

GY27-7237-1

Systems

**IBM System/360
Operating System
Machine Check Handler for
The IBM System/370
Models 135 and 145**

Program Number 360S-DN-539

OS Release 21

IBM

PREFACE

This publication describes the design of the Machine-Check Handler (MCH) program and what it does to prevent or minimize down-time for System/370 Models 135 and 145.

ORGANIZATION OF THIS MANUAL

The "Introduction" summarizes the operation of MCH. This section contains definitions and descriptions needed to understand the second section "Method of Operation."

The "Method of Operation" describes the functions of the program and shows how the major data areas are used by MCH.

The "Program Organization" section describes the modules that constitute MCH and the operation of each of these modules. Flowcharts of each module are provided at the end of this section.

"MCH Data Areas" describes the fields of information used by MCH in its principal data area, the MCH Common Area.

The "Diagnostic Aids" section describes several techniques that can be used to determine the source of problems that arise in MCH.

The "MCH Module Directory" section is a guide to the named areas of code in the program listing.

The appendixes contain a table showing where MCH messages originate, a detailed description of the machine-check interruption code, and the MCH wait state codes.

PREREQUISITE PUBLICATIONS

To use this manual effectively, the reader should be familiar with the System/360 Operating System and have available the following publication:

IBM System/370 Principles of Operation,
GA22-7000.

Second Edition (March 1972)

This is a major revision of, and obsoletes, GY27-7237-0. This edition applies to Release 21 of the IBM System/360 Operating System and to all subsequent versions of the operating system unless otherwise indicated in new editions or Technical Newsletters. Changes are periodically made to the information contained here; any such changes will be reported in subsequent revisions or Technical Newsletters.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Programming Publications, Department 636, Neighborhood Road, Kingston, New York 12401.

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained from an IBM 1403 Printer using a special print chain.

CONTENTS

SECTION 1: INTRODUCTION 1
Recovery Design of the Models 135 and 145 1
Hardware Recovery Features of the Models 135 and 145 1
 Automatic Recovery Features 1
 CPU Retry 1
 ECC Validity Checking 3
 Fixed Storage Locations 3
 Fixed Logout Area 3
 Extended Logout Area 3
 Control Registers 3
 Modes of Recovery Operation 3
 Modes of Recovery Operation of the Model 135 4
 Modes of Recovery Operation of the Model 145 5
 Use of the Mode Commands 5
 Mode Command for the Model 135 5
 Mode Command for the Model 145 6
MCH Error Recovery 6
 System Recovery 6
 System-Supported Restart 6
 System Repair 7
Physical Characteristics 7
 Main Storage Requirements 7
Auxiliary Storage Requirements 7
Overlay Structure of MCH 8

SECTION 2: METHOD OF OPERATION 10
The Logic of MCH 10
 Communications 10
 Initialization 10
 Saving the Environment 14
 Module Loading 14
 Hardware Error Analysis 17
 Types of Hardware Malfunctions 17
 System Damage 17
 Instruction Processing Damage 17
 Program Damage Recovery 19
 Recording and Termination 19
 Error Recording 20
 Emergency Recording 20
 Interface with the Channel-Check Handler (CCH) 23

OPERATION DIAGRAMS 25

SECTION 3: PROGRAM ORGANIZATION 47
MCH Initialization 47
MCH Nucleus 48
MCH Module Loader 48
Soft Machine-Check Handler (Model 135 Only) 50
Soft Machine-Check Handler (Model 145 Only) 51
Preliminary Error Analysis 53
System Analysis 54
MVT System Analysis 1 (Model 145 Only) 54
MVT System Analysis 2 (Model 145 Only) 54
MVT System Analysis 3 (Model 145 Only) 55
MFT System Analysis 1 56
MFT System Analysis 2 56
MFT System Analysis 3 57
PDAR Terminator 57
TSO Subsystem Analysis (Model 145 only) 58
Error Recorder 58
Console Write Routine 59
Emergency Recorder 59

Machine Status Control (Model 145 only)	59
Machine Status Control (Model 135 Only)	59
FLOWCHARTS	61
SECTION 4: MCH DATA AREAS104
Model Dependent Common Area104
MCH Independent Common Area104
Record Buffer Build Area109
Fixed Logout112
Extended Logout112
Damage Assessment Field Buffer Area113
Subsystem Data Area (Model 145 Only)113
Machine Status Block113
Model 135 Machine Status Block113
Model 145 Machine Status Block114
SECTION 5: DIAGNOSTIC AIDS118
Register Conventions118
Common Interruption Code Settings118
Unexpected Errors118
MCH History Table118
SECTION 6: MCH MODULE DIRECTORY121
APPENDIX A: MCH MESSAGE TABLE AND WAIT STATE CODES123
APPENDIX B: MACHINE-CHECK INTERRUPTION CODE125
INDEX127

ILLUSTRATIONS

Figure 1.	Machine-check handler overview	2
Figure 2.	Modes of recovery operation	4
Figure 3.	MCH resident area	7
Figure 4.	Main storage and auxiliary storage relationships	8
Figure 5.	MCH overlay structure	9
Figure 6.	General processing of hard errors	11
Figure 7.	General processing of Model 145 soft errors	12
Figure 8.	General processing of Model 135 soft errors	13
Figure 9.	MCH responses to error-on-error conditions	14
Figure 10.	MCH and environment before initialization	15
Figure 11.	MCH and environment after initialization	16
Figure 12.	Finding and loading MCH modules (Model 145 modules illustrated)	18
Figure 13.	MCH error record	21
Figure 14.	CCH error record	24
Figure 15.	Main storage error, soft error, system damage, and CCH error	27
Figure 16.	Flow of control for SPF key error	28
Figure 17.	Flow of control for CPU error	29
Figure 18.	Initialization	30
Figure 19.	Hardware error analysis	32
Figure 20.	Program damage recovery	37
Figure 21.	Recording and termination	41
Figure 22.	Use of buffers and the lost-record counter in recording	52
Figure 23.	MCH independent common area	105
Figure 24.	Fields of MCHDEB	107
Figure 25.	Fields of MCHINTEL	107
Figure 26.	Fields of MCHIOB	108
Figure 27.	Fields of MCHLSUM	108
Figure 28.	PDAR control and action bytes	110
Figure 29.	Fields of ABREC	112
Figure 30.	Possible MCH error records	114
Figure 31.	Fields of the Fixed Logout	115
Figure 32.	Fields of the Extended Logout for the Model 145	116
Figure 33.	Fields of the Extended Logout for the Model 135	117
Figure 34.	Register conventions	118
Figure 35.	Sample machine-check interruption codes for the Model 145	119
Figure 36.	MCH history table	120
Figure 37.	MCH message table	123

IGFMCHF0.	MCH Initialization	62
IGFMCHE0.	MCH Nucleus	64
IGFMCH40.	Model 145 Soft Machine-Check Handler	73
IGFMCH50.	Model 135 Soft Machine-Check Handler	77
IGFMCH41.	Preliminary Error Analysis	81
IGFMVTF1.	MVT System Analysis 1	82
IGFMVTF2.	MVT System Analysis 2	84
IGFMVTF3.	MVT System Analysis 3	86
IGFMFTF1.	MFT System Analysis 1	87
IGFMFTF2.	MFT System Analysis 2	89
IGFMFTF3.	MFT System Analysis 3	90
IGFMCHF5.	PDAR Terminator	91
IGFMCHF6.	TSO Subsystem Analysis	93
IGFMCHE2.	Error Recorder	94
IGFMCHE1.	Console Write Routine	98
IGFMCHE3.	Emergency Recorder	99
IGF29701.	Model 145 Machine Status Control	102
IGF13501.	Model 135 Machine Status Control	103

This publication describes the operations of the Machine-Check Handler (MCH) program for the IBM System/370 Models 135 and 145. The Machine-Check Handler for the Model 135 is a standard component of the MFT version of the System/360 Operating System. The Machine-Check Handler for the Model 145 is a standard component of both the MFT and MVT versions of the System/360 Operating System. The purpose of the Machine-Check Handler is to minimize the effects of machine malfunctions on jobs in process. MCH does this, on the occurrence of a machine-check interruption, by attempting to correct the malfunction and by producing diagnostic records and messages to help system maintenance personnel find the cause of the problem. See Figure 1 for an overview of the Machine-Check Handler.

RECOVERY DESIGN OF THE MODELS 135 AND 145

Machine malfunctions originate in the CPU, main storage, and control storage. When one of these fails, hardware facilities attempt to correct the malfunction. CPU malfunctions are corrected through microprogram routines which retry the failing operation. This is called CPU retry. Malfunctions in main and control storage are corrected by Error Checking and Correction (ECC). These two recovery features are described in more detail later in this section.

CPU retry and ECC are not always successful in their attempts to correct a malfunction. For this reason there are two types of machine-check interruptions. A "soft" machine-check interruption (sometimes called a recovery report) is generated when:

- CPU retry has corrected the malfunction,
- ECC has corrected the malfunction (Model 145 only), or
- ECC has encountered a solid, single-bit error that has reached an Error Frequency Limit such that ECC is correcting the error 256 times within 416 micro-seconds (Model 135 only).

A "hard" machine-check interruption (sometimes called a damage report) is generated when the malfunction has not been corrected.

When a machine-check interruption occurs, MCH immediately gets control. If a soft machine-check interruption occurs, MCH records information about the malfunction. If a hard machine-check interruption occurs, in addition to recording information about the malfunction, MCH attempts to shield the operating system from the adverse effects of the malfunction.

The machine logs out information describing the cause of the malfunction and the status of the system at the time of the interruption. This information is used by the Machine-Check Handler to carry out its recovery and recording operations.

HARDWARE RECOVERY FEATURES OF THE MODELS 135 AND 145

The operation of the Machine-Check Handler depends on certain recovery actions taken by the hardware. It also depends on information given to it by the hardware. Some of the features of the hardware are described here.

AUTOMATIC RECOVERY FEATURES

The Models 135 and 145 have two "built-in" methods of recovering from machine malfunctions: CPU retry and ECC. Whenever circumstances permit, these two hardware features recover from machine malfunctions without assistance from the software.

CPU Retry

CPU errors are automatically retried by microprogram routines. These routines save source data before it is altered by the operation. When an error is detected, a microprogram routine returns the CPU to the beginning of the operation, or to a point where the operation was executing correctly, and the operation is repeated. After eight unsuccessful retries, the error is considered permanent.

The CPU retry feature allows the machine to recover from temporary CPU failures that would otherwise make it necessary to reload the operating system or terminate the executing program.

After each successful use of CPU retry, there is a soft machine-check interruption unless CPU retry is in quiet mode. After eight unsuccessful retries, there is a hard machine-check interruption.

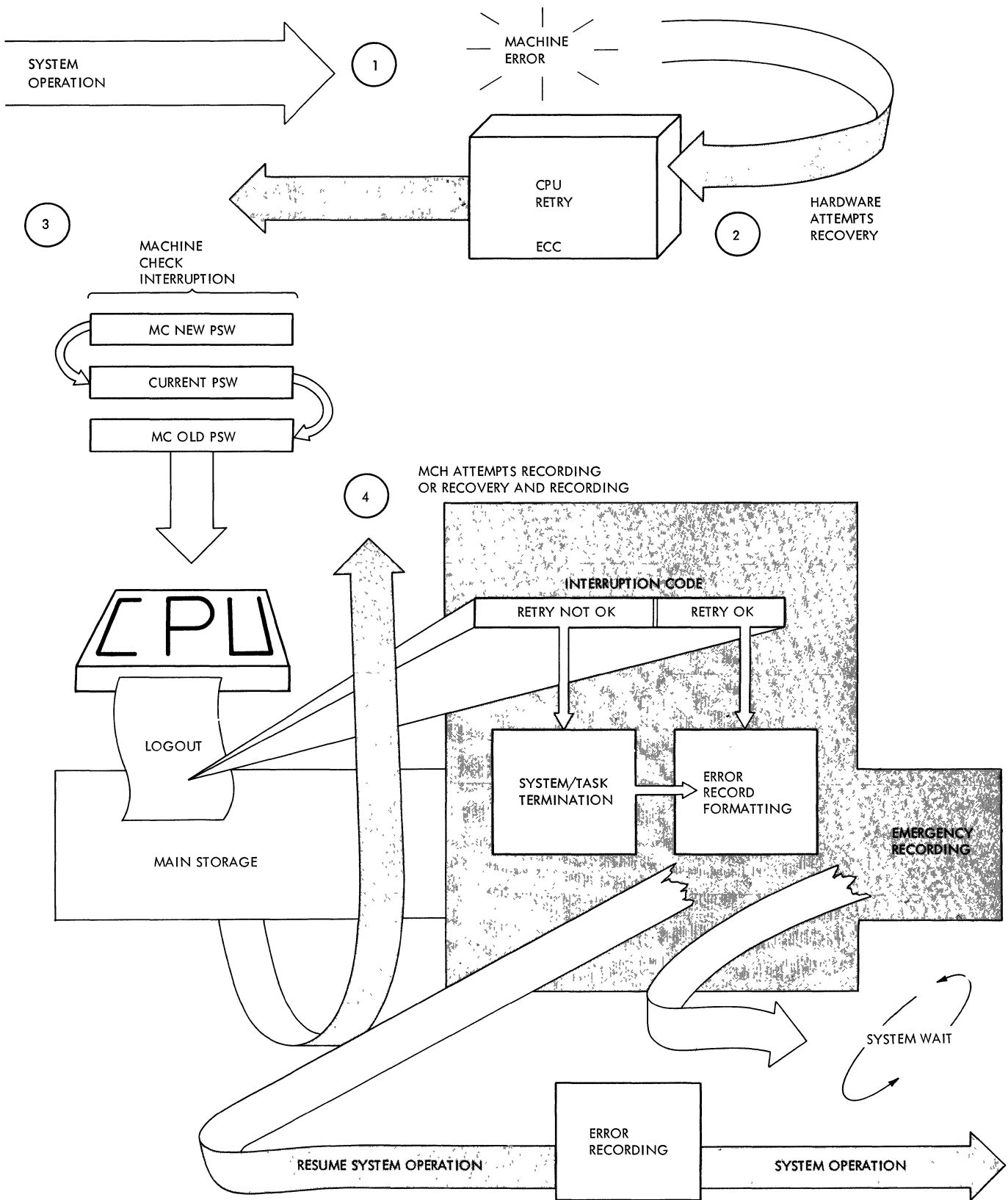


Figure 1. Machine-check handler overview

ECC Validity Checking

ECC checks the validity of data from main and control storage, automatically correcting single-bit errors. It also detects multiple-bit errors but does not correct them.

Data enters and leaves storage through a storage adapter unit. This unit makes the ECC validity check on each doubleword by insuring that the doubleword contains the appropriate parity bit for each byte. If a single-bit error is detected, the erroneous bit is corrected. The corrected doubleword is then sent back into main or control storage and on to the CPU. MCH is notified by a machine-check interruption and retrieves the failing storage address from the fixed logout. Note that with MCH for the Model 135, the threshold of such soft machine checks must be exceeded before a machine-check interruption occurs.

When a multiple-bit storage error is detected, a machine-check interruption is generated, and the error location is placed in the fixed logout. MCH gains control and attempts to recover from the error.

FIXED STORAGE LOCATIONS

There are four fixed storage locations in the Models 135 and 145: the fixed area in decimal locations 0-127, the I/O communications area in locations 160-191, the fixed logout area in locations 232-511, and the extended logout area. On the Model 135, the extended logout is a 14 byte field contained within the fixed logout area at location 256. On the Model 145, the extended logout is in locations 512-703, unless the pointer to the Model 145 logout area (control register 15) specifies otherwise.

Fixed Logout Area

Data is put into the fixed logout area (232-511) when any type of machine-check interruption occurs. The data stored is processed by the Machine-Check Handler. The layout of this area is model independent among the System/370 models; however, all models do not use every field in the fixed logout. The fixed logout area contains the machine-check interruption code which indicates the reason for the interruption. Other fields in the area preserve the status of the system at the time of the machine-check interruption and the contents of the general purpose, floating point, and control registers.

Extended Logout Area

The extended logout area contains data that is model-dependent. On the Model 145, the extended logout begins at the address specified in control register 15 and is a maximum length of 192 bytes long. Control register 15 is set to point to decimal location 512 by the hardware during IPL or system reset.

On the Model 135, the extended logout is contained in decimal locations 256 through 269 (an area within the fixed logout). If the extended logout mask bit in control register 14 is enabled for logouts, data is logged into the extended logout area for all types of machine-check interruptions. This data is recorded by MCH in the SYS1. LOGREC data set.

CONTROL REGISTERS

Two control registers are used by MCH for loading and storing control information.

Control register 14 contains mask bits which specify whether certain conditions can cause machine-check interruptions and mask bits which control conditions under which an extended logout may occur.

Control register 15, used only on the Model 145, contains the address of the extended logout area.

The control registers are referred to by MCH through the use of two privileged instructions: LOAD CONTROL and STORE CONTROL. LOAD CONTROL furnishes a means of loading control information from main storage to control registers; STORE CONTROL permits information to be transferred from control registers to main storage.

The publication IBM System/370 Principles of Operation, GA22-7000, contains a detailed description of the use of control registers.

MODES OF RECOVERY OPERATION

The type of recording done by MCH depends upon the current "mode" of the CPU, main storage, and control storage. There are three possible modes: quiet mode, recording mode, and threshold mode.

In quiet mode, machine checks corrected by CPU retry or ECC do not cause machine-check interruptions.

In recording mode, machine failures corrected by these features do cause interruptions for recording purposes.

In threshold mode, a preset frequency of such errors must occur before a soft machine-check interruption occurs. Note that hard (uncorrected) machine failures always result in a machine-check interruption regardless of mode.

There is a MODE command that can be used to vary the current mode (see "Use of the MODE Commands" in this section).

MODES OF RECOVERY OPERATION OF THE MODEL 135

The Model 135 can operate in either recording mode or quiet mode (see Figure 2).

CPU Malfunctions

When the CPU is in recording mode, a

soft machine-check interruption occurs each time a machine malfunction is repaired by CPU retry. When 20 such soft machine checks have occurred, the Soft Machine-Check Handler will automatically switch the CPU from recording mode to quiet mode.

Note: Main and control storage are switched automatically to quiet mode along with the CPU. See "Use of the Mode Commands" in this section.

When the CPU is in quiet mode, no machine-check interruption is issued for a soft error. Switching from quiet mode to recording mode can be accomplished by issuing the MODE command.

Mode	Error Location	Error Type	Solid/Intermittent	MCH Entered (Interruption)	
				135	145
Quiet	Main Storage	Multiple-bit (hard)	NA	Yes	Yes
		Single-bit (soft)	NA	No	No
	Control Storage	Multiple-bit	NA	Yes	Yes
		Single-bit	NA	No	No
	CPU	NA	NA	NA	NA
	Recording	Main Storage	Multiple-bit	NA	Yes
Single-bit			Solid	No	No
Single-bit			Intermittent	Yes	Yes
Control Storage		Multiple-bit	NA	Yes	Yes
		Single-bit	Solid	No*	No*
		Single-bit	Intermittent	Yes	Yes
CPU		Multiple-bit	NA	Yes	Yes
		Single-bit	NA	Yes	Yes
Threshold	Main Storage	NA	NA	NA	NA
	Control Storage	Multiple-bit	NA	NA	Yes
		Single-bit	NA	NA	No*
	CPU	NA	NA	NA	NA

*Single-bit errors in control storage will generate an interruption only if the hardware specified threshold is exceeded.

Figure 2. Modes of recovery operation

Main and Control Storage

In recording mode, a machine-check interruption occurs for each malfunction except solid, single-bit errors that occur below a certain rate. If the rate (or frequency) of single-bit errors becomes too high, a soft machine-check interruption occurs and main and control storage are automatically switched to quiet mode by the Soft Machine-Check Handler.

Note: The CPU is not automatically switched to quiet mode along with main and control storage. See "Use of the Mode Commands" in this section.

In quiet mode, no soft machine-check interruptions occur. A switch from quiet to recording mode can be made by issuing the MODE command.

MODES OF RECOVERY OPERATION OF THE MODEL 145

Three modes of operation for the Model 145 are used; recording mode, quiet mode, and threshold mode. Depending on the source of the malfunction, one, two, or all three modes may apply (see Figure 2):

CPU Malfunctions

Only the recording mode applies to CPU operations. In this mode, a machine-check interruption occurs for each malfunction.

Main Storage Malfunctions

In recording mode, a machine-check interruption occurs for each malfunction except solid, single bit errors. In quiet mode, only hard errors cause machine-check interruptions. Soft ECC errors do not cause interruptions when main storage is in quiet mode.

Control Storage Malfunctions

In recording mode, a machine-check interruption occurs for each malfunction. In quiet mode, only hard errors generate machine-check interruptions; soft errors do not cause interruptions. In threshold mode, no interruptions are generated for soft ECC errors unless a specified number of soft errors occur within a specified time. The frequency of errors that will be tolerated is preset by the hardware. When that frequency is exceeded, a machine-check interruption occurs, control storage is automatically switched into quiet mode and a message is sent to the operator.

USE OF THE MODE COMMANDS

The MODE command is an operator command used to switch between recording, quiet, and (145 only) threshold modes. The 135 MODE command can also be used to display the current mode status.

Mode Command for the Model 135

The format of the Model 135 MODE command is:

Operation	Operand
MODE	{ STATUS HIR RECORD ECC RECORD }

STATUS

causes the current status of both CPU retry (HIR) and ECC to be displayed in a message (IGF053I). The message also contains the CPU retry current error count and error count threshold for soft machine checks. The response to the command MODE STATUS is:

```
IGF053I MODE STATUS-ECC { QUIET }
                        { RECORD }
                        HIR { QUIET }
                        { RECORD }
count-nn THRESHOLD-nn
```

When the current error count equals the error count threshold (20), the Soft Machine-Check Handler switches both CPU retry and ECC to quiet mode. ECC is automatically switched to quiet mode along with CPU retry because the bit used to mask off CPU retry recording mode (bit 4 in control register 14) also masks off ECC recording mode.

When an Error Frequency Limit (EFL) of 256 single-bit error corrections within 416 micro-seconds has been reached, a soft ECC interruption occurs and the Soft Machine-Check Handler switches ECC (main and control storage) to quiet mode. CPU retry is not automatically switched to quiet mode along with ECC because ECC can be masked off independently with a DIAGNOSE instruction. Notice that solid, single-bit error corrections do not cause machine-check interruptions unless they occur with a greater frequency than the Error Frequency Limit. Also note that only Control Storage is referenced frequently enough for the EFL to be exceeded.

HIR RECORD

causes the CPU retry feature to enter recording mode. When the command to

enter recording mode is issued, the CPU current error count is reset to zero. If the CPU retry feature is already in recording mode when this form of the command is issued, the current error count is still reset to zero.

ECC RECORD

causes the ECC feature to enter recording mode. If this form of the MODE command is issued when CPU retry is in quiet mode, it is rejected as a command error. One bit in control register 14, which is used to mask off CPU retry recording mode, also masks off ECC recording mode. Therefore, CPU retry must be in recording mode before ECC can be switched to recording mode.

Mode Command for the Model 145

The format of the Model 145 MODE command is:

Operation	Operand
MODE	{ MAIN } { ,RECORD } { CNTR } { ,QUIET } { ,THRES }

MAIN

causes main storage to be placed in the specified mode.

CNTR

causes control storage to be placed in the specified mode. Note that control storage is physically identical to main storage and that both are contained within the same unit. Main storage contains problem programs and control (supervisor) programs. Control storage contains the basic instruction set.

RECORD

causes the specified storage to be set to recording mode. In recording mode, a machine-check interruption occurs for all machine errors (except solid, single-bit errors)¹ whether they have been corrected or not.

QUIET

causes the specified storage to be set to quiet mode. In quiet mode, machine

¹Solid, single-bit errors in Control Storage can sometimes reach such a frequency as to exceed a preset Error Frequency Limit. In such cases, a solid, single-bit error causes a machine-check interruption.

checks corrected by CPU Retry or ECC do not cause machine-check interruptions.

THRES

causes control storage to be set to threshold mode. This operand can only be used for control storage. When in threshold mode, a pre-specified number of soft errors must occur before a soft machine-check is issued and control storage is automatically switched to quiet mode. Notice that solid, single-bit error corrections do not cause soft machine-check interruptions unless it is a solid, single-bit error in control storage that exceeds the preset threshold.

WARNING: The Model 145 MODE command is intended for use only by IBM personnel. Issuing the RECORD form of this command at the wrong time can cause significant degradation in performance.

MCH ERROR RECOVERY

Recovery from a machine malfunction is handled by both hardware recovery facilities and the MCH program. MCH recovery can be classified into three categories: system recovery, system-supported restart, and system repair. These levels of error recovery are discussed in the order in which they are attempted.

SYSTEM RECOVERY

When the hardware cannot recover the system from the machine check, system recovery takes place. MCH attempts to keep the system working at the expense of the task in which the error appeared. The processing of the task containing the error is terminated either by normal methods of job termination (ABEND) or by marking the task nondispatchable. System recovery only takes place if the task in question is not critical to continued system operation. An error in a critical task would require a system-supported restart.

SYSTEM-SUPPORTED RESTART

System-supported restart (warm start) requires the operator to re-IPL the system. The operator is notified that a critical error has occurred and that system continuation is impossible. This type of recovery is used when system recovery has failed or has been judged impossible.

SYSTEM REPAIR

System repair takes place at the discretion of the operator. Usually, the operator will have tried to recover by system-supported restart one or more times with no success. An example of this type of error is when a hard error occurs so frequently that system-supported restart would not be successful. System repair always requires the services of maintenance personnel.

PHYSICAL CHARACTERISTICS

MAIN STORAGE REQUIREMENTS

The Machine-Check Handler operates within the MCH Resident Area. The MCH Resident Area, as shown in Figure 3, occupies 4.9K bytes in the fixed area of main storage. The MCH Resident Area is divided into three sections: the MCH Nucleus Area, the MCH Transient Area, and the MCH Common Area.

MCH Nucleus Area: The MCH Nucleus Area contains the control module of the program. It is 2.3K bytes long and its contents remain in storage unchanged.

MCH Transient Area: The MCH Transient Area occupies 1K bytes of main storage adjacent to the MCH Nucleus Area. It is used by the MCH transient (or overlay) routines, which reside on SYS1.SVCLIB.

MCH Common Area: The MCH Common Area is used for intermodule communication and construction of the MCH error record. A portion of the MCH Common Area, called the Subsystem Data Area, is used for communication between MCH and any subsystems which may be running under the operating system. The MCH Common Area is partitioned into several smaller data areas. The contents of these partitions are described in Section 4.

The fixed and extended logout areas are reserved to log the data about the machine malfunction. The fixed logout area primarily contains information that is model independent. It extends from location 176 to location 511 (decimal). MCH only uses that portion from 232 through 511. The extended logout area contains only model-dependent data and is 192 bytes long for the Model 145 and 14 bytes long for the Model 135.

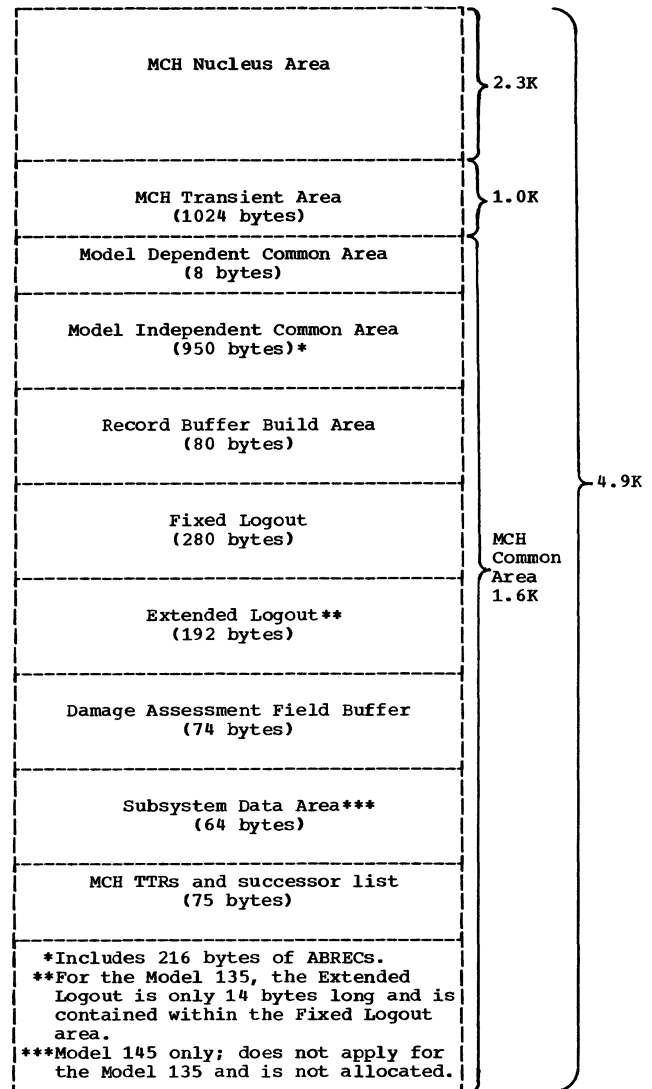


Figure 3. MCH resident area

AUXILIARY STORAGE REQUIREMENTS

The MCH transient modules occupy 13K bytes (Model 145) or 10K bytes (Model 135) in SYS1.SVCLIB on the primary SYSRES device. The MCH Nucleus and the MCH Initialization module (used to initialize MCH during NIP operations) are allocated 8.7K bytes on SYS1.LINKLIB.

The configuration of the system in use determines the amount of space required in SYS1.LOGREC for MCH to write its error records. (See IBM System/360 Operating System Storage Estimates, GC28-6551, for details.) Figure 4 shows the Machine-Check Handler in main storage and auxiliary storage.

OVERLAY STRUCTURE OF MCH

While the MCH Nucleus remains in main storage at all times, most other MCH modules are in main storage only when they are being used and are called transient modules. These nonresident modules are stored in the SYS1.SVCLIB data set. Figure 5 illustrates the overlay structure of MCH.

When the Machine-Check Handler is not being used, the Soft Machine-Check Handler occupies the transient area. The Soft Machine-Check Handler is an MCH module that prepares the recovery report for soft machine-check interruptions. Having the Soft Machine-Check Handler reside in the transient area eliminates the need to bring in modules from auxiliary storage when a soft machine-check interruption occurs.

The MCH Nucleus, which can be thought of as the control program for the Machine-Check Handler, resides permanently in the MCH Nucleus Area. The Module Loader is included in the MCH Nucleus. When a

machine-check interruption occurs, and the Nucleus determines that transient modules are needed to continue processing the machine-check interruption, control is given to the Module Loader to load a transient module from SYS1.SVCLIB. The first module brought into the transient area then overlays the Soft Machine-Check Handler. Each transient module can determine which transient module will succeed it. When the current transient module finishes its processing, it specifies the logical path number of the successor module to the Module Loader. The Module Loader then transfers control to the I/O Supervisor, which reads the next module into the transient area. After all processing has been completed, the Soft Machine-Check Handler is read back into the transient area. Except for system termination, the Soft Machine-Check Handler is always the final successor module, since it must be resident when MCH is again given control. When the system must be terminated, the Emergency Recorder is the last module in the Transient Area.

