a questionable group data list.

The above common block appears in subroutines JADE and REFLECT.

4. COMMON/BLK4/LTR(100)

LTR(100) is used to store unpacked words of track store data starting with word 5. The unpacking is done by entry TRSPLT when called by ZAP6.

The above common block appears only in subroutine JADE

5. COMMON/BLK5/KMC(19)

JMC(19) is loaded from parameter Card 2. It contains the message word count for each of the message codes, 1 through 14, and 16 through 19. Since message code 15 has a variable word count a 99 is used in JMC(15). These word counts are used to update JP, the index indicating the first word of a message in buffer JB.

The above common block appears in subroutine GEN and RUBY.

6. COMMON/BLK6/LF(13), LG(7), LH(4), NCID(45), JTM(5), JAL(3), JID(7)

LF(13) is used to store the first 13 bits of an azimuth word (Message type 1) reading from left to right.

LF(9) - Set to 1 means Mode A replies follow

LF(10) - Set to 1 means Mode C replies follow

A complete format definition of LF(13) is attached

LG(7) and LH(4) are used for temporary buffering

NCID(45) contains a table of BCD code

JIM(5), JAL(3), JID(7) are used for temporary buffering

The above common block appears only in subroutine JADE.

APPENDIX B

GUIDE FOR READING FLOW CHART 1

This flow chart describes the flow logic of the main program. The first three blocks of the chart prepare the program for the basic loop which drives the program. One pass through this basic loop is required to process one record of tape input.

The variables used have meaning as listed below:

NCNT - Record Count.

NPSS - Number of records to be skipped.

JPSS - Number of records to be processed.

NUMB - Record # N.

NUMA - Records # N-1.

LY4 - Set at 4 or 5 by user option.

LY2 - Set at 2 or 3 by user option.

Where branching occurs, the following abbreviations are used:

LE - Less than or equal

GT - Greater than

EQ - Equal

NE - Not equal

Subroutine names shown in the chart are SPLIT and ZAP.

A call to SPLIT is used to split one record of 60-bit words into 30-bit words.

A call to ZAP is used to process one record of 30-bit words.

A description of action during a pass through the main loop is as follows:

At the top of the loop, the input buffer is cleared and a new record of tape input is brought into the buffer, or end of data is indicated.

A test is made for end of data, no more data terminate, otherwise bump the record counter.

Check the record count against the number of records to be skipped. If less than or equal, return to the top of the loop and pick up the next record.

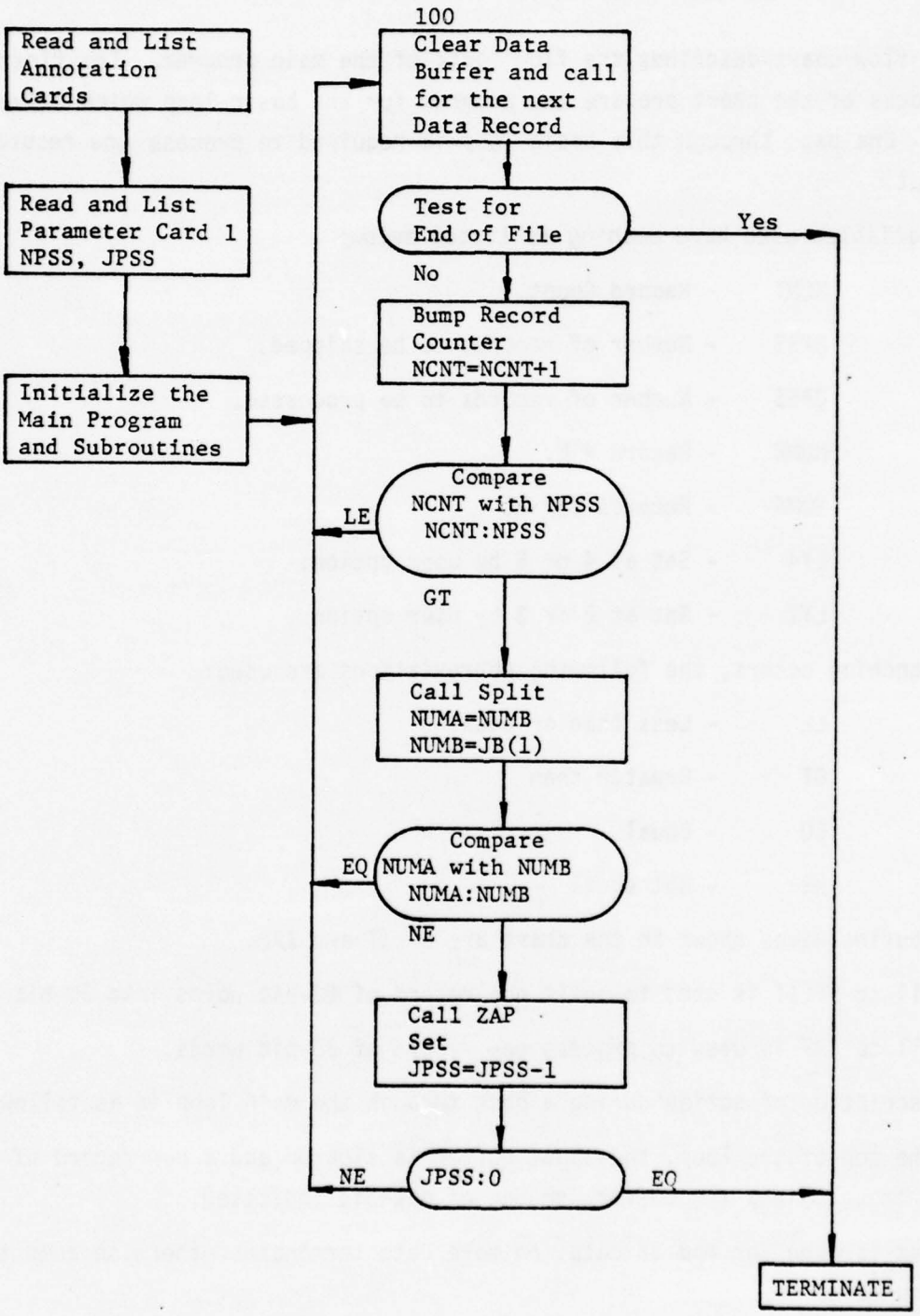


Figure B1. Flow Chart 1 - JASPER.

Otherwise, call SPLIT, save the current record number in NUMA and pick up the next record number from the record just brought in and store in NUMB. If this record has the same number as the preceding record, an error in tape recording is indicated. Return to the top of the loop and pick up the next record. Otherwise a record of 30-bit words is ready for processing.

Processing is initiated with a call to ZAP. Upon return from ZAP, JPSS is decremented by 1. If JPSS records have been processed, JPSS will have a value of zero and the run terminates. Otherwise a return is made to the top of the loop where the next record is brought in and the whole sequence is repeated.

APPENDIX C

GUIDE FOR READING FLOW CHART 2

The first two blocks of the chart are used only during program initialization.

The third block shown is at entry ZAP. In this block the basic loop is initialized. One pass through the loop is required to process one message from a given record. Passes are continued until all messages of the current record have been processed.

The variables used have meaning as listed below:

MC - Message code

MCX - Message code of the preceding message

JP - Index in buffer JP which indicates the first word of a message

NUM - Record number (always the first record word)

Buffer JP - Used for one record of 30-bit words

Buffer JX - Used to store one message

Buffer JMC - Used to store message word counts.

Where branching occurs, the following abbreviations are used:

EQ - Equal

LT - Less than

GT - Greater than

Note from the chart that only message type 1, 4, 2 and 6 are processed.

The use of LY4 and LY2 permits the user to select data from either subsystem 1 (LY2=2 / LY4=4) or subsystem 2 (LY2=3 / LY4=5).

Two special cases have to be accommodated. Message type 1 is followed by a variable number of reply words which must be identified and skipped.

Message type 15 has a variable word count which is extracted directly from the data.

All other message word counts are indexed by MC from the buffer, JMC.

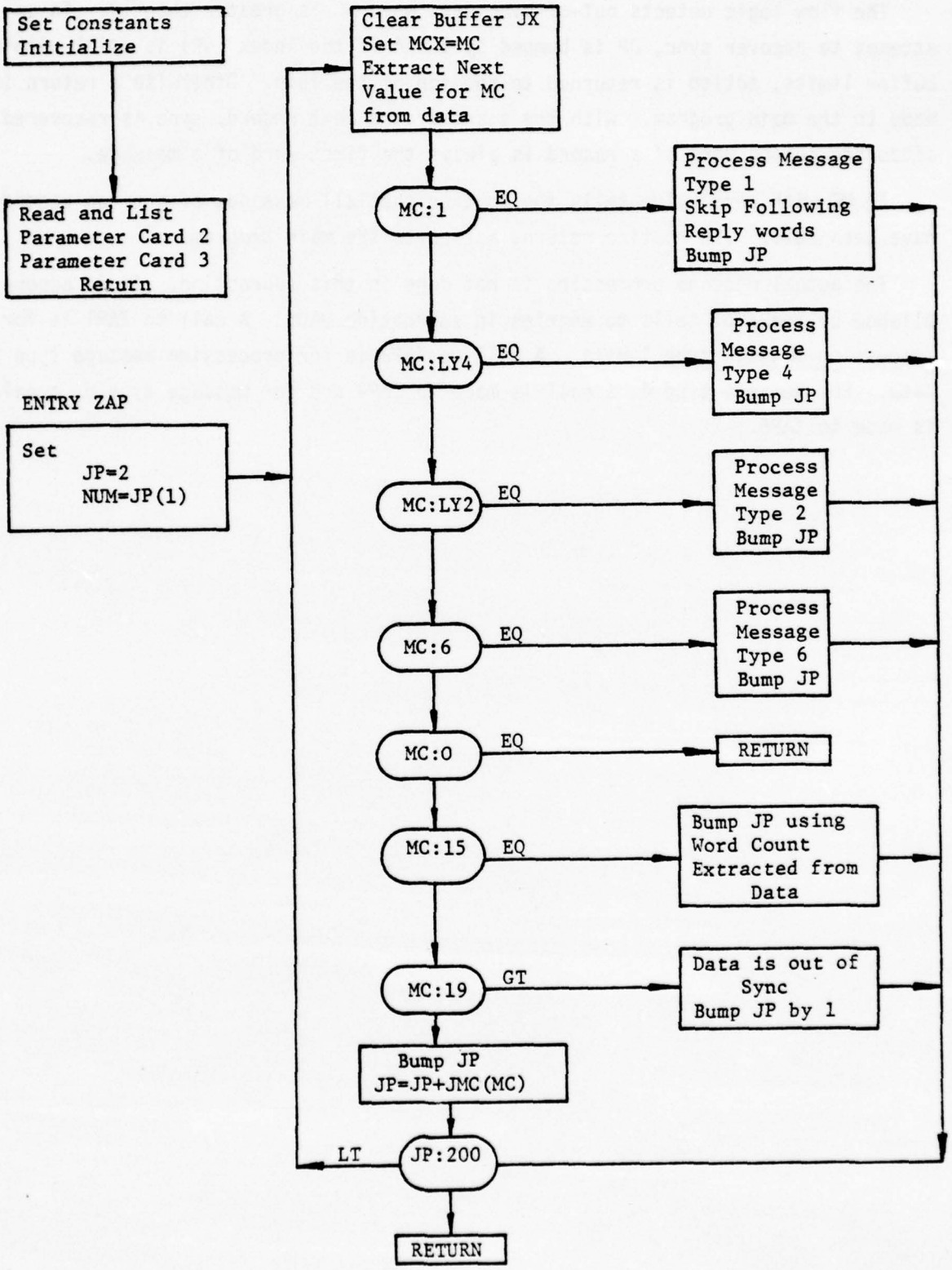


Figure C1. Flow Chart 2 - Subroutine GEN.

The flow logic detects out-of-sync data when MC is greater than 19. In an attempt to recover sync, JP is bumped by 1 and if the index (JP) is still within buffer limits, action is returned to the top of the loop. Otherwise a return is made to the main program. With the start of the next record, sync is recovered since the second word of a record is always the first word of a message.

An MC with zero value tells the routine that all messages of a given record have been read. The routine returns action to the main program.

The actual message processing is not done in this subroutine. It is accomplished by means of calls to entries in subroutine JADE. A call to ZAP1 is for processing message type 1 data. A call to ZAP2 is for processing message type 2 data. For message type 4, a call is made to ZAP4 and for message type 6, a call is made to ZAP6.

APPENDIX D

GUIDE FOR READING FLOW CHART 3

The first two blocks are used only during program initialization. This is done with a call to JADE from the main program. Other calls are all made from Subroutine GEN. They are used for processing a message of a given type. The number used as the last character of the entry name indicates the message type. For instance, Entry ZAP2, is used for processing message type 2 data. The variables used have meaning as listed below:

MBCD	- This is a beacon code supplied by the user
JABCD	- Assigned beacon code
JRBCD	- Reported beacon code

The abbreviation EQ for equal is used in some of the flow branching logic, also GT for greater than, LT for less than.

Entry ZAP1 processes message type 1 data which includes an azimuth word. This azimuth reading is used to update the scan count.

Entry ZAP2 requires no branching. The latest 100 target reports are maintained in storage using end around indexing. The current target report along with the scan number is passed on to REFLECT. This routine examines the current report and preceding reports for a "questionable data group".

Entry ZAP4 processes the sector time word and updates a clock register. The current time is thus available for listing with a "questionable data group".

Entry ZAP6 processes the track store data and selectively stores the data. If the user sets MBCD to 7777 the data are stored regardless of the beacon code number, otherwise the data are stored only when MBCD can be matched with either the assigned beacon code number or the reported beacon code number.

The data stored provide a listing of the latest selected track store data with the listing of "questionable data group".

