FAA-RD-78-96

Project Report
ATC-87
Volume 2

The Aircraft Reply and Interference Environment Simulator (ARIES) Volume 2: Appendices to the Principles of Operation

M. Goon
D. A. Spencer

22 March 1979

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Federal Aviation Administration,
Washington, D.C. 20591

Technical Report Documentation Page

1. Report No. FAA-RD-78-96	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle The Aircraft Reply and Interference Environment Simulator (ARIES) Volume 2: Appendices to the Principles of Operation		5. Report Date 22 March 1979 6. Performing Organization Code
7. Author(s) Michael Goon and David A. Spencer		8. Performing Organization Report No. ATC-87, Volume 2
9. Performing Organization Name and Address Massachusetts Institute of Technology Lincoln Laboratory P.O. Box 73 Lexington, MA 02173		10. Work Unit No. Proj. No. 052-241-04 11. Contract or Grant No. DOT-FA77-WAI-261
12. Sponsoring Agency Name and Address Department of Transportation Federal Aviation Administration Systems Research and Development Service Washington, DC 20591		13. Type of Report and Period Covered Project Report 14. Sponsoring Agency Code

15. Supplementary Notes

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the support of the Department of the Air Force under Contract F19628-78-C-0002.

16. Abstract

The Aircraft Reply and Interference Environment Simulator (ARIES) makes possible the performance assessment of a Discrete Address Beacon System (DABS) sensor under its specified maximum aircraft load. To do this ARIES operates upon a taped traffic model to generate simulated aircraft replies and fruit, feeding them to the sensor at RF. Support documentation for ARIES, of which this is the second volume, consists of:

Volume 1: Principles of Operation
Volume 2: Appendices to the Principles of Operation
Volume 3: Programmer's Manual

The Appendices to the Principles of Operation is comprised of a set of technical articles providing (a) explanations of design and programming aspects of the ARIES system not covered in the main volume, (b) detailed data format and data structure definitions, (c) detailed explanations of the meaning of ARIES error messages, and (d) an analysis of certain effects to be expected when more than one ARIES simulator are interconnected.

17. Key Words Discrete Address Beacon System (DABS) ARIES Fruit generation Simulation Air traffic models Secondary radar	18. Distribution Statement Document is available to the public through the National Technical Information Service, Springfield, Virginia 22151.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	
Unclassified	Unclassified	112	

Appendices to the Principles of Operation

This volume is comprised of a set of technical articles providing (a) explanations of design and programming aspects of the ARIES system not covered in the main volume, (b) detailed data format and data structure definitions, (c) detailed explanations of the meaning of ARIES error messages, and (d) an analysis of certain effects which may be expected when more than one ARIES simulators are interconnected to permit testing adjacent DABS sensors. Each appendix is independent of the other appendices, and each appendix is referenced at appropriate points within the parent volume.

The contents of these articles will be especially meaningful to those seeking to delve further into the reasons for certain ARIES design and operational features, and for those intending to alter or extend ARIES hardware or software to provide additional capability.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

APPENDIX A

PROGRAMMING THE CONTROLLER

A.1 Programming Steps and Mnemonics

Programming the controller consists of writing the source program using the defined mnemonics, assembling the source program using the ARIES microprocessor assembler, and punching the paper tape.

Machine instructions for the ARIES controller are single words, each resident in a programmable storage memory (PROM). There are seven classes of instructions: arithmetic, logical, data movement, shift, program control, register manipulation, and special function. The mnemonic for each instruction is described in Table A-1. Symbols used are as follow:

Symbol	Meaning
RA	RØ - R15: as specified in the A address field A_0, A_1, A_2, A_3 (A_0 = MSB)
RB	RØ - R15: as specified by the B address field B_0, B_1, B_2, B_3, (B_0 = MSB)
BA_i	Branch address field: as specified by BA_0 --- BA_8 (BA_0 = MSB)
D_i	Data field: as specified by $D_Ø$ --- D_{15} (D_0 = MSB)
D_8	PROM D_8
H_8	PROM H_8
E_8	PROM E_8
F_8	PROM F_8
G_8	PROM G_8
Q	Q Register

TABLE A-1

ARIES TARGET CONTROLLER MICROPROCESSOR MNEMONICS

ARITHMETIC INSTRUCTIONS:

MNEMONIC: ADD R_A R_B

DESCRIPTION: $R_A + R_B \longrightarrow R_B$

FORMAT:

			D8								H8									E8									F8								G8		
1	0	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	B_0	B_1	B_2	B_3	A_0	A_1	A_2	A_3

MNEMONIC: SUB R_A R_B

DESCRIPTION: $R_A - R_B \longrightarrow R_B$

FORMAT:

			D8								H8									E8									F8								G8		
1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	B_0	B_1	B_2	B_3	A_0	A_1	A_2	A_3

MNEMONIC: ADDQ R_B

DESCRIPTION: $Q + R_B \longrightarrow R_B$

FORMAT:

			D8								H8									E8									F8								G8		
1	0	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	B_0	B_1	B_2	B_3	1	1	1	1

TABLE A-1

(Continued)

ARIES TARGET CONTROLLER MICROPROCESSOR MNEMONICS

DATA MOVEMENT INSTRUCTIONS:

MNEMONIC: IN R_B

DESCRIPTION: INPUT TO REGISTER B FROM D-BUS

FORMAT:

D_8	H_8	E_8	F_8	G_8
1 0 1 1 1 1 1 1	1 1 1 0 1 1 0 1	0 0 1 1 1 1 1 1	1 1 1 1 1 1 1 1	B_0 B_1 B_2 B_3 1 1 1 1

MNEMONIC: OUT R_A

DESCRIPTION: OUTPUT TO Y-BUS FROM REGISTER A

FORMAT:

D_8	H_8	E_8	F_8	G_8
1 0 1 1 1 1 1 1	1 1 0 1 1 1 0 1	0 1 1 0 0 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 A_0 A_1 A_2 A_3

MNEMONIC: MOVE R_A R_B

DESCRIPTION: MOVE DATA FROM REGISTER A TO REGISTER B

FORMAT:

D_8	H_8	E_8	F_8	G_8
1 1 1 1 1 1 1 1	1 1 1 0 1 1 0 1	1 1 1 0 0 1 1 1	1 1 1 1 1 1 1 1	B_0 B_1 B_2 B_3 A_0 A_1 A_2 A_3

MNEMONIC: LCR R_A

DESCRIPTION: LOAD CONTROL REGISTER FROM REGISTER A

FORMAT:

D_8	H_8	E_8	F_8	G_8
1 0 1 1 1 1 1 1	1 1 0 1 1 1 0 1	1 0 1 0 0 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 A_0 A_1 A_2 A_3

TABLE A-1

(Continued)

ARIES TARGET CONTROLLER MICROPROCESSOR MNEMONICS

DATA MOVEMENT INSTRUCTIONS (Continued):

MNEMONIC: LCRQ

DESCRIPTION: LOAD CONTROL REGISTER FROM Q REGISTER

	D_8								H_8								E_8								F_8								G_8							
FORMAT:	1	0	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

MNEMONIC: INQ

DESCRIPTION: INPUT TO Q REGISTER FROM D-BUS

	D_8								H_8								E_8								F_8								G_8							
FORMAT:	1	0	1	1	1	1	1	1	1	1	0	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TABLE A-1

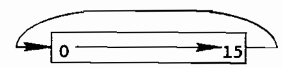
(Continued)

ARIES TARGET CONTROLLER MICROPROCESSOR MNEMONICS

SHIFT INSTRUCTIONS:

MNEMONIC: RTR R_B

DESCRIPTION: ROTATE REGISTER B ONE BIT TO THE RIGHT

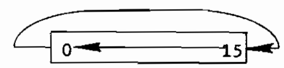


FORMAT:

D_8	H_8	E_8	F_8	G_8
1 1 1 1 1 1 0 1	1 1 0 1 1 1 1 1	0 1 0 1 1 1 1 1	1 1 1 1 1 1 1 1	B_0 B_1 B_2 B_3 1 1 1 1

MNEMONIC: RTL R_B

DESCRIPTION: ROTATE REGISTER B ONE BIT TO THE LEFT

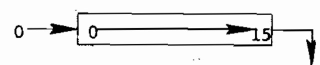


FORMAT:

D_8	H_8	E_8	F_8	G_8
1 1 1 1 1 1 0 1	1 1 1 1 1 1 1 1	0 1 0 1 1 1 1 1	1 1 1 1 1 1 1 1	B_0 B_1 B_2 B_3 1 1 1 1

MNEMONIC: SRLL R_B

DESCRIPTION: SHIFT REGISTER B RIGHT LOGICAL LOW

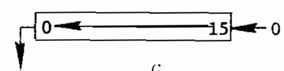


FORMAT:

D_8	H_8	E_8	F_8	G_8
1 1 1 1 1 1 0 0	1 1 0 1 1 1 1 1	0 1 0 1 1 1 1 1	1 1 1 1 1 1 1 1	B_0 B_1 B_2 B_3 1 1 1 1

MNEMONIC: SLLL R_B

DESCRIPTION: SHIFT REGISTER B LEFT LOGICAL LOW



FORMAT:

D_8	H_8	E_8	F_8	G_8
1 1 1 1 1 1 0 0	1 1 1 1 1 1 1 1	0 1 0 1 1 1 1 1	1 1 1 1 1 1 1 1	B_0 B_1 B_2 B_3 1 1 1 1

TABLE A-1

(Continued)

ARIES TARGET CONTROLLER MICROPROCESSOR MNEMONICS

SHIFT INSTRUCTIONS (Continued):

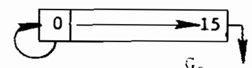
MNEMONIC: SLLH R_B

DESCRIPTION: SHIFT REGISTER B LEFT LOGICAL HIGH

	D_8								H_8								E_8								F_8								G_8							
FORMAT:	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	B_0	B_1	B_2	B_3	1	1	1	1

MNEMONIC: SRA R_B

DESCRIPTION: SHIFT REGISTER B RIGHT ARITHMETIC



	D_8								H_8								E_8								F_8								G_8							
FORMAT:	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	B_0	B_1	B_2	B_3	1	1	1	1

TABLE A-1

(Continued)

ARIES TARGET CONTROLLER MICROPROCESSOR MNEMONICS

PROGRAM CONTROL INSTRUCTIONS:

MNEMONIC: JMP BAi

DESCRIPTION: JMP TO ADDRESS SPECIFIED BY BAi

FORMAT:

			D_8								H_8								E_8								F_8								G_8				
1	0	1	1	1	1	1	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	BA_0	BA_1	BA_2	BA_3	BA_4	BA_5	BA_6	BA_7	BA_8	1	1	1	1	1	1	1	1

MNEMONIC: BRZ BAi

DESCRIPTION: BRANCH IF ZERO

FORMAT:

			D_8								H_8								E_8								F_8								G_8				
1	0	1	1	0	0	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	BA_0	BA_1	BA_2	BA_3	BA_4	BA_5	BA_6	BA_7	BA_8	1	1	1	1	1	1	1	1

MNEMONIC: BNZ BAi

DESCRIPTION: BRANCH IF NOT ZERO

FORMAT:

			D_8								H_8								E_8								F_8								G_8				
1	0	1	0	0	0	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	BA_0	BA_1	BA_2	BA_3	BA_4	BA_5	BA_6	BA_7	BA_8	1	1	1	1	1	1	1	1

MNEMONIC: BRN BAi

DESCRIPTION: BRANCH IF NEGATIVE

FORMAT:

			D_8								H_8								E_8								F_8								G_8				
1	0	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	BA_0	BA_1	BA_2	BA_3	BA_4	BA_5	BA_6	BA_7	BA_8	1	1	1	1	1	1	1	1

TABLE A-1

(Continued)

ARIES TARGET CONTROLLER MICROPROCESSOR MNEMONICS

PROGRAM CONTROL INSTRUCTIONS (Continued):

MNEMONIC: BNN BAi

DESCRIPTION: BRANCH IF NOT NEGATIVE

FORMAT:

D_8								H_8								E_8								F_8								G_8							
1	0	1	0	1	0	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	BA_0	BA_1	BA_2	BA_3	BA_4	BA_5	BA_6	BA_7	BA_8	1	1	1	1	1	1	1	1

MNEMONIC: BRC BAi

DESCRIPTION: BRANCH IF CARRY

FORMAT:

D_8								H_8								E_8								F_8								G_8							
1	0	1	0	0	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	BA_0	BA_1	BA_2	BA_3	BA_4	BA_5	BA_6	BA_7	BA_8	1	1	1	1	1	1	1	1

MNEMONIC: BRO BAi

DESCRIPTION: BRANCH IF OVERFLOW

FORMAT:

D_8								H_8								E_8								F_8								G_8							
1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	BA_0	BA_1	BA_2	BA_3	BA_4	BA_5	BA_6	BA_7	BA_8	1	1	1	1	1	1	1	1

MNEMONIC: JSR BAi

DESCRIPTION: JUMP TO SUBROUTINE

FORMAT:

D_8								H_8								E_8								F_8								G_8							
1	0	1	1	1	1	1	1	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	BA_0	BA_1	BA_2	BA_3	BA_4	BA_5	BA_6	BA_7	BA_8	1	1	1	1	1	1	1	1

TABLE A-1

(Continued)

ARIES TARGET CONTROLLER MICROPROCESSOR MNEMONICS

PROGRAM CONTROL INSTRUCTIONS (Continued):

MNEMONIC: RTN

DESCRIPTION: RETURN FROM SUBROUTINE

FORMAT:

			D_8									H_8								E_8								F_8								G_8			
1	0	0	0	1	1	1	1	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

MNEMONIC: JAR

DESCRIPTION: JUMP TO ADDRESS IN ADDRESS REGISTER

FORMAT:

			D_8									H_8								E_8								F_8								G_8			
1	1	1	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TABLE A-1

(Continued)

ARIES TARGET CONTROLLER MICROPROCESSOR MNEMONICS

LOGICAL INSTRUCTIONS:

MNEMONIC: AND R_A R_B

DESCRIPTION: $R_A \wedge R_B \longrightarrow R_B$

FORMAT:

			D_8								H_8								E_8								F_8								G_8				
1	0	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	B_0	B_1	B_2	B_3	A_0	A_1	A_2	A_3

MNEMONIC: OR R_A R_B

DESCRIPTION: $R_A \vee R_B \longrightarrow R_B$

FORMAT:

			D_8								H_8								E_8								F_8								G_8				
1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	B_0	B_1	B_2	B_3	A_0	A_1	A_2	A_3

MNEMONIC: XOR R_A R_B

DESCRIPTION: $R_A \oplus R_B \longrightarrow R_B$

FORMAT:

			D_8								H_8								E_8								F_8								G_8				
1	0	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	B_0	B_1	B_2	B_3	A_0	A_1	A_2	A_3

MNEMONIC: XNOR R_A R_B

DESCRIPTION: $\overline{R_A \oplus R_B} \longrightarrow R_B$

FORMAT:

			D_8								H_8								E_8								F_8								G_8				
1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	B_0	B_1	B_2	B_3	A_0	A_1	A_2	A_3

TABLE A-1

(Continued)

ARIES TARGET CONTROLLER MICROPROCESSOR MNEMONICS

LOGICAL INSTRUCTIONS (Continued):

MNEMONIC: COM R_B

DESCRIPTION: COMPLEMENT REGISTER B

FORMAT:

				D_8								H_8								E_8								F_8								G_8			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	B_0	B_1	B_2	B_3	1	1	1	1

MNEMONIC: ANDIQ

DESCRIPTION: AND IMMEDIATE Q, $Q \wedge Di \longrightarrow Q$

FORMAT:

				D_8								H_8								E_8								F_8								G_8			
1	D_0	1	1	1	1	D_{14}	D_{15}	1	1	0	0	1	1	0	0	1	0	1	1	0	0	1	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	1	1	1	1	D_1	D_2	D_3	D_4

MNEMONIC: ORIQ

DESCRIPTION: OR IMMEDIATE Q, $Q \vee Di \longrightarrow Q$

FORMAT:

				D_8								H_8								E_8								F_8								G_8			
1	D_0	1	1	1	1	D_{14}	D_{15}	1	1	0	1	1	1	0	0	0	1	1	1	0	0	1	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	1	1	1	1	D_1	D_2	D_3	D_4

MNEMONIC: XORIQ

DESCRIPTION: EXCLUSIVE OR IMMEDIATE Q, $Q \oplus Di \longrightarrow Q$

FORMAT:

				D_8								H_8								E_8								F_8								G_8			
1	D_0	1	1	1	1	D_{14}	D_{15}	1	1	0	0	1	1	0	0	1	1	1	1	0	0	1	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	1	1	1	1	D_1	D_2	D_3	D_4

TABLE A-1

(Continued)

ARIES TARGET CONTROLLER MICROPROCESSOR MNEMONICS

REGISTER MANIPULATION INSTRUCTIONS:

MNEMONIC: LIM D R_B

DESCRIPTION: LOAD REGISTER B WITH DATA Di

FORMAT:

D_8								H_8								E_8								F_8								G_8							
1	D_0	1	1	1	1	D_{14}	D_{15}	1	1	1	0	1	1	0	1	1	1	1	1	1	0	1	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	B_0	B_1	B_2	B_3	D_1	D_2	D_3	D_4

MNEMONIC: CLR R_B

DESCRIPTION: CLEAR REGISTER B

FORMAT:

D_8								H_8								E_8								F_8								G_8							
1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	B_0	B_1	B_2	B_3	1	1	1	1

MNEMONIC: DEC R_B

DESCRIPTION: DECREMENT REGISTER B

FORMAT:

D_8								H_8								E_8								F_8								G_8							
1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	B_0	B_1	B_2	B_3	1	1	1	1

MNEMONIC: INC R_B

DESCRIPTION: INCREMENT REGISTER B

FORMAT:

D_8								H_8								E_8								F_8								G_8							
1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	B_0	B_1	B_2	B_3	1	1	1	1

TABLE A-1

(Continued)

ARIES TARGET CONTROLLER MICROPROCESSOR MNEMONICS

REGISTER MANIPULATION INSTRUCTIONS:

MNEMONIC: INCQ

DESCRIPTION: INCREMENT REGISTER Q

FORMAT:

D_8								H_8								E_8								F_8								G_8							
1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

MNEMONIC: DECQ

DESCRIPTION: DECREMENT REGISTER Q

FORMAT:

D_8								H_8								E_8								F_8								G_8							
1	0	1	1	1	1	1	1	1	1	0	1	1	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TABLE A-1

(Continued)

ARIES TARGET CONTROLLER MICROPROCESSOR MNEMONICS

SPECIAL FUNCTION INSTRUCTIONS:

MNEMONIC: RSTCR

DESCRIPTION: CLEAR CONTROL REGISTER

	D_8	H_8	E_8	F_8	G_8
FORMAT:	1 0 1 1 1 1 1 1	1 1 0 1 1 1 0 1	1 0 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1

MNEMONIC: LIMCR Di

DESCRIPTION: LOAD CONTROL REGISTER WITH DATA Di

	D_8	H_8	E_8	F_8	G_8
FORMAT:	1 D_0 1 1 1 1 D_{14} D_{15}	1 1 0 1 1 1 0 1	0 1 1 1 1 0 1 D_5	D_6 D_7 D_8 D_9 D_{10} D_{11} D_{12} D_{13}	1 1 1 1 D_1 D_2 D_3 D_4

MNEMONIC: NOP

DESCRIPTION: NO OPERATION

	D_8	H_8	E_8	F_8	G_8
FORMAT:	1 1 1 1 1 1 1 1	1 1 0 0 1 1 0 1	1 1 0 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1

Initially, the paper tape produced by the assembler can be used to load a ROM simulator. This way the source program can be debugged before it is permanently programmed in the ROM.

A.2 Using the ARIES-Microprocessor Assembler

The ARIES-Microprocessor Assembler (hereafter simply called "assembler") is a set of programs written for the (Eclipse) computer which will translate a set of mnemonics into the appropriate bit patterns to execute the desired instruction on the ARIES microprocessor. The assembly process consists of (1) writing the input program, (2) assembling the input program, and (3) punching the paper tape.

1. Writing the Input Program

The input program, consisting of ARIES microprocessor mnemonics, pseudo-operations, and macro commands (readers not familiar with the Nova macro commands and pseduo cperations should refer to the D.G. Nova macro assembler manual), is entered into a file using the editor. All numbers are assumed to be octal. The 16 registers in the ARIES microprocessor can be represented by the names "R0, R1, ... R15". Some special instructions at the beginning and end of the file are needed to make the tape punching program work properly.

Start of Program

.TITL File

.RDXO 16

This will make the listing produced during assembly print in hexadecimal.

.LOC 0

.END-START

.LOC _______ (any desired starting address other than zero)

START: first instruction

.....

..... desired set of instructions

.....

Last instruction

```
END: .END
```

End of Program

The purpose of the .LOC instructions and the END-START is to put a count of the number of words in the program at location zero. This count will be used by the punching program later. The label START is also used by the branch and jump commands in the ARIES microprocessor instruction set to generate relative addresses.

2. Assembling the Input Program

Assuming that the input program resides in file FILE.SR, then the following command will invoke the assembler.

```
MAC FILE.SR $LPT/L
```

This will print a listing on the line printer. Error comments (if any) will appear on the console as well as the listing. If no listing is desired, or no printer is available, then omit the $LPT from the command line.

If the assembly has been completed satisfactorily, then a load file must be made. This file will be used by the punching program. A load file is produced by the following command:

```
RLDR/C FILE
```

This will produce a core-image file on disk with the name FILE.SV. It is this file which will be used by the punching program.