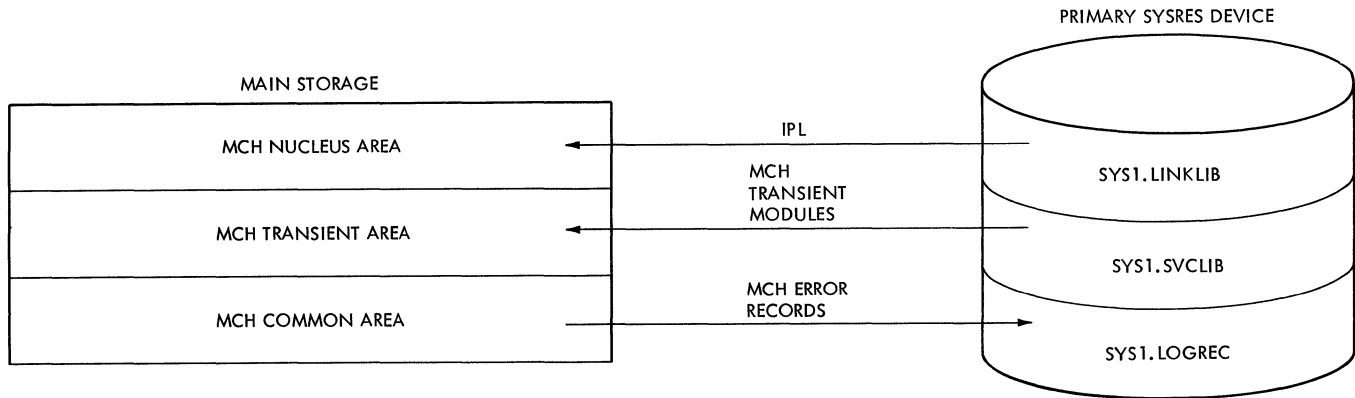


Figure 4. Main storage and auxiliary storage relationships

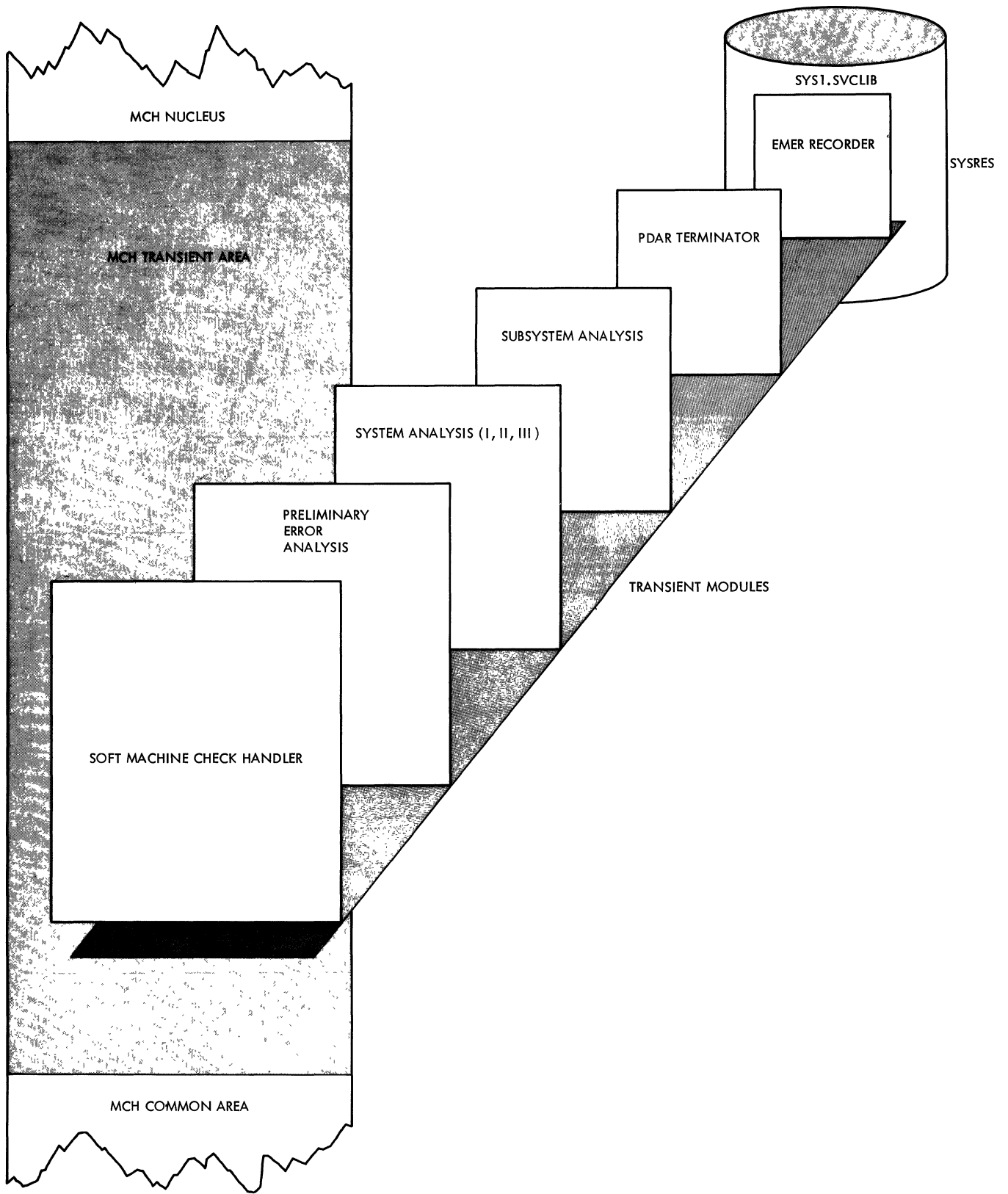


Figure 5. MCH overlay structure

SECTION 2: METHOD OF OPERATION

This section describes the functions of the Machine-Check Handler. For the reader who is unfamiliar with the operation of the program, this section will serve as an introduction to the logic described in the "Program Organization" section of the manual. For the reader familiar with MCH, this section, especially the illustrations, can be used for review.

This section is divided into two parts: the first tells why MCH operates as it does, and the second shows the operations that take place.

THE LOGIC OF MCH

MCH has two basic methods of operation: one for hard machine-check interruptions and one for soft (see Figures 6, 7, and 8). In processing a hard machine-check interruption, the Machine-Check Handler goes through four stages of operation:

1. Initialization
2. Hardware error analysis
3. Program damage recovery
4. Recording and termination

For a soft machine-check interruption, step 3, program damage recovery, is omitted. Since, by definition, a soft machine-check interruption signifies that the error has already been corrected by the circuitry (CPU retry and ECC), program damage recovery is not necessary. Figures 6, 7, and 8 illustrate the general processing involved in handling each type of machine-check interruption.

In addition to the four steps mentioned above, the Machine-Check Handler controls whether the machine will operate in recording mode or quiet mode. This function is logically independent from normal MCH processing.

The machine must communicate with the Machine-Check Handler and the components of the Machine-Check Handler must communicate with each other. In addition, MCH externally communicates with the operator and system maintenance personnel.

COMMUNICATIONS

To deal effectively with each machine-check interruption, MCH must have certain data concerning the nature of the malfunction. The hardware produces a logout that gives the Machine-Check Handler the information it needs to properly analyze the error. The Machine-Check Handler moves this information into the MCH Common Area. The transient modules use the Common Area to communicate with each other. The MCH Nucleus also uses the Common Area; it stores and retrieves data about the attempted recovery. Section 4 of this manual describes the Common Area fully.

INITIALIZATION

MCH normally receives control through a machine-check interruption. Should another machine-check interruption occur while MCH has control, processing would stop and control would go back to the beginning of MCH. As a result, the first machine-check interruption would never be processed since the information about it in the logout area would be lost. To minimize this possibility, MCH receives control with the system disabled for further interruptions. Disabling, however, is only a temporary measure to give the MCH Nucleus time to make some emergency provisions. The following initializing steps are taken by the MCH Nucleus:

1. It disables soft machine-check interruptions. Since soft errors have already been corrected, priority to interrupt MCH processing is given to hard errors. If the error being handled is hard and MCH is attempting to recover, there is no need to interrupt processing to report a soft error. If the present error is soft, there is no reason for one soft error to have priority over another.
2. It saves the contents of the fixed logout area in the MCH Common Area. If a hard machine-check interruption occurs now, the original data will not be overlaid by data from the second error. Also, extended logouts are prevented via a mask setting in control register 14 so that the extended logout area will not be overlaid.

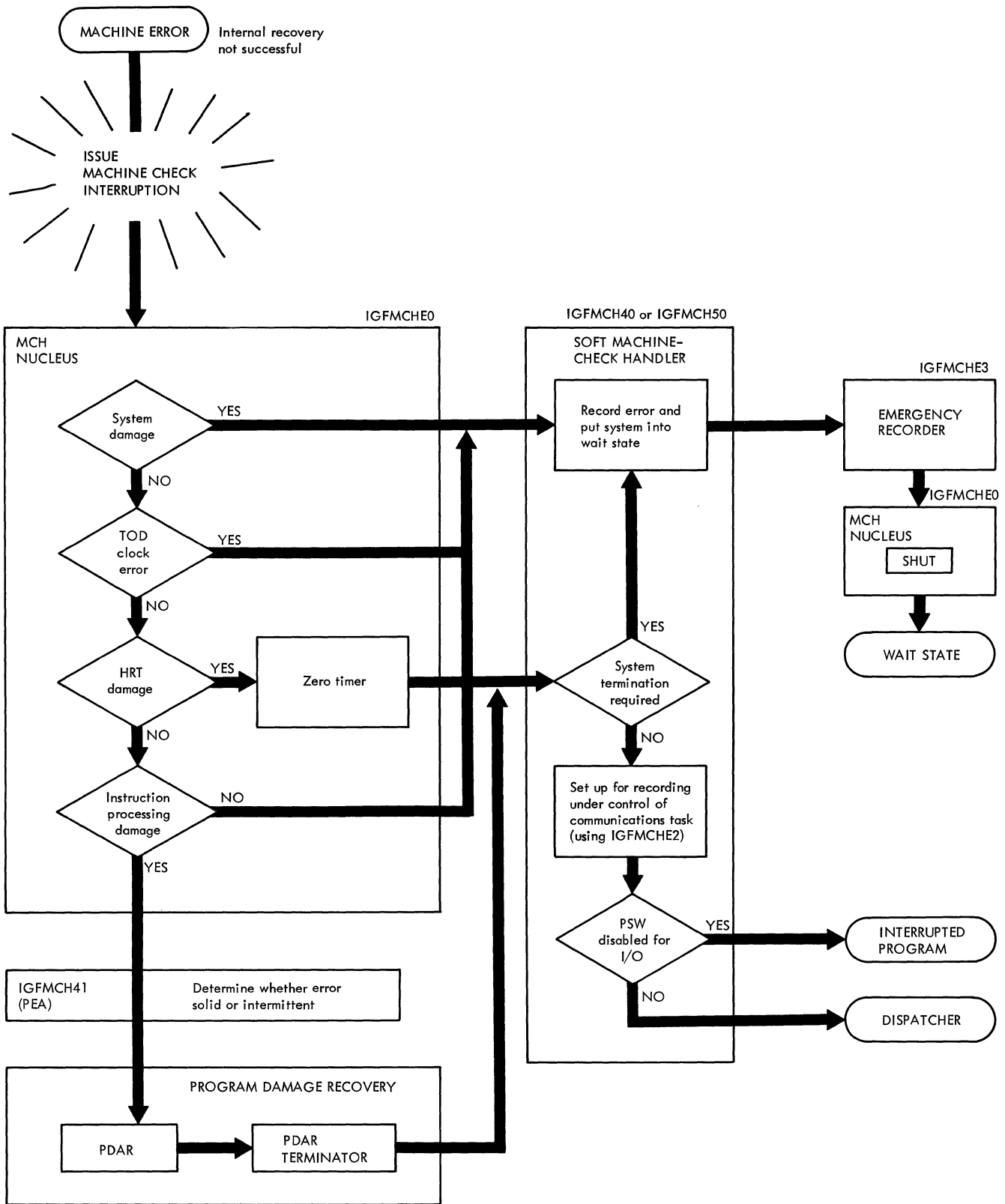


Figure 6. General processing of hard errors

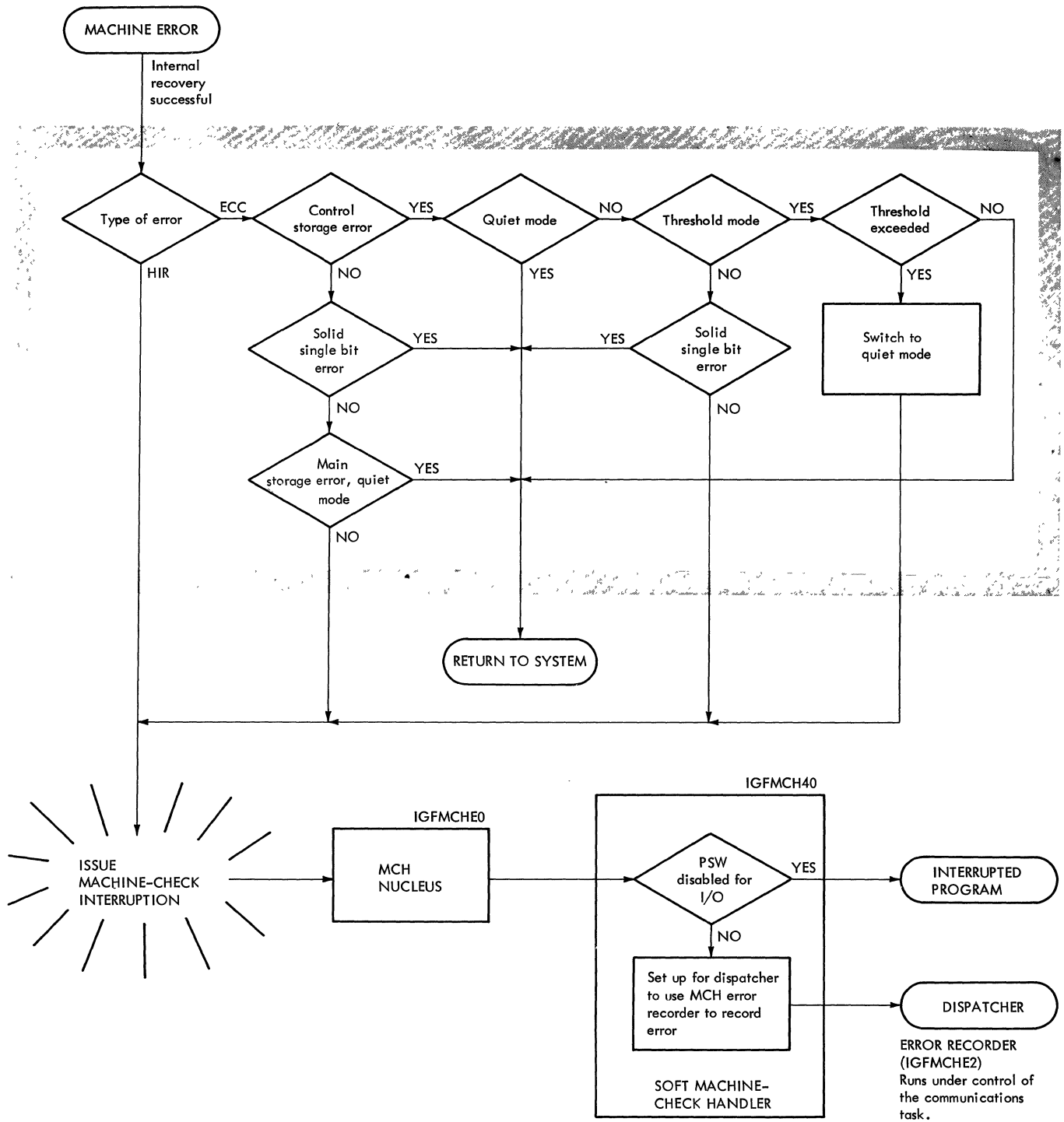


Figure 7. General processing of Model 145 soft errors

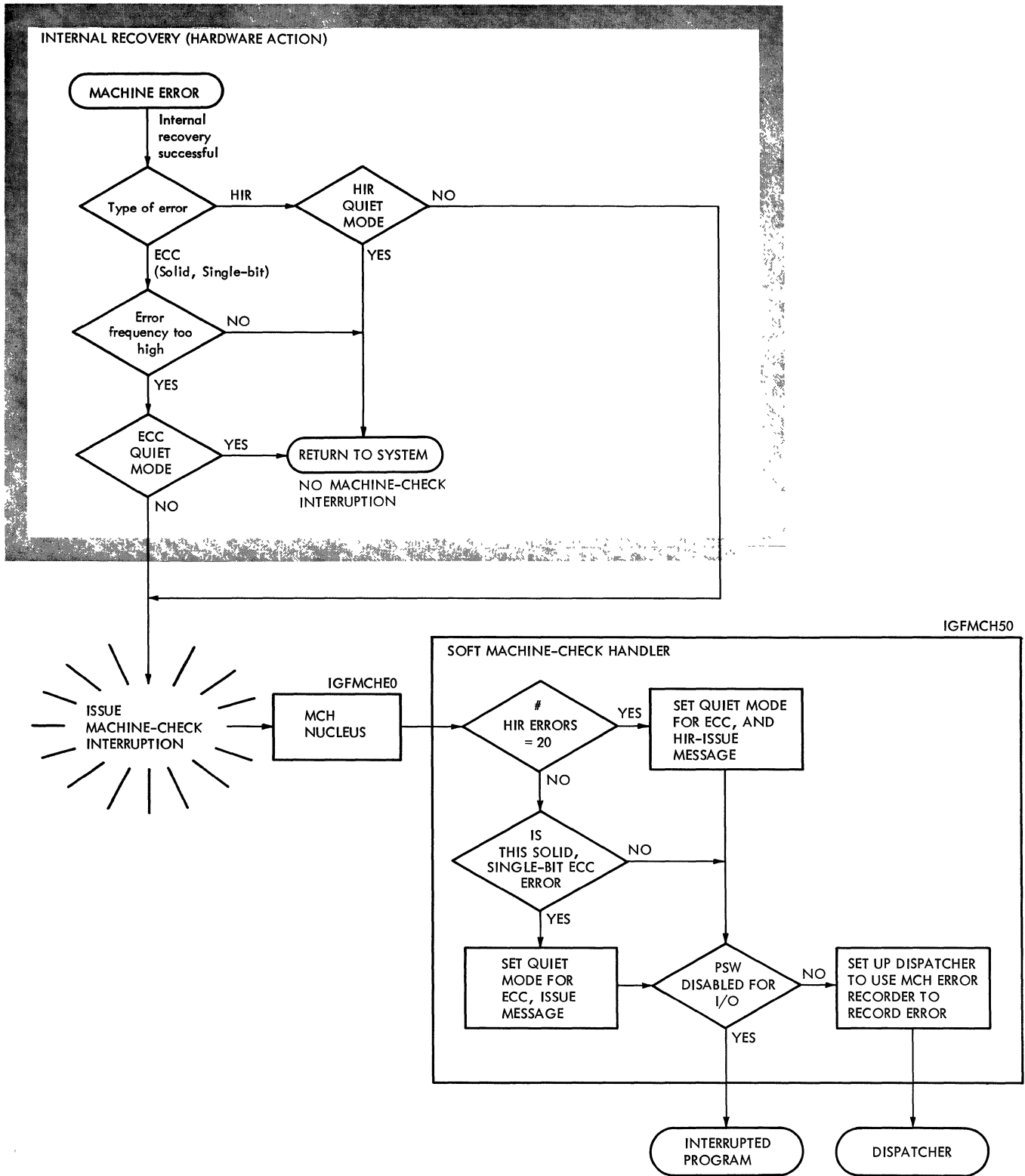


Figure 8. General processing of Model 135 soft errors

3. It saves the machine check old PSW. Then if a second error should occur, causing the current PSW to replace the old, control can be given back (through an LPSW instruction) to the program that was interrupted first (provided the error was corrected with the original system status intact).
4. It alters the address in the machine check new PSW to point to the SHUT (Special Handler for Unusual Termination) routine. A second machine-check interruption now sends control to the SHUT routine, rather than to the beginning of MCH. Note that a second machine-check interruption implies an error within MCH or IOS. If the machine-check new PSW were never altered and the error recurred, the Machine-Check Handler would go into a loop. Also, since the second error is within MCH, it is recognized that MCH is operating in a degraded state and might not be able to recover from the original error.
5. It alters the address in the program check new PSW to point to a special program check handler that intercepts and recognizes all program-check interruptions. If the interruption is caused by a Monitor Call, it is ignored. If the interruption is not caused by a Monitor Call, control is passed to the SHUT routine.
6. It enables hard machine-check interruptions. Soft machine-check interruptions remain disabled until error recording is completed.

There is always the danger that a machine malfunction may occur immediately after MCH is entered and the system is disabled for interruptions. If this happens,

the machine comes to a hard stop, no instructions are executed and no interruptions occur. The machine can only be removed from the hard stop by a system reset or IPL.

Figure 9 shows MCH responses to various error-on-error conditions.

Saving the Environment

The Machine-Check Handler saves the fixed logout and the machine check old PSW to protect them from a second machine-check interruption. The address where the fixed logout is saved is contained in MCHINLOG in the Common Area. Once the system has been reenabled for interruptions, MCH saves the permanent storage assignment (PSA). The PSA extends from location 0 through location 128 (decimal). The four data areas are saved in the following locations:

- Fixed Logout - address contained in MCHINLOG in the MCH Common Area
- Extended Logout of Model 135 - a fixed save area (decimal location 256) within the Fixed Logout
- Extended Logout of Model 145 - address contained in control register 15
- Machine-Check Old PSW - MCHPSW (REMCOPSW) in the Common Area
- Permanent Storage Assignment - MCHPSA in the MCH Common Area

Figures 10 and 11 illustrate the MCH environment before and after initialization.

Module Loading

The Machine-Check Handler uses the facilities of the I/O Supervisor to bring the

Error Condition	Special Circumstances	MCH Response
Hard on Soft	Error within MCH	An analysis is made to determine the severity of the error. For any error other than system damage, an attempt is made to record the original error and return control to the point of interruption. The occurrence of another hard error during this attempt will result in a wait state, with a message if possible.
Hard on Hard	Recovery made from original error	An attempt is made to record the original error and control is returned to the system.
	Recovery not made from original error	The system is placed in a disabled wait state with a corresponding message written to the operator.

Figure 9. MCH responses to error-on-error conditions

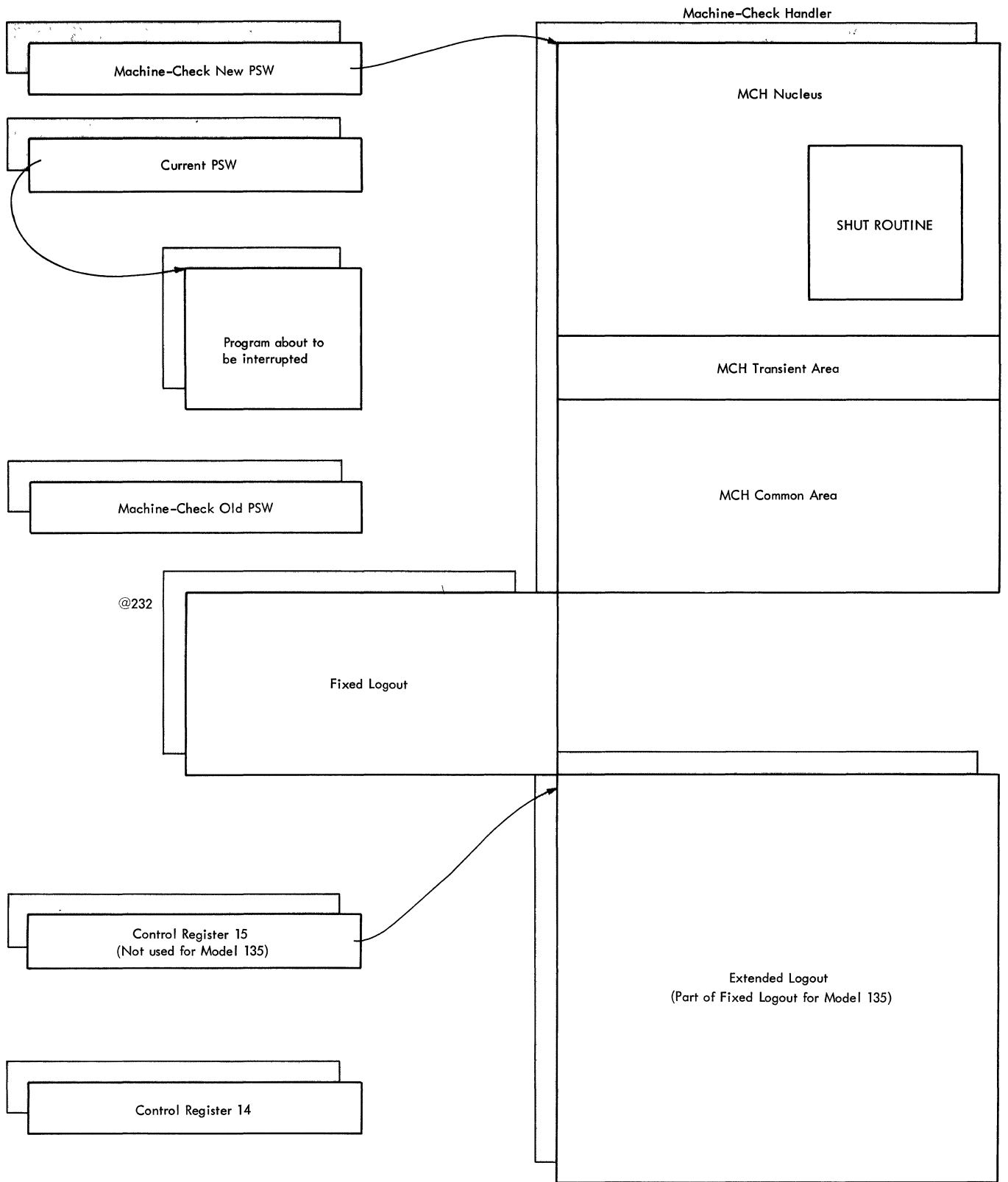


Figure 10. MCH and environment before initialization

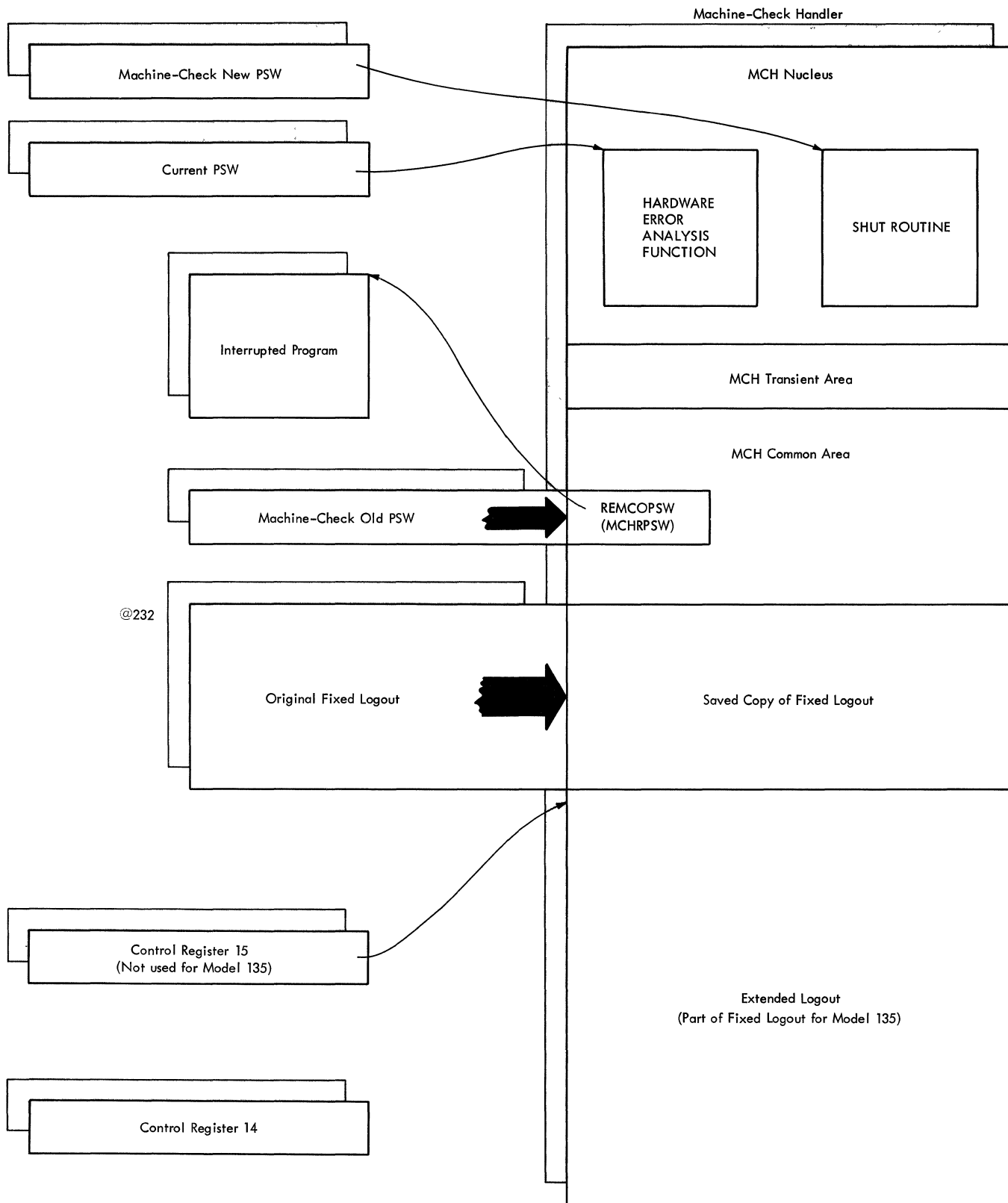


Figure 11. MCH and environment after initialization

MCH transient modules into main storage. Since an I/O interruption takes place after the new module has been read into the MCH transient area, MCH saves the address portion of the I/O new PSW and replaces it with the address of a section of its own code. This permits the Machine-Check Handler to service the I/O interruption. The original address contained in the I/O new PSW is replaced prior to returning to the system.

Figure 12 shows the module loading operation and explains the logic of scheduling a successor module.

HARDWARE ERROR ANALYSIS

To accurately assess the extent of the damage at the time of the machine-check interruption, the MCH Nucleus and the Preliminary Error Analysis (PEA) modules analyze the hardware error. MCH must identify the type of error, where it occurred, and under what special circumstances, if any.

To understand the hardware analysis function, some of the major fields of the machine-check interruption code are discussed first. Appendix B describes the interruption code fully.

MACHINE CHECK SUBCLASSES: Bits 0 through 4, the subclasses, indicate the machine check condition causing the interruption. On each interruption, at least one of these bits must be set. If multiple errors have occurred, several bits may be on.

TENSE: This field indicates the timeliness of the interruption status. For example, bit 14, when one, indicates that the instruction address in the machine check old PSW points to the instruction in which the error occurred. If the bit were set to zero, the instruction address would be pointing to an instruction beyond the point of error.

STORAGE ERRORS: This field informs MCH that the error was in main storage.

VALIDITY: The validity bits represent the various fields stored during the machine-check interruption. Any bit that is zero indicates that the associated data (general registers, condition code, etc.) has been affected by the error.

EXTENDED LOGOUT LENGTH: (Model 145 only) This field indicates the length in bytes of the extended logout area pointed to by control register 15.

Types of Hardware Malfunctions

The following types of hardware failures can be identified by the MCH Nucleus from the machine-check interruption code:

- **System Damage** - An error occurred that could not be attributed to the instruction referred to by the machine-check old PSW.
- **Instruction Processing Damage** - An error occurred during the processing of the instruction indicated by the machine-check old PSW. The instruction was either unretriable or unsuccessfully retried, or the damage resulted from a multiple bit failure in main storage or a Storage Protect Feature (SPF) error.
- **CPU Retry Successful** - (Soft error) The CPU instruction was successfully retried.
- **ECC Successful** - (Soft error) A single bit storage error was corrected by the ECC facility.
- **Time of Day Clock Damage** - An error occurred in the time of day clock making it invalid for time stamping.
- **Timer Damage** - The high-resolution timer at location 80 contains a parity error.

System Damage

System damage occurs when the machine circuitry or the microcode in the CPU has failed. Multiple-bit errors in control storage are included in this category. By presenting the malfunction as system damage in the machine-check interruption code, the machine informs MCH that system operation must stop. In this case the MCH Nucleus places the system in the wait state.

Instruction Processing Damage

Any type of instruction processing damage has some program damage or potential program damage associated with it. MCH must therefore ultimately associate the error with a system or user task and then take whatever action is necessary to keep the system operating. The first step is to determine from the bit settings in the interruption code the type of error that occurred. The common bit settings for various types of machine malfunctions are shown in Appendix B.