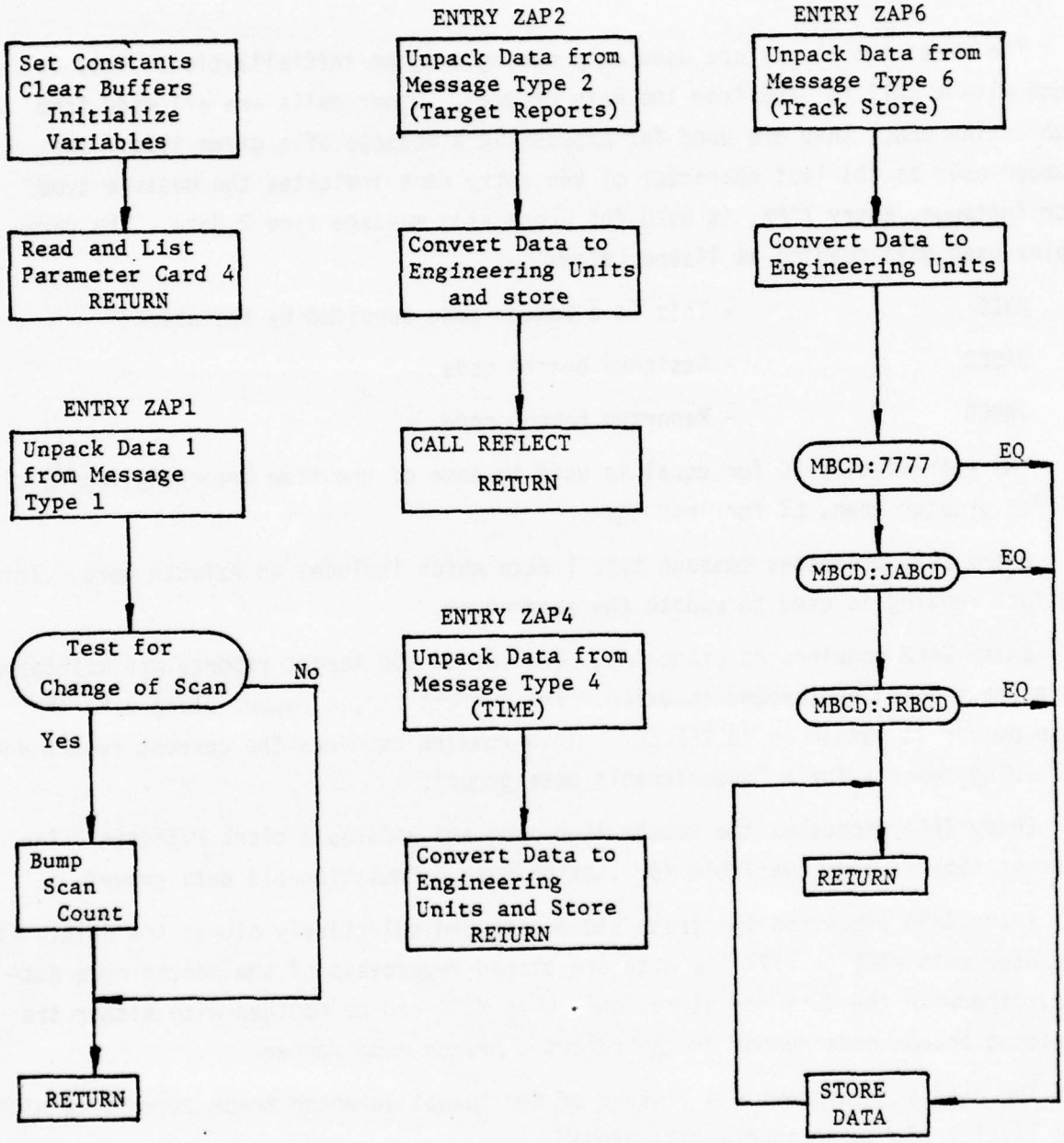


Figure D1. Flow Chart 3 - Subroutine JADE.

