X-324-72-413
PREPRINT

NASA TM X- 66086

ATMOSPHERE EXPLORER CONTROL SYSTEM SOFTWARE (VERSION 1.0)

ANTHONY VILLASENOR

OCTOBER 1972

GSFC
GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

X-324-72-413

ATMOSPHERE EXPLORER CONTROL SYSTEM SOFTWARE

(VERSION 1.0)

Anthony Villasenor

October 1972

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

INTRODUCTION

This document describes the basic design of the Atmosphere Explorer Control System (AECS) software used in the testing, integration, and flight control of the AE spacecraft and experiments. The same basic system is employed by the Test and Evaluation Division at GSFC, the Spacecraft and Test System at RCA, Hightstown, New Jersey, and the Operations Control Center, GSFC.

CONTENTS

		Page
I.	SUMMARY	1
	1. Executive Control Section	1
	2. Telemetry Decommutation Section	1
	3. Command Generation Section	2
	4. EDITOR Section	3
	5. Utility Section	3
II.	EXECUTIVE CONTROL SECTION - ECS	4
	1. ECS Hardware Configuration	4
	2. ECS Message Interpreter	4
	3. ECS Flag Scan	9
	4. CRT Displays and Keyboards	11
	5. Pages	12
	6. Stripcharts	14
	7. Snapshots	14
	8. Procedure - &	14
	9. Schedule	14
	10. Test Conductors Console	14
III.	TELEMETRY DECOMMUTATION SECTION (IMD)	16
	1. Telemetry Notation	17
	2. Ping-Pong Buffers	17
	3. Latest Received Value Table (LVR)	19
	4. Limit Checking	22
	5. Conversion	22
	6. History Tape Recording	22
	7. Command Memory Dump	24
IV.	COMMAND GENERATION SECTION - CMD	24
	1. Verification Modes	25
	2. Commanding Sequence	25
	3. Command Programs	27
	4. Command Word Format	28

CONTENTS (continued)

		Page
V.	EDITOR	30
	1. Procedure Statements	30
	2. EDITOR Control Cards	32
	3. Directions for Using EDITOR	33
	4. Examples	34
VI.	THE UTILITY SECTION (UTL)	37
VII.	PROGRAMMING DETAILS	37
	1. AECS Structure	38
	2. Directives	40
	3. Page Programs	40
	4. Cathode Ray Tubes	41
	5. Event Printer	45
	6. Snapshot Printer	46
	7. Command Calls	47
	8. LRVINDEX	47
	9. CONVERT	48
VIII.	SYSTEM GENERATION	48
	1. RBM System Generation	48
	2. RBM Modifications	49
	3. AECS Overlay Program	49
REFERENCES		49
APPENDIX A.	SYSTEM DIRECTIVES	A-1
APPENDIX B.	COMMAND MNEMONICS	B-1
APPENDIX C.	AE MAIN FRAME FORMAT	C-1
APPENDIX D.	AE 8-SECOND SUBCOM FORMAT	D-1
APPENDIX E.	AE 4-SECOND SUBCOM FORMAT	E-1
APPENDIX F.	TELEMETRY ITEM DESIGNATION	F-1

ILLUSTRATIONS

Figure		Page
1	Hardware Configuration for the AECS System	5
2	RBM and AECS Operational Labels	6
3	AECS Interrupt and Trap Assignments	7
4	RAD Storage Areas Assigned by AECS	8
5	Basic Flow Control of the Executive Control System (ECS)	10
6	System Keyboard on Each CRT Device	13
7	Test Conductors Console	15
8	Core Layout for the AECS System	16
9	Basic Flow of TMD Program	18
10	Main Frame Sub-Commutated Channels	19
11	Complete Major Frame Map	20
12	Latest Recorded Value (LVR) Table. Byte Addressing	21
13	Format of Each Record on the History Tape	23
14	Basic Flow of CMD Program	26
15	Command Work Structure - Real-Time Command	29
16	Basic Logic Flow of EDITOR	31
17	AECS Overlay Structure	39
18	Flow of *PAGE Directive	42
19	Flow of Resident PAGE Processor	43
20	Flow of PAGE Interrupt/Execution Routine	44

ATMOSPHERE EXPLORER CONTROL SYSTEM SOFTWARE

I. SUMMARY

The Atmosphere Explorer Control System — AECS — has been developed to provide automatic computer control of the AE spacecraft and experiments. The software performs several vital functions, such as issuing commands to the spacecraft and experiments, receiving and processing telemetry data, allowing for extensive data processing by experiment analysis programs, etc.

The AECS has been written for a 48K XEROX Data System Sigma 5 computer, and coexists in core with the XDS Real-time Batch Monitor (RBM) executive system. RBM is a flexible operating system designed for a real-time foreground/background environment, and hence is ideally suited for this application. Existing capabilities of RBM have been used as much as possible by AECS to minimize programming redundancy.

The most important functions of the AECS are to send commands to the spacecraft and experiments, and to receive, process, and display telemetry data. In order to avoid interference between these functions and at the same time permit the execution of other processing and system overhead tasks, the AECS is logically divided into several major processing sections. The duties of each of these sections are as follows:

1. Executive Control Section

The Executive Control Section (ECS) schedules all real-time processing in the computer. Every 100 milliseconds the ECS scans through the system to initiate real-time operations such as the recording of telemetry data on a "history tape," or the updating of visual display screens. All telemetry parameters displayed on the screens can be regularly updated to show the most recently received values. ECS also interprets messages entered through the console keyboards and performs the requested tasks. Such messages can be commands to the spacecraft or experiments, calls for the execution of one or more programs, requests for hardcopy printouts of the display screens, requests for snapshots of selected telemetry data, updates on strip-chart recorders, etc. The ECS also provides the priority scheduling needed to interface real-time input/output operations with the RBM I/O routines.

2. Telemetry Decommutation Section

The Telemetry Decommutation Section (TMD) receives and processes telemetry data at the rate of 16,384 bits per second. The term "main frame"

designates the 128 8-bit telemetry words that repeat each 1/16 of a second, beginning with the frame sync pattern (1111 1010 1111 0011 0010 0000). The words are designated 1 through 128. The term main frame applies to these words independent of the subcom counter, which ranges in value from 0 through 127 and defines the 128 minor frames that make up a major frame. A "minor frame" is one of the 128 main frames that occur during the subcom count from 0 through 127. A minor frame can be identified by the subcom count value. The only difference between the terms main frame and minor frame is that main frame refers to the 128 words in general and without regard to the subcom count value, while a minor frame refers to a specific main frame with a particular subcom count value. A "major frame" contains the words from all 128 minor frames as the subcom counter goes from 0 to 127, and requires 8 seconds for completion.

A frame sync interrupt is triggered as each new minor frame is received, thereby initiating processing of the previously received minor frame; this processing includes checking of upper and lower limits, scanning and stripping of selected experiment data, establishing command verification data, etc. The most recent data for each telemetry variable is taken from the minor frame and stored in a Latest Received Value (LRV) table, which is continuously accessible by both real-time and batch programs. Several flags are set by TMD to indicate, for example, that the latest full second of data is ready to be dumped onto the History Tape, or that data from a selected experiment is ready for processing.

3. Command Processing Section

The Command Processing Section (CMD) transmits one or more 64-bit commands to the experiments and spacecraft at a maximum rate of 60 milliseconds per command. A BCH cyclic check code, with generator polynomial.

$$g(X) = (X^6 + X + 1)\ (X + 1)$$

is computed and stored in bits 58-64 of every command.

Three modes of commanding are possible:

Mode 0 - Commands are issued with no verification.

Mode 1 - Commands are issued at the maximum rate until the command output buffer is empty. Commands which the spacecraft fails to verify will be written into a table for output after all commands within the command output buffer have been issued. The procedure will be repeated until the failed-command table is empty

or until an operator-specified maximum number of re-tries have failed. If the latter occurs a message will be output to a CRT listing those commands which have failed the specified maximum number of re-transmissions.

Mode 2 - Command N+1 will not be issued until command N has been transmitted and verified. Again, after an operator-specified maximum number of re-transmissions have failed, a message will be output and the operator can elect to halt Mode 2 commanding or to delete the failed command.

Each experiment or commandable spacecraft subsystem is associated with a "command program" which is executed whenever the system receives a sequence of command mnemonics related to that subsystem. The command programs are responsible for providing CMD with the proper bit patterns for minor or major mode commanding, operation code contents, and minor mode instructions. CMD can handle up to 64 commands at a time.

4. EDITOR Section

The EDITOR Section is used to create, modify and update automatic control sequences and procedures. The automatic control sequence, hereafter referred to as a "Schedule", contains a sequence of statements which may be commands to the experiments or spacecraft, or directives to the real-time programs of the AECS system. The Schedule may be used for experiment testing, checkout during spacecraft integration, or for directing in-flight operations. A "Procedure" is a short sequence of statements grouped together for convenience; a typical Procedure would be a set of directives or commands to turn on an experiment, or to turn it off if certain telemetry limits are exceeded.

5. Utility Section

The Utility Section (UTL) provides a powerful programming tool for the on-line debugging of real-time computer programs. UTL was conceived as a result of the schedule pressures placed on the AECS programming effort; the additional work in created UTL saved considerable time in the later checkout stages of AECS applications programs.

Each of the above sections will be described in more detail on the following pages.

II. EXECUTIVE CONTROL SECTION - ECS

The ECS supervises all real-time processing operations. It receives control of the computer every 100 milliseconds through a clock interrupt and proceeds to scan a series of system flags associated with numerous processing tasks. Some of these tasks perform basic housekeeping functions, such as surveying and servicing the Test Conductors' Panel, connecting and arming all AECS interrupts, updating the History Tape, and driving the analog strip chart recorders. Other tasks initiated include interpreting all messages submitted to the system, loading and executing AECS overlay programs, and performing system utility functions.

1. ECS Hardware Configuration

A diagram of the AECS hardware configuration is shown in Figure 1. ECS controls one printer and two 9-track IBM-compatible magnetic tapes. These devices are not accessible by standard RBM system calls because they are processed directly through a modified RBM I/O handler. The line printer (LPA02) is referred to as the Event Printer, and has an AECS operational label of EP. A magnetic tape, either unit 9TB82 or 9TB83, is defined as the History Tape HT and is used for saving raw telemetry data for later off-line processing and analysis. Another tape, 9TA80, is defined as the Control Tape CT and is used for Schedule executions.

In addition, ECS controls several XDS System Keyboard Display cathode ray tubes (CRTs). These television screen/keyboards are general purpose interactive communications devices that allow input to and output from the system. All AECS operational labels are shown in Figure 2.

There are 16 interrupts available in the Sigma 5 computer, and most of these are used by ECS. The interrupt assignments are shown in Figure 3. The priority ordering scheme, while still evolving, has been quite satisfactory. AECS usage of the RAD files is shown in Figure 4.

2. ECS Message Interpreter

Subroutine MSGINTRP decodes and interprets all messages submitted to the AECS system. MSGINTRP immediately displays each 80-character input message simultaneously on line 19 of all CRT screens and also on the Event Printer. It separates, sequences, and stores up to 20 parameters in a buffer later used by the overlay programs. Then, depending on the type of message, it transfers control to the relevant message post-processor, in which the various ECS scan flags are set.

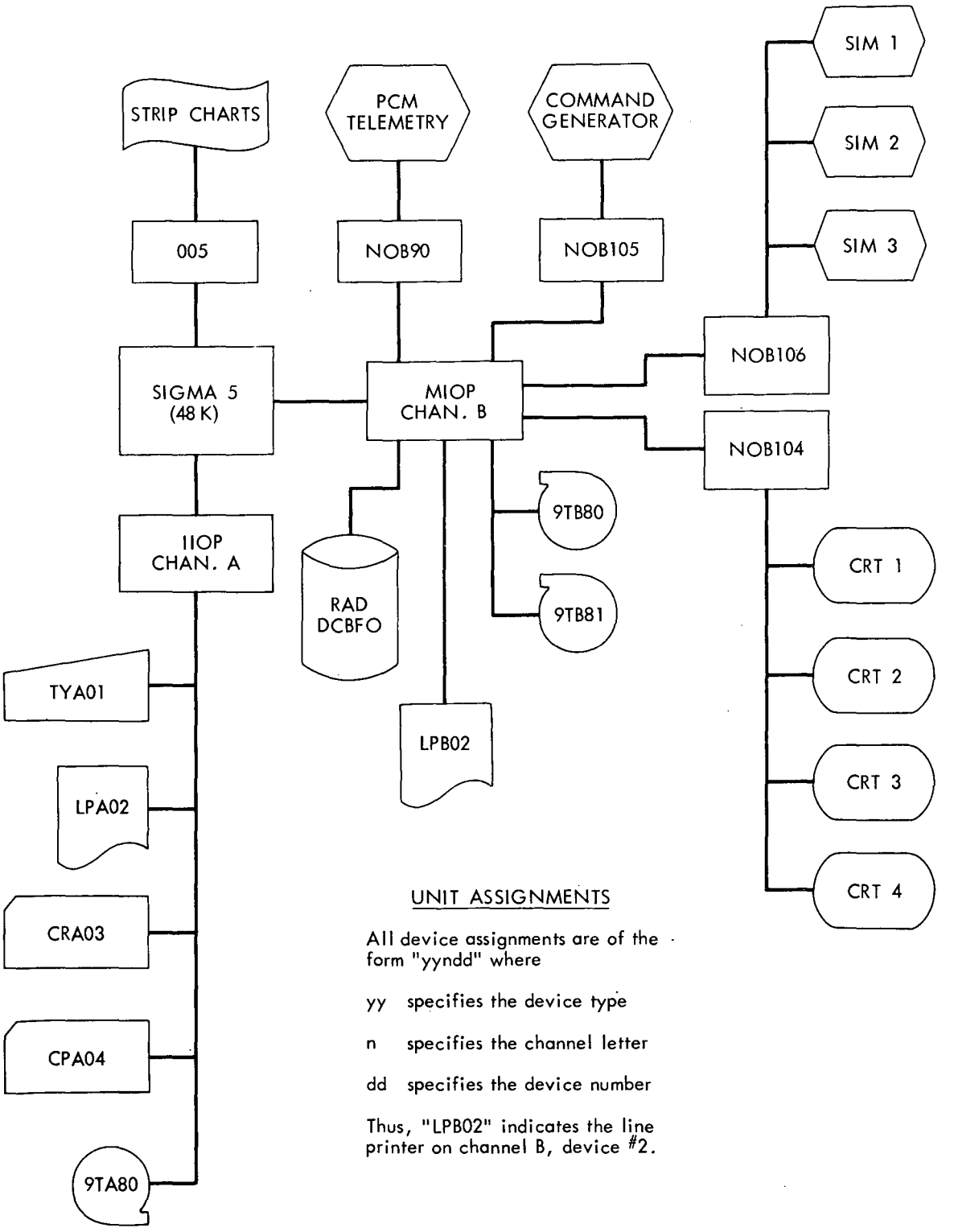


Figure 1. Hardware Configuration for the AECS System

LABEL	UNIT	FUNCTION
HT	9TB82	History Tape
CT	9TA80	Control Tape for Schedules
EP	LPA02	Event Printer (800 1 pm) (RCA 1100 1 pm)
SN	LPA02	Snapshot Printer (800 1 pm) (RCA 1100 1 pm)
LP	LPB02	Line Printer (1100 1 pm)
C	TYA01	Sigma 5 Control Command Input
OC	TYA01	Operator's Console
LO	LPB02	Listing Output
LL	LPB02	Listing Log (RBM Job Cards)
DO	LPB02	Diagnostic Output (Dumps)
CO	9TB80	Compressed Output
CI	9TB81	Compressed Input
BO	9TB80	Binary Output
BI	CRA03	Card or Binary Input
SI	CRA03	Symbolic (Source) Input
SO	9TA80	Symbolic Output

Figure 2. RBM and AECS Operational Labels

All messages to AECS are identified by special symbols beginning in column 10 of the card image. Use of these special symbols speeds up system reaction time by reducing the number of table look-up operations needed to convert the mnemonic string to the desired form for the post-processors. The symbols and corresponding message types are, in order of priority:

* - System Directive

/ - Subsystem Command (Real-Time)

- Spacecraft Command (Stored)

$ - Simulator Directive

INTERRUPT	DESCRIPTION
X'58'	Clock 1 counter (2000 hz), used to generate simulated real-time telemetry data for checkout.
X'5A'	Clock 3 counter (500 hz), triggers ECS X'6B' every 100 milliseconds, providing ECS is not busy.
X'60'	I/O Endaction for command transmission through the 7601.
X'61'	I/O Endaction for transmission of spacecraft simulator data through the 7929.
X'62'	Telemetry FRAME SYNC INTERRUPT, occurring every 62.5 milliseconds at word 3 of real-time data.
X'63'	-
X'64'	-
X'65'	-
X'66'	-
X'67'	I/O endaction on telemetry History Tape.
X'68'	I/O endaction for ECS special devices (CRTs, event printer, snapshot printer, and stripcharts).
X'69'	I/O endaction for SIM printer
X'6A'	I/O endaction for RAD transmission of real-time telemetry data. Used by TMD program.
X'6B'	ECS system flag scan. Main program of AECS.
X'6C'	ECS queue processing.
X'6D'	ECS Page execution.
X'6E'	ECS directive and command program execution.
X'6F'	RBM Control Task interrupt.
CAL2	Trap for CRT display instructions.
CAL3	Trap for Event Printer and Snapshot Printer operations.

Figure 3. AECS Interrupt and Trap Assignments

AREA	TRACKS	FUNCTION
SP	70	System programs - RBM, FORTRAN, MACRO-SYMBOL
FP	157	Foreground programs - AECS root and segments, directives, commands, snapshots, etc.
BP	75	Background program area
D1	30	Foreground data files - PEND, ACTV, STD, RAWFILE
D2	20	Background data files
D3	50	Procedures
XA	10	IOEX file
CK	10	Checkpoint area for background programs
BT	90	Background temporary area - OV and GO files, scratch files X1, X2, , , X9.
512 Tracks Total		

Figure 4. RAD Storage Areas Assigned by AECS

? - Utility Directive

& - Procedure Call

A "directive" (*) is defined as a control instruction for the AECS which effects the state of the system; examples of directives are requests for page displays, starting or stopping real-time telemetry processing, entering a wait state, etc. A "command" (/) is an experiment-related overlay program which generates one or more 64-bit commands to be transmitted to the experiments. A "simulator directive" ($) is a program which generates one or more 32-bit words for the spacecraft simulators. A "utility directive" (?) is a subroutine which dynamically lists, displays, or modifies computer memory locations in order to assist in program debugging. A "procedure" (&) is group of combined directives and commands which are linked together as a convenience.

As soon as each message code is identified, control is passed to one of several post-processing routines:

* goes to DIRINPUT and DIRLOAD

/ and # go to CMDINPUT and DIRLOAD

$ goes to SIMPROC

? goes to UTILITY

& goes to PROCINP

3. ECS Flag Scan

A number of flags — locations in core — are periodically inspected by ECS to determine the status of real-time operations and initiate processing tasks. Flags currently examined by ECS are: (See Figure 5.)

NBREADY — if non-zero, indicates that the latest one second (16 minor frames) of PCM telemetry data is ready for transfer onto the History Tape HT. Byte 0 of NBREADY contains a pointer indicating which of 8 groups of 16 minor frames each is ready to be dumped, and halfword 1 of NBREADY contains the first word address of the data. This flag is set by TMD and cleared by HTDUMP.

TMDREADY — if non-zero, indicates that the TMD program must connect and arm the telemetry frame sync interrupt in order to begin receiving real-time telemetry data. This flag is set by the *START directive and reset by TMD.

CRTBUSY — if non-zero, indicates that the CRT displays are busy and not available for input or output operations at the current time. If zero, the CRTs are available, and each of the CRT keyboards are queried to find out if any new messages have been submitted to the system. This flag is set by the CRT handler and cleared by the ECS I/O endaction routine.

