

UDR-TR-79-72

DEVELOPMENT AND EVALUATION OF L/ESS
DATA COMPRESSION TECHNIQUES

Ronald K. Newman
Ronald M. Janning

University of Dayton
Research Institute
Dayton, Ohio

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

August 1979

Contract Number F33615-78-C-5208

Prepared for:

Aeronautical Systems Division (ASD/ENFSL)
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio

DTIC QUALITY INSPECTED 3

19980114 153

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Development and Evaluation of L/ESS Data Compression Techniques		5. TYPE OF REPORT & PERIOD COVERED
7. AUTHOR(s) Ronald K. Newman Ronald M. Janning		6. PERFORMING ORG. REPORT NUMBER UDR-TR-79-72
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Dayton Research Institute 300 College Park Avenue Dayton, Ohio 45469		8. CONTRACT OR GRANT NUMBER(s) F33615-78-C-5208
11. CONTROLLING OFFICE NAME AND ADDRESS Aeronautical Systems Division (ASD/ENFSL) Air Force Systems Command Wright-Patterson Air Force Base, OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE August 1979
		13. NUMBER OF PAGES
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes the programs developed for the evaluation of two different methods of compressing the data from the loads/environmental spectra survey (L/ESS) program on three different aircraft. The two different compression techniques involve: (a) saving sections of data when any parameter exceeds a given threshold, and (b) saving the value of an individual parameter when it changes by more than a given window value.		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Results of the application of these two compression techniques applied to data recorded on the A-10, C-130, and C-141 aircraft are presented. Complete documentation of the programs is provided for the user and for the programmer who might wish to use the methods on these aircraft or other aircraft.

UNCLASSIFIED

FOREWORD

This report was prepared by the Aerospace Mechanics Section of the University of Dayton Research Institute, Dayton, Ohio, under Air Force Contract NO. F33615-78-C-5208. The work was initiated and monitored by the Loads Branch, Structures Division, Air Frame Directorate, Deputy for Engineering, Aeronautical Systems Division, Wright-Patterson Air Force Base, Ohio with Mr. Steven Guilfoos and Mr. Cyril Peckham as technical monitors.

The project was conducted under the general direction of Dale H. Whitford, Supervisor, Aerospace Mechanics Section with Mr. Ronald K. Newman as Project Engineer. Ronald M. Janning provided the programming effort and assisted in documentation.

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
1 INTRODUCTION	1
2 PROGRAM INPUT	2
2.1 R/T TAPE FORMAT	2
2.2 CARD INPUT	14
2.2.1 Input to COMPAl	14
2.2.2 Input to COMPA2	15
2.2.3 Input to COMPB1	15
2.2.4 Input to COMPB2	16
3 DESCRIPTION OF PROGRAMS	17
3.1 METHOD A	18
3.1.1 COMPAl Program	20
3.1.2 COMPA2 Program	20
3.2 METHOD B	20
3.2.1 Program COMPB1	21
3.2.2 Program COMPB2	23
3.3 SUBROUTINE PACK	23
3.4 SUBROUTINE UNPACK	24
4 PROGRAM OUTPUT	25
5 RESULTS	53
6 CONCLUSIONS AND RECOMMENDATIONS	63
APPENDIX	
A DATA COMPRESSION PROGRAM LISTINGS	64

LIST OF ILLUSTRATIONS

Figure		Page
2.1	BIT and CAL Cycles	12
4.1	Method A Output	27
4.2	Method B Output	30
A.1	Flowchart of Method A	120
A.2	Flowchart of Method B	128

LIST OF TABLES

Table		Page
2.1	General Description of R/T Output Tape	3
2.2	ECU-68/A Multiplexing Format A-10	4
2.3	ECU-67/A Multiplexing Format C-141A/C-130	6
2.4	Documentary Data Encoder	13
3.1	Sampling Rates of Parameters for C-130/C-141	22
4.1	COMPAl Tape Output Format	33
4.2	COMPAl Tape Output Format	38
4.3	COMPBl Tape Output Format Record Type 1	43
4.4	COMPBl Tape Output Format Record Type 1	48
5.1	Active Data by Parameter Using Method A (A-10)	54
5.2	Active Data by Parameter Using Method A (C-141)	55
5.3	Active Data by Parameter Using Method A (C-130)	56
5.4	A-10 Method B Samples of Active Data	57
5.5	C-141 Method B Samples of Active Data	58
5.6	Comparison of Methods for C-141 Data	60
5.7	Comparison of Methods for A-10 Data	62
A.1	Program COMPAl	66
A.2	Program COMPAl	78
A.3	Program COMPBl	90
A.4	Program COMPBl	101
A.5	Subroutine Pack/Unpack	113
A.6	Programs COMPAl and COMPAl Definition of Variables	115
A.7	Program COMPBl and COMPBl Definition of Variables	118

SECTION 1

INTRODUCTION

Four separate computer programs were written for the CYBER 175 computer at Wright-Patterson Air Force Base for use in evaluating two different data compression techniques on two different aircraft types. Two different programs were written for the two aircraft types utilizing the same compression technique in order to simplify programming and provide more efficient programs.

The input for the data compression is the Reformatter/Transcriber (R/T) tape generated at Oklahoma City Air Logistics Center by reformatting and transcribing the data recorded on the MXU-553/A Airborne Signal Recorder. The first compression method evaluated (Method A) saves all recorded data during a second when any of the selected parameters examined deviates by more than a specified value from its prescribed mean or when any flight condition changes by more than a specified value from the previous stored value.

The second compression method (Method B) saves values of each individual parameter when it changes by more than a specified value from the previous stored value of that parameter. The maximum (Min) excursion upon change of direction of the parameter is also saved as a "peak". The time of occurrence of these window crossings and peaks is stored with the value.

Comparison of the two compression methods applied to A-10, C-130 and C-141 data are presented in Section 5. A real test of the degree of compression attained with the two methods can be obtained from the sample problem data, for the A-10, presented in Figures 4.1 and 4.2. The total amount of data on the RT tape was 52,810 seconds (102.6×10^6 bits). Using Method A, there were 28,657 records (43.0×10^6 bits) written to the output tape for a compression ratio of 2.36. With Method B, there were 3554 records (3.82×10^6 bits) written to the output tape for a compression ratio of 26.9. These results were obtained with the window values in Method B equal to the threshold values used in Method A where applicable. Representative window values were chosen for all other parameters

examined in Method B that are not analyzed in Method A. Ground data were processed in both of these runs. The Central Processor times used on these runs were 220 seconds and 360 seconds for Method A and Method B, respectively.

One additional run of Method A and Method B was made on the A-10 data, not processing ground data, at the request of the Project Engineer. For Method A thresholds of 23, 13, 8, 13, 13, 4, 4, and 4 were used for NZ, NY, P, Q, R, ALT, A/S and FCTR, respectively. For Method B windows of 7, 6, 10, 9, 9, 12, 14, 14, 4, 1, 1, 8, 4, and 4 were used for NY, NZ, P, Q, R, PDOT, QDOT, RDOT, RCTR, PLA, DSB, SIGMA, ALT, and A/S, respectively. The compression ratios of Method A and Method B were 3.51 and 11.52, respectively.

SECTION 2

PROGRAM INPUT

The input to the data compression programs comes from two sources. Program control is provided by card input and the data to be compressed comes from tape. The tape is that produced by the R/T located at Oklahoma City Air Logistics Center.

2.1 R/T TAPE FORMAT

The output tape from the R/T, the input tape for the computer, contains the inflight recorded data plus some supplementary or header data entered via the ASR33 teletype unit which is part of the R/T. The R/T has the capability of writing either a 7-track or a 9-track tape. Only the 9-track format will be discussed here.

The data on the 9-track cartridge tape is written in physical records of 479 8-bit bytes and the physical records are separated by a one-byte gap. The R/T adds seven 8-bit samples of information to each 479-sample physical record of data, making a set of 486 8-bit samples of data. The R/T groups five of these 486-sample data sets together and writes them on the output tape as a single physical record containing 2430 data samples.

Each set of 486 samples is referred to as a "frame" of data. There are two seconds worth of data in each frame. The first 239 samples are for the first second. The next 240 samples are for the following second of data. The final seven samples of data, supplied by the R/T, contain information regarding the quality of the "frame" of data as it was recorded on the tape cartridge, and are discussed in Table 2.1. Thus, a physical record containing 5 frames contains 10 seconds of data.

The content of each of the 479 samples of data contained in a frame of the R/T output tape for the A-10 is shown in Tables 2.2. The content of the tape for the C-130 and C-141A aircraft is shown in Table 2.3. The two sample numbers refer to the two seconds of data contained in each frame. The following should be

TABLE 2.1

GENERAL DESCRIPTION OF R/T OUTPUT TAPE

Byte or Sample	Description
1-479	Each byte corresponds to a respective multiplexer recording format (Table 2.2). That is, byte 1 is multiplexer sample 2, byte 2 is sample 3, ..., byte 479 is sample 480. (Note that sample 1 for the multiplexer is really a one-byte cartridge tape gap which is not transcribed or represented on the output tape.)
480	This byte or sample represents two pieces of information. The four least significant bits are used to designate whether a normal length record was read from the recorder cartridge. When a normal length record is read, an integer nine is represented in these bits. Any other bit pattern indicates a short frame, hence the data in the frame may be invalid. The four most significant bits of this byte are used to indicate an end-of-flight, or more realistically a beginning of flight, since these bits are all on for a minimum of six frames and a maximum of ten frames after the beginning of a flight.
481-482	Value from 1 to 479 which indicates the last correct input byte in those frames in which parity errors have occurred. These two bytes together must be interpreted as a 16-bit unsigned binary integer. (If there are no parity errors the value is set to 482.)
483-484	Number of parity errors encountered by the playback hardware in reading this frame of data from the recorder cartridge. It is also interpreted as a 16-bit unsigned integer.
485-486	If all bits are ones, this is a padding frame which follows the final data frame. At least one such padding frame will always be written on the output tape. More than one will be written if required to complete the last physical record on the tape. If all bits are <u>not</u> ones, then this 16-bit unsigned integer contains the number of this frame as counted from the beginning of the tape.

TABLE 2.2
ECU-68/A MULTIPLEXING FORMAT A-10

Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter
1&241	tape gap & T	31&271	q	61&301	r	91&331	N _y
2&242	V _i	32&272	N _z	62&302	p	92&332	σ
3&243		33&273		63&303	q	93&333	r
4&244	PLA	34&274		64&304	N _z	94&334	p
5&245		35&275		65&305	DD1	95&335	q
6&246	δ_{SB}	36&276	PLA	66&306	DD2	96&336	N _z
7&247	FCTR	37&277		67&307	DD3	97&337	DD7
8&248		38&278	δ_{SB}	68&308	PLA	98&338	DD8
9&249		39&279	FCTR	69&309		99&339	h _p
10&250		40&280		70&310	δ_{SB}	100&340	PLA
11&251	N _y	41&281		71&311	FCTR	101&341	
12&252	σ	42&282		72&312		102&342	δ_{SB}
13&253	r	43&283	N _y	73&313		103&343	FCTR
14&254	p	44&284	σ	74&314		104&344	
15&255	q	45&285	r	75&315	N _y	105&345	
16&256	N _z	46&286	p	76&316	σ	106&346	
17&257	E ₁ -E ₂	47&287	q	77&317	r	107&347	N _y
18&258	E ₅ -E ₉	48&288	N _z	78&318	p	108&348	σ
19&259		49&289		79&319	q	109&349	r
20&260	PLA	50&290		80&320	N _z	110&350	p
21&261		51&291		81&321	DD4	111&351	q
22&262	δ_{SB}	52&292	PLA	82&322	DD5	112&352	N _z
23&263	FCTR	53&293		83&323	DD6	113&353	DD9
24&264		54&294	δ_{SB}	84&324	PLA	114&354	DD10
25&265		55&295	FCTR	85&325		115&355	DD11
26&266		56&296		86&326	δ_{SB}	116&356	PLA
27&267	N _y	57&297		87&327	FCTR	117&357	
28&268	σ	58&298		88&328		118&358	δ_{SB}
29&269	r	59&299	N _y	89&329		119&359	FCTR
30&270	p	60&300	σ	90&330		120&360	

TABLE 2.2 (Concluded)
ECU-68/A MULTIPLEXING FORMAT A-10

Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter
121&361		151&391	FCTR	181&421		211&451	
122&362		152&392		182&422	δ_{SB}	212&452	PLA
123&363	N_Y	153&393		183&423	FCTR	213&453	
124&364	σ	154&394		184&424		214&454	δ_{SB}
125&365	r	155&395	N_Y	185&425		215&455	FCTR
126&366	p	156&396	σ	186&426		216&456	
127&367	q	157&397	r	187&427	N_Y	217&457	
128&368	N_Z	158&398	p	188&428	σ	218&458	
129&369	DD12	159&399	q	189&429	r	219&459	N_Y
130&370		160&400	N_Z	190&430	p	220&460	σ
131&371		161&401		191&431	q	221&461	r
132&372	PLA	162&402		192&432	N_Z	222&462	p
133&373		163&403		193&433		223&463	q
134&374	δ_{SB}	164&404	PLA	194&434		224&464	N_Z
135&375	FCTR	165&405		195&435		225&465	
136&376		166&406	δ_{SB}	196&436	PLA	226&466	
137&377		167&407	FCTR	197&437		227&467	
138&378		168&408		198&438	δ_{SB}	228&468	PLA
139&379	N_Y	169&409		199&439	FCTR	229&469	
140&380	σ	170&410		200&440		230&470	δ_{SB}
141&381	r	171&411	N_Y	201&441		231&471	FCTR
142&382	p	172&412	σ	202&442		232&472	
143&383	q	173&413	r	203&443	N_Y	233&473	
144&384	N_Z	174&414	p	204&444	σ	234&474	
145&385		175&415	q	205&445	r	235&475	N_Y
146&386		176&416	N_Z	206&446	p	236&476	σ
147&387		177&417		207&447	q	237&477	r
148&388	PLA	178&418		208&448	N_Z	238&478	p
149&389		179&419		209&449		239&479	q
150&390	δ_{SB}	180&420	PLA	210&450		240&480	N_Z

TABLE 2.3

ECU-67/A MULTIPLEXING FORMAT C-141A/C-130

NOTE: C-130 has same format with S1 instead of NY

Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter
1	GAP	21	NY/S1	41	PS	61	NY/S1	81	DELR	101	NY/S1
2	PITCH	22	STR4	42	STR2	62	PITCH	82	STR4	102	STR2
3	YAW	23	STR5	43	STR3	63	YAW	83	STR5	103	STR3
4	NZ	24	DELF	44	NZ	64	DELE	84	NZ	104	STR6
5	NY/S1	25	DELNG	45	NY/S1	65	E1-4	85	NY/S1	105	DELR
6	STR2	26	PITCH	46	STR4	66	STR2	86	PITCH	106	STR4
7	STR3	27	YAW	47	STR5	67	STR3	87	YAW	107	STR5
8	STR6	28	NZ	48	PT	68	NZ	88	DELE	108	NZ
9	DELR	29	NY/S1	49	DD1	69	NY/S1	89	DD2	109	NY/S1
10	STR4	30	STR2	50	PITCH	70	STR4	90	STR2	110	PITCH
11	STR5	31	STR3	51	YAW	71	STR5	91	STR3	111	YAW
12	NZ	32	STR6	52	NZ	72	DELF	92	NZ	112	DELE
13	NY/S1	33	DELR	53	NY/S1	73	DELNG	93	NY/S1	113	E5-9
14	PITCH	34	STR4	54	STR2	74	PITCH	94	STR4	114	STR2
15	YAW	35	STR5	55	STR3	75	YAW	95	STR5	115	STR3
16	DELE	36	NZ	56	STR6	76	NZ	96	DD3	116	NZ
17	VG	37	SY/S1	57	DELR	77	NY/S1	97	DD4	117	NY/S1
18	STR2	38	PITCH	58	STR4	78	STR2	98	PITCH	118	STR4
19	STR3	39	YAW	59	STR5	79	STR3	99	YAW	119	STR5
20	NZ	40	DELE	60	NZ	80	STR6	100	NZ	120	DELF

TABLE 2.3 (Continued)
ECU-67/A MULTIPLEXING FORMAT C-141A/C-130

Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter
121	DELNG	141	NY/S1	161	VG	181	NY/S1	201	DELNR	221	NY/S1		
122	PITCH	142	STR4	162	STR2	182	PITCH	202	STR4	222	STR2		
123	YAW	143	STR5	163	STR3	183	YAW	203	STR5	223	STR3		
124	NZ	144	DD6	164	NZ	184	DELE	204	NZ	224	STR6		
125	NY/S1	145	DD7	165	NY/S1	185	DD3	205	NY/S1	225	DELNR		
126	STR2	146	PITCH	166	STR4	186	STR2	206	PITCH	226	STR4		
127	STR3	147	YAW	167	STR5	187	STR3	207	YAW	227	STR5		
128	STR6	148	NZ	168	DELF	188	NZ	208	DELE	228	NZ		
129	DELNR	149	NY/S1	169	DELNG	189	NY/S1	209	DELP	229	NY/S1		
130	STR4	150	STR2	170	PITCH	190	STR4	210	STR2	230	PITCH		
131	STR5	151	STR3	171	YAW	191	STR5	211	STR3	231	YAW		
132	NZ	152	STR6	172	NZ	192	DD9	212	NZ	232	DELE		
133	NY/S1	153	DELNR	173	NY/S1	193	DD10	213	NY/S1	233	DD11		
134	PITCH	154	STR4	174	STR2	194	PITCH	214	STR4	234	STR2		
135	YAW	155	STR5	175	STR3	195	YAW	215	STR5	235	STR3		
136	DELE	156	NZ	176	STR6	196	NZ	216	DELF	236	NZ		
137	DD5	157	NY/S1	177	DELNR	197	NY/S1	217	DELNG	237	NY/S1		
138	STR2	158	PITCH	178	STR4	198	STR2	218	PITCH	238	STR4		
139	STR3	159	YAW	179	STR5	199	STR3	219	YAW	239	STR5		
140	NZ	160	DELE	180	NZ	200	STR6	220	NZ	240	DD12		

TABLE 2.3 (Concluded)

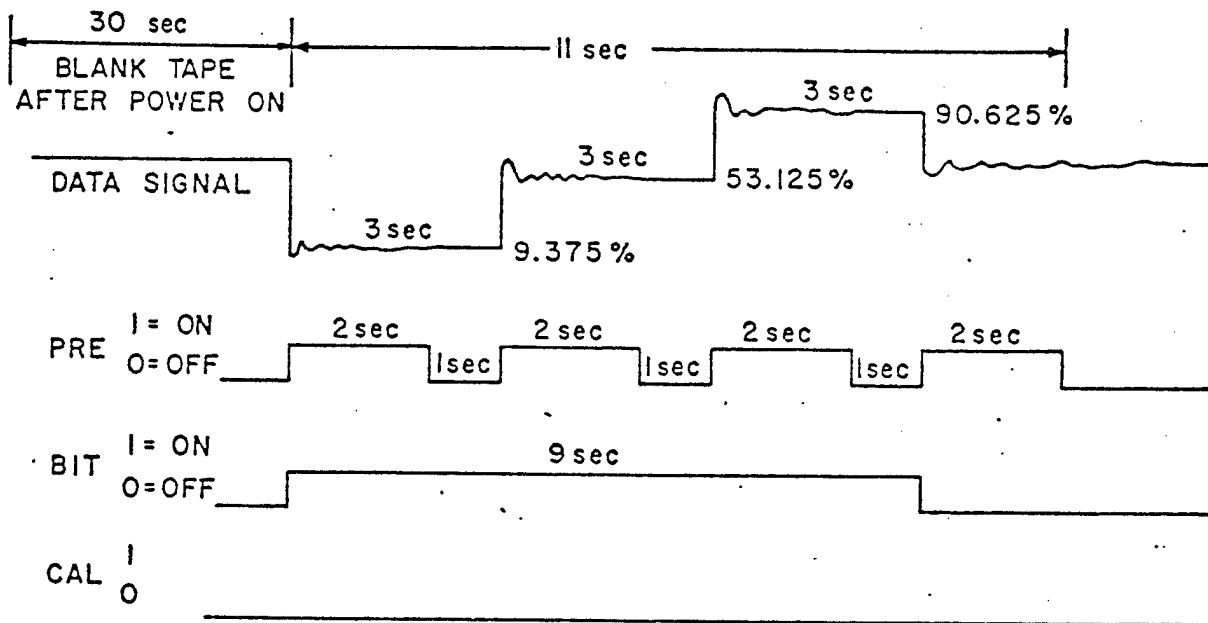
ECU-67/A MULTIPLEXING FORMAT C-141A/C-130

Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter
361	DELNG	381	NY/S1	401	VG	421	NY/S1	441	DELR	461	NY/S1		
362	PITCH	382	STR4	402	STR2	422	PITCH	442	STR4	462	STR2		
363	YAW	383	STR5	403	STR3	423	YAW	443	STR5	463	STR3		
364	NZ	384	DD6	404	NZ	424	DELE	444	NZ	464	STR6		
365	NY/S1	385	DD7	405	NY/S1	425	DD8	445	NY/S1	465	DELR		
366	STR2	386	PITCH	406	STR4	426	STR2	446	PITCH	466	STR4		
367	STR3	387	YAW	407	STR5	427	STR3	447	YAW	467	STR5		
368	STR6	388	NZ	408	DELF	428	NZ	448	DELE	468	NZ		
369	DELR	389	NY/S1	409	DELNG	429	NY/S1	449	DELP	469	NY/S1		
370	STR4	390	STR2	410	PITCH	430	STR4	450	STR2	470	PITCH		
371	STR5	391	STR3	411	YAW	431	STR5	451	STR3	471	YAW		
372	NZ	392	STR6	412	NZ	432	DD9	452	NZ	472	DELE		
373	NY/S1	393	DELR	413	NY/S1	433	DD10	453	NY/S1	473	DD11		
374	PITCH	394	STR4	414	STR2	434	PITCH	454	STR4	474	STR2		
375	YAW	395	STR3	415	STR3	435	YAW	455	STR5	475	STR3		
376	DELE	396	NZ	416	STR6	436	NZ	456	DELF	476	NZ		
377	DD5	397	NY/S1	417	DELR	437	NY/S1	457	DELNG	477	NY/S1		
378	STR2	398	PITCH	418	STR4	438	STR2	458	PITCH	478	STR4		
379	STR3	399	YAW	419	STR5	439	STR3	459	YAW	479	STR5		
380	NZ	400	DELE	420	NZ	440	STR6	460	NZ	480	DD12		

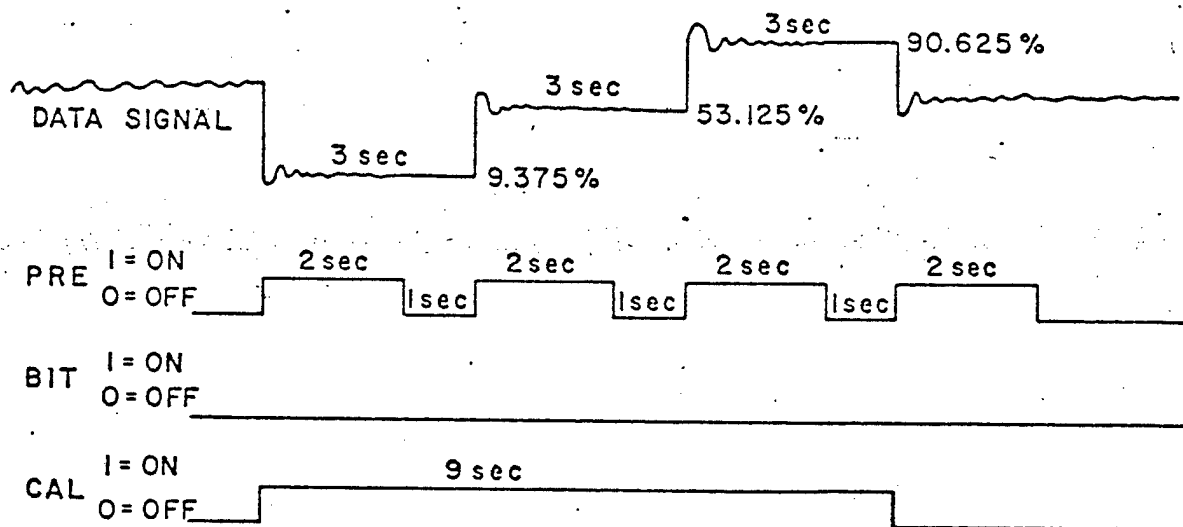
carefully noted by the reader! Sample 1 as indicated in Table 2.2 and Table 2.3 is really the one-byte tape gap on the cartridge tape and is not written on the R/T output tape, thus, even though Table 2.2 and Table 2.3 indicate 480 data samples, only 479 (plus the additional 7) are actually written out.

A description of the notations used in Table 2.2 is as follows:

T/CTR	=	frame counter (0-255)
V_i	=	differential pressure
PLA	=	power lever angle
δ_{SB}	=	speed brake position
FCTR	=	fuel counter
N_y	=	lateral acceleration
σ	=	strain
r	=	yaw rate
p	=	roll rate
q	=	pitch rate
N_z	=	normal acceleration
$E_1 - E_4$	=	events 1-4 stored in 4 high order bits
$E_5 - E_9$	=	events 5-9 stored in 5 high order bits
DD1-DD12	=	documentary data (see Table 2.4)
h_p	=	static pressure
STR1-STR6	=	strain 1 - strain 6
DELE	=	elevator deflection
VG	=	ground speed
DELF	=	flap deflection
DELNG	=	nose gear angle
PS	=	static pressure
PT	=	total pressure
DELR	=	rudder deflection
DELP	=	cabin pressure



(a) BUILT-IN TEST (BIT) CYCLE



(b) CALIBRATION TEST (CAL) CYCLE

Figure 2.1. BIT and CAL Cycles.

TABLE 2.4
DOCUMENTARY DATA ENCODER

Discrete Data	Number of Decimal Digits	Programmer Samples for BCD Formatting (Ref. MIL-C-83166 Slash Specifications)	
		1st Sample	2nd Sample
Aircraft serial number	4	DD ₁	DD ₂
Gross weight	4	DD ₃	DD ₄
Fuel weight	4	DD ₅	DD ₆
Base	2	DD ₇	
Mission	2	DD ₈	
Configuration	3	DD ₉	*DD ₁₀ (1st half)
		DD ₁₀ (2nd half)	
Date	5	DD ₁₁ **	DD ₁₂ ***

*The first digit of DD₁₀ is the third digit of configuration and the second digit of DD₁₀ is the last digit of the year

DD ₁₀ 1st half	DD ₁₀ 2nd half
2^7 2^6 2^5 2^4	2^3 2^2 2^1 2^0
third digit of configuration	last digit of year

**This sample also contains the BIT, CAL, and PRE indicators

DD₁₁

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
BIT	CAL	PRE	Tens digits of month	Unit digits of month			

***This sample is broken down as follows:

DD₁₂

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Power on	AGE on	Tens digit of day		Units digit of day			

NOTE: The bits in the above samples are represented as powers of two for clarity.

The CAL cycle, as shown in Figure 2.1, is very similar to the BIT cycle. No blank tape is run off, however. This calibration cycle is triggered every 90 gaps (or 3 minutes) by an automatic system unless an auto-cal disable switch on the recorder control panel is turned to "off". This automatic cal is normally not used due to an excessive loss of data.

For each indicator, BIT, CAL, and PRE, a value of one indicates it is on and a value of 0 indicates it is off.

In summary, the computer input data tape contains frames having 486 8-bit bytes. Five frames of 486 8-bit bytes make up one physical record. The reformatted tape should begin with a frame of header information followed by a BIT cycle. The end-of-flight is designated by all ones in the first four bits of byte 480 of the frame. The last frame of data on the tape is followed by at least one complete frame with all bits on. Following the end of the last physical record would be an octal 17 end of file mark.

2.2 CARD INPUT

Card input for the four compression programs varies because of the number of parameters being examined and because of the two different methods of data compression. Therefore, the inputs are specified for each of the programs in the following paragraphs. The programs are COMPAL, COMPA2, COMPB1, and COMPB2 representing method A for the A-10, Method A for the C-130/C-141, Method B for the A-10, and Method B for the C-130/C-141 respectively.

2.2.1 Input to COMPAL

This is used for Method A on the A10 tape format:

<u>Column</u>	<u>Description</u>	<u>Format</u>
1-4	Aircraft Title	A4
5	Process Ground Data. Enter: 0 or blank - do not wish to examine ground data 1 - examine all data on tape	I1
6-29	Threshold Values. The eight thresholds are in the following order: NZ, NY, P, Q, R, ALT, A/S, FUEL	8I3

30-44 Constant Mean Values. The five mean(normal) 513 values are in the following order:
NZ, NY, P, Q, R

45-80 Blank

2.2.2 Input to COMPA2

This is used for the Method A on the C130 & C141 tape format:

<u>Column</u>	<u>Description</u>	<u>Format</u>
1-4	Aircraft Title	A4
5	Process Ground Data. Enter: 0 or blank - do not wish to examine ground data 1 - examine all data on tape	I1
6-23	Threshold values. The six thresholds are in the following order: NZ, NY/S1*, Q, R, ALT, and AS.	6I3
24-35	Constant Mean Values. Digit numbers are as follows: NZ, NY/S1, Q, R	4I3
36-80	Blank	
*NOTE:	S1 is recorded in place of NY on the C-130. A value of 999 must be used for the S1 threshold to exclude it from the compression process.	

2.2.3 Input to COMPB1

This is used for Method B on the A10 tape format.

