MC 104

project
mercury

BERMUDA MONITOR PROGRAMS

prepared for
National Aeronautics and Space Administration
Contract No. NAS 1-430

1 march 1961

International Business Machines Corporation

in association with

WESTERN ELECTRIC COMPANY, INC.

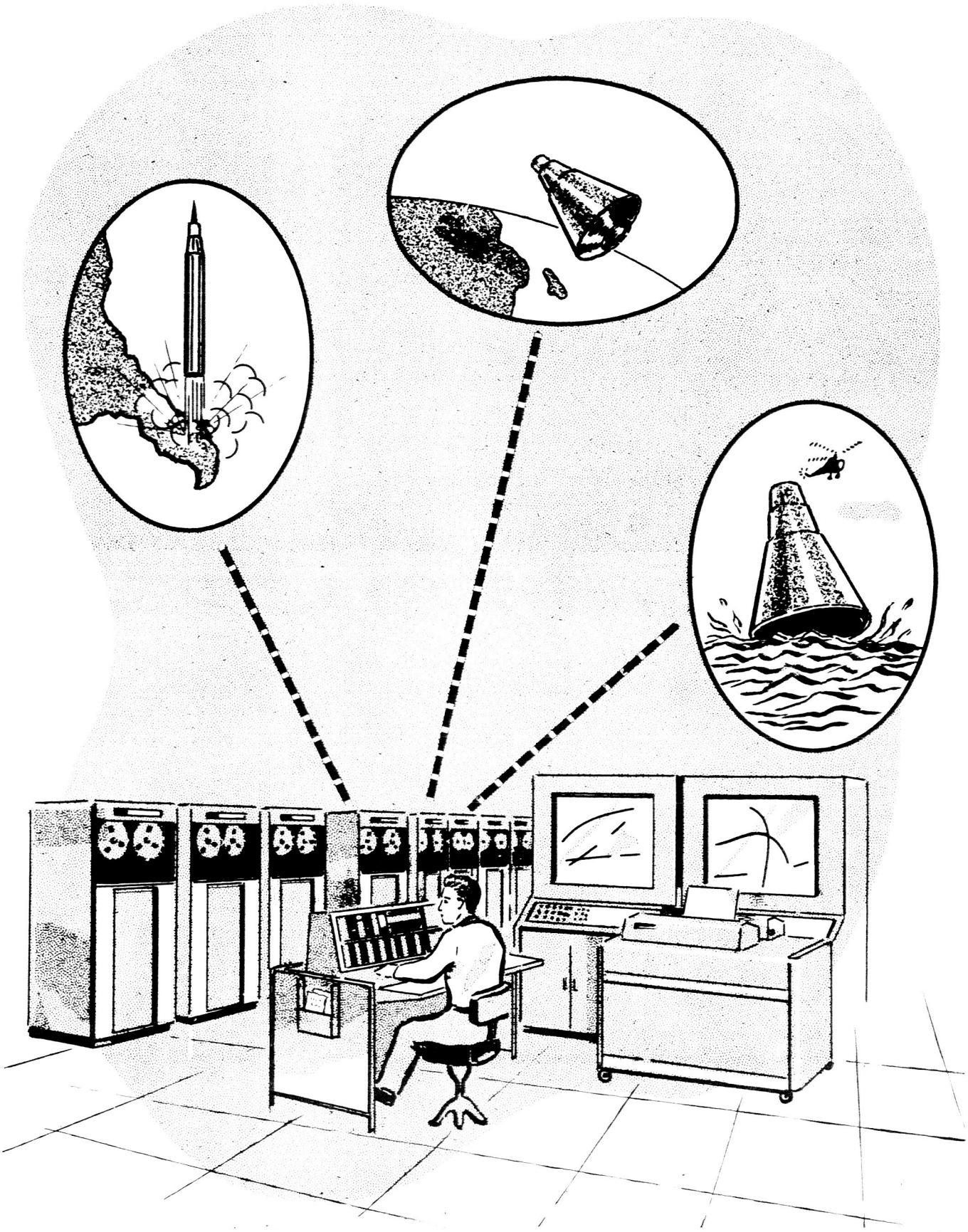


TABLE OF CONTENTS

Page

Section 1. INTRODUCTION

1.1 Data Communications Channel 1-2

1.2 Mercury Monitor 1-6

1.3 Scope of the Manual 1-11

Section 2. BASIC MONITOR CONTROL PROGRAMS

2.1 Main Controller Priority Program (M0PRIO) 2-2

2.2 DCC Main Controller Program (M0DCCC) 2-6

2.3 Main Controller Save Program (M0SAVE). 2-9

2.4 Main Controller Return Program (M0RTRN) 2-12

2.5 Main Controller Diagnostic Program (M0DIAG) 2-15

2.6 Main Controller Queue Program (M0QUEU) 2-17

2.7 Main Controller Unqueue Program (M0UNQU) 2-21

2.8 Turn On Macro (TRNON) 2-24

2.9 Turn Off Macro (TRNOF). 2-27

2.10 Queue Macro (QUEUE) 2-30

2.11 Unqueue Macro (UNQUE) 2-32

2.12 Enable Macro (QENBA)/Enable Program (MYENBA) 2-34

2.13 Disable Macro (QENBZ)/Disable Program (MYENB0). 2-37

2.14 Present Sense Line Macro (QPSLF)/Present Sense Line Program (MYPSLF). 2-40

TABLE OF CONTENTS (Cont'd)

Page

2.15 Save Macro (SAVE) 2-43

2.16 Reference Macro (REFR) 2-46

Section 3. DCC INPUT CONTROL

3.1 High-Speed Input: Radar Data 3-1

3.2 Monitor High-Speed Input Trap Processor (MTHSIN) 3-5

3.3 Monitor Prefix to BIHSIN (MPHSIN) 3-11

3.4 Monitor Suffix to BIHSIN (MFHSIN) 3-17

3.5 Low-Speed Input: Paper Tape 3-21

3.6 Monitor Teletype Input Trap Processor (MTTTIN) 3-23

3.7 Monitor Prefix to BIMANI (MPTIPT) 3-27

3.8 Monitor Suffixes to BIMANI (MFMANI, MFMAN1) 3-29

3.9 Low-Speed Input: Telemetry 3-33

3.10 Monitor Telemetry Input Trap Processor (MTTMIN) 3-35

3.11 Monitor Telemetry Processor (MYTMIN) 3-39

3.12 Monitor Engine Cut-Off Early Processor (MYECOE) 3-55

3.13 Monitor Sustainer Engine Cut-Off Processor (MYSECO) 3-59

3.14 Timing Control 3-63

3.15 Monitor Initial WWV Trap Processor (MTIWWV) 3-64

3.16 Monitor WWV Trap Processor (MTWWWV) 3-69

3.17 Monitor Half-Second Trap Processor (MTHFSC) 3-73

3.18 Monitor Minutes Processor (MYMINS) 3-79

TABLE OF CONTENTS (Cont'd)

Page

Section 4. CONTROL DURING PROCESSING

4.1 The Processing Cycle 4-1

4.2 Monitor Prefix to BEDITS (MPEDIT) 4-9

4.3 Monitor Suffix to BEDITS (MFEDIT) 4-19

4.4 Monitor Bad Points Suffix to BEDITS (MFEBAD) 4-23

4.5 Monitor Prefix to BSMOOS (MPSMOO) 4-27

4.6 Monitor Suffix to BSMOOS (MFSMOO) 4-35

4.7 Monitor Bad Points Processor (MYEBAD) 4-37

4.8 Monitor Prefix to BARSCH (MPSARC) 4-47

4.9 Monitor Suffix to BARSCH (MFSARC) 4-53

4.10 Monitor Prefix to BOCOMP (MPCOMP) 4-55

4.11 Monitor Phase Setting Subroutine (MSETPH) 4-61

4.12 Monitor Suffix to BOCOMP (MFCOMP) 4-67

Section 5. DCC OUTPUT CONTROL

5.1 High-Speed Output to Bermuda Displays 5-1

5.2 Monitor Prefix to BODSPL (MPDSPL) 5-5

5.3 Monitor Suffix to BODSPL (MFDSPL) 5-9

5.4 Monitor High-Speed Output Processor (MYHSBM) 5-11

5.5 Monitor High-Speed Output Trap Processor (MTHSBM) 5-15

5.6 Teletype Output to Goddard 5-19

5.7 Monitor Prefix to BOTOUT (MPTOUT) 5-21

5.8 Monitor Suffix to BOTOUT (MFTOUT) 5-25

TABLE OF CONTENTS (Cont'd)

Page

5.9 Monitor Teletype Output Processor (MYTTOT) 5-29

5.10 Monitor Teletype Output Trap Processor (MTTTOT) 5-33

5.11 Sense Output to the Output Status Console 5-37

5.12 Monitor Sense Output Processor (MYSENS) 5-38

5.13 Monitor Sense Output Trap Processor (MTSENS) 5-41

Section 6. CONTROL OF CHANNELS A AND B

6.1 Station Characteristics Tape Reading Cycle 6-2

6.2 Monitor Station Characteristics Tape Reader Processor (MYSCRD) 6-5

6.3 Monitor Station Characteristics Tape Reader Trap Processor (MTENST) 6-11

6.4 The On-Line Message System 6-15

6.5 Monitor On-Line Message Processor (MYMESS) 6-21

6.6 Binary Number Card Image Macro (BINNO) 6-27

6.7 Monitor Message Check Trap Processor (MTMSCK) 6-31

6.8 Monitor Message Check Processor (MYMSCK) 6-35

6.9 Monitor End-of-Printing Trap Processor (MTENPR) 6-47

6.10 Monitor On-Line Card Processor (MYCDRD) 6-50

6.11 The Logging Process 6-51

6.12 Monitor Logging Subroutine (MSLOGG) 6-57

6.13 Monitor Logging Processor (MYSTLT) 6-63

6.14 Monitor End-of-Logging Trap Processor (MTENLG) 6-67

TABLE OF CONTENTS (Cont'd)

Page

Section 7. LOADING THE SYSTEM

7.1 Mercury System Tape Writer Program (MXSTW1) 7-2

7.2 Error Correction Code Writer Subroutine (MSWECC) 7-11

7.3 Mercury System Tape Loader (MXLOAD) 7-17

7.4 Error Correction Code Reader Subroutine (MSRECC) 7-23

7.5 Main Controller Initialization Program (M0INTL) 7-24

7.6 Monitor Initialization Processor (MYINTL) 7-29

Section 8. MONITOR SYMBOL REFERENCE LISTS

8.1 Program Lists 8-2

8.2 Communication Cells 8-7

8.3 Tables 8-11

Appendix--SPECIAL SYSTEMS FEATURES BULLETIN:

IBM 7281 I DATA COMMUNICATIONS CHANNEL

LIST OF ILLUSTRATIONS

Figure		Page
1-1	Basic Monitor Control.	
2-1	M0PRIO Program Flow Chart.	2-3
2-2	M0DCCC Program Flow Chart	2-7
2-3	M0SAVE Program Flow Chart	2-10
2-4	M0RTRN Program Flow Chart	2-13
2-5	M0DIAG Program Flow Chart.	2-16
2-6	M0QUEU Program Flow Chart	2-18
2-7	M0UNQU Program Flow Chart	2-22
2-8	TRNON Macro Flow Chart.	2-25
2-9	TRNOF Macro Flow Chart.	2-28
2-10	QUEUE Macro Flow Chart.	2-31
2-11	UNQUE Macro Flow Chart.	2-33
2-12	QENBA Macro Flow Chart.	2-35
2-13	QENBZ Macro Flow Chart.	2-38
2-14	QPSLF Macro Flow Chart	2-41
2-15	SAVE Macro Flow Chart	2-44
2-16	Basic Monitor Tables	2-49
3-1	DCC Input Control	3-2
3-2	Message Format: Radar to IBM 709	3-3
3-3	MTHSIN Program Flow Chart.	3-7/3-9

LIST OF ILLUSTRATIONS (Cont'd)

Figure		Page
3-4	MPHSIN Program Flow Chart	3-14/3-15
3-5	MFHSIN Program Flow Chart	3-19/3-20
3-6	MTTTIN Program Flow Chart	3-25
3-7	MPTIPT Program Flow Chart	3-28
3-8	MFMANI and MFMAN1 Program Flow Chart	3-31
3-9	Message Format: External Telemetry Buffer to IBM 709	3-34
3-10	MTTMIN Program Flow Chart	3-38
3-11	General Flow Diagram, MYTMIN Processor	3-45
3-12	MYTMIN Program Flow Chart	3-46/3-54
3-13	MYECOE Program Flow Chart	3-57
3-14	MYSECO Program Flow Chart	3-61
3-15	MTIWWV Program Flow Chart	3-67
3-16	MTWWWV Program Flow Chart	3-71
3-17	MTHFSC Program Flow Chart	3-77/3-78
3-18	MYMINS Program Flow Chart	3-81
4-1	General Flow Diagram, Bermuda Cycle Data Flow	4-2
4-2	General Flow Diagram, Bermuda Edit Cycle	4-3
4-3	Flow Chart: Prefix and Suffix to BEDITS	4-12/4-17
4-4	Flow Chart: Prefix and Suffix to BSMOOS	4-30/4-33
4-5	MYEBAD Program Flow Chart	4-42/4-46
4-6	MPSARC Program Flow Chart	4-50/4-51
4-7	MFSARC Program Flow Chart	4-54

LIST OF ILLUSTRATIONS (Cont'd)

Figure		Page
4-8	MPCOMP Program Flow Chart	4-59
4-9	MSETPH Program Flow Chart	4-64/4-66
4-10	MFCOMP Program Flow Chart	4-69
5-1	DCC Output Control	5-2
5-2	MPDSPL Program Flow Chart	5-7
5-3	MFDSPL Program Flow Chart	5-10
5-4	MYHSBM Program Flow Chart	5-13
5-5	MTHSBM Program Flow Chart	5-17
5-6	MPTOUT Program Flow Chart	5-24
5-7	MFTOUT Program Flow Chart	5-27/5-28
5-8	MYTTOT Program Flow Chart	5-31
5-9	MTTTOT Program Flow Chart	5-35
5-10	MYSENS Program Flow Chart	5-40
5-11	MTSENS Program Flow Chart	5-43
6-1	Station Characteristics Tape Reading Cycle	6-3
6-2	MYSCRD Program Flow Chart	6-8/6-9
6-3	MTENST Program Flow Chart	6-13
6-4	On-Line Message Cycle	6-16
6-5	MYMESS Program Flow Chart	6-25/6-26
6-6	BINNO Macro Flow Chart	6-29
6-7	MTMSCK Program Flow Chart	6-33
6-8	General Flow Diagram, MYMSCK Processor	6-41

LIST OF ILLUSTRATIONS (Cont'd)

Figure		Page
6-9	MYMSCK Program Flow Chart	6-42/6-45
6-10	MTENPR Program Flow Chart	6-49
6-11	Logging Cycle	6-52
6-12	MSLOGG: Logging Block and Calling Sequence	6-53
6-13	MSLOGG Subroutine Flow Chart	6-62
6-14	MYSTLT Program Flow Chart	6-64
6-15	MTENLG Program Flow Chart	6-68
7-1	MXSTW1 Program Flow Chart	7-3/7-7
7-2	Input and Output Tapes for MXSTW1	7-8
7-3	MSWECC and MSRECC Flow Charts	7-12
7-4	MXLOAD Program Flow Chart	7-18
7-5	MSLOAD Program Flow Chart	7-19/7-21
7-6	M0INTL Program Flow Chart	7-25
7-7	MYINTL Program Flow Chart	7-30

LIST OF TABLES

Table		Page
1-1	Monitor Control of Priority Indicators	1-8/1-9
2-1	Monitor Tables	2-50
2-2	Monitor Tables	2-51
2-3	Monitor Tables	2-52
3-1	Time Cells Used by MYTIMIN	3-43
4-1	Smoothing Parameters and Criteria Initially Set by MPSMOO	4-29
6-1	On-Line Messages	6-20

Section 1
INTRODUCTION

The principal role of Bermuda in overall Project Mercury computation is one of the most critical and exacting functions of mission surveillance and control. Bermuda must gather, process and evaluate real time information to decide if capsule insertion into orbit is adequate, an all-important decision of whether to continue the flight or command an abort. The Bermuda Program System is oriented principally toward accomplishing this task. During orbit and re-entry Bermuda is a most necessary link in ground data gathering, computing and monitoring, but these are supplemental compared to its essential importance during the capsule's late-launch and insertion periods of flight.

All logical decisions affecting the capsule's flight as well as all predictions of the capsule's future location, including the impact point, are dependent upon the IBM 709 computer's knowledge of the exact present position and velocity of the capsule. Only a real time system could provide this knowledge to the computer. A real time transmission device, the Data Communications Channel, was specifically developed to enable the transmission of vast quantities of real time data directly into computer core storage, and a real time program was designed to handle the complex data processing required to support the Mercury mission.

Unlike ordinary programming, in which the flow of the process can be predetermined prior to execution, the Bermuda Program System involves factors not under program control. Hence, the program system must control its own flow in response to outside events which are channeled to the IBM 709 through the Data Communications Channel, the DCC. The Monitor control programs must decide what to do next on the basis of what has happened in the past and what must happen in the future.

The programs which, when assembled and compiled, together compose the Bermuda Program System are classified as the Monitor programs and the processor programs. As the name implies, the Monitor programs provide the system supervision; the processor programs perform the mathematical operations.

The dynamic programming of a monitor concept brings to a complex and expansive system an inherent advantage in flexibility. The processor programs operate as programs independently of Monitor in the same sense that subroutines operate independently of the main program. These processors need know only the location of available input data and the required locations for storing the calculated results.

The Data Communications Channel and the Monitor methods of program control are described in this section. A complete description of the Bermuda processing programs is contained in MC 106, Bermuda Processing Programs.

1.1 THE DATA COMMUNICATIONS CHANNEL

In general, the most critical data must be handled in real time. To accomplish this, a real time transmission device, the IBM 7281 I Data Communications Channel (DCC), replaces channel F of the input/output system of the Bermuda IBM 709. This provides for direct subchannel connection of a variety of input/output devices operating at various transmission rates. Real time data is read in and out of assigned buffer areas in computer core storage simultaneously with the execution of the main unit central processing and the operation of the other data channels. A program interrupt occurs automatically whenever the assigned buffer area has been filled or emptied, depending upon whether an input or an output operation is taking place.

When an interrupt occurs, processing control is taken from the main program and given to those instructions in lower core memory specified by the particular interrupt. After the computer has serviced the subchannel and has satisfied the resulting conditions imposed upon the system, the pre-interrupt processing state is restored (insofar as is possible, because an immediate consequence of the interrupt may be an alteration in the processing sequence or data flow) and processing control is returned to the main program at the exact point from which it was exempted.

A multiplexor-sequencer within the DCC examines subchannels in turn, gives priority to the higher-speed subchannels as required, requests program interrupts for data transmission, and provides controls needed for subchannel operation. Data transfer through any subchannel occurs only when the sequencer is examining, i.e., is positioned at, this subchannel.

Subchannels may be turned on (activated) or turned off (deactivated) independently upon instruction from the main program, and subchannels must be activated to transfer data or initiate program interrupts (traps). Further, a main program instruction must "enable" the DCC for trapping before an interrupt request is honored. If the DCC is "disabled," traps are remembered and executed when enabled, but no data transfer takes place until the trap request is granted. Disabling the DCC has no effect, therefore, on data transmission unless there is a waiting trap.

When any of the subchannels requests a trap, all DCC data transfers are stopped until the trap is granted. For example, at any given time in the mission it may be essential that a program be completed without interruption. In programming terminology such a program would "run disabled," i.e., it would begin with an instruction to disable the computer. Normal data transfer through the DCC continues in this case. However, if an input buffer area is filled or an output

buffer area is emptied, the particular subchannel involved requests a trap. At this point the data transfer through the DCC is suspended. Data could be lost to the system in this manner except that in actual usage the speed of the IBM 709 computer and methods employed by the Monitor programs virtually preclude this possibility. The trap request is remembered and, after the program is completed, an enabling instruction is given, the trap request is granted and the transfer of data by the DCC continues. Therefore, the DCC may interrupt the main program only to the extent that the main program allows. The DCC controls the transmission of data; the main program controls the action of the DCC.

Whenever a trap is initiated (from any source) the DCC is inhibited until an enable instruction is executed. Subchannels cannot initiate traps during the time the DCC is inhibited. All trap requests that occur during this time are remembered and executed when the inhibit conditions is removed. Data from all activated subchannels is transferred while the DCC is inhibited, provided none of the subchannels requests a trap. When any of the subchannels requests a trap, all DCC data transfers are stopped until the trap is granted.

The Bermuda Program System uses 10 subchannels composed of the following four types:

a) The High-Speed subchannel for input (subchannel 1) receives a 16-bit character in parallel every eight milliseconds. This character is divided into two eight-bit parts and each is stored in separate core storage locations. For output (subchannel 2) the high-speed subchannel transmits data serially at a rate of 1000 bits per second. High-speed subchannels take priority over slower-speed subchannels.

b) The Low-Speed teletype subchannels for input (subchannels 10 and 11) receive five-bit characters in parallel from teletype equipment at a rate of as many as ten characters per second. For output (subchannel 9) a low-speed subchannel transmits a five-bit character in parallel to teletype equipment at a rate of up to ten characters per second.

c) The Sense subchannel for input (subchannel 3) receives an eight-bit character in parallel at a rate not to exceed one character every 500 microseconds. For output (subchannel 12) the sense subchannel allows the computer program to set up eight voltages and/or relays to control external equipment or display devices.

d) The Timing Control subchannel (subchannel 4) is the basic timing control mechanism for the system and provides an automatic program interrupt every one-half second. With all modifications the electronic clock requires three subchannels to: increment a core storage location every 8-1/3 milliseconds; interrupt the program after a program-specified number of either half-seconds or 8-1/3 millisecond increments, up to a maximum of 255 (subchannel 6); and be synchronized with an external timing pulse (subchannel 8).

The specific functions of the individual subchannels are described in the following paragraphs. The underscored subchannel number preceding the explanation refers to the sequencer position and, since the sequencer examines each subchannel sequentially, it also refers to the subchannel's relative priority.

Subchannel 1: High-speed input receives real time data directly from the Bermuda site's AN/FPS-16 and Verlort radars and transmits the data into assigned blocks in core storage. A complete message is received from each radar every tenth of a second. As its priority indicates, this subchannel transmits the most important data to the Bermuda Program System.

Subchannel 2: High-speed output transmits computer-processed data to the local digital/analog converters to drive the Bermuda displays.

Subchannel 3: The telemetry input subchannel allows for the transmission of telemetry messages from the capsule directly to the IBM 709 computer without manual intervention. A special storage buffer externally connected to the DCC receives telemetry messages continuously during capsule transit over the Bermuda site. At every half-second throughout the transit the computer activates subchannel 3 and the complete telemetry message is transmitted from the external buffer into core storage. Since direct input of telemetry event information may induce major changes in the data processing logic, protection is provided against invalid information or transmission errors. If, for example, the telemetry input indicates to the computer that the SECO has occurred, the computer delays action on the information for nine seconds to allow for a manual override, if necessary. The manual override is received in the telemetry message in the same manner as data directly from the capsule.

Subchannel 4: The one-half second trap subchannel provides a program interrupt every one-half second. Since much of the computer processing is based on a time period, the incrementing or decrementing of core storage locations for each half-second trap enables the computer to keep count of time. In synchronization with the operation of subchannel 4, a core storage location is incremented from 0 to 59 to keep count of the number of elapsed 8-1/3 millisecond intervals since the last one-half-second trap (60 intervals of 8-1/3 milliseconds span a half second). After the 60th interval the cell is reset to zero.

Subchannel 5: The WWV trap subchannel receives a pulse from the National Bureau of Standards radio station WWV every minute. This signal initiates a program interrupt which enables the computer to increment a cell containing a one-minute count and to ascertain the accuracy of the computer's internal time count. If the difference between the combined half-second and 8-1/3-millisecond counts and the WWV standard exceeds 8-1/3 milliseconds, an audible error signal is sounded at the Output Status Console and its WWV ERROR light is turned on.

Subchannel 6: The interval timer subchannel is available, but, at present, has no function in Project Mercury.

Subchannels 7 and 8: Not used; open.

Subchannel 9: Low-speed output transmits radar data and telemetry summaries to the Goddard Computing and Communications Center.

Subchannel 10: Low-speed input provides the Bermuda 709 computer with facilities for direct input of teletype messages. At present there is no functional requirement for direct teletype input and this subchannel is not used.

Subchannel 11: Low-speed manual input provides the Bermuda 709 computer with facilities for direct insertion of information on paper tape.

Subchannel 12: The sense output subchannel transmits to the Output Status Console an eight-bit output from a specified word in computer core storage. Each of two pairs of eight bits control an indicator on the panel of the Output Status Console. A 1 in the off bit position turns off the indicator regardless of the on bit position; a 1 in the on bit position with a zero in the off bit position turns on the indicator.

The Output Status Console contains an amber light with an audible tone indicator, and seven white lights. The amber light indicates WWV ERROR and the audible tone sounds if, when the one-minute trap occurs, the computer's timing count differs from WWV by an amount in excess of 8-1/3 milliseconds. The seven white lights indicate major computer processing conditions: LAUNCH, ABORT, HOLD, ORBIT, RE-ENTRY, BAD POINTS FPS/16, and BAD POINTS VERLORT.

The sense output subchannel transmits immediately after it is activated and then requests a trap. When the trap request is granted, the subchannel must be deactivated to prevent further transmission.

1.2 MERCURY MONITOR

The control of the Bermuda Program System depends upon the input data, upon the results of computations by the program itself, and upon a preset priority table which is capable of continuous effective modification. In non-real-time computer programs, control normally proceeds along the lines of flow prescribed by the program being executed. When real time trapping is introduced into the system, control is no longer restricted to the preset line of flow. When operating in any trapping mode, a signal is generated when a specific condition is met. Upon receipt of the signal, control immediately transfers to a predetermined location regardless of the preset lines of flow which would otherwise have been used.

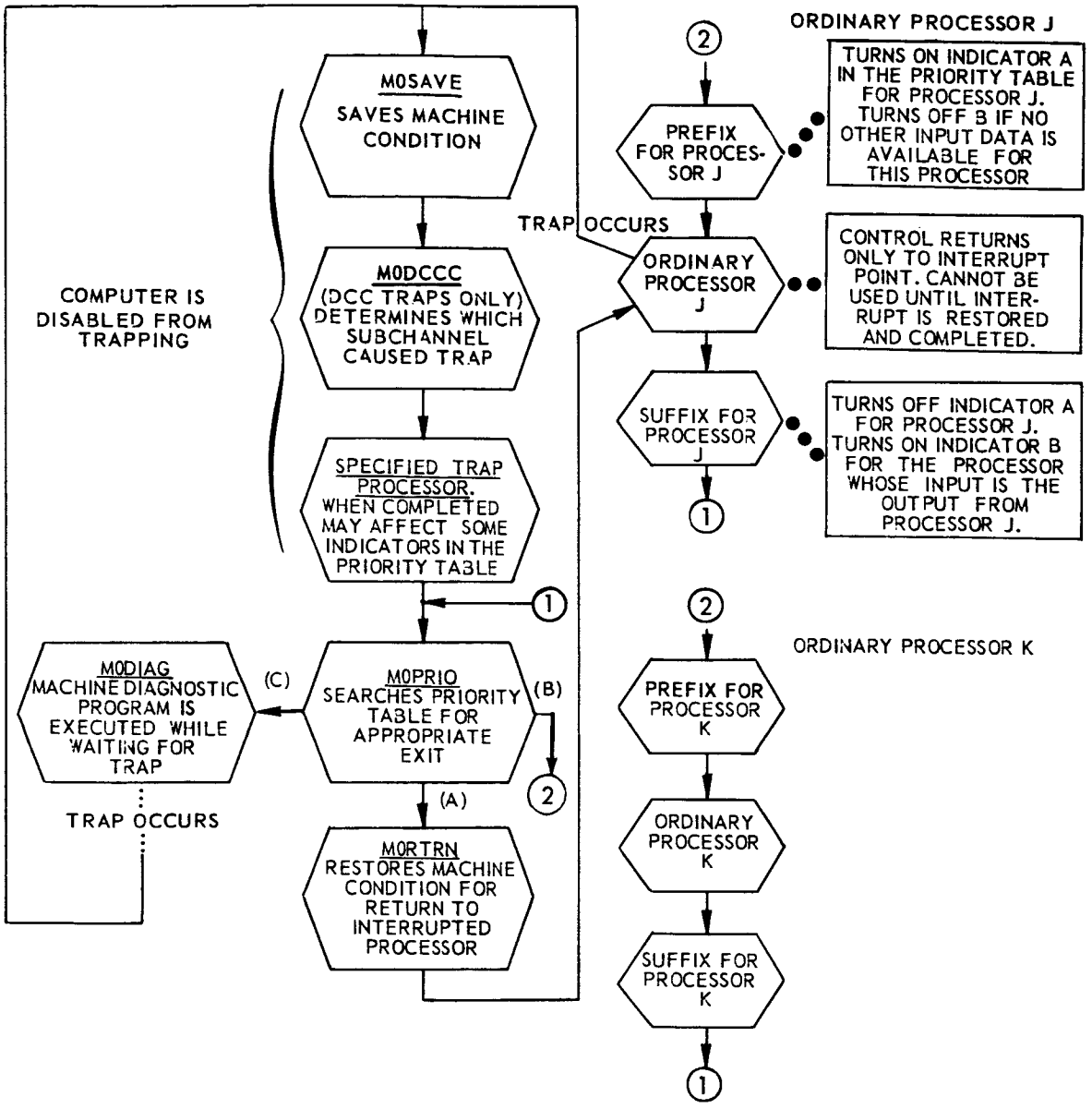
There are three states in which a computer may be operating. It may be enabled, disabled, or inhibited. When the computer is enabled, traps may occur. When disabled or inhibited, traps cannot occur even though an interrupt signal has been received. The computer may be enabled or disabled by executing the appropriate instruction. The inhibited state, however, is not entered through programming. When a trap occurs, the computer is immediately inhibited, thereby permitting no further traps until the computer is re-enabled by the program.

All input/output data transmission occurs through the Data Communications Channel simultaneously with the execution of the main program. Only when an input buffer is filled, and output buffer emptied, a half-second interval elapses, or the one-minute pulse from WWV occurs, must the present computer program be interrupted. This situation gives rise to the primary control mechanism in the Mercury Monitor, the program trap.

When the Bermuda Program System is executing any of the non-trap processors two possibilities exist; the processor may be completed or a trap may interrupt and take control from the processor.

If the computer is enabled for trapping on a given data channel, all trap requests on that channel are honored immediately. The trap may be necessary to relocate input data if the particular subchannel buffer area must be filled more than once to provide a quantity of data sufficient for processing (a reverse situation applies for output data transmission) or the trap may provide sufficient input data for the execution of a processor. The trap processor, operating while the computer is inhibited from trapping, generates control information for the Monitor controller programs. When the trap processor is completed, control passes to the basic priority routine, M0PRIO. If, in the original example, the program completes the processor, control also returns to M0PRIO. When any trap or ordinary processor is completed, control always returns to M0PRIO.

The main priority routine, M0PRIO, together with M0SAVE, M0DCCC, M0RTRN and M0DIAG comprise the Monitor system's main controller programs (see Figure 1-1). When a trap takes control from an ordinary processor, M0SAVE stores the pre-trap condition of the computer so the ordinary processor may be



EXIT FROM MOPRIO DEPENDS UPON THE PRIORITY TABLE.

GIVEN: PROCESSOR J HAS HIGHER PRIORITY THAN PROCESSOR K.

MOPRIO EXITS TO: (A) WHEN J IS HIGHEST UNSUPPRESSED PROCESSOR
(B) WHEN K IS HIGHEST UNSUPPRESSED PROCESSOR (THE INTERRUPTED PROCESSOR, J, MUST BE SUPPRESSED)
(C) WHEN NO PROCESSOR HAS BEEN INTERRUPTED OR IS READY, OR WHEN ALL INTERRUPTED OR READY PROCESSORS ARE SUPPRESSED.

FIGURE 1-1. BASIC MONITOR CONTROL

TABLE 1-1. MONITOR CONTROL OF PRIORITY INDICATORS (Sheet 1 of 2)

PROGRAM (By Priority)	INDICATOR A (In-Process) On	INDICATOR A (In-Process) Off	INDICATOR B (Ready) On	INDICATOR B (Ready) Off	INDICATOR C (Suppress) On	INDICATOR C (Suppress) Off	INDICATOR D (Suppress) On	INDICATOR D (Suppress) Off	INDICATOR E (Suppress) On	INDICATOR E (Suppress) Off
BIHSIN	MPHSIN	MPHSIN MFHSIN	MTHSIN	MPHSIN						
MYTMIN	MYTMIN	MYTMIN	MTTMIN	MYTMIN	MPCOMP	MPCOMP				
MYHSBM	MYHSBM	MYHSBM MFDSPL	MFDSPL	MYHSBM	MYHSBM	MYHSBM	MPDSPL	MFDSPL		
MYSTLT	MYSTLT	MYSTLT	MSLOGG	MYSTLT	MYSTLT	MTENLG				
MYTTOT	MYTTOT	MYTTOT	MFTOUT	MYTTOT	MPTOUT	MFTOUT MPTOUT	MYTTOT	MTTTOT		
MYSENS	MYSENS	MYSENS	MFEBAD* MSETPH* MTWWWV* MYEBAD*	MYSENS**	MYSENS	MTSENS				
MYMINS	MYMINS	MYMINS	MTHFSC	MYMINS						
MYMESS	MYMESS	MYMESS	MFCOMP* MFEBAD* MFHSIN * MFMANI * MFMAN1* MFSMOO* MFTOUT* MSLOGG * MTIWWV * MYMINS * MYEBAD* MYSCRD * MYTMIN *	MYMESS**	MYMSCK	MYMSCK	MYMESS	MTENPR	MYSCRD	MTENST MYSCRD
MYMSCK	MYMSCK	MYMSCK	MTMSCK	MYMSCK						
MYSCRD	MYSCRD	MYSCRD	MYINIL MTENST	MYSCRD	MYMESS	MTENPR				
BODSPL	MPDSPL	MFDSPL MPDSPL	MFCOMP	MPDSPL						

TABLE 1-1. MONITOR CONTROL OF PRIORITY INDICATORS (Sheet 2 of 2)

PROGRAM (By Priority)	INDICATOR A (In-Process) On	INDICATOR A (In-Process) Off	INDICATOR B (Ready) On	INDICATOR B (Ready) Off	INDICATOR C (Suppress) On	INDICATOR C (Suppress) Off	INDICATOR D (Suppress) On	INDICATOR D (Suppress) Off	INDICATOR E (Suppress) On	INDICATOR E (Suppress) Off
MYSECO	MYSECO	MYSECO	MTHFSC	MYSECO						
BIMANI	MPTIPT	MFMANI MFMAN1	MTTTIN*	MPTIPT						
BOTOUT	MPTOUT	MFTOUT MPTOUT	MTHFSC MFCOMP MYTTOT	MPTOUT	MFTOUT	MYTTOT				
MYCDRD					MYMESS	MYSCRD			MYSCRD	MTENST MYSCRD
MYECOE	MYECOE	MYECOE	MTHFSC	MYECOE						
BOCOMP	MPCOMP	MFCOMP MPCOMP	MFSARC MPSARC	MPCOMP						
BARSCH	MPSARC	MFSARC	MFSMOO	MPSARC						
BSMOOS	MPSMOO	MFMOO	MFEBAD MFEDIT	MFEDIT MPSMOO	MPEDIT MPSARC	MFEBAD MFEDIT MFSARC MPSARC				
MYEBAD	MYEBAD MFEBAD	MYEBAD	MFEDIT	MYEBAD MFEBAD	MPEDIT	MFEDIT MFEBAD				
BEDITS	MPEDIT	MFEDIT MFEBAD	MFHSIN	MPEDIT						
MYINTL	MYINTL	MYINTL	M0INTL	MYINTL						

*All indicator bits are controlled by the TRNON and TRNOF macros except for those programs marked with an asterisk, which turn on the B indicators with the QUEUE macro.

** The double asterisk indicates the B indicator is turned off with the UNQUE macro.

restored and completed at some future time. If the trap occurred through the DCC, M0DCCC (Data Communications Channel Controller) determines which subchannel of the DCC is responsible for the trap. After the trap processor is executed, control passes to M0PRIO.

The transfer of control from M0PRIO is determined from the priority table, TMPRIO (see Table 2-1). Each ordinary processor is represented in this table by a routine entry and associated control indicator bits, A, B, C, D, etc. The A, or in-process, indicator for a processor is turned on when that processor is entered, and turned off when the processor is completed. The B, or ready, indicator is turned on when input data is available for that processor, and turned off when the processor is entered, if no other input data remains. If input data may at times be supplied to a processor at a rate exceeding the processor's ability to accept this data, input queue tables are provided. But, generally, the B indicator is turned off as the A indicator is turned on. The C, D, . . . K, or suppression, indicators are turned on when for any reason a processor must not be entered, and are turned off when this restriction is no longer present. Since a processor may be suppressed for various reasons, there are as many suppression indicators as there are reasons for suppression.

When M0PRIO receives control, it scans the priority table, TMPRIO, sequentially seeking the highest priority routine for which none of the suppression indicators (C, D, . . . K) is on. If such a situation exits, M0PRIO examines the in-process and ready indicators, A and B, respectively. If neither is on, M0PRIO continues to examine the suppression indicators for the remaining entries in TMPRIO. If M0PRIO finds an unsuppressed routine with indicators A and/or B on, indicator A takes precedence over indicator B. That is, a routine may be interrupted any number of times, but it must be completed before new input data may be accepted. If M0PRIO finds that every unsuppressed routine has neither its in-process nor its ready indicator on, the program has, in effect, caught up with the data transmission and must wait for the next trap.

While waiting for the next trap, the computer executes a special machine diagnostic program, M0DIAG. This program is entered only if all ordinary processors lack input data or are suppressed and, since it is always entered at the initial location, the machine condition at the point of interrupt need not be remembered.

Control enters each non-Monitor processor through its prefix and exits through its suffix. These control the indicator bits. When the prefix is entered, the A indicator is turned on—the processor is now "in process." If no other input data is available and waiting for the processor, the B indicator is turned off. After the processor is completed, the suffix turns off the A indicator and, if the output data from this processor serves as input to another processor, the B indicator for that processor is turned on. Control always returns to M0PRIO.

1.3 SCOPE OF THE MANUAL

The Monitor programs for the IBM 709 computing system supervise the logical flow of all data processing and associated transmission at the Bermuda Control Center.

The basic Monitor control programs are described in Section 2. The Monitor control of the Bermuda computational cycle is described in logical sequence from the DCC input control, Section 3, through the control of the processing, Section 4, to the DCC output control, Section 5. Section 6 discusses the function of channels A and B in system operation: the on-line message printing, the logging of data and the reading of the Station Characteristics Tape—which operations are entirely the responsibility of Monitor.

Section 7 follows the loading of the Bermuda Program System, from the writing of the absolute tape to the initialization of the system for data processing. Section 8 provides a series of reference listings for the programs, communication cells and tables utilized in Monitor control. The mathematical or processing, programs for the Bermuda Program System are described only briefly in this manual. A detailed description of the programs appears in Bermuda Processing Programs, MC 106.

Section 2.
BASIC MONITOR CONTROL PROGRAMS

The basic monitor control functions for the Bermuda Program System are exercised by the main controller programs, by the queueing and unqueueing programs and by the associated system macros. The Main Controller Priority Program, M0PRIO, gives control to other programs in the Bermuda Program System on the basis of a dynamic priority table in which the priority indicators are controlled by the Monitor Macros, TRNOF and TRNON. These macros are used throughout the Bermuda Program System by various monitor programs—processors, trap processors, prefixes and suffixes to the processor programs.

When any program is interrupted, the Save Macro, SAVE, gives control to the Main Controller Save Program, M0SAVE, which preserves the condition of the interrupted program. The DCC Main Controller Program, M0DCCC, interprets all interrupts occurring on the Data Communications Channel and gives control to the proper trap processor. When the trap processor which services the interrupt is completed, priority conditions permitting, M0PRIO gives control to the Main Controller Return Program, M0RTRN, which restores the conditions and returns control to the interrupted program at the interrupt point. When a program must be free from interruption the Disable Macro, QENBZ, and its associated Disable Program, MYENB0, are executed. When conditions permit interrupt, the Enable Macro, QENBA, and its associated Enable Program, MYENBA, are executed.

When input data may be held in waiting for a program, the execution of the Queue Macro, QUEUE, and its associated Queue Program, M0QUEU, places an entry in the input queue table for that program. When the program is free to accept new data, the execution of the Unqueue Macro, UNQUE, and its associated Unqueue Program, M0UNQU, extracts the oldest waiting input from the first location in the queue table.

The Reference Macro, REFR, is executed only in the compilation of the Mercury Program System and generates the queue tables and the basic Monitor tables used by both the main controller programs and the system macros.

The programs and macros listed above are described in detail on the following pages.

2.1 MAIN CONTROLLER PRIORITY PROGRAM (M0PRIO)

The Monitor priority program, M0PRIO, is the center of the Monitor control system and gives precedence to the routines of the Bermuda Program System according to a priority table which is continuously modified by mission conditions.

The flow chart for the M0PRIO program is shown in Figure 2-1.

2.1.1 Input Requirements

Input to M0PRIO consists of:

a) N—parameter designating the total number of routines listed as entries in the Monitor reference table, TMREFR.

b) TMPRIO—the Monitor priority table, N cells, in order by priority number, containing in the address the location of the first instruction (the entry) of the routine. Bits S, 1, 2-9 contain the indicator bits, A, B, C, D, E, F, G, H, J and K. An indicator is on if the bit specified by the indicator is a 1, off if the bit is a zero. When indicator A is on, the routine has been initiated—the routine is in-process. When indicator B is on, input data is ready for the routine—the routine is ready. When any one of the indicators C through K is on, there is a reason why the routine should <u>not</u> be given control. The routine is suppressed.

c) TMSAVE—the Monitor save table, N cells, in order by routine number, containing in the address the location of the block in TMPANL which is used to store the machine condition when a trap occurs during the execution of that routine.

d) TMFRPR—the Monitor priority table, N cells, in order by priority number, containing in the address the routine number corresponding to the routine of the particular priority.

e) M0RTRN—the location of the Monitor Return program.

f) M0DIAG—the location of the Monitor Diagnostic program.

(The following three masks are used to determine if the specified indicator bits are on.)

g) TMNMSK - 1 + Y—mask, 177 400 000 000; indicators C through K (Y= 11, TMNMSK - 1 + Y = TMNMSK + 10).

h) TMNMSK - 1 + Z—mask, -200 000 000 000; indicators A or B (Z = 12, TMNMSK - 1 + Z = TMNMSK + 11).

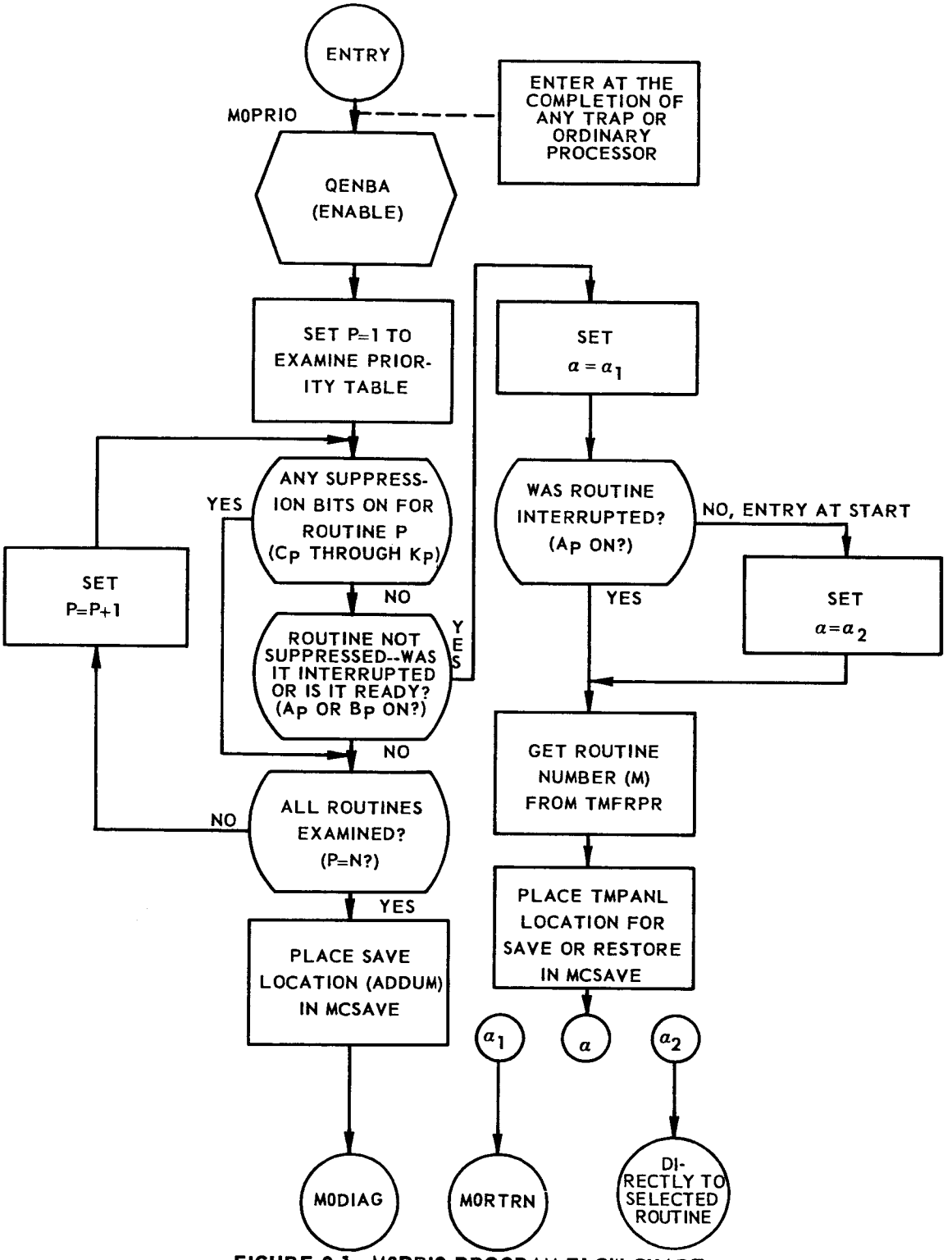


FIGURE 2-1. MOPRIO PROGRAM FLOW CHART

i) TMNMSK - 1 + A—mask, -000 000 000 000; indicator A (A = 1, TMNMSK - 1 + A = TMNMSK).

j) ADDUM—the address of a six-cell storage block within M0PRIO. When a trap takes program control from M0DIAG, this block holds the machine condition as TMPANL holds the machine condition when a routine is interrupted.

k) Macro Used—QENBA.

2.1.2 Output Requirements

M0PRIO (1) selects either one of the routines in the Bermuda Program System or the diagnostic program, M0DIAG, (2) places an appropriate entry in MCSAVE and (3) transfers control, directly or indirectly, to the selected routine.

If control returns to an interrupted routine, M0PRIO places the TMPANL location for that routine in MCSAVE and gives control to M0RTRN. M0RTRN restores the machine condition and transfers control to the interrupted routine at the exact point of the trap.

If control may be given to a routine at its initial location, M0PRIO places the TMPANL location for that routine in MCSAVE and transfers control directly to the selected routine.

If control may not be given to any routine, M0PRIO places the location of ADDUM in MCSAVE and transfers control directly to M0DIAG.

2.1.3 Method

M0PRIO receives control under one of two conditions: either a trap or an ordinary (non-trap) processor has been completed. Since all trap processors are executed while the computer is completely inhibited, and since M0PRIO has no way of knowing from where program control came, the first operation by M0PRIO is to enable the computer for trapping.

M0PRIO examines the indicators in TMPRIO from the first entry, seeking the highest priority routine for which none of the suppression indicators (C through K) is turned on. If such a situation exists, M0PRIO examines the in-process and ready indicators, A and B, respectively. If neither is on, M0PRIO continues to examine the suppression indicators for the remaining entries in TMPRIO. If M0PRIO finds that every unsuppressed routine has neither its in-process nor its ready indicator on, there is nothing for the Bermuda Program System to do. M0PRIO gives program control to M0DIAG.

If M0PRIO finds an unsuppressed routine with indicators A and/or B on, indicator A takes precedence over indicator B. That is, a routine may be interrupted any number of times, but it must be completed before new input data may

be accepted. The address of the save block in TMPANL is placed in MCSAVE and control transfers to the routine directly (indicator B on, indicator A off) or indirectly (indicator A on, indicator B may be on or off) via M0RTRN.

2.1.4 Usage

M0PRIO is entered at the completion of any routine or trap processor; the program exits to a previously interrupted routine, a new routine, or to M0DIAG.

a) Storage Required—29 locations

b) M0PRIO uses:

1) Macro—QENBA

2) Parameters—N, Y, Z and A

3) Communication Cell—MCSAVE

4) Tables—TMPRIO, TMSAVE, TMREFR, TMFRPR and ADDUM

5) Masks—TMNMSK + Y - 1, TMNMSK + Z - 1 and TMNMSK + A - 1

c) Time Required:

1) Exit to M0DIAG—5.61 + 0.048S milliseconds

2) Exit to new routine—0.30 + 0.048S + 0.144P milliseconds

3) Exit to M0RTRN—0.32 + 0.048S + 0.144P milliseconds

where:

S = the number of examined routines for which one (or more) of the suppression indicators is on.

P = the priority of the routine which receives control from M0PRIO.

2.2 DCC MAIN CONTROLLER PROGRAM (M0DCCC)

All program interrupts resulting from data transmission through the DCC are interpreted by the real time channel control program, M0DCCC.

The flow chart for the M0DCCC program is shown in Figure 2-2

2.2.1 Input Requirements

Input to M0DCCC consists of:

a) Location 00003—control word set by the hardware when a DCC trap occurs. The address part contains the location counter at the instant the trap occurred; bits 4-8 contain the number of the DCC subchannel responsible for the trap.

b) Location 00004—This location supplies the instruction following a DCC trap and must contain a transfer to M0DCCC (the instruction is actually TTR M0DCCC).

c) KMSK48—mask, 047 000 000 000, used to clear a word of all bits except 4 through 8, inclusive.

d) TMDCCC—the Monitor real time channel reference table containing an entry for each DCC subchannel used by the Bermuda Program System. The entries are arranged in inverse subchannel order, i.e., the last entry in the table contains information relative to subchannel number 1. For each entry the decrement contains the location of the DCC core storage buffer serviced by this subchannel and the address contains the location of the trap processor which services the buffer.

e) Macro Used—SAVE

The computer is inhibited during the execution of M0DCCC.

2.2.2 Output Requirements

The appropriate trap processor is given control with the accumulator containing the subchannel number in the decrement and the location of the associated core storage buffer in the address.

2.2.3 Method

M0DCCC begins by saving the machine condition with the execution of the SAVE macro and the M0SAVE program. M0DCCC then examines location 00003 to determine which DCC subchannel is responsible for the trap. With this information, M0DCCC places the subchannel number in the decrement of the accumulator and the subchannel's buffer block location into the address of the accumulator. Control is transferred directly to the specified trap processor.

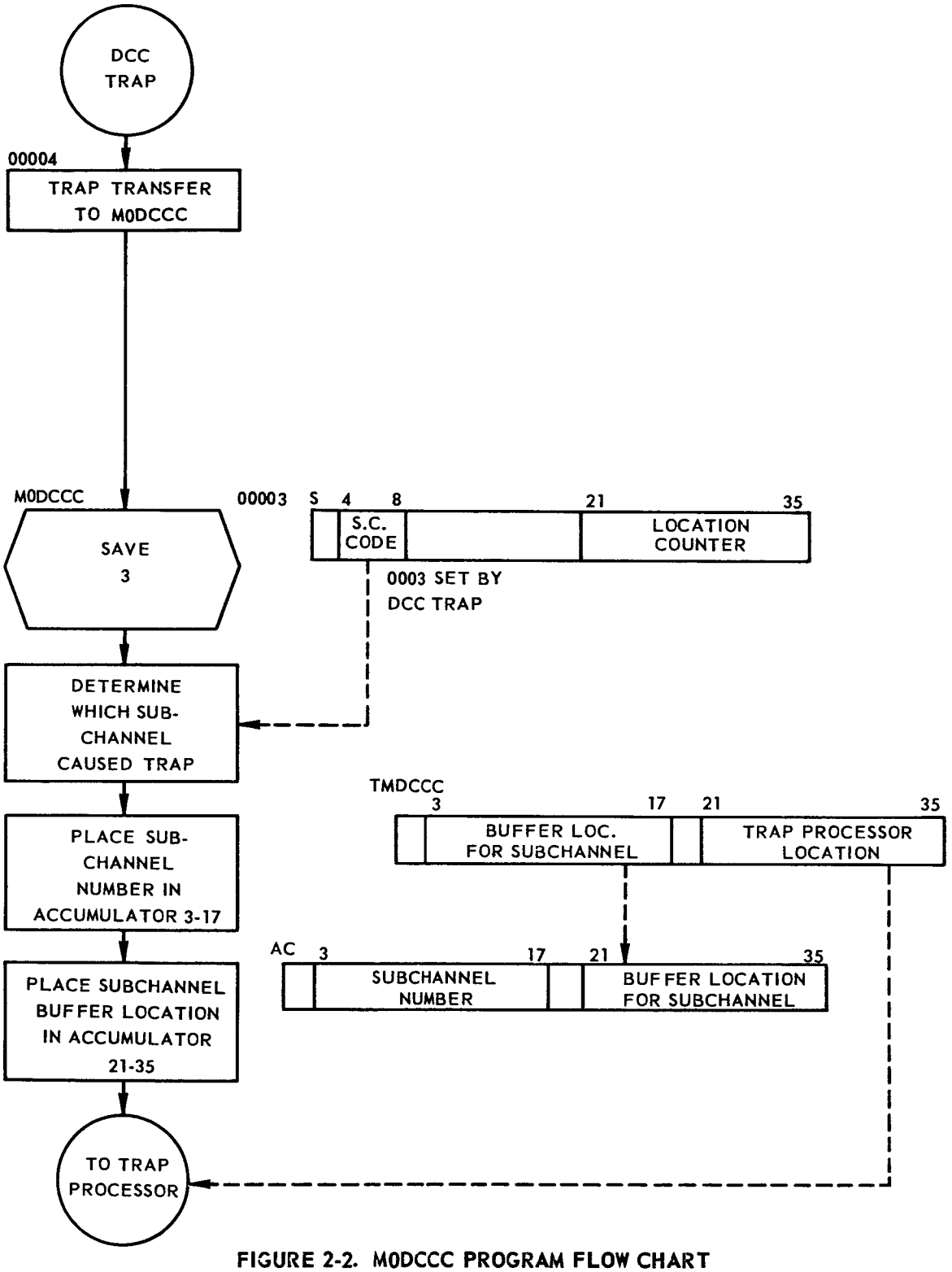


FIGURE 2-2. M0DCCC PROGRAM FLOW CHART

2.2.4 Usage

M0DCCC is entered from location 00004 following a DCC trap, and exits to the selected trap processor.

a) Storage Required—11 locations

b) M0DCCC uses:

1) Macro—SAVE

2) Communication Cell—MCX4RA

3) Table—TMDCCC

4) Mask—KMSK48

c) Time Required—if DCC trap interrupted:

1) M0PRIO—0.482 millisecond

2) M0RTRN—0.539 millisecond

3) Any other routine—1.23 milliseconds (maximum)

2.3 MAIN CONTROLLER SAVE PROGRAM (M0SAVE)

Every time a program interrupt occurs, the machine condition must be preserved if control is to return to the interrupted program at the exact point of the interrupt. M0SAVE receives control from the SAVE macro following a program interrupt and stores the machine condition in a specified block within TMPANL.

The flow chart for the M0SAVE program is shown in Figure 2-3.

2.3.1 Input Requirements

Input to M0SAVE consists of:

a) M0PRIO, AD4—the locations corresponding to the first and last instructions in M0PRIO.

b) M0RTRN, AC5—the locations corresponding to the first and last instructions in M0RTRN.

c) MCX4RA—communication cell containing in the decrement the contents of index register 4, and in the address the contents of the location counter when trap occurred.

d) MCSAVE—communication cell containing the location in TMPANL of the block in which is stored the machine condition when an interrupt occurs. This address is set by M0PRIO just prior to the transfer of program control to the selected routine.

e) MCTRAP—communication cell containing non-zero data if the interrupted program was attempting to disable from trapping when the trap occurred.

The computer is inhibited during the execution of M0SAVE.

2.3.2 Output Requirements

If M0SAVE has received control after an interrupt of M0PRIO or M0RTRN, there is no output. Otherwise, a six-cell block provided in TMPANL for the interrupted routine is filled (as shown in Figure 2-3).

2.3.3 Method

The SAVE macro sets up the calling sequence, which includes the contents of both index register 4 and the location counter when the interrupt occurred, and transfers control to M0SAVE. By examining the saved location counter value, M0SAVE determines whether or not the trap took control from M0PRIO or M0RTRN. If either of these two programs was interrupted, there is no need to save the machine condition—both programs are entered only at the initial

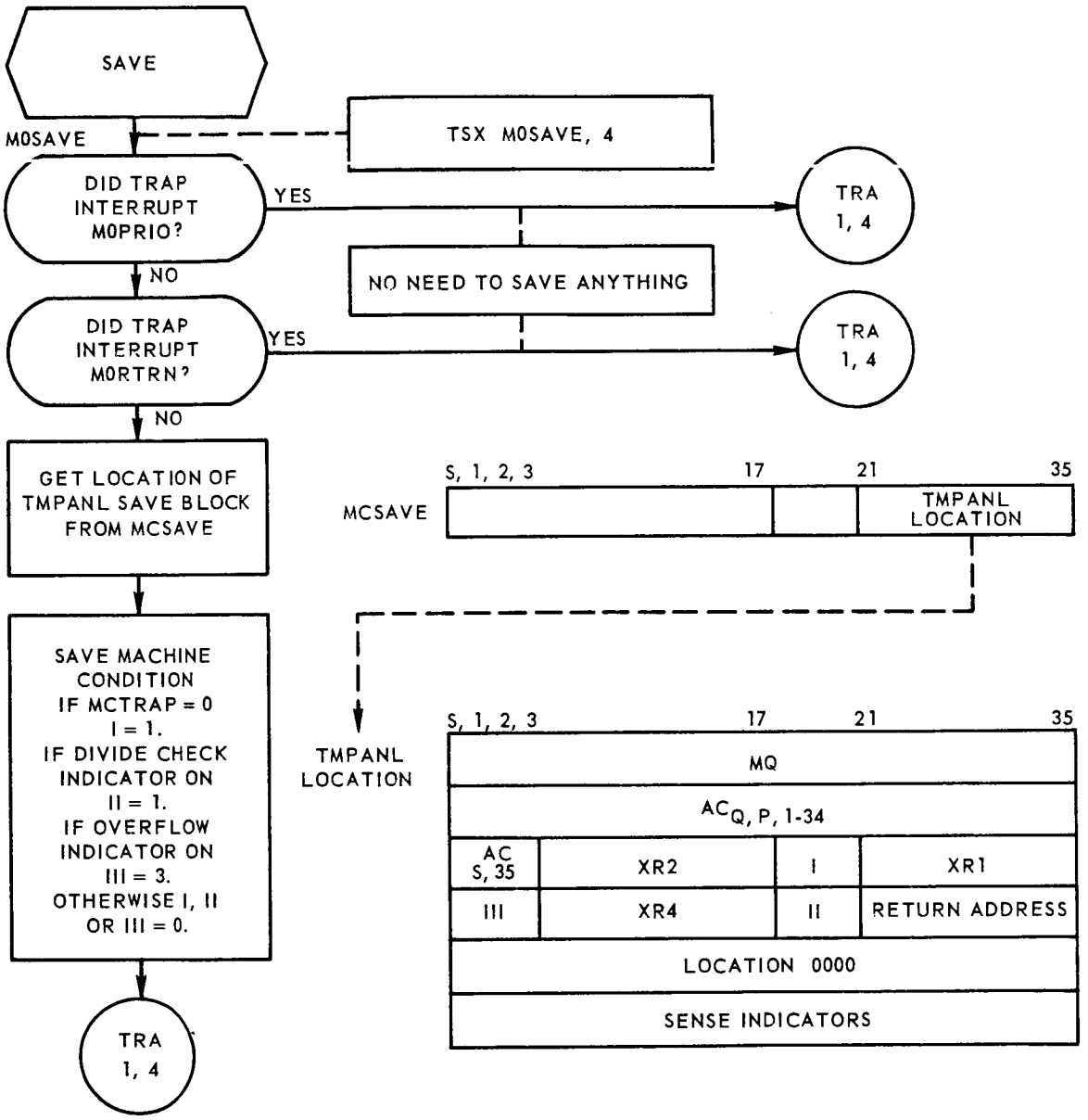


FIGURE 2-3. MOSAVE PROGRAM FLOW CHART

instruction—and control is given directly to the appropriate trap processor. Otherwise M0SAVE stores the machine condition in the reserved block in the TMPANL table, taking the location of this block from the address of the communication cell MCSAVE.

Three tests are performed in the process of saving the machine condition in TMPANL. If the divide check indicator is on, it is turned off and a notation is made in the tag of the fourth word of the TMPANL block. If the overflow indicator is on, it is turned off and a notation is made in the prefix of the fourth word of the TMPANL block. The third test is an examination of MCTRAP. If MCTRAP contains non-zero data, the interrupted program was attempting to disable when the trap occurred, and a notation is made in the tag in the third word of the TMPANL block.

The information contained in, and the arrangement of the storage block in TMPANL are illustrated in Figure 2-3.

2.3.4 Usage

M0SAVE is entered from the SAVE macro.

a) Calling Sequence:

```
α      TSX    M0SAVE, 4
α+ 1   Return
```

b) Storage Required—33 locations

c) M0SAVE uses:

1) Communication Cells—MCX4RA, MCSAVE and MCTRAP

2) Table—TMPANL

3) Locations—M0PRIO, AD4 and M0RTRN, AC5

d) Time Required:

1) If M0PRIO is interrupted—0.216 millisecond

2) If M0RTRN is interrupted—0.264 millisecond

3) Maximum—0.960 millisecond

2.4 MAIN CONTROLLER RETURN PROGRAM (M0RTRN)

When M0PRIO determines that the highest priority assignment consists of completing a routine which has been previously interrupted by a trap, control is first given to M0RTRN. This program restores the conditions of the interrupted routine exactly as they were immediately prior to the trap.

The flow chart for the M0RTRN program is shown in Figure 2-4.

2.4.1 Input Requirements

Input to M0RTRN consists of:

a) MCSAVE—the location in TMPANL of the six-cell block in which the machine condition was saved when the interrupt occurred; this cell is set by M0PRIO before control transfers to M0RTRN.

b) K00000—constant, 0.

c) KTGMSK—mask, 000 000 700 000, used to keep the contents of the tag.

d) Macro Used—QENBZ

2.4.2 Output Requirements

The machine condition is restored and the interrupted routine begins from the point of the interrupt as if it had not occurred.

M0RTRN is essentially the converse of M0SAVE and provides a restoration of: $AC_{S, Q, P, 1-35}$, $MQ_{S, 1-35}$, XR1, XR2, XR4, sense indicators, the divide check indicator, the overflow indicator, and location 00000.

If the routine had been trapped while attempting to disable, the computer is disabled by M0RTRN before control returns to the trapped routine.

2.4.3 Method

M0RTRN receives control from M0PRIO with the location of the TMPANL save block in MCSAVE. Index register 4 is restored. If the divide check or the overflow indicators were on, M0RTRN restores these indicators. Index registers 1 and 2, the entire accumulator and the multiplier-quotient registers are restored. If the routine is disabled upon return, M0RTRN executes the QENBZ macro. Location 00000 and the sense indicators are restored and control returns to the interrupted routine.

2.4.4 Usage

M0RTRN is entered from M0PRIO; it exits to the selected routine.

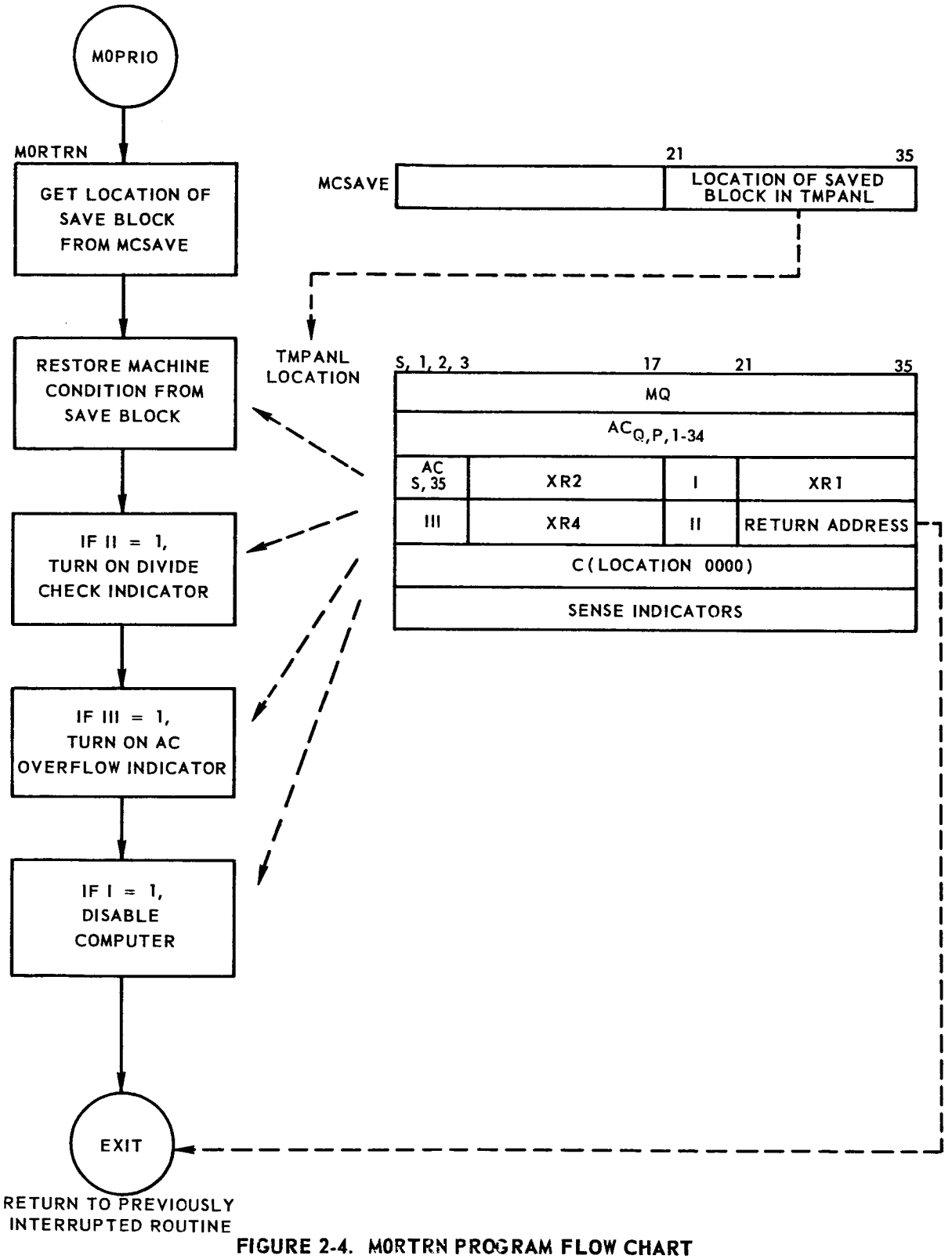


FIGURE 2-4. MORTRN PROGRAM FLOW CHART

a) Storage Required—36 locations

b) M0RTRN uses:

1) Macro—QENBZ

2) Mask—KTGMSK

3) Communication Cells—MCSAVE and MCTRAP

4) Table—TMPANL

5) Constant—K00000

c) Time Required:

1) Minimum—0.827 millisecond

2) Maximum—0.960 millisecond.

2.5 MAIN CONTROLLER DIAGNOSTIC PROGRAM (M0DIAG)

When M0PRIO is unable to give control to any other routine in the Bermuda Program System, control is given to M0DIAG—where it remains until a program interrupt occurs.

The flow chart for the M0DIAG program is shown in Figure 2-5.

2.5.1 Input Requirements

Input to M0DIAG consists of:

a) MCDIAG—communication cell containing the count of the number of times M0PRIO has given control to M0DIAG.

b) K00001—constant, 1.

2.5.2 Output Requirements

The contents of MCDIAG are incremented by one.

2.5.3 Method

M0PRIO, unable to give control to one of the referenced Bermuda Program System routines, gives control to M0DIAG. M0DIAG takes the value contained in MCDIAG, adds one to this value, and stores it back in MCDIAG. M0DIAG halts and waits for a trap.

2.5.4 Usage

M0DIAG is entered from M0PRIO; it exits to the lower core location specified by the trap.

a) Storage Required—four locations

b) M0DIAG uses:

1) Communication Cell—MCDIAG

2) Constant—K00001

c) Time Required—0.096 millisecond

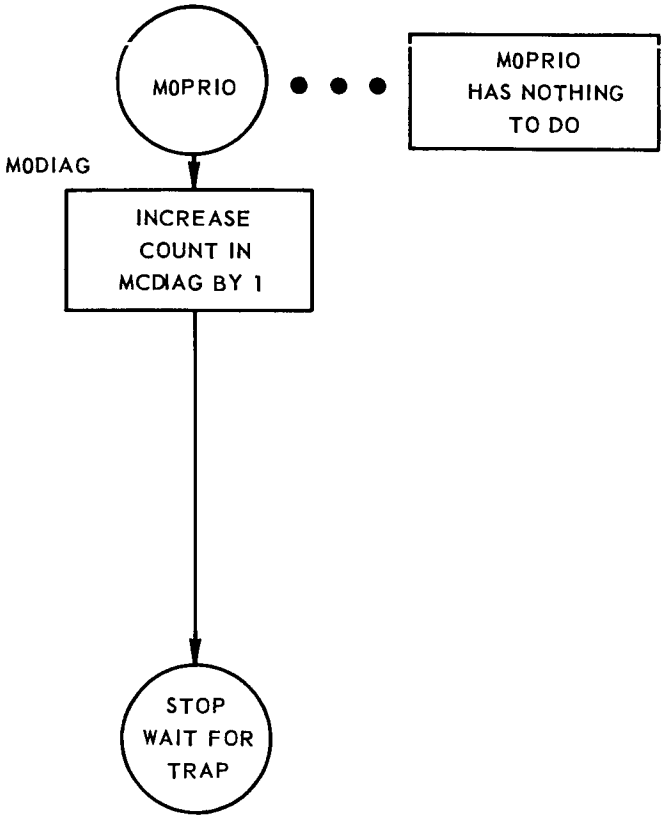


FIGURE 2-5. MODIAG PROGRAM FLOW CHART

2.6 QUEUE PROGRAM (M0QUEU)

For some routines in the Bermuda Program System, situations exist where input data is supplied to a routine at a rate exceeding the routine's ability to accept it or to dispose of the computed output. Furthermore, a rate of output determined by the output computations may not be desirable since output devices frequently cannot operate at speeds comparable to an IBM 709 computer and generally operate asynchronously with data processing.

To preserve this input, queue tables allow for the stacking of data until such time as the restrictions of the situation permit processing or transmission. M0QUEU stores entries in the queue tables.

The flow chart for the M0QUEU program is shown in Figure 2-6.

2.6.1 Input Requirements

Input to M0QUEU consists of:

a) XR4—contains the 2's complement of the location counter at the instruction which transfers control to M0QUEU.

b) 1, 4—contains in the address the location (DL) of the word to be placed in the queue table, and in the decrement the routine number (N) which specifies the routine to be queued. (For routines whose input requirements for execution exceed the capacity of one core location, DL is the location of a block of input data. For those routines whose input consists of data contained within a single location, DL may be the actual location of the data, e.g., the data itself is contained within the queue table. In either case, the QUEUE macro and the M0QUEU program operate identically; the distinction of DL is the responsibility of the routines which use queue tables.)

c) 2,4—the return address at the completion of M0QUEU.

d) TMQKEY + N - 1—contains in the decrement the current number of entries in the queue table, and in the address the location of the queue table for routine N.

e) TMQKY2 + N - 1—contains in the decrement the 2's complement of the maximum number of queue entries for routine N.

f) TMREFR + N - 1—contains the location of the entry in the priority table, TMPRIO, for routine N.

g) TMNMSK + B - 1—mask, 200 000 000 000, used to turn on indicator B in TMPRIO for routine N (B = 2, TMNMSK + B - 1 = TMNMSK + 1).

QUEUE	MACRO	DL, N
	TSX	MOQUEU
(1,4)	PZE	DL, 0, N
(2,4)	RETURN	

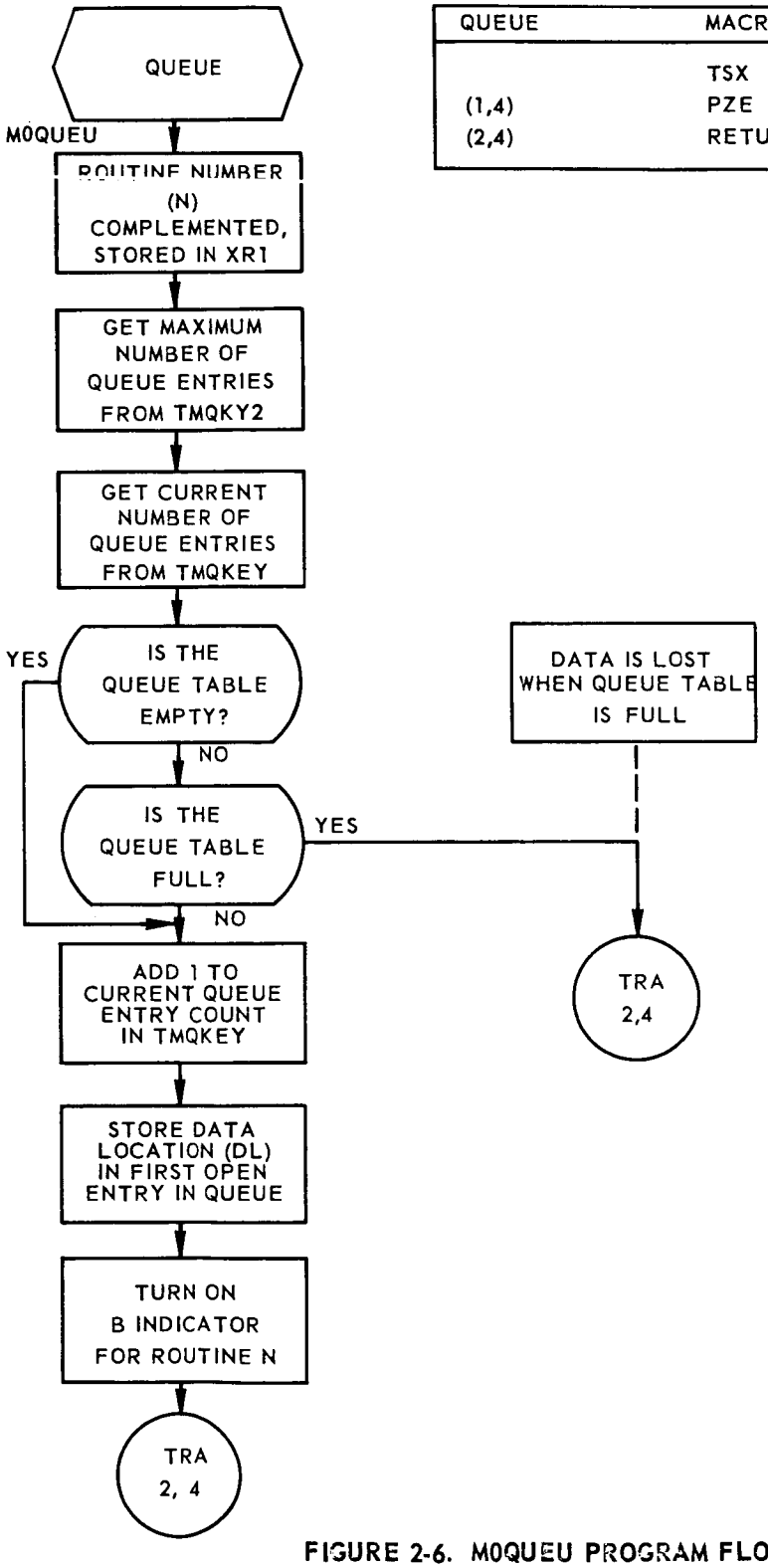


FIGURE 2-6. MOQUEU PROGRAM FLOW CHART

h) KD0001—constant, 000 001 000 000.

The computer is disabled during the execution of M0PRIO.

2.6.2 Output Requirements

If the queue table is full upon entry, there is no output—the new entry is lost.

If the queue table is not full:

a) TMQKEY + N - 1—the current number of queue entries in the decrement is incremented by one.

b) QN—the word in location DL is stored as an entry in the queue table, QN. The location in the queue table is determined by the sum of the queue table location and the number of current entries in the queue, that is, A(TMQKEY + N - 1) + D(TMQKEY + N - 1).

c) TMPRIO + P_N - 1—indicator B for routine N (having priority P_N) is turned on.

No provision is made by M0QUEU to save the machine condition. The contents of all index registers, as well as the accumulator, are destroyed by this program.

2.6.3 Method

The QUEUE macro supplies the calling sequence and transfers control to M0QUEU. The current entry count is compared to the maximum allowable number of entries in the queue table. If the queue table is not already full, the location of the input data is stored in the first empty location, the count of entries is increased by one, indicator B for the routine using the input queue is turned on, and control returns to the routine.

If the comparison determines that the queue table was already full, control returns immediately to the routine.

2.6.4 Usage

M0QUEU is entered from the QUEUE macro.

a) Calling Sequence:

α	TSX	M0QUEU,4
$\alpha + 1$	PZE	DL, 0, N
$\alpha + 2$	Return	

b) Storage Required—17 locations.

c) M0QUEU uses:

1) Parameter—B

2) Tables—TMQKEY, TMQKY2, TMREFR and TMPRIO

3) Constant—KD0001

4) Mask—TMNMSK + B - 1

d) Time Required:

1) Queue full—0.240 millisecond

2) Queue empty—0.407 millisecond

3) Queue neither full nor empty—0.432 millisecond

2.7 UNQUEUE PROGRAM (M0UNQU)

The M0UNQU program supplies input data for those routines which have queue tables.

The flow chart for the M0UNQU program is shown in Figure 2-7.

2.7.1 Input Requirements

Input to M0UNQU consists of:

a) XR4—contains the 2's complement of the location counter at the instruction which transfers control to M0UNQU.

b) 1,4—contains in the decrement the routine number (N) for the routine to which M0UNQU is to supply input data, and in the address the location (DL) into which the first entry in the queue table is placed. (For routines whose input requirements for execution exceed the capacity of one core location, DL is the address of a block of input data. For those routines whose input consists of data contained within a single location, DL may be the actual location of the data, so the queue tables contain the actual input data. The distinction of DL is the responsibility of the routines which use the queue tables.)

c) 2,4—the return address at the completion of M0UNQU.

d) TMQKEY + N - 1—contains in the decrement the current number of entries in the queue table, and in the address the location of the queue table for routine N.

e) TMREFR + N - 1—contains in the address the location of the entry in the priority table, TMPRIO, for routine N.

f) TMFMSK - 1 + B—mask, -177 777 777 777, used to turn off indicator B in TMPRIO for routine N (B = 2, TMFMSK - 1 + B = TMFMSK + 1).

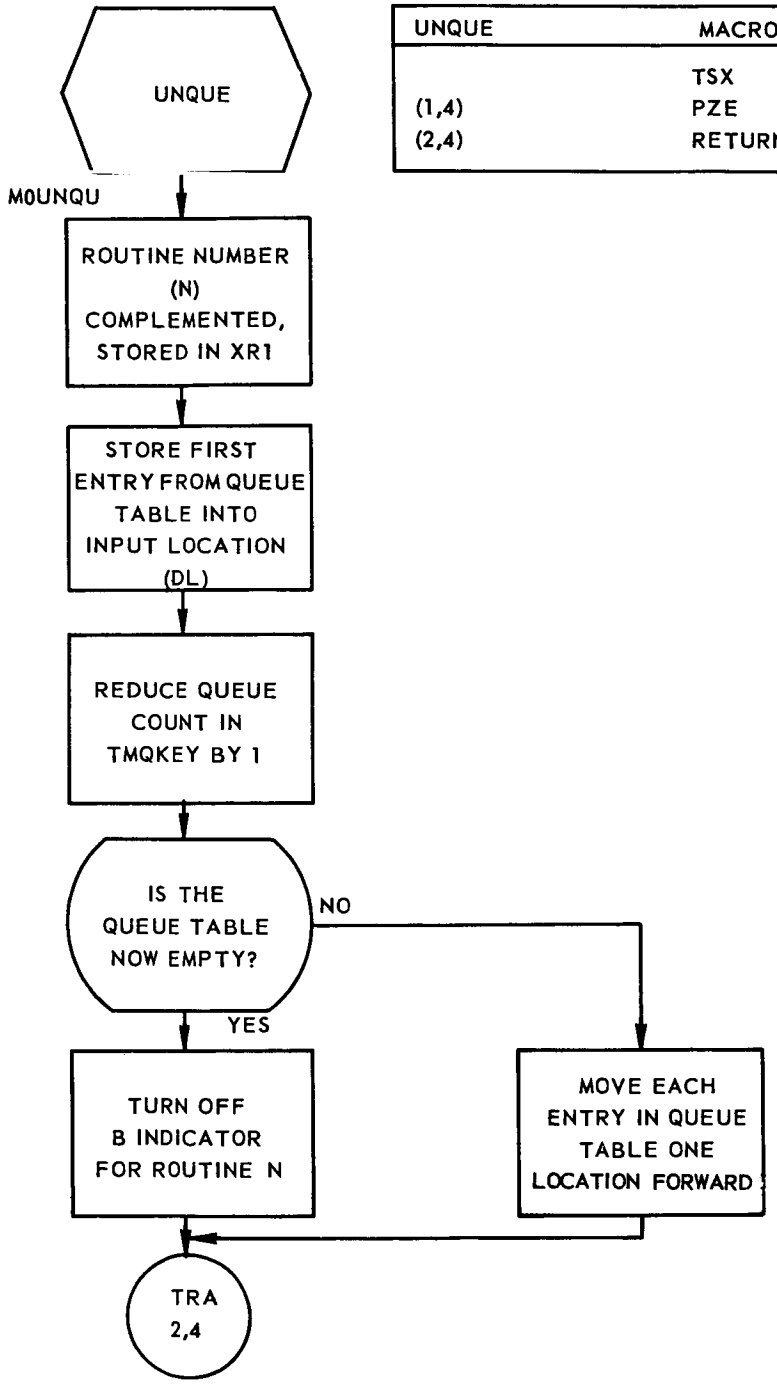
g) KD7777—constant, 077 777 000 000.

h) K00001—constant, 000 000 000 001.

The computer is disabled during the execution of M0UNQU.

2.7.2 Output Requirements

The first entry in the queue table, QP_N, is placed in the specified location, DL. The contents of the decrement (the queue entry count) are decreased by one (TMQKEY + N - 1).



UNQUE	MACRO	DL, N
	TSX	M0UNQU,4
(1,4)	PZE	DL,0,N
(2,4)	RETURN	

FIGURE 2-7. M0UNQU PROGRAM FLOW CHART

If the queue table has been emptied, indicator B for routine N is turned off (TMPRIO + P - 1).

If the queue table contains further input entries, each entry in the queue table is advanced one location so the oldest entry is always at the first location in the queue table.

No provision is made by M0UNQU to save the machine condition. The contents of all index registers and the accumulator are destroyed by this program.

2.7.3 Method

The UNQUE macro supplies the calling sequence and transfers control to M0UNQU. The first entry in the queue table is placed in the location specified by the calling sequence. The current entry count is reduced by one. If the queue is now empty, indicator B in the priority table is turned off. If the queue is not empty, each entry is advanced one location toward the front of the table and indicator B for this routine in the priority table remains on to designate that more input data remains for the routine. Control returns to the routine.

2.7.4 Usage

M0UNQU is entered from UNQUE.

a) Calling Sequence:

α	TSX	M0UNQU, 4
$\alpha+1$	PZE	DL, 0, N
$\alpha+2$	Return	

b) Storage Required—20 locations

c) M0UNQU uses:

1) Parameter—B

2) Tables—TMQKEY, TMREFR, TMQKYZ and TMPRIO

3) Constants—KD7777 and K00001

4) Mask—TMFMSK + B - 1

d) Time Required:

1) One entry in queue—0.348 millisecond

2) More than one entry in queue—(0.096E + 0.240) millisecond (where E is the number of entries in the queue table).

2.8 TURN ON MACRO (TRNON)

The TRNON macro turns on an indicator in the Monitor priority table and is used throughout the Monitor programs.

The flow chart for the TRNON macro is shown in Figure 2-8.

2.8.1 Input Requirements

The TRNON macro is defined:

```
TRNON    MACRO    ALPHA, I
         CAL      TMNMSK + ALPHA - 1
         ORS*     TMREFR + I - 1
         END
```

where the parameters are:

a) ALPHA—the indicator to be turned on. The indicators are located in the first ten bits (S, 1 through 9) of the entries in the Monitor priority table, TMPRIO. There are ten indicators: A, B, C, D, E, F, G, H, J and K. A is numerically equal to 1 (the first bit), B is equal to 2 (the second bit), etc. . . . K is equal to 10 (the tenth bit).

b) I—the routine number of the routine controlled by the TRNON macro.

c) TMNMSK + ALPHA - 1—mask with a zero in every bit position except that corresponding to the indicator designated by ALPHA. There are ten masks in the table, one for each value of ALPHA.

d) TMREFR + I - 1—contains in the address the location of the entry in TMPRIO for routine I.

2.8.2 Output Requirements

An indicator is turned on for routine I in the priority table (TMPRIO + P_I - 1).

2.8.3 Method

The mask containing a zero in every bit position except that of the indicator to be turned on is supplied by the TMNMSK table. The mask is matched by an ORS instruction to the routine's location in TMPRIO and the designated indicator is turned on.

2.8.4 Usage

Entry into TRNON has no restrictions; the macro exits to the second instruction following the entry.

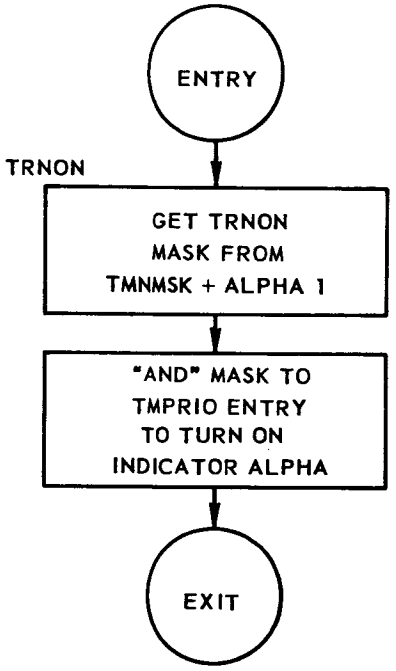


FIGURE 2-8. TRNON MACRO FLOW CHART

a) Storage Required—two locations

b) TRNON uses:

1) Parameters—ALPHA and I

2) Tables—TMREFR and TMPRIO

3) Mask—TMNMSK + ALPHA - 1

c) Time Required—0.060 millisecond

2.9 TURN OFF MACRO (TRNOF)

The TRNOF macro turns off an indicator in the Monitor priority table and is used throughout the Monitor programs.

The flow chart for the TRNOF macro is shown in Figure 2-9.

2.9.1 Input Requirements

The TRNOF macro is defined:

```
TRNOF    MACRO    ALPHA, I
         CAL      TMFMSK + ALPHA - 1
         ANS*     TMREFR + I - 1
         END
```

where the parameters are:

a) ALPHA—the indicator to be turned off. The indicators are located in the first ten bits (S, 1 through 9) of the entries in the Monitor priority table, TMPRIO. There are ten indicators: A, B, C, D, E, F, G, H, J and K. A is numerically equal to 1 (the first bit), B is equal to 2 (the second bit), etc. . . . K is equal to 10 (the tenth bit).

b) I—the routine number of the routine controlled by the TRNOF macro.

c) TMFMSK + ALPHA - 1—mask with a 1 in every bit position except that corresponding to the indicator designated by ALPHA. There are ten masks in the table, one for each value of ALPHA.

d) TMREFR + I - 1—contains in the address the location of the entry in TMPRIO for routine I.

2.9.2 Output Requirements

An indicator is turned off for routine I in the priority table (TMPRIO + P_I - 1).

2.9.3 Method

The mask containing a 1 in every bit position except that of the indicator to be turned off is supplied by the TMFMSK table. The mask is matched by an ANS instruction to the routine's location in TMPRIO and the designated indicator is turned off.