PROCFLAG — if byte 0 is non-zero, indicates that the next statement of a currently executing procedure has been read into core and is ready for operation. Byte 0 is actually the RBM I/O completion code set by the CAL1 READ operation and reset by the PROC processor.

SCHDFLAG — if non-zero, indicates that the schedule processor is in operation: that is, messages are being read in automatically and continuously from the schedule tape. This flag is set by the *SCHEDULE directive and reset by an end-of-data condition on the schedule tape.

MSGFLAG — if non-zero, indicates that a new message has been submitted to the AECS system from some input device. Byte 0 contains the identification of the calling unit, and halfword 1 contains the address of the message buffer.

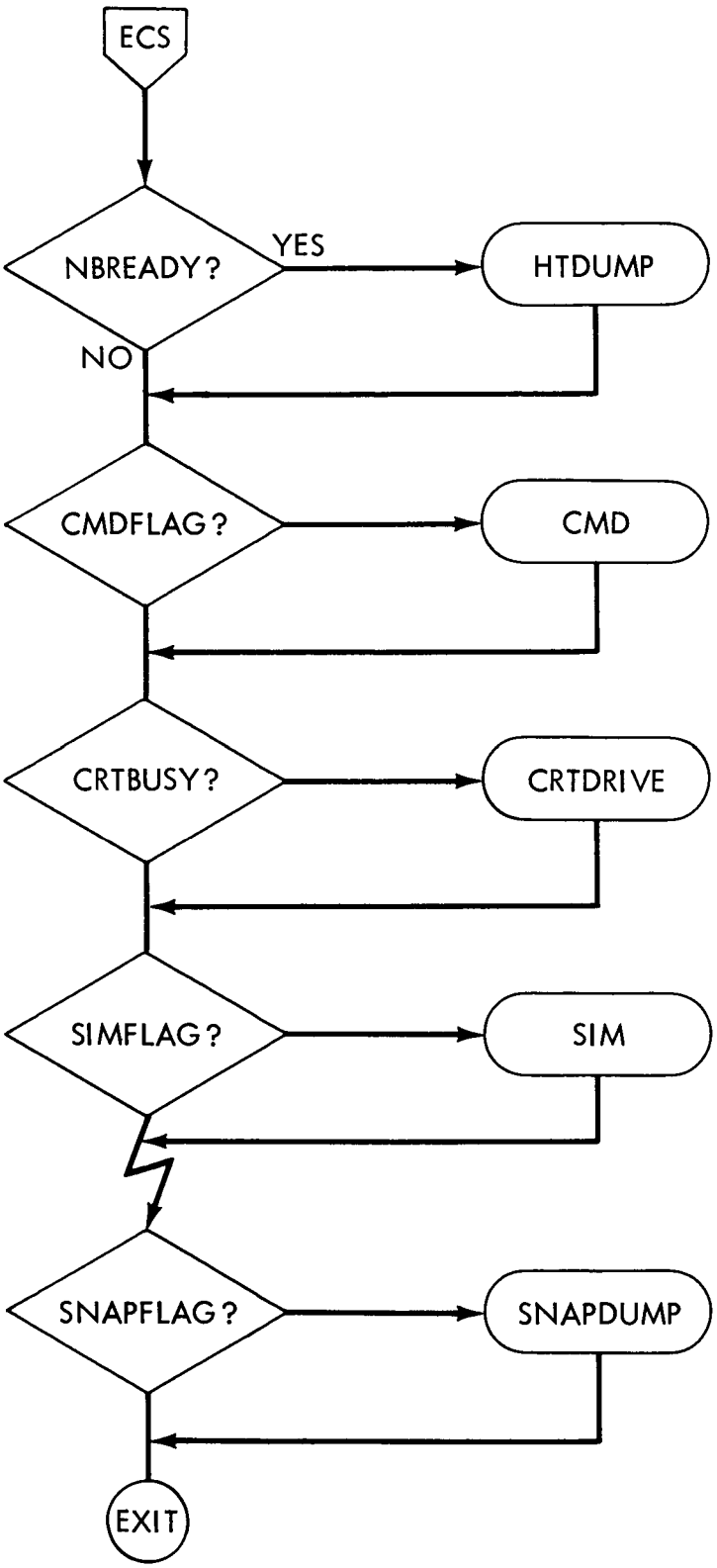


Figure 5. Basic Flow of the Executive Control System (ECS)

SIMFLAG — if non-zero, indicates that a simulator directive request has been received and needs processing. SIMFLAG is set by MSGINTRP and cleared when the desired program has completed execution. SIMFLAG contains the mnemonic of the simulator directive.

CMDFLAG — if non-zero, indicates that a command program has been requested. CMDFLAG is set by MSGINTRP and cleared when the command program is finished. CMDFLAG contains the 4-character experiment mnemonic.

DIRFLAG — if non-zero, contains the 8-character mnemonic of the directive program when has been requested. DIRFLAG is set by MSGINTRP and cleared after the directive program has finished.

PAGEFLAG — if non-zero, indicates that one or more CRT display pages must be called in (overlaid) and executed in order to update the display. PAGEFLAG is set and cleared by the *PAGE directive.

UTLFLAG — if non-zero, contains the mnemonic of the desired Utility directive. UTLFLAG is set by MSGINTRP and cleared after the desired utility program has been initiated.

UPDATE — if non-zero, indicates that the currently executing directive or command program will remain in core and be re-executed at one second intervals, unless another message request is received or the program is cancelled. UPDATE is set by the individual programs and cleared by MSGINTRP or CANCEL.

CRFLAG — if non-zero, indicates that messages are expected from the card reader, defined as Unit B in AECS. CRFLAG is set by setting Sense Switch 1 on the computer console, or by depressing the "CARD" button on the Test Conductors' Panel. CRFLAG is cleared when the sense switch is reset or when there are no more cards in the hopper.

TTYFLAG — if non-zero, indicates that messages are expected from the computer teletype console (AECS Unit A). TTYFLAG is set by setting Sense Switch 2 on the computer console and cleared when the (one) message is transmitted by the End-of-Message (EOM) or New Line (NL) key on the teletype.

4. CRT Displays and Keyboards

ECS controls three XEROX Data Systems "System Keyboard Display" cathode ray tubes (CRTs). These television screen/keyboards, which are identified as CRT#1, CRT#2, and CRT#3, are general purpose interactive communications

devices which allow input to and output from the system. Each CRT screen displays alphanumeric characters on a matrix 20 lines down by 80 columns across. The top line is defined as line #1 and the leftmost column is column #1. Character transmission occurs at the rate of 1200 characters per second.

Messages are typed at the CRT keyboards in LOCAL mode. Figure 6 shows the keyboard arrangement. Transmission of messages is performed by depressing the SEND key. Errors in typing may be eliminated by backspacing and typing over the erroneous characters, by depressing the DELETE key which removes the character under the cursor, or by depressing the LINE ERASE key which erases the entire line to the right of the cursor.

The bottom three lines of each display screen are reserved by AECS for the following purposes:

Line 18 - System error messages

Line 19 - Statement which is currently executing

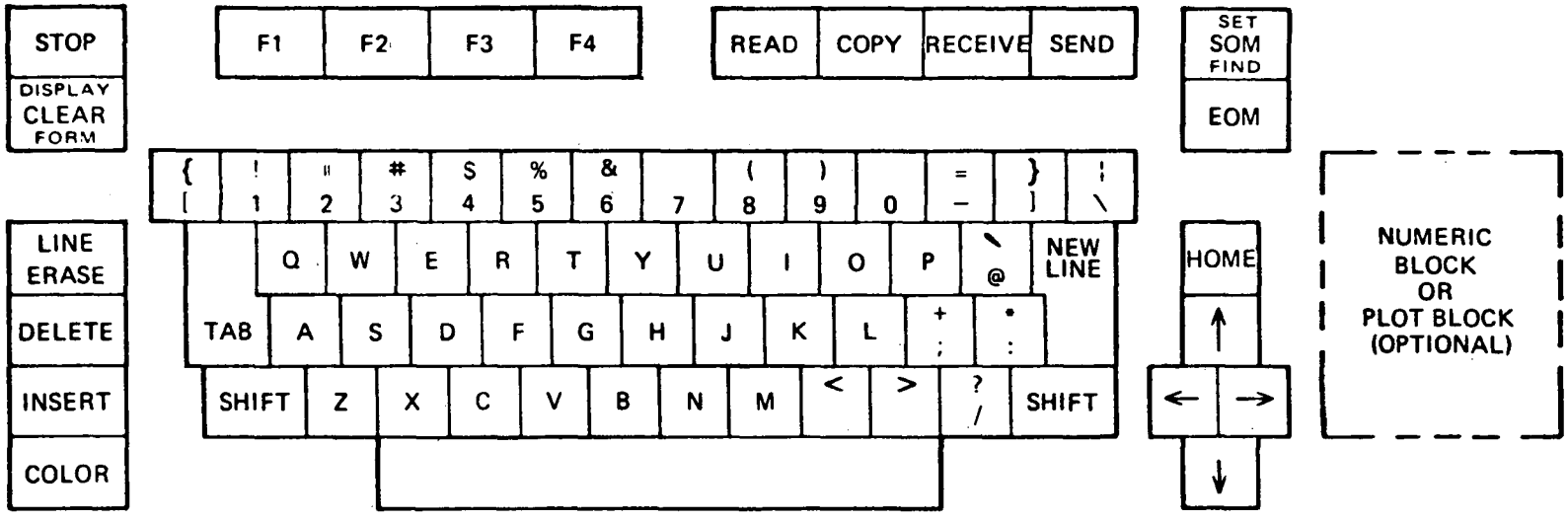
Line 20 - Next statement to be executed.

Each CRT screen, therefore, is limited to displaying no more than 17 lines of user-oriented data.

While each CRT operator can issue commands to the spacecraft and experiments, he may be enabled or disabled from doing so by the *ENABLE and *DISABLE directives which can be submitted only through CRT #1. CRT #1 thus enjoys executive control over all other CRTs.

5. Pages

A "Page" is an overlay segment program which displays real-time telemetry data in raw or processed form on any specified CRT screen. Each Page program receives control at regular time intervals (every 3 seconds), and operates autonomously in the system; that is, there is no restriction on its making use of the full range of CRT display capabilities, such as blinking, alarming, and plotting. When a Page program is initiated, it generates title and header information on the screens, then converts and displays the desired telemetry data; subsequent executions of the same Page program can bypass the transmission of unchanged title data and proceed directly with updating the telemetry data. Pages can operate in core simultaneously with command and directive programs.



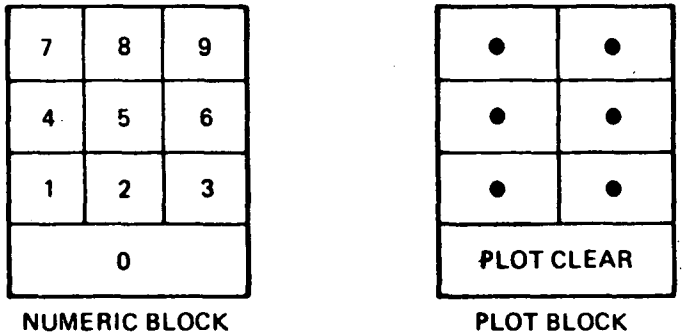


Figure 6. System Keyboard on Each CRT Device

6. Stripcharts

ECS services any number of analog strip chart pen records and event markers. These pens operate independently and simultaneously, with plotting speed and scale regulated by controls at the recorder console. Access to these pens is through the directives *CHARTON and *CHARTOFF.

7. Snapshots

A "snapshot" is an overlay segment program similar to a directive or command, which produces a fixed format printout of telemetry or other data associated with the spacecraft, experiments, or simulators. The Snapshot Printer SN is used exclusively as the output device.

8. Procedure or "PROC"

A "Procedure" is a logically related set of system statements (directives, commands, snapshots, etc.) which are grouped together for compactness and operational efficiency. A Procedure call is a single statement resulting in the execution of all statement contained in that Procedure. Procedures are created by the AECS EDITOR program and placed on the RAD for rapid access. All Procedure calls are indicated by the prefix symbol &.

9. Schedule

A "Schedule" is an automatic control sequence. It can contain up to 10,000 directives, commands, procedures, snapshots, simulator directives, etc., which are executed either sequentially or as directed by the statements themselves. A Schedule is actually a magnetic tape created from an input card deck by the AECS EDITOR program. One of the functions of EDITOR is to add, delete, or modify existing schedules, as well as provide a listing of the schedule tape contents, including a full expansion of all imbedded Procedures. A Schedule is called for execution by the *SCHEDULE directive.

10. Test Conductors Console

A push-button console is provided to simplify the system operator's task of submitting requests to the system. The console allows requests for CRT Pages, snapshots, Group commands, Matrix commands, etc., and also displays the Greenwich Mean Time (GMT) and Spacecraft Elapsed Time (SET) in digital form. ECS acknowledges each request by illuminating the depressed push-button. Only one push-button operation can occur at a time.

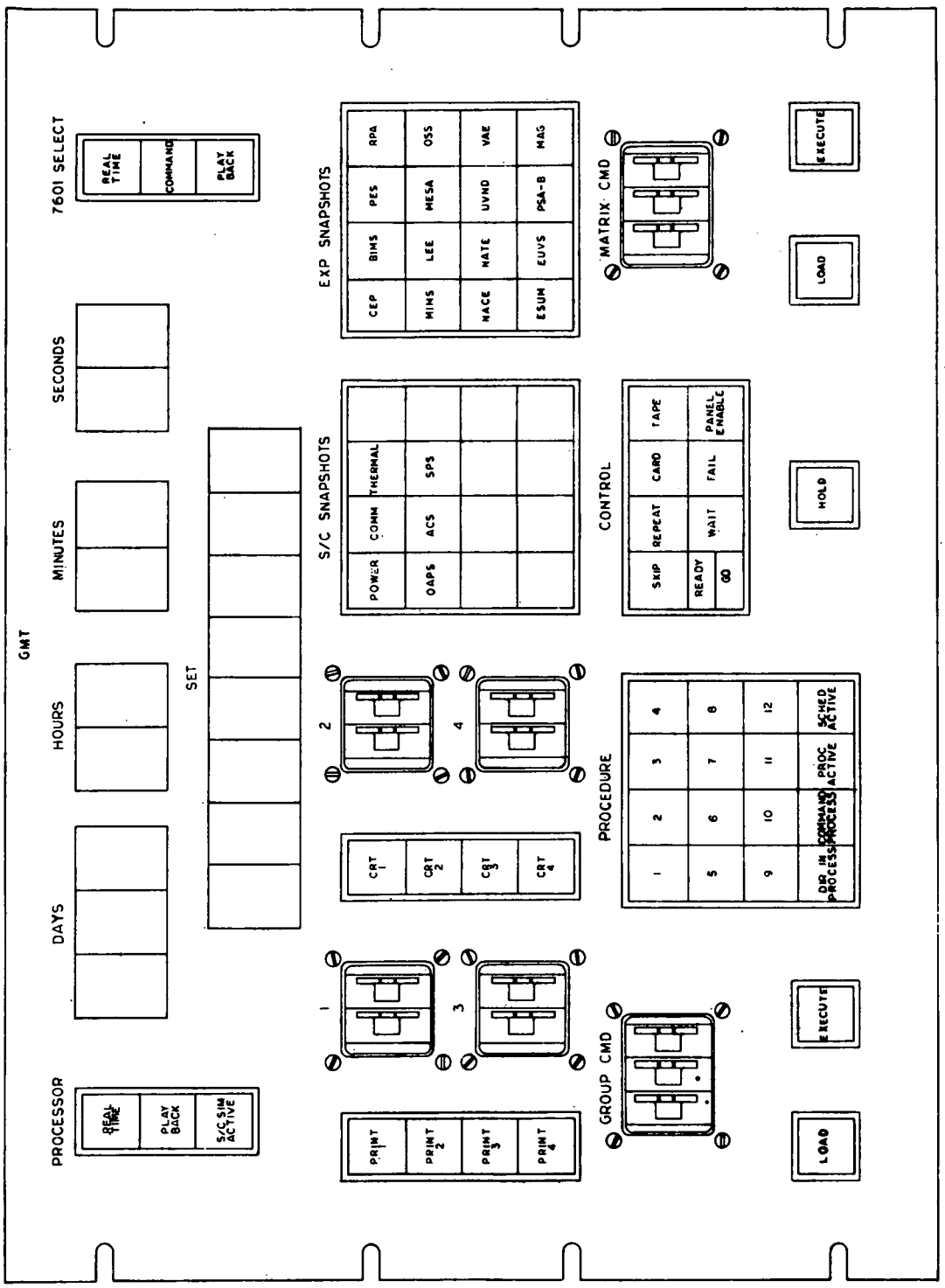


Figure 7. Test Conductors Console

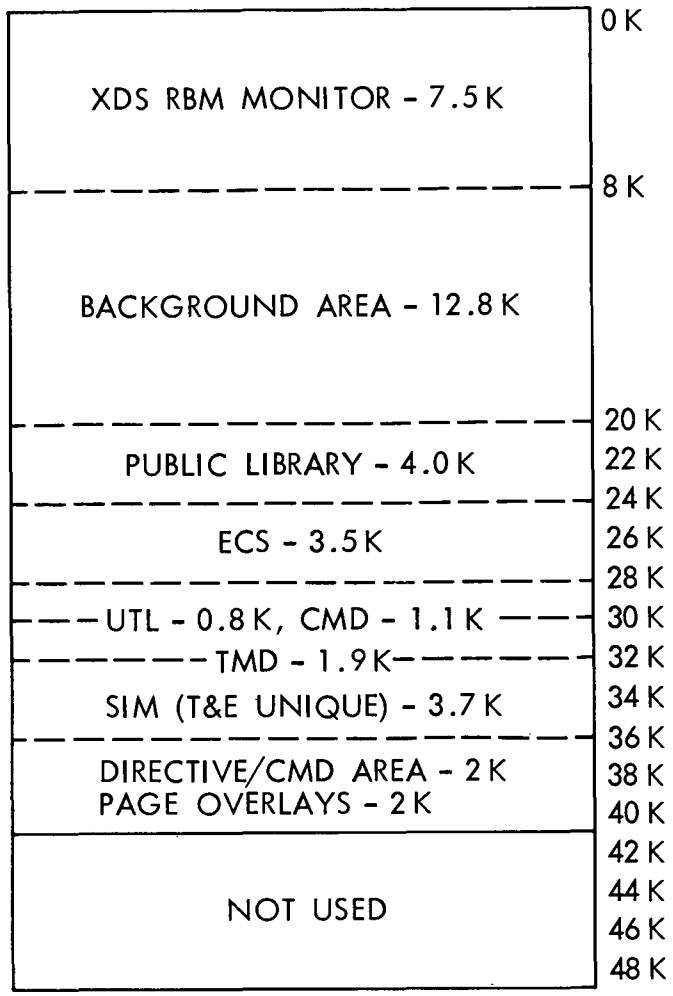


Figure 8. Core Layout for the AECS System

III. TELEMETRY DECOMMUTATION SECTION (TMD)

TMD receives and processes narrowband PCM telemetry data at the minor frame rate of 128 8-bit words (bytes) every 62.5 milliseconds. Each minor frame is checked for frame synchronization (coded as X 'FAF320' in words 1, 2 and 3) and specified words are limit-checked against preset limits (*LIMITS directive) to establish a gross estimate of experiment and spacecraft performance. If commands are concurrently being issued to the experiments or spacecraft, the command verification information from words 4 and 5 is extracted for use by the Command Processing Section. A specified number of telemetry words may be stripped out of each minor frame and saved for real-time use by data processing programs. Telemetry housekeeping is performed, and the relevant flags, pointers, counters, etc., are updated to reflect the current state of the system. Updated status can then be displayed in hard copy on the

Snapshot Printer by "snapshot" programs, or dynamically on the CRTs through Page routines. Raw telemetry data is saved on the History Tape in 560-word records containing one second of telemetry data and the latest one second's worth of verified commands. (See Figure 9.)