<u>Column</u>	<u>Description</u>	<u>Format</u>
1-4	Aircraft Title	A4
5	Process Ground Data. Enter: 0 or blank - do not wish to examine ground data 1 - wish to examine all data recorded	I1
6-50	Window Value. The fifteen values are in the following order: NY, NZ, P, Q, R, $\dot{P}$, $\dot{Q}$, $\dot{R}$, FCTR, PLA, δ_{SB} , σ , h_p , and V_i	14I3
51-80	Blank	

2.2.4 Input to COMPB2

This is used for Method B on the C130 and C141 tape
format:

<u>Column</u>	<u>Description</u>	<u>Format</u>
1-4	Aircraft Title	A4
5	Process Ground Data. Enter: 0 or blank - do not wish to examine ground data 1 - examine all data recorded on tape	I1
6-62	Window Values. The nineteen values are in the following order: NY/S1*, NZ, Q, R, Q̇, Ṙ, S2, S3, S4, S5, S6, DELR, DELE, DELF, DELNG, VG, PS, PT, DELP *C130 - NY is replaced by the S1 parameter	19I3
63-80	Blank	

SECTION 3

DESCRIPTION OF PROGRAMS

Four separate programs were written for the evaluation of the two methods of data compression. A separate program was written for use on the A-10 data and the C-130/C-141 data for each of the compression techniques. This was done to keep the programming simpler in order to make a more accurate comparison of run times. Each of these four programs will be described in the following paragraphs followed by a description of the general subroutines PACK and UNPACK which are used in all of the programs.

Each of the four programs must do a certain amount of editing of the raw data. After program initialization and the program control cards are read, the data is processed by reading five frames or ten seconds of data from the RT tape. The very first frame from each tape, the header frame, is ignored. Thereafter, each frame is handled as an entity. The frame is first examined for parity errors as indicated by a non-zero value in bytes 483 and 484 combined. If a parity error is detected, the frame is counted and then ignored for further processing. The next check is made for a short record which is indicated by the four least significant bits of byte 480 containing something other than a 9. If a short record is detected, it is counted and ignored. A check is then made for the presence of a BIT, CAL, or PRE indicator in either half of the frame. This is indicated by the presence of a one in the appropriate bits of DD11 as indicated in Table 2.4. If the BIT, CAL, or PRE bit is on, the second of data is counted and then ignored with the exception of the BIT bit being on. This condition is necessary for the beginning of a flight if the aircraft is not airborne. If the aircraft is airborne, the data is ignored. If the BIT bit is on exclusively, a single sample of each of the parameters is saved for inclusion in the header record as calibration data. Each of the three calibration levels is handled in this manner. The first occurrence of a BIT cycle also signals the end of a previous flight and an end of flight record is written. A BIT cycle can be initiated by a power interruption while the aircraft is airborne which will be ignored by the program.

If the data is valid data, having passed all of the previous tests, a check is made to determine whether the aircraft is airborne. If the aircraft is airborne, the specific compression technique is applied. If the aircraft is not in the air and the decision has been made to not process ground data, the second of data is ignored; otherwise the data is processed.

The initial values of airspeed, altitude, fuel counter and events are obtained from the first valid data record following a BIT cycle for Method A and for all parameters for Method B.

If no exceedance or window crossings are found in a segment of tape between BIT cycles, there will not be a header record or any other evidence of the segment on the output tape. If, however, ground data is being processed and exceedances or window crossings are found, the segment will be designated as a flight even though there is no inflight data in the segment.

If the data is good data, having passed all of the previous tests, a check is made to determine whether the aircraft is airborne. If the aircraft is airborne, the specific compression technique is applied. If the aircraft is not in the air and the decision has been made to not process ground data, the second of data is ignored; otherwise the data is processed.

The initial values of airspeed, altitude, fuel counter and events are obtained from the first good data record following a BIT cycle for Method A and for all parameters for Method B.

3.1 METHOD A

Data compression using Method A is accomplished by testing selected parameters for exceedance of predetermined thresholds. Specifically, if $v_m - T_v > v_i$ or $v_i > v_m + T_v$

where v_m = normal value

T_v = threshold,

then the data is considered active and the entire second of data is written to the output tape. If the data is not determined active by this criterion, a further check is made for a change in

flight condition. A flight condition change is defined as a change by more than a preselected value of altitude, airspeed, etc. from the last change in that parameter. A change in any event is also considered a flight condition change. If a change in flight condition occurs, the entire second of data is written to the output tape. One second of additional data is written to the output tape following a period of active data as determined by either of the prior criteria.

For each second of data written to the output tape, computations of the roll, pitch and yaw acceleration are made, where applicable, using the following equation

$$\dot{P} = \frac{S}{12} [(P_{i-2} - P_{i+2}) + 8(P_{i+1} - P_{i-1})]$$

where $\dot{P}$ = roll, pitch or yaw acceleration (counts/sec²)

S = sample rate (samples/sec)

P = roll, pitch or yaw rate (counts)

The accelerations are not computed for the first two or the last two samples in each continuous interval of active data. The acceleration values are biased by 128 to avoid a negative sign which would require an additional bit for storage.

A small peculiarity of the above equation should be noted. Each time there is a change in the acceleration from the zero position, the first value computed is of the wrong sign. For example, if we have four consecutive values of 128 followed by a 129, the acceleration computed from the equation is -1.25 counts/sec for a sampling rate of 15 samples/sec. This is due to the fact that the quartic equation must pass through all five points of the curve. This should, however, present no real problem since the magnitude of these values should be quite small and well below the threshold of interest.

All recorded parameters need not be examined for activity with this compression technique since all data is retained if any of the analyzed parameters become active.

3.1.1 COMPAL Program

The COMPAL program was written to compress data recorded on the A-10 aircraft specifically. The parameters examined for threshold exceedances are N_y , N_z , pitch, roll and yaw, all of which are sampled at a rate of 15 samples/sec. Flight condition parameters are altitude, airspeed, fuel counter and the two event samples all of which are sampled at the rate of one sample/sec with the exception of the fuel counter which is sampled at 15 samples/sec. Inflight data on the A-10 aircraft is determined by the presence of a one in event 3.

3.1.2 COMPA2 Program

The COMPA2 program will work on both C-130 and C-141 data. However, when running C-130 data the threshold value for S_1 , which is recorded in place of N_y , must be set to a large number to avoid processing the strain channel. A strain channel, unless it is at some location which has a constant mean stress, cannot be processed by looking for excursions from a constant mean.

The parameters that are examined for threshold exceedances are N_y , N_z , pitch and yaw. N_y and N_z are sampled at 30 times/sec while pitch and yaw are sampled at 20 times/sec. Flight condition parameters are altitude, airspeed and the two event samples which are sampled at 1 sample/sec. If the aircraft is not airborne and ground data are being processed, ground speed becomes the only flight condition parameter. Ground speed is sampled at two samples/sec. Inflight data on both the C-130 and C-141 aircraft is determined by the presence of a one in the first event.

3.2 METHOD B

The technique of data compression designated as Method B involves the location of window crossings and peaks on selected parameters. A window crossing is determined by the following equation

$$v_i - w > v \quad \text{or} \quad v > v_i + w$$

where v = present value
 v_i = last stored value
 w = window value

Upon detecting a window crossing the value along with a parameter ID and time of occurrence are written to the output tape. If a minimum or a maximum is reached by the parameter between window crossings, it is also written to the output tape. A maximum is generated by a change from positive slope of two or more window crossings to a negative slope and a minimum is generated by a change from negative to positive slope.

All parameters of interest are handled in the same manner with this compression technique, however, window values cannot be varied for the events which are treated as combined values of events 1-4 and events 1-9. The event windows are set, within the program, to zero.

A continuous computation of the roll, pitch, and yaw accelerations, where applicable, is made using all samples of roll, pitch and yaw. These accelerations are then analyzed for window crossings and peaks in the same way as the other parameters. An input option allows for processing or ignoring ground data at the discretion of the user.

A major difference between the two methods of data compression is that in using Method B all parameters of interest must be included in the analysis for window crossings since coincident values of other recorded parameters are not saved when activity occurs on one parameter. Therefore, if a parameter is not included in the analysis, that data is lost during the compression process.

3.2.1 Program COMPB1

The COMPB1 program was written to process the A-10 data using the Method B technique. The parameters that are examined for window crossings are: N_y , N_z , pitch, roll, yaw, fuel counter, speed brake position, power lever angle, strain, altitude,

TABLE 3.1
SAMPLING RATES OF PARAMETERS FOR C-130/C-141

<u>Parameter</u>	<u>Sampling Rate (Samples/sec)</u>	
	<u>C-130</u>	<u>C-141</u>
Pitch	20	20
Yaw	20	20
N _z	30	30
N _y		30
S1	30	
S2	20	20
S3	20	20
S4	20	20
S5	20	20
S6	10	10
DELR	10	10
DELE	10	10
DELF	5	5
DELNG	5	5
PS	1	1
PT	1	1
VG	2	2
DELP	1	1
E1-9	1	1

airspeed, and events 1-9. The pitch, roll, and yaw accelerations are computed and analyzed. The altitude, airspeed and events are sampled at 1 sample/sec and all of the other parameters at 15 samples/sec.

3.2.2 Program COMPB2

The COMPB2 program was written to process C-130/C-141 data. The C-141 parameters analyzed for window crossings and peaks are N_y , N_z , pitch, yaw, strains 2-6, rudder deflection, elevator deflection, flap deflection, nose gear angle, ground speed, cabin pressure, altitude, airspeed, and events 1-9. The same parameters, with the exception of strain 1 replacing N_y , are analyzed for the C-130 data. The sampling rates for the various parameters are shown in Table 3.1. Airborne data is detected by the presence of a one in event 1 for both aircraft. In addition to the recorded parameters, the pitch and yaw accelerations are computed continuously and are monitored for window crossings and peaks.

3.3 SUBROUTINE PACK

Subroutine PACK is a generalized routine for packing data into a condensed format for output to tape. The routine is capable of extracting any number of bits from any number of unpacked words and consecutively packing these into an array of 60 bit packed words. The parameters in the call statement to subroutine PACK and their usage are as follows:

- UP - array containing the unpacked data
- NUP - number of words from UP to be packed
- P - array to receive the packed data
- J - first available word in the P array to receive packed data
- NB - first available bit in the Jth word of P array to receive packed data ($1 \leq NB \leq 60$)
- ITBL - array of length NUP containing the number of bits from each word in UP to be packed into P (if positive bits are to be taken from right side of packed word and if negative from the left side of packed word).

The values of J and NB are saved internally within PACK so that in a subsequent call to the routine it will continue packing into the P array where it left off, if the values of J and NB have not been altered in the calling program.

3.4 SUBROUTINE UNPACK

Subroutine UNPACK is a generalized routine for unpacking data into 60 bit words. In the present application it is used to unpack the 486 consecutive 8 bit samples from a frame of RT data. The parameters in the call to PACK and their usage are as follows:

- UP - array to contain the unpacked data.
- NUP - number of words in UP the unpacked data will occupy
- P - array containing the packed data
- J - word in the P array at which unpacking is to begin
- NB - bit in the Jth word of the P array at which unpacking is to begin
- ITBL - array of length NUP containing the number of bits to unpack into each word of the UP array (if positive, bits are to be right justified with leading bits zero filled and if negative, left justified with trailing bits zero filled unless ITBL is a multiple of 6 in which case trailing bits are blank filled)

The variation in the possible values contained in the ITBL array allows for unpacking integer, real or alphanumeric information.

The values of J and NB are saved internally within UNPACK so that in a subsequent call to the routine it will continue to unpack from the P array at the point it left off, if the values of J and NB have not been altered by the calling program.

SECTION 4

PROGRAM OUTPUT

The output from the data compression programs consists of printed output and tape. The printed output lists the input parameters, a summary of each flight encountered on the RT tape and a summary of the total tape. Each time a BIT cycle is encountered, all counters are reinitialized and the summary data for a flight is printed if appropriate. A flight on the output tape is either an actual flight or a segment of ground data depending upon whether ground data is being processed. If ground data is being processed, data will be written to the output tape only if at least one exceedance or window crossing was detected. If ground data is not being processed, only the data occurring inflight will be written to the output tape and if no lift-off occurs during the segment, no trace of the segment will occur on the output tape or the printed output.

The printed output from programs COMPAL and COMPAL2 using Method A of data compression are identical with the exception of the parameters analyzed and their thresholds. A detailed explanation, using output from COMPAL, is shown in Figure 4.1. The printed output from programs COMPB1 and COMPB2 are also identical with the exception of the parameters analyzed and their window values. The output from COMPB1 is used in Figure 4.2 to provide a detailed explanation.

The tape output of the four compression programs is different for each program. All of the programs write flight header records (Type 1) followed by data records (Type 2) and end of flight records (Type 3). The difference in the formats of these records is due to the compression technique and the number of parameters being analyzed. The detailed format of the three record types for each of the compression programs is shown in Tables 4.1, 4.2, 4.3, and 4.4. Although it might be desirable from the standpoint of generalization to standardize the output formats for a given compression technique, considerable savings in output space is

realized in the case of the A-10 using Method A with the present format. This is due to the large number of vacant samples. In fact, a 37 percent compression of the data can be realized by only removing the unused samples from the data.

```

AIRCRAFT# A10
PROCESSING GROUND DATA
THRESHOLDS# NZ= NY= P= Q= R=AL= AS= EC
          11 13 5 6 6 9 4 4
VALUES OF# NZ= NY= P= Q= R
          85 128 128 128 128

FLIGHT# 1 STATISTICS
1. THE NUMBER OF TYPE2 RECORDS PRINTED# 2501
2. SECONDS OF GROUND DATA# 482
3. SECONDS OF ACTIVE DATA# 1526
4. TOTAL NUMBER OF SECONDS# 4456

FLIGHT# 2 STATISTICS
1. THE NUMBER OF TYPE2 RECORDS PRINTED# 6213
2. SECONDS OF GROUND DATA# 3085
3. SECONDS OF ACTIVE DATA# 4448
4. TOTAL NUMBER OF SECONDS# 10376

FLIGHT# 3 STATISTICS
1. THE NUMBER OF TYPE2 RECORDS PRINTED# 3940
2. SECONDS OF GROUND DATA# 2111
3. SECONDS OF ACTIVE DATA# 2335
4. TOTAL NUMBER OF SECONDS# 6318

FLIGHT# 4 STATISTICS
1. THE NUMBER OF TYPE2 RECORDS PRINTED# 5635
2. SECONDS OF GROUND DATA# 1010
3. SECONDS OF ACTIVE DATA# 4164
4. TOTAL NUMBER OF SECONDS# 8170

FLIGHT# 5 STATISTICS
1. THE NUMBER OF TYPE2 RECORDS PRINTED# 921
2. SECONDS OF GROUND DATA# 1910
3. SECONDS OF ACTIVE DATA# 546
4. TOTAL NUMBER OF SECONDS# 1974

FLIGHT# 6 STATISTICS
1. THE NUMBER OF TYPE2 RECORDS PRINTED# 4111
2. SECONDS OF GROUND DATA# 1504
3. SECONDS OF ACTIVE DATA# 2911
4. TOTAL NUMBER OF SECONDS# 8152

FLIGHT# 7 STATISTICS
1. THE NUMBER OF TYPE2 RECORDS PRINTED# 5043
2. SECONDS OF GROUND DATA# 1922
3. SECONDS OF ACTIVE DATA# 3690
4. TOTAL NUMBER OF SECONDS# 9090

FLIGHT# 8 STATISTICS
1. THE NUMBER OF TYPE2 RECORDS PRINTED# 277
2. SECONDS OF GROUND DATA# 680
3. SECONDS OF ACTIVE DATA# 158
4. TOTAL NUMBER OF SECONDS# 682

```

Figure 4.1. Method A Output.

PRINTOUT FOR COMPA1 STATISTICS.

1. THE NUMBER OF TYPE1 RECORDS PRINTED:	8
2. THE NUMBER OF TYPE2 RECORDS PRINTED:	28641
3. THE NUMBER OF TYPE3 RECORDS PRINTED:	8
4. SECONDS OF GROUND DATA:	16018
5. TOTAL ACTIVE SECONDS:	19777
6. TOTAL NUMBER OF SECONDS:	52810

EOJ

Figure 4.1 (Continued)

Notes:

1. The Number of Type 2 Records Printed

The number of seconds of data found where a parameter exceeded the input threshold plus the additional second written at the end of each active period.

2. Seconds of Ground Data

The number of recorded seconds while the aircraft was not airborne.

3. Seconds of Active Data

The number of seconds of data found where a parameter exceeded threshold.

4. Total Number of Seconds

The total time for the flight from recorder start to stop.

5. The Number of Type 1 Records Printed

The total number of flight header records written to the output tape. This is the number of flights encountered on the tape.

6. The Number of Type 3 Records Printed

The total number of end of flight records written to the output tape.

Figure 4.1. (Concluded)

AIRCRAFT: A10									
PROCESSING GROUND DATA									
WINDOW VALUES:	NY	NZ	P	Q	R	P-DOT	Q-DOT	R-DOT	
	13	11	5	6	6	20	20	20	
FCT:	FLA	CSB	SIGMA	ALT	AS				
	4	1	1	5	9	4			

FLIGHT# 1 STATISTICS	
1. THE NUMBER OF TYPE2 RECORDS WRITTEN:	106
2. SECONDS OF GROUND DATA:	402
3. SAMPLES OF ACTIVE DATA:	2419
4. TOTAL NUMBER OF SECONDS:	456

FLIGHT# 2 STATISTICS	
1. THE NUMBER OF TYPE2 RECORDS WRITTEN:	906
2. SECONDS OF GROUND DATA:	3085
3. SAMPLES OF ACTIVE DATA:	20938
4. TOTAL NUMBER OF SECONDS:	10376

FLIGHT# 3 STATISTICS	
1. THE NUMBER OF TYPE2 RECORDS WRITTEN:	130
2. SECONDS OF GROUND DATA:	2111
3. SAMPLES OF ACTIVE DATA:	2982
4. TOTAL NUMBER OF SECONDS:	6318

FLIGHT# 4 STATISTICS	
1. THE NUMBER OF TYPE2 RECORDS WRITTEN:	1581
2. SECONDS OF GROUND DATA:	1017
3. SAMPLES OF ACTIVE DATA:	36345
4. TOTAL NUMBER OF SECONDS:	8170

FLIGHT# 5 STATISTICS	
1. THE NUMBER OF TYPE2 RECORDS WRITTEN:	55
2. SECONDS OF GROUND DATA:	2554
3. SAMPLES OF ACTIVE DATA:	1260
4. TOTAL NUMBER OF SECONDS:	2646

FLIGHT# 6 STATISTICS	
1. THE NUMBER OF TYPE2 RECORDS WRITTEN:	366
2. SECONDS OF GROUND DATA:	1504
3. SAMPLES OF ACTIVE DATA:	8409
4. TOTAL NUMBER OF SECONDS:	8152

FLIGHT# 7 STATISTICS	
1. THE NUMBER OF TYPE2 RECORDS WRITTEN:	811
2. SECONDS OF GROUND DATA:	1922
3. SAMPLES OF ACTIVE DATA:	18645
4. TOTAL NUMBER OF SECONDS:	9630

FLIGHT# 8 STATISTICS	
1. THE NUMBER OF TYPE2 RECORDS WRITTEN:	14
2. SECONDS OF GROUND DATA:	1524
3. SAMPLES OF ACTIVE DATA:	307
4. TOTAL NUMBER OF SECONDS:	1532

Figure 4.2. Method B Output.

PRINTOUT FOR COMB1 STATISTICS:

1. THE NUMBER OF TYPE1 RECORDS WRITTEN:	8
2. THE NUMBER OF TYPE2 RECORDS WRITTEN:	3969
3. THE NUMBER OF TYPE3 RECORDS WRITTEN:	0
4. SECONDS OF GROUND DATA:	16018
5. TOTAL ACTIVE SAMPLES:	91295
6. THE NUMBER OF PEAKS STOPPED:	31091
7. THE TOTAL NUMBER OF SECONDS:	52810

Figure 4.2 (Continued)

Notes:

1. The Number of Type 2 Records Written

This is the number of active data records written. Each record, with the exception of the last for each flight, contains 23 window crossings and peaks.

2. Seconds of Ground Data

The number of recorded seconds where the aircraft was not airborne.

3. Samples of Active Data

The number of window crossings and peaks found in the flight.

4. Total Number of Seconds

The total time for the flight from recorder start to stop.

5. The Number of Type 1 Records Written

The total number of flight header records written to the output tape. This is the number of flights encountered on the tape.

6. The Number of Type 3 Records Written

The total number of end of flight records written to the output tape.

7. The Number of Peaks Stored

This is the total number of actual peaks encountered on the tape.

Figure 4.2 (Concluded)

TABLE 4.1
COMPAL TAPE OUTPUT FORMAT

RECORD TYPE 1

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
Record type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
ALT threshold	8	4
AS threshold	8	
Fuel threshold	8	
NY threshold	8	
NZ threshold	8	
P threshold	8	
Q threshold	8	
Blank	4	
R threshold	8	5
NY Cal 1	8	
NZ Cal 1	8	
P Cal 1	8	
Q Cal 1	8	
R Cal 1	8	

TABLE 4.1 (Continued)
COMPAL TAPE OUTPUT FORMAT

RECORD TYPE 1 (Continued)

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
Delta-SB Cal 1	8	
Blank	4	
Sigma Cal 1	8	6
PLA Cal 1	8	
Airspeed Cal 1	8	
Altitude Cal 1	8	
Fuel Counter Cal 1	8	
NY Cal 2	8	
NZ Cal 2	8	
Blank	4	
P Cal 2	8	7
Q Cal 2	8	
R Cal 2	8	
Delta SB Cal 2	8	
Sigma Cal 2	8	
PLA Cal 2	8	
Airspeed Cal 2	8	
Blank	4	
Altitude Cal 2	8	8
Fuel Counter Cal 2	8	
NY Cal 3	8	
NZ Cal 3	8	
P Cal 3	8	
Q Cal 3	8	
R Cal 3	8	
Blank	4	
Delta SB Cal 3	8	9
Sigma Cal 3	8	
PLA Cal 3	8	
Airspeed Cal 3	8	

TABLE 4.1 (Continued)
COMPAL TAPE OUTPUT FORMAT

RECORD TYPE 1 (Continued)

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
Altitude Cal 3	8	
Fuel Counter Cal 3	8	
Blank	12	
Blank	960	10-25

TABLE 4.1 (Continued)
COMPAL TAPE OUTPUT FORMAT

RECORD TYPE 2

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
Record type	4	1
Time*	16	
Airspeed	8	
Altitude	8	
Fuel Counter	8	
E1-4	8	
E5-9	8	
NY	8	} Repeat 15 times
NZ	8	
P	8	
Q	8	
R	8	
·P	8	
·Q	8	
·R	8	
Delta SB	8	
Sigma	8	
PLA	8	
Blank	8	
8 x 180 = 1440		
TOTAL		25 words

*Time is from start of recorder.

TABLE 4.1 (Concluded)
COMPAL TAPE OUTPUT FORMAT

RECORD TYPE 3

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
Record type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DDL	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD0	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
Flight duration (in seconds)	60	4
Blank	1260	5-25

TABLE 4.2
COMPA2 TAPE OUTPUT FORMAT

RECORD TYPE 1

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
Record Type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
ALT threshold	8	4
AS threshold	8	
NY threshold	8	
NZ threshold	8	
Q threshold	8	
R Threshold	8	
Blank	12	
Q Cal 1	8	5
R Cal 1	8	
NZ Cal 1	8	
NY - (S1) Cal 1	8	
S2 Cal 1	8	
S3 Cal 1	8	
S6 Cal 1	8	
Blank	4	

TABLE 4.2 (Continued)
COMPA2 TAPE OUTPUT FORMAT

RECORD TYPE 1 (Continued)

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
DELR Cal 1	8	6
S4 Cal 1	8	
S5 Cal 1	8	
DELE Cal 1	8	
VG Cal 1	8	
DELF Cal 1	8	
DELNG Cal 1	8	
Blank	4	
PS Cal 1	8	7
PT Cal 1	8	
DELP Cal 1	8	
Q Cal 2	8	
R Cal 2	8	
NZ Cal 2	8	
NY Cal 2	8	
Blank	4	
S2 Cal 2	8	8
S3 Cal 2	8	
S6 Cal 2	8	
DELR Cal 2	8	
S4 Cal 2	8	
S5 Cal 2	8	
DELE Cal 2	8	
Blank	4	
VG Cal 2	8	9
DELF Cal 2	8	
DELNG Cal 2	8	
PS Cal 2	8	
PT Cal 2	8	
DELP Cal 2	8	
Q Cal 3	8	
Blank	4	

TABLE 4.2 (Continued)
COMPA2 TAPE OUTPUT FORMAT

RECORD TYPE 1 (Continued)

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
R Cal 3	8	10
NZ Cal 3	8	
NY Cal 3	8	
S2 Cal 3	8	
S3 Cal 3	8	
S6 Cal 3	8	
DELR Cal 3	8	
Blank	4	
S4 Cal 3	8	11
S5 Cal 3	8	
DELE Cal 3	8	
VG Cal 3	8	
DELF Cal 3	8	
DELNG Cal 3	8	
PS Cal 3	8	
Blank	4	
PT Cal 3	8	12
DELP Cal 3	8	
Blank	44	
Blank	1560	13-38

TABLE 4.2 (Continued)
COMPA2 TAPE OUTPUT FORMAT

RECORD TYPE 2

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
Record type	4	1
Time *	16	
One second of data as is	$8 \times 239 = 1912$	
$\dot{Q}$	$8 \times 20 = 160$	
$\dot{R}$	$8 \times 20 = 160$	
Blank	28	
TOTAL		38 words

*Time is from start of recorder

TABLE 4.2 (Concluded)
COMPA2 TAPE OUTPUT FORMAT

RECORD TYPE 3

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
Record Type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
Flight Duration	60	4
(time in seconds)		
Blank	2040	5-38

TABLE 4.3

COMPBI TAPE OUTPUT FORMAT
RECORD TYPE 1

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
Record Type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
NY Window	8	4
NZ Window	8	
P Window	8	
Q Window	8	
R Window	8	
P-DOT Window	8	
Q-DOT Window	8	
Blank	4	
R-DOT Window	8	5
Delta SB Window	8	
Sigma Window	8	
Delta PLA Window	8	
Airspeed Window	8	
Altitude Window	8	
Fuel Counter Window	8	
Blank	4	
NY Cal 1	8	6
NZ Cal 1	8	

TABLE 4.3 (Continued)
COMPB1 TAPE OUTPUT FORMAT
RECORD TYPE 1

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
P Cal 1	8	
Q Cal 1	8	
R Cal 1	8	
Delta SB Cal 1	8	
Sigma Cal 1	8	
Blank	4	
Delta PLA Cal 1	8	7
Airspeed Cal 1	8	
Altitude Cal 1	8	
Fuel Counter Cal 1	8	
NY Cal2	8	
NZ Cal2	8	
P Cal2	8	
Blank	4	
Q Cal2	8	8
R Cal2	8	
Delta SB Cal2	8	
Sigma Cal2	8	
Delta PLA Cal2	8	
Airspeed Cal2	8	
Altitude Cal2	8	
Blank	4	
Fuel Counter Cal2	8	9
NY Cal3	8	
NZ Cal3	8	
P Cal3	8	
Q Cal3	8	
R Cal3	8	
Delta SB Cal3	8	
Blank	4	

TABLE 4.3 (Continued)

COMPBL TAPE OUTPUT FORMAT
RECORD TYPE 1

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
Sigma Cal3	8	10
Delta PLA Cal3	8	
Airspeed Cal3	8	
Altitude Cal3	8	
Fuel Counter Cal3	8	
Blank	20	
Blank	360	11-16

TABLE 4.3 (Continued)
COMPBI TAPE OUTPUT FORMAT
RECORD TYPE 2

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
Record Type	4	1
Blanks	36	
ID	8	
Time	12	
Time	4	2
Sample number	8	
Value	8	
ID	8	
Time	16	
Sample number	8	
Value	8	
ID	8	3
Time	16	
Sample number	8	
Value	8	
ID	8	
Time	12	
Time	4	4
Sample number	8	
Value	8	
ID	8	
Time	16	
Sample number	8	
Value	8	

Repeat 7
times
(15-16)

TABLE 4.3 (Concluded)
COMPBI TAPE OUTPUT FORMAT
RECORD TYPE 3

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
Record Type	4	1
Aircraft Title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
Flight Duration (in seconds)	60	4
Blank	720	5-16

TABLE 4.4

COMPB2 TAPE OUTPUT FORMAT
RECORD TYPE 1

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
Record Type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
Q window	8	4
R window	8	
Q-DOT Window	8	
R-DOT window	8	
NZ window	8	
NY window	8	
S2 window	8	
Blank	4	
S3 window	8	5
S6 window	8	
DWLR window	8	
S4 window	8	
S5 window	8	
DELE window	8	
VG window	8	
Blank	4	
DELF window	8	6

TABLE 4.4 (Continued)

COMPB2 TAPE OUTPUT FORMAT
RECORD TYPE 1

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
DELNG window	8	
PS window	8	
PT window	8	
DELP window	8	
Blank	20	
Q Cal 1	8	7
R Cal 1	8	
NZ Cal 1	8	
NY -(S1) Cal 1	8	
S2 Cal 1	8	
S3 Cal 1	8	
S6 Cal 1	8	
Blank	4	
DELR Cal 1	8	8
S4 Cal 1	8	
S5 Cal 1	8	
DELE Cal 1	8	
VG Cal 1	8	
DELF Cal 1	8	
DELNG Cal 1	8	
Blank	4	
PS Cal 1	8	9
PT Cal 1	8	
DELP Cal 1	8	
Q Cal 2	8	
R Cal 2	8	
NZ Cal 2	8	
NY (S1) Cal 2	8	
Blank	4	
S2 Cal 2	8	10

TABLE 4.4 (Continued)
COMPB2 TAPE OUTPUT FORMAT
RECORD TYPE 1

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>
S3 Cal 2	8	
S6 Cal 2	8	
DELR Cal 2	8	
S4 Cal 2	8	
S5 Cal 2	8	
DELE Cal 2	8	
Blank	4	
VG Cal 2	8	11
DELF Cal 2	8	
DELNG Cal 2	8	
PS Cal 2	8	
PT Cal 2	8	
DELP Cal 2	8	
Q Cal 3	8	
Blank	4	
R Cal 3	8	12
NZ Cal 3	8	
NY (S1) Cal 3	8	
S2 Cal 3	8	
S3 Cal 3	8	
S6 Cal 3	8	
DELR Cal 3	8	
Blank	4	
S4 Cal 3	8	13
S5 Cal 3	8	
DELE Cal 3	8	
VG Cal 3	8	
DELF Cal 3	8	
DELNG Cal 3	8	
PS Cal 3	8	
Blank	4	
PT Cal 3	8	14
DELP Cal 3	8	
Blank	44	
Blank	120	15-16

TABLE 4.4 (Continued)
 COMPB2 TAPE OUTPUT FORMAT
 RECORD TYPE 2

<u>Data Item</u>	<u>Bits</u>	<u>Word</u>	
Record Type	4	1	
Blanks	36		
ID	8		
Time	12		
Time	4	2	
Sample number	8		
Value	8		
ID	8		
Time	16		
Sample number	8		
Value	8		
ID	8	3	
Time	16		
Sample number	8		
Value	8		
ID	8		
Time	12		
Time	4	4	
Sample number	8		
Value	8		
ID	8		
Time	16		
Sample number	8		
Value	8		

Repeat 7
 times
 (5-16)

TABLE 4.4 (Concluded)
 COMPB2 TAPE OUTPUT FORMAT
 RECORD TYPE 3

<u>Data Item</u>	<u>Bits</u>	(tail record) <u>Word</u>
Record type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
Flight Duration (in seconds)	60	4
Blank	720	5-16

SECTION 5

RESULTS

All four of the data compression programs were run extensively on RT data from the A-10, C-130, and C-141 aircraft. In order to establish the sensitivity of the Method A compression to the threshold size and the effect of the individual parameters on compression, runs were made with given values of the threshold on an individual parameter with the threshold on all other parameters set to a high value. Thus, the number of seconds containing data above threshold for each parameter was determined for three different threshold levels. The results of these runs for the three aircraft are shown in Tables 5.1, 5.2, and 5.3. It should be noted that the upper number in each block is the number of active seconds triggered by the specific parameter at the indicated threshold while the lower number is the number of times the flight condition or event changed when the parameter did not change. In all cases the windows on the flight condition parameters (altitude, airspeed, and fuel counter) were maintained at four counts. All runs were made with the option to process ground data in effect. After noting that the number of flight condition changes were small compared to the total number of active seconds, Programs COMPAL and COMPA2 were modified to write Type 2 data records when these changes occurred. The programs were originally designed to write data records upon detecting a parameter outside threshold and a time slice record when a flight condition parameter changed. Elimination of the time slice records simplified program logic with only a small addition of data to the output tape.