2.9.4 Usage

Entry into TRNOF has no restrictions; TRNOF exits to the second instruction following the entry

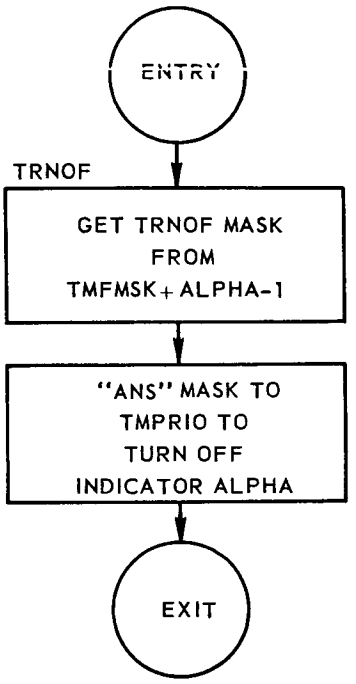


FIGURE 2-9. TRNOF MACRO FLOW CHART

a) Storage Required—two locations

b) TRNOF uses:

1) Parameters—ALPHA, I

2) Tables—TMREFR and TMPRIO

3) Mask—TMFMSK + ALPHA - 1

c) Time Required—0.072 millisecond

2.10 QUEUE MACRO (QUEUE)

The QUEUE macro establishes the calling sequence for, and transfers control to, M0QUEU, which services the queue tables.

The flow chart for the QUEUE macro is shown in Figure 2-10.

2.10.1 Input Requirements

The QUEUE macro is defined:

```
QUEUE    MACRO    N, DL
         TSX      M0QUEU,4
         PZE      DL, 0, N
         END
```

where the parameters are:

a) N—the routine number which identifies the routine to be queued.

b) DL—the location of the word to be stored as an entry in the queue table.

The computer is disabled during the execution of the QUEUE macro.

2.10.2 Output Requirements

Control is transferred to M0QUEU with the data location (DL) in the address and the routine number (N) in the decrement of the second word of the calling sequence. Index register 4 contains the 2's complement of the location counter at the transfer instruction.

2.10.3 Method

Since the second word of the calling sequence is established during compilation, QUEUE executes only one instruction, a transfer (TSX) to M0QUEU.

2.10.4 Usage

Entry into QUEUE has no restrictions, other than that the computer must be disabled; QUEUE exits to M0QUEU.

a) Storage Required—two locations

b) QUEUE uses the parameters, DL and N.

c) Time Required—0.024 millisecond

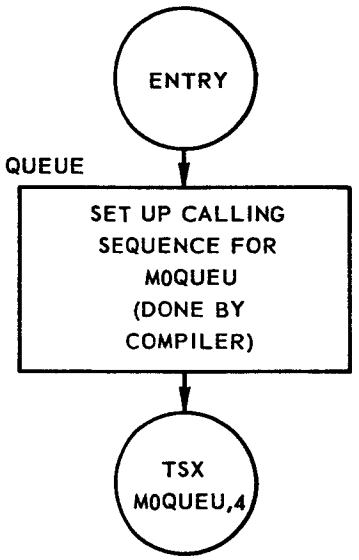


FIGURE 2-10. QUEUE MACRO FLOW CHART

2.11 UNQUEUE MACRO (UNQUE)

The UNQUE macro establishes the calling sequence for and transfers control to M0UNQU.

The flow chart for the UNQUE macro is shown in Figure 4-11.

2.11.1 Input Requirements

The UNQUE macro is defined:

```
UNQUE    MACRO    N, DL
         TSX      M0UNQU, 4
         PZE      DL, 0, N
```

where the parameters are:

a) N—the routine number which identifies the routine to be unqueued.

b) DL—the location into which the first entry in the queue table for routine N is placed.

The computer is disabled during the execution of the UNQUE macro.

2.11.2 Output Requirements

Control is transferred to M0UNQU with the data location (DL) in the address and the routine number (N) in the decrement of the second word of the calling sequence. Index register 4 contains the 2's complement of the location counter at the transfer instruction.

2.11.3 Method

Since the second word of the calling sequence is established during compilation, UNQUE executes only one instruction, a transfer (TSX) to M0UNQU.

2.11.4 Usage

Entry into UNQUE has no restrictions, other than that the computer must be disabled; UNQUE exits to M0UNQU.

a) Storage Required—two locations

b) UNQUE uses the parameters, N and DL.

c) Time Required—0.024 millisecond

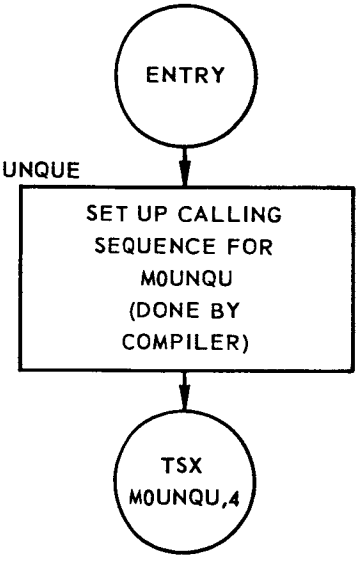


FIGURE 2-11. UNQUE MACRO FLOW CHART

2.12 ENABLE MACRO (QENBA)/ENABLE PROGRAM (MYENBA)

The QENBA macro enables the computer for data channel traps through the execution of the MYENBA program. The QENBA macro is used throughout the Mercury Program System, but generally following the completion of a routine for which the computer is disabled.

The flow chart for the QENBA macro is shown in Figure 2-12.

2.12.1 Input Requirements

The QENBA macro is defined:

```
QENBA    MACRO
         STZ      MCTRAP
         XEC      MYENBA
         END
```

where MYENBA is defined:

a) MYENBA ENB K40003

b) K40003—constant, 000 000 040 003

2.12.2 Output Requirements

The computer is enabled for data channel traps on channel A, channel B and the DCC. MCTRAP, a communication cell, is set to zero. This communication cell is used by M0SAVE and is explained in the QENBZ macro writeup (Subsection 4.13).

2.12.3 Method

The communication cell, MCTRAP, is set to zero and the MYENBA program is executed. MYENBA consists of an enable (ENB) from a location in which bits 21, 34, and 35 contain 1's. These bits represent the DCC, channel B and channel A, respectively.

2.12.4 Usage

Entry into QENBA has no restrictions, except that the macro may not be used from the point at which a trap occurs until the completion of the trap processor. QENBA exits to the second location following entry.

a) Storage Required—two locations

b) QENBA uses:

1) Program—MYENBA

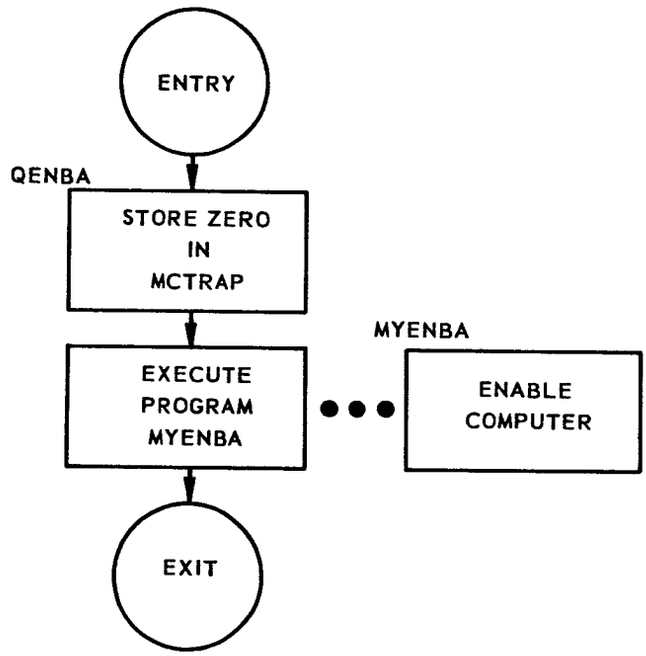


FIGURE 2-12. QENBA MACRO FLOW CHART

2) Communication Cell—MCTRAP

3) Constant—K40003

c) Time Required—0.060 millisecond

2.13 DISABLE MACRO (QENBZ)/DISABLE PROGRAM (MYENB0)

The QENBZ macro disables the computer from trapping through the execution of the MYENB0 program. The QENBZ macro is used throughout the Mercury Program System whenever any series of instructions must be completed without interruption. The most common examples of routines which run disabled are MSLOGG, M0QUEU and M0UNQU.

The flow chart for the QENBZ macro is shown in Figure 2-13.

2.13.1 Input Requirements

The QENBZ macro is defined:

```
QENBZ    MACRO
         STL      MCTRAP
         XEC      MYENB0
         NOP
         END
```

where MYENBA is defined:

a) MYENBA ENB K00000

b) K00000—constant, 0

2.13.2 Output Requirements

The computer is disabled from data channel traps; the MCTRAP communication cell is set to non-zero.

2.13.3 Method

The communication cell, MCTRAP, is set to non-zero and the program MYENB0 is executed. MYENB0 consists of an enable (ENB) from a location containing all zeroes.

MCTRAP, used in conjunction with QENBZ, QENBA and M0SAVE, is non-zero when the computer is disabled or when the program is attempting to disable through the execution of the QENBZ macro. Under certain conditions a trap may occur during the instruction to disable (ENB K00000). Before the trap processor is given control, the condition of the interrupted program is saved for the return and, if MCTRAP is non-zero M0SAVE makes a notation in the save block in TMPANL. When priority considerations allow control to return to the interrupted program, M0RTRN determines from the notation set by M0SAVE whether the computer should be disabled before return.

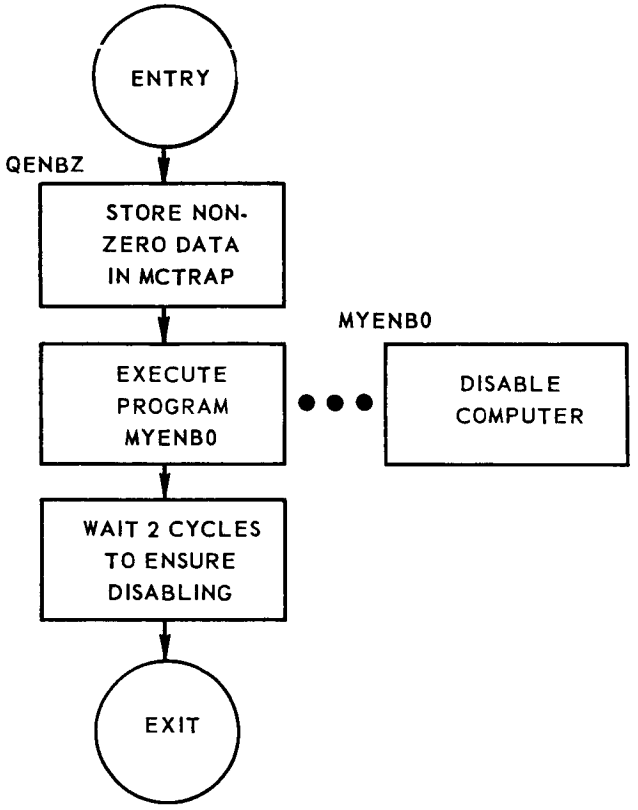


FIGURE 2-13. QENBZ MACRO FLOW CHART

The NOP instruction is further protection against a trap interrupting a routine which should run disabled. After the NOP instruction has been completed, a trap signal does not interrupt the program (but trap requests are remembered until the computer is enabled).

2.13.4 Usage

Entry into QENBZ has no restriction; QENBZ exits to the third location following entry.

a) Storage Required—three locations

b) QENBZ uses:

1) Program—MYENB0

2) Communication Cell—MCTRAP

3) Constant—K00000

c) Time Required—0.084 millisecond

2.14 PRESENT SENSE LINE MACRO (QPSLF)/PRESENT SENSE LINE PROGRAM (MYPSLF)

The QPSLF macro activates and/or deactivates the individual subchannels of the Data Communications Channel through the execution of the MYPSLF program. The QPSLF macro is used throughout the Mercury Program System.

The flow chart for the QPSLF macro is shown in Figure 2-14.

2.14.1 Input Requirements

The QPSLF macro is defined:

```
QPSLF     MACRO
          XEC       MYPSLF
          END
```

where MYPSLF is defined:

a) MYPSLF PSLF MCACTV (NOTE: PSLF [Present Sense Lines on Channel F] is recognized by Mercury SOS but is not a standard SCAT instruction. The operation code for PSLF is -0666.)

b) MCACTV—communication cell used to activate and/or deactivate subchannels of the DCC. Bits 1-32 correspond to subchannels 1-32, respectively. A 1 in bit position 1 activates (turns on) subchannel 1 of the DCC; a zero in bit position 2 deactivates (turns off) subchannel 2 of the DCC, etc. A 1 in the S bit position of the mask turning on the 8.3-millisecond clock restarts the clock from zero.

2.14.2 Output Requirements

Any or all of the DCC subchannels are activated and/or deactivated according to the bit configuration of MCACTV.

2.14.3 Method

The QPSLF macro executes the single instruction program, MYPSLF, which presents sense lines on channel F (the DCC physically replaces data channel F) from the location MCACTV.

2.14.4 Usage

Entry into QPSLF has no restrictions; it exits to the first location following entry.

a) Storage Required—one location

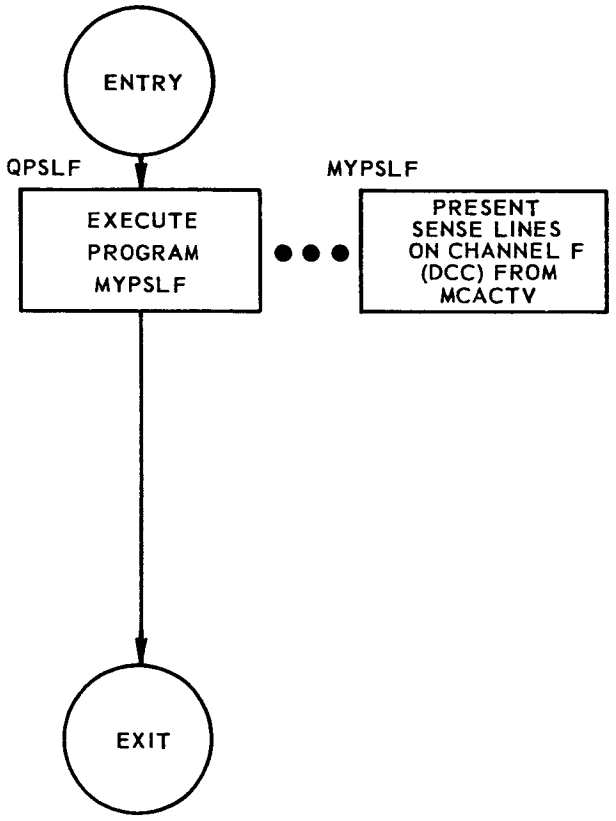


FIGURE 2-14. QPSLF MACRO FLOW CHART

b) QPSLF uses:

1) Program—MYPSLF

2) Communication Cell—MCACTV

c) Time Required—0.036 millisecond

2.15 SAVE MACRO (SAVE)

Every time a program interrupt occurs, the machine condition must be preserved if control is to return to the interrupted program at the exact point of the interrupt. The SAVE macro prepares information for, and transfers control to, the saving routine, M0SAVE. All non-DCC trap processors execute the SAVE macro before processing data. SAVE is executed by M0DCCC for DCC trap processors.

The flow chart for the SAVE macro is shown in Figure 2-15.

2.15.1 Input Requirements

```
SAVE    MACRO   CW
        SXD     MCX4RA,4
        LXA     CW, 4
        SXA     MCX4RA,4
        TSX     M0SAVE,4
        END
```

where the single parameter, CW, is the decimal equivalent of the absolute location of the cell set by the IBM 709 computer when a trap occurs. CW is zero for floating point traps, 3 for DCC traps, 10 for channel A traps, and 12 for channel B traps. The location specified by CW contains in the address the value of the location counter when the trap occurred. For DCC traps, 3_{4-8} contains the number of the subchannel responsible for the trap.

The computer is inhibited during the execution of the SAVE macro.

2.15.2 Output Requirements

The output of SAVE consists of a single communication cell, MCX4RA, which contains in the address the contents of the location counter when the trap occurred, and contains in the decrement the contents of index register 4.

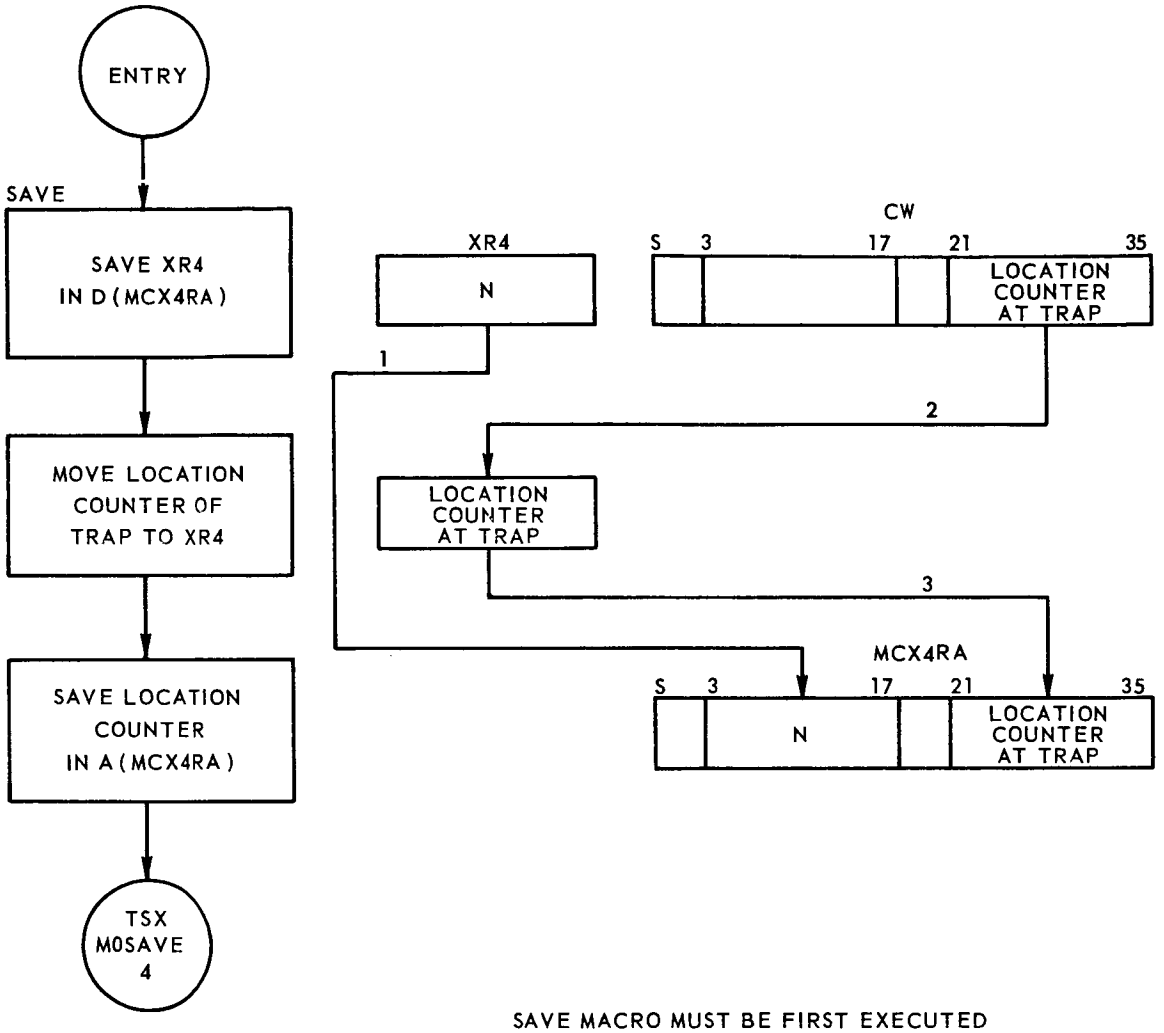
Index register 4 contains the 2's complement of the location counter at the instruction which transfers control to M0SAVE.

2.15.3 Method

The SAVE macro stores the contents of index register 4 into the decrement of MCX4RA. With index register 4 available for data relocation, SAVE uses it to move the location counter value at the interrupt from CW into the address of MCX4RA. SAVE then transfers program control to M0SAVE and sets index register 4 to the 2's complement of the current value of the location counter.

2.15.4 Usage

Entry into save is made from a transfer instruction in lower core, following a trap; SAVE exits to M0SAVE.



SAVE MACRO MUST BE FIRST EXECUTED PROGRAM AFTER EVERY TRAP.

LOCATION COUNTER READING SET IN CONTROL WORD (CW) BY THE COMPUTER WHEN TRAP OCCURS.

FIGURE 2-15. SAVE MACRO FLOW CHART

a) Storage Required—four locations

b) SAVE uses:

   1) Parameter—CW

   2) Communication Cell—MCX4RA

c) Time Required—0.096 millisecond

2.16 REFERENCE MACRO (REFR)

The REFR macro constructs the basic Monitor tables for the Bermuda Program System: TMPRIO, TMREFR, TMFRPR, TMSAVE, TMQKEY, TMQKY2 and the blocks for queuing.

2.16.1 Input Requirements

The REFR macro is defined:

```
REFR     MACRO    N, ENTRY, P, QNAME, QLENG
QNAME    BSS      QLENG
         ORG      TMREFR +N-1
         PZE      TMPRIO+ P-1
         ORG      TMPRIO+ P-1
         PZE      ENTRY
         ORG      TMFRPR+ P-1
         PZE      N
         ORG      TMSAVE + N-1
         PZE      TMPANL+ 6*N-11
         ORG      TMQKEY + N-1
         PZE      QNAME
         ORG      TMQKY2 + N-1
         PZE      0, 0, -QLENG
         ORG      QNAME +QLENG
         END
```

where the parameters are defined:

a) N—the routine number which is relative to the position of the routine in the reference table. Each ordinary (non-trap) processor in the Bermuda Program is represented by both a routine number and a priority number. The location TMREFR + N - 1 contains in the address the location of the routine in the priority table, TMPRIO.

b) ENTRY—the location of the first instruction of the routine, stored in address position of the routine's location in TMPRIO.

c) P—the priority number of a routine which is its relative location (TMPRIO + P - 1) in the priority table, TMPRIO. The highest priority routine (P = 1) would, therefore, have the symbolic location TMPRIO.

d) QNAME—the name of the input queue block for each routine (formed by replacing NAME in the definition by the priority number. For example, Q1 is the name of the queue block for the highest priority routine and QP is the name of the queue block for the Pth priority routine).

e) QLENG—the queue length, i.e., the number of input entries to a routine that may be stacked and held in waiting until the routine is available. While every routine has a QNAME and a QLENG, for many routines the queue length is zero which means that the queue block does not physically exist.

To use the REFR macro, the following locations must be defined: TMPRIO, TMREFR: TMFRPR, TMSAVE, TMQKY2, and TMPANL by a BSS of the appropriate length.

2.16.2 Output Requirements

The following tables are generated as output from REFR:

a) TMREFR—the Monitor reference table, N cells, in order by routine number containing in the address the location of the routine in TMPRIO.

b) TMPRIO—the Monitor priority table, N cells, in order by priority number containing in the address the location of the first instruction (the entry) of the routine. The first ten bits in each location of TMPRIO contain the priority indicator bits.

c) TMFRPR—the Monitor priority reference table, N cells, in order by priority number containing in the address the routine number corresponding to the particular priority routine.

d) TMSAVE—the Monitor save table, N cells, in order by routine number containing in the address the location of the block in TMPANL which is used to store the machine condition when a program trap interrupts this routine. The machine condition is restored from the TMSAVE table before the routine may be reentered.

e) TMQKEY—the Monitor queue table, N cells, in order by routine number containing in the address the location of the queue block for this routine. A routine with a queue length of zero will have an entry location in TMQKEY equal to the next location after the last non-zero queue block previously generated. (The decrement of the cells in TMQKEY are used to keep a running count of the current entries in the routine's queue block).

f) TMQKY2—the Monitor queue length table, N cells, in order by routine number containing in the decrement the 2's complement of the routine's queue block length.

g) Queue blocks are generated as specified by the input data.

2.16.3 Method

The REFR macro is composed entirely of pseudo-instructions* and is executed for each ordinary processor, or routine, in the Mercury Program System.

The pseudo-instruction QNAME BSS QLENG assigns the symbolic location QNAME to the current value of the location counter and the location counter is incremented by the number of cells specified by QLENG.

The pseudo-instruction ORG TMREFR + N - 1 sets the location counter to the value specified by TMREFR + N - 1, and the following pseudo-instruction PZE TMPRIO + P - 1 defines the contents of the location thus specified. Six pairs of ORG and PZE pseudo-instructions produce this result (see Figure 2-16):

Symbolic Location		Contents
TMREFR + N - 1	PZE	TMPRIO + P - 1
TMPRIO + P - 1	PZE	ENTRY
TMFRPR + P - 1	PZE	N
TMSAVE + N - 1	PZE	TMPANL + 6*N - 11
TMQKEY + N - 1	PZE	QNAME
TMQKY2 + N - 1	PZE	0, 0, -QLENG

By incorporating both a reference and a priority table the Mercury System Program allows for maximum flexibility. A rigid reference table provides a permanent location for any program in the system, while the priority table may change as experience dictates the need for change.

2.16.4 Usage

REFR does not actually appear in the Mercury tracking program and has no timing considerations. After compilation the REFR macro exists only in effect.

*A discussion of the pseudo-instructions used in the Mercury Program System is contained in Program System Standards, MC 102.

TMPRIO (PRIORITY ORDER)

	S	1	2	3	4	17	21	35
TMPRIO + P-1	A	B	C	D	. . .K		Entry to Routine P	

TMFRPR (PRIORITY ORDER)

	S	1	17	21	35
TMFRPR + P-1				Routine Number N	

TMREFR (ROUTINE ORDER)

	S	1	17	21	35
TMREFR + N-1				Routine Priority, P	

TMSAVE (ROUTINE ORDER)

	S	1	17	21	35
TMSAVE + N-1				Save Location in TMPANL	

TMQKEY (ROUTINE ORDER)

	S	1	17	21	35
TMQKEY + N-1		Current entries in queue		Location of Queue Table	

TMQKY2 (ROUTINE ORDER)

	S	1	17	21	35
TMQKY2 + N-1		Length of Queue table			

P = ROUTINE PRIORITY, N = ROUTINE NUMBER

FIGURE 2-16. BASIC MONITOR TABLES

TABLE 2-1. MONITOR TABLES

Monitor Priority Reference Table

Location		21 - 35
TMFRPR		MNHSIN
+ 1		MNTMIN
+ 2		MNHSBM
+ 3		MNSTLT
+ 4		MNTTOT
+ 5		MNSENS
+ 6		MNMINS
+ 7		MNMESS
+ 8		MNMSCK
+ 9		MNSCRD
+10		MNDSPL
+11		MNSECO
+12		MNTIPT
+13		MNTOUT
+14		MNCDRD
+15		MNECOE
+16		MNCOMP
+17		MNSARC
+18		MNSMOO
+19		MNEBAD
+20		MNEDIT
+21		
+22		
+23		
+24		
+25		
+26		
+27		
+28		MNTTIN
+29		MNINTL

Monitor Priority Table

Location	S,1,2 . . . 9	21 - 35
TMPRIO	A,B,C . . . K	MPHSIN
+ 1		MYTMIN
+ 2		MYHSBM
+ 3		MYSTLT
+ 4		MYTTOT
+ 5		MYSENS
+ 6		MYMINS
+ 7		MYMESS
+ 8		MYMSCK
+ 9		MYSCRD
+10		MPDSPL
+11		MYSECO
+12		MPTIPT
+13		MPTOUT
+14		MYCDRD
+15		MYECOE
+16		MPCOMP
+17		MPSARC
+18		MPSMOO
+19		MYEBAD
+20		MPEDIT
+21		
+22		
+23		
+24		
+25		
+26		
+27		
+28		MPTTIN
+29		MYINTL

TABLE 2-2. MONITOR TABLES

Monitor Queue Table

Location	3 – 17			21 – 35
TMQKEY		*		Y** (Q6)
+ 1				Y (Q17)
+ 2				Y (Q13)
+ 3				Y (Q5)
+ 4				Y (Q19)
+ 5				Y (Q18)
+ 6				Y+10 (Q20)
+ 7				Y+10 (Q1)
+ 7				Y+110 (Q3)
+ 9				Y+110 (Q16)
+10				Y+110 (Q22)
+11				Y+110 (Q10)
+12				Y+110 (Q21)
+13				Y+110 (Q14)
+14				Y+110 (Q2)
+15				Y+110 (Q15)
+16				Y+110 (Q12)
+17				Y+110 (Q11)
+18				Y+110 (Q8)
+19				Y+110 (Q9)
+20				Y+110 (Q7)
+21				
+22				
+23				
+24				
+25				
+26				
+27				
+28				Y+110 (Q3)
+29				Y+110 (Q30)

Monitor Queue Length Table

Location	3 – 17	
TMQKY2	0	
+ 1	0	
+ 2	0	
+ 3	0	
+ 4	0	
+ 5	–10	
+ 6	0	
+ 7	–100	
+ 8	0	
+ 9	0	
+10	0	
+11	0	
+12	0	
+13	0	
+14	0	
+15	0	
+16	0	
+17	0	
+18	0	
+19	0	
+20	0	
+21		
+22		
+23		
+24		
+25		
+26		
+27		
+28	0	
+29	0	

*The decrement (3 – 17) of each location in TMQKEY contains the current number of entries, up to the maximum allowable, waiting as input to the routine. As REFR generates TMQKEY, the decrement is zero for all locations. For those routines which have a maximum queue length of zero, the decrement of TMQKEY never changes.

**When the macro constructs the TMQKEY table, every QNAME (Q1, Q2, Q3 . . . QN) generates a symbolic location. If, however, the queue specified by QNAME does not physically exist, i.e., has zero length, the absolute location will not be unique. In the development of the queue tables, the location counter is incremented only when a real (non-zero length) queue is defined in the macro parameters. In the tables above only Q1 and Q18 physically exist. Q6, Q17, Q13, Q5 and Q19 are assigned the same location in core, but the location counter is not stepped until after Q18 has been assigned the same location (shown as Y in the table), when the compiler finds a non-zero value in the variable field of the BSS pseudo-instruction. The queues of zero length are never referred to in the program.

TABLE 2-3. MONITOR TABLES

Referenced Program	Monitor Reference Table			Monitor Save Table		
Name	Location		21 – 35	Location		21 – 35
1 MNMESS	TMREFR		TMPRIO+ 7	TMSAVE		TMPANL
2 MNCDRD	+ 1		TMPRIO+14	+ 1		TMPANL+ 6
3 MNMSCK	+ 2		TMPRIO+ 8	+ 2		TMPANL+ 12
4 MNTTIN	+ 3		TMPRIO+28	+ 3		TMPANL+ 18
5 MNSTLT	+ 4		TMPRIO+ 3	+ 4		TMPANL+ 24
6 MNHSIN	+ 5		TMPRIO	+ 5		TMPANL+ 30
7 MNEDIT	+ 6		TMPRIO+20	+ 6		TMPANL+ 36
8 MNSMOO	+ 7		TMPRIO+18	+ 7		TMPANL+ 42
9 MNEBAD	+ 8		TMPRIO+19	+ 8		TMPANL+ 48
10 MNSECO	+ 9		TMPRIO+11	+ 9		TMPANL+ 54
11 MNSARC	+10		TMPRIO+17	+10		TMPANL+ 60
12 MNCOMP	+11		TMPRIO+16	+11		TMPANL+ 66
13 MNHSBM	+12		TMPRIO+ 2	+12		TMPANL+ 72
14 MNTOUT	+13		TMPRIO+13	+13		TMPANL+ 78
15 MNECOE	+14		TMPRIO+15	+14		TMPANL+ 84
16 MNSCRD	+15		TMPRIO+ 9	+15		TMPANL+ 90
17 MNTMIN	+16		TMPRIO+ 1	+16		TMPANL+ 96
18 MNSENS	+17		TMPRIO+ 5	+17		TMPANL+102
19 MNTTOT	+18		TMPRIO+ 4	+18		TMPANL+108
20 MNMINS	+19		TMPRIO+ 6	+19		TMPANL+114
21 MNTIPT	+20		TMPRIO+12	+20		TMPANL+120
22 MNDSPL	+21		TMPRIO+10	+21		TMPANL+126
23						
24						
25						
26						
27						
28						
29						
30 MNINTL	+29		TMPRIO+29	+29		TMPANL+174

Section 3.
DCC INPUT CONTROL

Three types of input are received into the Mercury Program System via the DCC: high-speed radar data, low-speed telemetry, teletype and paper tape data, and timing controls. The low-speed teletype input has no functional requirement in the Bermuda System and at present is not used. All of the other DCC input data, except low-speed telemetry, enter the IBM 709 computer via the DCC as received and are accepted synchronously with data processing. The telemetry information is entered via the DCC on request by the computer program.

All input utilizes the internal trapping feature of the IBM 709, i.e., a program interrupt occurs at the completion of the data transmission. All interrupts produced by the DCC are interpreted by the DCC Main Controller Program, M0DCCC. This controller program saves the condition of the interrupted program and gives control to the Monitor trap processor appropriate for the input information.

The Monitor programs which control DCC input data are described in detail in this section. The DCC input to the Bermuda Mercury Program System is illustrated in Figure 3-1.

3.1 HIGH SPEED INPUT: RADAR DATA

The primary input data at Bermuda is that from the AN/FPS-16 and the Verlort radar units. Every tenth of a second the time generator reads the radar data out of each radar's data buffer register, supplies a time tag for the message and transmits the data to the IBM 709 computer. The time generator operates independently from the radars, and if the radar is not in operation, each bit in the buffer is zero. An on or off-track bit in the radar message is set manually at the radar unit.

Each radar transmits over one subline of the duplexed high-speed input line via DCC subchannel 1. Each subline has a separate address register and stores its data in a separate block in the IBM 709 computer core storage. Date arrives in 8-bit bytes to fill bits 28-35 of a cell in core storage. After each byte is received, the address register is stepped by one. Each complete frame contains 12 words of eight bits which comprise one radar point—range, azimuth, elevation, time, and on or off-track indication, and several special bits not used in data computations. A complete radar message from the AN/FPS-16 and the Verlort radars is illustrated in Figure 3-2.

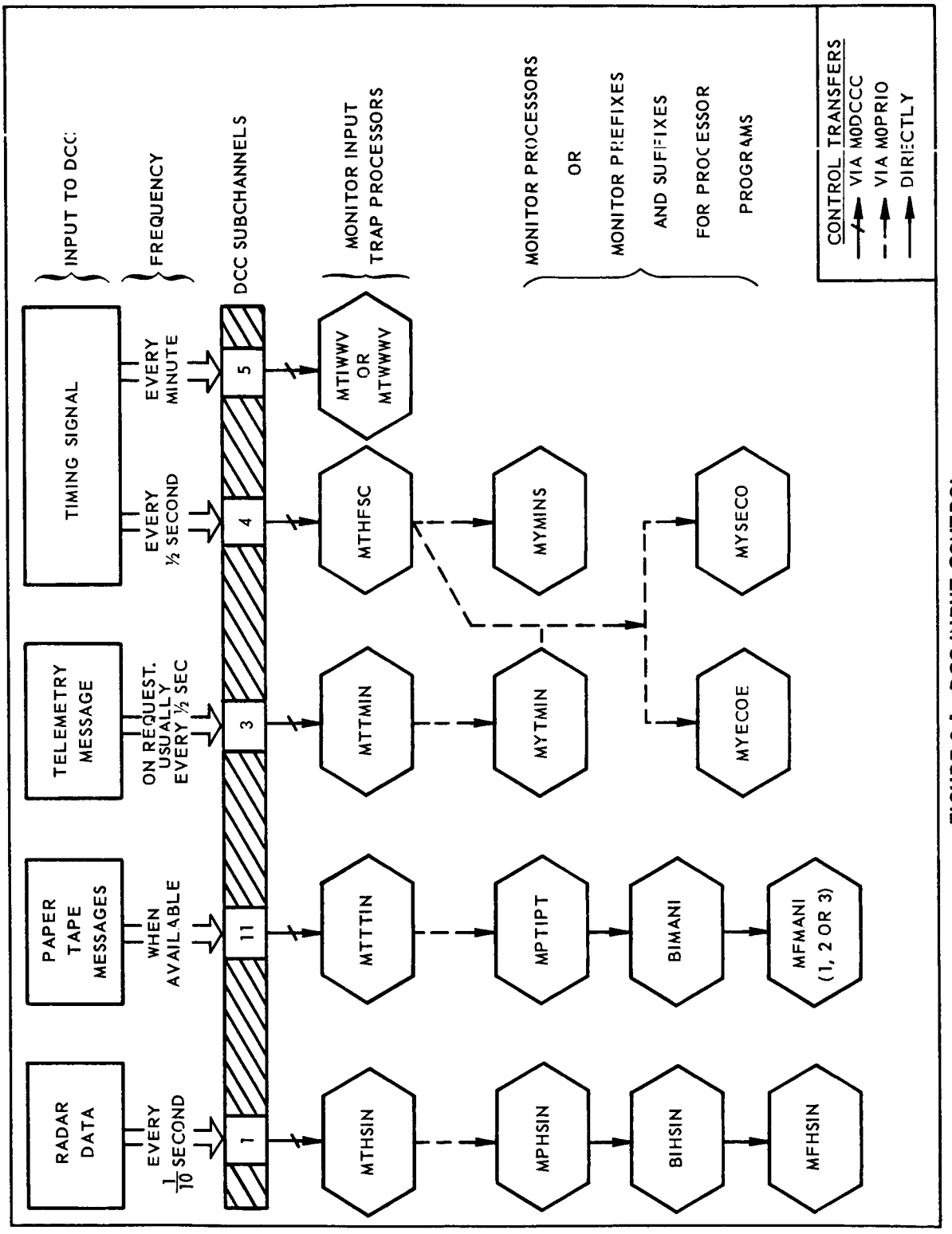


FIGURE 3-1. DCC INPUT CONTROL

AN/FPS - 16 Data

	28							35
TMHSIF	R_{17}	R_{18}	R_{19}	R_{20}	0	0	0	0
	R_9	R_{10}	R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	R_{16}
	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
	A_{17}	0	0	0	0	0	0	0
	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}
	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
	E_{17}	0	0	O.T.	0	0	0	0
	E_9	E_{10}	E_{11}	E_{12}	E_{13}	E_{14}	E_{15}	E_{16}
	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8
	T_{17}	T_{18}	T_{19}	T_{20}	T_{21}	T_{22}	T_{23}	T_{24}
	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}
+ 11	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8

Verlort Data

	28							35
TMHSIV	R_{17}	R_{18}	R_{19}	R_{20}	0	0	0	0
	R_9	R_{10}	R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	R_{16}
	0	R_2	R_3	R_4	R_5	R_6	R_7	R_8
	0	0	0	0	0	0	0	0
	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}
	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
	0	0	0	O.T.	0	0	0	0
	E_9	E_{10}	E_{11}	E_{12}	E_{13}	E_{14}	E_{15}	E_{16}
	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8
	T_{17}	T_{18}	T_{19}	T_{20}	T_{21}	T_{22}	T_{23}	T_{24}
	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}
+ 11	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8

Least significant bit of each quantity is received first. Words 1 to 12 are transferred from each receiver into Real Time Channel in sequence. The data receivers are synchronized so that the FPS-16 message and the Verlort message are in synchronism. The blank bits are either zero or contain indications not used by the computer programs.

Range, azimuth, and elevation are digital values. The units of each quantity are such that the quantity represented by 1 bit is:

	FPS-16	Verlort
Range	1 yd.	9.765625 yd.
Azimuth	0.048828125 mil.	0.09765625 mil.
Elevation	0.048828125 mil.	0.09765625 mil.

Time T_1 to T_{24} is coded as follows:

T_{24} = 0.1 sec.	T_{18} = 4.0 sec.	T_{12} = 2.0 min.	T_6 = 1.0 hour
T_{23} = 0.2 sec.	T_{17} = 8.0 sec.	T_{11} = 4.0 min.	T_5 = 2.0 hour
T_{22} = 0.4 sec.	T_{16} = 10.0 sec.	T_{10} = 8.0 min.	T_4 = 4.0 hour
T_{21} = 0.8 sec.	T_{15} = 20.0 sec.	T_9 = 10.0 min.	T_3 = 8.0 hour
T_{20} = 1.0 sec.	T_{14} = 40.0 sec.	T_8 = 20.0 min.	T_2 = 10.0 hour
T_{19} = 2.0 sec.	T_{13} = 1.0 min.	T_7 = 40.0 min.	T_1 = 20.0 hour

O.T., when 1, indicates radar on track.

FIGURE 3-2. MESSAGE FORMAT: RADAR TO IBM 709

The input of a complete frame or an end-of-message indication causes an interrupt of the computer. M0DCCC interprets the interrupt and gives control to the Monitor High Speed Input Trap Processor, MTHSIN, which examines the quantity of the input data and moves the data from the DCC input buffers into working tables in core storage. A radar data frame is processed if, and only if, exactly 12 words were received from either the AN/FPS-16 or Verlort radars.

MTHSIN prepares a code identifying the number of words received in relation to the required 12 and entry is set for the Monitor prefix for high-speed input data, MPHSIN.

MPHSIN logs all of the received radar data. If exactly 12 words were received from either radar, or both, a code is prepared describing the good radar data present and control is given directly to the high-speed input processor, BIHSIN. If neither radar had transmitted exactly 12 words, control returns to Monitor.

For each radar BIHSIN must maintain a table of up to 200 observations of range, azimuth and elevation, converted from radar message format to fixed-point format. Each table is headed by the valid "start time," the GMT of the first entry, and contains range, azimuth and elevation observations at one-tenth of a second intervals. Each observation after the initial entry is time tagged by its relative location to the first observation (the second, third and fourth locations) in the table. For off-track, missing or other erroneous data extrapolated values are substituted; however, these values are biased to incur deletion by the editing processor. When either table is filled, BIHSIN examines parameters set by the editing processor and the smoothing processor to determine which observations have been used and may be deleted. The remaining observations are moved to the top of the table and other parameters are adjusted to enable the editing processor to locate the relocated data.

BIHSIN exits to its suffix, MFHSIN, which provides on-line messages indicating the on or off-track status for either radar if this status has changed since the last entry to MFHSIN. Controls are set for entry to the editing processor, BEDITS, and MFHSIN exits to Monitor.

MTHSIN, MPHSIN and MFHSIN are described in detail in subsections 3.2, 3.3, and 3.4, respectively. BIHSIN is described in detail in MC 106, Bermuda Processing Programs.

3.2 MONITOR HIGH SPEED INPUT TRAP PROCESSOR (MTHSIN)

The Monitor High-Speed Input Trap Processor examines the quantity of input data from AN/FPS-16 and Verlort radars, constructs a mask which describes the data quantity, and moves the data from the DCC input buffer.

The flow chart for the MTHSIN processor is shown in Figure 3-3.

3.2.1 Input Requirements

Input to MTHSIN consists of:

a) TMHSFP, TMHSVE—two tables which can contain up to 24 words of AN/FPS-16 and Verlort input radar data. Each cell of the table is initially equal to minus zero. When radar data enters each cell, the sign is set positive. The format of the radar messages is illustrated in Figure 3-2.

b) MCTHSM—mask, -0, used to reset the DCC radar input tables after the data is moved.

c) BNF—mask, 000 001 011 000, which is the basic logging mask for input on DCC subchannel 1.

d) Macro Used—TRNON.

3.2.2 Output Requirements

Output from MTHSIN consists of:

a) TMHSFP, TMHSVE—the first 12 cells of both buffers are reset to minus zero. If more than twelve words are received from either radar, the data in the last 12 cells of TMHSFP is stored in the decrement of the last 12 cells in TMHSVE for logging.

b) TMHSIF—table of 12 cells containing the first 12 data words from FPS-16 radar.

c) TMHSIV—table of 12 cells containing the first 12 data words from both radars: Verlort data in the address, AN/FPS-16 data in the decrement. This table is used by MPHSIN to log the first twelve words of radar input data.

d) MCTHSN—the basic log mask from BNF with the modifications indicating data quantity from both radars is stored in MCTHSN.

e) The B indicator for BIHSIN is turned on.

3.2.3 Method

When a trap occurs on DCC subchannel 1, M0DCCC gives control to MTHSIN which examines the two input buffers, TMHSFP and TMHSVE, of the subchannel for complete points. A complete point is one radar observation of range, azimuth, and elevation and occupies 12 data words in an input buffer table (see Figure 3-4). MTHSIN places bits in a mask for logging which indicates whether TMHSFP and TMHSVE contain more than, less than, or exactly 12 data words—one complete point.

The data from the first 12 cells of TMHSFP is moved into the decrement of the 12 cells of TMHSIV, and the data from the first 12 cells of TMHSVE is moved into the address of the 12 cells of TMHSIV. Any data in the remaining 12 cells of TMHSFP is moved into the decrement of the last 12 cells of TMHSVE. If TMHSFP contains one good point, the first 12 words are moved into TMHSIF to be used as input to BIHSIN, the processing program for high-speed input data.

The first 12 cells of TMHSFP and TMHSVE are reset to minus zero. The logging mask is stored in MCTHSN for the use of MPHSIN and the B indicator for BIHSIN is turned on. Program control transfers to M0PRIO.

3.2.4 Usage

MTHSIN is entered from M0DCCC; MTHSIN exits to M0PRIO.

a) Storage Required—46 locations

b) MTHSIN uses:

1) Macro—TRNON

2) Parameters—B and MNHSIN

3) Communication cell—MCTHSN

4) Tables—TMHSFP, TMHSVE, TMHSIF and TMHSIV

5) Mask—BNF

c) Time Required

1) Less than 12 words in TMHSFP—2.57 milliseconds

2) Twelve words in TMHSFP—2.93 milliseconds

3) More than 12 words in TMHSFP—3.96 milliseconds

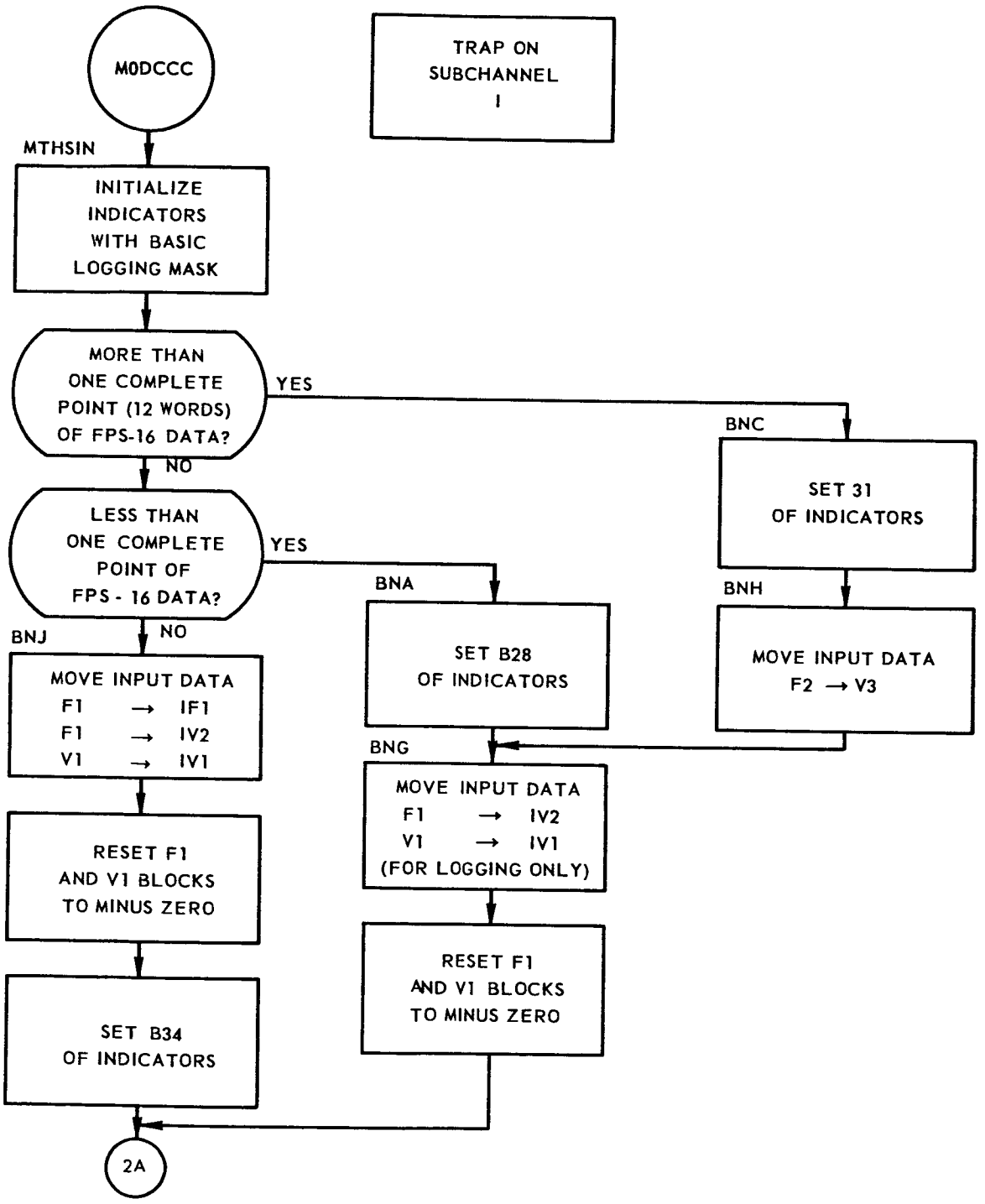


FIGURE 3-3. MTHSIN PROGRAM FLOW CHART *(Sheet 1 of 3)*

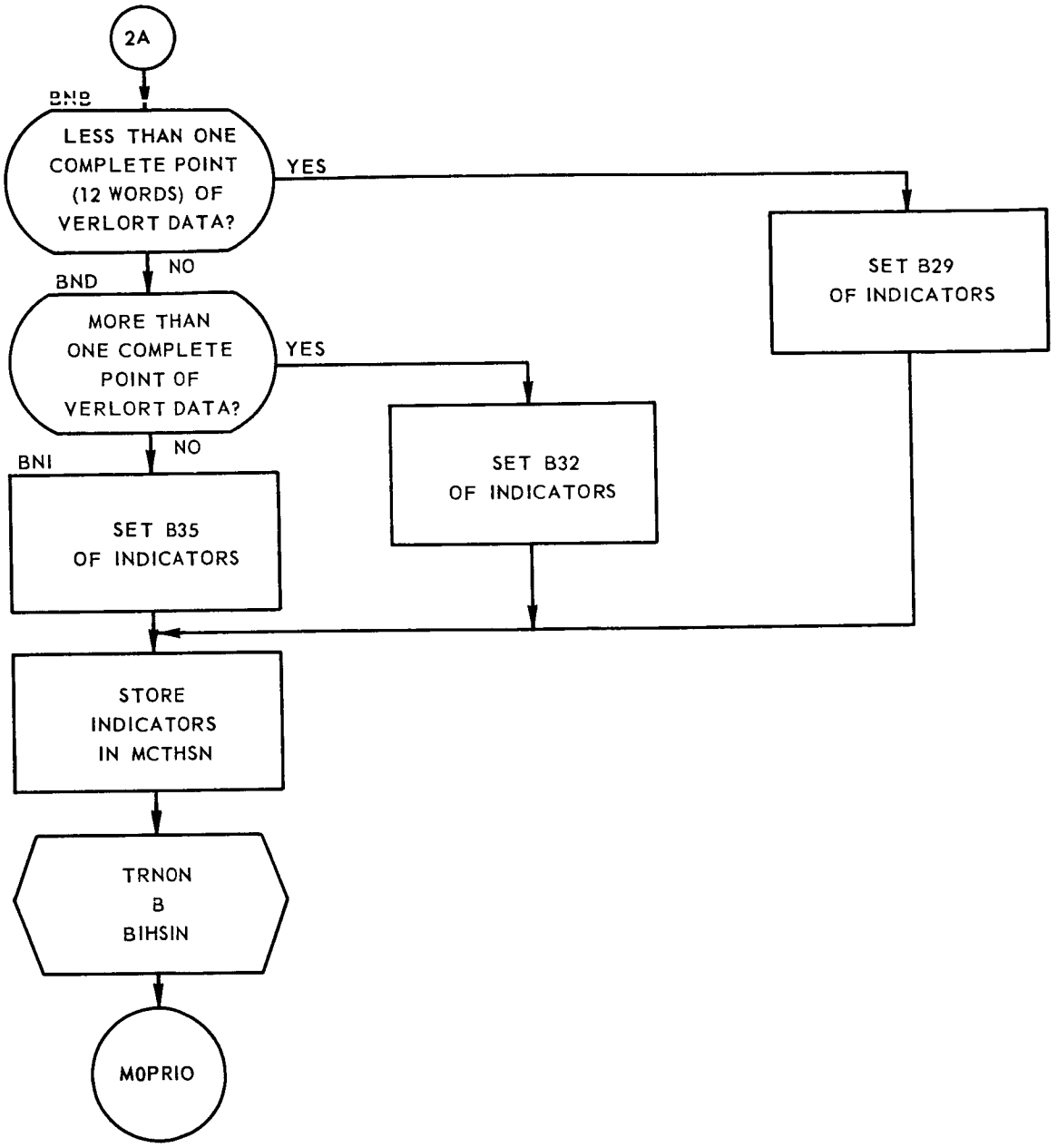
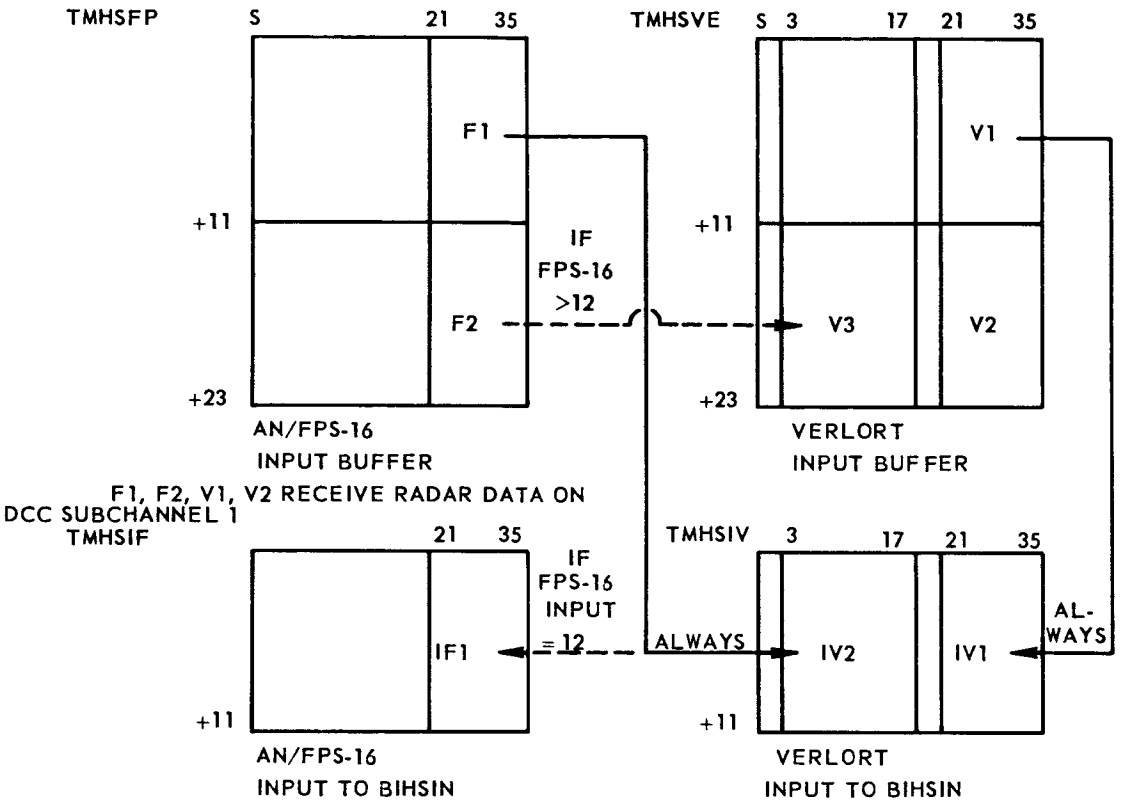
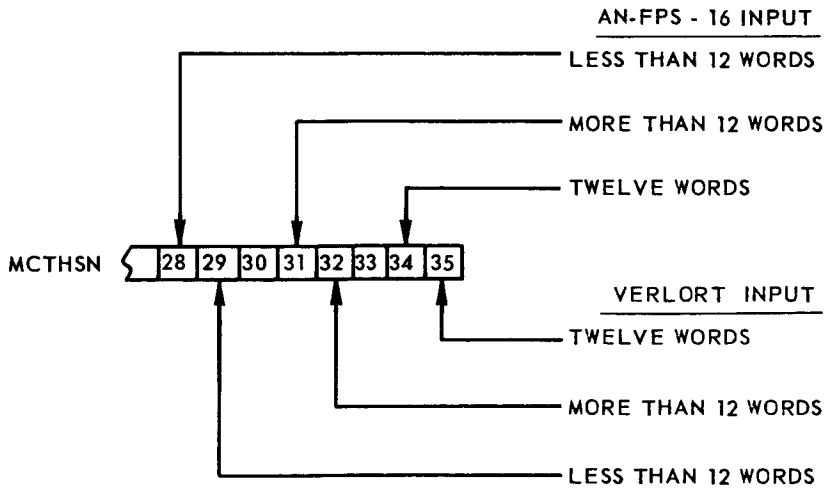


FIGURE 3-3. MTHSIN PROGRAM FLOW CHART *(Sheet 2 of 3)*



RELOCATION OF RADAR DATA BY MTHSIN



INDICATORS FOR LOGGING MASK SET BY MTHSIN

FIGURE 3-3. MTHSIN PROGRAM FLOW CHART *(Sheet 3 of 3)*

3.3 MONITOR PREFIX TO BIHSIN (MPHSIN)

MPHSIN logs the input radar data and, if sufficient data is available for processing, transfers control to BIHSIN.

The flow chart for the MPHSIN prefix is shown in Figure 3-4.

3.3.1 Input Requirements

Input to MPHSIN consists of:

a) TMHSIF—table of 12 cells set by MTHSIN containing one complete point of AN/FPS-16 radar data. If the input radar data does not contain one complete point, this table is not used.

b) TMHSIV—table of 12 cells set by MTHSIN. TMHSIV contains the first 12 words of input data from the AN/FPS-16 radar in the decrement, the first 12 words of Verlort data in the address. If the data from either radar is less than 12 words, the unused cells in TMHSIV contain minus zero.

c) TMHSVE—table of 24 cells, the last 12 cells containing radar input data for logging if more than 12 words were received from either radar.

d) TM8.3M—one-cell table containing the count of 8-1/3 millisecond intervals since the last exact half second.

e) MCHFSC—communication cell containing the current GMT, expressed in fixed-point half seconds.

f) MCTHSN—communication cell set by MTHSIN to indicate the amount of radar data received.

g) MCTHSM—mask, -0, used to reset the DCC subchannel 1 input buffers, TMHSFP and TMHSVE.

h) K00000—constant, 0.

i) Subroutine Used—MSLOGG.

j) Macros Used—QENBA, QENBZ, TRNOF and TRNON.

3.3.2 Output Requirements

Output from MPHSIN consists of:

a) TMHSFP, TMHSVE—both input buffer tables are reset to minus zero for all cells.

b) TMHSIF—The table is not changed by MPHSIN.

c) TMHSIV—If the Verlort radar data contains a complete point, zero is stored in the left half of each cell (erasing the AN/FPS-16 data contained there).

d) MCRADR—communication cell which contains a code indicating radar data acceptable for input to BIHSIN: B34 = 1, AN/FPS-16 acceptable; B35 = 1, Verlort acceptable.

e) The radar data is logged.

3.3.3 Method

MPHSIN receives control from M0PRIO, turns indicator A on and indicator B off for BIHSIN, and examines MCTHSN to determine if the amount of radar data received requires the MSLOGG subroutine be used more than one time. The standard logging time tag is constructed from the half/second and 8-1/3 millisecond counters and the radar data is logged.

If a complete point has been received from either the AN/FPS-16 or the Verlort radar, or both, indicators designating this are set in MCRADR. If the Verlort data is acceptable, the AN/FPS-16 radar data which was stored in the decrement of 12 cells of TMHSIN for logging, is replaced by zero. MPHSIN transfers control directly to the input processor, BIHSIN, with TMHSIF and/or TMHSIV as input data for BIHSIN.

If neither radar has provided a complete point, indicator A for BIHSIN is turned off and control returns to M0PRIO.

3.3.4 Usage

MPHSIN is entered from M0PRIO; MPHSIN exits to BIHSIN if the input radar data is acceptable, or to M0PRIO if the input radar data is not acceptable.

a) Storage Required—58 locations

b) MPHSIN uses:

1) Subroutine—MSLOGG

2) Macros—QENBA, QENBZ, TRNOF and TRNON

3) Parameters—A, B, and MNHSIN

4) Communication cells—MCHFSC, MCRADAR, MCTHSM and MCTHSN

5) Tables—TMHSIF, TMHSIV, TMHSFP and TMHSVE

6) Constant—K00000

c) Time Required:

1) Less than 12 data words from both radar—3.62 milliseconds

2) Twelve data word from both radars—4.17 milliseconds

3) More than 12 data word from one radar—8.08 milliseconds

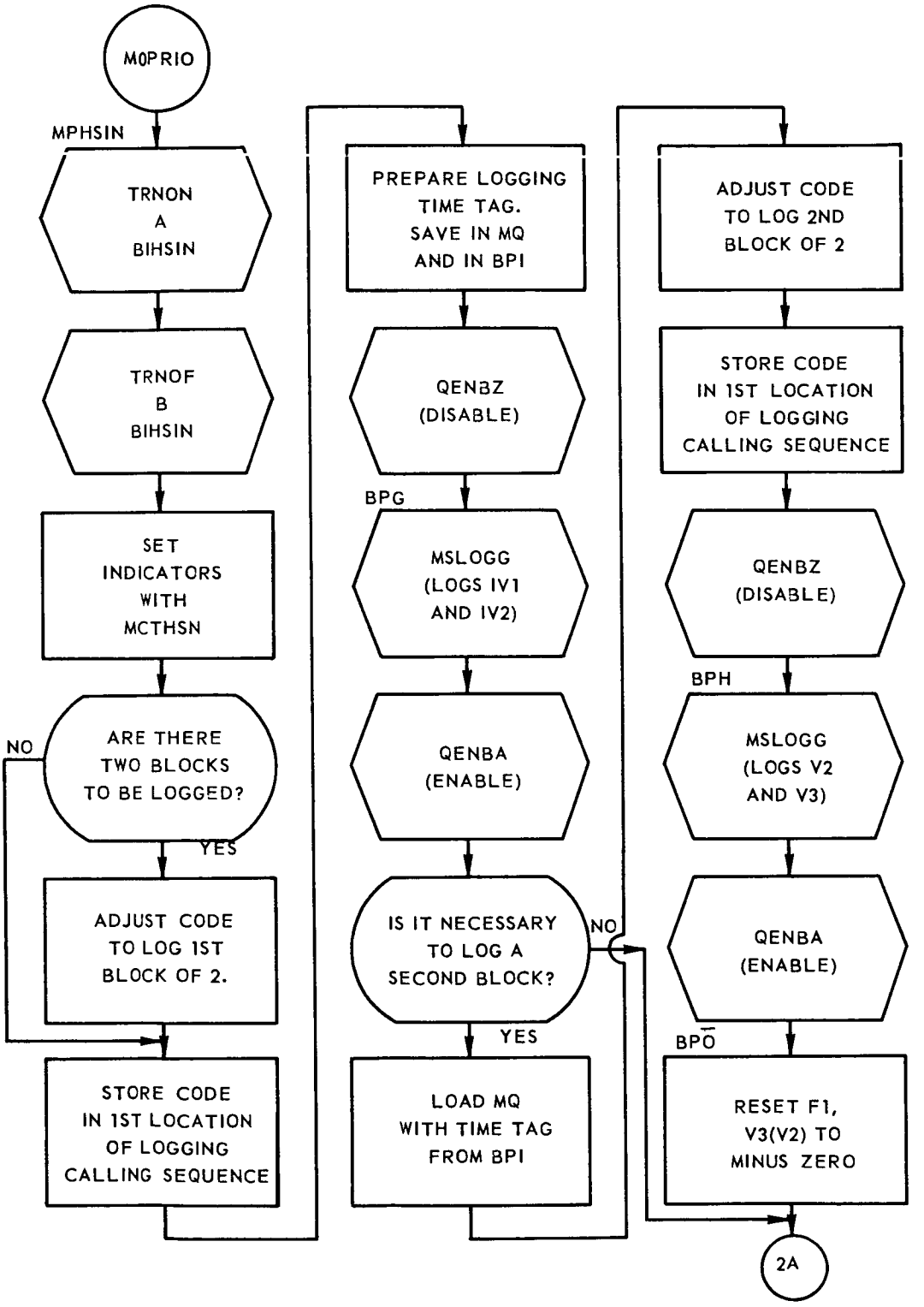


FIGURE 3-4. MPHSIN PROGRAM FLOW CHART *(Sheet 1 of 2)*

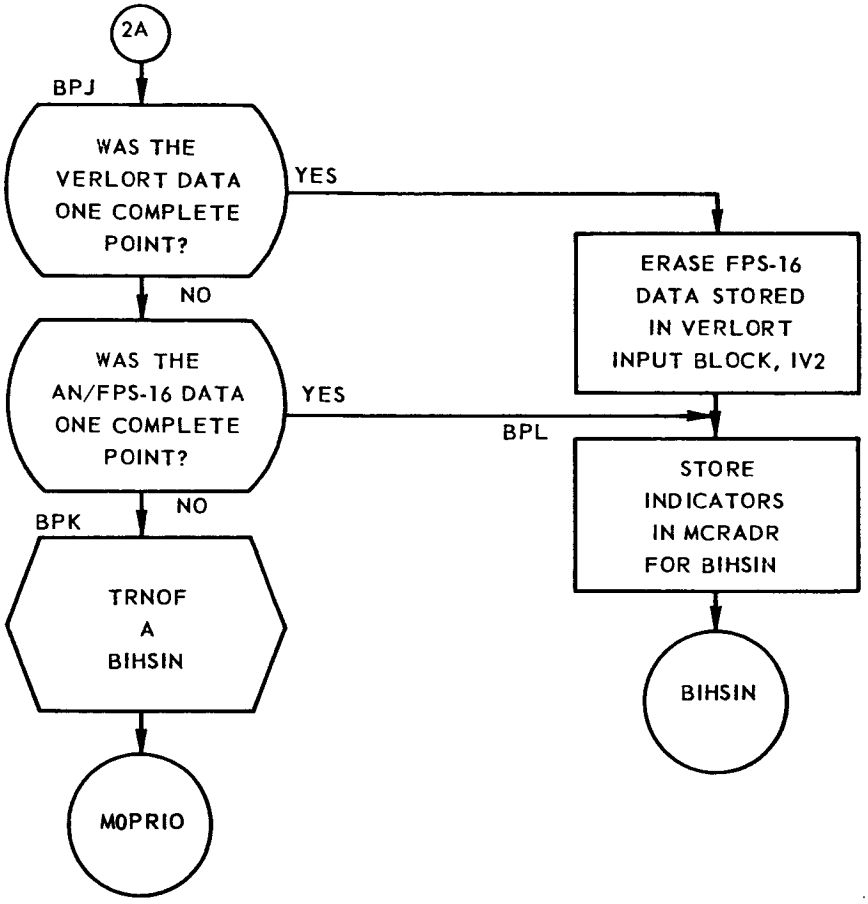


FIGURE 3-4. MPHSIN PROGRAM FLOW CHART *(Sheet 2 of 2)*

3.4 MONITOR SUFFIX TO BIHSIN (MFHSIN)

MFHSIN receives control at the completion of BIHSIN. This suffix checks whether or not there has been a change in the tracking mode—on or off-track—for either radar. A change results in an on-line message print. MFHSIN also sets controls for entry into the editing processor, BEDITS.

The flow chart for the MFHSIN suffix program is shown in Figure 3-5.

3.4.1 Input Requirements

Input to MFHSIN consists of:

a) TMONTF—one-cell table which, when zero, indicates the AN/FPS-16 radar data was sent when the radar was off track. TMONTF is set by BIHSIN.

b) TMONTV—one-cell table which, when zero, indicates the Verlort radar data was sent when the radar was off track.

c) FHST1—internal cell which, when non-zero, indicates the message FPS-16 HAS BEGUN TRACKING was the last AN/FPS-16 message provided by MFHSIN.

d) FHST2—internal cell which, when non-zero, indicates the message FPS-16 HAS CEASED TRACKING was the last AN/FPS-16 message provided by MFHSIN.

e) FHST3—internal cell which, when non-zero, indicates the message VERLORT HAS BEGUN TRACKING was the last Verlort message provided by MFFISIN.

f) FHST4—internal cell which, when non-zero, indicates the message VERLORT HAS CEASED TRACKING was the last Verlort message provided by MFHSIN.

g) Macros Used—QENBA, QENBZ, QUEUE, TRNOF and TRNON.

3.4.2 Output Requirements

MFHSIN queues the on-line message processor to print one of the four messages described above whenever there has been a change in the tracking mode of either radar since the last entry into MFHSIN. The B indicator for BEDITS is turned on and the A indicator for BIHSIN is turned off.

3.4.3 Method

MFHSIN receives control at the completion of BIHSIN and examines two cells, TMONTF and TMONTV, set by BIHSIN to indicate whether or not the

radars are on-track. When MFHSIN determines a change in either radars tracking mode, an on-line print specifies the tracking mode corresponding to the radar data currently being processed.

MFHSIN turns on the B indicator for BEDITS, turns off the A indicator for BIHSIN, and exits to M0PRIO.

3.4.4 Usage

MFHSIN is entered from BIHSIN: MFHSIN exits to M0PRIO.

a) Storage Required—49 locations.

b) MFHSIN Uses:

1) Macros—QENBA, QENBA, QUEUE, TRNOF and TRNON.

2) Parameters—A, B, MNEDIT (BEDITS), MNHSIN (BIHSIN) and MNMESS.

3) Tables—TMONTF and TMONTV.

c) Time Required:

1) Minimum (no change in tracking mode)—0.25 millisecond.

2) Maximum (both radars changed tracking mode)—1.28 milliseconds.

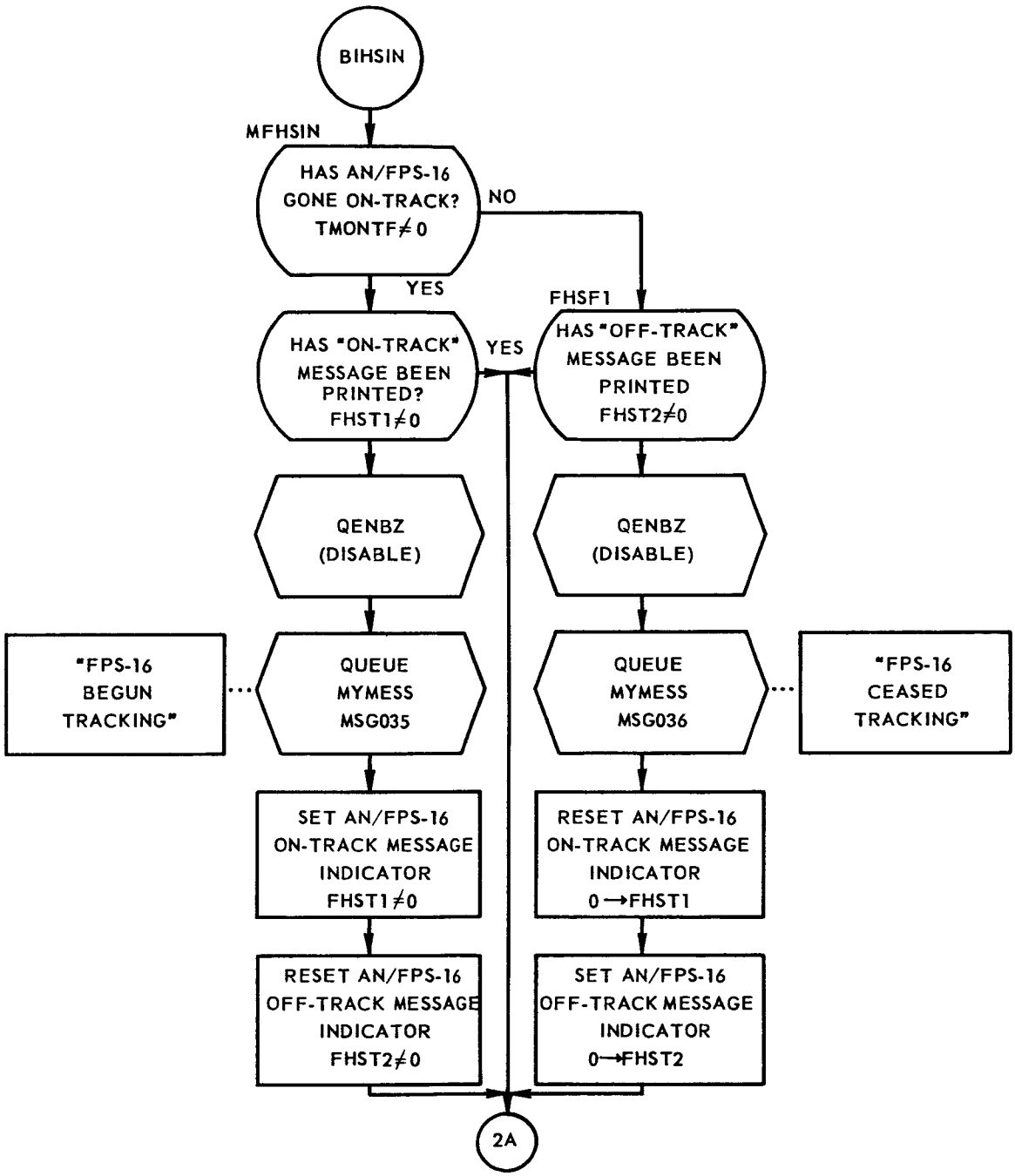


FIGURE 3-5. MFHSIN PROGRAM FLOW CHART *(Sheet 1 of 2)*

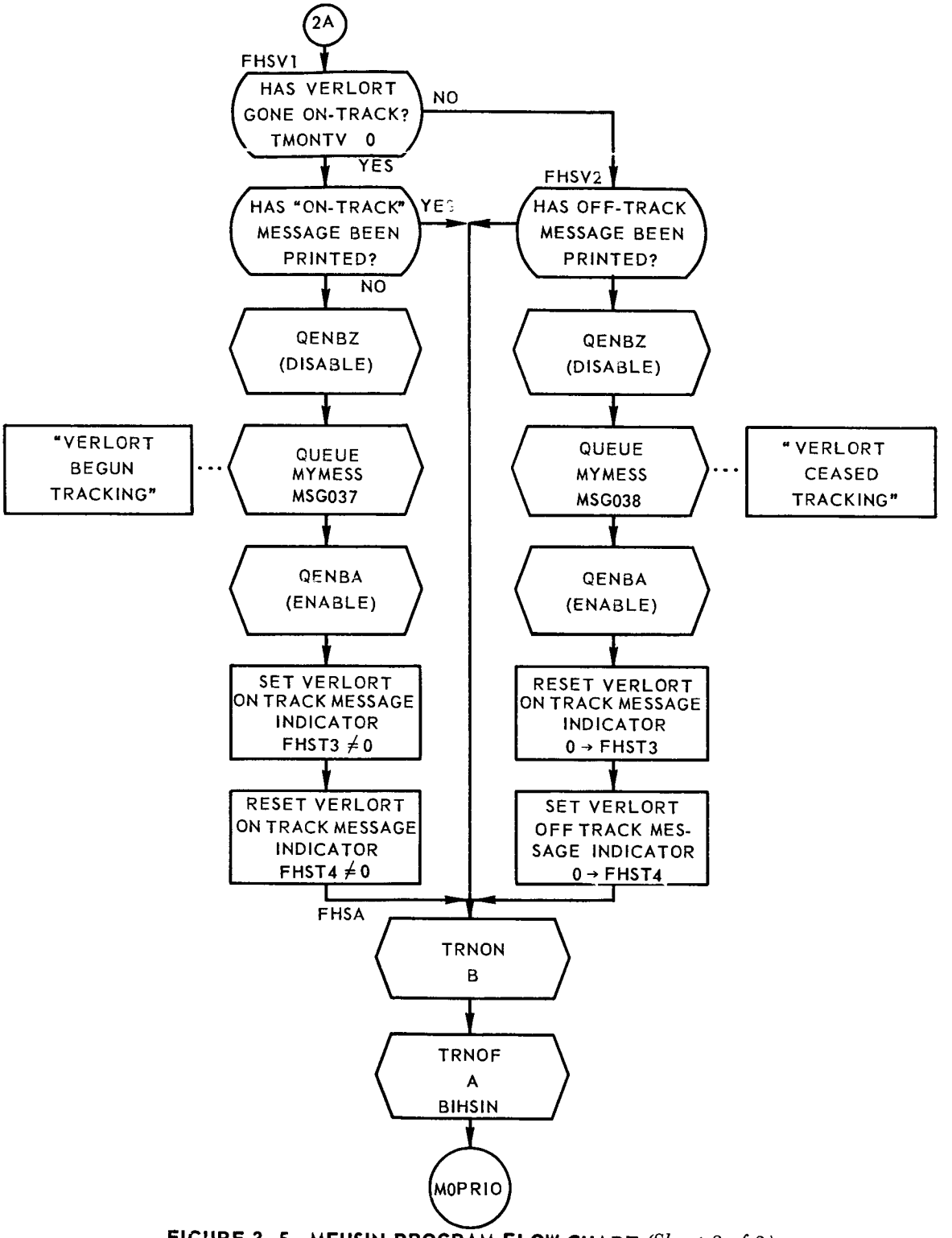


FIGURE 3-5. MFHSIN PROGRAM FLOW CHART *(Sheet 2 of 2)*

3.5 LOW-SPEED INPUT: PAPER TAPE

Paper tape is utilized to insert special numeric information into the Bermuda Program System via DCC subchannel 11. The complete paper tape message consists of 36 or more teletype format characters. Three types of paper tape messages may be processed by the manual insertion processor, BIMANI, although at present only the first of these messages, as listed below, may be accepted by the Monitor programs for paper tape input.

In the paper tape input each 5-bit character is stored in a separate cell in core storage and is right-justified, the fifth "bit" of the teletype character code occupying bit position 35 of the core memory cell. After the transmission of the sixth character into core storage, a program interrupt occurs on DCC subchannel 11. M0DCCC interprets the interrupt and gives control to the Monitor Teletype Input Trap Processor, MTTTIN. This trap processor packs the six characters into one computer word. When six such packed computer words have been received, MTTTIN informs Monitor that data sufficient for processing has been received. MTTTIN then logs the six packed words.

Monitor gives control to the prefix for paper tape input, MPTIPT, which sets the parameters for, and gives control to, BIMANI, the paper tape processor. BIMANI converts the message from the teletype code into binary information suitable for computer processing and stores the converted data into specified locations in core storage. BIMANI exits to one of its suffixes depending upon the message type, quality or quantity. If the message was incomplete, BIMANI exits to MFMANI with parameters specifying the words required to complete the message. If the message could not be processed, exit is also to MFMANI, but parameters are given to request an on-line print stating that the paper tape message be re-entered. For a successfully completed message, BIMANI exits to MFMAN1 (or MFMAN2 or MFMAN3, when available) and an on-line print specifies the information received.

MTTTIN, MPTIPT, and MFMANI and MFMANI are described in detail in subsections 3.6, 3.7, and 3.8, respectively. A detailed description of BIMANI is contained in MC 106, Bermuda Processing Programs.

Each type of message must strictly adhere to the format prescribed below. Messages as shown enter the computer from left to right.

3.5.1 Greenwich Mean Time of Lift-off

↑↑↑↑↑↑↑↑↑↑↑/000401/hhmmss/↓↓↓↓↓↓↓↓↓↓↓

a) A minimum of 11 figures shifts (↑) must precede the control word of the message.

b) The figures shifts are followed by a slash (/).

c) The slash must be followed by a six figure number in which the fourth number (4) indicates the classification code and the sixth number (1) indicates the item count, i.e., the number of six character words in the message.

d) The control word must be followed by a slash (/).

e) The next six characters define the tens and units, respectively, of hours, minutes and seconds for GMT of lift-off. If hours, minutes or seconds are less than ten, the first character of the pair must be a zero. The time characters must be followed by a slash.

f) A minimum of 11 letters shifts (↓) must terminate the message.

3.5.2 Capsule Clock Information

↑↑↑↑↑↑↑↑↑↑↑/000606/GMT_1/ECT_1/GMT_2/ECT_2/ECTRS/ ECTRS′/↓↓↓↓↓↓↓↓↓↓↓

a) The characters have the same relative meaning as those described above; the times have the same format as GMT of lift-off (hhmmss).

b) GMT_1—GMT of first reading of capsule clock

c) ECT_1—first reading of capsule elapsed time

d) GMT_2—GMT of last reading of capsule clock

e) ECT_2—last reading of capsule clock

f) ECTRS—present setting of retrofire clock

g) ECTRS′—previous setting of retrofire clock

3.5.3 Retro rocket Firing Time

↑↑↑↑↑↑↑↑↑↑↑/000802/00000F/TIME/↓↓↓↓↓↓↓↓↓↓↓

a) F—the number of retrorockets fired (1, 2 or 3).

b) The other characters have the same relative meaning as those described in 3.5.1). TIME has the same format as GMT of lift-off (hhmmss).

3.6 MONITOR TELETYPE INPUT TRAP PROCESSOR (MTTTIN)

When a paper tape message is read into core storage via the Data Communication Channel, a trap occurs after every six 5-bit teletype-format characters have been entered. M0DCCC interprets the trap and gives control to MTTTIN which takes data from the input subchannel buffer and packs the six characters into one computer word. On the sixth entry to MTTTIN, the six packed words (comprising 36 teletype characters) are logged and controls are set to inform M0PRIO that a complete message has been entered and may now be processed by the paper tape processor, BIMANI.

The flow chart for the MTTTIN program is shown in Figure 3-6.

3.6.1 Input Requirements

MTTTIN is entered from M0DCCC with the accumulator containing the DCC subchannel number (11) right justified in the decrement and the location of the DCC input buffer for that subchannel in the address. Other input to MTTTIN consists of:

a) L—parameter (9) designating the difference between the DCC subchannel number and the low-speed input line number for teletype format messages. Therefore, the paper tape input line number is 2.

b) TMTIPT—a six-cell table, each cell containing a 5-bit teletype-format character, right justified.

c) TMTTIN—an 18-cell table which provides the location in which the packed word of six characters is to be stored. The entries in TMTTIN are in order by low-speed line number, i.e., TMTTIN + 2 contains data relative to the paper tape messages. The entry for paper tape messages in TMTTIN is illustrated and described below:

Location	Operation	Address, Tag, Decrement
TMTTIN + 2	MZE	AAS02,1,D

1) MZE—the sign of TMTTIN is initially minus and indicates on the first entry the need to generate a log time for logging the complete message. The sign is set positive on the first entry and is reset minus on the last entry when the complete message is logged.

2) AAS02—the address of TMTTIN + 2 contains the symbolic location minus one of the last location of the block in which the packed input characters are to be stored.

3) D—the decrement of TMTTIN + 2 defines the number of packed computer words (6) to be stored in the block located by AAS02. The

number in the decrement is reduced by one for each pass through MTTTIN and is restored to 6 on the last pass. (The decrement also provides the location for storing the packed word for each pass: AAS02 - 6 for the first pass, AAS02 - 5 for the second, etc.)

d) TM8.3M—one-cell table containing the number of 8-1/3 milliseconds elapsed since the last exact half second.

e) MCHFSC—communication cell containing the GMT in fixed-point half seconds. TM8.3M and MCHFSC are used in generating the time tag for logging.

f) Subroutine used—MSLOGG.

g) Macro Used—QUEUE.

3.6.2 Output Requirements

For each pass MTTTIN packs the 6 teletype-format characters into one computer word and stores this packed word into location AAS02 - 6 + P, where P denotes the number of previous passes for this input message. The contents of TMTTIN + 2 are altered (as described above). On the sixth pass the complete message is logged and the complete message is queued as input to the paper tape processor, BIMANI.

3.6.3 Method

MTTTIN receives control from M0DCCC after a trap occurs on subchannel 11 indicating six 5-bit characters have been entered into computer core storage. The subchannel number is saved for the logging mask. Then, the subchannel number is converted to a low-speed input line number. This line number locates in the TMTTIN table the location for storing the input message. On each pass the six input characters are packed, right justified, in the location obtained from the TMTTIN table and controls are set for the next entry.

On the first pass the time-tag for logging is generated. On the last pass, the complete message is logged and the message is queued as input to the paper tape processor, BIMANI.

3.6.4 Usage

MTTTIN is entered from M0DCCC following a trap on DCC sub-channel 11; MTTTIN exits to M0PRIO.

a) Storage Required—262 locations

b) MTTTIN Uses:

1) Subroutine—MSLOGG

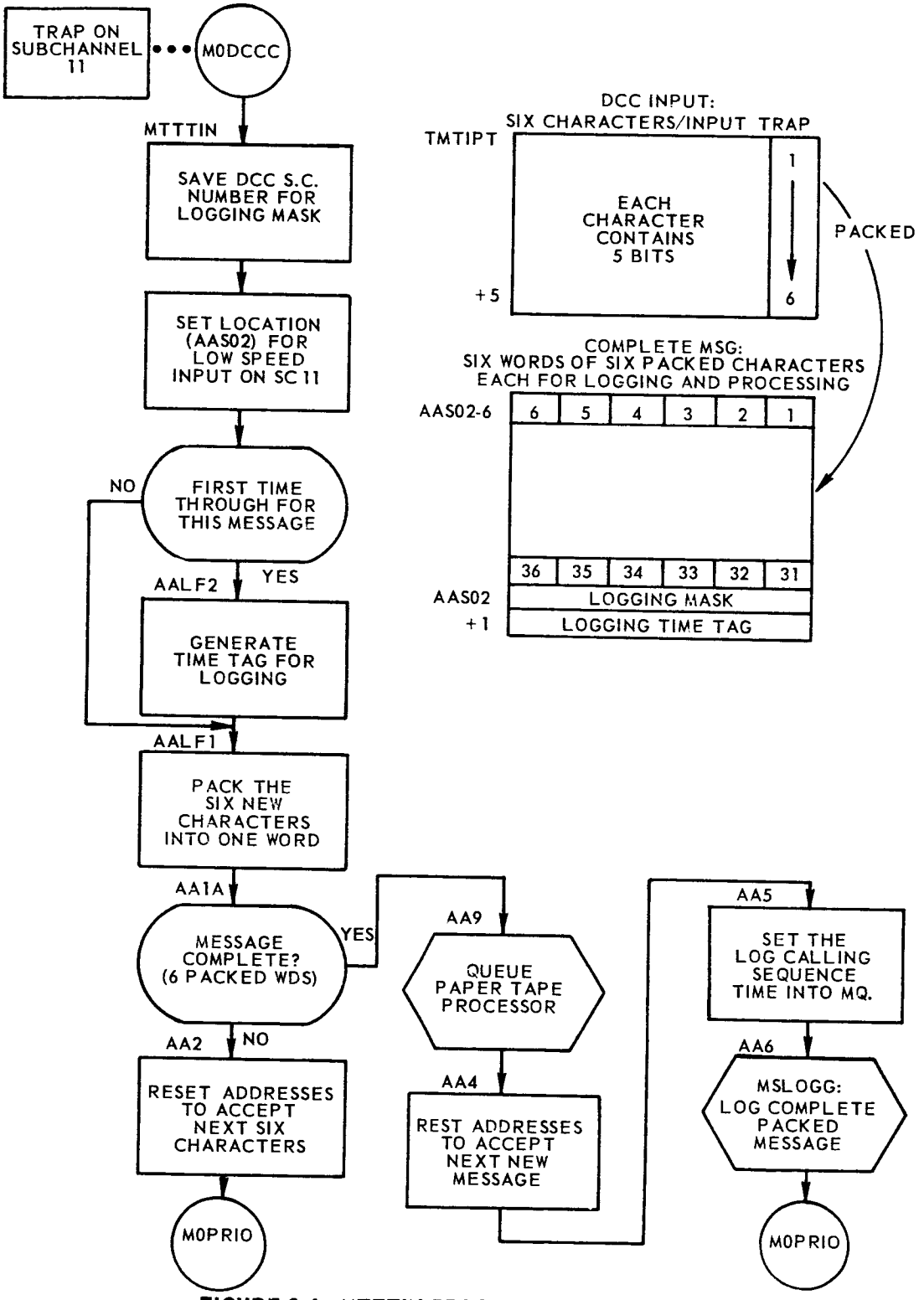


FIGURE 3-6. MTTTIN PROGRAM FLOW CHART

2) Macro—QUEUE

3) Parameters—L and MNTIPT

4) Communication cell—MCHFSC

5) Tables—TM8.3M, TMTIPT and TMTTIN

6) Internal table—AAS02

c) Time Required:

1) First pass—1.29 milliseconds

2) Second to fifth pass—1.02 milliseconds

3) Last pass—4.58 milliseconds

3.7 MONITOR PREFIX TO BIMANI (MPTIPT)

MPTIPT sets the input parameters for, and transfers control to, the paper tape processor, BIMANI.