1. Telemetry Notation

Telemetry words may be referenced in AECS through the following notation convention:

TM(I) or TM(I,J)

where

I = the telemetry word index, from 1 through 128

and

J = the subcom step number, also from 1 through 128. (See Figures 10 & 11.)

Individual programs requiring this notational form for input are responsible for checking the values of the subscripts to avoid erroneous references. The more sophisticated notational scheme described in Appendix F of this document has not been implemented in the AECS software.

2. Ping-Pong Buffers

There are two one-second ping-pong buffers in TMD, each containing 16 minor frames. As soon as the first ping-pong buffer is full, the second begins to receive data. When either ping-pong buffer has been filled, the NBREADY flag in ECS is set so that the data can be dumped onto the History Tape, if requested.

These buffers are accessed each time the frame sync interrupt (X'62') fires, at which time TMD executes a direct read sequence to bring the next 128 bytes into the appropriate slot in core. A "window" is provided to verify that the frame sync interrupt does indeed fire at the prescribed rate of 62.5 milliseconds; if it fires before 60 milliseconds has elapsed, or after 67 milliseconds has elapsed, a message appears on the CRTs and the Event Printer. The next minor frame is stored in the location specified by the contents of the sub-com counter, TM(37).

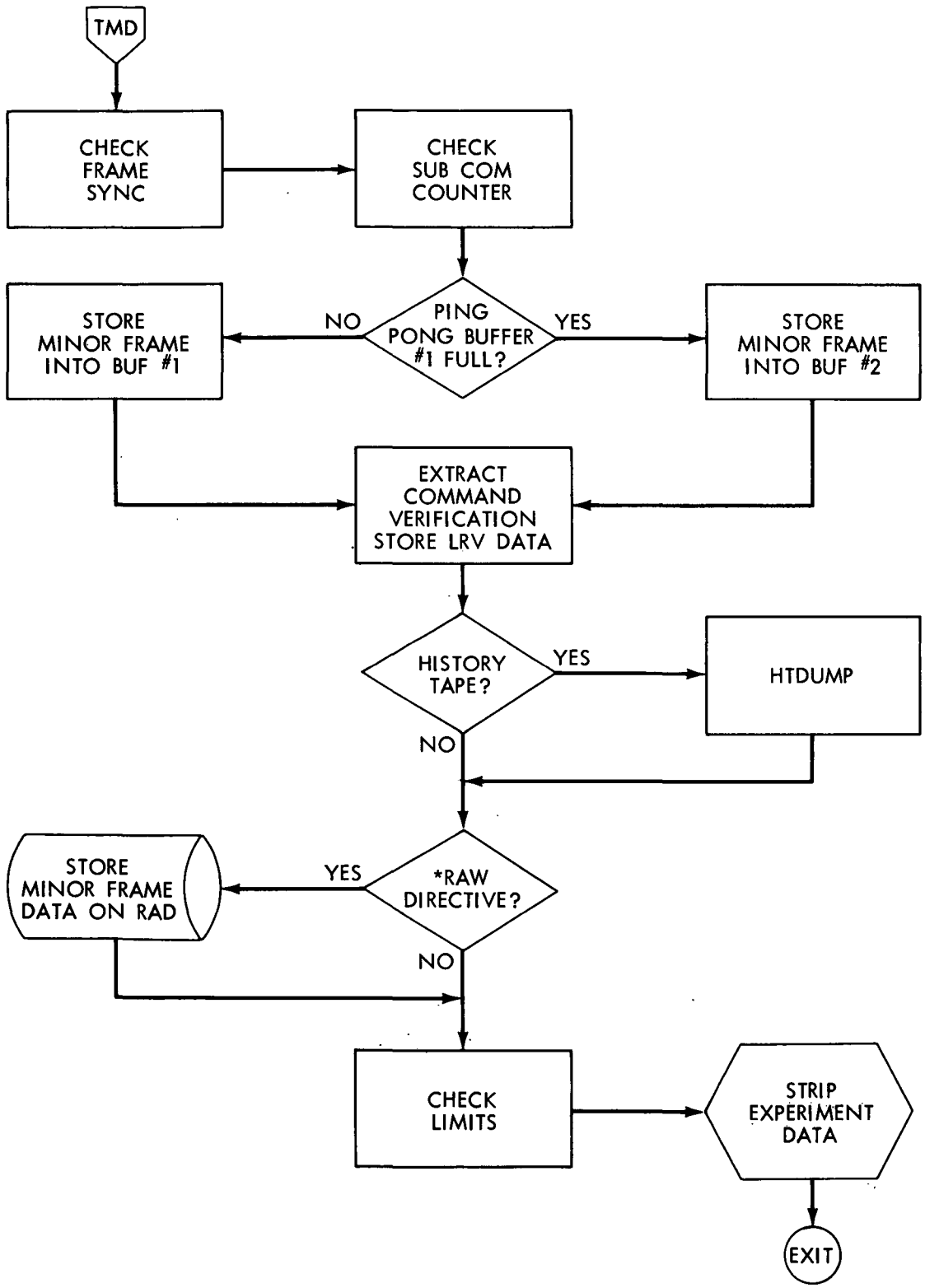


Figure 9. Basic Flow of TMD Program

CHANNEL	SUB-COM LEVEL	IDENTIFICATION
TM(17,i)	i = 1 to 4	Spacecraft Attitude Control
TM(18,i)	i = 1 to 4	Spacecraft Attitude Control
TM(20,i)	i = 1 to 8	SPS
TM(34,i) TM(35,i)	i = 1 to 4 i = 1 to 4	ESUM ESUM
TM(46,i) TM(47,i) TM(48,i)	i = 1 to 2 i = 1 to 2 i = 1 to 2	VAE VAE VAE
TM(65,i)	i = 1 to 64	Spacecraft Data
TM(66,i)	i = 1 to 128	Spacecraft Data
TM(67,i)	i = 1 to 64	Experiment Data
TM(68,i)	i = 1 to 128	Experiment Data
TM(98,i) TM(99,i)	i = 1 to 4 i = 1 to 4	ESUM ESUM
TM(110,i) TM(111,i) TM(112,i)	i = 1 to 2 i = 1 to 2 i = 1 to 2	VAE VAE VAE
TM(119,i) TM(120,i)	i = 1 to 4 i = 1 to 4	MESA MESA

Figure 10. Main Frame Sub-Commutated Channels

3. Latest Received Value Table (LVR)

TMD maintains an LRV table which contains the most recently received telemetry words, and which is accessible by both real-time and background programs for data processing. The LRV table actually consists of two arrays, LRVMINOR and LRVSUBCM, corresponding respectively to 32 full minor frames and to all 128 steps of sub com channels 65, 66, 67, and 68. LRVMINOR is 1024 words long, and is updated every 62.5 milliseconds; LRVSUBCM is 512 bytes long, and similarly updated. (See Figure 12.)

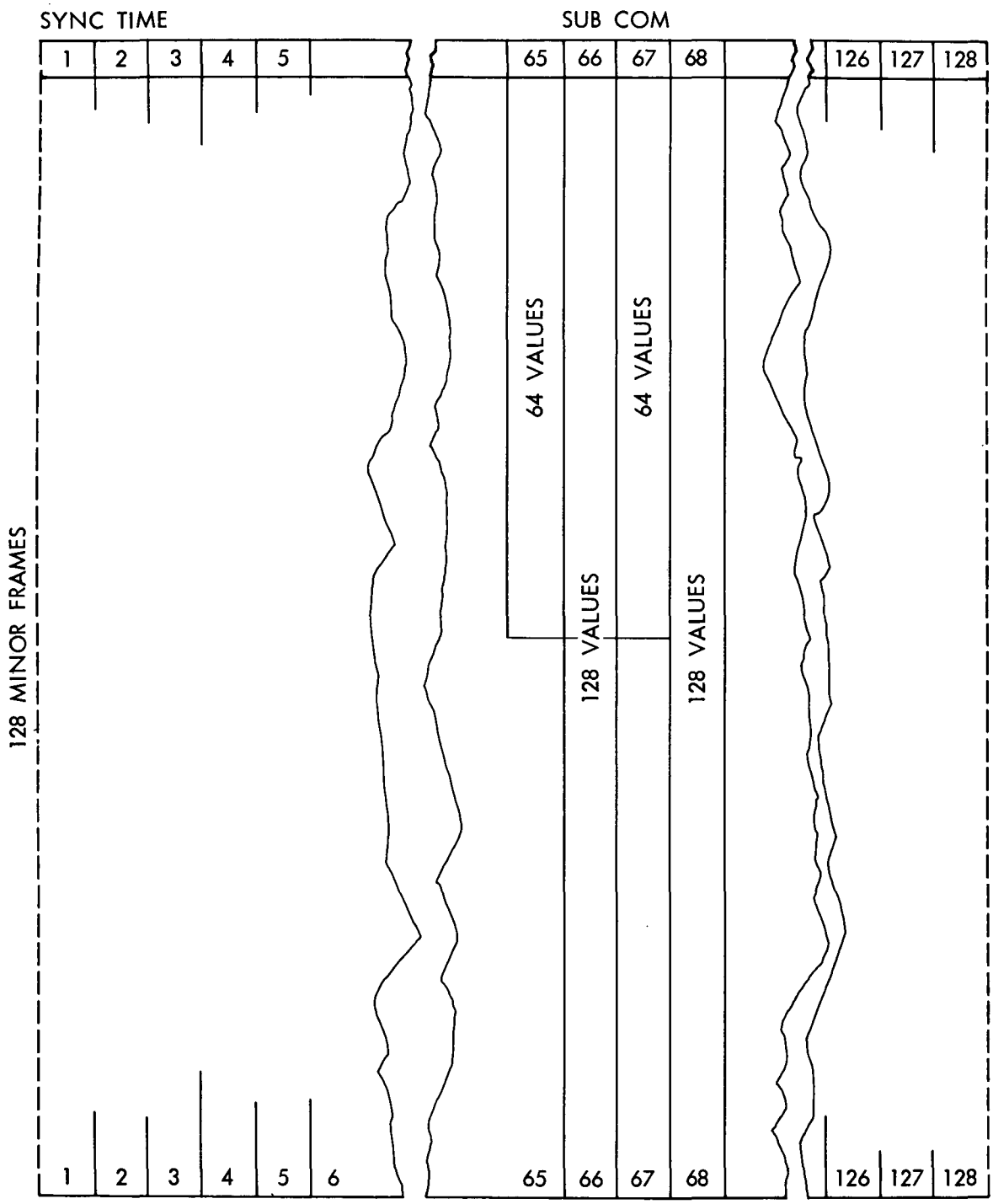


Figure 11. Complete Major Frame Map

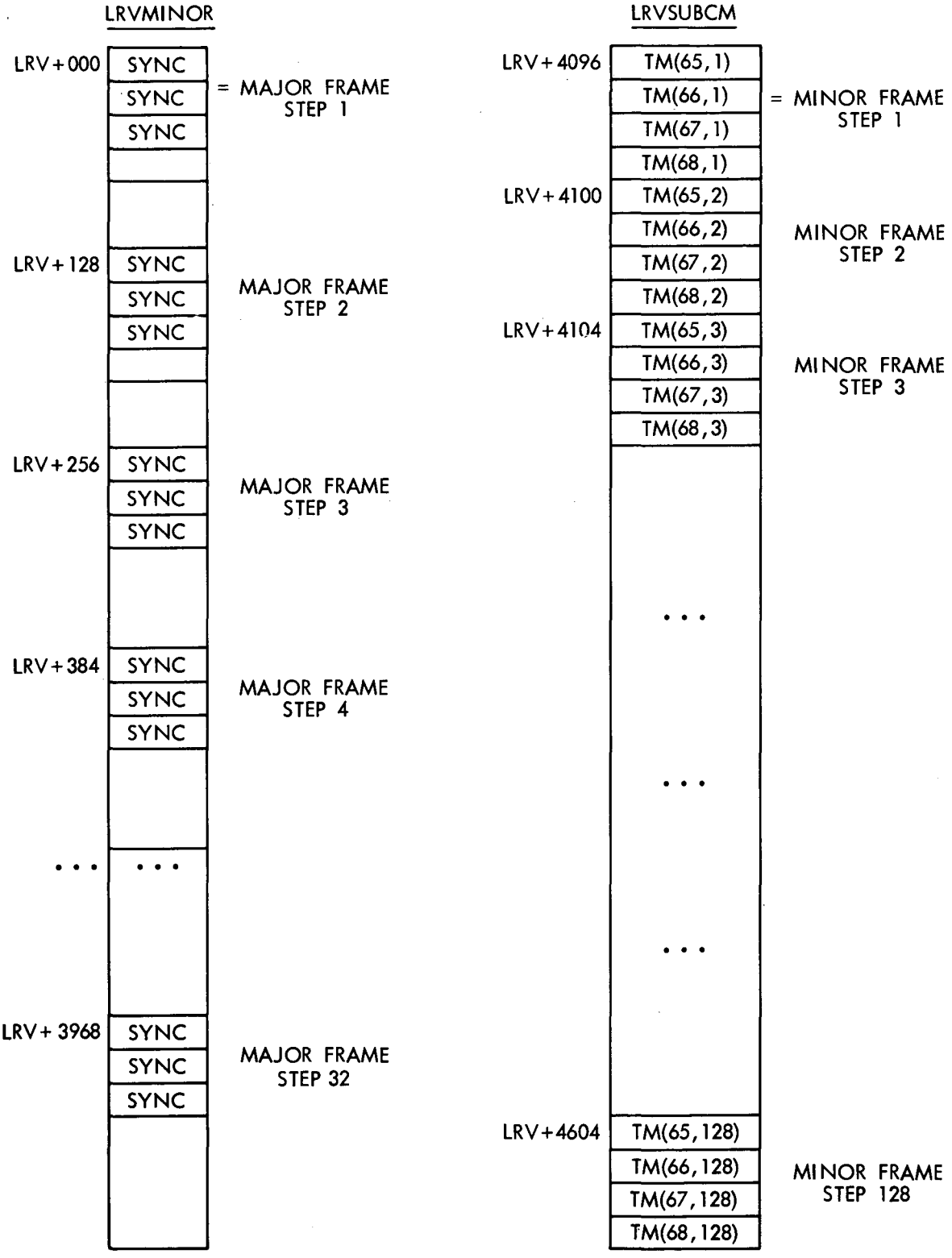


Figure 12. Latest Recorded Value (LRV) Table. Byte Addressing

4. Limit Checking

The capability exists for TMD to check each telemetry word against specified high or low limits, and to enter a small resident subroutine if these limits are exceeded. The limit values are entered through the *LIMITS directive; the resident subroutine, which can contain commands, must be compiled separately and added to the resident TME programs. See *LIMITS and *LIMOUT directives.

Limit checking may be stopped temporarily and then re-instated through the *LIMOFF and *LIMON directives.

5. Conversion

While TMD does not itself convert raw telemetry data into engineering units, the capability is provided for other real-time or data processing programs to do so. TMD maintains the coefficient table (TMDCOEF) which is built up by the *CONVCOEF directive. The contents of TMDCOEF are the polynomial coefficients A0, A1, A2,,,. . ., A7, which form the equation

$$Y = A0 + X^* (A1 + X^* (A2 + X^* (A3 + \ldots X^* A7)))))))$$

6. History Tape Recording

All PCM narrowband telemetry data can be recorded on magnetic tape. If the *HTSTART directive has been issued, TMD will dump telemetry data every second onto 9TB82 or 9TB83 through the HTDUMP program. HTDUMP is activated by the NBREADY system flag; this program first checks to insure that the History Tape is ready to receive data, and then it proceeds to execute the I/O instructions. The actual format of the 560-word History Tape record is shown in the explanation of the *HTSTART directive. This particular record size was chosen in view of the size of the telemetry ping-pong buffer, the possibility of tape channel lock-out during I/O, the readability of 560-word records by background programs, and the compatibility of this size with other computer systems.

It should be noted that provision has been made, through the *HTSTOP directive, to stop recording data at any time in order to conserve tape usage and system overhead. In addition, if one of the History Tapes is nearing end-of-reel, another tape may be readied and the *HTSTART directive re-issued with reference to the new tape; in this way, no data is lost during important recording sessions.

Each 560-word record occupies 2.7 inches on tape (at 800 bpi). The inter-record gap is approximately 0.65 inches. Each record dump thus takes 3.35 inches of tape. For a full 2400-foot reel, over 8,700 records can be dumped, or about 2.4 hours worth of data. (See Figure 13.)

Words 1 - 32	Minor Frame I I = 1,17,33,..,113
Words 33 - 64	Minor Frame I + 1
Words 65 - 96	Minor Frame I + 2
Words 97 - 128	Minor Frame I + 3
Words 129 - 160	Minor Frame I + 4
Words 161 - 192	Minor Frame I + 5
Words 193 - 224	Minor Frame I + 6
Words 225 - 256	Minor Frame I + 7
Words 257 - 288	Minor Frame I + 8
Words 289 - 320	Minor Frame I + 9
Words 321 - 352	Minor Frame I + 10
Words 353 - 384	Minor Frame I + 11
Words 385 - 416	Minor Frame I + 12
Words 417 - 448	Minor Frame I + 13
Words 449 - 480	Minor Frame I + 14
Words 481 - 512	Minor Frame I + 15
Words 513 - 516	GMT Time in EBCDIC - GMT DDD:HH:MM:SS
Words 517 - 520	Unassigned
Words 521 - 524	First command verified in the past 1 second (16 minor frames) of the form NNNNTTTTCCCCCCCC, where NNNN = Experiment mnemonic TTTT = GMT time, 32-bit binary CCCCCCCC = 64-bit command
Words 525 - 528	Second Command
Words 529 - 532	Third Command
Words 533 - 536	Fourth Command
Words 537 - 540	Fifth Command
Words 541 - 544	Sixth Command
Words 545 - 548	Seventh Command
Words 549 - 552	Eighth Command
Words 553 - 556	Ninth Command
Words 557 - 560	Tenth Command verified in the past second

Figure 13. Format of Each Record on the History Tape

7. Command Memory Dump

Certain telemetry words in the minor frame are stripped out by TMD and stored in the array CMEMDUMP. These words correspond to the command memory dump transmission down the 7.2KB line. The 46 words are allocated as follows:

1.	TM(1)	SYNC	2.	TM(2)	SYNC	3.	TM(3)	SYNC
4.	TM(4)	STATUS	5.	TM(5)	STATUS	6.	TM(7)	UVNO
7.	TM(8)	UVNO	8.	TM(15)	RPA	9.	TM(16)	BIMS
10.	TM(17)	S/C	11.	TM(18)	S/C	12.	TM(19)	MIMS
13.	TM(20)	EUVS	14.	TM(27)	NATE	15.	TM(28)	NATE
16.	TM(37)	SUBCOM	17.	TM(39)	PES	18.	TM(40)	PES
19.	TM(47)	VAE	20.	TM(48)	VAE	21.	TM(51)	RPA
22.	TM(52)	RPA	23.	TM(59)	NATE	24.	TM(60)	NATE
25.	TM(65)	SUBCOM	26.	TM(66)	SUBCOM	27.	TM(67)	SUBCOM
28.	TM(68)	SUBCOM	29.	TM(71)	TAL	30.	TM(72)	UVNO
31.	TM(79)	RPA	32.	TM(80)	BIMS	33.	TM(81)	S/C
34.	TM(82)	S/C	35.	TM(83)	MIMS	36.	TM(84)	MIMS
37.	TM(91)	NATE	38.	TM(92)	NATE	39.	TM(103)	PES
40.	TM(104)	PES	41.	TM(111)	VAE	42.	TM(112)	VAE
43.	TM(115)	RPA	44.	TM(116)	NACE	45.	TM(123)	NATE
46.	TM(124)	NATE						

IV. COMMAND PROCESSING SECTION - CMD

CMD generates one or more commands from command mnemonics and controls their transmission to the spacecraft or experiments. Commands may be transmitted singly or in groups, and in one of three verification modes. Commands can be submitted to the system through the card reader, teletype, RAD (via a Procedure), schedule tape, or individually through any of the CRT keyboards—providing the appropriate *ENABLE or *DISABLE statements have been issued by the system operator at CRT #1. (If a command is identified as being "dangerous", a message is displayed to the system operator, and the system enters a "HOLD" state during which time no new processing tasks can be

initiated; no dangerous commands are sent without operator approval, which is furnished by means of the *GO or *DELETE directive.) (See Figure 14.)