The same type of sensitivity runs were made using Method B on the A-10 and C-141 data. Results of these runs are presented in Tables 5.4 and 5.5. The data contained in these tables are window crossings on individual parameters by flight and total peaks by parameter for all flights. Although it is unknown what practical use one might make of the information, it is interesting to note that the ratio of total peaks to total window crossings

TABLE 5.1
ACTIVE DATA BY PARAMETER USING METHOD A (A-10)

Flight No.	Flight Time Sec.	Threshold = 4					Threshold = 9					Threshold = 19				
		N _Z	N _Y	P	Q	R	N _Z	N _Y	P	Q	R	N _Z	N _Y	P	Q	R
1	7470	620 7	1263 4	179 7	692 6	1013 7	189 7	141 7	65 7	193 6	392 7	28 7	21 7	14 7	21 7	87 7
2	10380	3065 8	3525 4	1389 12	2509 6	2959 17	1651 10	921 14	467 16	1547 8	1279 21	909 11	132 19	75 20	736 12	119 21
3	6310	983 6	1595 6	131 6	569 5	1036 7	255 7	203 7	18 7	173 6	299 7	62 7	2 7	1 7	38 7	72 7
4	8590	3093 5	2544 5	1751 7	2977 3	2936 5	2032 5	1135 10	1030 10	1873 3	1493 6	1447 5	498 11	480 12	1294 7	392 11
5	8150	1667 10	2796 8	595 11	1617 9	2138 11	881 12	287 11	292 12	891 12	779 12	489 12	90 11	84 13	428 12	93 13
6	9090	2459 5	3142 2	1355 8	2273 3	2478 8	1525 6	1075 4	661 8	1408 6	1212 9	952 9	336 7	220 8	803 6	76 9
TOTAL	49990	11883 41	14865 29	5400 51	10637 32	12560 55	6533 47	3862 53	2533 60	6085 41	5454 62	3887 48	1079 62	874 67	3320 51	839 68

NOTE: Upper entry is seconds of active data.
Lower entry is number of flight condition changes.

TABLE 5.2
ACTIVE DATA BY PARAMETER USING METHOD A (C-151)

Flight No.	Flight Time Sec.	Threshold = 4			Threshold = 9			Threshold = 19		
		N _z	N _y	Q	R	N _z	N _y	Q	R	R
1	19720	5871 10	22 12	90 8	366 11	31 12	1 12	22 12	209 11	1 12
2	32740	847 3	387 4	374 1	724 5	3 6	368 6	39 4	354 6	1 6
3	2150	193 1	3 2	6 1	50 2	3 2	2 2	4 1	22 2	2 2
TOTAL	54610	6911 14	412 18	470 10	1140 18	37 20	371 20	65 17	585 19	4 20
										155 19

NOTE: Upper entry is seconds of active data.
Lower entry is number of flight condition changes.

TABLE 5.3
ACTIVE DATA BY PARAMETER USING METHOD A (C-130)

Flight No.	Flight Time Sec.	Threshold = 4			Threshold = 9			Threshold = 19		
		N _Z	Q	R	N _Z	Q	R	N _Z	Q	R
1	16180	4468 26	2690 23	5720 41	879 49	192 63	3189 64	19 62	1 66	273 66
2	2940	625 2	224 2	433 5	117 4	14 4	155 6	16 6	1 6	62 6
3	18930	6129 24	3801 16	6000 29	1773 53	851 51	3615 59	109 62	13 64	427 64
4	6360	1524 8	956 8	1994 8	306 15	157 11	1025 14	8 16	8 16	160 16
5	10950	1960 10	930 14	3545 6	423 15	169 16	1482 14	64 17	38 16	121 17
TOTAL	55360	14726 70	8610 63	17692 89	3498 136	1383 145	9466 157	216 163	61 168	1042 169

NOTE: Upper entry is seconds of active data.
Lower entry is number of flight condition changes.

TABLE 5.4
A-10 METHOD B SAMPLES OF ACTIVE DATA

		Window/Parameter														
Flight No.	Flight Time Sec.	4NZ	4NY	4P	4Q	4R	9NZ	9NY	9P	9Q	9R	19NZ	19NY	19P	19Q	19R
1	7470	*	1137	367	704	638	175	181	143	165	220	47	46	68	47	67
2	10380		6245	2888	4139	1764	1952	1295	960	1243	530	639	318	247	466	161
3	6310	1848	2528	422	514	647	269	386	59	84	161	45	36	25	43	72
4	8590	5521	7284	5799	6715	3382	2512	2961	2587	2662	1277	974	1161	1041	1091	424
5	8150	2166	2269	1332	1635	1956	722	554	542	498	357	311	185	196	201	137
6	9090	4672	5128	2866	3367	1779	1682	1858	1169	1150	580	733	382	394	481	157
Total Active Data	49990	*	24591	13674	17074	9266	7312	6735	5460	5802	3125	2749	2128	1971	2329	1018
Total Peaks		*	8632	3390	5090	2820	1938	2242	1584	1720	1040	846	*	666	884	388
Peaks/Total		*	.35	.25	.30	.30	.26	.33	.29	.30	.33	.31	*	.34	.38	.38

*Values not available due to tape break late in the program.

TABLE 5.5
C-141 METHOD B SAMPLES OF ACTIVE DATA

Flight	Time	Wind/Parameter											
		4NY	4NZ	4Q	4R	9NY	9NZ	9Q	9R	19NY	19NQ	19Q	19R
1	1970	28	22	21	104	21	21	21	62	21	21	21	40
2	19722	77	1567	137	195	41	92	45	103	41	44	41	69
3	32734	7951	650	1353	1816	5912	68	175	399	3670	31	31	53
Totals	54426	8056	2239	1511	2115	5964	181	241	564	3732	96	93	162
Total Peaks		1851	1107	705	891	1661	65	95	227	1445	19	19	51

does not vary from parameter to parameter for the A-10 data. This ratio also does not vary much as the window size is increased. It should be kept in mind that for Method A the data presented is seconds of active data and the data presented for Method B is samples of active data.

A direct comparison of the amount of output tape used for the two methods is not straight forward. It was first believed that by choosing the window sizes for Method B to be the same as the thresholds used in Method A, with the exception of the $\dot{p}$, $\dot{q}$, and $\dot{r}$ windows, a reasonable comparison might be made. Using thresholds and windows of 19 counts for $\dot{q}$ and $\dot{r}$ and 9 counts for all other parameters on C-141 data, the results shown in the first two columns of Table 5.6 were obtained. Since this seemed to be a large number of active samples using Method B, a run was made eliminating $\dot{q}$ and $\dot{r}$. The last column of Table 5.6 shows these results. If we assume that there will be a like number of window crossings and peaks on $\dot{q}$ and $\dot{r}$ to the number of peaks and window crossings on N_y , N_z , q and r we can linearly extrapolate for the total number of peaks and window crossings. This, of course, assumes equivalent window values on all parameters. Using the values from Table 5.5, with a 9 count window there are 1,800 windows and peaks per parameter for this data. This would give us 12,171 windows and peaks for all parameters on the tape. Since these values are blocked 23 per 16 word record on the tape, we would have a total of 8,464 words counting the header and end of flight records. The output from Method A is written as a 38-word record for each active second which gives a total of 46,322 words. Therefore, on this particular flight there is approximately 5.5 times more output for Method A than for Method B. It should be noted, however, that there appears to be an instrumentation problem with N_y on this tape. This can be seen by examination of Tables 5.2 and 5.5. The number of exceedances using Method A does not drop off significantly as the threshold is increased and the number of window crossings and peaks also does not decrease rapidly enough as the window value is increased. This leads one to believe that the actual ratio of output data for the two methods would be different for valid data. The CYBER 175

TABLE 5.6
COMPARISON OF METHODS FOR C-141 DATA

Flight No.	Time	Method A	Method B	
		Active	Active	Active
		Seconds ⁽¹⁾	Samples ⁽²⁾	Samples ⁽³⁾
1	48	1	23	21
2	1,970	118	434	430
3	19,722	274	9,299	781
4	32,734	601	39,761	7,297
5	92	1	113	21
6	142	1	22	21
Totals	54,708	996	49,652	8,571
CP Time ⁽⁴⁾ (Sec/Flt.Hour)		11.5	28.4	28.1

- Notes:
- (1) Threshold of 9 counts on all parameters.
 - (2) Window of 9 counts on all parameters except 19 counts/sec on $\dot{q}$ and $\dot{r}$.
 - (3) Window of 9 counts on all parameters except 999 counts/sec on $\dot{q}$ and $\dot{r}$.
 - (4) Central processor time on CYBER 175.

central processor time, however, is approximately 2.5 times greater for Method B than for Method A. This is due to the greater number of parameters examined for window crossings in Method B and in particular to the continuous computation of $\dot{p}$ and $\dot{r}$.

The same runs and extrapolation were made for the A-10 data. These runs were made on a shortened A-10 tape due to a stretching of the tape late in the program and the first two flights do not agree with previous runs. The results are shown in Table 5.7. The total number of windows and peaks were calculated to be 49,350. The total number of words output for Method B was calculated to be 34,704 and for Method A it was 386,700 which is about 11 times that of Method B. The amount of central processor time for the A-10 using Method B was only about 1.7 times that using Method A. This is due to the fact that there are not as many additional parameters analyzed for the A-10 using Method B over Method A.

The two runs made with realistic thresholds and windows for the A-10 tend to verify the calculation done above. The output of these runs was shown in Figures 4.1 and 4.2. Calculations made using this data show the compression ratio for Method A to be 2.36 and for Method B to be 26.9.

TABLE 5.7
COMPARISON OF METHOD FOR A-10 DATA

<u>Flight Number</u>	<u>Time</u>	<u>Method A Active Samples</u>	<u>Method B Active Samples</u>
1	14,832	4,428	7,771
2	6,318	1,496	1,001
3	8,170	3,581	12,523
9	8,152	2,312	3,128
10	9,090	3,330	7,130
TOTALS	46,562	15,147	31,553
CP Time (Sec/Flt. Hour)		13.9	23.5

SECTION 6

CONCLUSIONS AND RECOMMENDATIONS

The following conclusions and recommendations have been reached concerning the two data compression techniques:

1. Method A may be more suitable for fighter-type aircraft since more precise coincident values of parameters are obtained when any parameter peaks.

2. Method B can be used to obtain coincident values of other parameters if one can accept the degradation in accuracy due to using the previous window crossing on the coincident parameter.

3. Method B uses considerably less tape for output, thus a greater compression ratio is obtained. (The actual ratio of Method A to Method B is believed to be about 10 to 1.)

4. Method B uses significantly more computer time than Method A, but time used in subsequent analysis would be less. (This ratio will vary with the aircraft from about 1.5 to 2.5.)

5. The compression ratio for Method A, processing ground data, on the A-10 data using 11 counts on N_z , 13 counts on N_y , 5 counts on P, 6 counts on q and r, 9 counts on altitude, and 4 counts on airspeed and fuel counter was 2.36. This is the direct ratio of bits on the RT tape to bits on the output tape. The compression ratio using these same values as windows in Method B was 26.9.

6. The compression ratio using Method B would be considerably greater.

7. A study should be performed to examine the validity of the output data of the compression programs. This would require writing an edit program to select peaks and block the data.

8. Neither method of compression would affect peaks except for some data lost in the threshold needed to define a peak.

APPENDIX A
DATA COMPRESSION PROGRAM LISTINGS

APPENDIX A

DATA COMPRESSION PROGRAM LISTINGS

A complete source listing of each of the four data compression programs is provided in the following pages. Programs COMPA1, COMPA2, COMPB1, and COMPB2 are presented in Tables A.1 through A.4 while the general subroutines PACK and UNPACK are presented in Table A.5. In addition to the program listings, Table A.6 and A.7 contain definitions of variables used in the programs and Figures A.1 and A.2 show the general flow of Method A and Method B, respectively.

TABLE A.1

PROGRAM COMPAL

PROGRAM COMPAL 7474 OP1=1 TRACE FIN 4,7+476 10/19/79 15,41-39 PAGE 1

```

1  PROGRAM COMPAL(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE7)
C ***** NON-JAMMING-UBFI 547-CONTAGI#-93579 *****
C ***** THIS PROGRAM IS THE FIRST COMPRESSION METHODOLOGY USING INITIALLY
C ***** DETERMINED THRESHOLDS AS DETERMINANTS AS TO WHETHER A SECOND
C ***** IS ACTIVE OR INACTIVE DATA *****
C ***** DIMENSION NTBLK(324),ITBL(486) *****
COMMON INDEX,IREC(486),IOVP,IOVR,IOVQ,IOVNY,IOVNZ,IOVALT,IOVAS,
*IOVFC,IOVE1,IOVES
COMMON/CH1/ LAST
COMMON/CH3/ TAG
COMMON/CH4/ INCRST1,ITH
COMMON/CH7/ KOUNT,LKOUNT,NKOUNT,KNTK1,KNTK2,KNTK3
COMMON/CH2/ IKUNT,IGRCNT,IERRCNT,ISHTCNT
COMMON/CHTR/ KNT,LINKNT,NKNT,KNTK2
COMMON/CHTR2/ ITKNT,ITGRCNT,ITERCNT,ITSTCNT
COMMON/SHCH/ ISHTCNT,ISHTCNT2,ISHTCNT3,IEND
COMMON/TECH/ ITEM(6)
COMMON/THRS/ ITP,ITQ,ITR,IINY,IFNZ,IALT,IAS,IFG
DATA ITBL/486*8/

C ***** INITIALIZATION *****
C 0 IOV=INTEGER OLD-VALUE FOR...
C 0 CNT=COUNTERS FOR DIFFERENT AREAS...

IOVP=IOVR=IOVQ=IOVNY=IOVNZ=IOVALT=IOVAS=IOVFC=IOVE1=IOVE5=0
IERRCNT=ISHTCNT=IGRCNT=KOUNT=LKOUNT=NKOUNT=IKOUNT=0
ISHTCH2=ISHTCH4=KNTK1=KNTK2=KNTK3=0
ITERCNT=ITSTCNT=ITGRCNT=KNTK2=KNTK3=0
ITKNT=NTKNT=0
INFLGHT=ITM=IEND=0
ISHTCH3=1
ISHTCH1=INCRST1=2
READ(5,1)IAC,NGR,INZ,IINY,ITP,ITQ,ITR,IALT,IAS,
*IFG,IOVNZ,IOVNY,IOVP,IOVR,IOVR

1  FORMAT(A4,I1,13I3)
WRITE(6,7)IAC
7  FORMAT(10H1AIRCRAFT,2X,A4)
IF(NGR.EQ.0)WRITE(6,9)
IF(NGR.EQ.0)WRITE(6,9)
8  FORMAT(23H PROCESSING GROUND DATA)
9  FORMAT(27H NOT PROCESSING GROUND DATA)
WRITE(6,11)IINZ,IINY,ITP,ITQ,ITR,IALT,IAS,IFG
45  11  FORMAT(43H THRESHOLDS: NZ- NY- P- Q- R-ALT- AS- FC,
*12X,0(13,1X))
WRITE(6,14)IOVNZ,IOVNY,IOVP,IOVQ,IOVR
14  FORMAT(30H VALUES OF: NZ- NY- P- Q- R-ALT,
*5(13,1X))
LAST=0
10  BUFFER IN(1,1)NTBLK(1),NTBLK(324))
IF (UNIT(1))=20,900,15
15  NKOUNT=NKOUNT+10
GO TO 10
55  20  IM=JN=1
NKOUNT=NKOUNT+10

```

PROGRAM COMPAL

```

C *****MAIN LOOP DEGLADATIO*****
C 2 SECONDS/UNPACK...10 SECONDS/BUFFER IH.
C
C 60 DO 80 K=1,5
C CALL UNPACK(IREC,486,NTBLK,IN,JM,ITBL)
C IF(I SWITCH.EQ.0)GO TO 73
C
C 65 *****PAFITY ERROR CHECK*****
C
C IF(OR(SHIFT(IREC(483),0),IFEC(484)).NE.0) GO TO 71
C
C *****SHORT RECORD CHECK*****
C
C IF(AND(IREC(480),15).NE.9) GO TO 72
C INDEX=0
C DO 70 L=1,2
C IF(L.EQ.2)INDEX=240
C JBIT=IREC(114+INDEX)/128
C JCAL=AND((IREC(114+INDEX)/64),1)
C JPRE=AND((IREC(114+INDEX)/32),1)
C
C *****BIT CYCLE DETERMINATION*****
C
C IF(JBIT.EQ.1.AND.INFLIGHT.NE.32.AND.IEND.NE.1)CALL PR3
C IF(JBIT.EQ.1.OR.JCAL.EQ.1.OR.JPRE.EQ.1)IKOUNT=IKOUNT+1
C IF(JCAL.EQ.1.OR.JPRE.EQ.1)GO TO 70
C IF(JBIT.EQ.1)GO TO 35
C
C 67
C
C 65 C CALCULATION OF THE SECONDS INTO FLIGHT
C
C JTH=NKOUNT-10+2*(K-1)+L
C IEND=0
C
C *****IN FLIGHT CHECK*****
C
C INFLIGHT=AND(IREC(116+INDEX),32)
C IF(INFLIGHT.EQ.0)IGKDCNT=IGKDCNT+1
C IF(IGR.EQ.1)GO TO 25
C IF(INFLIGHT.EQ.0)GO TO 70
C
C 95 C *****
C
C 100 C DONE ONLY ONCE AFTER BIT CYCLE HAS BEEN SHUT DOWN
C INITIATION OF THE MEAN VALUES OF ALT,AS,FC,E(1-9)
C *****
C
C 25 IF(I SWITCH1.NE.0)GO TO 30
C I SWITCH1=2
C DO 26 M=6,22,16
C DO 26 N=1,3
C M1=M+N+INDEX
C IREC(M1)=128
C CONTINUE
C DO 27 M=30,198,16
C DO 27 N=1,3
C M1=M+N+INDEX
C IREG(M1)=1+25*((IREC(M1-27))-IREC(M1+37))+
C 8*(IFEC(M1+21)-7*EC(M1-11))+128
C
C 110
C
C 1

```

PROGRAM COMPAL

```

115 27 CONTINUE
    SWITCH2=0
    CALL PR12
    IOVALT=IREC(90+INDEX)
    IOVAS=IREC(1+INDEX)
    IOVEC=IREC(16+INDEX)
    IOVE1=IREC(16+INDEX)
    IOVE5=IREC(17+INDEX)
    CALL ACTDATA
    30 GO TO 70

125 C *****
    C *****
    C STORE THE CALIBRATED DATA INTO THE TYPE1 RECORDS.
    C *****
    C *****

130 C *****
    35 IF(INFLIGHT.NE.32)CALL PR11
    70 CONTINUE
    GO TO 80

    71 IERRCNT=IERRCNT+2
    GO TO 80

    72 ISHTCNT=ISHTCNT+2
    GO TO 80

    73 ISWCH4=2
    AC CONTINUE
    GO TO 10

140 C *****
    C *****
    C FINAL TAIL AND STATISTICAL EVALUATION PRINTOUT.
    C *****
    C *****

145 C *****
    900 IF(ISWCH4.EQ.0)GO TO 10
    SWITCH3=1
    CALL PR13
    WRITE(6,1013)
    1014 FORMAT(32H)PRINTOUT FOR COMPA1 STATISTICS.)
    1050 FORMAT(41H01. THE NUMBER OF TYPE1 RECORDS PRINTED:,I7)
    WRITE(6,1063)K1NR1
    1060 FORMAT(41H02. THE NUMBER OF TYPE2 RECORDS PRINTED:,I7)
    WRITE(6,1070)K1NR3
    1071 FORMAT(41H03. THE NUMBER OF TYPE3 RECORDS PRINTED:,I7)
    WRITE(6,1083)ITGCNT
    1080 FORMAT(20H04. SECONDS OF GROUND DATA:,I7)
    WRITE(6,1085)K1KNT
    1088 FORMAT(26H05. TOTAL ACTIVE SECONDS:,I7)
    WRITE(6,1095)N1KNT
    1095 FORMAT(29H06. TOTAL NUMBER OF SECONDS:,I7)
    WRITE(6,1103)
    1100 FORMAT(4H 00J)
    STOP
    END

165

```

TABLE A.1 (Continued)

PROGRAM COMPAL

SUBROUTINE ACTDATA 74/76 OPT=1 TRACE FIN 4,7+476 10/19/79 16.41.39 PAGE 1

```

1  SUBROUTINE ACTDATA
   DIMENSION LASTVAL(12)
   COMMON INDEX, IREC(486), IOVP, IOVR, IOVQ, IOVNY, IOVNZ, IOVALT, IOVAS,
   * IOVFC, IOVE1, IOVES
5  COMMON/CH1/ LAST
   COMMON/CH2/ INCKST, ITH
   COMMON/CH3/ KOUNT, LKOUNT, NKOUNT, KNTR1, KNTR2, KNTR3
   COMMON/CH4/ IREKNT, IREKNT, IREKNT, IREKNT, IREKNT, IREKNT
   COMMON/CH5/ KTKN1, KTKN2, KTKN3, KTKN4, KTKN5, KTKN6
10  COMMON/CH6/ IREKNT, IREKNT, IREKNT, IREKNT, IREKNT, IREKNT
   COMMON/CH7/ ISWICH1, ISWICH2, ISWICH3, IEND
   COMMON/CH8/ ITEMP(6)
   COMMON/THRES/ ITP, ITQ, ITR, IFNY, ITNZ, ITALT, ITAS, ITFC
15  C *****
   C ***** ACTIVE DATA DETERMINED BY IF OLD VALUE - THRESHOLD < VALUE -
   C OLD VALUE - THRESHOLD
   C VALUES ARE INVTZTP, GRALT, ASCTE5, FGR
   C OLD VALUES DEFINED PREVIOUSLY
20  C ***** THRESHOLDS ARE INPUT *****
   C *****
   N1=6
   DO 10 N=1,22,16
      M1=M+9*INDEX
      IF(IABS(IREC(M1)-IOVNY),GT,ITNY,OR,
1      IABS(IREC(M1+2)-IOVR),GT,ITR,OR,
2      IABS(IREC(M1+3)-IOVP),GT,ITP,OR,
3      IABS(IREC(M1+4)-IOVQ),GT,ITQ,OR,
4      IABS(IREC(M1+5)-IOVNZ),GT,ITNZ),GO TO 20
10  CONTINUE
      IF(IABS(IREC(98+INDEX)-IOVALT),GT,ITALT,OR,
1      IABS(IREC(1+INDEX)-IOVAS),GT,ITAS,OR,
2      IOVE1,ME,IREC(16+INDEX),OR,
3      IOVE5,ME,IREC(17+INDEX))GO TO 20
      DO 15 N1=6,23,16
15  CONTINUE
      C *****
      C ***** INACTIVE DATA RECOGNIZED.
      C ***** LAST WILL BE SET TO ONE IF PREVIOUS DATA WAS ACTIVE. THIS WILL
      C ***** ALLOW FOR EACH ACTIVE DATA TO BE FOLLOWED BY ONE SECOND *****
40  C *****
      LKOUNT=LKOUNT+1
      IF(LAST,EG,ITGO TO 50
      RETURN
20  IF(KOUNT,NE,0)GO TO 25
      ISWICH1=1
      CALL PRY1
      KOUNT=KOUNT+1
      ISWICH2=1
      DO 23 M=1,6
23  CONTINUE
      CALL PKT2
      IOVALT=IREC(98+INDEX)

```

TABLE A.1 (Continued)

PROGRAM COMPAL

```

SUBROUTINE AGTUA-A 74774 OPT=1, KAGE 10/12/77 10041-37 PAUL C
PROGRAM COMPAL

IOVA=IREC(1+INDEX)
IOVFO=IREC(N+INDEX)
IOVE1=IREC(16+INDEX)
IOVE5=IREC(17+INDEX)
INCP=0
KOUNT=KOUNT+1

C *****
C *****RESET THE OLD VALUES ACTIVE DATA FOUND.....
C SAVE LAST FOUR VALUES OF P-Q-R, (LASTVAL)
C THEN CALCULATE P-Q-R. FOR 3-13
C SET 1-2 P-Q-R TO 120
C *****
C *****
DO 30 M=107,235,16
DO 30 N=1,3
INCR=INCR+1
LASTVAL(INCR)=IREC(M+N+INDEX)
30 CONTINUE
IF (LASTVAL(160) TO 140)
DO 35 M=6,22,16
DO 35 N=1,3
M1=M+N
IREC(M1)=120
35 CONTINUE
DO 40 M=30,190,16
DO 40 N=1,3
M1=M+N+INDEX
IREC(M1)=((IREC(M1-27)-IREC(M1+37))+
0*(IREC(M1+21)-IREC(M1-11)))+120
40 CONTINUE
LAST=1
CALL PRT2
RETURN

C *****
C *****ACTIVE DATA NOT FOUND CALCULATE LAST 2 P-Q-R. FOR 14-15 OF PR
C REGRD(I)MP(1-6)), STORE INFO TYPE 2, GO ON AND CALCULATE P-Q-R
C THIS RECORD AND STORE INTO TYPE 2.....
C *****
C *****
50 LAST=0
DO 70 M=1,3
ITEMP(M)=(LASTVAL(M)-IREC(M+11+INDEX))+
8*(LASTVAL(M+9)-LASTVAL(M+3))+120
1 ITEMEMP(M+3)=(1-25)*((LASTVAL(M+3)-IREC(M+27+INDEX))+
8*(IREC(M+11+INDEX)-LASTVAL(M+6)))+120
70 CONTINUE
ISWICH2=1
CALL PRT2
DO 80 M=1,3
IREC(6+M+INDEX)=(1-25)*((LASTVAL(M+6)-IREC(M+43+INDEX))+
8*(IREC(M+27+INDEX)-LASTVAL(M+9)))+120
1
80 CONTINUE
DO 90 N=1,3
IREC(22+M+INDEX)=(1-25)*((LASTVAL(M+9)-IREC(M+59+INDEX))+
8*(IREC(M+43+INDEX)-IREC(M+11+INDEX)))+120
1

```

TABLE A.1 (Continued)

PROGRAM COMPAL

SUBROUTINE ACTDATA 74774 OPT=1 TRACE 10/19/79 16.41.39 PAGE 3

```

115      90 CONTINUE
      DO 100 M=36,196,16
      DO 100 N=1,3
      M1=M+N*INDEX
      IREC(M1)=(1.25)*((IREC(M1-27)-IREC(M1+37)) +
      8*(IREC(M1+21)-IREC(M1-11))) + 128
120      1
      100 CONTINUE
      CALL PR2
      ISWICH2=1
      DO 105 M=1,6
      ITEMP(M)=128
125      105 CONTINUE
      CALL PR2
      IOVALT=IREC(98*INDEX)
      IOVAS=IREC(1+INDEX)
130      IOVFG=IREC(11*INDEX)
      IOVEI=IREC(16*INDEX)
      IOVES=IREC(17*INDEX)
      RETURN
C *****
135      C *****
C ***** ACTIVE DATA FOUND AND FOLLOWS ANOTHER ACTIVE DATA **
C      CALCULATE LAST 2 VALUES OF P.-Q.-R. FOR 14-15
C      PREVIOUS RECORD (ITEMP(1-6)). STORE INTO TYPE 2,60 ON AND
C      CALCULATE P.-Q.-R. FOR 1-13. NOT TO BE STORED UNTIL 14-15
C      ARE CALCULATED.
C *****
140      C *****
C      110 DO 130 M=1,3
      ITEMP(M)=(1.25)*((LASTVAL(M)-IREC(M+11*INDEX)) +
      8*(LASTVAL(M+9)-LASTVAL(M+3))) + 128
145      1
      ITEMP(M+3)=(1.25)*((LASTVAL(M+3)-IREC(M+27*INDEX)) +
      8*(IREC(M+11*INDEX)-LASTVAL(M+6))) + 128
150      130 CONTINUE
      ISWICH2=1
      CALL PR2
      DO 150 M=1,3
      IREC(16*M+INDEX)=(1.25)*((LASTVAL(M+6)-IREC(M+43*INDEX)) +
      8*(IREC(M+27*INDEX)-LASTVAL(M+9))) + 128
155      1
      IREC(22*M+INDEX)=(1.25)*((LASTVAL(M+9)-IREC(M+59*INDEX)) +
      8*(IREC(M+43*INDEX)-IREC(M+11*INDEX))) + 128
160      1
      150 CONTINUE
      DO 160 M=36,196,16
      DO 160 N=1,3
      M1=M+N*INDEX
      IREC(M1)=(1.25)*((IREC(M1-27)-IREC(M1+37)) +
      8*(IREC(M1+21)-IREC(M1-11))) + 128
165      1
      160 CONTINUE
      CALL PR2
      RETURN
      END