3. Punching the Paper Tape

The punching and reformatting is done by a FORTRAN program in a file called MIKES and a subroutine in a file called TOUT. The programs use the core-image file produced during Step 2 as input, and produce a paper-tape image file called PTAPE as output. The punching program may be invoked simply by using the command:

```
MIKES
```

The program will first ask for the input file name. This is the file whose load image was produced at the end of Step 3. The name will always have the appendage ".SV". In the case of this example, the input file name is

FILE.SV

The program will now begin to reformat the contents of the file in five passes. A message will be printed out at the end of each pass. Also printed out is the length of the file in words. This is the value that was placed at location zero. At the completion of pass five, the file PTAPE contains the image of the desired paper tape. Its contents should then be transferred to the punch by the following command.

XFER PTAPE $PTP

The paper tape produced by the assembler is in the SMS Format (Scientific Micro-System). (This format is compatible with the Data-I/O ROM Programmer and the SMS ROM Simulator) and is suitable for programming the INTEL 3624 PROM. The contents of the paper tape is in a sequence (blocks of punched holes spaced by unpunched tape) corresponding to the order of the PROMS as follows:

D_8 H_8 E_8 F_8 G_8.

A.3 Sample Programs

Program "A" and Program "B" (Figs. A-1 and A-2) included in this appendix serve as illustrations of ARIES microprocessor program format, mnemonics, and NOVA assembler pseudo-operations and commands. Program A is used to sequence replies from the CPU (Eclipse) to three Controlled ARIES Targets (CAT's) and Program B is used to control the Fruit ARIES Target (FAT) operational modes.

PROGRAM A (Fig. A-1)

This program as resident in the CAT-Controller ROM's, is used to sequence replies from the CPU (Eclipse) to three Controlled ARIES Targets (CAT's). The program will initialize when an IORST is executed. (IORST is an Eclipse instruction which can be executed manually from the CPU front panel switch or under program control). Upon initialization the controller will reset all three CAT's and proceed to sequence replies to them in CAT-1, CAT-2, CAT-3 order. Two conditions must be met before a reply is transferred: a reply must be available from the interface buffer, and the CAT must be ready to accept one. (If a CAT fails to accept a reply within 5 ms, a resetting bit will be set by the Eclipse to inform the controller to re-initialize the program.) The number of words transferred in a reply depends upon the type: four for ATCRBS and ten for DABS. After each reply is sent, the reply counter in the interface will be decremented. Please refer to the ARIES Programmer's Manual for further details. Table A-2 is a list of symbols and their meanings as used in this program.

PROGRAM B (Fig. A-2)

This program, as resident in the FAT-controller ROMs, provides four modes of operation for the FAT-controller: 1) normal, 2) FAT diagnostic, 3) RPG diagnostic, and 4) loop test. As in the CAT controller, the FAT controller also initializes itself when an IORST is executed from the CPU. The initialization process involves clearing all three FAT's and resetting the RPG with initial fruit parameters. (Fixed code = 1200, mainbeam to sidelobe ratio = 50%, 50% fixed code, fruit rate = 2000 fruit per sec).

The mode of operation can be selected by the CPU by specifying two mode bits in the interface:

1 1 Normal

1 0 RPG Diagnostic

0 1 FAT Diagnostic

0 0 Interface Loop Test

If the mode selected by the CPU is normal (BEGIN), the FAT controller will begin sequencing replies from the RPG to each of the FAT's in a round robin fashion. Also, periodically it will update the RPG with new fruit parameters when they are provided by the CPU.

Before each reply is sent to the FAT, the controller will examine the delay to trigger time and the ATCRBS code portion of the reply. If the sum of the current and the two previous delay to trigger times is less than 28 μs, the current time will be modified to meet the requirement. This is necessary since a FAT cannot start another reply until the previous reply is over. If the ATCRBS code contains all '1's the controller will replace it with a code specified by the first word of the RPG parameter. Furthermore, if any FAT remains busy for more than 5 msec (i.e., does not respond to a poll) an interrupt will be sent to the CPU to signal the problem. Note that each FAT should reply within 4.096 ms since the clock is wrapped around at that point.

TABLE A-2

PROGRAM A SYMBOL DEFINITIONS

START:	The address label for the first instruction in the program. It must be used since all branch instructions use it to calculate the proper address.
RSTC 1, 2, 3	Control word for resetting each CAT.
BUSY 1, 2, 3	Control word for checking the 'busy' bit of each CAT.
CAT 14, 24, 34	Control word for loading reply words 1, 2, 3, 4, in CAT -1, -2. -3.
CAT 15, 25, 35	Control word for loading reply words 5, 6, 7, 8, 9, 10 in CAT -1, -2, -3.
BEGIN:, CAT:	A segment of the program that actually sequences replies from the CAT controller interface to each CAT.
STATUS	Control word for checking the status in the interface.
REPLY	Mask for 'reply available' bit from the interface.
BUSY	Mask for 'busy' bit in the CAT.
RESET	Mask for 'reset' bit in the interface.
CHCK:	A segment of the program that checks the 'busy' bit in the CAT and 'reset' bit in the interface.
CLRST	Control word for clearing the 'reset' bit in the interface.
RD14	Control word for reading reply word 1-4.
TYPE	Mask for the 'type' bits of the third reply word.
RCD	Control word for decrementing the reply counter in the interface.
RD510	Control word for reading the reply words 1-5.
END:	Address label for the last instruction in the program. Normally the instruction would be the Macro Operator .END, which is used to terminate the source program. However, if the label for the numerical constants are used, the first label should be inserted here, followed by others if any, then finally the Macro Operator .END.

```
        .TITL CAT
        .RDXO 16
        .LOC Ø
        END-START
        .LOC 1ØØØØ
START:  LIMCR RSTC1
        LIMCR RSTC2    ;RESET CAT-1,CAT-2,CAT-3
        LIMCR RSTC3


BEGIN:  LIM BUSY1,R15    ;CHECK TO SEE IF CAT-1 IS BUSY
        LIM CAT14,R14
        LIM CAT15,R13
        JSR CAT


        LIM BUSY2,R15    ;CHECK TO SEE IF CAT-2 IS BUSY
        LIM CAT24,R14
        LIM CAT25,R13
        JSR CAT


        LIM BUSY3,R15    ;CHECK TO SEE IF CAT-3 IS BUSY
        LIM CAT34,R14
        LIM CAT35,R13
        JSR CAT


        JMP BEGIN        ;REPEAT


CAT:    LIMCR STATUS   ;CHECK FOR REPLY IN INTERFACE BUFFER
        NOP           ;LOOP UNTIL REPLY IS AVAILABLE
        INQ
        ANDIQ REPLY
        BNZ . -2
CHCK:   LCR R15
        NOP
        NOP
        INQ
        ANDIQ BUSY
        BNZ . +9.        ;LOOP UNTIL THE CAT IS NOT BUSY,
        LIMCR STATUS   ;AND CHECK FOR RESETTING BIT
        NOP
        INQ
        ANDIQ RESET
        BNZ CHCK
        LIMCR CLRST
        RSTCR
        JMP START


        LIMCR RD14        ;READ 4 WORDS FROM THE MEMORY
        NOP
        NOP
        IN RØ
        IN R1
        IN R2
        IN R3
        RSTCR
```

Fig. A-1. Program A.

```
        LCR R14          ;LOAD THE CAT WITH WORDS 1-4
        OUT RØ
        OUT R1
        OUT R2
        OUT R3
        RSTCR

        LIM TYPE,R12
        AND R2,R12           ;CHECK FOR REPLY TYPE.
        BRZ . +4             ;IF IT IS A DABS REPLY GET 6 MORE
                             ;WORDS FROM THE BUFFER

        LIMCR RCD            ;DECREMENT REPLY COUNTER
        RSTCR
        RTN

        LIMCR RD51Ø
        NOP
        NOP
        IN R4
        IN R5
        IN R6
        IN R7
        IN R8
        IN R9
        RSTCR
        LCR R13           ;LOAD CAT WITH WORDS 5-1Ø
        OUT R4
        OUT R5
        OUT R6
        OUT R7
        OUT R8
        OUT R9
        RSTCR
        LIMCR RCD       ;DECREMENT REPLY COUNTER
        RSTCR
        RTN

END:    RSTC1=2Ø2
        RSTC2=4Ø2
        RSTC3=1ØØ2

        BUSY1=2Ø1
        BUSY2=4Ø1
        BUSY3=1ØØ1

        CAT14=21Ø
        CAT24=41Ø
        CAT34=1Ø1Ø

        CAT15=22Ø
        CAT25=42Ø
        CAT35=1Ø2Ø

        STATUS=5ØØØ
        REPLY=1

        BUSY=1
        RD14=42ØØ
        RD51Ø=44ØØ
```

Fig. A-1. (Con't).

```
TYPE=1
RCD=6ØØØ
RESET=2
CLRST=41ØØ
.END
```

Fig. A-1. (Con't).

```
          LCR R14          ;LOAD THE CAT WITH WORDS 1-4
          OUT RØ
          OUT R1
          OUT R2
          OUT R3
          RSTCR

          LIM TYPE,R12
          AND R2,R12           ;CHECK FOR REPLY TYPE.
          BRZ . +4             ;IF IT IS A DABS REPLY GET 6 MORE
                               ;WORDS FROM THE BUFFER

          LIMCR RCD            ;DECREMENT REPLY COUNTER
          RSTCR
          RTN

          LIMCR RD51Ø
          NOP
          NOP
          IN R4
          IN R5
          IN R6
          IN R7
          IN R8
          IN R9
          RSTCR
          LCR R13           ;LOAD CAT WITH WORDS 5-1Ø
          OUT R4
          OUT R5
          OUT R6
          OUT R7
          OUT R8
          OUT R9
          RSTCR
          LIMCR RCD       ;DECREMENT REPLY COUNTER
          RSTCR
          RTN

END:      RSTC1=2Ø2
          RSTC2=4Ø2
          RSTC3=1ØØ2

          BUSY1=2Ø1
          BUSY2=4Ø1
          BUSY3=1ØØ1

          CAT14=21Ø
          CAT24=41Ø
          CAT34=1Ø1Ø

          CAT15=22Ø
          CAT25=42Ø
          CAT35=1Ø2Ø

          STATUS=5ØØØ
          REPLY=1

          BUSY=1
          RD14=42ØØ
          RD51Ø=44ØØ
```

Fig. A-1. (Con't).

```
TYPE=1
RCD=6ØØØ
RESET=2
CLRST=41ØØ
.END
```

Fig. A-1. (Con't).

In the FAT diagnostic mode (FATTST), the controller will do exactly the same thing as in the normal mode, except the replies now come from the CPU instead of from the RPG. This way a known reply can be sent and verified (see the section on the Self Test Unit in the Programmer's Manual).

In order to check out the characteristics of the RPG, the RPG diagnostic mode (RPGTST) is provided. In this mode, the FAT controller will send replies generated by the RPG to the CPU, thus providing a means of verifying the data generated by the RPG against the software simulated data.

Lastly, the loop test (LOOPTST) mode provides a means to check out the data path and memories of the FAT Controller interface. However, the controller does not participate in this test. (It does constantly monitor the mode bits).

Table A-3 is a list of symbols and their meanings as used in this program.

TABLE A-3

PROGRAM B SYMBOL DEFINITIONS

SYMBOL	MEANING
OUTFF	Mask for output buffer status.
LOAD:	Segment of the program that sequences replies to FAT's.
RDIO	Control word for reading the interface.
RSTRPG	Control word for checking the busy bit of each FAT.
BUSY 1, 2, 3	Control word for loading FAT-1, -2, -3 respectively.
BEGIN:	Segment of the program for executing the normal mode of operation.
RSTF 1, 2, 3	Control word for resetting each FAT.
LDRPG	Control word for loading the RPG with new parameters.
STATUS	Control word for reading the status from the interface.
BA	Mask word for reading the mode.
DIAGNOSTIC:	A segment of program that is used to determine the mode bit and execute the appropriate task accordingly.
. (period)	NOVA Macro assembler-pseudo operator for the current address (i.e., .+2 means current address plus two).
TIMER	Constant (7640_8) for 5 msec time out.
BUSY	Mask for busy bit in interface.
FAULT:	Segment of the program that generates interrupt to the CPU (Eclipse).

TABLE A-3 (Continued)

SYMBOL	MEANING
RDRPG	Control word for reading the RPG.
DELAA	Constant $(177710)_8$ for comparison with delay of trigger time.
DELAB	Constant $(70)_8$ for 3.5 us.
INTR	Control word for generating an interrupt.
START:	Address label for the first instruction of the program, it must be used in order for all branch instructions to execute properly.
HERE:	JMP Table.
LOOPTST:	A segment of the program that executes the loop test mode.
FATTST:	A segment of the program that executes that FAT diagnostic mode.
RPGTST:	A segment of the program that executes the RPG diagnostic mode.
RPG:	A segment of the program in RPGTST that takes data from the Random Process Generator and transfers them to the CPU.
INFF	Mask for the input buffer status.
WRIO	Control word for writing the interface.
END:	Address label for the last instruction in the program. Normally the instruction would be the Macro operator .END, which is used to terminate the source program. However, if labels for the numerical constants are used, the first label should be inserted here, followed by others,and finally .END.

```
        .TITL FAT
        .RDXO 16
        .LOC Ø
        END-START
        .LOC  1ØØØØ
START:  LIMCR RSTF1         ;RESET FAT-1,FAT-2,FAT-3
        LIMCR RSTF2
        LIMCR RSTF3

        LIM 12ØØ42,R4         ;FIXED CODE=12ØØ
        LIM Ø4ØØ1Ø,R5         ;M/S=5Ø%,5Ø% FIXED CODE
        LIM ØØ4ØØØ,R6         ;FR=2ØØØ FRUITS PER SEC

        LIM ATCRB,R11              ;ATCRB=-35US

        LIMCR LDRPG
        RSTCR                 ;INITIALIZE RPG
        OUT R4
        OUT R5
        OUT R6

        LIMCR RSTRPG           ;RESET RPG
        LIM ØØØ144,R7         ;DELAY 5Ø US
        DEC R7
        BNZ . -1

BEGIN:  LIMCR STATUS
        NOP
        INQ
        ANDIQ BA                   ;IF BA=ØØ,CONTINUE;ELSE GOTO DIAGNOSTIC ROUTINE
        BRZ . +2
        JMP DIAGNOSTIC
        INQ
        ANDIQ OUTFF                ;IF OUTFF=1,GET 4 FRUIT PARAMETERS
        BRZ LOAD
        LIMCR RDIO                 ;FROM ECLIPSE AND LOAD THEM INTO THE RPG
        RSTCR
        NOP                        ;ELSE GOTO LOAD
        IN R4
        IN R5
        IN R6
        IN R7
        LIMCR LDRPG
        RSTCR
        OUT R4
        OUT R5
        OUT R6

        LIMCR RSTRPG          ;RESET RPG
        LIM ØØØ144,R7         ;DELAY 5Ø US
        DEC R7
        BNZ . -1
```

Fig. A-2. Program B.

```
LOAD:   LIM BUSY1,R15          ;LOAD FAT1 A FRUIT REPLY FROM RPG
        LIM FAT1,R14
        JSR FAT


        LIM BUSY2,R15          ;LOAD FAT2 A FRUIT REPLY FROM RPG
        LIM FAT2,R14
        JSR FAT


        LIM BUSY3,R15          ;LOAD FAT3 A FRUIT REPLY FROM RPG
        LIM FAT3,R14
        JSR FAT


        JMP BEGIN             ;REPEAT BEGIN



FAT:    LCR R15              ; SUBROUTINE FAT---THIS ROUTINE CONTINUOUSLY POLL
        LIM TIMER,R13          ;THE FAT TO SEE IF IT CAN ACCEPT ANOTHER FRUIT REPLY
        NOP
        INQ                     ;WHEN THE FAT IS NOT BUSY,A REPLY IS LOAD INTO
        ANDIQ BUSY             ;THE FAT FROM THE RPG.HOWEVER IF THE FAT FAILED
        BNZ . +4               ;TO RESPONDSE WITHIN 5 MS, A INTERRUPT WILL BE SENT
        DEC R13                ;TO THE ECLIPSE TO SIGNIFY THE PROBLEM.
        BRZ FAULT
        JMP . -5
        LIMCR RDRPG
        RSTCR
        NOP
        NOP
        NOP
        NOP
        IN RØ
        IN R1
        IN R2
        IN R3




        LIM 177376,R7         ;IF THE CODE IS EQUALS TO ALL 1'S
        SUB R3,R7             ;REPLACE R3 WITH A FIXED CODE
        BNZ . +2
        MOVE R4,R3


        ADD R9,R1Ø          ;IF THE SUM (X) OF THE CURRENT TRIGGER
        BRC ATC             ;TIME AND THE PREVIOUS TWO IS LESS THAN 28 US
        ADD RØ,R1Ø          ;THEN ADD (28US-X) TO THE CURRENT TIME
        BRC ATC
        MOVE R1Ø,R12
        ADD R11,R12            ;R11 CONTAINS THE VALUE OF -28US
        BRC ATC

        COM R12
        ADD R12,R1Ø
```

Fig. A-2. (Con't).

```
        MOVE R1Ø,RØ


ATC:    MOVE R9,R1Ø
        MOVE RØ,R9



        LCR R14
        OUT RØ
        OUT R1
        OUT R2
        OUT R3
        RSTCR
        RTN


FAULT:  LIMCR INTR               ;GENERATE A INTERRUPT, WHEN A FAT FAILED TO RESPONDSE
        RSTCR                     ;A POLL WITHIN 5 MS.
        JMP START


DIAGNOSTIC:     LIMCR STATUS     ;DIAGNOSTIC ROUTINE
        RSTCR
        IN R15
        LIM 17,R14
        AND R14,R15
        SRLL R15
        SRLL R15
        LIM 3,R14
        SUB R14,R15
        LIM ((HERE-START)/3),R12
        ADD R15,R12
        JAR
HERE:   JMP LOOPTST
        JMP FATTST
        JMP RPGTST
        JMP START


RPGTST: LIMCR STATUS             ;RPG DIAGNOSTIC-OUTPUT FROM THE RPG IS ACCUMULATED
        NOP                       ;TO CHECK FOR PROPER PATTERN.
        IN R15
        LIM 14,R14
        AND R14,R15
        LIM BA2,R14
        SUB R15,R14
        BRZ . +2
        JMP DIAGNOSTIC
        INQ                       ;IF BA=Ø1,THEN CONTINUE. ELSE  GOTO
        ANDIQ OUTFF              ;DIAGNOSTIC
        BRZ RPG                  ;IF OUTFF=1,UPDATE RPG WITH FRUIT PARAMETER
        LIMCR RDIO
        RSTCR
        NOP
        IN R4
        IN R5
        IN R6
        IN R7
        LIMCR LDRPG
        RSTCR
        OUT R4
        OUT R5
```

Fig. A-2. (Con't).

```
        OUT R6


        LIMCR RSTRPG        ;RESET RPG
        LIM ØØØ144,R7       ;DELAY 5Ø US
        DEC R7
        BNZ . -1


RPG:    LIMCR STATUS
        NOP              ;IF INFF=1,THEN GET A FRUIT REPLY FROM RPG
        INQ              ;AND WRITE IT INTO THE UP RAM,ELSE GOTO RPGTST
        ANDIQ INFF
        BRZ RPGTST
        LIMCR RDRPG
        RSTCR
        NOP
        NOP
        NOP
        NOP
        IN RØ
        IN R1
        IN R2
        IN R3
        LIMCR WRIO
        OUT RØ
        OUT R1
        OUT R2
        OUT R3
        RSTCR
        JMP RPGTST

LOOPTST:        JMP DIAGNOSTIC          ;LOOP TEST-CPU AND UP RAM ARE LOOP TESTED

FATTST: LIM BUSY1,R15           ;FAT DIAGNOSTIC-GENERATE FRUIT FROM ECLIPSE
        LIM FAT1,R14
        JSR GET
        LIM BUSY2,R15
        LIM FAT2,R14
        JSR GET
        LIM BUSY3,R15
        LIM FAT3,R14
        JSR GET
        JMP FATTST

GET:    LIMCR STATUS
        NOP
        IN R13
        LIM 14,R12
        AND R12,R13
        LIM BA3,R12
        SUB R13,R12
        BRZ . +2
        JMP DIAGNOSTIC
        INQ              ;IF BA=1Ø,CONT.ELSE EXIT
        ANDIQ OUTFF
        BRZ . -9.        ;REPEAT TESTING OUTFF AND BA BIT UNTIL OUTFF=1

        LIMCR RDIO
```

Fig. A-2. (Con't).

```
         RSTCR
         NOP
         IN RØ
         IN R1
         IN R2
         IN R3

         LIM DELAA,R13      ;IF DELAY OF TRIGGER IS LESS THAN 3.5US
         ADD RØ,R13         ;REPLACE IT WITH A 3.5US TRIGGER TIME
         BRC . +2
         LIM DELAB,RØ


         LCR R15
         LIM TIMER,R13
         NOP
         INQ
         ANDIQ BUSY
         BNZ . +4
         DEC R13
         BRZ FAULT
         JMP . -5
         LCR R14
         OUT RØ
         OUT R1
         OUT R2
         OUT R3
         RSTCR
         RTN

END:     STATUS=6ØØØ
BA1=ØØ
BA2=Ø4
BA3=1Ø
BA4=14
BA=14
INFF=1
OUTFF=2
RDIO=44ØØ
WRIO=5ØØØ
INTR=4ØØ1
TIMER=764Ø
LDRPG=11Ø
RDRPG=1Ø4
RSTRPG=1Ø2
BUSY=1
RSTF1=2Ø2
RSTF2=4Ø2
RSTF3=1ØØ2
FAT1=21Ø
FAT2=41Ø
FAT3=1Ø1Ø
BUSY1=2Ø1
BUSY2=4Ø1
BUSY3=1ØØ1
DELAA=17771Ø
DELAB=ØØØØ7Ø
ATCRB=17672Ø
         .END
```

Fig. A-2. (Con't).