1. Verification Modes

MODE 0 — No command verification. Commands are transmitted as soon as they are composed, with no concern for verification of previous commands.

MODE 1 — Commands will be issued from the ground at the maximum rate of 16 per second until the command output buffer is empty. This buffer holds up to 64 commands. Commands which the spacecraft fails to verify will be flagged and re-transmitted after all commands within the command output buffer have been issued. This procedure will be repeated until all commands have been verified or until a specified (*VERIFY directive) maximum number of retries have failed. If the latter occurs, a message will be output to a CRT listing those commands which have failed the specified maximum number of retries.

MODE 2 — Command N+1 will not be issued until Command N has been transmitted and verified. Again, after a specified (*VERIFY) maximum number of re-transmissions have failed, a message will be displayed in the CRTs and the system operator can either force another attempt at transmission or delete the failed command, using the *DELETE directive. The operator may also switch to a different commanding mode.

2. Commanding Sequence

When a command statement (beginning with the symbol /) is submitted to AECS, the following sequence occurs:

1. MSGINTRP separates the parameters in the command mnemonic string.

2. DIRLOAD loads the specified command decoding program, providing that no other program is currently executing.

3. The command program beings execution and builds portions of the final command through a series of table look-up operations.

4. The command program transfers control to COMMAND for final processing, transmission, and verification.

When COMMAND receives control, it performs the following steps:

1. Checks the validity of the command mnemonic in the case of experiment commands.

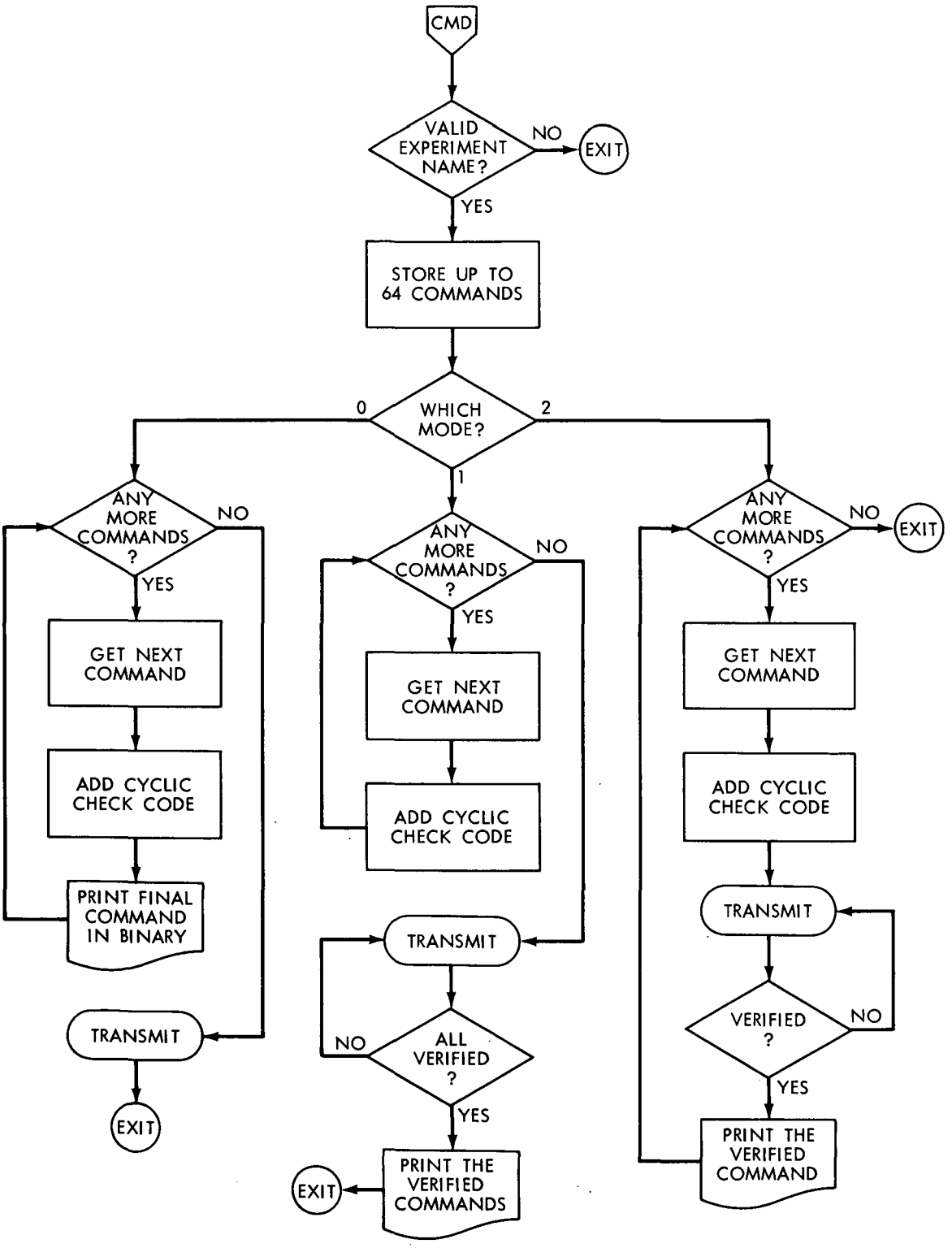


Figure 14. Basic Flow of CMD Program

2. Checks the eligibility of the device issuing the command. If an *ENABLE directive was not issued to allow the unit to generate commands, an error message is displayed and the command is ignored; control returns to the command decoding program.

3. Picks up all of the commands (up to 64) to be sent and transfers them to the command output buffer.

4. Checks the number of transmissions allowed and the command verification mode. Both parameters are specified by the *VERIFY directive.

5. Constructs and inserts the cyclic check code bits into the 64-bit command word, then transmits the command. The cyclic check code is a BCH code whose generator polynomial is

$$g(X) = (X^6 + X + 1)\,(X + 1)$$

   Actual I/O transmission of a string of one or more 64-bit commands requires preceding the string by a 64-bit "preamble" for bit synchronization. This preamble contains 63 zeros followed by a one.

6. If command verification was specified, COMMAND waits until the transmitted commands have been verified, then prints the verified command on the Event Printer in binary (0 and 1) form, together with the GMT time of verification. Failure of commands to verify after a specified number of re-transmissions causes a HOLD condition in the system and the display of error messages.

7. If all commands in the command output buffer have been sent, COMMAND returns control to the command decoding program.

3. Command Programs

A command program is an overlay program within AECS designed to interpret the English-language mnemonics established for an experiment and construct the basic fields of the 64-bit command. Each experiment is uniquely associated with a command program. The mnemonics are used to simplify the commanding process, and correspond to one or more bit fields in the command word. When these fields have been constructed, the command program transfers control to CMD, which then transmits the command.

A typical command has the form

/NNNN,P1,P2,P3,,,,,.

where

NNNN - is the experiment name (3 or 4 characters)

Pi - are mnemonics which are to be interpreted.

4. Command Word Format

The actual format of the 64-bit command word is as follows:

Bits 1-7 - X'EO' - Satellite address code

Bit 8 - Spare bit = 0

Bits 9, 10 - Decoder select code; 01 = decoder 1, 10 = decoder 2 and 00 or 11 signified neither decoder

Bits 11, 12 - Data bus control code; 00, 01, 10 - no minor mode data, 11 signifies minor mode data is present

Bit 13 - Spare = 0

Bit 14 - Spare = 0

Bit 15 - Spare = 0

Bits 16-24 - 9 bit OP CODE

Bits 25-56 - 32 bit MINOR MODE DATA; these bits are all 0's if minor mode data is not included

Bit 57 - Spare = 0

Bits 58-64 - Cyclic code check bits

It should be noted that for all command transmissions, whether for individual 64-bit commands, or a group of such commands, the command string is always preceeded by a special "frame sync" word containing 63 logic 0 bits followed by a logic 1 bit. (See Figure 15 for Command Word Structure.)

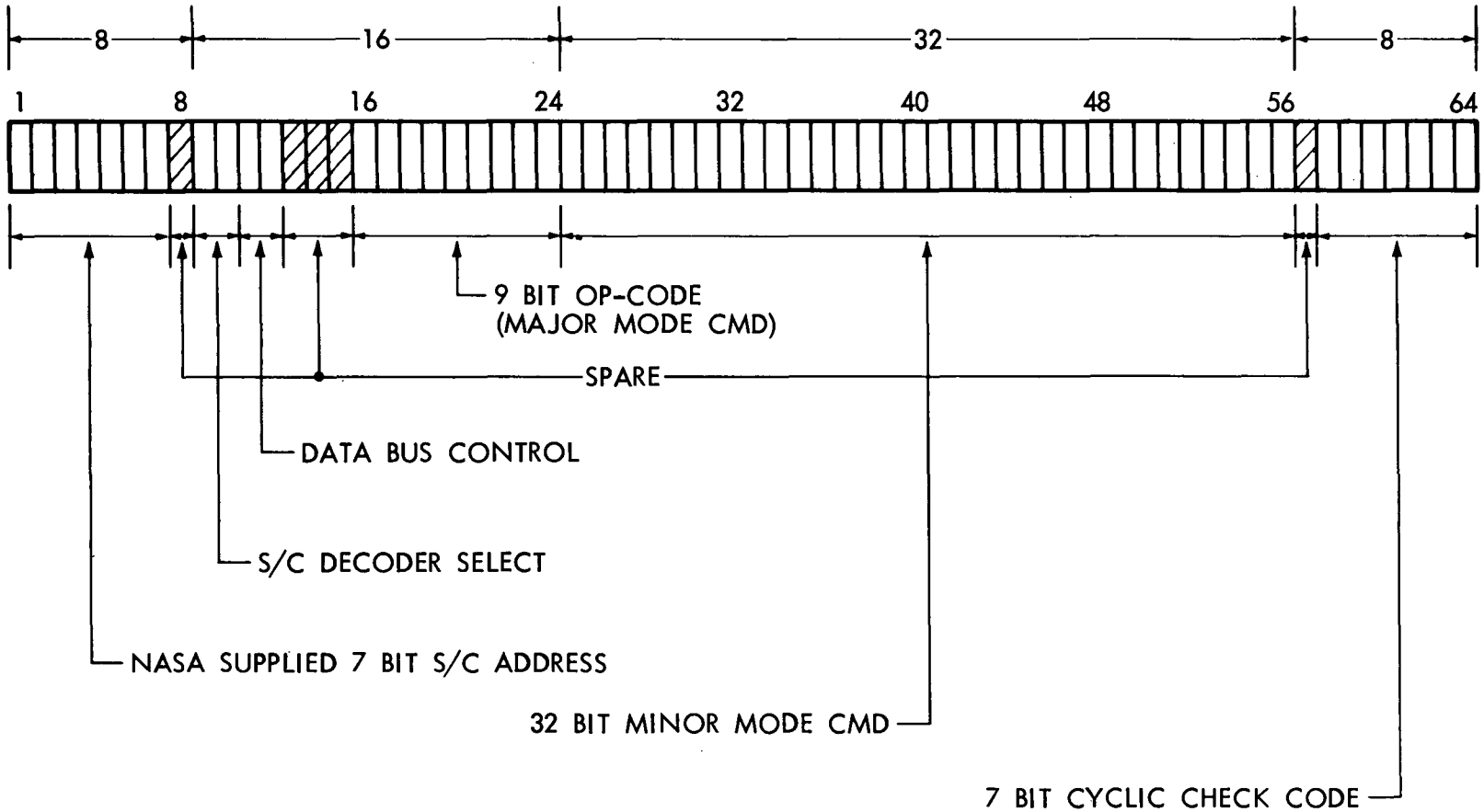


Figure 15. Command Word Structure - Real-Time Command

V. EDITOR

EDITOR is a real-time program that creates Procedures and Schedules. A Schedule is a chain of directives, commands, and Procedure calls combined together, sequenced, and given a name for later reference. Procedures are groups of commands and directives that are related to a specific function. A Schedule is put onto a magnetic tape, Procedures onto the RAD (D3 area). Procedures are also referred to as "PROCS".

To clarify the difference between a Schedule and a Procedure, an analogy can be drawn between EDITOR and the FORTRAN compiler; a schedule corresponds to a main program, a procedure to a subroutine, and the directives and commands to individual instructions.

Input to EDITOR consists of (1) EDITOR Control Cards which determine the activity of EDITOR, and (2) Procedure and schedule statements; in the case of Procedures, the statements are directives and commands, and for a Schedule, the statements are directives, commands, and Procedures as well. (See Figure 16.)

1. Procedure Statements

Procedure statements have the following general format:

```
S  NNNNNNNN,PARM1,PARM2,,,,,,,.  COMMENTS
```

where

S - The symbol denoting a directive (*), a command (/), a utility function (?), a simulator directive ($), or a Procedure call (&). The symbol must be in column 10.

NNNNNNNN - The name of the directive, command, etc., being called. In the case of a Procedure call, NNNNNNNNN is the name of the Procedure.

PARMi - parameters

COMMENTS - descriptive or otherwise useful information which will be displayed on the CRTs and Event Printer.

Columns 1 through 9 and 73 through 80 are used by EDITOR and must be left blank, or the contents will be ignored.

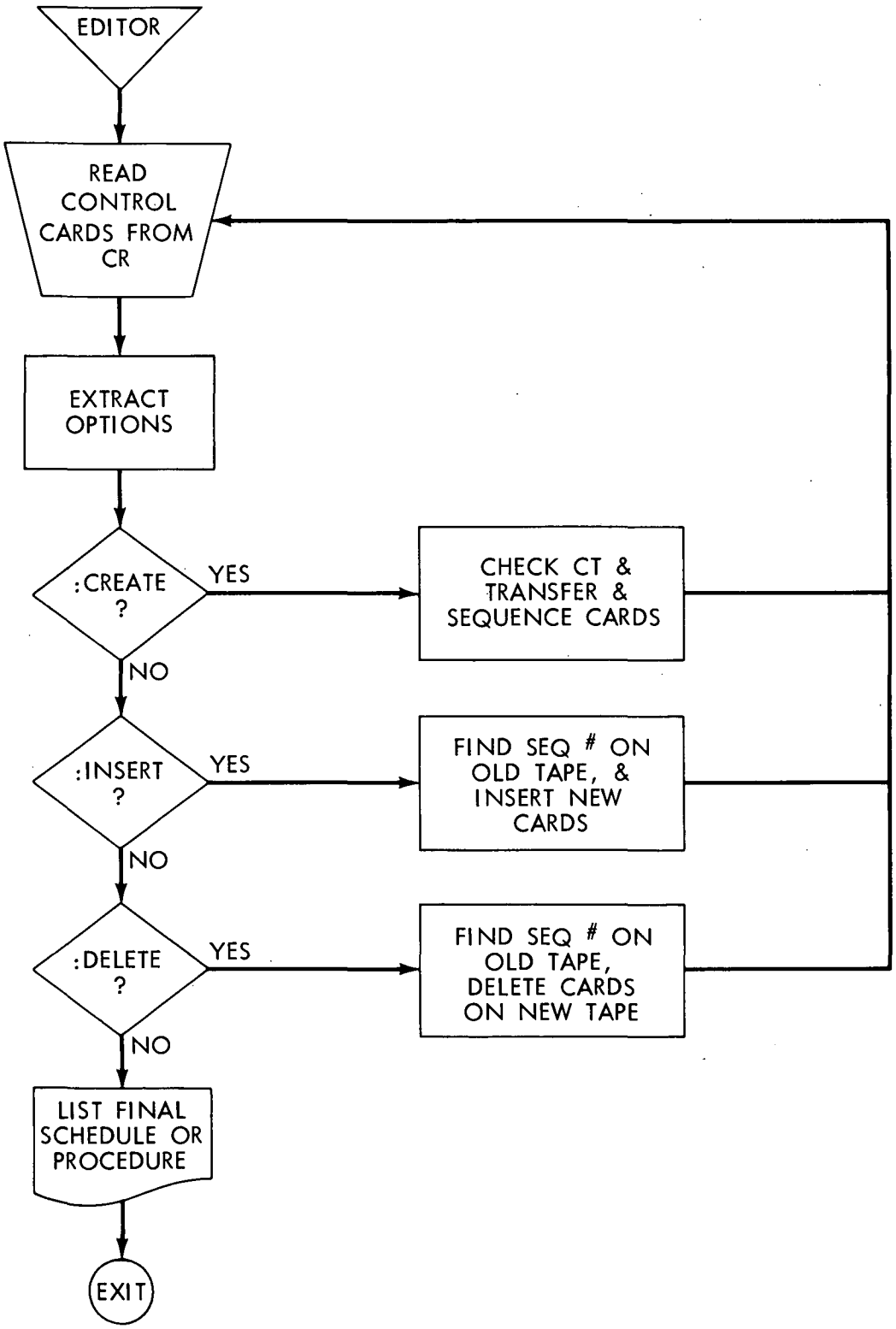


Figure 16. Basic Logic Flow of EDITOR

EDITOR sequences all statements and copies them onto the RAD for a Procedure or onto the CT (Control Tape) for a Schedule. The sequence numbers are of the form NNNNxxxx, where NNNN is the first four characters of the name used, and xxxx is a decimal sequence number.

2. EDITOR Control Cards

All EDITOR control cards are identified by a colon (:) in the first column of the card. The comment field of each EDITOR card begins in column 45.

(1) :CREATE,XXXX,NAME

XXXX - { SCHD - to create a schedule
       { PROC - to create a procedure

NAME - The name of the schedule or procedure being created.

When EDITOR reads a :CREATE card, it assumes that all cards in the card reader between the :CREATE and the next :END card are the statements of the schedule or procedure being created. EDITOR sequences these statements, and writes them to the RAD if a procedure is being generated or to the CT tape if a schedule is being made.

(2) :INSERT,NAMESEQ#

NAMESEQ# - The first 8 characters of the sequence number of the schedule statement AFTER which cards are to be inserted. A special NAMESEQ# - NAMEBEFO - is used to insert statements BEFORE the first schedule statement.

When the :INSERT card is read, EDITOR copies the schedule statements on the CT tape up to and including the schedule statement NAMESEQ#. EDITOR inserts into the schedule the statements between the :INSERT card and the next EDITOR control card.

(3) :DELETE,NAMEFIRS,NAMELAST

NAMEFIRS - The first 8 characters of the sequence number of the first schedule statement to be deleted.

NAMELAST - The first 8 characters of the sequence number of the last statement to be deleted.

When EDITOR reads the :DELETE card, it copies the schedule statements on the CT tape up to but not including NAMEFIRS, and skips the schedule statements NAMEFIRS through NAMELAST.

The :INSERT and :DELETE cards can be stacked as long as the numeric portion of the sequence numbers are ascending. The entire modification stack must be terminated with a :END card which causes EDITOR to copy the remainder of the schedule.

(4) :END

The :END card is used to separate stacks and delimit procedure and schedules. Two :END cards terminate EDITOR.

3. Directions for Using EDITOR

EDITOR is a real-time program that can be run from the card reader or the OC device. To run EDITOR from the card reader the user must

(a) Place the RBM control cards and EDITOR commands into the card reader.

(b) Press the INTERRUPT button on the SIGMA 5 console.

(c) Type in C on the OC device. When PAUSE FGC prints on the OC device, key in FGC.

EDITOR will now read all the cards in the reader up to two :END cards, and produce a listing of the schedule or procedures being made.

To run EDITOR from the OC device, the user must

(a) Place only the EDITOR control cards and procedure (schedule) statements in the card reader.

(b) Press the INTERRUPT button on the SIGMA 5 console and type in RUN EDITOR on the OC device.