APPENDIX E

JASPER PROGRAM LISTING

The following is a listing of the complete JASPER I Program which consists of the MAIN program and the six subroutines. The listing is given in the same order as the list of subroutines in table 2. For a short description of the major subroutines, see Section IV. Also the COMMON BLOCK variables are explained in detail in Appendix A as well as guides to FLOW CHARTS for main program and subroutine GEN and JADE are in Appendixes B, C, and D.

```
PROGRAM JASPER     74/74   OPT=1                                FTN 4.2+74261     01/
                      PROGRAM JASPER(OUTPUT,TAPE1,INPUT)
                      COMMON/BLK1/ JA(101),JB(201),LEN,LAST,MC,JX(50),JP,KNT,KX
                      COMMON/BLK2/ SCNDS,LHR,LMN,FSCS,JAFC(5),ASEC,KAFC(4),FAFC(8)
                      DIMENSION ID(8)
5           900       FORMAT(/10X*WRD COUNT IN REC NUM*2XI5,2X*IS NOT 100, BUT*2XI3/)
            901       FORMAT(1H1)
            902       FORMAT(8A10)
            903       FORMAT( /10X8A10)
            904       FORMAT(//)
10          905       FORMAT(1H1/5X*CARD INPUT FOR PROGRAM JASPER*//
                     15X*THE FIRST CARDS READ ARE FOR COMMENTS OR DOCUMENTATION*/
                     25X*TO BE LISTED ON PRINTER OUTPUT.  UP TO NINE CARDS MAY BE*/
                     35X*USED. THE LAST CARD MUST HAVE AN X IN COLUMN 1.*//
                     45X*THE COMMENT CARDS ARE FOLLOWED BY FOUR PARAMETER CARDS.*//
15                   55X*  CARD 1   FORMAT  3I5 NPSS,JPSS,LRCRD*/
                     15X*  LRCRD IS SET AT 1 FOR READING THE ORIGINAL FAA TAPE*/
                     15X*  LRCRD IS SET AT 5 FOR READING A REFORMATED FAA TAPE*/
                     35X*  THE REFORMATED TAPE IS STACKED INTO 500 WRD RCRDS*/
                     45X*  PROGRAM RECORD COUNT ALWAYS REFERS TO 100 WRD RCRDS*//
20                   65X*  THE PROGRAM SKIPS NPSS RECORDS AND THEN PROCESSES JPSS*/
                     75X*  RECORDS FROM THE DATA INPUT TAPE.*//)
            9052     FORMAT(5X*  CARD 2  FORMAT  19I3*/
                     95X*  THIS CARD CONTAINS THE MESSAGE WORD COUNT FOR EACH*/
                     X5X*  OF THE MESSAGE CODES, 1 THRU 14 AND 16 THRU 19. USE*/
25                   15X*  99 IN FIELD NUMBER 15.*//
                     25X*  CARD 3   FORMAT  2I5*/
                     35X*  THESE TWO WORDS ARE REQUIRED SINCE SOME TAPES COMBINE*/
                     45X*  DATA FROM TWO CONTROL TOWERS UNDER SUBSYSTEM ONE AND*/
                     55X*  SUBSYSTEM TWO.  ENTER 2 AND 4  FOR SUBSYSTEM ONE.*/
30                   75X*  ENTER 3 AND 5 FOR SUBSYSTEM TWO.*//)
            9051     FORMAT(5X*  CARD 4  FORMAT 2I5*/
                     15X*  WORD 1 IS A FOUR DIGIT BECON CODE.  THE LATEST*/
                     25X*  TRACK STORE INFORMATION ON THAT CODE WILL BE*/
                     35X*  INCLUDED WITH THE QUESTIONABLE GROUP PRINT OUT.*/
35                   45X*  WORD 2 IS A MAXIMUM TARGET ALTITUDE IN FEET.*//)
            906      FORMAT(5I5)
            907      FORMAT(//5X*NPSS EQUALS*I10,5X*JPSS EQUALS*I10/
                     15X*LRCD SET AT*I10/
                     25X*LRCD USED FOR INPUT RECORD SIZE OPTION*/
40                   35X*LCRD SET AT 1 FOR 100 WRD RCDS*/
                     45X*LRCD SET AT 5 FOR 500 WRD RCDS*)
                      PRINT905
                      PRINT9052
                      PRINT9051
45                    NTST=1044
                      NCNT=0
                      PRINT901
                C  READ AND LIST ANNOTATION CARDS
                      DO 10 I=1,10
50                    READ902,ID  $ PRINT903,ID
                      IF(ID(1).EQ.NTST)GO TO 20
                   10 CONTINUE
                   20 CONTINUE
                      PRINT904
55              C  READ AND LIST PARAMETER CARD 1
                      READ906,NPSS,JPSS,LRCD
                      PRINT907,NPSS,JPSS,LRCD
```

```
      C  INITIALIZE THE MAIN PROGRAM AND SUBROUTINES
            NUMB=-1
            CALL SET
            CALL GEN
            CALL JADE
            CALL RUBY
            CALL JADER
            IF(LRCD.LT.5)GO TO 100
            CALL FLID
      100   CONTINUE
      C  CLEAR AND LOAD INPUT BUFFER
            CALL ZERO
            IF(LRCD.LT.5)GO TO 103
            CALL LOADX
            GO TO 104
      103   CONTINUE
            CALL LOAD
      104   CONTINUE
      C  TEST FOR END OF FILE *** YES,TERMINATE *** NO,CONTINUE
            IF(LAST.GT.0)GO TO 800
      C  BUMP RECORD COUNTER
            NCNT=NCNT+1
      C  TEST FOR NPSS RECORDS SKIPPED ** NO,GET NEXT RECORD *** YES,CONTINUE
            IF(NCNT.LT.NPSS)GO TO 100
      C  SUB SPLIT CONVERTS 60-BIT WORDS FROM JA TO 30-BIT WORDS STORED AT JB
            CALL S2.IT
      C  NUMB IS RECORD NUM N *** NUMA IS RECORD NUM N-1
            NUMA=NUMB
            NUMB=JB(1)
      C  TEST FOR NUMB EQ NUMA  ***  YES,GET NEXT RECORD *** NO,CONTINUE
      101   IF(NUMA.EQ.NUMB)GO TO 702
            IF(LRCD.EQ.5)GO TO 102
      C  TEST FOR SHORT RECORD
            IF(LEN.NE.100)GO TO 701
      102   CONTINUE
      C  SUB ZAP PROCESSES A RECORD OF DATA
            CALL ZAP
            IF(LAST.EQ.2)GO TO 800
      C  TEST FOR JPSS RECORDS PROCESSED *** YES,TERMINATE *** NO,CONTINUE
            JPSS=JPSS-1 & IF(JPSS.GT.0)GO TO 100
            GO TO 800
      701   CONTINUE
      C  PRINT RECORD NUM AND LENGTH ON SHORT RECORDS
            PRINT911,NUMB,LEN
            GO TO 102
      702   CONTINUE
            GO TO 100
      800   CONTINUE
            END
```

```
      SUBROUTINE SET
      COMMON/BLK1/ JA(100),JB(200),LEN,LAST,MC,JA(50),JP,KNT,KX
      DIMENSION JAX(500)
  903 FORMAT(//10X*READ END OF FILE*I10//)
  904 FORMAT(//10X*PARITY ERROR RECORD NUM*I10,5X*RECORD LENGTH*I10//)
  905 FORMAT(//5X*DUMP OF JA*2I10/(5022))
  906 FORMAT(//5X*DUMP OF JB*2I10/(10(2X010)))
  951 FORMAT(///)
  951 FORMAT(//20X*FILE ID RECORD*/)
  952 FORMAT(/5X,A10)
  953 FORMAT(//10X*READ A DOUBLE END OF FILE ON TAPE1*//)
  954 FORMAT(//10X*PARITY ERROR ON FILE ID RECORD*//)
      MSK=77777777773
      KNT=0
      KNTX=0
      JACK=600
      RETURN
C  FLID READS AND LISTS THE REFORMATED TP ID RORD
C  NOT CALLED WHEN PROCESSING AN ORIGINAL FAA TP
      ENTRY FLID
      BUFFER IN(1,1)(JA(1),JA(100))
      IF(UNIT(1))252,253,254
  253 CONTINUE
      LAST=2 $ PRINT953 $ GO TO 258
  254 CONTINUE
      PRINT954
  252 CONTINUE
      LEN=LENGTH(1)
      PRINT951
      PRINT952,(JA(I),I=1,LEN)
      PRINT950
  258 CONTINUE
      RETURN
C  ZERO CLEARS DATA INPUT BUFFERS
      ENTRY ZERO
      DO 101 I=1,100
      JA(I)=JB(I)=0
  101 CONTINUE
      DO 102 I=101,200
      JB(I)=0
  102 CONTINUE
      DO 103 I=1,50
      JX(I)=0
  103 CONTINUE
      RETURN
C  LOAD IS USED TO READ AN ORIGINAL FAA TP RORD
      ENTRY LOAD
      KNT=KNT+1
      LAST=0
      BUFFER IN(1,1)(JA(1),JA(100))
      IF(UNIT(1))202,203,204
  203 CONTINUE
      LAST=1
      PRINT903,KNT $ GO TO 803
  204 CONTINUE
      LEN=LENGTH(1)
      PRINT904,KNT,LEN
```

```
      GO TO 303
  212 CONTINUE
      LEN=LENGTH(1)
  303 CONTINUE
      RETURN
C  LOADX IS USED TO READ A DATA WORD FROM THE REFORMATED TP
      ENTRY LOADX
      JACK=JACK+100
      IF(JACK.LT.LEN)GO TO 405
      JACK=0
      DO 401 I=1,500
      JAX(I)=1
  401 CONTINUE
      BUFFER IN(1,1)(JAX(1),JAX(500))
      IF(UNIT(1))402,403,404
  403 CONTINUE
      LAST=1 $ PRINT803,KNTX $ GO TO 406
  404 CONTINUE
      LEN=LENGTH(1)
      PRINT804,KNTX,LEN
      GO TO 405
  402 CONTINUE
      KNTX=KNTX+1
      LEN=LENGTH(1)
  405 CONTINUE
      J1=JACK+1 $ J2=JACK+100 $ K=0
      DO 406 J=J1,J2
      K=K+1
      JA(K)=JAX(J)
  406 CONTINUE
      KNT=KNT+1
  408 CONTINUE
      RETURN
C       SPLIT CONVERTS 60-BIT WORDS FROM JA TO 30-BIT WORDS STORED AT JB
      ENTRY SPLIT
      K=0
      DO 301 I=1,100
      JHLD=JA(I)
      DO 301 J=1,2
      K=K+1 $ JJ=J*30
      JB(K)=SHIFT(JHLD,JJ).AND.MSK
  301 CONTINUE
      RETURN
      ENTRY PRNTA
      PRINT805,KNT,LEN,JA
      RETURN
      ENTRY PRNTB
      PRINT806,KNT,LEN,JB
      RETURN
      END
```

```
      SUBROUTINE GEN
      COMMON/BLK1/ JA(100),JB(200),LEN,LAST,MC,JX(50),JP,KNT,KX
      COMMON/BLK2/ SCODE,LHF,LIN,FSCS,JAFC(5),ASEC,KAFC(4),FAFC(3)
      COMMON/BLK5/ JMC(19)
315   FORMAT(//10X*FLIGHT PLANS*18X
     1*PRECEDING MSG CD/WRD CNT *I2*/*I2,5X
     2*CURRENT MSG CD/WRD CNT *I2*/*I2,1X*IN*I4,1X*RN*I5,1X*RC*I5)
302   FORMAT(/2X+I10/(10(2X010)))
304   FORMAT(19I3)
305   FORMAT(//5X*MSG CODE*7X,4X*1*4X*2*4X*3*4X*4*4X*5*4X*6*
     14X*7*4X*8*4X*9*3X*10*3X*11*3X*12*3X*13*3X*14*
     23X*15*3X*16*3X*17*3X*18*3X*19*/
     35X*MSG WRD COUNT*2X,19I6//)
306   FORMAT(2I5)
307   FORMAT(//5X*SELECT SYSTEM ONE OR TWO*2I5)
370   FORMAT(/2X2I10/(10(2X010)))
351   FORMAT(2X*MCX,MC,JP,NUM,LEN*2X5I10 )
C  SET CONSTANTS *** INITIALIZE MC
      MSK1=377
      MSK2=777773
      MSK3=4000000000003
      MC=99
C  READ AND LIST PARAMETER CARD 2,CARD 3
      READ314,JMC
      PRINT315,JMC
      READ306,LY2,LY4
      PRINT307,LY2,LY4
      RETURN
C  ZAP PROCESSES ONE RECORD OF DATA.  A RECORD OF DATA MAY CONTAIN
C  SEVERAL MESSAGES.  THE MESSAGE TYPE IS IDENTIFIED BY ITS
C  MESSAGE CODE(MC) FOUND IN THE RIGHT-MOST 5 BITS OF THE
C  FIRST MESSAGE WORD
      ENTRY ZAP
C  JP INDEXES THE FIRST WORD OF THE CURRENT MESSAGE
      JP=2 $ NUM=JB(1)
100   CONTINUE
C  CLEAR THE MESSAGE BUF JX
      DO 750 I=1,50
      JX(I)=0
750   CONTINUE
C  JM IS USED AS A BRANCHING INDEX AT 700
C  MC IS MESSAGE CODE N *** MCX IS MESSAGE CODE N-1
      JM=1
      MCX=MC
      MC=JB(JP).AND.MSK1
C  BRANCHING LOGIC INDEXED BY MC
      IF(MC.EQ.1)GO TO 1
      IF(MC.EQ.LY4)GO TO 4
      IF(MC.EQ.LY2)GO TO 2
      IF(MC.EQ.6)GO TO 6
      IF(MC.EQ.0)GO TO 900
      IF(MC.EQ.16)GO TO 16
      IF(MC.GT.19)GO TO 80
C  BUMP JP BY THE MESSAGE WORD COUNT FOUND IN JMC
70    CONTINUE
      JP=JP+JMC(MC) $ IF(JP.GE.200)GO TO 800
      GO TO 100
```

```
C  PROCESS MESSAGE TYPE 1 AND ANY FOLLOWING REPLY WORDS
C  REPLY WORDS ARE IDENTIFIED BY BIT 23 EQUAL 1
1     CONTINUE
      L=JMC(MC)
      K=KFLG=1
300   CONTINUE
      IF(JP.GT.200)GO TO 800
      JRPLY=JB(JP).AND.MSK3
      IF(JRPLY.GT.0)GO TO 301
      IF(KFLG.GT.0)GO TO 302
      CALL ZAP1
C     IF(K.LT.2)GO TO 700
C     KX=K $ CALL ZAPR
      GO TO 700
301   CONTINUE
      K=K+1 $ IF(K.GT.50)GO TO 701
302   CONTINUE
      JX(K)=JB(JP)
      KFLG=0
      JP=JP+L
      GO TO 300
C  PROCESS MESSAGE TYPE 2 (MC MAY BE 2 OR 3 BY USER OPTION)
2     CONTINUE
      L2=JMC(MC) $ L1=L2-1
      JL=JP+L1 $ IF(JP.GT.200)GO TO 800
      JRPLY=JB(JP).AND.MSK3
      IF(JRPLY.GT.0)GO TO 60
      K=0
      DO 202 J=JP,JL
      K=K+1
      JX(K)=JB(J)
202   CONTINUE
      CALL ZAP2
      JP=JP+L2 $ IF(JP.GT.200)JM=2
      GO TO 700
C  PROCESS MESSAGE TYPE 4 (MC MAY BE 4 OR 5 BY USER OPTION)
4     CONTINUE
      JX(1)=JB(JP) $ CALL ZAP4
      GO TO 70
C  PROCESS MESSAGE TYPE 6 (TRACK STORE DATA)
6     CONTINUE
      L2=JMC(MC) $ L1=L2-1
      JL=JP+L1 $ IF(JL.GT.200)GO TO 800
      K=0
      DO 206 J=JP,JL
      K=K+1
      JX(K)=JB(J)
206   CONTINUE
      CALL ZAP6
      GO TO 70
C  PROCESS MESSAGE TYPE 15, MESSAGE WORD COUNT(JJ) IS EXTRACTED FROM DATA
15    CONTINUE
      JJ=SHIFT(JB(JP),45).AND.MSK2
      IF(JJ.EQ.0)JJ=1
      IF(JJ.GT.50)JJ=50
      PRINT815,MCX,JMC(MCX),MC,JJ
      JL=JP+JJ-1 $ IF(JL.GT.200)GO TO 800
```

```
            K=0
            DO 215 J=JP,JL
            K=K+1
            JX(K)=JB(J)
      215   CONTINUE
            JP=JP+JJ $ IF(JP.GT.200)JM=2
            GO TO 700
C  ENTER HERE IF MESSAGE TYPE 1 WITH REPLY WORDS EXCEEDS BUF JX CAPACITY
      701   CONTINUE
            JM=3 $ K=1
            CALL ZAP1
            JX(1)=0
C  BRANCHING CONTROLLED BY JM
C  TRANSFER TO 100 FOR THE NEXT MESSAGE
C  TRANSFER TO 800 FOR THE NEXT RECORD