APPENDIX B

DERIVATION OF POWER ROM CONTENTS

The fruit power distribution required of ARIES is specified as follows:

"The mainlobe fruit will have a power level referred to the sensor RF port determined as follows:

$$P_{m\ell} = -20-20 * \log r_{m\ell} \text{ dBm},$$

where $r_{m\ell}$ is random and uniformly distributed between 1 and 100,...

"The sidelobe fruit will have a power level referred to the sensor RF port determined as follows:

$$P_{s\ell} = -55-20 * \log r_{s\ell} \text{ dBm},$$

where $r_{s\ell}$ is random and uniformly distributed between 1 and 32. Sidelobe fruit is not generated at power levels below -85 dBm".

Two mechanisms are used in ARIES to meet these requirements. The random process generator (RPG) provides a 6 bit power field distributed according to the variable part of either the mainlobe or sidelobe power specifications. In addition, it provides a single bit which determines whether the reply is to be a mainlobe or sidelobe reply. The power field is used to control the main power attenuator of the analog portion of a fruit ARIES target (FAT), giving power levels in the range from -20 to -60 dBm. The mainbeam/sidelobe bit, if set, switches in an additional 35 dB of attenuation, to obtain the sidelobe power distribution.

To produce the desired power distribution, the RPG produces an 8 bit uniformly distributed random number, which is used in conjunction with the mainbeam/sidelobe bit to address a 512 word ROM. This ROM produces a 6 bit power control signal with the desired distribution. The first 256 locations contain the mainbeam distribution, and the upper 256 locations contain the sidelobe distribution. The rest of this appendix addresses the question of what the ROM content should be.

The ARIES power attenuator has 1 dB steps. Zero attenuation corresponds to a -20 dBm signal, referred to the DABS sensor's RF port. We will assume that the output power level will be set to integral power P whenever the desired real power $P_{m\ell}$ satisfies $P+.5 > P_{m\ell} > P-.5$ dBm. P = -20 will be used for $-20 > P_{m\ell} > -20.5$ and P = -60 will be used for $-59.5 > P_{m\ell} \geq 60$. To find the corresponding range brackets, we invert the equation for $P_{m\ell}$ to obtain

$$r_{m\ell} = 10^{-\left[1+\frac{P_{m\ell}}{20}\right]}$$

Therefore, the power should be set to P whenever

$$10^{-\left[1+\frac{P+.5}{20}\right]} < r_{m\ell} < 10^{-\left[1+\frac{P-.5}{20}\right]}$$

Since $r_{m\ell}$ is uniformly distributed between 1 and 100, the probability that it is in the above range is:

$$Pr_{m\ell}(P) = 1/99 \left[10^{-\left[1+\frac{P-.5}{20}\right]} - 10^{-\left[1+\frac{P+.5}{20}\right]}\right]$$

$$= 1/990 \left[10^{-\left[\frac{P-.5}{20}\right]} - 10^{-\left[\frac{P+.5}{20}\right]}\right] \qquad (-21 > P > -59)$$

Similar expressions apply for the end points P = -20 and P = -60. If the same derivation is applied to $r_{s\ell}$, a similar expression for $Pr_{s\ell}(P)$ may be obtained as follows:

$$Pr_{s\ell}(P) = \frac{1}{310} \left[10^{-\left[\frac{P-.5}{20}\right]} - 10^{-\left[\frac{P+.5}{20}\right]}\right] \qquad (-21<P<-49)$$

Similar equations also apply to the two end points P=-20 and P=-50.

Therefore, we wish to make the output of the ROM equal to P with the above probabilities. Since each ROM address is equally likely to be addressed, the way to do this is to have the number of ROM locations with P stored in them in each of the two distributions be proportional to $Pr_{m\ell}(P)$ or $Pr_{s\ell}(P)$. The proper distribution is selected by the mainbeam/sidelobe bit. Since there are 512 entries in the ROM, the number of entries containing P will therefore be $256\ Pr_{m\ell}(P) + 256\ Pr_{s\ell}(P)$. This number must be rounded up or down to obtain an integral number of entries.

APPENDIX C

GENERATION OF EXPONENTIALLY DISTRIBUTED FRUIT INTER-ARRIVAL TIMES

The statistics of fruit arrivals are assumed to correspond to those of a Poisson process, and therefore the inter-arrival time is a random variable with an exponential probability density. The probability that the interval between two successive events is between t and t + dt is therefore

$$P(t) = \lambda e^{-\lambda t}\, dt$$

where λ is the fruit rate. In ARIES, the fruit rate must be variable over the range from 1000 to 64000 fruit/second.

The Poisson Sequence Generator in the ARIES Random Process Generator (RPG) is responsible for controlling the inter-arrival times in accordance with the above expression. One of the inputs to this generator is related to $1/\lambda$. The output is a series of 16 bit exponentially distributed inter-arrival times, with the least significant bit corresponding to the least significant bit of the ARIES range clock, 62.5 nsec.

The approach taken is to approximate the exponential distribution by a step function which can be generated by combining the output of several random number generators with uniform distributions. Such a step function approximation is shown in Fig. C-1, with only a small number of steps for the sake of clarity. In the actual implementation, 1024 steps are used.

To generate an exponentially distributed random number, a uniform number n between 1 and N is first generated. This is used to obtain t_n from a table (implemented as a read-only memory). This essentially corresponds to picking one of the horizontal strips with probability 1/N, which matches the probability that each strip represents.

Next, another uniform random number generator is used to obtain a number between 0 and t_n. The probability of picking a given number is $1/t_n$ (the t_n are integers), and the overall probability is $1/N.t_n$, representing a small area of probability such as the filled in rectangle in Fig. C-1. The total probability of picking any t corresponds to the sum of all such rectangles in the column for t, and can be seen to be the desired step function.

To calculate the t_n, an expression for the area of each strip is set equal to 1/N, and the resulting equality solved for t_{n-1} in terms of t_n.

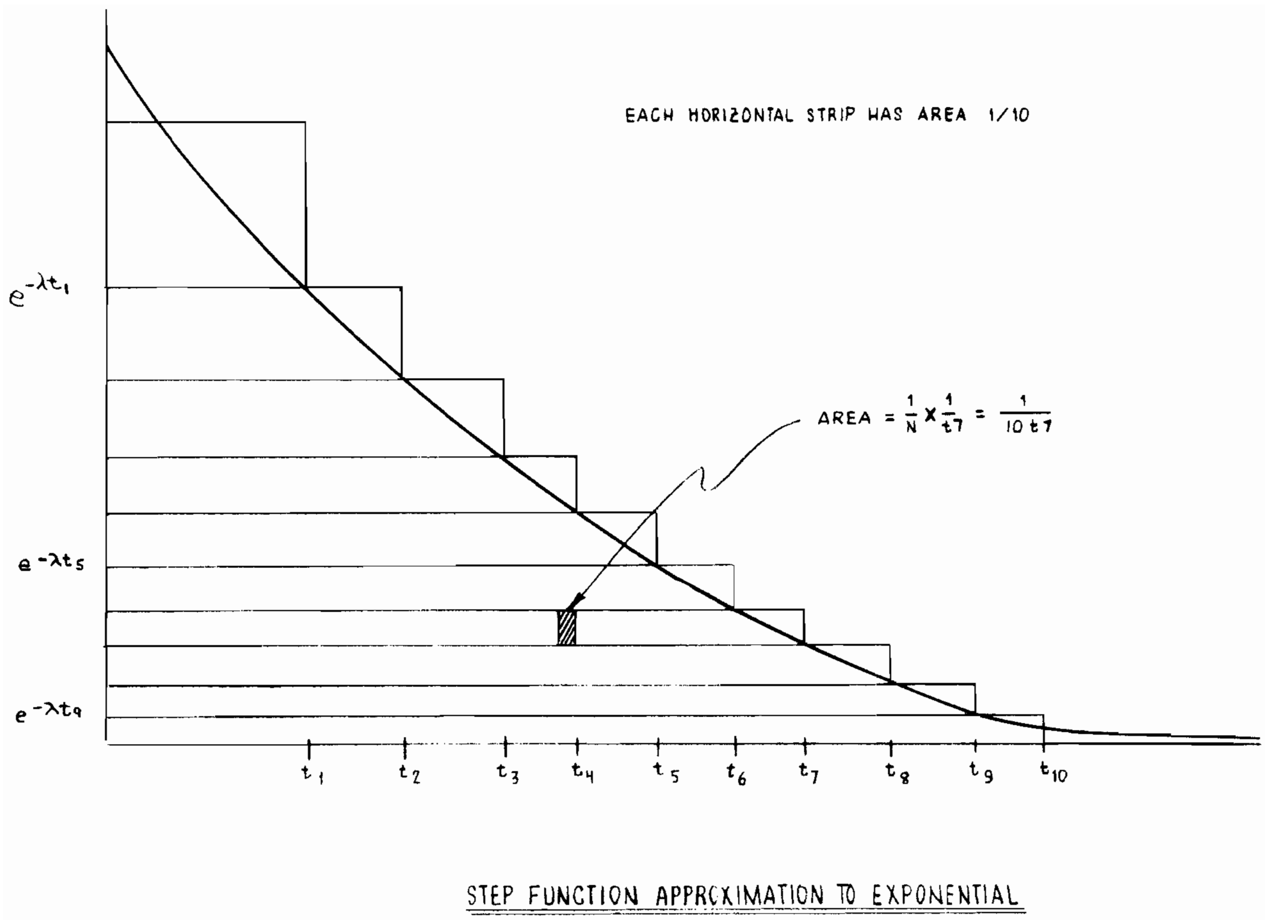


Fig.C-1. Step function approximation to exponential.

$$t_n (\lambda \exp(-\lambda t_{n-1}) - \lambda \exp(-\lambda t_n)) = 1/N$$

$$t_{n-1} = -\frac{1}{\lambda} \ell n \left[\frac{1}{N\lambda t_n} + \exp(-\lambda t_n)\right]$$

The value t_N is set equal to the maximum value that can be stored in the lookup table, and then the above recursion is used to generate the other entries. To preserve numerical accuracy, all calculations must be performed in floating point or double precision and the resulting t_n then converted to integers of the desired size.

To provide for varying λ, the t_n stored in the lookup table are chosen for the maximum fruit rate, λ_M, to be generated. Then, after some t_n is obtained, it is multiplied by λ_M/λ, where $\lambda < \lambda_M$ is the desired fruit rate. Then a uniformly distributed random number between 0 and $t_n.\lambda_M/\lambda$ is generated.

This essentially corresponds to dilating all the inter-arrival times by the factor λ_M/λ. To see that the $t'_n = t_n.\lambda_M/\lambda$ are the correct values that would have been stored in a lookup table generated specifically for the desired λ, substitute λ and $t'_n = t_n.\lambda_M/\lambda$ in the above iteration. Then:

$$\lambda t'_n = \lambda t_n.\lambda_M/\lambda = \lambda_M t_n$$

and so the expression within the logarithm does not change value, and only the change in $1/\lambda$ has an effect. Therefore $t'_{n-1} = t_n.\lambda_M/\lambda$ for all n, by induction.

By this means all fruit rates $\lambda < \lambda_M$ can be generated from the lookup table for λ_M simply by multiplying the t_n from that table by λ_M/λ. Note, however, that because of the limited precision of the t_n as actually stored, $t_n.\lambda_M/\lambda$ may not be exactly the value that would have been calculated by using the recursion above with floating point arithmetic of greater precision. This is not a significant effect for ARIES' purpose.

APPENDIX D

DABS MESSAGE FIELD DEFINITIONS

The message field definitions contained in this appendix are those specified (in FAA-RD-74-62) for the DABS sensors which ARIES is designed to test*.

1.0 GENERAL FIELDS

1.1 F: Format Type

The first bit of the data block, designating format type:

F	Format Type
0	Normal (Surveillance, Comm-A, Comm-B formats)
1	Special (All-Call, ELM formats)

1.2 L: Length

The second bit of the data block, designating block length:

L	Length
0	56 bits
1	112 bits

1.3 IT: Interrogator Type

A one-bit field designating whether the interrogation is transmitted from a standard (IT=1) or auxiliary (IT=0) interrogator. The IT bit determines the meaning of the DABS All-Call lockout (DL) code.

In general, a standard interrogator is a DABS sensor with a directional antenna performing normal ATC surveillance and communications, while an auxiliary interrogator is an omnidirectional facility whose function is to provide synchronizing interrogations to aircraft not under DABS surveillance.

2.0 ALL-CALL INTERROGATION/REPLY FIELDS

2.1 Capability

A six-bit field which designates to the interrogator the data link input/output capability of the aircraft. Each of the bits in this field is dedicated to a specific capability. A "1" in a particular bit position indicates the following:

---

*Some modifications to the DABS message formats have taken place since FAA-RD-74-62 was issued. For the updated formats and protocols the reader is referred to FAA-RD-77-143.

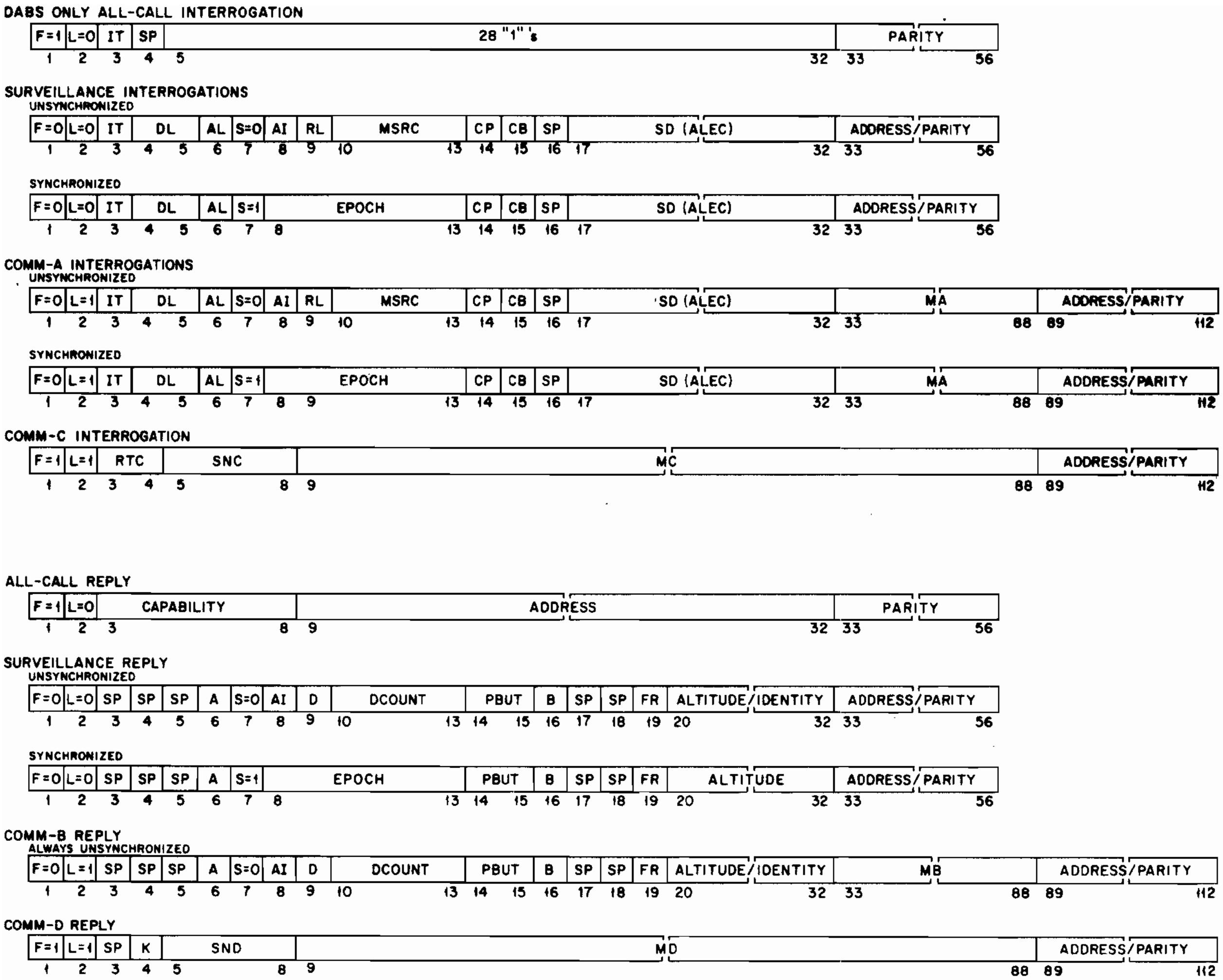


Fig.D-1. Interrogation and reply formats.

FIELD NAME ABBREVIATIONS (Fig. D-1).

A:	Alert
AI:	Altitude/Identity Designator
AL:	ATCRBS Lockout
ALEC:	Altitude Echo
B:	Air-to-Ground Data Link Message Waiting
CB:	Clear Comm-B
CP:	Clear PBUT
D:	Air-to-Ground Extended-Length Message Waiting
DCOUNT:	Number of Segments in Air-to-Ground Extended-Length Message
DL:	DABS All-Call Lockout
EPOCH:	Synchronous Reply Time
F:	Format Type
FR:	Flight Rules Indicator
IT:	Interrogator Type
K:	Extended-Length Message Control Indicator
L:	Data-Block Length
MA:	Ground-to-Air Data Link Message
MB:	Air-to-Ground Data Link Message
MC:	Ground-to-Air Extended-Length Message Segment
MD:	Air-to-Ground Extended-Length Message Segment
MSRC:	Air-to-Ground Data Link Message Source
PBUT:	Pilot Acknowledgment Buttons
RL:	Reply Length
RTC:	Reply Type for Comm-C Interrogations
S:	Synchronization Indicator
SD:	Special Data
SP:	Spare
SNC:	Segment Number of Ground-to-Air ELM Segment
SND:	Segment Number of Air-to-Ground ELM Segment

TABLE D-1
INDEX OF FIELD DEFINITIONS

A (4.1)
Address (2.2)
Address/Parity (3.13 and 4.12)
AI (3.4 and 4.3)
AL (3.2)
ALEC (3.11)
Altitude (4.10)
AR (3.12)
B (4.8)
Capability (2.1)
CB (3.9)
CP (3.8)
D (4.4)
DC (4.1)
DCOUNT (4.5)
DE (4.1)
DL (3.1)
EPOCH (3.7 and 4.6)
F (1.1)
FR (4.9)
Identity (4.10)
IT (1.3)
K (5.3)
L (1.2)
MA (3.12)
MB (4.11)
MC (5.2)
MD (5.3)
MDES (4.1)
MSRC (3.6)
Parity (2.3)
PBUT (4.7)
RL (3.5)
RTC (5.1)
S (3.3 and 4.2)
SA (4.1)
SD (3.10)
SNC (5.2)
SND (5.3)

Bit No.	
3	Equipped with IPC/PWI Display
4	Equipped with ATC numeric display (Freq., Alt., Heading, Airspeed)
5	Equipped with ATC alphanumeric display (32 character)
6	Unassigned
7	Unassigned
8	Request extended capability and/or variable call sign (see 3.6)

2.2 Address

A 24-bit field which contains the aircraft's discrete address code.

2.3 Parity

A 24-bit field which contains 24 parity check bits generated by applying a parity check code to the preceeding bits in the data block. The algorithms for generating the parity check bits are described in FAA-RD-74-62.

3.0 SURVEILLANCE/COMM-A INTERROGATION FIELDS

3.1 DL: DABS All-Call Lockout

A two-bit field which allows the interrogator to prevent the transponder from replying to All-Call interrogations or auxiliary discrete address interrogations. The meaning of the DL code is determined by the value of the IT bit received in the same transmission, as summarized in Table D-2.

TABLE D-2
LOCKOUT CODES

DL	Function When IT = 1 (Standard Interrogator)	Function When IT = 0 (Auxiliary Interrogator)
00	Clear all DABS lockouts	Clear Aux. All-Call lockout
01	Lockout Std. All-Calls in addition to previously-set lockouts	Lockout Aux. All-Calls in addition to previously-set lockouts
10	Lockout Aux. All-Calls and Aux. Discrete-Address Interrogations	Hold current lockout state
11	Lockout Std. All-Calls, Aux. All-calls, and Aux. Discrete-Address Interrogations	Unassigned

(Note: Standard All-Calls are ATCRBS/DABS All-Calls. Auxiliary All-Calls are DABS-only All-Calls with IT = 0. DABS-only All-Calls with IT = 1 cannot be locked out.)

3.2 AL: ATCRBS Lockout

A one-bit field which allows the interrogator to prevent the transponder from replying to ATCRBS interrogations. A "1" in this field inhibits replies to ATCRBS interrogations.

3.3 S: Synchronization Indicator

A one-bit field which designates whether the interrogation is synchronized, i.e., timed such that the resulting reply occurs at a precise clock time. A "1" in this field indicates that the interrogation is synchronized.