EDITOR uses at most two tape drives. The assignment of the tapes is as follows

CT (9TA80) - Output tape if creating a schedule. If EDITOR is modifying an existing schedule CT will be the OLD schedule.

9TB81 - The output tape when modifying a schedule. The new schedule will be on this unit.

4. Examples

(1) Creating a procedure using RBM control cards

Below are the RBM and EDITOR control cards needed to generate a procedure called TURNON. The RBM control cards begin with a !. The executable statements of the procedure are between the :CREATE and :END cards. The entire deck is placed in the card reader and executed from the C device.

```
!JOB
!PAUSE FOC
!RUN FP,EDITOR
:CREATE,PROC,TURNON
         *START.                          SIMULATORS
         *START.                          TELEMETRY
         *HOLD.                           UNTIL GET SYNC
         *HTSTART,3.                      RECORD DATA ON UNIT 3
         *CHARTON,1,TM(37).               STRIP CHART FOR SUBCOM COUNT
         *HOLD.
:END
:END
!FIN
```

(2) Creating several procedures using RBM TYC control

Several procedures may be stacked. Each procedure begins with a :CREATE card and ends with an :END card. Another :END card is added at the end of the card deck. The entire deck is placed in the card reader and executed via a "RUN EDITOR" key-in on the teletype.

```
:CREATE,PROC,UTILEXER          :END
         ?X,5001,5100,5050.    :CREATE,PROC,SENDCOMD
         ?D,5001,5108.                  /BIMS,VSMRC,005.
         ?R,5001=1.                     /BIMS,ANLG3.
         ?C,5001,5002,1.                /BIMS,ANLG4.
         ?C,5001,5003,2.                /BIMS,ANLG5.
         ?C,5001,5005,4.       :END
         ?C,5001,5100,8.
         ?C,5100,5050,8.
         ?%
         ?D,5000,5108.
```

```
:CREATE,PROC,COEFASGN                          ASSIGN COEFICIENTS
        *CONVCOEF,TM(37),0,1.
        *CONVCOEF,TM(9),D'1.',D'0.0',D'20.0'.
        *CONVCOEF,TM(11),D'1.0',D'0.0',D'1.0'.        1.X**2
:END
:CREATE,PROC,TURNOFF
        *HTSTOP.
        *STOP.                                 STOP TELEMETRY
        $STOPP.                                SIMULATORS
        *CHARTOFF.                             CHARTS
:END
:END
```

(3) Creating a schedule

Below are the EDITOR control cards to generate a schedule called DEMO. DEMO calls the previously created procedures UTLEXER, COEFASGN, SENDCOMD, TURNON, and TURNOFF. EDITOR expands these procedures and puts their code in line on the schedule tape.

```
:CREATE,SCHD,DEMO
        &UTILEXER.
        *HOLD.
        &TURNON.                       EVERYTHING
        $PRA.                          PRINT ACTIVE ARRAY &HOLD.
        *HOLD.
        *LIMITS,TM(37),2,128.
        *SKIP,5.
        *LIMITS,TM(37),3,128.
        $MINOR,9,X'10',X'11',X'12'.
        *WAIT,10.
        $SWITCH.
        *WAIT,10.
        $PRA.
        *WAIT,80.                      8 SECOND WAIT.
        ?L,500A,502A,504A,506A,508A,50AA,50CA,50EA.
        *HOLD.
        &COEFASGN.                     SET COEFICIENTS FOR CONVERSIONS.
        *VAESNAP.                      SNAPSHOT FOR VAE
        *PRTCOEFS.
        *PRTLIMITS.
        *WAITA,TM(37),100,300.
        *LRV,PRT,TM(37).
        *VERIFY,0.                     NO VERIFICATIO .
        &SENDCOMD.                     SEND
        *TESTD,TM(37),1,1,-1,1.         SET AGAIN IF ODD FRAME.
        *IF,TM(37)-64.,1,2,2.
        *FAIL,GO.                      SUBCOM LT 64
        *LRV,CRT,TM(37),TM(1),TM(2).
        *PRCRT,1.
        *HOLD.
```

```
          *WAITD,TM(37),X'FF',X'40'.  EAIT UNTIL FRAME 64
          *TESTA,TM(37),128,132,1,7.
          *LRV,PRT,TM(37).
          *STATUS,0.
          *IF,TM(37)=68.,1,2,3.
          *LIMOFF,TM(37).
          *FAIL,HOLD.                    ODD FRAME
          *SKIP,2.
          &SENDCOMD.
          *TESTD,TM(37),1,0,=30,1.
          *LRV,CRT,TM(37).
          ?D,5000,5400.
          *WAIT,90.
          *LIMOFF,TM(37).
          ?M,0,'TURNING OFF EVERYTHING'.
          &TURNOFF.
          ?M,0,'ITS BEEN MY PLEASURE.'.
:END
:END
```

VI. THE UTILITY SECTION (UTL)

UTL is a resident program consisting of several control instructions designed to facilitate the debugging of real-time programs. The instructions have a standard format, and are keyed-in through any of the CRT keyboards. These instructions are:

?L,X1,X2,X3, , , X20. — List on the requesting CRT screen the contents of the specified computer core locations. The locations are specified as hexadecimal values. Up to 20 locations can be listed.

?X,X1,X2,X3, , , X20. — Like L above, except that the CRT display is updated every 100 milliseconds with the latest contents of the specified locations.

?% — Cancel the currently operating utility function.

?D,X1,X2. — Dump the contents of core between locations X1 and X2 onto the Event Printer.

?R,A1=X1,A2=X2, , , A10=X10. — Replace the contents of locations Ai by the hex value Xi. Up to 10 replacement parameters may be used.

?C,X1,X2,N. — Copy N Sigma 5 words from the area beginning at location X1 in core to the area beginning at location X2. All parameters are in hex.

?M,N,'message' — Transmit the message to CRT #N. If N=0, the message is broadcast to all CRTs. The message appears on line 22 of the screen.

In addition to the above, a stand-alone dump routine called "DUMP" can be loaded into upper core and executed as needed. DUMP uses no system parameters and gives a complete or partial dump of core contents. If the console "I/O RESET" button is depressed before initiating DUMP, the RBM system will be re-loaded when the dump is terminated.

VII. PROGRAMMING DETAILS

This section contains information essential for designing and writing programs for the AECS system. This information will be expanded in the near future and become part of the AECS Programmers' Manual.

D

1. AECS Structure

AECS is a simple overlay program with two levels of overlay segments. The root (resident portion) contains all the ECS, TMD, CMD, UTL, SIM, and related programs. The first overlay level is used for directives, command programs, snapshots, and other special purpose data processing programs. The second level overlay is for CRT Page display programs. The first overlay operates at the X'6E' interrupt priority level, and the second operates at the X'6D' level. (See Figure 17.)

New programs such as directives or pages can easily be added to AECS. Consider, for example, the addition of a new command program called XXXXXXXX; the following control cards should be used:

```
!JOB    XXXXXXXX - COMMAND PROGRAM OF NNNN EXPERIMENT
!ATTEND
!PAUSE SYC - ALLOWS RAD ACCESS
!RADEDIT
:DELETE (FILE,FP,XXXXXXXX)
:ALLOT (FILE,FP,XXXXXXXX), (RSIZE,30), (FORMAT,B), (FSIZE,10)
!ASSIGN (M:GO,FP,XXXXXXXX)
!MACRSYM SI,LO,GO
        . . .
        . . .
        . . .
        END
!FIN
```

The above sequence of control cards allots space on the RAD,FP area, then compiles the program with the object deck being put on the new file XXXXXXXX. What remains is simply to link this segment onto the rest of the AECS overlay program. The job control cards for this operation are as follows:

```
!JOB AECS OVERLAY PROGRAM
!ATTEND
!PAUSE SFC
!RADEDIT
:DELETE (FILE,FP,AECS)
:ALLOT (FILE,FP,AECS), (FSIZE,400)
!OLOAD (FILE,FP,AECS), (FORE,6000), (PUBLIB,AELIB), (TASKS,16)
:ROOT (FILE,FP,ECS), (FILE,FP,TMD), . . . . .
:SEG (LINK,1), (FILE,FP,PAGESNAP) DIRECTIVE OVERLAY
```

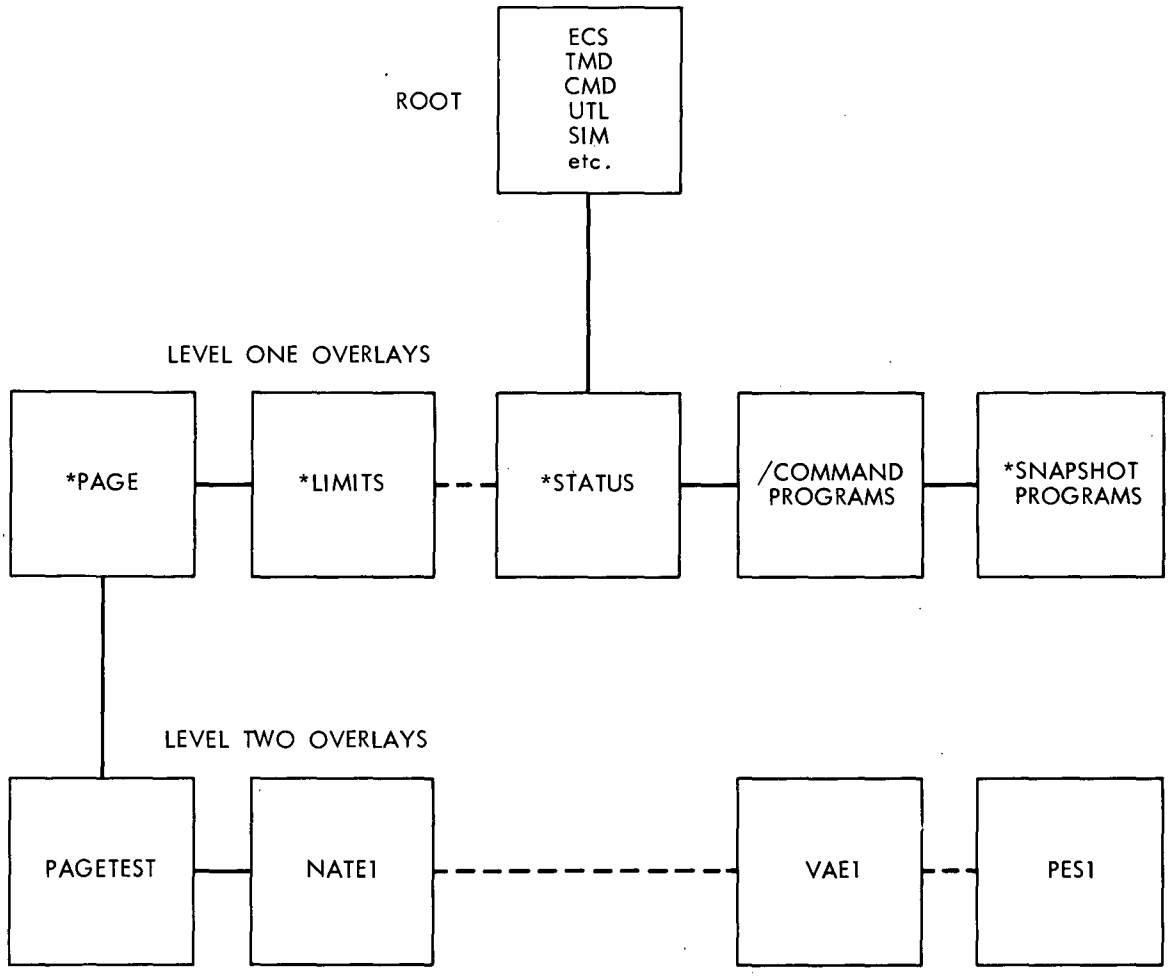


Figure 17. AECS Overlay Structure

```
:SEG (LINK,40,ONTO,1), (FILE,FP,PAGETEST) PAGE OVERLAY
 . . . .
 . . . .
:SEG (LINK,nn), (FILE,FP,XXXXXXXX) NEW PROGRAM
 . . . .
 . . . .
!FIN
```

At this point, the new program XXXXXXXX has been fully integrated into the AECS system. The system operator now can issue the call

RUN AECS

to execute the entire AECS program.

2. Directives

The following procedures are recommended for the programming of system directives:

(1) Directives are overlay segments which are called in for execution by operator key-ins, card inputs, Schedule statements, etc. As segments, the length of each directive program should not exceed 2,048 decimal words.

(2) The first word of a directive program (relative location 00000) should be the entry point — first executable instruction — of the program. ECS transfers control to this address via a

```
BAL,11   *PROGRAM
```

Control should be returned to ECS by an instruction analogous to

```
B   *11
```

(3) If there is any loop within the directive program which could last more than one second, for any reason, the variable KILLFLAG should be checked as part of the loop. That is, the following code should be added to the loop:

```
LW,2   KILLFLAG
BNEZ   EXIT - DIRECTIVE PROGRAM EXIT
```

KILLFLAG is normally 0, until set to 1 by the Cancel (%) code.

(4) If a directive program contains data which is to be updated dynamically, the first executable instruction after entry should be

```
MTW,1   UPDATE
```

where UPDATE is the flag which signals dynamic updates. If this UPDATE flag is set, the directive program is re-entered by ECS every second. The UPDATE flag is cancelled when a new directive is loaded into the system, or when the Cancel (%) code is used.

3. Page Programs

Pages are second level overlay programs. Each Page program is brought into core every 3 seconds and receives control at interrupt priority level X'6D'. The program should reference three flags, PAGEONE, PAGEUNIT, and PAGEDONE.

PAGEONE is a word (32-bit) which has an initial value of zero. It is provided so that the Page program may have some way of knowing when its first execution occurred. That is, it is up to each Page program to reset PAGEONE to some non-zero value for subsequent executions. Then a typical execution procedure could be as follows:

(1) During execution, PAGEONE is checked. If 0, this must be the first time through, so display all the static titles and header information. Also, set PAGEONE to 1 (or any other number).

(2) During execution, PAGEONE is checked. If non-zero, bypass the sending of title data and only display the dynamic telemetry data.

The advantage of this kind of operation is that I/O time to the CRT devices is thereby minimized; I/O transfer occurs at the rate of 1200 characters per second.

PAGEUNIT is another 32-bit word containing 00N00000, where N is the CRT number/screen on which this Page is to be displayed. This word needs only to be merged with the CAL2,0 instruction for I/O to the appropriate CRT. (See Figures 18, 19, and 20.)

PAGEDONE is a resident 32-bit word which is not modified by the individual Page overlays, but which is referenced in the very last CAL2 call as the end action flag. This flag is used by the PAGE driver only.

4. Cathode Ray Tubes

The cathode ray tubes (XEROX System Keyboard Displays) on the AE system may be accessed through the statement

```
CAL2,N   CRT:FPT
B        REJECT
```

(Note: The flag CRTBUSY may be checked first; it should be 0.)

where

N - CRT unit address = 0 for all units
= 1 for CRT #1 (System CRT)
= 2 for CRT #2
= 3 for CRT #3
= 4 for CRT #4 (RCA only) etc.

CRT:FPT - CRT function parameter table

REJECT - address of a CRT-busy routine. This address may simply be the branch, $-1