There are three types of malfunctions that are classified as instruction processing damage. Bit one, the instruction-processing-damage bit, is on (set to 1) in

FINDING SUCCESSOR MODULE

LOADING SUCCESSOR MODULE

Procedure

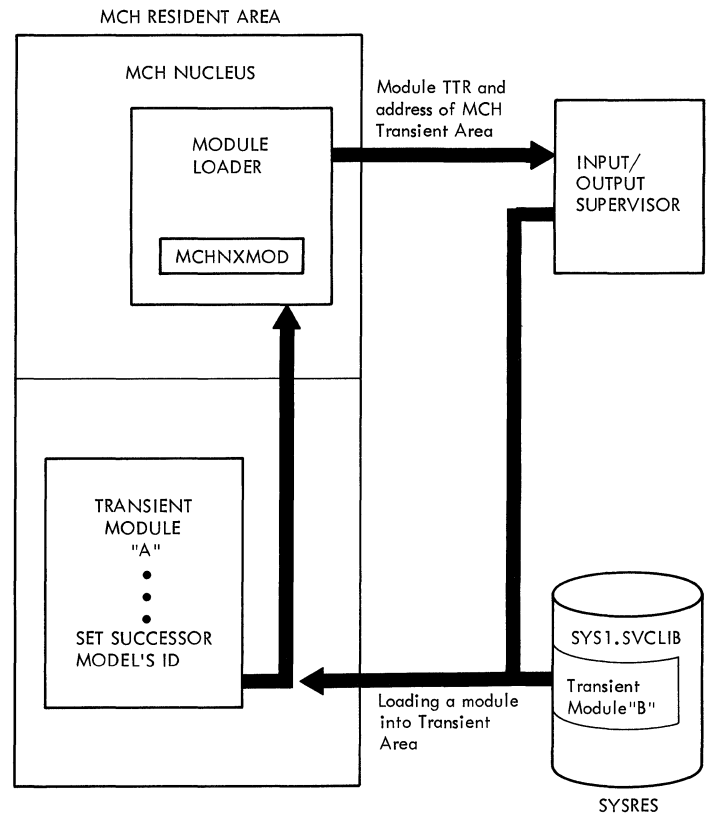
1. Module Loader saves displacement to successor list for module just loaded.
2. Transient module places a code into MCHNXMOD to designate its successor.
3. Module Loader adds saved displacement and subtracts 1 to determine successor.

Example

1. If PEA (IGFMCH41) is loaded, the Module Loader saves X'1B' from its Displacement Table.
2. If PEA wants control passed to the Soft Machine-Check Handler (IGFMCH40), it places a X'02' in MCHNXMOD.

3. Module Loader adds:
MCHNXMOD X'02'
Saved Displacement X'1B'
X'1D'
and subtracts 1
- X'01'
X'1C'

The result of which points to 40 (IGFMCH40) in the successor list.



Module Loader Successor List *			
Displacement	ID	Displacement	ID
00	00	0F	91
01	F5	10	00
02	F2	11	41
03	F3	12	00
04	00	13	00
05	00	14	00
06	F5	15	00
07	00	16	00
08	F3	17	00
09	F6	18	F1
0A	00	19	40
0B	F5	1A	E3
0C	F6	1B	F1
0D	40	1C	40
0E	F5	1D	F5

Module Loader Displacement Table		
Caller	Displacement	TTR
E3	00	005D02
F1	00	005D06
F2	05	005D0A
F3	0A	005D0E
F5	0D	005C04
F6	0E	005C08
40	10	005C10
41	1B	005C0C
91 **	1D	000000

* An ID of 00 indicates that a module has specified an invalid successor.

** IGFMC91 is the TSO Analysis module. Its original name is IKJEAM00 but it is linked into the SVCLIB at System Generation time as IGFMC91.

Figure 12. Finding and loading MCH modules (Model 145 modules illustrated)

all cases. The remaining related bit settings indicate the type of instruction processing damage that occurred. (Appendix B describes the use of each bit in the interruption code.)

1. **Retry failed** - This condition is indicated when the instruction processing damage bit is on and the error is neither a multiple-bit error nor an SPF key error. Since the PSW is pointing to the failing instruction and the instruction address is valid, MCH assumes that the CPU has retried the instruction but has not been successful.
2. **Multiple-bit error in main storage** - MCH, through the Preliminary Error Analysis routine, determines whether this type of error is solid or intermittent by finding the location of the error and doing a series of stores and fetches using that area. This is termed exercising a location.

If data changes during a store or fetch, or if another machine-check interruption occurs during the exercise, MCH labels the error solid. Otherwise, MCH labels the error intermittent.

Since a valid machine-check interruption must be anticipated each time data is fetched from or stored into the location, the address in the machine-check new PSW is altered to point to code that services the expected interruption. The result of this test is placed in the Common Area. MCH eventually uses this information to assess the damage to the task occupying that particular section of main storage.

3. **SPF key error** - The severity of an SPF key error is determined in a way similar to that used for the multiple-bit storage error. The machine-check new PSW is made to point to a section of code that will service the expected machine-check interruption. A succession of fetches using all 16 possible key patterns is made to determine whether the error is solid or intermittent. The result of this analysis is placed in the MCH Common Area.

PROGRAM DAMAGE RECOVERY

Having identified the hardware characteristics of the malfunction, MCH investigates the extent of the damage to the program executing at the time of the machine-check interruption. After assessing the damage to the program, MCH attempts to

recover the system by associating the damage with a particular task and terminating that task. This keeps the system in operation at the expense of only one job. If the supervisor is damaged, the system must be reloaded.

The modules responsible for system recovery are collectively known as the program damage assessment and repair modules or PDAR. To accurately assess the extent of the damage to the system, the PDAR modules use information placed in the MCH Common Area by the MCH Nucleus and Preliminary Error Analysis. In turn, each PDAR module uses the Common Area to convey the results of its operation to its successor.

In general, program damage recovery assists in the following:

- **Damage assessment** - associating what is known about the hardware characteristics of the failure with the task occupying the location that was affected.
- **Task Termination** - terminating any task that is in the problem program area.
- **System Termination** - putting the system in the Wait state when the error occurs in the supervisor or Link Pack Area, making further system operation impossible.

Program damage recovery procedures are necessary for:

- Intermittent or solid SPF key error
- Intermittent or solid main storage errors
- Retry failed error

To recover from an intermittent main storage or SPF key error in a problem program, the task is terminated by ABEND.

To recover from a solid main storage error or SPF key error in a problem program, the task is terminated by setting its TCB nondispatchable.

For an uncorrectable error caused by a failing instruction in a problem program, the task is terminated by ABEND.

For all errors (SPF key, main storage, or failing retry) occurring in the supervisor or Link Pack Area, MCH places the system in the Wait state.

RECORDING AND TERMINATION

The recording function of the Machine-Check Handler has two parts. The first is

the normal error recording procedure of formatting an error record and eventually writing it on the SYS1.LOGREC data set. The second is emergency recording; that is, the recording attempted when MCH has determined that system continuation is impossible.

Error Recording

The typical MCH error record is illustrated in Figure 13 . It consists of the MCH abbreviated record (ABREC), the fixed logout, the extended logout, and the damage-assessment field of the MCH Common Area. This record is produced for all machine-check interruptions.

Error recording involves formatting a record and writing it into the SYS1.LOGREC data set. Before MCH terminates its operation it formats the error MCH record. The actual writing of the record on the data set takes place after MCH terminates. MCH terminates before writing to decrease the chances of a second machine-check interruption's occurring while MCH is executing. In other words, if an interruption takes place during the I/O operation and MCH has not yet terminated, the interruption would have to be handled as an error-on-error condition. If MCH has terminated and a machine-check interruption occurs, MCH can handle the interruption in the normal manner.

MCH does the following to put an error record in SYS1.LOGREC:

1. It formats the complete error record.
2. It establishes the communications task (MCH Error Recorder) as an active task.
3. It terminates itself by giving control to the Dispatcher or to the interrupted program.

If the dispatcher gets control it will dispatch the next ready task. This should be the communications task. The MCH Error Recorder then writes the MCH records into the SYS1.LOGREC data set.

Should another machine-check interruption occur before the error record is written, the Error Recorder writes the short record of the first interruption and the complete error record of the second interruption. If a third interruption occurs after the record is formatted but before it is written, the short record of the first and second interruptions and the complete record of the third interruption are recorded.

The maximum number of error records that can be formatted for each recording operation is three. Therefore, if more than three interruptions occur before any recording is done, the record for the current interruption replaces the most recent soft error record. If no soft error records have been formatted, the most recent hard error record is replaced. Consequently, when the Error Recorder finally puts the error records into SYS1.LOGREC, they represent the three most recent machine checks with the hard machine checks taking priority. Also, the number of lost records and their characteristics are included in the error record.

The system remains in quiet mode until formatted error records have been written. Therefore, soft errors cannot cause formatted error records to be overlaid.

Note: The MCHDAMAG field of the MCH error record reflects the error analysis and action taken by MCH in its processing. The field is designed to be model independent in content. Thus, for a specific model all bits in the damage area or error type bytes of the field may not be implemented. Specifically, the buffer, control storage, extended main storage, address and mark bits are not set for all machines.

In addition, some RMS action data bits in this field are not set for all machines. The retry bit is set only if some type of software retry is attempted by MCH. Repair indicates that MCH has attempted to repair an SPF key failure. The Reconfigure bit indicates that MCH has performed some type of main storage reconfiguration. The refresh bit is set only if MCH has refreshed a portion of main storage. The setting of any of these bits indicates that MCH has performed the indicated action but does not imply that MCH was able to resume the task that was in control at the time of the error. For instance, task or system termination may be necessary if the retry was unsuccessful, a valid return point to the interrupted program is not available or an instruction is non-retryable.

Finally, for certain machine-check interruptions MCH makes an early determination that system termination is necessary and thus does not perform any further analysis as to type of error or area of damage. In these cases only the termination bit is set and possibly a system down code in the RMS action data area of the field.

Emergency Recording

Emergency recording is necessary when the system cannot continue to operate. Instead of giving control to the operating

Offset DECIMAL (HEX)				Field Name	Description
Model 135		Model 145			
				<u>Header</u>	24 bytes from MCHABREC in the MCH Common Area.
0	(0)	0	(0)	Byte 1	Record ID.
1	(1)	1	(1)	Byte 2	
				xx..	System ID. OS=00.
				..x.	Not used.
				...x xxxx	Release level.
2	(2)	2	(2)	Byte 3	
				1...	Operator action message.
				.1..	System/370 machine.
				..xx xxxx	Not used.
3	(3)	3	(3)	Byte 4	Record type information:
				1...	Short form of record.
				.1..	Record incomplete.
				..1.	MCH terminated the system.
				...x xxxx	Not used.
4	(4)	4	(4)	Bytes 5-8	Not used.
8	(8)	8	(8)	Bytes 9-16	Date and time.
16	(10)	16	(10)	Bytes 17-24	CPU Serial number.
				<u>Program ID</u>	8 bytes from MCHABREC in the MCH Common Area.
24	(18)	24	(18)		
				<u>Job ID</u>	8 bytes from MCHABREC in the MCH Common area.
32	(20)	32	(20)		
				<u>MC Old PSW</u>	8 bytes from MCHABREC in the MCH Common Area.
40	(28)	40	(28)		
				<u>MC Independent Logout</u>	280 bytes from the Fixed Logout Save Area.
48	(30)	48	(30)		
				<u>Extended Logout of Model 135</u>	14 bytes contained within a scratch area of the Fixed Logout.
256	(100)	*	*		
				<u>Extended Logout of Model 145</u>	192 bytes from the Extended Logout field (pointed to by control register 15).
*	*	328	(148)		
*Note: Because of the difference in Extended Logouts for the Models 135 and 145, MCH error record displacements differ for the two models from this point on.					
				<u>Damage Assessment Data</u>	74 bytes.
Bytes 1-6 from MCHPDAR in the MCH Common Area.					
328	(148)	520	(208)	Bytes 1-2	Length of this field.
330	(14A)	522	(20A)	Bytes 3-6	Address of the Machine Dependent Common Area.
Bytes 7-10 from MCHLOGIC in the MCH Common Area.					
334	(14E)	526	(20E)	Bytes 7-10	First level interrupt control field.
Bytes 11-18 from MCHDAMAG in the MCH Common Area.					

Figure 13. MCH error record (part 1 of 2)

Offset DECIMAL (HEX)		Field Name	Description
Model 135	Model 145		
<u>Damage Assessment Data (continued)</u>			
338 (152)	530 (212)	Byte 11	System status.
		1... ..	Hardware recovery.
		.1.. ..	Software recovery.
		..1.	Task aborted.
		...1	Task set nondispatchable.
	 1...	Termination.
	xxx	Not used.
339 (153)	531 (213)	Byte 12	Damage Area.
		1... ..	Main storage.
		.1.. ..	Buffer.
		..1.	Control storage.
		...1	Extended main storage.
	 1...	Processor.
	1..	Channel error.
	1.	Time-of-day clock error.
	x	Not used.
340 (154)	532 (214)	Byte 13	Error type.
		1... ..	Intermittent.
		.1.. ..	Solid.
		..1.	Data.
		...1	Address
	 1...	Mask.
	1..	Protect.
	xx	Not used.
341 (155)	533 (215)	Byte 14	RMS Action data.
		1... ..	Retry.
		.1.. ..	Repair.
		..1.	Reconfigure.
		...1	Refresh.
	 1...	Machine-check interruption in MCH.
	001	System damage.
	010	Machine-check interruption in CCH.
	011	I/O error.
	100	Invalid Logout.
342 (156)	534 (216)	Byte 15	Machine status data.
		1... ..	HIR in record mode.
		.1.. ..	ECC in record mode.
		..1.	Buffer disabled.
		...x xxxx	Not used.
343 (157)	535 (217)	Bytes 16-18	Not used.
		Bytes 19-26	from MCHLSUM in the MCH Common Area.
346 (15A)	538 (21A)	Bytes 19-24	Lost record summary. This field represents the number of lost records and their characteristics.
352 (160)	544 (220)	Bytes 25-26	Not used.
		Bytes 27-34	from MCHHISTY in the MCH Common Area.
354 (162)	546 (222)	Bytes 27-34	History of executed transient modules.
		Bytes 35-74	from MCHPDAR in the MCH Common Area.
362 (16A)	554 (22A)	Bytes 35-74	PDAR data.

Figure 13. MCH error record (part 2 of 2)

system to write the error record -- since the system is known to be unreliable -- MCH writes it. The Emergency Recorder, an MCH transient module, determines the number of records formatted in the buffer and whether there is room in SYS1.LOGREC to record them. The writing is done by the Module Loader in the MCH Nucleus. When the error records have been placed in SYS1.LOGREC, control is given to the SHUT routine to write a message to the operator informing him of the status of the error and to terminate the system.

Interface with the Channel-Check Handler (CCH)

The Channel-Check Handler (CCH) is a resident program which receives control from the I/O Supervisor after detection of a channel data check, channel control check, or interface control check.

The Machine-Check Handler may receive control during CCH operations because:

- A machine-check interruption occurs during CCH processing, or
- CCH determines that the operating system must be terminated.

When CCH is entered, it places a code X'01' in bits 24-31 of the machine-check

new PSW. This indicates to MCH, when a machine-check interruption occurs, that CCH is the affected program and that the system must be terminated. Only a machine-check error record is written in this case (see Figure 13).

When CCH determines that the system must be terminated because of a channel error, it:

1. Constructs a full channel-check record entry (see Figure 14).
2. Puts a code of X'0F' in bits 24-31 of the machine-check new PSW to indicate that CCH has created a record to be written and that the system must be terminated because of a channel error.
3. Loads the machine-check new PSW to pass control to MCH.

MCH uses error recording procedures to write the channel record on SYS1.LOGREC. The address of the channel record is in register 13 and its length is in the count field of the header (see Figure 14).

After MCH writes the record, it writes a console message that a channel error record has been recorded and it places the system in a Wait state with a code of A0A.

Dec	Hex		
0	0	Count Field 8	
8	8	Key 2	Reserved 4
		Switches 2	
16	10	Date 4	Time 4
24	18	CPU ID and Serial Number 8	
32	20	Job ID 8	
40	28	Active I/O Units 16	
56	38	Failing CCW 8	
64	40	CSW 8	
72	48	Extended Channel Status Word 4	Device Type 4
80	58	Channel ID	Multiprocessing Information 4
		Control Unit Address 3	
88	58	Channel Log (variable length)	

Figure 14. CCH error record

OPERATION DIAGRAMS

The first set of operation diagrams (Figures 15 through 17) traces the general path for processing each type of error. The major decisions along the way are related to the module that makes that decision.

The second set of operation diagrams (Figures 18 through 21) is more detailed. Each diagram is a series of frames that show the progress of an operation, such as initialization. Comments under each frame indicate what is being depicted.

The frames are numbered sequentially. When a decision results in a break from the normal sequence, an off-page connector composed of the figure number, figure part, and frame number directs the reader to the next frame. For example, a jump to Figure 20, part 3, frame 15 is shown as F20, 3-15. When a decision results in a break from the normal sequence to another frame on the same page, only the frame number is given inside the connector. For more detail, the flowcharts or the microfiche should be consulted.



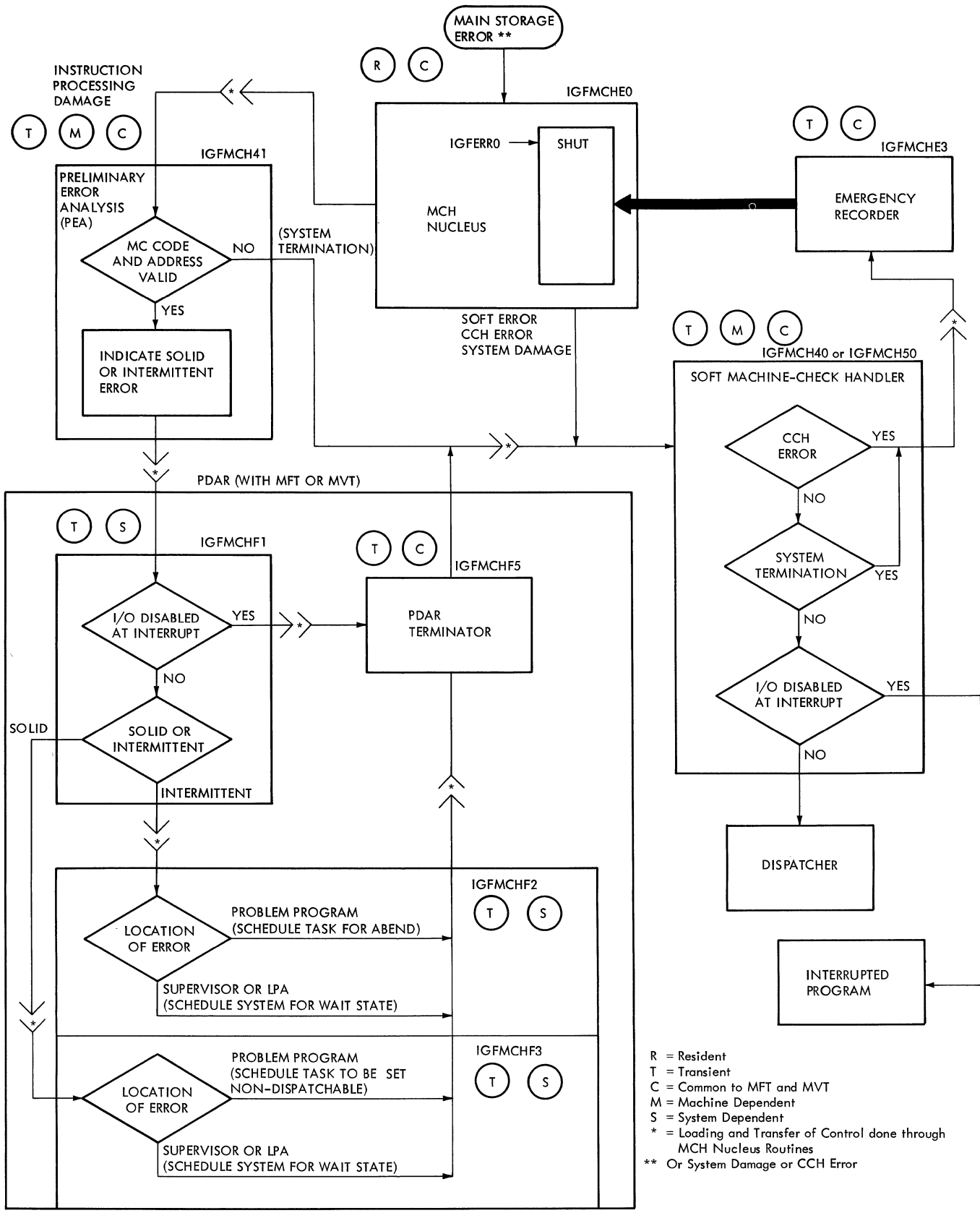


Figure 15. Main storage error, soft error, system damage, and CCH error

R = Resident
 T = Transient
 C = Common to MFT and MVT
 M = Machine Dependent
 S = System Dependent
 * = Loading and Transfer of Control
 done through MCH Nucleus
 Routines

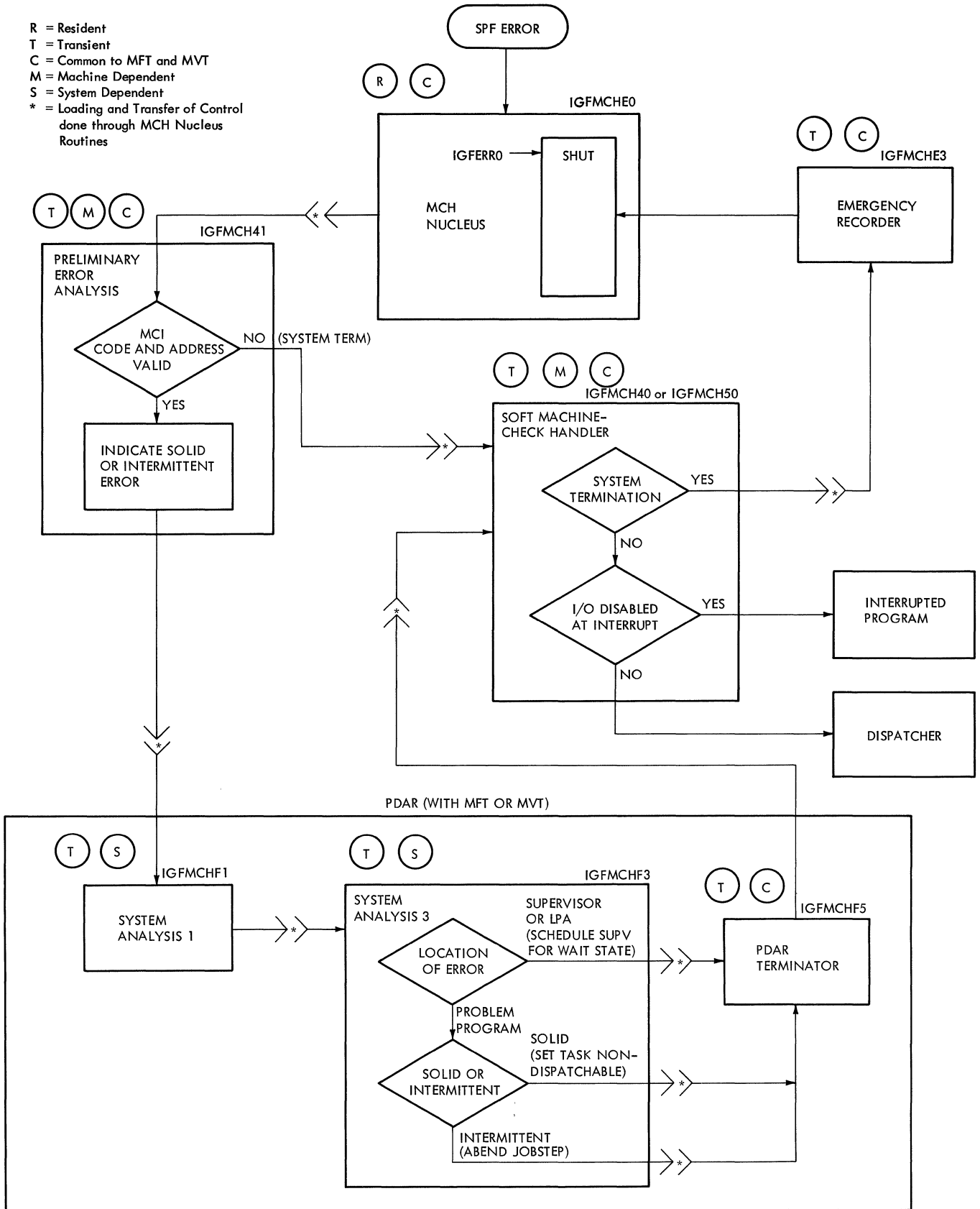


Figure 16. Flow of control for SPF key error

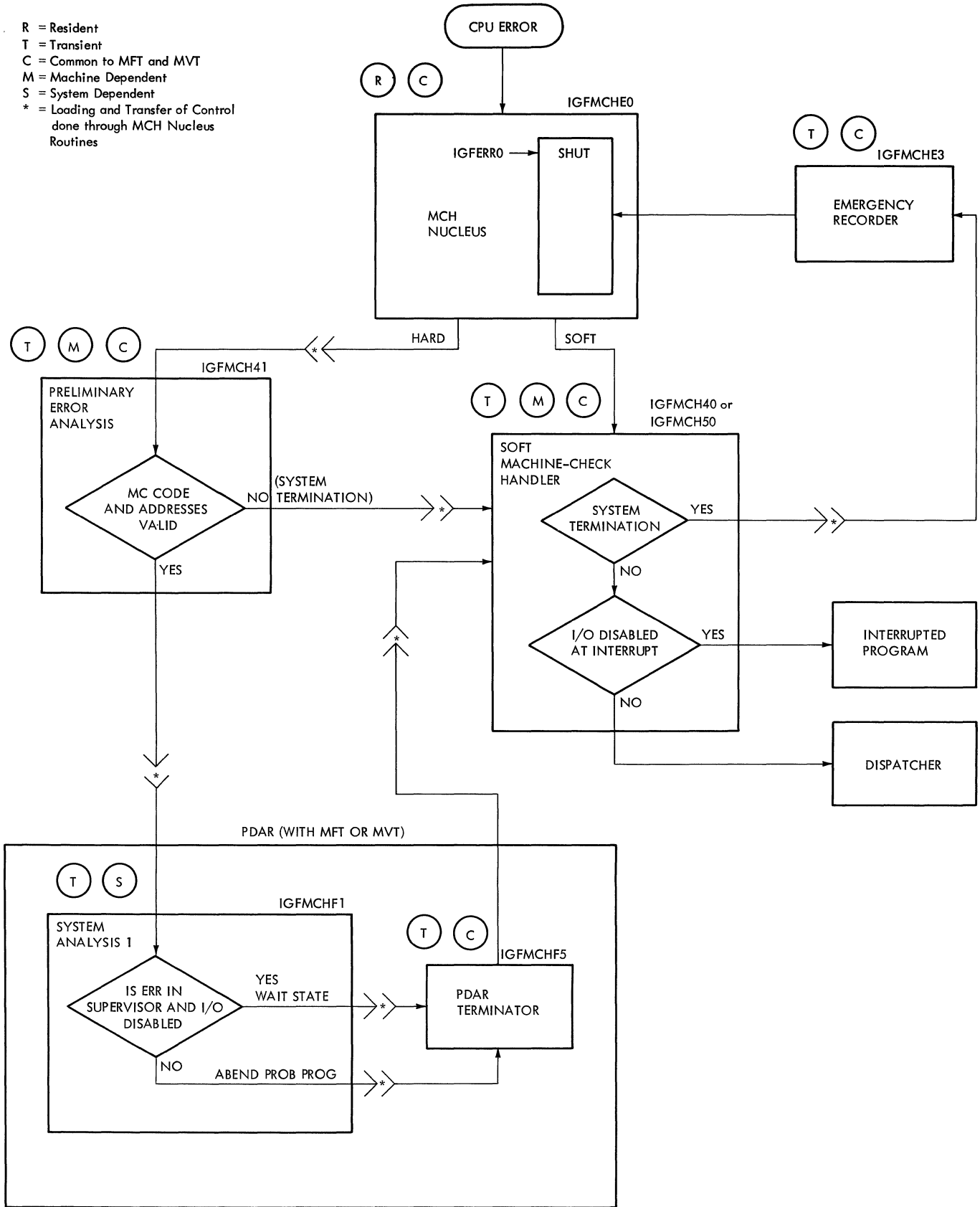


Figure 17. Flow of control for CPU error

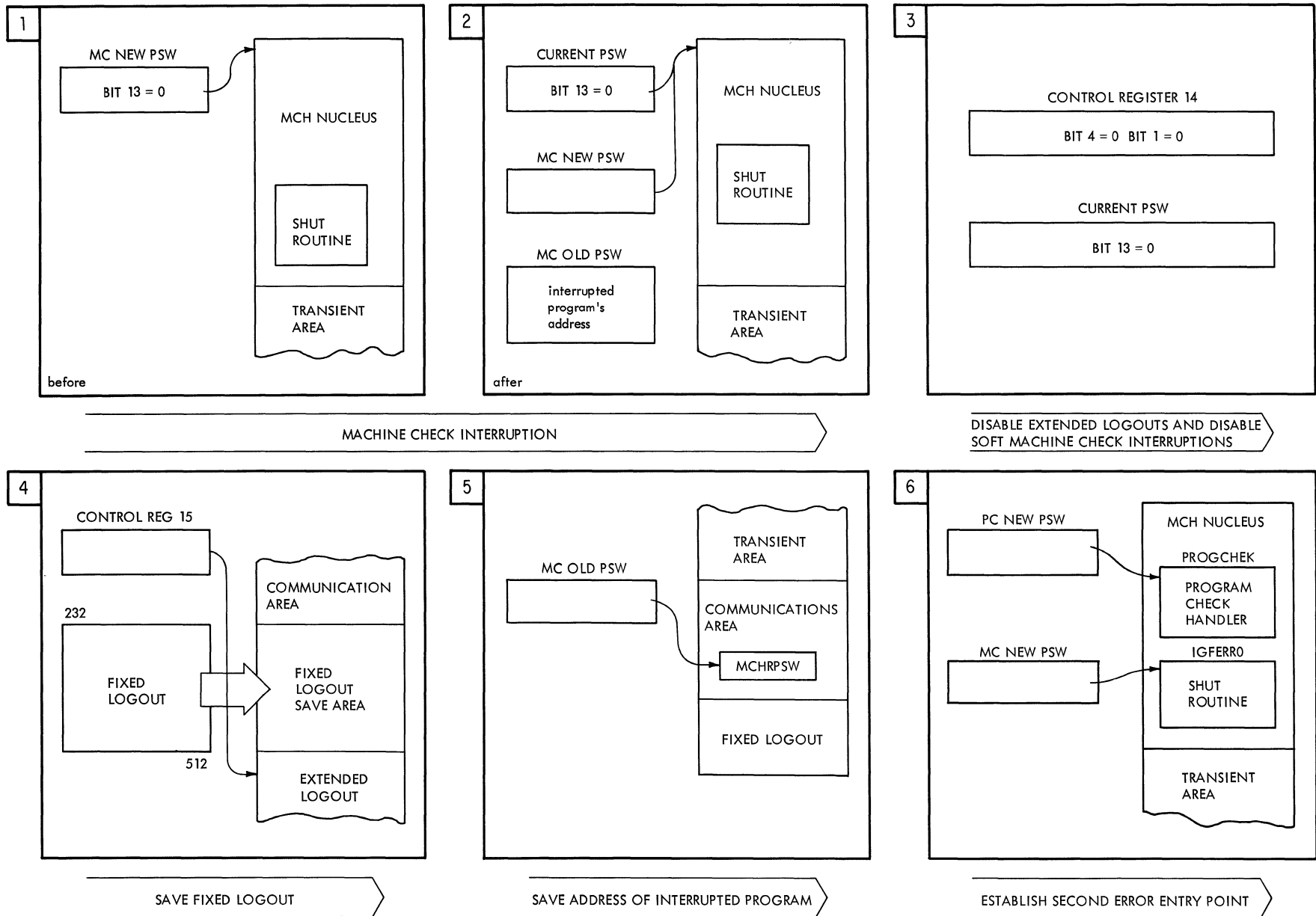


Figure 18. Initialization (Part 1 of 2)