The flow chart for the MPTIPT program is shown in Figure 3-7.

3.7.1 Input Requirements

Input to MPTIPT consists of:

a) PTX—communication cell containing in the address the location following the input data block (which contains the packed message) and in the decrement the length of input data block.

b) Macros Used—TRNOF and TRNON.

3.7.2 Output Requirements

Output from MPTIPT consists of:

a) The accumulator contains in the address the starting location of the input message lbock and in the decrement the complement of the block length.

b) The A indicator for MPTIPT is turned on and the B indicator turned off.

3.7.3 Method

MPTIPT receives control from M0PRIO and turns its A indicator on and its B indicator off. The input block length is placed in the address of the accumulator and from this is subtracted the contents of PTX. The accumulator now contains in the address the starting location of the input block and in the decrement the complement of the block length. The accumulator sign is set positive and program control transfers to BIMANI, where the message is interpreted for content.

3.7.4 Usage

MPTIPT is entered from M0PRIO; MPTIPT exits to BIMANI.

a) Storage Required—10 locations

b) MPTIPT Uses:

1) Macros—TRNOF and TRNON

2) Communication cell—PTX.

c) Time Required—0.25 millisecond.

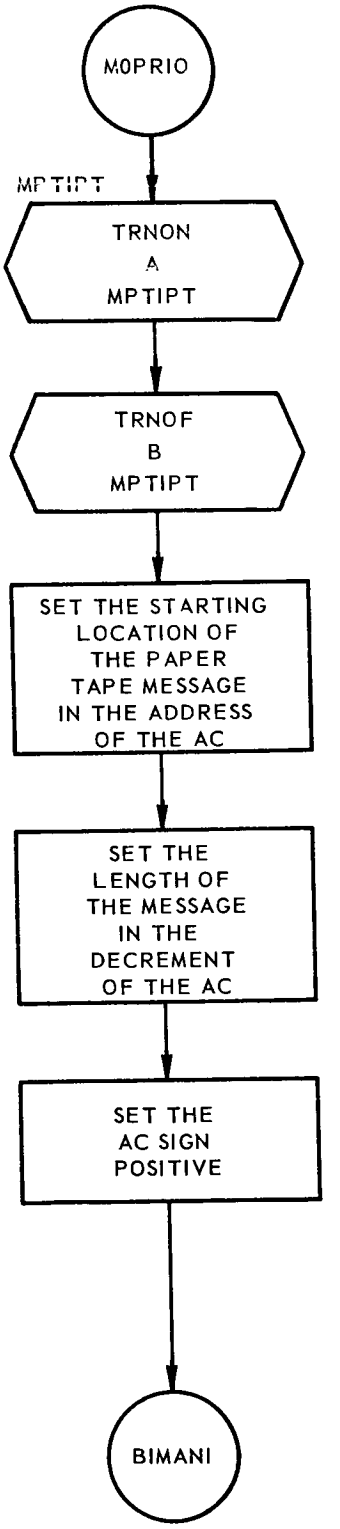


FIGURE 3-7. MPTIPT PROGRAM FLOW CHART

3.8 MONITOR SUFFIXES TO BIMANI (MFMANI, MFMAN1)

When BIMANI has successfully processed the paper tape input message, control transfers to MFMAN1 which queues an on-line print describing the input data. If BIMANI cannot completely or successfully process the message, control is given to MFMANI which sets controls dependent upon the two possibilities: message was incomplete, remainder of message must be entered; message was in error, message must be entered again.

The flow chart for the BIMANI suffixes is shown in Figure 3-8.

3.8.1 Input Requirements

Input depends upon the suffix to which BIMANI exits:

a) MFMAN1—the GMT of lift-off, converted from teletype format into fixed-point seconds, is contained in MCGTLO.

b) MFMANI—the accumulator sign designates whether the message was incomplete or in error: if the AC sign is negative the complete message must be entered again; if the AC sign is positive, the decrement of the AC contains the number of packed words necessary to complete the message.

c) Macros Used—QENBA, QENBZ, QUEUE and TRNOF.

3.8.2 Output Requirements

Output depends upon the suffix to which BIMANI exits:

a) MFMAN1—an on-line print which states the GMT of lift-off given in the paper tape message, is queued.

b) MFMANI—if the message was incomplete, the number of packed words necessary to complete the message is stored in the decrements of PTX (in MPTIPT) and TMTTIN + 2. If the message was rejected, the decrements of PTX and TMTTIN + 2 are reset to their initial value and an on-line message stating that the paper tape must be entered again, is queued.

3.8.3 Method

The method of the suffixes is illustrated in Figure 3-8.

3.8.4 Usage

MFMAN1 and MFMANI are entered from BIMANI; both suffixes exit to M0PRIO.

a) Storage Required—31 locations

b) MFMAN1 and MFMANI use:

1) Macros—QENBA, QENBZ, QUEUE and TRNOF

2) Parameters—A, MNMESS and MNTIPT

3) Communication cells—MCGTLO and PTX

4) Table—TMTTIN

c) Time Required:

1) MFMAN1—0.75 millisecond

2) MFMANI (incomplete message)—0.24 millisecond

3) MFMANI (error in message)—0.79 millisecond

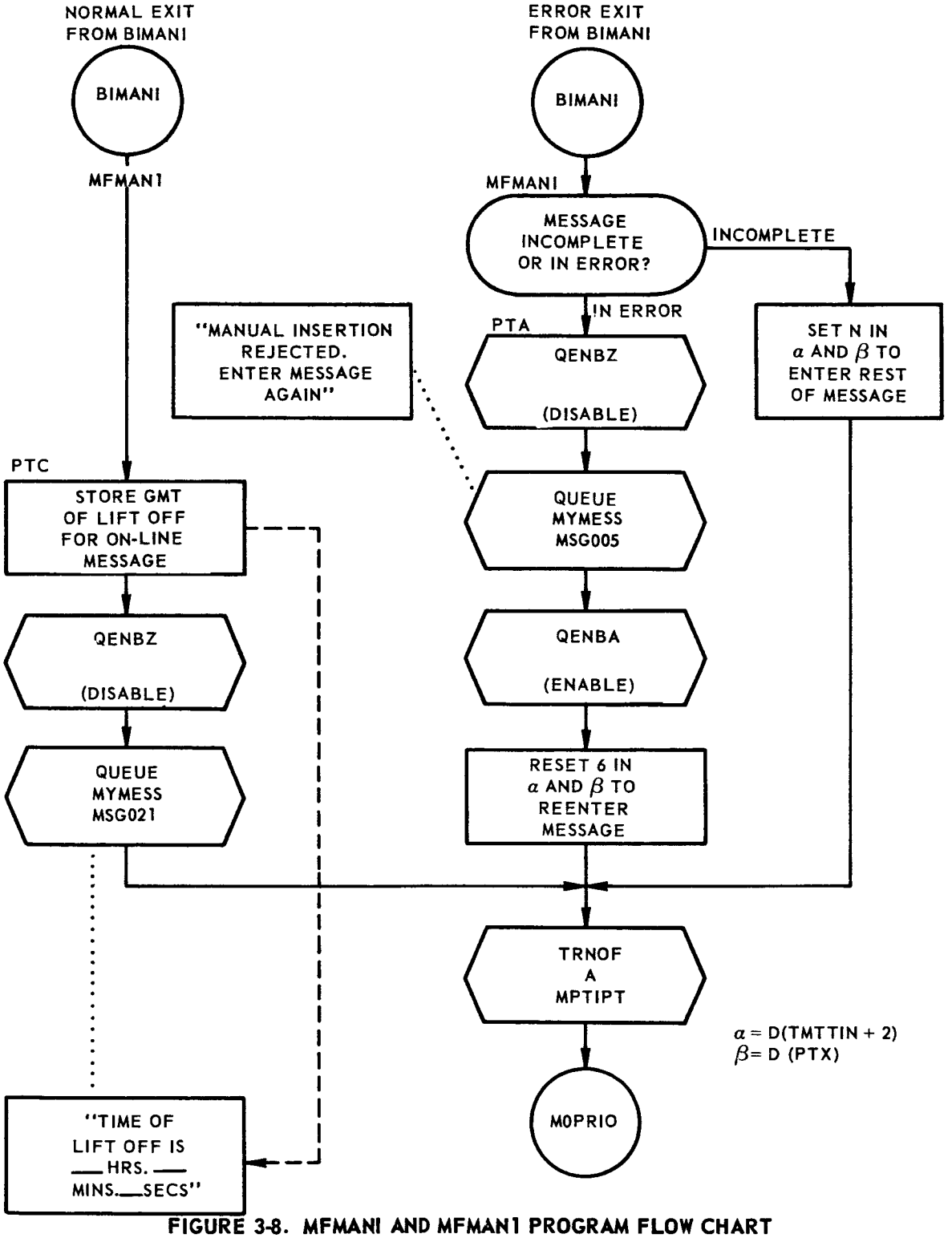


FIGURE 3-8. MFMANI AND MFMAN1 PROGRAM FLOW CHART

3.9 LOW SPEED INPUT: TELEMETRY

The telemetry data entered into core consists of eight 8-bit bytes, each byte filling bits 28-35 of a core storage cell. The message contains two capsule time readings—the elapsed time since lift-off and the retrofire mechanism setting—in a binary-coded-decimal format and eight bits which indicate the occurrence of discrete events. (At present, not all of the event bits are examined by the Bermuda Program System.) The telemetry input message is illustrated in Figure 3-9.

Telemetry messages, unlike the radar data, are not transmitted directly into computer core storage. Instead, they are received by the telemetry subsystem and selected information is placed in the Telemetry Event Buffer which is external to the IBM 709 computer system. When the computer program activates the DCC subchannel 3, the contents of external telemetry buffer begin to fill a preselected block in core storage. When the complete message is transmitted into core, a program interrupt occurs.

Every half-second, a program interrupt gives control to the Monitor Half-Second Trap Processor, MTHFSC (see subsection 3.17). This trap processor activates DCC subchannel 3 and the telemetry information begins to fill this subchannel's input buffer in core storage. If an 8-bit byte was changing during its transmission, the sign of the corresponding cell in core storage is set negative. When the complete message is transmitted, a program interrupt occurs. M0DCCC interprets the interrupt and gives control to the Monitor Telemetry Trap Processor, MTTMIN. This trap processor deactivates the subchannel to stop transmission and sets a switch in MTHFSC to prevent another telemetry message from entering until the current message is completely processed. MTTMIN records the time of message receipt and constructs an octal mask to indicate words changing during transmission and words different from corresponding words of the last message. MTTMIN sets controls for entry to the Monitor Telemetry Processor, MYTMIN.

MYTMIN logs the new message and examines the MTTMIN data mask. If words one through six had changed during transmission, subchannel 3 is activated and the external buffer is retransmitted into core again. If words seven or eight had changed, MYTMIN resets the switch in MTHFSC to allow a new telemetry input at the next half-second trap.

If no words had changed during transmission, the MTTMIN mask is examined for words changing since the previous message. If any information had changed, the message is moved from the DCC input core storage buffer into a working table. The changes are examined and control cells are set or updated depending upon the information received. If discrete bits indicate the occurrence of engine cut-off early or sustainer engine cut-off, MYTMIN sets cells in the timing tables of MTHFSC so that nine seconds after the event was reported, if no override occurs, the Monitor processors, MYECOE or MYSECO, for these two conditions are given control.

MTTMIN, MYTMIN, MYECOE and MYSECO are described in detail in the following four subsections.

	SIGN	28	29	30	31	32	33	34	35	
TMTMIN	S	TENS OF HOURS		UNITS OF HOURS				TENS OF MINUTES		CAPSULE ELAPSED TIME
+ 1	S	MINUTES (CON'T)	UNITS OF MINUTES				TENS OF SECONDS			
+ 2	S	UNITS OF SECONDS								
+ 3	S	TENS OF HOURS		UNITS OF HOURS				TENS OF MINUTES		RETROFIRE SETTING
+ 4	S	MINUTES (CON'T)	UNITS OF MINUTES				TENS OF SECONDS			
+ 5	S	UNITS OF SECONDS							ICTRC REQ'ST	
+ 6	S	CAP SEP	1 POS	1 + 2 POS	1 + 2 + 3 POS	ABORT INIT	1 RETRO	1 + 2 RETRO	1 + 2 + 3 RETRO	
+ 7	S									

DEFINITIONS:

ICTRC	ABORT ROUTINE COMMANDED
CAP SEP	CAPSULE RELEASED FROM SUSTAINER
1 POS, ETC.	ONE OF THREE POSIGRADES FIRED, ETC.
ABORT INIT	ABORT SEQUENCE INITIATED
1 RETRO, ETC.	RETRO 1 FIRED, ETC.

For any word, S = 1 denotes word was changing during transmission of word into computer core storage. The unlabled bit position are cleared.

FIGURE 3-9. MESSAGE FORMAT: EXTERNAL TELEMETRY BUFFER TO IBM 709

3.10 MONITOR TELEMETRY INPUT TRAP PROCESSOR (MTTMIN)

Once every half second subchannel 3 of the Data Communications Channel is activated and an eight word, 64-bit message is transmitted from the external telemetry input buffer via the DCC into a preselected buffer in computer core storage. When the 64-bit message has been transmitted, the computer is trapped and control is given to MTTMIN, the Monitor Telemetry Trap Processor. This program examines the message for changes occurring during transmission into core storage, compares the message with the most recently received message, generates for logging the time associated with the message, and informs the priority routine, M0PRIO, that data is available for MYTMIN, the Monitor Telemetry Processor.

The flow chart for the MTTMIN trap processor is shown in Figure 3-10.

3.10.1 Input Requirements

Input to MTTMIN consists of:

a) TMTMIN—eight-cell table which receives the telemetry input through DCC subchannel 3. Each word consists of eight bits, right justified. A negative sign indicates the word was changing when transmitted into TMTMIN.

b) TMTMEX—eight-cell table containing the last previous telemetry message. TMTMEX has the same format as TMTMIN, except that the sign of every cell is positive.

c) TM8.3M—one-cell table which contains the count of 8-1/3-millisecond intervals since last exact half second.

d) MCHFSC—communication cell containing the current GMT, expressed in half seconds.

e) MCTM02—mask, -040 000 000 000, used to activate or (when complemented) deactivate subchannel 3.

f) CHCN—sixteen-cell table of masks used to set indicator bits to describe the quality of telemetry message received. CHCN has a 1 in bit 35, CHCN + 1 has a 1 in bit 34, . . . CHCN + 7 has a 1 in bit 28. CHCN + 8 has a 1 in bit 26, CHCN + 9 has a 1 in bit 25, . . . CHCN + 15 has a 1 in bit 19. The first eight masks are used to designate words one through eight, respectively, have changed since last telemetry message. The last eight masks are used to indicate words one through eight were changing during the transmission into TMTMIN.

g) CHC1—mask, 000 003 000 000, used as the basic code mask for logging input data on subchannel 3.

h) K00012—constant, 12 (decimal).

i) K00005—constant, 5.

j) Macros Used—QPSLF and TRNON.

3.10.2 Output Requirements

Output from MTTMIN consists of:

a) MSEXTM—single-instruction subroutine in MTHFSC controlled by the telemetry processors, both trap and ordinary. This subroutine effectively acts as a switch to prevent a new telemetry message from entering the computer before the previous message is completely processed.

b) MCTM01—communication cell containing the appropriate mask indicators from the CHCN table.

c) MCTM03—communication cell containing the time the telemetry message was received in standard logging time tag format.

d) TMGMTR—one-cell table containing the time the message was received, expressed in fixed-point tenths of seconds.

e) Subchannel 3 is deactivated.

f) The B indicator for the Monitor Telemetry Processor, MYTMIN, is turned on.

3.10.3 Method

MTTMIN receives control from the DCC trap following a telemetry transmission from the external buffer into core. Subchannel 3 is immediately deactivated to prevent further transmission.

The new telemetry message is examined for two criteria: 1) change during transmission and 2) change since last telemetry message. Indicators are set in a communication cell to show these changes. The time the message was received is computed and stored in another communication cell. MYTMIN uses both cells in examining the content of the message. The time the message was received is also placed in the standard logging time tag format and this is used with the first communication cell by MYTMIN to log the telemetry message.

The B indicator for MYTMIN is turned on and control returns to M0PRIO.

3.10.4 Usage

MTTMIN is entered from M0DCCC; MTTMIN exits to M0PRIO.

a) Storage Required—54 locations

b) MTTMIN uses:

1) Subroutine—MSEXTM

2) Macros—QPSLF and TRNON

3) Parameters—B and MNTMIN

4) Communication cells—MCACTV, MCTM01, MCTM03 and MCHFSC

5) Tables—TMGMTR, TM8.3M, TMTMIN and TMTMEX

6) Constants—K00005 and K00012

7) Masks—MCTMO2, CHCN - CHCN + 15 and CHC1

c) Timing

1) Minimum—2.18 milliseconds

2) Maximum—2.95 milliseconds

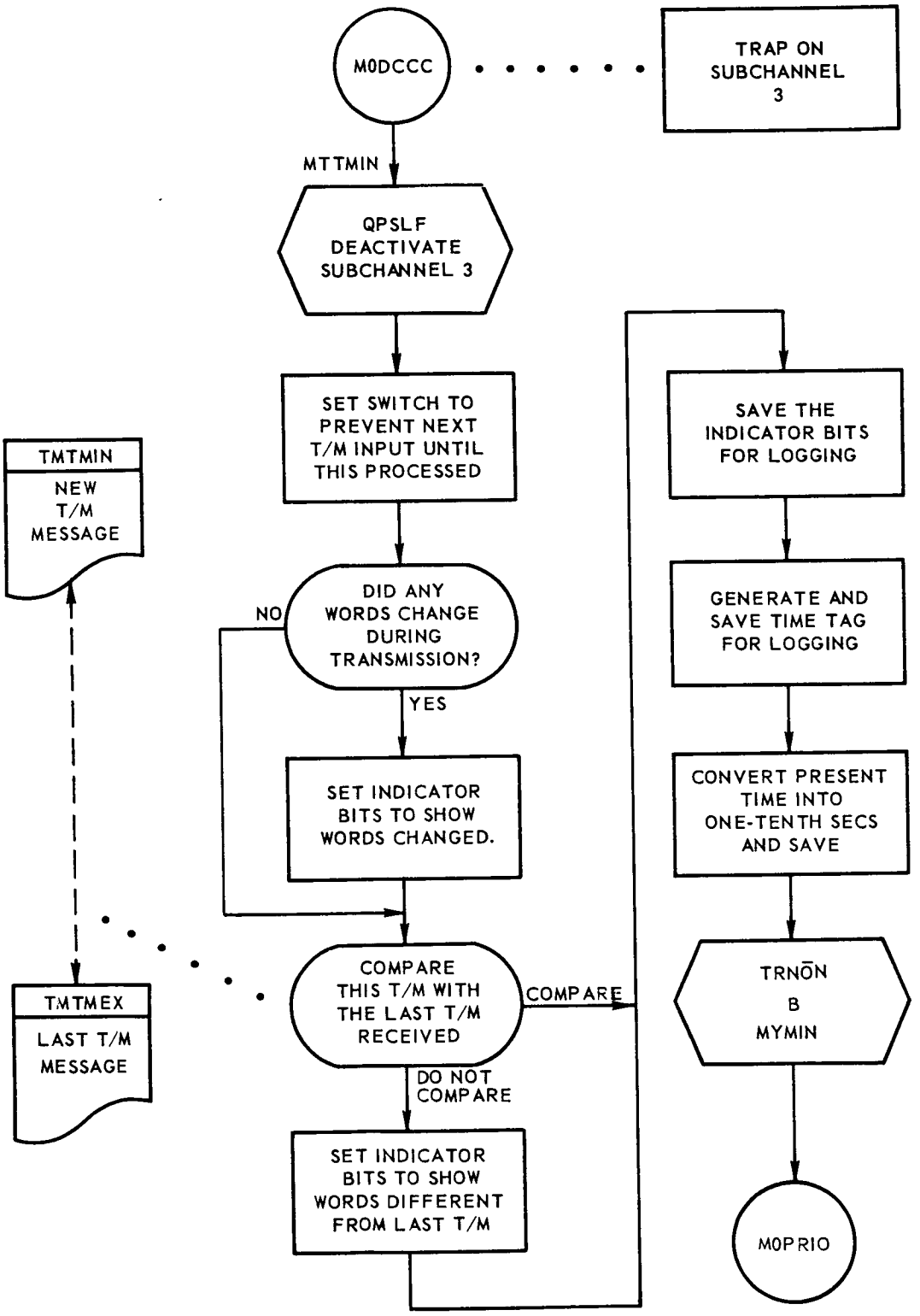


FIGURE 3-10. MTTMIN PROGRAM FLOW CHART

3.11 MONITOR TELEMETRY PROCESSOR (MYTMIN)

MYTMIN receives control from M0PRIO after a new telemetry (T/M) message has been received and given an initial examination by the trap processor, MTTMIN. MYTMIN logs the new message, examines it in detail and stores control information in various specified locations depending upon the information contained in the message.

The general flow diagram and the flow chart for the MYTMIN processor are shown in Figure 3-11 and Figure 3-12, respectively.

3.11.1 Input Requirements

Input to MYTMIN consists of:

a) TMTMIN—eight-cell table containing the new telemetry message. The telemetry message consists of eight words; each word contains eight bits and is right justified in the computer word. This table is shown in Figure 3-9.

b) TMECTR—one-cell table containing the elapsed capsule time. This cell, initially zero, is set by MYTMIN each time the telemetry input message contains a change in words 1, 2 or 3. The time in TMECTR is expressed in fixed point tenths of seconds.

c) TMGMTR—one-cell table containing the time the telemetry message was received. This time is generated by MTTMIN and is expressed in fixed point tenths of seconds.

d) TMRDLY—three-cell table containing time correction factors for the delay in reporting retro 1 has fired, retros 1 + 2 have fired and retros 1 + 2 + 3 have fired, respectively. (These cells are currently set to zero.)

e) BTFORF—three-cell table, set by MYTMIN, to indicate retrorocket firings. A 1 in BTFORF denotes that a previous telemetry message reported retro 1 has fired. A 1 in BTFORF + 1 denotes that a previous telemetry message reported retros 1 + 2 have fired. A 1 in BTFORF + 2 denotes that a previous telemetry message reported retros 1 + 2 + 3 have fired.

f) MCTM01—communication cell, set by MTTMIN, containing 1's in bit positions to indicate telemetry message words changing during transmission and words different from corresponding words of the last telemetry message.

g) MCTM02—mask used to activate or, when complemented, deactivate DCC subchannel 3.

h) MCTM03—communication cell containing the time the telemetry message was received. This cell is set by MTTMIN.

i) MCTM04—communication cell that indicates the number of times a new telemetry message has been received during the current half-second interval. This cell is reset each half second by MTHFSC.

j) MCACTV—communication cell used by the QPSLF macro to activate/deactivate subchannels of the DCC.

k) MCESPI—communication cell which, when non-zero, indicates the termination of the powered flight phase of the mission, in that nine seconds have passed since early engine cut-off (ECOE) or sustainer engine cut-off (SECO). This cell is set by MYSECO or MYECOE.

l) CUT5—internal cell containing word 6 of the last processed telemetry input message.

m) CULST—internal cell containing word 7 of the last processed telemetry input message.

n) KCH233—mask, 233 000 000 000, used in converting a fixed-point number to floating-point.

o) K00001—constant, 1.

p) K00010—constant, 10 (decimal).

q) K00600—constant, 600 (decimal).

r) K0036K—constant, 36,000 (decimal).

s) Subroutine used—MSLOGG

t) Macros used—QENBA, QENBZ, QPSLF, QUEUE, TRNOF and TRNON

3.11.2 Output Requirements

For every entry MYTMIN logs the new telemetry message. Other output from MYTMIN is conditional:

a) If words 7 or 8 of the telemetry message had changed during transmission, the switch (MSEXTM) in MTHFSC is set to allow a new telemetry input message at the next half-second trap. There is no other output.

b) If words 1 through 6 had changed during transmission and if the external telemetry buffer has been read less than three times during

this half-second interval, subchannel 3 is activated and a new telemetry message is read. If three telemetry messages have been entered this half-second interval, the switch in MTHFSC is set to allow a new telemetry input message at the next half-second trap. There is no other output.

c) If this telemetry message is identical to the previous message, there is no output unless the elapsed capsule time reading (TMECTR) is equal to zero in which case the time at which the telemetry message was received (TMGMTR) is stored as the difference between Greenwich Mean Time and the elapsed capsule time reading (in TMGMER). There is no other output.

d) If none of the three mutually exclusive conditions listed above apply, the telemetry message contains significant information. MYTMIN moves the message into table TMEXTM where the message is examined for the following conditions:

Elapsed Capsule Time Reading

Changed—the elapsed capsule time is converted into fixed-point tenths of seconds and stored in TMECTR.

Unchanged—no output if the elapsed capsule time is unchanged from zero. Otherwise, the difference between the GMT and the elapsed capsule time (TMGMTR - TMECTR) is converted to floating-point tenths of seconds and stored in TMGMER.

Capsule Separation Event Bit Changed
(Does not apply if SECO + 9 seconds has been reported.)

Event—18 is stored in the decrement of TMCYCL to begin timing SECO + 9 seconds; -0 is stored in the phase indicator cell BCCPHS; the time of the event is converted into fixed-point tenths of seconds and stored in TMTSEP and TMSEPO; the appropriate on-line message is queued.

Override—32,767 (an effective infinity) is stored in the decrement of TMCYCL to stop timing SECO + 9 seconds; TMTSEP is cleared; the appropriate on-line message is queued.

Abort Initiate Event Bit Changed
(Does not apply if SECO + 9 seconds has been reported)

Event—1 is stored in MCABRT; the time is stored in TMTABT in fixed-point tenths of seconds; 18 is stored in the decrement of TMCYCL + 1 to wait nine seconds before entering abort phase; the appropriate on-line message is queued.

Override—MCABRT and TMTABT are cleared; 32767 is stored in the decrement of TMCYCL + 1 to stop timing for abort phase; the appropriate on-line message is queued.

Retro 1 Fired Event Bit Changed

Event—1 is stored in BTFORF; the corrected time of retrofire (TMGMTR - TMRDLY) is stored in BTTORF in floating point tenths of seconds; -1 is stored in BCCPHS; the appropriate on-line message is queued.

Override—BTFORF and BTTORF are cleared; the appropriate on-line message is queued.

Retros 1 + 2 Fired Event Bit Changed

Event—1 is stored in BTFORF + 1; the corrected time of retrofire (TMGMTR - TMRDLY + 1) is stored in BTTORF + 1 in floating-point tenths of seconds; -1 is stored in BCCPHS; the appropriate on-line message is queued.

Override—BTFORF + 1 and BTTORF + 1 are cleared; the appropriate on-line message is queued.

Retros 1 + 2 + 3 Fired Event Bit Changed

Event—1 is stored in BTFORF + 2; the corrected time of retrofire (TMGMTR - TMRDLY + 2) is stored in BTTORF + 2 in floating-point tenths of seconds; -1 is stored in BCCPHS; the appropriate on-line message is queued.

Override—BTFORF + 2 and BTTORF + 2 are cleared; the appropriate on-line message is queued.

Abort Command (ICTRC) Event Bit Changed

Event—1 is stored in MCABOT; the time is stored in TMABOT in fixed-point tenths of seconds; the appropriate on-line message is queued.

Override—MCABOT and TMABOT are cleared; the appropriate on-line message is queued.

Retrofire Mechanism Time
(Does not apply if Retros 1 + 2 + 3 fired reported)

Changed—the packed, binary-coded-decimal time (from telemetry message words 4-6) is saved in BTTFRR; the time is stored in TMRETS in fixed-point tenths of seconds.

TABLE 3-1. TIME CELLS USED BY MYTMIN

Symbolic Location	Explanation	Source	Format	Units
BTTORF	Corrected GMT Retro 1 Fired	TMGMTR - TMRDLY	Floating Point	Tenths of Seconds
BTTORF + 1	Corrected GMT Retros 1 + 2 Fired	TMGMTR - TMRDLY + 1	Floating Point	Tenths of Seconds
BTTORF + 2	Corrected GMT Retros 1 + 2 + 3 Fired	TMGMTR - TMRDLY + 2	Floating Point	Tenths of Seconds
BTTFRR	GMT Setting of Retrofire Mechanism	T/M words 4,5,6	Packed BCD	Hrs - Mins - Secs
TMTABT	GMT of Abort Initiate	TMGMTR	Fixed Point	Tenths of Seconds
TMABOT	GMT of Abort Command (ICTRC)	TMGMTR	Fixed Point	Tenths of Seconds
TMECTR	Elapsed Capsule Time Reading (Interval since Lift-off)	T/M words 1,2,3	Fixed Point	Tenths of Seconds
TMGMER	Difference Between GMT and Elapsed Capsule Time	TMGMTR - TMECTR	Floating Point	Tenths of Seconds
TMGMTR	GMT T/M Input Message Received	MCHFSC and TM8.3M by MTTMIN	Fixed Point	Tenths of Seconds
TMRDLY	Delay in T/M Report of Retro 1 Fired	Given	Floating Point	Tenths of Seconds
TMRDLY + 1	Delay in T/M Report of Retros 1 + 2 Fired	Given	Floating Point	Tenths of Seconds
TMRDLY + 2	Delay in T/M Report of Retros 1 + 2 + 3 Fired	Given	Floating Point	Tenths of Seconds
TMRETS	GMT Setting of Retrofire Mechanism	T/M Words 4,5,6	Fixed Point	Tenths of Seconds
TMSEPO	GMT of Capsule Separation	TMGMTR	Fixed Point	Tenths of Seconds
TMTSEP	GMT of Capsule Separation	TMGMTR	Fixed Point	Tenths of Seconds

Unchanged—no output.

e) If changes had occurred in either the sixth or seventh words of the telemetry input message, the new sixth and/or seventh words are stored in CUT5 and CULST, respectively.

3.11.3 Method

The method of MYTMIN is illustrated in the general flow diagram, Figure 3-11, and in the flow chart, Figure 3-12.

3.11.4 Usage

MYTMIN is entered from, and exits to, M0PRIO.

a) Storage Required—223 locations.

b) MYTMIN Uses:

1) Subroutines—MSLOGG and MSEXTM.

2) Macros—QENBA, QENBZ, QPSLF, QUEUE, TRNOF and TRNON.

3) Parameters—A, B, MNMESS and MNTMIN.

4) Communication cells—BCCPHS, MCABOT, MCABRT, MCACTV, MCESPI, MCTM01, MCTM02, MCTM03 and MCTM04.

5) Internal cells—CULST and CUT5.

6) Tables—BTFORF, BTTFRR, BTTORF, TMABOT, TMTABT, TMCYCL, TMECTR, TMGMTR, TMRDLY, TMRETS, TMSEPO, TMTMEX, TMTMIN and TMTSEP.

7) Constants—K00001, K00010, K00600 and K0036K.

8) Mask—KCH233.

c) Time Required:

1) Minimum—3.0 milliseconds.

2) Maximum—12.7 milliseconds.

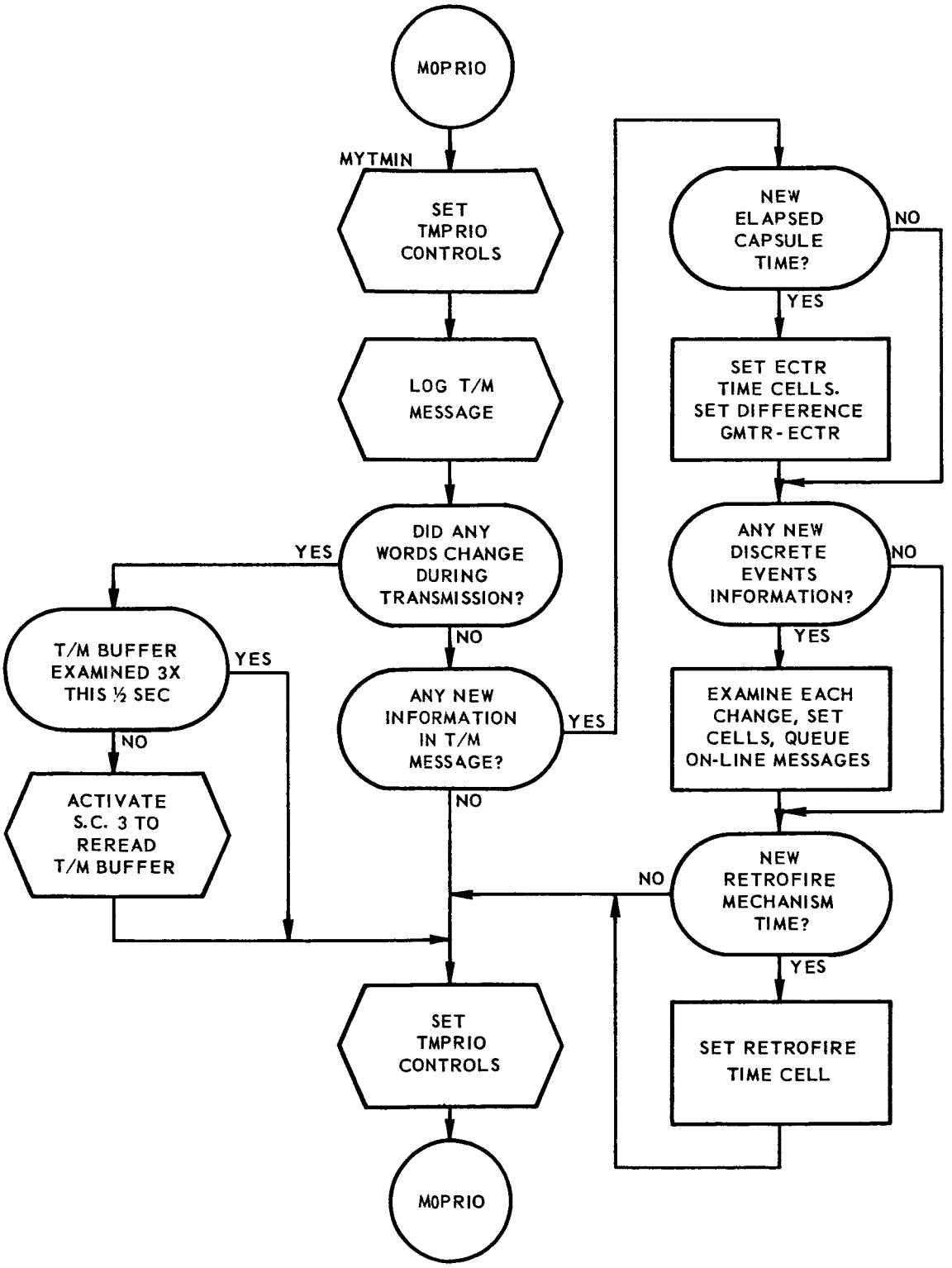


FIGURE 3-11. GENERAL FLOW DIAGRAM, MYTMIN PROCESSOR

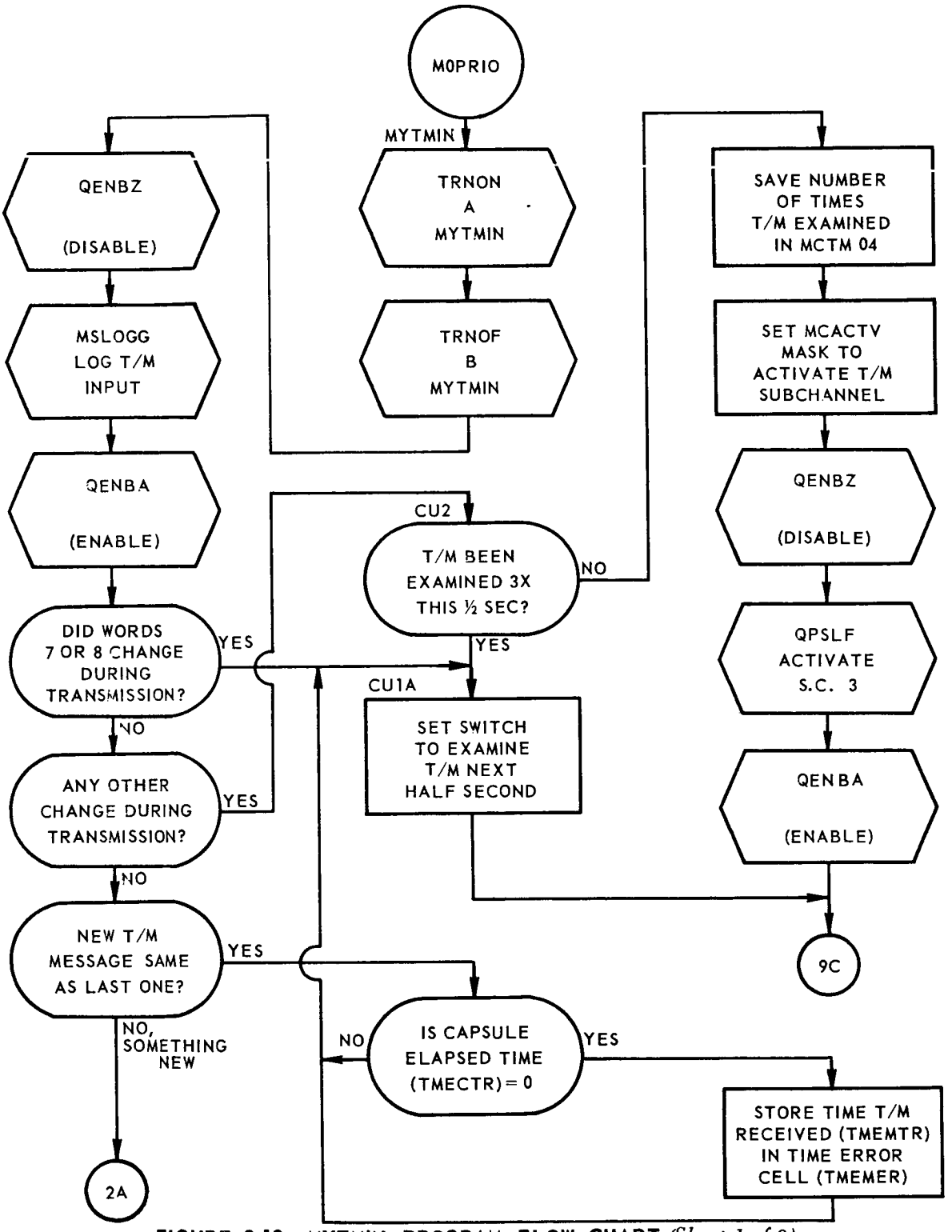


FIGURE 3-12. MYTMIN PROGRAM FLOW CHART *(Sheet 1 of 9)*

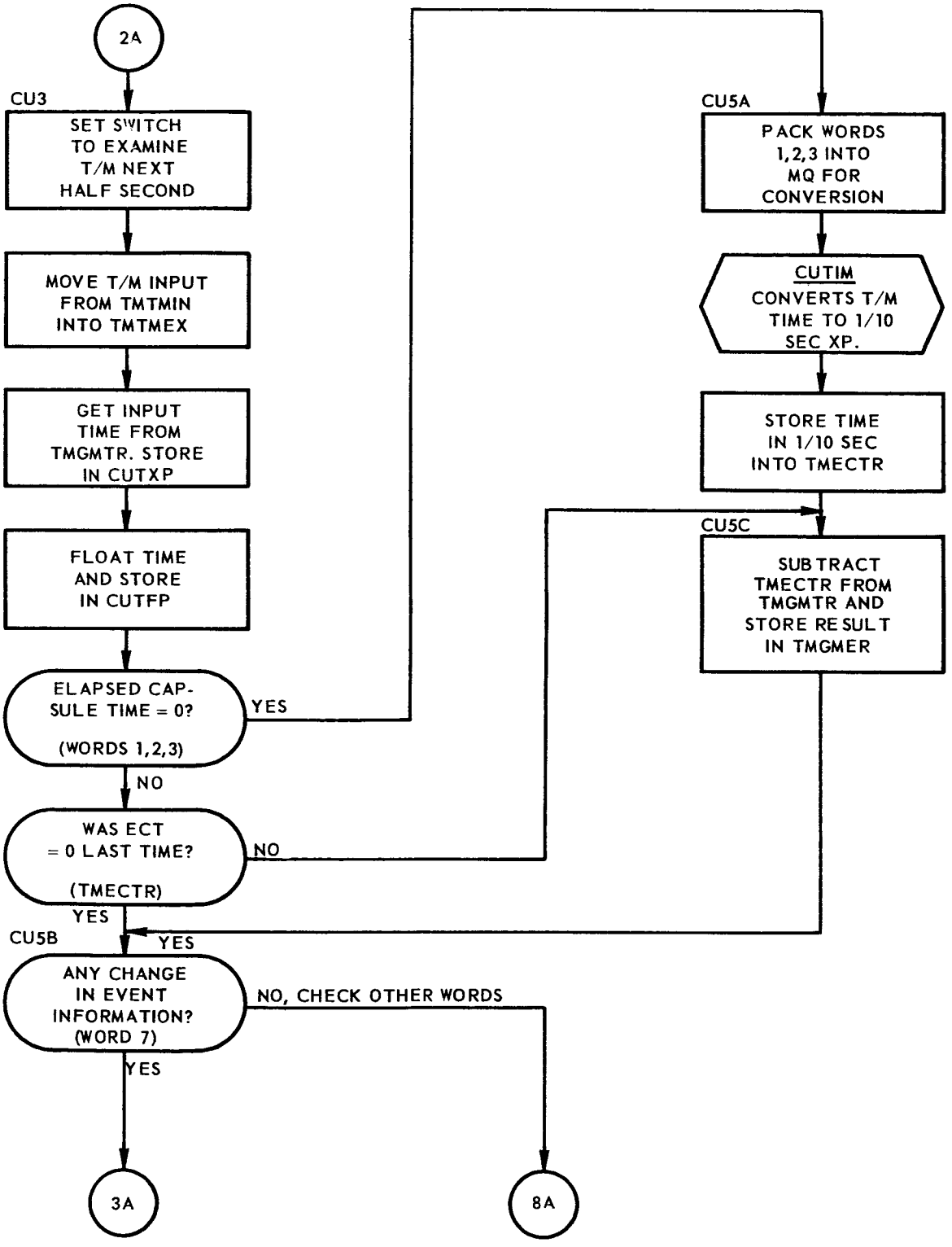


FIGURE 3-12. MYTMIN PROGRAM FLOW CHART *(Sheet 2 of 9)*

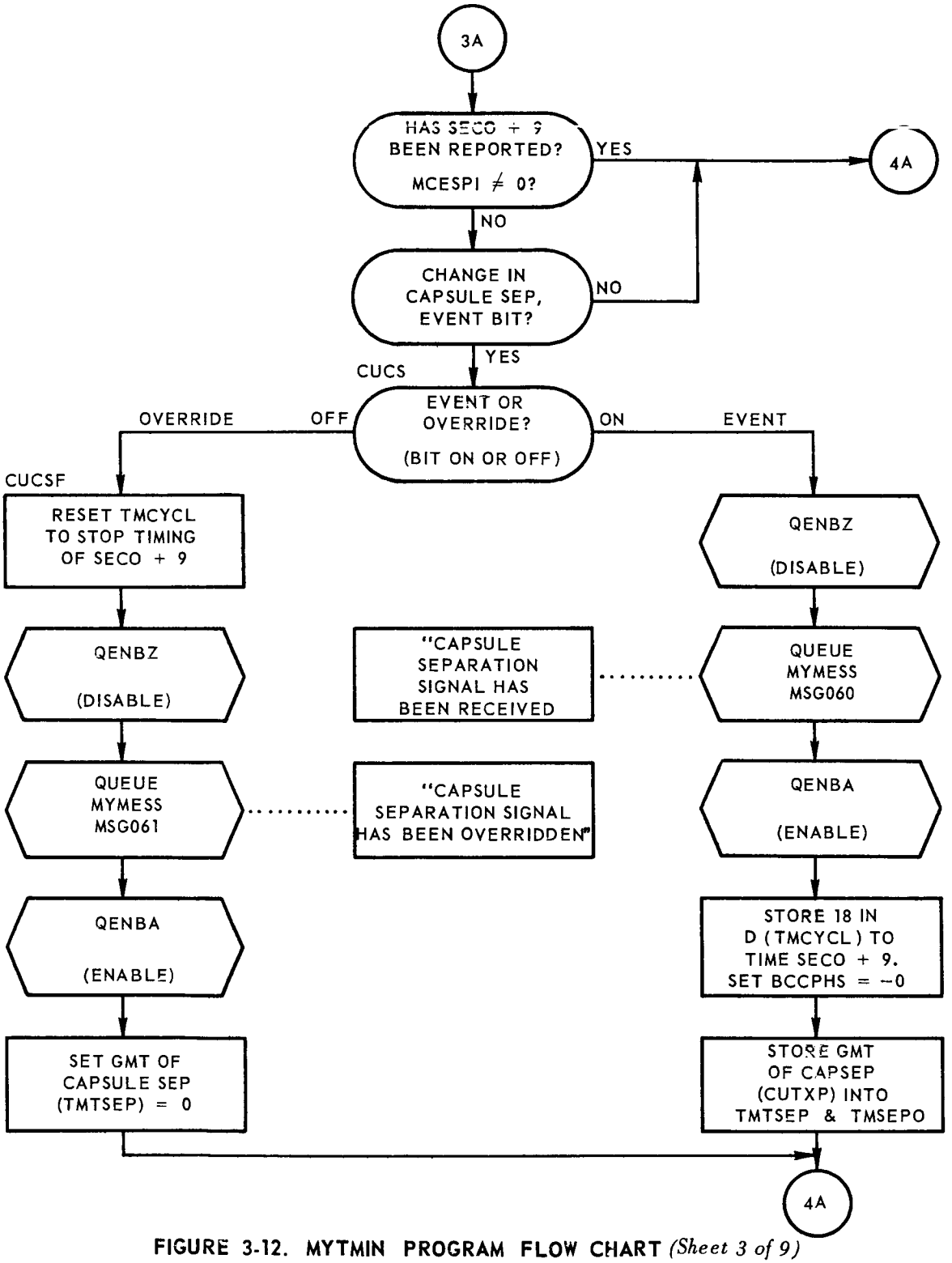


FIGURE 3-12. MYTMIN PROGRAM FLOW CHART *(Sheet 3 of 9)*

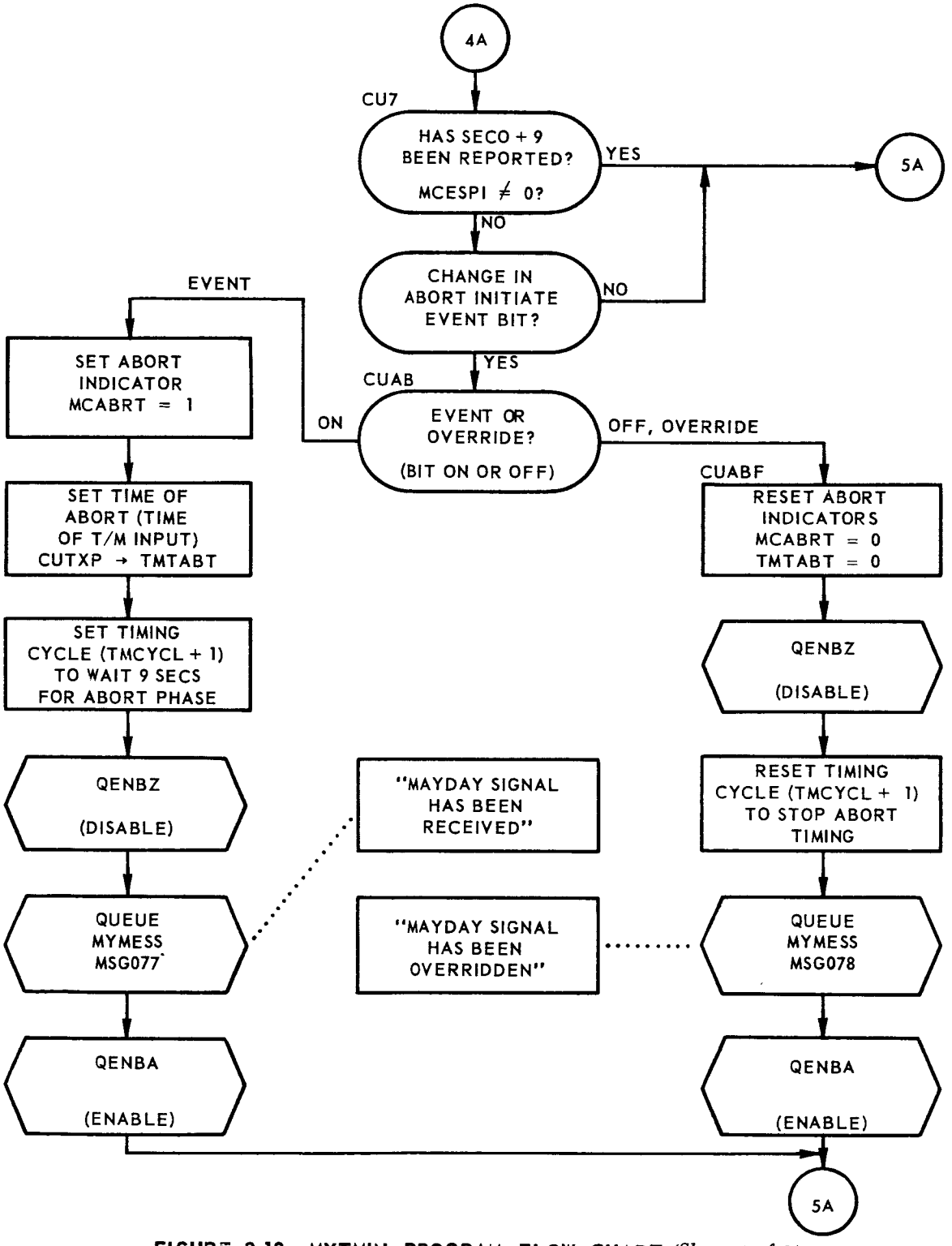


FIGURE 3-12. MYTMIN PROGRAM FLOW CHART *(Sheet 4 of 9)*

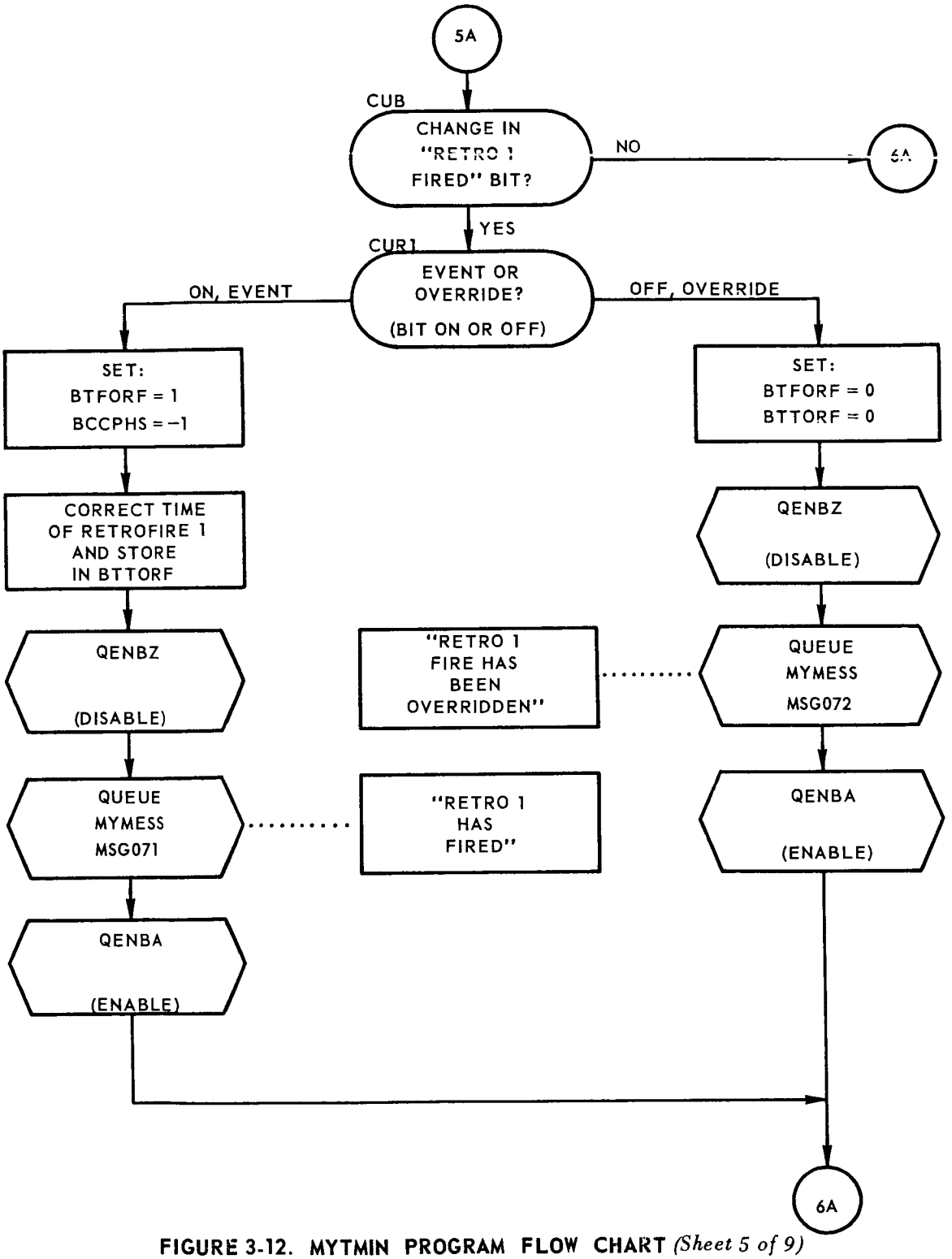


FIGURE 3-12. MYTMIN PROGRAM FLOW CHART *(Sheet 5 of 9)*

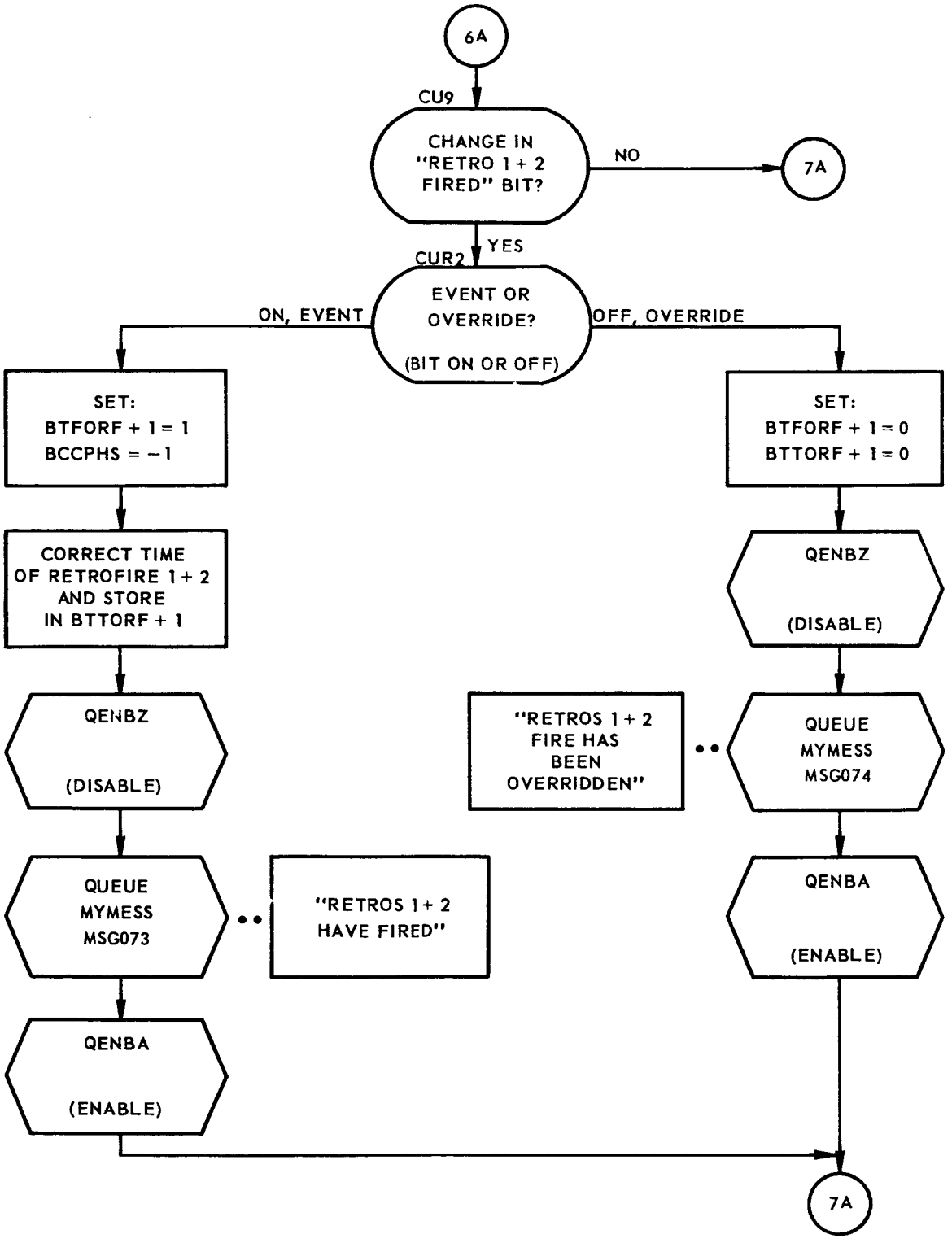


FIGURE 3-12. MYTMIN PROGRAM FLOW CHART *(Sheet 6 of 9)*

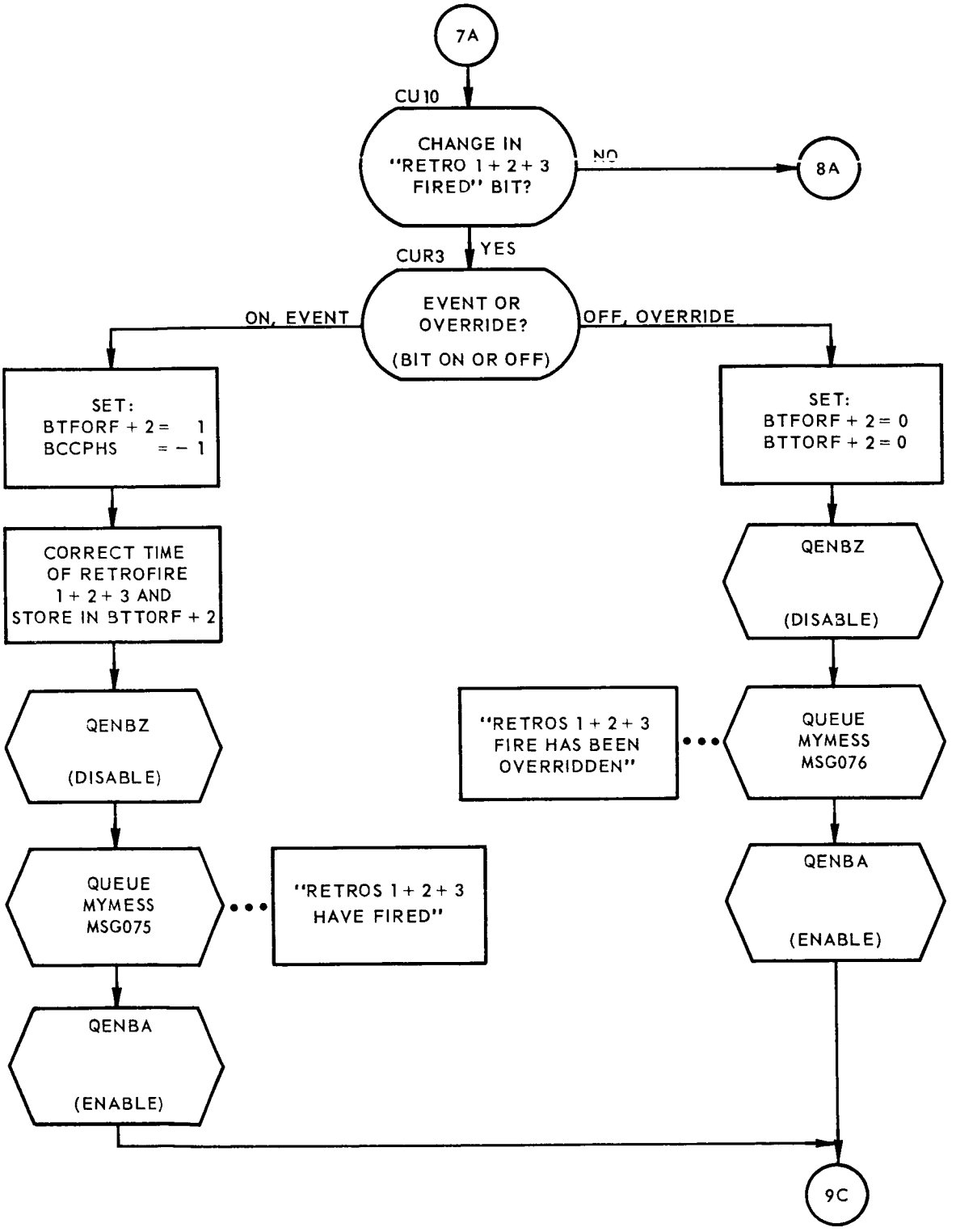


FIGURE 3-12. MYTMIN PROGRAM FLOW CHART *(Sheet 7 of 9)*

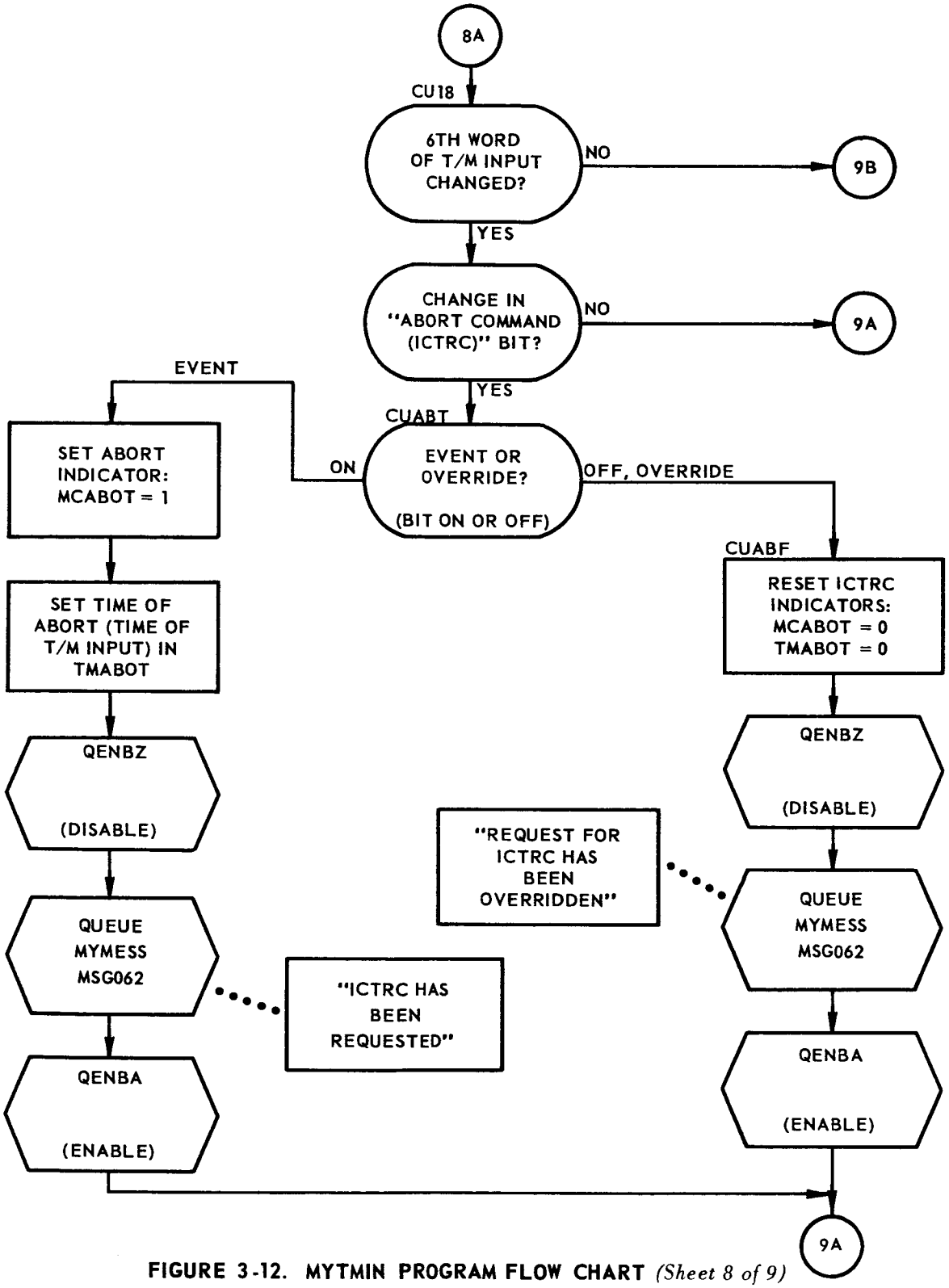


FIGURE 3-12. MYTMIN PROGRAM FLOW CHART *(Sheet 8 of 9)*

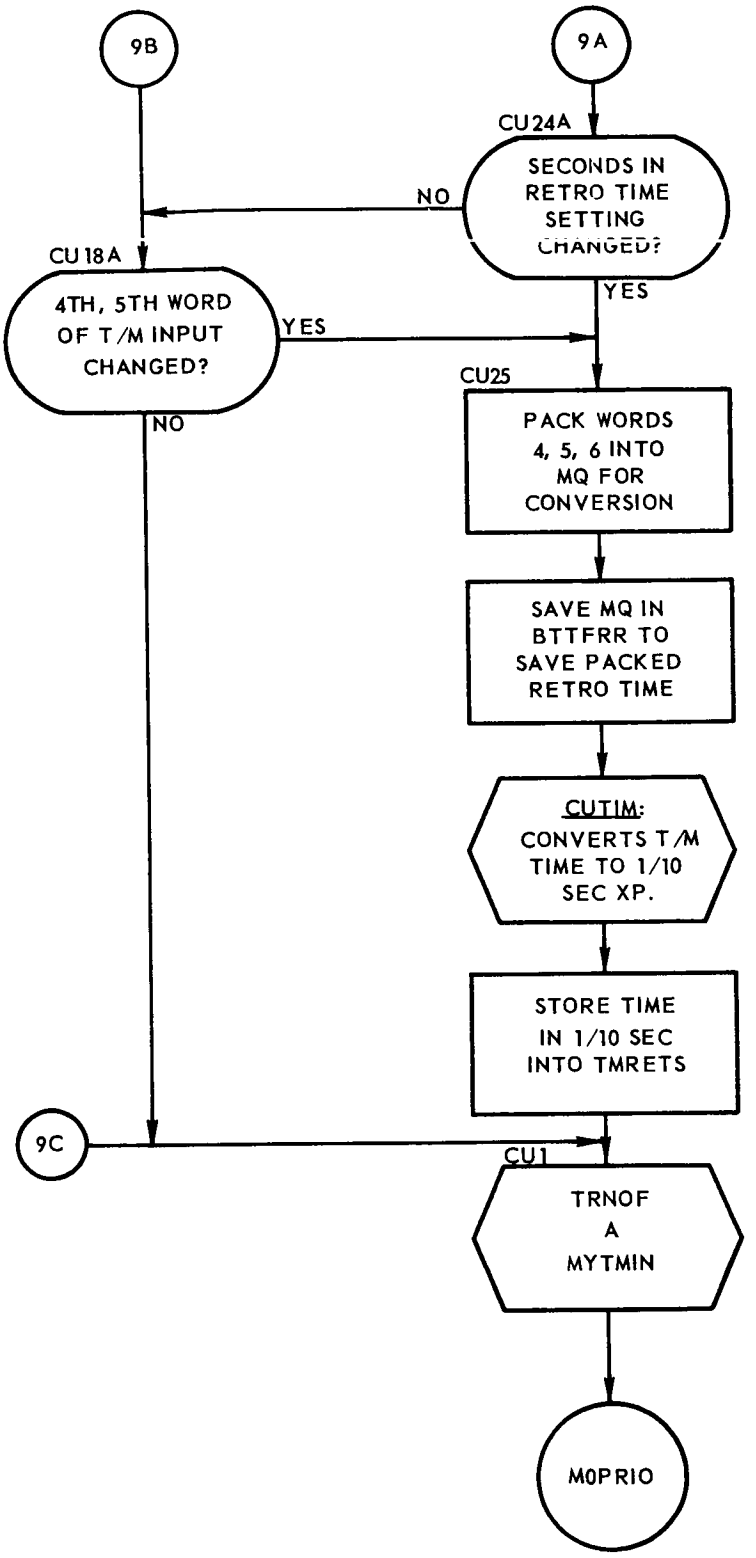


FIGURE 3-12. MYTMIN PROGRAM FLOW CHART *(Sheet 9 of 9)*

3.12 MONITOR ENGINE CUT-OFF EARLY PROCESSOR (MYECOE)

When a telemetry message indicates an abort has been initiated due to the premature cut-off of the launch vehicle's sustainer engine, a countdown clock begins timing for nine seconds. If, after nine seconds, the abort initiate signal has not been overridden, MYECOE receives control and sets communication cells which indicate to other programs in the Bermuda Program System the occurrence of the abort initiated due to early engine cut-off.

The flow chart for the MYECOE program is shown in Figure 3-13.

3.12.1 Input Requirements

Input to MYECOE consists of:

a) TMTABT—one-cell table containing the time of the abort initiate (ECOE). This cell is set by the telemetry processor MYTMIN.

b) TMCYNO—contains in the decrement the cycle (half-second) time for MYECOE and in the address the location of MYECOE. Since MYECOE may be executed only once, the cycle time is infinite. The decrement therefore contains an effective infinite bound of 32,767.

c) K00001—constant, 1.

d) Macros Used—TRNOF and TRNON.

3.12.2 Output Requirements

Output from MYECOE consists of:

a) TMTSEP—the time of the abort initiate is stored in TMTSEP to indicate the time of the capsule's separation from the launch vehicle.

b) MCESPI—1 is stored in MCESPI to indicate the time of capsule separation is now available.

c) MCSECO—1 is stored in MCSECO to indicate the time of sustainer engine cut-off is now available.

d) TMCYCL—32,767 is stored in the decrement of TMCYCL to stop the timing of capsule separation (ECOE) plus nine seconds and to prevent further entry to MYECOE.

3.12.3 Method

MYECOE receives control from M0PRIO after nine seconds have elapsed since the telemetry report of the sustainer engine's early cut-off. MYECOE turns

its A indicator on, its B indicator off and then stores a 1 in MCESPI and MCSECO to indicate the time of both capsule separation and abort initiate is available. The time of the abort initiate is then stored in the cell provided for the time of the capsule separation. MYECOE resets its own entry (TMCYCL) in the half-second timing table to effectively stop the timing of capsule separation plus nine seconds. MYECOE turns off its A indicator and program controls returns to M0PRIO.

3.12.4 Usage

MYECOE is entered from, and exits to, M0PRIO.

a) Storage Required—14 locations

b) MYECOE Uses:

1) Macros—TRNOF and TRNON

2) Parameters—A, B and MNECOE

3) Communication cells—MCESPI and MCSECO

4) Tables—TMCYCL, TMCYNO, TMTABT and TMTSEP

5) Constant—K00001

c) Time Required—0.42 millisecond

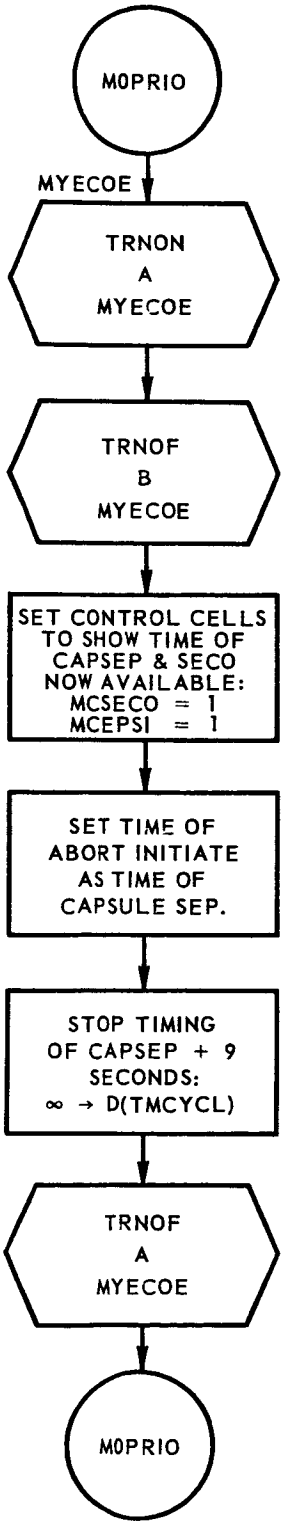


FIGURE 3-13. MYECOE PROGRAM FLOW CHART

3.13 MONITOR SUSTAINER ENGINE CUT-OFF PROCESSOR (MYSECO)

When a telemetry message indicates sustainer engine cut-off (SECO) has occurred, a countdown clock begins timing for nine seconds. If, after nine seconds, the SECO signal has not been overridden, MYSECO receives control and sets communication cells which indicate to the other programs of the Bermuda Program System the occurrence of SECO.

The flow chart for the MYSECO program is shown in Figure 3-14.

3.13.1 Input Requirements

Input to MYSECO consists of:

a) TMCYNO+1—contains in the decrement the cycle time for MYSECO. Since MYSECO may be executed only once, the cycle time is infinite. The decrement therefore contains an effective infinite bound of 32.767.

b) K00001—constant, 1.

c) Macros Used—TRNOF and TRNON.

3.13.2 Output Requirements

Output from MYSECO consists of:

a) TMCYCL+1—32,767 is stored in the decrement of TMCYCL+1 to stop the timing of sustainer engine cut-off (SECO) plus nine seconds and to prevent further entry to MYSECO.

b) MCESPI—1 is stored in MCESPI to indicate the time of capsule separation is now available.

c) MCSECO—1 is stored in MCSECO to indicate the time of sustainer engine cut-off is now available.

3.13.3 Method

MYSECO receives control from M0PRIO after nine seconds have elapsed since the report of sustainer engine cut-off. MYSECO turns its A indicator on, its B indicator off and stores a 1 in MCESPI and MCSECO to indicate the time of capsule separation and sustainer engine cut-off is now available. MYSECO resets its own entry (TMCYCL+1) in the half-second timing table to prohibit further entry to MYSECO. The processor turns off its A indicator and returns program control to M0PRIO.

3.13.4 Usage

MYSECO is entered from, and exits to, M0PRIO.

a) Storage Required—12 locations

b) MYSECO Uses:

1) Macros—TRNOF and TRNON

2) Parameters—A, B and MNSECO

3) Communication cells—MCESPI and MCSECO

4) Tables—TMCYCL and TMCYNO

5) Constant—K00001

c) Time Required—0.35 millisecond

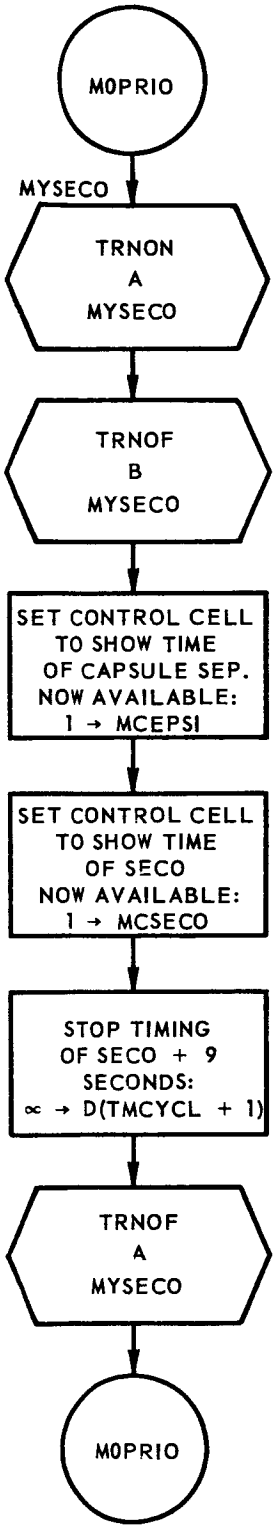


FIGURE 3-14. MYSECO PROGRAM FLOW CHART

3.14 TIMING CONTROL

The Data Communications Channel provides the Bermuda Program System with all necessary timing facilities. Three DCC subchannels supply timing control. The timing subchannels utilized in the Mercury Program System and their associated Monitor Processors are described below:

Subchannel 4, when activated, increments a cell in computer core storage every 8-1/3 milliseconds. When the count reaches 59, the next 8-1/3 milliseconds increment resets the count to zero—60 increments of 8-1/3 milliseconds comprise a half-second interval—whereupon subchannel 4 causes an automatic interrupt of the computer. The half-second interrupt subchannel is activated when the Greenwich Mean Time is entered into the computer (see subchannel 5, below) and is never deactivated until the processing for the mission is terminated. When the half-second interrupt subchannel is activated, a 1 in the sign bit of the mask addressed by the PSLF instruction resets the 8-1/3 millisecond counter to zero. The sign bit of the PSLF mask must be zero thereafter.

Interrupts on subchannel 4 are serviced by the Monitor Half-Second Trap Processor, MTHFSC. This trap processor controls a half-second timing table which provides a means of timing set intervals within Monitor as well as a means of informing the Monitor Main Controller Priority Program, M0PRIO, at set intervals that the cyclic, or time related, programs should be entered. The Monitor Minute Processor, MYMINS, receives control every time its entry in the half-second timing table is reduced to zero.

MTHFSC and MYMINS are described in detail in subsections 3.19 and 3.20, respectively.

Subchannel 5, when activated, interrupts the computer in synchronization with an external timing pulse. The external timing pulse for the Mercury Program System is received from the National Bureau of Standards' radio station WWV and produces a computer interrupt every minute (accurate to one millisecond). This subchannel is serviced by two trap processors. The Monitor Initial WWV Trap Processor, MTIWWV, receives control with each WWV interrupt until after the GMT time base is entered into the computer. When MTIWWV has established the time base and initializes the timing cells of the Bermuda Program System, MTIWWV replaces its own entry in the DCC control table with the entry for the Monitor WWV Trap Processor, MTWWWV. This trap processor maintains a minute time cell and examines the other time cells for deviations from the WWV standard.

MTIWWV and MTWWWV are described in detail in subsections 3.17 and 3.18, respectively.

Subchannel 6, when activated, decrements a specified cell in computer core storage every 8-1/3 milliseconds. When the cell is reduced to zero, subchannel 6 causes an automatic interrupt of the computer and the interrupt deactivates the

subchannel. Subchannel 6 may time any interval from 8-1/3 milliseconds (1 in the specified cell) to 4.25 seconds (250 in the specified cell).

Although at present this subchannel is not utilized in the Mercury Program System, provisions have been made to allow its use if the need should arise.

3.15 MONITOR INITIAL WWV TRAP PROCESSOR (MTIWWV)

MTIWWV provides a standard time base for all processing and data transmission in Project Mercury by entering the Greenwich Mean Time into the Mercury Program System.

The flow chart for the MTIWWV trap processor is shown in Figure 3-15.

3.15.1 Input Requirements

Input to MTIWWV consists of:

a) The Console Keys—the Greenwich Mean Time at which the time initialization of the system is desired is entered into the console keys with the following format: the octal equivalent of the hours is entered into the decrement and the octal equivalent of the minutes is entered into the address, both right justified. The sign key must be depressed less than one minute prior to the time thus set.

b) CYMSK—mask, 000 000 000 077, used to clear all bit locations except the last six in a computer word.

c) CYMS1—contains the location of the entry to MTWWWV, the program which processes all one-minute WWV traps after the time initialization has taken place.

d) CYPS4—mask, -030 000 000 000, used to activate DCC subchannels 4 and 5 to reset the 8-1/3 millisecond counter of the DCC.

e) K00015—constant, 15 (decimal).

f) K00060—constant, 60 (decimal).

g) K00120—constant, 120 (decimal).

h) Macro used—QUEUE.

3.15.2 Output Requirements

Subchannel 5 of the DCC is activated, and the 8-1/3 millisecond counter is synchronized with the one-half second trap. Other output from MTIWWV consists of:

a) TMDCCC+7 —the location of the entry to the trap processor (MTWWWV) to handle all succeeding WWV traps is stored in the address of the entry in the table TMDCCC which corresponds to the DCC subchannel for WWV interrupt signals. This table is used by M0DCCC to interpret DCC traps.

b) MCMINS—contains the current time in minutes in fixed point.

c) MCHFSC—contains the current time in half seconds in fixed point.

d) MCWWWV—contains the current time in 8-1/3 milliseconds in fixed point.

e) MSG020—contains the current time (which is the time entered into the Bermuda Program System) in minutes is stored in MSG020. This cell is used by the on-line message programs to provide the time field for the message: "WWV Time entered is . . . . . ."

3.15.3 Method

During the execution of MYINTL subchannel 5 is activated and every minute a trap occurs with the one minute pulse signal from radio station WWV. M0DCCC examines the DCC trap and transfers control to MTIWWV which examines the contents of the computer's console keys. If the sign key is up, MTIWWV transfers control directly back to M0PRIO. The time at which initialization is desired is entered into the keys. Within one minute prior to the time set in the keys, the sign key must be depressed. When MTIWWV finds the sign key down, the time entered is processed as the current Greenwich Mean Time, that is, the time base for the Bermuda Program System.

Subchannel 4 (for half-second interrupts) is activated and the 8-1/3 millisecond counter is reset. MTIWWV converts the Greenwich Mean Time into fixed point values in minutes, half-seconds, and 8-1/3 milliseconds and stores these values into the communication cells provided for them. In the table used by M0DCCC for control transfer after DCC traps, the entry for subchannel 5 is changed so that all succeeding WWV traps will be processed by MTWWWV. MTIWWV queues the message program to print two on-line messages stating that normal operations have begun and that the WWV time has been entered. The latter message will also include the time at which GMT was entered into the computer.

Control transfers to M0PRIO.

3.15.4 Usage

MTIWWV is entered from M0DCCC following a trap on DCC subchannel 5; MTIWWV exits to M0PRIO.

a) Storage Required—33 locations

b) MTIWWV uses:

1) Macro—QUEUE

2) Parameter—MNMESS

3) Table—TMDCCC

4) Communication cells—MCMINS, MCWWWV, MCHFSC and CYMS1

5) Masks—CYPS4 and CYMSK

6) Constants—K00120, K00060 and K00015

c) Time Required—1.72 milliseconds

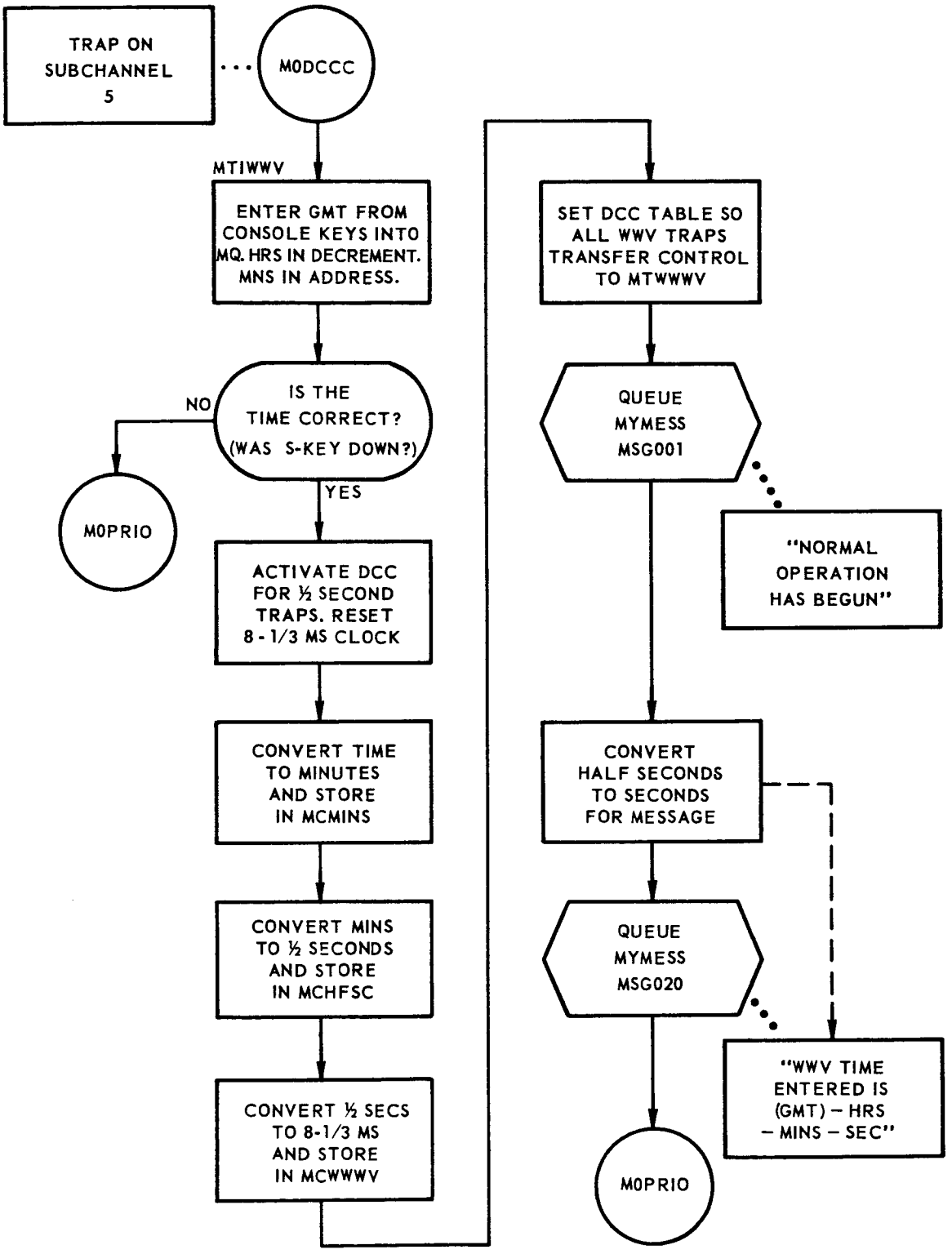


FIGURE 3-15. MTIWWV PROGRAM FLOW CHART

3.16 MONITOR WWV TRAP PROCESSOR (MTWWWV)

After MTIWWV enters the time base into the computer, the Initial WWV Trap Processor replaces its own entry in TMDCCC with the entry for MTWWWV, so that all subsequent WWV traps will be processed by the normal WWV Trap Processor.

MTWWWV compares the time generated by the computer from the half-second traps and the 8-1/3 millisecond counter to the time standard as received from WWV. If these times differ by an amount in excess of 8-1/3 milliseconds, an audible and visible indicator is set on the Output Status Console.

The flow chart for the MTWWWV trap processor is shown in Figure 3-16.

3.16.1 Input Requirements

Input to MTWWWV consists of:

a) TM8.3M—single-cell table containing the count of 8-1/3 milliseconds since last exact half second.

b) MCHFSC—communication cell containing the current GMT, expressed in half seconds.

c) MCWWWV—communication cell containing the count of 8-1/3 milliseconds one minute ago.

d) CSC5—mask, 000 000 000 001, used by MYSENS to turn on the "WWV ERROR" indicator on the Output Status Console (the indicator has an associated audible tone).

e) CSC6—mask, 000 001 000 000, used by MYSENS to turn off the "WWV ERROR" indicator on the Output Status Console.

f) K07200—constant, 7200 (decimal).

g) K00015—constant, 15 (decimal).

h) K00001—constant, 1.

i) Macro Used—QUEUE.

3.16.2 Output Requirements

Output from MTWWWV consists of:

a) MCWWWV—the contents of MCWWWV are incremented by 7200 (1 minute = 7200 8-1/3 milliseconds).

b) The "WWV ERROR" indicator light and the audible tone are turned on if |(MCHFSC x 15 x 2 x 2) + |TM8.3M| - MCWWWV| >1. If the difference is one or less, the indicator light and audible tone are turned off.

3.16.3 Method

MTWWWV receives control from M0DCCC when the one-minute WWV signal is received by the DCC. MTWWWV increments the 8-1/3 millisecond count in MCWWWV by 7200, the number of 8-1/3 milliseconds in one minute.

The internal time is computed by converting the half-second count (MCHFSC) into a 8-1/3 millisecond count. The 8-1/3 millisecond count since the last exact half-second (the magnitude of TM8.3M) is added to give the computer internal time as a count of 8-1/3 milliseconds. This time is compared to the count contained in MCWWWV. If the difference exceeds 1 (one 8-1/3 millisecond), the "WWV ERROR" indicators—light and audible tone—on the Output Status Console are turned on. If the difference is one or less, the indicators are turned off.

Program control transfers to M0PRIO.