3.4 AI: Altitude/Identity Designator

A one-bit field used to designate whether the reply is to contain the pressure altitude code or ATCRBS identity code in its altitude/identity field. A "1" is used to request transmission of the ATCRBS identity code.

3.5 RL: Reply Length

A one-bit field used to designate the reply type (length), as follows:

RL	Reply Type
0	Surveillance (56 bits)
1	Comm-B (112 bits)

3.6 MSRC: Air-to-Ground Data Link Message Source

A four-bit field used with RL=1 to initiate readout of message input devices on the aircraft (e.g., extended capability information, MLS position, automatic weather sensors, etc.). Specific codes are presently unassigned, except for MSRC=0000 and 0001, which are assigned as follows: an interrogation with RL=1 and MSRC=0000 is used following an earlier reply containing B=1 to elicit a Comm-B reply with a pilot-initiated message in MB. An interrogation with RL=1 and MSRC=0001 is used to read out the extended-capability field of an aircraft equipped for Comm-B transmission.

3.7 EPOCH: Synchronous Reply Time

A six-bit field which designates the six most significant bits of the resulting synchronized reply time.

3.8 CP: Clear PBUT

A one-bit field which is used to acknowledge and reset the pilot acknowledgment buttons. A "1" in this field indicates acknowledgment.

3.9 CB: Clear Comm-B

A one-bit field which is used to acknowledge receipt of an air-to-ground Comm-B message. A "1" in this field indicates acknowledgment and causes the B bit and the contents of the MB field to be cleared. When an interrogation is received with CB=1, the B bit shall be cleared in the reply to that interrogation unless another Comm-B message is waiting; in that case, the B bit shall remain set.

3.10 SD: Special Data

A 16-bit field used for transmission of short, specialized data words. If the first four bits of SD (Bits 17-20) are all zeros, the remainder of SD (bits 21-32) contains ALEC. (No other uses of SD are presently defined. However, if bits 17-20 are not all zeros, the transponder is thereby instructed not to display the contents of the remainder of the SD field in the ALEC display.)

3.11 ALEC: Altitude Echo

A 12-bit sub-field of SD used when bits 17-20 are all zeros to transmit to the aircraft its altitude as reported on the previous transponder response. For convenience in display, ALEC is encoded as follows: bits 21-24 transmit the decimal integers 0-12, representing 10,000-ft increments through 120,000-ft: bits 25-28 transmit the decimal integers 0-9, representing 1,000-ft increments: bits 29-32 transmit the decimal integers 0-9, representing 100-ft increments. For aircraft flying below 18,000 ft, MSL, ALEC is corrected for local barometric pressure in the aircraft's operating area, so that the ALEC display should correspond to the aircraft altimeter readout.

(The ALEC display allows the pilot to continuously verify that the aircraft's reported altitude is the same as that indicated by the aircraft altimeter. Such a check is necessary to insure the validity of the reported altitude when used for automatic conflict detection/resolution. In addition, the ALEC display provides continuous assurance to the pilot that he is under DABS surveillance and his equipment is functioning properly.)

3.12 MA: Ground-to-Air Data Link Message

A 56-bit field which contains the ground-to-air data-link message. The first bit of MA, designated AR (Acknowledgment Request), indicates whether a pilot acknowledgment of the message is desired. The first eight bits of MA include, in addition to AR, the display device address code and display control bits.

3.13 Address/Parity

A 24-bit field which contains the 24-bit discrete address of the interrogated aircraft combined with 24 parity bits generated by applying a parity check code to the preceding bits in the data block. The algorithms for generating the parity check bits and for their combination with the address code are described in FAA-RD-74-62.

4.0 SURVEILLANCE/COMM-B REPLY FIELDS

4.1 A: Alert

A one-bit field requesting the sensor to transmit an interrogation with AI=1 so that the ATCRBS identity (4096) code setting may be transmitted to the ground. This is the manner in which a pilot indicates an emergency condition when under DABS surveillance. It could also be used for transmitting other downlink messages if specific codes were designated for that purpose. The A bit is ordinarily set by means of a momentary-contact switch on the transponder control panel. In addition, the transponder will be designed so that the A bit is automatically set when the first two digits of the ATCRBS 4096 code are set to 76 or 77. When set by the momentary-contact switch, the A bit is cleared by the receipt of an interrogation with AI=1. When set by the ATCRBS code selector, the A bit is transmitted in all DABS replies and cannot be cleared from the ground.

4.2 S: Synchronization Indicator

A one-bit field which echos the S bit in the corresponding interrogation, indicating (by S=1) that the reply is synchronized and can be used by suitably equipped aircraft for air-to-air ranging. A synchronized reply will always be a Surveillance reply with altitude. Thus, the S bit of a Comm-B reply will always be "0".

4.3 AI: Altitude/Identity Designator

A one-bit field used to designate whether the altitude/identity field contains the pressure altitude code or the ATCRBS identity code. A "1" is used to indicate the presence of the ATCRBS identity code.

4.4 D: Extended-Length "Message Waiting" Indicator

A one-bit field which designates to the interrogator that the transponder has an Extended-Length Message waiting to be transmitted. A "1" in this field indicates the presence of such a message.

4.5 DCOUNT

A four-bit field used to indicate the length of an Extended-Length Message waiting to be sent. The four-bit binary integer in DCOUNT is one less than the number of segments in the Extended-Length Message.

4.6 EPOCH: Synchronous Reply Time

A six-bit field which repeats, in a synchronized reply, the contents of the EPOCH field in the corresponding synchronized interrogation (paragraph 3.7).

4.7 PBUT: Pilot Acknowledgment Buttons

A two-bit field set by the pilot to respond to an Acknowledgment Request (AR=1), or to request the transmission of a display test sequence, as follows:

PBUT	Meaning
0 0	None
0 1	Cannot comply
1 0	Will comply
1 1	Request test transmission

4.8 B: Data Link "Message Waiting" Indicator

A one-bit field which designates to the interrogator that the transponder has a Comm-B message waiting to be transmitted. A "1" in this field indicates the presence of such as message.

4.9 FR: Flight Rules Indicator

A one-bit field set by the pilot to indicate whether the aircraft is operating under visual or instrument flight rules. A "1" in this field designates IFR operation.

4.10 Altitude/Identity

A 13-bit field which contains the ATCRBS Mode A identity code or Mode C altitude code (including the "X" bit), as indicated by the AI field.

4.11 MB: Air-to-Ground Data Link Message

A 56-bit field which contains the air-to-ground data link message. The first eight bits of MB are reserved for input device address code and associated control/status bits.

4.12 Address/Parity

A 24-bit field which contains the 24-bit discrete address of the replying aircraft combined with 24 parity check bits generated by applying a parity check code to the preceding bits in the data block. The algorithms for generating the parity check bits and for their combination with the address code are described in FAA-RD-74-62.

5.0 SPECIAL COMM-C/COMM-D FIELDS

5.1 RTC: Reply Type for Comm-C Interrogations

A special two-bit reply type and control field used in conjunction with an Extended-Length Message transmission. The RTC codes are defined in Table D-3.

5.2 SNC and MC

Respectively 4- and 80-bit segment number and message fields included in Comm-C transmissions. The interpretation of these fields depends on the RTC code, as defined in Table D-3.

5.3 K, SND, and MD

Respectively, 1-, 4-, and 80-bit fields which are used for Extended-Length Message transfer, as follows:

K	SND	MD
0	Segment number of message segment in MD	80-bit segment of air-to-ground Extended-Length Message
1	Not used	Cumulative transponder technical acknowledgment (TTA) of ground-to-air Extended-Length Message

TABLE D-3
DEFINITION OF COMM-C FIELDS

RTC	SNC	MC	Reply	Function of Comm-C Transmission
0 0	0 0 0 1 to 1 1 1 1	Initial 80-bit Message Segment	No Reply	Contains highest-numbered segment of uplink ELM and indicates number of segments to follow.
0 1	0 0 0 1 to 1 1 1 0	Intermediate 80-bit Message Segment	No Reply	Contains an intermediate segment of an uplink ELM and its segment number.
1 0	0 0 0 0 to 1 1 1 0	Final 80-bit Message Segment	Comm-D with K=1 and TTA in MD	Contains last segment of uplink ELM and requests cumulative transponder technical acknowledgment (TTA).
1 1	0 0 0 0	Segment Request (SR) Field (16 bits)	Multiple Comm-D transmissions	Specifies which downlink ELM segments the transponder should transmit.
1 1	0 0 0 1	Bit 9 (first bit of MC) = 1, others = 0	Single Comm-D, contents arbitrary	Concludes an uplink ELM transaction.
1 1	0 0 1 0	Bit 9 (first bit of MC) = 1, others = 0	Single Comm-D, contents arbitrary	Concludes a downlink ELM transaction.
1 1	0 0 1 1 to 1 1 1 1	---	---	Undefined

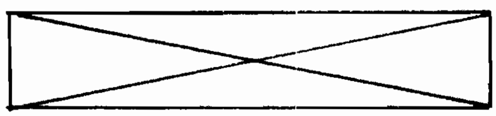
APPENDIX E

SYSTEM DATA STRUCTURES

This appendix describes the formats of certain major data structures referred to in Section 3.0. Included are the interrogation data block formats for the receiver and the reply data block formats for the controlled reply generator. Formats are presented in terms of diagrams representing the layout of data in memory, and showing the number of words used for the structure and which bits are allocated to each of the fields within the data structure. Associated with each format diagram is a table which describes the purpose of each field and its units.

For a complete description of the I/O formats and protocols of all ARIES devices please see Volume 3 of this document.

The following notation is used in the format diagrams to denote bits or fields which are unused and whose values are undefined. Programmers and digital designers should make no assumptions about the values of these fields.



unused field

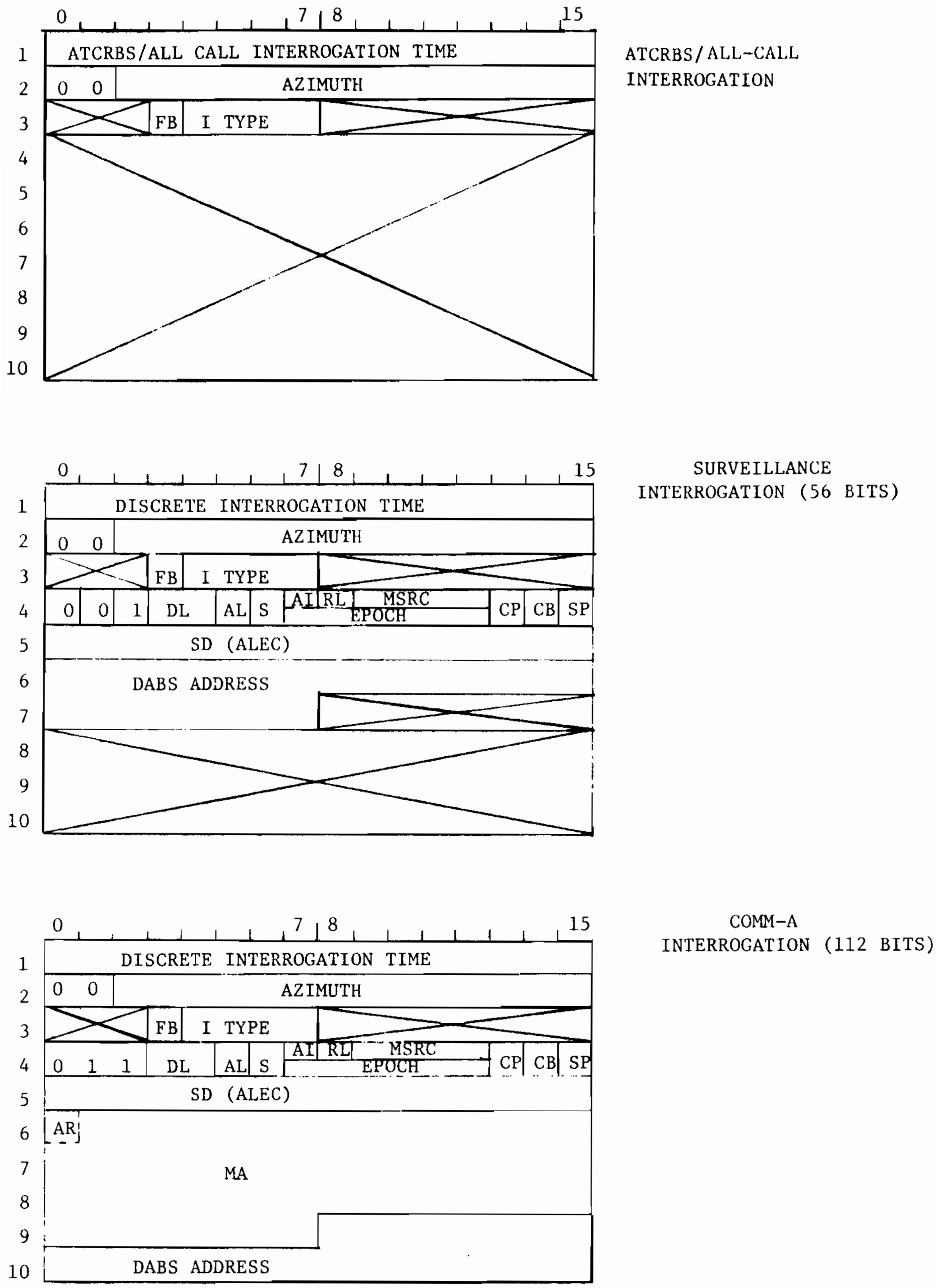


Fig.E-1. Interrogation data formats (receiver).

TABLE E-1

INTERROGATION DATA FIELD DEFINITIONS

Field	Significance
ATCRBS/ALL-CALL INTERROGATION TIME	The time since the last ATCRBS/DABS All-Call interrogation; LSB = 1 μsec. Only used to measure the inter-interrogation interval and therefore does not need range-clock resolution.
DISCRETE INTERROGATION TIME	The time since the last ATCRBS/DABS All-Call interrogation modulo 4.096 msec; LSB = 1/16 μsec. Used to calculate discrete reply times; therefore, range-clock resolution is required.
FB	Front/back bit. Indicates which antenna, front or back, transmitted the interrogation. 0 Front Antenna 1 Back Antenna
ITYPE	Interrogation type. Decimal Value — Significance 0 — Used to indicate an illegal uplink type. 1 — ATCRBS/DABS All-Call, Mode A. 2 — ATCRBS/DABS All-Call, Mode C. 3 — DABS-only All-Call. 4 — DABS surveillance (56 bits). 5 — ATCRBS Mode A (no P_4 pulse). 6 — ATCRBS Mode C (no P_4 pulse). 7 — ATCRBS Mode D (no P_4 pulse). 8 — DABS COMM-A (112 bits). 9-15 — Undefined.
AZIMUTH	LSB = $2\pi/2^{14}$ radian; measured clockwise from true north.

Note: The following fields are included in a DABS discrete interrogation. A detailed description of these fields appears in Appendix D.

Field	Significance
DL	DABS lockout. 00 Clear all DABS lockouts. 01 Lockout Mode A and Mode C All-Calls. 10 Lockout auxilliary interrogations (ignored by ARIES). 11 Lockout auxilliary interrogations and standard All-Calls (treated like 01 by ARIES).

TABLE E-1
INTERROGATION DATA FIELD DEFINITIONS (Continued)

Field	Significance
AL	ATCRBS lockout. Ignored by ARIES.
S	Synchronization indicator (synchro-DABS). 0 Unsynchronized interrogation. AI, RL, MSRC appear in the following bits. 1 Synchronized interrogation. EPOCH appears in the following bits.
AI	Altitude/identity bit. 0 Reply with Mode C altitude. 1 Reply with Mode A code.
RL	Reply length requested. 0 Surveillance reply (56 bits). 1 COMM-B reply (112 bits).
MSRC	Air-to-ground data link message source. Ignored by ARIES.
EPOCH	Synchro-DABS epoch. Merely copied into corresponding reply field.
CP	Clear PBUT. If set to 1, clears the transponder's pilot acknowledgment.
CB	Clear B bit. If set to 1, clears the transponder's downlink request.
SP	Spare bit, ignored by ARIES.
SD (ALEC)	Special data, or altitude echo field. Ignored by ARIES.
MA	COMM-A message field. Ignored by ARIES except for the bit marked AR, which initiates the acknowledgment request protocol if set to 1.
DABS ADDRESS	A 24-bit transponder identifier. The low order 10 bits are used by ARIES as a track number.

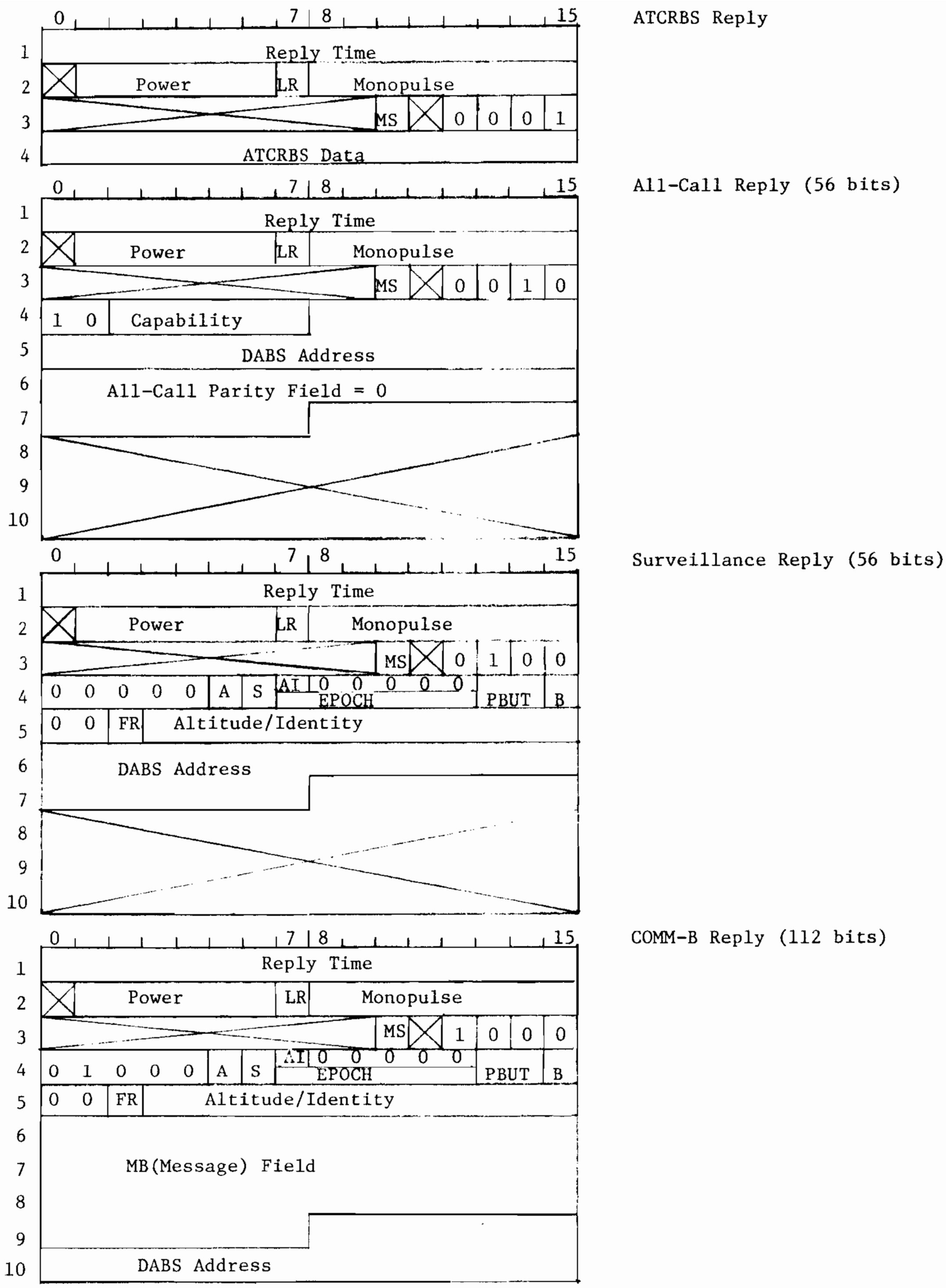


Fig.E-2. Reply data formats (controlled reply generator).

TABLE E-2

REPLY DATA FIELD DEFINITIONS

Field	Significance
REPLY TIME	The time that the reply is to be generated; LSB = 1/16 μsec. Referenced to the time of the last ATCRBS/DABS All-Call interrogation (i.e., the reply time counter is set to zero at that time).
POWER	Reply power level; LSB increment = 1 dB. 0 -20 dBm 63 -83 dBm } referred to the sensor's RF port (i.e., the reply will appear to the sensor as if it had these levels)
MONOPULSE	The off-boresight angle of the reply. Values (and therefore units) are determined by the calibration procedure. This value is calculated by table lookup based on off-boresight angle. The table is derived during the calibration procedure. In terms of amplitude, each step represents a .125 dB attenuation of Δ relative to Σ. A value of 0 gives $\Delta=\Sigma$.
MS	Mainbeam/Sidelobe. Determines whether the reply will appear in the mainbeam (MS=0) or in the sidelobe (MS=1). Normally, this value will be zero. A sidelobe reply is attenuated by an additional amount specified on front panel switches.
LR	Monopulse sign. Determines whether the monopulse angle is to be measured to the left (LR=0) or to the right (LR=1) of boresight.
ATCRBS DATA	The 13 data bits of a standard ATCRBS reply, plus F_1, F_2 (always 1) and SPI. The ordering is: 1 $C_1A_1C_2A_2C_4A_4OB_1D_1B_2D_2B_4D_4$ 1 SPI where the standard ATCRBS pulse nomenclature is used.