C  TRANSFER TO 301 FOR MORE REPLY WORDS
      700   CONTINUE
            GO TO (100,800,301)JM
      800   CONTINUE
            RETURN
C  MC IS NOT AN ACCEPTED MESSAGE CODE
      60    CONTINUE
            PRINT881,MCX,MC,JP,NUM,LEN
            CALL PRNTB
            JP=JP+1 $ IF(JP.GT.200)GO TO 800
            GO TO 100
            END
```

```
      SUBROUTINE JADE
      COMMON/BLK1/ JA(100),JB(200),LEN,LAST,MC,JX(50),JP,KNT,KX
      COMMON/BLK2/ SCNDS,LHR,LMN,FSCS,JAFC(5),ASEC,KAFC(4),FAFC(8)
      COMMON/BLK3/ NHR(100),NMN(100),FSC(100),FRNG(100),FAZ(100),
     1NALT(100),NCD(100),NST(100),NVC(100),NVA(100),NSP(100),KAP
     2,LAP,NKNT(100)
      COMMON/BLK4/ LTR(100)
      COMMON/BLK6/ LF(13),LG(7),LH(4),NCID(45),JTM(5),JAL(3),JID(2)
      COMMON/BLK7/ NAZB,NSWP
      DATA(NCID=1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HA,1HB,1HC,
     11HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1HO,1HP,
     21HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ,
     31H-,1H*,1H-,1H/,1H*,1H ,1H0,1H0,1H0)
  902 FORMAT(/2X2I10,F8.3,2XA7,2XA3)
  903 FORMAT(//(10(3XO7)))
  904 FORMAT(5I5)
  905 FORMAT(/5X*MONITOR TRACK STOPE ON BEACON CODE*I3/
     15X*MAX ALT*I8/)
  906 FORMAT(2x2I3,F7.3,2F7.2,8I5)
C  READ AND LIST PARAMETER CARD 4
      READ904,MBCD,NALM
      PRINT905,MBCD,NALM
C  SET CONSTANTS,CLEAR BUFFERS,INITIALIZE VARIABLES
      ASEC=0.0 $ JAFC(5)=0
      DO 21 I=1,4
      JAFC(I)=KAFC(I)=0
   21 CONTINUE
      DO 22 I=1,8
      FAFC(I)=0.0
   22 CONTINUE
      DO 100 I=1,100
      NHR(I)=NMN(I)=NALT(I)=NCD(I)=NST(I)=NVC(I)=NVA(I)=NSP(I)=0
      FSC(I)=FRNG(I)=FAZ(I)=0.0
  100 CONTINUE
      NRP=100
      KNZ=0 $ KAP=1
      NAZB=NSWP=0
      NSWP=1
      MBT11=2000B $ MBT8=400B
      MSK1=1B $ MSK2=3B $ MSK3=7B $ MSK4=17B
      MSK6=77B
      MSK8=377B
      MSK10=1777B $ MSK12=7777B
      MSK24=77777777B
      MSK4=7776B
      JPSS=5
      LAP=0
      RETURN
C  UNPACK DATA FROM MESSAGE TYPE 1
C  UPDATE SCAN COUNT IF NEEDED
      ENTRY ZAP1
      IF(JX(1).EQ.0)GO TO 106
      JHLD=SHIFT(JX(1),30)
      DO 101 I=1,13
      JHLD=SHIFT(JHLD,1)
      LF(I)=JHLD.AND.MSK1
  101 CONTINUE
```

```
      JHLD=SHIFT(JHLD,12)
      LACP=JHLD.AND.MSK12
      NAZA=NAZB $ NAZB=LACP
      IF(NAZA.GT.NAZB+1000)NSWP=NSWP+1
105   CONTINUE
      RETURN
C  UNPACK DATA FROM MESSAGE TYPE 2
C  CONVERT DATA TO ENGINEERING UNITS AND STORE
C  CALLS REFLECT
      ENTRY ZAP2
      KNZ=KNZ+1 $ IF(KNZ.LT.101)GO TO 205
      KAP=MOD(KAP,100)+1
205   CONTINUE
      JHLD=SHIFT(JX(1),30)
      JHLD=SHIFT(JHLD,13)
      LACP=JHLD.AND.MSK12
      JHLD=SHIFT(JHLD,1) $ LG(1)=JHLD.AND.MSK1
      JHLD=SHIFT(JHLD,1) $ LG(2)=JHLD.AND.MSK1
      JHLD=SHIFT(JHLD,10) $ LRNG=JHLD.AND.MSK10
      JHLD=SHIFT(JX(2),30)
      JHLD=SHIFT(JHLD,1) $ LG(3)=JHLD.AND.MSK1
      JHLD=SHIFT(JHLD,2) $ LG(4)=JHLD.AND.MSK2
      DO 201 I=1,4
      JHLD=SHIFT(JHLD,3) $ LH(I)=JHLD.AND.MSK3
201   CONTINUE
      LCODE=LH(1)*1000+LH(2)*100+LH(3)*10+LH(4)
      JHLD=SHIFT(JHLD,1) $ LG(5)=JHLD.AND.MSK1
      JHLD=SHIFT(JHLD,2) $ LG(6)=JHLD.AND.MSK2
      DO 202 I=1,3
      JHLD=SHIFT(JHLD,4) $ LH(I)=JHLD.AND.MSK4
202   CONTINUE
      LALT=LH(1)*100+LH(2)*10+LH(3)
      ACP=LACP $ AZIM=ACP*(360.0/4096.0)
      RNG=LRNG $ RANG=RNG*(1.0/16.0)
      NRP=MOD(NRP,100)+1
      NHR(NRP)=LHR $ NMN(NRP)=LMN $ FSC(NRP)=FSCS
      RNG(NRP)=RANG $ AZ(NRP)=AZIM
      NALT(NRP)=LALT $ NCO(NRP)=LCODE
      NST(NRP)=LG(5) $ NVC(NRP)=LG(6)
      NVA(NRP)=LG(4) $ NSP(NRP)=NSWP
      NKNT(NRP)=KNT
      IF(LAP.LT.1)GO TO 203
      PRINT906,LHR,LMN,FSCS,RANG,AZIM,LALT,LCODE,
     1LG(5),LG(6),LG(4),NSWP,JPSS,KNT
      IF(NSWP.NE.KSWP)LAP=0
203   CONTINUE
      KSWP=NSWP
      LSWP=NSWP
      IF(LALT.GT.NALM)GO TO 802
      CALL REFLECT(AZIM,RANG,LCODE,LSWP)
802   CONTINUE
      RETURN
C  UNPACK DATA FROM MESSAGE TYPE 4
C  CONVERT DATA TO ENGINEERING UNITS AND STORE
      ENTRY ZAP4
      JHLD=SHIFT(JX(1),30)
      JHLD=SHIFT(JHLD,25)
```

```
      JHLD=JHLD.AND.MSK24 $ HLD=JHLD $ SCNDS=HLD/128.0
      LSCS=SCNDS $ LHR=LSCS/3600
      LMN=(LSCS-3600*LHR)/60
      FSCS=SCNDS-3600*LHR-60*LMN
      RETURN
C  UNPACK DATA FROM MESSAGE TYPE 6
C  CONVERT TO ENGINEERING UNITS
C  SELECTIVELY STORE (USER CONTROLLED)
      ENTRY ZAP6
      SEC=JX(2)*(1.0/1024.0)
      KSEC=SEC
      JHR=KSEC/3600
      JMN=KSEC/60-JHR*60
      JSEC=KSEC-JHR*3600-JMN*60
      SEC=SEC-JHR*3600.0-JMN*60.0
      JHLD=SHIFT(JX(3),30)
      DO 601 I=1,5
      JHLD=SHIFT(JHLD,6)
      L=(JHLD.AND.MSK6)+1 $ IF(L.GT.37)L=37
      JID(I)=NCID(L)
601   CONTINUE
      JHLD=SHIFT(JX(4),30)
      DO 602 I=6,7
      JHLD=SHIFT(JHLD,6)
      L=(JHLD.AND.MSK6)+1 $ IF(L.GT.37)L=37
      JID(I)=NCID(L)
602   CONTINUE
      DO 603 I=1,3
      JHLD=SHIFT(JHLD,6)
      L=(JHLD.AND.MSK6)+1 $ IF(L.GT.31)L=37
      JAL(I)=NCID(L)
603   CONTINUE
      ENCODE(7,916,IDAC)(JID(I),I=1,7)
916   FORMAT(7A1)
      ENCODE(3,926,NAL)(JAL(I),I=1,3)
926   FORMAT(3A1)
C  TRSPLT COMPLETES THE UNPACKING OF THIS MESSAGE STORING RESULTS
C  IN COMMON/BLK4/ LTR(100)
      CALL TRSPLT
C CTST
      IUT=LTR(6) $ IA=LTR(7) $ ITP=LTR(8)
C  RGAZT        RANGE AND AZIMUTH
      AZM=LTR(11)*(360.0/4096.0)
      KIC=LTR(12) $ KC=LTR(13)
      RNG=LTR(14)*(1.0/16.0)
      KFIRM=LTR(15)
C  CORDT       PREDICTED TARGET COORDINATES
      JYP=LTR(16) $ JSGN=JYP.AND.1BT11
      IF(JSGN.GT.0)JYP=JYP-2047
      FYP=JYP*(1.0/16.0)
      JXP=LTR(18) $ JSGN=JXP.AND.1BT11
      IF(JSGN.GT.0)JXP=JXP-2047
      FXP=JXP*(1.0/16.0)
      NTCNT=LTR(17) $ JTC=LTR(20) $ JPP=LTR(21)
C  XYDOT        X COMPONENT VEL AND Y COMPONENT VEL
      JYD=LTR(22) $ JSGN=JYD.AND.1BT9
      IF(JSGN.GT.0)JYD=JYD-511
```

```
      FYD=JYD*(450.0/128.0)
      JXD=LTR(25) $ JSGN=JXD.AND.M8T9
      IF(JSGN.GT.0)JXD=JXD-511
      FXD=JXD*(450.0/128.0)
      FYD=FYD/3600.0 $ FXD=FXD/3600.0
      TST=ABS(FYD)+ABS(FXD)
      IF(TST.GT.0.01)GO TO 691
      HDGN=999.0 $ VEL=0.0 $ GO TO 692
  691 CONTINUE
      HDGN=90.0-ATAN2(FYD,FXD)/0.017453
      VEL=SQRT(FYD**2+FXD**2)
  692 CONTINUE
      LTR23=LTR(23) $ LTR24=LTR(24) $ LTR26=LTR(26)
C  TIMET      TIME OF LAST CORRELATION IF UT=01
      MSGNUM=LTR(27)
      CRLTM=0.0
      IF(IUT.EQ.1)CRLTM=LTR(28)*(1.0/1024.0)
C  TRACT      NEW INSERTED WORD
      NTRCT=LTR(33)
C  CELGI      (PRINT WITH FORMAT(21I5),LTR(I),I=34,54)
C  ABEAT      ASSIGNED BEACON CODE
      JNT=LTR(29)
      JHLD=SHIFT(LTR(30),48)
      DO 604 I=1,4
      JHLD=SHIFT(JHLD,3)
      LH(I)=JHLD.AND.MSK3
  604 CONTINUE
      JABCD=LH(1)*1000+LH(2)*100+LH(3)*10+LH(4)
      LTR31=LTR(31) $ LTR32=LTR(32)
C  LGCI       LAST GOOD CODE WORD
      JLT=LTR(55) $ LPS=LTR(57)
      ICPTRIC=LTR(58)
C  TEST FOR TABULAR COAST (UT=01,A=0,TP=01)
      JTCC=0 $ LGC=0 $ LGCX=0
      IF(IUT.EQ.1.AND.IA.EQ.0.AND.ITP.EQ.1)GO TO 606
      IF(JLT.EQ.2 )GO TO 607
      IF(JLT.EQ.3 )GO TO 607
      JHLD=SHIFT(LTR(56),+8)
      DO 605 I=1,4
      JHLD=SHIFT(JHLD,3)
      LH(I)=JHLD.AND.MSK3
  605 CONTINUE
      JLGC=LH(1)*1000+LH(2)*100+LH(3)*10+LH(4)
      IF(JLT.EQ.0)LGC=JLGC
      IF(JLT.EQ.1)LGCX=JLGC
      GO TO 607
  606 CONTINUE
      JTCC=LTR(56)
  607 CONTINUE
C  RBEAT      REPORTED BEACON CODE
      LTR59=LTR(59) $ LTR60=LTR(60) $ LTR61=LTR(61)
      LTR62=LTR(62)
      JHLD=SHIFT(LTR(62),48)
      DO 608 I=1,4
      JHLD=SHIFT(JHLD,3)
      LH(I)=JHLD.AND.MSK3
  608 CONTINUE
```

```
      JRBCD=LH(1)*1000+LH(2)*100+LH(3)*10+LH(4)
      IF(MBCD.EQ.ZZZZ)GO TO 651
      IF(MBCD.EQ.JABCD)GO TO 651
      IF(MBCD.EQ.JRBCD)GO TO 651
  651 CONTINUE
      JUMP=1 $ IF(JUMP.EQ.1)GO TO 806
      PRINT903,(LTR(I),I=1,70)
      PRINT912,JHR,JMN,SEC,IDAC,NAL
      PRINT931,IUT,IA,ITP
  931 FORMAT(/2X*CTST WRD UT,A,TP,*2X3I5)
      PRINT932,AZM,RNG,KIC,KC,KFIRM
  932 FORMAT(/2X*RGAZT WRD AZ,RNG,IC,C,FIRM*2X2F10.2,3I5)
      PRINT933,FXP,FYP,NTCNT,JTC,JPP
  933 FORMAT(/2X*CORDT WRD XP,YP,TC1/TC2,TC,P*2X2F10.2,3I8)
      PRINT934,FXD,FYD,VEL,HDGN,LTR23,LTR24,LTR26
  934 FORMAT(/2X*XYDOT WRD XVEL,YVEL,VEL,HDGN*2X4F10.3,3I3)
      PRINT935,CRLTM,MSGNUM
  935 FORMAT(/2X*TIMET WRD CORTIME,MSGNUM*2XF10.3,I8)
      PRINT936,NTPCT
  936 FORMAT(/2X*IRACT(NEW WRD)*2X010)
      PRINT937,(LTR(I),I=34,54)
  937 FORMAT(/2X*CFLGT WRD*/2X21I5)
      PRINT938,JABCD,JNT,LTR31,LTR32
  938 FORMAT(/2X*ABEAT WRD ABCD,NT,TT1,TT2*2X4I5)
      PRINT939,LGC,LGCX,JTCC,JLT,LPS,IPTRID
  939 FORMAT(/2X*LGCT WRD LGC,TLGC,IBCSIC,LT,PS,CID*2X5I5,2X05)
      PRINT94),JRBCD,LTR59,LTR60,LTR61,LTR63
  941 FORMAT(/2X*RBEAT WRD RBCD,R,CF,Z,TRKNUM*2X4I5,2X05)
      PRINT941,LTR(64),LTR(65),LTR(66)
  941 FORMAT(/2X*CTSST AND OBIT WRDS*2X2(2X07),2X010)
  806 CONTINUE
      RETURN
  650 CONTINUE
      JAFC(1)=IUT $ JAFC(2)=IA
      JAFC(3)=ITP $ JAFC(4)=JHR
      JAFC(5)=JMN $ ASEC=SEC
      KAFC(1)=IDAC $ KAFC(2)=NAL
      KAFC(3)=JABCD $ KAFC(4)=JRBCD
      FAFC(1)=AZM $ FAFC(2)=RNG
      FAFC(3)=FXP $ FAFC(4)=FYP
      FAFC(5)=FXD $ FAFC(6)=FYD
      FAFC(7)=VEL $ FAFC(8)=HDGN
      GO TO 651
      END
```

```
      SUBROUTINE RUBY
      COMMON/BLK1/ JA(100),JB(200),LEN,LAST,MC,JX(50),JP,KIT,XX
      COMMON/BLK4/ LTR(100)
      COMMON/BLK5/ JMC(19)
      DIMENSION JSKP(12)
      DATA(JSKP=12(1))
      IF(JMC(6).EQ.19)JSKP(7)=0
      IF(JMC(6).EQ.17)JSKP(1)=JSKP(7)=JSKP(12)=0
      MSK1=1B $ MSK2=3B $ MSK3=7B $ MSK4=17B
      MSK5=37B $ MSK6=77B $ MSK8=377B $ MSK9=777B
      MSK10=1777B $ MSK11=3777B $ MSK12=7777B $ MSK13=17777B
      MSK14=37777B $ MSK15=77777B $ MSK18=777777B
      RETURN
C  IRSPLT COMPLETES THE UNPACKING OF MESSAGE TYPE 6 DATA FOR ZAPS
C  UNPACKED WORDS ARE STORED IN COMMON/BLK4/ LTR(100)
      ENTRY IRSPLT
      DO 101 I=1,100
      LTR(I)=0
101   CONTINUE
      J=4
      DO 300 I=1,12
      JFLG=JSKP(I)
      IF(JFLG.EQ.0)GO TO 300
      GO TO(301,302,303,304,305,306,307,308,309,310,311,312)I
301   CONTINUE
C  IRK WRD NUM 5 CTST
      K=0
      J=J+1
      JHLD=SHIFT(JX(J),30)
      JHLD=SHIFT(JHLD,2) $ K=K+1
      LTR(K)=JHLD.AND.MSK2
      JHLD=SHIFT(JHLD,4) $ K=K+1
      LTR(K)=JHLD.AND.MSK4
      JHLD=SHIFT(JHLD,5) $ K=K+1
      LTR(K)=JHLD.AND.MSK5
      JHLD=SHIFT(JHLD,3) $ K=K+1
      LTR(K)=JHLD.AND.MSK3
      JHLD=SHIFT(JHLD,1) $ K=K+1
      LTR(K)=JHLD.AND.MSK1
      JHLD=SHIFT(JHLD,2) $ K=K+1
      LTR(K)=JHLD.AND.MSK2
      JHLD=SHIFT(JHLD,1) $ K=K+1
      LTR(K)=JHLD.AND.MSK1
      JHLD=SHIFT(JHLD,2) $ K=K+1
      LTR(K)=JHLD.AND.MSK2
      JHLD=SHIFT(JHLD,5) $ K=K+1
      LTR(K)=JHLD.AND.MSK5
      JHLD=SHIFT(JHLD,5) $ K=K+1
      LTR(K)=JHLD.AND.MSK5
      GO TO 300
302   CONTINUE
C  TRK WRD NUM 6  RGAZT
      K=10
      J=J+1
      JHLD=SHIFT(JX(J),30)
      JHLD=SHIFT(JHLD,13) $ K=K+1
      LTR(K)=JHLD.AND.MSK12
```

```
      JHLD=SHI[illegible]JHLD,1) $ K=K+1
      LTR(K)=JHLD.AND.MSK1
      JHLD=SHIFT(JHLD,1) $ K=K+1
      LTR(K)=JHLD.AND.MSK1
      JHLD=SHIFT(JHLD,10) $ K=K+1
      LTR(K)=JHLD.AND.MSK10
      JHLD=SHIFT(JHLD,5) $ K=K+1
      LTR(K)=JHLD.AND.MSK5
      GO TO 300
  303 CONTINUE
C  TRK WRD NUM 7 CORDT
      K=15
      J=J+1
      JHLD=SHIFT(JX(J),30)
      JHLD=SHIFT(JHLD,11) $ K=K+1
      LTR(K)=JHLD.AND.MSK11
      JHLD=SHIFT(JHLD,4) $ K=K+1
      LTR(K)=JHLD.AND.MSK4
      JHLD=SHIFT(JHLD,11) $ K=K+1
      LTR(K)=JHLD.AND.MSK11
      JHLD=SHIFT(JHLD,2) $ K=K+1
      LTR(K)=JHLD.AND.MSK2
      JHLD=SHIFT(JHLD,1) $ K=K+1
      LTR(K)=JHLD.AND.MSK1
      JHLD=SHIFT(JHLD,1) $ K=K+1
      LTR(K)=JHLD.AND.MSK1
      GO TO 300
  304 CONTINUE
C  TRK WRD NUM 8  XYDOT
      K=21
      J=J+1
      JHLD=SHIFT(JX(J),30)
      JHLD=SHIFT(JHLD,9) $ K=K+1
      LTR(K)=JHLD.AND.MSK9
      JHLD=SHIFT(JHLD,4) $ K=K+1
      LTR(K)=JHLD.AND.MSK4
      JHLD=SHIFT(JHLD,2) $ K=K+1
      LTR(K)=JHLD.AND.MSK2
      JHLD=SHIFT(JHLD,9) $ K=K+1
      LTR(K)=JHLD.AND.MSK9
      JHLD=SHIFT(JHLD,6) $ K=K+1
      LTR(K)=JHLD.AND.MSK6
      GO TO 300
  305 CONTINUE
C  TRK WRD NUM 9 TIMET