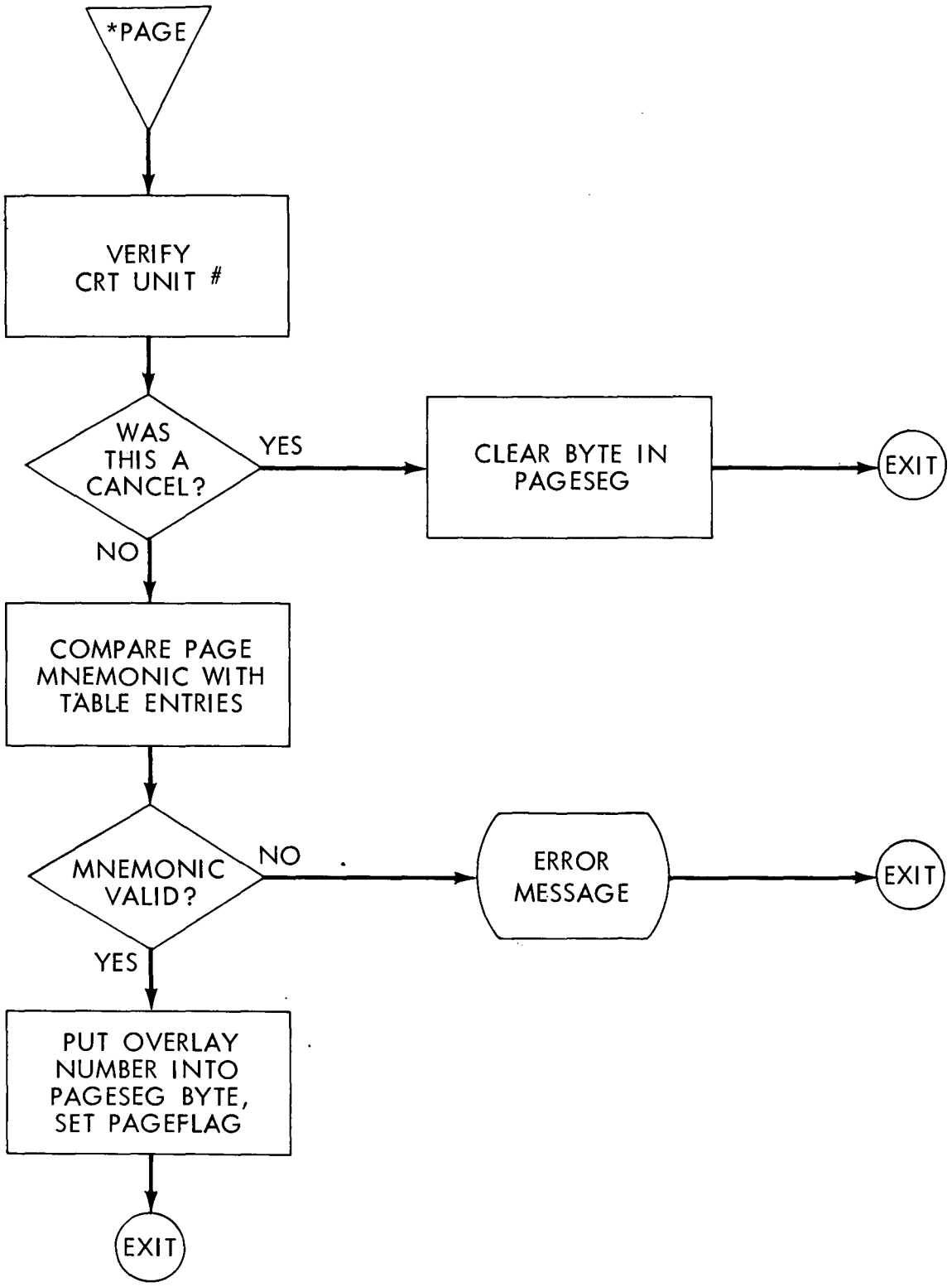


Figure 18. Flow of *PAGE Directive

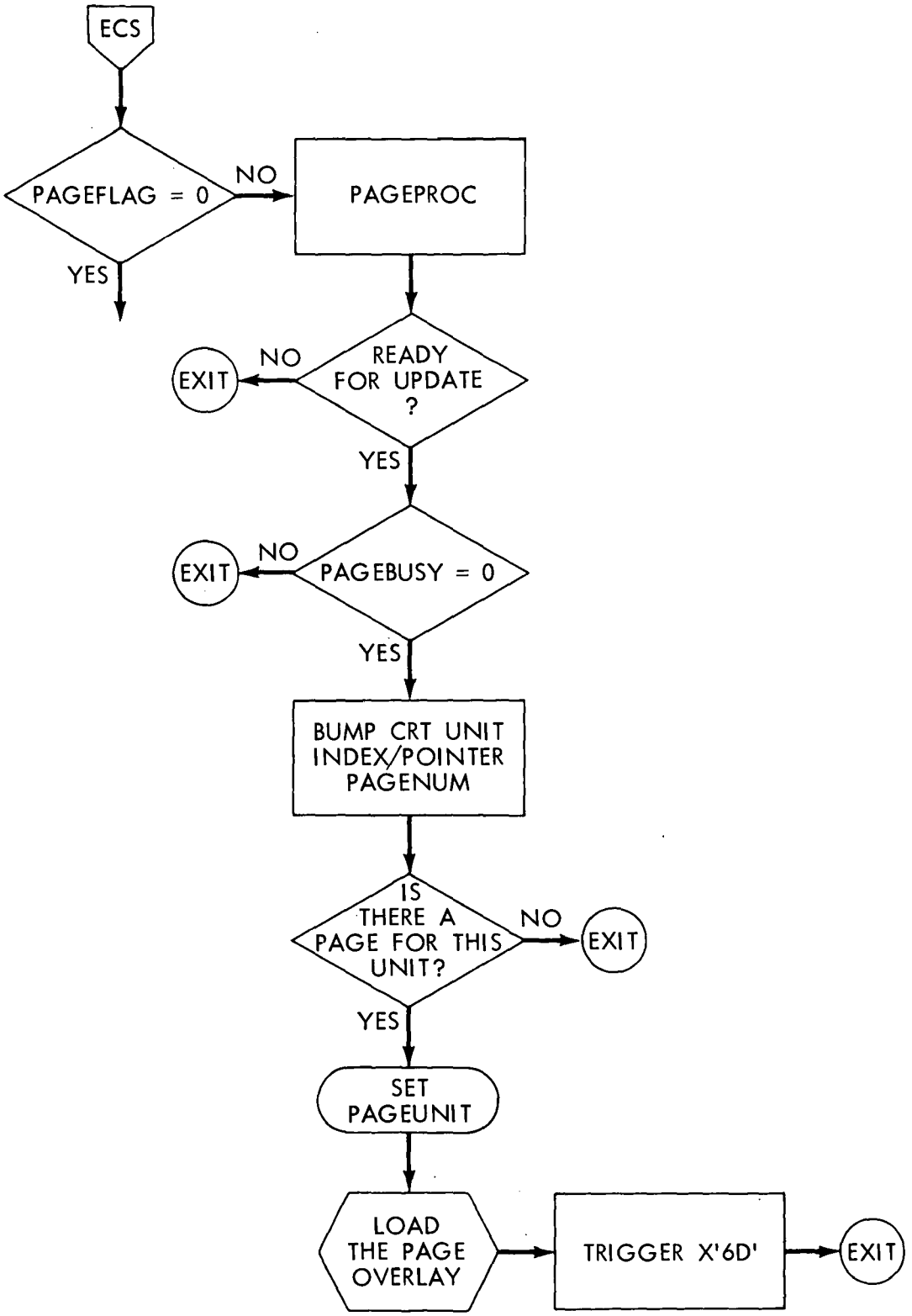


Figure 19. Flow of Resident PAGE Processor

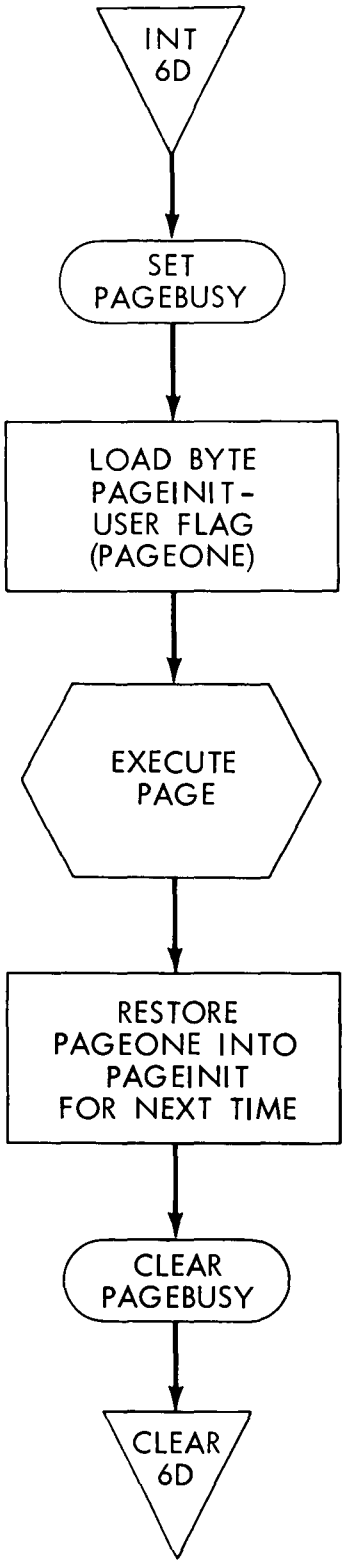


Figure 20. Flow of PAGE Interrupt/ Execution Routine

The CRT:FPT function parameter table is set up as follows:

```
            BOUND       8
CRT:FPT     GEN,8,8,16  NBYTES,CODE,TEXT
            GEN,8,8,16  LINE,COLUMN,FLAG
```

where

NBYTES — number of characters to be displayed (from 00 to 254). A full screen (17 x 80) can be displayed by specifying NBYTES = X'FF' and LINE=COLUMN=1

CODE — 0 for normal write to screen

1 for write with blinking characters

2 for alarm and write

TEXT — address of text message to be displayed (TEXT is expected in EBCDIC form)

LINE — line number of CRT at which message will begin

COLUMN — column at which message will begin

FLAG — set to 1 by AECS when this message has been sent out. This flag must be PAGEDONE for Page overlay programs.

5. Event Printer

The Event Printer (LPA02) is accessed through the following statement:

```
CAL3,1    EP:FPT
B         $-1         Reject
```

or

```
LW,2      EPBUSY      Busy flag
BNEZ      $-1         Wait loop
CAL3,1    EP:FPT
B         $-1
```

where

EPBUSY — Event Printer busy flag

EP:FPT — Function parameter table, which is set up as follows:

```
            BOUND       8
EP:FPT      GEN,8,8,16  NBYTES,CODE,TEXT
            GEN,32      GMTBINRY
```

where

NBYTES — number of characters to be printed (maximum 88)

CODE — printer format code (e.g., X'40' of X'00' for normal print, X'F1' for top of page before printing, etc.)

TEXT — address of text message to be printed

GMTBINRY — contains 0 or the 32-bit binary GMT time which must be inserted here by the user

The resulting printed message appears left-adjusted on the Event Printer, followed by status information including the GMT time, orbit number, identification of statement generation device, etc.

6. Snapshot Printer

The Snapshot Printer (LPA02 or LPB02) may be accessed through the CAL3 statement

```
CAL3,2    SN:FPT
B         REJECT
```

or

```
LW,2      SNBUSY
BNEZ      $-1          Wait until ready
CAL3,2    SN:FPT
B         $-1
```

where

SN:FPT is the address of the Snapshot Printer function parameter table, and SNbusy is the printer busy flag, which is 0 if the printer is not

busy. The function parameter table is similar to that for the Event Printer, except for the GMT word:

```
            BOUND       8
SN:FPT      GEN,8,8,16  NBYTES,CODE,TEXT
            GEN,8,8,16  0,0,ENDACT
```

where

ENDACT is the address of an end-action routine, or zero. The end-action routine is entered at the X'6C' priority level. The Snapshot Printer line contains 132 characters.

7. Command Calls

In order to transmit commands to the spacecraft/simulator, the following procedure is used:

(1) Put NNOOAAAA into Register 2, where NN=number of 64-bit commands to be sent, and AAAA=address of the basic command list, which must begin on a doubleword boundary. Each entry in the basic command list is

MMOOOCCC DDDDDDDD

where

MM — Minor Mode X'03' or Major Mode X'00', X'01' or X'02'

CCC — 9-bit op code

DDDDDDDD — 32-bit Minor Mode Data

(2) Execute the instruction BAL,15 COMMAND

(3) On return from COMMAND, Register 2 contains the number of commands which have been verified.

8. LRVINDEX

The subroutine LRVINDEX may be used to obtain the address of a desired telemetry word from the LRV table. Its usage is as follows:

(1) Load Register 2 with the telemetry word index (1 through 128).

(2) Load Register 3 with the sub com step index, if any.

(3) Execute the instruction BAL,15 LRVINDEX

(4) On return, Register 2 contains the byte offset desired

(5) On return, Register 3 contains an error code, if non-zero.

(6) Fetch the desired telemetry value by the instruction

LB,2 LRV,2

9. CONVERT

The subroutine CONVERT can be used to convert a specified telemetry word into engineering units, assuming that the appropriate conversion has been submitted to the system. The calling sequence is as follows:

(1) Load Register 2 with the telemetry word index

(2) Load Register 3 with the sub com step number, if any

(3) BAL,15 CONVERT

(4) On return, Register 2 contains the floating point value

(5) For normal returns, Register 3 = 0. Otherwise, Register 3 contains an error code and Register 2 contains the unconverted telemetry word (byte).

VIII. SYSTEM GENERATION

System generation is the process by which a software system is created on a computer in order to enable general usage of that computer. As far as AECS is concerned, system generation involves two steps: generating the XEROX RBM standard monitor, and, generating the AECS overlay program.

1. RBM System Generation

The following basic steps are involved in a typical RBM system generation — "SYSGEN". More detailed information is available in the RBM Reference Manual.

(1) Mount the following tapes on the specified tape units:

9TA80 — Binary Input (BI) of RBM, supplied by XDS

9TB80 — FORTRAN IV/H LIBRARY

9TB81 — FORTRAN IV/H COMPILER

(2) Place the SYSGEN deck in the card reader. A listing of the AE SYSGEN deck is shown in Figure 21.

(3) Set the Sigma 5 Unit Address to '080', and LOAD and RUN the computer.

(4) In response to the "INPUT DEVICE" message on the teletype, type in "9TA80".

(5) In response to the teletype message "IN/OUT DEVICES?", type in the response ":SYSGEN (IN,CRA03), (OUT,LPB02)". The SYSGEN deck will then be read in from the card reader. At this point, other messages for mounting or dismounting tapes will be displayed.

2. RBM Modifications

In order to perform I/O in AECS with the CRTs, Event and Snapshot printers, the RBM X'5C' routine has been modified. The modifications to this portion of the system are listed in Figure 22.

3. AECS Overlay Program

After the system is generated, the AECS overlay deck may be run. A listing of this deck is shown in Figure 23.

```
:MONITOR (CORE,48),(LPP,52)
:RESERVE (RSDF,56),(FFPOOL,10),(FRGD,20),(BRAD,7),(FIOQ,6),(FRAD,56);
:(BIOQ,8),(FMBOX,64),(MPATCH,50)
:DEVICE TYA01
:DEVICE CRA03
:DEVICE LPA02
:DEVICE (CPA04,LP)
:DEVICE 9TA80
:DEVICE 9TA81
:DEVICE LPB02
:DEVICE 9TB80
:DEVICE 9TB81
:DEVICE 9TB82
:DEVICE 9TB83
:DEVICE (DC8F0,S),(ENTRACK,511),(NSPT,16),(NWPS,90),(SP,70),(FP,157);
:(BP,75),(CK,10),(XA,10),(BT,90),(D1,30,F),(D2,20,B),(D3,50,F)
:STDLB (C,CRA03),(OC,TYA01),(LO,LPB02),(LL,LO),(DO,LO),(CO,9TA80);
:(BO,9TB80),(CI,9TA81),(SI,C),(BI,C),(SO,9TA80),(HT,9TB82),(CT,9TA80)
:CTINT (CT,6F),(HI,6F)
:ALLOBT (GO,15),(OV,30)
:SYSLD (IN,9TA80),(V,AECS),(MAP,LPB02),ALL
!JOB
!ATTEND
!MESSAGE  9TA80 - NEW BINARY INPUT TAPE
!MESSAGE  9TB81 - FORTRANH COMPILER
!MESSAGE  9TB80 - FORTRANH LIBRARIES
!STDLB (BI,9TA80)
!ALLOBT (FILE,GO),(FSIZE,0),SAVE
!ALLOBT (FILE,X1),(FORMAT,B),(FSIZE,300),(RSIZE,30)
!MESSAGE  ALLOTING FILE FOR PROCESSORS
!PAUSE KEYIN SYC
!LOAD MAP,(SEGS,4)
!ROV
:ALLOT (FILE,SP,RADEDIT),(FSIZE,110)
:ALLOT (FILE,SP,OLOAD),(FSIZE,136)
:ALLOT (FILE,SP,MACRSYM),(FSIZE,120)
:ALLOT (FILE,SP,SYMBOL),(FSIZE,44)
:ALLOT (FILE,SP,FORTRANH),(FSIZE,95)
:ALLOT (FILE,SP,MODIR),(FSIZE,12)
:ALLOT (FILE,SP,EBCDIC),(FSIZE,14)
:ALLOT (FILE,SP,DEFREF),(FSIZE,12)
:ALLOT (FILE,SP,MODULE),(FSIZE,800),(FOR,B),(RSIZE,30)
:COPY (FILE,BT,OV),(FILE,SP,RADEDIT)
!ALLOBT (FILE,OV),(FSIZE,0),SAVE
!ALLOBT (FILE,X1),(FORMAT,B),(FSIZE,700),(RSIZE,30)
!MESSAGE LOADING OVERLAY LOADER ONTO RAD
!LOAD (OUT,SP,OLOAD),(SEGS,6),MAP
!ALLOBT (FILE,X1),(FORMAT,B),(FSIZE,800),(RSIZE,30)
!MESSAGE LOADING MACRO-SYMBOL ONTO RAD
!OLOAD LIB,(FILE,SP,MACRSYM),(MAP,ALL)
:ROOT (OPLB,BI,EOD)
:SEG (LINK,1),(OPLB,BI,EOD)
:SEG (LINK,2),(OPLB,BI,EOD)
:SEG (LINK,3),(OPLB,BI,EOD)
:SEG (LINK,4),(OPLB,BI,EOD)
!REWIND 9TA80
!STDLB (BI,9TB81)
!PAUSE MOUNT FORTRANH COMPILER TAPE ON 9TB81 + KEYIN SYC
!ALLOBT (FILE,OV),(FSIZE,0),SAVE
!ALLOBT (FILE,X1),(FORMAT,B),(FSIZE,900),(RSIZE,30)
!OLOAD LIB,(FILE,SP,FORTRANH),(MAP,ALL)
:ROOT (OPLB,BI,EOD)
!REWIND 9TB81
!PAUSE MOUNT FORTRAN LIBRARIES ON 9TB80
!STDLB (BI,9TB80)
!RADEDIT
:COPY (IN,BI),(LIB,SP)
!REWIND 9TB80
!MESSAGE  MOUNT SCRATCH ON 9TB80
!JOB      CREATE RAD SAVE TAPE
!STDLB (BO,9TB81)
!PAUSE B81 - FIRST SAVE TAPE- SYC
!RADEDIT
:SAVE ALL
:MAP ALL
!JOB
!STDLB (BO,9TB80)
!RADEDIT
:SAVE ALL
```

Figure 21. Listing of AE SYSGEN Deck

```
!JOB
!ATTEND
!ALLOBT (FILE,OV),(FSIZE,0),(SAVE)
!STDLB (BO,D3,RBM)
!STDLB (CI,9TA80)
!REW CI
!MESSAGE UPDATING RBM
!SFIL CI,6
!MACRSYM SI,CI,BO,LO,LU
+4958
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         LH,7      AEUNIT1        GET NUMBER OF AECS SPECIAL DEVICES
         CH,2      AEUNIT1,7      CHECK EACH UNIT
         BE        AECSUNIT       FOUND AECS UNIT
         BDR,7     $-2           LOOK FOR OTHERS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
+4963
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
AEUNIT   EQU       X'10'
AECSUNIT STW,1     AEUNIT,7       SAVE AIO STATUS IN LOCATIONS 10-1F
         LH,2      AEUNIT2,7      LOAD INTERRUPT MARKER
         WD,2      X'1702'        AND TRIGGER THE INTERRUPT
         B         I040           AND GET OUT
*
AEUNIT1  DATA,2    9              9 SPECIAL AECS DEVICES       3/24/72
         DATA,2    X'000'         LPB02 - SNAPSHOT PRINTER
         DATA,2    X'002'         LPA02 - EVENT PRINTER
         DATA,2    X'104'         SKD SCREENS
         DATA,2    X'005'         STRIP CHART RECORDERS
         DATA,2    X'183'         9TB83 - HISTORY TAPE BACKUP
         DATA,2    X'182'         9TB82 - HISTORY TAPE
         DATA,2    X'103'         NARROWBAND PCM TELEMETRY INPUT
         DATA,2    X'106'         COMMAND GENERATION END ACTION
         DATA,2    X'105'         SPACECRAFT SIMULATOR ENDACTION
*
         BOUND     4
AEUNIT2  DATA,2    X'00'     I N T E R R U P T   A S I G N M E N T S
         DATA,2    X'80' - INTERRUPT 68 - ECS IO - SNAPSHOT PRINTER
         DATA,2    X'80' - INTERRUPT 68 - ECS IO-- EVENT PRINTER
         DATA,2    X'80' - INTERRUPT 68 - ECS IO - SKD SCREENS
         DATA,2    X'80' - INTERRUPT 68 - STRIPCHART RECORDER ENDACTION
         DATA,2    X'100' -INTERRUPT 67 - TMD I/O ENDACTION
         DATA,2    X'100' -INTERRUPT 67 - TMD I/O ENDACTION
         DATA,2    X'100'  INTERRUPT 67 - PCM TELEMETRY INPUT
         DATA,2    X'8000' - INTERRUPT 60 - COMMAND END ACTION
         DATA,2    X'4000' - INTERRUPT 61 - SIMULATOR
         BOUND     4
+END
```

Figure 22. Modifications to RBM X'5C'