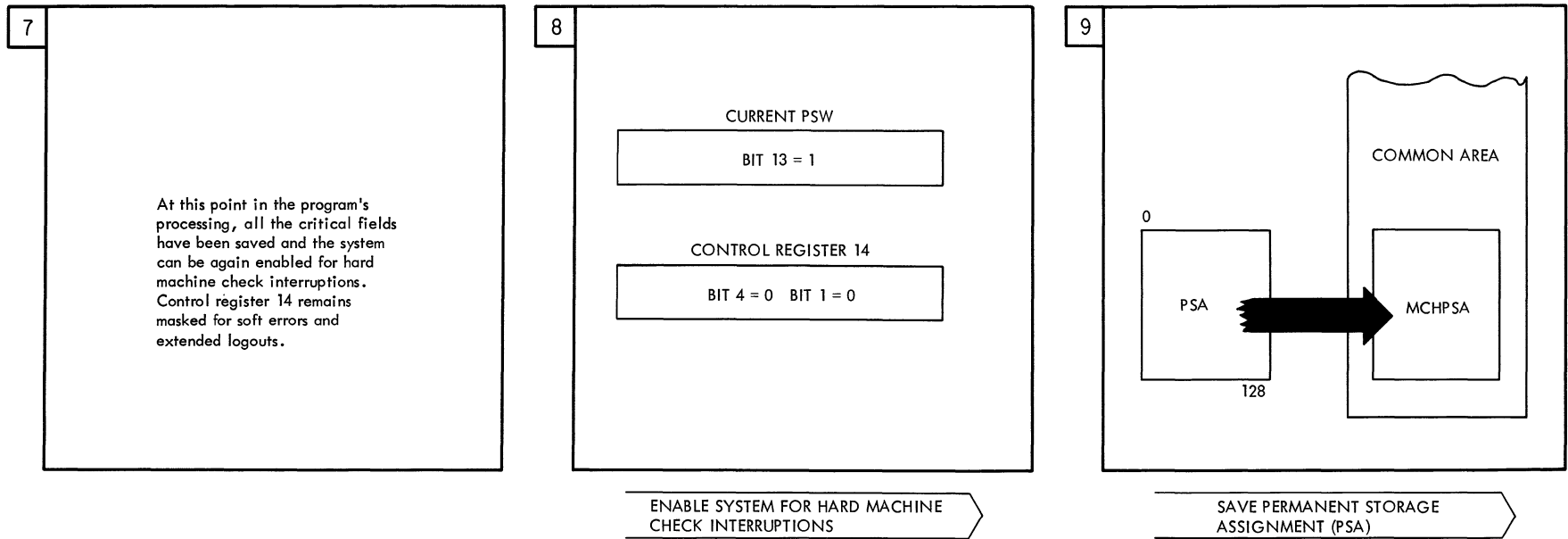


Figure 18. Initialization (Part 2 of 2)

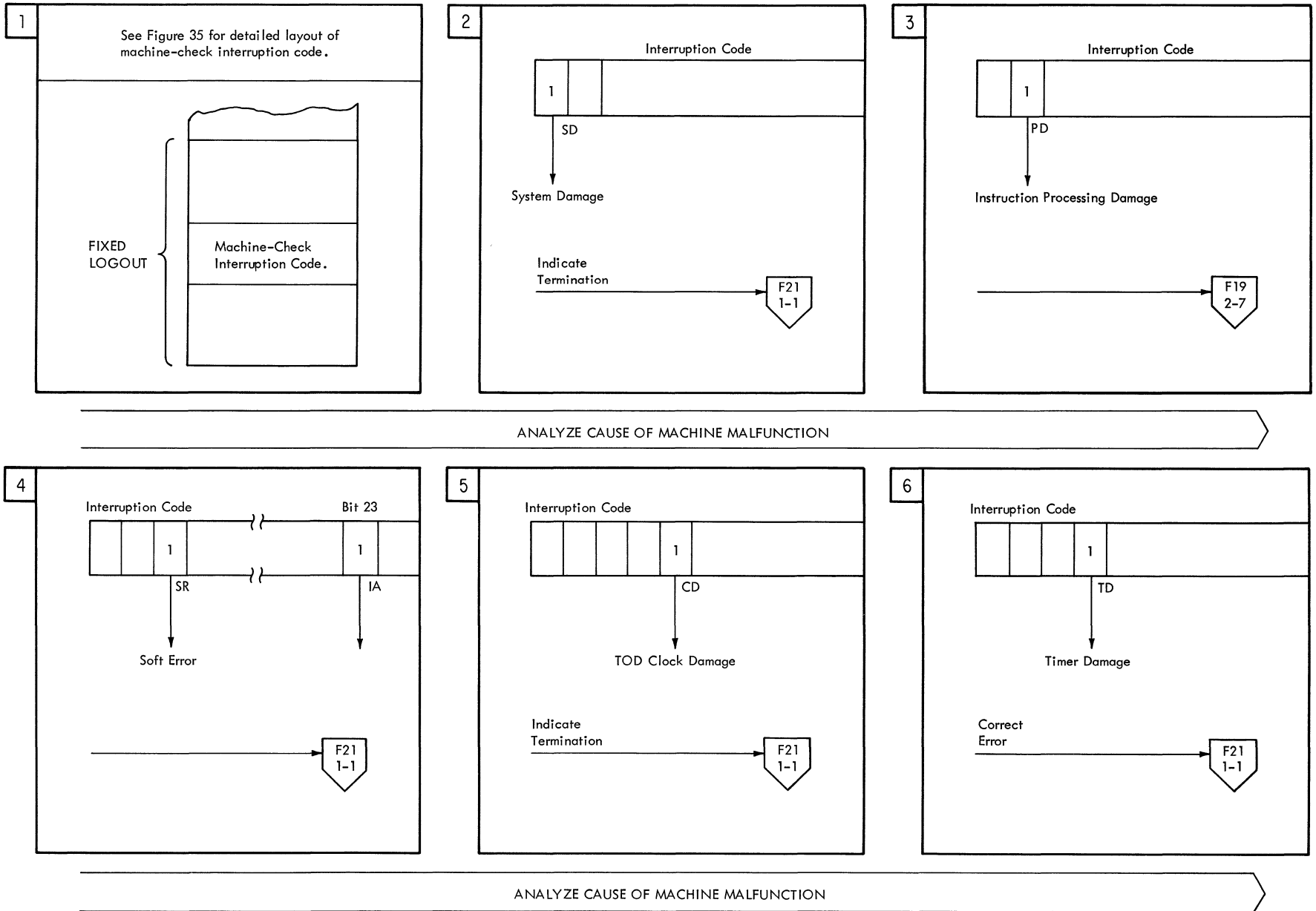


Figure 19. Hardware error analysis (Part 1 of 5)

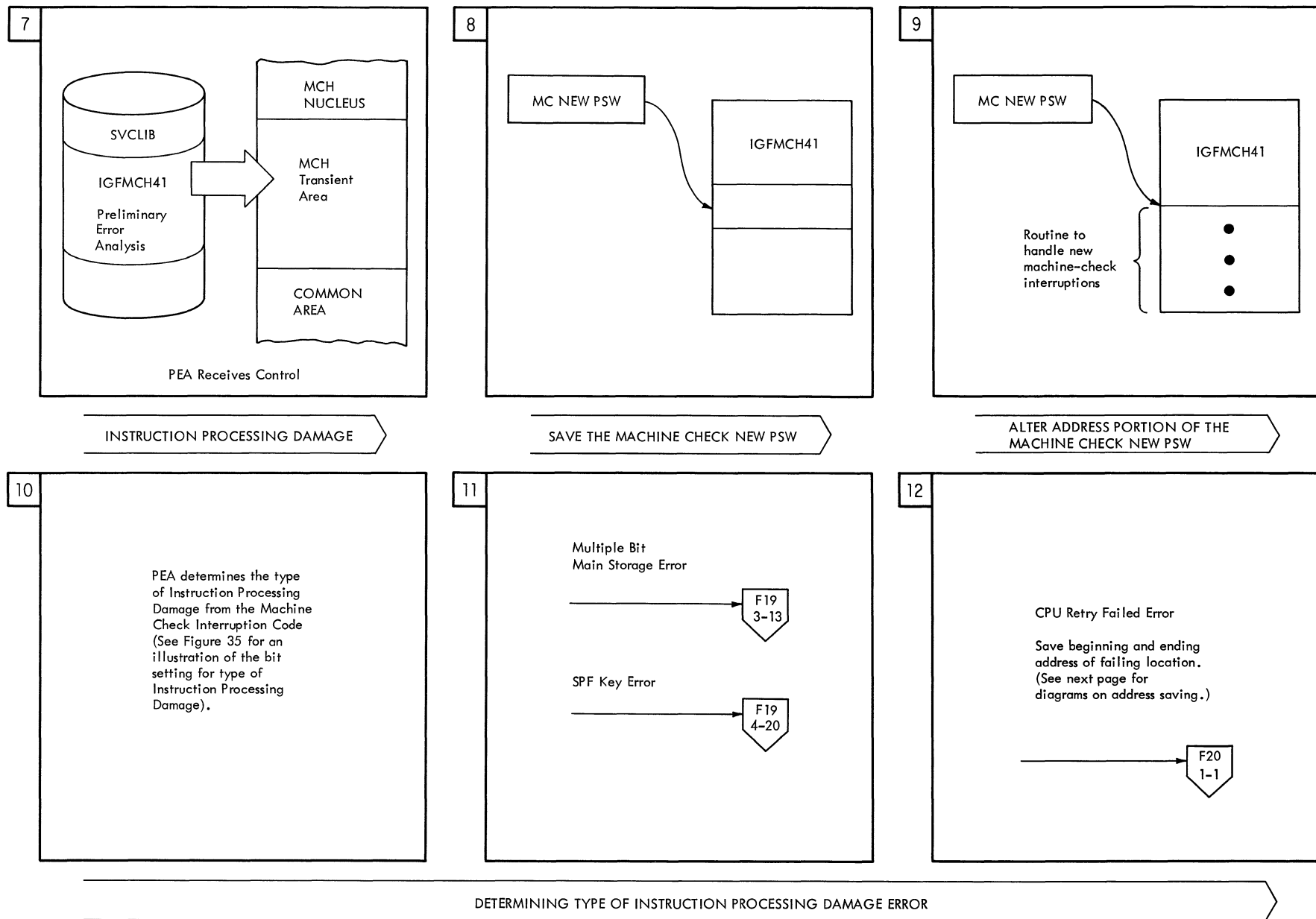


Figure 19. Hardware error analysis (Part 2 of 5)

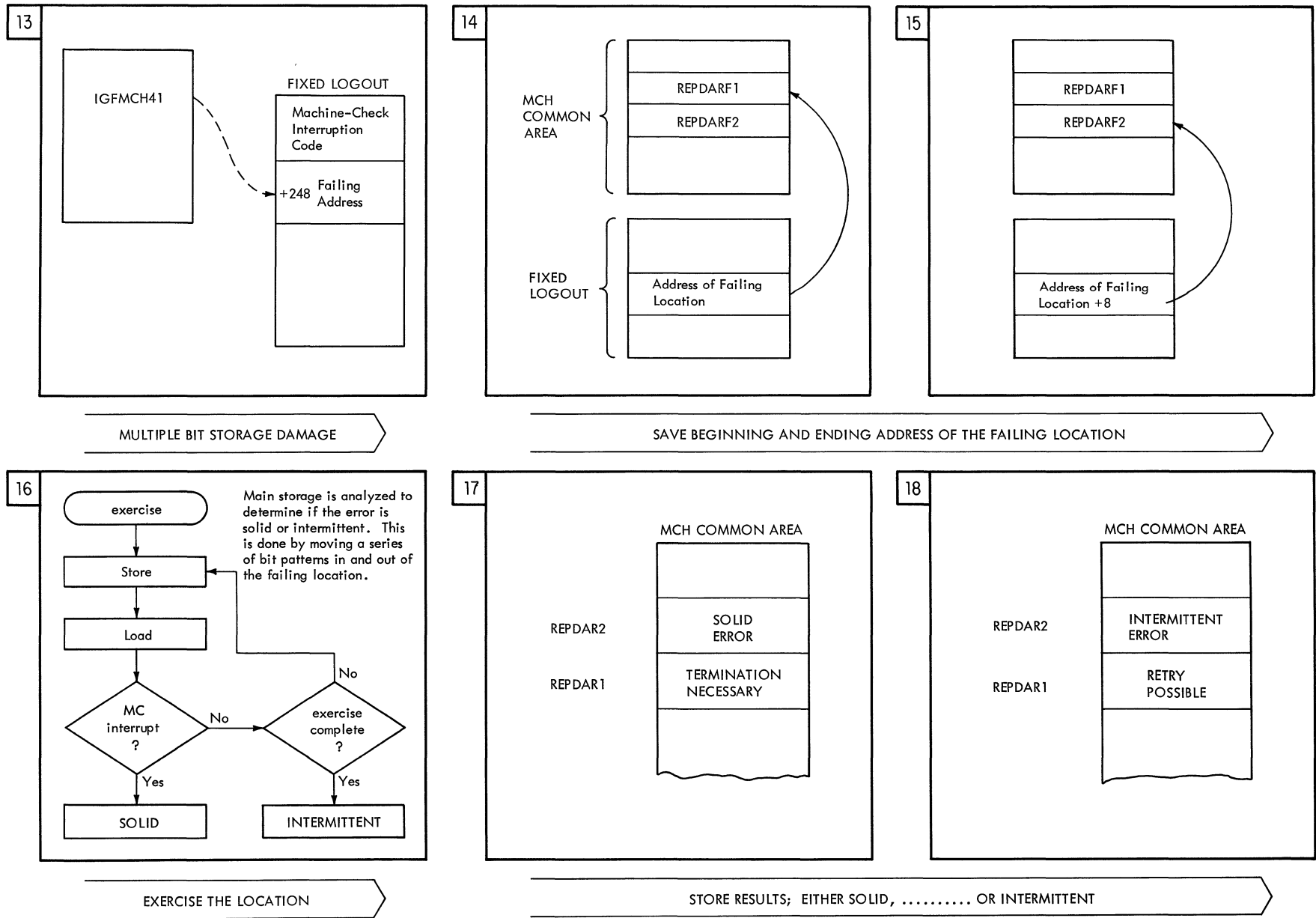


Figure 19. Hardware error analysis (Part 3 of 5)

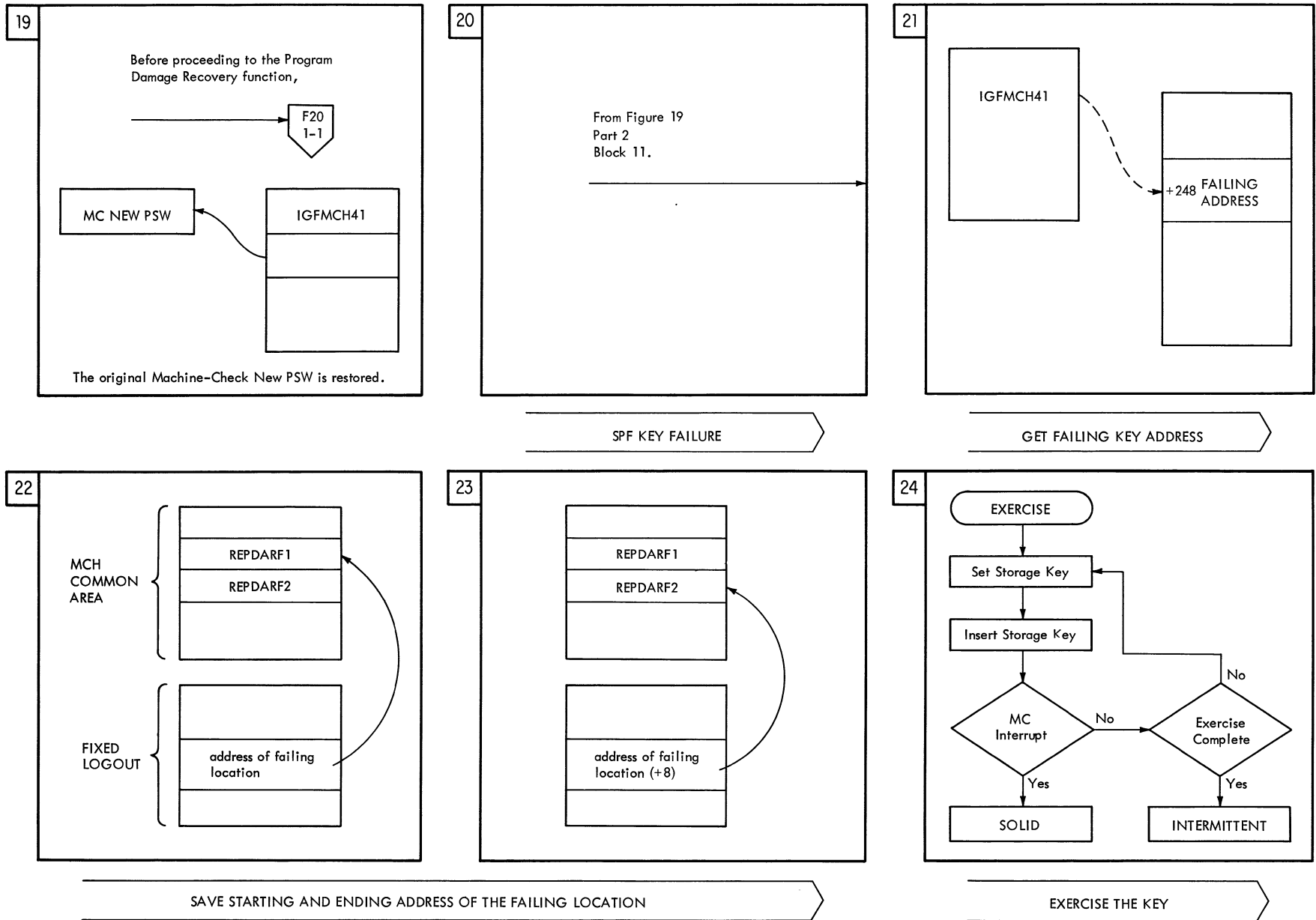


Figure 19. Hardware error analysis (Part 4 of 5)

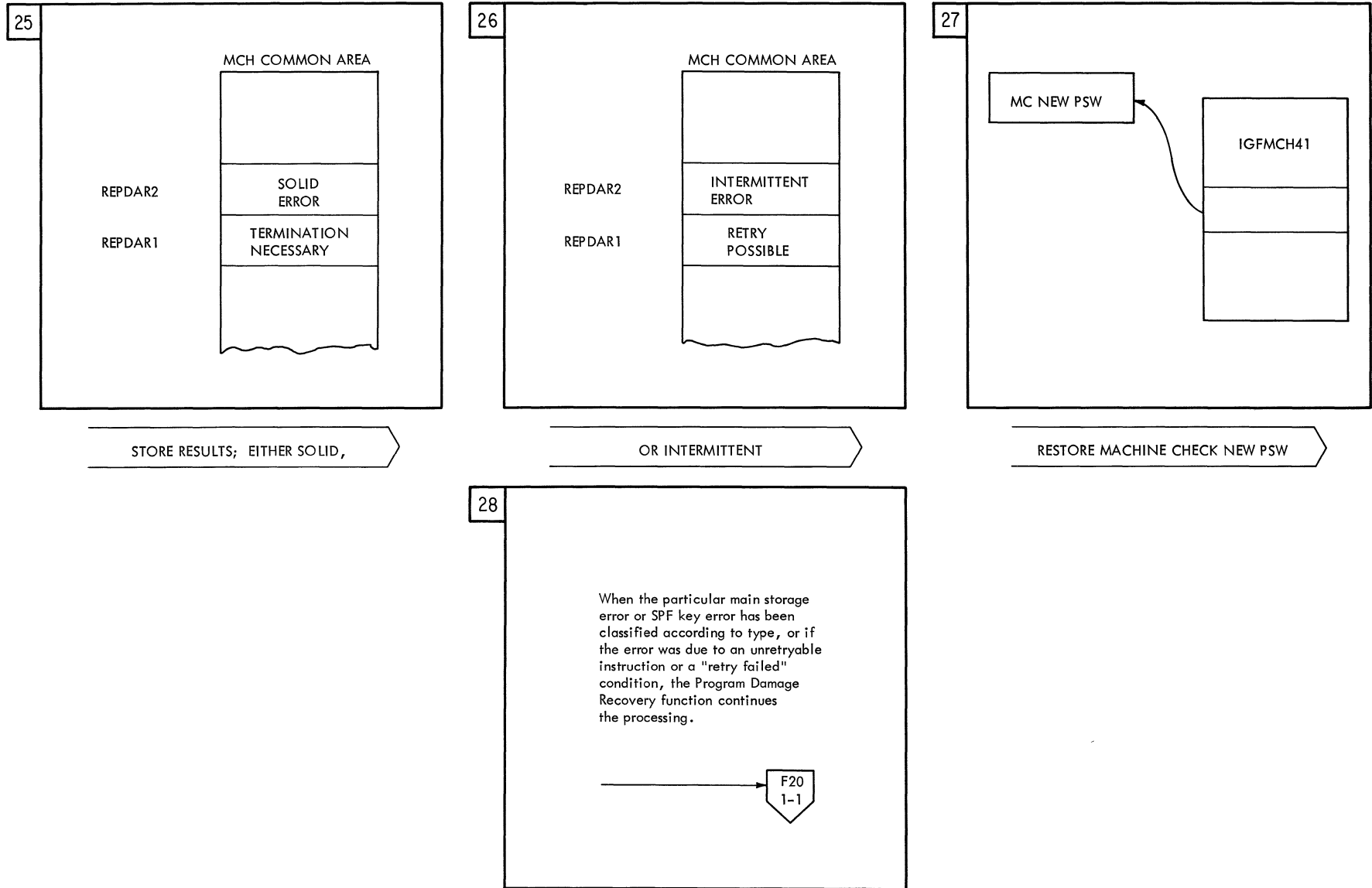


Figure 19. Hardware error analysis (Part 5 of 5).

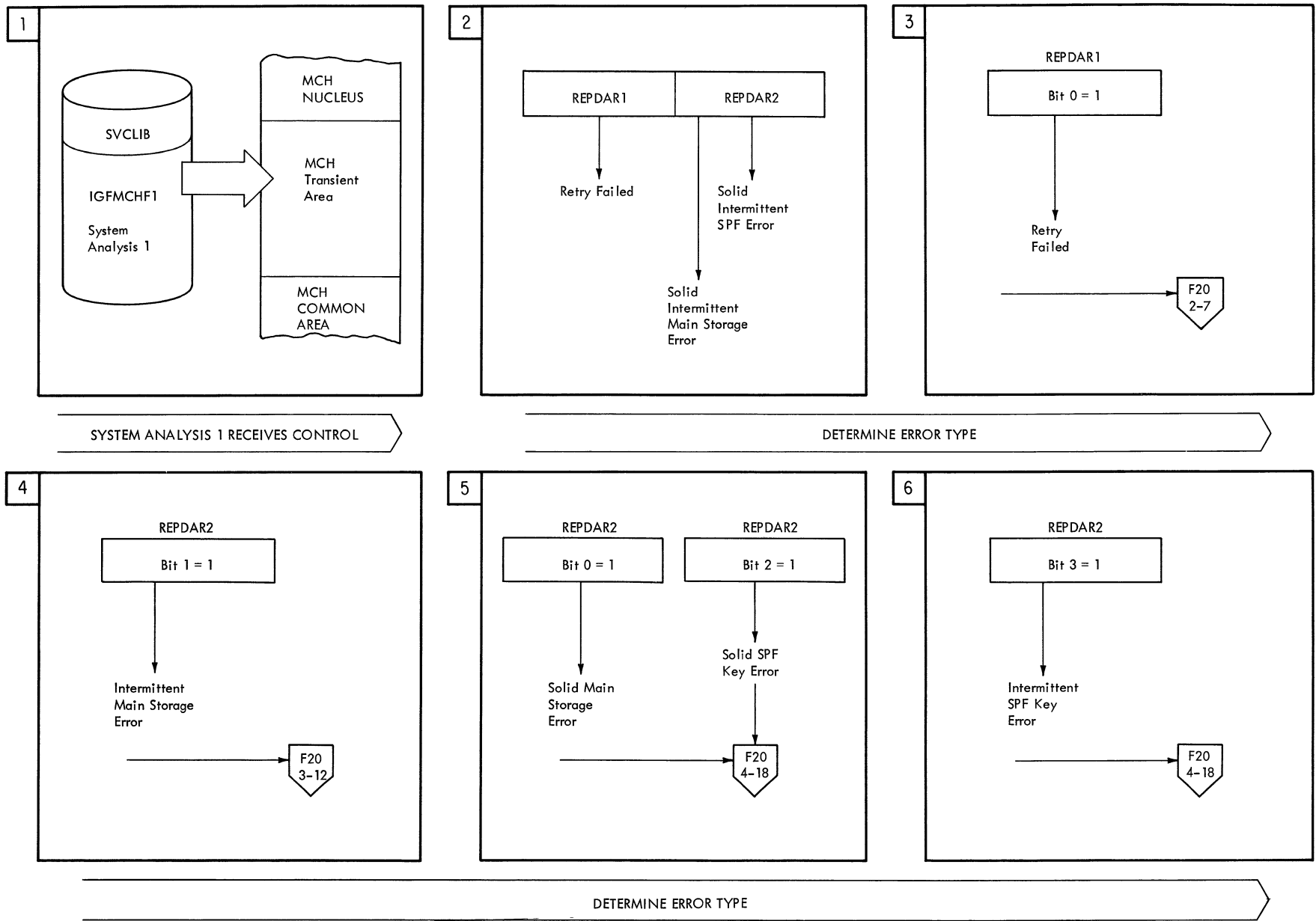


Figure 20. Program damage recovery (Part 1 of 4)

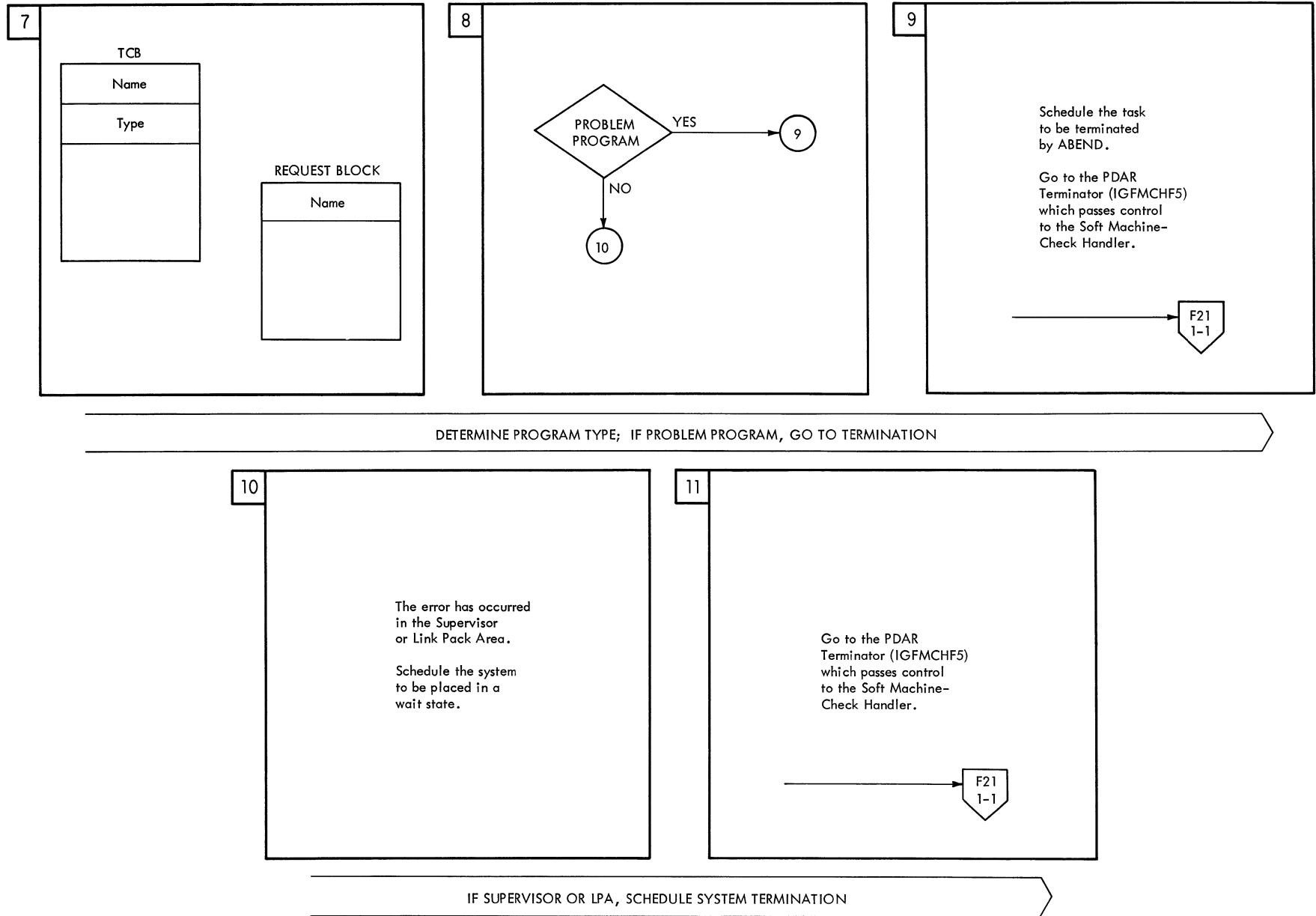


Figure 20. Program damage recovery (Part 2 of 4)

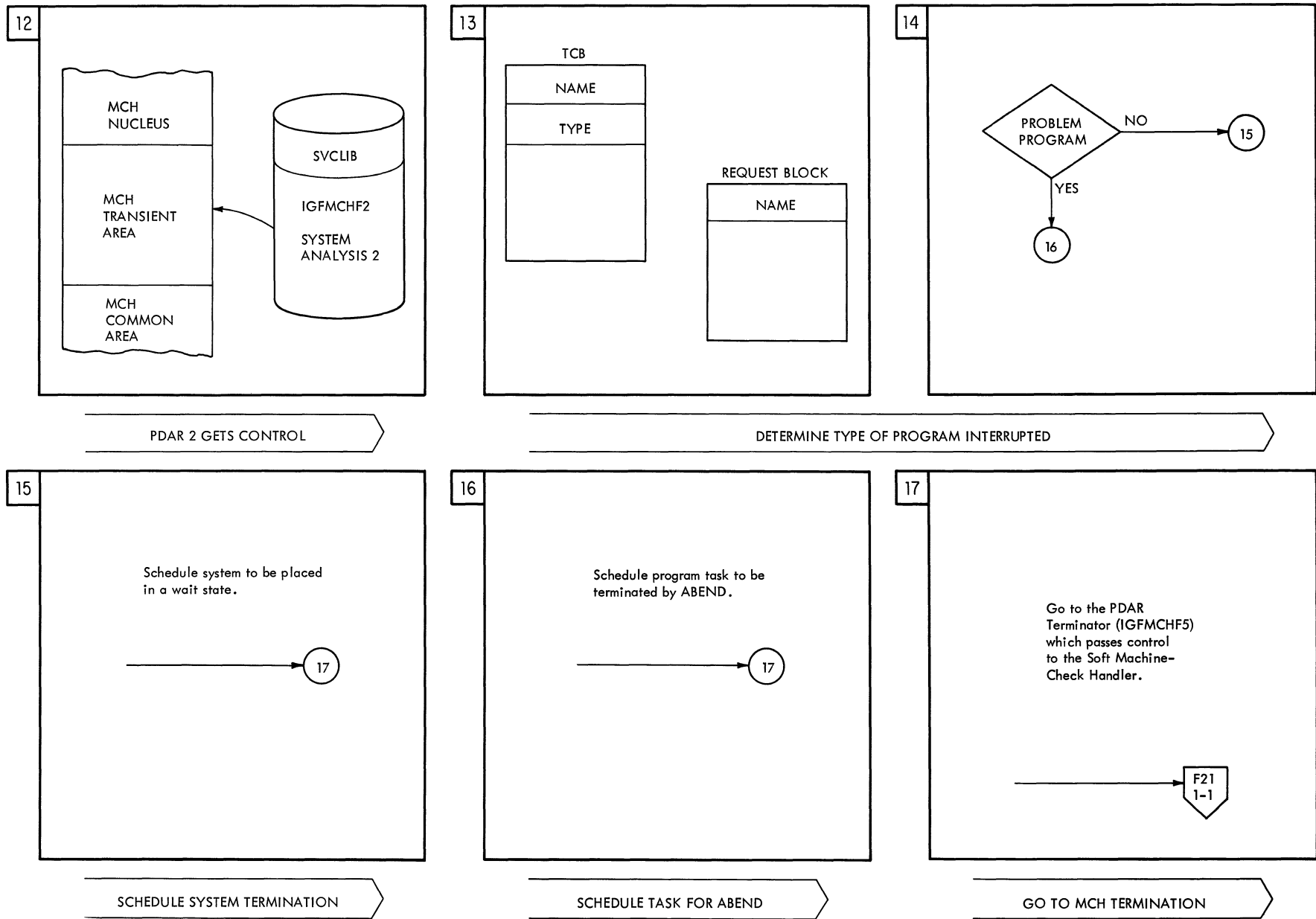


Figure 20. Program damage recovery (Part 3 of 4)