3.16.4 Usage

MTWWWV is entered from M0DCCC following a trap on DCC subchannel 5; MTWWWV exits to M0PRIO.

a) Storage Required—18 locations

b) MTWWV uses:

1) Macro—QUEUE

2) Parameter—MNSENS

3) Communication cells—MCWWWV and MCHSFC

4) Table—TM8.3M

5) Constants—K07200, K00015 and K00001

6) Masks—CSC5 and CSC6

c) Time Required—0.74 millisecond

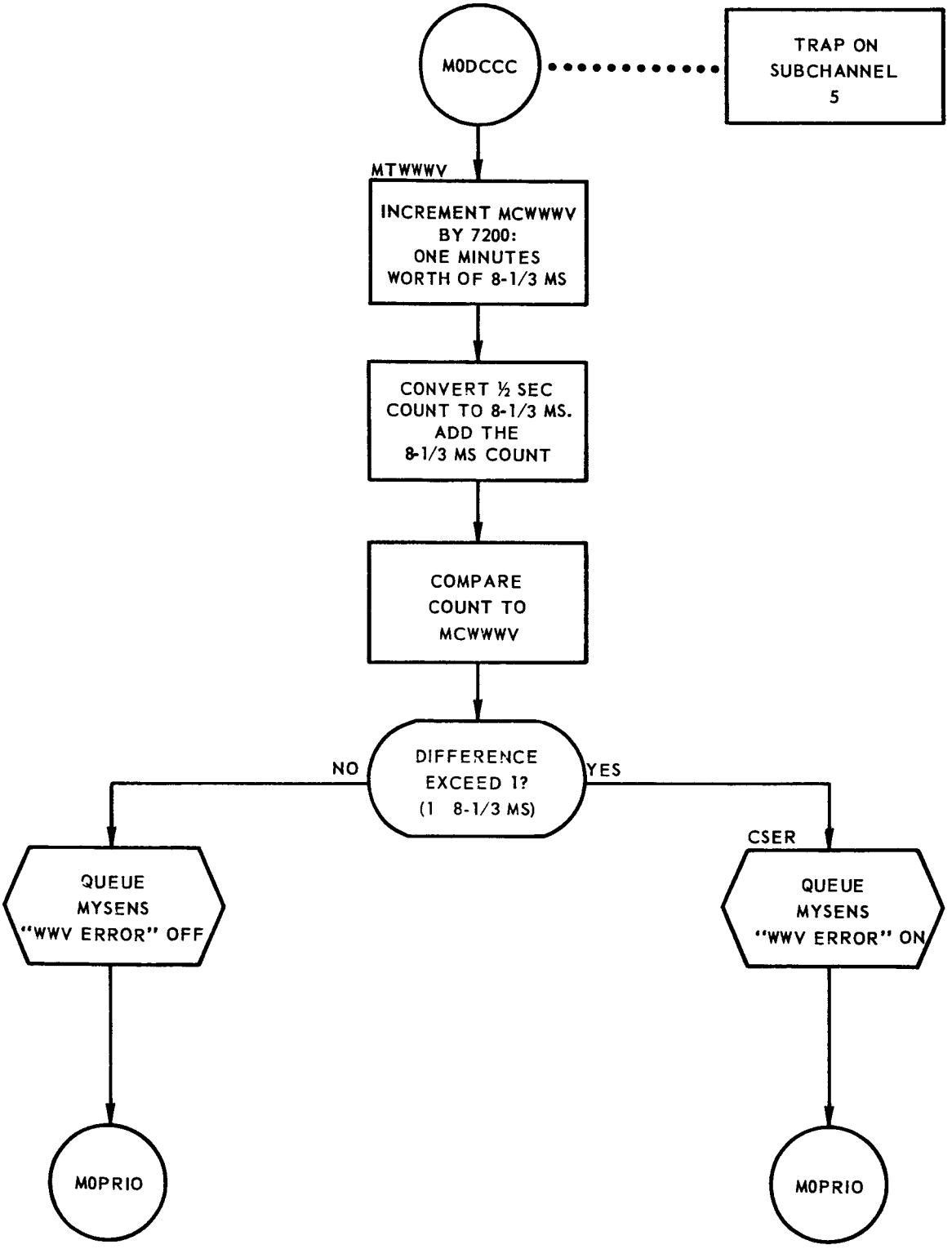


FIGURE 3-16. MTIWWV PROGRAM FLOW CHART

3.17 MONITOR HALF-SECOND TRAP PROCESSOR (MTHFSC)

Every half-second a trap on DCC subchannel 4 gives control to MTHFSC. This trap processor updates time cells, initiates the transmission of a telemetry input message and sets controls for programs in the Bermuda Program System which are executed with a specified frequency.

The flow chart for the MTHFSC trap processor is shown in Figure 3-17.

3.17.1 Input Requirements

Input to MTHFSC consists of:

a) M—parameter, designating the number of entries in the timing tables, TMCYNO and TMCYCL.

b) TMCYCL, TMCYNO—the Monitor Timing Tables, TMCYCL and TMCYNO, when used with the half-second trap processor, provide a means of giving control on a set time period to programs in the Bermuda Program System. The decrement of the entry in TMCYNO provides the half-second period of a program. The decrement of the corresponding entry in TMCYCL provides the half seconds remaining before control is given to the program. The address of TMCYCL provides the routine number which locates the program in the basic monitor tables. By changing the contents of the decrement for its entry in TMCYNO, a program's frequency of execution may be changed at will, and time delays may be generated for any program having an entry in the timing tables. For example, if program A is to be used once during the Mercury mission, and only X half seconds after a specific event has occurred, the decrement of the entry in TMCYNO for program A is initially set to the maximum value, 32,768. Program A is effectively delayed for a period of over five hours. When the event occurs, the quantity X is stored in the decrement of the entry in TMCYNO. X half seconds later program A receives control, and, after completing processing, program A places the maximum value back in the timing count, thereby suppressing itself for another five hours. (An alternate method of suppressing a periodic routine is discussed under MCHFS1, see below).

c) TMNMSK + B - 1—mask, 200 000 000 000, used to turn on indicator B in TMPRIO.

d) TMREFR—Monitor Reference Table which provides the location in the priority table for the routine whose indicator B is to be turned on.

e) MCHFS1—communication cell originally set by MYINIT. This cell controls the selection in the examination of the timing count by MTHFSC in that TMCYCL is examined for as many entries as are specified by

MCHFS1. If the contents of MCHFS1 are 2 and M is equal to 5, MTHFSC will skip over the first three (5-2) entries in TMCYCL. However, the entries are always examined in the order that they appear in the table. As an example of control provided by MCHFS1, if program A is executed every 20 seconds until a specific event in the Mercury mission occurs, and is not needed thereafter, the entries for program A should be the first in TMCYCL and TMCYNO. When the event occurs, some program discerns the fact and decrements the contents of MCHFS1 by one. The first entry in TMCYCL is no longer examined by MTHFSC and, consequently, program A is suppressed. If program A were needed at a later time, the opposite approach—restoring MCHFS1 to its initial value—would be needed.

f) MCHFSC—communication cell containing the current GMT, expressed in half seconds.

g) MCTM02—mask, 040 000 000 000, used to control—activate or deactivate—subchannel 3 of the DCC.

h) K00001—constant, 1.

i) K00003—constant, 3.

j) KD0001—constant, 000 001 000 000 (1 in the decrement).

k) Macro Used—QPSLF.

3.17.2 Output Requirements

Output from MTHFSC consists of:

a) TMCYCL—the timing count in the decrement of each examined entry in TMCYCL is reduced by one. If the count for any entry is reduced to zero, the B indicator in the priority table TMPRIO is turned on for the routine corresponding to that entry.

b) MCHFSC—the contents of MCHFSC (the half-second GMT) are incremented by one.

c) MCTM04—the maximum number of times (3) the external telemetry input buffer may be transmitted into core storage during this half second is set in MCTM04.

d) If the last telemetry transmission has been completely processed, DCC subchannel 3 is activated and a new telemetry message is read into core storage.

3.17.3 Method

MTHFSC receives control from M0DCCC when the half-second trap occurs on DCC subchannel 4. The contents of MCHFSC are incremented by one.

The program flow now depends upon the switch MSEXTM ("Examine Telemetry"). This switch determines the activation of DCC subchannel 3, the telemetry input subchannel, and is controlled by the monitor trap and ordinary processors for telemetry data (see MTTMIN and MYTMIN). If the switch is on, subchannel 3 is activated and the count for the number of times the telemetry buffer may be examined during the next half second is reset.

MTHFSC examines C(MCHFS1) last entires in the timing table, TMCYCL, skipping over the M-C(MCHFS1) initial entries (where M is a parameter specifying the number of entries in TMCYCL). For each entry examined the timing count in the decrement is reduced by one. When the timing count is reduced to zero, the B indicator for that routine is turned on and the count is reset to its original value.

Program control returns to M0PRIO.

3.17.4 Usage

MTHFSC is entered from M0DCCC following a trap on DCC subchannel 3; MTHFSC exits to M0PRIO.

a) Storage Required—22 locations.

b) MTHFSC uses:

1) Subroutine (switch)—MSEXTM

2) Macro—QPSLF

3) Parameters—B and M

4) Communication cells—MCACTV, MCHFS1, MCHFSC and MCTM04

5) Tables—TMCYCL, TMCYNO and TMREFR

6) Constants—K00001, K00003 and KD0001

7) Masks—MCTM02 and TMNMSK +B - 1

c) Time Required:

1) MSEXTM on—0.175 + 0.144 (MCHFS1) + 0.084B milliseconds

2) MSEXTM off—0.300 + 0.144 (MCHFS1) + 0.084B milliseconds where B is the number of examine entries in the timing tables for which the timing count is reduced to zero.

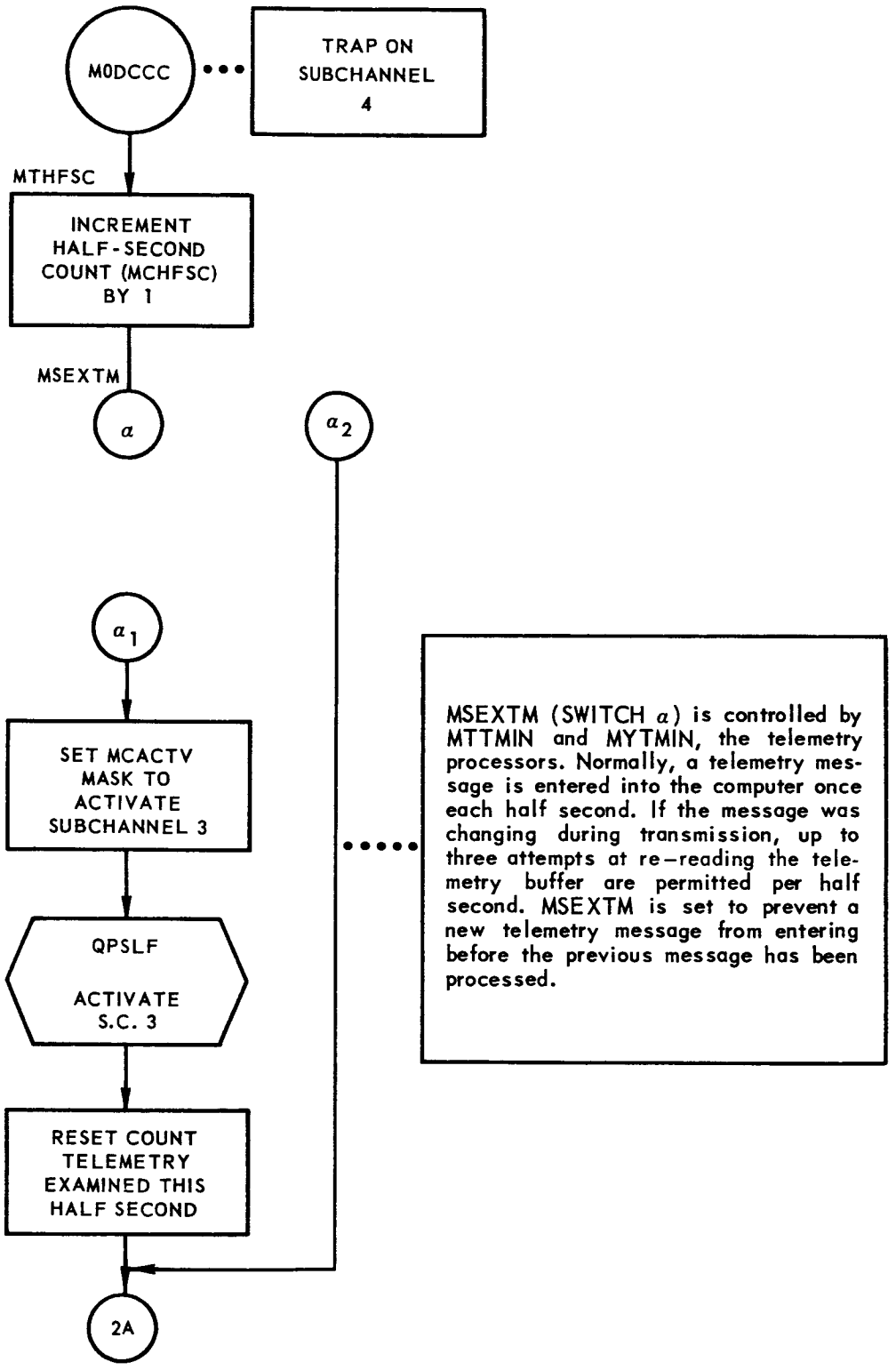


FIGURE 3-17. MTHFSC PROGRAM FLOW CHART *(Sheet 1 of 2)*

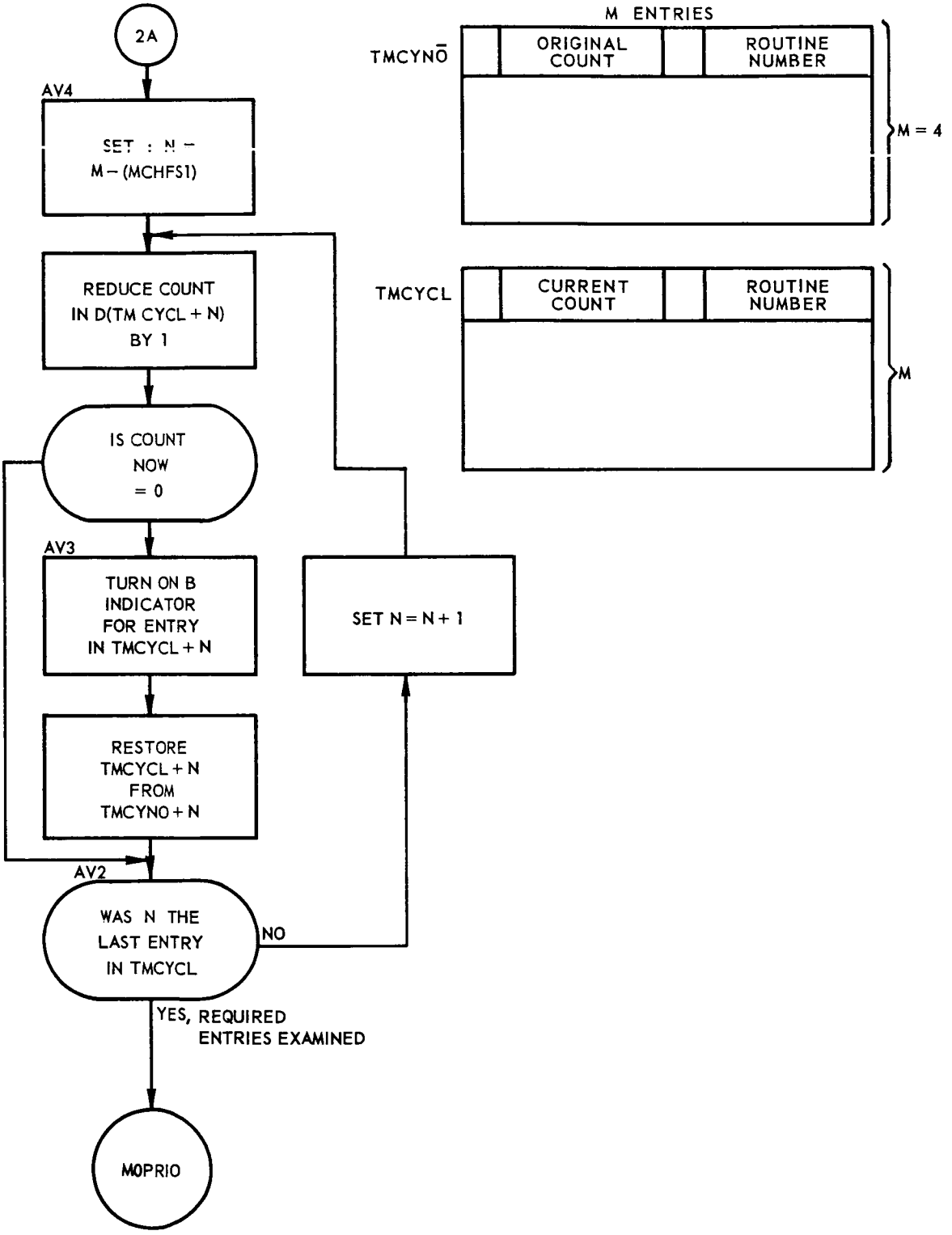


FIGURE 3-17. MTHFSC PROGRAM FLOW CHART *(Sheet 2 of 2)*

3.18 MONITOR MINUTES PROCESSOR (MYMINS)

Every minute (120 half-second intervals) a countdown clock in the half-second timing table is reduced to zero by MTHFSC. When this occurs, MTHFSC informs M0PRIO that control may be given to the Monitor Minutes Processor, MYMINS. This processor increments two minute timing cells and queues an on-line message which provides an external indication of the one-minute interval.

The flow chart for the MYMINS program is shown in Figure 3-18.

3.18.1 Input Requirements

Input to MYMINS consists of:

a) MCMINS—communication cell containing the Greenwich Mean Time in fixed-point seconds. This cell is originally set by the Monitor Initial WWV Trap Processor when GMT is entered into the computer. After the time initialization MCMINS is controlled entirely by MYMINS.

b) MINT1—internal cell containing the number of minutes elapsed since the initialization of time.

c) K00001—constant, 1.

d) Macros Used—QENBZ, QUEUE, TRNOF and TRNON.

3.18.2 Output Requirements

Output from MYMINS consists of:

a) MCMINS, MINT1—the GMT and elapsed minute cells, respectively, are incremented by one.

b) MSG022—the number of minutes elapsed since GMT initialization is stored in the address field of MSG022 to provide the variable field for the message: "Number of Minutes Pass Since Time Initialized . . ."

c) On-line message MSG022 is queued.

3.18.3 Method

MYMINS receives control from M0PRIO, turns its own A indicator on, its B indicator off and increments the two one-minute timing cells, MCMINS and MINT1. The number of elapsed minutes is stored in the address field of MSG022 for input to the on-line message processor. The computer is disabled and the on-line message processor queued to print out the number of minutes elapsed since time was entered into the computer. MYMINS turns off its A indicator and exits to M0PRIO.

3.18.4 Usage

MYMINS is entered from, and exits to, M0PRIO.

a) Storage Required—20 locations

b) MYMINS uses:

1) Macros—QENBZ, QUEUE, TRNOF and TRNON

2) Parameters—A, B, MNMESS and MNMINS

3) Communication cells—MCMINS

4) Internal cell—MINT1

5) Constant—K00001

c) Time Required—0.93 millisecond

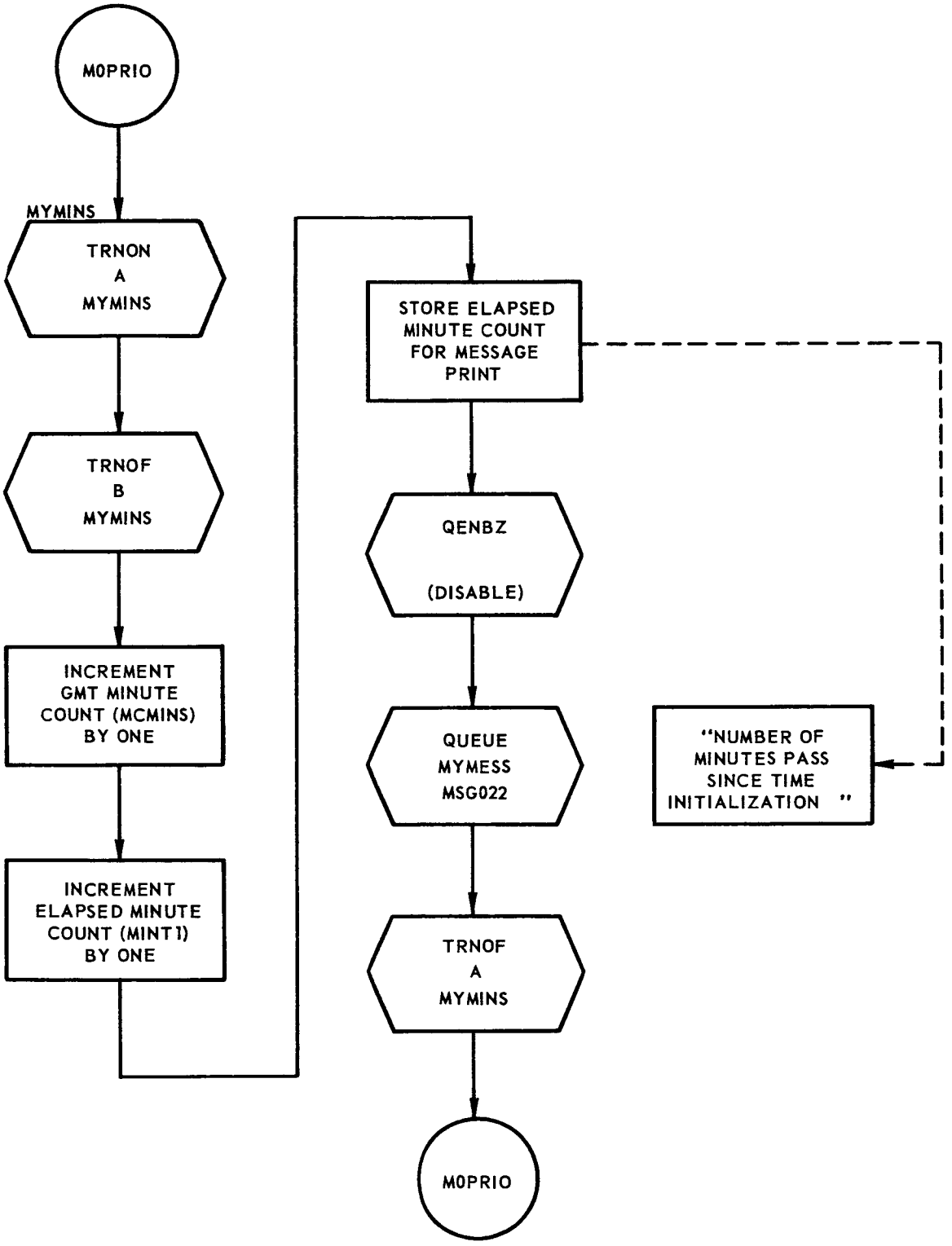


FIGURE 3-18. MYMINS PROGRAM FLOW CHART

Section 4.
CONTROL DURING PROCESSING

The basic computing cycle is initiated by the input of raw radar data and ends with the transmission of output messages. The Monitor control of the input processing is described in Section 3. The Monitor control of output format conversion and transmission is described in Section 5. The Monitor control of the programs which process the input data to produce the output data is described in this section.

4.1 THE PROCESSING CYCLE

The normal sequence of operations in the basic processing cycle is: editing, smoothing, short arc orbit computations and output computations (see Figure 4-1). A variation in the normal sequence occurs when the editing program finds excessive bad points from either radar. The normal sequence is described in general below followed by a general description of the modification in the basic cycle which result from excessive bad points.

4.1.1 The Normal Sequence

Editing: When raw radar data has been received and placed in the RAE tables for each radar, M0PRIO gives control to the editing program's prefix, MPEDIT (see Figure 4-2). This prefix turns the A indicator on and the B indicator off for BEDITS and turns on the C indicator for both BSMOOS and MYEBAD to suppress these programs. Then MPEDIT determines whether or not sufficient AN/FPS-16 data has been received for editing—four points are required for initial entry, one new point for each successive entry. If the program may be entered, MPEDIT sets the parameters and criteria locations in a specific cell for BEDITS and enters the program. The initial entry also requires that MPEDITS initialize the editing parameters.

BEDITS checks all new entries in the RAE table to insure that no gross errors exit. Erroneous values are replaced by interpolated points to keep the RAE tables complete, since most of the subsequent programs require for input a continuous set of data at equal time intervals. BEDITS examines the first four radar points to determine if they form, within acceptable limits, a straight line. If these points are acceptable, a fifth point is tested, etc. One or two bad points are replaced with interpolated values if the next point lies along the straight line. More than two bad points in succession causes BEDITS to search for a new four-point straight line and to interpolate for the missing points after a new line is

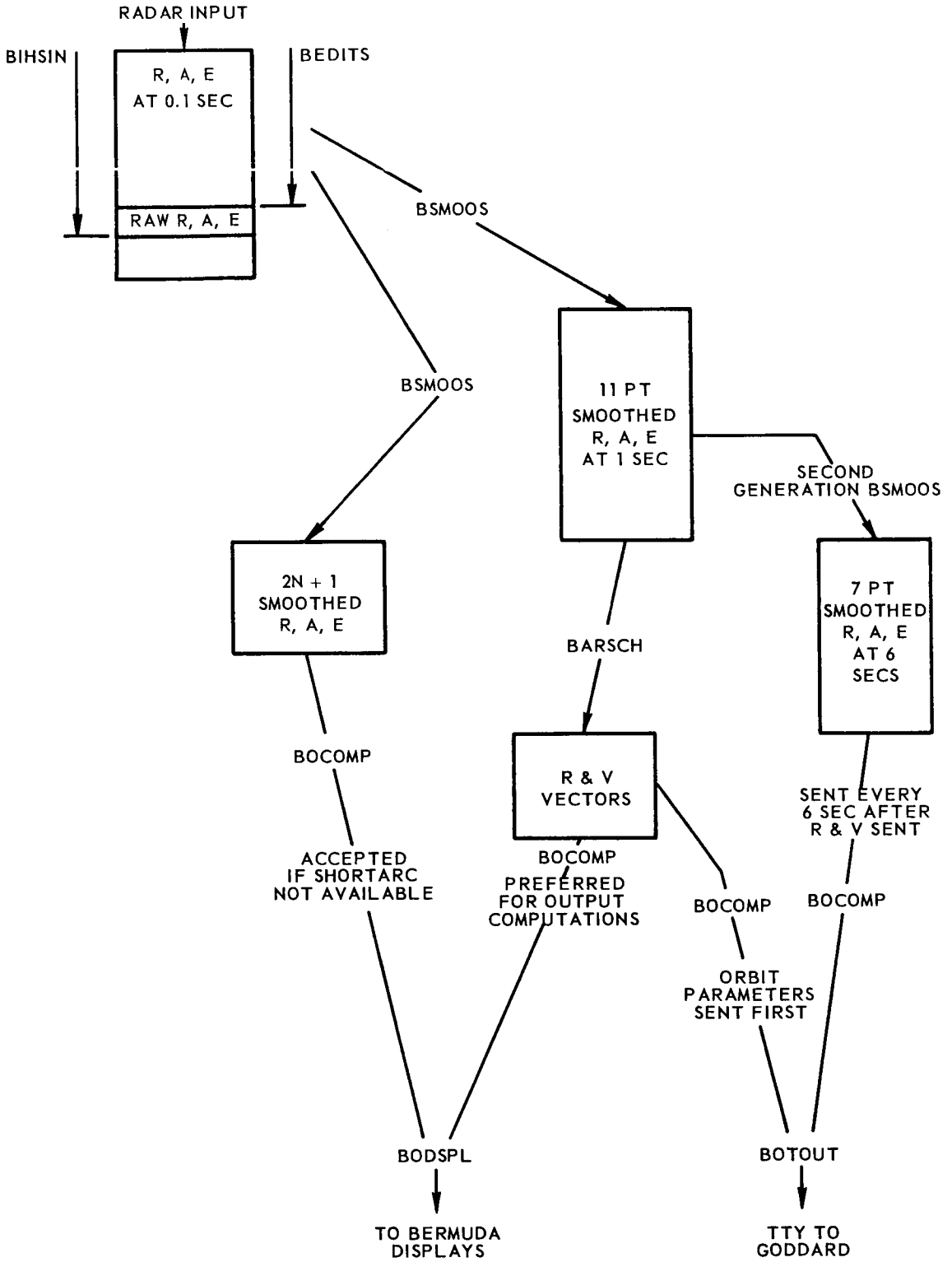


FIGURE 4-1. GENERAL FLOW DIAGRAM, BERMUDA CYCLE DATA FLOW

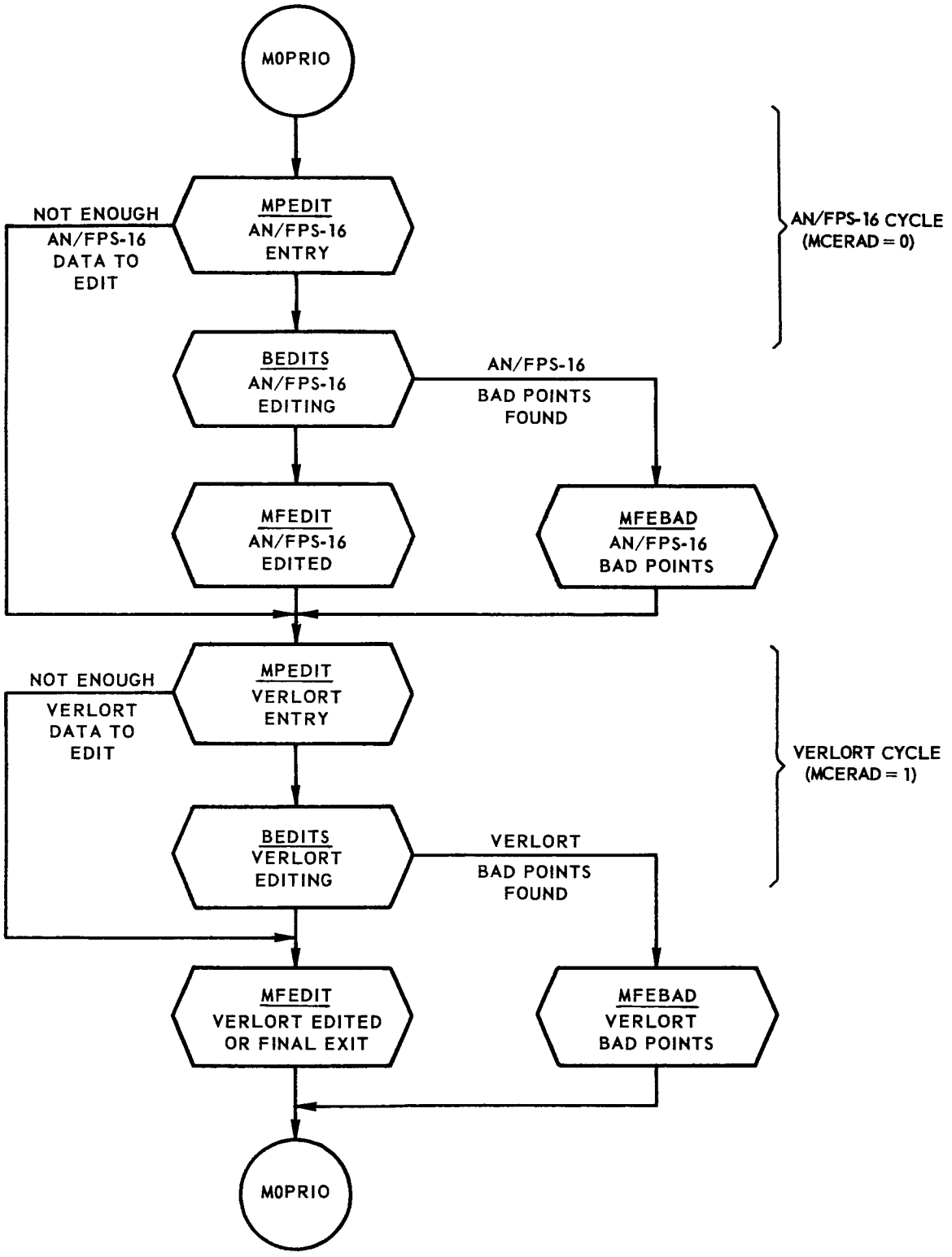


FIGURE 4-2. GENERAL FLOW DIAGRAM, BERMUDA EDIT CYCLE

determined. (If the number of consecutive bad points exceeds a preset limit, the normal basic cycle is interrupted. This event is discussed in 4.1.2.)

When BEDITS has completed editing AN/FPS-16 data, it exits to its suffix, MFEDIT, which turns on the B indicator for BSMOOS and returns control to the prefix to prepare the parameters and criteria for a Verlort editing entry. If the AN/FPS-16 entry could not be made, the prefix retains control and prepares the parameters and criteria for a Verlort entry to editing.

MPEDIT now determines whether or not sufficient Verlort data is available to enter editing. After editing Verlort BEDITS again exits to MFEDIT, as would MPEDIT if Verlort could not be edited. However, the B indicator for BSMOOS is turned on only if BEDITS is entered for at least one radar.

Each time a new radar point is received from either radar (excepting equipment malfunction, both radars simultaneously provide one radar point every tenth of a second), the high-speed input processor program, BIHSIN, must be given control or data could be lost. When BIHSIN completes processing, it exits to its suffix which turns on the B indicator for the editing processor program, BEDITS. At the completion of editing, BEDITS exits to its suffix, MFEDIT. If one second has elapsed since the last exit from BEDITS, the B indicator for the smoothing processor program, BSMOOS is turned on. Since the editing processor has a lower priority position than any other processor program in the basic cycle, BEDITS is not re-entered until the basic cycle is completed—the suffix of the smoothing processor program turns on the B indicator for the short arc computation program, the short arc suffix turns on the B indicator for the output computations processor, etc. While that part of the basic cycle which follows editing is being completed, the B indicator for BEDITS is being turned on every tenth of a second, but the priority position of BEDITS prevents its entry until the cycle begins anew. If, however, one second has <u>not</u> elapsed between entries to BEDITS, the B indicator for BSMOOS is turned off, since insufficient new data is available to profitably complete the basic cycle.

Following editing, whether the B indicator BSMOOS is turned on or off, the C indicators for BSMOOS and MYEBAD are turned off to unsuppress these programs, the B indicator for BEDITS is turned off and MFEDIT exits to M0PRIO.

BEDITS is described in detail in MC 106, <u>Bermuda Processing Programs</u>.

<u>Smoothing</u>: If sufficient new radar points have been edited, the suffix to BEDITS turns on the B indicator for BSMOOS, and M0PRIO gives control to MPSMOO, the prefix to the smoothing program. MPSMOO turns the A indicator on and the B indicator off for BSMOOS. For the initial BSMOOS entry for either radar, MPSMOO sets indicators requesting BSMOOS to perform 2N + 1 smoothing and to test the radar data for evidence of the termination of powered flight (SECO). For the initial entry for each radar, parameters and indicator cells must be initialized and, for every entry, the location of the smoothing parameters and tables must be provided for BSMOOS. As with editing, each radar has a separate and distinct set of parameters and tables.

From the edited radar data, BSMOOS produces all the smoothed radar data used by the remainder of the programs of the basic cycle. BSMOOS is composed of seven programs. The first of these is an 11 point smoothing program which uses edited but unsmoothed radar points to produce a smoothed point at one second intervals. The second program is a seven point smoothing program which uses seven 11-point smoothed values to produce a smoothed point at six second intervals (for later teletype transmission to Goddard). Three of the remaining five programs produce a general least-squares second degree smoothing. The last two programs use the output of the first three: one of them computes the range acceleration and examines the results for evidence of SECO; the other converts the second-degree smoothed RAE values to position and velocity vectors in XYZ coordinates. (After the short arc processing program—the next stage in the basic processing cycle—has generated a valid solution, the general least-squares second degree smoothing is no longer needed, although the short arc processing program makes use of the coordinate conversion program in BSMOOS.)

After smoothing is completed, as far as possible, for both radars, BSMOOS exits to its suffix. If BSMOOS had computed a time of SECO, an on-line message is printed. MFSMOO then turns on the B indicator for the short arc processing program, turns off the A indicator for BSMOOS and exits to M0PRIO.

BSMOOS is described in detail in MC 106, Bermuda Processing Programs.

Short Arc: M0PRIO enters the short arc program, BARSCH, via the prefix, MPSARC, which turns the A indicator on and the B indicator off for BARSCH. MPSARC then turns on the C indicator for BSMOOS to suppress it while the short arc program is in progress. MPSARC then determines whether or not SECO has been computed, since a short arc solution is valid only in free flight. Furthermore, MPSARC must provide BARSCH with the best estimate of the SECO time. If a SECO time is available from both capsule telemetry and smoothing computations, MPSARC accepts the smoothing-computed time unless it differs from the telemetry time by more than two seconds. If only one source has determined the time of SECO, MPSARC accepts that time as valid and gives control to BARSCH. If no time of SECO is available, MPSARC turns off the C indicator for BSMOOS, turns on the B indicator for the output computations program BOCOMP, turns off the A indicator for BARSCH and returns control to M0PRIO. The basic cycle continues—BARSCH is merely bypassed until its equations become valid.

The short arc program uses a method based on the Taylor Series expansion of the inertial position from the center of the earth to the capsule, utilizing the equations of motion of a two-body problem to obtain the expansion as a function of position and velocity. The input to BARSCH consists of the smoothed range, azimuth and elevation data produced by the 11 point smoothing program. When a short arc solution—the position and velocity vectors—has been obtained, the 2N+ 1 smoothing program is suppressed.

BARSCH exits to its suffix, MFSARC, which turns off the C indicator for BSMOOS, turns on the B indicator for BOCOMP, turns off the A indicator for BARSCH and exits to M0PRIO.

BARSCH is described in detail in MC 106, Bermuda Processing Programs.

Output Computations: M0PRIO enters the output computations program, BOCOMP, via the prefix, MPCOMP. The prefix turns the A indicator on and the B indicator off for BOCOMP. Since MPCOMP must examine communication cells and tables controlled by the telemetry processor, MPCOMP turns on the C indicator to suppress this higher-priority processor while the prefix has control. On the first entry, MPCOMP must initialize various cells used by BOCOMP. On every entry, MPCOMP computes the time of output and sets the flight phase indicators, using the MSETPH subroutine, and the times of retrofire, if available.

BOCOMP is actually a control program for a series of programs which provide the following output data: the capsule height, velocity, flight path angle; latitude and longitude of capsule's present position; latitude and longitude of impact for an immediate retrofire; the time of retrofiring for the next recovery area; the GO NO-GO decision; and several other quantities. Input to BOCOMP consists of: the proper phase designation; the position and velocity vectors from short arc or, if a short arc solution is not available, the output of the 2N + 1 smoothing program; and the time for output.

BOCOMP exits to its suffix, MFCOMP, at one of five different locations, where each location specifies a processing condition determined by BOCOMP. If the entry location designates new data has been prepared by BOCOMP, the B indicator for the output-to-displays processor, BODSPL, is turned on. If the orbit parameters have been prepared, MFCOMP turns on the B indicator for the teletype output-to-Goddard processor, BOTOUT. For every entry to MFCOMP, the A indicator for BOCOMP is turned off and control returns to M0PRIO.

BOCOMP is described in detail in MC 106, Bermuda Processing Programs.

4.1.2 Bad Points Variation in the Normal Sequence

When the editing process finds an excessive number of bad points of radar data, the exit from the editing processor is to the bad points suffix, MFEBAD. This suffix eliminates the waiting period for entry to smoothing since the bad points situation requires smoothing to proceed as far as possible with the good data now available. MFEBAD then sets control cells which indicate radar type, initial entry to Monitor Bad Points Processor, MYEBAD, and that smoothing should be suppressed for this radar, after the available good points are smoothed, until a linear fit has been re-established by the editing processor. MFEBAD turns on the B indicators for MPSMOO and MYEBAD, and exits to M0PRIO. Smoothing is entered and the basic cycle proceeds normally to completion.

On the next cycle, BEDITS edits the new data, but now the suffix MFEDIT suppresses the smoothing processor and turns on the B indicator for MYEBAD. Following the completion of editing on this cycle, M0PRIO finds smoothing suppressed and gives control to MYEBAD. On this initial entry MYEBAD only adjusts the editing parameters. The basic cycle continues but BSMOOS is bypassed.

On the following cycle, BEDITS is instructed by its prefix to find a new fit from the new points which have arrived. The cycle continues to flow from BEDITS to MYEBAD and on, skipping BSMOOS (only for the radar involved, unless both radars have simultaneous bad points).

The new fit entry to MYEBAD produces no results until BEDITS has produced a new fit and enough points are involved for artificial smoothing. When this happens, MYEBAD uses its artificial smoothing program to fill in the missing points in the bad data gap. If MYEBAD cannot artificially construct satisfactory smooth points, a special mask (-1) is stored in the missing locations of the smoothed point tables. The smoothing parameters are adjusted and MYEBAD exits to M0PRIO. With the next editing entry the program flow becomes normal again.

4.2 MONITOR PREFIX TO BEDITS (MPEDIT)

After the High-Speed Input Processor Program, BIHSIN, places complete radar points—range, azimuth and elevation—into the input RAE table for either radar, M0PRIO gives control to the Monitor Prefix to the editing processor, MPEDIT, which supplies BEDITS with the information necessary to edit the raw radar data. If sufficient radar points are available for editing, MPEDIT sets control cells which indicate to BEDITS the type of editing entry and which provide the locations of the editing parameters and criteria. If the radar data from either radar cannot be edited, MPEDIT returns control to M0PRIO via the suffix to the editing processor, MFEDIT.

The flow chart for the MPEDIT prefix is shown in Figure 4-3.

4.2.1 Input Requirements

The input requirements for MPEDIT are listed below for the AN/FPS-16 editing entry, with the corresponding Verlort entry input requirements given in parentheses. Unless otherwise stated, the definitions apply to both types of radar data.

a) Initial or restart entry—MCKAPA (MCLAMA) indicates, when non-zero, that the present entry is an initial or restart editing entry for the AN/FPS-16 (Verlort) radar data; BRT8 (BRT+1) indicates, when zero, that the present entry is an initial entry to editing for the particular radar type; MCFLOC (MCVLOC) contains the location of the last range entered in the RAE table by BIHSIN: TMHSFV (TMHSFV+1) contains last location of the third point in the RAE table; BTFRAE+1 (BTVRAE+1) is the location of the first range in the RAE table.

b) Normal entry—BTFEDT (BTVEDT) is the location of a table of seven editing parameters, the first parameter contains the location of the last range received by BIHSIN. BTFCRT (BTVCRT) is the location of a table of nine editing criteria. (The parameters and criteria for editing are described under output requirements in subsection, 4.3.2). MCFLOC (MCVLOC) and TMHSFV (TMHSFV+1) are as defined above.

c) New fit entry—MCMUMU (MCNUNU), when non-zero, indicates a new fit entry to the editing processor program. All other requirements defined in b) apply to a new fit entry.

d) K00001—constant, 1.

e) Macros used—TRNOF and TRNON.

4.2.2 Output Requirements

Output from MPEDIT depends upon where MPEDIT exits to. For every entry into BEDITS, MPEDIT must provide the following:

a) MCERAD—communication cell indicating the type of radar data being edited. MCERAD contains zero for AN/FPS-16 data, a 1 for Verlort. MCERAD is not used by BEDITS, but is required for the suffixes to the editing processor.

b) BTEINP—this input word contains all of the external information necessary for BEDITS to edit the raw radar data. The address of BTEINP contains the location of the radar's editing parameters—BTFEDT for AN/FPS-16 and BTVEDT for Verlort. The decrement of BTEINP contains the location of the radar's editing criteria—BTFCRT for AN/FPS-16 and BTVCRT for Verlort. The sign of BTEINP, when minus, indicates that the radar data previously edited contained excessive bad points and that BEDITS must attempt to find a new fit. The editing parameters and criteria are defined:

Location	Contents (Bits 21-35)	Parameter Designation
BTFEDT(BTVEDT)	Location of last range to be edited	LRTBE
+1	Location of first edited range	FER
+2	Location of last edited range	LER
+3	Location of first edited azimuth	FEA
+4	Location of last edited azimuth	LEA
+5	Location of first edited elevation	FEE
+6	Location of last edited elevation	LEE

Location	Contents (Full Word)
BTFCRT(BTVCRT)	$\sqrt{1.5}$ x allowable range error
+1	$\sqrt{3}$ x allowable range error
+2	$\sqrt{5}$ x allowable range error
+3	$\sqrt{1.5}$ x allowable azimuth error
+4	$\sqrt{3}$ x allowable azimuth error
+5	$\sqrt{5}$ x allowable azimuth error
+6	$\sqrt{1.5}$ x allowable elevation error
+7	$\sqrt{3}$ x allowable elevation error
+8	$\sqrt{5}$ x allowable elevation error

An initial entry LER is set equal to FER, LEA is set equal to FEA and LEE is set equal to LEA.

c) BTFEDT (BTVEDT) is updated from MCFLOC(MCVLOC) to the location of the last range entered in the RAE table by BIHSIN. This cell is the first in the table of editing parameters and specifies the last range to be edited, the LRTBE.

d) On the initial entry the timing cell MCHEST is set to zero, the initial entry indicator, BRT8 (BRT8+1), is set non-zero, and the initial/restart entry indicator, MCKAPA (MCLAMA), is set to zero.

e) If the AN/FPS-16 radar points entered by BIHSIN are insufficient for entering the editing processor, MPEDITS stores a 1 in MCERAD to indicate Verlort radar data is now being tested for entry to editing.

4.2.3 Method

The method of MPEDIT is illustrated in the flow chart, Figure 4-3.

4.2.4 Usage

MPEDIT is initially entered from M0PRIO. If an AN/FPS-16 entry to editing can be made, MPEDIT exits to BEDITS and is re-entered from the suffix to which BEDITS exits, MFEDIT or MFEBAD. If an AN/FPS-16 entry to editing cannot be made, MPEDIT determines whether or not an editing entry can be made for Verlort. If the Verlort data is sufficient for editing, MPEDIT exits to BEDITS; if not, MPEDIT exits to MFEDIT.

a) Storage Required—64 locations

b) MPEDIT uses:

1) Macros—TRNOF and TRNON

2) Parameters—A, B, C, MNEBAD, MNEDIT and MNSMOO

3) Communication cells—MCERAD, MCMUMU, MCKAPA, MCFLOC, MCHEST, MCNUNU, MCLAMA and MCVLOC

4) Tables—BTFRAE, BTFEDT, BTFCRT, BTVRAE, BTVEDT and BTVCRT

5) Internal cell—BRT8

6) Constant—K00001

c) Time Required

1) AN/FPS-16 entry (average)—0.635 millisecond

2) Verlort entry (average)—0.384 millisecond

3) Maximum—1.15 milliseconds

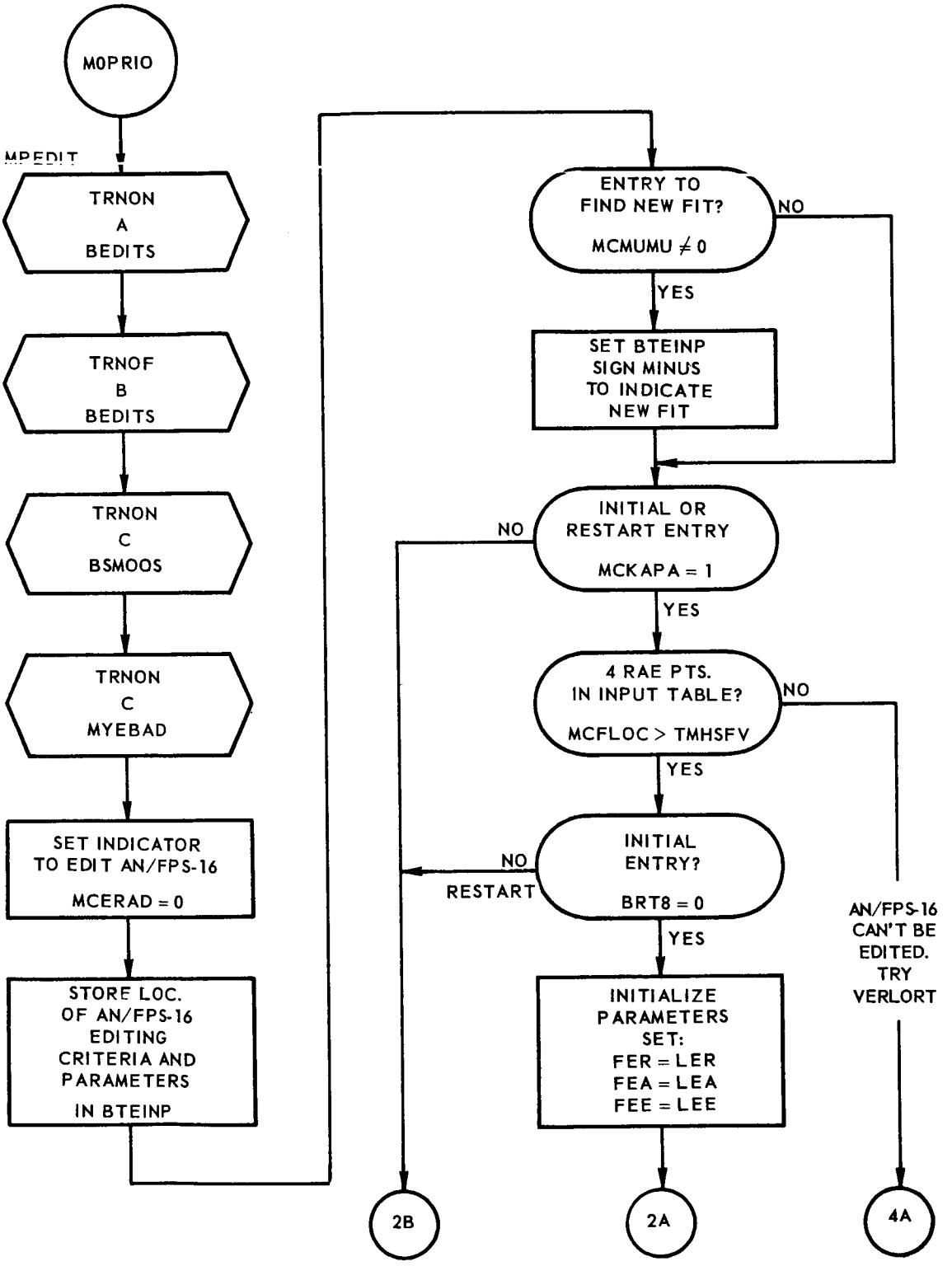


FIGURE 4-3. FLOW CHART: PREFIX AND SUFFIX TO BEDITS *(Sheet 1 of 6)*

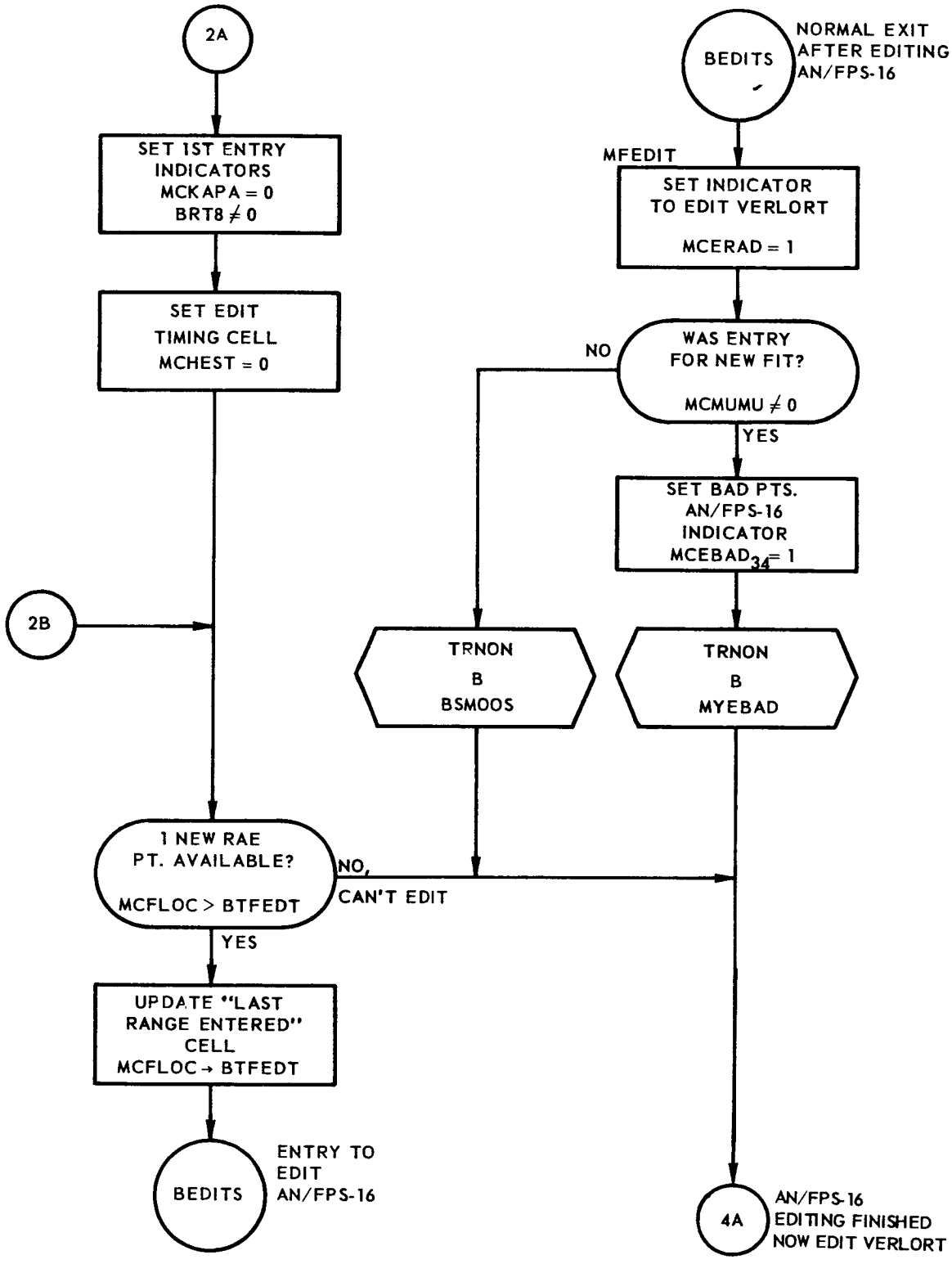


FIGURE 4-3. FLOW CHART: PREFIX AND SUFFIX TO BEDITS *(Sheet 2 of 6)*

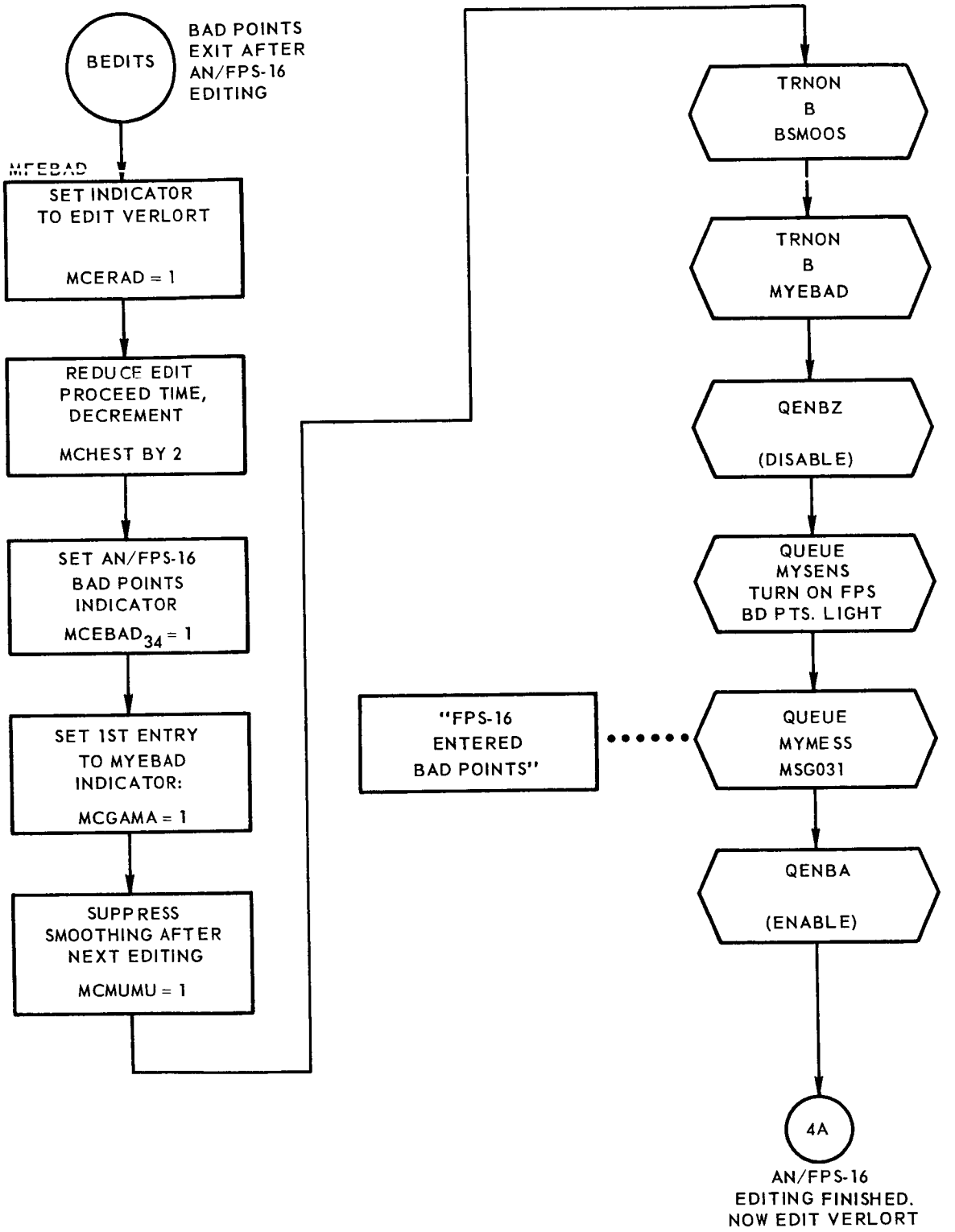


FIGURE 4-3. FLOW CHART: PREFIX AND SUFFIX TO BEDITS *(Sheet 3 of 6)*

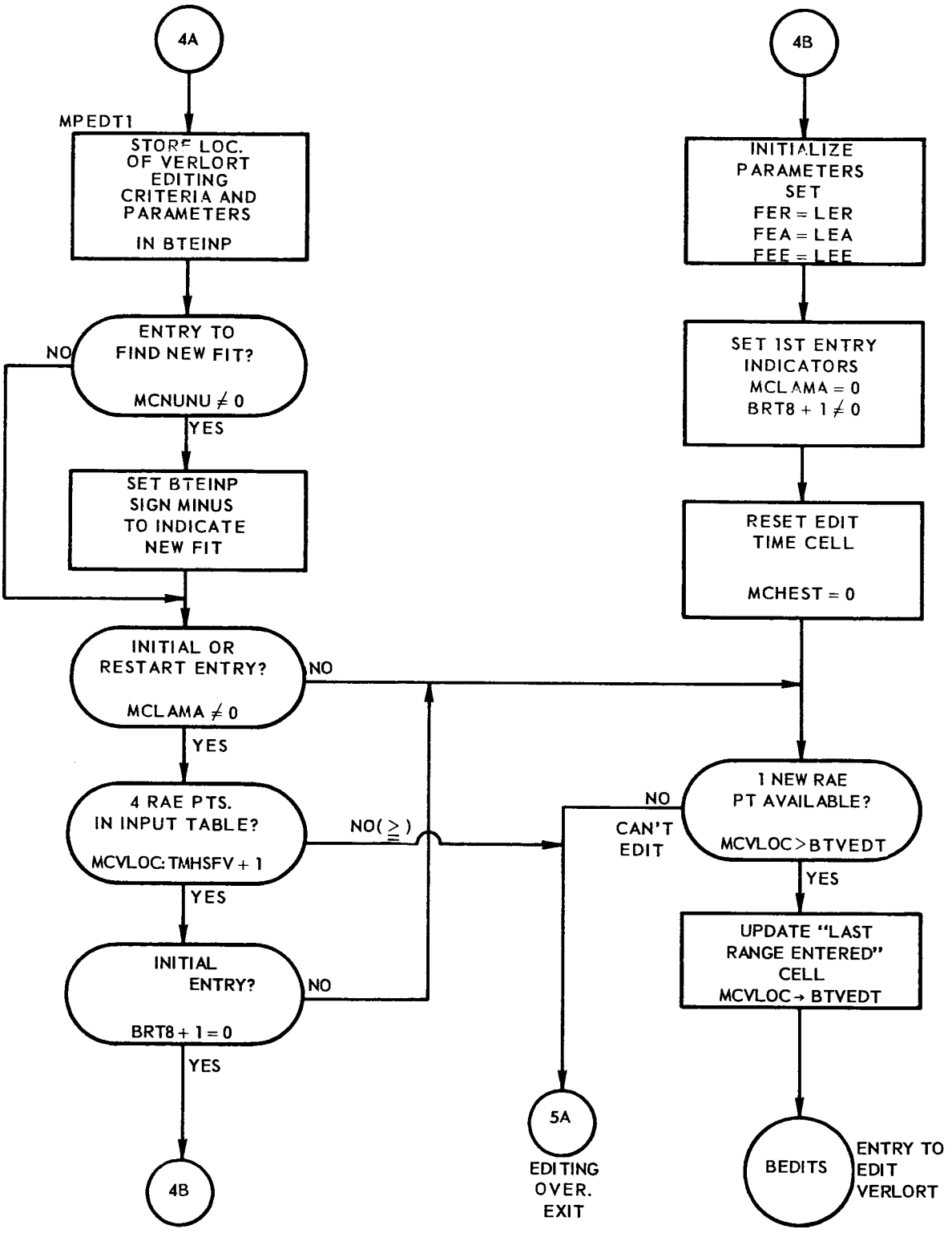


FIGURE 4-3. FLOW CHART: PREFIX AND SUFFIX TO BEDITS *(Sheet 4 of 6)*

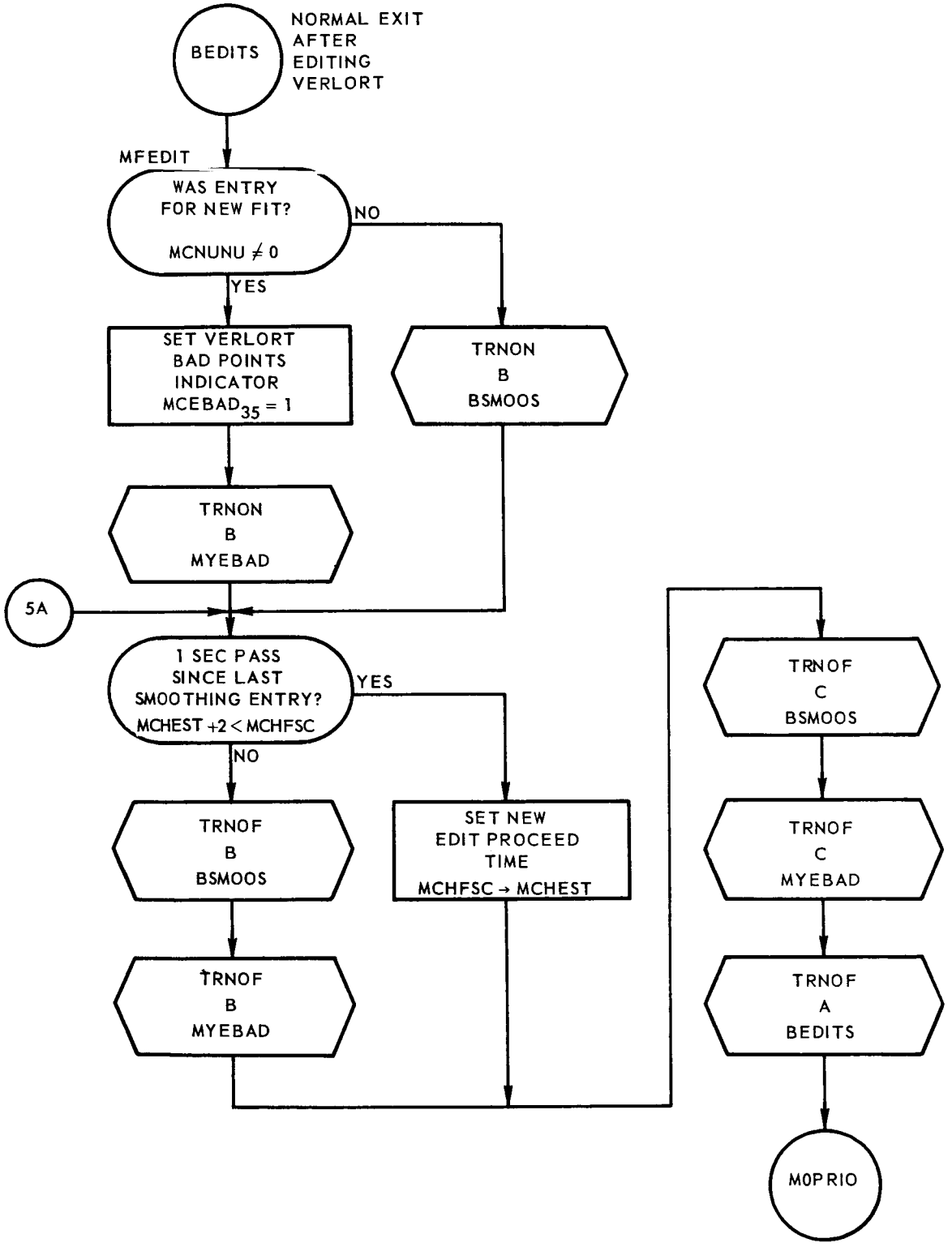


FIGURE 4-3. FLOW CHART: PREFIX AND SUFFIX TO BEDITS *(Sheet 5 of 6)*

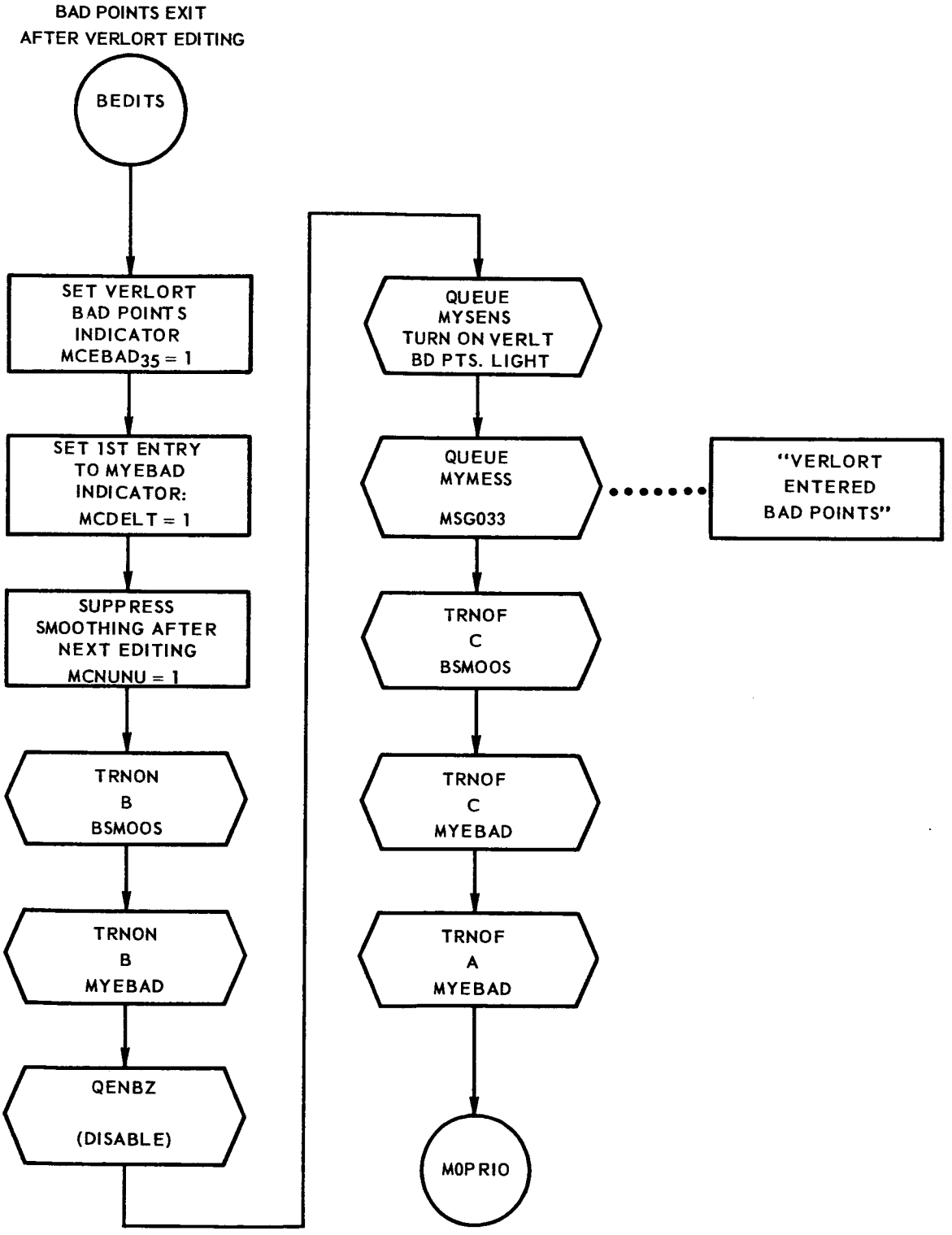


FIGURE 4-3. FLOW CHART: PREFIX AND SUFFIX TO BEDITS *(Sheet 6 of 6)*

4.3 MONITOR SUFFIX TO BEDITS (MFEDIT)

After BEDITS had completed editing the radar data, program control is given to one of BEDIT's suffixes—MFEDIT or MFEBAD. The bad points suffix, MFEBAD, is used only after the initial discovery of bad point by BEDITS and is not entered until the bad points situation has been corrected. MFEBAD, described in subsection 4.4, is entered once and only once for any bad point situation. All other editing entries while in bad points are termed "new fit" entries. The BEDITS exit from either a new fit entry or a normal entry is to MFEDIT.

If the MFEDIT entry follows AN/FPS-16 editing, controls must be set to initiate Verlort editing. In every entry MFEDIT must determine whether normal smoothing by BSMOOS is required or a bad points situation must be corrected by MYEBAD. On the final entry to MFEDIT, BSMOOS and MYEBAD must be unsuppressed.

The flow chart for the MFEDIT suffix is shown in Figure 4-3.

4.3.1 Input Requirements

Input to MFEDIT depends upon the source of the entry.

a) Entry to MFEDIT following AN/FPS-16 editing:

   1) MCERAD—communication cell contains zero to indicate AN/FPS-16 data was edited.

   2) MCMUMU—communication cell which, when non-zero, denotes that the editing entry was for a new fit (following bad points). In this case the smoothing processor is not entered with AN/FPS-16 data after editing; instead, the bad points processor is used.

b) Entry to MFEDIT following Verlort editing:

   1) MCERAD—communication cell contains a 1 to indicate Verlort data was edited.

   2) MCNUNU—communication cell which, when non-zero, denotes that the editing entry was for a new fit (following bad points). In this case the smoothing processor is not entered with Verlort data; instead, the bad points processor is used.

   3) MCHEST—communication cell which contains the fixed-point half-second time at which the last editing cycle was completed. This cell is used to determine the time interval between successive entries to the smoothing processor. If less than one second has elapsed since the last entry to the smoothing processor, the new input data is insufficient for smoothing—radar data enters at the rate of ten

points every second—and another editing cycle must precede smoothing. (The presence of bad points in AN/FPS-16 data reduces the time delay to zero).

4) MCHFSC—communication cell containing the current GMT in fixed-point half seconds.

c) Entry to MFEDIT from MPEDIT when the Verlort data is insufficient for entry to BEDITS.

1) MCHEST—as defined above.

2) MCHFSC—as defined above.

d) K00001—constant, 1.

e) K00002—constant, 2.

f) Macros used—TRNOF and TRNON.

4.3.2 Output Requirements

Output from MFEDIT depends upon the source of the entry and upon type of editing performed.

a) Entry to MFEDIT following AN/FPS-16 editing:

1) MCERAD—a 1 is stored in MCERAD to indicate a Verlort entry for the next entry to BEDITS.

2) If MCMUMU contains a 1, the B indicator for MYEBAD is turned on and a 1 is stored in bit position 34 of MCEBAD. If MCMUMU contains zero, the B indicator for BSMOOS is turned on.

3) Control transfers to MPEDIT to prepare for Verlort editing.

b) Entry to MFEDIT following Verlort editing:

1) If MCNUNU contains a 1, the B indicator for MYEBAD is turned on and a 1 is stored in bit position 35 of MCEBAD. If MCNUNU contains zero, the B indicator for BSMOOS is turned on.

2) If one second has not passed since the last entry to smoothing (MCHFSC < MCHEST+2), the smoothing and bad points processors cannot be entered and the B indicators for BSMOOS and MYEBAD are turned off. If at least one second has passed, the current half-second time is used to update the "edit proceed" time cell, MCHEST.

3) The C indicators for BSMOOS and MYEBAD are turned off.

4) The A indicator for BEDITS is turned off and program control returns to M0PRIO.

c) Entry to MFEDIT when the Verlort data is insufficient for editing—output requirements 2) through 5) under b) are applicable.

4.3.3 Method

The method of MFEDIT is illustrated in the flow chart, Figure 4-3.

4.3.4 Usage

MFEDIT is entered from BEDITS, or from MPEDIT if Verlort radar data is insufficient for editing. When MFEDIT is entered from BEDITS following AN/FPS-16 editing, MFEDIT exits to MPEDIT. For any other entry, MFEDIT exits to M0PRIO.

a) Storage Required—44 locations

b) MFEDIT uses:

1) Macros—TRNOF and TRNON

2) Parameters—A, B, C, MNEBAD, MNEDIT and MNSMOO

3) Communication cells—MCEBAD, MCERAD, MCHEST, MCHFSC, MCMUMU and MCNUNU

4) Constants—K00001, K00002

c) Time Required:

1) AN/FPS-16 entry—0.334 millisecond

2) Verlort entry—0.672 millisecond

4.4 MONITOR BAD POINTS SUFFIX TO BEDITS (MFEBAD)

When, in editing the raw radar points, BEDITS discovers an excessive number of non-linear points, BEDITS exits to its bad points suffix, MFEBAD. This suffix sets controls for the bad points corrective procedure. If the bad points were found in AN/FPS-16 editing, controls must also be established to edit the Verlort data. If the bad points were found in Verlort editing, MFEBAD controls are set for the bad points correction procedure and controls are also set which terminate editing until new radar data has arrived.

The flow chart for the MFEBAD suffix is shown in Figure 4-3.

4.4.1 Input Requirements

Input to MFEBAD consists of:

a) MCERAD—communication cell denoting the radar data currently being edited and also currently in bad points: when MCERAD contains zero, AN/FPS-16 is in bad points; when MCERAD contains a 1, Verlort is in bad points.

b) MCHEST—communication cell containing the time at which the last editing was completed. (This cell is affected only when AN/FPS-16 is in bad points.)

c) CJT11, CJT12—masks used to turn on two of the Output Status Console's lights which indicate AN/FPS-16 bad points and Verlort bad points.

d) K00001—constant, 1.

e) Macro Used—QENBA, QENBZ, QUEUE, TRNOF and TRNON.

4.4.2 Output Requirements

Output from MFEBAD depends upon the radar in bad points.

a) AN/FPS-16 in bad points:

1) MCERAD—communication cell set to 1 to indicate Verlort data still remains to be edited.

2) MCHEST—the contents of the "edit proceed" time cell are decremented by two, insuring no delay in entering the smoothing processor after the editing processor has completed editing the current radar data.

3) MCEBAD—a 1 is stored in bit position 34 of MCEBAD to indicate AN/FPS-16 has entered bad points.

4) MCGAMA—a 1 is stored in MCGAMA to indicate to MYEBAD the initial, or bad points, entry to the Monitor Bad Points Processor for this radar. After MYEBAD has set the parameters for the later correction of bad points, MCGAMA is reset to zero.

5) MCMUMU—a 1 is stored in MCMUMU to indicate to MPEDIT that a new edit fit is needed and that AN/FPS-16 smoothing must be suppressed after the next AN/FPS-16 editing entry and must remain suppressed until after the bad points situation has been corrected.

6) The B indicators for BSMOOS and MYEBAD are turned on.

7) The on-line message "FPS-16 ENTERED BAD POINTS" is queued.

8) The input mask is queued for the sense output processor to turn on the "BAD POINTS FPS-16" indicator light.

9) Control returns to MPEDIT to prepare for the Verlort editing entry to BEDITS.

b) Verlort in bad points:

1) MCEBAD—a 1 is stored in bit position 34 to indicate Verlort has entered bad points.

2) MCDELT—a 1 is stored in MCDELT to indicate to MYEBAD the initial, or bad points, entry to the Monitor Bad Points Processor for this radar. After MYEBAD has set the parameters for the later correction of bad points, MCDELT is reset to zero.

3) MCNUNU—a 1 is stored in MCNUNU to indicate to MPEDIT that a new edit fit is needed and that Verlort smoothing must be suppressed after the next Verlort editing entry and must remain suppressed until the bad points situation has been corrected.

4) The B indicators for BSMOOS and MYEBAD are turned on.

5) The on-line message "VERLORT ENTERED BAD POINTS" is queued.

6) The input mask is queued for the sense output processor to turn on the "BAD POINTS VERLORT" indicator light.

7) The exit to MFEBAD after editing Verlort data completes the current editing cycle. Therefore, MFEBAD unsuppresses BSMOOS and MYEBAD by turning off their C indicators. MFEBAD turns off the A indicator for BEDITS and returns control to M0PRIO.

4.4.3 Method

The method of MFEBAD is illustrated in the flow chart, Figure 4-3.

4.4.4 Usage

MFEBAD is entered from BEDITS. If the entry followed AN/FPS-16 editing, MFEBAD exits to MPEDIT. If the entry followed Verlort editing, MFEBAD exits to M0PRIO.

a) Storage Required—57 locations

b) MFEBAD Uses:

1) Macros—QENBA, QENBZ, QUEUE, TRNOF and TRNON.

2) Parameters—A, B, C, MNEBAD, MNEDIT, MNMESS, MNSENS and MNSMOO

3) Communication cells—MCEBAD, MCERAD, MCDELT, MCGAMA, MCHEST, MCMUMU and MCNUNU

4) Masks—CJT11, CJT12

5) Constant—K00001

c) Time Required:

1) AN/FPS-16 entry—1.55 milliseconds

2) Verlort entry—1.64 milliseconds

4.5 MONITOR PREFIX TO BSMOOS (MPSMOO)

If, at the completion of editing, sufficient new radar points have been entered and edited the Monitor Prefix to BSMOOS is entered to prepare the parameters for the smoothing processor.

The flow chart for the MPSMOO prefix is shown in Figure 4-4.

4.5.1 Input Requirements

Input to MPSMOO consists of:

a) BTT2—cell internal to MPSMOO which initially contains a 1 to indicate MPSMOO has not yet been entered.

b) BTT4 (BTT4+1)—internal cell which, when non-zero, indicates the initial entry for AN/FPS-16 (Verlort) editing data.

c) MCPIPI (MCRHOO)—communication cell which, when non-zero, indicates BEDITS and MYEBAD are attempting to find a new fit after AN/FPS-16 (Verlort) bad points. BSMOOS cannot be entered until the new fit is completed.

d) BTFSMO (BTVSMO)—contains the location of the AN/FPS-16 (Verlort) editing parameters. If smoothing may be used for AN/FPS-16 (Verlort) radar data, the contents of BTFSMO (BTVSMO) are stored in BTSMOO. The input requirements for other smoothing parameters set initially, and only once, by MPSMOO are given in Table 4-1.

e) K00001—constant, 1.

f) Macros Used—TRNOF and TRNON.

4.5.2 Output Requirements

Output from MPSMOO consists of:

a) BTT2—internal cell set to zero on the initial entry only.

b) BT2NST—on the initial entry only a 1 is stored in BT2NST to indicate 2N +1 smoothing requested. BT2NST is set only on the first entry to MPSMOO.

c) BTSECO—communication cell which, when non-zero, indicates SECO time has been computed by BSMOOS. This cell is set to zero on the initial entry to MPSMOO.

d) BTT4 (BTT4+1)—internal cell set to zero on the initial AN/FPS-16 (Verlort) smoothing entry. This cell is set only once.

e) BTSMOO—the location of the table of editing parameters, BTFSMO (BTVSMO), is stored in BTSMOO for every AN/FPS-16 (Verlort) entry to BSMOOS.

f) MCSRAD—communication cell set to indicate type of radar data being smoothed. A 1 in MCSRAD indicates Verlort; zero indicates AN/FPS-16.

Other output from MPSMOO consists of smoothing parameters set initially, and only once, for each radar. These parameters are described in Table 4-1.

4.5.3 Method

The method for MPSMOO is shown in Figure 4-4.