ε

```
!JOB        AECS OVERLAY
!PAU SFC
!ATT
!RADEDIT
:DELETE (FILE,FP,AECS)
:TRUNCATE FP
:SQUEEZE FP
:ALLOT (FILE,FP,AECS),(FSIZE,650),RF
!MESSAGE - RLS AECS, MOUNT 9TB80 FOR RAD DUMP
!OLOAD (FILE,FP,AECS),(FORE,6000),(PUBLIB,AELIB),(TASKS,16),(MAP,ALL)
:ROOT (FILE,FP,ECS),(FILE,FP,DECODE),(FILE,FP,UTILITY),(FILE,FP,TMD);
:(FILE,FP,CRT),(FILE,FP,SIM),(FILE,FP,SCHDPROC),(FILE,FP,COMMAND);
:(FILE,FP,SRENDACT),(FILE,FP,CRTPAGES),(FILE,FP,DIRECTS);
:(FILE,FP,PESCAL);
:(FILE,FP,UVOVFLOW);
:(FILE,FP,VAELIM);
:(FILE,FP,UVNDRCRT);
:(FILE,FP,PESTORE);
:(FILE,FP,NATEDAT);
:(FILE,FP,RPADATA);
:(FILE,FP,ECS:IO),(FILE,FP,TMDSTRIP),(FILE,FP,UVNADIR),(FILE,FP,TM:NATE)
:SEG (LINK,1),(FILE,FP,PAGESNAP)        PAGE,SNAP,GRIMES
:SEG (LINK,40,ONTO,1),(FILE,FP,PAGETEST)
:SEG (LINK,50,ONTO,1),(FILE,FP,UVNO1)
:SEG (LINK,51,ONTO,1),(FILE,FP,VAE1)
:SEG (LINK,56,ONTO,1),(FILE,FP,NATE1)
:SEG (LINK,64,ONTO,1),(FILE,FP,PES1)
:SEG (LINK,70,ONTO,1),(FILE,FP,RPA1)
:SEG (LINK,2),(FILE,FP,STATUS)
:SEG (LINK,3),(FILE,FP,START)
:SEG (LINK,4),(FILE,FP,STOP)
:SEG (LINK,5),(FILE,FP,HTSTART)
:SEG (LINK,6),(FILE,FP,LRV)
:SEG (LINK,7),(FILE,FP,PRCRT)
:SEG (LINK,8),(FILE,FP,SCHEDULE)
:SEG (LINK,9),(FILE,FP,HOLD)
:SEG (LINK,10),(FILE,FP,LRVSET)
:SEG (LINK,11),(FILE,FP,LIMITS)
:SEG (LINK,12),(FILE,FP,WAIT)
:SEG (LINK,13),(FILE,FP,RAW)
:SEG (LINK,14),(FILE,FP,VERIFY)
:SEG (LINK,15),(FILE,FP,DANGER)
:SEG (LINK,16),(FILE,FP,PRTCOEFS)
:SEG (LINK,17),(FILE,FP,GOTO)
:SEG (LINK,18),(FILE,FP,PRTLIMIT)
:SEG (LINK,20),(FILE,FP,TESTD)
:SEG (LINK,21),(FILE,FP,WAITD)
:SEG (LINK,22),(FILE,FP,SKIP)
:SEG (LINK,23),(FILE,FP,HTSTOP)
:SEG (LINK,24),(FILE,FP,WAITA)
:SEG (LINK,25),(FILE,FP,TESTA)
:SEG (LINK,26),(FILE,FP,OSS)
:SEG (LINK,27),(FILE,FP,FAIL)
:SEG (LINK,28),(FILE,FP,CONVCOEF)
:SEG (LINK,29),(FILE,FP,IF)
:SEG (LINK,30),(FILE,FP,DIRECT)
:SEG (LINK,31),(FILE,FP,CHARTON)
:SEG (LINK,32),(FILE,FP,CHARTOFF)
:SEG (LINK,33),(FILE,FP,LIMON)
:SEG (LINK,34),(FILE,FP,UVNO)
:SEG (LINK,35),(FILE,FP,VAE)
:SEG (LINK,36),(FILE,FP,PROCLEAR)
:SEG (LINK,37),(FILE,FP,CONTROL)
:SEG (LINK,38),(FILE,FP,BIMS)
:SEG (LINK,39),(FILE,FP,VAESNAP)
:SEG (LINK,41),(FILE,FP,PES)
:SEG (LINK,42),(FILE,FP,RPA)
:SEG (LINK,43),(FILE,FP,MIMS)
:SEG (LINK,44),(FILE,FP,NATE)
:SEG (LINK,45),(FILE,FP,NACE)
:SEG (LINK,46),(FILE,FP,UVNOSNAP)
:SEG (LINK,47),(FILE,FP,DEFSTD)
:SEG (LINK,48),(FILE,FP,STDOSS)
:SEG (LINK,49),(FILE,FP,CEP)
:SEG (LINK,52),(FILE,FP,UVNONADR)
:SEG (LINK,53),(FILE,FP,STARTDAT)
:SEG (LINK,54),(FILE,FP,VAECT)
:SEG (LINK,55),(FILE,FP,PESSNAP)
:SEG (LINK,57),(FILE,FP,NATEON)
:SEG (LINK,58),(FILE,FP,UVNOFLOW)
:SEG (LINK,59),(FILE,FP,VAETIME)
:SEG (LINK,60),(FILE,FP,CURVTEST)
:SEG (LINK,61),(FILE,FP,LIMOUT)
:SEG (LINK,63),(FILE,FP,PESCALIB)
:SEG (LINK,65),(FILE,FP,STOPDAT)
:SEG (LINK,66),(FILE,FP,PESSTORE)
:SEG (LINK,67),(FILE,FP,EVMARKON)
:SEG (LINK,68),(FILE,FP,EVMARKOF)
:SEG (LINK,69),(FILE,FP,NATESNAP)
:SEG (LINK,71),(FILE,FP,SIMSNAP)
:ASSIGN (F:STD,D1,STD)
:ASSIGN (F:SIM,LO)
:ASSIGN (F:ACTV,D1,ACTV)
:ASSIGN (F:PROC,D3,PROCFILE)
:ASSIGN (F:DCBRAW,D1,RAWFILE)
:ASSIGN (F:PEND,D1,PEND)
:ASSIGN (F:COEF,D3,COEFFILE)
!RADEDIT
:TRUNCATE FP
:SQUEEZE FP
:SAVE ALL
!FIN
```

Figure 23. Listing of the AECS Overlay Deck

REFERENCES

1. Preliminary Document "OSO SOFTWARE DESIGN REQUIREMENT," Code 511, dated 11/11/71.

2. XDS SIGMA 5 COMPUTER REFERENCE MANUAL - 90 09 59D

3. XDS MACRO-SYMBOL Language and Operations - 90 15 78A

4. XDS REAL-TIME BATCH MONITOR (RBM) Reference - 90 15 81D

5. XDS REAL-TIME BATCH MONITOR (RBM) User's Guide - 90 16 53A

6. XDS SYSTEM KEYBOARD DISPLAY, Pub. #988021

7. Several individuals assisted in the design of the AECS as well as in the preparation of this document, especially Mr. James McGuire of the AE Project Office:

OCC	T & E	AE PROJECT
Gardiner Hall	James Bailey	James McGuire
	Roy Morgan	David Haykin
RCA	David Christofalo	Wayne Hembree
	Russel Beard	
	David Provost	
Phillip Brandt		
Mary Theus		

8. Individuals responsible for the programming of AECS:

   A. Villasenor, Code 324
   W. Mocarsky, Code 324
   W. Blakeslee, Code 324

APPENDIX A

SYSTEM DIRECTIVES

System directives are statements which result in a response from the system; History Tape recording may be initiated, the telemetry input stream can be stopped, etc. All directives are processed by the ECS section, and must conform to the following format:

*NNNNNNNN,ARG1,ARG2, , ,ARG20. COMMENTS

where NNNNNNNN is the directive mnemonic, ARGi are numeric or character strings, and any statement appearing after the period is treated as comments and is not processed by the system.

The basis rules for directives are:

(1) The symbol * should appear in column 10 of card inputs

(2) NNNNNNNN and ARGi may be as long as desired, but only the first 8 characters are recognized by the system.

(3) A period must complete the directive statement.

(4) ARGi may be a telemetry word, a constant, or an expression. Values of telemetry words are taken from the LRV table.

(5) Constants have the following format capabilities:

- X'NNNNNNNN' - hexadecimal (0-9, A-F); 16 digits maximum
- D'NN.NNNN' - decimal or floating point number
- NNNNNNNN - decimal integer; 16 digits maximum
- B'NNNNNNNN' - binary number (1 or 0); 64 digits maximum
- O'NNNNNNNN' - octal (0-7); 31 digits maximum

Any number without a prefix X,D,B, or O will be treated as an integer.

(6) When directives are submitted to the system, they will be processed as soon as the currently executing statement finishes. If the user wishes to cancel a directive, the Cancel (%) code should be used.

Special Cancel Character (%)

The special character % is used to Cancel a currently executing directive, command, or data processing (overlay) program. Utility and SIM directive operations can also be cancelled by %. If the currently executing program is contained within a Schedule or a Procedure, a HOLD mode is entered before continuing on with the next statement in the queue.

*CHARTOFF,P1,P2,,,Pn. — Terminates the assignment of the specified strip chart pens on the Brush recorders. If no pens are specified, all strip chart assignments are cleared. The value of Pi ranges from 1 through the maximum number of pens available.

EX: *CHARTOFF,3,7. — Cancel operation of pens 3 and 7.

*CHARTOFF. — Cancel all pen activity.

*CHARTON,P1,TM(I1,J1),P2,TM(I2,J2),,,,. — Assigns the specified pens to the corresponding telemetry words. Any error in the list specification cancels the entire list. Digital-to-analog conversion is linear: a telemetry value of 00000000 corresponds to 0.0 volts, and a value of 11111111 corresponds to 5.11 volts. Speed of the recording (Brush Mark 200 8-channel recorders) is controlled at the recorder console.

EX: *CHARTON,1,TM(37). — Assigns strip chart pen #1 to the sub com counter, TM(37).

*CLEAR,N. — Clears the entire display screen of CRT #N.

*CONVCOEF,TM(I,J),A0,A1,A2,A3,A4,A5,A6,A7. — Assigns the string of coefficients to the telemetry word TM (I,J) for engineering conversion. One or more coefficients may be entered in either integer or floating point decimal form. Each *CONVCOEF statement results in an update of the resident TMDCOEF coefficient table, and the saving of the updated table on the RAD. When the AECS system is re-started afterwards, the latest coefficient table is automatically read into core.

EX: *CONVCOEF,TM(4),0,1,2,3. — Stores the specified coefficients for use in computing the value V, where $V = 0 + 1*TM(4) + 2*TM(4)^2 + 3*TM(4)^3$.

EX: *CONVCOEF,TM(9),D'1.5",D'5.02'. — Converts TM(9) by the polynomial $V = 1.5 + 5.02*TM(9)$.

*CRT,N,L,C,XXXXXXXXXX. — Generates new page display programs or modifies existing ones. N is the Page number, L is the line number and C is the column. The resulting page overlay is permanently saved on the RAD and can be called in for execution by the call *PAGE,N,X. where X is the desired screen number. Up to 20 Pages (N between 1 and 20) can be created in this way. There are 3 types of *CRT statements:

Type 1 — *CRT,N,L,C,TEXT. — Display up to 40 characters of EBCDIC text starting on line L, column C of Page N.

Type 2 — *CRT,N,L,C,TM(I,J),IDENT. — Display the telemetry value TM(I,J) on line L, column C of Page N. IDENT contains up to 8 characters of identification. The telemetry value is displayed in binary form unless a conversion has been supplied by *CONVCOEF.

Type 3 — *CRT,N,L,C,TM(I,J),MASK, TEXT0, TEXT1,,,,. — On line L column C of Page N, display the 8-character text messages corresponding to the value of TM(I,J) masked by MASK. That is, if MASK = X'00011000' and bits 4 and 5 of M(I,J) are 1 and 0 respectively, display TEXT2 (2 = B'10').

The page N can be cleared by the statement *CRT,N,CLEAR.

*DELETE. — Cancels execution of dangerous commands. This statement is normally issued during a HOLD mode.

*DISABLE,N,NEXP. — Disables the communication unit N from command transmission. That is, after execution of this statement, unit N will not be able to generate commands to the experiment whose mnemonic is NEXP. If the parameter NEXP is omitted, unit N cannot issue commands to any experiment. This statement can only be issued from Unit 1, the system console.

EX: *DISABLE,2,VAE. — Prevents CRT #2 from issuing any commands to the VAE experiment.

EX: *DISABLE,3. — Stops all commanding from CRT #3.

EX: *DISABLE,3,ALL. — Stops all activity whatever on CRT #3. No commands, directives, etc. can be issued thereafter.

*ENABLE,N,NEXP. — Permits unit N to send commands to experiment NEXP. This statement can only be generated at the system console, CRT #1. If NEXP is omitted, unit N is permitted to send commands to all experiments.

EX: *ENABLE,X'C',BIMS. — Allow unit C (SCHEDULE TAPE) to send commands to the BIMS experiment.

EX: *ENABLE,2. — Allow unit CRT #2 to transmit commands to all experiments.

EX: *ENABLE,2,ALL. — Allow CRT #2 to send commands, directives, etc.

*FAIL,GO. or *FAIL,HOLD. — Causes the comments (following the period) to be displayed on the Event Printer and on all CRT screens. No interruption in the sequence of operations occurs if the parameter GO is used; the parameter HOLD results in the system entering a HOLD mode.

EX: *FAIL,HOLD. DATA OUT OF LIMITS. — If this statement is reached by a *SKIP or *GOTO, the statement "DATA OUT OF LIMITS" is shown on all system devices, and a HOLD mode is entered. The HOLD mode can be cleared only by the *GO directive.

*GO. — Clears the system HOLD mode and continues on with the next statement to be executed in a SCHEDULE. This statement has no effect if the system is not currently in a HOLD mode, and can therefore be used to display comments during SCHEDULE execution.

*GOTO,N. — Transfers control to the SCHEDULE statement N. If no SCHEDULE is in operation, this is a null statement. If N is greater than 9999 or less than 0000, the SCHEDULE is aborted. N is the statement number in columns 72-80 of the SCHEDULE card image.

EX: *GOTO,345. — Issued during a HOLD mode, this statement causes the SCHEDULE to skip to statement number 345. The HOLD mode is still in effect, so that the system operator can verify that 345 is the desired statement; if so, the operator issues a *GO. and statement 345 is executed. If a mistake was made, another *GOTO,N. can be issued.

*HOLD. — Sets the system HOLD mode. The HOLD mode is simply a means of bypassing the automatic execution of a SCHEDULE. All statements in the system can be issued during a HOLD mode, but the SCHEDULE cannot be continued until the *GO directive is issued.

*HTPLAY,N. — Directs the system to read data from the History Tape on 9TB8N and treat the data as if it were raw telemetry. This History Tape "playback" can be used for subsystem checkout. If N is omitted, the process is cancelled.

*HTSTART,N. — Starts the History Tape recording real-time telemetry data. Each record on the History Tape is 560 32-bit words long. Another *HTSTART,M. can be issued to switch tape drives from N to M without loss of data. N is either 2 or 3.

EX: *HTSTART,3. — Start History Tape recording on 9TB83.

*HTSTOP. — Stops data from being placed on the History Tape.

*IF,ARITH,NEG,ZERO,POS. — Depending on the value of ARITH, causes a branch to one of three PROCEDURE statements. This statement is similar to the FORTRAN 3-way IF instruction.

ARITH — any arithmetic expression consisting of combinations of the arithmetic operations +,-,*, and /, and telemetry words or real numbers. The telemetry words, denoted by TM(I,J), are converted if a conversion polynomial exists. Grouping by parenthesis is permitted. Up to 26 numbers are allowed in the expression.

NEG — the relative position in the PROCEDURE to which control will be transferred if ARITH is negative.

ZERO — the relative position in the PROCEDURE to which control will be transferred if ARITH is zero.

POS — the relative position in the PROCEDURE to which control will be transferred if ARITH is positive.

EX: *IF,TM(65,4)-(6.*TM(37))/8.,-1,0,2. — If this is statement P in the PROCEDURE, control transfers to PROCEDURE statements P-1, P, or P+2 if the value of the expression TM(65,4)-(6.*TM(37))/8. is negative, zero, or positive, respectively.

*LIMITS,TM(I,J),HIGH,LOW. — Establishes limit values for the telemetry word TM(I,J). The parameters HIGH and LOW are specified in hundredths of volts (20 millivolts per count), and may be interchanged. This directive merely defines limits -- it does not actually initiate limit checking; the directive *LIMON triggers limit checking.

EX: *LIMITS,TM(17),020,350. — The lower limit for TM(17) is 0.20 volts and the upper limit is 3.50 volts; these correspond to counts of 10 and 175 respectively. If either limit is exceeded, the message "DOL - TM(17)=81" appears on the Event Printer. DOL means Data-Out-of-Limits, and 81 is the hexadecimal value of TM(17).

*LIMOFF,TM(I,J). — Prevents further limit checking on telemetry word TM(I,J). Absence of TM(I,J) cancels all limit-checking.

EX: *LIMOFF,TM(17). — Cancels limit checking for TM(17).

*LIMON,TM(I,J) — Triggers limit checking on telemetry word TM(I,J), assuming that limits have already been defined by the *LIMITS directive.

EX: *LIMON,TM(17). — Initiates limit checking for TM(17).

*LIMOUT,TM(I,J),&PPPPPPPP. — Forces execution of the RAD PROCEDURE PPPPPPPP if TM(I,J) exceeds the limits defined in a *LIMITS directive. The PROCEDURE PPPPPPPP must exist on the RAD. If an out-of-limits conditions occurs while another PROCEDURE is in operation, that PROCEDURE is aborted and replaced immediately by PPPPPPPP. PPPPPPPP can contain commands, directives, etc. to turn off the subsystem. The procedure PPPPPPPP must complete execution within 30 seconds.

EX: *LIMOUT,TM(1),&SYNC. — Whenever TM(1) exceeds its limits, the PROCEDURE "SYNC" is immediately called in for execution.

*LRV,XXX,TM(I),,,TM(J). — displays raw and converted (if a conversion exists) data for the specified telemetry words on the CRT (XXX=CRT) or Event Printer (XXX=PRT). If another directive or command is not imediately issued, the values are updated every second.

EX: *LRV,CRT,TM(37),TM(17,1),TM(108). — The specified telemetry words will be displayed on the screen of the CRT device from which this directive was issued. If this statement was part of a PROCEDURE, CRT #1 receives the data.

*LRVSET,TM(I,J),X1,TM(K,L),X2,... — Inserts the given value into the designated telemetry word, providing that telemetry is not currently being received. This directive is used primarily for debugging telemetry-dependent programs.

EX: *LRVSET,TM(65,9),255,TM(37),X'03'. — Inserts the value 11111111 into TM(65,9) and 00000011 into TM(37).

*PAGE,CCCCCCCC,N. — Causes the PAGE "CCCCCCCC" to be displayed on CRT #N. CCCCCCCC is the name of the PAGE, and usually is an easily identifiable mnemonic; from 1 to 8 characters are permitted. If N is omitted, the screen of the CRT which issued the directive will be used. Any page may be cancelled when CCCCCCCC = CANCEL.

EX: *PAGE,NATE1,3. — Display the PAGE called NATE1 on CRT 3.

EX: *PAGE,CANCEL,3. — Cancel the PAGE on CRT #3.

*PRABORT. — Aborts the current RAD PROCEDURE.

*PRCRT,N. — Prints the entire screen image of CRT #N on the Snapshot Printer. If N is omitted, the screen of the CRT which issued this directive will be printed.

*PRTCOEFS. — Prints the telemetry conversion coefficients from the core-resident tables TMDINDEX and TMDCOEF. While there is no limit to the number of coefficients for each telemetry word, only the first 8 are printed.

*PROCLEAR,BBBBBBBB. — Clears the previously defined PROCEDURE BBBBBBBB from the RAD PROC file, D3. If BBBBBBBB is omitted, the statement is ignored. If BBBBBBBB=ALL, all procedures are removed.

EX: *PROCLEAR,VAESTOP. — Eliminate the PROCEDURE VAESTOP from the RAD files.

EX: *PROCLEAR,ALL. — Remove all PROCEDURES from the RAD.

*PROCLIST. — Lists the names of all PROCEDURES available on the RAD.

*PRTLIMIT. — Prints the contents of the limit checking tables on the Snapshot Printer.

*RAW,N. — Prints minor frame N on the Snapshot Printer in hexadecimal form. If N is omitted, the entire major frame is dumped. The printout is performed once each major frame, unless another directive is submitted to the system.

EX: *RAW,1,3,5 — Dump minor frames 1, 3 and 5 each major frame.

*SCHEDULE,NNNNNNNN,LIST. — Begin execution of the SCHEDULE TAPE called NNNNNNNN. The SCHEDULE is assumed to be on magnetic tape

9TA80, and have the name NNNNNNNN. The first statement to be executed, regardless of the first statement of the SCHEDULE, is the directive *HOLD.

EX: *PROC,OSSX. — Begin executing the SCHEDULE called OSSX from magnetic tape 9TA80.

EX: *PROC,OSSX,LIST. — List the statements, without execution, of the SCHEDULE called OSSX on 9TA80. The listing is placed on the system LO device.

*TESTD,TM(I,J),MASK,VALUE,MATCH,NOMATCH. — Performs a test on the digital telemetry word TM(I,J) as follows:

(1) Form the logical product of TM(I,J) with MASK

(2) Compare with result with VALUE

(3) If identical, skip to MATCH statements relative to this one

(4) Otherwise, skip to NOMATCH statements relative to this one.

EX: *TESTD,TM(37),B'00000001',00000001,-1,1. — Whenever the telemetry word TM(37) is odd, the statement previous to this one will be executed; when it is even, the following statement will be executed.