```

PROGRAM COMPAL

```

1 SURROUTINE PRI1  

   DIMENSION IP(125),IREG01(6*),YBL1(6*),LIST(11)  

   COMMON INDIX,IREC(486)  

5  COMMON/GNTR/ KOUNT,KOUNT,NKUNT,KNTF1,KNTF2,KNTF3  

   CUMON/CNTF2/ IKOUNT,IRODCNT,IERCNT,IISICNT  

   GNHON/IGNR/ KKNY,IKNT,IKNTF1,IKNTF2  

   COMMON/IGNR2/ IKNY,ITGDCNT,IERCNT,IISICNT  

   COMMON/GN3/ IAG  

10  COMMON/CM4/ INCR,II,IIIN  

   GNHON/SWICH/ ISWICH,ISWICH2,ISWICH3,IEND  

   COMMON/THRES/ IIP,ITQ,ITR,IIIN,IIINZ,ITALL,ITAS,ITFC  

   DATA YBL1/4,-2,5,12,8,7,7-9,4,7,8,4,7,8,4,7,8,4,7,8,4,7,8,4,  

   DATA LIST /10,15,13,14,12,5,11,19,1,90,6/  

15  C*****REPEAT BEFORE EACH FLIGHT...IE,FIRST RECORD***  

   C *****SWITCH1 DEFINITION*****  

   C SWITCH1=0 WHEN IT HAS GONE THROUGH PREVIOUS CAL.  

   C SWITCH1=1 WHEN IT HAS FOUND THE FIRST GOOD SEG.  

   C OF DATA---PRINT TYPE 1 RECORD  

   C SWITCH1=2 WHEN IT IS TO PROCESS GAL DATA THIS  

   C ALLOWS FOR INITIALIZATION INTO ARRAY.  

   C *****  

   IF(IISWICH1.EQ.1)GO TO 135  

   IF(IISWICH1.EQ.2) INCR$1=26  

25  C *****CALIBRATED INFORMATION***  

   C *****  

   INCR=INGRS11-1  

   DO 30 M=1,11  

30  INCR=INCR+1  

   IF(MOD(INCR,8).NE.0)GO TO 20  

   IREG01(INCR)=0  

   INCR=INCR+1  

   IREG01(INCR)=IREG(LIST(M)+INDEX)  

35  CONTINUE  

   ISWICH1=0  

   IF(INCR.GE.62)ISWICH1=2  

   INGR$1=INCR  

   RETURN  

40  135 INCR=INGRS11-1  

   C *****CREATE TYPE3 REGIONS*****  

   C *****  

   ISWICH3=0  

45  CALL PRT3  

   DO 137 N=INCR,64  

   IREC01(N)=0  

   137 CONTINUE  

   IREC01(1)=1  

   IREG01(2)=IAG  

   IREC01(3)=0  

50  C *****DOCUMENTARY DATA*****  

   C *****  

   INCR=4  

55  DO 140 M=53,127,16  

   DO 140 F=1,3
```

TABLE A.1 (Continued)

PROGRAM COMPAL

PAGE 2

10/19/79 15.41.39

FTN 47476

74/74 OPT=1 TRACE

SUBROUTINE PRT1

IF (M+N.EQ.98.OR.M+N.GT.128) GO TO 140

IREC01(INCR)=IREC(M+N*INDEX)

INCR=INCR+1

140 CONTINUE

IREC01(16)=0

C ***THRESHOLDS***

IREC01(17)=ITALI

IREC01(18)=ITAS

IREC01(19)=ITFC

IREC01(20)=ITNY

IREC01(21)=ITNZ

IREC01(22)=ITP

IREC01(23)=ITQ

IREC01(24)=0

IREC01(25)=ITR

IJ=IK-1

DO 155 M=1,25

IP1(M)=0

155 CONTINUE

CALL PACK(IREC01,6,IP1,IJ,IK,IT0L1)

ISWICH1=2

KNTRI=KNTRI+1

C WRITE(7)(IP1(J),J=1,25)

RETURN

END

73

TABLE A.1 (Continued)

PROGRAM COMPAL

SUBROUTINE PR12 7474 ORI=1 IRAGE 10/19/79 16.41.39 PAGE 1

```

1  SUBROUTINE PR12
   DIMENSION IP2(25),IREG02(107),IFBL2(107)
   COMMON INDEX,IFEC(46)
   COMMON/CH1/ KOUNT,LKOUNT,NKNT,KNTK1,KNTK2,KNTK3
   COMMON/CH2/ IKUNT,IGDCNT,IERRCNT,ISHTCNT
   COMMON/CH3/ KTKNT,LTKNT,NTKNT,KTKR2
   COMMON/CH4/ IKNT,ITGDCNT,ITERCNT,ITSICNT
   COMMON/CH5/ INCRST1,ITH
   COMMON/CH6/ ISWICH1,ISWICH2,ISWICH3,IEND
   COMMON/TEMP/ ITEMP(6)
   DATA IFBL2/4,16,105*8/

C *****DATA IN ACTIVE PERIODS****
C *****OUTPUTED IF ACTIVE CONDITIONS ARE MET,OUTPUT 1-SEG AFTER GRITE
C *****FAILS****
15  IF (ISWICH2.EQ.1)60 TO 20
C *****TITLE****
C *****TYPE,TIME SEC.,AIR/SPEED,ALT,FCI,E(1-4),E(5-9)*****
20  IREC02(1)=2
   IREG02(2)=ITH
   IREC02(3)=IREC(1+INDEX)
   IREG02(4)=IREC(9+INDEX)
   IREC02(5)=IREC(6+INDEX)
   IREG02(6)=IREC(16+INDEX)
   IREC02(7)=IREC(17+INDEX)

C *****NY,NZ,P,Q,R,P,Q,R,SB,PLA*****
30  INCR=8
   DO 10 M=3,227,16
     IREC02(INCR)=IREC(7+M+INDEX)
     IREG02(INCR+1)=IREC(12+M+INDEX)
     IREC02(INCR+2)=IREC(10+M+INDEX)
     IREG02(INCR+3)=IREC(11+M+INDEX)
     IREC02(INCR+4)=IREC(9+M+INDEX)
     IREG02(INCR+5)=IREC(5+M+INDEX)
     IREC02(INCR+6)=IREC(6+M+INDEX)
     IREG02(INCR+7)=IREC(4+M+INDEX)
     IREC02(INCR+8)=IREC(2+M+INDEX)
     IREG02(INCR+9)=IREC(11+M+INDEX)
     IREG02(INCR+10)=IREC(11+M+INDEX)
     IREG02(INCR+11)=8
     INCR=INCR+12
10  CONTINUE
   RETURN
20  INCR=8
   ISWICH2=0
   KNTK2=KNTK2+1
   DO 30 M=160,180,12
     DO 30 N=1,3
       INCR=INCR+1
       IREG02(M+N)=ITEMP(INCR)
30  CONTINUE
   IK=J=1
   DO 40 M=1,25

```

TABLE A.1 (Continued)

PROGRAM COMPAL

PAGE 2

10/19/79 16.41 39

FIN 4.7+476

74/74 OPT=1 TRACE

SUBROUTINE PR12

```

        IP2(H)=0
        40 CONTINUE
        CALL PACK(LREC02,187,IP2,IJ,IK,ITBL2)
        C      WRITE(7)(IP2(I),J=1,25)
        RETURN
        END
    
```

PROGRAM COMPAL

```

1  SUBROUTINE PRT3
   DIMENSION IP3(25),IREC03(17),ITBL3(17)
   COMMON INDEX,IREC(486)
   COMMON/CH3/ IAC
5  COMMON/CH4/ INCRST1,ITH
   COMMON/CHIR/ KOUNT1,LKOUNT1,KKOUNT,KNIR1,KNIR2,KNIR3
   COMMON/CNTR2/ IKOUNT,IGROCHT,IERRCHT,ISHTCHT
   COMMON/TCNTR/ KTKNT,LTKN1,NTKNT,KNIR2
10  COMMON/TCNTR2/ IIKNT,IIGDOCHT,IIERCHT,IISHTCHT
   COMMON/SWICH/ ISWICH1,ISWICH2,ISWICH3,IEND
   DATA ITBL3/4,-24,52,12*0,4,60/
C *****END OF FLIGHT (TAIL)*****
C *****SWITCH3 DEFINITION*****
15  C ISWICH3=0 WHEN INITIAL ENTRY STOR DUC DATA.
   C ISWICH3=1 WHEN ITS TO SAVE TIME SEC. AND WRITE TYPE 3 REC.
C *****
   IF (ISWICH3.EQ.1)GO TO 11
   IREC03(1)=3
   IREC03(2)=IAC
   IREC03(3)=0
20  C *****DOCUMENTARY DATA*****
C
25  C INCR=4
   DO 10 M=63,127,16
     DO 10 N=1,3
       IF (M+N.EQ.98.OR.M+N.GT.128)GO TO 10
       IREC03(INCR)=IREC(M+N*INDEX)
       INCR=INCR+1
30  C
   10 CONTINUE
   ISWICH3=1
   RETURN
   11 IREC03(16)=0
   ISWICH3=0
35  C
   IEND=1
   IJ=IK=1
   IREC03(17)=ITH
   IF (KOUNT.EQ.0)GO TO 30
   DO 20 M=1,25
     IP3(M)=0
40  C
   20 CONTINUE
   CALL PACK(IREC03,17,IP3,IJ,IK,ITBL3)
   KNIR3=KNIR3+1
45  C
   WRITE(7)(IP3(J),J=1,25)
C *****
C *****INDIVIDUAL FLIGHT STATISTICS*****
C *****
50  C
   WRITE(6,1010)KNIR1
1010  FORMAT(8H9FLIGHT#,I3,11H STATISTICS)
   WRITE(6,1020)KNIR2
1020  FORMAT(41H 1. THE NUMBER OF TYPE2 RECORDS PRINTED,I7)
55  C
   WRITE(6,1060)IGROCHT
1060  FORMAT(28H 2. SECONDS OF GROUND DATA,I7)
   WRITE(6,1090)KOUNT
1090  FORMAT(10H9KOUNT,I7)

```

1090 FORMAT(28H 3. SECONDS OF ACTIVE DATA,I7)

WRITE(6,1110)ITM

1110 FORMAT(29H 4. TOTAL NUMBER OF SECONDS,77)

$$30-KTHTR2=KNTK2+KTHTR2$$

ITERCNT=ITERCNT+1*ERCNT

~~IT IS TENTATIVE~~
$$I(TGDCNT) = I(TGDCNT) + I(TGDCNT)$$

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

KT KNI = KUONI + KI KNI
LT KNT = L KONT + LI KNT

$\text{EITKNIT} = \text{EKNIT} + \text{EITKNIT}$
 $\text{RITKNIT} = \text{ITM} + \text{RITKNIT}$

KNTRF,2=IE*RCNT=ISHTCNT=IGRCNT=0

$$\Delta KOUNT_i = KOUNT = LKOUNT = 0$$
~~ACCOUNT=NK6UNT-ITM~~

RETURN

END

TABLE A.2

PROGRAM COMPA2

PROGRAM COMPA2 7474 OPT=1 TRACE FTN 4074670 11/19/79 1.2.29 PAGE 1

1 C PROGRAM COMPA2(OUTPUT,OUTPUT,TAPE5=OUTPUT,TAPE6=OUTPUT,TAPE1,TAPE7)

2 C *****

3 C *****

4 C *****

5 C THIS ROUTINE IS THE FIRST COMPRESSION METHODOLOGY DESIGNED

6 C FOR THE CIA AND C130 MODELS. USING INITIALLY DETERMINED

7 C THRESHOLDS AS DETERMINANTS AS TO WHETHER A SECOND IS ACTIVE

8 C OR INACTIVE DATA.

9 C *****

10 C *****

11 C *****

12 C *****

13 C *****

14 C *****

15 C *****

16 C *****

17 C *****

18 C *****

19 C *****

20 C *****

21 C *****

22 C *****

23 C *****

24 C *****

25 C *****

26 C *****

27 C *****

28 C *****

29 C *****

30 C *****

31 C *****

32 C *****

33 C *****

34 C *****

35 C *****

36 C *****

37 C *****

38 C *****

39 C *****

40 C *****

41 C *****

42 C *****

43 C *****

44 C *****

45 C *****

10 DIMENSION NTRK(324), ITBL(486)

COMMON INDEX, IRECT(486)

COMMON /DLDVL/ IOVR, IOVQ, IOVNY, IOVNZ, IOVALI, IOVAS, IOVE1, IOVES

COMMON /CM1/ LAST

COMMON /CM3/ IAC

COMMON /CM4/ ITH

COMMON /CNTR/ KOUNT, LKOUNT, NKOUNT, KNTR1, KNTR2, KNTR3

COMMON /CNTR2/ IKOUNT, IGRDONT, IERECNT, ISHTCNT

COMMON /CNTRP/ KTKNT, LTKNT, NTKNT, KNTR2

COMMON /CNTR2/ ITKNT, ITGDCNT, ITERCNT, ITSTCNT

COMMON /SWTCH/ ISWCH1, ISWCH2, ISWCH3, IEND

COMMON /TEMP/ IQOUT(20), IQOUT(20)

COMMON /THRES/ ITO, ITR, ITNY, ITNZ, ITALI, IIAS

DATA ITBL/486*87

25 C *****

26 C *****

27 C *****

28 C *****

29 C *****

30 C *****

31 C *****

32 C *****

33 C *****

34 C *****

35 C *****

36 C *****

37 C *****

38 C *****

39 C *****

40 C *****

41 C *****

42 C *****

43 C *****

44 C *****

45 C *****

46 C *****

47 C *****

48 C *****

49 C *****

50 C *****

51 C *****

52 C *****

53 C *****

54 C *****

10

15

20

25

30

35

40

45

50

55

60

65

70

75

80

85

90

95

100

105

110

115

120

125

130

135

140

145

150

155

160

165

170

175

180

185

190

195

200

205

210

215

220

225

230

TABLE A.2 (Continued)

PROGRAM COMPA2

FIN 1, 1, 5

PROGRAM COMPA2 74/74 OPT=1 TRACE

13/19/ 9 1 2 29

-AOL 2

```

C ***** MAIN LOOP DECLARATION *****
C 2 SECONDS/UPACK...10 SECONDS/BUFFER IN.
C
C DO 100 K=1,5
C   CALL UNPACK(IREC,486,NIBLK,JH,JH,ITBL)
C   IF(ISHUTCH.EQ.0)GO TO 93
C ***** PARITY ERROR CHECK *****
C
C   IF(OR(SHIFT(IREC(483),8),IREC(484)).NE.0) GO TO 91
C ***** SHORT RECORD CHECK *****
C
C   IF(AND(IREC(480),15).NE.9) GO TO 92
C   INDEX=0
C   DO 90 L=1,2
C     IF(L.EQ.2) INDEX=240
C ***** BIT CYCLE DETERMINATION *****
C
C   JBIT=IREC(232+INDEX)/128
C   JCAL=AND((IREC(232+INDEX)/64),1)
C   JPRE=AND((IREC(232+INDEX)/32),1)
C   IF(JBIT.EQ.1.AND.INFLIGHT.NE.64.AND.IEND.NE.1)CALL PR13
C   IF(JBIT.EQ.1.OR.JCAL.EQ.1.OR.JPRE.EQ.1)IKOUNT=IKOUNT+1
C   IF(JCAL.EQ.1.OR.JPRE.EQ.1)GO TO 90
C   IF(JBIT.EQ.1)GO TO 35
C
C 79  C ***** CALCULATION OF THE SECONDS INTO THE FLIGHT *****
C
C   ITM=NKOUNT-10+2*(K-1)*L
C ***** IN FLIGHT CHECK *****
C
C   IEND=0
C   INFLIGHT=AND(IREC(64+INDEX),64)
C   IF(INFLIGHT.EQ.0)IGROCHT=IGROCHT+1
C   IF(IGR.EQ.1)GO TO 25
C   IF(INFLIGHT.EQ.0)GO TO 90
C
C ***** DONE ONLY ONCE AFTER BIT CYCLE TEST HAS BEEN SHUT DOWN *****
C   TO INITIALIZE THE MEAN VALUES OF THE ALT,AS,E(1-9)
C *****
C
C 25  IF(ISHUTCH1.NE.0)GO TO 30
C   ISHTCH1=2
C   DO 26 M=1,2
C     IQDOT(M)=128
C     IRDOT(M)=128
C   CONTINUE
C   DO 27 N=3,10
C     IQDOT(M)=1.6667*((IREC(M-3)*12+1+INDEX)-
C       IREC(M+2)*12+1+INDEX))
C       +8*(IREC(M+1)*12+1+INDEX)-
C       IREC(M-1)*12+1+INDEX))+128
C     IRDOT(M)=1.6667*((IREC(M-3)*12+2+INDEX)-

```


SUBROUTINE ACTDATA 74/7/73 CPT=1 TRACE

```

1 SUBROUTINE ACICATA
   DIMENSION LASTIQ(4),LASTR(4)
   COMMON INDEX,IREC(486)
5  COMMON /CLOVL/ IDVR,IOVQ,IOVNZ,IOVALT,IOVAS,IOVE1,IOVES
   COMMON /CM1/ LAST
   COMMON /CM4/ ITM
   COMMON /CHTR/ KOUNT,LKOUNT,NKOUNT,KHTR1,KHTR2,KHTR3
   COMMON /CNTR2/ IKOUNT,IGRCMT,IERRCNT,ISHTCNT
   COMMON /CNTR/ KTNT,LTNT,NTKT,KNTR2
10  COMMON /CHTR2/ ITKNT,LGDCNT,ITERCNT,ISTCNT
   COMMON /SWICH/ ISWICH1,ISWICH2,ISWICH3,IEND
   COMMON /TEMP/ IQDOT(20),IRDOI(20)
   COMMON /THRES/ ITQ,ITR,ITNY,ITNZ,ITALY,ITAS
      C
15  C *****
      C ACTIVE DATA DETERMINED BY IF OLD VALUE > THRESHOLD <
      C VALUE OR IF OLD VALUE - THRESHOLD > VALUE
      C VALUES ARE NY,NZ,Q,R
      C OLD VALUES DEFINED FROM THE MEAN OF THE DISTRIBUTION
      C THRESHOLD ARE INITIALIZED FROM THE INPUT
20  C *****
      C
      C TO IO M=1,229,12
      C IF(IABS(IREC(M+INDEX)-IOVQ).GT.IIQR.OR.
25  # IABS(IREC(M+1+INDEX)-IOVR).GT.ITR)GO TO 20
      C
10 CONTINUE
      DO 15 M=3,235,8
      C IF(IABS(IREC(M+INDEX)-IOVNZ).GT.ITNZ.OR.
      C IABS(IREC(M+1+INDEX)-IOVNY).GT.ITNY)GO TO 20
15 CONTINUE
      C IF(IABS(IREC(40+INDEX)-IOVALT).GT.ITALY.OR.
      C IABS(IREC(47+INDEX)-IOVAS).GT.ITAS.OR.
30  1 IOVE1.NE.IREC(64+INDEX).OR.
      C
      C 3 IOVE5.NE.IREC(112+INDEX))GO TO 20
35  C *****
      C INACTIVE DATA RECOGNIZED: LAST WILL BE SET TO ONE IF
      C PREVIOUS DATA ACTIVE.. THIS WILL ALLOW FOR EACH ACT
      C IVE DATA TO BE FOLLOWED BY ONE SECOND OF INACTIVE
      C DATA.
40  C *****
      C
      C LKOUNT=LKOUNT+1
      C IF(LAST.EQ.1)GO TO 50
      C RETURN
45  20 IF(KCOUNT.NE.0)GO TO 25
      C ISWICH1=1
      C CALL PR1
      C KOUNT=KOUNT+1
      C ISWICH2=1
      C M 23 M=19,20
      C IQDOT(M)=128
      C IRDOI(M)=128
      C
      C 23 CONTINUE
      C CALL PR2
55  25 IOVALT=IREC(40+INDEX)
      C IOVAS=IREC(47+INDEX)

```

TABLE A.2 (Continued)

PROGRAM COMPA2

SUBROUTINE ACTDATA 74/74 OPT=1 TRACE FIN 4.7+476 10/19/79 1.24.29 PAGE 2

```

IOVEI=IREC(4+INDEX)
IOVEF=IREC(112+INDEX)
COUNT=COUNT+1

```

60

```

C RESET THE OLD VALUES, ACTIVE DATA HAS BEEN FOUND..

```

```

C SAVE THE LAST FOUR VALUES OF Q,R..

```

```

C THEN CALCULATE Q-R. FOR 3-18.

```

```

C SET 1-2 Q-R. TO 128

```

65

```

C INCR=0

```

```

DO 30 M=193,229,12

```

```

  INCR=INCR+1

```

```

  LASTQ(INCR)=IREC(M+INDEX)

```

```

  LASTR(INCR)=IREC(M+1+INDEX)

```

```

  30 CONTINUE

```

```

  IF(LAST.EQ.1)GO TO 110

```

```

  DO 35 M=1,2

```

```

    IQDOT(M)=128

```

```

    IRDOT(M)=128

```

```

  35 CONTINUE

```

```

  DO 40 M=3,18

```

```

    IQDOT(M)=(1.6667)*((IREC(M-3)*12+1+INDEX)-IREC(M+2)*12+1+INDEX)

```

```

    X)))+(IREC(M+1)*12+1+INDEX)-IREC(M-1)*12+1+INDEX))*128

```

```

    IRDOT(M)=(1.6667)*((IREC(M-3)*12+2+INDEX)-IREC(M+2)*12+2+INDEX)

```

```

    X)))+(IREC(M+1)*12+2+INDEX)-IREC(M-1)*12+2+INDEX))*128

```

```

  40 CONTINUE

```

```

  LAST=1

```

```

  CALL PRY2

```

```

  RETURN

```

90

```

C ACTIVE DATA NOT FOUND CALCULATE LAST 2 Q-R. FOR 19-20

```

```

C OF PREVIOUS RECORD. STORE INTO TYPE2, GO ON AND CALC-

```

```

C ULATE Q-R. FOR THIS RECORD 1-18 AND STORE INTO TYPE2.

```

95

```

  50 LAST=0

```

```

  IQDOT(19)=(1.6667)*((LASTQ(1)-IREC(1+INDEX))+8*(LASTQ(4)-LASTQ(2))

```

```

  )+128

```

```

  IQDOT(20)=(1.6667)*((LASTQ(2)-IREC(13+INDEX))+8*(IREC(1+INDEX)-LAS

```

```

  TR(3)))+128

```

```

  IRDOT(19)=(1.6667)*((LASTR(1)-IREC(2+INDEX))+8*(LASTR(4)-LASTR(2))

```

```

  )+128

```

```

  IRDOT(20)=(1.6667)*((LASTR(2)-IREC(14+INDEX))+8*(IREC(2+INDEX)-LAS

```

```

  TR(3)))+128

```

```

  ISNICH2=1

```

```

  CALL PRY2

```

```

  IQDOT(1)=(1.6667)*((LASTQ(3)-IREC(25+INDEX))+8*(IREC(13+INDEX)-LAS

```

```

  TR(4)))+128

```

```

  IQDOT(2)=(1.6667)*((LASTQ(4)-IREC(37+INDEX))+8*(IREC(25+INDEX)-IRE

```

```

  C(1+INDEX)))+128

```

```

  IRDOT(1)=(1.6667)*((LASTR(3)-IREC(26+INDEX))+8*(IREC(1+INDEX)-LAS

```

```

  TR(4)))+128

```

```

  IRDOT(2)=(1.6667)*((LASTR(4)-IREC(39+INDEX))+8*(IREC(26+INDEX)-IRE

```

```

  C(1+INDEX)))+128

```

110

TABLE A.2 (Continued)

PROGRAM COMPA2

3

PAGE

11/19/79 17.24.29

FTN 1173

SUBROUTINE ACIDATA 74/74 OPT=1 IPACE

```

115      DO 60 F=3,10
          IQDOT(M)=(1.6667)*((IREC(M-3)*12+1+INDEX)-IREC(M+2) 12+1+INDE
          X))+8*(IREC(M+1)*12+1+INDEX)-IREC(M-1)*12+1+INDEX))+1
          28
          #
120      IROOT(M)=(1.6667)*((IREC(M-3)*12+2+INDEX)-IREC(M+2) 12+2+INDE
          X))+8*(IREC(M+1)*12+2+INDEX)-IREC(M-1)*12+2+INDEX))+1
          28
          #
          60 CONTINUE
          CALL PR12
          ISWCH2=1
          DO 70 N=19,20
              IQDOT(M)=128
              IROOT(M)=128
          70 CONTINUE
          CALL PR12
          IOVALT=IREC(40+INDEX)
          IOVAS=IREC(47+INDEX)
          IOVEL=IREC(54+INDEX)
          IOVES=IREC(112+INDEX)
          RETURN
          C
135      C *****
          C ACTIVE DATA FOUND AND FOLLOWS ANOTHER ACTIVE SECOND.
          C CALCULATE LAST 2 VALUES OF Q.-R. FOR 19-20 OF PREV.
          C IONS RECORD. STORE INFO TYPE3, GO ON AND CALCULATE
          C Q.-R. FOR 1-10 NOT TO BE STORED UNIL 19-20 ARE
          C CALCULATED.
          C *****
          C
140      IQDOT(19)=(1.6667)*((LASTQ(1)-IREC(1+INDEX))+8*(LASTQ(4)-LASTQ(2))
          )+128
          IQDOT(20)=(1.6667)*((LASTQ(2)-IREC(13+INDEX))+8*(IREC(1+INDEX)-LAS
          TQ(3)))+128
          IROOT(19)=(1.6667)*((LASTR(1)-IREC(2+INDEX))+8*(LASTR(4)-LASTR(2))
          )+128
          IROOT(20)=(1.6667)*((LASTR(2)-IREC(14+INDEX))+8*(IREC(2+INDEX)-LAS
          TR(3)))+128
          ISWCH2=1
          CALL PR12
          IQDOT(1)=(1.6667)*((LASTQ(3)-IREC(25+INDEX))+8*(IREC(13+INDEX)-LAS
          TQ(4)))+128
          IQDOT(2)=(1.6667)*((LASTQ(4)-IREC(37+INDEX))+8*(IREC(25+INDEX)-IRE
          C(1+INDEX)))+128
          IROOT(1)=(1.6667)*((LASTR(3)-IREC(26+INDEX))+8*(IREC(14+INDEX)-LAS
          TR(4)))+128
          IROOT(2)=(1.6667)*((LASTR(4)-IREC(39+INDEX))+8*(IREC(26+INDEX)-IRE
          C(2+INDEX)))+128
          DO 120 M=3,10
              IQDOT(M)=(1.6667)*((IREC(M-3)*12+1+INDEX)-IREC(M+2) 12+1+INDE
              X))+8*(IREC(M+1)*12+1+INDEX)-IREC(M-1)*12+1+INDEX))+1
              28
              #
              IROOT(M)=(1.6667)*((IREC(M-3)*12+2+INDEX)-IREC(M+2) 12+2+INDE
              X))+8*(IREC(M+1)*12+2+INDEX)-IREC(M-1)*12+2+INDEX))+1
              28
              #
          120 CONTINUE
          CALL PR12
          RETURN

```

TABLE A.2 (Continued)

PROGRAM COMPA2

FTN 1.7+476

10/19/ 9 1.2.29

PAGE

74/74 CPT=1 TRACE

SUBROUTINE ACTDATA

END

PROGRAM COMP A2

```

1 SUBROUTINE PRT1
  DIMENSION IP(30), IREC01(87), ITBL1(87), INDD(12), LIST(17)
  COMMON INDEX, IREC(486)
  COMMON/CNTR/ KCNT,LKOUNT,NKOUT,I,KFIR1,KHTR2,KNTR3
  COMMON /CHTR2/ IKOUNT,IGROCNT,TERRCNT,ISHTCNT
  COMMON /TKHTR/ TKHTR,LIKHT,NIKHT,KHTR2
  COMMON /TCNTR2/ ITKNT,ITGCHT,ITERCNT,ITSTCNT
  COMMON /CM3/ IAC
  COMMON /CM4/ ITH
  COMMON /SWICH/ ISWICH1,ISWICH2,ISWICH3,IEND
  COMMON /THRES/ ITQ,ITR,ITY,ITRZ,ITALY,ITAS
  DATA ITBL1/4,-24,52,12*8,4,6*8,12*7*8,7*8,4,7*8,4,7*8,4,
    # 7*8,4,7*8,4,7*8,4/
  DATA INDD/48,86,95,96,136,143,144,184,191,192,232,239/
  DATA LIST/1,2,3,4,5,6,7,8,9,10,15,16,23,24,40,47,208/
  C *****SWICH1 DEFINITION*****
  C ISWICH1=0 WHEN IT HAS GONE THROUGH PREVIOUS CALDATA
  C ISWICH1=1 WHEN IT HAS FOUND THE FIRST GOOD SEC. OF DATA--
  C PRINT TYPE 1 RECORD.
  C ISWICH1=2 WHEN IT IS TO PROCESS CALDATA THIS
  C ALLONS FOR INITIALIZATION INTO ARRAY.
  C *****
  IF(ISWICH1.EQ.1) GO TO 100
  IF(ISWICH1.EQ.2) INCRST1=24
  INCR=INCRST1-1
  C
  C ***** CALIBRATION INFORMATION *****
  C
  DO 50 M=1,17
    INCR=INCR+1
    JINCR=INCR+1
    IF(MOD(JINCR,8).NE.0)GO TO 40
    IREC01(INCR)=0
    INCR=INCR+1
  35 40 IREC01(INCR)=IREC(LIST(M)+INDEX)
    50 CONTINUE
    ISWICH1=0
    IF(INCR.GE.81) ISWICH1=2
    INCRST1=INCR
    RETURN
  40 100 INCR=INCRST1-1
  C
  C ***** GREATE TYPES RECORD *****
  C
  ISWICH3=0
  CALL PRT3
  DO 110 M=INCR,87
    IREC01(M)=0
  45 110 CONTINUE
    IREC01(1)=1
    IREC01(2)=IAC
    IREC01(3)=0
  50
  C ***** DOCUMENTARY DATA *****
  C
  CO 120 M=1,12
    IREC01(M+3)=IREC(INDD(M)+INDEX)