      K=26
      J=J+1
      JHLD=SHIFT(JX(J),30)
      JHLD=SHIFT(JHLD,12) $ K=K+1
      LTR(K)=JHLD.AND.MSK12
      JHLD=SHIFT(JHLD,18) $ K=K+1
      LTR(K)=JHLD.AND.MSK18
      GO TO 300
  306 CONTINUE
C  TRK WRD NUM 10 ADBEAT
      K=28
      J=J+1
```

```
            JHLD=SHIFT(JX(J),30)
            JHLD=SHIFT(JHLD,2) $ K=K+1
            LTR(K)=JHLD.AND.MSK2
            JHLD=SHIFT(JHLD,12) $ K=K+1
            LTR(K)=JHLD.AND.MSK12
            JHLD=SHIFT(JHLD,8) $ K=K+1
            LTR(K)=JHLD.AND.MSK8
            JHLD=SHIFT(JHLD,8) $ K=K+1
            LTR(K)=JHLD.AND.MSK8
            GO TO 300
  307       CONTINUE
C   TRK WRD NUM 10.1 TRACT
            K=32
            J=J+1
            K=K+1
            JHLD=SHIFT(JX(J),30)
            LTR(K)=JX(J)
            GO TO 300
  308       CONTINUE
C   TRK WRD NUM 11 CELGT
            K=33
            J=J+1
            JHLD=SHIFT(JX(J),30)
            DO 604 MI=1,4
            JHLD=SHIFT(JHLD,1) $ K=K+1
            LTR(K)=JHLD.AND.MSK1
  604       CONTINUE
            JHLD=SHIFT(JHLD,3) $ K=K+1
            LTR(K)=JHLD.AND.MSK3
            JHLD=SHIFT(JHLD,4) $ K=K+1
            LTR(K)=JHLD.AND.MSK4
            JHLD=SHIFT(JHLD,4) $ K=K+1
            LTR(K)=JHLD.AND.MSK4
            DO 605 MI=1,13
            JHLD=SHIFT(JHLD,1) $ K=K+1
            LTR(K)=JHLD.AND.MSK1
  605       CONTINUE
            JHLD=SHIFT(JHLD,2) $ K=K+1
            LTR(K)=JHLD.AND.MSK2
            GO TO 300
  309       CONTINUE
C   TRK WRD NUM 12 LGCT
            K=54
            J=J+1
            JHLD=SHIFT(JX(J),30)
            JHLD=SHIFT(JHLD,2) $ K=K+1
            LTR(K)=JHLD.AND.MSK2
            JHLD=SHIFT(JHLD,13) $ K=K+1
            LTR(K)=JHLD.AND.MSK13
            JHLD=SHIFT(JHLD,1) $ K=K+1
            LTR(K)=JHLD.AND.MSK1
            JHLD=SHIFT(JHLD,14) $ K=K+1
            LTR(K)=JHLD.AND.MSK14
            GO TO 300
  310       CONTINUE
C   TRK WRD NUM 13 RBEAT
            K=58
```

```
      J=J+1
      JHLD=SHIFT(JX(J),30)
      JHLD=SHIFT(JHLD,1) $ K=K+1
      LTR(K)=JHLD.AND.MSK1
      JHLD=SHIFT(JHLD,1) $ K=K+1
      LTR(K)=JHLD.AND.MSK1
      JHLD=SHIFT(JHLD,1) $ K=K+1
      LTR(K)=JHLD.AND.MSK1
      JHLD=SHIFT(JHLD,12) $ K=K+1
      LTR(K)=JHLD.AND.MSK12
      JHLD=SHIFT(JHLD,15) $ K=K+1
      LTR(K)=JHLD.AND.MSK15
      GO TO 300
311   CONTINUE
C  TRK WRD NUM  14  CTSST
      K=63
      J=J+1
      JHLD=SHIFT(JHLD,12) $ K=K+1
      LTR(K)=JHLD.AND.MSK12
      JHLD=SHIFT(JHLD,18) $ K=K+1
      LTR(K)=JHLD.AND.MSK18
      GO TO 300
312   CONTINUE
C  TRK WRD NUM  15 DBIT
      K=65
      J=J+1
      JHLD=SHIFT(JX(J),30)
      K=K+1
      LTR(K)=JX(J)
300   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE JADER
      COMMON/BLK1/ JA(100),JB(200),LEN,LAST,MC,JX(50),JP,KNT,KX
      COMMON/BLK6/ LF(13),LG(7),LH(4),NCID(45),JTM(5),JAL(3),JID(7)
      COMMON/BLK7/ NAZB,NSWP
      DIMENSION GRAY1(8),GRAY2(16),GRAY3(16),JRPLY(10)
      DIMENSION NB(12),NBC(16),NF(4)
      INTEGER GRAY1,GRAY2,GRAY3
      DATA
     1(GRAY1=0,-2,0,-1,2,0,1,0),
     2(GRAY2=0,5,15,10,35,30,20,25,75,70,60,65,40,45,55,50),
     3(GRAY3=0,80,240,160,560,480,320,400,1200,1120,960,
     41040,640,720,880,800)
      MSK1=1B $ MSK10=1777B $ MSK12=7777B $ MSK3=7B
  907 FORMAT(//10X*A AND C BOTH 1 OR BOTH 0*//)
  905 FORMAT(2X*A*2X,I4,2X,F7.2,I5,2X,A1,2X,15A1,4X,A1,
     12X,102,1A1,102,2X1A1)
  906 FORMAT(2X*C*1X2F7.2,I5,2X,A1,2X,15A1,-X,A1,
     12X,102,1A1,102,2X1A1)
      MPZ=0 $ MPX=1B $ MP1=1H1 $ MP2=1H- $ MSK6=77B
      RETURN
C  ZAPR PROCESSES REPLY WORDS(NOT CURRENTLY IN USE)
C  TO USE ZAPR SEE COMMENTS FOLLOWING STATEMENT 300 IN SUB GEN
C  WHEN USED ZAPR LOOKS FOR REPLY WRDS IN(JX(I),I=2,KX)
C  UNLESS RESTRICTION IS MADE ON PRINTER OUTPUT,
C  THE AMOUNT OF OUTPUT IS EXCESSIVE
      ENTRY ZAPR
      PRINT900,(JX(I),I=1,KX)
  900 FORMAT(//2X,5O22)
      K=1 $ A=LF(9) $ C=LF(10)
  100 CONTINUE
      K=K+1 $ IF(K.GT.KX)GO TO 900
      JHLD=SHIFT(JX(K),30)
      DO 101 I=1,3
      JHLD=SHIFT(JHLD,1)
      JRPLY(I)=JHLD.AND.MSK1
  101 CONTINUE
      JHLD=SHIFT(JHLD,12)
      JRPLY(4)=JHLD.AND.MSK12
      DO 102 I=5,9
      JHLD=SHIFT(JHLD,1)
      JRPLY(I)=JHLD.AND.MSK1
  102 CONTINUE
      JHLD=SHIFT(JHLD,10)
      JRPLY(11)=JHLD.AND.MSK12
      JHLD=SHIFT(JRPLY(4),48)
      DO 103 I=1,12
      JHLD=SHIFT(JHLD,1)
      J=13-I $ NB(J)=JHLD.AND.MSK1
  103 CONTINUE
      DO 104 I=1,16
      NBC(I)=NCID(37)
  104 CONTINUE
      NBC(1)=NCID(2)
      NBC(15)=NCID(2)
      IF(JRPLY(7).EQ.1)NBC(8)=NCID(34)
      IF(NB(4).EQ.1)NBC(2)=NCID(2)
      IF(NB(11).EQ.1)NBC(3)=NCID(2)
```

```
      IF(NB(5).EQ.1)NBC(4)=NCID(2)
      IF(NB(11).EQ.1)NBC(5)=NCID(2)
      IF(NB(6).EQ.1)NBC(6)=NCID(2)
      IF(NB(12).EQ.1)NBC(7)=NCID(2)
      IF(NB(7).EQ.1)NBC(9)=NCID(2)
      IF(NB(1).EQ.1)NBC(10)=NCID(2)
      IF(NB(8).EQ.1)NBC(11)=NCID(2)
      IF(NB(2).EQ.1)NBC(12)=NCID(2)
      IF(NB(9).EQ.1)NBC(13)=NCID(2)
      IF(NB(3).EQ.1)NBC(14)=NCID(2)
      IF(JRPLY(3).EQ.1)NBC(16)=NCID(2)
      MGARB=NCID(42)
      IF(JRPLY(2).EQ.1)MGARB=NCID(17)
      JAKE1=JAKE2=0
      DO 751 JS=2,7
      KS=JS+7
      JAKE1=SHIFT(JAKE1,1) $ JAKE2=SHIFT(JAKE2,1)
      MAKE1=NBC(JS) $ MAKE2=NBC(KS)
      LAKE1=LAKE2=MPZ
      IF(MAKE1.EQ.MP1)LAKE1=MPX
      IF(MAKE2.EQ.MP1)LAKE2=MPX
      JAKE1=JAKE1.OR.LAKE1 $ JAKE2=JAKE2.OR.LAKE2
  751 CONTINUE
      RANG=JRPLY(10) $ RANG=RANG/16.0
      ALT=0.0
      IF(A.EQ.1)GO TO 620
      IF(C.EQ.0)GO TO 650
C  MODE C      ALT IN GRAY CODE
      NALT=JRPLY(4)
      IF(NALT.EQ.0)GO TO 500
      J=4*NB(3)+2*NB(2)+NB(1)
      TMP1=GRAY1(J+1)
      LSVK=K $ K=0
      DO 520 J=4,11
      IF(NB(J).NE.0)K=K+1
  520 CONTINUE
      TEST=FLOAT(K)/2.0 $ ITEST=K/2 $ FTEST=ITEST
      IF(TEST-FTEST)5210,521,5210
 5210 TMP1=-TMP1
  521 J=8*NB(7)+4*NB(6)+2*NB(5)+NB(4)+1
      TMP2=GRAY2(J)
      K=0
      DO 522 J=8,11
      IF(NB(J).NE.0)K=K+1
  522 CONTINUE
      TEST=FLOAT(K)/2.0 $ ITEST=K/2 $ FTEST=ITEST
      IF(TEST-FTEST)5230,523,5230
 5230 TMP2=75.0-TMP2
  523 J=8*NB(11)+4*NB(10)+2*NB(9)+NB(8)+1
      TMP2=TMP2+GRAY3(J)-10.0
      ALT=TMP2+TMP1
      K=LSVK
  500 CONTINUE
      PRINT906,ALT,RANG,NAZB,MGARB,(NBC(J),J=1,16),
     1JAKE1,NBC(8),JAKE2,NBC(16)
      GO TO 610
  620 CONTINUE
```

```
            IF(C.EQ.1)GO TO 650
      C   MODE A    BCN CODE
            JRACK=JRPLY(4)
            JHLD=SHIFT(JRACK,48)
            DO 761 I=1,4
            JHLD=SHIFT(JHLD,3)
            NF(I)=JHLD.AND.MSK3
      761   CONTINUE
            JRACK=NF(1)*1000+NF(2)*100+NF(3)*10+NF(4)
            PRINT905,JRACK,RANG,NAZB,MGARB,(NBC(J),J=1,16),
           1JAKE1,NBC(8),JAKE2,NBC(16)
            GO TO 600
      650   CONTINUE
            PRINT907
      600   CONTINUE
            GO TO 100
      800   CONTINUE
            RETURN
            END
```

```
      SUBROUTINE REFLECT(AZR,RANGE,CODE,SCAN)
      COMMON/BLK2/ SCNDS,LHR,LMN,FSCS,JAFC(5),ASEC,KAFC(4),FAFC(8)
      COMMON/BLK3/ NHR(100),NMN(100),FSC(100),FRNG(100),FAZ(100),
     1NALT(100),NCD(100),NST(100),NVC(100),NVA(100),NSP(100),KAP
     2,LAP,NKNT(100)
      INTEGER SCODE(100)
      INTEGER IKT,SSCAN(100,9),SCAN,CODE
      DIMENSION RNG(100,9),AZ(100,9)
      DIMENSION X(2),Y(2)
      DIMENSION THRF(200),RRF(200),THDR(200)
      INTEGER K(100)
      INTEGER ICREF
      INTEGER IGO
      DATA IKT/0/
      DATA ICREF/0/
      DATA K/100*0/
  905 FORMAT(2X2I3,F7.3,2F7.2,3I3)
  906 FORMAT(/)
  907 FORMAT(//)
  908 FORMAT(6X*H*2X*M*4X*SEC*4X*RNG*4X*AZM*2X*ALT*2X*BCD*3X*VA*
     14X*W*3X*VC*4X*S*4X*F*1X*RCRD*)
  909 FORMAT(5X*S*7X*AZM*7X*RNG*)
      LSCN=SCAN-1
      DEL=.05
      RAD=3.14159/180.
      IF( CODE.EQ.0) GO TO 7000
      IF(CODE.EQ.7305)GO TO 7000
C  PLACE THE INLINE CODE HERE
C       IGO DEFINED
      IGO=MOD(CODE,100)
      IF(IGO.EQ.0) RETURN
      IF(IKT.EQ.0) GO TO 3
      DO 200 IC=1,IKT
      IF(CODE.EQ.SCODE(IC)) GO TO 300
  200 CONTINUE
      IF(IKT.GE.100) GO TO 7000
    3 IKT=IKT+1
      IC=IKT
      PRINT196,CODE
  196 FORMAT(1X,*ADDITIONAL CODE =*,I8)
      SCODE(IKT)=CODE
  300 IF(AZR.GT.357.5) SCAN =SCAN-1
  301 IF(K(IC).GT.0) GO TO 400
      K(IC)=1
      GO TO 3010
C
C IF AZIMUTH WITHIN 370 OF HEAD OF LIST
  400 IF(SCAN.EQ.SSCAN(IC,1)) GO TO 3000
      IF(SCAN.NE.SSCAN(IC,1)+1) GO TO 1000
  550 IF(AZR.LT.AZ(IC,1)+10.) GO TO 2000
C
      IF(K(IC).GT.1) GO TO 1000
      K(IC)=1
      GO TO 3011
C
C UPDATE THE SAVED LIST DELETEING THE FIRST ELEMENT, MOVING THEM ALL UP
 1000 K(IC)=K(IC)-1
```

```
      IF(K(IC).EQ.0) GO TO 3000
      II=K(IC)
      DO 1500 III=1,II
      AZ(IC,III)=AZ(IC,III+1)
      RNG(IC,III)=RNG(IC,III+1)
      SSCAN(IC,III)=SSCAN(IC,III+1)
1500  CONTINUE
C
      GO TO 400
2000  IF(K(IC).GT.1) GO TO 2400
      K(IC)=2
      GO TO 3010
C
C IS THE RANGE>PRESENT ANGR IE.DEFINED FALSE TARGET
2400  IF(RNG(IC,1).GT. RNG(IC.2)) GO TO 1000
      NN=2
2410  IF(RNG(IC,NN).LT.RANGE) GO TO 3000
      N1=0
C
C PROCESS FOR REFLECTOR
      PRINT111,SCODE(IC),LHR,LMN,FSCS
111   FORMAT(/1X,*QUESTIONABLE GROUP CODE=*,I8,
     12X*LAST SECTOR TIME*2I3,F7.3)
      NR=K(IC)
      PRINT900
      DO 113 IR=1,NR
      PRINT112,SSCAN(IC,IR),AZ(IC,IR),RNG(IC,IR)
112   FORMAT(1X,I5,2(2X,F8.2))
113   CONTINUE
      PRINT114,SCAN,AZR,RANGE
114   FORMAT(1X,I5,2(2X,F8.2),2X,I8)
C
C CHOOSING 0<X1
      ZK=(AZ(IC,NN)-AZ(IC,1))/360.
2470  IF(ZK.GE.0.) GO TO 2475
      ZK=ZK+1.
      GO TO 2470
2475  IF(ZK.LE.1) GO TO 2480
      ZK=ZK-1.
C
      GO TO 2475
2480  RA=RNG(IC,1)+ZK*(RANGE-RNG(IC,1))
      THETAA=AZ(IC,1)+ZK*(AZR-AZ(IC,1))
      GO TO 2511
2490  RNG(IC,NN)=2*RNG(IC,NN)-RA
      N1=N1+1
2500  THOR(ICREF)=(AZ(IC,NN)+THETAA  )/2.
2510  IF(THOR(ICREF).GE.0) GO TO 2520
      THOR(ICREF)=THOR(ICREF)+180.
      GO TO 2510
2520  IF(THOR(ICREF).LE.180. ) GO TO 2530
      THOR(ICREF)=THOR(ICREF)-180.
      GO TO 2520
2530  THRF(ICREF)=AZ(IC,NN)
      RN= .5*(RNG(IC,NN)*RNG(IC,NN)-RA*RA)
      RD=RNG(IC,NN)-RA*COS((AZ(IC,NN)-THETAA)*RAD)
      IF(RD.EQ.0.)PRINT 99,IC,K(IC),CODE,NN,RD,RA,RNG(IC,NN)
```

```
SUBROUTINE REFLECT    74/74   OPT=1                    FTN 4.2+74261      01/
      IF(RD.LT.0.0000001)RD=0.0000001
   33 FORMAT(4(2X,I6),3(5X,F12.4))
      RRF(ICREF)=RN/RD
      PRINT4000,THRF(ICREF),RRF(ICREF),THOR(ICREF)
 4000 FORMAT(1X,*ANGLE OF REFLECTOR =*,F7.2,2X,*RNG OF RELCTR=*,F7.2
     1,2X,*ORIENTATION *,F7.2)
      PX  =SIN(THRF(ICREF)*RAD)*RRF(ICREF)
      PY = COS(THRF(ICREF)*RAD)*RRF(ICREF)
      DY=DEL*COS(THOR(ICREF)*RAD)
      DX=DEL*SIN(THOR(ICREF)*RAD)
      X(1)=PX+DX
      Y(1)=PY+DY
      X(2)=PX-DX
      Y(2)=PY-DY
      IF(N1.EQ.0.AND.RANGE.LE.8) GO TO 2490
      PRINT950,JAPC,ASEC,KAPC,FAPC
  950 FORMAT(/4X*UT*3X*A*2X*TP*2X*HP*1X*MIN*5X*SEC*
     14X*ACID*5X*ALT*4X*ABCD*4X*RBCD*5X*AZM*5X*RNG*
     26X*XP*6X*YP*6X*XD*6X*YD*5X*VEL*4X*HDNG*/
     32X3I4,F8.3,1XA7,5XA3,2I8,4F8.1,3F8.3,F8.1/)
      JAP=KAP
      NFLG=0
      LAP=1
      PRINT904
      DO 602 JPP=1,100
      IF(NSP(JAP).LT.LSCN)GO TO 601
      IF(NFLG.GT.0)GO TO 621
      IF(NSP(JAP).GT.LSCN)GO TO 620
  621 CONTINUE
      PRINT905,NHR(JAP),NMN(JAP),FSC(JAP),FRNG(JAP),FAZ(JAP),
     1NALT(JAP),NCD(JAP),NVA(JAP),NST(JAP),NVC(JAP),NSP(JAP)
     2,LAP,NKNT(JAP)
  601 CONTINUE
      JAP=MOD(JAP,100)+1
  602 CONTINUE
      NN=NN+1
      ICREF=ICREF+1
      IF(ICREF.GT.200)GO TO 7000
      IF(K(IC).GE.NN) GO TO 2410
      K(IC)=0
 3000 IF(K(IC).LT.9) GO TO 3001
      PRINT444,K(IC)
  444 FORMAT(1X,*MORE THAN 9 ELEMENTS *,I10)
      K(IC)=K(IC)-1
      II=K(IC)
      DO 4500 III=1,II
      AZ(IC,III)=AZ(IC,III+1)
      RNG(IC,III)=RNG(IC,III+1)
      SSCAN(IC,III)=SSCAN(IC,III+1)
 4500 CONTINUE
 3001 K(IC)=K(IC)+1
 3010 L=K(IC)
      AZ(IC,L)=AZP
      RNG(IC,L)=RANGE
      SSCAN(IC,L)=SCAN
 7000 CONTINUE
      RETURN
```

```
      620   CONTINUE
            NELG=1 $ PRINT917
            GO TO 621
            END
```