TABLE E-2

REPLY DATA FIELD DEFINITIONS (Continued)

The following fields are included in a DABS reply. A detailed description of these fields appears in Appendix D.

Fields	Significance
CAPABILITY	Bits indicating the on-board devices with which the transponder is equipped. In ARIES, this information is obtained from the model file.
DABS ADDRESS	A 24-bit transponder identifier. The low order 10 bits are used by ARIES as a track number.
ALL-CALL PARITY FIELD	Must always be 0.
A	Pilot alert bit. If set to 1, it indicates an on-board emergency condition whose nature is indicated by the value of the Mode A code. Set from the model file.
S	Synchronization indicator. Copied from the same bit in the interrogation which caused this reply. 0 Unsynchronized. AI bit follows. 1 Synchronized. EPOCH field follows.
AI	Altitude/Identity. Copied from the same bit in the interrogation which caused this reply. 0 Reply contains Mode C altitude. 1 Reply contains Mode A code.
EPOCH	Synchro-DABS epoch time. Copied from the interrogation which caused this reply.
PBUT	Pilot acknowledgment bits. Set from the model file whenever a pilot acknowledgment is requested. 00 No acknowledgment. 01 Cannot comply. 10 Will comply. 11 Request test transmission (not used by ARIES).
B	Air-to-ground downlink request (B=1). The ground is requested to allocate channel time for a COMM-B downlink. Set from the model file.

TABLE E-3
REPLY DATA FIELD DEFINITIONS (Continued)

Fields	Significance
FR	Flight rules. Set from the model file. 0=VFR; 1=IFR.
ALTITUDE/IDENTITY	If AI=0, this field contains the Mode C altitude. If AI=1, this field contains the Mode A code. The bit order, using the standard ATCRBS pulse nomenclature is: $C_1A_1C_2A_2C_4A_4OB_1D_1B_2D_2B_4D_4$
MB	A fixed format downlink message field. The same MB is returned by ARIES in all COMM-B downlinks.

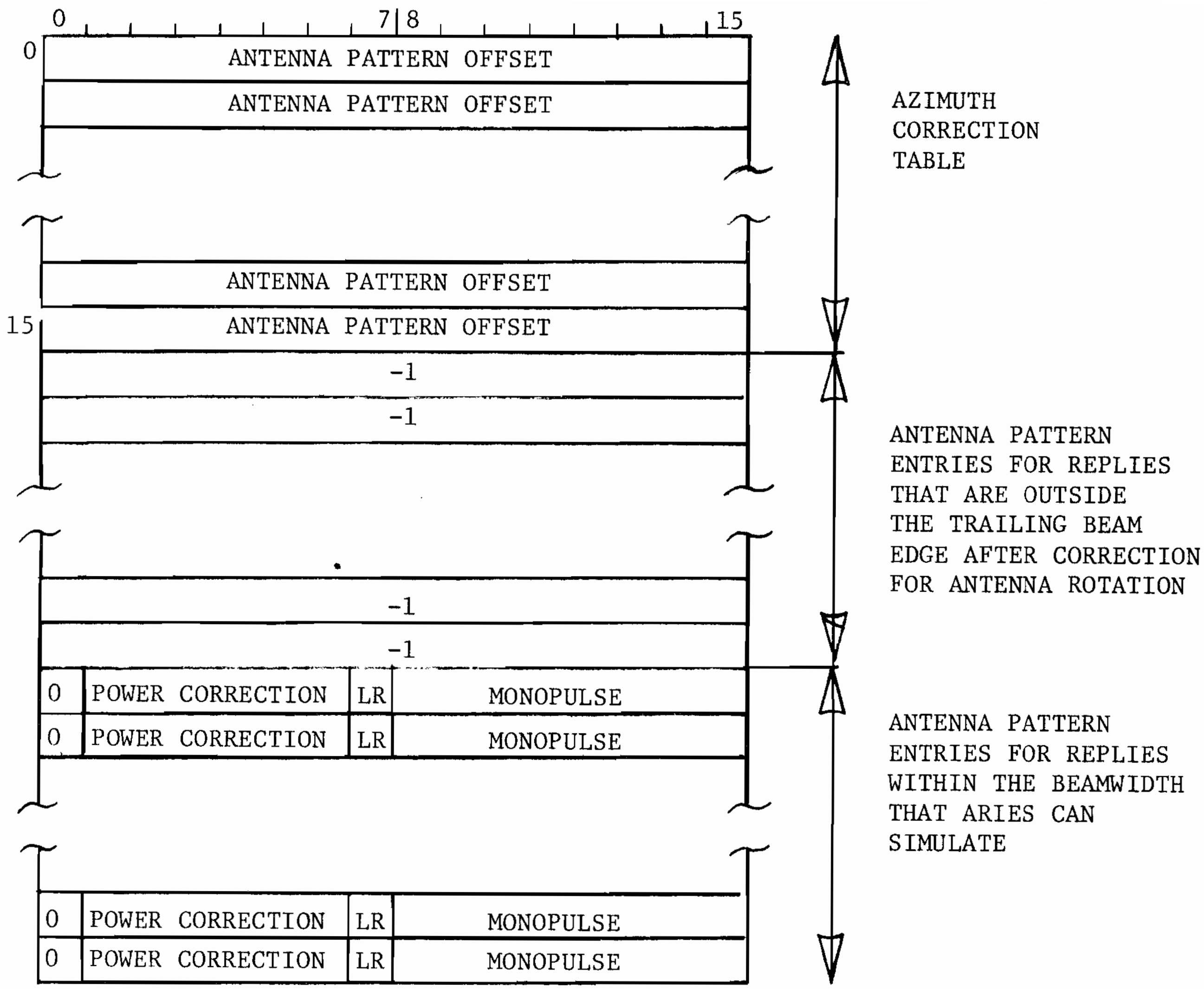


Fig.E-3. Azimuth correction table and antenna pattern lookup table.

TABLE E-3

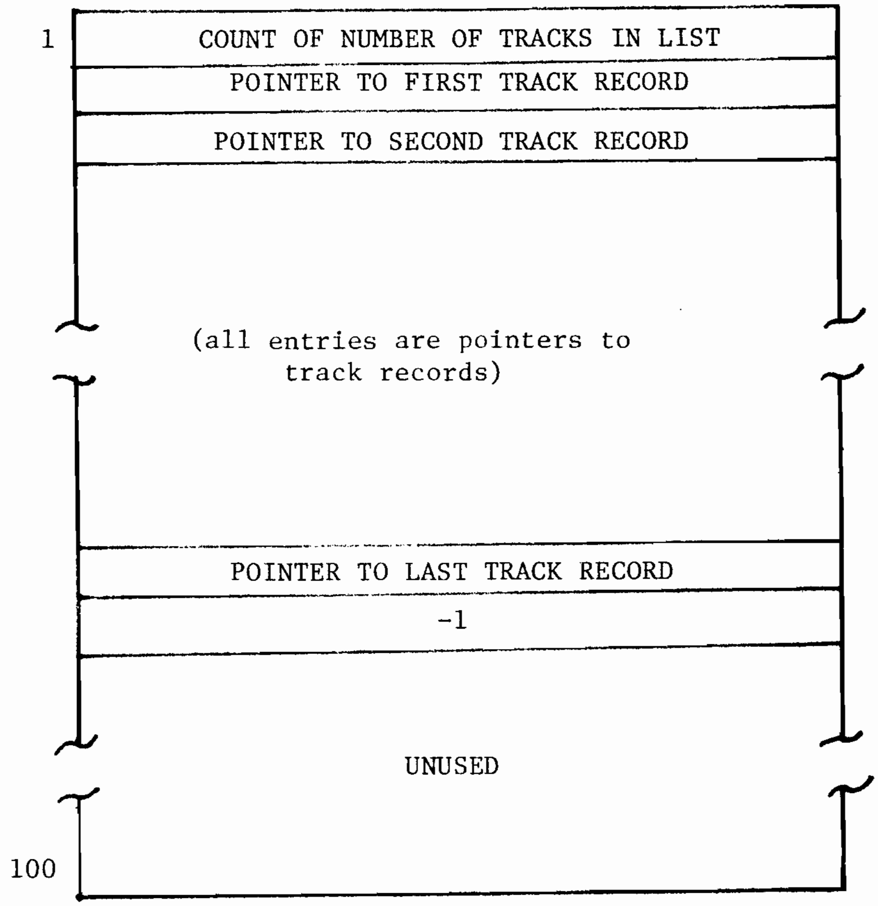
FIELD DEFINITIONS AND NOTES FOR THE AZIMUTH CORRECTION TABLE AND THE ANTENNA PATTERN LOOKUP TABLE

Fields	Significance
ANTENNA PATTERN OFFSET	This is an offset from the base of the Antenna Pattern Lookup Table (word 16) to the entry in that table that corresponds to the offboresight angle at reply time of a target that is interrogated on boresight. The difference in angle is due to antenna rotation during the round trip time. The Azimuth Correction Table is accessed by using the most significant 4 bits of the round trip time as an index, thus obtaining the appropriate correction for the given range.
POWER CORRECTION	A correction to the reply power to account for the sensor antenna's gain pattern. LSB = -1 dB 0 = NO ADJUSTMENT 63 = The reply power is reduced by 63 dB.
LR	Monopulse Sign 0 = Left of boresight 1 = Right of boresight
MONOPULSE	The amount of attenuation of Δ relative to Σ required to simulate the offboresight angle corresponding to this entry. LSB = .125 dB. A value of 0 gives $\Delta = \Sigma$.

Notes:

1) The number of -1 entries is equal to the maximum azimuth correction due to antenna rotation, in units of $2\pi/2^{14}$ radians (1 Au). This provides for a target at trailing beam edge at interrogation time and at the maximum range. Location 16 (i.e., the first location of the Antenna Pattern Lookup Table) would be accessed for this target.

2) The number of valid entries in the Antenna Pattern Lookup Table is determined by the beamwidth of the antenna measured between the two $\Sigma = \Delta$ crossover points in units of Au. An entry must be provided for each offboresight angle where a reply could be generated. This is determined by the cutoff angle parameter used by Interrogation Processing. The table size must be 2*cutoff + 1 entries long, where cutoff is measured in Au.



A POINTER IS THE WORD ADDRESS OF THE FIRST WORD OF THE TRACK RECORD.

THERE ARE THREE SUCH INDICES, ONE EACH FOR THE FRONT AND BACK ANTENNAS, AND ONE BEING CREATED BY PRE-INTERROGATION PROCESSING.

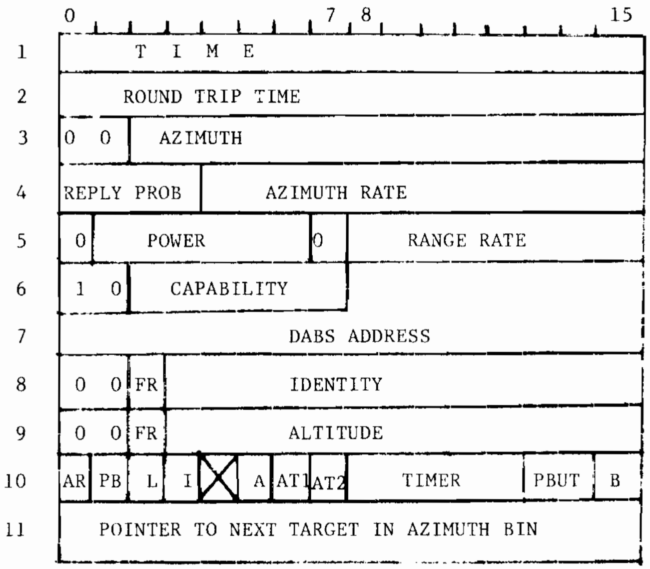
Fig.E-4. Reply time sorted index.

NOTE: The Interrogation Pattern File consists of up to 20 words in this format. Only as many words are used as are needed to describe the sensor's interrogation pattern. Additional words may be used to store interrogation data during pattern acquisition.

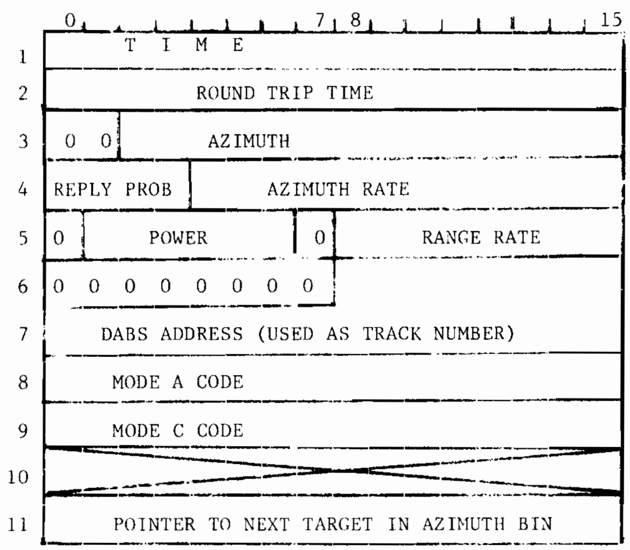
DEFINITION OF FIELDS

FIELD	SIGNIFICANCE
AC	MODE A(0) OR MODE C(1) INTERROGATION.
FB	FRONT (0) OR BACK (1) ANTENNA.
INTERVAL	INTERVAL BETWEEN THIS INTERROGATION AND THE PRECEDING ATCRBS/ALL-CALL INTERROGATION. LSB = 4 μSEC.

Fig.E-5. Interrogation pattern file word format.



a) DABS Target*



b) ATCRBS Target*

*Note: Any updates to these track records must be made with interrupts disabled, or by uninterruptable instructions, or by single store operations without prior reading. This avoids nondeterministic interactions with Interrogation Processing.

Fig.E-6. Track file record formats.

TABLE E-4

TRACK FILE FIELD DEFINITIONS

Field	Significance
TIME	This field contains the low order 16 bits of the system time to which the track's position has been updated; LSB = 1 msec. Note that this time will wrap around (to zero) once every 65,536 milliseconds. Algorithms using this field must be prepared to handle ambiguities occurring near such times.
ROUND TRIP TIME	This field contains the total round trip time to the transponder and back, and includes transponder turn-around time (3 μsec for ATCRBS, 128 μsec for DABS); LSB = 1/16 μsec.
AZIMUTH	Measured clockwise from true north; LSB = $2\pi/2^{14}$ radians.
REPLY PROB	Reply probability; LSB = 1/32 0 No reply will be sent. 15 A reply will always be sent if the aircraft is in the antenna beam. N≠0 Replies will be generated for (N+17)/32 of the interrogations.
AZIMUTH RATE	LSB = $(2\pi/2^{14})$ (1/4096) radian/msec. Range of $\pm$ $1/2(2\pi/2^{14})$ = $\pm$ 0.00019 radian/msec (10.88 degrees/sec) in two's complement notation. The extra resolution of 1/4096 of the basic azimuth measurement accuracy allows the azimuth to be predicted over approximately four seconds with no significant azimuth error. A target at 1.0 nmi range with 1000 ft/sec tangential velocity has an azimuth rate of 1/6076 = 0.00016 radian/msec.
POWER	Reply power level before losses due to off-boresight gain of sensor antenna. LSB = -1 db 0 = -20 dbm 63 = -83 dbm
RANGE RATE	LSB = (1/16) (1/4096) (μsec/msec). Rate of change of the round-trip time. Extra resolution of 1/4096 is provided for the same reason as the extra azimuth rate resolution. The range of values $\pm$(1/32) (1/16) (μsec/msec) = $\pm$ 960 ft/sec (569 kt) in one-way range rate.
CAPABILITY	Reply data bits provided by the model file and included in all-call replies. They indicate what devices the transponder has attached to it.

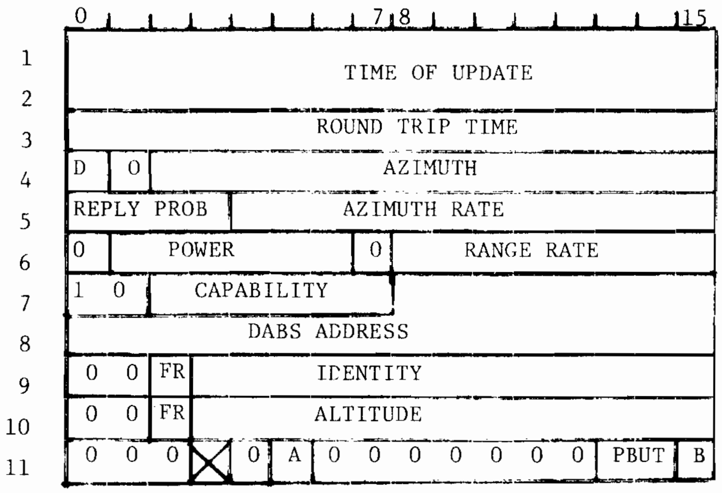
TABLE E-4

TRACK FILE FIELD DEFINITIONS (Continued)

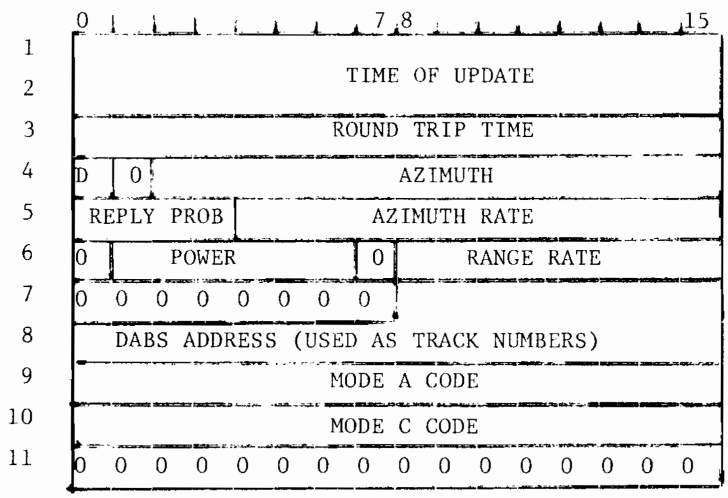
Field	Significance
DABS ADDRESS	A 24-bit transponder identifier. The low order 10 bits are used by ARIES as a track number.
FR	Flight rules bit. 0 = VFR, 1 = IFR.
IDENTITY	Mode A code for discrete replies. Bit format is: $C_1A_1C_2A_2C_4A_4 0 B_1D_1B_2D_2B_4D_4$
ALTITUDE	Mode C code for discrete replies. Bit format is identical to that for Identity.
AR	A 1 indicates that an acknowledgment request cycle is in progress.
PB	A 1 indicates that the PBUT field is to be inserted into discrete replies.
L	A 1 indicates that the simulated transponder is locked out to all-calls.
I	A 1 indicates that the track is in the process of being initialized by the procedures described in Section 3.5.7.
A	Value of the pilot alert (A) bit to be inserted in discrete replies. Obtained from the model file.
AT1, AT2	Indicate the timeout status of this track at two other ARIES sites. A 0 indicates that the track timed out. A 1 indicates that a discrete interrogation was received recently enough to prevent timeout.
TIMER	This field simulates the transponder's loss-of-contact timeout function; LSB = 1 sec. See Section 3.5.6.
PBUT	Value of the pilot acknowledgment bits to be inserted in discrete replies if PB=1. Obtained from the model file.
B	A 1 indicates a downlink message initiated by the track is in progress. This bit is set by the model file and cleared by receipt of a CB bit in an interrogation.

TABLE E-4
TRACK FILE FIELD DEFINITIONS (Continued)

Field	Significance
MODE A CODE	ATCRBS reply data bits for Mode A interrogations. The bit format is: $1\ C_1A_1C_2A_2C_4A_4 0 B_1D_1B_2D_2B_4D_4\ 1\ SPI$
MODE C CODE	ATCRBS reply data bits for Mode C interrogations. The bit format is identical to the Mode A code.
POINTER TO ... BIN	This field contains the word address of the first word in the track record pointed to.



a) DABS TARGET



b) ATCRBS TARGET

NOTE: REFER TO TABLE E-4 FOR THE FIELD DEFINITIONS, EXCEPT FOR THE FOLLOWING.

FIELD	SIGNIFICANCE
TIME OF UPDATE	SYSTEM CLOCK TIME AT WHICH THIS MODEL RECORD IS TO BE ENTERED INTO THE TRACK FILE. LSB = 1 MSEC. NOTE: IF THE MSB OF THIS FIELD IS SET, TRAFFIC MODEL INPUT TREATS THAT RECORD AS THE END OF THE TRAFFIC MODEL FILE.
D	A 1 INDICATES THAT THIS TRACK IS TO BE DROPPED FROM THE TRACK FILE.

Fig.E-7. Traffic model input format.