```

TABLE A.2 (Continued)

PROGRAM COMPA2

10/19/ 9 1 .25.25

PAGE 2

SUBROUTINE PR14 74/74 CPI=1 TRACE FRI 4.7.77

125 CONTINUE

IRECO1(16)=0

60

C ***** THRESHOLDS *****

65

IRECO1(17)=ITALY

IRECO1(18)=ITAS

IRECO1(19)=ITNY

IRECO1(20)=ITNZ

IRECO1(21)=ITQ

IRECO1(22)=ITR

IRECO1(23)=0

70

IJ=1

IK=1

CO 130 M=1,38

IP1(N)=0

130 CONTINUE

75 CALL PACK(IRECO1,87,IP1,IJ,IK,118L1)

ISN7CHI=2

KNTR1=KNTR1+1

C WRITE(7)(IP1(J),J=1,38)

RETURN

END

80

86

TABLE A.2 (Continued)

PROGRAM COMPA2

FILE 1, 2, 29

PAGE 1

SURROUTINE PR2 74/74 OPT=1 IRACE

```

1  SURROUTINE PR2
   DIMENSION IP2(38),IREC02(20),ITBL2(20)
   COMMON INDEX,IREC(486)
   COMMON /CH4/ ITH
5  COMMON /CHTR/ KOUNT,LKOUNT,KCOUNT,KNTR1,KNTR2,KNTR3
   COMMON /SWICH/ ISWICH1,ISWICH2,ISWICH3,IEHD
   COMMON /TEMP/ IQDOT(20),IRDOT(20)
   DATA ITBL2/4,16,279*8/
C ***** DATA IN ACTIVE PERIODS *****
10 C ***** OUTPUTTED IF ACTIVE CONDITIONS ARE MET *****
C ***** OUTPUT 1 SECOND AFTER CRITERIA FAILS *****
C
   IF(ISWICH2.EQ.1)GO TO 20
15 IREC02(1)=2
   IREC02(2)=ITH
   DO 10 M=3,241
     IREC02(M)=IREC(M-2*INDEX)
20 10 CONTINUE
   RETURN
20 ISWICH2=0
   DO 30 M=242,261
     IREC02(M)=IQDOT(M-241)
25 30 CONTINUE
   DO 40 M=262,281
     IREC02(M)=IRDOT(M-261)
40 CONTINUE
   IJ=IK=1
   DO 50 M=1,38
     IP2(M)=0
50 CONTINUE
   CALL PACK(IREC02,20,IP2,IJ,IK,ITBL2)
C   WRITE(7)(IP2(M),M=1,38)
   KNTR2=KNTR2+1
35 RETURN
   END

```

TABLE A.2 (Continued)

PROGRAM COMPA2

PAGE 1

11/13/79 17.17.23

FIN 4.7+476

74274 CPT=1 TRACE

SUBROUTINE PR13

```

1      SUBROUTINE PR13
      DIMENSION IP3(130), IREC03(17), IIBL3(17), INDD(12)
      COMMON INDEX, IREC(486)
      COMMON /CM3/ IAC
      COMMON /CM4/ ITH
      COMMON /CNTR/ KOUNT, LKOUNT, NKOUNT, KNTR1, KNTR2, KNTR3
      COMMON /CNTR2/ IKOUNT, IGRDNT, IERRCNT, ISHTCNT
      COMMON /TCNTR/ KIKNT, LKNT, NKNT, KINTR2
      COMMON /TCNTR2/ ITKNT, ITGCHT, IERRCNT, IISHTCNT
      COMMON /SWTCH/ ISWCH1, ISWCH2, ISWCH3, IEND
      DATA IIBL3/4, -24, 52, 12*0, 4, 60/
      DATA INDD/48, 88, 95, 96, 135, 143, 144, 144, 191, 192, 232, 239/

      C ***** END OF FLIGHT (TAIL) *****
      C ***** SWTCH3 DEFINITION *****
      C ***** SWTCH3=0 WHEN INITIAL ENTRY STORE DOC.DATA *****
      C ***** SWTCH3=1 WHEN ITS TO SAVE TIME SEC. AND WRITE *****
      C ***** TYPE3 RECORD *****
      C ***** DOCUMENTARY DATA *****

20      IF (ISWCH3.EQ.1) GO TO 11
      IREC03(1)=3
      IREC03(2)=IAC
      IREC03(3)=0

25      C ***** DOCUMENTARY DATA *****

      DO 10 M=1, 12
      IREC03(M+3)=IREC(INDD(M)+INDEX)
10      CONTINUE
      ISWCH3=1
      RETURN

35      11 IREC03(16)=0
      IREC03(17)=ITH
      IEND=1
      ISWCH3=0
      IF (KOUNT.EQ.0) GO TO 30
      KNTR3=KNTR3+1
      IJ=IK=1
      DO 20 M=1, 28
      IP3(M)=0
20      CONTINUE
      CALL PACK(IREC03, 17, IP3, IJ, IK, IIBL3)
      WRITE(7) (IP3(I), J=1, 36)

45      C *****
      C ***** FINAL INDIVIDUAL FLIGHT STATISTICS *****
      C ***** WRITE(6, 1010) KNTR1 *****
      1010 FORMAT(4H0FLIGHT#, I3, 11H STATISTICS)
      WRITE(6, 1020) KNTR2
      1020 FORMAT(4H1. THE NUMBER OF TYPE2 RECORDS PRINTED, I7)
      WRITE(6, 1060) IGRDNT
      1060 FORMAT(20H 2. SECONDS OF GROUND DATA, I7)
      WRITE(6, 1090) KOUNT
      1090 FORMAT(20H 3. SECONDS OF ACTIVE DATA, I7)

```

TABLE A.2 (Concluded)

PROGRAM COMPA2

SUBROUTINE PKIS 7/4/74 00111 1KAGE 10/19/79 17.24.29 AGE 2

```

1110 WRITE(6,1110)ITM
1110 FORMAT(29H 4. TOTAL NUMBER OF SECONDS:,I7)
30 KNTIR2=KNTIR2+KNTIR2
ITERCNT=IERRCNT+ITERCNT
ITSTCNT=ISHTCNT+ITSTCNT
ITGDCNT=IGDCNT+ITGDCNT
65 ITKNT=IKCNT+ITKNT
KTKNT=KOUNT+KTKNT
LTKNT=LKOUNT+LTKNT
NTKNT=NTKNT+ITM
KNTIR2=IERRCNT+ISHTCNT+IGDCNT=0
IKCNT=KOUNT=LKOUNT=0
70 NKCNT=NKOUNT-ITM
RETURN
END

```

TABLE A.3

PROGRAM COMPB1

PAGE 1

10/23/79 16.16.56

FIN 4.7+476

OPT=1 TRACE

74/74

```

1      PROGRAM COMPB1(INPUT,OUTPUT,TAPE3=INPUT,TAPE6=OUTPUT,TAPE1,TAPE7)

```

```

C      ***** RON JANNING UDRI 547 93579-00 *****
C      *****

```

```

C      THIS ROUTINE IS THE SECOND COMPRESSION METHODOLOGY

```

```

C      DESIGNED FOR THE A10 AIRCRAFT. USING A FLOATING MEAN

```

```

C      AND WINDOW VALUES AS DETERMINANTS TO WHETHER A

```

```

C      PARAMETER IS ACTIVE DATA.

```

```

C      *****

```

```

C

```

```

10     DIMENSION NTBLK(324),ITBL(486),LIST(16)

```

```

      DIMENSION LASTVAL(12),LISTA(16),LISTB(16)

```

```

      COMMON INDEX,IREC(486)

```

```

      COMMON/C7/IREC02(96),IDIR(16),IDVALUE(16)

```

```

      COMMON/CH3/IAC,MRECHT,INCRST1

```

```

      COMMON/SWITCH/ISWITCH1,ISWITCH2,ISWITCH3,IEND

```

```

      COMMON/CA/IBOUND(32),LFRAME(32),LSAMPLE(32)

```

```

      COMMON/CH4/ITM

```

```

      COMMON/CNTR/KOUNT,MPEAK,NKOUNT,KNTR1,KNTR2,KNTR3

```

```

      COMMON/CNTR2/IKOUNT,IGRDCNT,IERRCNT,ISHTCNT

```

```

      COMMON/THRES/INWINDH(16)

```

```

      COMMON/TCNTR/KTKNT,NTKNT,KTKNT2

```

```

      COMMON/TCNTR2/ITKNT,ITGDCNT,IERRCNT,IISTCNT

```

```

      DATA ITBL/486*0/

```

```

      DATA LIST/10,15,13,11,12,8,9,7,6,3,5,11,98,1,15,17/

```

```

      DATA LISTA/15,15,15,15,13,13,13,13,15,15,15,15,1,1,1,1/

```

```

      DATA LISTB/16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,0/

```

```

      IERRCNT=ISHTCNT=IGRDCNT=IERRCNT=ITSTCNT=ITGDCNT=0

```

```

      ISWITCH2=ITM=ISWITCH4=INFLIGHT=ISWITCH5=IEND=0

```

```

      KOUNT=MPEAK=NKOUNT=IKOUNT=NTKNT=KTKNT=0

```

```

      KNTR1=KNTR2=KNTR3=KNTR2=INWINDH(15)=INWINDH(16)=0

```

```

      MRECHT=ISWITCH3=1

```

```

      DO 3 M=1,16

```

```

        IDIR(M)=1

```

```

      3 CONTINUE

```

```

      ISWITCH1=INCRST1=2

```

```

      READ(5,1)IAC,NGR,(INWINDH(N),N=1,14)

```

```

      1 FORMAT(A4,I1,14I3)

```

```

      WRITE(6,5)IAC

```

```

      5 FORMAT(12H1AIRCRAFT: ,A4)

```

```

      IF(NGR.EQ.1)WRITE(6,6)

```

```

      IF(NGR.EQ.0)WRITE(6,7)

```

```

      6 FORMAT(23H PROCESSING GROUND DATA)

```

```

      7 FORMAT(27H NOT PROCESSING GROUND DATA)

```

```

      WRITE(6,10)(INWINDH(N),N=1,8)

```

```

      10 FORMAT(52H WINDOW VALUES NY NZ P Q R P-DOT Q-DOT R-DOT,

```

```

      1/15X,13,1X,13,1X,13,1X,13,1X,13,1X,13,1X,13,1X,13)

```

```

      WRITE(6,15)(INWINDH(N),N=9,14)

```

```

      15 FORMAT(41H

```

```

      1/15X,13,1X,13,1X,13,1X,13,1X,13,1X,13,1X,13)

```

```

C

```

```

C      C READING IN 10 SECONDS

```

```

C

```

```

20     BUFFER IN(1,1)NTBLK(1),NTBLK(224))

```

```

      IF (UNIT(1))25,900,22

```

```

22     NKOUNT=NKOUNT+10

```

```

      GO TO 20

```

```

25     NKOUNT=NKOUNT+10

```

TABLE A.3 (Continued)

PROGRAM COMPB1

74/74 OPT=1 TRACE

FTN 4,7+476

10/23/79 16,16,56

PAGE 2

IN=JM=1

60 C C PROCESSING THE 10 SECONDS 2 AT A TIME

DO 100 K=1,5

CALL UNPACK(IREC,486,NIBLK,IM,JM,ITBL)

IF(ISWITCH4.EQ.1)GO TO 93

65 C C PARITY ERROR CHECK

IF(ORISHIFT(IREC(483),0),IREC(484)).NE.0)GO TO 91

70 C C SHORT RECORD CHECK

IF(AND(IREC(480),15).NE.9)GO TO 92

INDEX=0

DO 90 L=1,2

IF(L.EQ.2)INDEX=240

75 C C BUILT IN CYCLE TEST

80 C JBIT=IREC(INDEX+114)/128

JCAL=AND((IREC(INDEX+114)/64),1)

JPRE=AND((IREC(INDEX+114)/32),1)

IF(JBIT.EQ.1.AND.INFLIGHT.NE.32.AND.IEND.NE.1)CALL PR13

IF(JCAL.EQ.1.OR.JBIT.EQ.1.OR.JPRE.EQ.1)IKOUNT=IKOUNT+1

IF(JCAL.EQ.1.OR.JPRE.EQ.1)GO TO 87

IF(JBIT.EQ.1)GO TO 86

INFLIGHT=AND(IREC(16+INDEX),32)

ITM=IKOUNT-10+2*(K-1)*L

IEND=0

90 C C FLIGHT CHECK

IF(INFLIGHT.EQ.0)IGRODNT=IGRODNT+1

IF(INGR.EQ.1)GO TO 26

IF(INFLIGHT.EQ.0)GO TO 87

95 C C DONE ONCE AFTER BIT CYCLE HAS BEEN SHUT DOWN.

100 C 26 IF(ISWITCH1.NE.0)GO TO 30

ISWITCH1=2

DO 28 M=1,16

N2=LIST(M)

ISTO=LIST(M)+INDEX

IF(M.LT.6.OR.M.GT.8)GO TO 27

IREC(ISTO)=128

IOVALUE(M)=IREC(ISTO)

IBOUND(M)=IBOUND(M+16)=IOVALUE(M)

LFRAME(M)=LFRAME(M+16)=ITM

LSAMPLE(M)=LSAMPLE(M+16)=N2

CONTINUE

ITEMPTH=ITM

110 C

C CALCULATION OF LAST 2 PITCH, ROLL, AND YAW VALUES

C AFTER PREVIOUS CALCULATIONS

C ISWICH5 IS A SWITCH TO DETERMINE

TABLE A.3 (Continued)

PROGRAM COMPB1

PROGRAM COMPB1 74/74 OPT=1 TRACE

FTN 4,7+476

10/23/79 16,16,56

PAGE 3

```

115 C IF THERE IS A PREVIOUS CALCULATION.
C ISWITCH5=0 WHEN THERE IS NO PREVIOUS LASTVAL
C VALUES HAVE BEEN STORED.
C ISWITCH5=1 WHEN THERE ARE PREVIOUS VALUES FOR
C CALCULATIONS OF PITCH, YAW AND ROLL
120 C SLOPE VALUES OF THE 14TH AND 15TH OF
C THE PREVIOUS SECOND AND ALSO THE 1ST
C AND 2ND VALUES OF THE PRESENT SECOND.
C
125 C
30 IF (ISWITCH5.EQ.0) GO TO 56
DO 35 N=1,3
1 IREG(N+214+INDEX)=1.25*((LASTVAL(N*4-3)-IREC(N+11+INDEX))
+8*(LASTVAL(N*4)-LASTVAL(N*4-2)))+128
35 CONTINUE
DO 40 N=1,3
1 IREG(N+230+INDEX)=1.25*((LASTVAL(N*4-2)-IREC(N+27+INDEX))
+8*(IREC(N+11+INDEX)-LASTVAL(N*4-1)))+128
40 CONTINUE
C
135 C CHECK MAX/MIN VALUES OF LAST TWO PITCH, ROLL, AND YAW
C VALUES THEN CHECK TO SEE IF IT IS A WINDOW
C CROSSING
C
DO 50 N=6,8
ITEMP=LIST(N)+200+INDEX
JTEMP=ITEMP+16
DO 50 M=ITEMP,JTEMP,16
IF (IREC(M).GE.IBOUND(N)) GO TO 42
IBOUND(N)=IREC(M)
LFRAME(N)=ITH-1
LSAMPLE(N)=M+5-INDEX
IF (IREC(M).LE.IBOUND(N+16)) GO TO 44
IBOUND(N+16)=IREC(M)
LFRAME(N+16)=ITH-1
LSAMPLE(N+16)=M+5-INDEX
N1=ITH-1
N2=M+5-INDEX
IF (IREC(N).LE.IOVALUE(N)+IWINDOW(N)) GO TO 47
IF (KOUNT.NE.0) GO TO 46
ISWITCH1=1
CALL PRT1
DO 45 M4=1,16
CALL PRT2(M4,ITEHPM,LIST(M4),IOVALUE(M4),MRECNT,-1)
CONTINUE
45
46 CALL PRT2(N,N1,N2,IREC(N),MRECNT,1)
47 IF (IREC(N).GE.IOVALUE(N)-IWINDOW(N)) GO TO 50
IF (KOUNT.NE.0) GO TO 49
TSWITCH1=1
CALL PRT1
DO 48 M4=1,16
CALL PRT2(M4,ITEHPM,LIST(M4),IOVALUE(M4),MRECNT,-1)
CONTINUE
48
49 CALL PRT2(N,N1,N2,IREC(N),MRECNT,0)
50 CONTINUE
C
170 C CALCULATION OF PITCH, ROLL, AND YAW 1-13 SAMPLES

```

TABLE A.3 (Continued)

PROGRAM COMB1

PROGRAM COMB1 74/74 OPT=1 TRACE ETN 4,7+76 16/23/79 16,16,56 PAGE 4

```

C
DO 52 N=1,3
  IREC(N+6+INDEX)=1.25*((LASTVAL(N*4-1)-IREC(N+43+INDEX))
    +8*IREC(N+27+INDEX)-LASTVAL(N*4))+128
175 1
52 CONTINUE
DO 54 N=1,3
  IREC(N+22+INDEX)=1.25*((LASTVAL(N*4)-IREC(N+59+INDEX))
    +8*IREC(N+43+INDEX)-IREC(N+11+INDEX))+128
180 1
54 CONTINUE
DO 56 N=6,8
  ITEMPL=LIST(N)+32+INDEX
  JTEMP=ITEMPL+160
DO 58 M=ITEMPL,JTEMP,16
  IREC(M)=1.25*((IREC(M-27)-IREC(M+37))+8*
    (IREC(M+21)-IREC(M-11))+128
185 1
58 CONTINUE
DO 70 M=1,16
  M1=LISTA(M1)
190 C
C CHECK MAX/MIN VALUES FOR ANY POSSIBLE
C CHANGE AND CHECK FOR WINDOW CROSSING
C
DO 69 M2=1,M1
  IF(M1.LT.6.OR.M1.GT.8)GO TO 61
  IF(1SWTCH5.EQ.0.AND.M2.LT.3)GO TO 69
  N2=LIST(M1)+(M2-1)*LISTB(M1)
  ISTO=N2+INDEX
  IF(M1.GT.5.AND.M1.LT.9)N2=N2+5
  IF(IREC(ISTO).GE.IBOUND(M1))GO TO 62
  IBOUND(M1)=IREC(ISTO)
  LFRAME(M1)=ITM
  LSAMPLE(M1)=N2
  IF(IREC(ISTO).LE.IBOUND(M1+16))GO TO 63
  IBOUND(M1+16)=IREC(ISTO)
  LFRAME(M1+16)=ITM
  LSAMPLE(M1+16)=N2
  IF(IREC(ISTO).LE.IOVALUE(M1)+IWINDOW(M1))GO TO 66
  IF(KOUNT.NE.0)GO TO 65
  ISWTCM1=1
  CALL PR1
  DO 64 M4=1,16
    CALL PR2(M4,ITEMPTH,LIST(M4),IOVALUE(M4),MRECN,-1)
  CONTINUE
215 64
65 CALL PR2(M1,ITM,N2,IREC(ISTO),MRECN,1)
66 IF(IREC(ISTO).GE.IOVALUE(M1)-IWINDOW(M1))GO TO 69
  IF(KOUNT.NE.0)GO TO 68
  ISWTCM1=1
  CALL PR1
  DO 67 M4=1,16
    CALL PR2(M4,ITEMPTH,LIST(M4),IOVALUE(M4),MRECN,-1)
  CONTINUE
220 67
68 CALL PR2(M1,ITM,N2,IREC(ISTO),MRECN,0)
  IF(IREC(ISTO).LT.IOVALUE(M1)-IWINDOW(M1))
    CALL PR2(M1,ITM,N2,IREC(ISTO),MRECN,0)
225 1
69 CONTINUE
70 CONTINUE
N=0

```

TABLE A.3 (Continued)

PROGRAM COMPB1

PROGRAM COMPB1 74/74 OPT=1 TRACE

11/13/79 17.37.24

PAGE 5

```

230      DO 75 M=187,235,16
           N=N+1
           DO 75 M1=1,3
               LASTVAL(N+4*(M1-1))=I*EC(M+M1*INDEX)
           75 CONTINUE
               ISWCH5=1
               GO TO 90
           235      IF (INFLIGHT.NE.32) CALL FRI1
               ISWCH5=0
               CONTINUE
               GO TO 100
           240      IERRCNT=IERRCNT+2
               ISWCH5=0
               GO TO 100
           92      ISHTCNT=ISHTCNT+2
               ISWCH5=1
               GO TO 100
           245      ISWCH5=1
               CONTINUE
               GO TO 20
           C
           C ANY INCOMPLETE TYPE2 DATA? WRITE LAST TYPE2 RECORD.
           C
           900 ISWCH3=1
               CALL PRT3
               WRITE(6,1010)
           1010 FORMAT(32H1PRINTOUT FOR COMPB1 STATISTICS: )
               WRITE(6,1050)KNTR1
           1050 FORMAT(41H01. THE NUMBER OF TYPE1 RECORDS WRITTEN ,I7)
               WRITE(6,1060)KNTR2
           1060 FORMAT(41H02. THE NUMBER OF TYPE2 RECORDS WRITTEN ,I7)
               WRITE(6,1070)KNTR3
           1070 FORMAT(41H03. THE NUMBER OF TYPE3 RECORDS WRITTEN ,I7)
               WRITE(6,1074)IISDCNT
           1074 FORMAT(28H04. SECONDS OF GROUND DATA,I7)
               WRITE(6,1076)KTKNT
           1076 FORMAT(26H05. TOTAL ACTIVE SAMPLES,I7)
               WRITE(6,1080)MPEAK
           1080 FORMAT(35H06. THE NUMBER OF PEAKS STORED ,I7)
               WRITE(6,1100)NTKNT
           1100 FORMAT(35H07. THE TOTAL NUMBER OF SECONDS ,I8)
               STOP
               END

```

PROGRAM COMP1

[illegible]

TABLE A.3 (Continued)

PROGRAM COMPB1

FIN 4.7.476

10/23/79

16.16.56

PAGE

2

SUBROUTINE PR11

74/74 OPT=1 TRACE

IF (M+1.EQ.98.OR.M+N.GT.128) GO TO 140

IREC01(INCR)=IREC(M+N*INDEX)

INCR=INCR+1

140 CONTINUE

IREC01(16)=0

C

C

C

C ***WINDOW VALUES***

IREC01(17)=IWINDOW(1)

IREC01(18)=IWINDOW(2)

IREC01(19)=IWINDOW(3)

IREC01(20)=IWINDOW(4)

IREC01(21)=IWINDOW(5)

IREC01(22)=IWINDOW(6)

IREC01(23)=IWINDOW(7)

IREC01(24)=0

IREC01(25)=IWINDOW(8)

IREC01(26)=IWINDOW(11)

IREC01(27)=IWINDOW(12)

IREC01(28)=IWINDOW(10)

IREC01(29)=IWINDOW(14)

IREC01(30)=IWINDOW(13)

IREC01(31)=IWINDOW(9)

IREC01(32)=0

IK=IJ=1

DO 155 M=1,16

IP1(M)=0

155 CONTINUE

CALL PACK(IREC01,72,IP1,IJ,IK,ITBL1)

ISWCH1=2

KNTR1=KNTR1+1

C WRITE(7)(IP1(J),J=1,16)

RETURN

END

TABLE A.3 (Continued)

PROGRAM COMB1

SUBROUTINE PRI2 74774 OPT=1 TRACE FIN 4.7+476 10/23/79 16.16.56 PAGE 1

```

1  SUBROUTINE PRI2(IIDENT,NFRAME,NTIME,JVALUE,KRECPN,IN)
   DIMENSION IPACK(96),IP2(16)
   COMMON/CT/IREC2(96),IDIR(16),IOVALUE(16)
   COMMON/CB/IBOUND(32),LFRAME(32),LSAMPLE(32)
   COMMON/CNTR/KOUNT,MPEAK,NKOUNT,KNTR1,KNTR2,KNTR3
   DATA IPACK/4,3*12,23*(8,16,8,8)/
   IP=IB=1
   IF(IH.EQ.-1)GO TO 200
   IF(IH.EQ.IDIR(IIDENT))GO TO 200

10  C
   C STORAGE OF PEAKS IN THE TYPE2 RECORDS.
   C IDENTIFICATION, SECONDS, PARAMETER POSITION
   C AND PEAK VALUE.
   C

15  IREC2(4*(KRECPN)+1)=IIDENT
   IREC2(4*(KRECPN)+2)=LFRAME(IDIR(IIDENT)*16+IIDENT)
   IREC2(4*(KRECPN)+3)=LSAMPLE(IDIR(IIDENT)*16+IIDENT)
   IREC2(4*(KRECPN)+4)=IBOUND(IDIR(IIDENT)*16+IIDENT)
   IDIR(IIDENT)=IN
   KOUNT=KOUNT+1
   MPEAK=MPEAK+1
   IF(KRECPN.LT.23)GO TO 300
   IREC2(1)=2
   IREC2(12)=IREC2(13)=IREC2(4)=0
   DO 50 M=1,16
     IP2(M)=0
50  CONTINUE
   CALL PACK(IREC2,96,IP2,IP,IB,IPACK)
   WRITE(7)(IP2(I),I=1,16)
30  KNTR2=KNTR2+1
   KRECPN=0
300 KRECPN=KRECPN+1
   C
   C STORAGE OF WINDOW CROSSINGS.
   C IDENTIFICATION, SECONDS, PARAMETER POSITION
   C AND WINDOW CROSSING VALUES.
   C

200 IREC2(4*(KRECPN)+1)=IIDENT
   IREC2(4*(KRECPN)+2)=NFRAME
   IREC2(4*(KRECPN)+3)=NTIME
   IREC2(4*(KRECPN)+4)=JVALUE
   LFRAME(IIDENT)=LFRAME(IIDENT+16)=NFRAME
   LSAMPLE(IIDENT)=LSAMPLE(IIDENT+16)=NTIME
   IBOUND(IIDENT)=IBOUND(IIDENT+16)=JVALUE
   IOVALUE(IIDENT)=JVALUE
   KOUNT=KOUNT+1
   IF(KRECPN.LT.23)GO TO 100
   IP=IB=1
   IREC2(1)=2
   IREC2(2)=IREC2(3)=IREC2(4)=0
   DO 150 M=1,16
     IP2(M)=0
150  CONTINUE
   CALL PACK(IREC2,96,IP2,IP,IB,IPACK)
   WRITE(7)(IP2(I),I=1,16)
55  KNTR2=KNTR2+1
   KRECPN=0

```

TABLE A.3 (Continued)

PROGRAM COMPB1

SUBROUTINE PR12 74/74 OPT=1 TRACE FIN 4,7+476 10/23/79 16,16,56 PAGE 2

100 KREGPN=KREGPN+1

RETURN
END

60

TABLE A.3 (Continued)

SUBROUTINE PRT3 7/7/74 OPT=1 TRACE 11/13/79 17.37.24 PAGE 1

PROGRAM COMB1

```

1  SUBROUTINE PRT3
   DIMENSION IP3(16),IREC03(17),IIBL3(17)
   COMMON INDEX,IREC(486)
   COMMON/O7/IREC02(96),IDIR,IOWALU(16)
   COMMON/CH3/ IAC,MRECNT,INCRST1
   COMMON/CH4/ IIM
   COMMON/CNTR/ KOUNT,HPEAK,MKOUNT,KNTR1,KNTR2,KNTR3
   COMMON/CNTR2/ IKOUNT,IGRCNT,IERRCNT,ISHICNT
   COMMON/CNTR/ KIKNT,NIKNT,KINTR2
   COMMON/CNTR2/ IKNT,IIGDCNT,IERRCNT,IISICNT
   COMMON/SWICH/ ISWICH1,ISWICH2,ISWICH3,IEND
   DATA IIBL3/4,-24,52,12*8,4,60/

C  ***END OF FLIGHT (TAIL)***
C  *****SWICH3 DEFINITION*****
15 C  ISWICH3=0 WHEN INITIAL ENTRY SJOR DDC DATA.
C  ISWICH3=1 WHEN ITS TO SAVE TIME SEC. AND WRITE TYPE 3 REC.
C  *****
C  IF (ISWICH3.EQ.1)GO TO 11
20 C  IREC03(1)=1
C  IREC03(2)=IAC
C  IREC03(3)=0
C  ***DOCUMENTARY DATA***
C  INCR=4
C  DO 10 N=63,127,16
C  DO 10 N=1,3
C  IF (N+4.EQ.90.OR.M+H.GT.128)GO TO 19
C  IREC03(INCR)=IREC(N+INCR)
C  INCR=INCR+1
C  10 CONTINUE
C  ISWICH3=1
C  RETURN
35 C  11 IREC03(16)=0
C  IREC03(17)=IIM
C  IEND=1
C  ISWICH3=0
C  IF (KOUNT.EQ.0)GO TO 30
C  40 C  CLEAR OUT REMAINING TYPE2 RECORDS
C  C BEFORE WRITING OUT THE TYPE3 RECORD.
C
C  IF (MRECNT.EQ.1)GO TO 18
C  IF (MRECNT.EQ.23)GO TO 15
C  DO 12 K7=MRECNT,23
C  IREC02(4*(K7)+1)=IPREC02(4*(K7)+2)=0
C  IREC02(4*(K7)+3)=IPREC02(4*(K7)+4)=0
C  12 CONTINUE
C  MRECNT=23
C  15 CALL PRI2(1,0,0,0,MRECNT,-1)
C  KOUNT=KOUNT-1
C  16 KNTR3=KNTR3+1
C  IK=IK+1
C  DO 24 N=1,16
C  IP3(N)=0
C  20 CONTINUE
C  CALL PACK(IREC03,17,IP3,IJ,IK,IIBL3)