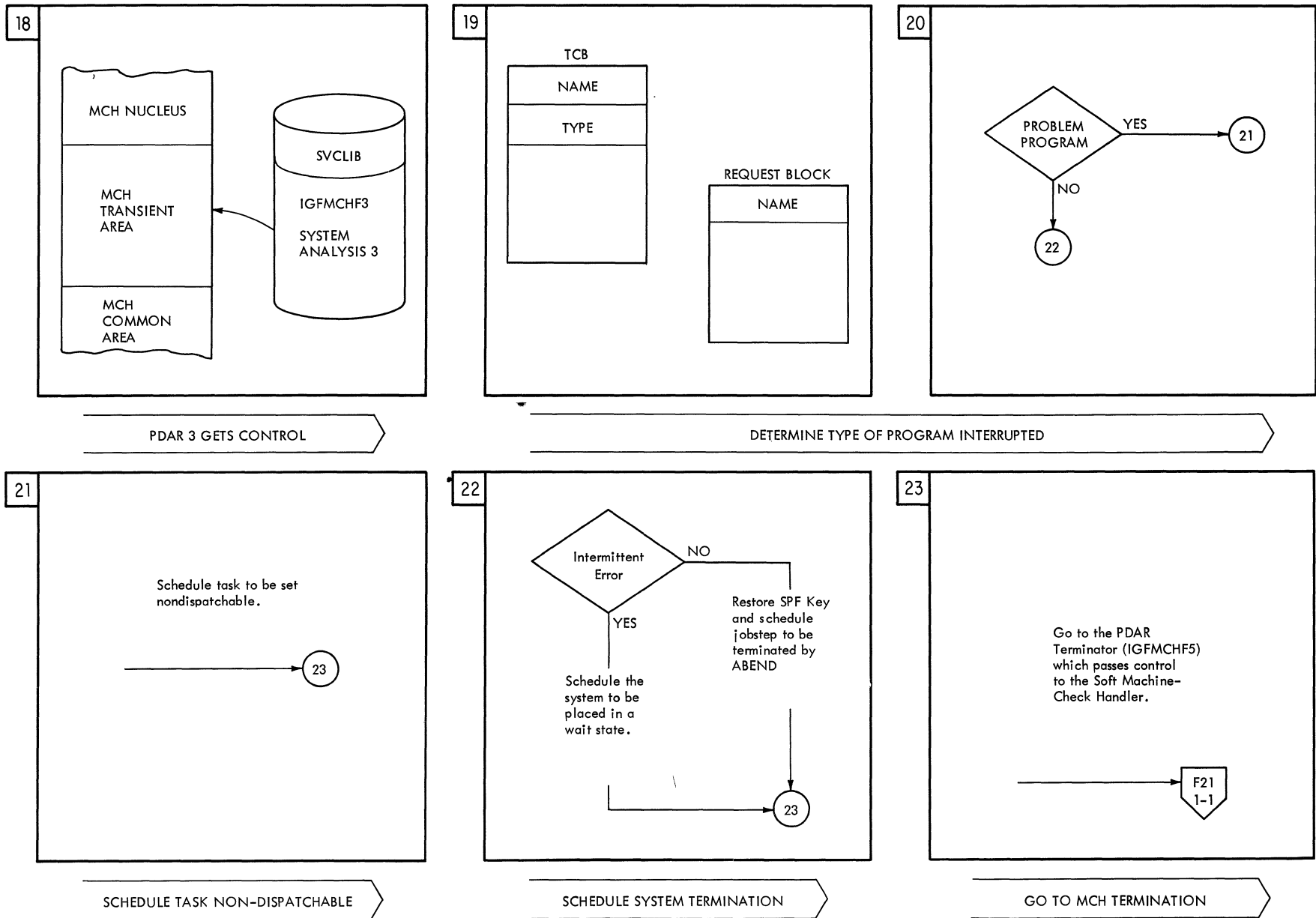


Figure 20. Program damage recovery (Part 4 of 4)

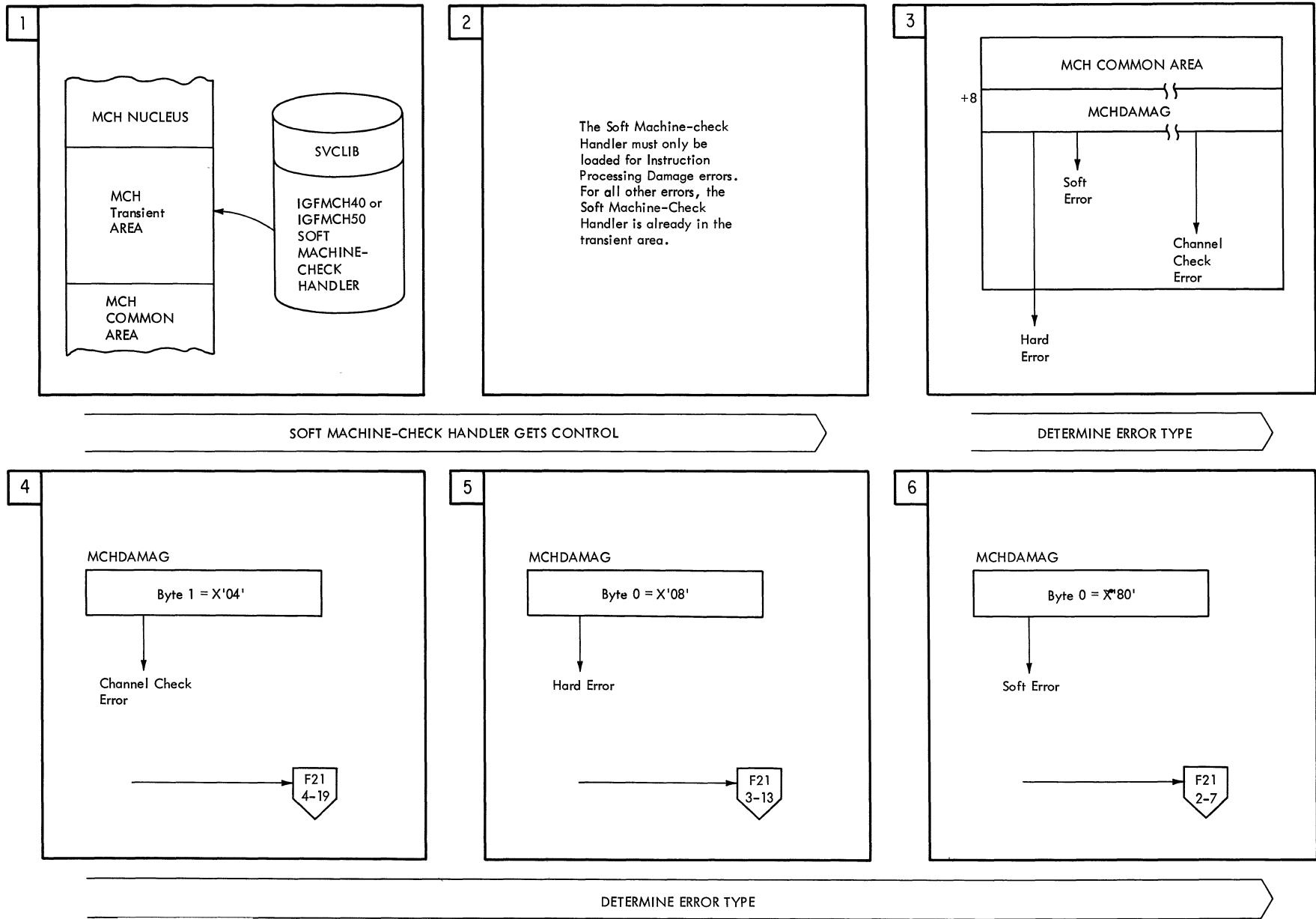


Figure 21. Recording and termination (Part 1 of 6)

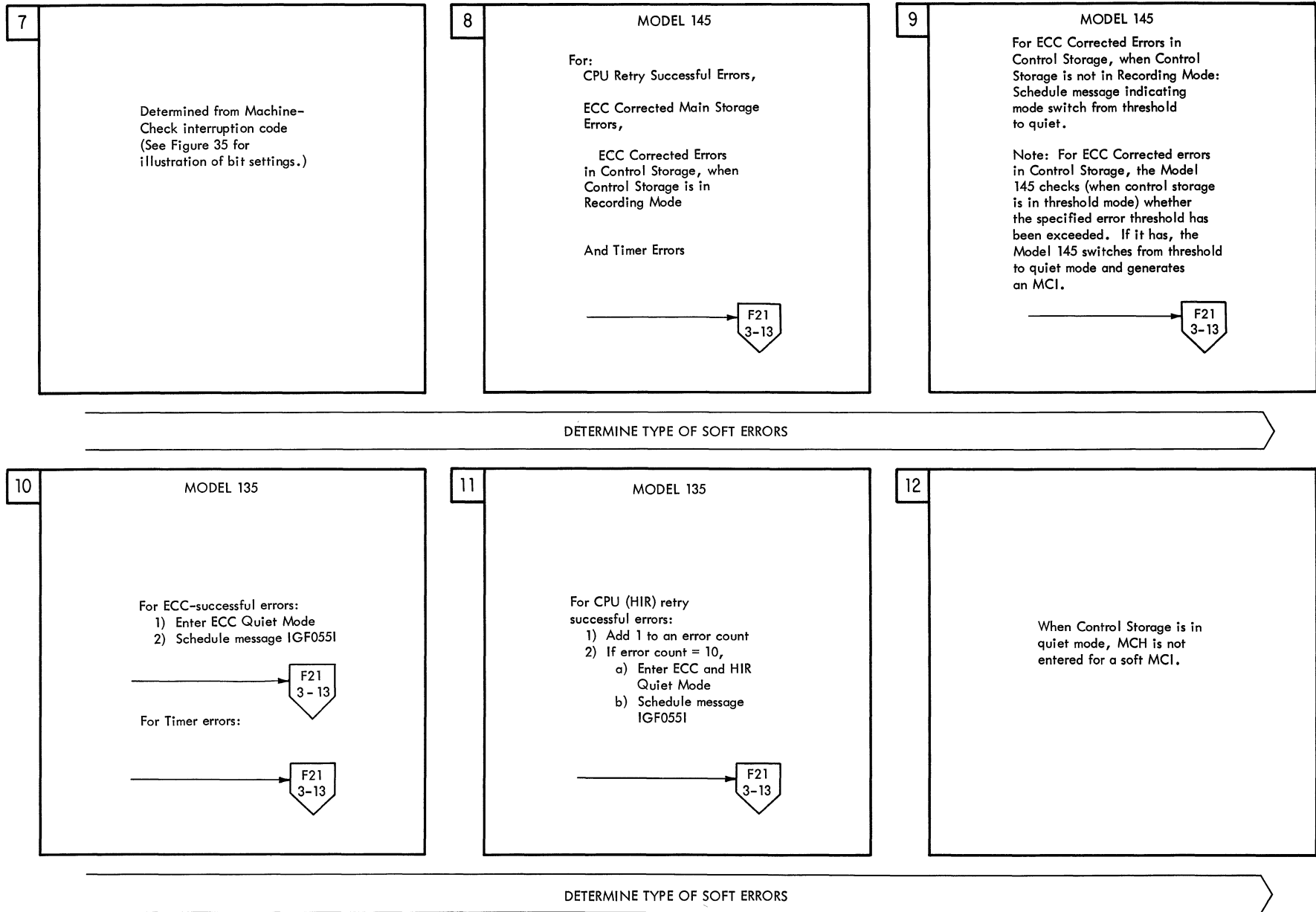


Figure 21. Recording and termination (Part 2 of 6)

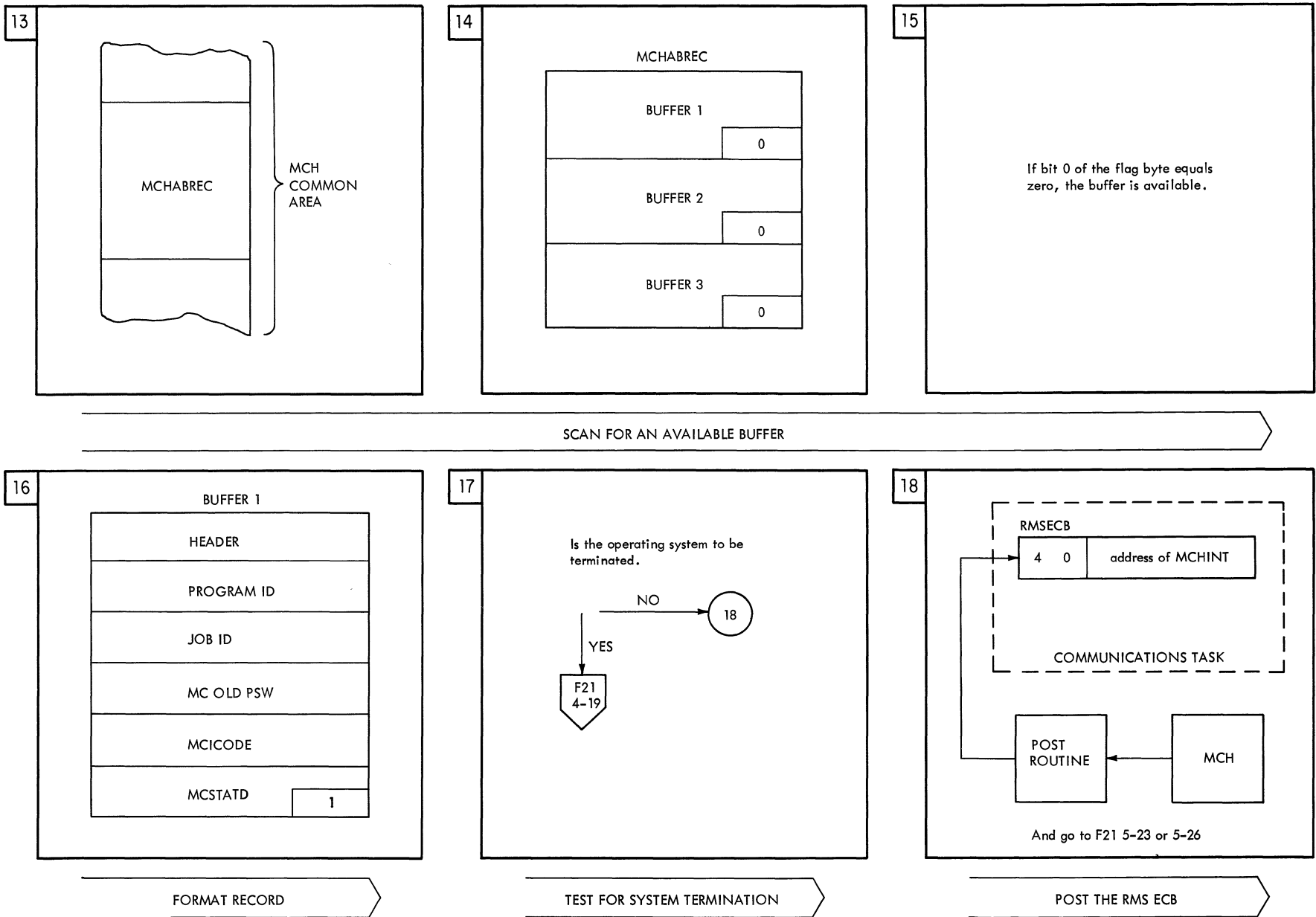


Figure 21. Recording and termination (Part 3 of 6)

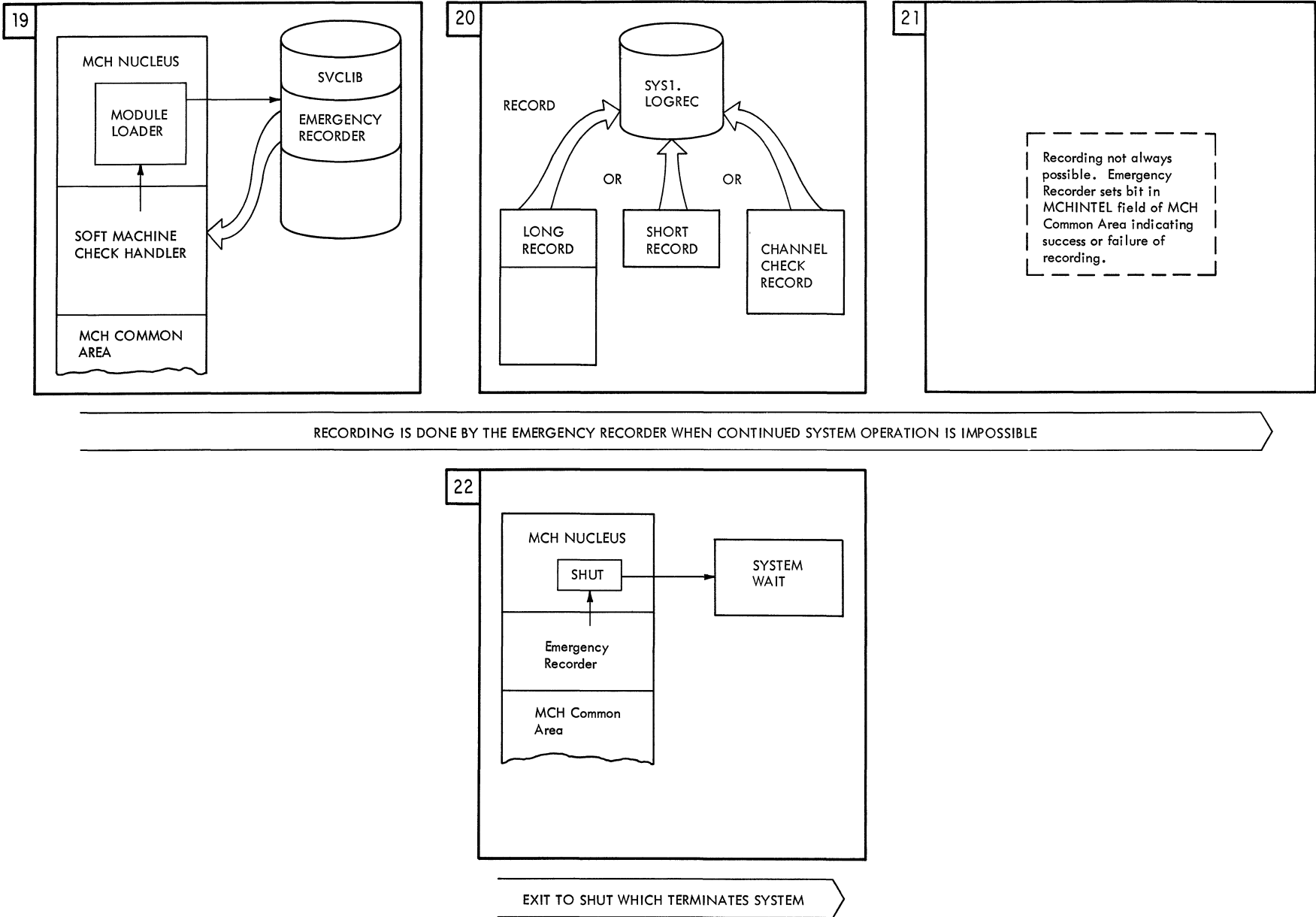


Figure 21. Recording and termination (Part 4 of 6)

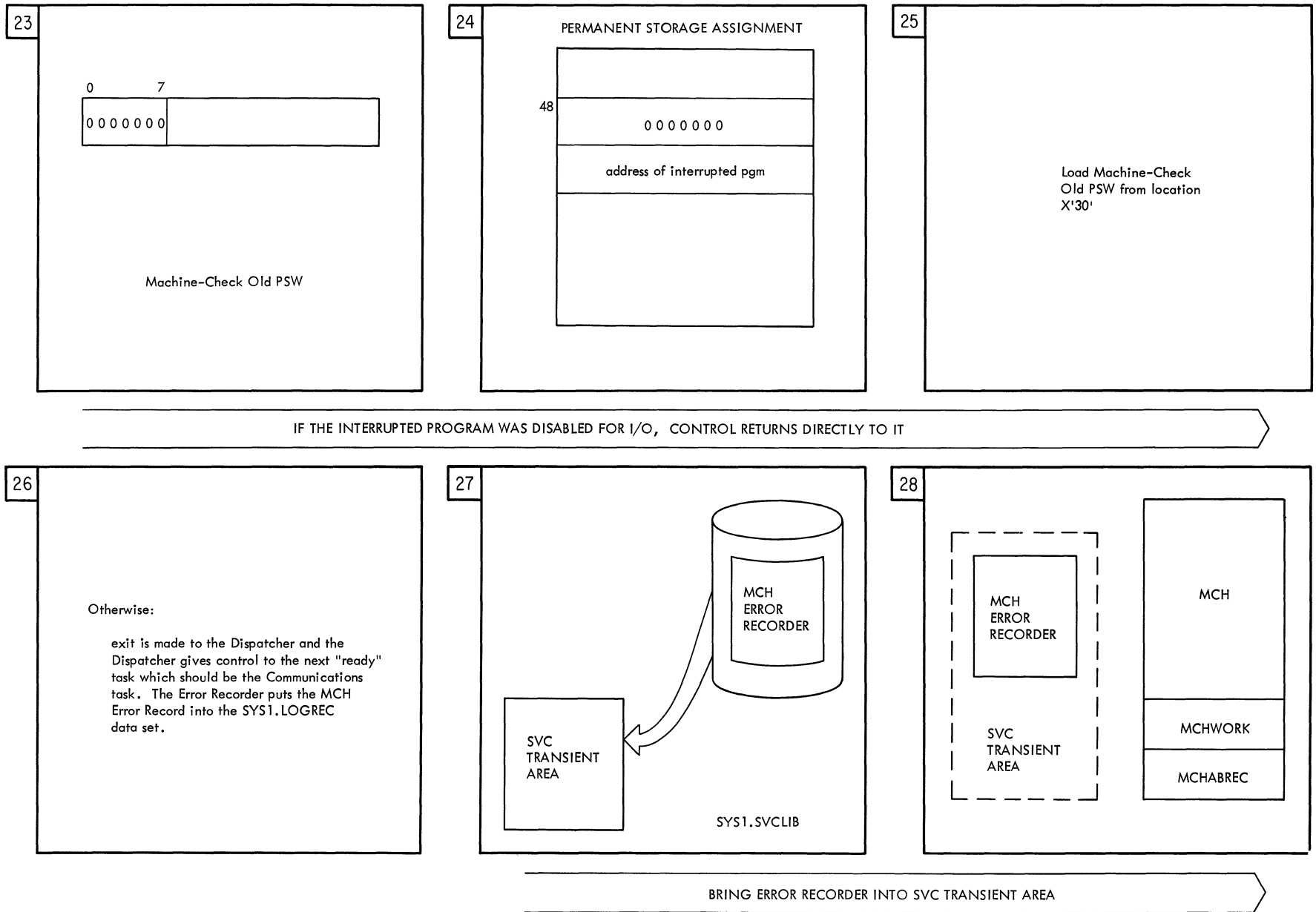


Figure 21. Recording and termination (Part 5 of 6)

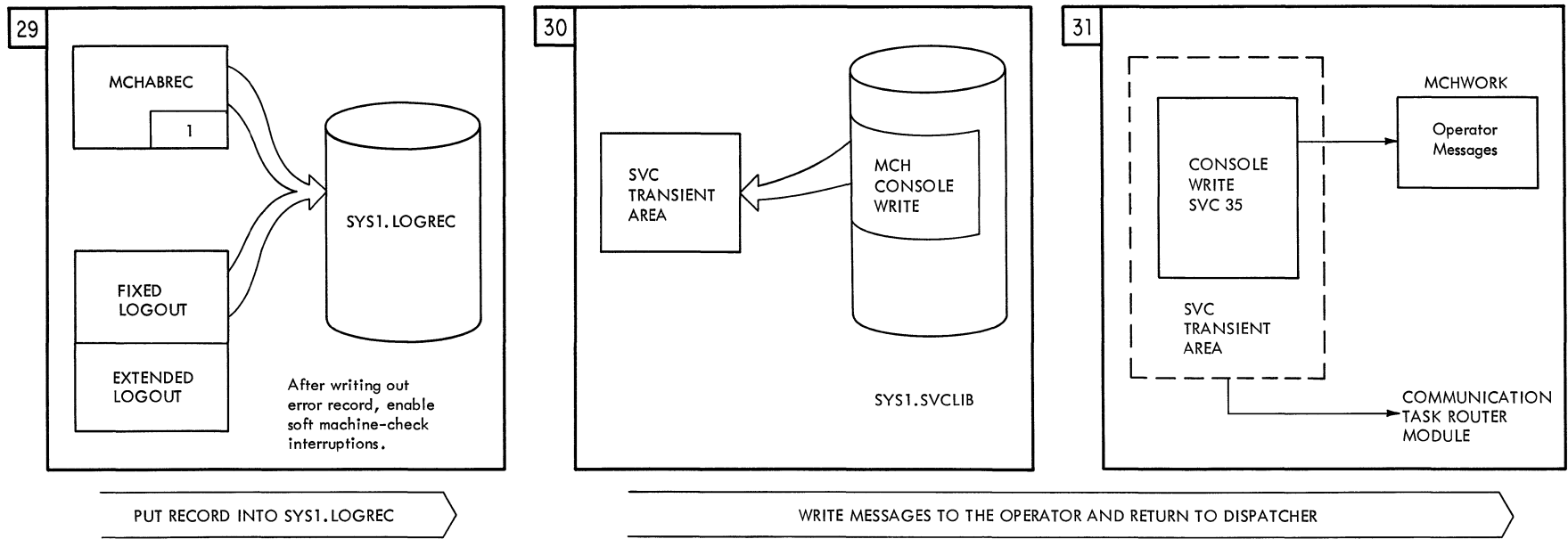


Figure 21. Recording and termination (Part 6 of 6)

This section contains descriptions and flowcharts for MCH modules. Each module description contains:

- English name of the module
- Module ID
- Functions
- Operation

Note: A module's ID is the same as its chart ID; the module ID corresponds to the module ID found in the microfiche listing. In addition, the charts are in the same sequence as the module descriptions.

MCH INITIALIZATION

Module ID: IGFMCHE0

Functions: MCH Initialization completes the initialization process started by the System Nucleus Initialization Program (NIP). Summarized here is that NIP Initialization of MCH.

Preliminary initialization of MCH during NIP processing insures that:

- MCH is incorporated into the operating system nucleus
- MCH is initialized with the values and addresses it needs for processing interruptions

Before passing control to IGFMCHE0, NIP:

- Checks whether the Machine-Check Handler is for the correct machine; if not, it issues a message to the operator informing him that MCH is inoperative.
- Loads the MCH Nucleus (IGFMCHE0) from the SYS1.LINKLIB into the dynamic area and passes control to it. One of the parameters NIP passes to the MCH Nucleus is a pointer to the first location in lower main storage which is not a part of the operating system nucleus. The MCH Nucleus relocates itself to that address, making itself contiguous with the operating system nucleus. MCH also updates the pointer to point to the new end of the operating system nucleus (including the MCH Nucleus).

The MCH Nucleus returns control to NIP, which deletes the copy of the MCH Nucleus

in the dynamic area, loads the MCH initialization module (IGFMCHF0) from the SYS1.LINKLIB, and passes control to it.

Operation: IGFMCHE0 initializes MCH in two stages.

During Stage 1 of its processing, IGFMCHE0:

1. Allocates space for the MCH Transient Area. Space is allocated by adding the number of bytes needed for the transient area to the address of the end of the operating system nucleus. For the Models 135 and 145, 1K bytes are needed. IGFMCHE0 then loads the appropriate Soft Machine-Check Handler into the transient area.
2. Allocates space for, and initializes, the Model-Dependent Common Area.
3. Initializes the Machine Status Block in the operating system nucleus with Multiple Console Support (MCS) control information for the nucleus.
4. Allocates space for, and initializes, the Model-Independent Common Area. This area serves as the MCH Communications area and is 1K bytes long.
5. Allocates space for the fixed logout save area. The fixed logout save area is 280 bytes long.
6. Allocates space for the Extended Logout. For the Model 145, the Extended Logout is 192 bytes long and is pointed to by control register 15. For the Model 135, the Extended Logout is 14 bytes long and is located within a scratch area of the Fixed Logout (at displacement 256, decimal).
7. Allocates a subsystem data area of 64 bytes if there is a subsystem present.
8. Initializes control register 14 with the machine-check mask.
9. Initializes pointers in the MCH Nucleus.
10. Initializes MCH fields in the Dispatcher. A pointer to the Post ECB routine in the MCH Nucleus is stored in the Dispatcher.
11. Initializes the machine-check new PSW.

12. Initializes fields for the Module Scheduler. The IDs and TTRs of the MCH transient modules on SYS1.SVCLIB are placed in the MCHTTRS field in the MCH Model-Independent Common Area. Successor IDs for those modules having successors are placed in the MCHNXIDS field.

After initializing the Module Scheduler, IGMCHF0 returns control to NIP, which saves the end-of-nucleus pointer and deletes the copy of IGMCHF0. Next, NIP constructs the System Queue Area (SQA), sets up the Link Pack Area, and relocates itself to just above the SQA. NIP then loads IGMCHF0 again and passes control to it.

During Stage 2 of its processing, IGMCHF0 initializes pointers in the MCH Common Area for the SVC and LINKLIB BLDL tables and then returns control to NIP, which deletes the copy of IGMCHF0 that was just loaded.

MCH NUCLEUS

Module ID: IGMCH0

Functions: The Nucleus:

1. Initializes the working environment of the Machine-Check Handler.
2. Screens out machine-check interruptions that occur during operation of the Channel-Check Handler.
3. Analyzes the cause of the malfunction and chooses the successor module.
4. Handles unexpected machine-check interruptions during MCH processing.
5. Intercepts Monitor Call interruptions and effectively No-op's them.
6. Handles module loading using the Input/Output Supervisor (IOS).
7. Handles system termination when required.

Operation: The MCH Nucleus receives control through the machine-check new PSW with the system disabled for all interruptions. It immediately masks out extended logouts and soft machine-check interruptions (the masking out is done in control register 14) and saves the data which is critical to the analysis of the error. This data includes:

1. The fixed logout (found in locations 232-511 decimal), which is saved immediately following the Record Buffer Build Area.

2. The machine-check old PSW, which is saved at MCHRPSW in the Common Area.

This data is saved immediately upon entry to the MCH Nucleus to avoid the destruction of data if a second machine check occurs. After the data is stored, the address in the machine-check new PSW is changed to point to the SHUT routine (IGFERR0) and the address in the program-check new PSW is changed to point to a special program-check handler routine (PROGCHK). The system is then reenabled for hard machine-check interruptions by setting the PSW bit 13 to 1. The Program Status Area (locations 0-127 decimal) is saved at MCHPSA in the Independent Common Area.

After the above initialization process has been completed, the MCH Nucleus screens out any machine-check interruptions originating in the Channel-Check Handler, passing control to the Soft Machine-Check Handler to initiate the recording of the Channel Check Record for those interruptions.

The MCH Nucleus then examines the machine-check interruption code in the fixed logout to determine the successor module. After the type of error condition is identified, the Module Loader portion of the MCH Nucleus is given control.

The Special Handler of Unusual Terminations (SHUT) subroutine is located in the MCH Nucleus and entered at IGFERR0 from the Emergency Recorder or from Preliminary Error Analysis when the operating system is to be placed in a wait state. SHUT writes a message containing the proper wait state code and loads a wait state PSW.

MCH MODULE LOADER

The Module Loader portion of the MCH Nucleus calls the I/O Supervisor to load the MCH transient modules. The loading is done by three subroutines, always resident within the MCH Nucleus:

- The module scheduling subroutine
- The I/O initialization subroutine
- The module loading subroutine

These subroutines, together with the services of the I/O Supervisor, handle all I/O for the Machine-Check Handler except recording errors and communications with operator.

Function of the Module Scheduler: The Module Scheduler interfaces between modules in the MCH Transient Area when they specify a successor module. The Module Scheduler

also maintains a history table of module execution. This history table (see Section 5) becomes part of the MCH error record when it is a full record.

Operation of the Module Scheduler: The Module Scheduler uses two tables to schedule a successor module:

- A successor table (pointed to by MCHNXIDS in the Independent Common Area) that contains the IDs of successor modules for each module that may call a successor. An ID is a one-byte field containing the last two digits of the module ID; for example, the XX of IGFMCHXX (see Figure 12).
- A displacement table (pointed to by MCHTTRS in the Independent Common Area) that contains the ID and relative track and record address (in TTR format) of each transient module as well as the displacement into the successor table to locate the successors for that module.

A module designates its successor by storing a one-byte path number in MCHNXMOD (in the Independent Common Area) before relinquishing control. The Module Scheduler then uses this path number plus a previously saved displacement value (see below) to index into the successor table, obtaining the ID of the desired successor module.

The Module Scheduler then searches the MCHTTRS table for a matching ID. When the ID is found, the corresponding TTR is obtained along with the displacement value for the new module's successor list. The TTR is saved for later conversion to an absolute track and record address (in MBBCCHHR format). The displacement is saved for use during the next execution of the Module Scheduler.