4.5.4 Usage

MPSMOO is entered from M0PRIO or MFSMOO. MPSMOO exits to BSMOOS or to MFSMOO.

a) Storage Required—78 locations

b) MPSMOO Uses:

1) Macros—TRNOF and TRNON

2) Parameters—A, B and MNSMOO

3) Communication cells—MCPIPI, MCRHOO and MCPIPI

4) Tables—BT2NST, BTF0AM, BTF0CA, BTF07P, BTF11P, BTFFEN, BTFLAR, BTFLSI, BTFLSR, BTFRS7, BTFSET, BTFSMO, BTFSR7, BTSECO, BTSMOO, BTV0AM, BTV0CA, BTV07P, BTVIIP, BTVFEN, BTVLAR, BTVLSI, BTVLSR, BTVRS7, BTVSET, BTVSMO and BTVSR7

5) Constant—K00001

6) Internal cells—BTT2, BTT4 and BTT4+1

c) Time Required:

1) AN/FPS-16 entry (average)—0.38 millisecond

2) AN/FPS-16 entry (maximum)—1.72 milliseconds

3) Verlort entry (average)—0.13 millisecond

4) Verlort entry (maximum)—1.46 milliseconds

TABLE 4-1. SMOOTHING PARAMETERS AND CRITERIA INITIALLY SET BY MPSMOO

Parameter Designation	Explanation	AN/FPS-16		VERLORT	
		Symbolic Location	Contents Set by MPSMOO	Symbolic Location	Contents Set by MPSMOO
SMO*	Contains the location of the table of editing parameters, BTFEDT or BTVEDT	BTSMOO	BTFSMO	BTSMOO	BTVSMO
LSR	Contains the location of the last 11-point smoothed range	BTFLSR	BTF11P-2	BTVLSR	BTVIIP-2
LRS7	Contains the location of the last range smoothed, 11 point to 7 point	BTFRS7	BTFIIP-8	BTVRS7	BTVIIP-8
LSR7	Contains the location of the last 7-point smoothed range.	BTFSR7	BTF07P-2	BTVSR7	BTF07P-2
SET UP	Contains a 1 in bit position 35 if 2N+1 smoothing set up.	BTFSET	0	BTVSET	0
C_a	Contains the value of C_a for the BSMOOS SECO determination program.	BTF0CA	0	BTV0CA	0
A_m	Contains the value of maximum acceleration for the BSSECO program.	BTF0AM	0	BTV0AM	0
FEN	Contains a 1 in bit position 35 if BSMOOS has been entered previously for this radar.	BTFFEN	0	BTVFEN	0
LAR	Contains the location of the last 11-point smoothed range used for Short Arc.	BTFLAR	BTFIIP-2	BTVLAR	BTFIIP-2
LSI	Contains a 1 in bit position 35 when a new 2N+1 position/velocity solution obtained.	BTFLSI	0	BTVLSI	0

*SMO is set for each entry to BSMOOS

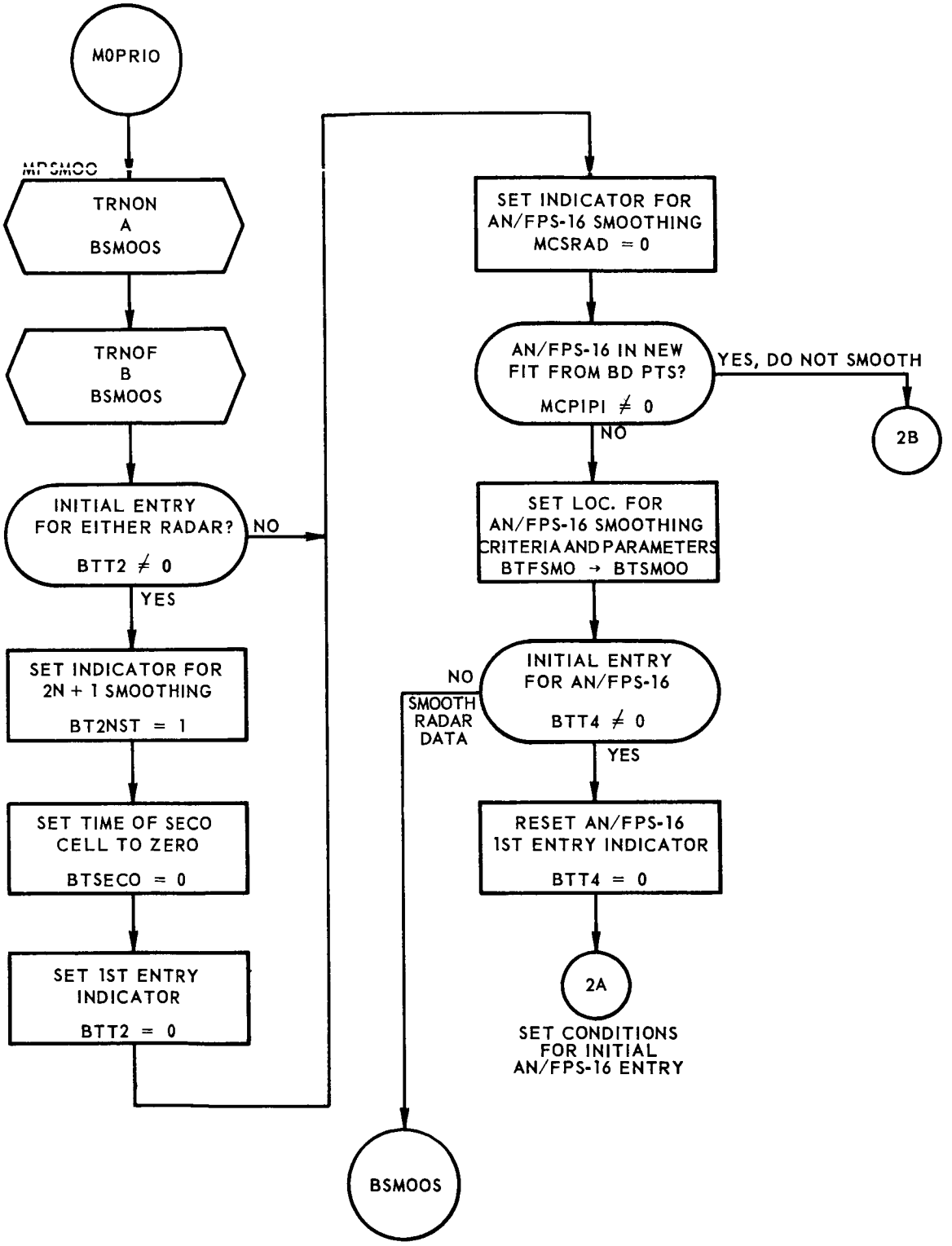


FIGURE 4-4. FLOW CHART: PREFIX AND SUFFIX TO BSMOOS *(Sheet 1 of 4)*

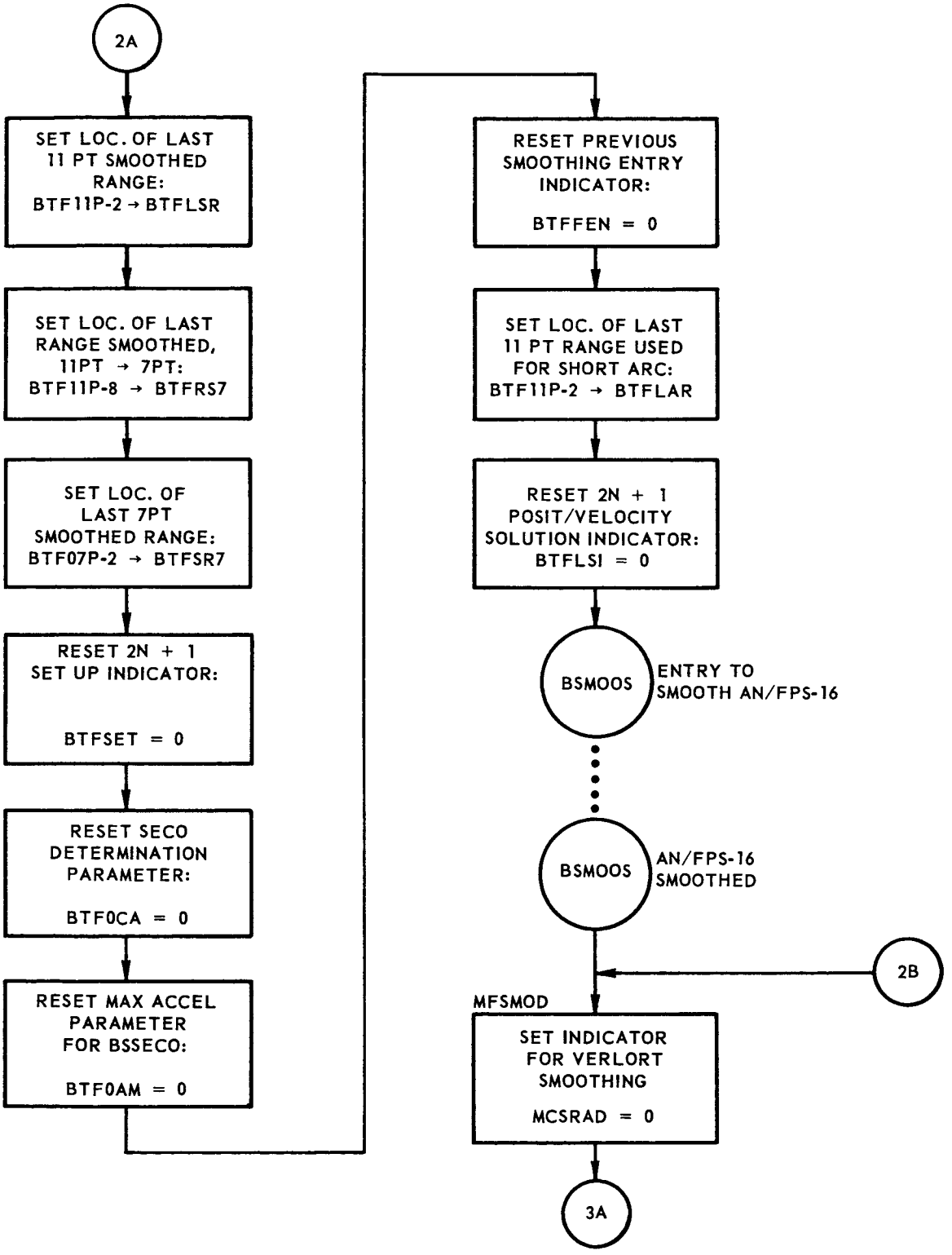


FIGURE 4-4. FLOW CHART: PREFIX AND SUFFIX TO BSMOOS *(Sheet 2 of 4)*

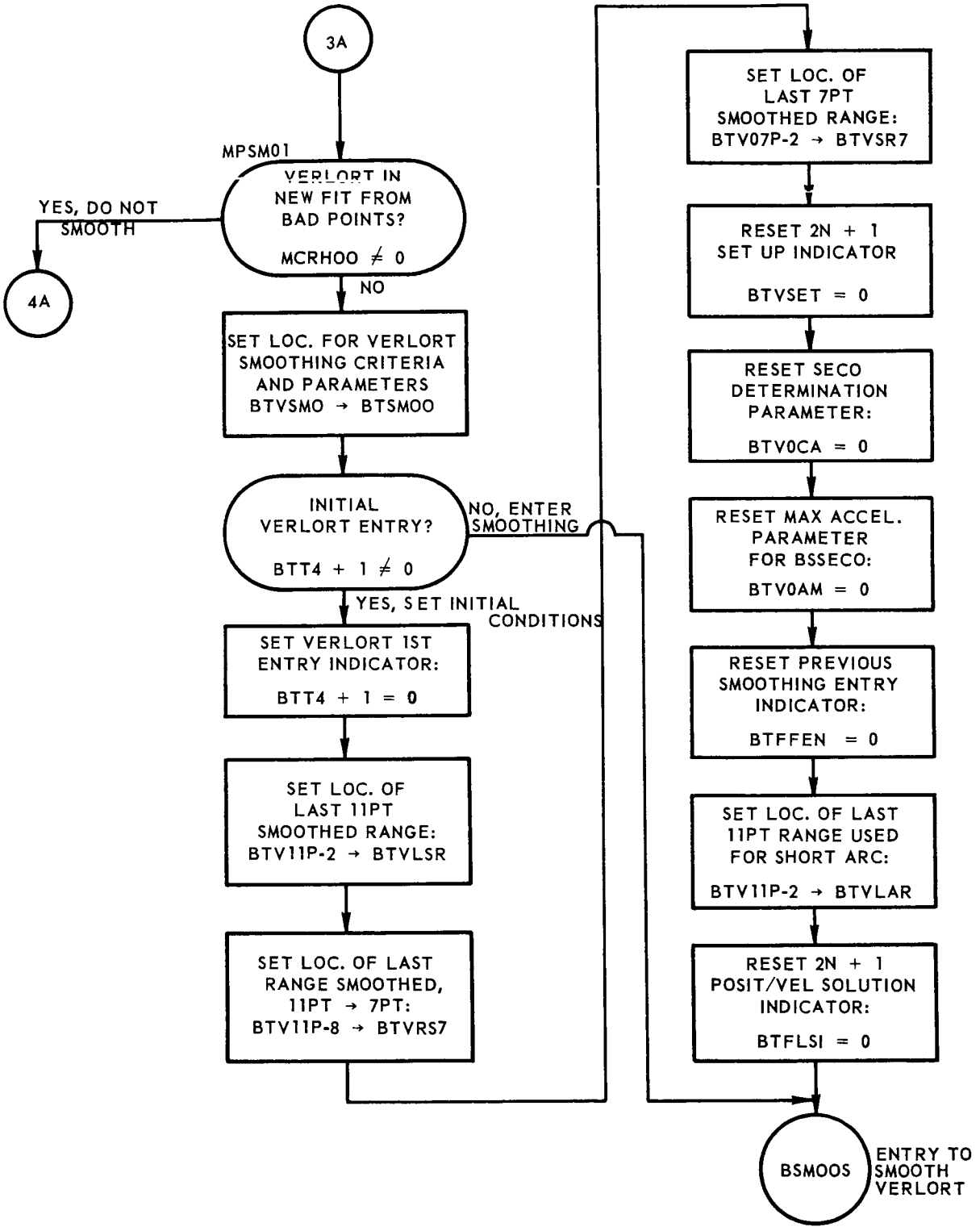


FIGURE 4-4. FLOW CHART: PREFIX AND SUFFIX TO BSMOOS *(Sheet 3 of 4)*

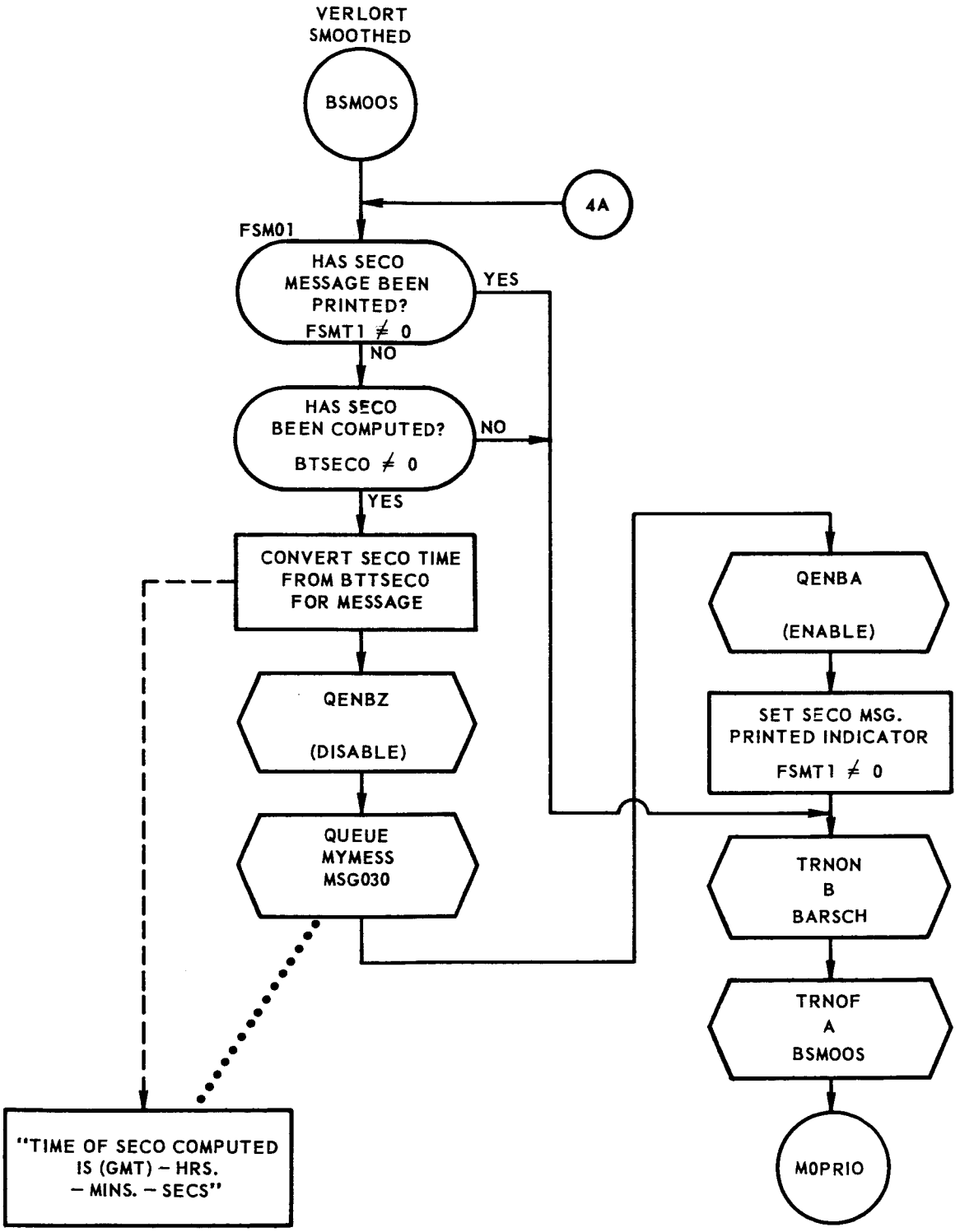


FIGURE 4-4. FLOW CHART: PREFIX AND SUFFIX TO BSMOOS *(Sheet 4 of 4)*

4.6 MONITOR SUFFIX TO BSMOOS (MFSMOO)

MFSMOO receives control, either indirectly via BSMOOS or directly from MPSMOO, after each radar entry to MPSMOO. When MFSMOO receives control following the AN/FPS-16 entry, MFSMOO returns control to the prefix for the Verlort entry. Following the Verlort entry, MFSMOO determines whether or not BSMOOS obtained a time for SECO. If a SECO time is available, MFSMOO provides an on-line message giving the computed time.

The flow chart for MFSMOO is shown in Figure 4-4.

4.6.1 Input Requirements

Input to MFSMOO consists of:

a) MCSRAD—communication cell which, when zero, indicates Verlort radar data remains to be smoothed. MCSRAD, when non-zero, indicates the final entry to MFSMOO for this smoothing pass.

b) FSMT1—internal cell which, when non-zero, indicates the SECO message has been printed.

c) BTSECO—one-cell table which, when non-zero, indicates the smoothing program has computed the time of SECO.

d) BTTSEC—one-cell table which, when non-zero, contains the time of SECO computed by BSMOOS. The time in BTTSEC is in fixed-point tenths of seconds.

e) K00001—constant, 1.

f) K00010—constant, 10.

g) Macros used—QENBA, QENBZ, QUEUE, TRNOF and TRNON.

4.6.2 Output Requirements

Output from MFSMOO consists of:

a) MCSRAD—after the AN/FPS-16 entry, a 1 is stored in MCSRAD to indicate the Verlort entry. When MFSMOO is entered following AN/FPS-16 smoothing, MCSRAD is the only output requirement and control returns to MPSMOO for the Verlort smoothing entry.

b) If, following the Verlort smoothing entry, MFSMOO finds that SECO has been computed, but that the SECO message has not yet been printed, the computed time of SECO is converted into fixed-point seconds and stored as the field in the queued on-line message "TIME OF

SECO COMPUTE IS (GMT)___HRS.___MINS.___SECS." The SECO message indicator cell, FSMT1, is set non-zero.

4.6.3 Method

The method of MFSMOO is illustrated in Figure 4-4.

4.6.4 Usage

MFSMOO is entered from MPSMOO or BSMOOS; MFSMOO exits to MPSMOO or M0PRIO.

a) Storage Required—29 locations

b) MFSMOO uses:

1) Macros—QENBA, QENBZ, QUEUE, TRNOF and TRNON

2) Parameters—A, B, MNMESS, MNSARC and MNSMOO

3) Communication cell—MCSRAD

4) Tables—BTSECO and BTTSEC

5) Internal cell—FSMT1

c) Time Required:

1) Entry after AN/FPS-16 smoothed—0.14 millisecond

2) Entry after Verlort smoothed (average)—0.25 millisecond

3) Entry after Verlort smoothed (maximum)—1.21 milliseconds

4.7 MONITOR BAD POINTS PROCESSOR (MYEBAD)

When the editing processor finds an excessive number of bad points from either radar, the editing processor exits to its Monitor Suffix for bad points, MFEBAD. This suffix sets controls for the bad points correction procedure.

At least two separate entries to MYEBAD are required to correct a bad points situation. On the first, or bad points, entry MYEBAD must adjust the editing parameters and set controls to allow the editing processor to find a new fit, i.e., a new set of colinear points at a projected time ahead of the bad points. MYEBAD must also suppress smoothing for the radar involved until the bad points have been corrected. On successive, or new fit, entries MYEBAD determines whether or not a new fit has been established by BEDITS. After a new fit is available, MYEBAD bridges, if possible, the gap between the two good fits—the good fit before bad points and the new fit found by BEDITS—with an artificial smoothing program. If MYEBAD cannot suffessfully fill in the gap caused by bad points, a non-data mask is used to complete the tables. When MYEBAD completes filling in the needed values, the smoothing parameters are adjusted to conform to the changes made by MYEBAD.

The flow chart for the MYEBAD processor is shown in Figure 4-5.

4.7.1 Input Requirements

The input requirements for the AN/FPS-16 entry to MYEBAD are listed below, with the corresponding input requirements for the Verlort entry given in parentheses. Unless otherwise stated, the definitions apply to both types of radar data. When only one requirement is listed for a given definition, this input is shared by both radar entries.

a) Bad points entry:

1) MCEBAD—communication cell indicating the radar involved in bad points. A 1 in bit position 34 indicates AN/FPS-16 in bad points; a 1 in bit position 35 indicates Verlort in bad points.

2) MCGAMA (MCDELT)—communication cell which, when non-zero, indicates the bad points, or initial, entry to MYEBAD for AN/FPS-16 (Verlort).

3) BTF0LR (BTV0LR)—one-cell table containing the location of the last completely edited point when the smoothing processor was last entered for this radar. The contents of this location is designated by the parameter LR.

4) BTMNRP—one-cell table whose contents define the maximum number of points BEDITS may reject before a new fit is required. When MYEBAD is entered, BEDITS has already rejected the number of

points specified by BTMNRP. MYEBAD uses BTMNRP to reset the editing parameters for the new fit entry to BEDITS. BTMNRP is designated by the parameter M. (BTMNRP actually represents three times the number of rejected points, since each point occupies three locations in core storage.)

b) New fit entry:

1) BTFEDT (BTVEDT)—a seven-cell table containing the editing parameters for AN/FPS-16 (Verlort). If the locations of the first and last edited range or azimuth, or elevation are identical, BEDITS has not yet found a new fit and MYEBAD cannot use its artificial smoothing routine until more data enters and/or is edited. (The editing parameters are described in subsection 4.2.2.)

2) If a new fit has been obtained by BEDITS, MYEBAD uses the editing parameters to determine GLR and GFR. GLR is the location of the range of last completely edited point after the bad data gap and GFR is the location the range of the first completely edited point after the bad data gap. If GLR and GFR do not differ by 12 locations (4 points), MYEBAD must await further edited points before using the artificial smoothing routine.

2) BTFFEN (BTVFEN)—one-cell table which, when non-zero, indicates the smoothing processor BSMOOS has been entered for AN/FPS-16 (Verlort) data at least one time. If BSMOOS has not been entered, the initial bad points are bypassed, i.e., bad points which occur before BSMOOS has smoothed good points, are not artifically reconstructed.

3) BFFLRS (BTVLRS)—one-cell table containing the location of the last AN/FPS-16 (Verlort) range smoothed, raw RAE to 11 point; this range location is designated by the parameter LRS.

4) BTFLSR (BTVLSR)—one-cell table containing the location of the last AN/FPS-16 (Verlort) 11-point smoothed range. This range location is designated by the parameter LSR.

5) BTF00N (BTV00N)—one-cell table containing the value of N for 2N+1 smoothing. This value has the parameter designation N.

6) BTF11P (BTV11P)—the location of the 11-point smoothed table. This location has the parameter designation 11PT.

7) BTF11T (BTV11T)—one-cell table containing the location of time in the 11-point smoothed table. This time has the parameter designation T11.

8) BTFLAR (BTVLAR)—one-cell table containing the location of the last 11-point smoothed range used by the short arc processor. This location has the parameter designation LAR.

9) MCERAD—as defined above.

10) BZT27 (BZT28)—location containing the mask used to turn off the AN/FPS-16 (Verlort) bad points light on the Output Status Console.

c) K00001—constant, 1.

d) K00002—constant, 2.

e) K00003—constant, 3.

f) K00009—constant, 9 (decimal).

g) K00012—constant, 12, (decimal).

h) K00015—constant, 12, (decimal).

i) K00030—constant, 30, (decimal).

j) K00090—constant, 90, (decimal).

k) K00178—constant, 178, (decimal).

l) K00300—constant, 300, (decimal).

m) Macros Used—QENBA, QENBZ, QUEUE, TRNOF and TRNON.

4.9.2 Output Requirements

The output requirements from MYEBAD are contingent upon the type of entry, the type of radar, and the condition of several indicators.

a) Bad points entry:

1) BTVEDT (BTVEDT)—the editing parameters are reset for BEDITS to find a new fit. FER and LER are set to LR+M-9, FEA and LEA are set to LR+M-8, and FEE and LEE are set to LR+M-7.

2) MGAMA(MCDELT)—the indicator for bad points entry to MYEBAD is reset to zero.

3) MCPIPI(MCRHOO)—the suppress smoothing indicator is set to non-zero. This suppresses smoothing for AN/FPS-16 (Verlort).

4) MCEBAD—bit position 34 is set zero on AN/FPS-16 entry; bit position 35 is set zero on Verlort entry.

b) New fit entry when a new edit fit is not available produces no output except MCEBAD (as listed above).

c) New fit entry when GLR/GFR is less than 12 produces no output except MCEBAD (as listed above).

d) New fit entry where GLR-GFR is greater than, or equal to, 12 and where BSMOOS has previously been entered produces the artificial smoothing as shown in Figure 4-5. If BSMOOS had not been entered, the artificial smoothing is bypassed. Additional output consists of:

1) MCMUMU(MCNUNU)—the edit bad points indicator is reset to zero.

2) The sense output processor is queued with a mask to turn off the "FPS-16 BAD POINTS" ("VERLORT BAD POINTS") light.

3) The on-line message processor is queued with input to produce the message "FPS-16 COMPLETED BAD POINTS" ("VERLORT COMPLETED BAD POINTS").

For every entry MYEBAD turns its A indicator on and B indicator off at entry; MYEBAD turns its A indicator off at exit.

4.7.3 Method

The method of MYEBAD is shown in Figure 4-5.

4.7.4 Usage

MYEBAD is entered from, and exits to, M0PRIO.

a) Storage Required—244 locations

b) MYEBAD Uses:

1) Macros—QENBA, QENBZ, QUEUE, TRNOF and TRNON

2) Parameters—A, B, MNEBAD, MNMESS and MNMESS

3) Communication cells—MCEBAD, MCDELT, MCGAMA, MCMUMU, MCNUNU, MCPIPI and MCRHOO

4) Tables—BTF00N, BTF0LR, BTF11P, BTFEDT, BTFFEN, BTFLAR, BTFLRN, BTFLRS, BTFLSR, BTFRS7, BTFSET, BTMNRP,

BTV00N, BTV0LR, BTV11P, BTVEDT, BTVFEN, BTVLAR, BTVLRN, BTVLRS, BTVLSR, BTVRS7 and BTVSET

5) Masks—BZT27 and BZT28

6) Constants—K00001, K00002, K00003, K00009, K00012, K00015, K00030, K00090, K00178 and K00300

c) Time Required:

1) Bad points entry for one radar—1.03 milliseconds

2) Bad points entry for both radars—1.80 milliseconds

3) New fit entry for one radar (maximum)—3.32 milliseconds

4) New fit entry for both radars (maximum)—5.98 milliseconds

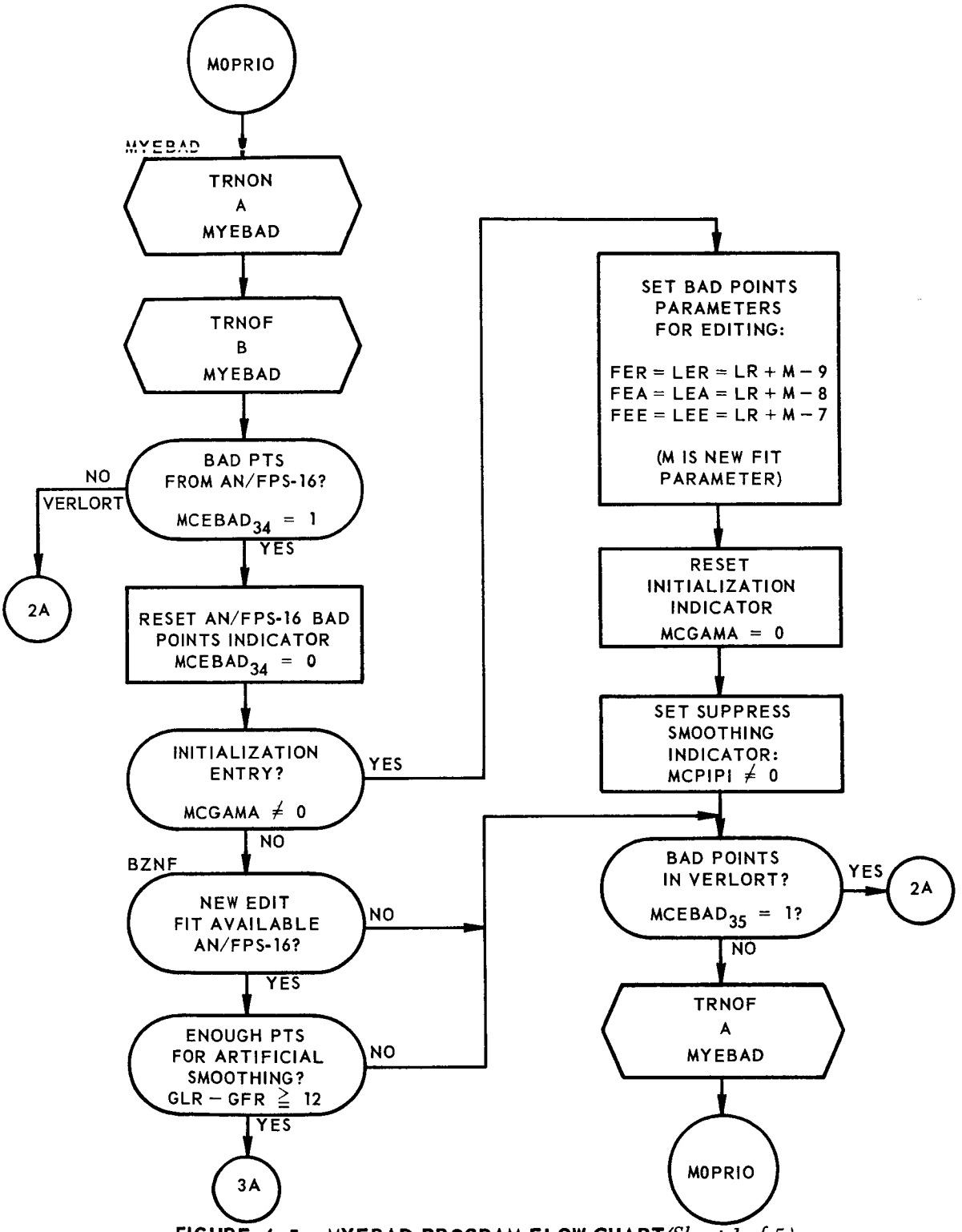


FIGURE 4-5. MYEBAD PROGRAM FLOW CHART *(Sheet 1 of 5)*

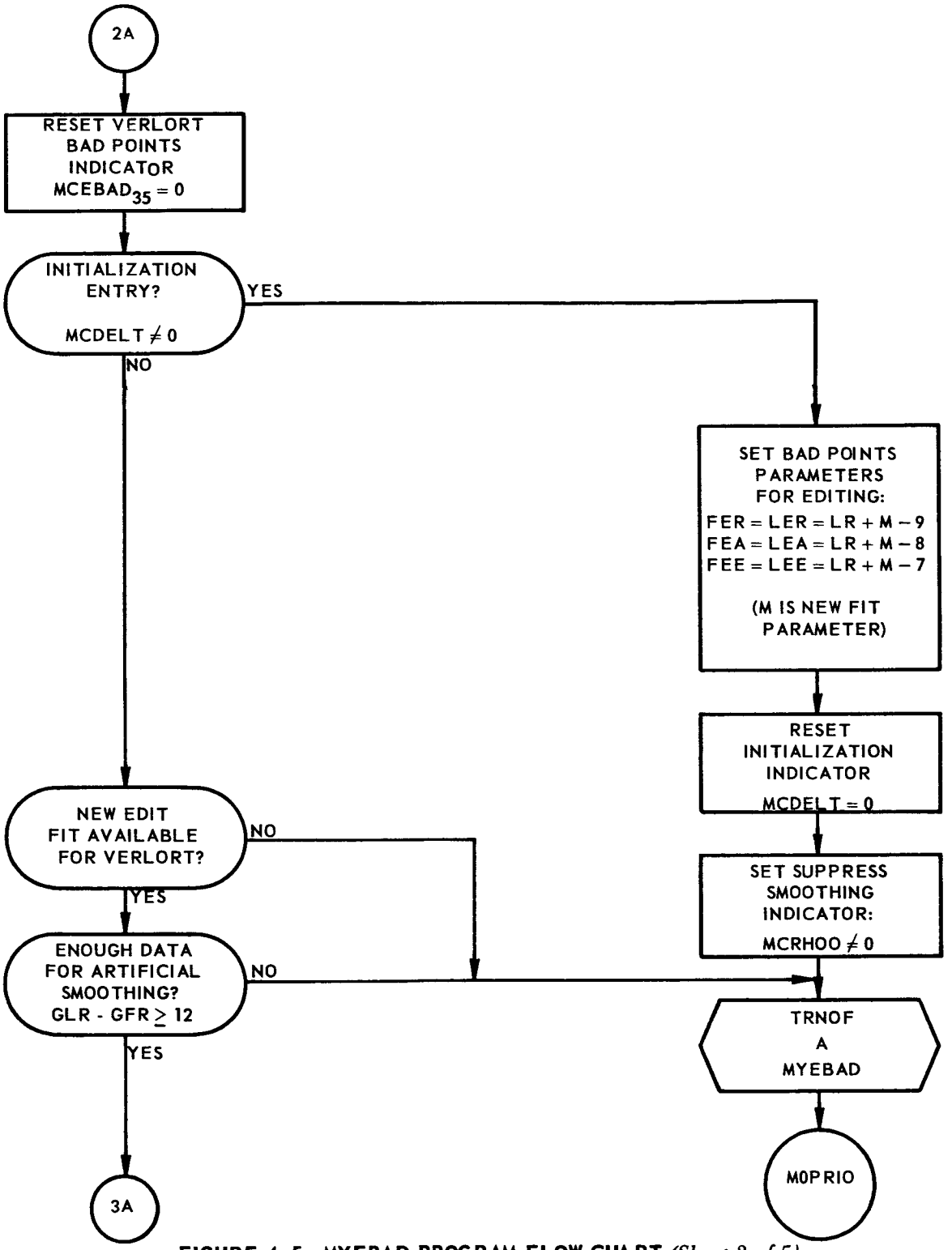


FIGURE 4-5. MYEBAD PROGRAM FLOW CHART *(Sheet 2 of 5)*

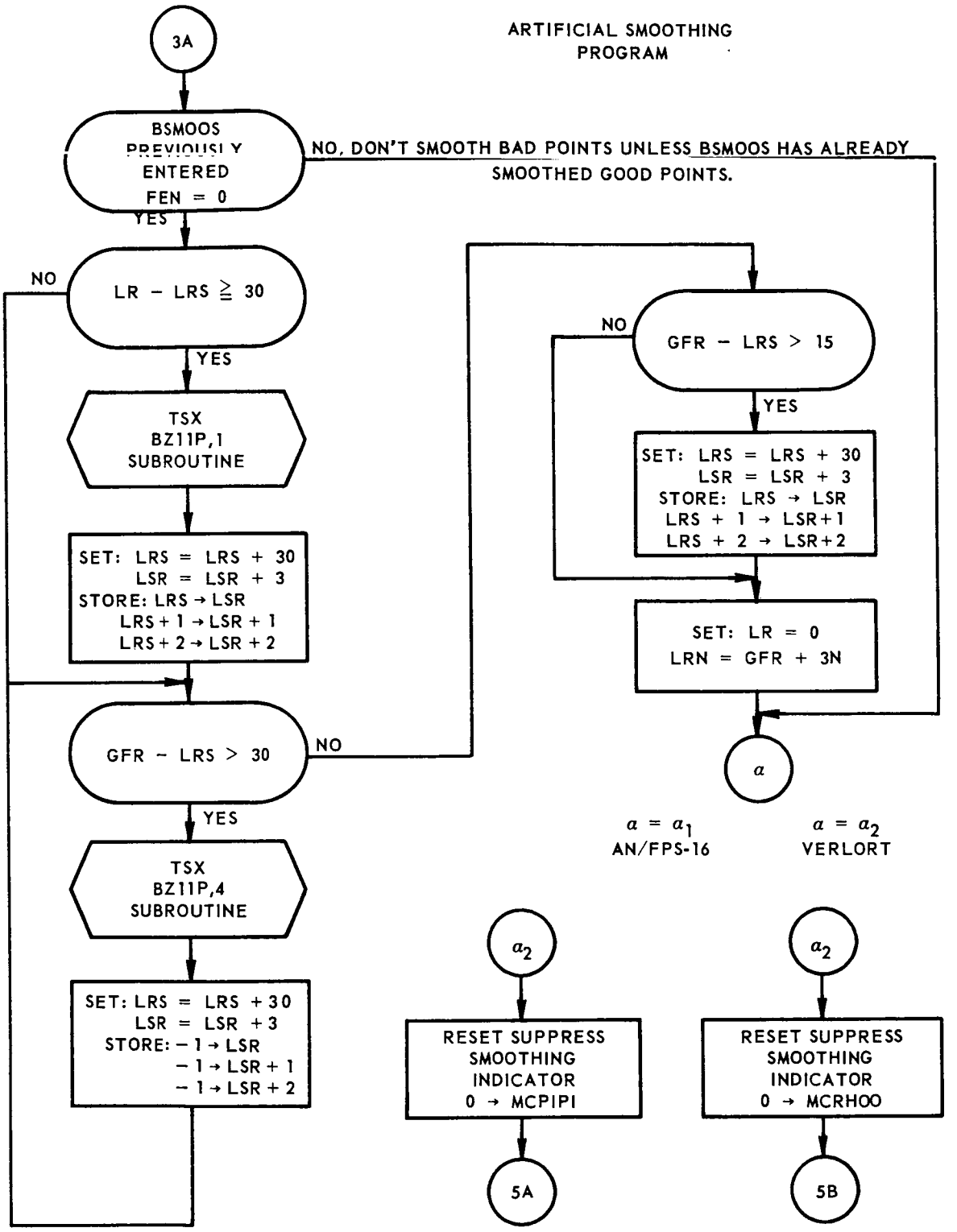


FIGURE 4-5. MYEBAD PROGRAM FLOW CHART *(Sheet 3 of 5)*

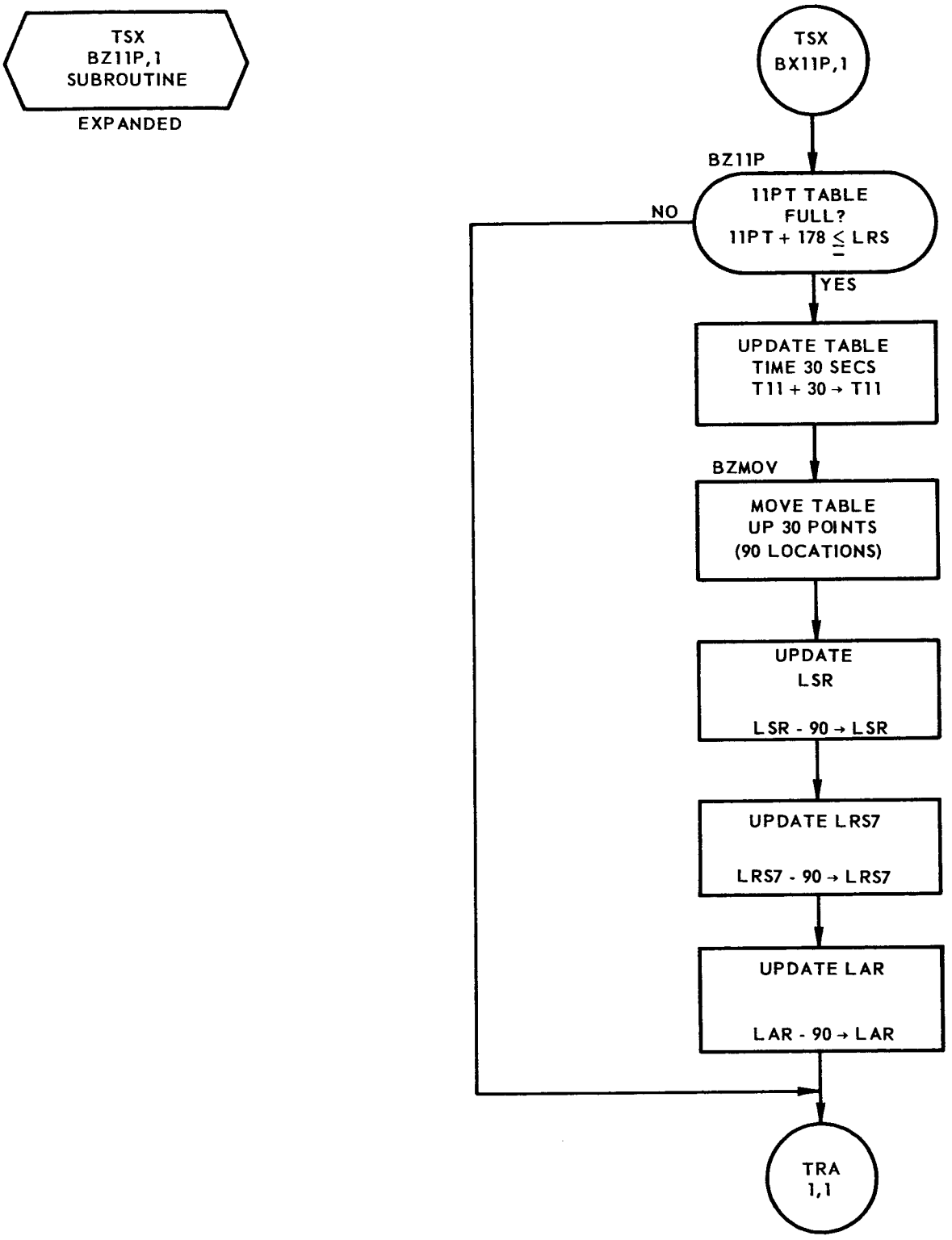


FIGURE 4-5. MYEBAD PROGRAM FLOW CHART *(Sheet 4 of 5)*

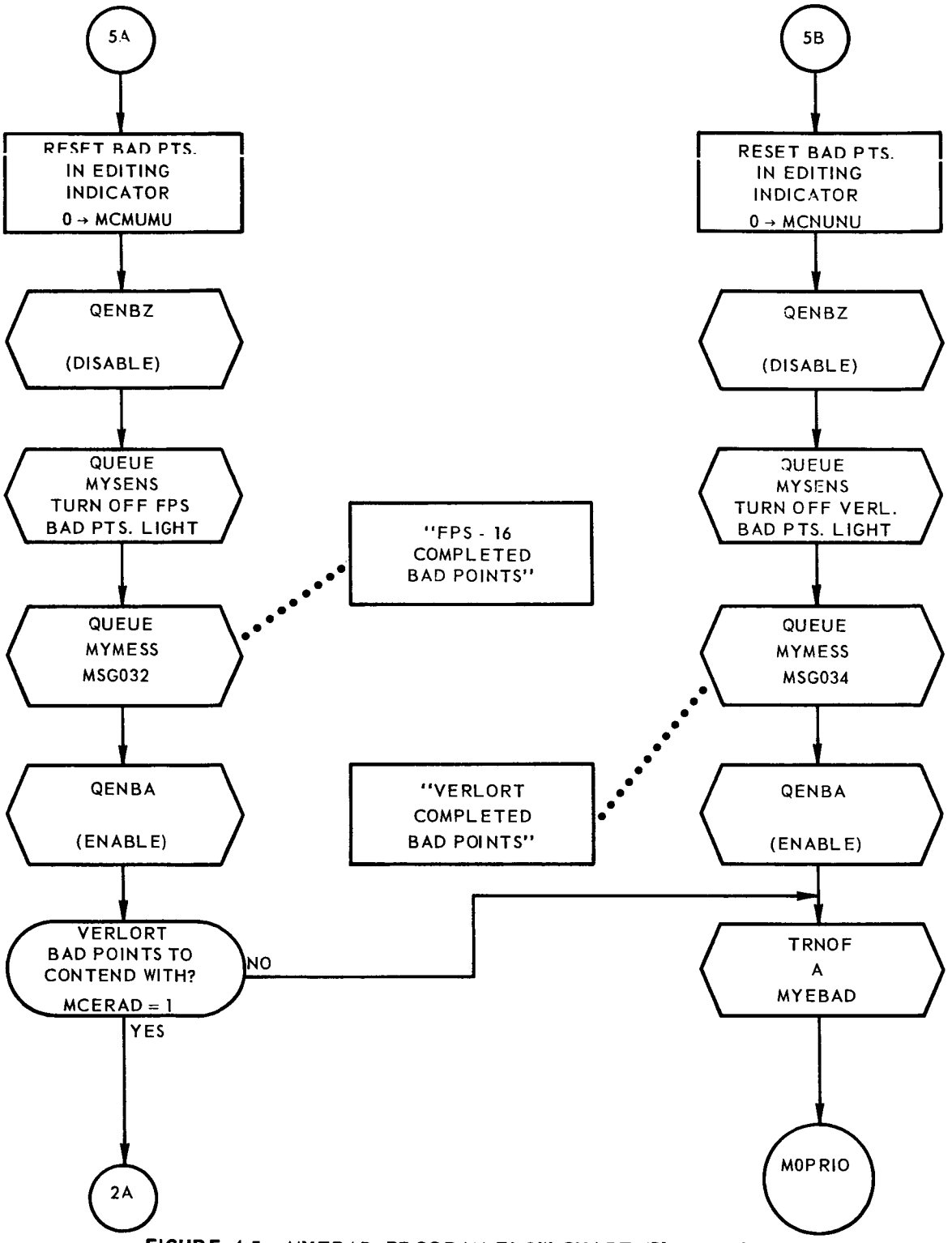


FIGURE 4-5. MYEBAD PROGRAM FLOW CHART *(Sheet 5 of 5)*

4.8 MONITOR PREFIX TO BARSCH (MPSARC)

MPSARC receives control from M0PRIO after the smoothing processor has completed and sets the required parameters for the short arc processor, BARSCH. MPSARC must determine whether or not sustainer engine cut-off (SECO) has occurred. If SECO has not been determined by the smoothing processor, or if nine seconds have not elapsed since SECO was reported via telemetry, BARSCH cannot be entered and MPARSC returns control to M0PRIO.

The flow chart for the MPSARC prefix is shown in Figure 4-6.

4.8.1 Input Requirements

Input to MPSARC consists of:

a) MCZETA—communication cell which, when non-zero, indicates the SECO time in floating-point tenths of seconds has been stored in TMFSEC.

b) BTSECO—one-cell table which, when non-zero, indicates the time of SECO has been determined by the smoothing processor, BSMOOS.

c) BTTSEC—one-cell table which contains the computed time of SECO, if available. The time is in fixed-point tenths of seconds.

d) MCSECO—communication cell which, when non-zero, indicates nine seconds have elapsed since SECO was reported via telemetry.

e) TMTSEP—one-cell table which, when non-zero, contains the time of capsule separation (CAPSEP) as reported via telemetry.

f) TMTABT—one-cell table which, when non-zero, contains the time of engine cut-off early (ECOE) as reported via telemetry. ECOE amounts to a premature termination of powered flight, i.e., an effective abort situation.

g) TMCYNO—timing entry for resetting and effectively stopping the timing of ECOE (see MTHFSC, subsection 3.17).

h) BTV1—internal cell which, when non-zero, indicates the initial entry to MPSARC.

i) K00001—constant, 1.

j) K00020—constant, 20 (decimal).

k) KCH233—mask, 233 000 000 000, used to convert a fixed-point number into floating-point format.

l) Macros used—TRNOF and TRNON.

4.8.2 Output Requirements

On the initial entry, MPSARC stores zero in the short arc parameter BTSAET and in the initial entry indicator BTV1. MPSARC must then determine whether the short arc processor may be entered. If SECO has not been determined by BSMOOS and if nine seconds have not elapsed since SECO was reported by telemetry, BARSCH may not be entered and MPSMOO returns control to M0PRIO.

If only one of the above SECO determinations is present, the cells listed below are set from the only time available: computed SECO from BSMOOS or time of CAPSEP from the telemetry processor, MYTMIN. If the only time present is from telemetry, a 1 is stored in BTSECO to indicate to BSMOOS that a capsule separation time has been obtained.

If a SECO time is present from both BSMOOS and telemetry, the CAPSEP time and ECOE (both from telemetry) are compared to obtain the earlier of the two. If this time is two or more seconds earlier than the computed SECO, the telemetry time is used as the SECO time for BARSCH. Otherwise, the computed SECO time is used as input to BARSCH. The following tables and cells are set after the proper time is selected:

a) BTTSEC—contains the selected time in fixed-point tenths of seconds.

b) TMFSEC and MCZETA—both contain the selected time in floating-point tenths of seconds

c) MCFSEC—contains a 1 to indicate the selected time is available in TMFSEC.

d) If the time computed by BSMOOS was the only time available 32,767 is stored in the decrement of both TMCYCL and TMCYCL+1 to stop SECO and ECOE timing and MCESPI is set non-zero to stop further ECOE or SECO testing.

4.8.3 Method

MPSARC is entered from M0PRIO and turns the A indicator on and the B indicator off for BARSCH. MPSARC turns on the C indicator for BSMOOS to suppress smoothing while the short arc processor is being used. If there is no indication of SECO or capsule separation, MPSARC turns off the C indicator for the smoothing processor, turns on the B indicator for the output computations processor, BOCOMP, turns off the A indicator for BARSCH and exits to M0PRIO.

The remainder of the method of MPSARC is described in subsections 4.8.1 and 4.8.2.

4.8.4 Usage

MPSARC is entered from M0PRIO; MPSARC exits either to BARSCH or to M0PRIO.

a) Storage Required—67 locations

b) MPSARC uses:

1) Macros—TRNOF and TRNON

2) Parameters—A, B, C, MNSARC and MNSMOO

3) Communication cells—MCESPI, MCFSEC, MCSECO and MCZETA

4) Tables—BTSAET, BTSECO, BTTSEC, TMCYCL, TMCYNO, TMTABT and TMTSEP

5) Internal cell—BTV1

6) Constants—K00001 and K00020

7) Mask—KCH233

c) Time Required:

1) Before SECO time established (maximum)—0.98 millisecond

2) After SECO time established—0.25 millisecond

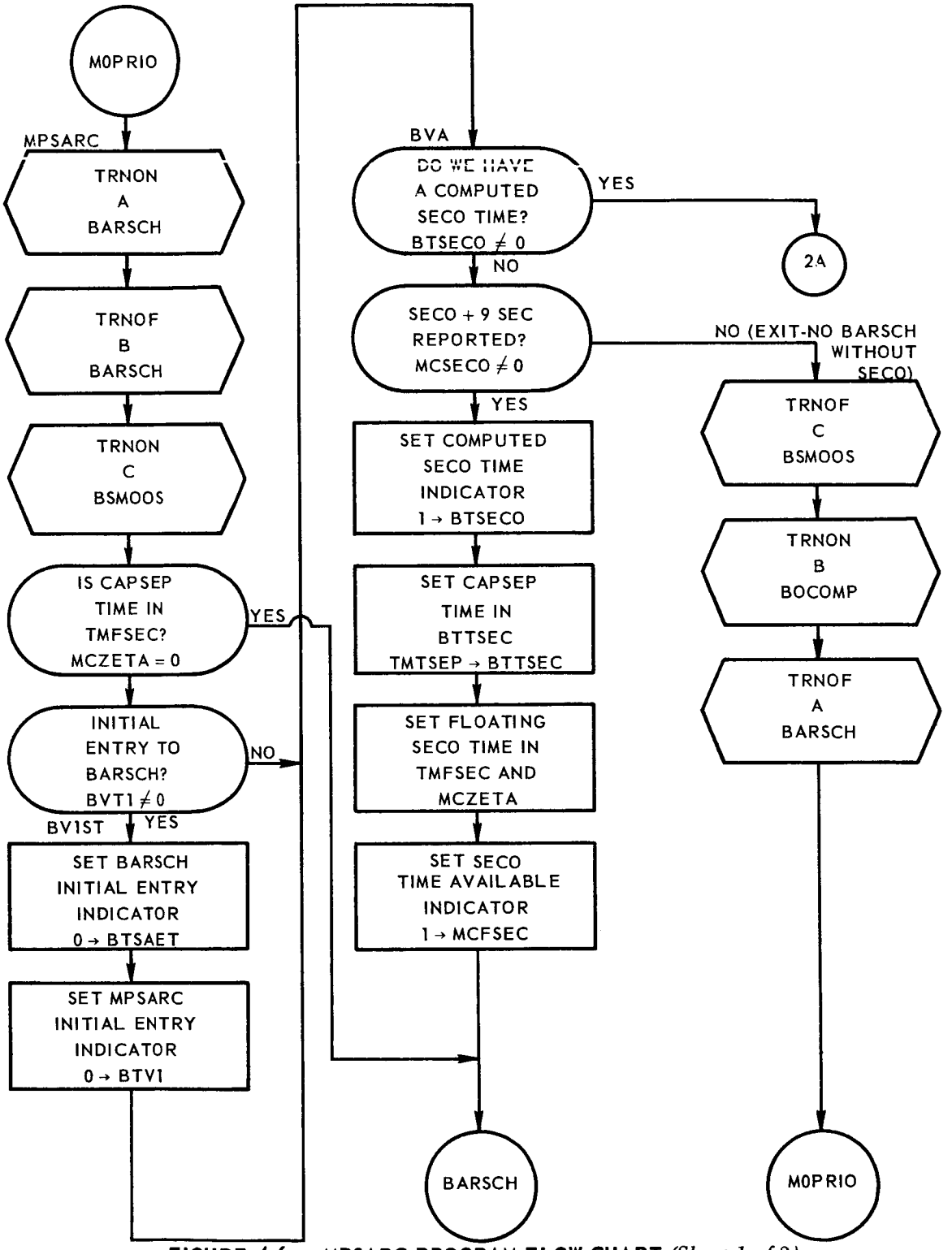


FIGURE 4-6. MPSARC PROGRAM FLOW CHART *(Sheet 1 of 2)*

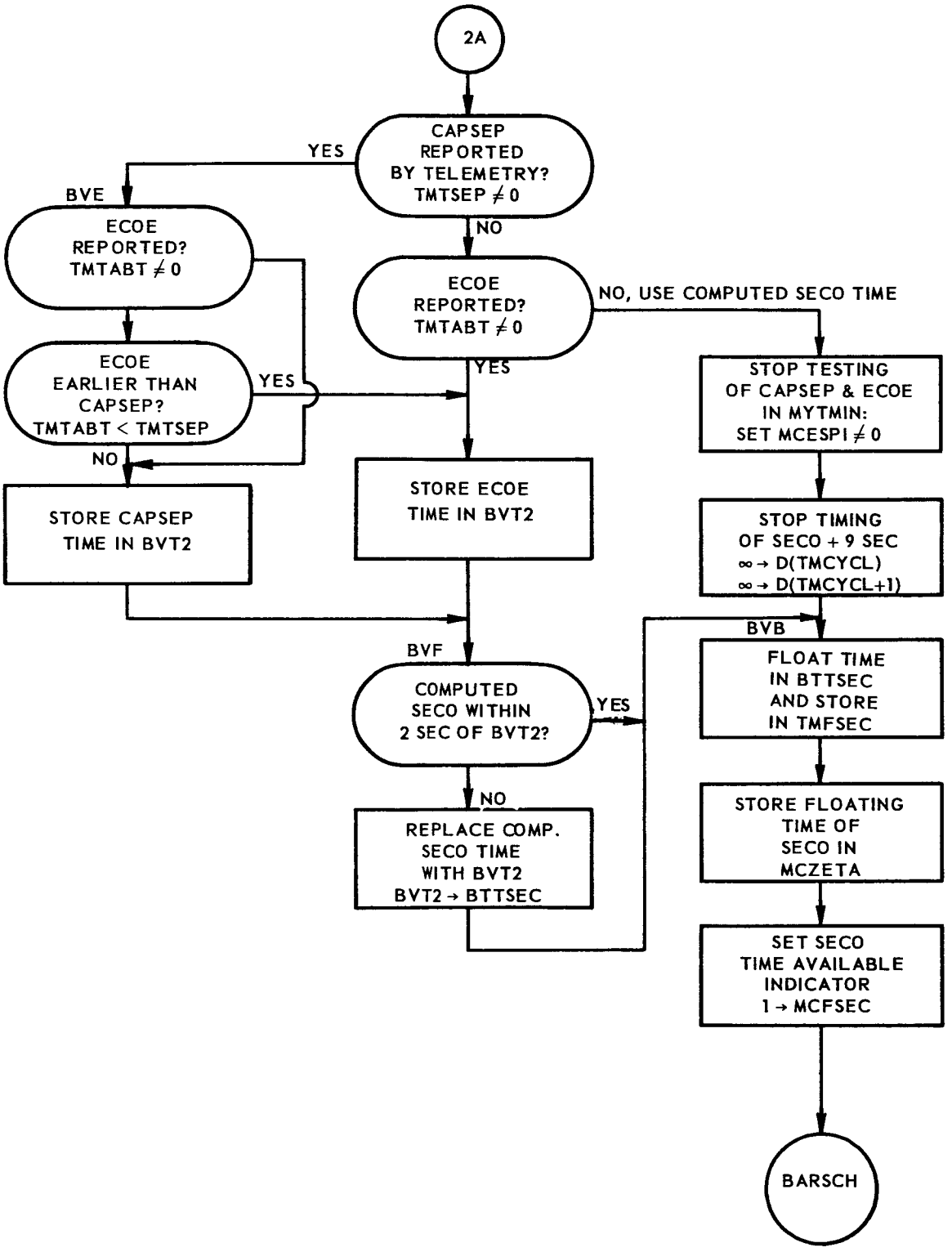


FIGURE 4-6. MPSARC PROGRAM FLOW CHART *(Sheet 2 of 2)*

4.9 MONITOR SUFFIX TO BARSCH (MFSARC)

MFSARC is entered at the completion of BARSCH. This suffix sets indicator bits in the priority table to direct the program flow to the next step in the processing cycle, output computations.

The flow chart for the MFSARC suffix is shown in Figure 4-7.

4.9.1 Input Requirements

Input to MFSARC consists of the TRNOF and TRNON macros.

4.9.2 Output Requirements

MFSARC sets indicator bits in the priority table and has no other output.

4.9.3 Method

MFSARC is entered from BARSCH and turns on the B indicator for the output computations processor BOCOMP, turns off the C indicator for BSMOOS, turns off the A indicator for BARSCH and exits to M0PRIO.

4.9.4 Usage

MFSARC is entered from BARSCH; MFSARC exits to M0PRIO.

a) Storage required—7 locations

b) MFSARC Uses:

1) Macros—TRNOF and TRNON

2) Parameters—A, B, C, MNCOMP, MNSARC and MNSMOO

3) Time Required—0.23 millisecond

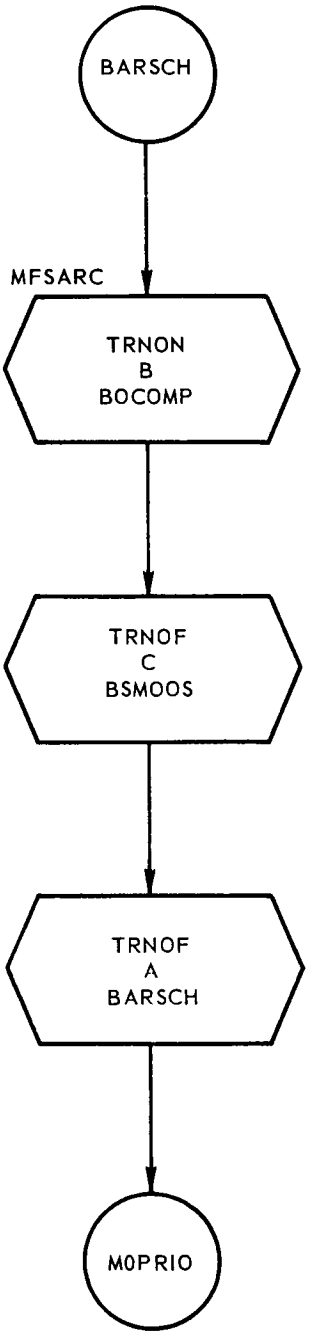


FIGURE 4-7. MFSARC PROGRAM FLOW CHART

4.10 MONITOR PREFIX TO BOCOMP (MPCOMP)

MPCOMP supplies initial entry parameters for the output computations processor, BOCOMP, and determines the time for the next high-speed output. MPCOMP also sets the mission phase lights on the Output Status Console and provides BOCOMP with the retrorocket firing indications and associated times.

The flow chart for the MPCOMP prefix is shown in Figure 4-8.

4.10.1 Input Requirements

Input to MPCOMP consists of:

a) PCPT1—internal cell which, when zero, indicates the initial entry into MPCOMP.

b) BTSECO—one-cell table which, when non-zero, indicates the time of SECO is available in TMFSEC.

c) TMFSEC—one-cell table containing the floating-point time of SECO.

d) BCLAOR—communication cell which indicates mission flight phase: 0, launch; -0, abort; 1, orbit; -1, re-entry.

e) BTFORF—three-cell table used to indicate the firing of retrorockets. BTFORF, when non-zero, indicates retro 1 has fired. BTFORF+1, when non-zero, indicates retros 1 +2 have fired. BTFORF+2, when non-zero, indicates retros 1 +2 +3 have fired.

f) BTTORF—three-cell table containing the times of retrofiring corresponding to the BTFORF table above. The BTFORF and BTTORF tables are set from telemetry input by MYTMIN.

g) MCHFSC—communication cell containing the current GMT in fixed-point half seconds.

h) TM8.3M—one-cell table containing the number of elapsed 8-1/3 milliseconds since the last exact half second.

i) K00001—constant, 1.

j) K00002—constant, 2.

k) K00005—constant, 5 (decimal).

l) K00012—constant, 12 (decimal).

m) K0250 —constant, 250 (floating point).

n) KCH233—mask, 233 000 000 000, used to convert a number from fixed-point to floating-point format.

o) Subroutine used—MSETPH (see subsection 4.11).

p) Macros used—TRNOF and TRNON.

Since MPCOMP uses the retrorocket fact-of-firing and time-of-firing indicator cells set by MYTMIN, this processor must be suppressed during the execution of MPCOMP to prevent these retro indicators from changing while MPCOMP is examining them.

4.10.2 Output Requirements

Output from MPCOMP depends upon the entry and the time elapsed since SECO.

a) On the initial entry only, the MPCOMP initial entry indicator, PCPT1, is set to non-zero. Zero is stored in the following indicator cells:

1) BTHOLD—the hold phase indicator.

2) BT0TRC—the GMTRC computed indicator.

3) BTSART—the short arc reset indicator.

4) BTSASO—the short arc solution (available) indicator.

5) MCOFOG—the teletype output initiated indicator.

6) MCFRVI—the orbit parameters computed indicator.

b) On every entry, the output from MPCOMP consists of the following cells, subject to the conditions given:

1) BTOUTT—the time of the next high-speed output is computed by incrementing the present time by two tenths of a second and storing this time in BTOUTT.

2) BTHOLD—if SECO has been reported and if the output time stored in BTOUTT is at least 25 seconds after SECO, a 1 is stored in BTHOLD to indicate the hold phase of the mission.

3) The MSETPH subroutine (see subsection 4.11) is executed to set the phase light on the Output Status Console

4) MCROAL—the phase setting (from MCLAOR) is stored in MCROAL to save the current phase for the next succeeding entry to BOCOMP.

5) MCRET1, MCRET2, MCRET3 are set to agree with BTFORT, +1, +2, respectively.

6) TMRET1, TMRET2, TMRET3 are set to agree with BTTORF, +1, +2, respectively.

4.10.3 Method

MPCOMP receives control from M0PRIO and turns the BOCOMP indicators A on and B off. The C indicator for MYTMIN is turned on to suppress the telemetry processor, since some of the input requirements for MPCOMP are controlled by MYTMIN.

If this initial entry to MPCOMP, six indicator cells used by BOCOMP are set to zero and the MPCOMP initial entry indicator is reset.

The time for the next output is computed by adding two tenths of a second to the current computer time and, if this time is 25 seconds later than SECO, the hold phase indicator is set.

MPCOMP executes the MSETPH subroutine which updates the phase lights of the Output Status Console. The present flight phase indication is stored for reference at the next entry to BOCOMP and the retrofire indicators used by BOCOMP are set from corresponding indicators controlled by MYTMIN. The C indicator for MYTMIN is turned off and MPCOMP exits to BOCOMP.

4.10.4 Usage

MPCOMP is entered from M0PRIO; MPCOMP exits to BOCOMP.

a) Storage Required—58 locations

b) MPCOMP uses:

1) Subroutine—MSETPH

2) Macros—TRNOF and TRNON

3) Parameters—A, B, C, MNCOMP and MNTMIN

4) Communication cells—MCFRVI, MCHFSC, MCLAOR, MCOFOG, MCRET1, MCRET2, MCRET3 and MCROAL.

5) Tables—BT0TRC, BTFORF, BTHOLD, BTOUTT, BTSART, BTSASO, BTSECO, BTTORF, TMFSEC, TMRET1, TMRET2, TMRET3 and TM8.3M

6) Internal cell—PCPT1

7) Constants—K00001, K00002, K00005, K00012 and K0250.

8) Mask—KCH233

c) Time Required—1.75 milliseconds

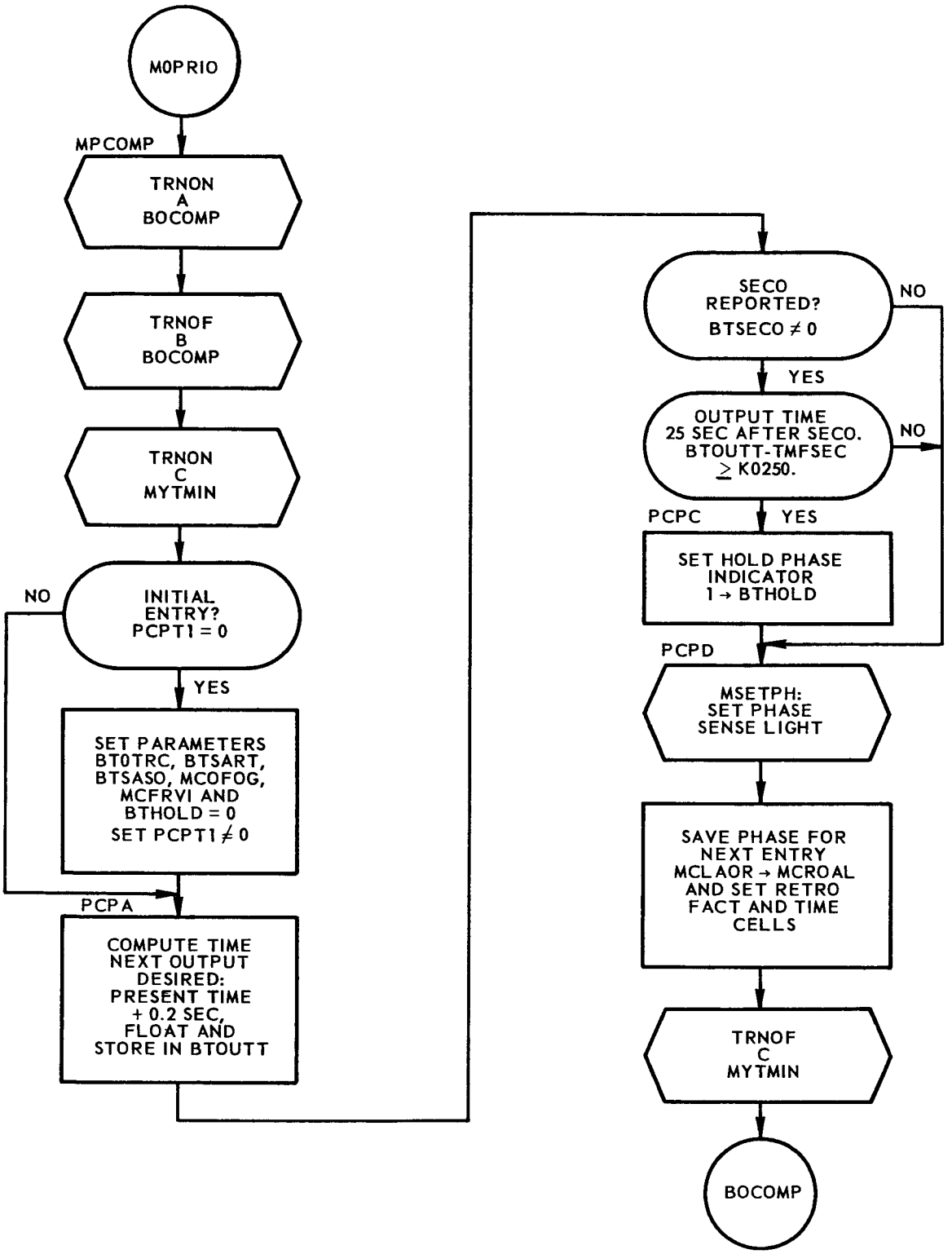


FIGURE 4-8. MPCOMP PROGRAM FLOW CHART

4.11 MONITOR PHASE SETTING SUBROUTINE (MSETPH)

MSETPH determines the current flight phase—launch, abort, hold, orbit or re-entry—and sets the phase indicator lights of Output Status Console to correspond to current phase. MSETPH is executed on every entry to MPCOMP.

The flow chart for the MSETPH subroutine is shown in Figure 4-9.

4.11.1 Input Requirements

Input to MSETPH consists of:

a) MCABOT—communication cell which, when non-zero, indicates an abort command (ICTRC request).

b) MCFSEC—communication cell which, when non-zero, indicates the time of SECO in floating-point has been stored in TMFSEC.

c) MCROAL—communication cell which indicates the flight phase at the last execution of MSETPH. The phase indications are: 0, launch; -0, abort; 1, orbit; -1, re-entry.

d) BT0RTC—one-cell table which, when non-zero, indicates the Greenwich Mean Time to retrofire (GMTRC) has been computed.

e) BTGMTR—one-cell table which, when non-zero, contains the GMTRC in floating-point tenths of seconds.

f) BTFORF, + 1, + 2—three-cell table used to indicate the firing of retro 1, retros 1 + 2 and retros 1 + 2 + 3, respectively. The indication is positive when the particular cell is non-zero.

g) BTOUTT—one-cell table containing the time for the next output in floating-point tenths of seconds. This cell is set by MPCOMP prior to entering MSETPH.

h) BTNPWC—a one-cell table which provides a criterion for entering the hold phase. The criterion is a function of the number of observations plus the weight given to these observations. The contents of BTNPWC must be equal to or greater than 44 before the hold phase may be entered.

i) TMFSEC—one-cell table which contains the time of SECO in floating-point tenths of seconds.

j) TMRETS—one-cell table containing the capsule clock retrofire setting in fixed-point tenths of seconds.

k) BK.12H—constant, 1200 (floating-point).

l) BK44.0—constant, 44 (floating-point).

m) K00000—constant, 0.

n) K00001—constant, 1.

o) K00002—constant, 2.

p) K00004—constant, 4 (decimal).

q) K00008—constant, 8 (decimal).

r) K00016—constant, 16 (decimal).

s) K0250.—constant, 250 (floating-point).

t) KCH233—mask, 233 000 000 000, used to convert a number from fixed-point to floating-point.

u) Macros used—SETPH (described on page 3 of Figure 4-). QENBA, QENBZ and QUEUE are used by the SETPH macro.

4.11.2 Output Requirements

The primary output from MSETPH is the input mask queued for sense output processor, MYSENS, to change, if necessary, the phase light on the Output Status Console. MSETPH sets the phase indicator cells, MCLAOR and BTHOLD. These and other cells which may be output from MSETPH are:

a) MCLAOR—set to indicate phase: 0, launch; -0, abort; +1, orbit; -1, re-entry.

b) BTHOLD—a 1 is stored in BTHOLD to indicate the hold phase; if the mission was in, but has left, the hold phase since the last entry to MSETPH, BTHOLD is cleared.

c) BT0TRC—zero is stored in BT0TRC if any retrofiring has been reported.

d) BTECTS—if an abort has been ordered and if the next output time is not less than the computed time to retrofire, the capsule clock setting for retrofire is converted to floating-point tenths of seconds and is stored in BTECTS.

e) BTODPL+5—when conditions are such that the mission enters the orbit or hold phase for the first time, a 1 is stored in BTODPL+5 to

indicate to BOCOMP that the current output computations cycle is an even cycle.

4.11.3 Method

The method of MSETPH is given in Figure 4-9.

4.11.4 Usage

MSETPH is entered from, and returns to, MPCOMP.

a) Calling Sequence:

```
TSX      MSETPH, 4
Return
```

b) Storage Required—140 locations

c) MSETPH uses:

1) Macros—QENBA, QENBZ, QUEUE and SETPH

2) Parameters—MNSENS, SETL, SETML, SETO, SETMO, SETMH, SETA, SETMA, SETR and SETMR

3) Communication cells—MCABOT, MCFSEC, MCLAOR and MCRAOL

4) Tables—BT0TRC, BTECTS, BTFORF, BTGMTR, BTHOLD, BTNPWC, BTODPL, BTOUTT, TMFSEC and TMRETS

5) Constants—K00000, K00002, K00004, K00008, K00016, K0250., BK.12H and BK44.0

6) Mask—KCH233

d) Time Required:

1) Minimum—0.22 millisecond

2) Maximum—1.63 milliseconds

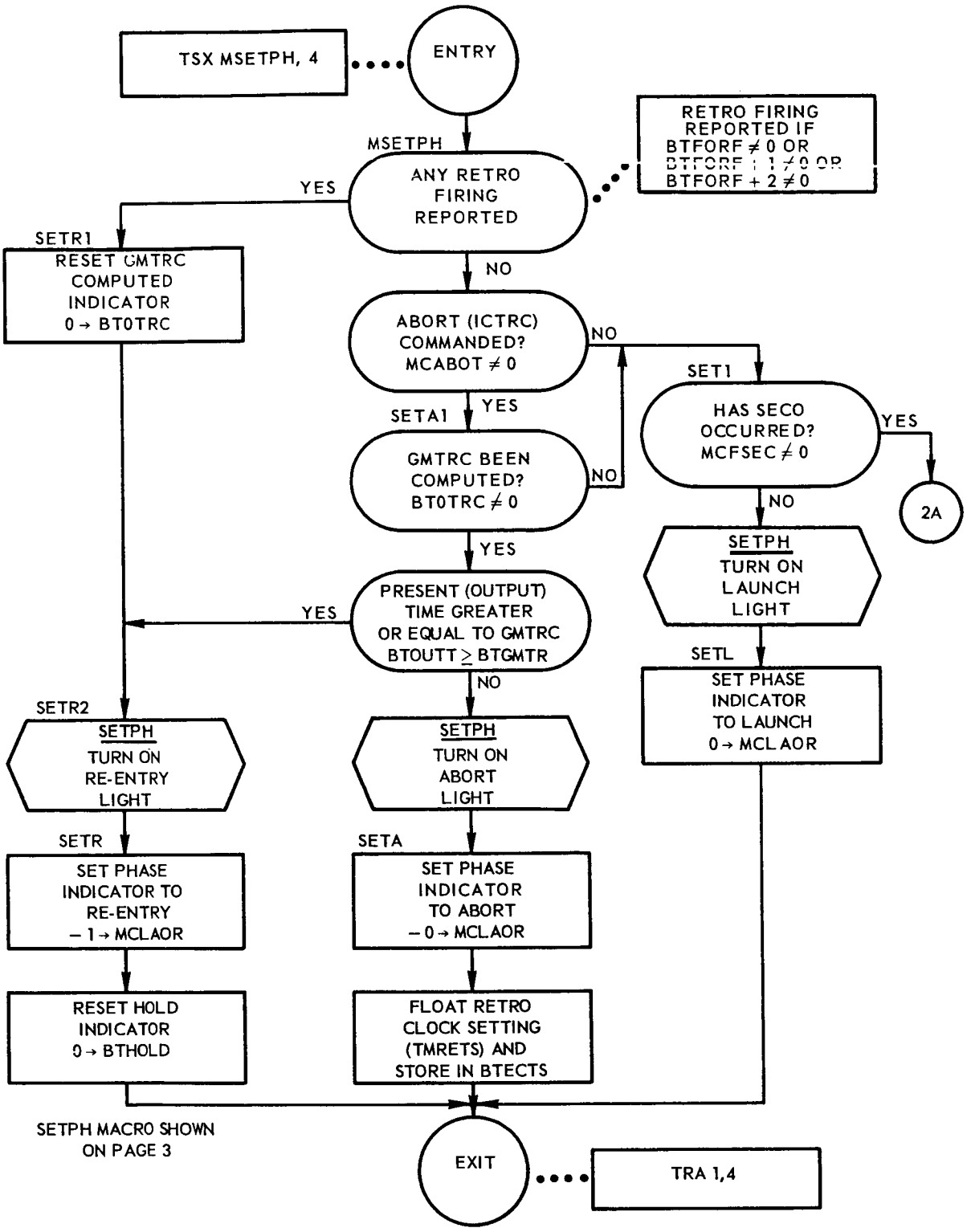


FIGURE 4-9. MSETPH PROGRAM FLOW CHART *(Sheet 1 of 3)*

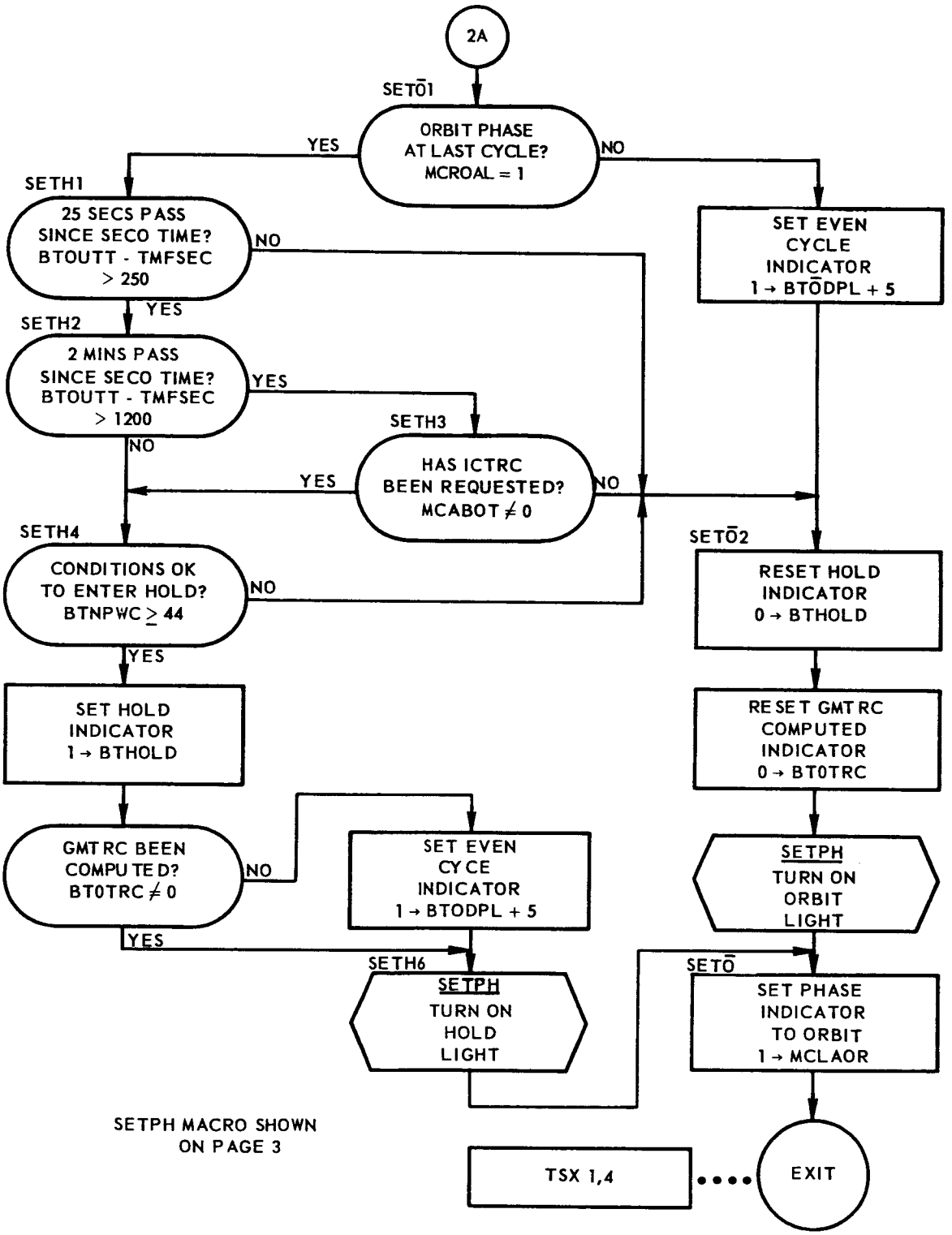
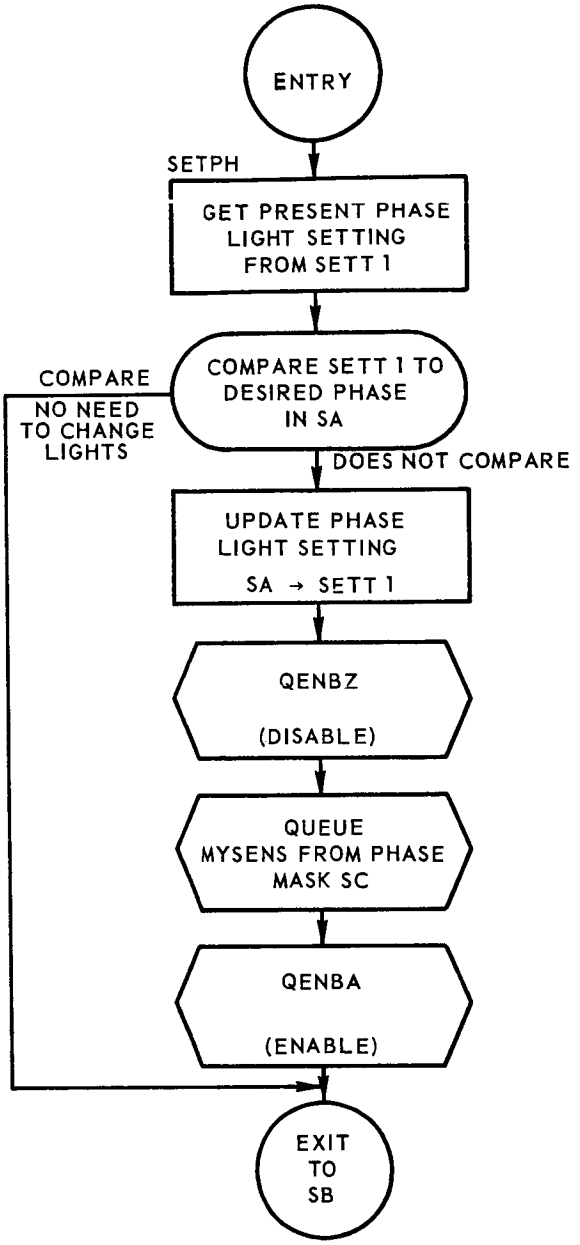


FIGURE 4-9. MSETPH PROGRAM FLOW CHART *(Sheet 2 of 3)*




```
SETPH    MACRO    SA, SB, SC
         CLA      SETT1
         ANA      SA
         TNZ      SB
         LDQ      SA
         STQ      SETT1
         QENBZ
         SXA      SETT2, 4
         QUEUE    MNSENS, SC
         LXA      SETT2, 4
         QENBA
         END
```

PHASE	SA	SB	SC
LAUNCH	K00001	SETL	SETML (PZE 32,, 30)
ORBIT	K00002	SETO	SETMO (PZE 16,, 46)
ORBIT-HOLD	K00004	SETO	SETMH (PZE 24,, 38)
ABORT	K00008	SETA	SETMA (PZE 4,, 56)
RE-ENTRY	K00016	SETR	SETMR (PZE 4,, 60)

FIGURE 4-9. MSETPH PROGRAM FLOW CHART *(Sheet 3 of 3)*

4.12 MONITOR SUFFIX TO BOCOMP (MFCOMP)

MFCOMP receives control when BOCOMP completes output processing and this suffix sets priority indicators which control the program flow. MFCOMP also queues on-line messages to indicate significant events which have been determined by BOCOMP and initiates teletype transmission to Goddard of the orbital parameters after these parameters have been prepared.

The flow chart for the MFCOMP suffix is shown in Figure 4-10.

4.12.1 Input Requirements

Input to MFCOMP consists of:

a) FCT1—internal cell which, when zero, indicates that the teletype transmission to Goddard has not yet been initiated.

b) MCFRVI—communication cell which, when non-zero, indicates that the orbit parameters have been computed.

c) Macros Used—QENBA, QENBZ, QUEUE, TRNOF and TRNON.

4.12.2 Output Requirements

If teletype transmission to Goddard has not been initiated and if the orbit parameters for transmission have been prepared, MFCOMP turns on the B indicator for the teletype output generator, BOTOUT, and sets FCT1 non-zero to indicate the initiation of teletype to Goddard. In addition MFCOMP may have other output depending upon the location at which BOCOMP enters its suffix. This output is listed with the associated entry location in MFCOMP:

a) BMOCHX entry—MFCOMP queues the on-line message processor with the message "HYPERBOLIC ORBIT HAS BEEN COMPUTED" and turns on the B indicator for the output-to-Bermuda displays processor, BODSPL.

b) BMONIN entry—MFCOMP queues the message "NO INTERSECTION WITH 450,000 FEET" and turns on the B indicator for BODSPL.

c) BMOPIO entry—MFCOMP queues the message "TIME TO FIRE RETROS FOR LAST RECOVERY AREA HAS PASSED" and turns on the B indicator for BODSPL.

d) BMOCNX entry—MFCOMP turns on the B indicator for BODSPL. This is the normal entry to MFCOMP.

e) BMONNO (no new output) entry—no additional output.

For every entry the A indicator for BOCOMP is turned off and MFCOMP exits to M0PRIO.

4.12.3 Method

The method of MFCOMP is described in subsection 4.12.2.

4.12.4 Usage

MFCOMP is entered (at one of five different locations) from BOCOMP; MFCOMP exits to M0PRIO.

a) Storage Required—38 locations

b) MFCOMP Uses:

1) Macros—QENBA, QENBZ, QUEUE, TRNOF and TRNON

2) Parameters—A, B and MNMESS

3) Communication cell—MCFRVI

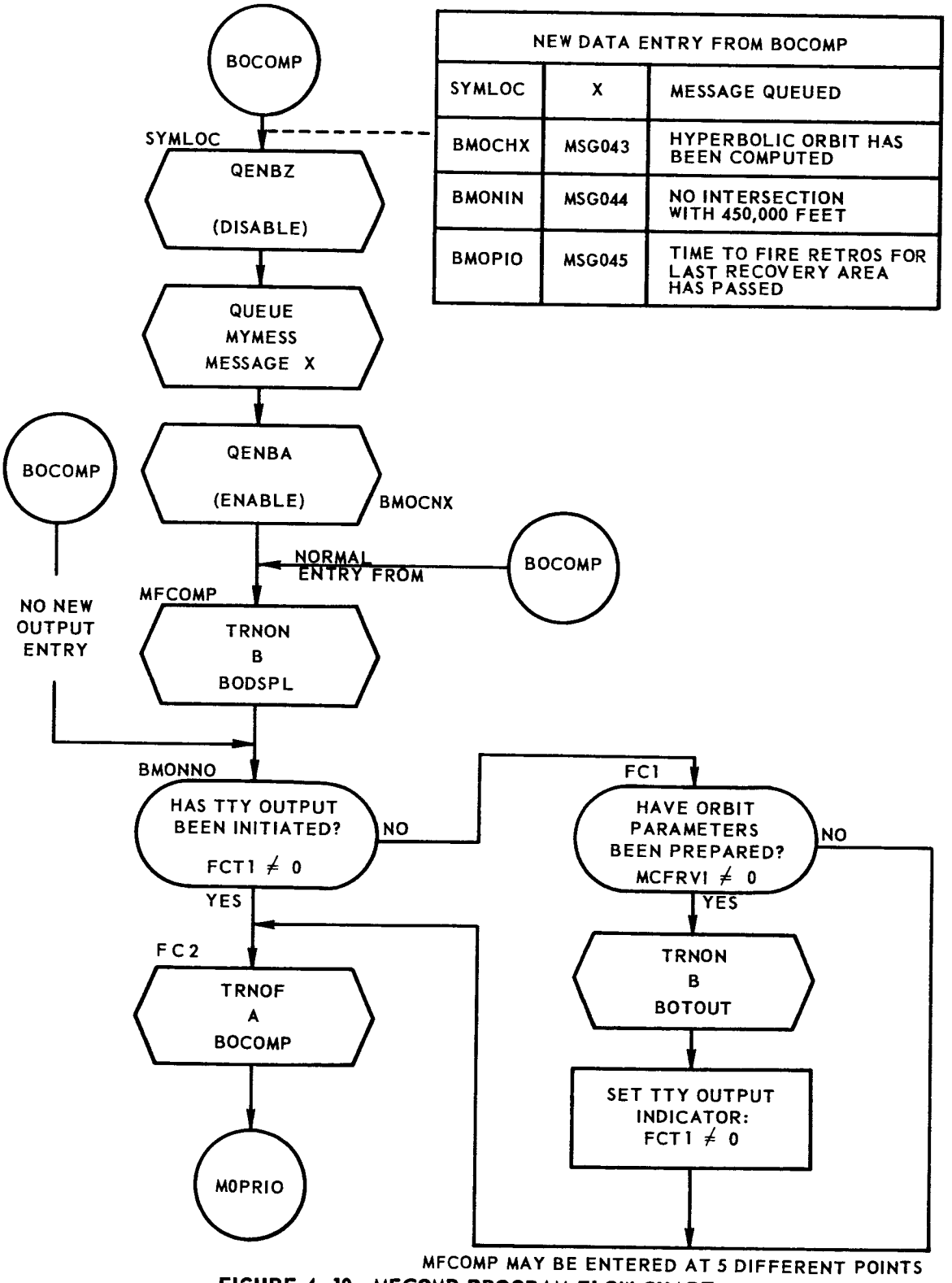
4) Internal cell—FCT1

c) Time Required:

1) Entry at BMOCHX, BMONIN or BMOPIO—0.84 millisecond

2) Entry at BMOCNX—0.30 millisecond

3) Entry at BMONNO—0.24 millisecond



NEW DATA ENTRY FROM BOCOMP		
SYMLOC	X	MESSAGE QUEUED
BMOCHX	MSG043	HYPERBOLIC ORBIT HAS BEEN COMPUTED
BMONIN	MSG044	NO INTERSECTION WITH 450,000 FEET
BMOPIO	MSG045	TIME TO FIRE RETROS FOR LAST RECOVERY AREA HAS PASSED

FIGURE 4-10. MFCOMP PROGRAM FLOW CHART

Section 5.
DCC OUTPUT CONTROL

Three types of output data are transmitted from the IBM 709 computer via the DCC: high-speed data to drive the Bermuda displays, low-speed teletype messages for the Goddard Communications Center and sense output for the indicator lights of the Output Status Console.

As with DCC input transmission, the internal trapping feature of the IBM 709 is utilized to minimize the processing time required for output transmission. Transmission is initiated by the computer program and only when the transmission is complete must the processing be interrupted. All interrupts produced by the DCC are interpreted by M0DCCC which saves the condition of the interrupted program and gives control to the appropriate Monitor trap processor.

The Monitor programs which control DCC output data are described in detail in this section. The DCC output from the Bermuda Mercury Program System is illustrated in Figure 5-1.

5.1 HIGH-SPEED OUTPUT TO BERMUDA DISPLAYS

High-speed output data is provided to the Bermuda displays at a frequency of once per second. This output is a function of the mission flight phase and is also dependent upon the frame. Alternating between odd and even frames, one frame is transmitted every second. The output data drives digital displays and a plotboard in the Bermuda Control Center.

5.1.1 Bermuda Displays

During the launch phase two items are displayed on the plotboard:

a) the latitude and longitude of impact point for immediate retrofire and

b) the velocity ratio plotted against the capsule's flight path angle. These quantities are updated every odd frame.

The digital displays provided on the odd frame during the launch phase consist of: (1) the capsule's height in nautical miles, to the nearest tenth of a mile, (2) the flight path angle to the nearest hundredth of a degree and the velocity ratio to four decimal places. During the even frame the GO NO-GO recommendation is provided. Until sufficient data has been received by the computer, the GO

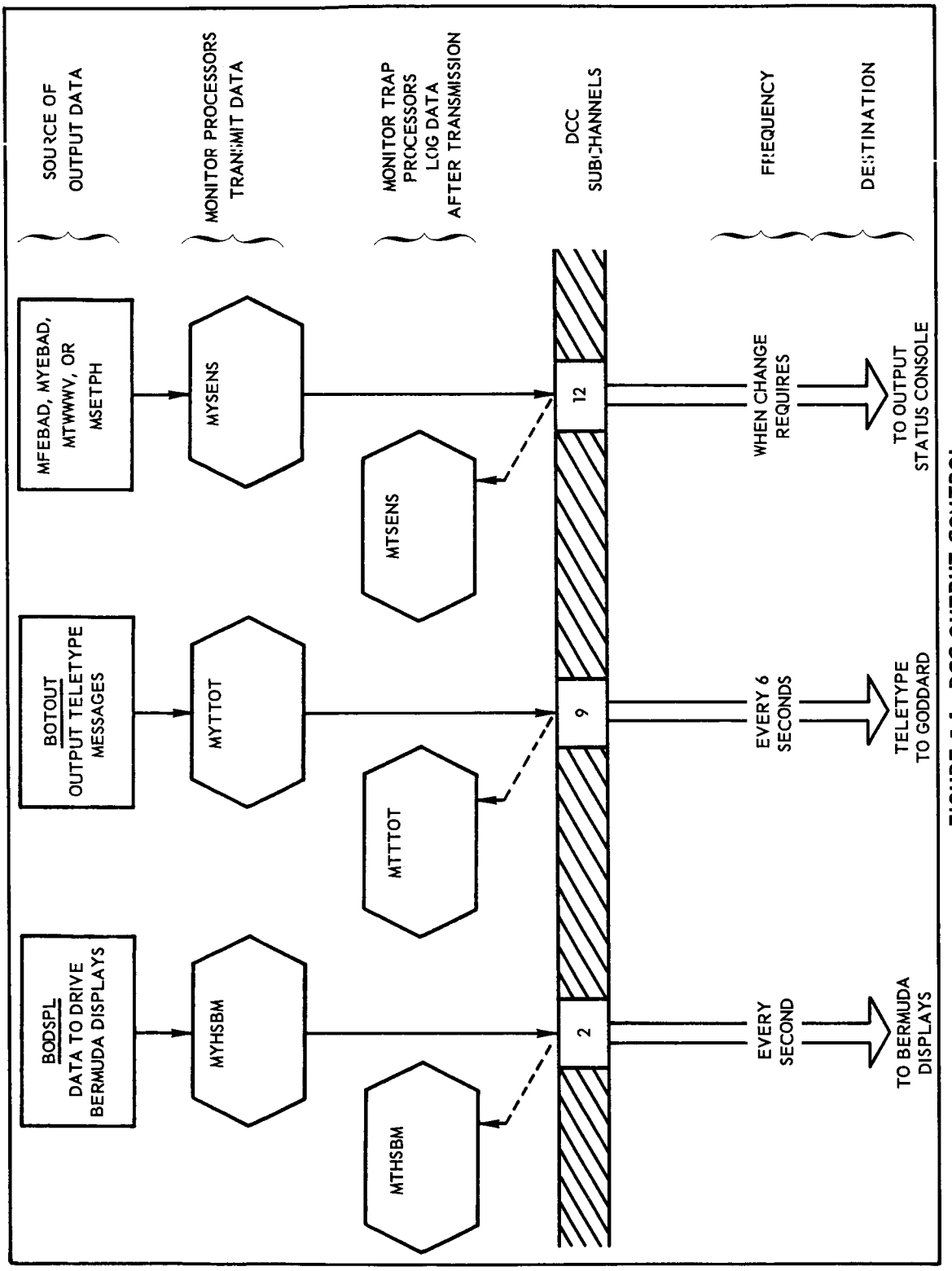


FIGURE 5-1. DCC OUTPUT CONTROL

NO-GO lamps are dormant. Thereafter, the lamp reflects the best estimate of the computer. The setting of the lamps when the capsule enters the hold phase, is fixed until re-entry, when the NO-GO lamp is always lit.