APPENDIX F

ILLUSTRATIVE RUNS

An illustrative run, using a data tape from a flight test conducted on 6 May 1975, is presented for the purpose of describing the input procedure and printout results. The aircraft was a T-39 (MIKE 61) with a beacon code of 2630. The aircraft was flown in the reflection area from the Advanced Radiation Test Facility (ARTF) hangar on Kirtland Air Force Base.

A copy of this data tape is available from AFWL (SUE) for those desiring to obtain JASPER and check it out against illustrative run. Only selected pages from the output have been included here. The total output is too voluminous to be included. The first sheet (page 60) of the output has all of the identification data related to this run, data tape used, and test aircraft and its code. Also all other aircraft in the terminal area are identified under "Additional Code". On the second sheet (page 61), the aircraft identified as questionable, are actually splits.

The next four sheets (pages 62 through 65) have been extracted from the total output to illustrate conventional false target indications from both the ARTF and the Manzano Area Fence.

```
FAA IP > MAY 75 6433,FILE1,40 <CR><LF>
NIKE61 CODE 2636

MESS EQUALS          4      JPOS EQUALS       9999
LRCD SET AT          5
LRCD USED FOR INPUT RECORD SIZE OPTION
LRCD SET AT 4 FOR 142 WRD RCDS
LRCD SET AT 5 FOR 561 WRD RCDS

                FILE ID RECORD

        I21  4336  CAIL 36,522/126  NIKE61/2636

        00  -9.094  96672  1.933  UPPER SCREEN WEST WALL


LTN   PCJ   SQN    HR   MIN     SEC     ALT     BCD     RNG      AZ   ACP GR MD

MSG CODE            1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19
MSG REC COUNT       1    2    2    1    1   20   11    7    4   10    4    4    1    1   99    6    1    1   13

SELECT STATE1 DIE OR TWO    2    4    8

MONITOR TRACK STORE ON BEACON CODE    2636
MAX ALT    9999

ADDITIONAL CODE =     3226
ADDITIONAL CODE =     3523
ADDITIONAL CODE =     3565
ADDITIONAL CODE =     3262
ADDITIONAL CODE =     2535
ADDITIONAL CODE =     1352
ADDITIONAL CODE =     2636
ADDITIONAL CODE =      342
ADDITIONAL CODE =      346
ADDITIONAL CODE =     3520
ADDITIONAL CODE =     3764
ADDITIONAL CODE =     3525
ADDITIONAL CODE =     2621

QUESTIONABLE GROUP CODE=    3262  LAST SECTOR TIME 22  7 35.669
    S        ZZY          ENG
    2    2.4.3*          3.64
    7    210.2*          3.7*
```

```
 5    215.49       9.79

                 VM          ZS    TRGT/AZ  TRGT/RNG  FALSE/AZ FALSE/RNG  PELTR/AZ PELTR/RNG PELTR/ANG  SPLT/CNT
SPLIT              2         .44     215.34      9.79     214.23      9.79                                    1

 UT   A  TD  HR MIN      SEC      ACID      ALT      ABCD      RBCD      AZM      RNG       XP       YP       XD       YD      VEL     HDNG
  1   3   2  22   3   21.670 MTK661-       ---      2670      2630      21.4      4.8      1.5      4.7     -.044     .144     .151   -18.4

 H  M      SEC      RNG      AZM     ALT     RCD   VA   M   MC   S   F RCRD
22  3   23.819     3.41    16.52     169  2670    1    1    3   2   1    6
22  3   24.422    17.69    22.94     196  7305    3    1    3   6   1    7
22  3   23.422     5.41    24.49       0  2630    3    1    1   6   1    7
22  3   29.567     5.26    37.57      67   342    3    1    3   2   1    7
22  3   29.677     5.19    55.76      64   340    3    1    3   2   1    7
22  3   29.737     5.09    68.96     141  3226    3    1    3   2   1    8
22  3   29.923    16.13    86.92     203  3764    3    1    3   2   1    8
22  3   30.555    31.41   122.17       3  1466    3    0    1   2   1    9
22  3   31.253    33.50   171.47     114  3226    3    1    3   2   1   10
22  3   31.603     9.16   177.57      72  3523    3    1    3   2   1   10
22  3   31.653     3.03   177.94       0  3555    3    1    0   2   1   11
22  3   31.193    34.06   182.72       3     0    3    1    0   2   1   11
22  3   31.429     4.03   209.36       0  3262    3    0    1   2   1   12
22  3   31.369     6.79   241.35     331  2635    3    1    3   2   1   12
22  3   32.079    22.19   266.34     330  1352    3    1    3   2   1   13
22  3   32.961    12.36   334.78       0  1200    3    1    0   2   1   14

22  3   33.711     4.76     5.45       0  2630    3    0    1   3   1   15
22  3   33.491     5.19    93.72     144  3525    3    1    3   3   1   17
22  3   33.981    15.75    66.36     201  3764    3    1    3   3   1   17
22  3   33.861     5.25    72.95      54   340    3    0    3   3   1   17
22  3   34.464     3.19   112.14       0  1200    3    0    0   3   1   18
22  3   34.459    35.41   114.44     294  2621    3    1    3   3   1   18
22  3   34.852    32.26   123.13       0  1400    3    0    0   3   1   19
22  3   35.142    33.81   171.74     104  3260    3    1    3   3   1   20
22  3   35.102     6.38   173.35      72  3523    3    1    3   3   1   20
22  3   35.274     3.44   174.47       0  3555    3    1    0   3   1   20
22  3   35.736    34.06   182.72       3     0    3    1    0   3   1   20
22  3   35.444     9.79   215.23     350  3262    3    1    3   3   1   21
22  3   35.609     9.79   215.95     350  3262    3    1    3   3   1   21
22  3   36.117    21.83   269.09     330  1352    1    1    3   3   5   22
22  3   36.857    12.75   334.34       0  1200    1    0    3   3   5   24
22  3   37.575     4.73    12.22     163  2630    1    2    3   4   5   25
ADDITIONAL CODE =     3521

QUESTIONABLE GROUP CODE=    2630  LAST SECTOR TIME 22  3 49.508
  S       AZM       RNG
  6      11.46      5.13
  7       4.22      5.44
  7      11.34      5.44

                 VM          ZS    TRGT/AZ  TRGT/RNG  FALSE/AZ FALSE/RNG  PELTR/AZ PELTR/RNG PELTR/ANG  SPLT/CNT
SPLIT              2         .98     11.32      5.13      4.22      5.44                                    2

 UT   A  TD  HR MIN      SEC      ACID      ALT      ABCD      RBCD      AZM      RNG       XP       YP       XD       YD      VEL     HDNG
  1   3   2  22   3   45.718 MTK661-       169      2670      2630      10.8      5.0       .9      4.9     -.015     .067     .069   -12.3

 H  M      SEC      RNG      AZM     ALT     RCD   VA   M   MC   S   F RCRD
22  3   45.421     5.13    11.95     169  2630    3    1    3   6   1   42
22  3   45.516    17.59    22.76     196  7305    3    1    3   6   1   43
22  3   45.844     4.88    44.47      64   342    3    1    3   6   1   43
22  3   45.844     1.46    43.33      54   340    3    1    3   6   1   43
22  3   45.954     5.13    53.62     144  3521    3    1    3   6   1   44
22  3   46.039    14.50    66.78     201  3764    3    1    3   6   1   44
22  3   46.612     1.11   112.08       0  1200    3    1    0   6   1   46
```

```
22 26 17.504   59.16 192.72     0     0    3    1    0  340    1 3142
22 26 17.713   44.38 193.39   324  3731    7    1    6  741    1 3142
22 26 18.227    2.76 237.64     5   361    3    1    2  340    1 3143
22 26 18.359   27.31 254.97   105  2265    3    1    3  340    1 3143
22 26 18.494   15.76 254.75     3   321    7    1    0  340    1 3144
22 26 18.496   47.50 259.37   271  2612    3    1    3  340    1 3144
22 26 18.629   32.50 269.13     4  1460    3    1    0  340    1 3144
22 26 18.629    9.58 277.42     0   321    3    1    0  340    1 3144
22 26 18.764   21.16 286.48    90  2636    3    1    3  341    1 3145
22 26 19.113     .69 111.94     0  1260    3    1    0  340    1 3145
22 26 19.367   16.25 346.34   260  2834    7    1    3  740    1 3146

22 26 19.497   13.64  21.03     3  7705    3    1    0  341    1 3147
22 26 20.230    7.31  99.62    55  2617    3    1    3  341    1 3149
22 26 20.363    7.13  88.90     0  1034    1    1    0  341    1 3149
22 26 20.629    7.26  89.97     4  1480    3    1    0  341    1 3149
22 26 21.641   42.13 191.65   135  1401    3    1    3  341    1 3151
22 26 21.641   43.81 192.29     4  1260    3    1    0  341    1 3151
22 26 21.641   54.15 182.72     0     0    3    1    0  341    1 3151
22 26 21.766   46.13 193.94   127  3731    3    1    3  341    1 3151
22 26 22.026    5.96 237.66     0   321    3    1    0  341    1 3152
22 26 22.359   27.10 254.94   104  2265    3    1    3  341    1 3152
22 26 22.628   17.44 254.43     0   320    3    1    0  341    1 3153
22 26 22.894   29.44 259.37   273  2612    3    1    3  341    1 3153
22 26 22.894   32.50 269.03     0  1460    3    1    0  341    1 3153
22 26 22.894    9.58 277.24     0   321    3    1    0  341    1 3153
22 26 22.763   20.04 287.31    91  2633    1    3    3  341    5 3154
22 26 23.031     .75 111.62     4  1260    1    1    3  341    5 3154
22 26 23.496   16.09 346.85   251  2634    1    3    3  341    5 3155
22 26 23.960   13.69  22.04   106  7305    1    3    3  342    5 3156
```