```

TABLE A.3 (Concluded)

PROGRAM COMPB1

SUBROUTINE PR13 7474 OPI=1 TRACE FIN 4.7476 10/23/79 16.16.56 PAGE 2

```

C      WRITE(7) (IP3(J), J=1, 16)
      WRITE(6, 1010) KNTR1
1010  FORMAT(8H0FLIGHT#, I3, 11H STATISTICS)
      WRITE(6, 1020) KNTR2
1020  FORMAT(41H 1. THE NUMBER OF TYPE2 RECORDS WRITTEN:, I7)
      WRITE(6, 1060) IGRDCNT
1060  FORMAT(28H 2. SECONDS OF GROUND DATA:, I7)
      WRITE(6, 1090) KOUNT
1090  FORMAT(28H 3. SAMPLES OF ACTIVE DATA:, I7)
      WRITE(6, 1110) ITH
1110  FORMAT(29H 4. TOTAL NUMBER OF SECONDS:, I7)
      30 NKOUNT=NKOUNT-ITH
      KNTR2=KNTR2+KNTR2
      IITERCNT=IERCNT+IITERCNT
      IISHTCNT=ISHTCNT+IISHTCNT
      IIGDCNT=IGDCNT+IIGDCNT
      ITKNT=IKOUNT+ITKNT
      KTKNT=KOUNT+KTKNT
      ITHNT=ITH+ITKNT
      KNTR2=IERCNT=ISHTCNT=IGDCNT=0
      IKOUNT=KOUNT=LKOUNT=0
      RETURN
      END

```

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525

TABLE A.4 (Continued)

PROGRAM COMPB2

PROGRAM COMPB2 7/74 OPT=1 TRACE

10/19/79 17.23.46

PAGE 2

```

DO 100 K=1,5
CALL UNPACK(IREC,486,NISLK,IN,JH,ITBL)
IF(ISWITCH4.EQ.0)GO TO 93

C
C PARITY ERROR CHECK
C
60      IF(OR(SHIFT(IREC(483),8),IREC(484)) NE.0) GO TO 91

C
C SHORT RECORD CHECK
C
65      IF(AND(IREC(480),15).NE.9) GO TO 92
INDEX=0
DO 90 L=1,2
IF(L.EQ.2) INDEX=240

C
C BUILT IN CYCLE TEST..
C
75      JBIT=IREC(INDEX+232)/128
JCAL=AND((IREC(INDEX+232)/64),1)
JPRE=AND((IREC(INDEX+232)/32),1)
IF(JBIT.EQ.1.AND.INFLIGHT.NE.64.AND.IEND.NE.1)CALL PRT3
IF(JCAL.EQ.1.OR.JBIT.EQ.1.OR.JPRE.EQ.1)IKOUNT=IKOUNT+1
IF(JCAL.EQ.1.OR.JPRE.EQ.1)GO TO 87
IF(JBIT.EQ.1)GO TO 86
ITH=IKOUNT-10+2*(K-1)+L
IEND=0
INFLIGHT=AND(IREC(64+INDEX),64)

85      C FLIGHT CHECK
C

90      IF(INFLIGHT.EQ.0)IGRODNT=IGRODNT+1
IF(INGR.EQ.1)GO TO 26
IF(INFLIGHT.EQ.0)GO TO 87

C
C DONE ONCE AFTER BIT CYCLE HAS BEEN SHUT DOWN.
C
26      IF(ISWITCH1.NE.0)GO TO 30
ISWITCH1=2
DO 29 N=1,21
IF(N.LT.5.OR.N.GT.6)GO TO 27
N2=LIST(N-2)
ISTO=N2+INDEX
IOVALUE(N)=128
GO TO 28

27      N2=LIST(N)
ISTO=N2+INDEX
IOVALUE(N)=IREC(ISTO)
IBOUND(N)=IBOUND(N+21)=IOVALUE(N)
LFRAME(N)=LFRAME(N+21)=ITH
LSAMPLE(N)=LSAMPLE(N+21)=ISTO
ITEMPT=ITH

29      CONTINUE

C
C ANALYSIS OF ALL 21 PARAMETERS OF THE SECOND.
C
C 1. CALCULATE THE PITCH AND YAW SLOPE VALUES OF 19-20
C 2. ANALYZE THE MAX/MIN CHECKS AND THE MINIMUM CROSSINGS OF 19-20
C 3. CALCULATE THE PITCH AND YAW SLOPE VALUES OF SAMPLES 1-18 OF THE

```

TABLE A.4 (Continued)

PROGRAM COMPB2

PROGRAM COMPB2 74/74 CNT=1 TRACE

FTN 4,7+7,7

11/19/79 1 23:40

PAGE 3

```

115      C      PRESENT SECOND.
120      C      4. ANALYZE THE MAX/MIN CHECKS AND ALL WINDOW CROSSING OF ALL
      C      PARAMETERS.
      C      30      IF (ISWCH5.EQ.0) GO TO 45

125      C      DEFINITION OF ISWCH5
      C      C      ISWCH5=0 WHEN NO PREVIOUS LASTVAL VALUES HAVE BEEN STORED.
      C      C      ISWCH5=1 WHEN THERE ARE PREVIOUS VALUES FOR THE CALCULATION
      C      C      OF THE PITCH AND YAW SLOPE VALUES OF THE 19TH AND 20TH
      C      C      SAMPLE VALUES OF THE PREVIOUS SECOND AND 1ST AND 2ND
      C      C      VALUES OF THE PRESENT SECOND.
      C
      C      32      N=1,2
      C      100 (N*20-1)=1.6667*((LASTVAL(N*4-3))-IREC(N+INDEX))
      C      C      +8*((LASTVAL(N*4)-LASTVAL(N*4-2)))+128
      C      32      CONTINUE
      C      33      N=1,2
      C      100 (N*20)=1.6667*((LASTVAL(N*4-2))-IREC(N+12+INDEX))
      C      C      +8*((IREC(N+INDEX)-LASTVAL(N*4-1)))+128
      C      33      CONTINUE
      C      N2=1TH-1
      C      43      N=18,38,20
      C      N1=4
      C      43      M=1,2
      C      N1=N1+1
      C      IF (IDOT(N*M).GE.180000(N1)) GO TO 34
      C      180000(N1)=IDOT(N*M)
      C      LFRAME(N1)=N2
      C      LSAMPLE(N1)=LIST(N1-2)+(M-1)*12+216
      C      IF (IDOT(N*M).LE.180000(N1+21)) GO TO 35
      C      180000(N1+21)=IDOT(N*M)
      C      LFRAME(N1+21)=N2
      C      LSAMPLE(N1+21)=LIST(N1-2)+(M-1)*12+216
      C      N3=LIST(N1-2)+(M-1)*12+216
      C      IF (IDOT(N*M).LE.180000(N1)+180000(N1)) GO TO 38
      C      IF (KOUNT.NE.0) GO TO 37
      C      ISWCH1=1
      C      CALL PR11
      C      36      M=1,21
      C      CALL PR12(M,1,TEMPH,L1ST(N4),IOVALUE(M4),MRECHT,-1
      C      1
      C      CONTINUE
      C      36      CALL PR12(N1,N2,N3,IDOT(N*M),MRECHT,1)
      C      37      IF (IDOT(N*M).GE.180000(N1)-180000(N1)) GO TO 43
      C      38      IF (KOUNT.NE.0) GO TO 41
      C      ISWCH1=1
      C      CALL PR11
      C      39      M=1,21
      C      CALL PR12(M,1,TEMPH,L1ST(N4),IOVALUE(M4),MRECHT,-1
      C      1
      C      CONTINUE
      C      39      CALL PR12(N1,N2,N3,IDOT(N*M),MRECHT,0)
      C      41      CONTINUE
      C      43      N=1,2
      C      100 (N*20-19)=1.6667*((LASTVAL(N*4-1))-IREC(N+24+
      C      C      INDEX))+8*((IREC(N+12+INDEX)-LASTVAL(N*4)))+128

```

TABLE A.4 (Continued)

PROGRAM COMPB2

PAGE

10/19/79 1.23.40

FIN 7.74.0

74/74 OPT=1 TRACE

PROGRAM COMPB2

```

44      CONTINUE
45      DO 45 N=1,2
175      IDOT(N*20-18)=1.6667*((LASTVAL(N*4)-IREC(N*36+INDEX
      ))+8*(IREC(N*24+INDEX)-IREC(N+INDEX)))+128
45      CONTINUE
46      DO 46 N=3,10
      N2=2
      DO 46 M=1,2
      N2=N2+1
180      IDOT(N+20*(N-1))=1.6667*((IREC(LIST(N2)+(N-3)*12
      )-IREC(LIST(N2)+(N+1)*12))+8*(IREC(LIST(N2)+N
      +12)-IREC(LIST(N2)+(N-2)*12)))+128
48      CONTINUE
49      DO 49 M=1,21
      N1=LISTA(M1)
      IF(M1.GT.4.AND.M1.LT.7)GO TO 59
      DO 50 M2=1,N1
      N2=LIST(M1)+(M2-1)*LISTB(M1)
      ISTO=N2+INDEX
190      IF(IREC(ISTO).GE.IBOUND(M1))GO TO 50
      IBOUND(M1)=IREC(ISTO)
      LFRAME(M1)=ITM
      LSAMPLE(M1)=N2
195      IF(IREC(ISTO).LE.IBOUND(M1+21))GO TO 51
      IBOUND(M1+21)=IREC(ISTO)
      LFRAME(M1+21)=ITM
      LSAMPLE(M1+21)=N2
51      IF(IREC(ISTO).LE.IOVALUE(M1)+IWINDOW(M1))GO TO 54
      IF(KOUNT.NE.0)GO TO 53
      ISWITCH=1
      CALL PR1
      DO 52 M4=1,21
      CALL PR12(M4,ITM,N2,IREC(ISTO),MRECNT,1)
      IF(IREC(ISTO).GE.IOVALUE(M1)-IWINDOW(M1))GO TO 58
      IF(KOUNT.NE.0)GO TO 56
      ISWITCH=1
      CALL PR1
      DO 55 M4=1,21
      CALL PR12(M4,ITEMPTH,LIST(M4),IOVALUE(M4),MRECNT,-1
      )
      CONTINUE
52      CALL PR12(M1,ITM,N2,IREC(ISTO),MRECNT,1)
53      IF(IREC(ISTO).GE.IOVALUE(M1)-IWINDOW(M1))GO TO 58
54      IF(KOUNT.NE.0)GO TO 56
      ISWITCH=1
      CALL PR1
      DO 55 M4=1,21
      CALL PR12(M4,ITEMPTH,LIST(M4),IOVALUE(M4),MRECNT,-1
      )
      CONTINUE
55      CALL PR12(M1,ITM,N2,IREC(ISTO),MRECNT,0)
56      CONTINUE
58      GO TO 69
59      DO 60 M2=1,18
      IF(ISWITCH5.EQ.0.AND.M2.LT.3)GO TO 68
      N2=N2+20*(M1-5)
      ISTO=LIST(M1-2)+(M2-1)*LISTB(M1-2)
      IF(IDCT(N2).GE.IBOUND(M1))GO TO 63
      IBOUND(M1)=IDCT(N2)
      LFRAME(M1)=ITM
      LSAMPLE(M1)=ISTO
225      IF(IDOT(N2).LE.IBOUND(M1+21))GO TO 61
      IBOUND(M1+21)=IDCT(N2)

```

TABLE A.4 (Continued)

PROGRAM COMPB2

PROGRAM COMPB2 74/74 OPT=1 TRACE

11/13/79 17.17.43

PAGE 5

```

230      LFRAME(M1+21)=ITM
        LSAMPLE(M1+21)=ISTO
        IF(IDOT(N2).LE.IOVALUE(M1))GO TO 64
        IF(KOUNT.NE.0)GO TO 63
        ISWCH1=1
        CALL PRT1
235      DO 62 M4=1,21
        CALL PRT2(M4,ITEMPTM,LIST(M4),IOVALUE(M4),MRECN, -1
1
        CONTINUE
        CALL PRT2(M1,ITM,ISTO,IDOT(N2),MRECN,1)
240      IF(IDOT(N2).GE.IOVALUE(M1))GO TO 68
        IF(KOUNT.NE.0)GO TO 66
        ISWCH1=1
        CALL PRT1
        DO 65 M4=1,21
        CALL PRT2(M4,ITEMPTM,LIST(M4),IOVALUE(M4),MRECN, -1
1
        CONTINUE
        CALL PRT2(M1,ITM,ISTO,IDOT(N2),MRECN,0)
250      CONTINUE
        N=0
        DO 70 M=192,220,12
        N=N+1
        DO 70 M1=1,2
        LASTVAL(N+M*(M1-1))=IREC(N+M1*INDEX)
255      CONTINUE
        ISWCH5=1
        GO TO 90
260      GO TO 90
        IF(INFLIGHT.NE.60)CALL PRT1
        ISWCH5=0
        GO TO 90
265      GO TO 100
        ISWCH1=ITEMCNT+2
        ISWCH5=0
        GO TO 100
        ISWCH1=ITEMCNT+2
        ISWCH5=0
        GO TO 100
        ISWCH4=1
        GO TO 20
270      GO TO 20
        C
        C ANY INCOMPLETE TYPE3 DATA? WRITE LAST TAIL RECORD.
        C
275      900 IF(ISWCH4.EQ.0)GO TO 21
        ISWCH3=1
        CALL PRT3
        WRITE(6,1010)
        1010 FORMAT(32HPRINTOUT FOR COMPB2 STATISTICS: )
        WRITE(6,1050)KNIR1
        1050 FORMAT(41H1. THE NUMBER OF TYPE1 RECORDS WRITTEN: ,I7)
        WRITE(6,1060)KNIR2
        1060 FORMAT(41H02. THE NUMBER OF TYPE2 RECORDS WRITTEN: ,I7)
        WRITE(6,1070)KNIR3
        1070 FORMAT(41H03. THE NUMBER OF TYPE3 RECORDS WRITTEN: ,I7)

```

TABLE A.4 (Continued)

PROGRAM COMPB2

PAGE 6

11/13/79 17.11.43

FIN 4.7+476

PROGRAM COMPB2 74/74 OPT=1 TRACE

```

WRITE(6,1074)ITGOCNT
1074 FORMAT(20H04. SECONDS OF GROUND DATA,I7)
WRITE(6,1076)KTKNT
1076 FORMAT(26H05. TOTAL ACTIVE SAMPLES,I7)
WRITE(6,1080)MPEAK
1080 FORMAT(35H06. THE NUMBER OF PEAKS STORED,I7)
WRITE(6,1090)NTKNT
1090 FORMAT(35H07. THE TOTAL NUMBER OF SECONDS,I7)
STOP
END
295

```

FILED 4.7.47b

74/74 CPT=1 TRACE

SUBROUTINE PRT1

[illegible]

TABLE A.4 (Continued)

PROGRAM COMPB2

PAGE 2

11/19/9 1.23.10

FIN ***

SUBROUTINE PRT1 7474 OPT=1 TRACE

```

C
60      ISWCH3=0
        CALL PRT3
        ISWCH3=1
        CO 137 M=INCR,103
            IPECO1(M)=0
137      CONTINUE
        IRECO1(1)=1
        IRECO1(2)=IAC
        IRECO1(3)=0
C
C ***** DOCUMENTARY DATA *****
70      C
        CO 120 M=1,12
            IRECO1(M+3)=IREC(INCD(M)+INDEX)
120      CONTINUE
        IRECO1(16)=0
75      C
C ***** WINDGWS *****
C *****NY=S1 C130***
C
        M1=17
80      CO 124 M=1,19
            IF(MOD(M1,8).NE.0)GO TO 123
            IRECO1(M1)=0
            M1=M1+1
123      IRECO1(M1)=IWINDOW(INDW(M))
            M1=M1+1
124      CONTINUE
        CO 125 M=30,39
            IRECO1(M)=0
125      CONTINUE
        IK=IJ=1
        CO 130 M=1,16
            IPI(M)=0
130      CONTINUE
        CALL PACK(IRECO1,103,IPI,IJ,IK,ITBL1)
        ISWCH1=2
95      KNTRI=KNTRI+1
        C      WRITE(7)(IPI(J),J=1,16)
        RETURN
        END

```

TABLE A.4 (Continued)

11/19/9 1.23.49

11/19/9 1.23.49

PROGRAM COMPB2

11/19/9 1.23.49

11/19/9 1.23.49

11/19/9 1.23.49

```

SUBROUTINE PRT2      74/74      CPI=1 TRACE      PROGRAM COMPB2
1  SUBROUTINE PRT2(IDENT,NFRAME,NTIME,JVALUE,KRECPN,IH)
   DIMENSION IPACK(96),IP2(16)
   COMMON/C7/IREC02(96),IDIR(21),IOVALUE(21)
   COMMON/C8/IBOUND(42),LFRAME(42),LSAMPLE(42)
   COMMON/CNTR/KOUNT,MPEAK,NKOUNT,KNTR1,KNTR2,KNTR3
   DATA IPACK/4,3*12,23*8,16,8,8/
   IP=IB=1
   IF(IH.EQ.-1)GO TO 200
   IF(IH.EQ.IDIR(IDENT))GO TO 200
10  C STORAGE OF PEAKS IN THE TYPE2 RECORDS
   C IDENTIFICATION
   C SECOND OF FLIGHT
   C PARAHUTER POSITION
15  C PEAK VALUE
   C
   IREC02(4*(KRECPN)+1)=IDENT
   IREC02(4*(KRECPN)+2)=LFRAME(IDIR(IDENT)+21+IDENT)
   IREC02(4*(KRECPN)+3)=LSAMPLE(IDIR(IDENT)+21+IDENT)
   IREC02(4*(KRECPN)+4)=IBOUND(IDIR(IDENT)+21+IDENT)
   IDIR(IDENT)=IH
   KOUNT=KOUNT+1
   MPEAK=MPEAK+1
   IF(KRECPN.LT.23)GO TO 300
   IREC02(1)=2
   IREC02(2)=IREC02(3)+IREC02(4)=0
   TO 50 M=1,16
   IP2(M)=0
50  CONTINUE
30  CALL PACK(IREC02,96,IP2,IP,IB,IPACK)
   C WRITE(7)(IP2(I),I=1,16)
   KNTR2=KNTR2+1
   KRECPN=0
300  KRECPN=KRECPN+1
   C
35  C STORAGE OF WINDOW CROSSINGS IN THE TYPE2 RECORDS
   C IDENTIFICATION
   C SECOND OF FLIGHT
   C PARAHUTER POSITION
40  C WINDOW CROSSING VALUE(NEW MEAN VALUE)
   C
200  IREC02(4*(KRECPN)+1)=IDENT
   IREC02(4*(KRECPN)+2)=NFRAME
   IREC02(4*(KRECPN)+3)=NTIME
   IREC02(4*(KRECPN)+4)=JVALUE
   IOVALUE(IDENT)=JVALUE
   LFRAME(IDENT)=LFRAME(IDENT+21)=NFRAME
   LSAMPLE(IDENT)=LSAMPLE(IDENT+21)=NTIME
   IBOUND(IDENT)=IBOUND(IDENT+21)=JVALUE
   KOUNT=KOUNT+1
   IF(KRECPN.LT.23)GO TO 100
   IP=IB=1
   IREC02(1)=2
   IREC02(2)=IREC02(3)+IREC02(4)=0
   DO 150 M=1,16
   IP2(M)=0
150  CONTINUE

```

TABLE A.4 (Continued)

PROGRAM COMPB2

SUBROUTINE PR12 7/4/74 OPT=1 TRACE 10/19/79 1.23 40 PAGE 2

CALL PACK(IREC02,96,IP2,IP,IB,IPACK)
 WRITE(7) (IP2(I),I=1,16)
 KNTR2=KNTR2+1
 KRECPN=0
 100 KRECPN=KRECPN+1
 RETURN
 END

PROGRAM COMPB2

```

1 SUBROUTINE PR13
   DIMENSION IP3(17),IREC03(17),ITBL3(17),INOD(12)
   COMMON INDEX,IREC(486)
5   COMMON /C7/ IREC02(96),IDIR(21),IOVALUE(21)
   COMMON /CN3/ IAC,MRENT
   COMMON /CH4/ ITH
   COMMON /CNTR/ KOUNT,HPEAK,NKOUNI,KNTR1,KNTR2,KNTR3
   COMMON /ICNTR2/ IKOUNT,IGRCNT,IERRCNT,ISHTCNT
   COMMON /ICNTR/ KIKNT,NIKNTR,KTNIR2
10  COMMON /ICNTR2/ IVKNT,ITGCR2,IFERCNT,IISTCNT
   COMMON /SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
   DATA ITBL3/4,-24,52,12,84,60/
   DATA INDD/48,88,95,96,136,143,144,184,191,192,232,239/

      C ***** END OF FLIGHT (TAIL: *****
15  C ***** SWITCH3 DEFINITION *****
      C SWITCH3=0 WHEN INITIAL ENTRY STORE DOC.DATA
      C SWITCH3=1 WHEN ITS TO SAVE TIME SEC. AND WRITE
      C TYPES RECORD
20  C ***** DOCUMENTARY DATA *****
      C IF(ISWICH3.EQ.1)GO TO 11
      IREC03(1)=3
      IREC03(2)=IAC
25  IREC03(3)=IAC
      C
      C DO 10 M=1,12
        IREC03(M+2)=IREC(INDD(M)+INDEX)
30  10 CONTINUE
        ISWICH3=1
        RETURN
        11 IREC03(16)=0
35  IREC03(17)=ITH
        IEND=1
        ISWICH3=0
        IF(KOUNT.EQ.0)GO TO 30
      C
40  C CLEAR OUT THE REMAINING TYPE2 RECORDS BEFORE WRITING
      C OUT THE TAIL RECORD.
      C
        IF(MRENT.FQ.1)GO TO 18
        IF(MRENT.EQ.23)GO TO 15
45  GO 12 K7=MRENT,23
        IREC02(4*(K7)+1)=IREC02(4*(K7)+2)=0
        IREC02(4*(K7)+3)=IREC02(4*(K7)+4)=0
      12 CONTINUE
        MRENT=23
50  15 CALL PR12(1,0,0,0,MRENT,-1)
        KOUNT=KOUNT-1
        18 KNTR3=KNTR3+1
        IJ=1
        IK=1
55  DO 20 M=1,16
        IF3(M)=0
      20 CONTINUE

```

TABLE A.4 (Concluded)

PROGRAM COMPB2

Page 2

19/19/19 1-23

FTR

SUBROUTINE PR13 (A/74 CPT=1 TRACE

CALL PACK(IREFC03,17,IF3,IJ,IG,119L3)

WRITE(7) (IP3(J),J=1,16)

WRITE(6,1010)KNIR1

1010 FORMAT(8HFLIGHT#,I3,11H STATISTICS)

WRITE(6,1020)KNIR2

1020 FORMAT(41H 1. THE NUMBER OF TYPE2 RECORDS PRINTED,I7)

WRITE(6,1060)IGRODNT

1060 FORMAT(29H 2. SECONDS OF GROUND DATA,I7)

WRITE(6,1090)KOUNT

1090 FORMAT(29H 3. SAMPLES OF ACTIVE DATA,I7)

WRITE(6,1110)ITH

1110 FORMAT(29H 4. TOTAL NUMBER OF SECONDS,I7)

30 PKOUNT=PKOUNT-ITH

KNIR2=KNIR2+KNIR2

ITERCNT=IERRCNT+ITERCNT

IISHTCNT=ISHTCNT+IISHTCNT

ITGDCNT=IGRODNT+ITGDCNT

ITKNT=IKOUNT+ITKNT

KTKNT=KOUNT+KTKNT

NTKNT=ITH+NTKNT

KNIR2=IERRCNT=ISHTCNT=IGRODNT=0

IKOUNT=KOUNT=0

RETURN

END

TABLE A.5
SUBROUTINE PACK/UNPACK

1	SUBROUTINE UNPACK(UP,NUP,P,J,NB,ITBL) INTEGER P,UP DIMENSION UP(1),P(1),ITBL(1),MSK(60),MSKR(10) DATA IM/0/
5	DATA MSKR/0,55P,5555P,555555P,55555555P,555555555P,5555555555P, *55555555555555P,55555555555555P,55555555555555P,55555555555555P/ IF(IM.NE.0) GO TO 10 MSK(1)=COMPL(0) DO 5 I=2,60
10	5 MSK(I)=COMPL(MSK(I-1)) IM=1 10 CONTINUE DO 500 I=1,NUP JJ=ITBL(I)
15	IF(IARS(JJ).GT.(61-NB)) GO TO 50 IF(JJ) 20,500,15 15 UP(I)=AND(SHIFT(P(J),NB+JJ-1),MSK(61-JJ)) NB=NB+JJ GO TO 30
20	20 JJ=-JJ UP(I)=SHIFT(AND(SHIFT(P(J),NB+JJ-1),MSK(61-JJ)),60-JJ) NB=NB+JJ 25 IF(MOD(JJ,6).NE.0) GO TO 30 LL=11-JJ/6
25	UP(I)=OR(UP(I),MSKB(LL)) 30 IF(NB.LT.61) GO TO 500 NB=1 J=J+1 GO TO 500
30	50 IF(JJ.LT.0) GO TO 60 KK=61-NB LL=JJ-KK UP(I)=SHIFT(AND(MSK(NB),P(J)),LL) J=J+1
35	UP(I)=OR(UP(I),AND(SHIFT(P(J),LL),MSK(61-LL))) NR=1+LL GO TO 30 60 JJ=-JJ UP(I)=SHIFT(AND(MSK(NB),P(J)),NB-1) J=J+1
40	KK=JJ-61+NB UP(I)=OR(UP(I),SHIFT(AND(SHIFT(P(J),KK),MSK(61-KK)),60-JJ)) NR=1+KK GO TO 25
45	500 CONTINUE RETURN END

TABLE A.5 (Concluded)
SUBROUTINES PACK/UNPACK

1	SUBROUTINE PACK(UP,NUP,P,J,NB,ITBL)
	C UP IS UNPACKED ARRAY
	C NUP IS NUMBER OF WORDS FROM UP TO BE PACKED
	C P IS PACKED ARRAY
5	C J IS FIRST AVAILABLE WORD IN P ARRAY
	C NB IS FIRST AVAILABLE BIT IN JTH WORD OF P ARRAY (NB GE 1 OR LE 60)
	C ITBL IS TABLE OF LENGTH NUP CONTAINING NUMBER OF BITS FROM
	C EACH VALUE IN UP TO PACKED IN P
10	INTEGER P,UP
	DIMENSION UP(1),P(1),ITBL(1),MSK(60)
	DATA IP/0/
	IF (IP.EQ.1) GO TO 10
	MSK(1)=COMPL(0)
	DO 5 I=2,60
15	F MSK(I)=COMPL(MSK(I-1))
	IF=1
	10 CONTINUE
	DO 500 I=1,NUP
	JJ=ITBL(I)
20	IF (IABS(JJ).GT.(61-NB)) GO TO 50
	IF (JJ) 20,500,15
	15 P(J)=OR(P(J),SHIFT(AND(UP(I),MSK(61-JJ)),61-NB-JJ))
	NB=NBJJ
	GO TO 30
25	20 JJ=-JJ
	P(J)=OR(P(J),SHIFT(AND(SHIFT(UP(I),JJ),MSK(61-JJ)),61-NB-JJ))
	NB=NBJJ
	30 IF (NB.LT.61) GO TO 500
	NB=1
30	J=J+1
	GO TO 500
	50 IF (JJ.LT.0) GO TO 60
	KK=121-JJ-NB
	P(J)=OR(P(J),AND(SHIFT(UP(I),KK),MSK(NB)))
35	J=J+1
	P(J)=OR(P(J),SHIFT(AND(UP(I),MSK(KK+1)),KK))
	NB=61-KK
	GO TO 30
	60 JJ=-JJ
40	P(J)=OR(P(J),AND(MSK(NB),SHIFT(UP(I),61-NB)))
	J=J+1
	KK=121-JJ-NB
	P(J)=OR(P(J),SHIFT(AND(MSK(KK+1),SHIFT(UP(I),JJ)),KK))
	NB=61-KK
45	GO TO 30
	500 CONTINUE
	RETURN
	END

TABLE A.6
PROGRAMS COMPAL AND COMPA2 DEFINITION OF VARIABLES

IAC - aircraft title

IERRCNT - counter incremented when a parity error has been found

IGRDCNT - counter incremented if second of data was recorded on the ground

IKOUNT - counter incremented when active data is found

INDEX - used to determine first or second half of frame

INFLIGHT - flag for in-flight

IOVALT - old value of the altitude

IOVAS - old value of the air speed

IOVE1 - old value of the events of 1-4

IOVE5 - old value of the events of 5-9

IOVNY - old value of the NY parameter

IOVNY - old value of the NZ parameter

IOVQ - old value of the Q parameter

IOVR - old value of the R parameter

IQDOT - array used for the integrated q value

IRDOT - array used for the integrated r value

IREC - array used for 2 seconds of unpacked data

ISHTCNT - counter incremented when a short record has been found

ISWTCH1 - switch used in PRT1 routine
 = 0 when the routine has gone through previous calibration data
 = 1 when the routine is to write the record
 = 2 when the routine is to initialize INCR variable