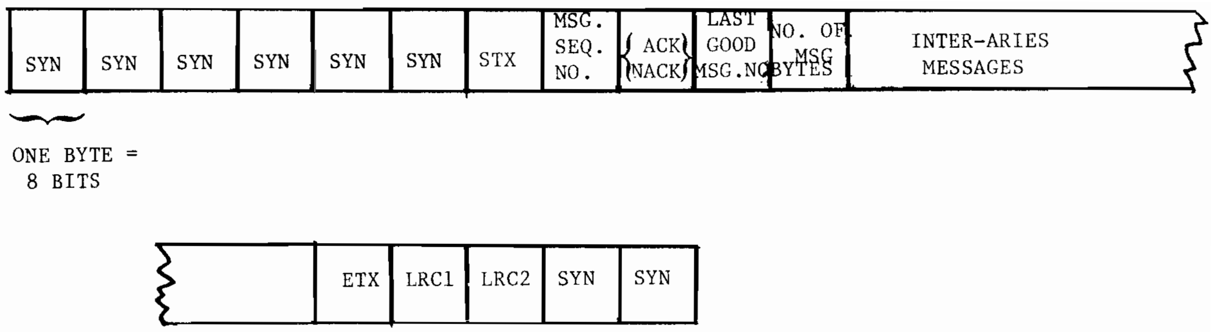
1	POINTER TO FIRST TARGET IN THIS BIN
2	"
	(All table entries are of the same type)
256	POINTER TO FIRST TARGET IN THIS BIN
257	POINTER TO FIRST TARGET IN < 5 NMI BIN

FRONT ANTENNA STARTING BIN INDEX
FRONT ANTENNA END BIN INDEX
BACK ANTENNA STARTING BIN INDEX
BACK ANTENNA END BIN INDEX

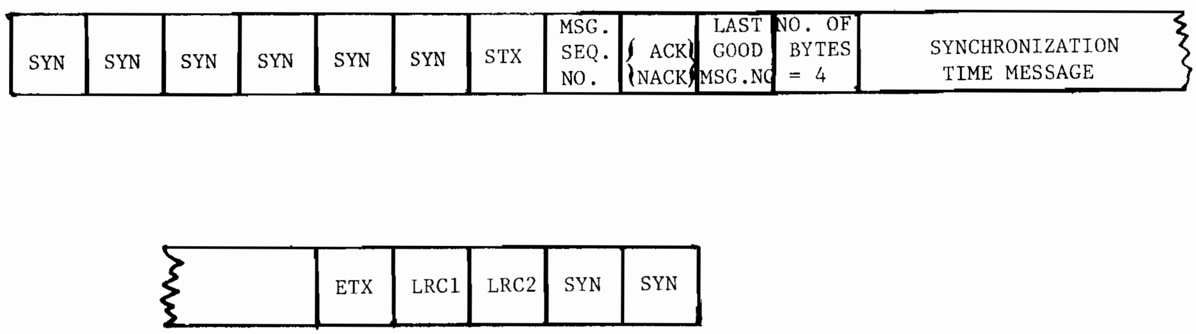
A POINTER IS THE WORD ADDRESS OF THE FIRST WORD OF THE TRACK RECORD. THE TRACKS IN AN AZIMUTH BIN ARE LINKED BY MEANS OF THE POINTERS IN WORD 11 OF THE TRACK RECORDS.

EACH AZIMUTH BIN CORRESPONDS TO A $2\pi/256$ RADIAN AZIMUTH WEDGE. A BIN INDEX IS AN INTEGER IN THE RANGE: 1 THROUGH 256.

Fig.E-8. Azimuth sorted index.



a) TRACK STATUS MESSAGES



b) SYSTEM STATUS MESSAGES

Fig.E-9. Inter-ARIES transmission block formats.

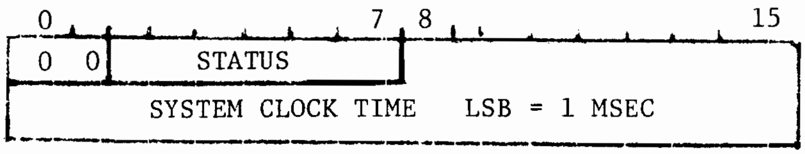
TABLE E-5

DEFINITION OF INTER-ARIES LINK CONTROL CHARACTERS

Symbol	Significance
ACK	Acknowledgment. This indicates that all messages through the message with the message sequence number contained in the following byte ("LAST GOOD MSG. NO.") have been received without error. The code for this character is 006_8.
ETX	End of text. This symbol marks the end of the data field of the message. It cannot be used to find the end of the data, however, as the data may contain bytes which match the ETX symbol. It can be used in conjunction with the message byte count ("NO. OF MSG. BYTES") to perform a consistency check, essentially confirming that a correct byte count was received. The code for this character is 003_8.
LRC1, LRC2	Longitudinal Redundancy Check. These are obtained by computing the exclusive-or of all the words in the output buffer between the STX and ETX characters, and including these characters. LRC1 is for bits 0-7, LRC2 for bits 8-15. The STX character is assumed to be in bits 0-7 of the first word included in this computation. If the message has an odd number of bytes, the ETX will appear in bits 0-7 of the last word and bits 7-15 will be set to 0 for the LRC calculation.
MSG. SEQ. NO.	A sequential numbering of messages sent over a given inter-ARIES link. Each link has its own numbering, independent of all other links. The receiver tests each message to verify that it is the next expected message in the sequence, and if it is not, requests retransmission of all messages starting with the first missing message.
NACK	Negative acknowledgment. As with ACK, all messages through the one indicated by "LAST GOOD MSG. NO." are acknowledged. In addition, however, the following message was received with errors and retransmission of that and all subsequent messages is requested. The code for this character is 025_8.
STX	Start of text. Marks the beginning of a message. If not present (i.e., some character other than SYN or STX is received before receiving STX) an error is signalled. The code for this character is 002_8.

TABLE E-5
DEFINITION OF INTER-ARIES LINK CONTROL CHARACTERS

Symbol	Significance
SYN	Synchronization/idle character. This character synchronizes the receiving circuits with the transmitting circuits, and also serves as the idle character transmitted when there is no data to send or when the transmitting CPU is late in providing a message byte. The latter case can be detected by the STX, ACK/NACK, or ETX characters not appearing at the expected locations. The code for this character is 026_8.



a) SYSTEM STATUS MESSAGE

STATUS

0 SYSTEM IS (RE)INITIALIZING TO THE SPECIFIED TIME ON THE MODEL FILE.

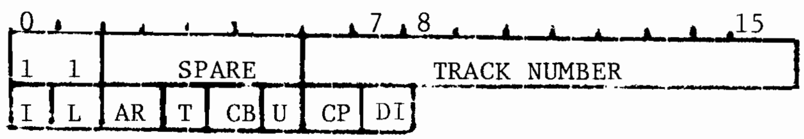
1 SYSTEM IS READY TO START AT THE SPECIFIED TIME

2 SYNCHRONIZATION TIME MESSAGE (SYSTEM RUNNING)

3 SYSTEM HAS HALTED AT SPECIFIED TIME.

4 SYSTEM STARTUP.

OTHER VALUES ARE UNDEFINED.



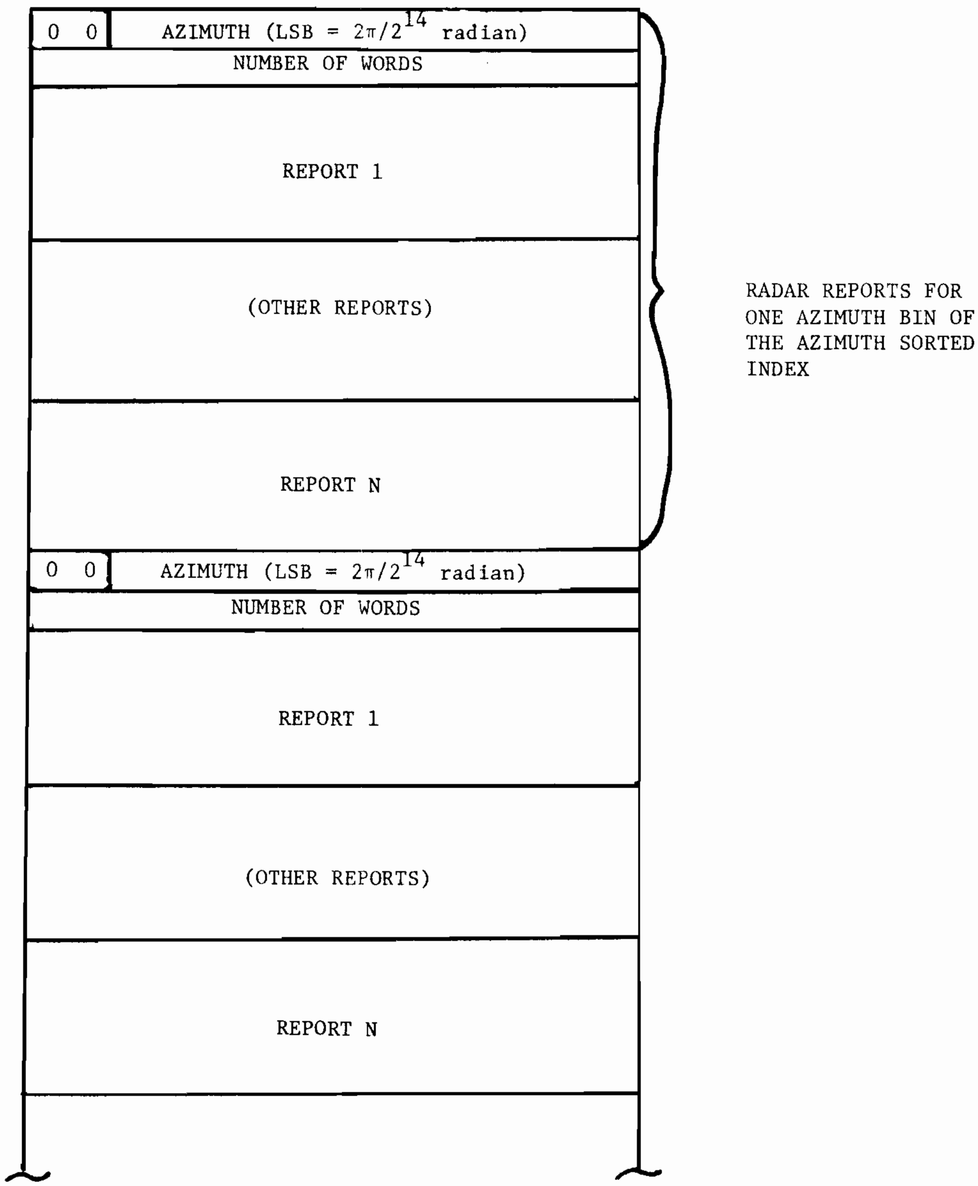
b) TRACK STATUS MESSAGE

Fig.E-10. Inter-ARIES message formats.

TABLE E-6

TRACK STATUS MESSAGE BIT DEFINITION

Symbol	Significance
AR	The transmitting ARIES received an acknowledgment request for this track. The receiving site should initiate the acknowledgment request protocol.
CB	The transmitting site received CB=1 in an interrogation for this track. The receiving site should clear the B bit in the track state word.
CP	The transmitting site received CP=1 in an interrogation for this track. The receiving site should set AR=PB=0 in the track state word.
DI	The transmitting site received a discrete interrogation for this track after the transponder had timed out. The receiving site should set the status bit for the transmitting site (AT1 or AT2) to 1 in the track's state word.
I	The transmitting site just started this track. The receiving site should return a message which will initialize the track state word to the correct value, unless the default value is correct.
L	The transmitting site received a message to lockout the track to all-call interrogations. The receiving site should do the same.
T	The transmitting site has timed out this transponder (i.e., the timer has gone to zero but the state bits have not necessarily been reset depending on the values of AT1 and AT2). The receiving site should set AT1 or AT2 (as appropriate) to 0 in the track status word. If this makes both AT1 and AT2 zero, and the timer is 0, the state word should be reset.
U	The transmitting site received a message to unlock this target to all-call interrogations. The receiving site should do the same.



THE AZIMUTH WORD SPECIFIES THE AZIMUTH THE ANTENNA MUST REACH BEFORE THE FOLLOWING REPORTS ARE RELEASED FOR TRANSMISSION.

Fig.E-11. Radar report buffer format.

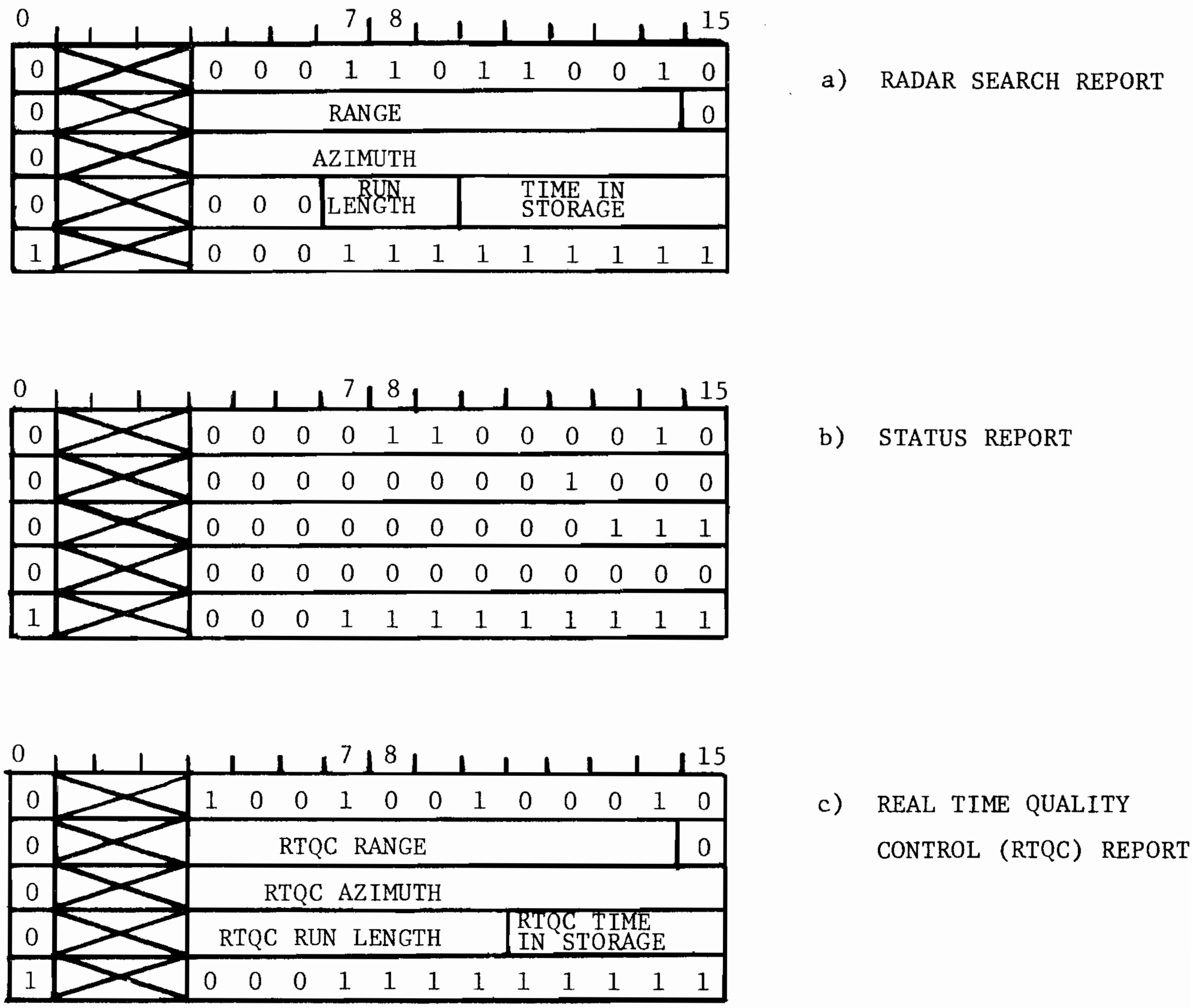


Fig.E-12. Radar report formats.

TABLE E-7
RADAR REPORT FIELD DEFINITIONS

a) Radar Search Report

RANGE -	LSB = 0.125 nm, maximum value 255.875 nm
AZIMUTH -	LSB = $2\pi/4096$ radian = 1 ACP
RUN LENGTH -	LSB = 4 ACP, maximum value 28 ACP. ARIES always sends a fixed value.
TIME IN STORAGE -	LSB = .125 second, maximum value 7.875 seconds. ARIES sends a constant.

Other fields with fixed values:

Word	Bit	Definition
1	14	FAA bit, always 1
1	15	AF (Air Force) bit, always 0
4	4	AIMS Present, always 0
4	5	MSB AIMS Code, always 0
4	6	LSB AIMS Code, always 0

b) Status Report

This report is a fixed message intended to convey to NAS that the simulated radar's status is satisfactory. To that end, the various status bits have been set as follows. Field nomenclature is taken from FAA-RD-74-63A, "Provisional Message Formats for the DABS/NAS Interface (Rev. 1)".

TABLE E-7
RADAR REPORT FIELD DEFINITIONS (Continued)

Word	Bit	Field	Value
1	14	FAA	1
1	15	AF	0
2	4	Radar alarm	0
2	5	Beacon alarm	0
2	6	CD alarm	0
2	9	AIMS alarm	0
2	10	Standby Radar Alarm	0
2	11	Standby Beacon Alarm	0
2	12	Standby CD Available	1
2	14	HPG Req Line Alarm	0
3	4	HPG Req Parity Alarm	0
3	6	DSG Alarm	0
3	11	Sens. Det. On	0
3	12	R.L. Discr. On	0
3	13	Normal Sector 3	1
3	14	Normal Sector 2	1
3	15	Normal Sector 1	1
4	4	Outer Contour	0
4	5	Inner Contour	0
4	6	Fixed Map On	0
4	7	High Speed Timing Alarm	0
4	8	1/2 Scan Inhibit Alarm	0
4	9	Buffer Overload Alarm	0
4	13	Sensitive Sector 3	0
4	14	Sensitive Sector 2	0
4	15	Sensitive Sector 1	0

c) Real Time Quality Control (RTQC) Report

This reports a fixed target once each scan for purposes of quality control checks by the NAS system receiving the data.

RTQC RANGE - A fixed range value chosen in agreement with the value expected by the NAS system receiving the data. LSB = 0.125 nm.

RTQC AZIMUTH - A fixed azimuth value for the test target chosen in accordance with the value expected by the NAS system receiving the data. LSB = $2\pi/4096$ radian = 1 ACP.

RTQC RUN LENGTH - Also a fixed value, as for the range and azimuth. LSB = 4 ACP.

RTQC TIME IN STORAGE - LSB = .125 second. ARIES sends a constant value.

1–2		SYSTEM CLOCK
3		STATUS BITS
4		ADJACENT ARIES 1 STATUS
5		ADJACENT ARIES 2 STATUS
6–13		INPUT FROM STATUS FORMATTER
14–15		SYSTEM TIME OF ANTENNA MEASUREMENT
16	0 0	ANTENNA BORESIGHT AZIMUTH
17		ANTENNA RATE
18		NUMBER OF ACTIVE DABS TRACKS
19		NUMBER OF ACTIVE ATCRBS TRACKS
20		% CPU IDLE TIME, LAST SECOND
21		% CPU IDLE TIME, CURRENT COUNT
22		% CPU IDLE TIME, MINIMUM
23		MODEM LINE 1, NACK'S SENT
24		MODEM LINE 1, NACK'S RECEIVED
25		MODEM LINE 2, NACK'S SENT
26		MODEM LINE 2, NACK'S RECEIVED
27–28		SYSTEM CLOCK AT INITIALIZATION

Fig.E-13. System status file.

TABLE E-8

SYSTEM STATUS FILE FIELD DEFINITIONS

	Significance
SYSTEM CLOCK	LSB = 1 msec. This 32 bit count is incremented by one every millisecond by the clock interrupt handler if the "system running" status bit is set.
STATUS BITS	The following bits have the meaning shown, if set. All other bits are not used. 0: Acquiring the sensor's interrogation pattern. 1: (Re)-initializing to a new point on the model file. 2: The last record from the model file has been processed. 3: ATCRBS recording is disabled. 14: The system is waiting to start the simulation. This implies that the interrogation pattern has been acquired and the model file is positioned correctly at this site. 15: The system is running. This bit causes the the system clock to be incremented.
ADJACENT ARIES STATUS	The bit definitions are the same as for the local status, but only bits 1, 14, and 15 are used.
INPUT FROM STATUS FORMATTER	This contains 8 words of device status data read from the status formatter either at system initialization or at the last status change interrupt. See FAA-RD-78-96, Volume 3 for the detailed format.

NOTE: THE NEXT THREE FIELDS CONSTITUTE THE ANTENNA AZIMUTH AND RATE DATA.

SYSTEM TIME OF ANTENNA MEASUREMENT	LSB = 1 msec. This is the 32 bit system clock value at the last ATCRBS/All-Call interrogation, which is when the azimuth data was last updated.
ANTENNA BORESIGHT AZIMUTH	The antenna boresight azimuth in the last ATCRBS/All-Call interrogation block. LSB = $2\pi/2^{14}$ radian.
ANTENNA RATE	LSB = .25 Au/msec, where 1 Au = $2\pi/2^{14}$ radian. This is calculated from the last two boresight azimuth samples and the interval between them.
NUMBER OF ACTIVE DABS TRACKS	The number of DABS equipped targets in the track file.
NUMBER OF ACTIVE ATCRBS TRACKS	The number of ATCRBS equipped targets in the track file.

TABLE E-8
SYSTEM STATUS FILE FIELD DEFINITIONS (Continued)

Field	Significance
% CPU IDLE TIME	The CURRENT COUNT word is incremented by one each time System Initialization cycles through the system idle loop. This requires 1 msec. This count is copied to LAST SECOND and then zeroed once a second by the real time clock interrupt handler. If the new LAST SECOND value is less than MINIMUM then it also becomes the new MINIMUM. Thus a rough measure of available CPU time averaged over a second is obtained.
MODEM ... NACK's SENT	The number of messages containing NACK sent over each of the inter-ARIES lines. This is a measure of line quality.
MODEM ... NACK's RECEIVED	Similar to NACK's SENT, but this is the count of NACK's received from the other ARIES sites.
SYSTEM CLOCK AT INITIALIZATION	This is the value which SYSTEM CLOCK contained when the system was started. Also, during system initialization, this field specifies the time to which the model file is to be positioned.