ADDITIONAL CODE = 2414
ADDITIONAL CODE = 322
ADDITIONAL CODE = 7101
ADDITIONAL CODE = 2664
ADDITIONAL CODE = 2244
ADDITIONAL CODE = 6460
ADDITIONAL CODE = 6664
ADDITIONAL CODE = 3664
ADDITIONAL CODE = 120
ADDITIONAL CODE = 310

QUESTIONABLE GROUP CODE= 2630 LAST SECTOR TIME 22 30 49.461

S	AZM	RNG
406	277.73	27.56
407	99.89	30.63
407	277.93	27.69

CHAN	HN	ZK	TRGT/AZ	TRGT/RNG	FALSE/AZ	FALSE/RNG	RFLTR/AZ	RFLTR/RNG	RFLTR/ANG	SPLT/CNT
	2	.43	277.46	27.81	94.68	30.69	59.68	1.71	167.53	

UT	A	TP	HR	MIN	SEC	ACID	ALT	APCD	RBCD	AZM	RNG	XP	YP	XD	YD	VEL	HDNG
1	2	L	22	30	45.322	NIK261=	117	2630	2630	277.8	27.6	-27.3	3.8	-.064	-.031	.072	244.1

```
H  M    SEC     RNG    AZM  ALT  RCD  VA   W   VC    S   F RCRD
22 30 42.343  14.31 194.74    0 1200    3   1   0  406   1 3719
22 30 42.641  13.69  22.04    2 7305    3   1   0  406   1 3719
22 30 42.766   1.96  40.61    0 2664    3   1   1  406   1 3720
22 30 43.031  26.36 143.31  330 3721    3   1   3  406   1 3722
22 30 44.071   4.19 192.94    0  344    3   1   0  406   1 3722
22 30 44.196  36.25 156.53    3 1200    7   1   0  406   1 3722
22 30 44.196  36.43 156.61   64 1200    3   1   3  406   1 3722
22 30 44.406  25.43 142.59  135 1440    3   1   3  406   1 3723
22 30 44.406  59.26 182.72    2    0    3   1   0  406   1 3723
22 30 45.039   7.78 237.68   42 2265    3   1   3  406   1 3724
```

22	31	45.287	25.41	358.53	3	1470	3	1	0	436	1	3725		
22	31	45.476	21.63	270.39	4	1408	3	1	0	436	1	3725		
22	31	45.632	27.06	277.73	117	2630	3	1	3	436	1	3725		
22	31	45.787	5.13	313.17	7	322	3	1	0	436	1	3726		
22	31	46.035	5.25	358.76	5	2676	3	1	3	436	1	3724		
22	31	46.658	16.63	358.77	4	1364	3	1	3	437	1	3724		
22	31	46.846	13.59	22.95	100	7305	3	1	3	437	1	3724		
22	31	46.939	1.69	77.27	6	2664	3	1	3	437	1	3724		
22	31	47.065	30.39	99.88	117	2630	3	0	3	437	1	3723		
22	31	47.252	45.24	145.98	329	3731	3	1	3	437	1	3731		
22	31	47.376	4.31	156.12	2	344	3	1	0	437	1	3731		
22	31	47.595	37.31	159.61	88	1720	3	1	3	437	1	3731		
22	31	47.845	29.63	182.64	106	1408	3	1	3	437	1	3732		
22	31	48.064	34.38	182.72	4	0	3	1	0	437	1	3732		
22	31	48.283	7.19	267.43	81	2205	3	1	3	437	1	3733		
22	31	48.408	4.75	270.14	0	1408	3	1	0	437	1	3734		
22	31	48.564	27.63	277.83	118	2630	3	1	3	437	1	3734		
22	31	48.814	3.25	310.25	0	322	1	0	3	437	5	3735		
22	31	50.408	16.44	357.99	0	1260	1	0	3	438	5	3737		

QUESTIONABLE GROUP CODE= 321 LAST SECTOR TIME 22 32 34.469

S	AZM	RNG
432	261.39	15.25
433	260.68	15.34
434	261.56	15.34

	NN	PK	IRGT/AZ	IRGT/RNG	FALSE/AZ	FALSE/RNG	RFLTR/AZ	RFLTR/RNG	RFLTR/ANG	SPLT/CNT
SPLIT	2	1.00	261.56	15.37	260.68	15.34				47

UT	A	T	HR	MIN	SEC	ACID	ALT	ABCD	RBCD	AZM	RNG	XP	YP	XD	YD	VEL	HDNG
1	3	L	22	32	36.919	MIK661-	121	2630	2630	278.1	32.8	-32.4	4.6	-.041	.061	.073	-34.1

H	M	SEC	SNG	AZM	ALT	RBCD	VA	W	VC	S	F	RCRD
22	32	27.656	10.66	357.94	0	1260	3	0	0	432	1	3954
22	32	27.756	13.69	23.03	0	7305	3	1	3	432	1	3955
22	32	28.431	34.34	129.55	328	3731	3	1	3	432	1	3957
22	32	28.873	5.34	159.36	2	344	3	1	0	432	1	3958
22	32	29.273	36.75	161.19	0	3557	3	0	3	432	1	3958
22	32	29.339	3.25	170.64	79	2205	3	1	3	432	1	3958
22	32	29.523	27.25	183.75	105	1408	3	1	3	432	1	3959
22	32	29.623	34.34	182.72	0	0	3	1	0	432	1	3959
22	32	30.351	5.25	223.96	75	2664	3	1	3	432	1	3960
22	32	30.414	36.44	260.24	0	1400	3	0	0	432	1	3961
22	32	30.414	15.25	261.39	0	321	3	1	0	432	1	3961
22	32	30.539	42.75	270.19	3	1408	3	1	0	432	1	3961
22	32	31.039	5.38	313.07	0	322	3	1	0	432	1	3962
22	32	31.727	13.69	23.03	100	7305	3	1	3	433	1	3964
22	32	32.439	12.94	83.27	253	2634	3	0	3	433	1	3965
22	32	32.976	36.31	129.26	329	3731	3	1	3	433	1	3967
22	32	33.317	5.75	159.03	0	344	3	1	0	433	1	3967
22	32	33.317	36.38	160.14	0	3557	3	1	0	433	1	3967
22	32	33.473	3.25	166.55	77	2205	3	0	3	433	1	3967
22	32	33.623	29.16	182.90	105	1408	3	1	3	433	1	3968
22	32	33.923	34.16	182.72	0	0	3	1	0	433	1	3968
22	32	34.673	5.44	226.49	76	2664	3	1	3	433	1	3969
22	32	34.936	15.25	261.68	0	321	0	0	0	433	1	3970
22	32	34.945	15.44	261.56	0	321	3	1	0	433	1	3970
22	32	34.978	42.38	270.19	0	1408	1	0	3	433	5	3972
22	32	34.763	37.06	279.44	121	2630	1	3	3	433	5	3971
22	32	35.078	6.44	312.94	0	322	1	0	3	433	5	3972

```
22 32 55.456  13.69  24.37  105 7765    1    3    1  434    5 1973

QUESTIONABLE GROUP CODE=     322  LAST SECTOR TIME 22 33  7.445
   S       AZM        RNG
 449     312.65       6.75
 441      33.32      16.31
 441     313.37       8.41

                NM        ZK   TRGT/AZ  TRGT/RNG  FALSE/AZ FALSE/RNG  RFLTR/AZ RFLTR/RNG RFLTR/ANG  SPLT/CNT
CNVNL            2       .36   312.79      6.77     33.32     16.31     83.32      5.32      7.24
STDELOW          2       .36   312.79      6.77     33.32     25.45     83.32     10.29      2.98

  HT   A  TP  HR MIN      SEC     ACID      ALT     ABCD     RBCD      AZM      RNG       XP       YP       XD       YD      VEL     HDNG
   1   0   0  22  33   7.324 MIK161-      119     2630     2630    279.1      6.8     -6.4      5.6     -.058     .097     .098    -93.2

  H  M     SEC      RNG     AZM  ALT  BCD  VA  W  VC   S  F RCRD
 22 32  59.859     4.63  359.57    0 1200   3  0   0 440  1 4029
 22 33    .117    15.23   22.08  105 7365   3  1   3 441  1 4030
 22 33    .742    19.56   79.61  254 2634   3  0   2 440  1 4031
 22 33   1.251    57.19  120.21  330 3731   3  1   3 440  1 4032
 22 33   1.745     3.09  171.64   74 2265   3  1   3 440  1 4032
 22 33   1.976     4.25  144.95    0  344   3  1   3 440  1 4032
 22 33   1.615    53.28  154.52    0 3557   3  0   0 440  1 4033
 22 33   1.993    14.26  193.43  105 1400   3  1   3 440  1 4034
 22 33   1.313    50.06  182.72    0    0   3  1   0 440  1 4034
 22 33   2.133     4.56  264.26   85 2684   3  1   3 440  1 4034
 22 33   2.766    16.44  266.51    0  321   3  1   0 440  1 4036
 22 33   2.898    46.09  278.62    0 1400   3  1   0 440  1 4036
 22 33   3.063    34.09  273.53  119 2630   3  1   3 440  1 4036
 22 33   3.339     6.75  312.63    0  322   3  1   0 440  1 4037

 22 33   3.976     5.44  353.47    0 1200   3  0   0 441  1 4038
 22 33   4.156    17.59   22.85  106 7365   2  1   3 441  1 4039
 22 33   4.719    15.51   83.32    0  322   3  0   0 441  1 4040
 22 33   5.237    50.31  125.86  330 3731   3  1   3 441  1 4041
 22 33   5.422     3.75  135.53   74 2265   3  1   3 441  1 4042
 22 33   5.970    18.38  193.62  105 1400   3  1   3 441  1 4043
 22 33   5.970    59.46  182.72    0    0   3  1   0 441  1 4043
 22 33   6.116     6.75  200.92   86 2664   3  1   3 441  1 4044
 22 33   6.419    13.56  260.24    0  321   3  1   0 441  1 4045
 22 33   6.954    46.75  269.74    0 1400   3  1   0 441  1 4045
 22 33   6.988    34.38  278.35  120 2630   3  1   3 441  1 4046
 22 33   7.445     6.81  313.07    0  322   3  1   0 441  1 4047
 22 33   7.965     4.25  353.05    0 1200   1  0   3 442  5 4048

QUESTIONABLE GROUP CODE=     322  LAST SECTOR TIME 22 33 43.727
   S       AZM        RNG
 449     312.65       7.69
 450      42.44      17.29
 450     312.19       7.79

                NM        ZK   TRGT/AZ  TRGT/RNG  FALSE/AZ FALSE/RNG  RFLTR/AZ RFLTR/RNG RFLTR/ANG  SPLT/CNT
CNVNL            2       .36   312.47      7.71     82.44     17.25     82.44      5.36      7.24
STDELOW          2       .36   312.47      7.71     82.44     26.79     82.44     10.37      2.99

  HT   A  TP  HR MIN      SEC     ACID      ALT     ABCD     RBCD      AZM      RNG       XP       YP       XD       YD      VEL     HDNG
   1   0   0  22  33  43.727 MIK161-      121     2630     2630    274.9     35.8    -35.6      3.1     -.035     -.057     .067    211.8

  H  M     SEC      RNG     AZM  ALT  BCD  VA  W  VC   S  F RCRD
 22 33  36.266     7.36    1.76    0 1200   0  0   0 449  1 4112
 22 33  36.516    15.09   22.59  106 7365   3  1   3 449  1 4113
 22 33  37.531     5.41  111.71   72 2265   3  1   3 449  1 4115
 22 33  38.031    43.56  154.31    0 3557   3  0   0 449  1 4116
    33  38.164     6.25  174.59    0 2664   3  0   1 449  1 4117
```