ISWTCH2 - switch used in PRT3 routine
 = 0 when the routine is to store time and data
 = 1 when the routine is to store values of $\dot{p}$, $\dot{q}$, and $\dot{r}$ and write the record

ISWTCH3 - switch used in PRT4 routine
 = 0 when the routine is to store the documentary data
 = 1 when the routine is to write the record

ISWTCH4 - switch used in the MAIN routine
 = 0 when the routine is to skip over seconds read in
 = 1 when processing is finished upon reading a tape mark

ITALT - altitude threshold

ITAS - airspeed threshold

TABLE A.6 (Continued)

PROGRAMS COMPAL AND COMPA2 DEFINITION OF VARIABLES

ITBL - unpacking scheme for RT record
 ITERCNT - total parity errors counter
 ITGDCNT - total seconds of ground data
 ITKNT - total active data counter
 ITM - time parameter
 ITNY - N_y threshold
 ITNZ - N_z threshold
 ITQ - Q threshold
 ITR - R threshold
 ITSTCNT - total short records counter
 JBIT - flag turned on during a built in cycle test
 JCAL - flag turned on during a calibration cycle
 JPRE - flag indicating PRE bit is on
 KNTR1 - counter used in counting the type 1 records
 KNTR2 - counter used in counting the type 2 records
 KNTR3 - counter used in counting the type 3 records
 KNOUNT - counter used in counting the active data found
 KTKNT - total active data found
 KTNTR2 - total type 2 records written
 LAST - flag signifying if the previous second was active or inactive
 LKOUNT - counter used in counting the inactive data found
 LTKNT - total inactive seconds found
 NGR - switch indicating whether to process ground data or not
 KNOUNT - counter used in counting number of seconds read in
 NTBLK - array into which data is read from tape
 NTKNT - total number of seconds read in

Subroutine ACTDATA

INCR - variable used to index the LASTQ and LASTR arrays
 LASTQ - array used to store the last four pitch values
 LASTR - array used to store the last four yaw values

TABLE A.6 (Concluded)
PROGRAMS COMPAL AND COMPA2 DEFINITION OF VARIABLES

Subroutine PRT1

INCR - variable used to index the IRECO1 array during different calibration periods

INCRST1 - variable used to save INCR when passed back to the main routine

INDD - array containing the location of the documentary data in IREC

IP1 - array returned from the PACK routine in packed format

IRECO1 - array given to the PACK routine in unpacked format

ITBL1 - packing scheme for type 1 record

Subroutine PRT2

IP2 - array returned from the PACK routine in packed format

IRECO2 array given to the PACK routine in unpacked format

ITBL2 - packing scheme for type 2 record

Subroutine PRT3

IP3 - array returned from the PACK routine in packed format

IRECO3 - array given to the PACK routine in unpacked format

ITBL3 - packing scheme for type 3 record

TABLE A.7
PROGRAM COMPB1 AND COMPB2 DEFINITION OF VARIABLES

IAC - aircraft title

IBOUND - array containing minimums of each parameter in the first half, and maximums in second half

IDIR - previous direction of parameter (1 = up, 0 = down)

IDOT - array of $\dot{p}$, $\dot{q}$ and $\dot{r}$ values

IERRCNT - number of seconds containing parity errors

IGRDCNT - number of seconds of ground data

IKOUNT - number of BIT/PRE/CAL seconds

INDEX - used to determine first or second half of frame

INFLGHT - flag for in-flight

IOVALUE - array containing last window crossing

IREC - unpacked input data

ISHTCNT - number of short record seconds

ISWTCH1 - used in PRT1 routine (same as in program COMPAL)

ISWTCH3 - used in PRT3 routine (same as in program COMPAL)

ISWTCH4 - used in main routine (same as in program COMPAL)

ITBL - unpacking scheme for RT record

ITERCNT - total seconds with parity errors

ITGDCNT - total seconds of ground data

ITKNT - total of BIT/PRE/CAL seconds

ITSTCNT - total short record seconds

ITM - time into flight

IW - current parameter direction (see IDIR)

IWINDOW - window values

JBIT, JCAL, JPRE - flags for BIT, CAL, PRE seconds, respectively

KNTR1, KNTR2, KNTR3 - counters for type 1, 2 3 records, respectively

KTKNT - total samples of active data

KTNTR2 - total number of type 2 records

LFRAME - frames containing minimums or maximums accessed like IBOUND

LIST - location of parameters in IREC array

LSAMPLE - sample numbers for minimums and maximums assessed like IBOUND

IRECO2 - Type 2 buffer area

MPEAK - number of peaks stored

TABLE A.7 (Concluded)

PROGRAM COMPB1 AND COMPB2 DEFINITION OF VARIABLES

MRECNT - type 2 data pointer
NGR - flag for inflight
NKOUNT - number of seconds
NTBLK - input buffer for packed data

USED ONLY IN C141, C130:

LISTA - number of samples/sec for each parameter
LISTB - displacement factor for each parameter

Subroutine PRT1

INCR - index IRECO1 for calibration
INCRST1 - save area for INCR
INDD - documentary data
INDW - window indexing scheme
IP1 - packed type 1 record
IRECO1 - unpacked type 1 record
ITBL1 - type 1 packing scheme

Subroutine PRT2

IDENT - parameter identification
IPACK - type 2 packing scheme
JVALUE - window crossing
KRECPN - position within type 2 record
IP2 - output buffer for packed type 2 record
NFRAME - frame number of value
NTIME - time of value

Subroutine PRT3

IP3 - packed type 3 data
IRECO3 - unpacked type 3 data
ITBL3 - packing scheme for type 3 records

MAIN PROGRAM

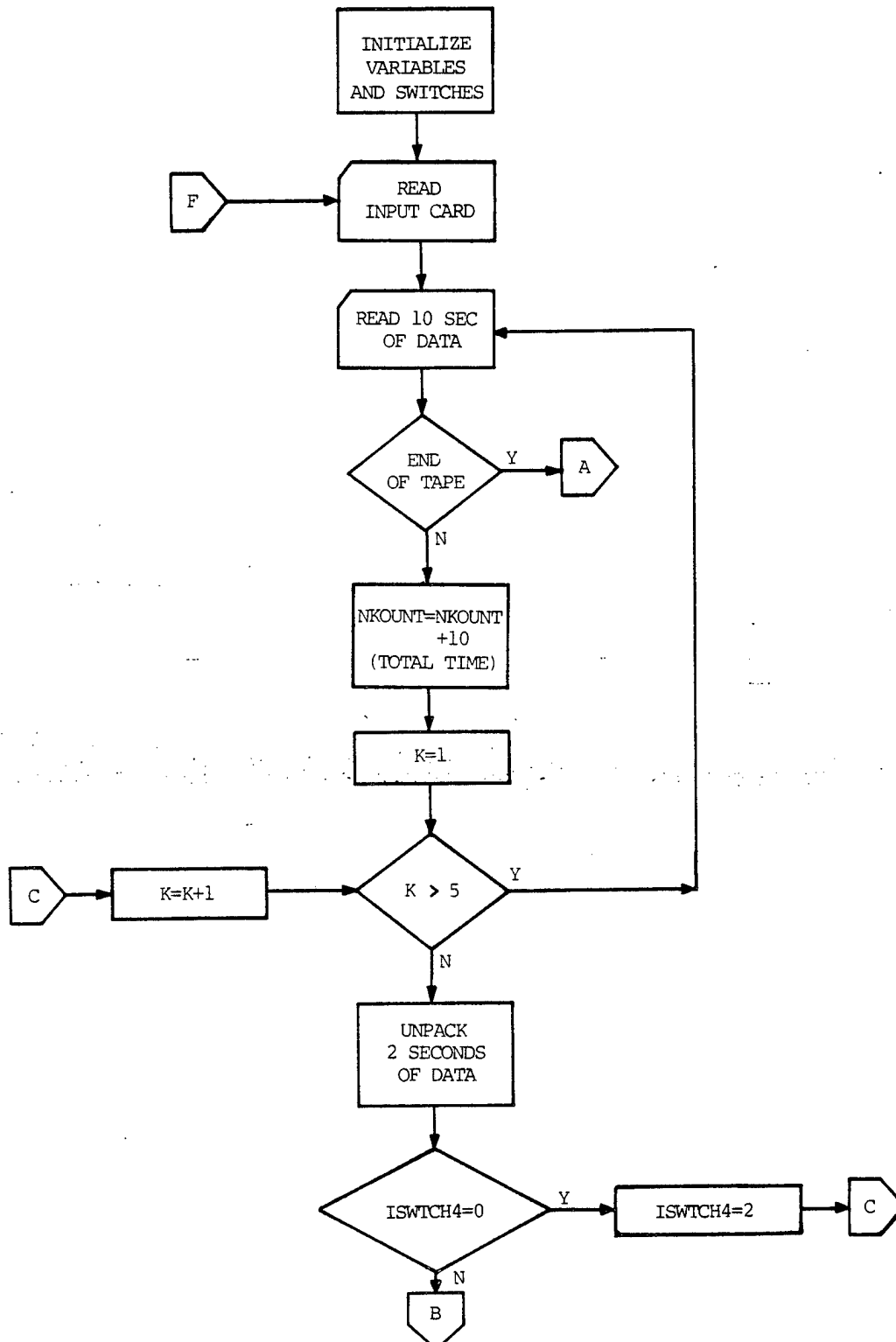


Figure A.1. Flowchart of Method A.

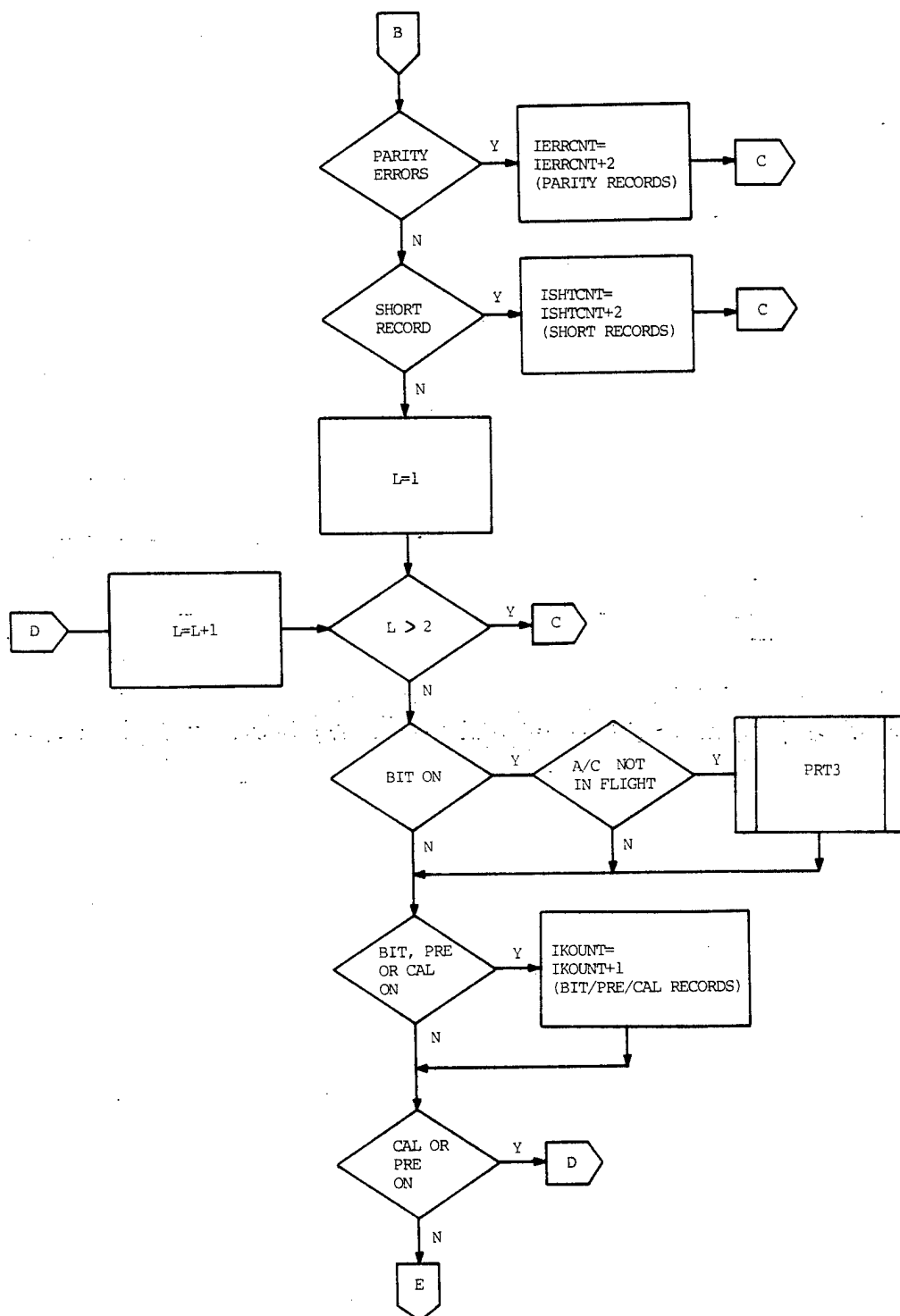


Figure A.1. (Continued)

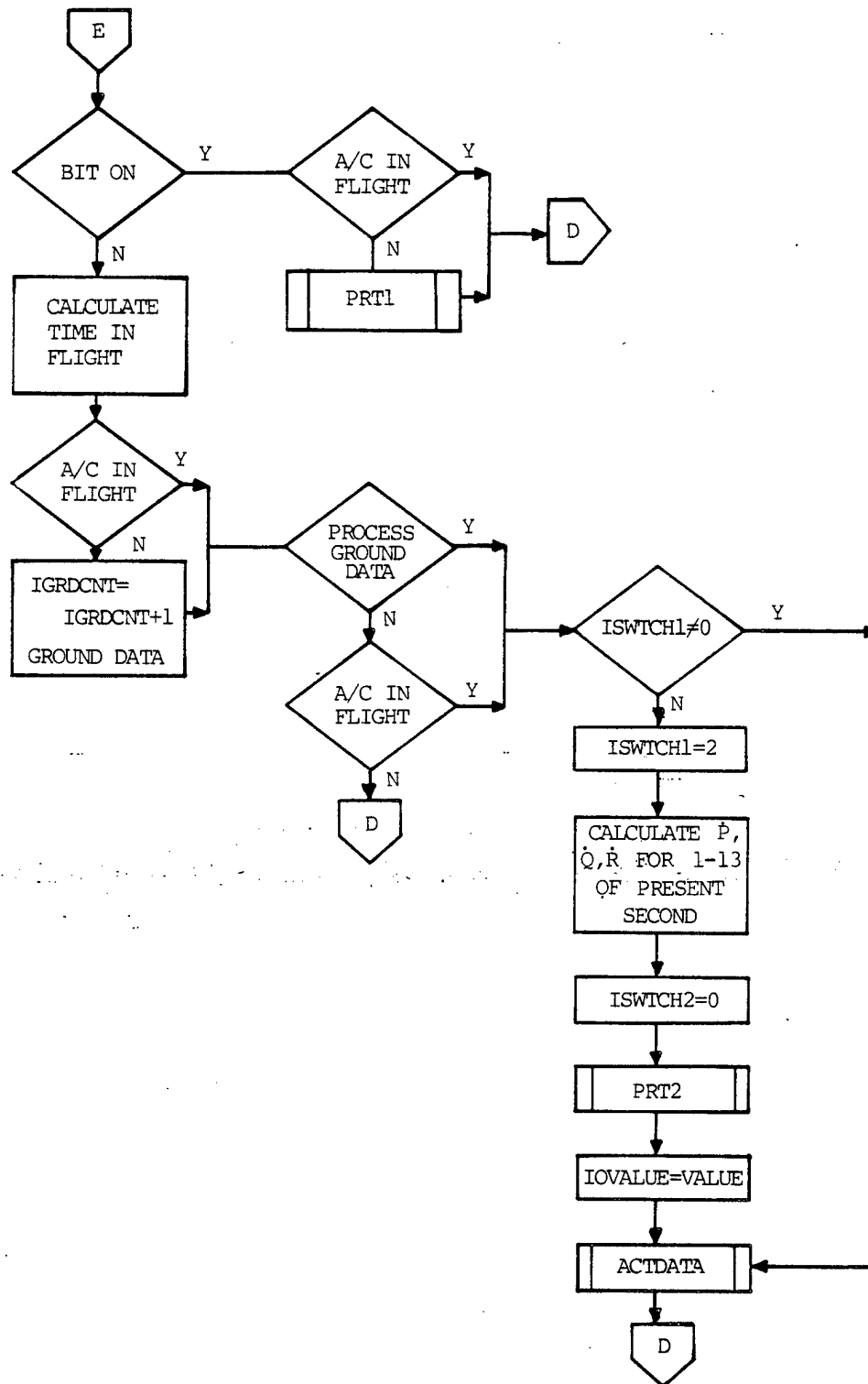


Figure A.1. (Continued)

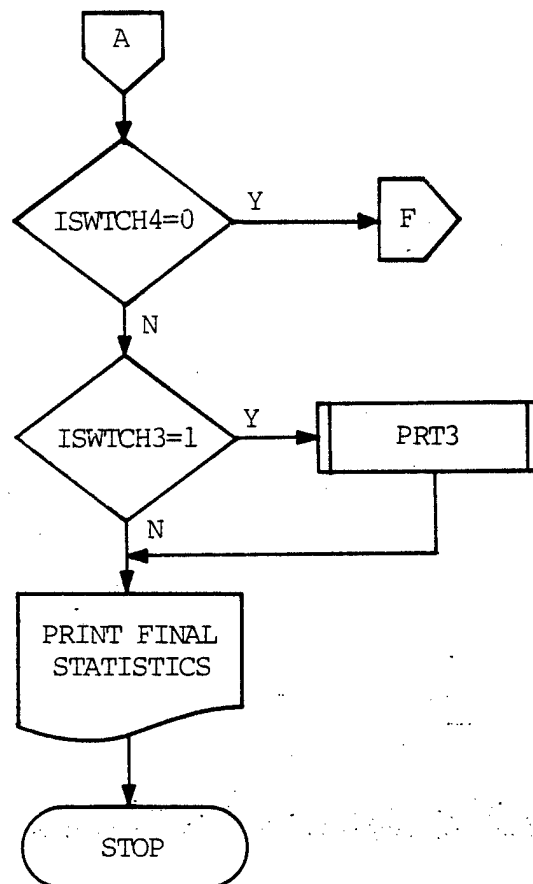


Figure A.1. (Continued)

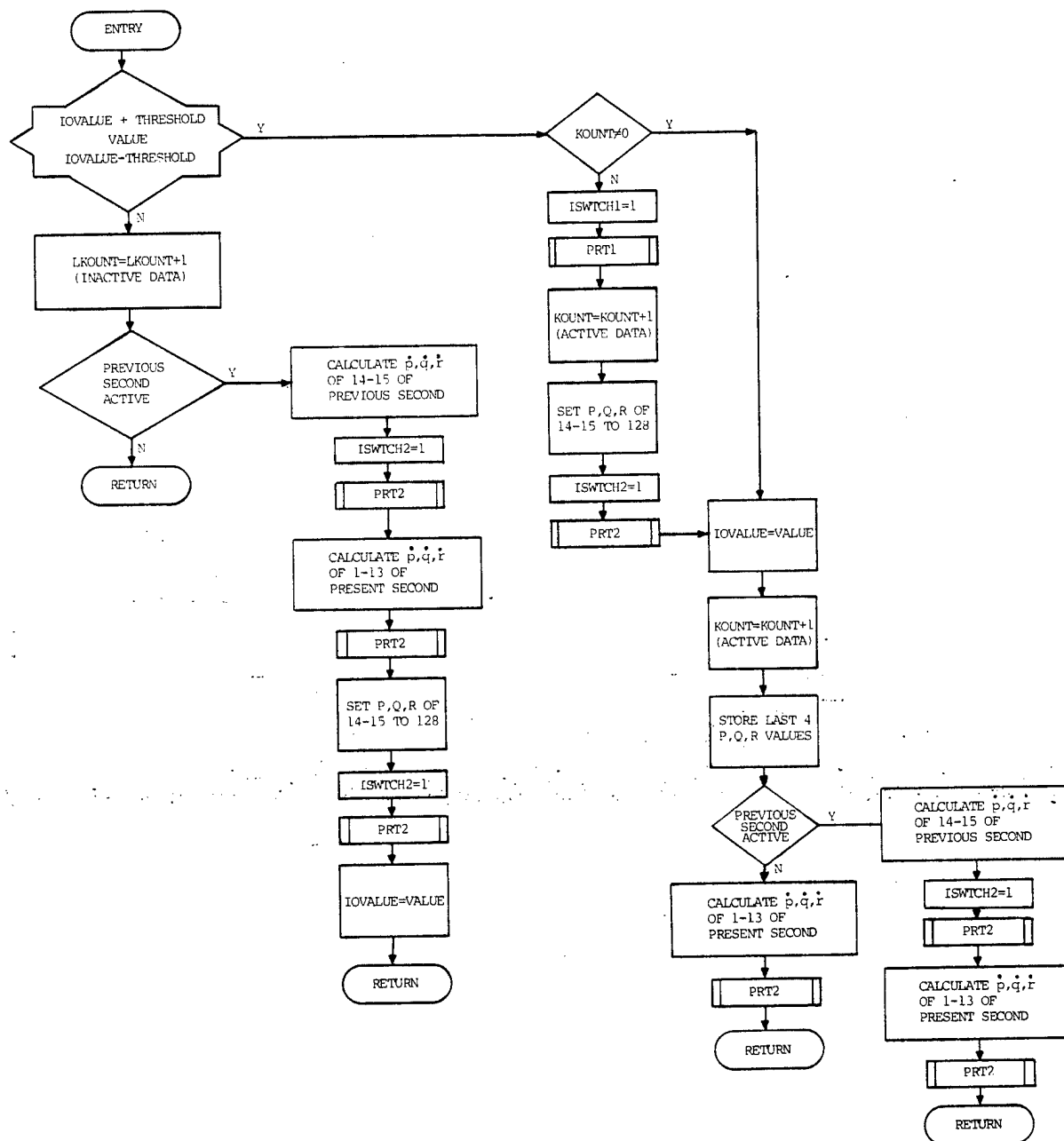


Figure A.1. (Continued)

SUBROUTINE PRT1

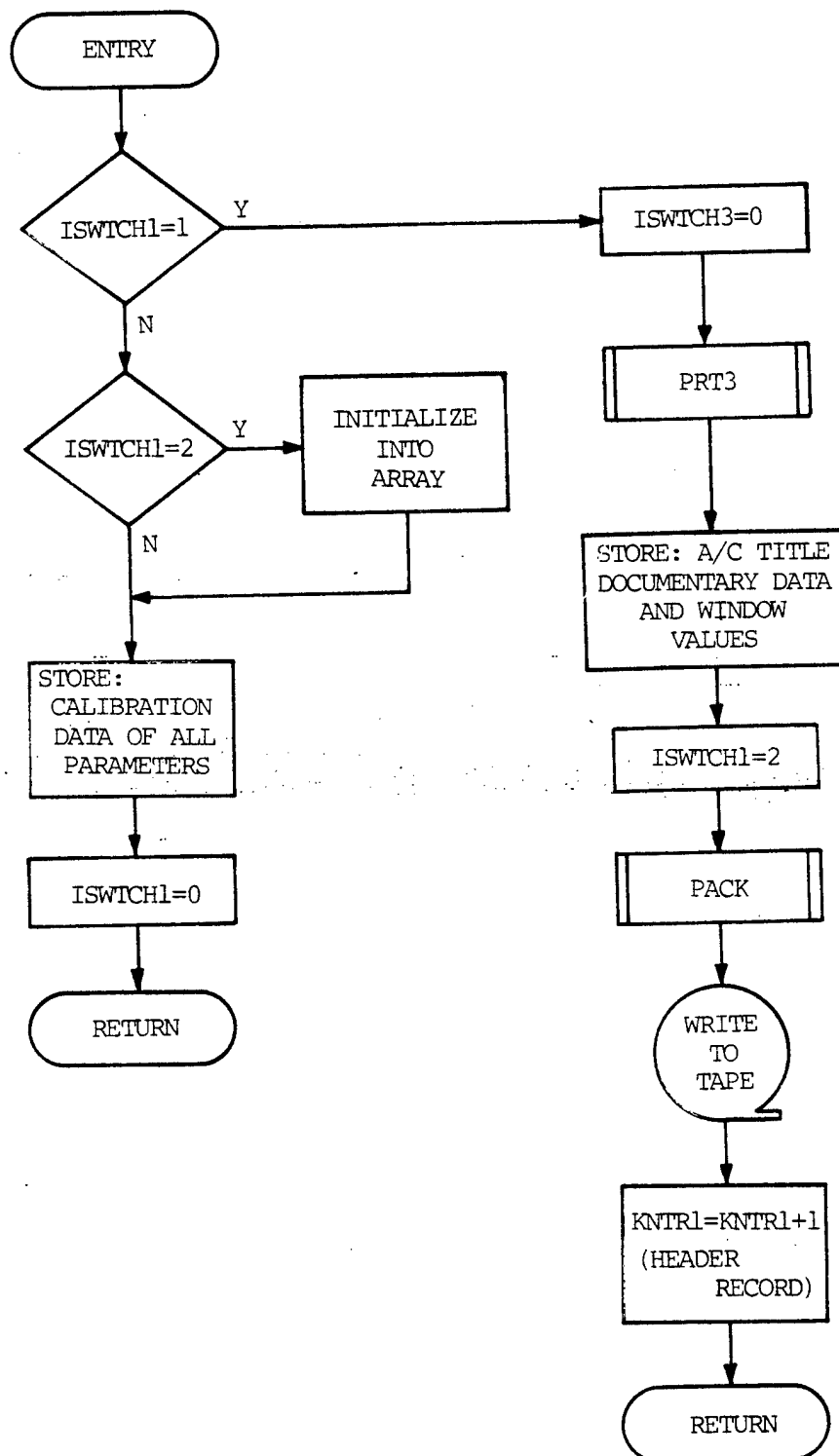


Figure A.1. (Continued)

SUBROUTINE PRT2

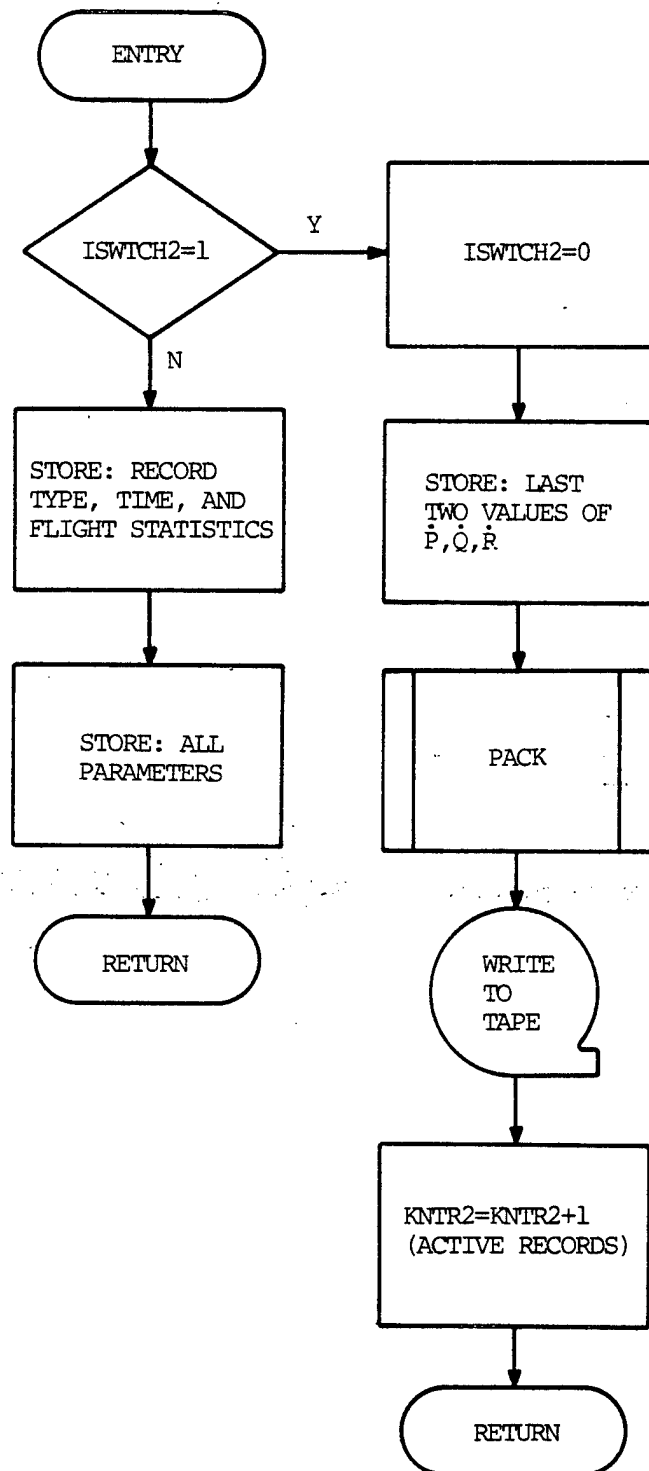


Figure A.1. (Continued).