The Module Scheduler then:

1. Saves the TTR of the module to be loaded in the MCHTTRIN field of the Independent Common Area.
2. Prepares the channel program to load the module, placing the channel program address in an IOB (labeled MCHIOB).
3. Loads the address of the SYS1.SVCLIB Data Extent Block (DEB) into register 1.
4. Passes control to IGFLOAD in the Nucleus to complete preparations for the I/O operation.

Upon return of control from IGFLOAD, the Module Scheduler passes control to the module that was just loaded.

Function of I/O Initialization (IGFLOAD): The I/O Initialization subroutine converts the relative device address (TTR format) to an absolute address (MBBCCHHR format). It also completes the initialization of those I/O control blocks required by the module loading routine.

Operation of I/O Initialization: I/O Initialization obtains an absolute device address using a system device dependent characteristics table (IECZDTAB) and the extent information in the DEB whose address is in register 1.

I/O Initialization then passes control to the module loading routine at entry point IGFIORTN.

Function of the Module Loader (IGFIORTN): The Module Loader uses a special interface with the I/O Supervisor to load the MCH transient modules from SYS1.SVCLIB. The Module Loader also performs the I/O operations for the Emergency Recorder (if the system is going into a wait state).

Operation of the Module Loader: Routines requesting a load operation enter the Module Loader after initializing the DCB, DEB, IOB, and CCW chain for the module to be loaded.

All registers and the I/O new PSW address are saved; the address in the new I/O PSW is replaced with the address of the MCH First Level Interrupt Handler (MCHFLIH) in the MCH Nucleus. The DCB specified in the DEB is chained to the MCHIOB. The address of an appendage table (APNTBL in the MCH Nucleus) is stored in the DEB.

When control is passed to IOS to Start I/O, register 14 contains the address of a parameter (MCHIOSWD) and register 1 contains the address of the IOB for execution. The action that IOS is to perform is indicated in the MCHIOSWD parameter as follows:

Bit 0=0	Normal MCH entry, honor the request
Bit 0=1	Final MCH entry, dequeue the MCH RQE
Bit 1=1	Clear busy and post indicators in the UCB
Bit 2=1	Internal MCH recursion indicator
Bits 3-31	Reserved

IOS returns to MCH 4 bytes past the beginning of the in-line coded parameter MCHIOSWD. MCH reestablishes a base register and loads register 2 with the address of the UCB. MCH issues the IOSGEN macro

instruction to enable the channel it has to use and then enters a pseudo wait state (a bit-spin loop). The completion of an MCH I/O event (via MCH appendage routines) causes exit from the loop.

If the MCH I/O event does not complete, the bit-spin loop eventually times out. When this happens, MCH attempts to load the Soft Machine-Check Handler to write an error record and enter a wait state. Should a second time out occur during this attempt, MCH immediately enters a wait state without writing an error record.

The IOSGEN macro instruction is then issued to turn off the enabled channel, the I/O new PSW is stored, and the MCH ECB is tested to determine the success of the load operation.

If the load operation is successful, the Module Loader restores registers and returns control to the Module Scheduler via register 14. If the I/O operation to perform a load is unsuccessful, the Module Loader returns control via register 14 plus a displacement of 4.

The Abnormal End Appendage routine is entered upon detection of an unrecoverable I/O error by IOS. IOS describes the error by posting a code in the MCHIOECB, a field of the MCHIOB in the Independent Common Area. When that code is X'7F', the operation in error was successfully retried, and the Abnormal End Appendage routine exits to IOS via register 14 (with a displacement of zero). When that code is X'44', a device-end error occurred, and the Abnormal End Appendage routine returns to IOS via register 14 with an offset of 8. When that code is X'41', a permanent error has occurred. In this case, the abnormal end appendage routine places the X'41' in the MCH ECB (in the MCH Nucleus) and returns to IOS via register 14 with a displacement of 12, bypassing the Post routine and returning the Request Queue Element (RQE) to a free list.

The Normal End Appendage routine sets a successful completion code in the MCH ECB and exits to IOS via register 14 plus a displacement of 12. IOS determines whether it was entered from the Abnormal or the Normal End Appendage routine by the MCH ECB posting.

SOFT MACHINE-CHECK HANDLER (MODEL 135 ONLY)

Module ID: IGMCH50

Functions: The Soft Machine-Check Handler (SMCH):

1. Prepares error records for the SYS1. LOGREC data set.

2. Determines the type of soft errors and performs any mode switching required as a result of soft errors. It issues a message whenever a mode switch is performed because of an error.
3. Performs normal termination of the Machine-Check Handler.

Operation: The Soft Machine-Check Handler is loaded into the MCH Transient Area during system initialization and may be overlaid by subsequent MCH modules loaded for the processing of hard machine-check interruptions.

Mode Handling for the Model 135: When ECC-corrected single-bit storage errors occur at a rate of at least 256 errors in 416 micro-seconds, the Error Frequency Limit (EFL) is exceeded and a soft machine-check occurs. To process these errors, the Soft Machine-Check Handler:

1. Issues a Diagnose instruction to disable the EFL function from performing additional interruptions.
2. Sets the ECC Quiet indicator in the Machine Status Block.
3. Indicates storage damage in the MCHDA-MAG field of the Common Area.
4. Schedules the message: IGF055I QUIET MODE ECC.
5. Continues with Record Buffering and Formatting (see below).

Note that no instruction stream can access main storage fast enough to exceed the EFL. For this reason, only control storage errors can generate soft machine-check interruptions as the result of exceeding the EFL.

When the twentieth CPU-retry corrected error occurs, the Soft Machine-Check Handler:

1. Sets bit 4 of control register 14 to disable all soft machine-check interruptions (both ECC and CPU retry).
2. Sets HIR (CPU retry) and ECC Quiet Indicators in the Machine Status Block. Whenever CPU retry is set to quiet mode, ECC is also set to quiet mode since the bit set in control register 14 disables all soft machine-check interruptions.
3. Indicates storage damage in the MCHDA-MAG field of the MCH Common Area.
4. Schedules the message: IGF055I QUIET MODE ECC,HIR.

5. Continues with Record Buffering and Formatting (see below).

Note that soft machine-check interruptions not requiring a mode switch are merely handled as described under "Record Buffering and Formatting."

Record Buffering and Formatting: The record buffering routine of the Soft Machine-Check Handler (SMCH) scans the record, storing the address of the current record buffer in the MCHLONG field of the Independent Common Area for use by the record formatting routine. At the end of each record buffer, there is a flag byte, which indicates the status of the record:

- Bit 0=1 Active record
- Bit 1=1 Short record
- Bit 2=1 Full record (Fixed and extended logouts are still intact)
- Bit 3=1 The previous record in the buffer has been overlaid.

Since there is room in the MCH record buffer for only one record containing the fixed and extended logout, a record is overlaid if a second, hard machine-check interruption is generated before the previous record has been recorded. However, to prevent the complete loss of records, the most critical parts of each record are saved, and short records (up to 3) may be generated and queued in the short record buffer.

If more than three hard machine checks occur before recording, previous short error records are overlaid and a counter is updated, indicating the number of records lost. Figure 22 shows the extended and fixed logout areas (the extended logout for the Model 135, however, is only 14 bytes long and is contained within the fixed logout), the short record buffers, the order in which they may be overlaid, and the location and use of the lost record counter.

The record buffer routine exits to the record formatting routine, which completes the MCH record by setting up short records (ABRECS).

Note: See Figure 13 for the format of the MCH record.

MCH Termination: MCH relinquishes control at the end of its processing in one of the following manners:

- It returns control to the interrupted program if that program was disabled for I/O at the time of the interruption. Before returning to the interrupted program, MCH sets a No-operation instruction in the Dispatcher, so that

the error record is posted when the interrupted program gives up control.

- It gives control to the nucleus to post the error record and upon return it exits to the Dispatcher (when normal recording is to be done).
- It gives control to the Emergency Recorder (IGFMCHE3) when continuation of system operation is impossible (determined by a bit set in MCHDAMAG).

SOFT MACHINE-CHECK HANDLER (MODEL 145 ONLY)

Module ID: IGFMC40

Functions: The Soft Machine-Check Handler (SMCH):

1. Prepares error records for recording on the SYS1.LOGREC data set.
2. Determines the type of soft errors and the mode of operation for storage errors. It issues a message when the hardware has placed control storage in the quiet mode as a result of exceeding the allowable error frequency.
3. Performs normal termination for the Machine-Check Handler.

Operation: The Soft Machine-Check Handler is loaded into the MCH Transient Area during system initialization and may be overlaid by subsequent MCH modules loaded for the processing of hard machine-check interruptions.

Mode Handling for the Model 145: When a corrected CPU error or an ECC-corrected main storage error occurs, IGFMC40 sets appropriate bits in the damage assessment field (MCHDAMAG of the Independent Common Area); puts a message code into the buffer (indicating that the CPU retry or ECC successful message is to be issued); and passes control to the record buffer management portion of the Soft Machine-Check Handler (discussed below).

For control storage errors, SMCH tests for record mode. When in record mode, SMCH handles these errors in the same manner as a CPU retry or an ECC-corrected main storage error. Otherwise, SMCH schedules a message indicating a switch (which occurred before the machine-check interruption) from threshold to quiet mode.

Record Buffering and Formatting: The record buffering routine of SMCH scans the record, storing the address of the current record buffer in the MCHLONG field of the Independent Common Area for use by the record formatting routine. At the end of each

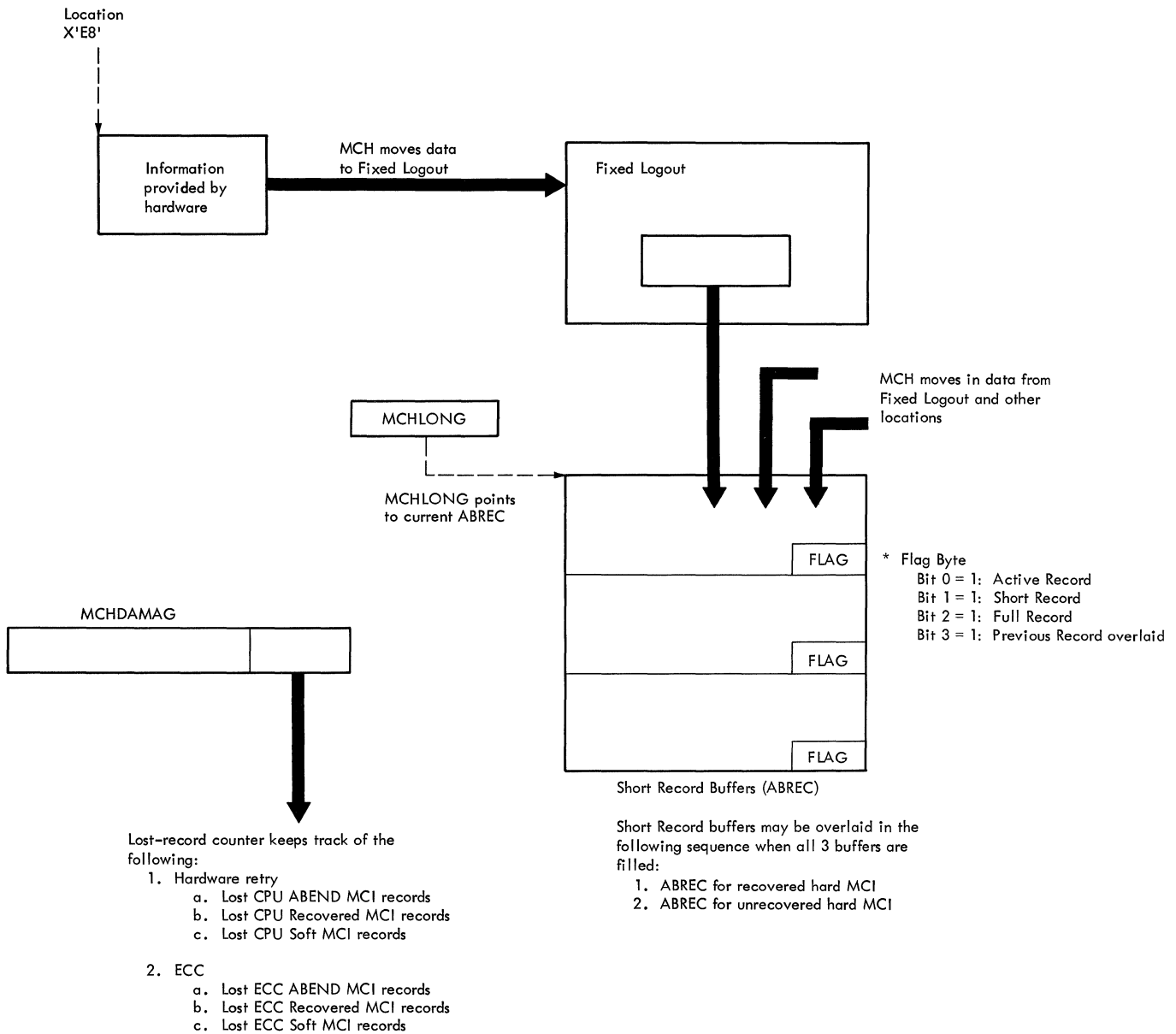


Figure 22. Use of buffers and the lost-record counter in recording

record buffer there is a flag byte, which indicates the status of the record:

- Bit 0=1 Active record.
- Bit 1=1 Short record.
- Bit 2=1 Full record (Fixed and extended logouts are still intact).
- Bit 3=1 The previous record in the buffer has been overlaid.

Since there is room in the MCH record buffer for only one record containing the

fixed and extended logout, a record will be overlaid if a second interruption is generated before the previous record has been recorded. However, to prevent the complete loss of records, the most critical parts of each record are saved, and short records (up to 3) may be generated and queued in the short record buffer.

If more than three machine checks occur before recording, previous short error records are overlaid and a counter is updated, indicating the number of records lost. Figure 22 shows the extended and fixed logout areas, the short record buf-

fers, the order in which they may be overlaid, and the location and use of the lost-record counter.

The record buffer routine exits to the record formatting routine, which completes the MCH record by setting up short records (ABRECS).

Note: See Figure 13 for the format of the MCH record.

MCH Termination: MCH relinquishes control at the end of its processing in one of the following manners:

- It returns control to the interrupted program if that program was disabled for I/O at the time of the interruption. Before returning to the interrupted program, MCH sets a No-operation instruction in the dispatcher, so that the error record will be posted when the interrupted program gives up control.
- It gives control to the nucleus to post the error record and upon return it exits to the Dispatcher (when normal recording is to be done).
- It gives control to the Emergency Recorder (IGFMCH3) when continuation of system operation is impossible (determined by a bit set in MCHDAMAG).

PRELIMINARY ERROR ANALYSIS

Module ID: IGFMC41

Functions: The Preliminary Error Analysis routine examines the machine-check interruption code and the fixed and extended logouts to determine the recovery strategy for MCH.

Operation: Preliminary Error Analysis (PEA) is the first transient MCH module to be loaded when the Instruction Processing Damage (PD) bit of the machine-check interruption code is found to be on. (When this bit is on, it indicates that an instruction or information in a register has been changed.) PEA determines whether the machine-check interruption code is valid. If it is not, PEA sets the "system termination necessary" bit in the MCHDAMAG field of the Common Area and designates the Soft Machine-Check Handler as its successor (the Soft Machine-Check Handler passes control to IGFMCHE3 to write MCH records, and IGFMCHE3 passes control to SHUT to stop the system).

If the machine-check interruption code is valid, PEA saves the machine-check new PSW, replacing it with one that points to a

section of PEA designed to handle all interruptions. PEA then determines whether the Instruction Processing Damage (PD) is the result of a storage error, an SPF key error, or a hardware-retry-failed error.

If a storage error occurred, PEA does the following:

1. It checks the machine-check interruption code for a valid, failing, storage address. If the address is invalid, PEA schedules the Soft Machine-Check Handler, which terminates the system.
2. If PEA finds a valid failing storage address, it stores the beginning and ending addresses of the doubleword that includes the location in error in the REPDARF1 and REPDARF2 fields of the Independent Common Area.
3. It uses the Store Multiple and Load instructions to test the failing location. The following bit patterns are stored and fetched:
 - a. All binary 0's
 - b. All binary 1's
 - c. Binary 1's and 0's (101010...)
 - d. Binary 0's and 1's (010101...)

If there is a machine-check interruption that can be identified as a result of the storing or fetching, or if any bits are altered, the solid error switch in REPDAR2 is turned on. Otherwise, the intermittent error switch in REPDAR2 is set since the original error cannot be duplicated. PEA restores the machine-check new PSW and returns control to the MCH Nucleus, which loads IGFMC41.

If a Storage Protect Feature (SPF) key error occurred, PEA first acts as described above in steps 1 and 2 for storage errors. PEA then tests the SPF key in the following manner to determine whether the error is solid or intermittent:

1. It uses the Set Storage Key (SSK) and Insert Storage Key (ISK) instructions to test all possible 4-bit combinations. If a machine-check interruption occurs on the execution of either instruction on any single pattern (that is, if the original error can be duplicated), the error is considered to be solid. The error is also considered to be solid if the bit pattern changes after it is set in the location (by SSK) or after it is inserted back into the register (by ISK).

2. For either type of solid error, the solid error indicator is set in REPDAR2. When all bit patterns are tested without a machine check, the error is considered to be intermittent. In this case, the intermittent indicator is set in REPDAR2, and control is passed to IGFMCHF1.
3. If a hardware-retry-failed error occurs, PEA sets a "termination necessary" bit in the MCH Common Area and passes control to IGFMCHF1.

SYSTEM ANALYSIS

There are two sets of system analysis modules: one set for MFT and one set for MVT. Only the MFT set is used with MCH for the Model 135 since only the MFT version of the operating system may be run on a Model 135. Either the MFT or the MVT set may be used on the Model 145.

Each set contains three modules. For an MFT operating system, these modules are:

IGFMFTF1
IGFMFTF2
IGFMFTF3

For an MVT operating system, the corresponding modules are:

IGFMVTF1
IGFMVTF2
IGFMVTF3

These system analysis modules are called Program Damage Assessment and Repair (PDAR) modules. They prepare the system for one of the following machine-check recovery procedures:

- Setting nondispatchable those jobsteps associated with solid storage or SPF errors, to circumvent the solid error.
- Abnormally terminating a jobstep using the ABEND routine.
- Placing the system in a wait state when an error that affects a critical system task cannot be corrected.

Note: In other parts of this manual, these PDAR modules are referred to as IGFMCHF1, IGFMCHF2, and IGFMCHF3 when either MFT or MVT modules may be used. This is done to simplify documentation; MFT and MVT modules are mutually exclusive.

MVT SYSTEM ANALYSIS 1 (MODEL 145 ONLY)

Module ID: IGFMVTF1

Functions: MVT System Analysis 1 determines what error occurred and schedules the appropriate routine to handle it (one of the other system analysis modules or the PDAR Terminator). MVT System Analysis 1 also refreshes any intermittent storage errors¹ within an SVC or error transient area and then passes control to the PDAR Terminator.

Operation: MVT System Analysis 1 receives control from the Preliminary Error Analysis module. It gives control to one of the other system analysis modules or to the PDAR Terminator as follows:

- When the machine check is a CPU error (not a storage or SPF error) and I/O interruptions are disabled for the interrupted task, IGFMVTF1 indicates, in REPDAR3 and REPDAR6, that a wait state message must be issued. When I/O interruptions are enabled for the task interrupted by a CPU error, IGFMVTF1 indicates that the task must be terminated by ABEND. In both cases, IGFMVTF1 schedules the PDAR Terminator (IGFMCHF5) as the next routine.
- When the machine check is an intermittent main storage error (indicated in REPDAR2) in an SVC or an error transient area, IGFMVTF1 tries to restore the original data by loading a fresh copy of the affected module. Then, IGFMVTF1 indicates that the interrupted task must be terminated by ABEND and schedules the PDAR Terminator as the next routine. For all other intermittent storage errors, IGFMVTF1 schedules System Analysis 2 (IGFMVTF2) as the next routine.
- When the machine check is a solid main storage error or an SPF error (either solid or intermittent as indicated in REPDAR2), IGFMVTF1 schedules System Analysis 3 (IGFMVTF3) as the next routine.

MVT SYSTEM ANALYSIS 2 (MODEL 145 ONLY)

Module ID: IGFMVTF2

Functions: MVT System Analysis 2 determines which parts of the system are affected by an intermittent main storage error (multiple bit error) and takes appropriate action.

¹Although such errors are refreshed (repaired), the task affected by the error is still terminated.

Note: Single-bit main storage errors are corrected by hardware and merely require recording.

Operation: MVT System Analysis 2 determines from flags in the REPDAR field whether the intermittent main storage error occurred in a supervisor area (excluding system transient areas) or the Link Pack Area (excluding SVC and LINKLIB BLDL Tables). If so:

1. IGFMTVF2 indicates the system must be placed in a wait state and schedules the PDAR Terminator (IGFMCHF5) as the next routine when one of the following conditions also exists:
 - a. The interrupted task is disabled for I/O interruptions.
 - b. The failing storage location is marked cleared.
 - c. The instruction counter (at the time of the interruption) addresses a location within the nucleus or Link Pack Area.
2. IGFMTVF2 indicates the interrupted task must be terminated by the ABEND routine and schedules System Analysis (IGFMCHF6) as the next routine whenever none of the conditions listed under 1 exist.

When the intermittent main storage error occurred within the SVC BLDL Table, the LINKLIB BLDL Table, or both, IGFMTVF2 deletes the affected table(s) and:

1. Indicates the system is to be placed in a wait state and schedules the PDAR Terminator (IGFMCHF5) as the next routine whenever the interrupted task is disabled for I/O interruptions.
2. Indicates the interrupted task must be terminated by the ABEND routine and schedules Subsystem Analysis (IGFMCHF6) as the next routine whenever the interrupted task is enabled for I/O interruptions.¹

When the intermittent main storage error occurred within a system transient area, IGFMTVF2:

1. Indicates the system is to be placed in a wait state and schedules the PDAR

¹When an intermittent main storage error overlaps an SVC or LINKLIB BLDL Table and another supervisor location, IGFMTVF2 deletes the affected BLDL table but bases subsequent action on which other supervisor locations are affected.

Terminator (IGFMCHF5) as the next routine whenever the interrupted task is disabled for I/O interruptions.

2. Indicates the interrupted task must be terminated by the ABEND routine and schedules Subsystem Analysis (IGFMCHF6) as the next routine whenever the interrupted task is enabled for I/O interruptions.

MVT SYSTEM ANALYSIS 3 (MODEL 145 ONLY)

Module ID: IGFMTVF3

Functions: MVT System Analysis 3 determines which parts of the system are affected by SPF errors (solid or intermittent) or by solid main storage errors, and it takes the appropriate action.

Operation: For intermittent SPF key errors, MVT System Analysis 3 determines whether the error location associated with the failing key is within the Nucleus, Link Pack Area, or dynamic area subpool 252. If it is, IGFMTVF3 sets the failing key to zero using a Set Storage Key (SSK) instruction. If the error is in a dynamic area subpool other than 252, IGFMTVF3 sets the failing key to the key value in the TCB of the interrupted task. In either case, IGFMTVF3 then indicates the interrupted task must be terminated by ABEND and schedules Subsystem Analysis (IGFMCHF6) as the next routine.

For a solid SPF key error or a solid main storage error, IGFMTVF3 indicates the system must be placed in a wait state and schedules the PDAR Terminator (IGFMCHF5) as the next routine whenever:

- The storage location associated with the error is the Nucleus or Link Pack Area.
- The storage location associated with the error is in the dynamic area and the interrupted task is disabled for the I/O interruption.

If the error location associated with the solid error is in the dynamic area and the interrupted task is enabled for I/O interruptions, IGFMTVF3 sets the following nondispatchable:

- The TCB of the interrupted task.
- The associated jobstep TCB.
- All subtask TCBs of that jobstep.

IGFMTVF3 then indicates it took this action and schedules Subsystem Analysis (IGFMCHF6) as the next routine.

MFT SYSTEM ANALYSIS 1

Module ID: IGFMTF1

Functions: MFT System Analysis 1 determines what error occurred and schedules the appropriate routine to handle it (one of the other system analysis routines or the PDAR Terminator). It also:

1. Repairs intermittent SPF key errors.
2. Sets nondispatchable the TCB of the interrupted task (and the TCBs of all associated tasks when the system uses subtasking) when the machine check is a solid SPF key error or a solid main storage error within the dynamic area.

Operation: MFT System Analysis 1 receives control from the Preliminary Error Analysis module. It gives control to one of the other system analysis modules or to the PDAR Terminator as follows:

- When the machine check is a CPU error (not a storage or SPF error) and I/O interruptions are disabled for the interrupted task, IGFMTF1 indicates, in REPDAR3 and REPDAR6, that a wait state message must be issued. When I/O interruptions are enabled for the task interrupted by a CPU error, this routine indicates the task must be terminated by ABEND. In both cases, IGFMTF1 schedules the PDAR Terminator (IGFMCHF5) as the next routine.
- When the machine check is an intermittent main storage error (indicated in REPDAR2), and the interrupted task is disabled for I/O interruptions, and termination is necessary (indicated in REPDAR1), IGFMTF1 indicates the system must be placed in a wait state and schedules the PDAR Terminator (IGFMCHF5) as the next routine. For all other intermittent main storage errors:
 - a. When the error location is within the fixed area of the operating system, IGFMTF1 schedules MFT System Analysis 2 (IGFMFTF2) as the next routine.
 - b. When the error is within a dynamic area, IGFMTF1 indicates the interrupted task must be terminated by ABEND and schedules the PDAR Terminator (IGFMCHF5) as the next routine.
- When the machine check is an intermittent SPF key error, and the interrupted task is disabled for I/O interruptions, and termination is necessary (indicated in REPDAR1), IGFMTF1 indicates the

system must be placed in a wait state and schedules the PDAR Terminator (IGFMCHF5) as the next routine. For all other intermittent SPF key errors:

- a. When the error is within the fixed area, IGFMTF1 resets the failing key to 0.
- b. When the error is within the dynamic area, IGFMTF1 resets the failing key to the key value in the interrupted task's TCB.

After resetting the failing key, IGFMTF1 indicates that the interrupted task must be terminated by ABEND and schedules the PDAR Terminator (IGFMCHF5) as the next routine.

- When the machine check is a solid error, in main storage or the SPF key, and the error is within a fixed area, IGFMTF1 schedules IGFMTF2 as the next routine. When the solid error is within a dynamic area, IGFMTF1:
 - a. Indicates the system is to be placed in a wait state and schedules the PDAR Terminator (IGFMCHF5) as the next routine, whenever the interrupted task is disabled for I/O interruptions.
 - b. Whenever the interrupted task is enabled for I/O interruptions, it sets nondispatchable the TCB of the interrupted task and, if the system is subtasking, sets nondispatchable the TCBs of all tasks within the TCB chain headed by the jobstep TCB that includes the interrupted task. IGFMTF1 then indicates it took this action and schedules the PDAR Terminator (IGFMCHF5) as the next routine.

MFT SYSTEM ANALYSIS 2

Module ID: IGFMTF2

Functions: MFT System Analysis 2 schedules the appropriate termination procedures for solid main storage and solid SPF key errors within the fixed area. It also refreshes intermittent main storage errors within transient areas, deletes BLDL Tables containing intermittent storage errors, or schedules IGFMTF3 to handle intermittent storage errors in other locations.

Operation: MFT System Analysis 2 is entered by IGFMTF1 for machine checks in the supervisor (fixed area) that are either intermittent main storage errors or solid errors in either main storage or the SPF key.

When the machine check is a solid error, IGFMTF2 indicates the system is to be placed in a wait state and schedules the PDAR Terminator (IGFMCHF5) as the next routine.

When the machine check is an intermittent error within the SVC or an error transient area, IGFMTF2 tries to restore (refresh) the damaged module by loading a new copy. When this refresh attempt is successful, IGFMTF2 schedules the PDAR Terminator (IGFMCHF5) as the next routine after doing one of the following:

- Indicating the system must be placed in a wait state when the interrupted task is disabled for I/O interruptions.
- Indicating the interrupted task is to be terminated by ABEND when it is enabled for I/O interruptions.

When the refresh attempt is unsuccessful:

1. IGFMTF2 indicates the system must be placed in a wait state and schedules the PDAR Terminator (IGFMCHF5) as the next routine when one of the following conditions also exists:
 - a. The interrupted task is disabled for I/O interruptions.
 - b. The failing storage location is marked cleared.
 - c. The instruction counter (at the time of the interruption) addresses a location within the supervisor.
2. IGFMTF2 indicates the interrupted task must be terminated by ABEND and schedules the PDAR Terminator (IGFMCHF5) as the next routine whenever the conditions listed under 1 do not exist.

When the machine check is an intermittent main storage error within the SVC BLDL Table, the LINKLIB BLDL Table, or both, IGFMTF2 deletes the affected table(s) and:

1. Indicates the system is to be placed in a wait state when the interrupted task is disabled for I/O interruptions.
2. Indicates the interrupted task is to be terminated by ABEND when it is enabled for I/O interruptions.

In either case, IGFMTF2 schedules the PDAR

Terminator (IGFMCHF5) as the next routine.¹

When the machine check is an intermittent main storage error not mentioned above (that is, not within an SVC or error transient area, the SVC BLDL Table, or the LINKLIB BLDL Table), IGFMTF2 schedules MFT System Analysis 3 (IGFMFTF3) as the next routine.

MFT SYSTEM ANALYSIS 3

Module ID: IGFMTF3

Functions: MFT System Analysis 3 indicates the system is to be placed in a wait state and passes control to the PDAR Terminator (IGFMCHF5).

PDAR TERMINATOR

Module ID: IGFMCHF5

Functions: When system analysis routines request it, the PDAR Terminator does one of the following:

- Prepares the job step associated with the interrupted task for abnormal termination by ABEND.
- Prepares a Resume PSW that will put the system into a wait state.

In either case, and when system analysis has set the interrupted task nondispatchable, the PDAR Terminator also schedules the appropriate message.

Operation: The routines that call the PDAR Terminator (system analysis modules and, in MVT systems only, TSO recovery routines) indicate one of the following actions:

1. Terminate the interrupted task using ABEND.
2. The interrupted task and associated tasks already have been set nondispatchable.
3. Place the system in a wait state.
4. An entire TSO subsystem already has been terminated by ABEND or set nondispatchable (only possible on MVT systems when TSO is affected by the machine check).

¹When an intermittent main storage error overlaps an SVC or LINKLIB BLDL Table and another supervisor location, IGFMTF2 deletes the affected table but bases subsequent action on the other supervisor location affected.

5. A TSO user already has been terminated by ABEND or set nondispatchable (only possible on MVT systems when TSO is affected by the machine check).

When the request is to terminate using ABEND, IGFMCHE5 calls ABTERM, passing an ABEND code and the address of the interrupted task's TCB. ABTERM sets up the interrupted task for ABEND and returns with the address of the ABEND SVC in the Resume PSW field of the task's Request Block (RB). IGFMCHE5 moves this address to the MCH Resume PSW so that it will also be set up for ABEND when MCH returns control to the operating system.

When a task has been set nondispatchable, IGFMCHE5 moves the Resume PSW from the interrupted task's RB to the MCH Resume PSW so that, upon exit from MCH, the system Resume PSW will not be changed.

When the request is to place the system in a wait state, IGFMCHE5 moves a wait state Resume PSW into the MCH Resume PSW.

When TSO is affected, either required termination has already been done or a system analysis module has indicated that one of the previously described actions must be performed.

The PDAR Terminator always sets up a descriptive message in the message buffer MCHIBUF. This can consist of a message code only, a message code and associated text, or a message code, text, and a wait state code.

The PDAR Terminator always schedules the Soft Machine-Check Handler as the next routine.

TSO SUBSYSTEM ANALYSIS (MODEL 145 ONLY)

Module ID: IGFMCHE6

Functions: The PDAR Subsystem Analysis module is the interface between MCH and the Time Sharing Option (TSO) subsystem. This module receives control when an uncorrectable main storage error or uncorrectable SPF Key error occurs in the dynamic area.

Operation: Upon entry, Subsystem Analysis (IGFMCHE6) tests a flag in the Common Vector Table (CVT) to determine whether TSO is active. If it is not, IGFMCHE6 exits to the PDAR Terminator for normal termination processing.

If TSO is active, and the error is in the Local System Queue Area (LSQA), IGFMCHE6 schedules the system to be placed in a disabled wait state by setting a request bit in REPDAR and passing control to

IGFMCHE5 via the module loader. If the error is in the TSO Link Pack Area rather than the LSQA area, IGFMCHE6 stores the appropriate information in MCHSUB fields for the TSO Recovery Module.