The orbit phase displays for the plotboard are the same as those provided during the launch phase. The digital displays provided during orbit are the same as those provided during the launch phase for both frames, except that two additional quantities are given on the even frame: the computed GMT for retrofire in hours, minutes and seconds, the computed elapsed capsule time for retrofire in hours, minutes and seconds, and a designation of the next recovery area. During the hold phase, the time from capsule separation or abort initiate to GMT of retrofire is displayed in lieu of the flight path angle.

The abort phase displays for the plotboard consist of the latitude and longitude of both the capsule's present position and the impact point for retrofire after a 30 second delay. The digital displays for abort are the same as those for orbit with the addition of the incremental capsule time to retrofire, i.e., the difference between the elapsed capsule time setting and the elapsed capsule time for retrofire.

The re-entry phase displays for plotboard consist of the latitude and longitude of both the capsule's present position and the refined impact point. The digital displays during re-entry are the same as those for orbit, except that the GMT of retrofire is displayed in place of GMT for retrofire, and elapsed time since retrofire is displayed in place of elapsed capsule time for retrofire.

5.1.2 The High-Speed Output Control Cycle

The high-speed output cycle begins with the entry to the Output Computation Program, BOCOMP. This program is given a phase and frame indication. From these indications and the data provided by the smoothing and short arc programs, BOCOMP produces an 11-cell output table. If new data has been placed in the table, the B indicator for Output-to-Displays Program, BODSPL, is turned on by the Monitor Suffix to BOCOMP.

BODSPL processes the 11-cell output table from BOCOMP, converting the data into the proper format and packing the data into the eight low-order bit positions in a 12-cell output table. BODSPL exits to its suffix, MFDSPL, which turns on the B indicator for the Monitor High-Speed Output Processor, MYHSBM. Since BODSPL has a lower priority than MYHSBM, the prefix to BODSPL suppresses MYHSBM and the suffix unsuppresses MYHSBM.

MYHSBM moves the data from the BODSPL output table into the output block for DCC subchannel 2 and clears the last word in the BODSPL output table to indicate to the BODSPL prefix that the table may be refilled. MYHSBM activates the subchannel to initiate transmission of the message and then suppresses itself. The trap following the transmission of the high-speed message gives control to the Monitor High-Speed Output Trap Processor, MTHSBM, which deacti-

vates subchannel 2, logs the data and removes the MYHSBM suppression indicator set by MYHSBM. When BOCOMP produces new output, the cycle repeats.

MPDSPL, MFDSPL, MYHSBM and MTHSBM are described in detail in subsections 5.2 through 5.5. BODSPL is described in detail in MC 106, Bermuda Processing Programs.

5.2 MONITOR PREFIX TO BODSPL (MPDSPL)

MPDSPL receives control from M0PRIO and determines whether or not the last output message for the Bermuda displays has been transmitted. If the last message has been transmitted MPDSPL gives control to BODSPL which prepares the next output message. Otherwise, MFCOMP returns control to M0PRIO until the current message has been transmitted.

The flow chart for the MPDSPL prefix is shown in Figure 5-2.

5.2.1 Input Requirements

Input to MPDSPL consists of the last location of the TMHSOT table. If this location contains zero, the last packed output message prepared by BODSPL has been transmitted and a new message may be prepared.

MPDSPL uses the TRNON and TRNOF macros.

5.2.2 Output Requirements

MPDSPL directs the program flow, but has no other output.

5.2.3 Method

MPDSPL receives control from M0PRIO after the output computations processor, BOCOMP, has indicated new output data is available. MPDSPL turns on the A indicator for BODSPL and examines the last location of the BODSPL output table, TMHSOT. If this location contains zero, the table is free to accept new data. MPDSPL turns off the B indicator for BODSPL, turns on the D indicator for MYHSBM (to suppress the output transmission processor until after the new message has been prepared) and exits to BODSPL.

If the last location in TMHSOT contains data, MPDSPL turns off the A indicator for BODSPL and control returns to M0PRIO. The B indicator for BODSPL remains on and M0PRIO re-enters MPDSPL, as priority considerations permit, until the BODSPL output table is free to accept new data.

5.2.4 Usage

MPDSPL is entered from M0PRIO; MPDSPL exits to BODSPL or to M0PRIO.

a) Storage Required—11 locations

b) MPDSPL uses:

1) Macros—TRNOF and TRNON

2) Parameters—A, B, C, MNDSPL and MNHSBM

3) Table—TMHSOT

c) Time Required:

1) MPDSPL exits to BODSPL—0.27 millisecond

2) MPDSPL exits to M0PRIO—0.18 millisecond

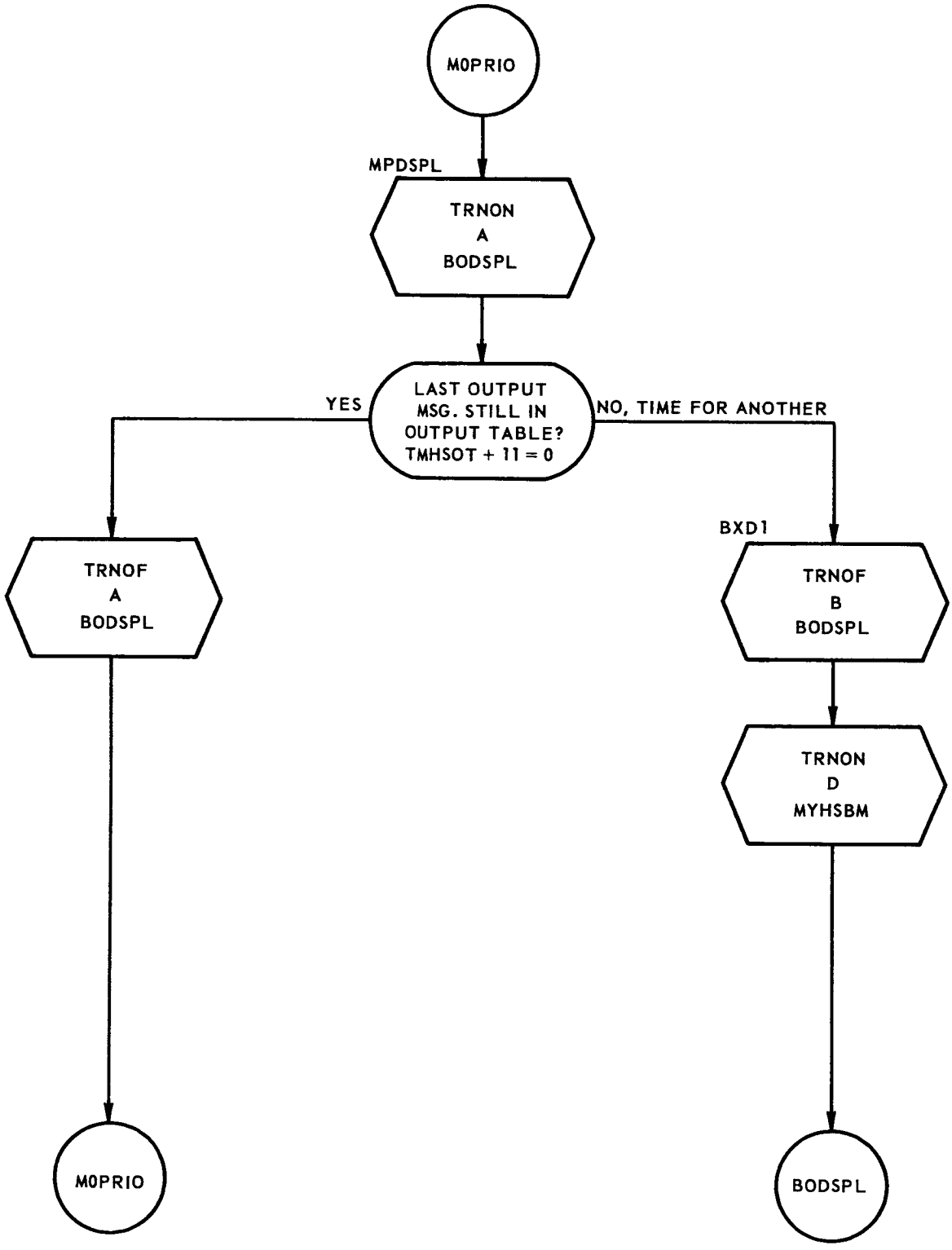


FIGURE 5-2. MPDSPL PROGRAM FLOW CHART

5.3 MONITOR SUFFIX TO BODSPL (MFDSPL)

When BODSPL has completed packing the output table for the Bermuda displays, BODSPL exits to its suffix. MFDSPL sets indicators in the priority table for the transmission processor, MYHSBM.

The flow chart for the MFDSPL suffix is shown in Figure 5-3.

5.3.1 Input Requirements

MFDSPL uses the TRNOF and TRNON macros, but has no other input.

5.3.2 Output Requirements

MFDSPL sets indicators in the priority table. There is no other output.

5.3.3 Method

MFDSPL is entered from BODSPL after the packed output table has been prepared. The D indicator is turned off for MYHSBM to unsuppress the transmission processor (MYHSBM was suppressed by MPDSPL before BODSPL was entered). The B indicator for MYHSBM is turned on, the A indicator for BOCOMP is turned off and MFDSPL exits to M0PRIO.

5.3.4 Usage

MFDSPL is entered from BODSPL; MFDSPL exits to M0PRIO.

a) Storage Required—seven locations

b) MFDSPL uses:

   1) Macros—TRNOF and TRNON

   2) Parameters—A, B and D

c) Time Required—0.23 millisecond

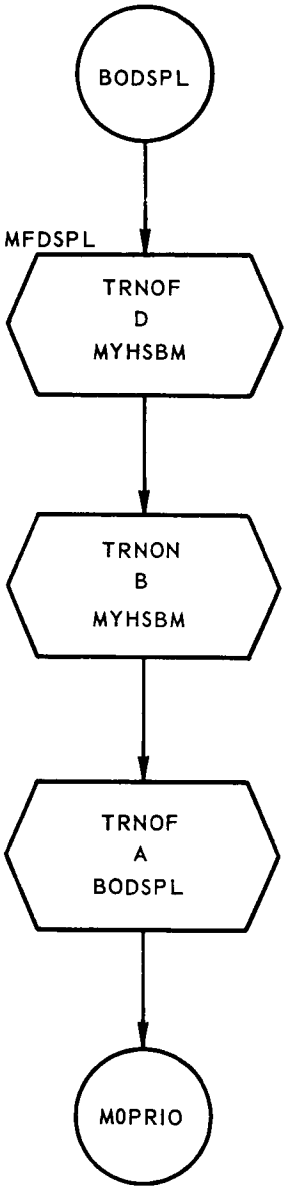


FIGURE 5-3. MFDSPL PROGRAM FLOW CHART

5.4 MONITOR HIGH-SPEED OUTPUT PROCESSOR (MYHSBM)

MYHSBM receives control from M0PRIO after BODSPL has prepared a new packed output message for the Bermuda displays. MYHSBM generates a logging time tag for the message and transmits the message to the displays via DCC subchannel 2.

The flow chart for the MYHSBM processor is shown in Figure 5-4.

5.4.1 Input Requirements

Input to MYHSBM consists of:

a) TMHSOT—twelve-cell table containing the packed output message prepared by BODSPL. The packed data is placed in bits 28 through 35 of the computer word.

b) MCHFSC—communication cell containing the GMT in fixed-point half seconds.

c) TM8.3M—one-cell table containing the number of elapsed 8-1/3 millisecond intervals since the last exact half-second.

d) CGMAS—internal cell containing a 1 in bit position 2, used to control the activation of DCC subchannel 2.

e) Macros used—QENBZ, QPSLF, TRNOF and TRNON.

5.4.2 Output Requirements

Output from MYHSBM consists of:

a) TMHSBM—the output message is moved from the BODSPL table, TMHSOT, into TMHSBM. The data occupies the same relative position in both 12-cell tables. The sign of the last word in TMHSBM is set negative to indicate the end of message.

b) TMHSOT—the last cell in the BODSPL output table is cleared.

c) MCCG01—the logging time tag is generated and stored in MCCG01.

d) Subchannel 2 is activated and the message is transmitted.

5.4.3 Method

MYHSBM receives control and turns its A indicator on and its B indicator off. The BODSPL output table is moved intothe output block (table TMHSBM) for DCC subchannel 2. The sign of the last word is set negative to indicate an end-

of-message. The last location in the BODSPL output table is cleared to indicate to MPDSPL that the table has been emptied and may be refilled.

MYHSBM generates the logging time tag and saves this time for the Monitor High-Speed Output Trap Processor, MTHSBM, which logs the data after transmission is complete. The computer is disabled from trapping, a 1 is stored in bit position 2 of the DCC subchannel mask, subchannel 2 is activated and the message is transmitted. The C indicator for MYHSBM is turned on to suppress this processor until the message is completely transmitted. MYHSBM turns off its A indicator and returns program control to M0PRIO.

5.4.4 Usage

MYHSBM is entered from, and exits to, M0PRIO.

a) Storage Required—28 locations

b) MYHSBM uses:

1) Macros—QENBZ, QPSLF, TRNOF and TRNON

2) Parameters—A, B, and MNHSBM

3) Communication cells—MCACTV, MCCG01 and MCHFSC

4) Tables—TM8.3M, TMHSBM and TMHSOT

5) Internal cell—CGMAS

c) Time Required—1.82 milliseconds

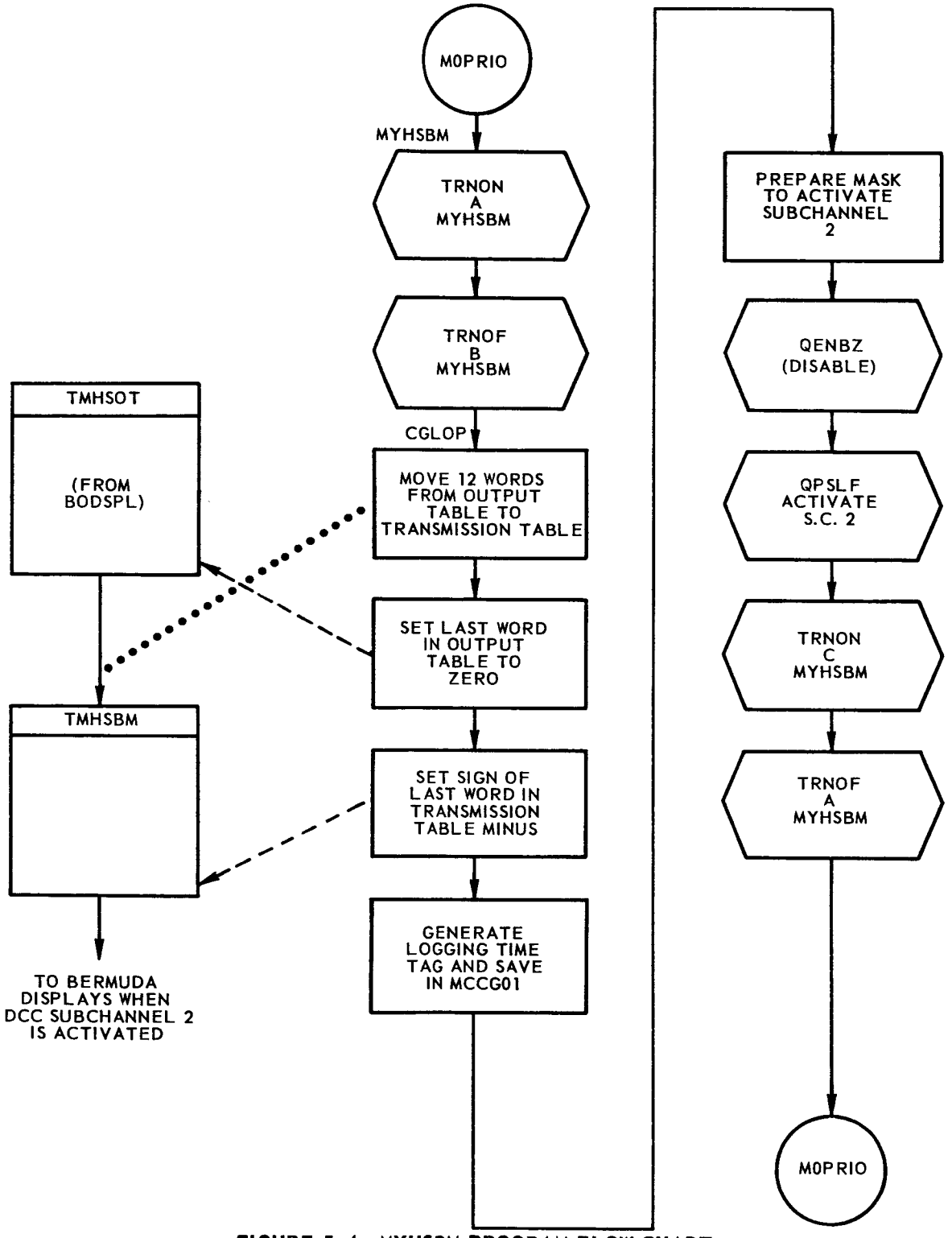


FIGURE 5-4. MYHSBM PROGRAM FLOW CHART

5.5 MONITOR HIGH SPEED OUTPUT TRAP PROCESSOR (MTHSBM)

MTHSBM receives control from M0DCCC following a trap on DCC subchannel 2. MTHSBM deactivates subchannel 2 and logs the transmitted message.

The flow chart for the MTHSBM processor is shown in Figure 5-5.

5.5.1 Input Requirements

Input to MTHSBM consists of:

a) TMHSBM—12-cell table containing the high-speed message transmitted.

b) MCCG01—communication cell containing the logging time tag associated with the transmitted message. This cell is set by MYHSBM.

c) Subroutine used—MSLOGG.

d) Macros used—QPSLF and TRNOF.

5.5.2 Output Requirements

MTHSBM deactivates subchannel 2, logs the transmitted message and unsuppresses MYHSBM.

5.5.3 Method

MTHSBM receives control from M0DCCC following a trap on DCC subchannel 2, MTHSBM deactivates the subchannel, loads the MQ with the logging time tag generated by MYHSBM and logs the output message. MTHSBM turns off the C indicator for, and unsuppresses, MYHSBM and exits to M0PRIO.

5.5.4 Usage

MTHSBM is entered from M0DCCC following a trap on DCC subchannel 2; MTHSBM exits to M0PRIO.

a) Storage Required—10 locations

b) MTHSBM uses:

1) Subroutine—MSLOGG

2) Macro—QPSLF and TRNOF

3) Parameters—C and MNHSBM

4) Communication cells—MCACTV and MCCG01

5) Table—TMHSBM

c) Time Required—2.95 milliseconds.

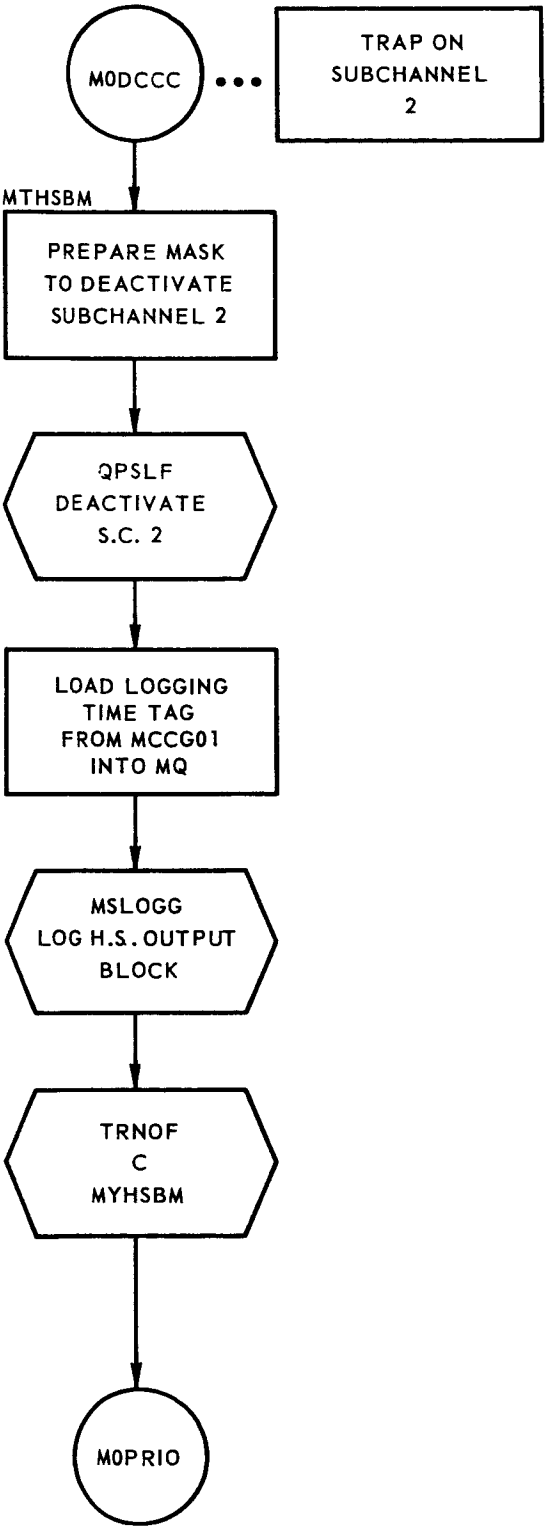


FIGURE 5-5. MTHSBM PROGRAM FLOW CHART

5.6 TELETYPE OUTPUT TO GODDARD

The Bermuda processing program provides two different types of teletype messages* for transmission to the Mercury Communications Center at Goddard: position/velocity components and smoothed radar data. These two types of messages and the programs which generate and transmitted the teletype messages are described in the following subsections.

5.6.1 Position/Velocity Components

This message consists of the three vector components of both range and elevation headed by two times. The first time is the time of sending the message, and the second time is associated with vector components. These vectors represent the capsule position/velocity at the first integral minute after a short arc solution has been obtained and effectively produces the capsule's range and velocity vectors at insertion into orbit. This is the first computer processed teletype message transmitted and is transmitted only once.

5.6.2 Smoothed Radar Data

This message consists of one frame comprised of a range, azimuth and elevation, headed by an associated time tag. The smoothed radar values are sent to Goddard continuously at a rate of one frame from each radar every six seconds, beginning after the position/velocity vectors have been transmitted and continuing until the capsule passes out of the range of the Bermuda radars.

5.6.3 The Teletype Output Control Cycle

The Teletype output cycle begins when the Bermuda Output Computations Program, BOCOMP, prepares the orbital parameters—the position and velocity vectors. The Monitor Suffix to BOCOMP checks at each entry a communication cell set by BOCOMP. If the cell specifies that the orbital parameters have been prepared, the suffix turns on the B indicator for the teletype output generator, BOTOUT, (shown as MPTOUT in the Monitor Priority Table, Table 2-1). BOTOUT has a higher ranking in the priority table than the editing, smoothing, short arc and computations programs and therefore, when the B indicator for BOTOUT is turned on, this program receives control in preference to those programs tested. When the prefix MPTOUT to BOTOUT receives control for the first time, control is given immediately to BOTOUT to convert the position and velocity components into the teletype format for transmission to Goddard. BOTOUT exits to MFTOUT when a message frame has been packed. MFTOUT queues an on-line message stating that teletype transmission has begun. MFTOUT then turns on the B indicator for the teletype transmission processor, MYTTOT.

*Raw radar data is also transmitted to Goddard from both radars, but this transmission is completely independent of the computer programs.

MYTTOT moves the packed teletype data from the BOTOUT output table into a temporary buffer, unpacks one word into the output block for DCC subchannel 9, activates the subchannel and initiates transmission of the message to Goddard. MYTTOT then turns on the B indicator for BOTOUT to generate more teletype output while the present message is being unpacked and transmitted. After the six characters have been transmitted, the resulting trap gives control to the teletype output trap processor, MTTTOT which unpacks another word and refills the subchannel 9 output block. (The speed of the IBM 709 computer programs relative to the transmission speed for teletype eliminates the need of deactivating subchannel 9 after each six characters are transmitted. The output block is refilled before the first word is retransmitted.)

After the initial entry to MTTTOT, it is the only program concerned with the present teletype output frame. Each time six characters are transmitted, a trap gives control to MTTTOT which unpacks another six for transmission. Meanwhile, BOTOUT is converting and packing more output to refill its output table. When MTTTOT finishes transmitting the frame complete MYTTOT which suppressed itself at its entry, is now unsuppressed and receives control. MYTTOT once again unpacks the first word, but does not activate the subchannel since it was not deactivated. MTTTOT once again is entered after the first six characters are transmitted and continues to unpack the message for transmission.

BOTOUT exits to MFTOUT at that entry location initially and each time thereafter until one type of message has been completed. When the first message—the orbit parameters—has been packed, BOTOUT enters its suffix at the MFTTOA location. The next entry to BOTOUT is for AN/FPS-16 smooth data and finally for Verlort. The AN/FPS-16 and Verlort entries alternate until the capsule completes its transit of the Bermuda tracking radars. After completing each AN/FPS-16 smooth data message, BOTOUT enters its suffix at the entry point MFTTOB; after completing each Verlort smooth data message, BOTOUT enters its suffix at the entry point MFTTOC. MPTOUT, MFTOUT, MYTTOT and MTTTOT are described in detail in subsections 5.7 through 5.10. BOTOUT is described in detail in MC 106, Bermuda Processing Programs.

5.7 MONITOR PREFIX TO BOTOUT (MPTOUT)

MPTOUT receives control from M0PRIO and determines if any data remains to be processed by the teletype output generator, BOTOUT. MPTOUT sets indicators which reflect the processing conditions and, if more output is required of BOTOUT, returns control to this program.

The flow chart for the MPTOUT prefix is shown in Figure 5-6.

5.7.1 Input Requirements

Input to MPTOUT consists of:

a) MCFRVI—communication cell which, when zero, indicates the orbit parameters have not yet been sent to Goddard.

b) MCTTOD—communication cell which, when non-zero, indicates AN/FPS-16 data has been transmitted to Goddard.

c) MCTTOE—communication cell which, when non-zero, indicates Verlort data has been sent to Goddard.

d) MCTTOF—communication cell which, when non-zero, indicates that BOTOUT is in progress, i.e., must process more data to complete the present teletype message.

e) BTF07P (BTV07P)—table containing the seven-point smoothed radar data for AN/FPS-16 (Verlort). The second cell of this table is examined to determine whether or not the table contains data.

f) K00001—constant, 1.

g) K00002—constant, 2.

h) Macros used—TRNOF and TRNON.

5.7.2 Output Requirements

Output from MPTOUT consists of:

a) BTRADR—contains a constant which indicates to BOTOUT which radar data is to be packed for transmission. A 2 indicates AN/FPS-16; a 1 indicates Verlort.

b) MCTTOC—communication cell set non-zero when teletype output processing is complete for the current pass.

c) MCTTOD—communication cell reset to zero when teletype output of AN/FPS-16 data is complete for the current pass.

d) MCTTOE—communication cell reset to zero when teletype output of Verlort data is complete for the current pass.

5.7.3 Method

MPTOUT receives control from M0PRIO and turns the A indicator on and the B indicator off for BOTOUT. The C indicator for MYTTOT is turned on to suppress the teletype output transmission processor while BOTOUT is generating the output message in teletype format.

If MPTOUT finds that BOTOUT is already engaged in processing a message (part of which has just been transmitted), MPTOUT immediately returns control to BOTOUT. Otherwise, MPTOUT tests MCFRVI to determine whether or not the orbit parameters have been transmitted. (MPTOUT is not entered until orbit parameters have been prepared and the orbit parameters are not prepared until smoothed data is available from at least one radar.) If the orbit parameters have not been transmitted, MPTOUT immediately gives control to BOTOUT. Otherwise, MPTOUT examines the AN/FPS-16 seven-point smoothing table to determine if data is available. If data is present, MPTOUT sets an indicator for AN/FPS-16 and enters BOTOUT. If there is no seven-point smoothed AN/FPS-16 data available, MPTOUT examines the corresponding table for Verlort. If data is present, MPTOUT sets the indicator for Verlort and enters BOTOUT.

If all of the above tests are positive—BOTOUT is not in progress, the orbit parameters have been sent, and seven-point smoothed data from both AN/FPS-16 and Verlort has been sent—MPTOUT resets the AN/FPS-16 and Verlort indicators to zero and sets the teletype output indicator to show that teletype processing is complete for this pass. MCFRVI is not reset since orbital parameters are sent once only. MPTOUT turns off the C and A indicators for BOTOUT and returns control to M0PRIO.

5.7.4 Usage

MPTOUT is entered from M0PRIO: MPTOUT exits to BOTOUT or to M0PRIO (when all of the available data has been packed for transmission).

a) Storage Required—32 locations.

b) MPTOUT uses:

1) Macros—TRNOF and TRNON

2) Parameters—A, B, C, MNTOUT and MNTTOT

3) Communication cells—MCFRVI, MCTTOC, MCTTOD, MCTTOE and MCTTOF

4) Tables—BTRADR, BTF07P and BTV07P

c) Time Required:

1) Minimum—0.24 millisecond

2) Maximum—0.60 millisecond

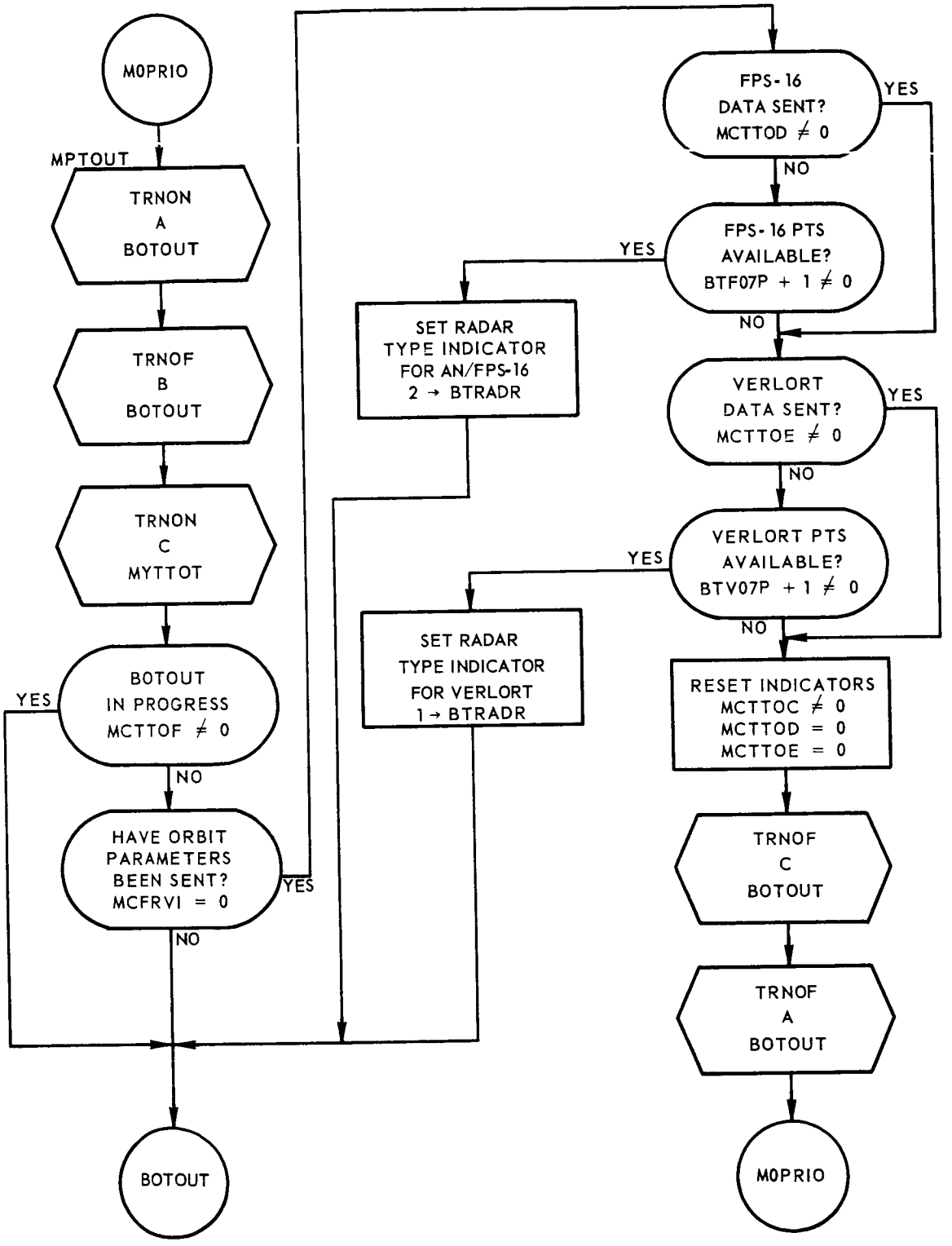


FIGURE 5-6. MPTOUT PROGRAM FLOW CHART

5.8 MONITOR SUFFIX TO BOTOUT (MFTOUT)

MFTOUT receives control from the teletype output generator, BOTOUT, sets indicators for entry to the teletype transmission processor, MYTTOT, and resets indicators for BOTOUT.

The flow chart for the MFTOUT suffix is shown in Figure 5-7.

5.8.1 Input Requirements

Input to MFTOUT consists of:

a) FTYT1—internal cell which, when non-zero, indicates an on-line message has been printed stating that the teletype transmission has begun. This cell is initially zero, and after the message has been queued on the first entry, is set non-zero.

b) Macros used—QENBA, QENBZ, QUEUE, TRNOF and TRNON.

There are four different entry points for MFTOUT for the four different types of exits from BOTOUT. These different entries effectively serve as input requirements: entry after one message frame packed, MFTOUT; entry after orbit parameters packed, MFTTOA; entry after AN/FPS-16 seven-point smoothed data packed, MFTTOB; entry after Verlort seven-point smoothed data packed, data packed, MFTTOB; entry after Verlort seven-point smoothed data packed, MFTTOC.

5.8.2 Output Requirements

Output requirements depend upon the type of entry to MFTOUT:

a) MFTOUT entry—MCTTOF is set non-zero to indicate more of this particular message remains to be processed by BOTOUT; the message "TTY TRANSMISSION TO GODDARD HAS BEGUN" is queued and FTYT1 is set non-zero to show the message has been queued.

b) MFTTOA entry—zero is stored in MCFRVI to indicate the orbit parameters have been sent; the cell indicating BOTOUT has completed this particular message, MCTTOF, is set to zero; the message "ORBIT DATA HAS BEEN TRANSMITTED TO GODDARD" is queued.

c) MFTTOB entry—zero is stored in MCTTOD to indicate AN/FPS-16 data has been completely packed for transmission; zero is stored in MCTTOF to indicate the BOTOUT has completed processing this particular message; the on-line message "FPS-16 SMOOTHED DATA HAS BEEN TRANSMITTED TO GODDARD" is queued.

d) MFTTOC entry—zero is stored in MCTTOE to indicate Verlort data has been completely packed for transmission; zero is stored in

MCTTOF to indicate that BOTOUT has completed processing this particular message; the on-line message "VERLORT SMOOTHED DATA HAS BEEN TRANSMITTED TO GODDARD" is queued.

For all entries, the B indicator for MYTTOT is turned on and the C indicator for MYTTOT is turned off, the C indicator for BOTOUT is turned on and the A indicator for BOTOUT is turned off.

5.8.3 Method

When MFTOUT is entered from BOTOUT, the location of the entry determines the condition of BOTOUT processing. If the entry is to the location MFTOUT, the entry is normal, i.e., the orbit parameters or radar data have been packed, but the message has not been completed. The in-progress indicator is set, MYTTOT is unsuppressed and requested transmitted that part of message which is complete, and BOTOUT is suppressed. Control returns to M0PRIO.

If the entry is to some location other than MFTOUT, the entry follows the processing of a complete message by BOTOUT. MFTOUT sets indicators to show the type of data available for transmission and that BOTOUT has completed processing of the particular type of message.

For the first entry to the location MFTOUT or on every entry to any other location, an on-line message specifies the processing done by BOTOUT.

5.8.4 Usage

MFTOUT is entered from BOTOUT (at any one of four different entry points); MFTOUT exits to M0PRIO.

a) Storage Required—45 locations

b) MFTOUT Uses:

1) Macros—QENBA, QENBZ, QUEUE, TRNOF and TRNON

2) Parameters—A, B, C, MNMESS, MNTOUT and MNTTOT

3) Communication cells—MCFRVI, MCTTOD, MCTTOE and MCTTOF

4) Internal cell—FTYT1

c) Time Required:

1) First entry MFTOUT—0.98 millisecond

2) Succeeding entries at MFTOUT—0.39 millisecond

3) Entry to MCTTOA, MFTTOB or MFTTOC—0.90 millisecond

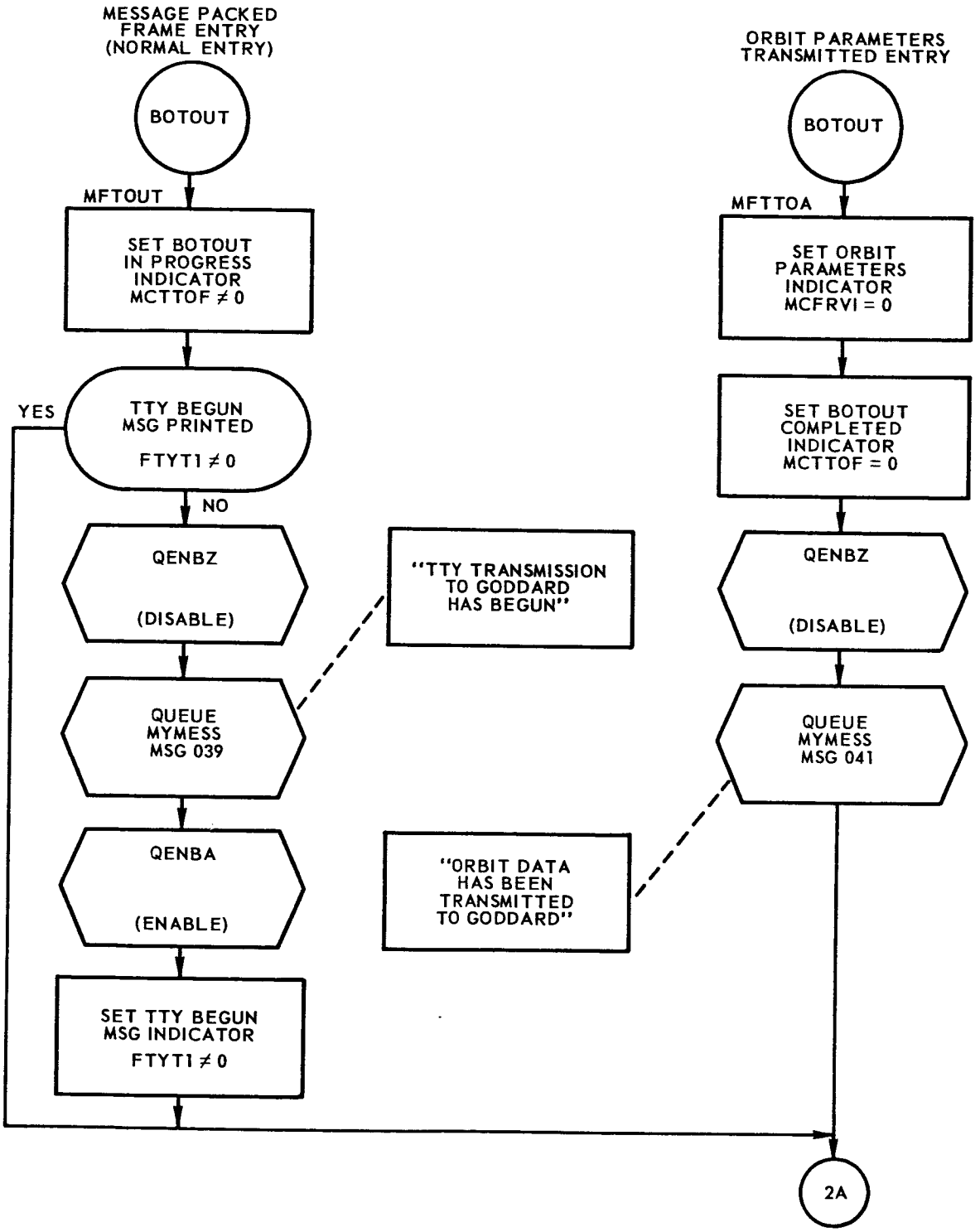


FIGURE 5-7. MFTOUT PROGRAM FLOW CHART *(Sheet 1 of 2)*

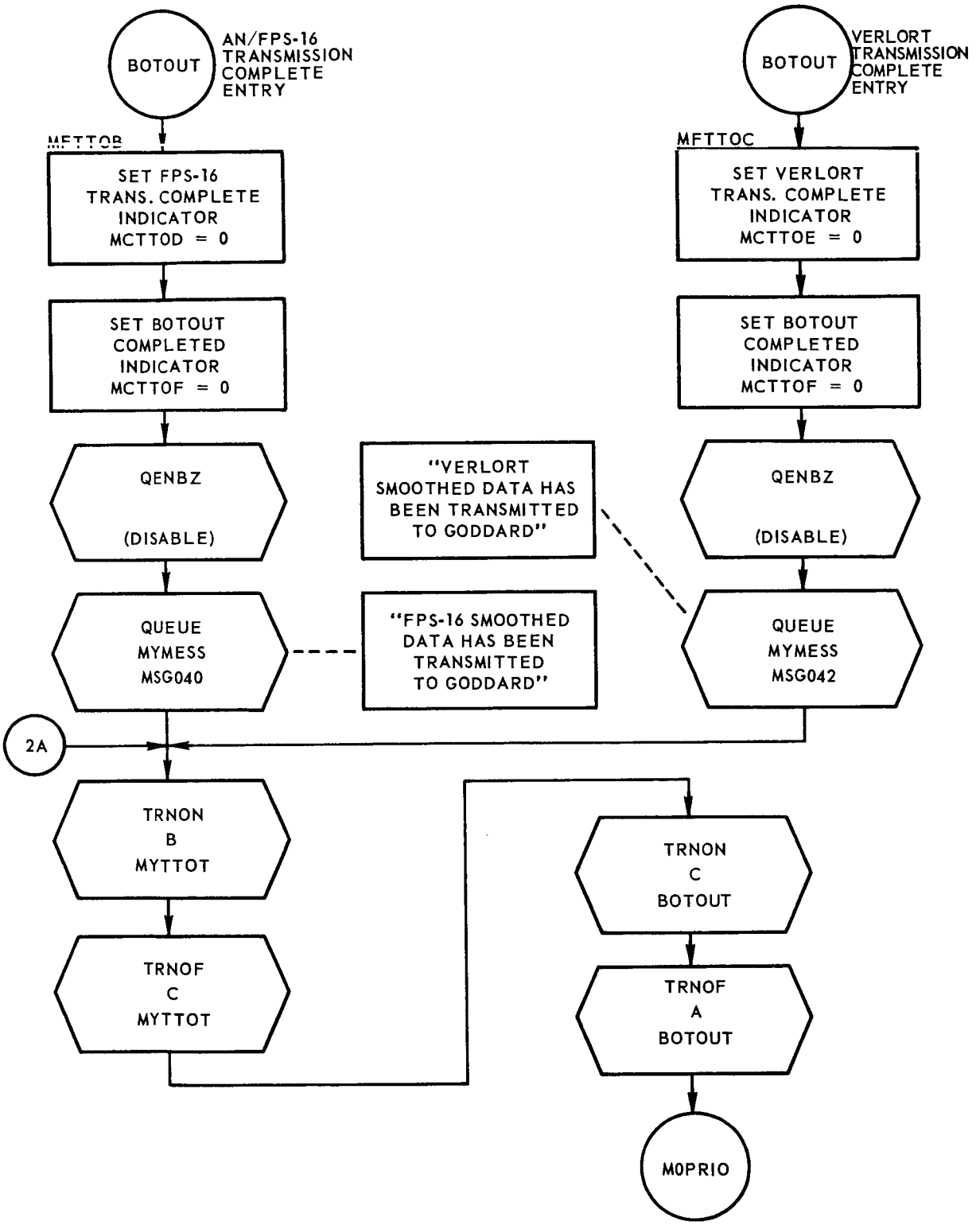


FIGURE 5-7. MFTOUT PROGRAM FLOW CHART *(Sheet 2 of 2)*

5.9 MONITOR TELETYP OUTPUT PROCESSOR (MYTTOT)

MYTTOT moves the packed teletype message from the BOTOUT output table into a temporary storage buffer, unpacks the first word and stores the six teletype characters in the DCC subchannel 9 output block, and generates the time tag for logging. On the initial entry MYTTOT activates subchannel 9 to initiate transmission of the teletype message.

The flow chart for the MYTTOT processor is shown in Figure 5-8.

5.9.1 Input Requirements

Input to MYTTOT consists of:

a) BTVROP—output table for BOTOUT containing the packed teletype message.

b) MCTOUT—communication cell containing in the decrement the number of words of packed teletype output and in the address the last location plus one of the output.

c) MCHFSC—communication cell containing the current GMT in fixed-point half seconds.

d) TM8.3M—one-cell table containing the number of elapsed 8-1/3 millisecond intervals since the last exact half second.

e) CWT1—internal cell which, when zero, indicates the initial entry to MYTTOT.

f) Macros Used—QENBZ, QPSLF, TRNOF and TRNON.

5.9.1 Output Requirements

Output from MYTTOT consists of:

a) TMTTBF—the contents of BTVROP are temporarily stored in TMTTBF, which makes the BSTVROP table available to BOTOUT for more packed teletype output.

b) TMTTOT—six-cell table containing each cell one character from the first word of the packed teletype message.

c) MTTTOT—the number of packed words in the teletype message is stored in the address of the first instruction, i.e., the entry location, of MTTTOT.

d) MCTTO1—communication cell containing the logging time tag for the packed message.

e) MCTT02—communication cell containing in the address the number of words in the packed output message.

Subchannel 9 is activated on the initial entry to MYTTOT.

5.9.3 Method

MYTTOT receives control from M0PRIO and turns its A indicator on and its B indicator off. The packed message is moved from the BOTOUT table, BTVROP, into the temporary storage buffer, TMTTBF, and the first word of the message is unpacked into the DCC subchannel 9 output table, TMTTOT. One character is placed in each cell.

On the initial entry MYTTOT activates subchannel 9 initiating transmission on the six unpacked characters. After transmission has begun MYTTOT sets the initial entry indicator.

On every entry MYTTOT generates a logging time tag and saves it and the packed word count of the message for reference by the Monitor Teletype Output Trap Processor, MTTTOT.

BOTOUT is unsuppressed and may now refill, if necessary, the original input table, BTVROP. The B indicator for BOTOUT is turned on to return control to this processor. The D indicator for MYTTOT is turned on to suppress the teletype output processor until after the complete message is transmitted. The A indicator for MYTTOT is turned off and control returns to M0PRIO.

5.9.4 Usage

MYTTOT is entered from, and exits to, M0PRIO.

a) Storage Required—41 locations

b) MYTTOT uses:

1) Macros—QENBZ, QPSLF, TRNOF and TRNON

2) Parameters—A, B, C, D, MNTOUT and MNTTOT

3) Communication cells—MCACTV, MCHFSC, MCTOUT, MCTT01 and MCTT02

4) Tables—BTVROP, TMTTBF, TMTTOT and TM8.3M

5) Internal cell—CWT1

c) Time Required:

1) Initial entry—1.73 milliseconds

2) Normal entry—1.44 milliseconds

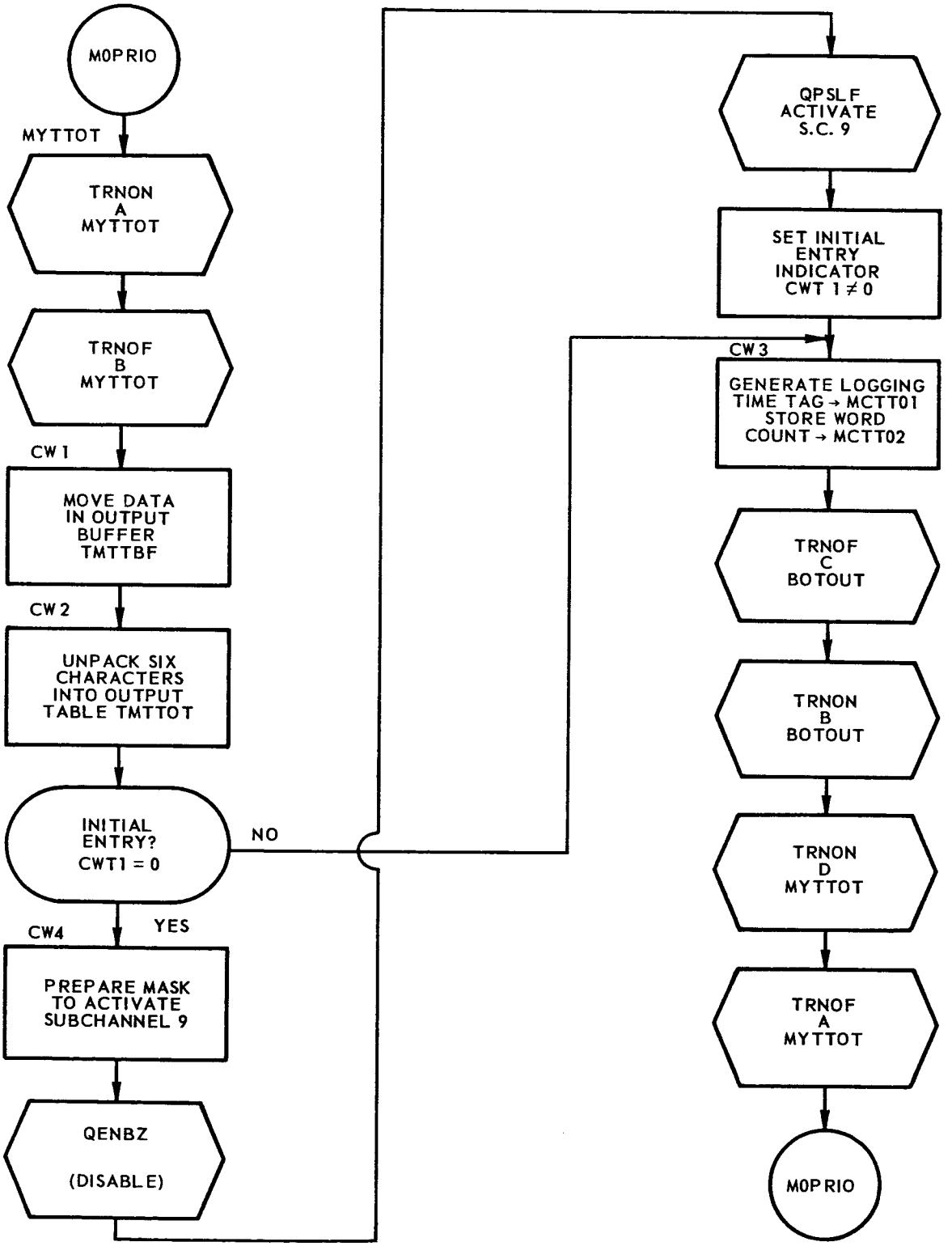


FIGURE 5-8. MYTTOT PROGRAM FLOW CHART

5.10 MONITOR TELETYPE OUTPUT TRAP PROCESSOR (MTTTOT)

When a teletype message is available for transmission, control is given to the Monitor Teletype Output Processor which unpacks the first word of the message and initiates transmission on subchannel 9 of the DCC. The trap following the transmission of the first word (six unpacked teletype characters) gives control to MTTTOT. The Monitor Teletype Output Trap Processor unpacks the second and subsequent words of each teletype message prepared by BOTOUT and transmits these in groups of six characters. When the last word is transmitted, MTTTOT logs the transmitted message. When all teletype transmission is complete, MTTTOT deactivates subchannel 9.

The flow chart for the MTTTOT trap processor is shown in Figure 5-9.

5.10.1 Input Requirements

Input to MTTTOT consists of:

a) MTTTOT—the entry location of the trap processor contains in the address the word count of the message.

b) TMTTBF—table containing the packed teletype message.

c) MCTT01—communication cell containing the time tag for logging the message after transmission is complete.

d) MCTTOC—communication cell which, when non-zero, indicates that all teletype transmission is complete for this pass.

e) Subroutine Used—MSLOGG.

f) Macros Used—QPSLF, QUEUE, and TRNOF.

5.10.2 Output Requirements

Output from MTTTOT consists of:

a) TMTTOT—six teletype characters are unpacked and stored in the DCC subchannel 9 output block.

b) MTTTOT—the word count in the address is decremented by one.

c) MCTTOC—communication cell is reset to zero on the final entry for each pass.

5.10.3 Method

The word count in the address of MTTTOT is examined to see if any words remain in TMTTBF. If there are more words to be sent for this message, one word is unpacked and placed in TMTTOT.

If no words remain, MCTTOC is examined to see whether or not any data remains to be sent for this pass. If all the data has been sent, subchannel 9 of the DCC is deactivated and an on-line message is queued which states that teletype output is complete for this pass. In either event the data which has been transmitted is logged in packed form and MYTTOT is unsuppressed.

5.10.4 Usage

MTTTOT is entered from M0DCCC following a trap on DCC subchannel 9; MTTTOT exits to M0PRIO.

a) Storage Required—33 locations

b) MTTTOT uses:

1) Subroutines—MSLOGG

2) Macros—QPSLF, QUEUE and TRNOF

3) Parameters—MNMESS and MNTTOT

4) Communication cells—MCTT01 and MCTTOC

5) Tables—TMTTBF and TMTTOT

c) Time Required:

1) Normal entry—3.43 milliseconds

2) Entry at completion of a message—3.07 milliseconds

3) Entry at completion of a pass—3.73 milliseconds

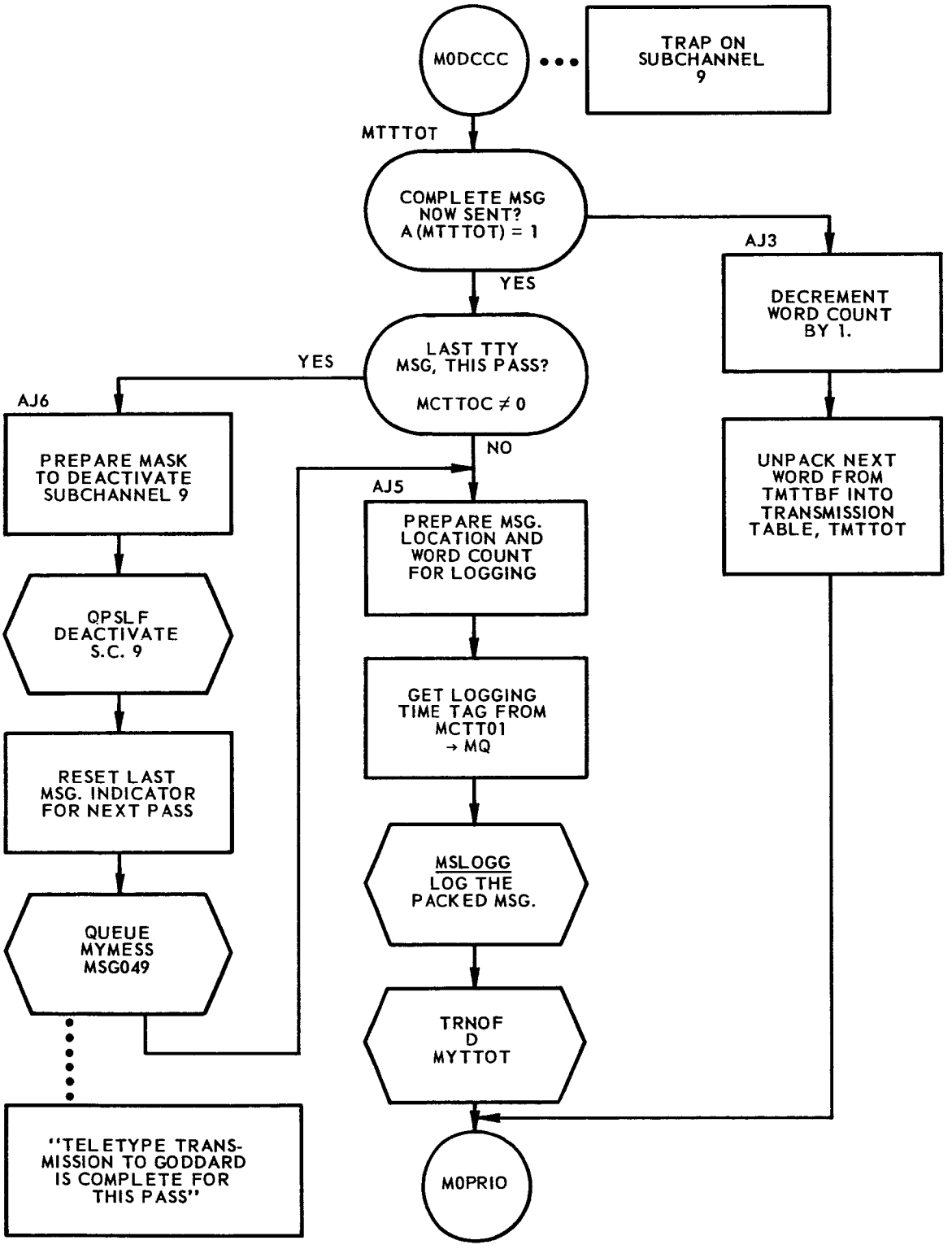


FIGURE 5-9. MTTTOT PROGRAM FLOW CHART

5.11 SENSE OUTPUT TO THE OUTPUT STATUS CONSOLE

The Output Status Console provides eight indicator lights which are used to provide external indications of mission flight phases, quality of input radar data and computer timing control. These eight lights are controlled by a fixed location in computer core storage (TMSENS) whenever DCC subchannel 12 is activated. A 1 in bit positions 28 through 35 of TMSENS turns on the corresponding indicators; a 1 in bit positions 10 through 17 turns off the corresponding indicators. If both bits for a given indicator contain 1, the off bit has precedence.

The function of the Output Status Console's sense indicator lights is illustrated below:

LAUNCH	ABORT	HOLD	ORBIT	RE-ENTRY	POINTS	BAD POINTS VERLORT	WWV ERROR

Whenever any program recognizes the need for change in the sense indicator lights, a mask, with a 1 in the proper bit position for the lights to be turned on or off, is stored in the sense output processor's queue table and the B indicator for MYSENS is turned on.

The launch abort, hold, orbit and re-entry lights are set on request from MSETPH (see subsection 4.11), a subroutine used exclusively by the prefix to BOCOMP, MPCOMP. The bad points lights are set on request from the bad-points suffix to BEDITS, MFEBAD (see subsection 4.4), and are turned off on request from the bad-points processor, MYEBAD. The "WWV Error" light is controlled by WWV trap processor, MTWWWV (see subsection 3.16).

5.12 MONITOR SENSE OUTPUT PROCESSOR (MYSENS)

MYSENS extracts the mask from the queue table, adjusts the bit configuration of TMSENS to reflect the changes generates a time tag for logging, and activates DCC subchannel 12 to initiate transmission.

The flow chart for the MYSENS processor is shown in Figure 5-10.

5.12.1 Input Requirements

Input to MYSENS consists of:

a) TMSENS—contains a 1 in bit positions which indicate the current setting of the indicator lights.

b) Q18—the location of the queue table for TMSENS. The first entry contains a 1 in bit positions of those indicator lights which are to be changed.

c) MCHFSC—communication cell containing the GMT in fixed-point half seconds.

d) TM8.3M—one-cell table containing the number of elapsed 8-1/3 millisecond intervals since the last exact half second

e) MCSEN2—mask used to deactivate or, when complemented, activate subchannel 12.

f) Macros used—QENBA, QENBZ, QPSLF, TRNOF, TRNON and UNQUE.

5.12.2 Output Requirements

Output from MYSENS consists of:

a) TMSENS—this cell is adjusted to reflect the requested changes in the indicator lights.

b) MCSEN1—communication cell containing the logging time tag.

c) MCACTV—a 1 is stored in the bit position for subchannel 12 of the communication cell which controls the activation of the DCC subchannels.

5.12.3 Method

MYSENS disables the computer, extracts the input mask for the queue table, and enables the computer. The input mask is used to reset the bit configuration in the sense output cell, TMSENS. The time tag for logging is generated and

stored for use by the Monitor Sense Output Trap Processor, MTSENS. The PSLF mask is adjusted to activate subchannel 12 of the DCC, the computer is disabled and subchannel 12 is activated to initiate transmission. MYSENS suppresses itself to prevent entry until this transmission is complete and returns control to M0PRIO.

5.12.4 Usage

MYSENS is entered from, and exits to, M0PRIO.

a) Storage Required—32 locations.

b) MYSENS uses:

1) Macros—QENBA, QENBZ, QPSLF, TRNOF and TRNON

2) Parameters—A, C and MNSENS

3) Communication cells—MCACTV, MCSEN1 and MCSEN2

4) Tables—Q18, TMSENS and TM8.3M

c) Time Required—1.16 milliseconds

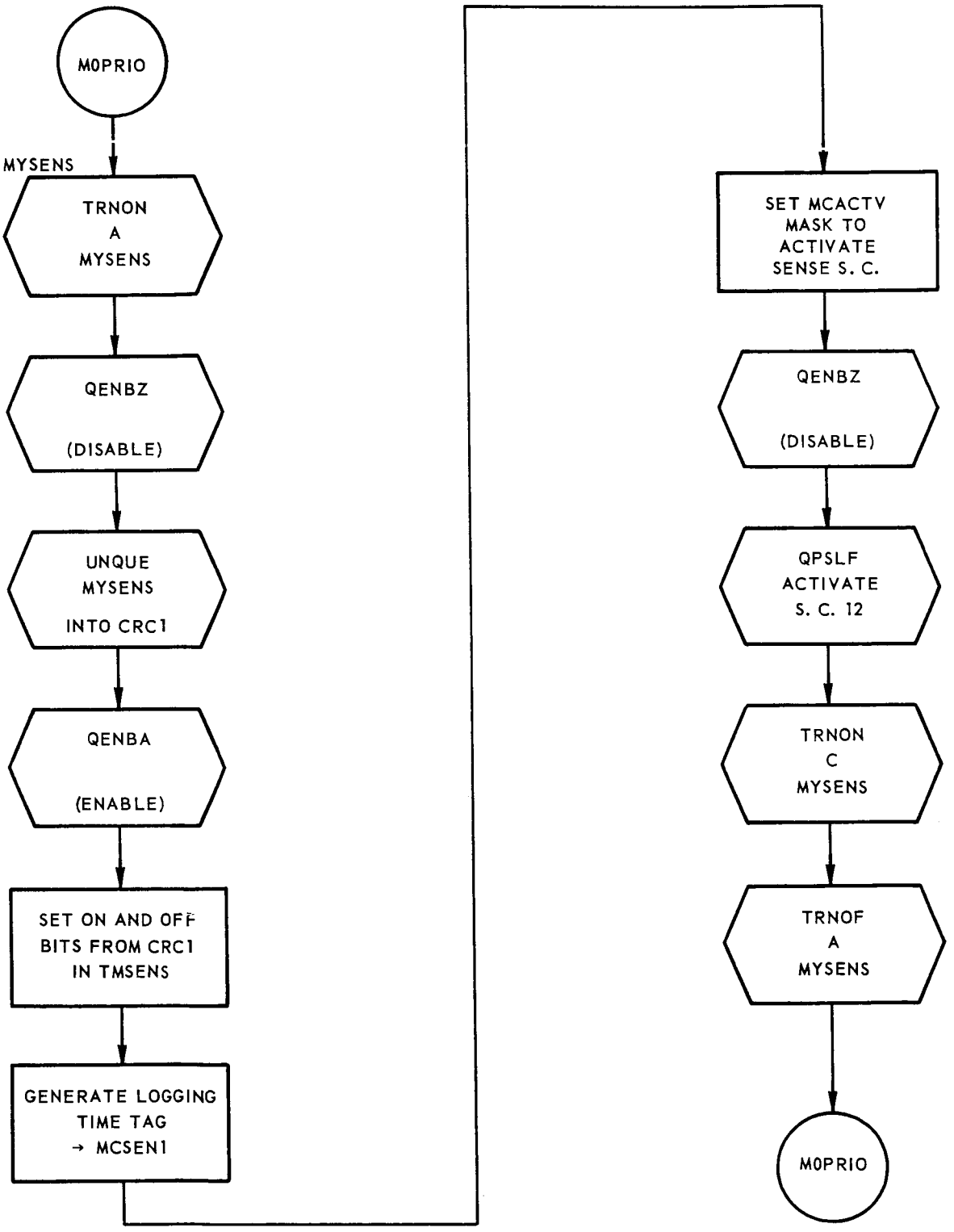


FIGURE 5-10. MYSENS PROGRAM FLOW CHART

5.13 MONITOR SENSE OUTPUT TRAP PROCESSOR (MTSENS)

MTSENS receives control on the trap which follows the transmission by MYSENS. MTSENS deactivates subchannel 12, logs the sense output transmission and unsuppresses MYSENS.

The flow chart for the MTSENS trap processor is shown in Figure 5-

5.12.1 Input Requirements

Input to MTSENS consists of:

a) TMSENS—one-cell table containing the output transmission.

b) MCSEN2—mask used to activate subchannel 12.

c) MCSEN1—contains the logging time tag generated by MYSENS

d) Subroutine used—MSLOGG.

e) Macro Used—QPSLF and TRNOF.

5.13.2 Output Requirements

MTSENS deactivates subchannel 12, logs the sense output transmission and unsuppresses MYSENS.

5.13.3 Method

MTSENS adjusts the PSLF mask and executes the QPSLF macro to deactivate the subchannel 12. The logging time tag is loaded into the MQ and the logging subroutine, MSLOGG, is used to log the output from TMSENS. MTSENS unsuppresses the sense output processor, MYSENS, and transfer program control to M0PRIO.

5.13.4 Usage

MTSENS is entered from M0DCCC following a trap on DCC subchannel 12; MTSENS exits to M0PRIO.

a) Storage Required—nine locations

b) MTSENS uses:

1) Subroutine—MSLOGG

2) Macro—QPSLF and TRNOF

3) Parameters—C and MNSENS

4) Communication cells—MCSEN1 and MCSEN2

5) Table—TMSENS

c) Time Required—2.95 milliseconds

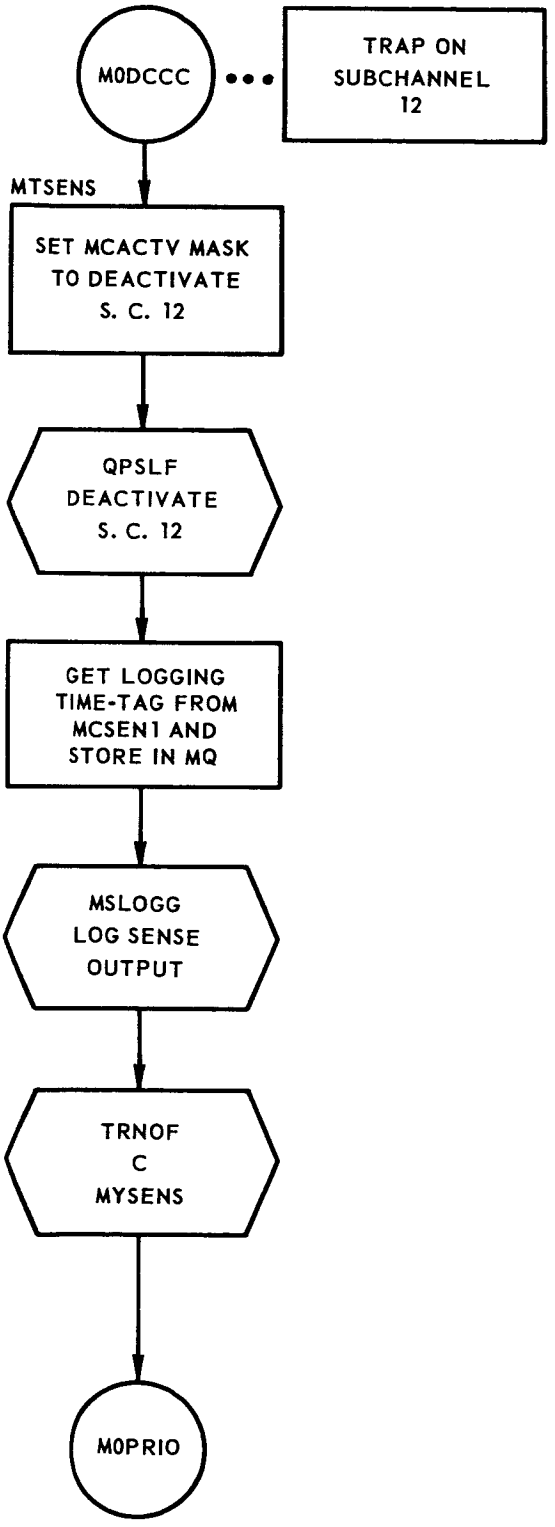


FIGURE 5-11. MTSENS PROGRAM FLOW CHART

Section 6.

CONTROL OF CHANNELS A AND B

To provide maximum versatility in the operational tracking program, the IBM 709 Data Channels A and B are utilized to perform the following functions:

a) Read the Station Characteristics tape into computer core storage during the pre-launch initialization of the Mercury Program System.

b) Provide the Mercury Program System with prepared messages for on-line printing throughout the mission.

c) Provide for the inclusion of on-line card input during the mission. (At present, this feature is not used.)

d) Provide a permanent log tape record for post-flight and other analyses of all on-line messages and all input/output data transmitted via the Data Communications Channel.

To promote maximum efficiency of input/output data transmission and to ensure minimum delay of the central processing unit, no data channel commands are given which would logically disconnect an I/O device at the termination of transmission without an accompanying data channel trap. This requires two programs for every I/O transmission process: the monitor processor (MYXXXX) which initiates the transmission and the monitor trap processor (MTXXXX) which services the trap following the completion of transmission.

When two or more transmission processors share the same data channel, certain restrictions must be observed. Prior to initiating transmission the processor must:

a) Suppress every other processor which may initiate transmission over the same data channel.

b) Set the lower core trap control location with a transfer instruction addressed to the trap processor appropriate for the data transmitted.

c) Suppress itself, if necessary to prevent successive entry and transmission, until after the trap processor has signaled the completion of the current transmission cycle.

6.1 STATION CHARACTERISTICS TAPE READING CYCLE

6.1.1 Composition of the Station Characteristics Tape

The Station Characteristics tape supplies geological and meteorological data used in the computations of the Bermuda Processor Programs. This tape is composed of one file containing four records. The first is a tape identification record and the remaining three records define the three types of stations (capsule tracking devices) at Bermuda: the second record contains the characteristics of the AN/FPS-16 radar, the third record contains the characteristics of the Verlort radar, and the fourth record contains the characteristics of the telemetry receivers. The three station's characteristics on the tape contain special error correction (Hamming) codes which permit the detection and correction of writing, transmission and reading errors.

The Station Characteristic tape is updated in the pre-launch countdown procedure and is read into core storage during the initialization of the Mercury Program System. Once read into core memory, the station characteristics become part of the IBM 709's stored knowledge of the Bermuda stations and, excepting a restart or other emergency condition, the tape is no longer used throughout the mission. The programs which generate (U0STCH) and update (U0STUP) the Stations Characteristics tape and specific data contained in the Station Characteristic block are described in MC 107, External System Programs. The process whereby the Station Characteristic tape is read into computer core storage is described in the following subsection.

6.1.2 Monitor Control of the Station Characteristics Tape

When the Mercury System Tape has been read successfully into the IBM 709 core storage, initialization of the Mercury Program System begins (see Section 7). After the GMT time base has been entered into the system, the Monitor Initialization Processor, MYINTL, turns on the B indicator for the Monitor Station Characteristics Tape Reader Processor, MYSCRD. MYINTL loops in an examination of the location TMSTCH for the indication that the Station Characteristics tape has been read successfully. The next one-minute WWV signal interrupts MYINTL and M0PRIO gives control to the higher priority program, MYSCRD. When MYSCRD receives control, the Station Characteristics tape reading cycle begins. This cycle is illustrated in Figure 6-1.

On the first entry MYSCRD initiates the reading of the first Station Characteristics record from tape into a temporary storage block in core. Control returns via M0PRIO to the interrupted processor, MYINTL, which continues to loop examining TMSTCH. When the transmission is completed, the Monitor Station Characteristics Tape Reader Trap Processor, MTENST, receives control. MTENST checks for errors in reading the tape and, if no errors occurred, turns on the B indicator for the next entry into MYSCRD.

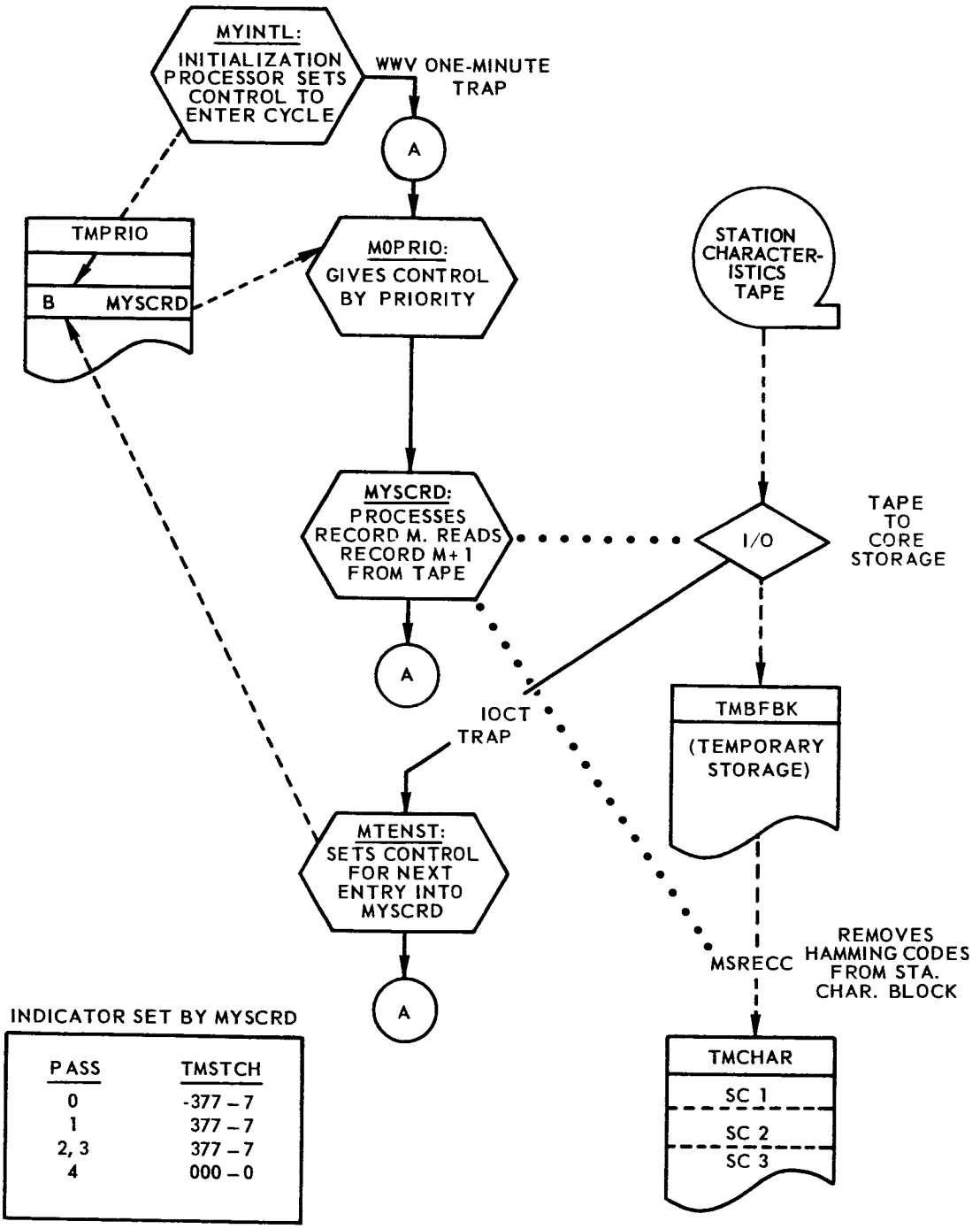


FIGURE 6-1. STATION CHARACTERISTICS TAPE READING CYCLE.

M0PRIO again gives control to MYSCRD where the record read is used as input to MSRECC, the monitor subroutine which examines the record and removes the error correction codes. After executing the subroutine, MYSCRD moves the corrected station block into the permanent Station Characteristics table in core storage. MYSCRD initiates transmission to read the second Station Characteristics record from the tape into the temporary storage block. MYSCRD again exits to M0PRIO. When transmission of the record is completed, the trap gives control to the trap processor MTENST.

The cycle continues until the last Station Characteristics block is read from tape. On the last entry to MYSCRD the final Station Characteristics block is processed and placed in the permanent table. MYSCRD now clears TMSTCH, queues the on-line message cycle to print a message indicating the Station Characteristics tape has been read successfully, and transfers control to M0PRIO. When MYINTL receives control, it finds TMSTCH equal to zero, and completes the initialization of the Mercury Program System.

The two processors which read the Station Characteristics tape are described in Subsections 6.2 and 6.3.

6.2 MONITOR STATION CHARACTERISTICS TAPE READER PROCESSOR (MYSCRD)

MYSCRD functions only during the initialization phase of the Mercury Program System. This processor is used in conjunction with its trap processor, MTENST, to read into core storage the Station Characteristics from the Station Characteristics Tape. MYSCRD initiates transmission of one record into core storage. When transmission is completed a trap occurs and MTENST receives control and checks for errors in transmission. MYSCRD receives control again, processes the record read, and, if more records remain on the Station Characteristics Tape, the processor initiates transmission of the next record. The cycle continues until all records have been read and processed, or until an error interrupts the cycle.

The flow chart for the MYSCRD processor is shown in Figure 6-2.

6.2.1 Input Requirements

Input to MYSCRD consists of:

a) MUSCTP—parameter designating the unit tape address of the Station Characteristics Tape, (A 7), see Subsection 6.1.

b) MNCHAR—parameter designating the number of words (length) in a Station Characteristics block, 34.

c) MNSCNO—parameter designating the number of Station Characteristics blocks on the tape to be read into core storage, 3. The first record on the tape is an identification record and each of the remaining records comprises one Station Characteristics block.

d) TMSTCH—a four cell table with the following format:

```
TMSTCH    OCT    -377 777 777 777
   +1     PZE    TMCHAR - 1
   +2     PZE    TMCHAR + MNCHAR - 1
   +3     PZE    TMCHAR + 2*MNCHAR - 1
```

The first cell is used as an internal indicator for MYSCRD and as a communication cell by MYINTL. Initially, TMSTCH has the value shown above. In the first pass through MYSCRD, the sign bit of TMSTCH is set positive. During the last pass through MYSCRD, TMSTCH is set to zero.

TMSTCH + 1 contains the location for storing the Station Characteristics block which references the Bermuda AN/FPS-16 radar. TMSTCH + 2 contains the location for storing the Station Characteristics block which references the Bermuda Verlort radar. TMSTCH + 3 contains

the location for storing the Station Characteristics block which references the Bermuda telemetry receivers.

e) Macros Used—QUEUE, QENBA, QENBZ, TRNON and TRNOF.

6.2.2 Output Requirements

Output from MYSCRD depends primarily upon the number of the pass through MYSCRD (each pass is a separate entry into the program):

a) First pass—Sign of TMSTCH set positive. Transmission of AN/FPS-16 Station Characteristics block into core is initiated.

b) Second pass—AN/FPS-16 Station Characteristics block is processed to remove error correction codes. The block is moved into the TMCHAR table. Transmission of Verlort Station Characteristics block into core is initiated.

c) Third pass—Verlort Station Characteristics block is processed to remove error correction codes. The block is moved into the TMCHAR table. Transmission of the telemetry Station Characteristic block into core is initiated.

d) Fourth (last) pass—telemetry Station Characteristic block is processed to remove error correction codes. The block is moved into the TMCHAR table. TMSTCH is set to zero. The on-line message routine is queued to print "STATION CHARACTERISTICS HAS BEEN READ SUCCESSFULLY".

For every pass, the instruction TTR MTENST is stored in the channel A trap control location, 00013.

6.2.3 Method

MYSCRD receives control from M0PRIO and turns its own A indicator on and B indicator off. MYSCRD then suppress the two other programs, MYMESS and MYCDRD, which transmit on channel A. The instruction TTR MTENST is stored in the channel A trap control location, 00013.

a) On the first pass through MYSCRD the sign of TMSTCH is set positive, the Station Characteristics tape, A7, is rewound and the commands and instructions are given to space past the first record and read the second record (the first Station Characteristics block) into a temporary storage buffer in core storage. MYSCRD turns off its A indicator and exits to M0PRIO.

b) On the second pass MYSCRD uses the Error Correction Subroutine, MSRECC, to check the record just read into core storage. If an

uncorrectable error is found, control transfers to the Error Subroutine, MSUERR. (In the event of error detection by MSRECC, the Mercury Program System must be reloaded as explained in section 7.) If no errors are detected, the Station Characteristics block is moved into its permanent location in the TMCHAR table and the command and instruction are given to read the third record into the temporary buffer. MYSCRD turns off its A indicator and exits to M0PRIO.

c) The third pass has the same form as the second, except that the third record is moved into TMCHAR and transmission is initiated for reading the last record.

d) On the fourth (and last) pass the last Station Characteristics block is moved into TMCHAR, and the control word TMSTCH is set to zero. MYSCRD disables the computer, queues the on-line message program with the message STATION CHARACTERISTICS TAPE HAS BEEN READ SUCCESSFULLY, and enables the computer. The suppression bits set by MYSCRD for MYMESS and MYSCRD are turned off, the A indicator for MYSCRD is turned off and control returns to M0PRIO.

6.2.4 Usage

MYSCRD is entered from M0PRIO; MYSCRD exits to M0PRIO or, if an error occurs, to MSUERR.

a) Storage Required—60 locations

b) MYSCRD uses:

1) Subroutines—MSRECC

2) Macros—QUEUE, QENBA, QENBZ, TRNOF and TRNON

3) Parameters—A, B, E, MNCDRD, MNCHAR, MNMESS, MNSCRD, MNSCNO and MUSCTP

4) Tables—TMBFBK, TMCHAR and TMSTCH

5) Constant—K00001

c) Time Required:

1) First pass—22.8 milliseconds

2) Last pass—26.9 milliseconds

3) Any other pass—26.4 milliseconds

Most of the time—approximately 22.7 milliseconds—is taken by MSRECC. However, time is not extremely critical for MYSCRD since the processor functions only in the pre-launch phase.

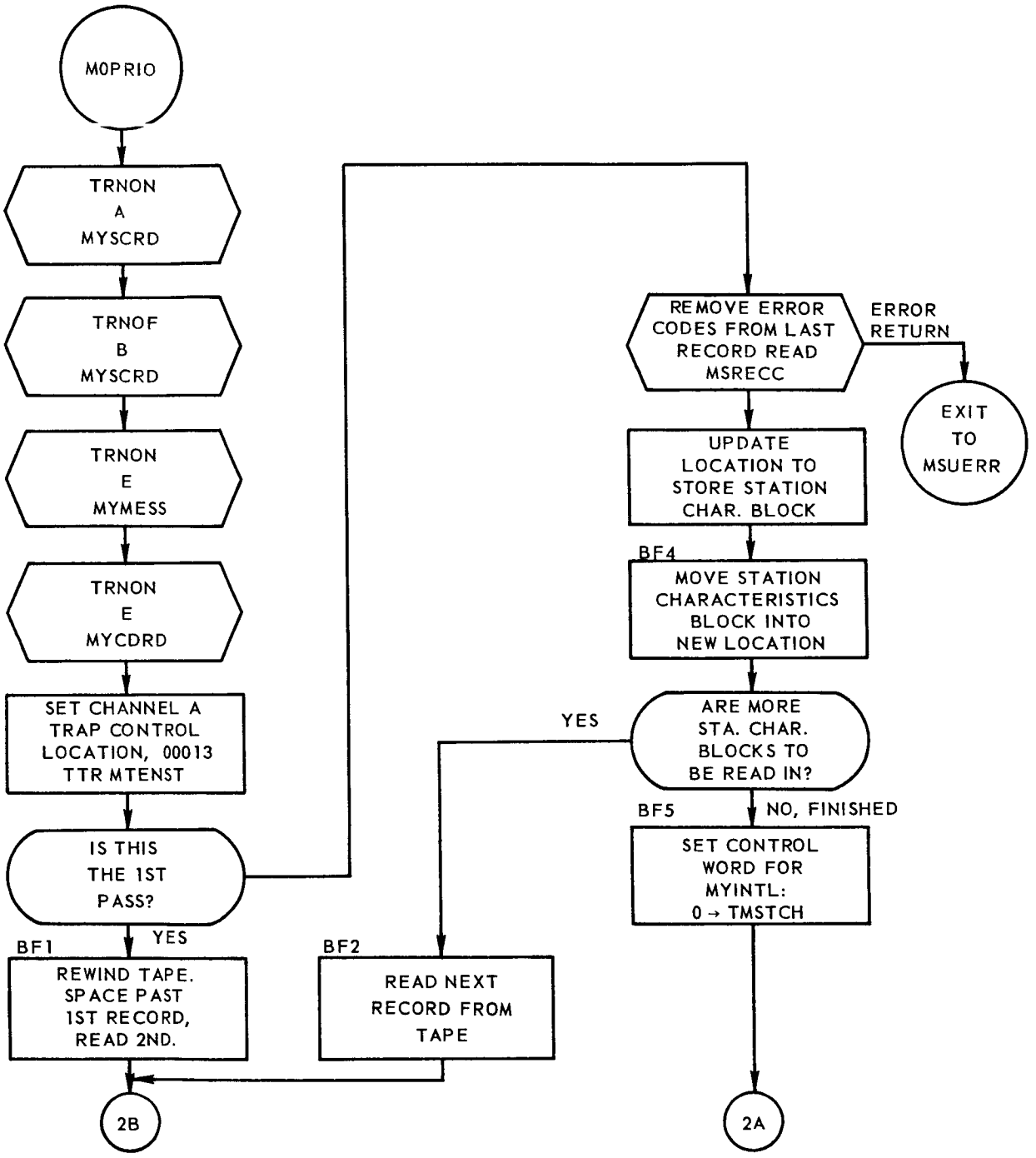


FIGURE 6-2. MYSCRD PROGRAM FLOW CHART *(Sheet 1 of 2)*

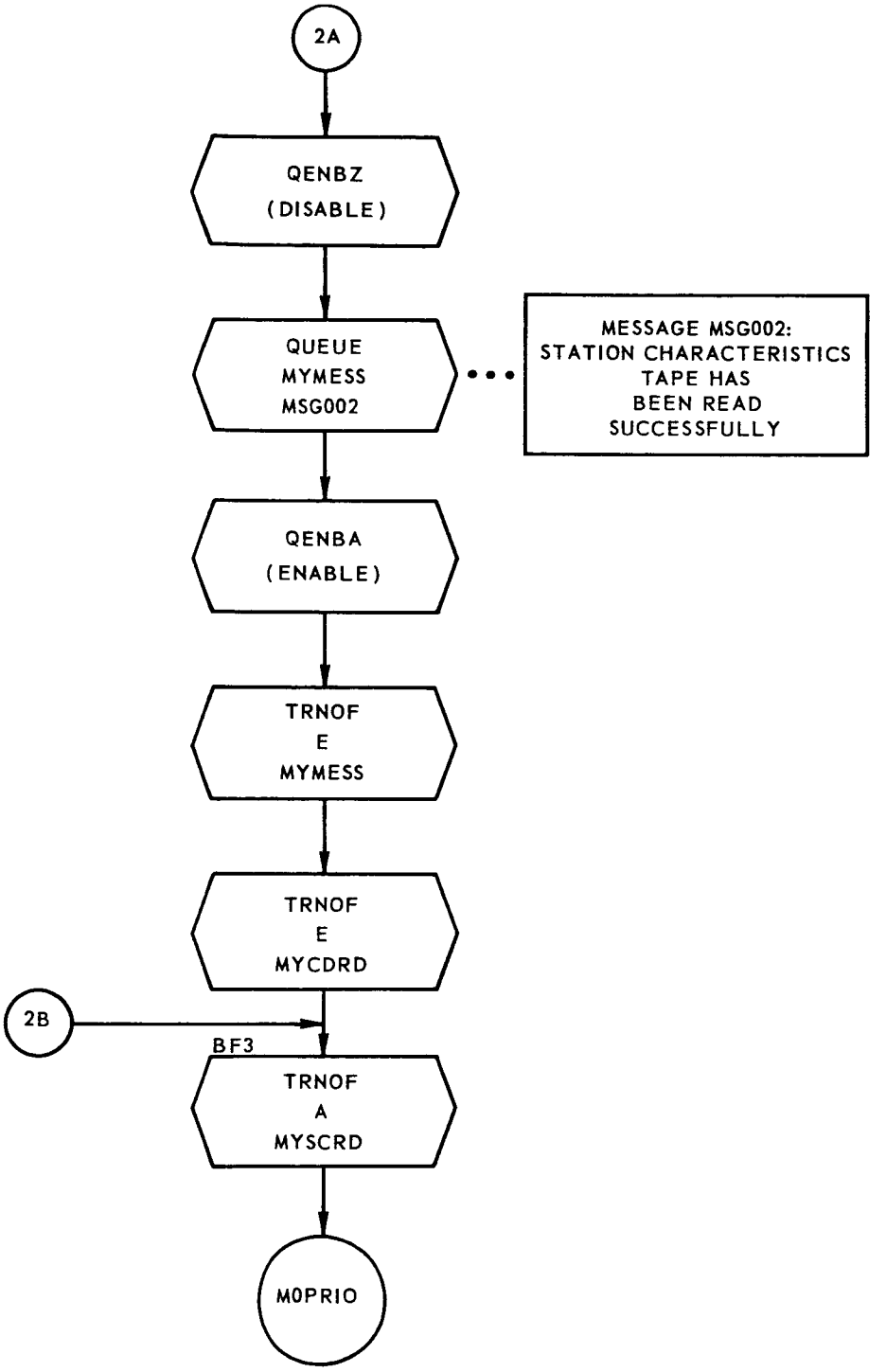


FIGURE 6–2. MYSCRD PROGRAM FLOW CHART *(Sheet 2 of 2)*

6.3 MONITOR STATION CHARACTERISTICS TAPE READER TRAP PROCESSOR (MTENST)

MTENST receives control from the channel A trap following the reading of a record from the Station Characteristics tape into core storage. This trap processor checks for an error in the tape reading and sets priority bits to control the reading of the next record from the tape.