SUBROUTINE PRT3

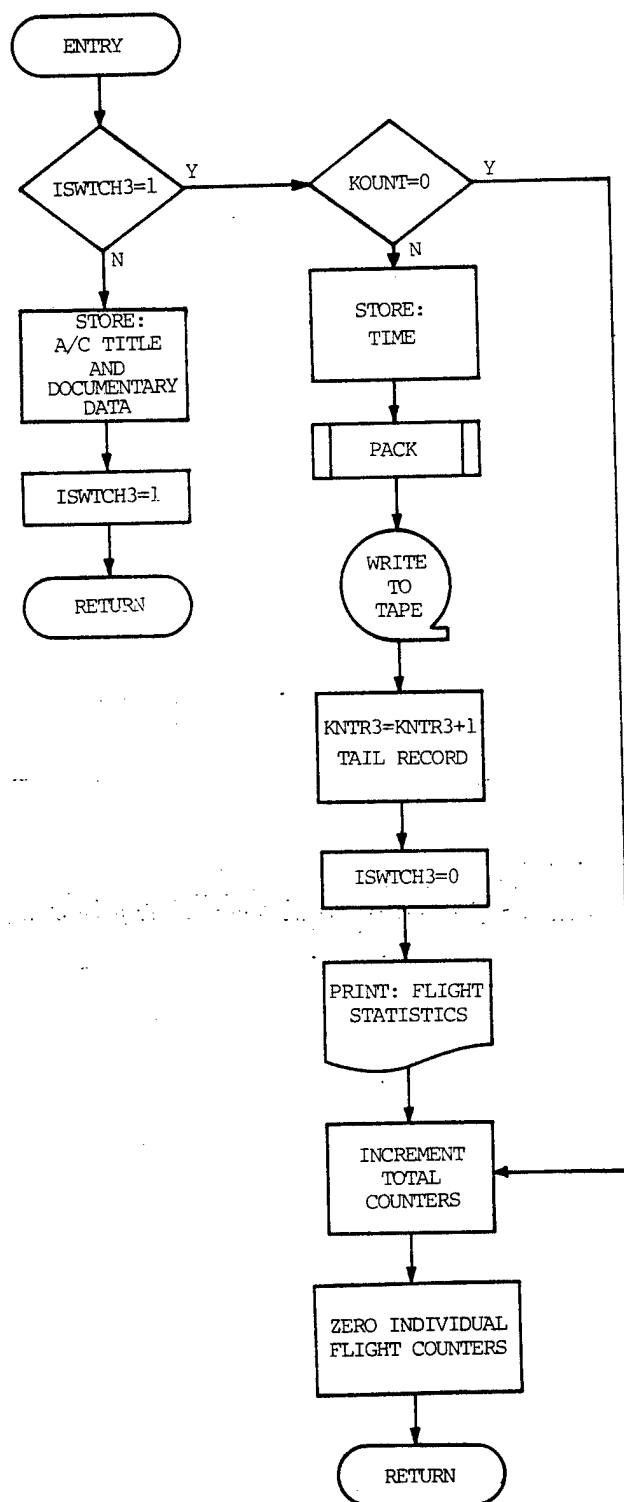


Figure A.1. (Concluded).

MAIN PROGRAM

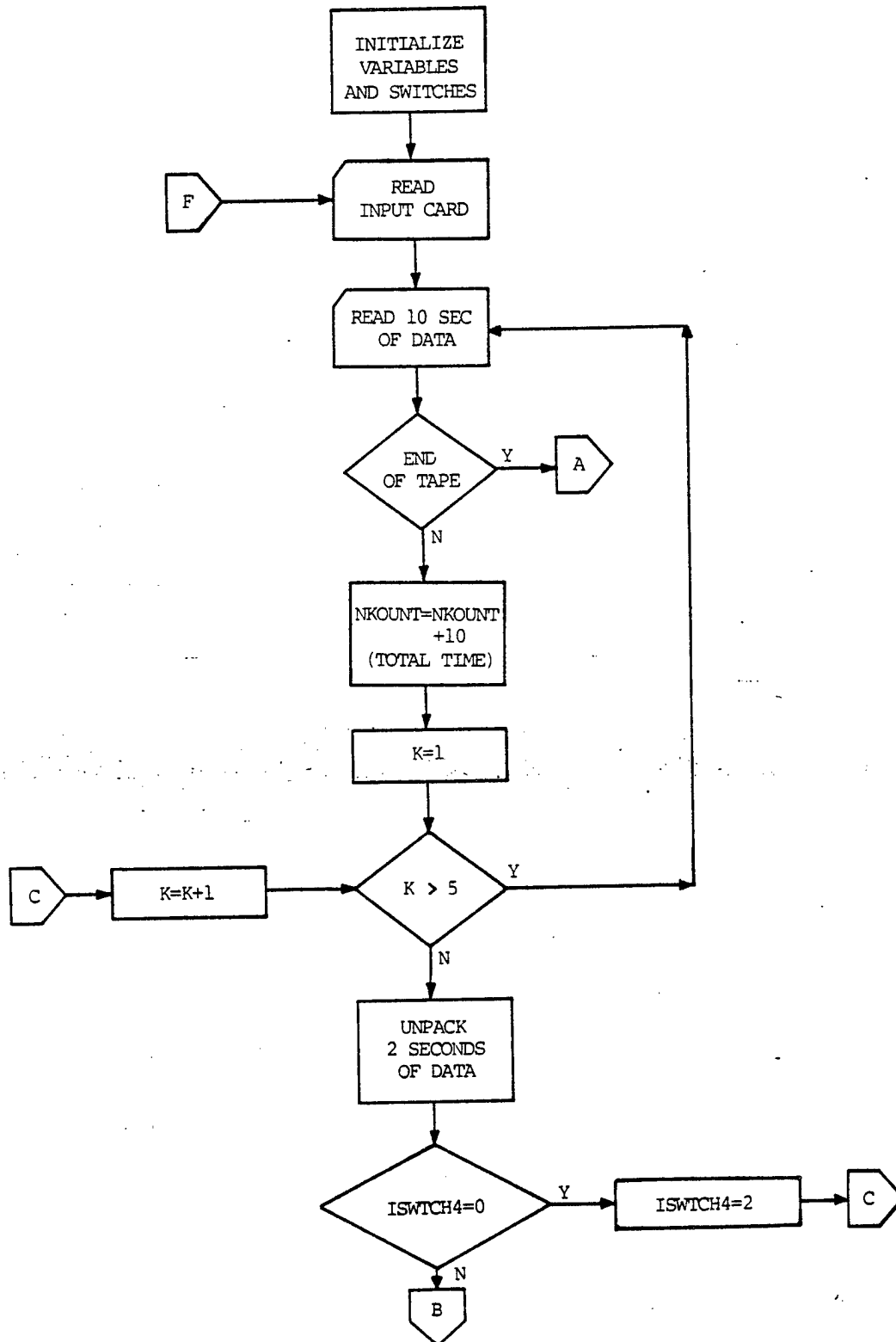


Figure A.2. Flowchart of Method B.

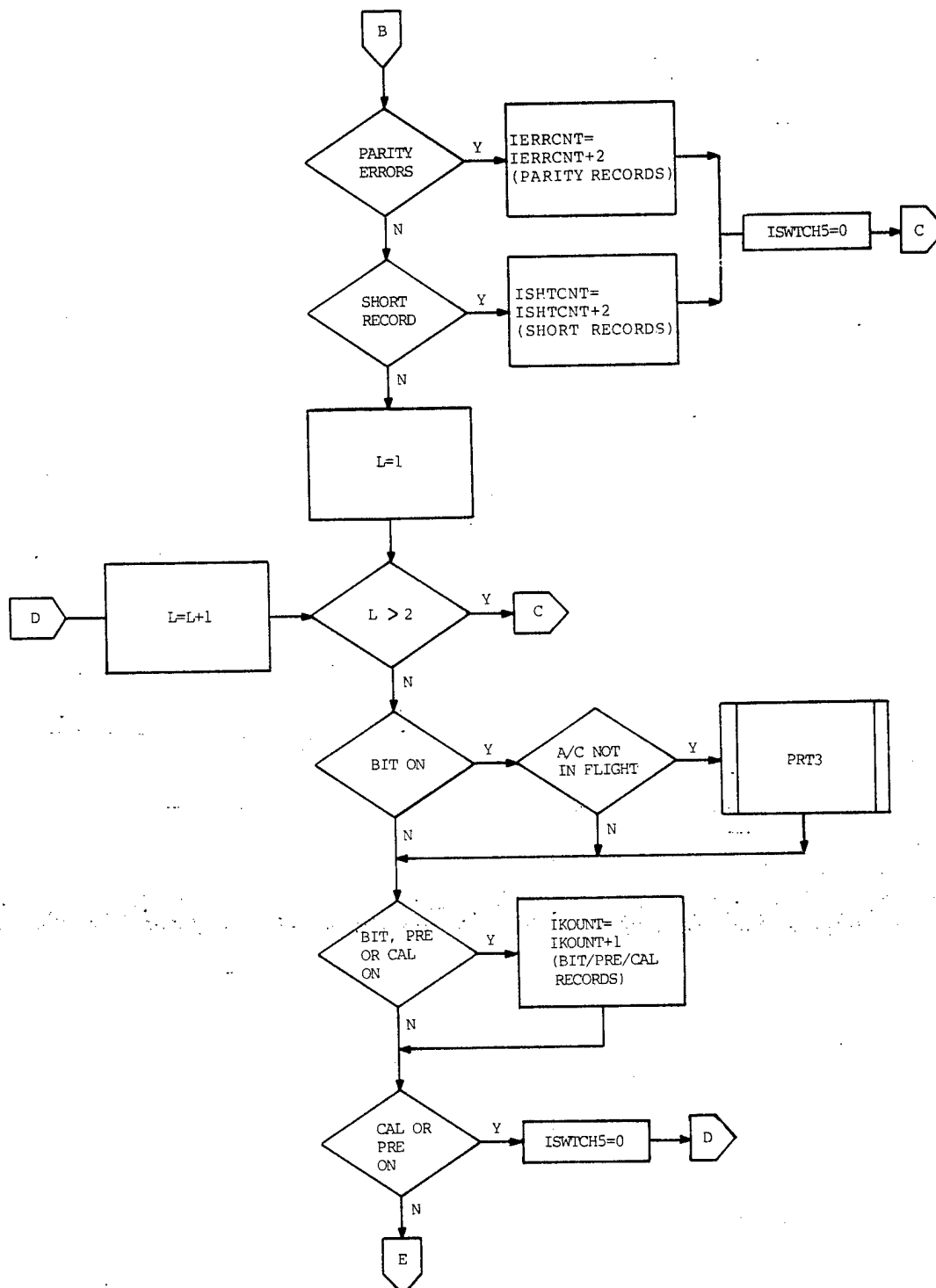


Figure A.2. (Continued).

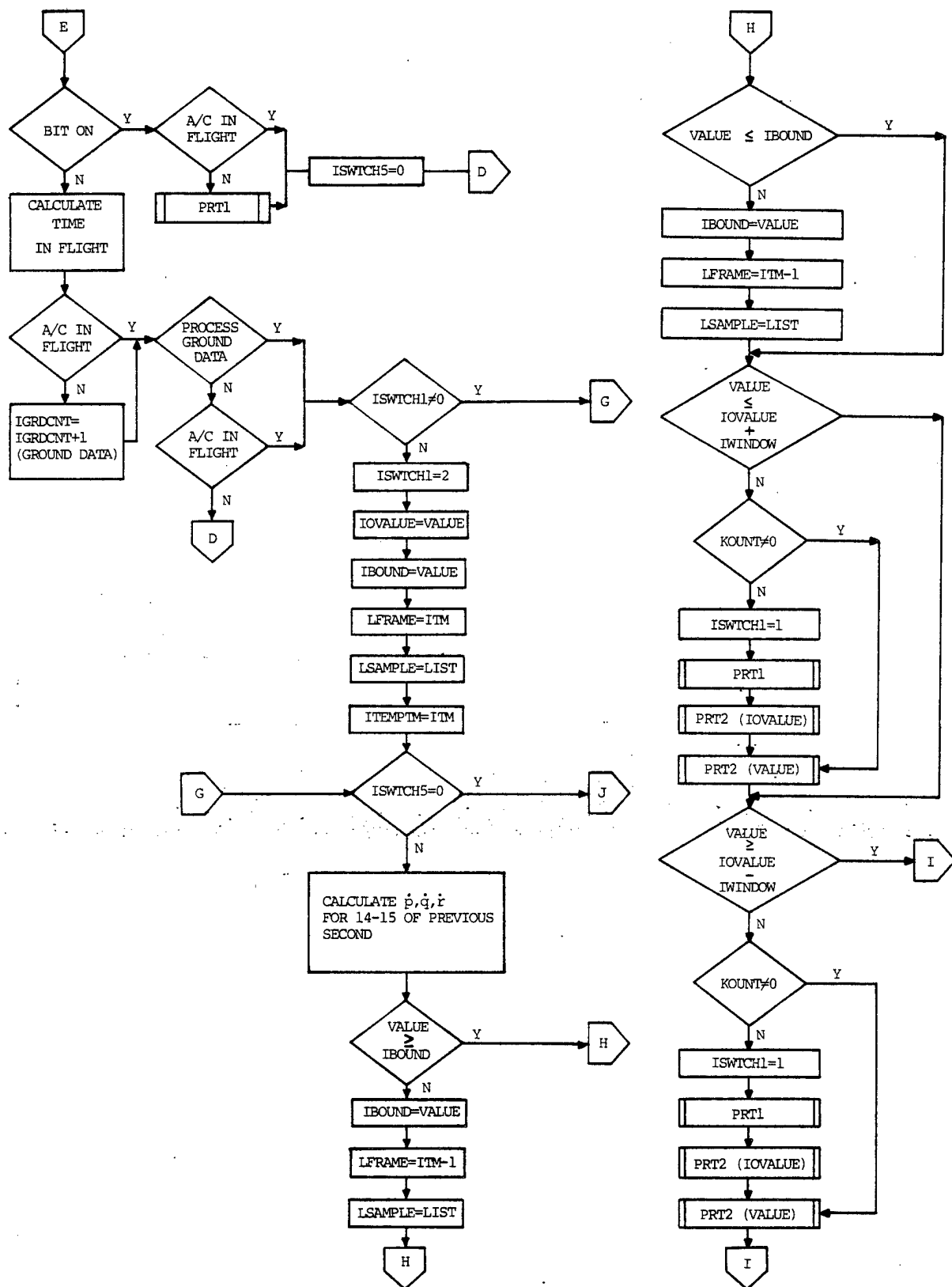


Figure A.2. (Continued).

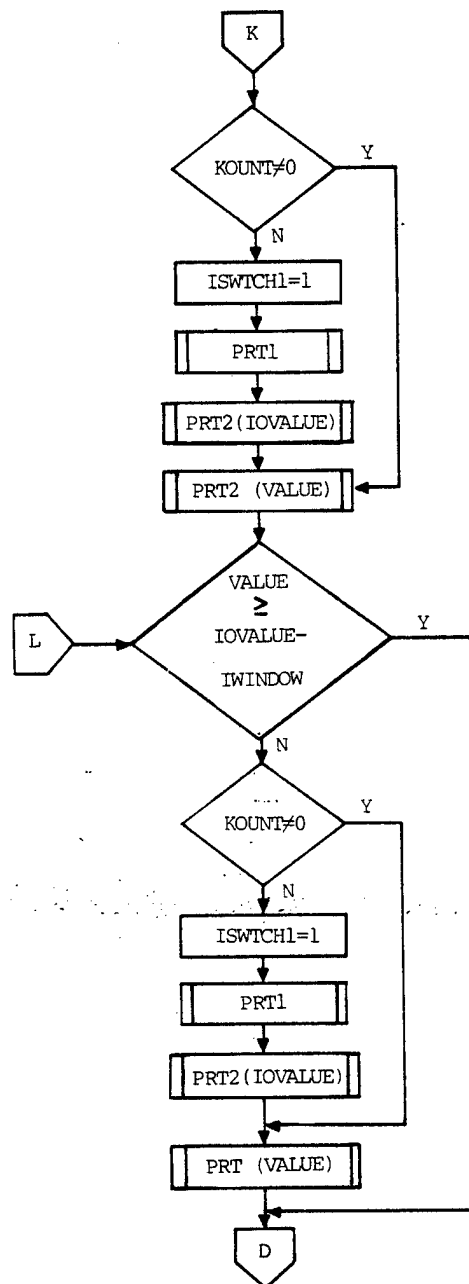
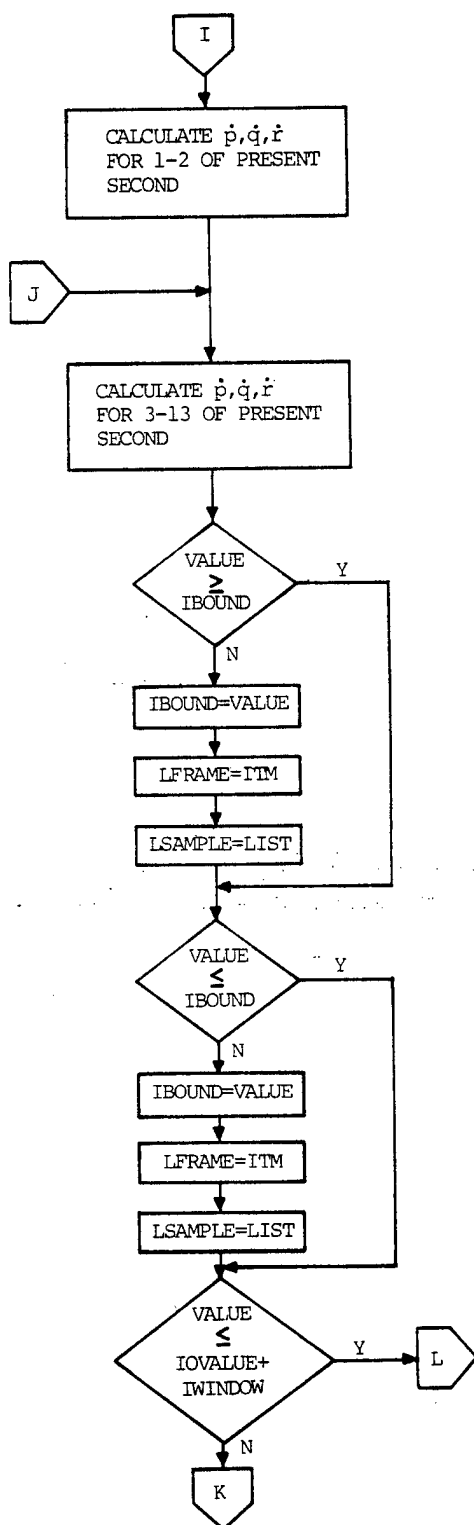


Figure A.2 (Continued).

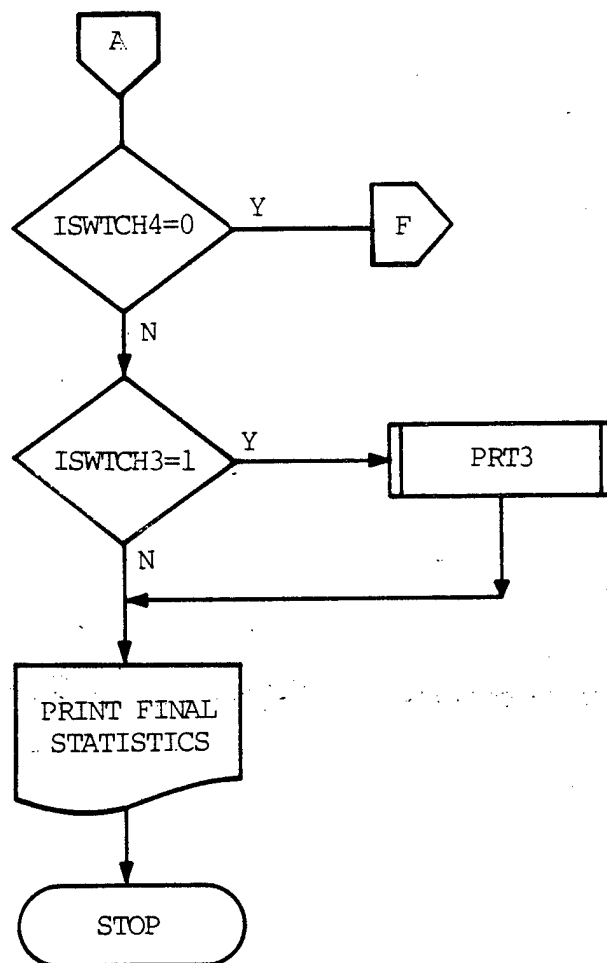


Figure A.2. (Continued).

SUBROUTINE PRT1

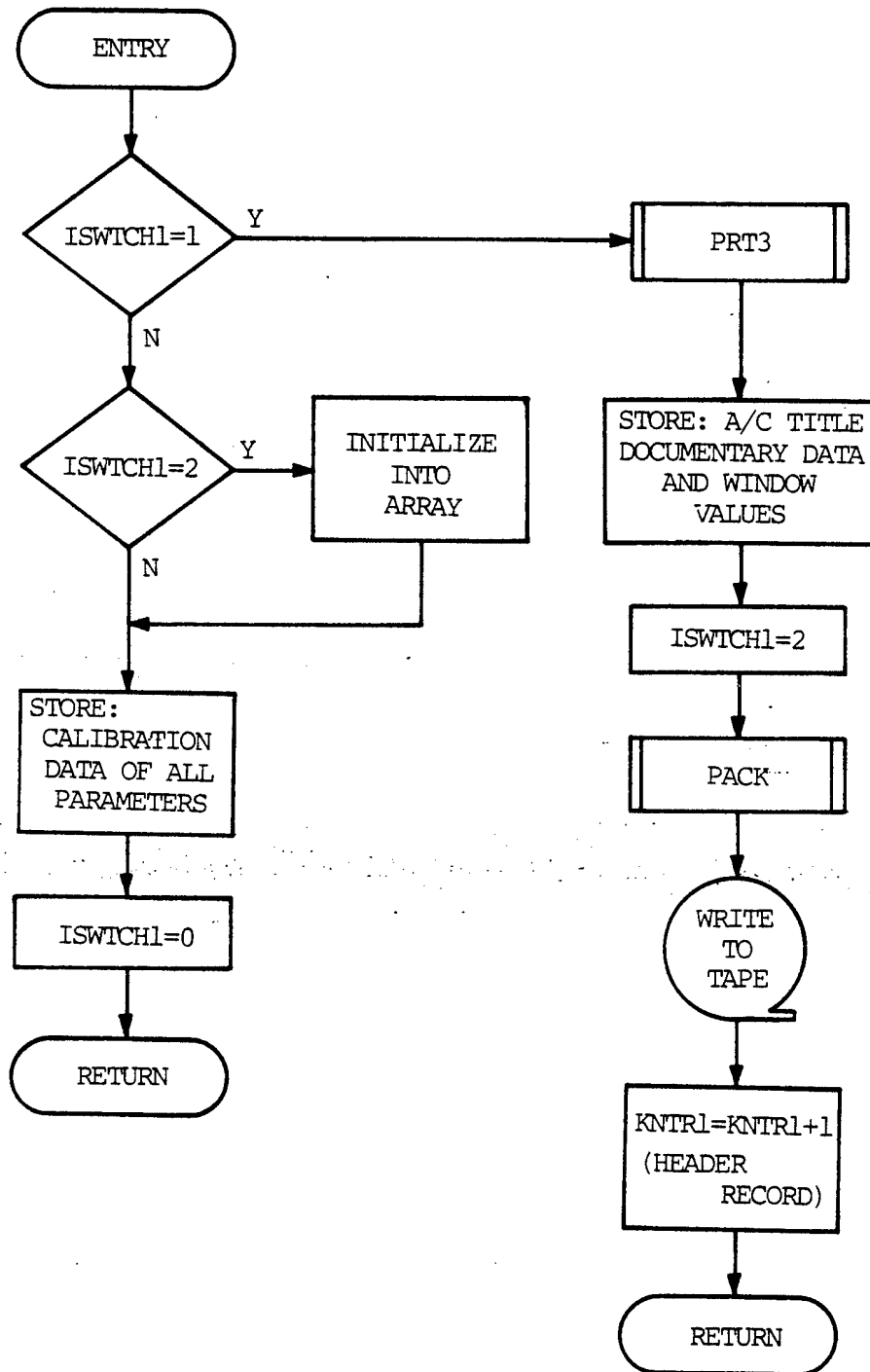


Figure A.2. (Continued).

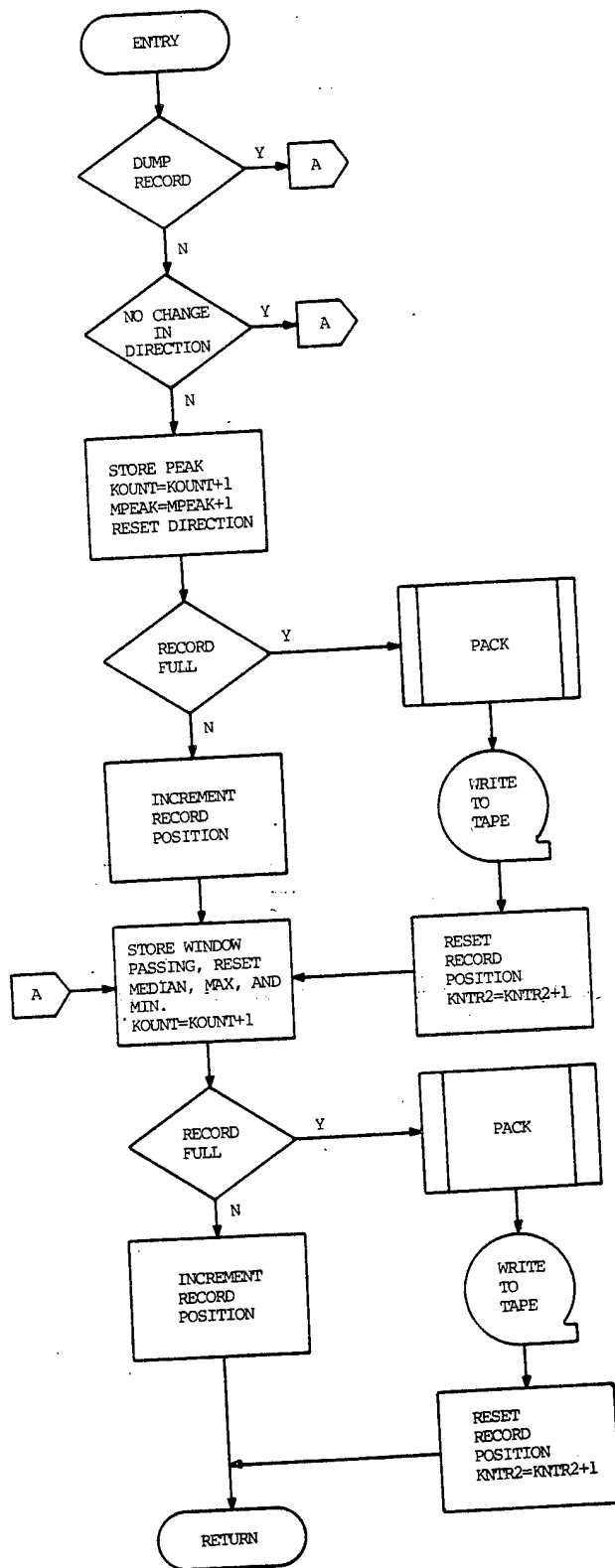


Figure A.2. (Continued).

SUBROUTINE PRT3

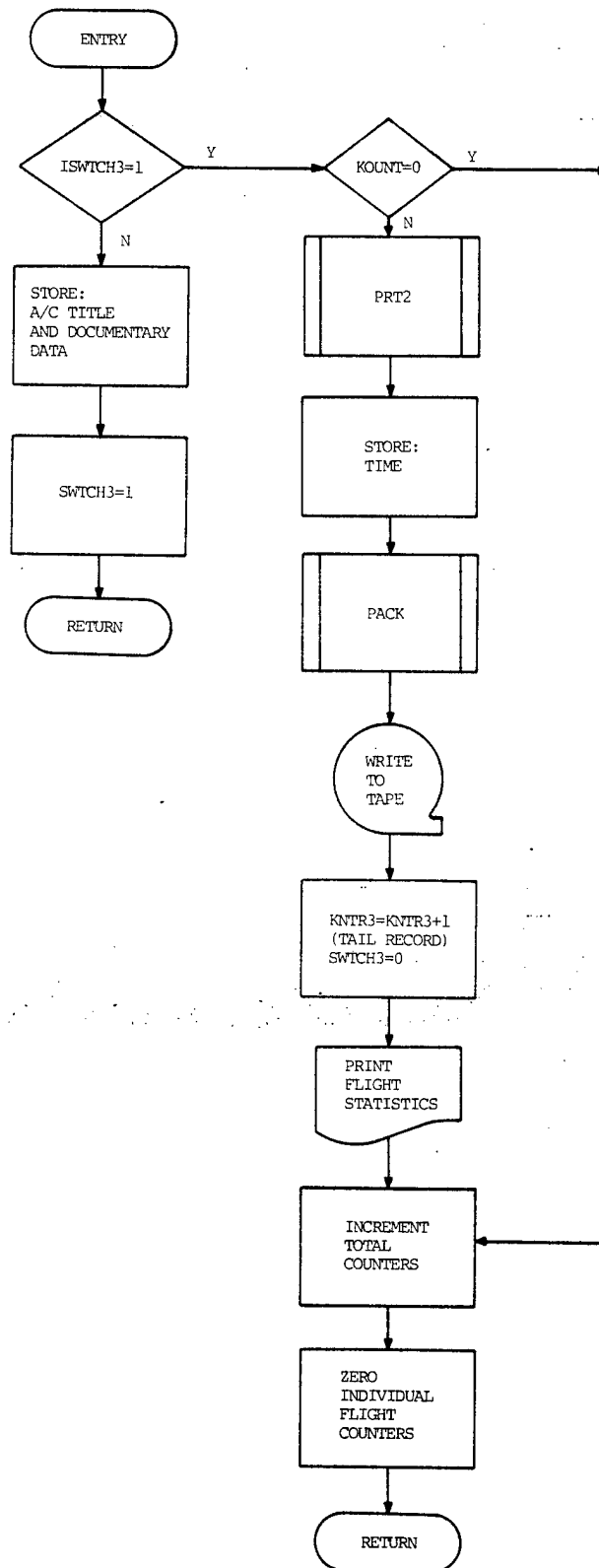


Figure A.2. (Concluded).