22	33	38.259	16.63	183.44	1.6	1460	3	1	3	449	1	4117	
22	33	38.259	94.65	182.72	4	4	3	1	0	449	1	4117	
22	33	39.172	17.41	268.64	4	321	3	1	0	449	1	4119	
22	33	39.375	46.49	269.91	3	1400	3	1	4	449	1	4119	
22	33	39.375	35.53	275.6	121	2630	3	1	3	449	1	4119	
22	33	39.419	7.53	312.97	4	322	3	1	0	449	1	4121	
22	33	40.963	17.63	23.83	106	7365	3	1	3	450	1	4122	
22	33	41.145	17.39	82.44	4	322	3	1	0	450	1	4124	
22	33	41.445	5.81	107.83	71	2265	3	1	3	450	1	4124	
22	33	41.628	7.56	142.12	2	344	3	1	0	450	1	4125	
22	33	42.078	49.19	159.17	0	3597	3	1	0	450	1	4125	
22	33	42.211	5.79	163.72	38	2664	3	1	3	450	1	4126	
22	33	42.336	59.56	182.72	0	0	3	1	0	450	1	4126	
22	33	42.336	16.44	144.57	0	1400	3	1	1	450	1	4126	
22	33	43.219	18.22	268.68	4	321	3	1	3	450	1	4129	
22	33	43.219	57.63	269.35	0	1401	3	1	0	450	1	4129	
22	33	43.344	45.31	269.74	4	1401	3	1	0	450	1	4129	
22	33	43.344	35.75	274.39	120	2630	3	1	3	450	1	4129	
22	33	43.727	7.75	312.19	4	322	3	1	0	450	1	4130	
22	33	44.399	6.75	2.66	3	1400	0	6	1	451	5	4131	

QUESTIONABLE GROUP CODE= 322 LAST SECTOR TIME 22 33 47.781

S	AZM	RNG
450	312.19	7.75
451	42.83	17.31
451	312.36	7.84

	NN	PK	TRGT/AZ	TRGT/RNG	FALSE/AZ	FALSE/RNG	RFLTR/AZ	RFLTR/RNG	RFLTR/ANG	SPLT/CNT
CHANNEL	2	.36	312.25	7.80	42.88	17.31	82.88	5.34	7.68	
SIDELOBE	2	.36	312.25	7.80	42.88	26.83	82.88	10.35	3.39	

UT	1	T2	HR	MIN	SEC	AGID	ALT	ARCD	RCD	AZM	RNG	XP	YP	XD	YD	VEL	HDNG
1	2	0	22	33	47.782	MIKE61=	120	2630	2630	274.5	35.9	-35.8	2.4	-.036	-.063	.071	208.7

H	M	SEC	RNG	AZM	ALT	IDD	VA	W	VC	S	F	RCDD
22	33	40.963	17.63	23.83	106	7365	3	1	3	450	1	4122
22	33	41.145	17.39	82.44	0	322	3	1	0	450	1	4124
22	33	41.445	5.81	107.83	71	2265	3	1	3	450	1	4124
22	33	41.628	7.56	142.12	2	344	3	1	0	450	1	4125
22	33	42.078	49.19	159.17	0	3597	3	1	0	450	1	4125
22	33	42.211	5.79	163.72	38	2664	3	1	3	450	1	4126
22	33	42.336	59.56	182.72	0	0	3	1	0	450	1	4126
22	33	42.336	16.44	144.57	0	1400	3	1	1	450	1	4126
22	33	43.219	18.22	268.68	4	321	3	1	3	450	1	4129
22	33	43.219	57.63	269.35	0	1401	3	1	0	450	1	4129
22	33	43.344	45.31	269.74	4	1401	3	1	0	450	1	4129
22	33	43.344	35.75	274.39	120	2630	3	1	3	450	1	4129
22	33	43.727	7.75	312.19	4	322	3	1	0	450	1	4130
22	33	44.399	6.75	2.66	3	1400	0	6	1	451	5	4131
22	33	44.659	15.69	23.84	4	7365	3	1	0	451	1	4132
22	33	45.242	17.31	82.88	0	322	3	6	0	451	1	4133
22	33	45.442	5.75	108.42	0	2265	3	1	1	451	1	4134
22	33	46.091	7.13	144.88	2	344	3	1	0	451	1	4134
22	33	46.133	49.09	159.73	4	3597	3	6	0	451	1	4135
22	33	46.266	5.81	166.11	107	2664	3	1	3	451	1	4135
22	33	46.391	59.56	182.72	0	0	3	1	4	451	1	4136
22	33	46.391	16.25	144.15	115	1410	3	1	3	451	1	4136
22	33	47.273	18.12	268.64	4	321	3	1	0	451	1	4138
22	33	47.394	45.36	270.66	3	1401	3	1	0	451	1	4138
22	33	47.781	7.68	312.36	0	322	3	1	4	451	1	4139

APPENDIX G

GLOSSARY OF PRINCIPAL VARIABLE NAMES USED FOR OUTPUT

PROGRAM SYMBOL	MEANING / UNITS
S	Scan, one complete revolution of antenna starting with "1" from beginning of data tape
AZM	Azimuth, degrees magnetic heading
RNG	Range, nautical miles (NM)
TRGT/AZ	Target azimuth, actual aircraft, degrees magnetic heading (M.H.)
TRGT/RNG	Target range, actual aircraft, nautical miles
FALSE/AZ	False target azimuth, degrees (M.H.)
FALSE/RNG	False target range, nautical miles
RFLTR/AZ	Reflector azimuth, degrees (M.H.)
RFLTR/RNG	Reflector range, nautical miles
RFLTR/ANG	Reflector orientation angle, degrees from magnetic north
SPLT/CNT	Split Count
H & HR	Hour, zulu time
M & MIN	Minutes
SEC	Seconds
ALT	Altitude, feet from sea level
ACID	Aircraft Identity Code
ABCD	Aircraft beacon code
X,Y	Range X and Y components, N.M.
VEL	Aircraft velocity, N.M./sec

PROGRAM SYMBOL	MEANING / UNITS
XD & YD	X & Y components of velocity in N.M./ sec
HDNG	Aircraft heading (MH)

APPENDIX H

DETERMINATION OF SYSTEM REPEATABILITY

This appendix describes a feature that has been incorporated in JASPER to determine the "sensitivity" of the system. The term "sensitivity" used here is not used in the true sense, but in a relative sense. It is used to indicate that there is a change in the system from flight test to flight test or from one data tape acquisition to the next.

In order to have some consistency in the measure of "sensitivity", a fixed signal beacon should be used instead of an aircraft system. This is because the same aircraft is not always available and systems vary between aircrafts. Also with the aircraft moving (i.e., altitude, azimuth and range changing), the acquired data varied excessively in a single flight test.

Therefore, here at Albuquerque, the FAA beacon atop Sandia Mountain was used. It has an output of 200 watts peak power and produced consistent data throughout the tape.

The "sensitivity" is obtained in the following manner and as shown in Figure H1. The angle, ΔAZ, is found between where the antenna beam first acquires the beacon and where it is last acquired. That angle, ΔAZ, which is normally given in ACP (Azimuth Change Pulse) is converted to degrees and beam half-angle. This half-angle is used in a table look-up in the Program which has been obtained from antenna patterns for a particular antenna system.

The half-angle is obtained in the following manner,

$$4096 \text{ ACP} = 360°$$

or

$$0.0879 \text{ ACP/DEG}$$

$$\Delta AZ = (\text{Total ACP}) * (0.0879)$$

$$\text{Beam } \tfrac{1}{2} \text{ angle} = \frac{\Delta AZ}{2}$$

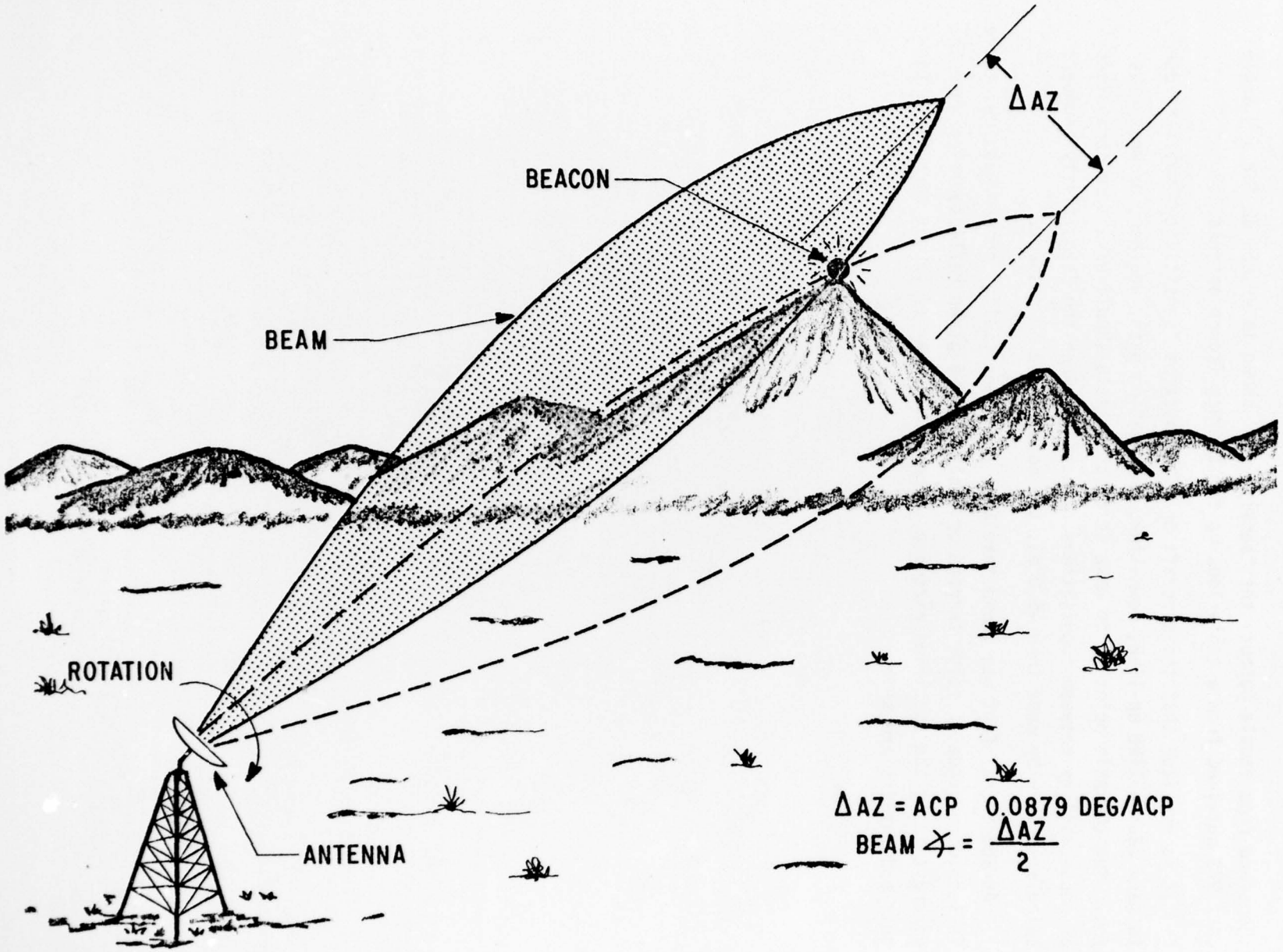


Figure H1. Beacon Half-Angle Definition

Therefore from sample output, the "sensitivity" shown is -9.069 dB for a ½ angle of 1.933 obtained from a table look-up for the Hog Trough antenna.

It is realized that the aircraft or ATCRB systems "sensitivity" may vary due to many causes like up-link, down-link, propagation path, weather, or what have you. But the main point to be made is that the data should have some consistency or repeatability between acquisitions. Therefore when the "sensitivity" numbers calculated vary by more than -5.0 dB, the data should be suspect.

An additional fact was found that when the "sensitivity" number was low, there is less occurrence of false targets or replies off a given reflector even when the aircraft is in the specified airspace. When the number if larger more false targets and replies are seen.