If neither of the above areas is affected, IGFMCHE6 determines whether the error affects TSO in any other area by comparing the address of the error with TSO region addresses located in the TSO extent list queue. If the error does affect an area in TSO, IGFMCHE6 stores the appropriate information in the MCH Subsystem Common Area for the TSO Recovery routine (IGFMCHE91). Control is then passed to that routine via the Module Scheduler.

If TSO is not affected in any manner, IGFMCHE6 exits to the PDAR Terminator via the Module Scheduler.

ERROR RECORDER

Module ID: IGFMCHE2

Functions: The Error Recorder writes either a short or long record in the SYS1.LOGREC data set.

Operation: The Error Recorder scans the record buffers and, when an active buffer is located, determines whether the record in it is a short or a long record. Before doing any I/O, the recorder enqueues on the SYS1.LOGREC recording queue to ensure sole access while recording. The Error Recorder then checks the header record of the SYS1.LOGREC data set to ensure that recording is possible and, if so, it writes the error record into the data set. After one record has been written, the Error Recorder scans remaining buffers, repeating the process until all active short record buffers have been written out. Finally, the recorder enables the system for soft machine-check interruptions if that is the normal system status.

Note: For recording to be possible, a flag byte at the end of the SYS1.LOGREC header record must contain a X'FF'. If this flag byte contains any other value, the Disk Format Error Message is scheduled by turning on a bit in MCHINT, and control passes to IGFMCHE1.

The Error Recorder builds CCWs in the recorder portion of the MCH Common Area (MCHWORK). It issues an EXCP macro instruction to perform the I/O.

Note: The Error Recorder uses a common area to build CCWs as the transient area may be overlaid before the completion of the requested I/O.

When the SYS1.LOGREC data set is full, or when a disk format or I/O error occurs, this routine sets a flag in MCHINT to indicate these conditions to the Console Write routine (IGFMCHE1).

When all active records have been written, the Error Recorder passes control to the Console Write routine via XCTL.

CONSOLE WRITE ROUTINE

Module ID: IGFMCHE1

Functions: The MCH Console Write routine uses SVC 35 to write messages to the operator.

Operation: The Console Write routine scans MCHIBUF for a full SYS1.LOGREC I/O error message or disk format error message (if indicated in the MCHINT field). When either of these priority messages is found, the Console Write routine issues an SVC 35 (WTO routine) to write it. Then, the Console Write routine issues SVC 35 to write any remaining messages (such as, normal messages scheduled by other MCH routines).

This routine must handle two kinds of messages: preformatted and dynamic messages. A preformatted message remains the same each time it is issued. For this type of message, a code is placed in MCHIBUF by the routine requesting the message. The code is matched to its corresponding message in the message table contained in the Console Write routine. The Console Write routine issues an SVC 35 specifying the address of the message in the table to be issued.

A dynamic message is placed directly into the MCHIBUF by the routine requesting the message. Since a dynamic message contains unique data each time it is issued, it cannot reside in the Console Write message table. The address of the dynamic message in MCHIBUF is specified in register 1 each time the Console Write routine issues an SVC 35 for a dynamic message.

EMERGENCY RECORDER

Module ID: IGFMCHE3

Functions: The Emergency Recorder writes short MCH records, long MCH records, or a channel error record when either MCH or CCH determines that the system cannot continue. These records are written into the SYS1.LOGREC data set prior to system termination.

Operation: The Emergency Recorder performs the same tasks as the Error Recorder with the exception that it uses the Module Loader in the MCH Nucleus to write the record into the SYS1.LOGREC rather than using the operating system I/O facilities (that is, rather than passing control to the system via XCTL).

A channel error inboard record, constructed by CCH, is recorded using the address of the record passed by CCH. The address of the record is originally passed in register 13 and is stored in the MCH Common Area by IGFMCHE0.

MACHINE STATUS CONTROL (MODEL 145 ONLY)

Module ID: IGF29701

Functions: The Machine Status Control routine allows the operator to change the mode of recording soft machine-check interruptions for either main storage or control storage.

Operation: This routine analyzes the command parameters entered by the operator and places main storage or control storage into the appropriate mode, using a Diagnose instruction. The format of the MODE command is:

```
MODE {CNTR,RECORD/QUIET/THRES}
      {MAIN,RECORD/QUIET}
```

Note: The one combination which cannot be issued is MODE MAIN,THRES.

See "Modes of Recovery Operation" in Section 1 for descriptions of MCH recovery operation when control and main storage are in the different modes.

In all cases, the Machine Status Control routine writes message ICF061I to indicate that mode switching has been completed and returns control to the supervisor by issuing an SVC 3.

WARNING: This MODE command is intended for use only by IBM personnel or at their request. Use of this MODE command can degrade system performance.

MACHINE STATUS CONTROL (MODEL 135 ONLY)

Module ID: IGF13501

Functions: The Machine Status Control routine allows the operator to change the mode of recording soft machine-check interruptions for either storage or CPU errors from quiet mode to recording mode. It also allows the operator to display the current mode status.

Operation: This routine analyzes the MODE command parameters entered by the operator and places main storage or control storage into recording mode using a Diagnose instruction (or it displays mode status). The format of the MODE command is:

```
MODE { STATUS
      { HIR RECORD
      { ECC RECORD }
```

This routine is entered from the Mode Command Router module, IGF2603D, when the MODE command is specified by the operator.

See "Use of the Mode Commands" in Section 1 for descriptions of MCH recovery operation involving the different modes.

When mode switching is requested, this routine issues message IGF061I to indicate that the mode switch is complete. When mode status is requested, this routine issues message IGF053I, which has the following format:

```
MODE STATUS-ECC { RECORD } HIR { RECORD }
                { QUIET }      { QUIET }
COUNT-nn THRESHOLD-nn
```

When a switch to ECC record mode is requested while CPU retry is still in quiet mode, it is considered an error, and message IEE305I is issued.

In all cases, the Machine Status Control routine returns to the supervisor by issuing an SVC 3.

FLOWCHARTS

The flowcharts are arranged in the order in which MCH modules are described in this section. The chart heading for each set of flowcharts corresponds to a module ID.

Subroutine blocks contain the name of the subroutine on the first line of the block. If the subroutine is flowcharted in this manual, its entry and page and block number are to the right and above the subroutine block.

Some of the code represented in these charts should not be executed and invalid logic paths are indicated by notes.

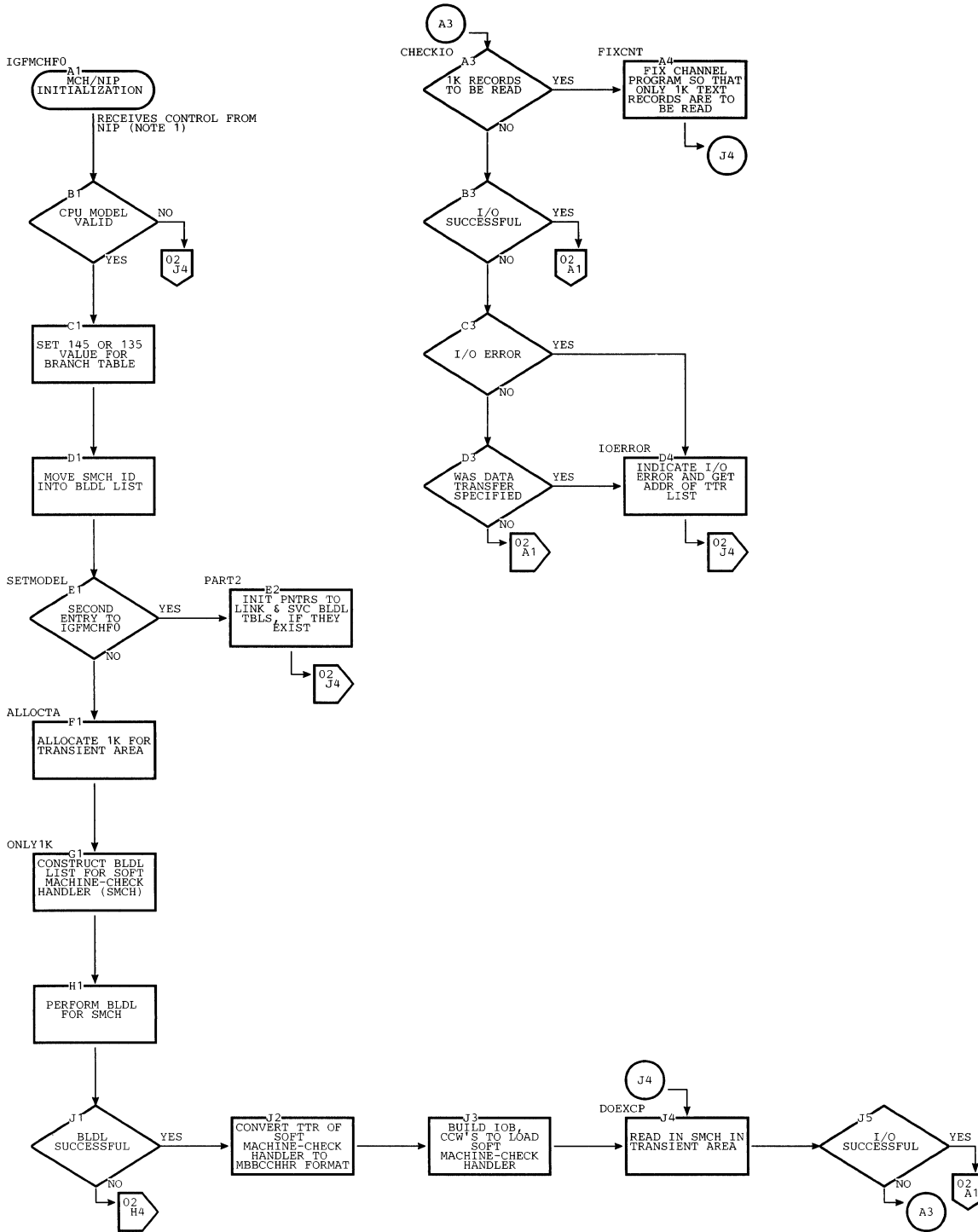
Note: The following terms appear inside some decision blocks and are defined here:

TERM NECESSARY -- Either system or task termination has been requested by IGMCH41 by setting REPDAR1, bit 0, to 1 (see Section 4, "MCH Independent Common Area").

INHIBIT TERMINATION ON or REQUESTED -- The "NO" branch must always be taken.

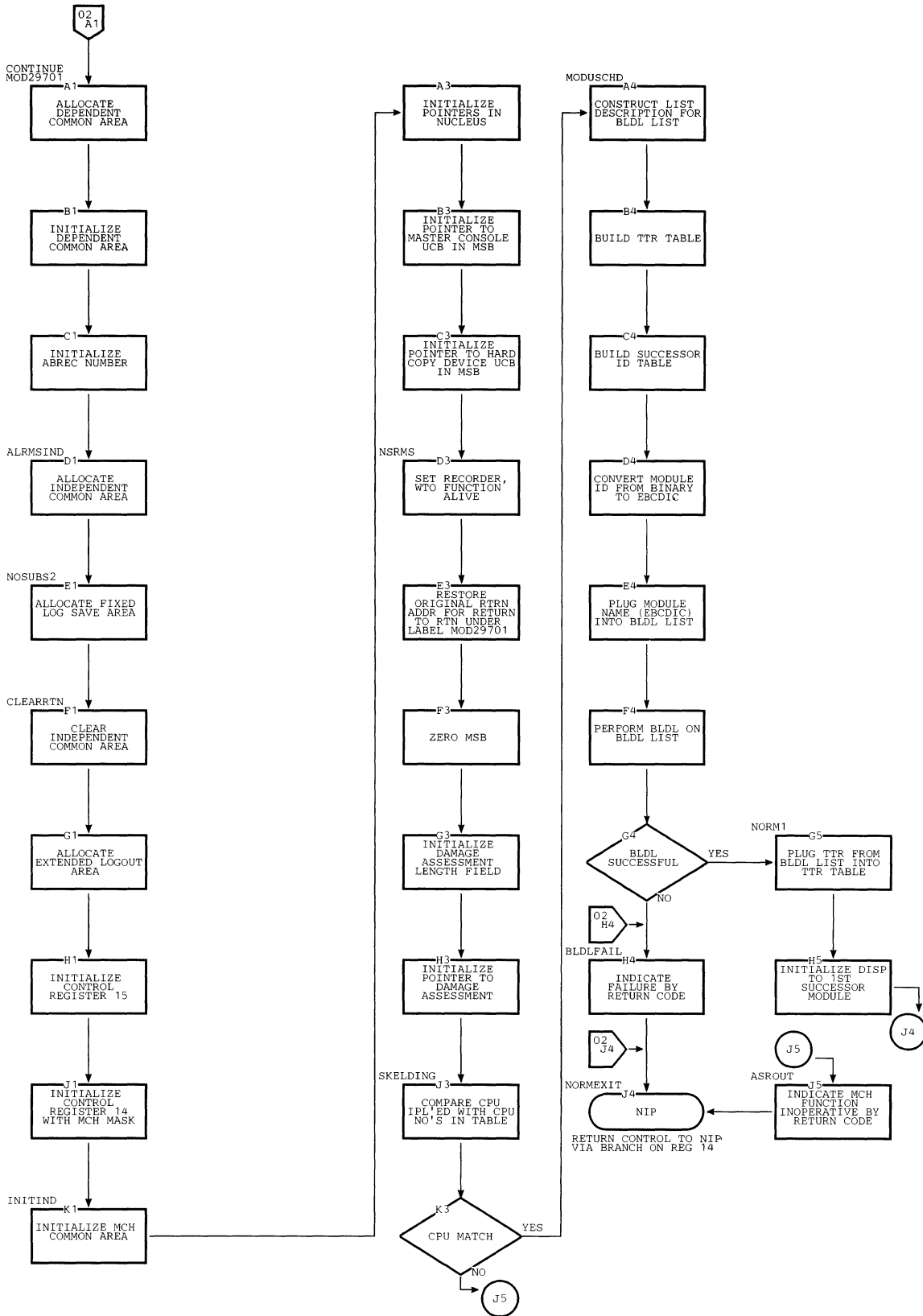
RETRY ON -- The "NO" branch must always be taken.

IGFMCHF0. MCH Initialization (Part 1 of 2)



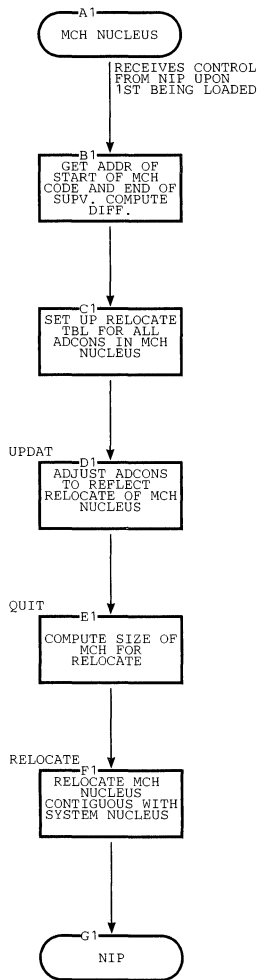
NOTE 1 : THIS FLOW CHART REFLECTS ONLY THE LOGIC APPLICABLE TO THE MODELS 135 AND 145

IGFMCHF0. MCH Initialization (Part 2 of 2)

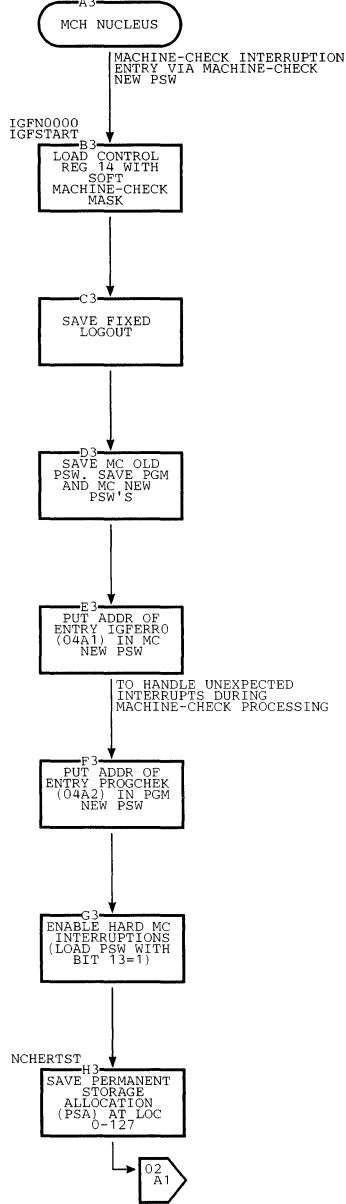


IGFMCHE0. MCH Nucleus (Part 1 of 9)

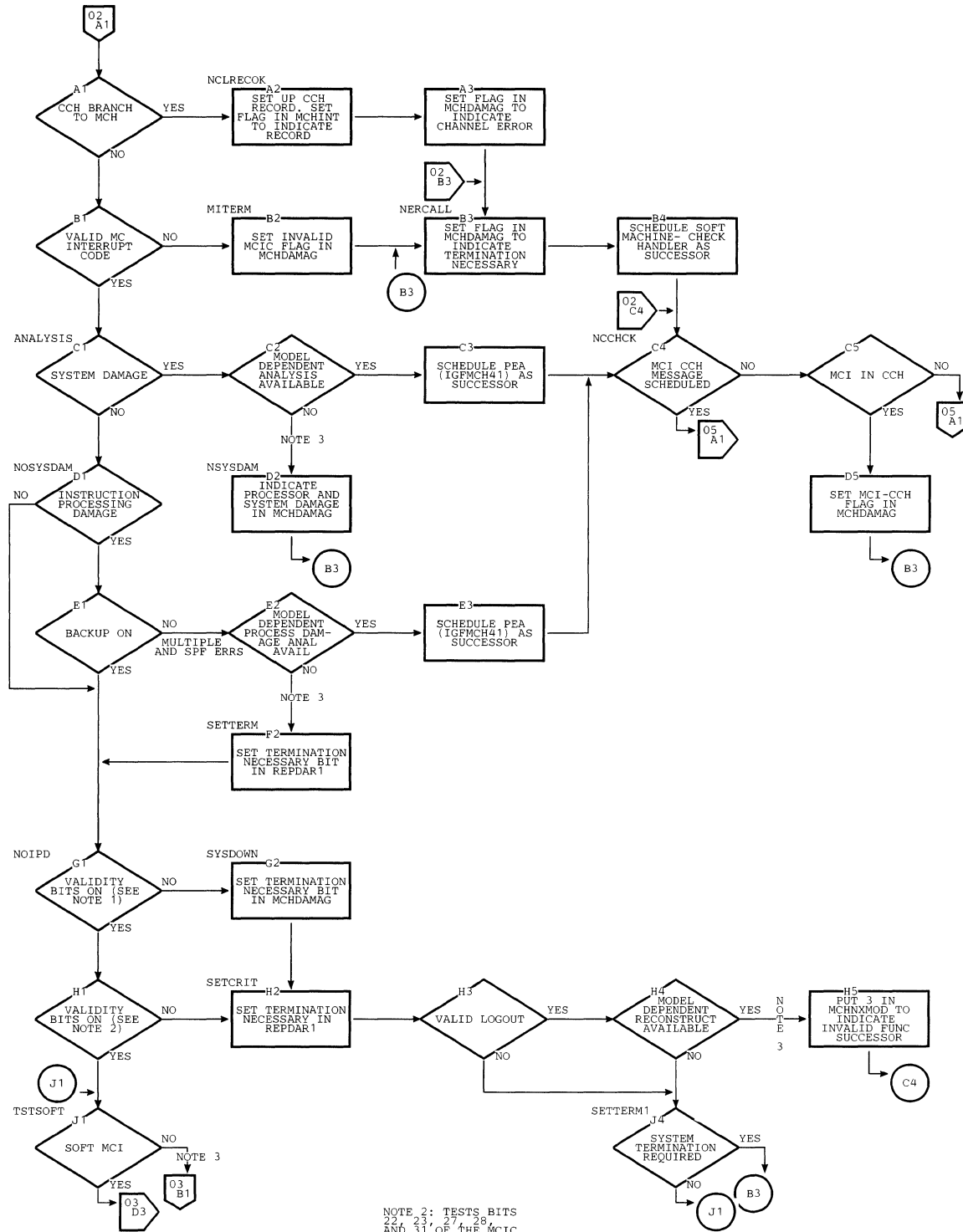
IGFMCHE0



IGFMCHE0



IGFMCHE0. MCH Nucleus (Part 2 of 9)

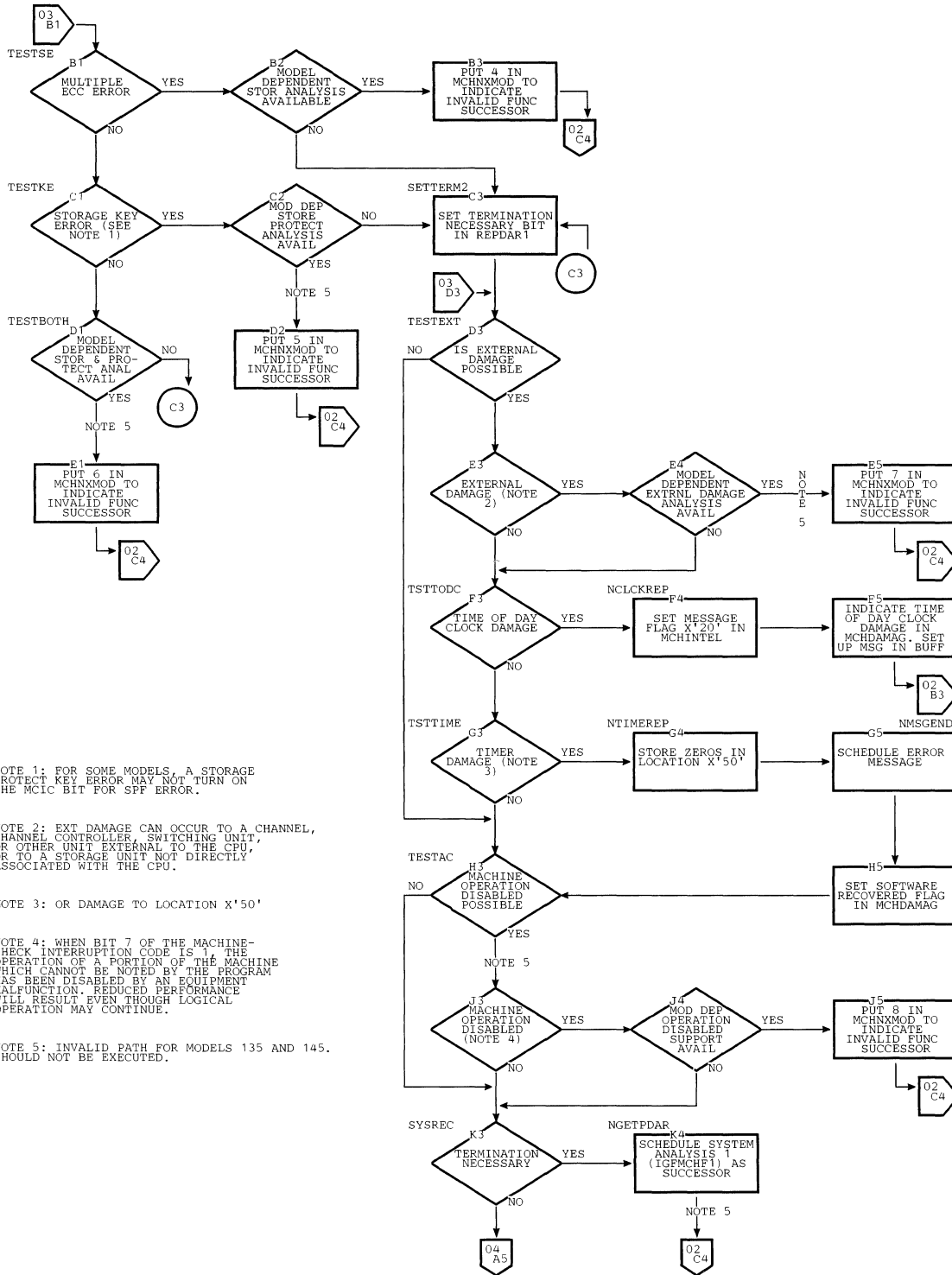


NOTE 1: BLOCK G1 TESTS VALIDITY BITS AT BITS 20, 21, AND 29 OF THE MACHINE-CHECK INTERRUPTION CODE. IF ALL THESE BITS ARE ON; THE AMWP, SYSTEM MASK, AND KEY (OF THE OLD PSW) ARE VALID AND THE CONTROL REGISTER SAVE AREA ACCURATELY REFLECTS THE CONDITION OF THE CONTROL REGS AT THE TIME OF THE INTERRUPTION.

NOTE 2: TESTS BITS 27, 23, 27, 28 AND 31 OF THE MCIC. IF ALL ARE ON; THE PROGRAM MASK, CONDITION CODE, AND INSTRUCTION ADDRESS OF THE OLD PSW ARE VALID; AND THAT THE FLOATING POINT AND GENERAL REG SAVE AREAS ACCURATELY REFLECT REG CONTENTS AT THE INTERRUPTION; AND THAT THOSE STORAGE LOCATIONS THAT ARE MODIFIED BY THE INSTRUCTION PROCESSING STREAM CONTAIN CORRECT INFO RELATIVE TO THE POINT OF THE INTERRUPTION.

NOTE 3: INVALID PATH FOR MODELS 135 AND 145. SHOULD NOT BE EXECUTED.

IGFMCHE0. MCH Nucleus (Part 3 of 9)



NOTE 1: FOR SOME MODELS, A STORAGE PROTECT KEY ERROR MAY NOT TURN ON THE MCIC BIT FOR SPF ERROR.

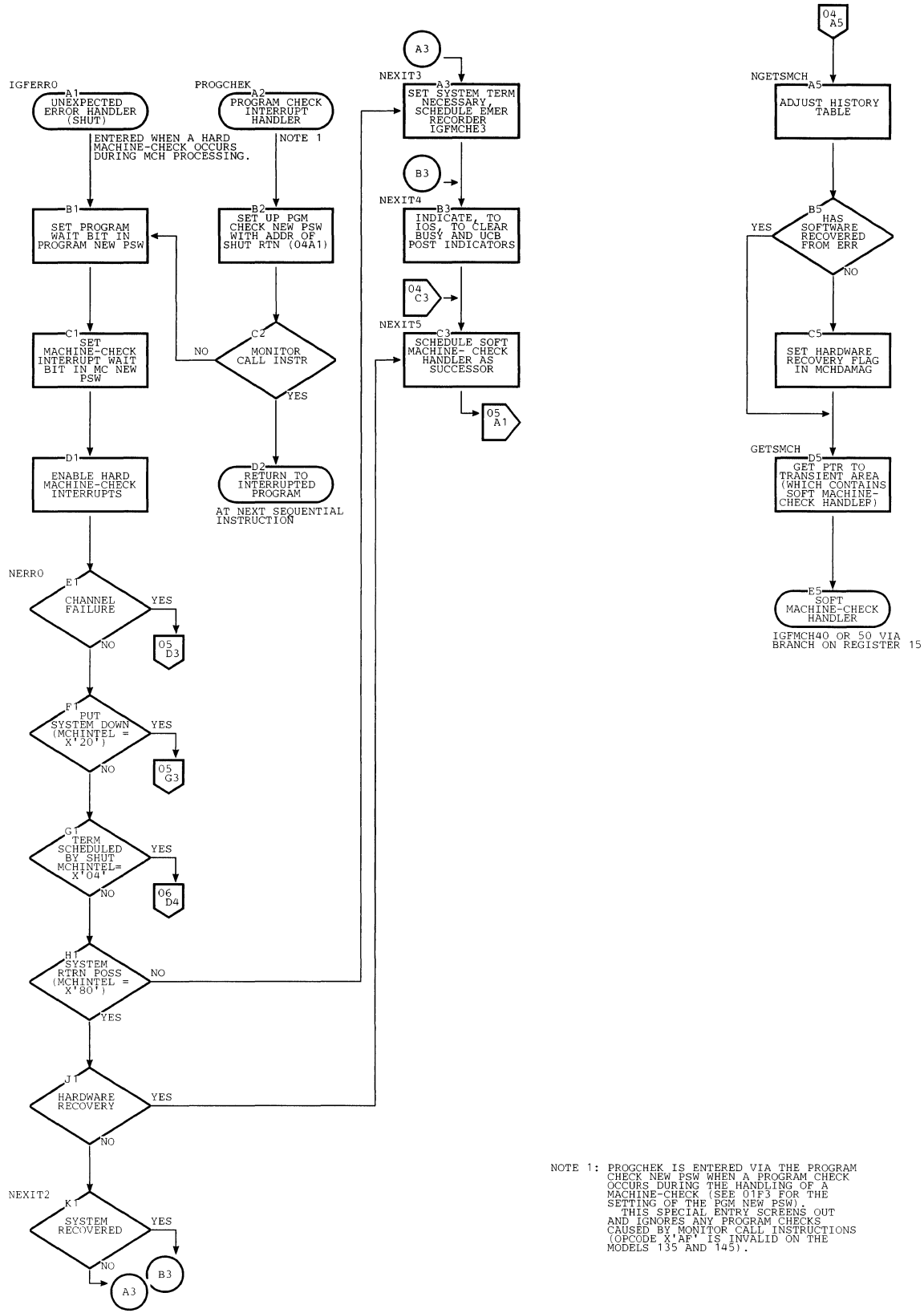
NOTE 2: EXT DAMAGE CAN OCCUR TO A CHANNEL, CHANNEL CONTROLLER, SWITCHING UNIT, OR OTHER UNIT EXTERNAL TO THE CPU, OR TO A STORAGE UNIT NOT DIRECTLY ASSOCIATED WITH THE CPU.

NOTE 3: OR DAMAGE TO LOCATION X'50'

NOTE 4: WHEN BIT 7 OF THE MACHINE-CHECK INTERRUPTION CODE IS 1, THE OPERATION OF A PORTION OF THE MACHINE WHICH CANNOT BE NOTED BY THE PROGRAM HAS BEEN DISABLED BY AN EQUIPMENT MALFUNCTION. REDUCED PERFORMANCE WILL RESULT EVEN THOUGH LOGICAL OPERATION MAY CONTINUE.

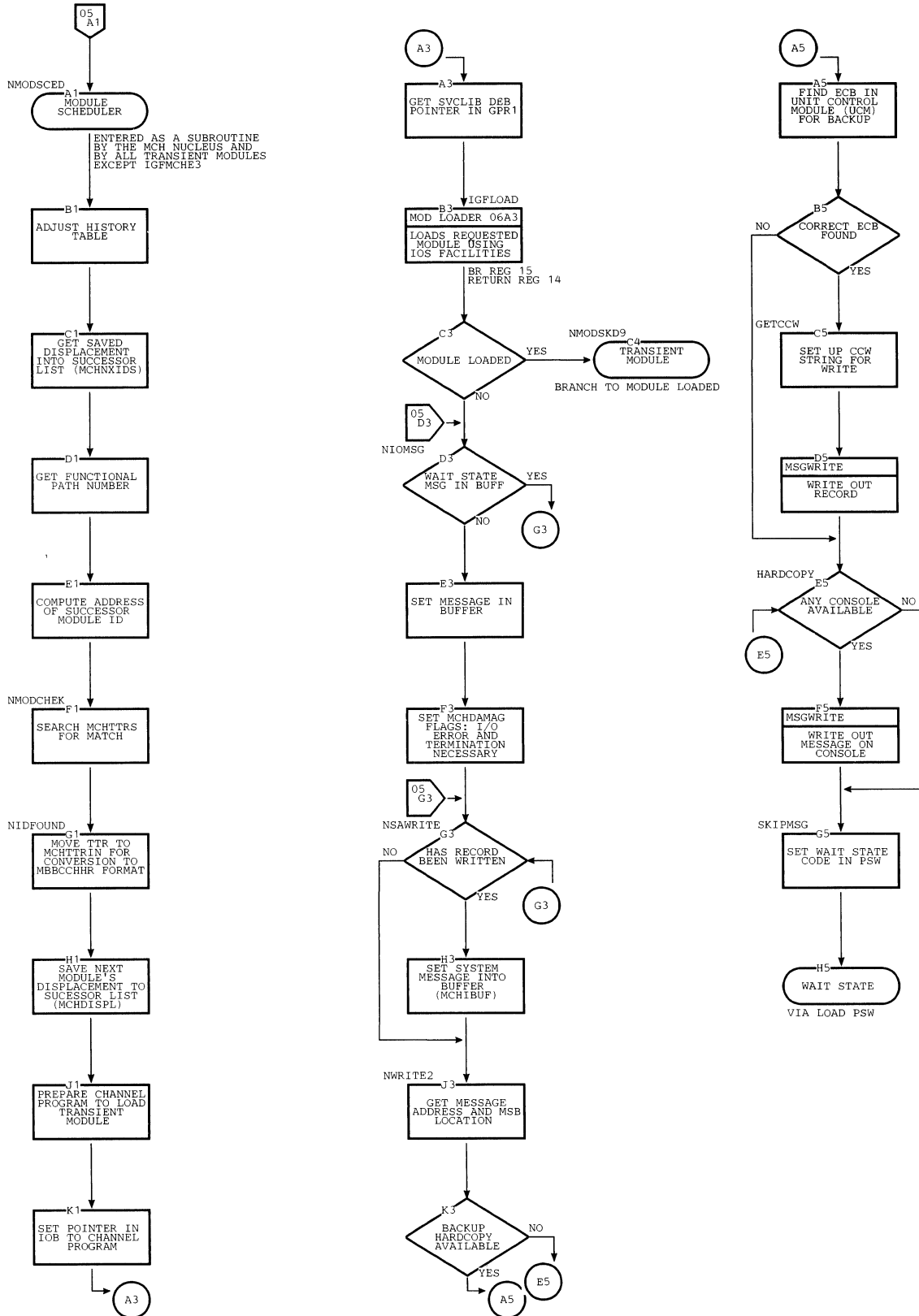
NOTE 5: INVALID PATH FOR MODELS 135 AND 145. SHOULD NOT BE EXECUTED.