The flow chart for the MTENST processor is shown in Figure 6-3.

6.3.1 Input Requirements

Input to MTENST consists of:

a) Location 00012—the location of the cell set by the computer at the channel A trap. The address of this location contains the location counter when the trap occurred. The three low-order bits in the decrement contain the reason code for the channel A trap. These are:

   1) A 1 in bit position 17 is the result of a trap following the completion of an IOCT, IORT or IOST command with no load channel instruction waiting. This is the expected condition.

   2) A 1 in bit position 16 is the result of a redundancy check during the I/O operation.

   3) A 1 in bit position 15 is the result of the end-of-file indicator being turned on in reading the tape. This error condition is tested by MTENST.

b) Macros Used—SAVE, TRNOF and TRNON.

6.3.2 Output Requirements

Output from MTENST depends upon the channel A trap which precedes entry into the trap processor. If the trap was the result of reading an end-of-file on the Stations Characteristics tape, there is no output from MTENST and program control transfers directly to the error subroutine, MSUERR. If an end-of-file was not read, MTENST turns on the B indicator for MYSCRD to continue the reading of the Station Characteristics tape. MTENST also turns off the suppression bits for MYCDRD and MYMESS which were set by MYSCRD.

6.3.3 Method

MTENST receives control from the channel A trap following the reading of a record (or an end-of-file if an error occurs) from the Station Characteristics tape into core storage. The trap processor executes the SAVE macro to preserve the condition of the interrupted program.

MTENST examines bit position 15 of location 00012. If the bit is a 1, an end-of-file has been read and program control transfers directly to the error subroutine, MSUERR.

If no end-of-file has been read, MTENST assumes that one record has been read successfully, and the B indicator for MYSCRD is turned on so MYSCRD can process the record read. The suppression bits set by MYSCRD for MYMESS and MYCDRD are turned off and control returns to M0PRIO.

6.3.4 Usage

MTENST is entered from a trap transfer instruction (TTR MTENST) in the channel A trap control location (00013); MTENST normally exits to M0PRIO or, if an error occurred in the tape reading, MTENST exits to MSUERR.

a) Storage Required—20 locations

b) MTENST uses:

1) Macros—SAVE, TRNOF and TRNON

2) Parameters—B, E, MNSCRD, MNMESS and MNCDRD

3) Location—00012

c) Time Required—1.46 milliseconds

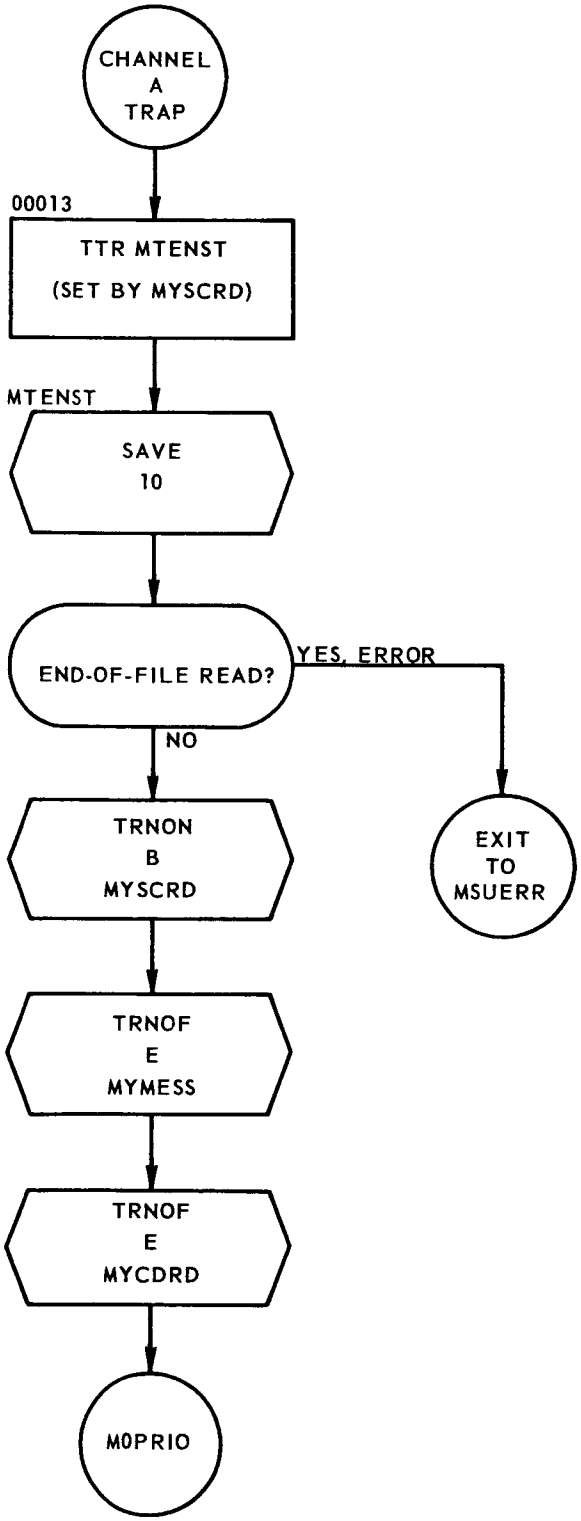


FIGURE 6-3. MTENST PROGRAM FLOW CHART

6.4 THE ON-LINE MESSAGE SYSTEM

6.4.1 The On-Line Message Cycle

In a normal situation the on-line message cycle requires the execution in the stated order of the following four processors: MYMESS, MTMSCK, MYMSCK and MTENPR. The general function and integration of these processors are illustrated in Figure 6-4. A brief description of the on-line message cycle follows:

a) MYMESS, the Monitor On-line Message Processor, must initiate every on-line message. MYMESS receives control from M0PRIO whenever priority restrictions permit after an entry has been placed in the MYMESS input queue. MYMESS suppresses all other processors which initiate transmission on data channel A (except MYMSCK which is never given control until after the transmission initiated by MYMESS is completed). The one-word input (see Subsection 6.4.2) from the queue specifies the message which must be read from the Message tape. MYMESS checks the number for possible error:

   1) If the number of the requested message is zero or greater than the highest message number, MYMESS sets the channel A trap control location to give control to MTENPR, generates the information necessary for logging and gives the commands and instructions to print a standard error message directly from core storage. When MYMESS finds an error in the message number, neither MTMSCK nor MYMSCK is used, nor is the Message tape.

   2) If the message number is valid, MYMESS sets the channel A trap control location to give control to MTMSCK, and generates the commands and instructions necessary to position the Message tape at the requested message and read the message into core storage.

   MYMESS suppresses itself and exits to M0PRIO.

b) MYMSCK, the Monitor Message Check Trap Processor, saves the control word set by the channel A trap, updates the tape position cell and turns on the B indicator for MYMSCK. MTMSCK exits to M0PRIO.

c) MYMSCK, the Monitor Message Check Processor, receives control from M0PRIO and sets the channel A trap control location to give control to MTENPR.

   1) If the trap was caused by an end-of-file or if the numbers of the message requested and the message read from tape are not identical, MYMSCK generates the information for logging and gives the commands and instructions necessary to print from core storage the applicable error message.

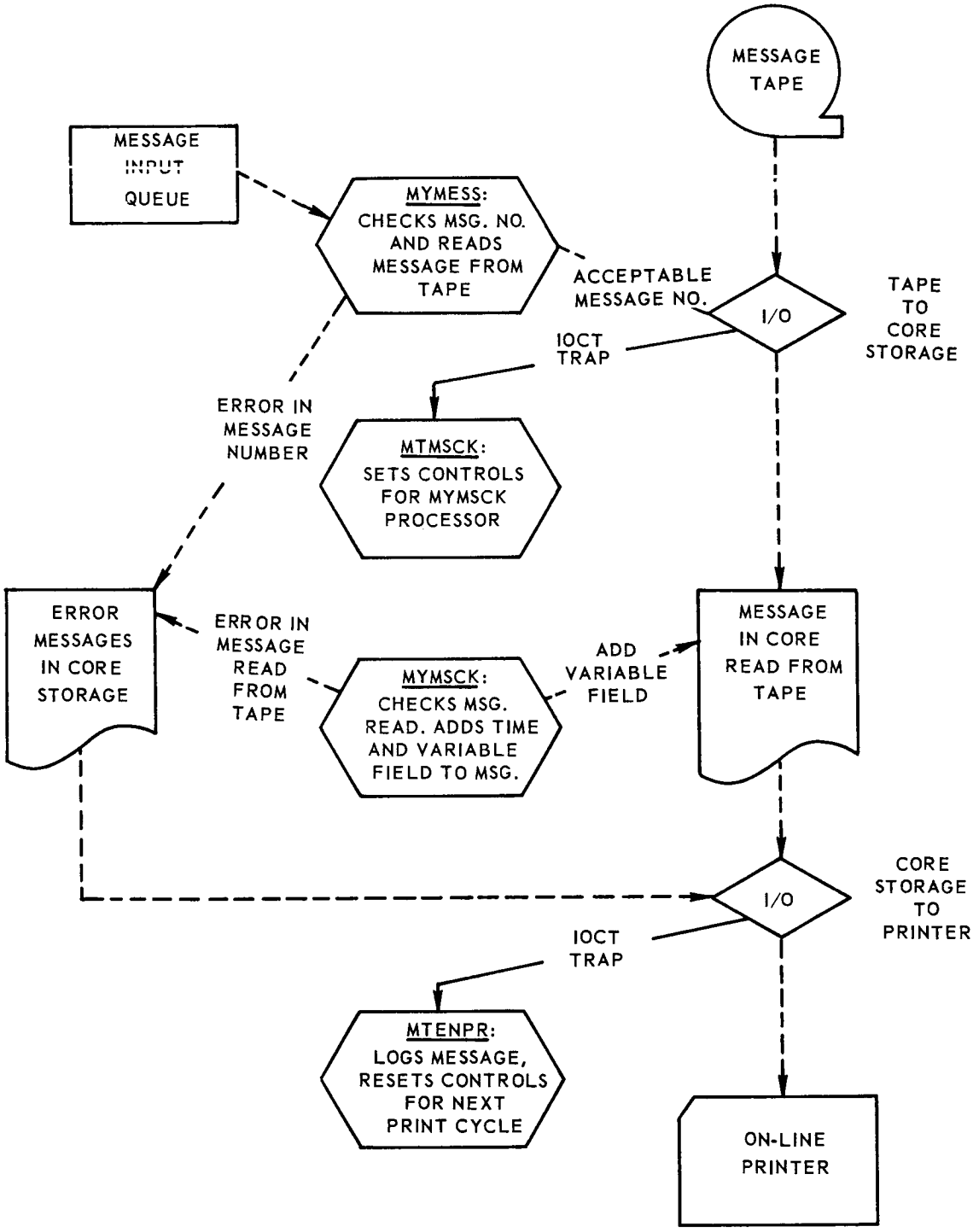


FIGURE 6-4. ON-LINE MESSAGE CYCLE

2) If the requested message was properly read into core, MYMSCK generates the information for logging, and if the message requires the addition of variable field data, MYMSCK converts the specified data into card image format and adds this to the message. The current time is also added to the message and the commands and instructions are given to print the completed message. MYMSCK exits to M0PRIO.

d) MTENPR, the Monitor End-of-Printing Trap Processor, receives control after the on-line message has been printed. MTENPR logs the message, unsuppresses the channel A transmission processors. MTENPR completes the message cycle.

6.4.2 Procedure for Queueing On-Line Messages

The on-line message cycle can currently process messages with or without variable fields. A message without a variable field is read from tape, given a time tag and printed. The variable field may be a fixed or floating point number or a time in hours minutes and seconds. The procedures for queueing each of these four types of messages are given below, with the added variable field underscored in the examples of printed messages.

a) Message without variable field

1) Method:

```
        QENBZ
        QUEUE     MNMESS, K
        QENBA
          .
          .
          .
K       PZE       Message Number
```

2) Example of message printed:

TTY TRANSMISSION TO GODDARD HAS BEGUN

b) Floating-point number variable field (this type of variable field is not currently being used).

1) Method:

```
        QENBZ
        QUEUE     MNMESS, K
        QENBA
          .
          .
          .
K       MTW       X,,Message number
X                 Floating point number
```

2) Restriction: $-999.9999 \leq C(X) \leq 999.99999$

3) Example of message printed:

THE COMPUTED LATITUDE OF IMPACT IS <u>15.75214</u>

c) Fixed-point number variable field

1) Method:

```
        CLA       Y
        STA       K
        STT       K
        QENBZ
        QUEUE     MNMESS, K
        QENBA
          .
          .
          .
K       MTH       **,,Message number
Y                 Fixed-point number
```

2) Restrictions: $0 \leq C(Y) < 2^{18}-1$

3) Example of message printed:

NUMBER OF MINUTES PASS SINCE TIME INITIALIZED 7

d) Time variable field

1) Method:

```
        CLA     Z
        STA     K
        STT     K
        QENBZ
        QUEUE   MNMESS, K
        QENBA
          .
          .
          .
K       MZE     **,,Message number
Z               Time in fixed-point seconds
```

2) Restrictions: $0 \leq C(Z) \leq 2^{18}-1$, values for hours are computed modulo 24.

3) Example of message printed:

TIME OF LIFT-OFF IS 11 HRS. 06 MINS. 21 SECS.

The time of printing is added to every non-error message printed. This time is GMT and appears as hours, minutes and seconds, left justified, on the message. A period separates the hours from the minutes and the minutes from the seconds. An example of a complete message is:

11.06.22 TIME OF LIFT-OFF IS 11 HRS. 06 MINS. 21 SECS.

Table 6-1 lists the on-line messages and the programs which queue these messages.

TABLE 6-1. ON-LINE MESSAGES

Message No.	Message	Used By
1	Normal Operation Has Begun	MTIWWV
2	Station Characteristics Tape Has Been Read Successfully	MYSCRD
3	Simulated Operation Has Begun	MOSENT
4	Floating Point Trat at Location Printed Below in Binary	MTFLPT
5	Manual Insertion Rejected. Enter Message Again	MFMANI
6	File Protect Tape on B6 and Set Up New Tape	MSLOGG
7	File Protect Tape on B7 and Set Up New Tape	MSLOGG
20	WWV Time Entered is (GMT) Hrs. Mins. Sec.	MOSENT MTIWWV
21	Time of Lift Off Is Hrs. Mins. Secs.	MOSENT MFMANI
22	Number of Minutes Pass Since Time Initialized	MYMINS
30	Time of SECO Computed is (GMT) Hrs. Mins. Secs.	MFSMOO
31	FPS-16 Entered Bad Points	MFEBAD
32	FPS-16 Completed Bad Points	MYEBAD
33	Verlort Entered Bad Points	MFEBAD
34	Verlort Completed Bad Points	MYEBAD
35	FPS-16 has Begun Tracking	MFHSIN
36	FPS-16 Has Ceased Tracking	MFHSIN
37	Verlort Has Begun Tracking	MFHSIN
38	Verlort Transmission to Goddard Has Begun	MFHSIN
39	TTY Transmission to Goddard Has Begun	MFTOUT
40	FPS-16 Smoothed Data Has Been Transmitted to Goddard	MFTOUT
41	Orbit Data Has Been Transmitted to Goddard	MFTOUT
42	Verlort Smoothed Data Has Been Transmitted to Goddard	MFTOUT
43	Hyperbolic Orbit Has Been Computed	MFCOMP
44	No Intersection With 450,000 Feet	MFCOMP
45	Time to Fire Retros for Last Recovery Area Has Passed	MFCOMP
49	Teletype Transmission to Goddard is Complete for This Pass	MTTTOT
60	Capsule Separation Signal Has Been Received	MYTMIN
61	Capsule Separation Signal Has Been Overridden	MYTMIN
62	ICTRC Has Been Requested	MYTMIN
63	Request for ICTRC Has Been Overridden	MYTMIN
71	Retro 1 Has Fired	MYTMIN
72	Retro 1 Fire Has Been Overridden	MYTMIN
73	Retros 1 + 2 Have Fired	MYTMIN
74	Retros 1 + 2 Fire Has Been Overridden	MYTMIN
75	Retros 1 + 2 + 3 Have Fired	MYTMIN
76	Retros 1 + 2 + 3 Fire Has Been Overridden	MYTMIN
77	Mayday Signal Has Been Received	MYTMIN
78	Mayday Signal Has Been Overridden	MYTMIN

6.5 MONITOR ON-LINE MESSAGE PROCESSOR (MYMESS)

MYMESS is the initial program in the cycle which prints on-line a prepared tape message. A one-word input identifies the requested message by number and type. If the message number refers to a nonexistent message, MYMESS initiates directly the printing of an error indication; otherwise, MYMESS locates the message and initiates transmission of the message into core storage.

MYMESS also sets bits in the Monitor priority table and establishes other controls for channel A operations.

The flow chart for the MYMESS processor is shown in Figure 6-5.

6.5.1 Input Requirements

Input to MYMESS consists of:

a) MUTAPA—parameter designating the unit tape address of the Message tape, (A 6).

b) MNAENO—parameter designating the number of messages on the Message tape. Since each message consists of one record, MNAENO also defines the number of records of the Message tape. Each message contains 25 words: a one-word message (record) number followed by the message itself in the remaining 24 words.

c) TMCMDA—Monitor table of input/output commands used by MYMESS and MYMSCK to control the on-line message printing.

d) TMMES2—24-cell table containing in card image format the message, BINARY NUMBER PRINTED BELOW WAS REQUESTED BUT DOES NOT EXIST.

e) MCMSNO—communication cell which contains the number of the requested message. This cell is set by the UNQUE macro within the MYMESS program and is obtained from the input queue for MYMESS.

f) MCMTPR—communication cell, used as a constant by MYMESS and MYMSCK; contains the instruction, TTR MTENPR.

g) MCTPOS—communication cell which designates the message (by record) position of the Message tape. Since the tape is initially set to the first message, MCTPOS initially contains the number, 1. For each message read from the Message tape, MYMSCK updates MCTPOS to correspond to the current tape position. MYMESS calculates from MCTPOS the required direction the Message tape must be spaced to be positioned at the requested message.

h) AETP—cell containing the instruction, TTR MTMSCK.

i) AENO—cell containing the parameter, MNAENO (described above).

j) K00000—constant, zero.

k) K00001—constant, 1.

l) K00002—constant, 2.

m) K00025—constant, 25.

n) Macros—QENBA, QENBZ, TRNON, TRNOF, UNQUE and BINNO (see Subsection 6.6).

6.5.2 Output Requirements

Output from MYMESS depends upon the number of the requested message. If the message number is nonexistent, i.e., equal to zero or greater than the number of messages on the Message tape, MYMESS initiates printing of the illegal number with the error message contained in TMMES2 (BINARY NUMBER PRINTED BELOW WAS REQUESTED BUT DOES NOT EXIST). MYMESS also provides the following output requirements:

a) TMPRLG—a two-cell table used by MSLOGG as input for the logging of on-line message printouts. MYMESS stores a 2 in the address of TMPRLG to indicate that the second error message was printed, and the illegal message number is stored in TMPRLG + 1.

b) Location 00013—the channel A trap control location contains the instruction, TTR MTENPR. If the message number is legal, the location of the message relative to the current Message tape position is determined and the instructions are generated to read into TMMESS the requested message.

c) Location 00013—contains the instruction, TTR MTMSCK.

d) MCMSNO—contains in the address the number of the requested message.

e) MCMVIP—contains the one-word requested message number as obtained from the input queue for MYMESS and originally contained in MCMSNO.

6.5.3 Method

MYMESS receives control from M0PRIO when one or more message requests are stored in the input queue to MYMESS. This program first turns on

its own A indicator and then suppresses the other programs (MYSCRD and MYCDRD) which transmit on channel A. The computer is disabled and the input data word is taken from the input queue for MYMESS. If the queue contains only one word, the B indicator is turned off. The computer is enabled.

The input data word is contained in location MCMSNO and is also stored into MCMVIP for later reference (if the message requires variable field data).

The prefix of the input data word specifies the type of message requested. If the prefix is zero, the message does not require a variable field addition and the message number is in the address of the input data word. If the prefix is non-zero (octal 4, 6 and 7 are the three possible alternatives), a variable field is required, and the message number is contained in the decrement of the input data word, with the address containing the location of the input data for the variable field conversion. For the non-zero prefix, MCMSNO is cleared and the message number stored in the address.

The requested message number is examined for legality:

a) If the message number is valid, MYMESS stores in the channel A trap control location (00013) a TTR MTMSCK instruction for the trap following the transmission of the message from tape into core storage. MYMESS then determines the position of the message on the tape relative to the tape position. If the requested message is past the read head, the Message tape is rewound. The computer is disabled and tape commands are generated to space past the preceding messages and read the requested message into core storage, storing the 25-word message record in the TMMESS table.

b) If the requested message number is unacceptable, i.e., equal to zero or greater than the largest message number on the Message tape, the illegal number is converted into a card image format by the BINNO macro (see Subsection 6.6). The two-word input table (TMPRLG) for logging is set with a 2 stored in the address of TMPRLG and the illegal number in the address of TMPRLG + 1. MYMESS stores in the channel A trap control location (00013) a trap transfer to MTENPR. The computer is disabled and the commands given to print on-line the error message (BINARY NUMBER PRINTED BELOW WAS REQUESTED BUT DOES NOT EXIST) from TMMES1 and the binary number from TMMES3.

After either step (a or b) is completed, MYMESS suppresses itself to prevent entry until the current message has been completely processed. The computer is enabled, the A indicator for MYMESS turned off, and program control returns to M0PRIO.

6.5.4 Usage

MYMESS is entered from, and exits to, M0PRIO.

a) Storage Required—80 locations

b) MYMESS uses:

1) Macros—BINNO, QENBA, QENBZ, TRNON, TRNOF and UNQUE

2) Parameters—MUTAPA, MNAENO, MNCDRD, MNMESS, MNSCRD, A, C and D

3) Communication Cells—MCMSNO, MCMTPR, MCMVIP and MCTPOS

4) Tables—TMCMDA, TMMESS, TMMES2 and TMPRLG

5) Constants—K00000, K00001, K00002, K00025 and AETP

6) Locations—00013 and AENO

c) Time Required:

1) Minimum—1.67 milliseconds

2) Most Probable—1.81 milliseconds

3) Maximum—2.89 milliseconds

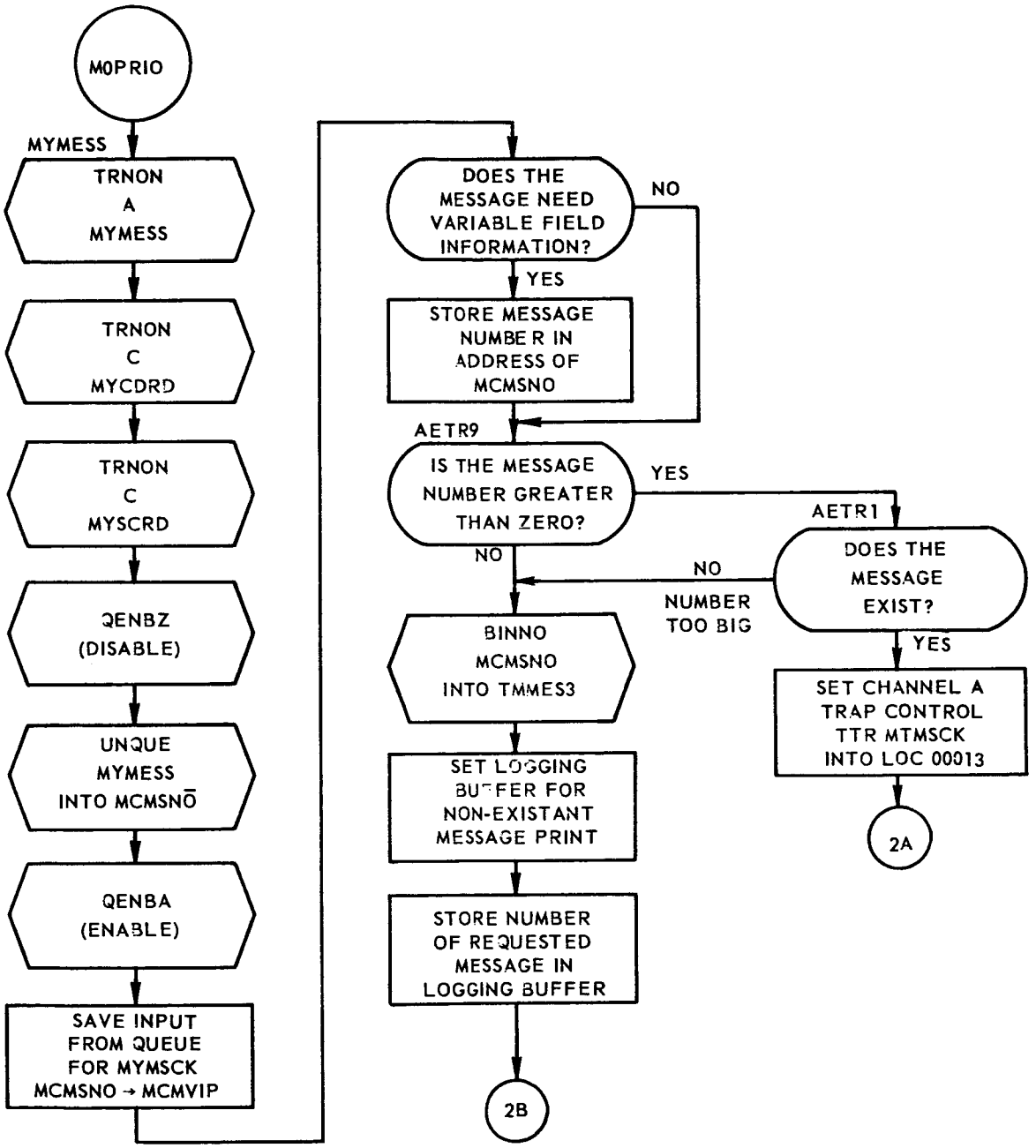


FIGURE 6–5. MYMESS PROGRAM FLOW CHART *(Sheet 1 of 2)*

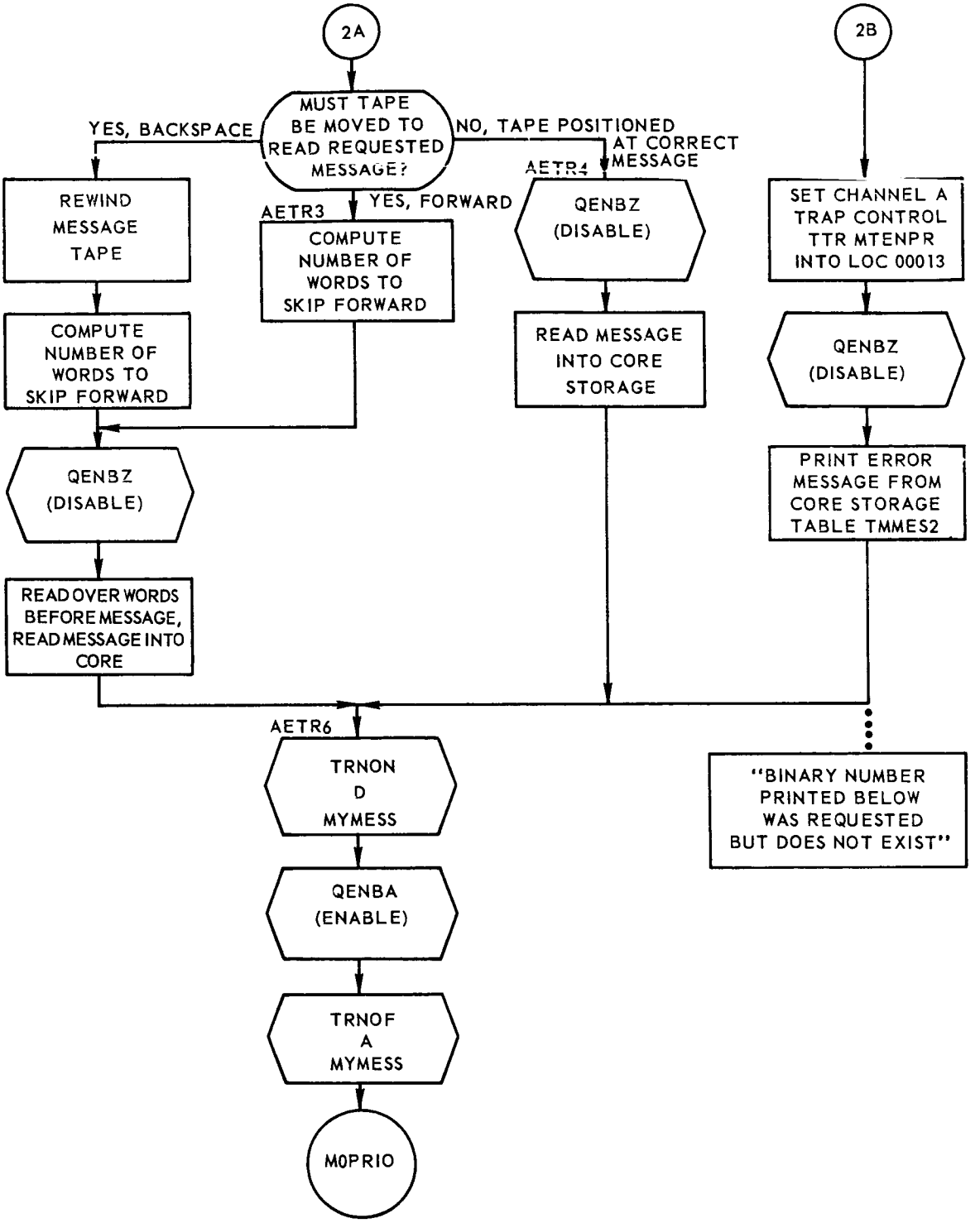


FIGURE 6-5. MYMESS PROGRAM FLOW CHART *(Sheet 2 of 2)*

6.6 BINARY NUMBER CARD IMAGE MACRO (BINNO)

BINNO converts the number in location MCMSNO into a Hollerith-coded card image so an on-line print may reproduce the number in 36-bit binary format.

The flow chart for the BINNO macro is shown in Figure 6-6.

6.6.1 Input Requirements

The BINNO macro is defined:

```
BINNO   MACRO
        AXT     24, 1
        STZ     TMMES3 + 24, 1
        TIX     *-1, 1, 1
        CAL     MCMSNO
        STO     TMMES3 + 16
        COM
        SLW     TMMES3 + 18
        END
```

The location MCMSNO contains the message number of the requested message.

6.6.2 Output Requirements

TMMES3 is a 24-cell table in which is placed the card image of the number from MCMSNO. Every location is cleared except TMMES3 + 16 and TMMES3 + 18, 1-row left and 0-row left, respectively, of the card image. TMMES3 + 16 contains the number in MCMSNO; TMMES3 + 18 contains the complement of this number.

6.6.3 Method

The 24 cells of the TMMES3 table are cleared. The contents of location MCMSNO are stored in the 17th word of the table; the complement of the contents of location MCMSNO is stored in the 19th word of this table. Consequently, for every 1 in the location, MCMSNO, the corresponding bit in TMMES3 + 16 becomes a 1. For every zero in location MCMSNO, the corresponding bit in TMMES3 + 18 becomes a 1. An on-line printout (in the decimal mode) of the TMMES3 table produces the number in binary format.

6.6.4 Usage

Entry into BINNO has no restrictions, except that the location MCMSNO must contain the number to be converted (BINNO is used only by MYMESS and MYMSCK). BINNO exits to the seventh location following entry.

a) Storage Required—seven locations

b) BINNO uses:

1) Communication Cell—MCMSNO

2) Table—TMMES3

c) Time Required—1.27 milliseconds

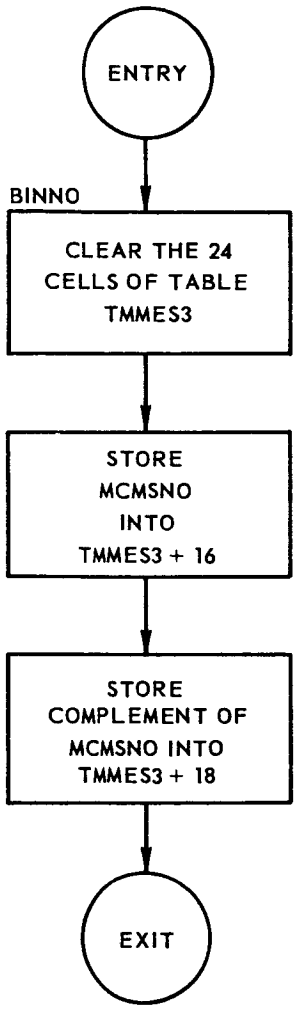


FIGURE 6-6. BINNO MACRO FLOW CHART

6.7 MONITOR MESSAGE CHECK TRAP PROCESSOR (MTMSCK)

MTMSCK receives control from the channel A trap following the reading of a tape message into core storage initiated by MYMESS. MTMSCK updates the tape position communication cell, saves the control word set by the trap and establishes controls for the message check processor, MYMSCK. The flow chart for the MTMSCK processor is shown in Figure 6-7.

6.7.1 Input Requirements

Input to MTMSCK consists of:

a) MCMSNO—communication cell, set by MYMESS, which contains the number of the requested message.

b) Location 00012—the location of the cell set by the computer at the channel A trap. The address of this location contains the location counter value when the trap occurred. The three low-order bits in the decrement are coded to indicate the reason for the trap.

c) K00001—constant, 1.

d) Macros Used—SAVE and TRNON.

6.7.2 Output Requirements

Output from MTMSCK consists of:

a) MCEFTS—contains the contents of location 00012.

b) MCTPOS—contains the current position of the Message tape, i.e., the number of the message now positioned at the tape read head.

c) MTMSCK also turns on the B indicator for MYMSCK in the priority table.

6.7.3 Method

MTMSCK receives control on a TTR MTMSCK in location 00013 following a trap on channel A. The SAVE macro is executed to preserve the condition of the interrupted program for a later return. The cell which indicates the position of the Message tape is updated and the B indicator for MYMSCK is turned on. The contents of location 00012 are stored in the communication cell, MCEFTS (this cell is interpreted by MYMSCK, the next step in the on-line message program cycle).

6.7.4 Usage

MTMSCK is entered by a trap transfer instruction (TTR MTMSCK) in the channel A trap control location (00013); MTMSCK exits to M0PRIO.

a) Storage Required—11 locations.

b) MTMSCK uses:

1) Macros—SAVE and TRNON

2) Parameters—B and MNMSCK

3) Communication Cells—MCEFTS, MCMSNO and MCTPOS

4) Constant—K00001

5) Location—00012

c) Time Required:

1) Maximum—1.28 milliseconds

2) Minimum—0.56 millisecond

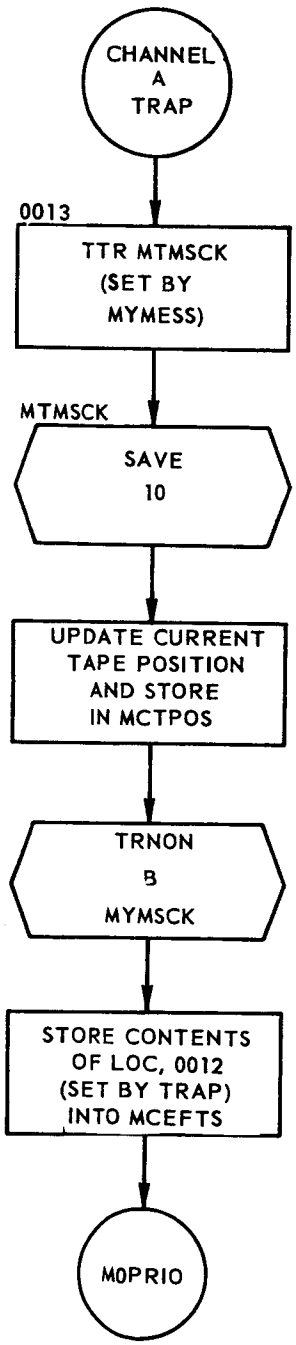


FIGURE 6–7. MTMSCK PROGRAM FLOW CHART

6.8 MONITOR MESSAGE CHECK PROCESSOR (MYMSCK)

MYMSCK receives control from M0PRIO after a message has been read into core storage from the Message tape. MYMSCK compares the number of the message read from tape with the number of the message requested, and prints on-line an error indication if these do not agree. If the requested message has been properly read into core storage, MYMSCK supplies any necessary variable fields, and the current Greenwich Mean Time (GMT) for each correct message printed. In every case MYMSCK provides the input for logging the on-line message.

The general flow diagram and the flow chart for the MYMSCK processor is shown in Figures 6-8 and 6-9.

6.8.1 Input Requirements

Input to MYMSCK consists of:

a) MUTAPA—parameter designating the unit tape address of the Message tape.

b) TMCMDA—Monitor table of input/output commands used by MYMESS and MYMSCK to control on-line message printing.

c) TMMESS—25-cell table containing the message read into core storage by MYMESS. The first cell contains the number of the message read into core storage; the remaining 24 cells contain the card image of the message.

d) TMMES1—24-cell table containing in card image format the following error message: MESSAGE NUMBER REQUESTED DISAGREES WITH ONE FOUND—BOTH PRINTED BELOW.

e) TMMES4—six-cell table of bit masks used in adding the card image of an integer to the card image of the message in TMMESS +1 through TMMESS+ 25. TMMES4 has a 1 in bit position 30, TMMES4+ 1 has a 1 in bit position 31, etc., TMMES4+ 5 has a 1 in bit position 35.

f) TMMES5—three-cell table used in computing GMT in hours, minutes and seconds from a value given in seconds. TMMES5 contains the number of seconds in 24 hours, TMMES5 +1 contains the number of seconds in one hour and TMMES5 +2 contains the number of seconds in one minute.

g) TMMES6—three-cell table of bit masks used in adding the card image of a time to the card image of the message in TMMESS+ 1 through TMMESS+ 25. TMMES6 contains a 1 in bit position 11, TMMES6+ 1 contains a 1 in bit position 20 and TMMES6+ 2 contains a 1 in bit position 29.

h) TMMES7—three-cell table of bit masks used in adding a card image of the current time (in GMT hours, minutes and seconds) to the card image of the message in TMMESS 1 through TMMESS+ 25. TMMES7 contains a 1 in the sign position, TMMES7+ 1 contains a 1 in bit position 3 and TMMES7+ 2 contains a 1 in bit position 5.

i) TMMES8—four-cell table of bit masks used in adding a card image of a floating point number to the card image of the message in TMMESS 1 through TMMESS +25. TMMES8 contains a 1 in bit position 26, TMMES8 +1 contains a 1 in bit position 27, etc., TMMES8+ 3 contains a 1 in bit position 29.

j) TMFALI—four-cell table used in separating a fixed point number into integrals which represent tens of thousands, thousands, hundreds, tens, and units. TMFALI contains 10,000; TMFALI +1 contains 1000; TMFALI +2 contains 100; and TMFALI+ 3 contains 10 (all values are decimal).

k) MCHFSC—contains the current GMT in fixed point half-seconds.

l) MCEFTS—contains the contents of location 00012 when the trap occurred. The cell is saved by MTMSCK and contains the reason code for the channel A trap in the three low-order bits (15, 16 and 17) of the decrement. The remaining bits of the decrement are cleared. The reason codes are:

1) A 1 in bit position 17 is the result of a trap following the completion of an IOCT, IORT, or IOST command with no load channel instruction waiting. This is the expected condition.

2) A 1 in bit position 16 is the result of a redundancy check during the I/O operation. This is an error condition.

3) A 1 in bit position 15 is the result of the end-of-file indicator being turned on in reading the tape. This is an error condition.

m) MCMSNO—communication cell containing in the address the number of the requested message.

n) MCMTPR—communication cell containing the instruction TTR MTENPR.

o) MCMVIP—communication cell containing the original requested message input to MYMESS. If the requested message does not require a variable field, MCMVIP and MCMSNO are identical. If a variable field is necessary to complete the message, the decrement of MCMVIP contains the message number (corresponding to the address of MCMSNO) and the address contains the location of the data to be used in generating the variable field.

p) K00001—constant, 1.

q) K00010—constant, 10 (decimal).

r) K00032—constant, 32 (decimal).

s) K7007S—mask, -377 777 776 000, used to clear columns in TMMESS +1 through TMMESS+25 for the addition of an integer in card image format.

t) KCH233—mask, 233 000 000 000, used to convert a number from floating point to fixed point format.

u) KMNMSK—mask, 000 777 777 777, used to clear columns in TMMESS +1 through TMMESS+25 for the addition of the present GMT in hours, minutes and seconds.

v) KMSKFT—mask, -377 407 017 017, used to clear columns in TMMESS +1 through TMMESS+25 for the addition of a time (correlating to the message) in hours, minutes and seconds.

w) KPEROD—mask, 011 000 000 000, used to supply the two periods which separate hours, minutes and seconds in the present (GMT) time. The present time appears left-justified for every non-error message printed.

x) Macros Used—TRNON, TRNOF and BINNO.

6.8.2 Output Requirements

Output from MYMSCK is a function of the success in reading the requested message from the Message tape into core storage:

a) If the message has been read successfully, the present (GMT) time is converted and added to the card image of the message in core. If a variable field must be supplied to the message, MYMSCK constructs a card image of the field (a time in hours, minutes and seconds—a fixed point or floating point number) and adds this to the card image for on-line printing. The logging input table is generated as follows: the decrement of TMPRLG contains the message number; the remainder of TMPRLG and all of TMPRLG+1 are cleared.

b) If an end-of-file was read on the Message tape, the tape is rewound, a 1 is stored in MCTPOS to update the tape position cell, and the logging input table is set as follows: a 1 is stored in TMPRLG and TMPRLG + 1 is cleared. The following message is printed on-line: BINARY NUMBER REQUESTED DISAGREES WITH ONE FOUND—BOTH PRINTED BELOW. (There is, however, no additional printout for this

error. This distinguishes an end-of-file error from the error condition described in the following paragraph.)

c) If the number of the message read into core storage does not agree with the number of the requested message (TMMESS ≠ MCMSNO), the Message tape is rewound, a 1 is stored in MCTPOS to update the tape position cell, and the logging input table is set as follows: a 1 is stored in TMPRLG and the requested message number (MCMSNO) is stored in TMPRLG+1. The requested message number is stored in the TMMES3 table in card image format and the following messages are printed on-line: MESSAGE NUMBER REQUESTED DISAGREES WITH ONE FOUND—BOTH PRINTED BELOW; the requested message number is in 36-bit binary format and the message is received in core storage.

For every condition, the TTR MTENPR instruction is stored in the channel A trap control location, 00013.

6.8.3 Method

MYMSCK receives control from M0PRIO and turns its A indicator on and its B indicator off. MYMSCK turns on the C indicator for MYMESS to suppress that routine until after MYMSCK has updated the tape position cell, MCTPOS. A trap transfer instruction to MTENPR is stored in the channel A trap control location, 00013 (an on-line print always follows the entry into MYMSCK).

MCEFTS is examined to determine the cause of the trap. If the trap was the result of an end-of-file, an error has occurred and the conditions described in Subsection 6.8.2, b, are established. MYMSCK unsuppresses MYMESS by turning off the C indicator for MYMESS. It then turns off its own A indicator and exits to M0PRIO.

If no end-of-file is detected, MYMSCK assumes a complete message has been read into the TMMESS block and examines the number to determine if the message received was the one requested. If the message numbers are not identical, an error has occurred and the conditions described in Subsection 6.8.2, c, are established. MYMSCK unsuppresses MYMESS by turning off the C indicator for MYMESS; MYMESS then turns off its own A indicator and exits to M0PRIO.

If the proper message has been read into core storage, MYMSCK establishes the logging input table and examines the original (to MYMESS) input word, MCMVIP, to determine whether the message requires additional variable field information. Every correct message requires the addition of the current GMT.

If the message requires a time (other than the current GMT), MYMSCK obtains the time in fixed point seconds from the tag and address of MCMVIP. The time is converted into hours, minutes and seconds. The values are converted into card image format and added to the message for printing.

If the message requires an integer, the integer is contained in the tag and address of MCMVIP. This number is separated into digits representing powers of ten. These digits are converted into card image format and added to the message for printing.

If the message requires a floating point number, the number is obtained from the location contained in the address of MCMVIP. The sign is set in the card image message and the number is converted to fixed point format. The fractional part is converted into card image format and added to the message. The same instructions used above for converting an integer are used to compute the variable field by converting the integral part of the floating point number.

After the variable field data has been added, MYMSCK adds the present time (obtained from MTHFSC) to the message and initiates transmission to print the message. MYMSCK unsuppresses MYMESS by turning off the C indicator for MYMESS. MYMSCK turns off its own A indicator and exits to M0PRIO.

6.8.4 Usage

MYMSCK is entered from, and exits to, M0PRIO.

a) Storage Required—179 locations.

b) MYMSCK uses:

1) Macros—TRNON, TRNOF and BINNO

2) Parameters—A, B, C, MNMSCK, MNMESS and MUTAPA

3) Communication Cells—MCEFTS, MCHFSC, MCMSNO, MCMTPR, MCMVIP and MCTPOS

4) Tables—TMCMDA, TMFALI, TMMESS, TMMES1, TMMES3, TMMES4, TMMES5, TMMES6, TMMES7, TMMES8 and TMPRLG

5) Constants—K00001, K00010, K00032, K7007S, KCH233, KMNMSK, KMSKFT and KPEROD

6) Location—00013

c) Time Required (all times are maximum):

1) EOF read on Message tape—0.73 millisecond

2) Error in reading message—2.16 milliseconds

3) No variable field—6.05 milliseconds.

4) Time variable field—8.78 milliseconds

5) Integer variable field—9.90 milliseconds

6) Floating point variable field—12.32 milliseconds

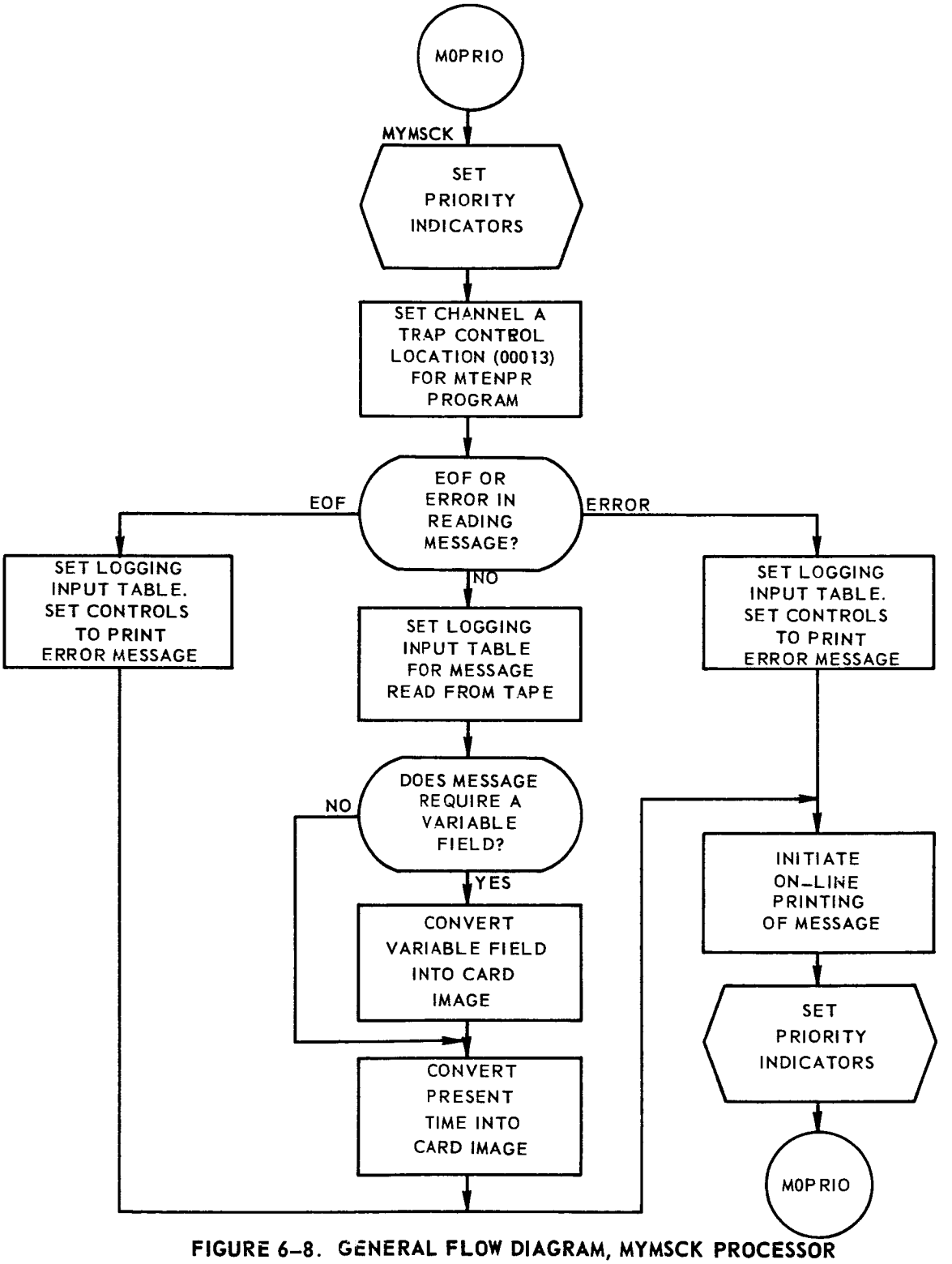


FIGURE 6-8. GENERAL FLOW DIAGRAM, MYMSCK PROCESSOR

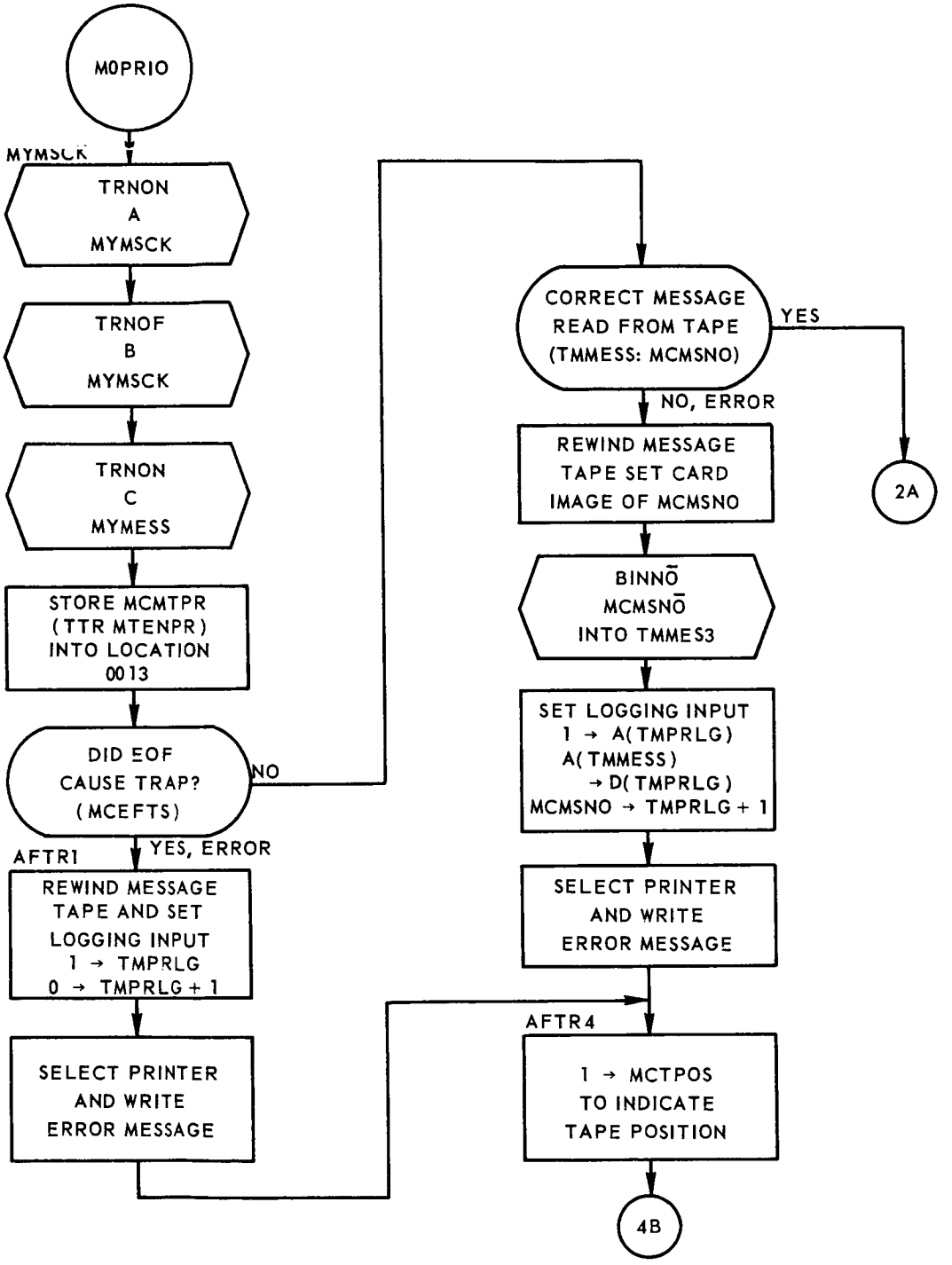


FIGURE 6-9. MYMSCK PROGRAM FLOW CHART *(Sheet 1 of 4)*

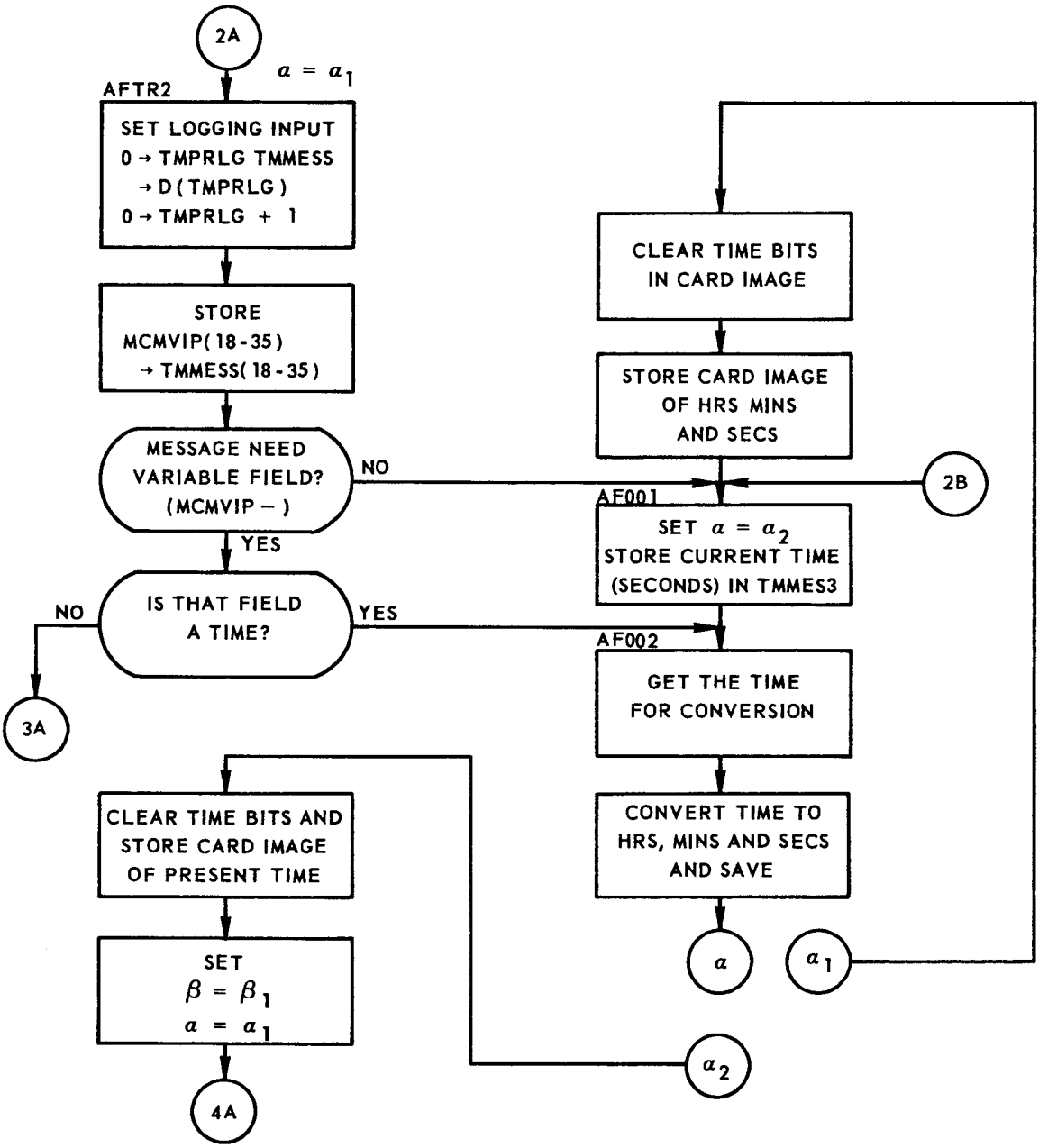


FIGURE 6–9. MYMSCK PROGRAM FLOW CHART *(Sheet 2 of 4)*

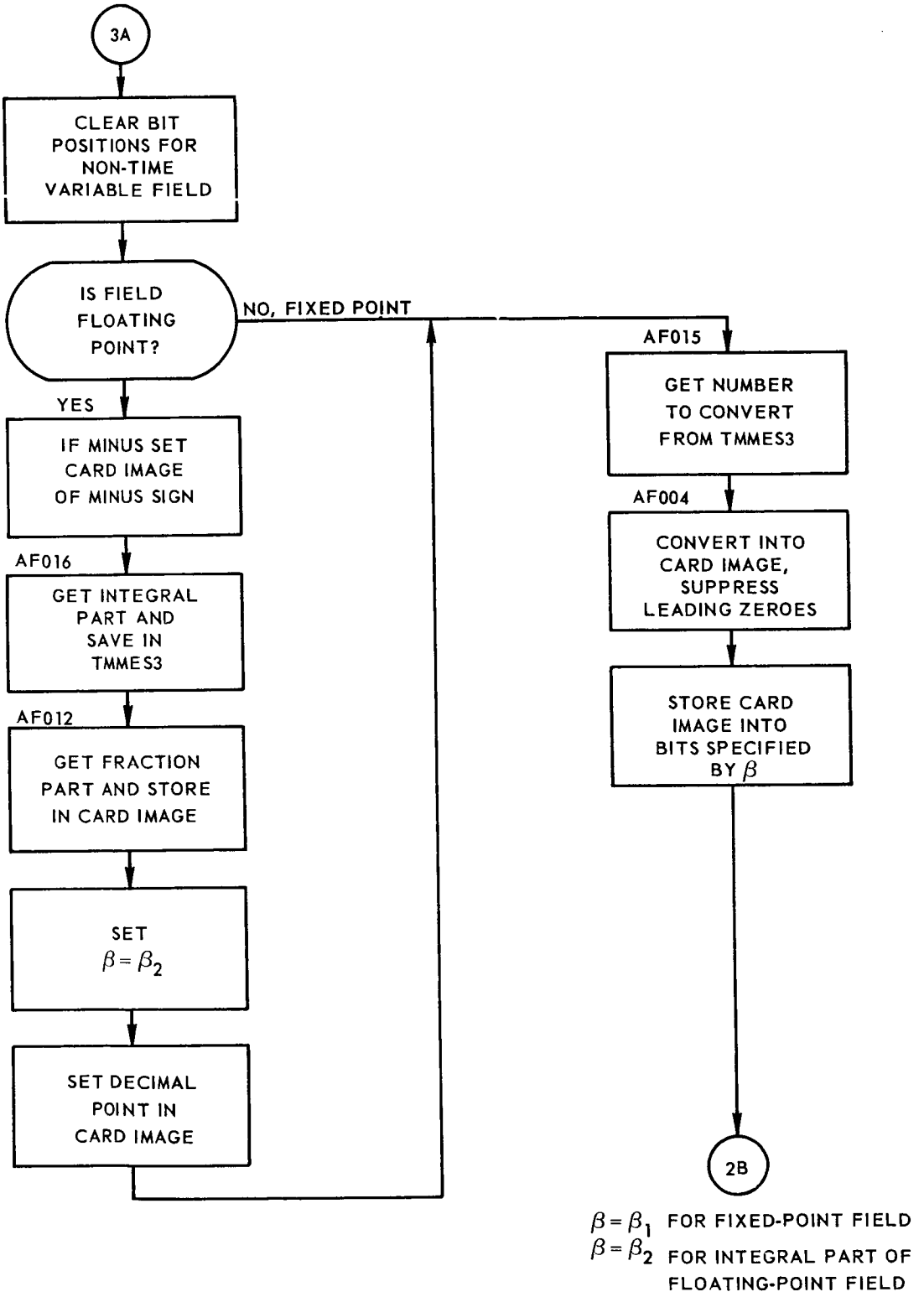


FIGURE 6–9. MYMSCK PROGRAM FLOW CHART *(Sheet 3 of 4)*

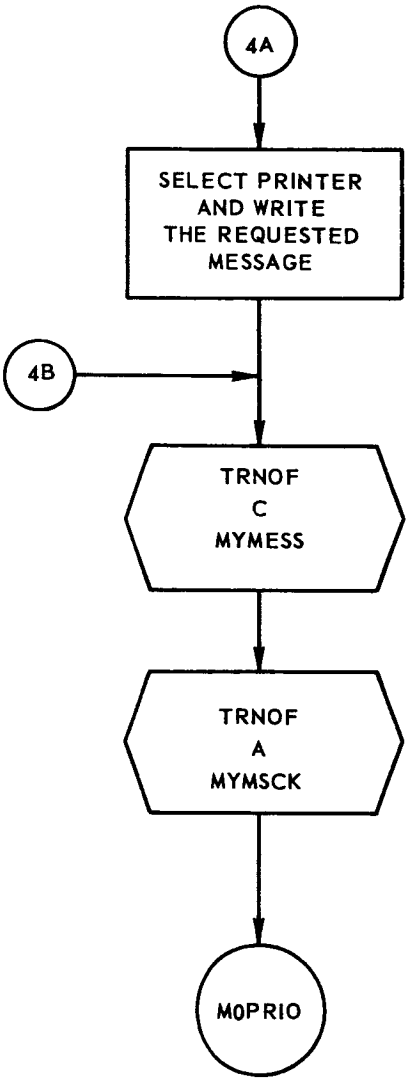


FIGURE 6-9. MYMSCK PROGRAM FLOW CHART *(Sheet 4 of 4)*

6.9 MONITOR END-OF-PRINTING TRAP PROCESSOR (MTENPR)

MTENPR, the last program in the on-line message cycle, receives control after the on-line message has been printed. This program exercises no control over the printed message, but simply logs the message, restores the on-line message cycle to its initial state in preparation for the next message request, and sets other channel A controls.

The flow chart for the MTENPR trap processor is shown in Figure 6-10.

6.9.1 Input Requirements

Input to MTENPR consists of:

a) TM8.3M—a one-cell table containing the count of the elapsed 8-1/3 milliseconds since the last exact half-second.

b) TMPRLG—a two-cell table which serves as input to the logging subroutine, MSLOGG, in logging on-line message prints. This table is established during the printing cycle by MYMSCK or, if an error occurs in the printing cycle, by either MY MSCK or MYMESS.

c) MCHFSC—communication cell containing the current GMT in fixed point half-seconds.

d) MCBETA—communication cell which, when positive, denotes that the computer is in the transmission mode for external transmission. (Since Bermuda is at present a simplex operation, the 709 must always be in the transmission mode, and the sign of MCBETA must be positive.)

e) Subroutine Used—MSLOGG.

f) Macros Used—SAVE and TRNOF.

6.9.2 Output Requirements

MTENPR logs the printed message and unsuppresses MYMESS, MYCDRD and MYSCRD.

6.9.3 Method

MTENPR receives control on a channel A trap following the on-line printing of a message. This trap processor executes the SAVE macro to preserve the condition of the program interrupted by the trap. MTENPR then constructs the standard logging time tag from the half-second and 8-1/3-millisecond timing cells. The time tag is stored in the MQ, the log calling sequence is completed and MTENPR executes the MSLOGG subroutine to log the on-line message just printed.

The suppression bits set by MYMESS for MYCDRD, MYSCRD and MYMESS are turned off and program control passes to M0PRIO.

6.9.4 Usage

MTENPR is entered by a trap transfer instruction (TTR MTENPR) in the channel A trap control location (00013); MTENPR exits to M0PRIO.

a) Storage Required—20 locations

b) MTENPR uses:

1) Subroutine—MSLOGG

2) Macros—SAVE and TRNOF

3) Parameters—C, D, MNCDRD, MNSCRD and MNMESS

4) Communication Cells—MCBETA and MCHFSC

5) Table—TM8.3M

c) Time Required—4.16 milliseconds (average, including MSLOGG time)

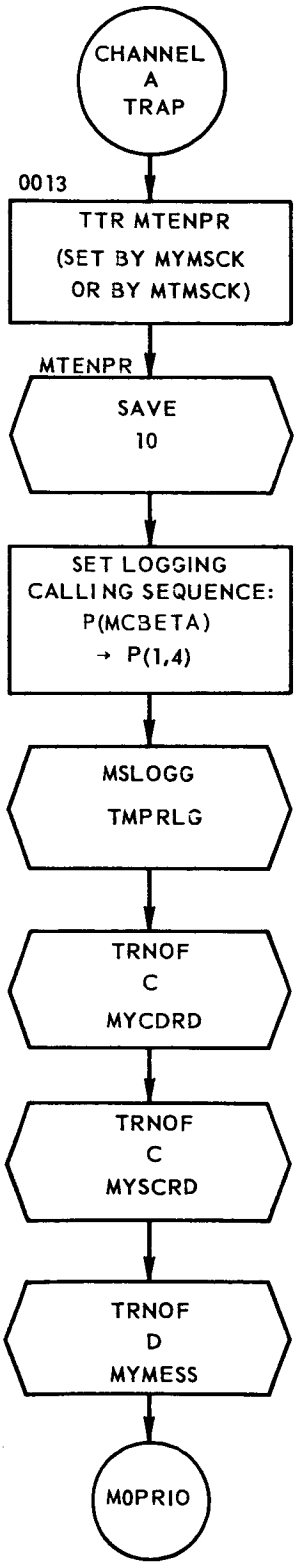


FIGURE 6-10. MTENPR PROCESSOR FLOW CHART

6.10 MONITOR ON-LINE CARD PROCESSOR (MYCDRD)

MYCDRD is not used in the present system. Its inclusion is a provision for a future incorporation of punch card input to the Mercury Program System. MYCDRD consists of a single instruction (to protect against an error which would give control to this processor) which transfers program control to the main priority controller, M0PRIO.

6.11 THE LOGGING PROCESS

6.11.1 The Logging Cycle

Three programs are required to complete the logging cycle: MSLOGG, MYSTLT and MTENLG. The general function and interrelation of these programs are illustrated in Figure 6-11. A brief description of the logging cycle is presented below:

a) MSLOGG, the Monitor Logging Subroutine stores the data for logging into a 17-cell block (illustrated in Figure 6-12) in a 170-cell logging buffer. There are two logging buffers, each having a capacity of ten blocks. When a buffer is filled, MSLOGG turns on the B indicator for the Monitor Logging Processor, MYSTLT. MSLOGG keeps a serial count of the logging blocks and, when the serial count reaches a specified increment, MSLOGG changes instructions so that the alternate log tape will now be used by MYSTLT. MSLOGG queues an on-line message which states that a log tape has been filled and should be replaced. MSLOGG is used by the following programs:

   MPHSIN, to log high-speed input radar data

   MTTTIN, to log low-speed paper tape input data

   MYTMIN, to log telemetry input data

   MTENPR, to log on-line messages

   MTTTOT, to log teletype output data

   MTHSBM, to log high-speed output data

   MTSENS, to log sense output for the Output Status Console

b) MYSTLT, the Monitor Logging Processor, generates the I/O commands to write the logging buffer on the log tape (the logging buffer and log tape addresses are controlled by MSLOGG). MYSTLT initiates the data transmission and suppresses itself until after the writing is completed.

c) MTENLG, the Monitor End-Of-Logging Trap Processor, checks for redundancy in the writing of the logging buffer and unsuppresses MYSTLT for the next writing on the log tape.

6.11.2 Format of the Logging Blocks

The logging block consists of 17 cells. The first five cells comprise the heading and contain information relative to the logged data. The last twelve

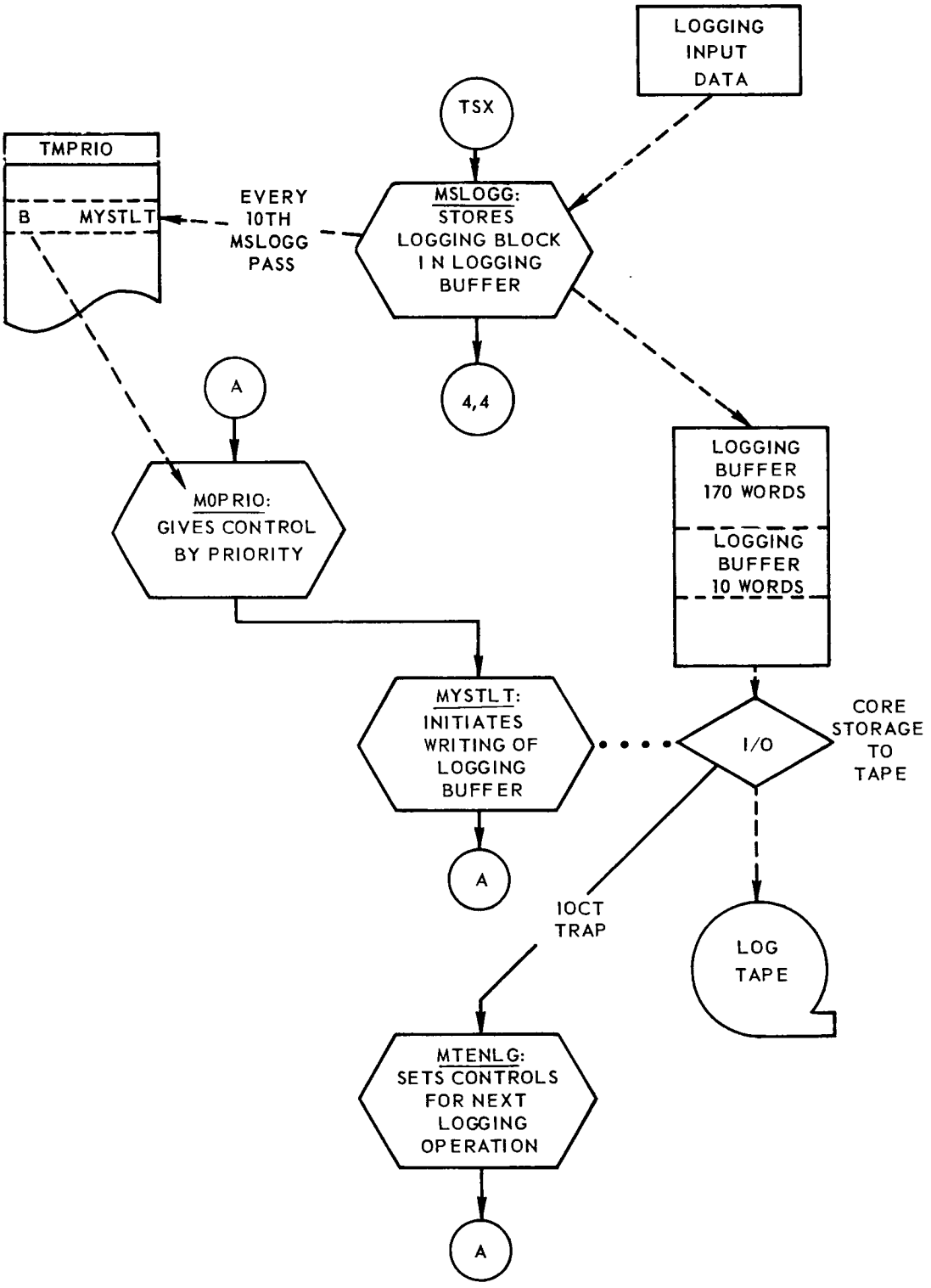


FIGURE 6-11. LOGGING CYCLE

MSLOGG LOGGING BLOCK

WORD	S 3	18	21 35	
1	IDENTIFICATION BY DCC SUBCHANNEL NO.			HEADING
2	TIME TAG: DATA AVAILABLE FOR LOGGING			
3	TIME TAG: DATA STORED IN THIS BLOCK			
4	*	SERIAL COUNT OF LOGGING BLOCKS		
5	**		DATA WORD COUNT IN THIS LOGGING BLOCK	
6	NON-DATA MASK (777 777 777 777)			LOGGED DATA
	NON-DATA MASK (777 777 777 777)			
	FIRST DATA WORD			
	SECOND DATA WORD			
	(DATA STORED FROM BOTTOM OF BLOCK)			
17	LAST DATA WORD			

* HOLD PHASE INDICATOR

** MISSION PHASE (L, A, O, R) INDICATOR

MSLOGG CALLING SEQUENCE

```
TSX   MSLOGG, 4
PZE   , , IDENTIFICATION
PZE   DATA LOCATION, , DATA WORD COUNT
RETURN
```

FIGURE 6-12. MSLOGG: LOGGING BLOCK AND CALLING SEQUENCE

cells contain the logged data which is stored forward from the last word in the block. If less than twelve data words are to be placed in the logging block, a non-data mask is stored in the unused cells following the heading but preceding the logged data. The second word of the heading contains the time the data was transmitted or received and the third word contains the time the data was placed in the logging block. Both time tags have the same format: bits S, 1-26 contain the Greenwich Mean Time in half-seconds, bits 27-35 contain the count of 8-1/3 millisecond intervals since last half-second. The fourth word contains the hold phase indicator, if applicable, in the prefix and the logging block serial count in the remainder of the word. The fifth word contains a mission phase indicator in the prefix and the logging block data word count in the address.

The first word of the heading identifies the DCC subchannel transmitting or receiving the data. The DCC subchannel, the data logged and the data word count for the various types of logging are described below:

a) High-speed input radar data

1) DCC Subchannel—1.

2) Data logged—input from both AN/FPS-16 and Verlort is logged simultaneously. Input from both radars consists of 12 data words, each word containing one 8-bit byte. For logging AN/FPS-16 data is right justified in the decrement of the logging block and Verlort data is right justified in the address.

3) Data Word Count—normally 12 words. More than 12 bytes from either radar requires two entries to MSLOGG.

b) Low-speed paper tape input

1) DCC subchannel—11.

2) Data logged—low-speed paper tape input messages are composed of 6-bit characters. These characters are packed six per logging word and left justified.

3) Data word count—varies from one to ten.

c) Telemetry input data

1) DCC subchannel—3.

2) Data logged—telemetry input messages are composed of eight words, each word containing 8 bits of information, right justified.

3) Data word count—12 words.

d) On-line message

1) Uses channel A, identification number is 33.

2) Data logged—the on-line message is logged from a 2-cell table (TMPRLG) generated during the on-line message cycle.

3) Data word count—two words.

e) Teletype output data

1) DCC subchannel—9.

2) Data logged—teletype output messages are composed of six 6-bit characters per computer word, left-justified, for transmission.

3) Data word count—six or seven words.

f) High-speed output data

1) DCC subchannel—2.

2) Data logged—the output messages for the local displays consist of a 12 word block, each word containing eight bits, right-justified.

3) Data word count—12.

g) Sense output for the Output Status Console

1) DCC subchannel—12.

2) Data logged—one word which controls the eight indicator lights on the Output Status Console.

3) Data word count—one.

6.12 MONITOR LOGGING SUBROUTINE (MSLOGG)

MSLOGG, MYSTLT and MTENLG provide a system for reproducing all input to and output from the Mercury Program System for post-mission and other analyses. MSLOGG stores logging blocks into the logging buffer in core storage and sets controls to write the logging buffer, when filled, onto the log tape. MYSTLT and MTENLG are the ordinary and trap processors, respectively, which control the writing of the logging buffer on the log tape.

The flow chart for the MSLOGG subroutine is shown in Figure 6-13.

6.12.1 Input Requirements

The calling sequence for MSLOGG is defined:

```
        TSX     MSLOGG,4
1,4     PZE     ,,Identification
2,4     PZE     Data Location,, Data Word Count
3,4     Return
```

The information which must be supplied in the calling sequence is described in a) through c) of the following list of input to MSLOGG:

a) Identification—the decrement of the second word of the calling sequence contains the number of the DCC subchannel sending or receiving the transmitted data. The number 33 identifies the logging of an on-line message printout.

b) Data Word Count—the decrement of the third word of the calling sequence contains the count of the data words to be logged. MSLOGG accepts a maximum word count of 12.

c) Data Location—the address of the third word of the calling sequence contains the location of the data to be stored in the logging block.

d) MQ—contains the time tag associated with the data to be logged. This time represents the time of reception for input data and the time of transmission for output data and on-line message printouts. The composition of the time tag is: $MQ_{S,\,1-26}$ contain the contents of MCHFSC (GMT in half-seconds); MQ_{27-35} contain the contents of TM8.3M (number of 8-1/3 milliseconds since last half-second.

e) MCHFSC, TM8.3M—communication cell and table, respectively, used by MSLOGG to construct a time tag of the same format as described in d) above. This time tag defines the time the logging block is constructed and stored in the logging buffer.

f) BCLAOR—communication cell specifying the mission phase:

BCLAOR = PZE	0,	launch phase
BCLAOR = MZE	0,	abort phase
BCLAOR = PZE	1,	orbit phase
BCLAOR = MZE	1,	re-entry phase

g) BTHOLD—one-cell table which, when non-zero, specifies that the mission has entered the hold phase.

h) MCALBL, MCALBT—communication cells containing in their addresses the locations of the logging buffer being filled and the logging buffer being written on tape, respectively. MCALBL and MCALBT contain the locations of TMALB1 and TMALB2 in opposition with each other as these two buffers are alternately filled and written on the log tape.

i) MCLTP1, MCLTP2—communication cells containing the write-select instructions WTBB MULTP1 and WTBB MULTP2. The instruction in MCLTP1 addresses the log tape being written; the instruction in MCLTP2 addresses the log tape in reserve. Normally, one log tape is written until it is filled; then MSLOGG exchanges MCLTP1 and MCLTP2.

j) MCALM1, MCALM2—communication cells containing in their addresses the message numbers for the following two messages: FILE PROTECT TAPE ON B6 AND SET UP NEW TAPE and FILE PROTECT TAPE ON B7 AND SET UP NEW TAPE. At any time the message number in MCALM1 corresponds to the tape being written, i.e., the tape unit addressed by instruction in MCLTP1 is the tape unit referred to by the message whose number is in MCALM1.

k) MULTP1, MULTP2—parameters designating the primary and secondary logging tapes, B6 and B7, respectively.

l) ALNOW—internal cell whose contents designate the total number of words (heading and data) per logging block, 17.

m) ALNOB—internal cell whose contents designate the maximum number of logging blocks per logging buffer, 10.

n) ALBFL—internal cell whose contents upon entry to MSLOGG designate the number of blocks stored in the logging buffer.

o) ALTFL—internal cell whose contents designate the serial count (ALSCT) at which the current log tape is filled and must be replaced.

Since each filled log tape contains 45,000 blocks, ALTFL contains N x 45,000 where N represents the number of the log tape currently being written.

p) ALSCT—internal cell containing the serial count of logging blocks stored in logging buffers. The contents of ALSCT are incremented by one for each logging block stored in a logging buffer.

q) ALONE—mask, 777 777 777 777. This mask is stored in each unused location of the logging block and indicates a non-data word. The total number of data and non-data words must equal 12 for each logging block.

r) ALTFN—constant, 45,000 (decimal).

s) K00001—constant, 1

t) Macros used—BEGIN*, RETURN*, QUEUE, TRNON.

The computer must be disabled during the execution of the MSLOGG subroutine.

6.12.2 Output Requirements

MSLOGG fills a 17-word logging block in the logging buffer, either TMALB1 or TMALB2, designated by MCALBL. The first five words of the block compose the heading and provide descriptive information relative to the logged data. The last 12 words of the block contain the logged data, with the last data word stored in the 17th word of the block, the second last data word in the 16th word of the block, etc. If less than 12 words are available for logging, the words following the heading but preceding the logged data contain the non-data octal mask, 777 777 777 777.

For every entry MSLOGG increments both the serial block count in ALSCT and the block per buffer count in ALBFL by one. The following output is conditional:

a) If the 17-word logging block fills the logging buffer (the contents of ALBFL and ALNOB are equal) the contents of MCALBL and MCALBT are exchanged so that the location of the alternate logging buffer will be contained in MCALBL. The B indicator is turned on for MYSTLT, the processor which writes the filled logging buffer on the log tape.

b) If MSLOGG finds that the log tape has been filled (contents of ALSCT and ALTFL are equal), MSLOGG also exchanges the contents of MCLTP1 and MCLTP2 so that the alternate tape will be used by

*BEGIN and RETURN are SOS macros and are described in MC 102, Program System Standards.

MYSTLT. An on-line message is queued to state that one of the log tapes is full and should be replaced. MCALM1 and MCALM2 are exchanged so that the message whose number is in MCALM1 refers to the tape now being written.

6.12.3 Method

Upon entry MSLOGG executes the BEGIN macro to save the index registers and the sense indicators. The logging block per buffer counter is incremented by one and the location of the block with the buffer is computed. MSLOGG begins setting up the five word heading in the logging block (see Figure 6-12). The second word of the calling sequence, the data identification, is stored in the first word of the heading. The data time tag from the MQ is stored as the second word of the heading. MSLOGG constructs a logging time tag of the same format and stores this as the third word of the heading. The logging block serial count is incremented by one and stored in the fourth word of the heading.

MSLOGG examines the hold phase indicator and, if the mission has entered the hold phase, stores a 7 in the prefix of the fourth word of the heading. The third word of the calling sequence, the data word count, is stored in the fifth word of the heading. MSLOGG sets the prefix of the fifth word of the heading to indicate mission phase: +0 for launch, -0 for abort, +1 for orbit, or -1 for reentry.

Every non-data location following the five word heading is filled with a non-data mask and MSLOGG completes the logging block by storing the data words forward from the last location in the logging block. The last data word is stored in the last location.

If the logging buffer is not filled, MSLOGG executes the RETURN macro to restore the index registers and sense indicators. Program control returns to the fourth word in the calling sequence.

If the logging buffer now contains the maximum number of logging blocks, the alternate buffer is set for the next logging. If the log tape is full, the primary and the alternate log tapes are effectively switched by exchanging the cells containing the tape data-select instructions. An on-line message states that one log tape has been filled and should be replaced with a blank. Whether or not the tape was full, MSLOGG turns on the B indicator for MYSTLT, the processor which writes the buffer on the log tape. The RETURN macro is executed and program control returns to the fourth word in the calling sequence.

6.12.4 Usage

MSLOGG is entered from the following calling sequence:

Location	Operation	Address, Tag, Decrement
α	TSX	MSLOGG, 4
$\alpha + 1$	PZE	,, Identification
$\alpha + 2$	PZE	Data Location,,Data Word Count
$\alpha + 3$	Return	

a) Storage Required—92 locations

b) MSLOGG uses:

1) Macros—BEGIN, RETURN, QUEUE and TRNON

2) Parameters—B, MNSTLT, MULTP1 and MULTP2

3) Communication cells—BCLAOR, MCALBL, MCALBT, MCALM1, MCALM2, MCHFSC, MCLTP1, MCLTP2, ALBFL, ALSCT and ALTFL

4) Tables—TMALB1, TMALB2 and TM8.3M

5) Constants—K00001, ALNOB, ALNOW and ALTFN

6) Mask—ALONE

c) Time Required—2.71 milliseconds (average)

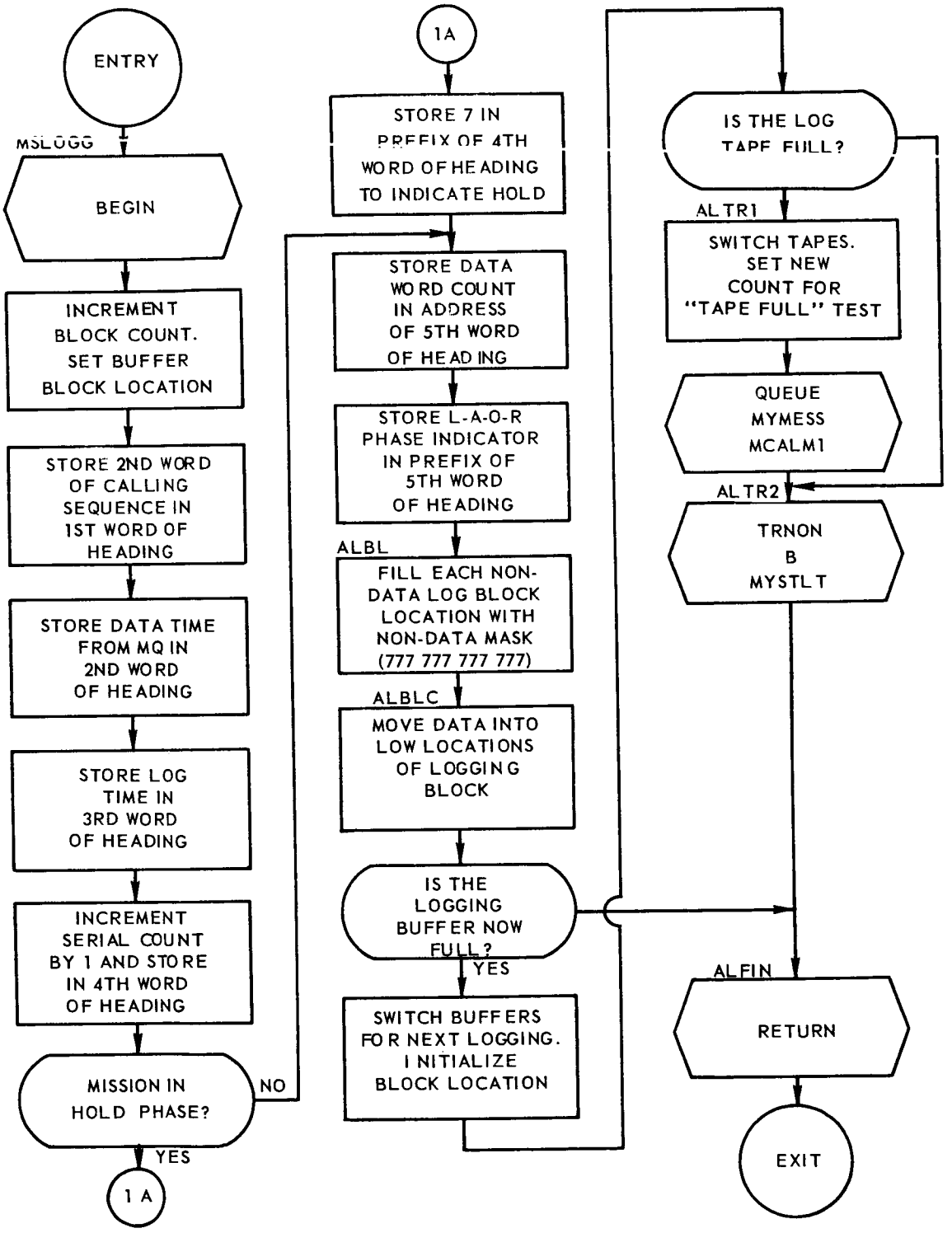


FIGURE 6-13. MSLOGG SUBROUTINE FLOW CHART

6.13 MONITOR LOGGING PROCESSOR (MYSTLT)

MYSTLT receives control from M0PRIO after the Logging Subroutine, MSLOGG, has completely filled a logging buffer. The Logging Processor initiates transmission to write the logging buffer on the log tape.