APPENDIX F

DATA RECORDING FORMATS

An overview of ARIES data recording is presented in Section 3.9. The buffer format and the general format of the data blocks is also described there. This appendix presents the details of the individual data blocks. The format of each block is presented, and the items within the blocks are defined. Included in the format descriptions, for the sake of completeness, are the initial three word block descriptor specifying the block number, the word count, and the time of recording. The definitions of these items are not repeated for each block, but are instead given in Section 3.9.

Title: Buffer time

Description: This block is recorded once at the beginning of each buffer of data (each physical record on tape, or every other 256 word disk block).

Word	Description
1	Block number = 1
2	Word count = 5
3	Recording time
4-5	The full 32 bit system clock value at the time this block was recorded, which in turn is the time that recording was switched to this buffer, LSB = 1 msec. This can be used to resolve any ambiguity in the 16 bit recording times in word 3 of all following data blocks.

Title: Error Messages

Description: This block may be recorded by any task detecting an error. In fact, most errors are not recorded on the data recording but simply increment counters in the computer's memory. These error counts can be read by means of Command 4 of the Operator Communications command list.

Word	Description
1	Block number = 2
2	Word count, variable with a minimum of 4.
3	Recording time
4	Error number
5-end of block	Additional optional data that may be recorded to help diagnose an error. The format is different for each error number.

See Appendix G for a list of all error numbers. Those errors which are also recorded are indicated.

Title: Azimuth Correction/Antenna Pattern Looking Table

Description: This block is recorded at system initialization. See Appendix E for format descriptions of the Azimuth Correction Table and the Antenna Pattern Lookup Table.

Word	Description
1	Block number = 3
2	Word count = 256
3	Recording time (meaningless, as this block is only recorded at initialization)
4-19	The Azimuth Correction Table
20-256	The Antenna Pattern Lookup Table. Note that since this table may vary in length depending on the antenna beamwidth, not all of these entries will be used. The unused entries will contain whatever data is stored in the disk file from which this table is read.

Title: Pattern File

Description: This block is recorded at system initialization following acquisition of the sensor's interrogation pattern.

Word	Description
1	Block number = 4
2	Word count. Variable with a minimum of 4 and a maximum of 23. As many words are recorded as are used by the sensor to describe the interrogation pattern.
3	Recording time (meaningless, as this block is only recorded at system initialization).
4-end of block	The entire Interrogation Pattern File. See Appendix E for a description of the format.

Title: Fruit Rate Table

Description: This block is recorded at system initialization, and records the table used by Interrogation Processing to determine the fruit rate in each of 32 azimuth sectors. For a description of the fruit reply generator's I/O formats and protocols, please see Volume 3 of this document.

Word	Description
1	Block number = 5
2	Word count = 37
3	Recording time (meaningless, as this block is recorded only at system initialization).
4-35	The Fruit Rate Table. Each entry represents the fruit rate for an 11 1/4 degree sector, beginning with the sector between 0 and 11 1/4 degrees and continuing clockwise. The units are those of the fruit reply generator hardware.
36	The reply code to be generated by the fruit reply generator for that fraction of replies specified to have a "fixed code". The bit order is $F_1C_1A_1C_2A_2C_4A_4XB_1D_1B_2D_2B_4D_4F_2SPI$
37	Bits 0-8: Fraction of replies to be generated as mainlobe replies. Bits 11-15: Fraction of fixed codes to be generated.

Title: Status Formatter Input

Description: This block is recorded only at system initialization, and records the entire status formatter input block which is read at that time. Subsequent updates due to status change interrupts are not recorded.

Word	Description
1	Block number = 6
2	Word count = 11
3	Recording time (meaningless, as this block is only recorded at system initialization).
4-11	All 8 words of the status formatter input block. Please see Volume 3 of this document for a description of the format.

Title: Model Input Record

Description: This is recorded by Traffic Model Input each time it begins processing a new 256 word block of model data from the disk.

Word	Description
1	Block number = 8
2	Word count = 6
3	Recording time
4	The disk block number. This defines exactly which record is being started. The record number of the first block in the model file is 1. Note that the record number is counted from the beginning of the file, not from the record at which the simulation was started.
5-6	The 32 bit update time from the first model record within the disk block (see Appendix E for the model record formats). This time can be compared with word 3 (recording time) to verify that updates from the model are being performed on time.

Title: ATCRBS/All-Call Replies - Short Form

Description: This is recorded by ATCRBS/All-Call Reply Generation for each interrogation for which replies are prepared. If no replies are to be generated for an interrogation, no recording takes place. The operator can select short or long form recording for this data, and can also disable recording of this block by means of teletype commands. Short form recording enabled is the mode in which the system is initialized. Short form records only targets which replied, while long form recording (see the following block description) records all targets that were in the Reply Time Sorted Index.

Word	Description
1	Block number = 9
2	Word count. Variable, with a minimum of 7.
3	Recording time. Note that since ATCRBS/All-Call Reply Generation is an interrupt driven task, it cannot directly record the data. Instead, the data is passed to a background task for recording. Thus the recording time will not be the exact time at which the reply data was generated.
4	The Interrogation Pattern File entry for the expected interrogation for which these replies will be the response. See Appendix E for the format of Interrogation Pattern File entries.
5	The predicted azimuth at which this interrogation should be received. LSB = $2\pi/2^{14}$ radians.
6	This has the same format as word 3 of the expected interrogation data block, containing the front/back and interrogation type fields. Please see Appendix E for the formats and field definitions.
7-end of block	Bits 0-5: Always 0. Bits 6-15: The track numbers (i.e., the low order 10 bits of the DABS ID) of targets for which replies were generated.

Title: ATCRBS/All-Call Replies - Long Form

Description: This block is similar to the short form ATCRBS/All-Call recording except that all targets listed in the Reply Time Sorted Index are recorded whether they replied or not. If this Index is empty, no recording takes place. The choice of long recording must be made by the operator using teletype commands. Recording of the block may also be disabled by teletype command. This block puts more of a load on the data recording function, but also provides more information.

Word	Description
1	Block number = 10
2	Word count. Variable length, minimum 7 words.
3	Recording time. (See the note for block #9).
4	The Interrogation Pattern File entry for the expected interrogation for which these replies are being generated. See Appendix E for the format of Interrogation Pattern File entries.
5	The predicted azimuth at which this interrogation should be received. LSB = $2\pi/2^{14}$ radian.
6	This has the same format as word 3 of the expected interrogation data block, containing the front/back and interrogation type fields. Please see Appendix E for the formats and field definitions.
7-end of block	Bits 0-3: Reply code. The possible values and their meanings are as follows:

0: Target replied.
1: Not used.
2: DABS target, L or I bit set in track record, no reply.
3: Zero reply probability, no reply.
4: Reply probability, random reply failure.
5: Target outside beam, no reply.
6: Target slant range < 1 nmi, no reply.
7: This reply would have overlapped the previous reply from its assigned reply generator, so the reply was suppressed.

Bits 4-5: Always zero.

Bits 6-15: The track number (i.e., the low order 10 bits of the DABS ID) of this target.

Title: DABS Interrogation - Short Form

Description: This is recorded by Interrogation Processing each time a discrete interrogation is processed. More than one interrogation may be recorded in this block, each pair of words representing a different interrogation. The operator may switch between long and short form recording or disable interrogation recording altogether by means of teletype commands. Short form recording enabled is the mode in which the system is initialized.

Word	Description
1	Block number = 11
2	Word count. Variable, minimum = 5.
3	Recording time. Note that since this is recorded by an interrupt driven task, the data must be passed to a background task before the actual recording can take place. Thus the recording time will not be the exact time at which the interrogation occurred.
4,6,8, etc.	The low order 16 bits of the DABS ID of the interrogated target.
5,7,9 etc.	Bit 0: 0 indicates a reply was generated. 1 indicates no reply was generated. Bits 8-15: If bit 0 is 0, these are the inter-ARIES message bits, as follows:

8							15
I	L	AR	T	CB	U	CP	DI

Please see Appendix E for a definition of these bits.

If bit 0 is 1, these bits encode the reason why no reply was generated. The following values have the meaning shown.

1: Target does not exist in track file.
2: I bit set in track file.
3: Zero reply probability.
4: Reply probability, random reply failure.
5: Target not in the beam.
6: Target within 1 nm. slant range.
7: The full 24 bit DABS ID did not match the track file ID, even though the low order 10 bits match.

Title: DABS Interrogation - Long Form

Description: This is recorded for each discrete interrogation. Only one interrogation is recorded in each block. Long form recording must be enabled by the operator via teletype command and replaces short form recording. All interrogation recording may be disabled by teletype command.

Word	Description
1	Block number = 12
2	Word count = 19
3	Recording time (see the note for short form recording).
4,5	Identical to short form recording (block #11).
6-9	The first 4 words of the interrogation data block. These include the time of interrogation, the boresight azimuth at interrogation, the interrogation type, and the first 16 data bits. See Appendix E for the detailed format and field definitions.
10-19	The entire reply data block as sent to the controlled reply generator. See Appendix E for the detailed format and field definitions. If no reply was generated the content of these words is undefined.

Title: Inter-ARIES Communications*

Description: This is recorded by Adjacent ARIES input for each block of messages received. Only messages received free of error and processed by Adjacent ARIES Input are recorded. Erroneous messages are handled independently by the modem input and output programs and are not seen by Adjacent ARIES Input.

Word	Description
1	Block number = 13
2	Word count. Variable, minimum = 6.
3	Recording time. This will be the time at which Adjacent ARIES Input begins to process the first message in the block.
4-end of block	The entire message block is recorded, starting with the STX character and ending with the ETX character (or the following character if there are an odd number of bytes). See Appendix E for the individual system status and track status message formats.

*Note: This data recording block has not yet been implemented.

APPENDIX G

ARIES ERROR CONDITIONS

All error conditions detected in ARIES are recorded in an array of error counters. Each possible error condition has its own counter, and this is incremented by one each time the particular error occurs. None of these errors cause ARIES to halt. If necessary, some sort of recovery action is taken to keep the system running. It is, of course, possible that some anomalous behavior may occur at about the same time as such an error. The counters may be read by executing operator command Number 4 from the teletype. This will display the current system status, including the values of non-zero error counts.

In addition to incrementing counters, some errors are also recorded on the data recording file. The tabulation below indicates which these are by means of an * next to the error number, and the nature of any optional debugging information included in the error recording (see Appendix B, Block No. 2 for the error recording format).

In the table below, the major system function which increments each error count is indicated. The acronyms used are as follows:

A/AIP	ATCRBS/All-Call Interrogation Processing
A/ARG	ATCRBS/All-Call Reply Generation
A/ADR	ATCRBS/All-Call Data Recording
DDR	Discrete (DABS) Data Recording
DIP	Discrete Interrogation Processing (also includes the receiver interrupt processing common to both this function and A/AIP.
DO	Data Output
ECCIH	Error Correcting Core Interrupt Handler
PIP	Pre-Interrogation Processing
SFIH	Status Formatter Interrupt Handler
TMI	Traffic Model Input

Error No.	System Function	Description
1	PIP/TMI	A divide overflow occurred while updating a track's position to the time of bore-sight crossing. The update was not performed.
2	PIP	There were too many targets to fit in the Reply Time Sorted Index being created by PIP. This may be incremented more than once for a given execution of PIP. It will be incremented once for every 5 targets beyond the maximum. At each increment the last 5 targets are dropped from the index (i.e., one bins' worth is discarded).

Error No.	System Function	Description
3*	PIP	The radar report buffer overflowed when PIP was trying to start a new report group (i.e., store the azimuth and initial count words). The new group was not started and reports will be added to the previous group.
4	PIP	Same as 3, from a different location.
5	PIP	The radar report buffer overflowed when an attempt was made to add a new radar report. The report was discarded.
6	TMI	No free track records were available when TMI tried to start a new track. The track was not started and the model record was discarded.
7	Unused error number	
8	A/AIP	A divide overflow occurred in calculating the updated antenna azimuth rate. The azimuth rate was not updated.
9	A/AIP	The fruit reply generator was not ready to accept new parameters when the software wanted to change the fruit rate.
10	A/AIP	The Interrogation Pattern File overflowed before a pattern was discovered. The pattern acquisition procedure was re-initialized.
11	A/ARG	The time since the last ATCRBS/All-Call interrogation was greater than 40 msec. While reply generation was performed as if no error had occurred, this error indicates a missing interrogation and ARIES may have lost synchronization with the interrogation pattern.
12	A/AIP	An ATCRBS/All-Call interrogation was received without a preceding interval timer interrupt. This indicates a timing problem. It could be caused by jitter in the interrogation times or extra or missing interrogations. Normal interrogation processing was performed.

Error No.	System Function	Description
13	A/AIP	The mode of the interrogation does not match the expected mode, or the interval does not match the expected interval to within 4 μsec. Normal interrogation processing was performed.
14	A/AIP	During the calculation of the antenna azimuth rate, the previous azimuth measurement time was found to be greater than the current system time. The update was continued normally.
15	A/AIP	During the calculation of the antenna azimuth rate, the difference of the current azimuth and the previous azimuth did not correspond to clockwise rotation of the antenna. The azimuth rate update was not completed.
16	DIP	An interrogation type was received that the ARIES software is not prepared to process. Included in this category are all the undefined receiver interrogations and DABS-only All-Calls. The interrogation was ignored.
17	A/AIP	An illegal interrogation type was passed to the A/AIP software. This error should never occur, as the test giving rise to error No. 16 should detect the error first.
18	DIP	A receiver buffer overflow interrupt occurred. The receiver was reinitialized and all interrogations in the buffer were discarded.
19	SFIH	A status change interrupt was received from the status formatter and the new status data read into the System Status File (see Appendix E). This occurs once as part of the normal system initialization procedures. If more than 512 interrupts are received during a run, the status formatter is disabled to prevent interference with the rest of the system.

Error No.	System Function	Description
20	FRGIH	An error interrupt was received from the fruit reply generator. This happens if a target card went for more than 5 msec without accepting new reply data. The fruit reply generator was reinitialized.
21	ECCIH	An interrupt was received from the error correcting memory, indicating that error correction was required. No recovery action is required, but the address of the failed word and the error syndrome are saved.
22*	A/ADR	A/ARG found all the buffer areas it uses to pass data to A/ADR full when it tried to record a group of replies. The reply data was lost. In the data recording for the error (Block No. 2) the optional data field indicates the number of interrogations for which data was discarded.
23*	DDR	DIP found the buffer it uses to transfer data to DDR full when it tried to record an interrogation. The data was discarded. In the data recording for the error (Block No. 2) the optional data field indicates the number of interrogations for which data was discarded.
24*	DO	An I/O error was detected by the operating system while writing a data recording buffer to disk. The operation was attempted again.
25	TMI	The disk input buffers were empty when an attempt was made to process the next model record. The Disk Input task was awakened and TMI terminated.
26	A/ARG	At the interval timer interrupt, when the primary and secondary buffers of the controlled reply generator were swapped to release the ATCRBS/All-Call replies, it was discovered that replies remained in the primary buffer. This probably indicates that some reply did not reach its assigned reply generator before its reply time occurred. The controlled reply generator was reinitialized.

APPENDIX H

AN ANALYSIS OF THE EFFECTS OF INTER-ARIES MESSAGE DELAYS

1.0 METHOD OF ANALYSIS

One purpose of the inter-ARIES track status messages is to keep the simulated DABS transponder state for a given target consistent among all ARIES sites. The existence of transmission delays on the inter-site links means that this goal cannot be perfectly realized, as there will always be delays between a state change at one site and the resulting changes at all other sites. The purpose of this appendix is to analyze this effect in detail and to point out areas where the simulated results will differ from a similar live aircraft test.

The state behavior of a DABS transponder as simulated by ARIES can be represented by a finite state machine diagram, in the usual way. The nodes of the graph represent the transponder's states, and the directed edges between the nodes represent allowable transitions. Each edge is labeled by input event(s) which cause the transition represented by that edge to be made. The set of all such input events is called the "input alphabet." A complete state diagram of this sort for the simulated transponders is very complex. Fortunately, it is composed of several smaller diagrams which, due to the nature of the protocols, can be treated independently. We will consider diagrams representing the all-call lockout protocol, the downlink request protocol, and the acknowledgment request protocol.

A system consisting of two ARIES sites will be considered here, but the analysis technique may be generalized to a larger number of sites. For a given protocol, the overall system state can be represented by two copies of the state diagram for that protocol, each copy representing the simulated transponder state at one of the sites. The desired result is that both these finite state machines be in identical states at all times, as they would then behave exactly as if they were a single real transponder. Thus, any condition which causes them to be in different states, however briefly, may cause a departure from a realistic simulation.

More formally, the result of combining the two finite state machines represented by the diagrams can be treated as a single finite state machine, known as the "cross-product" machine, resulting from these two machines. A finite state machine can be considered to be a structure consisting of a set of states and a set of transitions. The transitions, in turn, are 3-tuples consisting of a starting state, an ending state, and a letter from the (finite) input alphabet set which causes that transition to be made. For the cross-product machine, the states are the set cross-product of the state sets of the individual machines, and the input alphabet is the cross-product of the individual alphabets. The individual alphabets must be extended by a null character, which causes no transition, in order that the cross-product machine be able to represent cases where one of the component machines receives an input and the other does not. If A and B are states in one component machine, C and D are states in the other component machine, and X and Y represent letters in the two respective input alphabets, then the transition $AC \xrightarrow{(x,y)} BD$ exists in the cross-product machine if and only if $A \xrightarrow{x} B$ and $C \xrightarrow{y} D$ are transitions in the component machines.

In our case, the two state sets are identical. An <u>undesirable</u> state of the cross-product machine can then be defined as any state whose two component states are not identical, as this represents a condition where the simulated transponder appears to be in one state at one site, and a different state at another site. An <u>erroneous</u> state is one in which both component machines agree but are different from the state a real transponder would be in under identical conditions. Obviously, in an undesirable state one of the two sites can also obtain results different from a live aircraft test. The procedure followed here will be to:

1) Identify all such states.
2) Determine what input sequences can cause these states to occur. Since many of these states will be seen to occur transiently during normal operation, we will also be interested in determining input sequences which can cause semi-permanent occurances (i.e., longer than the message delay time).
3) Comment on the effects of being in these undesirable or erroneous states as far as the accuracy of the simulation results is concerned. Effects can be placed in two categories as far as seriousness. In the less serious cases, the results appear different from a live aircraft test, but do not differ from plausible results of a live aircraft test given slightly different timing, etc. In the more serious cases, the results are not possible in a live test and may violate some of the assumptions of the DABS software.

2.0 THE ALL-CALL LOCKOUT PROTOCOL

One of the simplest protcols to analyze, and therefore a good one to illustrate the technique, is the lockout protocol. This involves only two states at each site, these being the locked (L) and unlocked (U) states. The input "alphabet" is as follows:

Symbol	Meaning
IL	An inter-ARIES message was received to lockout this target. This is generated only when the other ARIES receives an L and the transponder was unlocked.

Symbol	Meaning
IU	An inter-ARIES message was received to unlock this target. This is generated only when the other ARIES receives a U and the transponder was locked.
L	An interrogation caused this target to be locked out to all-call interrogations.
T	Due to no discrete interrogations being received for 16 seconds, the transponder timed out.
U	An interrogation caused this target to be unlocked to all-call interrogations.
-	Null input - no transition is made.

Figure H-1 shows the state diagram for one site, and Figure H-2 shows the resulting cross-product machine for two sites. Note that in order to simplify the discussion of the cross-product machine, it has been assumed that simultaneous transitions of both component machines cannot occur. The net effect is to reduce the input alphabet of the cross-product machine from N^2 possibilities to 2N, where N is the size of the input alphabet of one component machine, and to eliminate certain transitions from the state diagram. The logical behavior does not change, however, as the state resulting from applying input (A, B) to any such cross-product machine is always equal to the states resulting from applying the sequences (A, -) (-, B) or (-, B) (A, -). The interval between the two inputs can be arbitrarily small. Each transition has been labeled with a number, and the set of input symbols that could cause each transition have been displayed separately.

The undesirable states are obviously UL and LU. It is also obvious that they are unavoidable, as any sequence involving, for example, (L, -) and later (-, IL) will go through one of the undesired states if the system starts in the state UU. This sequence will occur often, due to message delays. However, since an IL will always be transmitted by a site receiving an L that caused a state change, and similarly for U and UL, the situation is self-correcting with time. That is, given no further inputs from outside the two ARIES systems, the states will always end up in either the UU or LL states after an interval corresponding to the message delay.

This self-correcting characteristic also applies to cases when the two DABS sensors transmit contradictory information. An example would be the sequence (U, -) (-, L) (-, IU) when the system starts out in state LL. (Note: (IL, -) is not sent because the L did not cause a state change). The resultant state would be UU. In general, the "simultaneous" (i.e., within the message delay interval) transmission of conflicting commands will switch the lockout state at both sites if they both start in the same state. Note, however, that this can result in an "erroneous" state, as in the example. There, the L input would have been the last received by a real transponder, and yet the simulation ends up in the UU state.