*VERIFY,MODE,RETRIES. — Specifies commanding mode information. MODE = 0 implies no verification at all, and unless otherwise specified is the default mode. MODE = 1 forces command verification after a burst transmission of commands. MODE = 2 forces verification of a previously sent command; following commands cannot be transmitted until the previous command has been verified.

RETRIES — indicates the number of times a command will be re-transmitted if it fails to verify the first time. If the command is sent this number of times without verification, a HOLD mode is entered, to allow the system operator to skip this command (one command) or abort the command sequence.

EX: *VERIFY,2,2. — Select MODE = 2 for commanding, and only permit 2 re-tries for transmission.

*WAIT,N. — Pauses for the specified number of tenths of seconds.

EX: *WAIT,50. — Wait 5 seconds before continuing.

*WAITA,TM(I,J),HIGH,LOW,MAXTIME. — Waits until telemetry word TM(I,J) is between HIGH and LOW before continuing. If MAXTIME is specified, the checking continues until MAXTIME is reached, at which point a HOLD mode is entered. MAXTIME is in tenths of seconds. The sequence of operations is as follows:

(1) Compare TM(I,J) with HIGH

(2) If TM(I,J) is greater than HIGH, go back to step 1.

(3) Compare TM(I,J) with LOW

(4) If TM(I,J) is less than LOW, go back to step 1.

(5) Continue

HIGH and LOW are specified in hundredths of volts (20 millivolts per count), and may be interchanged.

EX: *WAITA,TM(75),050,400,MAXTIME. — Wait until TM(75) is between 0.50 volts (count = 25) and 4.00 volts (count = 200) before continuing.

*WAITD,TM(I,J),MASK,VALUE,MAXTIME. — Wait until the digital telemetry word TM (I,J) has the value VALUE when masked by MASK, or until MAXTIME is exceeded. When MAXTIME is reached, a HOLD mode is entered. The sequence of operations is as follows:

(1) Perform the logical product of TM(I,J) with MASK

(2) Compare the result with VALUE

(3) If equal, continue. Otherwise, go back to step 1.

EX: *WAITD,TM(37),X'FF',1. — Do not proceed until the value of the sub com counter is exactly 1.

APPENDIX B

COMMAND MNEMONICS

Each subsystem is associated with a unique command program that is responsible for interpreting all possible English-language mnemonics related to that subsystem and for producing the necessary fields of the 64-bit command. This command program then transfers control to the resident COMMAND routine for transmission and verification. The mnemonics for each command program depend on the subsystem, but in general have the following form:

/NNNN,ARG1,ARG2,,,,,ARGn.

where

NNNN - 1-4 character identification of the subsystem
ARGi - alphabetic or numeric parameters, to be interpreted by the individual command programs. Each ARGi contains 1 to 8 characters.

General Command

A general purpose command programs exists in the system for use in checkout of hardware. This program is /DIRECT,NNNN. where NNN is any string of numeric characters.

EX: /DIRECT,X'0123456789ABCDEF'. — Transmit the specified data bits through the command logic. Reserved command fields, such as the spacecraft address and poly check code, are replaced by the system. Spare bit fields are inserted with zeros.

Spacecraft Memory Commands

The following commands and directives relate to the loading of the spacecraft command memory.

#CMD FORMAT,AAAAA,BBBB,T. — Defines the commands which will be loaded into the spacecraft memories. The underlined portion of the command defines any AECS command format used to generate commands. The loads are saved on the RAD for later uplinking as one load execution.

AAAAA — The S.E.T. in decimal of command execution that is to be loaded into memory. Five characters may be specified. 0 AAAAA 16384. It

should be noted that since the spacecraft clock increments once every 4 seconds, there is a lapse of 4 seconds between consecutive values indicated in this field. For example, a value of 00005 would indicate a 20-second lapse from a SET of 00000. A null field implies usage of the last specified value for AAAAA.

BBBB — Indicates the memory address in octal where the generated load commands will be stored. If omitted, the default value will be the next available memory cell. 0 BBBB 37778. The CMU will be selected depending on the value of BBBB. A SET LOAD COMMAND will be generated as part of the load data each time this argument is used.

T — Indicates the last group of commands for this load block. It must be present on the last "#" command.

/MEMLD,X. — This command directs action to be performed on a previously generated load block.

X = U - Uplink the loads on the RAD load file

X = P - Print the loads on the RAD load file.

X = R - Update ground reference image with loads on the RAD file.

/MEMDP,E. — This command will uplink commands necessary to dump all of the memory specified by E and compare this to the current ground reference image.

E = 1 or E = 2 selects memory to be dumped.

*MEMREFOV,M. — All failures in the last memory-dump-compare are transferred to the memory reference image file on the RAD.

M = 1 or 2 specifies which memory image is updated.

*DUMPRINT,XX.

XX=ON - List all commands on the following dumps.

XX=OF - List only commands that failed to compare on all following dumps.

*TPREC,FF. — Start or stop recording of tape recorder TLM.

FF=ON - Command the recorders on, connect TR TLM interrupts.

FF=OF - Command the recorders off, disconnect TR TLM interrupts.

*MINDBASE,GGGGGGGG,XXXXXXXX. — Established minor mode command data base for use in load generation.

GGGGGGGG - Experiment name.

XXXXXXXX - 32-bit minor mode data to be used as data base for the following # commands. Leading zeros may be suppressed.

APPENDIX C

ATMOSPHERE EXPLORER TELEMETRY

MAIN FRAME FORMAT

(AE-C SATELLITE)

1 SYNC	2 SYNC	3 SYNC	4 COMMAND STATUS	5 COMMAND STATUS	6 RPA A	7 UVNO D	8 UVNO D
9 CEP A	10 BIMS A	11 EUVS D	12 EUVS D	13 EUVS D	14 EUVS D	15 RPA A	16 BIMS A
17 S/C	18 S/C	19 MIMS A	20 EUVS A	21 NACE D	22 NACE D	23 BIMS A	24 RPA A
25 CEP A	26 MIMS A	27 NATE D	28 NATE D	29 UVNO D	30 NATE A	31 NATE D	32 RPA A
33 RPA A	34 ESUM D	35 ESUM D	36 BIMS A	37 SUBCOM COUNTER	38 PES D	39 PES D	40 PES D
41 CEP A	42 RPA A	43 LEE D	44 LEE D	45 LEE D	46 VAE D	47 VAE D	48 VAE D
49 BIMS A	50 UVNO D	51 RPA A	52 RPA A	53 NACE D	54 NACE D	55 ESUM A	56 BIMS D
57 CEP A	58 MIMS A	59 NATE D	60 NATE D	61 RPA A	62 BIMS A	63 OSS D	64 OSS D
65 SUBCOM	66 SUBCOM	67 SUBCOM	68 SUBCOM	69 PSA A	70 RPA A	71 TAL A	72 UVNO D
73 CEP A	74 MIMS A	75 BIMS A	76 OSS D	77 OSS D	78 OSS D	79 RPA A	80 BIMS D
81 S/C	82 S/C	83 MIMS A	84 MIMS A	85 NACE D	86 NACE D	87 BIMS A	88 RPA A
89 CEP A	90 MIMS A	91 NATE D	92 NATE D	93 UVNO D	94 OSS A	95 NACE A	96 RPA A
97 RPA A	98 ESUM D	99 ESUM D	100 BIMS A	101 PSB A	102 PES D	103 PES D	104 PES D
105 CEP A	106 RPA A	107 MAG A	108 MAG A	109 MAG A	110 VAE D	111 VAE D	112 VAE D
113 BIMS A	114 UVNO D	115 RPA A	116 NACE D	117 NACE D	118 NACE D	119 MESA D	120 MESA D
121 CEP A	122 MIMS A	123 NATE D	124 NATE D	125 RPA A	126 BIMS A	127 RPA D	128 LEE A

July 19, 1972

APPENDIX D

ATMOSPHERE EXPLORER

8-SECOND SUBCOM FORMAT

(AE-C SATELLITE)

1 UVNO A	2 UVNO A	3 UVNO A	4 UVNO A	5 UVNO A	6 UVNO A	7 UVNO A	8 UVNO A
9 MESA A	10 MESA A	11 MESA A	12 RELAY STATUS D	13 BIMS A	14 VAE D	15 VAE D	16 VAE D
17 VAE A	18 VAE A	19 VAE A	20 VAE A	21 VAE A	22 VAE A	23 VAE A	24 VAE A
25 RPA A	26 RPA A	27 RPA A	28 LEE D	29 LEE D	30 LEE D	31 LEE D	32 NACE A
33 NACE A	34 NACE A	35 NACE A	36 NACE A	37 NACE A	38 NACE A	39 NACE A	40 NACE A
41 OSS A	42 OSS A	43 OSS A	44 OSS A	45 OSS A	46 OSS A	47 NACE D	48 NACE A
49 OSS A	50 OSS A	51 OSS A	52 OSS A	53 OSS A	54 OSS A	55 PES D	56 BIMS D
57 LEE A	58 LEE A	59 LEE A	60 LEE A	61 LEE A	62 RELAY STATUS D	63 NATE D	64 RPA D
65 CEP A	66 CEP A	67 CEP A	68 CEP A	69 CEP A	70 CEP A	71 NATE A	72 NATE A
73 NATE A	74 NATE A	75 NATE A	76 NATE A	77 NATE A	78 NATE A	79 NATE A	80 NATE A
81 MIMS A	82 MIMS A	83 MIMS A	84 MIMS A	85 MIMS A	86 MIMS A	87 MIMS A	88 MIMS A
89 MIMS A	90 RELAY STATUS D	91 PES A	92 PES A	93 PES A	94 PES A	95 PES A	96 PES A
97 ESUM A	98 ESUM A	99 ESUM A	100 ESUM A	101 ESUM A	102 ESUM A	103 ESUM A	104 ESUM A
105 ESUM A	106 RELAY STATUS D	107 NATE D	108 NATE D	109 NACE D	110 NACE D	111 NACE D	112 NACE D
113 EUVS A	114 EUVS A	115 EUVS A	116 EUVS A	117 BIMS D	118 BIMS D	119 PES D	120 BIMS D
121 BIMS A	122 BIMS A	123 BIMS A	124 BIMS A	125 BIMS A	126 BIMS A	127 NATE D	128 RPA D

March 8, 1972

APPENDIX E

ATMOSPHERE EXPLORER

4-SECOND SUBCOM FORMAT

(AE-C SATELLITE)

1 RPA D	2 CEP D	3 BIMS D	4 ESUM D	5 ESUM D	6 ESUM D	7 ESUM D	8 RPA A
9 S/C CLOCK	10 BIMS A	11 PSA A	12 RPA D	13 PSA A	14 PSA A	15 PSA A	16 RPA A
17 UVNO D	18 BIMS A	19 BIMS D	20 MIMS D	21 MAG A	22 RPA D	23 SPARE	24 RPA A
25 BIMS A	26 BIMS A	27 VAE A	28 VAE A	29 MESA A	30 MESA A	31 MESA A	32 RPA A
33 RPA D	34 MIMS D	35 BIMS D	36 OSS D	37 OSS D	38 OSS D	39 OSS D	40 RPA A
41 PSB A	42 BIMS A	43 PSB A	44 RPA D	45 MESA D	46 UVNO A	47 TAL A	48 RPA A
49 EUVS D	50 EUVS D	51 BIMS D	52 MIMS D	53 EUVS D	54 RPA D	55 PES D	56 RPA A
57 PSB A	58 BIMS A	59 PSB A	60 PSB A	61 MESA A	62 MESA A	63 MESA A	64 RPA A

July 21, 1972

APPENDIX F

TELEMETRY ITEM DESIGNATION

The designation of telemetry items, including supercommutated telemetry words, subcommutated telemetry words, and various combinations of words and partial words are necessary in the definition of the telemetry items and in the specification of data processing. An initial convention has been adopted for use in the Sigma 5 programs to display telemetry words on the CRT (System Directive *CRT). The convention proposed here adopts the previous Sigma 5 conventions and adds the facility of concatenation, partial word extraction, an "or" function, and the identification of bit patterns. The enhancement of the original Sigma 5 convention proposed here is for the purpose of defining telemetry items and to add in the specification of data reduction programs, and does not imply that these will be implemented in software in either the Sigma 5 or the central computer. It will, however, allow unambiguous written definition of telemetry items in a manner which is common to all groups and which will be easy to incorporate in flow diagrams.

A. Each main frame word is designated by TM(i) where i as a number between 1 and 128 identifying the word.

   EX: TM(27) refers to main frame word 27 - always a NATE experiment word.

   TM(67) is main frame word 67, which is a 4 second subcom that has a different meaning at each step.

B. Main frame words may have meaning in consecutive groups (concatenated). These groups can be designated by placing colons between the word identification numbers.

   EX: TM(47:111) refers to the combined 16 bits on main frame words 47 and 111.

C. Bits within a word or words may be designated. The bit numbering is considered to be MSB first (leftmost) and LSB last (rightmost). The bit numbering of a word would be:

   MSB = 12345678 = LSB

   The designation of the bits is accomplished by preceeding the bit identification sequence by the telemetry word notation. If a sequence of

numbers, separated by commas, is used, the bits are to be considered in the given order. Two hyphenated numbers represent the sequence included by the numbers.

EX: TM(27)5 refers to main frame word 27, bit 5

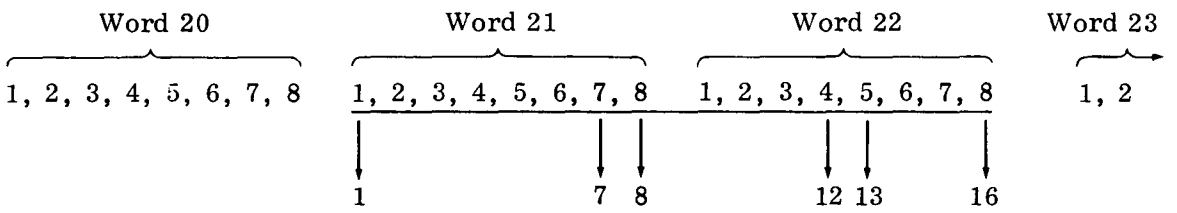
TM(27)5,7,2 refers to main frame word 27, bits 5, 7, and 2 in that order

TM(27)2,3,4,5 refers to main frame word 27, bits 2,3,4 and 5 in that order

TM(27)3-5 represents main frame word 27, bits 3,4, and 5

TM(27)5-3 represents main frame word 27, bits 5,4, and 3

TM(21:22)1,7,8,12,13 refers to the concatenated 16 bits of main frame words 21 and 22 with the selected bits as shown below:



D. Subcom words may be designated by the main frame word identification followed by the subcom count (step) number, separated by a comma.

EX: TM(68,28) is the 28th step of the 8 second subcom for experiments

TM(67,64) is the 64th word of the 4 second subcom for experiments

TM(17,4:18,4) refers to the spacecraft 1/4 second subcoms, the 4th step, and all 16 concatenated bits

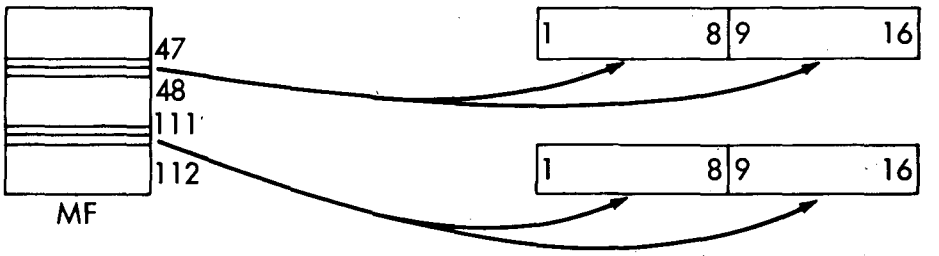
TM(68,28)5,10,7 are the 5th, 10th, and 7th bits of main frame word 68, step 28

TM(65,38:66,38#66,102)2,11 refers to the 2nd and 11th bits of the 16 bit value produced by the concatenation of TM(65,38) with TM(66,38), or by the concatenation of TM(65,38) with TM(66,102). This is the method by which a four second subcom word and its adjacent 8 second subcom word may be concatenated, as is done for some of the attitude data in the spacecraft subcoms.

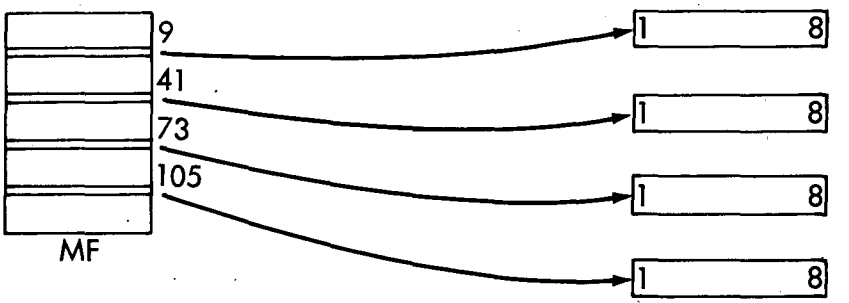
The symbol # indicates the "or" function. This example is repeated later in Example F except that a set of additional parentheses are added for clarity.

The use of the symbols may be expanded to cover other useful designations. These include concatenating individually specified items, use of parentheses to emphasize order of operations, and the designation of the value of individual bits for definition of a state or a sequence. A set of examples follow, illustrating possible uses:

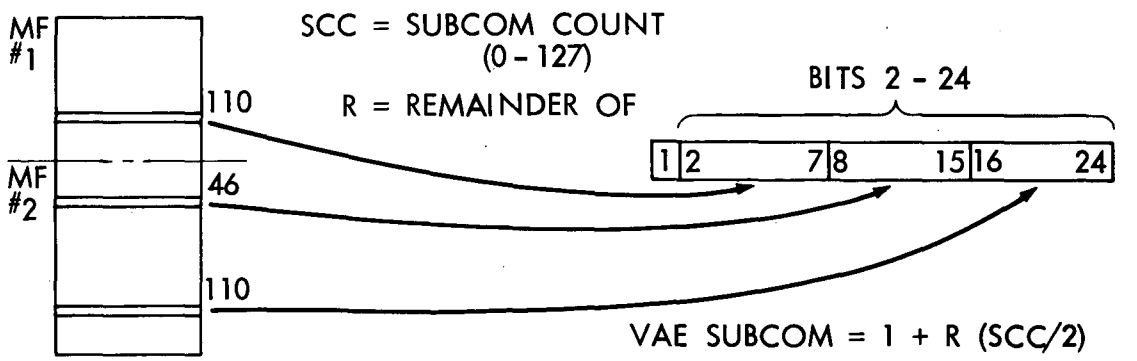
A. TM(47:48)#TM(111:112) = Supercom of VAE Channel 1



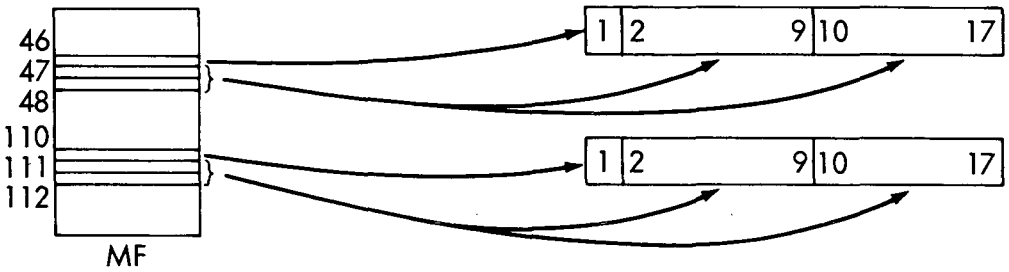
B. TM(9)#TM(41)#TM(73)#TM(105) = Supercom of CEP Electrometer 1 Output



C. TM(110,1:46,2:110,2)2-24 = VAE Subcom of Channel 2 data



D. (TM(46)7:TM(47:48))#(TM(110)7:TM(111:112)) = Supercom of VAE Channel 1 with SQNTS bits attached



* U.S. GOVERNMENT PRINTING OFFICE: 1972—735-963/180