The flow chart for the MYSTLT processor is shown in Figure 6-14.

6.13.1 Input Requirements

Input to MYSTLT consists of:

a) MCALBT—communication cell containing in the address the location of the logging buffer, either TMALB1 or TMALB2, to be written on the log tape. This cell is set by MSLOGG.

b) MCLPT1—communication cell containing the write-select instruction, either WTBB MULTP1 or WTBB MULTP2, for writing the log tape. MULTP1 and MULTP2 are parameters designating the primary and secondary, or alternate, logging tapes, B6 and B7, respectively. This communication cell is used by MYSTLT as a program; it is executed (XEC). MCALBT is set by MSLOGG.

c) AMCOM—cell internal to MYSTLT containing the data channel command IOCT **, 0 + 0, 170.

d) Macros used—QENBZ, TRNOF and TRNON.

6.13.2 Output Requirements

MYSTLT provides the address for the data channel command and initiates transmission to write a 170-word logging buffer on the log tape.

6.13.3 Method

MYSTLT receives control from M0PRIO after the Logging Subroutine, MSLOGG, has completely filled one of the two logging buffers and has turned on the B indicator for MYSTLT. When MSLOGG has filled a logging buffer, the subroutine provides all necessary input data for MYSTLT—MYSTLT makes no logical decisions.

MYSTLT turns on its own A indicator. Then the address of the buffer to be written on tape is obtained from MCALBT and stored in the address of the data channel command in location AMCOM. The computer is disabled and MYSTLT executes MCLTP1 to select the tape unit, and resets and loads the channel from AMCOM.

MYSTLT then turns on its own C indicator to suppress itself until after the transmission is completed (the C indicator is turned off by MTENLG). MYSTLT turns off its A indicator and exits to M0PRIO.

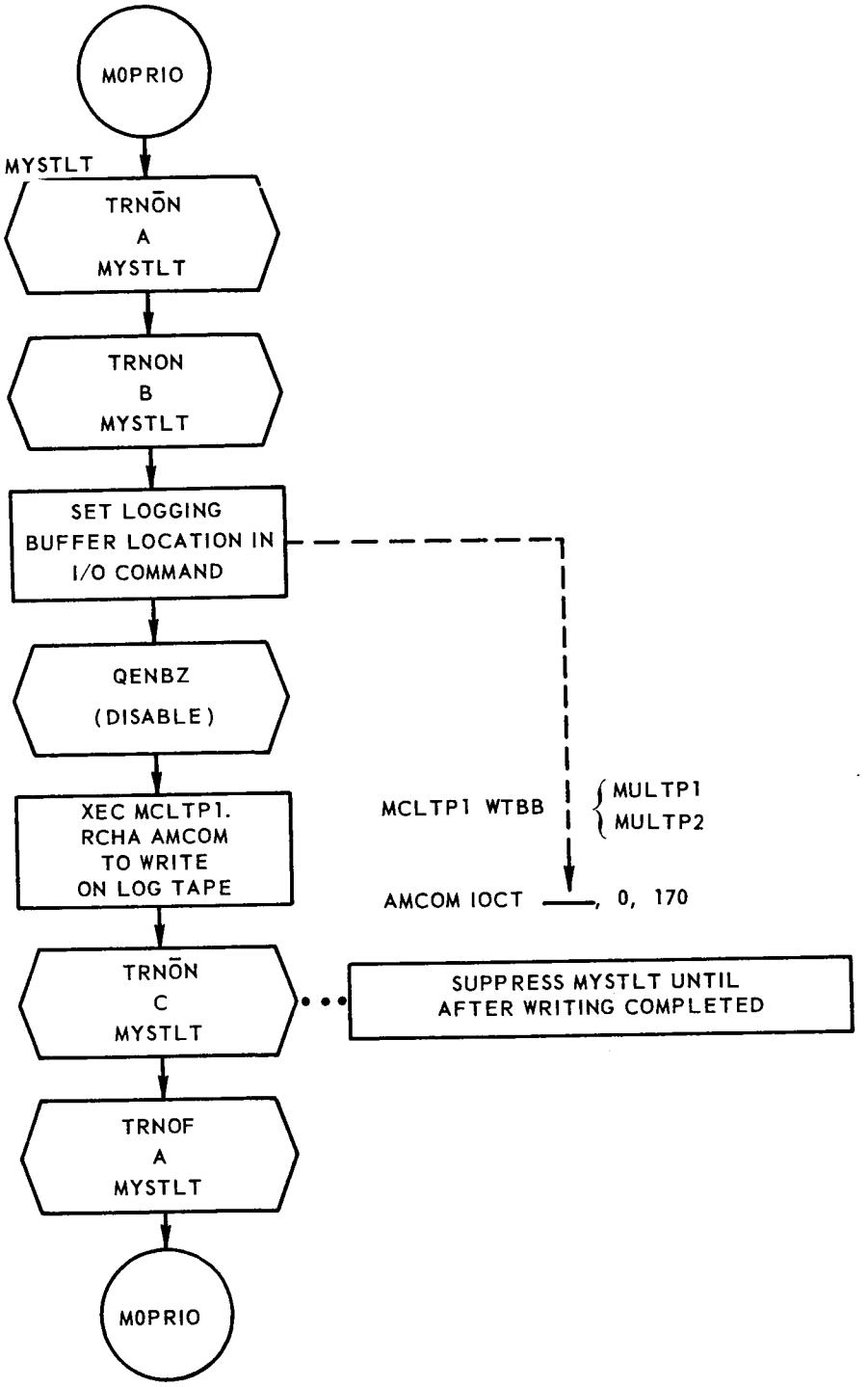


FIGURE 6-14. MYSTLT PROGRAM FLOW CHART

6.13.4 Usage

MYSTLT is entered from, and exits to M0PRIO.

a) Storage Required—14 locations.

b) MYSTLT uses:

1) Macros—QENBZ, TRNOF and TRNON

2) Parameter—A, C, MNSTLT, MULTP1 and MULTP2

3) Communication cells—MCALBT and MCLTP1

4) Tables—TMALB1 and TMALB2

c) Time Required—0.384 milliseconds

6.14 MONITOR END-OF-LOGGING TRAP PROCESSOR (MTENLG)

MTENLG receives control from the channel B trap following the writing of a logging buffer on the Log Tape. MTENLG checks for redundancy during the tape writing and removes the suppression condition from MYSTLT.

The flow chart for the MTENLG processor is shown in Figure 6-15.

6.14.1 Input Requirements

Input to MTENLG consists of:

a) Tape check indicator

b) KT0001—constant, 000 001 000 000, a 1 in the tag.

c) Macros used—SAVE, TRNOF.

6.14.2 Output Requirements

If a redundancy had occurred in writing the logging buffer, MTENLG stores a 1 in the tag of the next logging buffer, TMALB1 or TMALB2, to be written on the log tape.

MTENLG also unsuppresses MYSTLT so that the current logging buffer, when filled, may be written on the log tape.

6.14.3 Method

MTENLG receives control from a trap transfer instruction (TTR MTENLG) in the channel B trap control location. The trap processor executes the SAVE macro to preserve the condition of the interrupted program.

The tape check indicator is examined for a redundancy in writing the log tape. If a redundancy had occurred, a 1 is stored in the tag bit of the next buffer to be written on the log tape. (When interpreting the log tape, the tag of record n + 1 is checked to determine if redundancy had occurred in writing record n.)

MTENLG turns off the C indicator for MYSTLT which allows MYSTLT to write the next logging buffer as soon as it becomes filled. Program control returns to M0PRIO.

6.14.4 Usage

MTENLG is entered from a trap transfer in the channel B trap control location (00015); MTENLG exits to M0PRIO.

a) Storage Required—11 locations

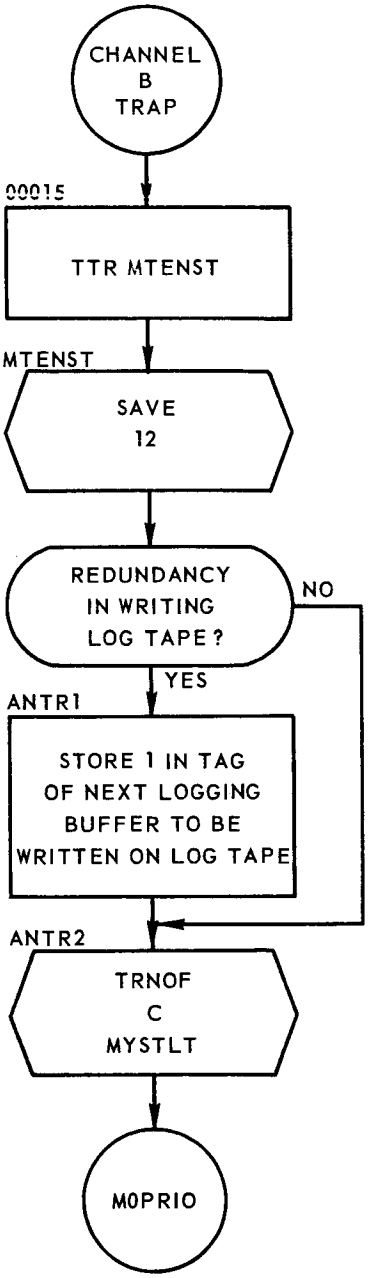


FIGURE 6-15. MTENLG PROGRAM FLOW CHART

b) MTENLG uses:

1) Macros—SAVE and TRNOF

2) Parameters—C and MNSTLT

3) Communication cell—MCALBL

4) Tables—TMALB1 and TMALB2

5) Constants—KT0001

c) Time Required—1.26 milliseconds

Section 7
LOADING THE SYSTEM

Prior to loading the Bermuda Program System for an actual mission, the final modifications are made to the program (for example, temporary instructions required for simulated testing are removed) and the system tape is written in absolute binary with special error-correcting codes and the program is preceded by a self-loading sequence.

The final modifications to the program are incorporated by MXMRGE, an external program which provides the capabilities for modifying a squoze tape. (MXMRGE is described in MC 107, External System Programs.) The output of MXMRGE is a job tape for an SOS compilation or execution. MXSTW1 edits the SOS B1 tape to produce a self-loading file as the first file on the Bermuda System tape. It then writes the remainder of the Bermuda System tape with the error correction codes using the MSWECC subroutine.

The self-loading tape is now placed on tape unit A1. When the Load Tape button is pressed, the self-loading sequence brings in the first file containing the loading program MXLOAD. This program reads in the remaining file on the tape, record by record, with the MSLOAD subroutine; it removes the error correcting codes using the MSRECC subroutine, and stores the record in the proper location in core storage.

After the complete tape has been read into core, control transfers to the Main Initialization Controller, M0INTL, which begins initializing the system. The last operation of M0INTL turns on the B indicator for the Monitor Initialization Processor, MYINTL, which completes the initialization process.

These programs—MXSTW1, MSWECC, MXLOAD, MSLOAD, MSRECC, M0INTL and MYINTL—are described in detail in this section.

7.1 MERCURY SYSTEM TAPE WRITER PROGRAM (MXSTW1)

MXSTW1 writes a self-loading Mercury System tape. The flow chart for the MXSTW1 program is shown in Figure 7-1.

7.1.1 Input Requirements

Input to MXSTW1 is the Bermuda System squoze tape, including all modifications intended for the final running program. MXSTW1 requires the MSWECC subroutine (see subsection 7.2) and the Share subroutine for converting binary-coded-decimal to card image for on-line printing.

7.1.2 Output Requirements

The output from MXSTW1 consists of an absolute self-loading Mercury System tape, illustrated in Figure 7-2.

7.1.3 Method

When SOS loads its System Mediary Input tape (SYSMIT) from B1 into the computer, a transfer card (TCD) gives control to MXSTW1. MXSTW1 writes a self-loading file (MXLOAD) as the first file on the Mercury System tape and then writes the file containing the Bermuda Program System. The latter file includes error correcting codes.

7.1.4 Usage

a) Operator's Procedures:

1) Ready the standard SOS system tapes, with A3 as the input tape.

2) Ready the on-line printer.

3) Press CLEAR and LOAD TAPE.

4) SOS writes its SYSMIT tape on B1 and transfers control to the MXSTW1 program. The program halts after printing an on-line message designating the control transfer.

5) Ready a blank tape on A6.

6) Pressing START produces the absolute self-loading Mercury System tape on A6.

7) MXSTW1 provides an on-line printout stating that the Mercury System tape has been successfully written. (A duplicate tape may be

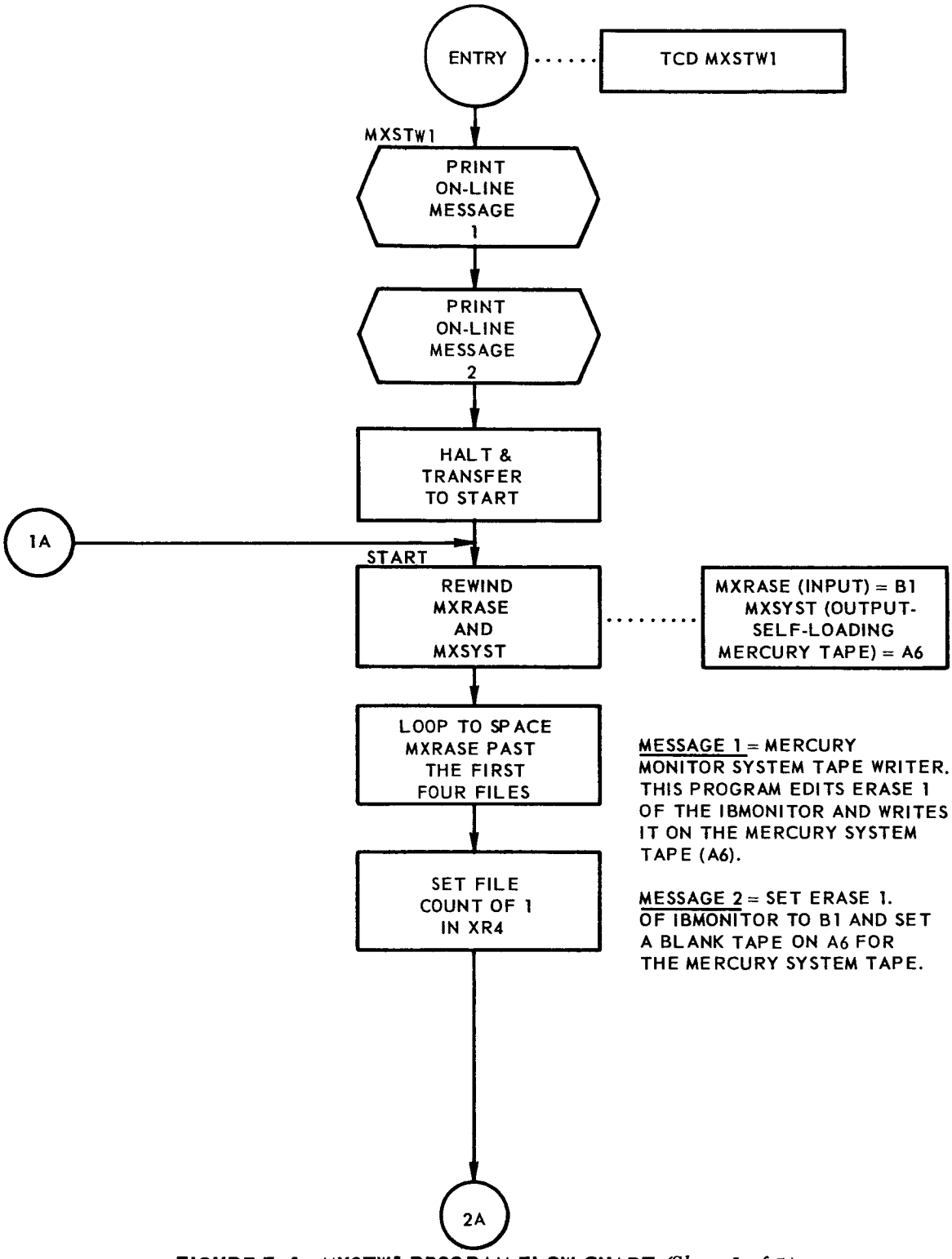


FIGURE 7-1. MXSTW1 PROGRAM FLOW CHART *(Sheet 1 of 5)*

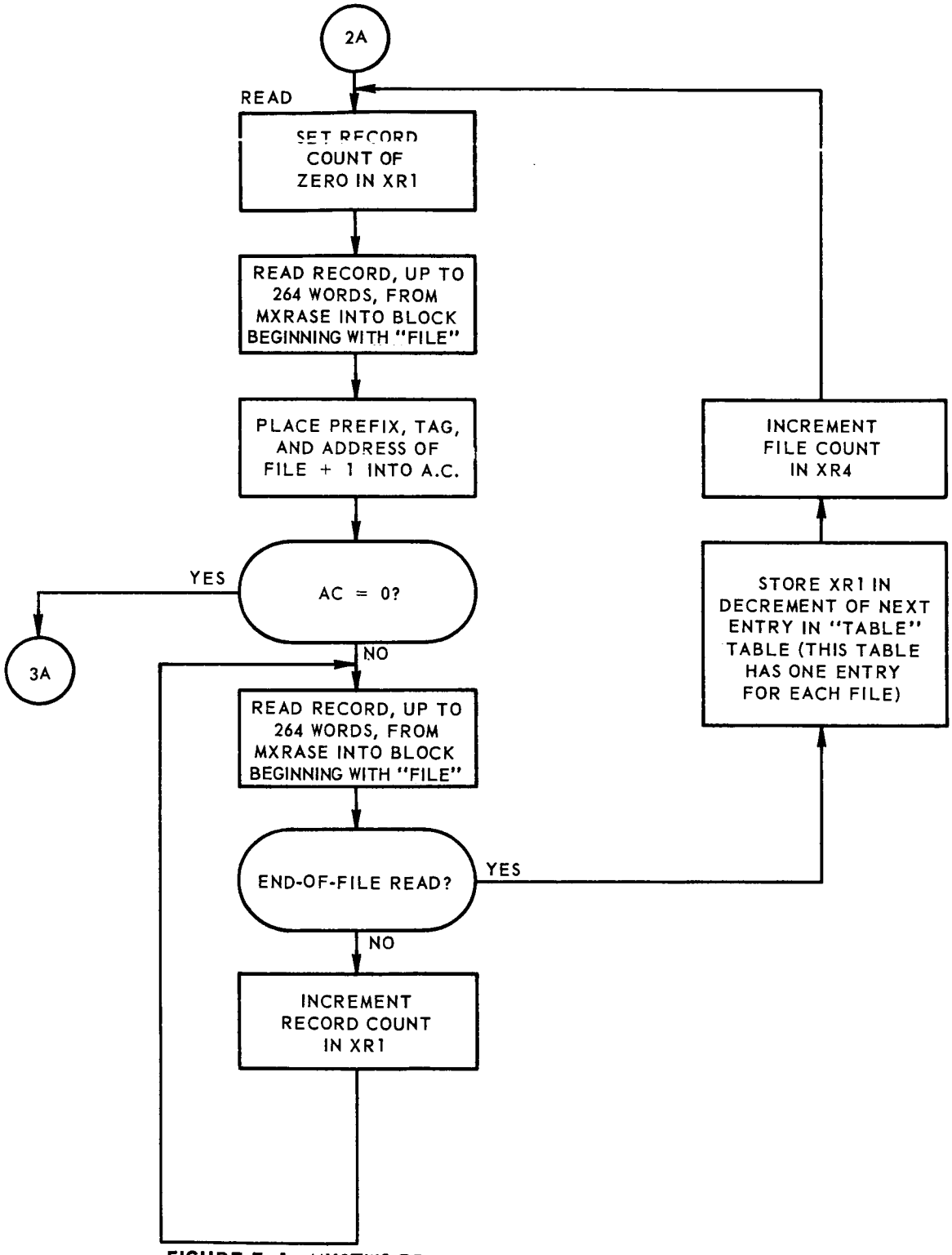


FIGURE 7-1. MXSTW1 PROGRAM FLOW CHART *(Sheet 2 of 5)*

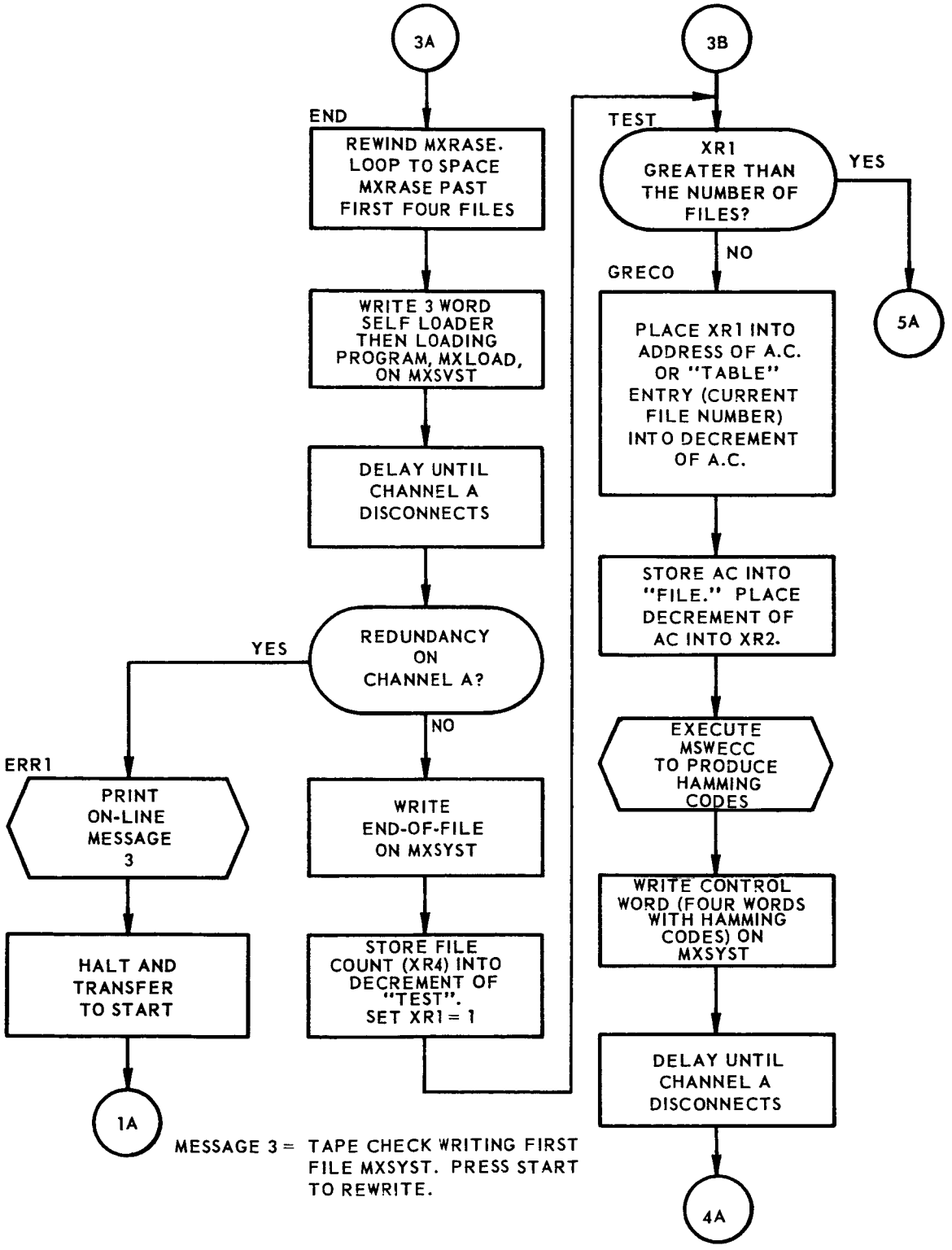


FIGURE 7-1. MXSTW1 PROGRAM FLOW CHART *(Sheet 3 of 5)*

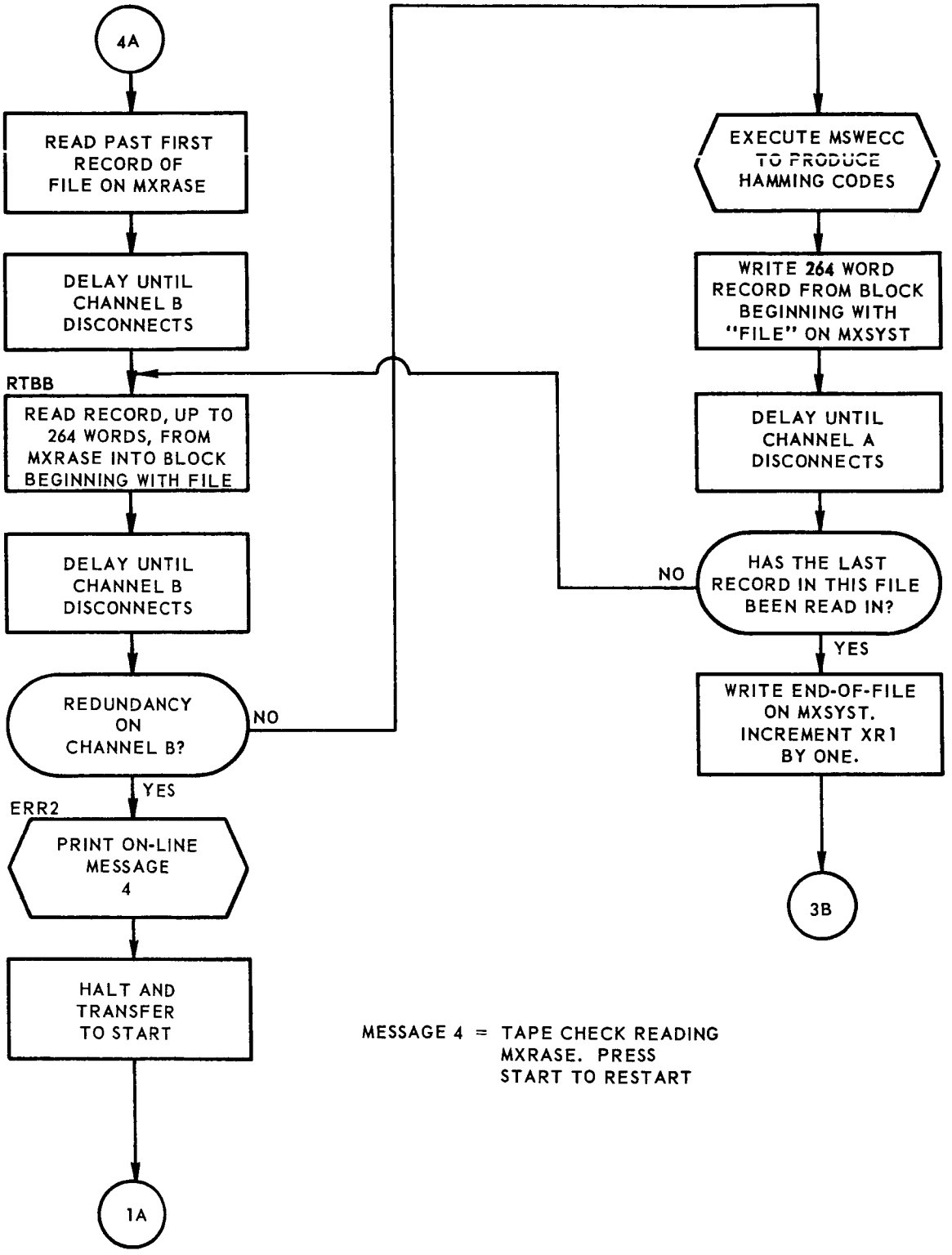


FIGURE 7-1. MXSTW1 PROGRAM FLOW CHART *(Sheet 4 of 5)*

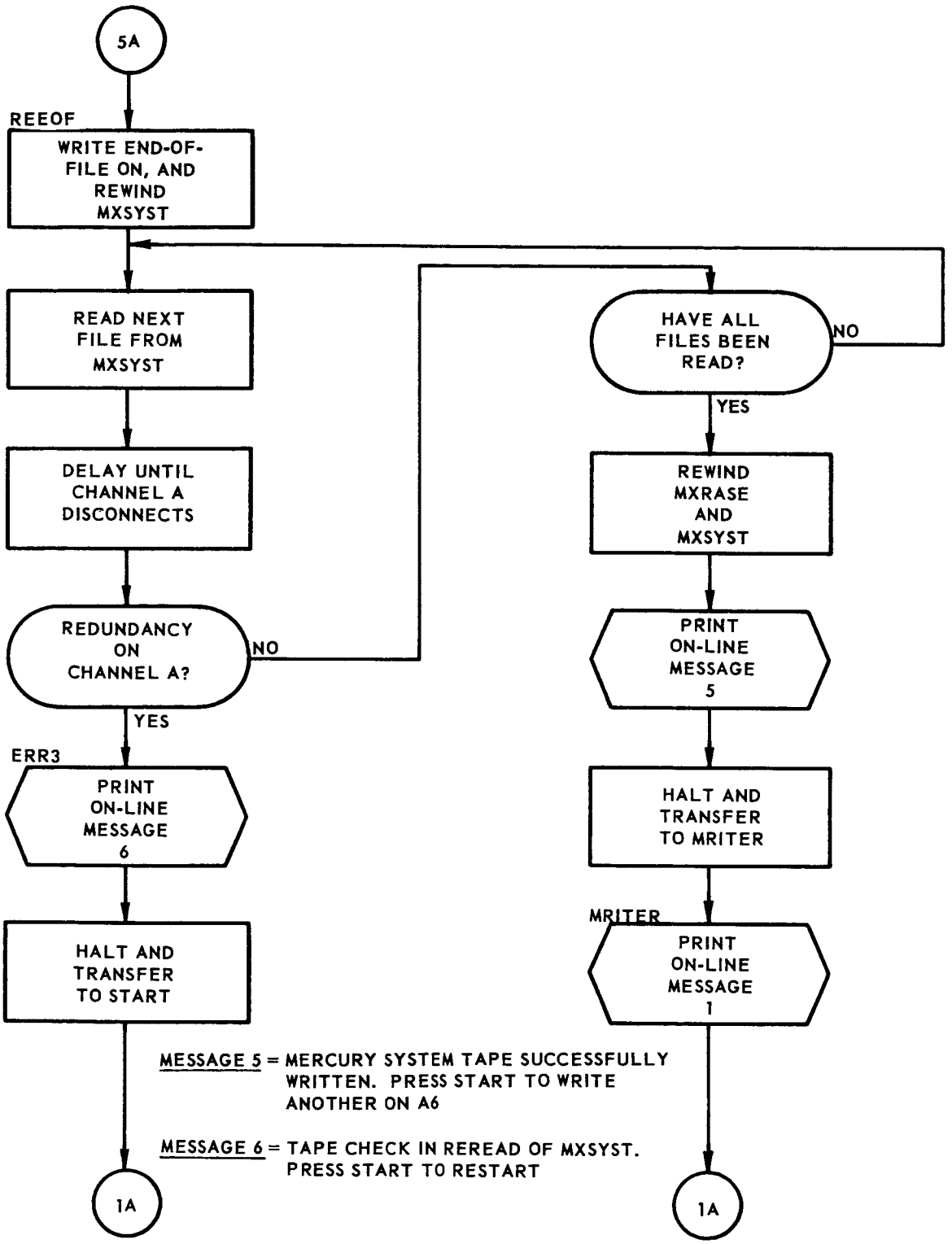


FIGURE 7-1. MXSTW1 PROGRAM FLOW CHART *(Sheet 5 of 5)*

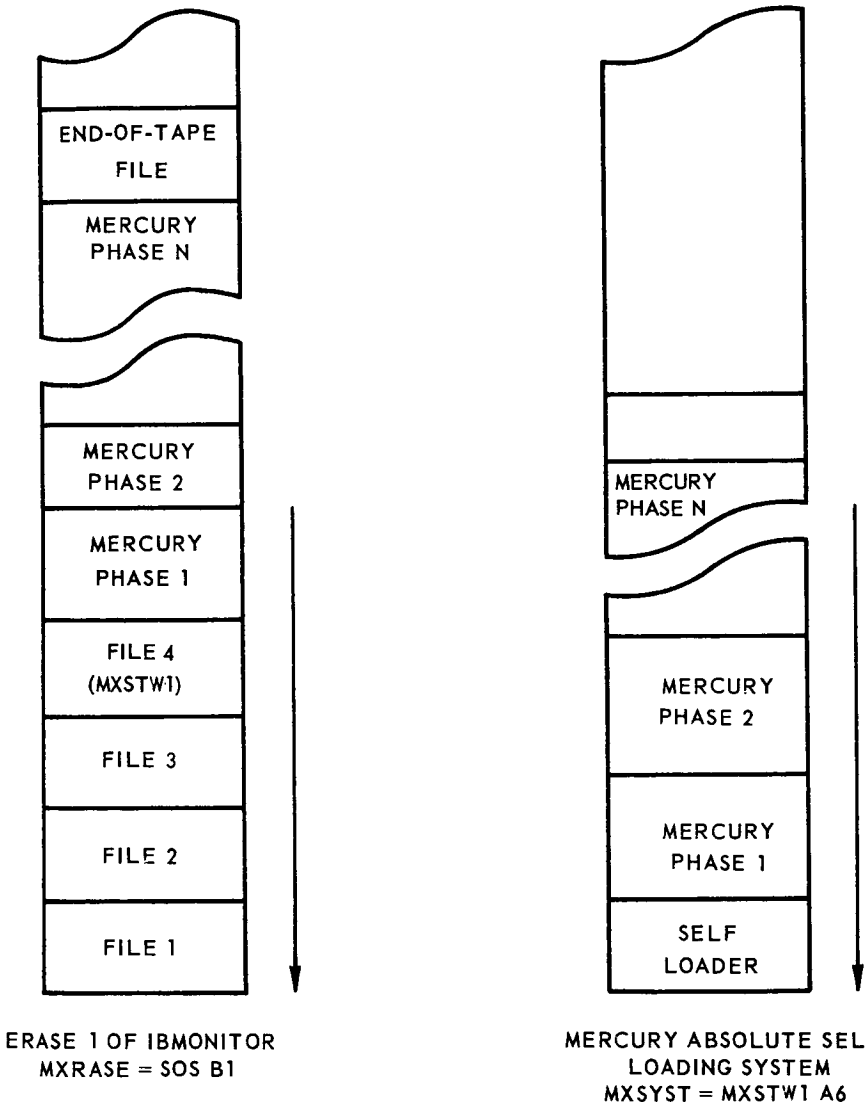


FIGURE 7-2. INPUT & OUTPUT TAPES FOR MXSTW1

written by placing a blank on A6 and pressing START.) The Mercury System tape is now placed on A1 and, when LOAD TAPE is pressed, it loads itself into the computer.

b) Error Conditions—Errors during the execution of MXSTW1 result in program halts. On-line messages identify the error condition and specify error correcting procedures.

7.2 ERROR CORRECTION CODE WRITER SUBROUTINE (MSWECC)

MSWECC writes information on tape with certain check words interleaved with the original record in such a way that automatic double error detection and automatic single error detection and correction is possible for each of the 36 bit positions of an IBM 709 computer word.

The flow chart for the MSWECC subroutine is shown in Figure 7-3.

7.2.1 Input Requirements

MSWECC requires an input of N consecutive words in core storage, where N is less than 503.

7.2.2 Output Requirements

MSWECC expands the input block N with interleaved check words to produce an expanded block, the size of which is given by the formula below (the size is placed in the address of the AC):

$$2^{K-1} + K + 1 \leq N < 2^{K} - K$$

where N is equal to the orginal block length and K is equal to the number of check words. The following table gives the values for the length of the output block (one additional word is required for parity checking):

N	K	Storage Required
1	2	4
2 - 4	3	6 - 8
5 - 11	4	10 - 16
12 - 26	5	18 - 32
27 - 57	6	34 - 64
58 - 120	7	66 - 128
121 - 247	8	130 - 256
248 - 502	9	258 - 512

7.2.3 Method

An example will best illustrate the method of MSWECC and MSRECC: If the original message is

1 0 1 0 1 0 1 0 1 0 1

room is made for the check bits by spacing in the following way:

position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
bit	X	X	1	X	0	1	0	X	1	0	1	0	1	0	1

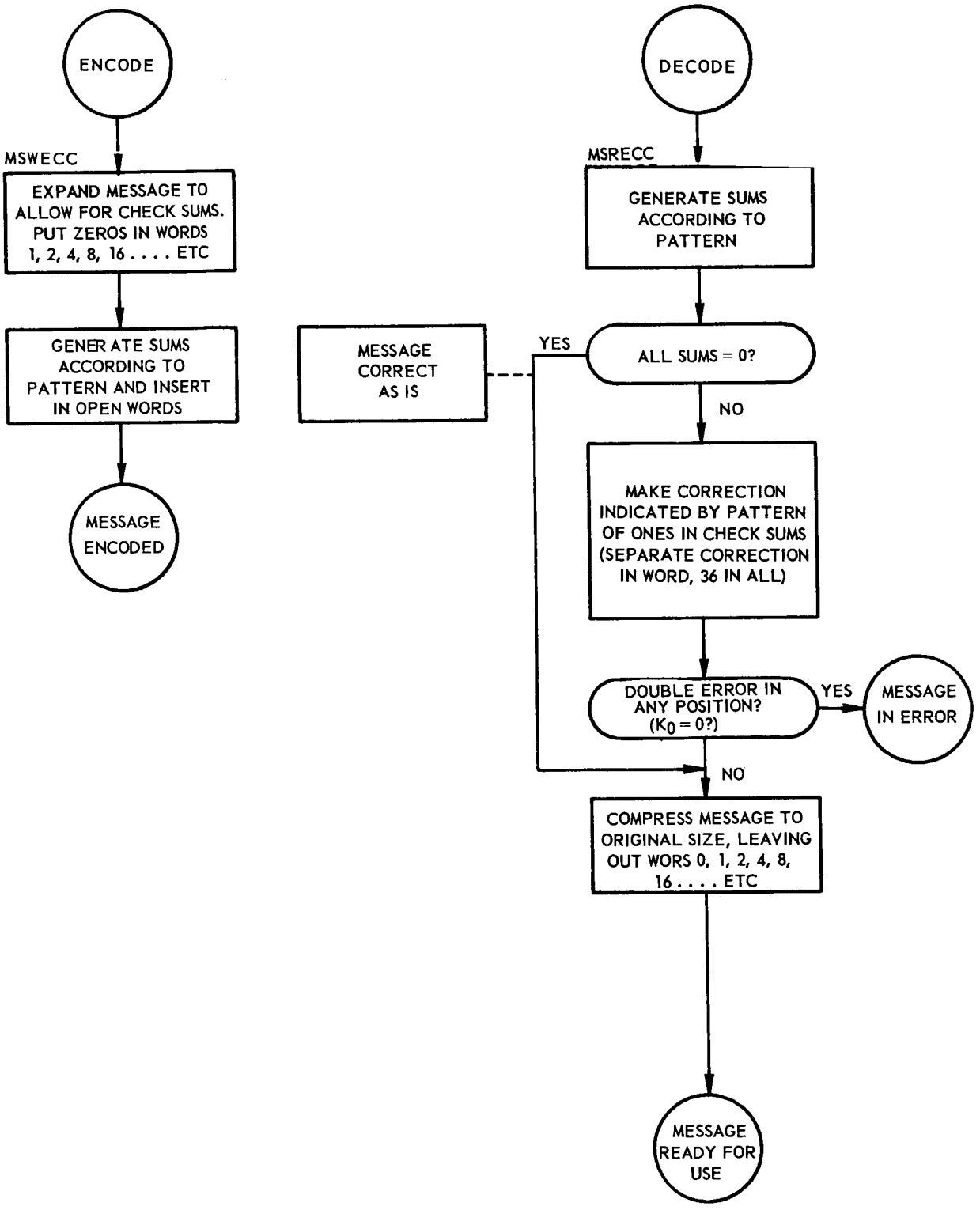


FIGURE 7-3. MSWECC AND MSRECC FLOW CHARTS

That is, the check bits go in the positions 1, 2, 4, and 8.

Thirteen may be considered as the sum of 8, 4, and 1; and the 13th bit is used in parity sums which make up the 8th, 4th, and 1st positions but does not enter the parity sum which goes into position 2. Similarly, bit 5 enters into sums 1 and 4; bit 10 enters into sums 8 and 2; and bit 15 enters into all four sums. Any bit enters into a unique combination of sums, since any number from 1 to 15 can be broken down into a sum of 8, 4, 2, and 1 in only one way if each is used at most once. The exact formulas for calculating the parity sums are:

$$C_1 = D_3 + D_5 + D_7 + D_9 + D_{11} + D_{13} + D_{15}$$

$$C_2 = D_3 + D_6 + D_7 + D_{10} + D_{11} + D_{14} + D_{15}$$

$$C_4 = D_5 + D_6 + D_7 + D_{12} + D_{13} + D_{14} + D_{15}$$

$$C_8 = D_9 + D_{10} + D_{11} + D_{12} + D_{13} + D_{14} + D_{15}$$

where the plus sign (exclusive or) represents addition without carries. For the message above

$$C_1 = 1 + 0 + 0 + 1 + 1 + 1 + 1 = 1$$

$$C_2 = 1 + 1 + 0 + 0 + 1 + 0 + 1 = 0$$

$$C_4 = 0 + 1 + 0 + 0 + 1 + 0 + 1 = 1$$

$$C_8 = 1 + 0 + 1 + 0 + 1 + 0 + 1 = 0$$

and the encoded message reads:

position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
bit	1	0	1	1	0	1	0	0	1	0	1	0	1	0	1

This is the form in which the message is stored or transmitted.

The decoding process is performed by MSRECC after the message is received but before it is used. The check sums are generated according to the same pattern (including the check position in each sum). An error can be detected because all sums are zero if there is no error. If a one-bit error has occurred, it can be located by the pattern of sums which are affected.

Position		Pattern			
C		1	2	4	8
*C	1	X			
C	2		X		
	3	X	X		
C	4			X	
	5	X		X	
	6		X	X	
	7	X	X	X	
C	8				X
	9	X			X
	10		X		X
	11	X	X		X
	12			X	X
	13	X		X	X
	14		X	X	X
	15	X	X	X	X

Single Error Detecting Codes

Position		Pattern				
		0	1	2	4	8
*C	0	X				
C	1	X	X			
C	2	X		X		
	3	X	X	X		
C	4	X			X	
	5	X	X		X	
	6	X		X	X	
	7	X	X	X	X	
C	8	X				X
	9	X	X			X
	10	X		X		X
	11	X	X	X		X
	12	X			X	X
	13	X	X		X	X
	14	X		X	X	X
	15	X	X	X	X	X

Double Error Detecting Codes

*C indicates check position.

The pattern for single error detecting code, illustrated above, shows that each bit yields a unique pattern when it is in error. For example, if bit 14 changes from a zero to a 1, it yields the following:

position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
bit	1	0	1	1	0	1	0	0	1	0	1	0	1	1	1

It also produces the following sums:

$$K_1 = 1 + 1 + 0 + 0 + 1 + 1 + 1 + 1 = 0$$

$$K_2 = 0 + 1 + 1 + 0 + 0 + 1 + 1 + 1 = 1$$

$$K_4 = 1 + 0 + 1 + 0 + 0 + 1 + 1 + 1 = 1$$

$$K_8 = 0 + 1 + 0 + 1 + 0 + 1 + 1 + 1 = 1$$

Since the only bit which enters into the three erroneous sums, K_2, K_4, and K_8, is D_{14} $(2 + 4 + 8 = 14)$, it is the erroneous one and must be changed from 1 to 0 to restore the message. (This technique also corrects erroneous check bits.)

If two bits are in error, the above method does not work. However, the code has been extended by adding a single check bit in the first position (position 0) to make it double error detecting. Since zero may be thought of as being part of each number ($13 = 8 + 4 + 1 + 0$; $5 = 4 + 1 + 0$; $10 = 8 + 2 + 0$; etc.), this bit, C_0, is the sum of all other bits, but does not enter into any other sums itself. With this arrangement none of the check bits is used in calculating any of the others. (See above illustration of double error detecting codes.) Thus, on any single error, the K_0 sum (over all bits) will be 1. On a double error, however, K_0 will be zero, but some of the other K's will not be zero. Therefore, by checking K_0 before attampting a correction, it can be determined whether this is an uncorrectable double error ($K_0 = 0$) or a correctable single error ($K_0 = 1$).

7.2.4 Usage

a) Calling Sequence:

```
TSX     MSWECC, 4

PZE     A, 0, N          (where A is the first location of the input
                          block and N is the block length)

Normal Return
```

b) Storage Required—171 locations.

c) Time Required—$(5.0 + 0.55\ N)$ milliseconds, where N is defined above.

7.3 MERCURY SYSTEM TAPE LOADER (MXLOAD)

MXLOAD receives control at the completion of the Load Tape sequence and using the MSLOAD subroutine completes the loading of the Bermuda System Tape. The flow charts for the MXLOAD program and the MSLOAD subroutine are shown in Figures 7-4 and 7-5, respectively.

7.3.1 Input Requirements

MXLOAD requires the MSLOAD subroutine, which in turn requires the MSRECC subroutine. The auxiliary Bermuda System Tape must be placed on tape unit A4.

7.3.2 Output Requirements

MXLOAD completes the loading of the Bermuda System Tape and transfers control to M0INTL.

7.3.3 Method

MXLOAD checks for an indication of redundancy while the MXLOAD and associated loading programs were being read into the computer. If a redundancy has occurred, MXLOAD rewinds the tape and attempts to load the first file once again. Another redundancy check will produce a halt in the program.

If the redundancy does not persist (or did not happen), MXLOAD transfers to MSLOAD to read in the second file of the Bermuda System Tape. MSLOAD reads in a record at a time, executes MSRECC to remove the error correction codes, and, if no uncorrectable errors are found by MSRECC, MSLOAD stores the record in the proper core storage location.

If uncorrectable errors are detected in reading the primary system tape, MXLOAD attempts to read in the auxiliary tape. If uncorrectable errors are detected in reading the system auxiliary tape, MXLOAD comes to a halt. If a further attempt at reading the primary tape is requested, the Start button must be pressed to repeat the cycle.

7.3.4 Usage

MXLOAD is entered from the instruction placed in core location 00003 by the load tape sequence. MXLOAD exits to M0INTL following a successful loading of the Bermuda System Tape.

a) Storage Required—401 locations

b) Error corrections—any halt signifies an error. If another attempt at reading the system tape is desired, the Start button is pressed.

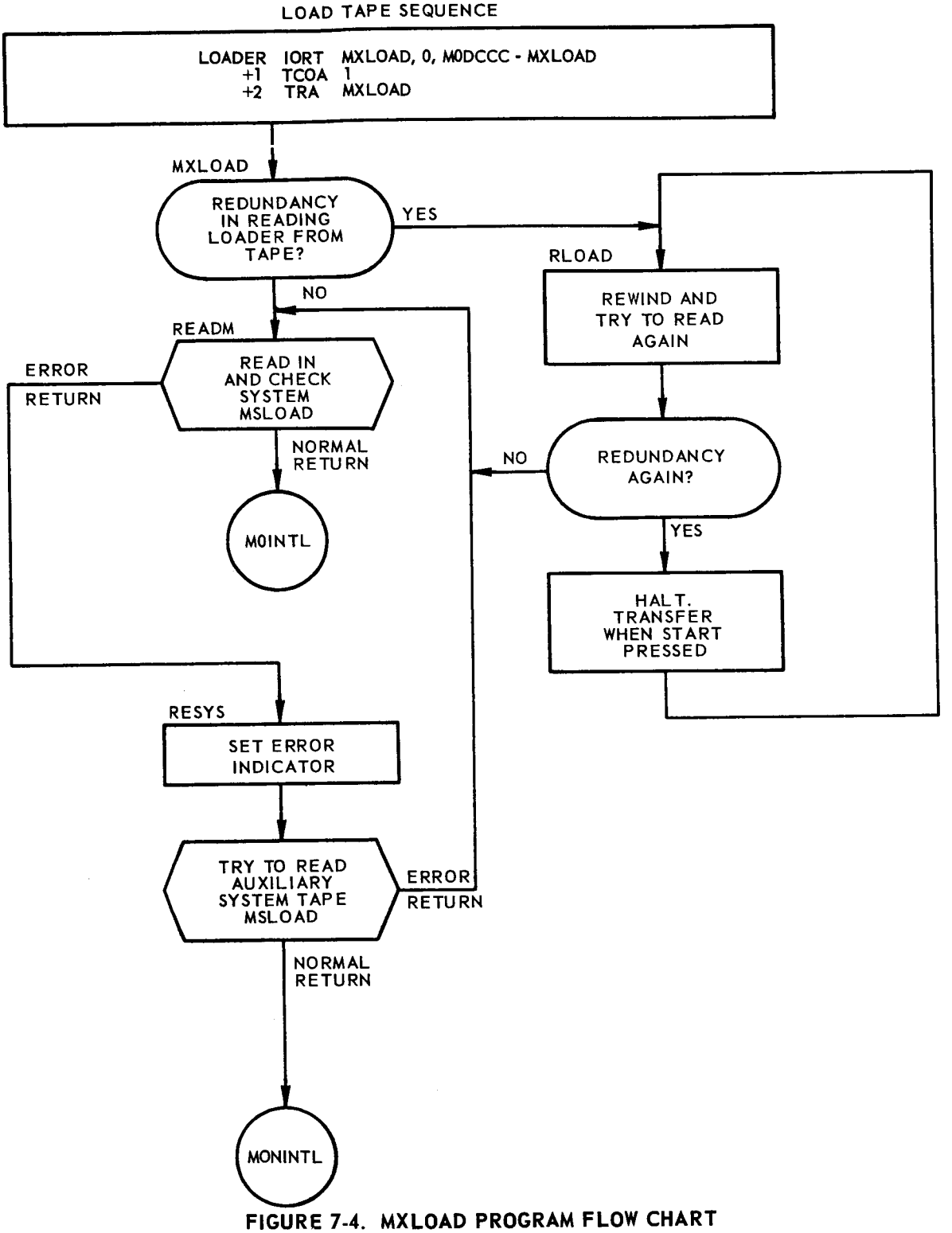


FIGURE 7-4. MXLOAD PROGRAM FLOW CHART

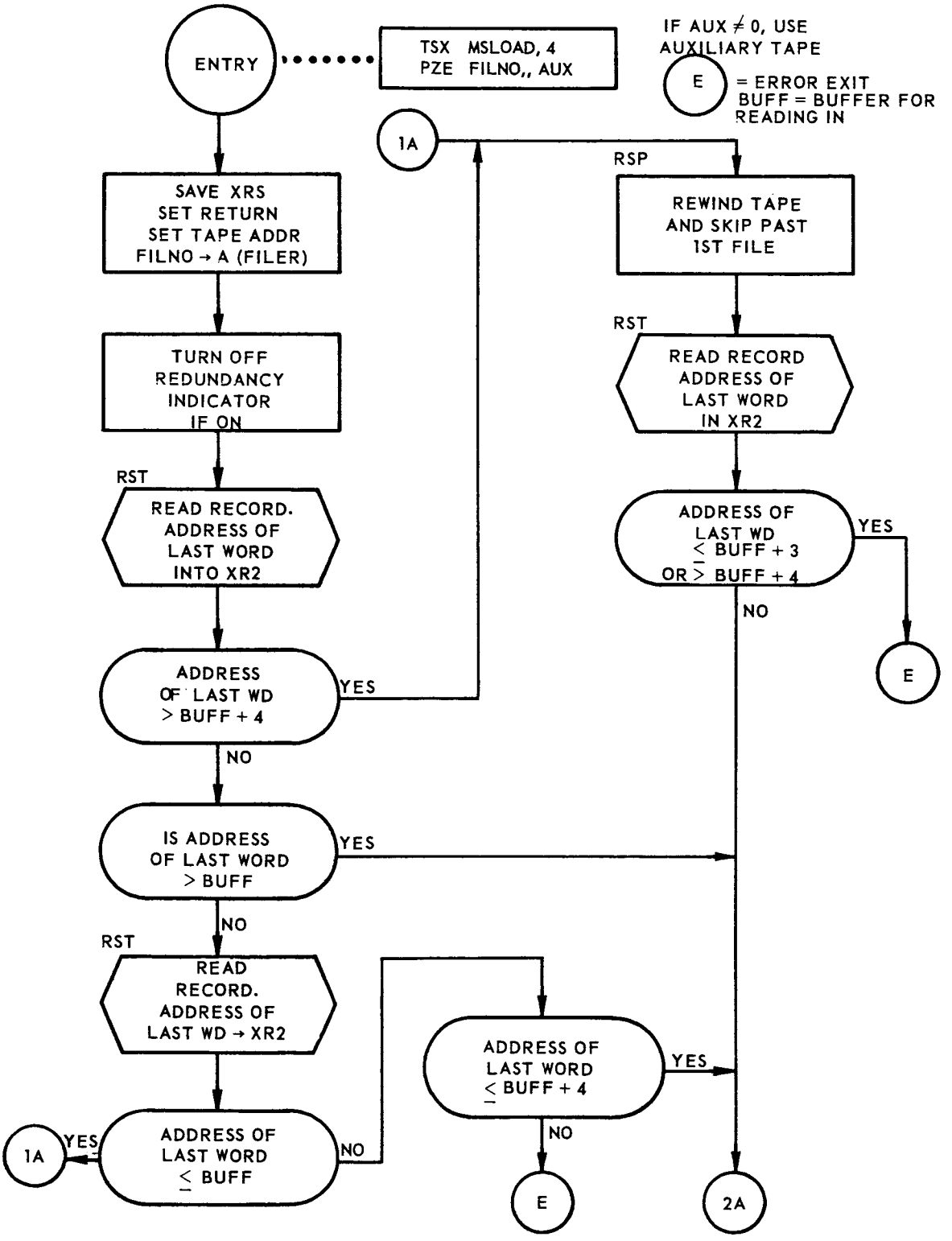


FIGURE 7-5. MSLOAD PROGRAM FLOW CHART *(Sheet 1 of 3)*

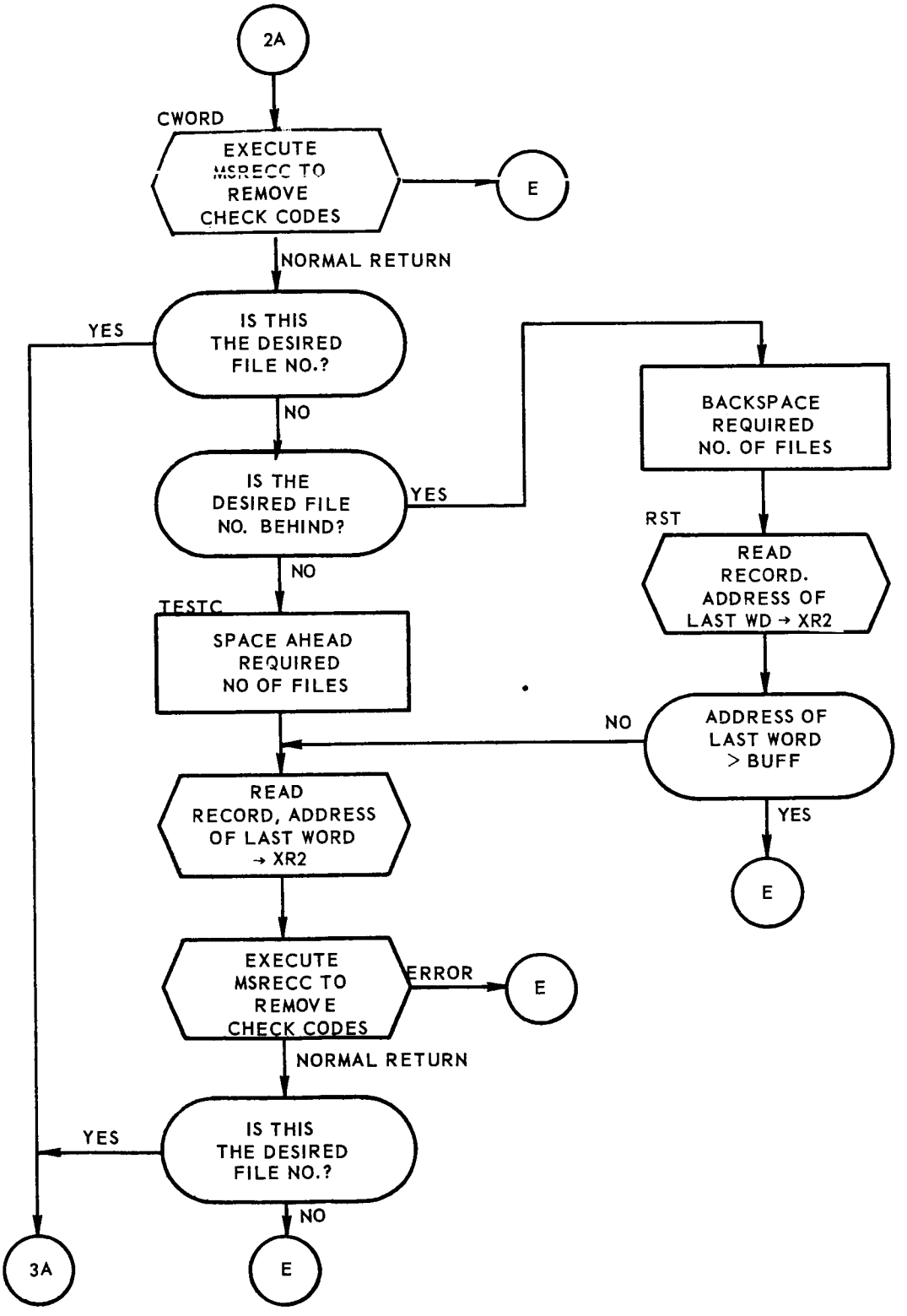


FIGURE 7-5. MXLOAD PROGRAM FLOW CHART *(Sheet 2 of 3)*

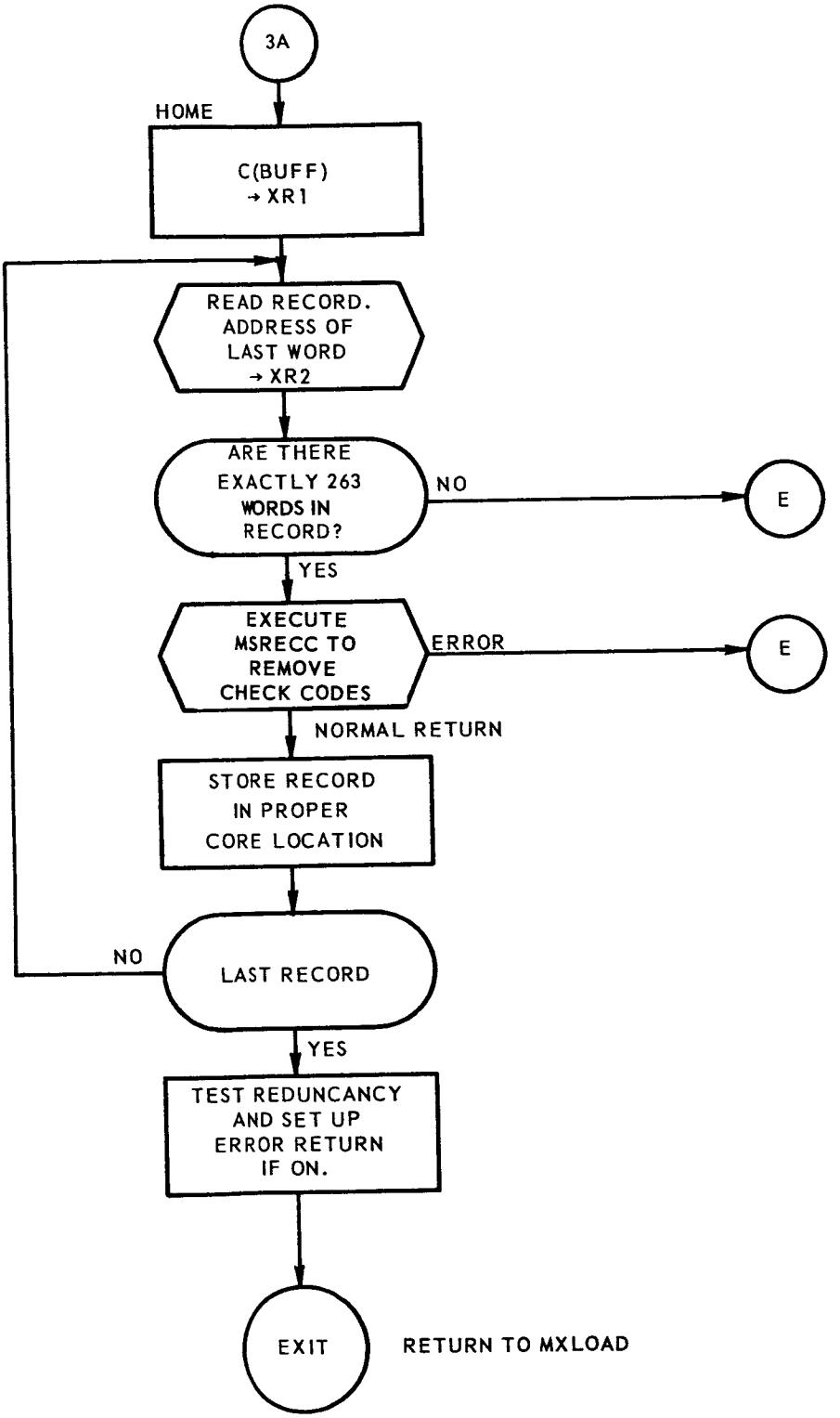


FIGURE 7-5. MXLOAD PROGRAM FLOW CHART *(Sheet 3 of 3)*

7.4 ERROR CORRECTION CODE READER SUBROUTINE (MSRECC)

MSRECC uses the check words formed by MSWECC to detect a double error or to detect and correct a single error in an IBM 709 computer word. The flow chart for the MSRECC subroutine is shown in Figure 7-3.

7.4.1 Input Requirements

MSRECC requires as input the expanded record which was produced by MSWECC.

7.4.2 Output Requirements

MSRECC examines the check sums and, if no errors or only single errors are observed in any of the 36 bit positions of the computer word, the check sums are removed and the original record is reconstructed. No notification is made of the corrected single errors. Double errors produce an error return from the subroutine.

7.4.3 Method

The method of MSRECC is given in subsection 7.2.3.

7.4.4 Usage

a) Calling Sequence:

```
TSX     MSRECC, 4

PZE     A, 0, N'        (where A is the first location of the input block
                         and N' is the total block length including parity
Error return             and check words)

Normal return
```

b) Storage Required—219 locations.

c) Time Required—(5.0 + 0.55N + 1.1E) milliseconds where N is the original block length and E is the number of errors.

7.5 MAIN CONTROLLER INITIALIZATION PROGRAM (M0INTL)

After the Bermuda Program System is read into core storage, various indicators and cells must be set to initial conditions in preparation for the mission computations. M0INTL begins the process of initialization of the Mercury Program System and sets controls for the Monitor Initialization Processor, MYINTL, to complete the process.

The flow chart for the M0INTL program is shown in Figure 7-6.

7.5.1 Input requirements

Input to M0INTL consists of:

a) The following parameters specify tape drives for the tapes used in the Mercury Program System:

1) MUPRIM—The primary Mercury Program System Tape, A1.

2) MUAUXL—The auxiliary Mercury Program System Tape, A4.

3) MUTAPA—The Message Tape, A6.

4) MUSCTP—The Station Characteristics Tape, A7.

5) MULTP1—The initial Log Tape, B6.

6) MULTP2—The reserve Log Tape, B7.

b) MCTHSM—communication cell used as a constant, -000 000 000 000.

c) CXMS2—internal cell containing the instruction: TTR MTENLG.

d) CXMS3—internal cell containing the instruction: TTR MTFLPT.

e) CXMS4—internal cell containing the instruction: TTR M0DCCC.

f) KD0001—constant, 000 001 000 000.

g) Macro used—TRNON.

7.5.2 Output Requirements

Output from M0INTL consists of:

a) TMHSFP, TMHSVE—two 24-cell buffer tables which receive the high-speed input data from the AN/FPS-16 and Verlort radars via DCC subchannel 1. Each cell is initially set to minus zero.

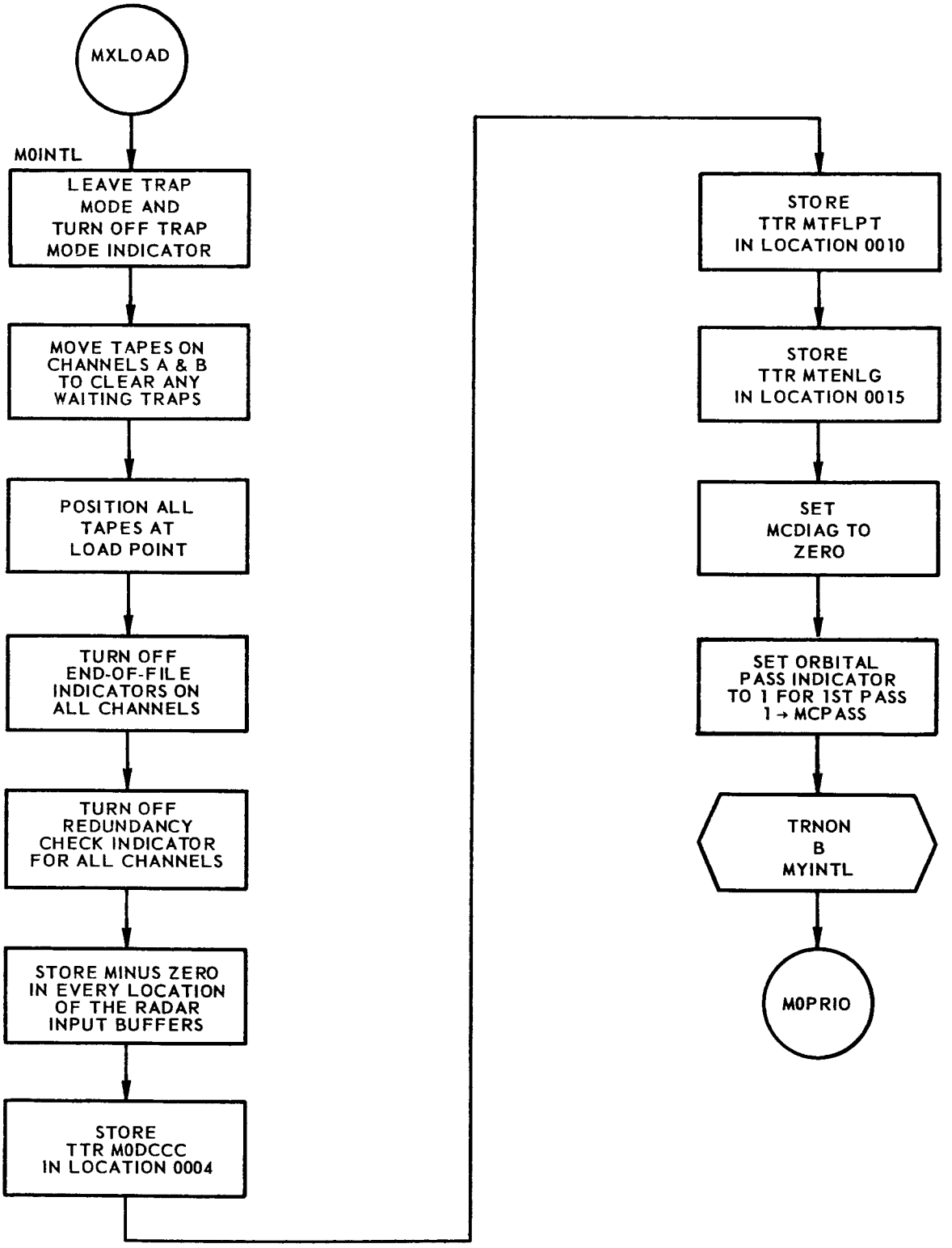


FIGURE 7-6. MOINTL PROGRAM FLOW CHART

b) MCDIAG—contains the count of the number of times M0PRIO gives program control to M0DIAG, is set to zero.

c) MCPASS—the decrement of MCPASS contains the orbital pass number. The decrement of MCPASS is set to 1.

d) Location 00004—contains instruction TTR M0DCCC which establishes the control path for all DCC traps.

e) Location 00015—contains instruction TTR MTENLG which establishes the initial control path for Channel B traps (MTENLG is the trap processor which receives control after output transmission on the log tape).

f) The end-of-file and redundancy tape check indicators for both Channels A and B are turned off.

g) The B indicator for MYINTL is turned on in the Monitor Priority Table.

7.5.3 Method

M0INTL receives control directly from MXLOAD after the Bermuda Program System has been read successfully into core storage. The (transfer) trapping mode indicator is turned off. M0INTL executes a read tape instruction on Channel A, and a write tape instruction on Channel B, both without associated load channel instructions, to clear any remembered traps that may have been waiting on Channels A and B.

All tapes used in the Mercury Program System are rewound to their load points. The end-of-file and redundancy tape check indicators on both Channels A and B are turned off.

The input buffer tables for the AN/FPS-16 and Verlort radar data are set to zero and the sign of each cell is set minus. The low core control locations for the DCC and Channel B traps are set to provide transfer to the appropriate trap processors.

The cell containing the count of entries to the Main Controller Diagnostic Program, M0DIAG, is set to zero. The orbital pass designation in the decrement of MCPASS is set to 1 (for the 1st pass).

M0INTL turns on indicator B for MYINTL in the Monitor Priority Table so that M0PRIO may give control immediately to MYINTL to complete the initialization of the Bermuda Program System.

7.5.4 Usage

M0INTL is entered from MXLOAD; M0INTL exits to M0PRIO.

a) Storage Required—33 locations

b) M0INTL uses:

1) Macro—TRNON

2) Parameters—B, MNINTL, MUTAPA, MULTP1, MUPRIM and MUAUXL

3) Communication cells—MCTHSM, MCDIAG and MCPASS

4) Tables—TMHSFP and TMHSVE

5) Absolute Locations—00004 and 00015

6) Constant—KD0001

c) Time Required—0.78 millisecond

7.6 MONITOR INITIALIZATION PROCESSOR (MYINTL)

The Monitor Initialization Processor continues to prepare the Mercury Program System for data processing and transmission by activating the one-minute WWV pulse subchannel, reading the Station Characteristic Tape into core storage, and activating all DCC input subchannels, except the telemetry subchannel.

The flow Chart for the MYINR processor is shown in Figure 7-7.

7.6.1 Input Requirements

Input to MYINTL consists of:

a) M—parameter defining the number of routines in the Bermuda Program System which receive control at a specified time interval (see MTHFSC, subsection 3.17).

b) TMSTCH—this cell is equal to zero until the Station Characteristics Tape has been read successfully into core storage.

c) CXPS1—mask, 010 000 000 000, used to activate subchannel 5 (providing the one-minute WWV pulse), de-activate all other subchannels of the DCC.

d) CXPS3—mask, 230 000 000 000, used to activate all input subchannels of the DCC.

e) Macros used—QENBA, QENBZ, QPSLF, TRNOF and TRNON.

7.6.2 Output Requirements

Output from MYINTL consists of:

a) The input subchannels of the DCC except the telemetry subchannel, are activated.

b) MSEXTM—the switch which controls the examination of the telemetry input buffer every half second is turned on (see MTHFSC, subsection 3.17).

c) MCHFS1—the parameter M is stored in MCHFS1.

d) Between the initial entry and the final exit, the standard GMT time base is entered into the computer and the Station Characteristic Tape is read into core storage (see Method).

7.6.3 Method

MYINTL receives control from M0PRIO immediately after the completion of M0INTL, since MYINTL, is the only routine with its indicator B on in TMPRIO

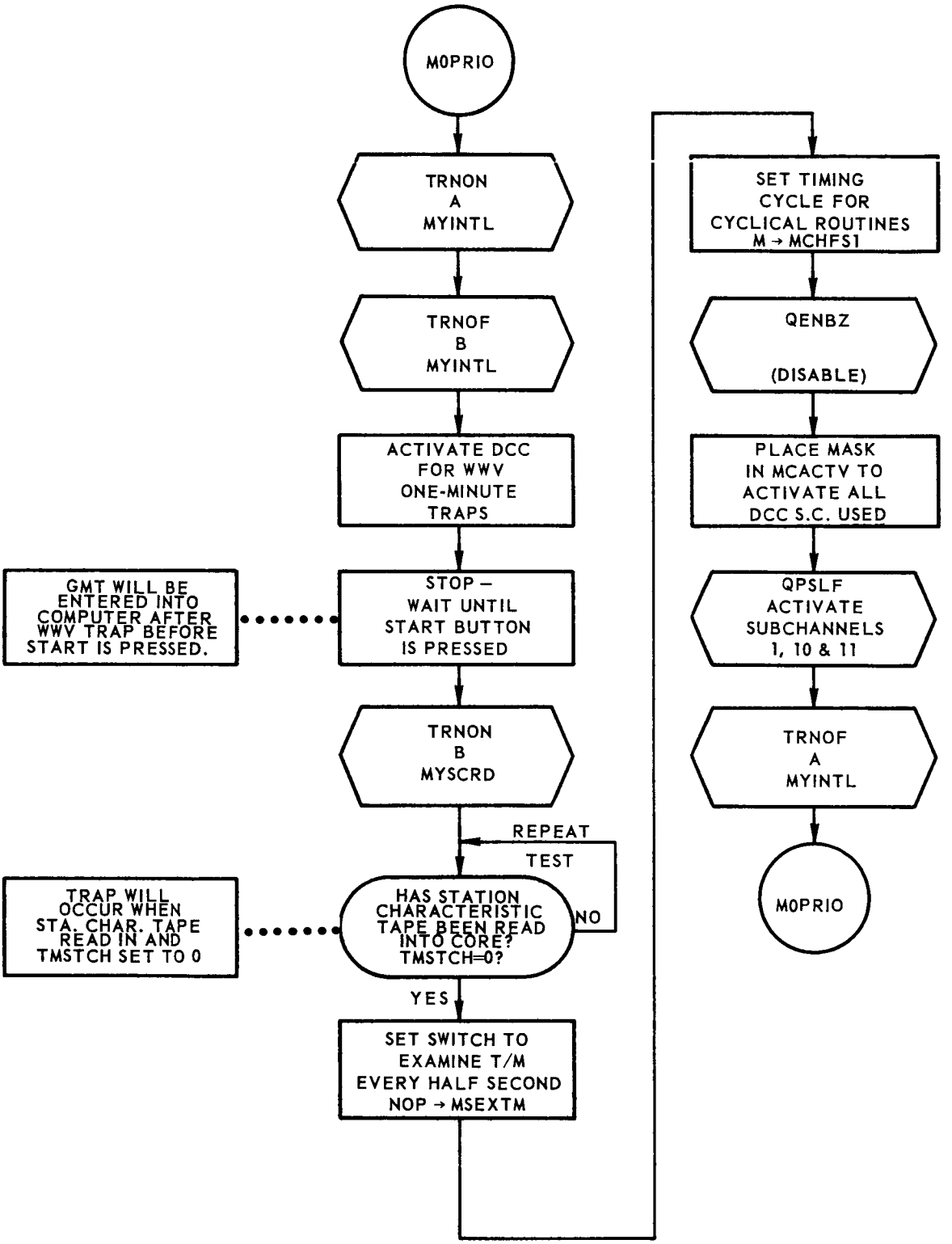


FIGURE 7-7. MYINTL PROGRAM FLOW CHART

and all DCC subchannels are deactivated. MYINTL turns on its A indicator, turns off its B indicator.

The WWV one minute pulse subchannel is activated and the program comes to a halt. MYINTL discontinues processing until the Start button is depressed. However, at the next exact minute the WWV (see writeup for MTIWWV) trap occurs and control transfers to MTIWWV. If the time of initialization is properly entered in the computer console keys, the GMT time base for the Bermuda Program System is entered into the computer. The Bermuda Program System will cycle between the MYINTL, MTIWWV, and the Main Monitor Controllers until the time base has been entered. MTIWWV provides two on-line messages to notify the computer operator that the WWV time has been entered. The operator now depresses the Start button and MYINTL continues from the point of the halt.

MYINTL turns on the B indicator for MYSCRD and loops on a non-zero test of the cell TMSTCH. Since MTIWWV has activated subchannel 4, the half-second trap subchannel, MYINTL will loop for a maximum of one half of a second. The half-second trap sends control to MTHFSC. At the completion of the half-second trap processor, control transfers to M0PRIO. M0PRIO examines the priority table and finds that indicator B for MYSCRD is turned on. Since MYSCRD has a higher priority than MYINTL, M0PRIO gives control to MYSCRD which reads the Station Characteristics Tape into core storage. When the tape is successfully read, TMSTCH is set non-zero.

MYINTL receives control, finds TMSTCH non-zero, and proceeds. The telemetry examination switch (MSEXTM) and the cyclical routine examination parameter (M) are set in the half-second trap processor.

MYINTL disables the computer, activates all DCC input subchannels, and enables the computer.

The A indicator for MYINYL is turned off—the Mercury Program System is now prepared for data processing.

Program control returns to M0PRIO.

7.6.4 Usage

MYINTL is entered from, and exits to, M0PRIO.

a) Storage Required—26 locations.

b) MYINTL uses:

1) Macros—QENBA, QENBZ, QPSLF, TRNON and TRNOF.

2) Parameters—A, B, M, MNSCRD and MNINTL.

3) Communication cell—MCHFS1.

4) Table—TMSTCH.

c) The timing of MYINTL is indeterminant since it depends upon the time initialization of the system by MTIWWV and the reading of the Station Characteristic Tape by MYSCRD and MTENST. However, since MYINTL is executed prior to data processing, the timing considerations should have no effect upon system operations.

Section 8.
MONITOR SYMBOL REFERENCE LISTS

The symbolic notations which identify Monitor programs, communication cells, tables, etc., follow the conventions outlined below. Each symbol consists of six characters: the first two conform to a standard code and specify a general function; the remaining four characters are generally mnemonic and indicate a specific function:

M0XXXX—Monitor controlling program. Any program name having M0 as the first two characters performs some basic controlling function.

MTXXXX—Monitor trap processor.

MYXXXX—Monitor processor.

MPXXXX—Monitor prefix to a processing program.

MFXXXX—Monitor suffix to a processing program.

MSXXXX—Monitor subroutine.

MNXXXX—the Monitor name for a Monitor processor or a non-Monitor processing program. This name is equivalent to a number which represents the location of the processor or processing program in the Monitor Reference table. The MNXXXX number is generally called the routine number of the program to which it refers.

MUXXXX—a tape unit addressed by a Monitor processor or trap processor.

MCXXXX—a communication cell, i.e., one word which contains some controlling information.

TMXXXX—a Monitor table.

MXXXXX—an external program, i.e., a support program which is not part of the operational program.

BXXXXX—a Bermuda processing program. There are eight processing programs in the Bermuda Program System: BARSCH, BEDITS, BIHSIN, BIMANI, BOCOMP, BODSPL, BOTOUT and BSMOOS.

BIMANI—Paper Tape Input Program; MNMANI. MPTIPT (3.7) and MFMANI and MFMAN1 (3.8).

BOCOMP—Output Computations; MNCOMP. MPCOMP (4.10) and MFCOMP (4.12).

BODSPL—Output to Displays Program; MNDSPL. MPDSPL (5.2) and MFDSPL (5.3).

BOTOUT—Output to Goddard Program; MNTOUT. MPTOUT (5.7) and MFTOUT (5.8).

BSMOOS—Smoothing Program; MNSMOO. MPSMOO (4.5) and MFSMOO (4.6).

8.1.4 Monitor Trap Processors

MTENPR—trap at completion of on-line printing (6.9).

MTENST—trap after reading record from Station Characteristics tape (6.3).

MTHFSC—half-second trap on DCC subchannel 4 (3.17).

MTHSBM—trap after transmitting to Bermuda displays on subchannel 2 (5.5).

MTHSIN—trap following radar input on subchannel 1 (3.2).

MTIWWV—Initial trap on WWV pulse on subchannel 5 (3.15).

MTMSCK—trap after reading message from tape (6.7).

MTSENS—trap following sense output on subchannel 12 (5.13).

MTTMIN—trap following telemetry input on subchannel 3 (3.10).

MTTTIN—trap following low-speed input on subchannels 10 or 11 (3.6).

MTTTOT—trap following transmission of teletype characters on subchannel 9 (5.10).

MTWWWV—normal WWV trap processor after MTIWWV enters GMT (3.16).

8.1.5 Monitor Subroutines

MSETPH—sets phase indicator cell and phase indicator light on Output Status Console. Used only by MPCOMP. (4.11).

MSEXTM—subroutine switch in MTHFSC set to examine or not examine new telemetry. (3.17).

MSLOGG—the logging subroutine (6.12).

MCEHST—time count for edit proceed; set and used by MFEBAD; used by MYEBAD.

MCERAD—radar type for editing; 0 for AN/FPS-16, 1 for Verlort; set and used by MPEDIT, MFEDIT and MFEBAD.

MCESPI—indicates GMT of capsule separation available; set by MYSECO, MYECOE; set and used by MYTMIN; used by MPSARC.

MCFLOC—last AN/FPS-16 point entered by BIHSIN; set by BIHSIN; set and used by MPEDIT.

MCFRVI—indicates orbit parameters have been prepared; set by BOCOMP; used by MFCOMP.

MCFSEC—GMT of capsule separation stored in TMFSEC; set by MPSARC; used by BARSCH.

MCGTLO—GMT of lift-off from paper tape input; set by BIMANI.

MCHFS1—cycled routines for MTHFSC to examine; set by MYINTL; used by MTHFSC.

MCHFSC—GMT in fixed-point half seconds; set by MTHFSC; used by any program that needs to know the current time.

MCKAPA—initial or restart edit entry for AN/FPS-16; set and used by MPEDIT.

MCLAMA—initial or restart edit entry for Verlort; set and used by MPEDIT.

MCLAOR—mission flight phase indicator: 0, launch; -0, abort; 1, orbit; -1, re-entry; set by MSETPH; used by BOCOMP.

MCLTP1—write select instruction addressing the current log tape; set by MSLOGG; used by MYSTLT.

MCLTP2—write select instruction addressing the reserve log tape: set by MSLOGG; used by MYSTLT.

MCMINS—GMT in fixed-point minutes; initially set by MTIWWV; set by MYMINS.

MCMSNO—message number and variable field data for input to MYMESS; set and used by MYMESS.

MCMTPR—contains TTR MTENPR instruction for channel A trap location; set by MYMESS and MYMSCK.

MCMUMU—editing to find new fit for AN/FPS-16; set by MFEBAD; used by MPEDIT and MFEDIT.

MCMVIP—temporary storage of MCMSNO while MYMESS is reading message from tape: used by MYMSCK; set by MYMESS.

MCNUNU—editing to find new fit for Verlort; set by MFEBAD; used by MPEDIT and MFEDIT.

MCPASS—orbit pass number; set by M0INTL.

MCPIPI—suppress smoothing for AN/FPS-16 until new fit found; set by MYEBAD; used by MPSMOO.

MCRADR—radar input to BIHSIN: 1 for Verlort, 2 for AN/FPS-16, 3 for both; set by MPHSIN; used by BIHSIN.

MCRECC—count of errors corrected by MSRECC; set by MSRECC.

MCRET1—retro 1 has fired; set by MPCOMP from telemetry input; used by BOCOMP.

MCRET2—retros 1 and 2 have fired; set by MPCOMP; used by BOCOMP.

MCRET3—retros 1, 2 and 3 have fired; set by MPCOMP; used by BOCOMP.

MCROAL—contents of MCLAOR last entry to BOCOMP; set by MPCOMP; used by MSETPH.

MCSAVE—save location for program trap; set by M0PRIO; used by M0SAVE.

MCSECO—valid indication of powered-flight termination obtained from telemetry; set by MYSECO or MYECOE; used by MPSARC.

MCSEN1—logging time tag for sense output; set by MYSENS; used by MTSENS.

MCSEN2—control mask for sense output subchannel; used by MYSENS and MTSENS.

MCSRAD—radar type for entry to BSMOOS: 0 for AN/FPS-16, 1 for Verlort; used and set by MPSMOO and MFSMOO.

MCTHSM—reset mask for input radar buffers; used by M0INTL and MPHSIN.

MCTHSN—radar quantity indicator; set by MTHSIN; used by MPHSIN.

MCTM01—indicates change in telemetry input for logging; set by MTTMIN; used by MYTMIN.

MCTM02—control mask for telemetry input subchannel; used by MYTMIN, MTTMIN and MTHFSC.

MCTM03—logging time tag for telemetry; set by MTTMIN; used by MYTMIN.

MCTM04—times telemetry examined this half second; set and used by MYTMIN.

MCTPOS—position of message tape; set by MYMSCK; used by MYMESS.

MCTRAP—trap while trying to disable; set by QENBA and QENBZ; used by M0SAVE and M0RTRN.

MCTT01—logging time tag for low-speed output; set by MYTTOT; used by MTTTOT.

MCTT02—word count for low-speed output; set by MYTTOT; used by MTTTOT.

MCTTOC—all teletype output has been transmitted; set by MPTOUT; used by MTTTOT.

MCTTOD—AN/FPS-16 teletype message in process; set by MFTOUT; set and used by MPTOUT.

MCTTOE—Verlort teletype message in process; set by MFTOUT; set and used by MPTOUT.

MCTTOF—BOTOUT working on uncompleted teletype message; set by MFTOUT; used by MPTOUT.

MCVLOC—last Verlort point entered by BIHSIN; set by BIHSIN; set and used by MPEDIT.

MCWWWV—GMT in 8-1/3 milliseconds; set by MTIWWV; set and used by MTWWWV.

MCX4RA—temporary storage of interrupt condition at trap; set by SAVE; used by M0SAVE.

MCZETA—a capsule separation time has been placed in TMFSEC; used and set by MPSARC.

8.3 TABLES

The Monitor controlled tables are listed below in alphabetical order.

TM8.3M—the 8-1/3 millisecond counter on subchannel 4.

TMALB1—first logging buffer filled by MSLOGG.

TMALB2—alternate logging buffer used when TMALB1 is full.

TMBFBK—temporary storage for Station Characteristics while error correcting codes are being removed.

TMCHAR—total length of all Station Characteristics blocks.

TMCMDA—table of I/O commands used by the on-line message programs.

TMCYCL—counter and transfer reference for cycled routines.

TMCYNO—reset cells for TMCYCL.

TMDCCC—reference table for control transfer on DCC traps.

TMECTR—elapsed capsule time as received in telemetry message.

TMFALI—tables of constants used for time conversion by MYMSCK.

TMFMSK—table of masks for TRNOF macro.

TMFRPR—basic Monitor table, contains programs in priority order referencing their routine numbers.

TMFSEC—best estimate of capsule separation time by MPSARC.

TMGMER—difference between GMT and elapsed capsule time.

TMGMTR—time telemetry input message received.

TMHSBM—output table for DCC subchannel 2.

TMHSF1—tags for initial or restart edit entry.

TMHSF2—first range used in BIHSIN output RAE table.

TMHSFP—input table for DCC subchannel 1, AN/FPS-16 data.

TMHSIF—temporary storage of AN/FPS-16 radar data for input to BIHSIN.

TMHSIV—temporary storage of both radar's input for logging, temporary storage of Verlort radar data for input to BIHSIN.

TMHSOT—packed message prepared by BODSPL for Bermuda displays.

TMHSVE—input table for DCC subchannel 1, Verlort data.

TMMES1—card image of error message used when wrong message read from tape.

TMMES2—card image of error message used when requested message number does not exist.

TMMES3—card image of message read from tape, if incorrect message read.

TMMES4—masks used in adding an integer to an on-line message.

TMMES5—factors for converting a time in seconds to hours, minutes and seconds.

TMMES6—masks used in adding a specified time to an on-line message.

TMMES7—masks used in adding the current time to an on-line message.

TMMES8—program switches used by MYMSCK.

TMMESS—location of message as read from tape.

TMNMSK—masks for TRNON macro.

TMPANL—blocks for saving condition of computer when trap occurs.

TMPRIO—basic Monitor table, contains priority indicators and entry locations to all Monitor processors and processing programs.

TMPRLG—input to MSLOGG for logging on-line messages.

TMQKEY—basic Monitor table, contains count of entries in queue and location of queue for all Monitor processors and processing programs which have input queue tables.

TMQKY2—basic Monitor table, contains number of maximum entries in queue tables

TMRDLY—table of corrections to apply to retrofire times reported by telemetry.

TMREFR—basic Monitor table, contains programs in routine number order referencing the locations in the priority table, TMPRIO.

TMRET1—GMT retro 1 fired.

TMRET2—GMT retros 1 and 2 fired.

TMRET3—GMT retros 1, 2 and 3 fired.

TMRETS—retrofire time setting of capsule clock.

TMSAVE—basic Monitor table, contains programs in routine number order referencing their save blocks in TMPANL.

TMSTCH—permanent location of Station Characteristics blocks.

TMTMEX—storage location for telemetry input message.

TMTMIN—input table for telemetry input message on DCC subchannel 3.

TMTABT—GMT of abort initiate reported by telemetry.

TMTIPT—input table for paper tape message on DCC subchannel 11.

TMTSEP—GMT of capsule separation reported by telemetry.

TMTTBF—temporary storage of packed teletype output message.

TMTTIN—teletype input reference tables.

TMTTOT—output table for teletype messages on DCC subchannel 9.