In general, it can be proven that if:

1) The cross-product machine is started in either state UU or LL with no inter-ARIES messages in progress, and

2) A DABS sensor is not allowed to "change its mind" about the lockout state of a target during the inter-ARIES message delay (i.e., transmit two conflicting lockout commands within this interval),

then all instances of states UL and LU are transient in the sense that if no further lockout commands are received the system will eventually stabilize in state UU or LL. The proof is by induction on the length of the input string. Assumption (2) is a necessary assumption, as the sequence $UU \xrightarrow{(L,-)} LU \xrightarrow{(-,L)} LL \xrightarrow{(-,U)} LU \xrightarrow{(IL,-)} LU \xrightarrow{(-,IL)} LL \xrightarrow{(IU,-)} UL$ demonstrates.

What, then, are the consequences of being in one of these undesirable states? One of the sensors will receive an all-call response and the other will not if they both interrogate the target while the cross-product machine is in one of these states. At this point we must take advantage of knowledge of the DABS system's operation.

In the normal case, all DABS sensors will immediately lockout all DABS transponders as soon as they begin receiving replies to discrete interrogations. While a sensor in this case might receive extra all-call replies that it would not have received in the "real world", it will never receive any such after it itself has requested lockout. The DABS sensors do not exchange data about the lockout state of a transponder, and so the sensor cannot distinguish this case from normal operation.

Once a transponder is locked out, it remains in that state unless one of the following conditions occur:

1) It enters a region of airspace where transponders are required to be unlocked (usually at the fringes of DABS coverage areas).

2) No discrete replies are received by the local sensor for a period of time.

3) An adjacent sensor failure is inferred by the local sensor and the failed sensor has primary collision avoidance responsibility for the target in question.

In these cases, the sensor will attempt to unlock the transponder. If the other sensor does not sense a corresponding condition, it will attempt to maintain the lockout state.

However, from either sensor's point of view this situation does not appear significantly different from the corresponding "real world" situation. All-call replies can be received irrespective of the last lockout command sent by a sensor. The transponder's state will oscillate back and forth in both cases. The only discrepancy will be that in cases where both sensor's send discrete interrogations within the message delay interval, the simulated transponder will always change state irrespective of which command would actually have been received last, and so the simulated state may disagree with the state of a real transponder under similar circumstances.

3.0 THE AIR-INITIATED COMM-B PROTOCOL

In analyzing the air-initiated Comm-B protocol an additional simplifying assumption can be made. The design of the DABS system allows only one sensor to process air-initiated downlinks from a given target at any instant of time. This sensor is designated as being the primary sensor for that target. This results in the state diagrams at the two ARIES sites having different input alphabets, as only one site can have the downlink cleared from the ground and the other site is cleared by means of an inter-ARIES message. Figure H-3 shows the two independent state diagrams, and H-4 shows the resulting cross-product machine. A "0" state indicates that the B bit in the track record is set to 0, and so no downlink is in progress. A "1" state indicates B=1. The input alphabet is as follows:

Symbol	Meaning
B	The B bit in the track record was set from the traffic model file.
CB	A CB bit was received in an interrogation from the sensor.
ICB	An inter-ARIES CB bit was received.
-	Null character - no state transition occurs.

As can be seen from the diagram, the "undesirable" states are again frequently entered in the normal course of the air-initiated Comm-B protocol. However, due to the designation of one of the sensors as primary, this has no undesired effect on the behavior of the DABS system. Even though the value of the B bit at the secondary site may differ from the value at the primary site there is no effect on the secondary sensor as it ignores the received B bit entirely. Thus, for the most common case, the effective behavior reduces to that of the state machine for the primary site, and so the overall simulation precisely mirrors the "real world".

Difficulties can occur if the rule that only one of the sensors is primary for the duration of the transaction is violated. This could conceivably happen, for example, after a transition from state 11 to state 01.

At this point the primary sensor considers the transaction complete. If the target has flown into an area where the other sensor is primary, the two sensors will exchange status. If this all occurs before the 01 to 00 transition is made by the ARIES system then there is a possibility of an additional copy of the downlink message being transmitted.

Another problem can arise if the two ARIES system clocks are out of synchronism by an amount larger than the message delay. The sequence (B, -) (CB, -) (-, ICB) (-, B) is then possible, which leaves the protocol in state 01 indefinitely. If a primary - secondary switch should take place, an additional copy of the message will be transmitted. This should be prevented by the inter-ARIES synchronization checks.

In a similar vein, if the sequence (B, -) (-, B) (CB, -) (B, -) (-, B) (-, ICB) should occur, and primary/secondary status is switched following (CB, -), ARIES will end up in state 10. Since the former secondary is now primary, no Comm-B downlink will result even though expected, and the former primary site is in the same situation described at the end of the preceding paragraph.

There is one other situation which can occur which is not a departure from "real world" behavior but which may cause results different from those expected when the traffic model was generated. If two downlink requests appear close together on the model file, the second may be lost if the first has not been cleared when the second arrives (i.e., (B, -) or (-, B) is received in state 11). In this case, not as many Comm-B downlinks will appear as may have been expected by the persons generating the traffic model.

To summarize the paragraphs above, it is best not to attempt closely spaced air-initiated Comm-B replies from the same target, and it is best not to attempt any air-initiated Comm-B replies for targets near a primary/secondary transition region. Both of these conditions are under the control of the traffic model generation process.

4.0 THE ACKNOWLEDGMENT REQUEST PROTOCOL

The last protocol to be analyzed is that for the acknowledgment request. Figure H-5 presents a state diagram for the corresponding finite state machine at one site. The cross-product machine is not diagrammed due to its complexity. Note that in addition to the communications between the sensor and ARIES, account must be taken of the transponder timeout function and of the fact that the value of the PBUT bits can be changed at any time from the traffic model file, unless PB is set. The states are labeled by the PBUT state (X = no response, Y = will comply, N = cannot comply), the state of the AR bit in the track record, and the state of the PB bit. It is only in states Y11 or N11 that the sensor gets a PBUT response that causes it to take any action. The input alphabet is as follows:

Symbol	Meaning
AR	An acknowledgment request was received from the sensor.
CP	CP=1 was received in an interrogation.
IAR	AR=1 was received in an inter-ARIES message. This is sent whenever an acknowledgment request is received at the other site.
ICP	CP=1 was received in an inter-ARIES message. This will only be sent if CP occurred at the other site while it was in states Y11 or N11.
N	The value of PBUT was changed to "cannot comply" on the model file input records.
PB	The PB bit in the track was set to 1 by Pre-interrogation Processing.
T	The track did not receive discrete interrogations for 16 seconds and timed out.
X	The value of PBUT was changed to "no response" on the model file input records.
Y	The value of PBUT was changed to "will comply" on the model file input records.

Note that for the sake of clarity not all transitions are shown on the state diagram. There is not necessarily a transition out of every state for every symbol in the input alphabet. Transitions not shown either leave the process in the same state (for example, receipt of CP while in states X00, Y00, or N00) or cannot occur for a given state (PB cannot occur in states X00, Y00, or N00).

We will first analyze the behavior of this protocol assuming no changes in the PBUT value. The undesirable states of interest are those when one machine is in a PBUT reply state (Y11 or N11) and the other is not. States when the machines are in different but non-responding states are not distinguishable by the sensor. Due to message delays, it will frequently be true that one site is responding while the other has not yet reached a responding state. However, it is not necessarily true that the site receiving the initial AR will be the first site ready to generate a PBUT response. This is due to the need for the PB transition, which occurs just before the aircraft enters the antenna beam. Depending on the antenna azimuths of the two sites, a sequence such as (AR, -) (-, IAR) (-, PB) is possible. This corresponds to a live aircraft test, however, in that the first sensor to scan by the aircraft after the pilot responds will obtain the acknowledgment. The only difference is that the minimum interval between the AR and the response is limited only by the message delay and may be shorter than the real delays in the message display designed to assure that the pilot is responding to the correct message.

Once an AR is received, the only input that can prevent either site from responding is an ICP. (T is ignored as not being relevant when the sensors are working properly). Receipt of additional AR's before responding to earlier ones may mean that there will not be a one-to-one correspondence between AR's and PBUT responses, but this again is realistic. With a real transponder it is always true that the last AR received (i.e., one that is not overridden by a subsequent AR before PBUT is generated) will result in a PBUT response (the assumption is that the pilot always responds). This can also be shown to be true with ARIES, although the sensor transmitting the AR may not receive the resulting PBUT. The proof involves the fact that receipt of the AR causes an IAR to be sent to the other site. This is the last message on that line, and so the other site will always respond unless either another AR (or IAR) is received or an ICP is received. The latter case implies that the site receiving the initial AR responded. The sequence (AR, -) (-, IAR) (-, PB) (-, CP) (AR, -) (ICP, -) ... demonstrates a case where the second AR is lost at the receiving site.

Note that because of message delays, responses will sometimes be generated that would not have been in a live aircraft test. An example is (AR, -) (-, IAR) (-, PB) (AR, -) (-, CP) (-, IAR) where the second AR would, in a real transponder, have prevented the response.

If we now consider the full cross-product machine, including the PBUT transitions, there are some additional undesirable states which can result, namely those where PBUT has different values at the two sites. One way in which this can happen is demonstrated by the following sequence: (AR, -) (-, IAR) (PB, -) (N, -) (-, N) (-, PB) (CP, -) (-, CP) (-, ICP) (ICP, -). Assuming the protocol is started in state Y00 Y00 then site 1 will send back PBUT=Y and site 2 will send back PBUT=N. The input character (N, -) has no effect, as site 1 is in state Y11 at that point, where PBUT changes are ignored (the pilot is not allowed to "change his mind").

The significance of the foregoing is that when generating traffic models it is best to avoid changing PBUT when acknowledgment requests are likely to be generated (or, better still, each target should be given a fixed PBUT value for the duration of the model).

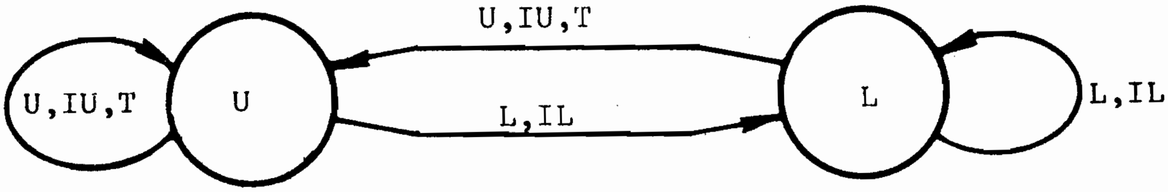
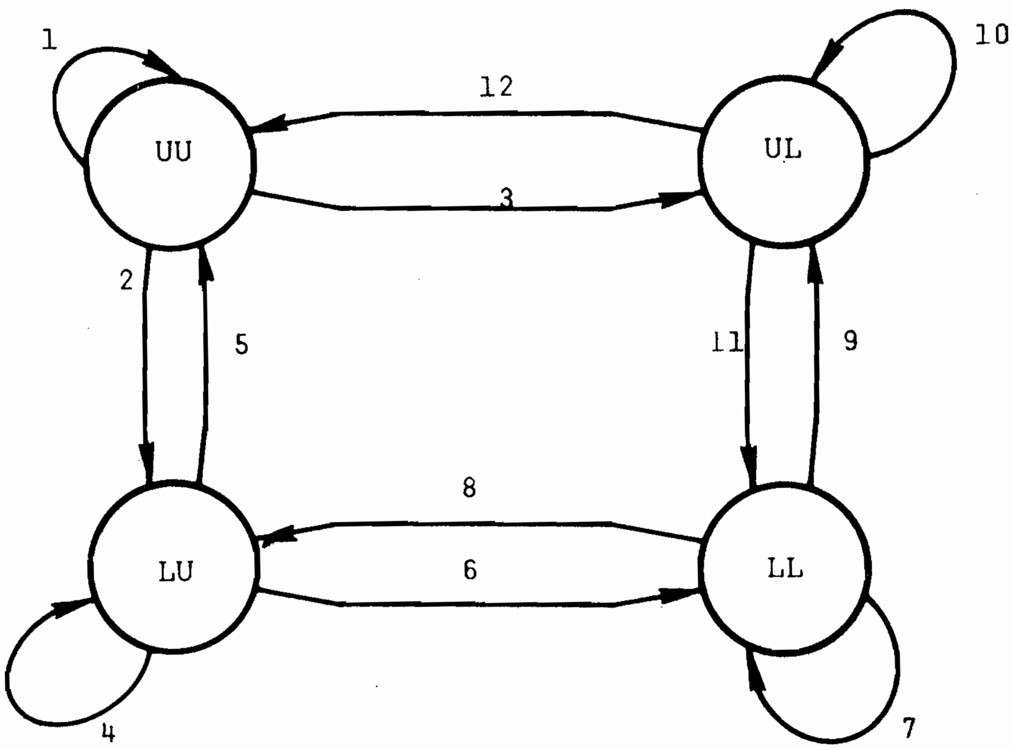


Fig.H-1. All-call lockout state diagram.



1: {(U,-) (IU,-) (T,-) (-,U) (-,IU) (-,T)}
2: {(L,-) (IL,-)}
3: {(-,L) (-,IL)}
4: {(L,-) (IL,-) (-,U) (-,IU) (-,T)}
5: {(U,-) (IU,-) (T,-)}
6: {(-,L) (-,IL)}
7: {(L,-) (IL,-) (-,L) (-,IL)}
8: {(-,U) (-,IU) (-,T)}
9: {(U,-) (IU,-) (T,-)}
10: {(-,L) (-,IL) (U,-) (IU,-) (T,-)}
11: {(L,-) (IL,-)}
12: {(-,U) (-,IU) (-,T)}

Fig.H-2. Cross-product machine derived from H-1.

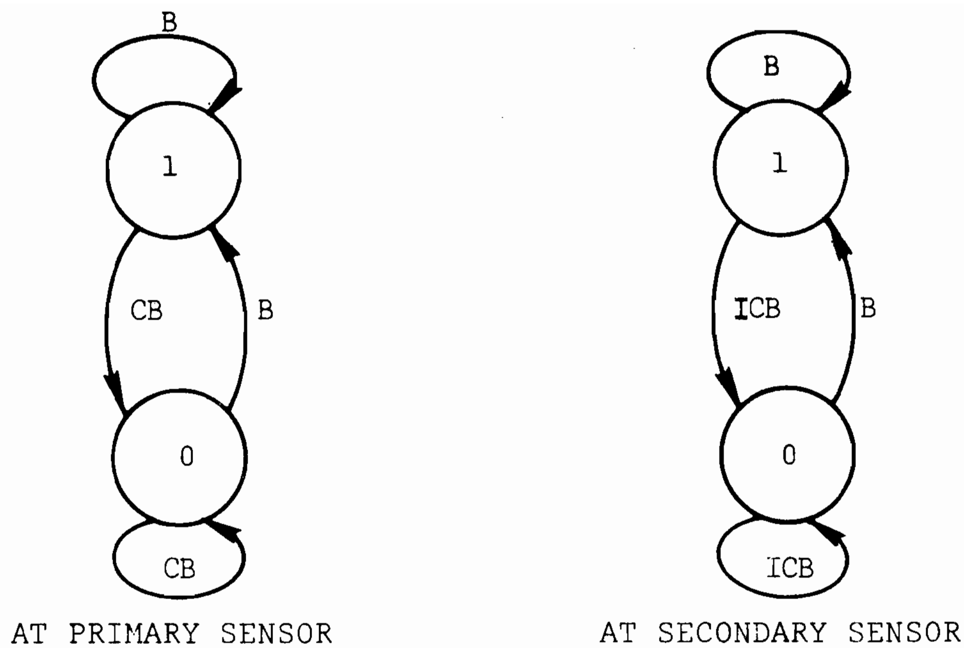


Fig.H-3. Air initiated comm-B state diagram.

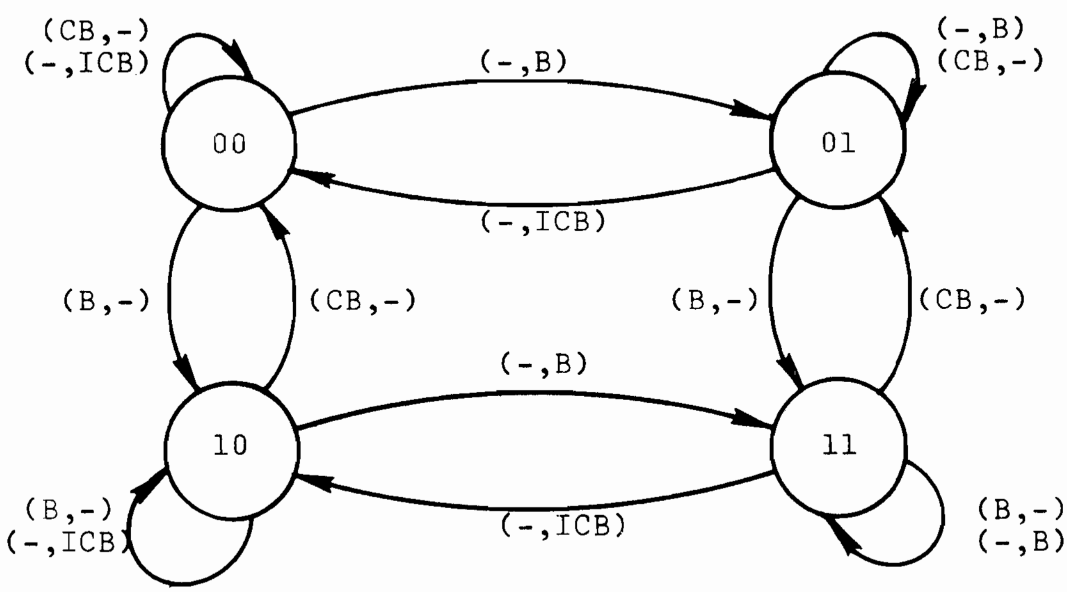


Fig.H-4. Cross-product machine resulting from H-3.

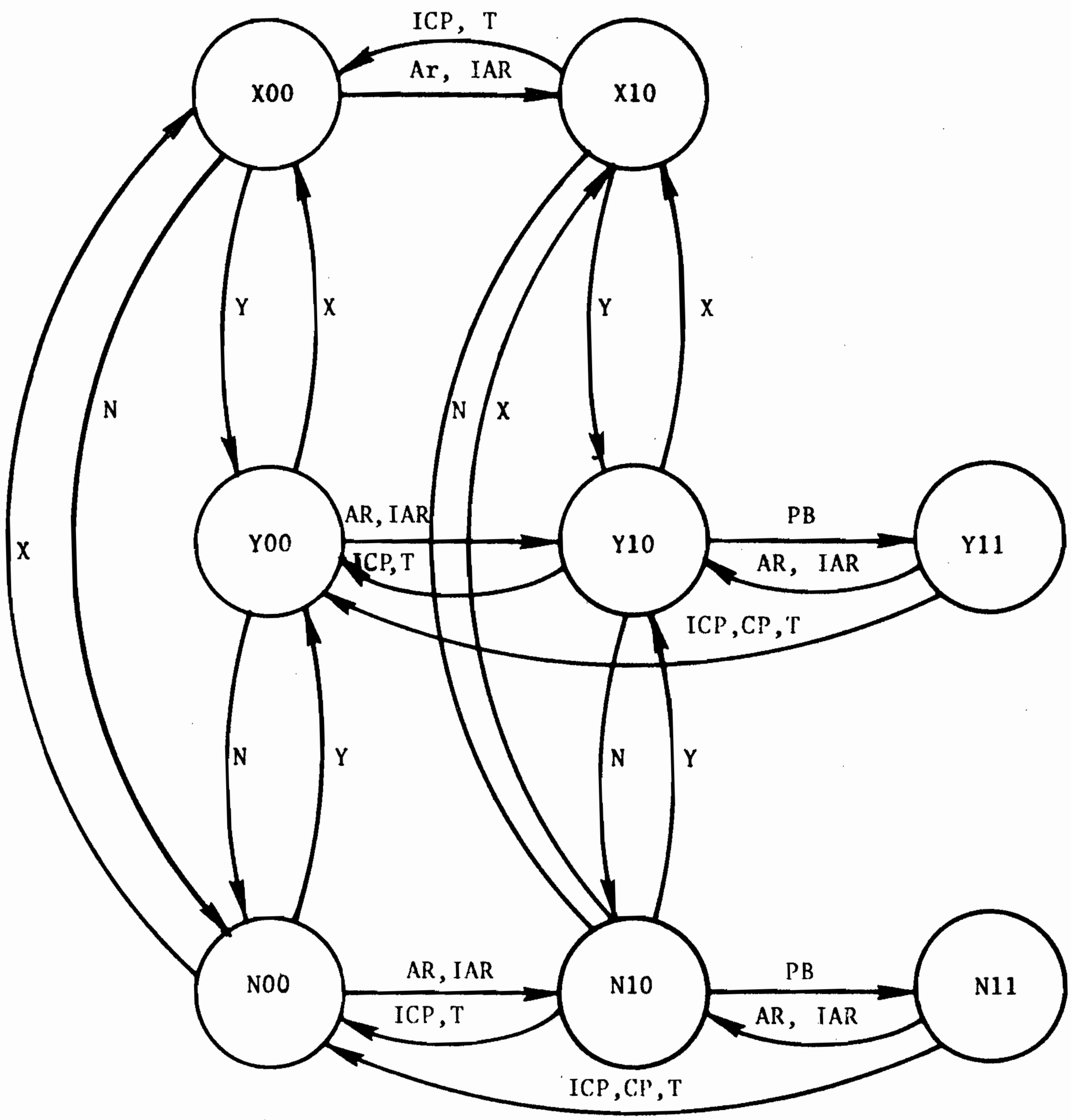


Fig.H-5. Acknowledgement request protocol.