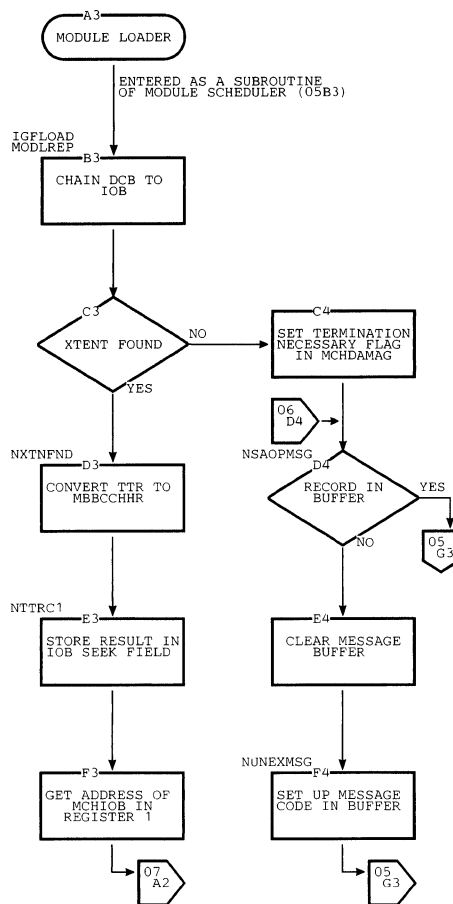
IGFMCHE0. MCH Nucleus (Part 4 of 9)



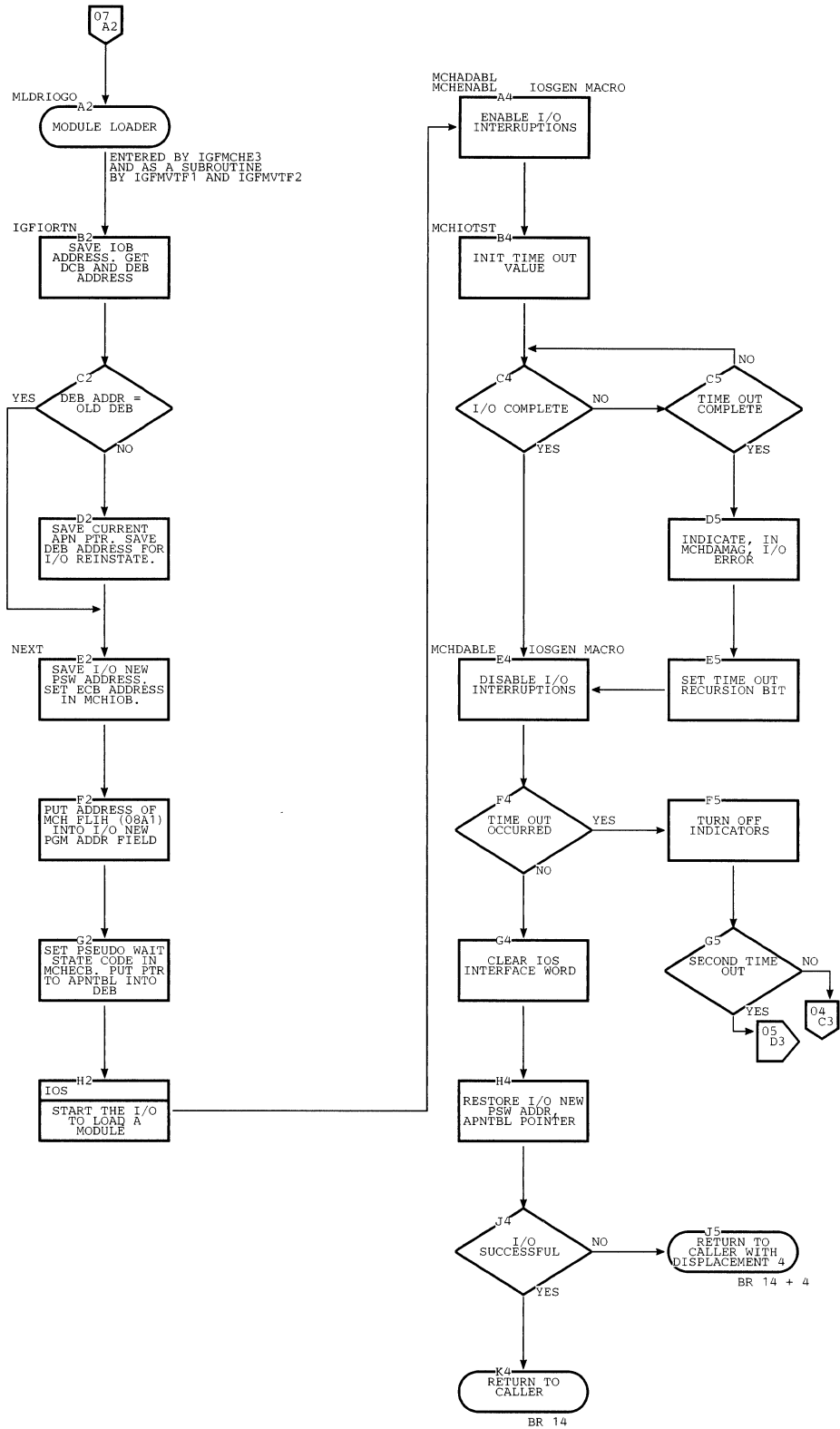
NOTE 1: PROGCHK IS ENTERED VIA THE PROGRAM CHECK NEW PSW WHEN A PROGRAM CHECK OCCURS DURING THE HANDLING OF A MACHINE-CHECK (SEE 01F3 FOR THE SETTING OF THE PGM NEW PSW). THIS SPECIAL ENTRY SCREENS OUT AND IGNORES ANY PROGRAM CHECKS CAUSED BY MONITOR CALL INSTRUCTIONS (OPCODE X'AF' IS INVALID ON THE MODELS 135 AND 145).

IGFMCHE0. MCH Nucleus (Part 5 of 9)

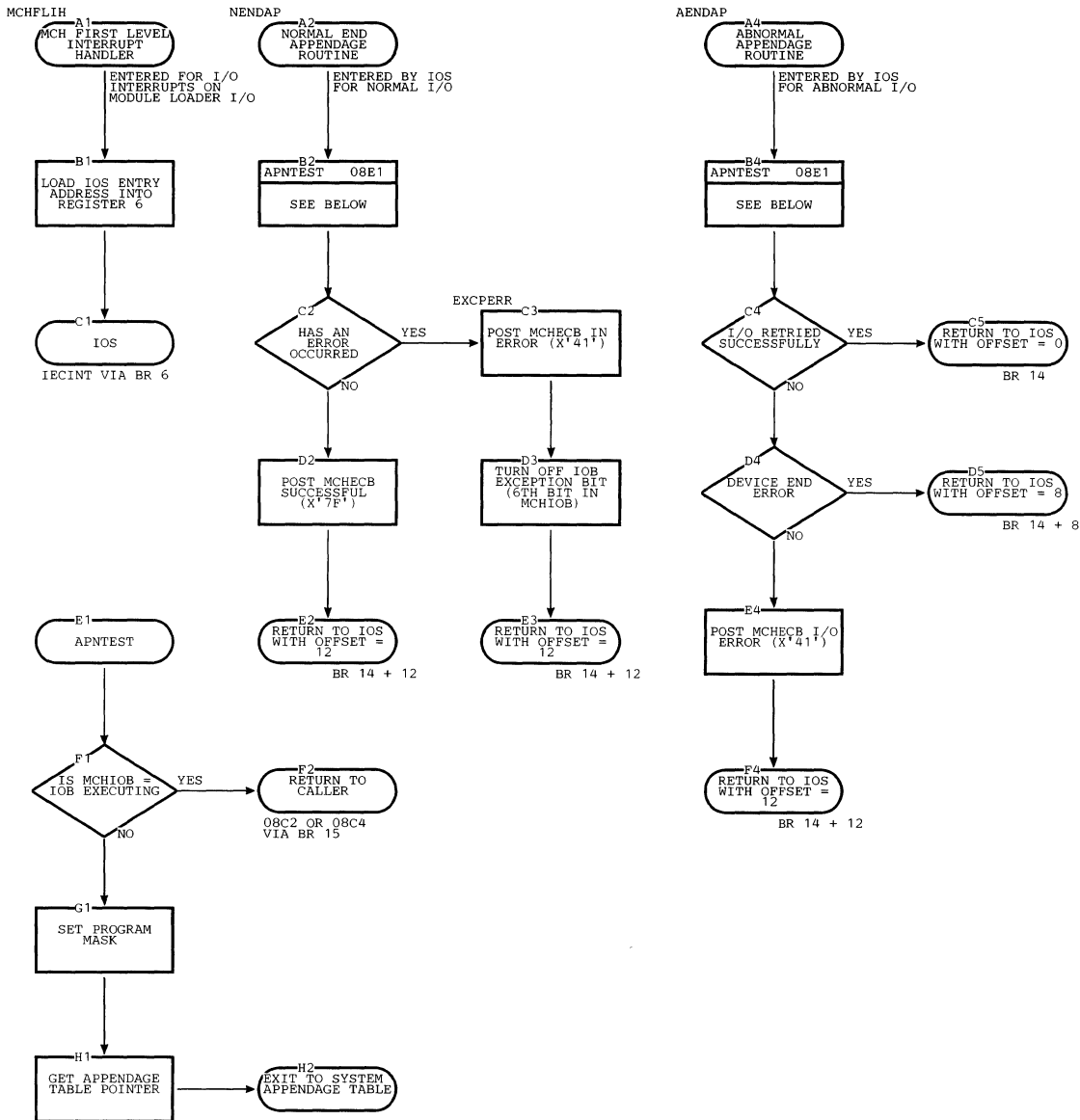




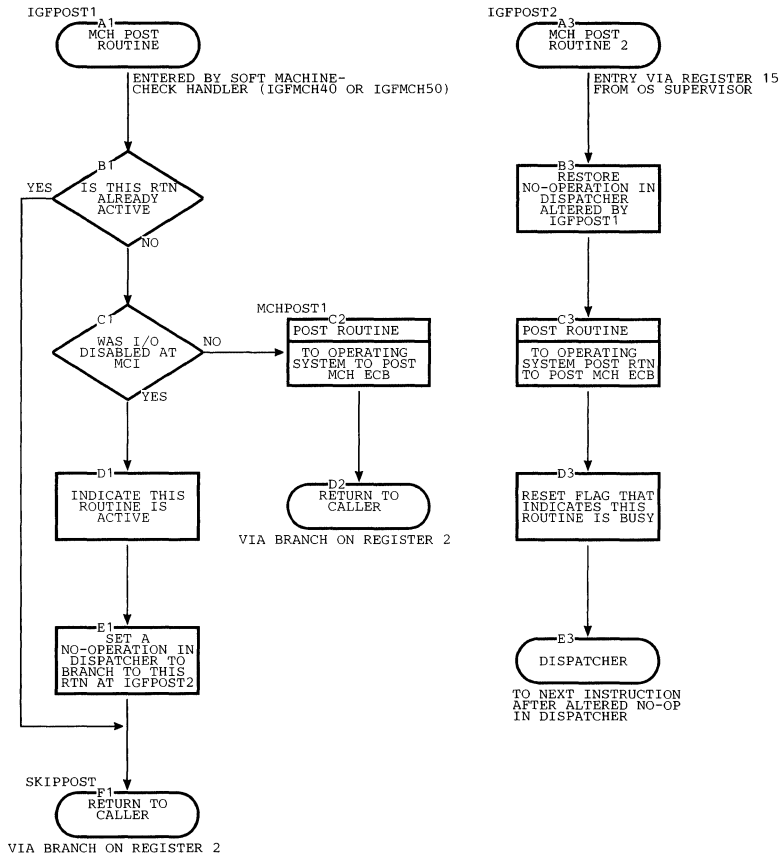
IGFMCHE0. MCH Nucleus (Part 7 of 9)



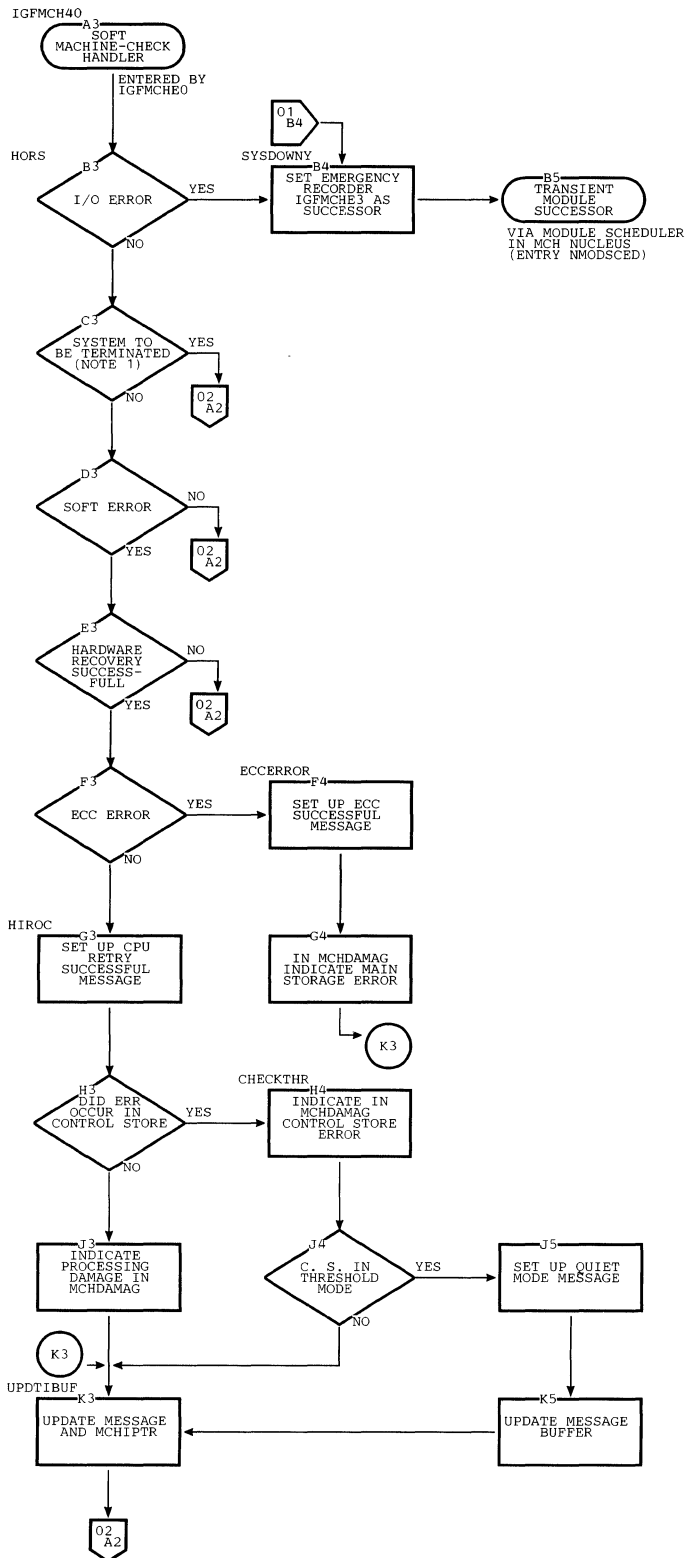
IGFMCHE0. MCH Nucleus (Part 8 of 9)



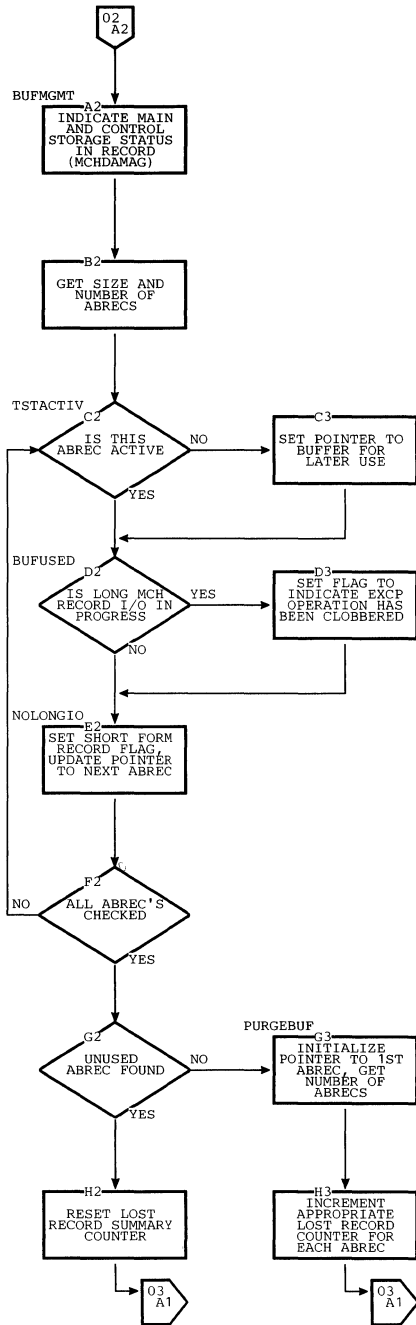
IGFMCH0. MCH Nucleus (Part 9 of 9)



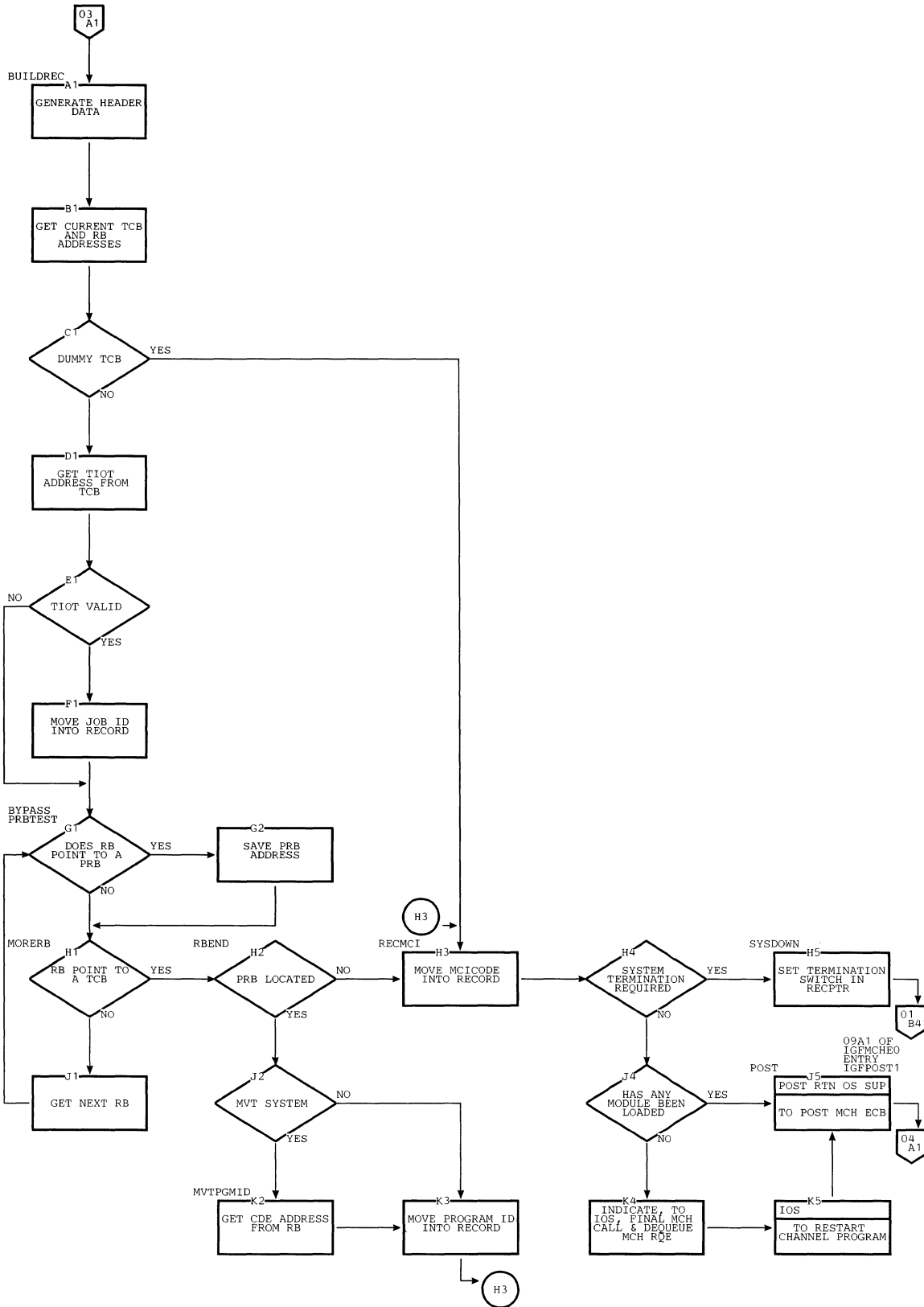
IGFMCH40. Model 145 Soft Machine-Check Handler (Part 1 of 4)



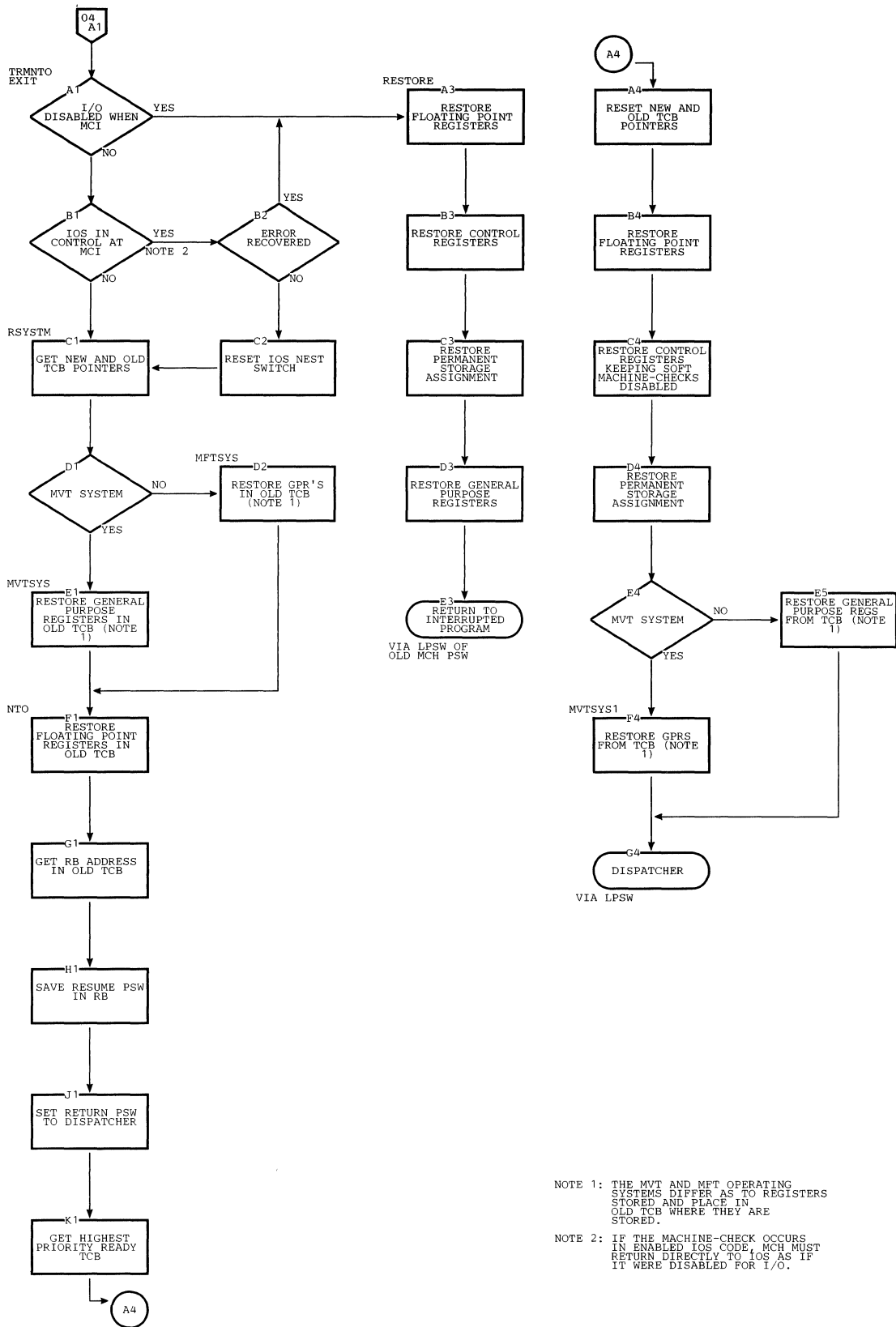
IGFMCH40. Model 145 Soft Machine-Check Handler (Part 2 of 4)



IGFMCH40. Model 145 Soft Machine-Check Handler (Part 3 of 4)



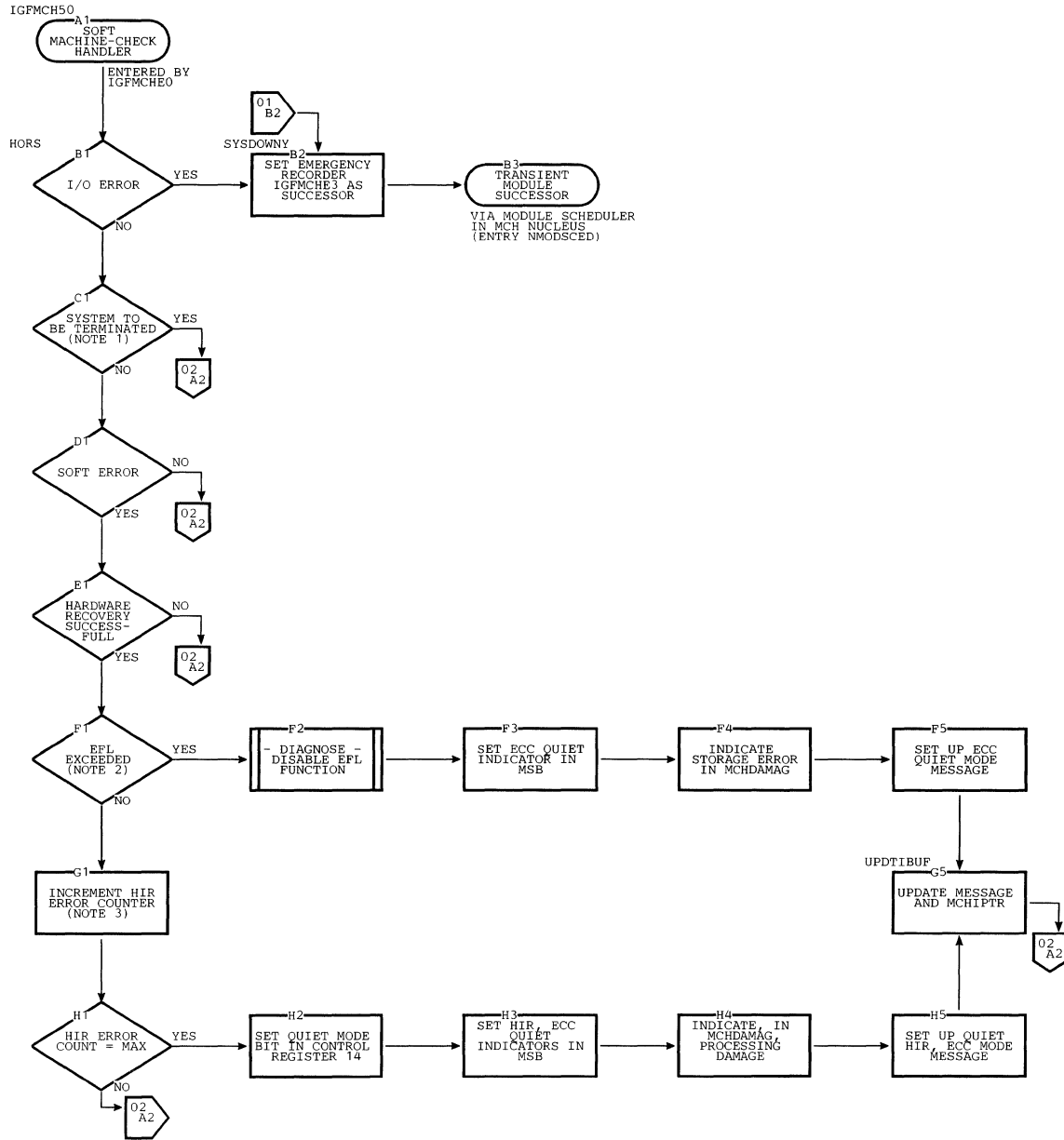
IGFMCH40. Model 145 Soft Machine-Check Handler (Part 4 of 4)



NOTE 1: THE MVT AND MFT OPERATING SYSTEMS DIFFER AS TO REGISTERS STORED AND PLACE IN OLD TCB WHERE THEY ARE STORED.

NOTE 2: IF THE MACHINE-CHECK OCCURS IN ENABLED IOS CODE, MCH MUST RETURN DIRECTLY TO IOS AS IF IT WERE DISABLED FOR I/O.

IGFMCH50. Model 135 Soft Machine-Check Handler (Page 1 of 4)

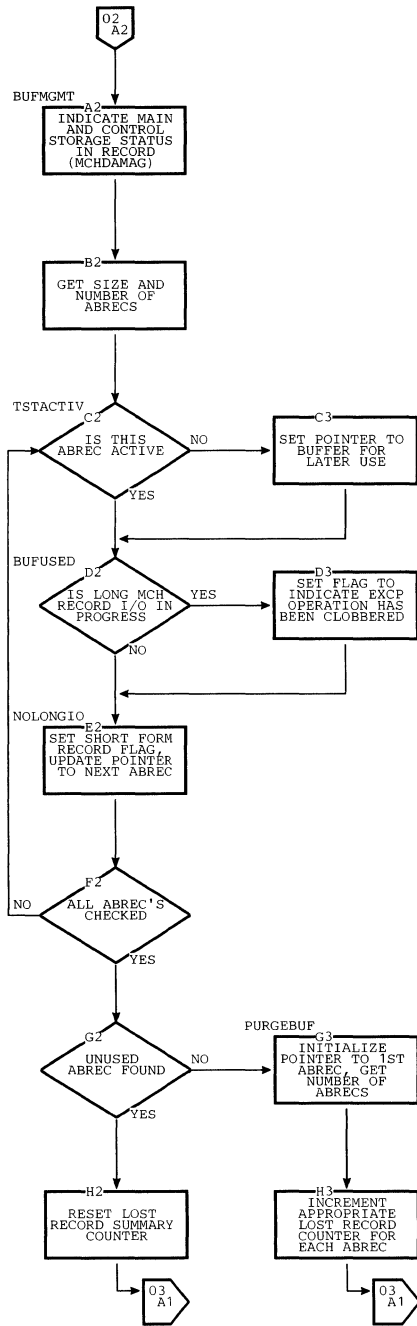


NOTE 2: EFL IS AN ERROR FREQUENCY LIMIT BUILT INTO THE HARDWARE THAT ISSUES A MACHINE CHECK WHEN SOLID SINGLE-BIT ERRORS REACH THE RATE OF 256 SINGLE-BIT ERRORS IN 416 MICRO-SECONDS.

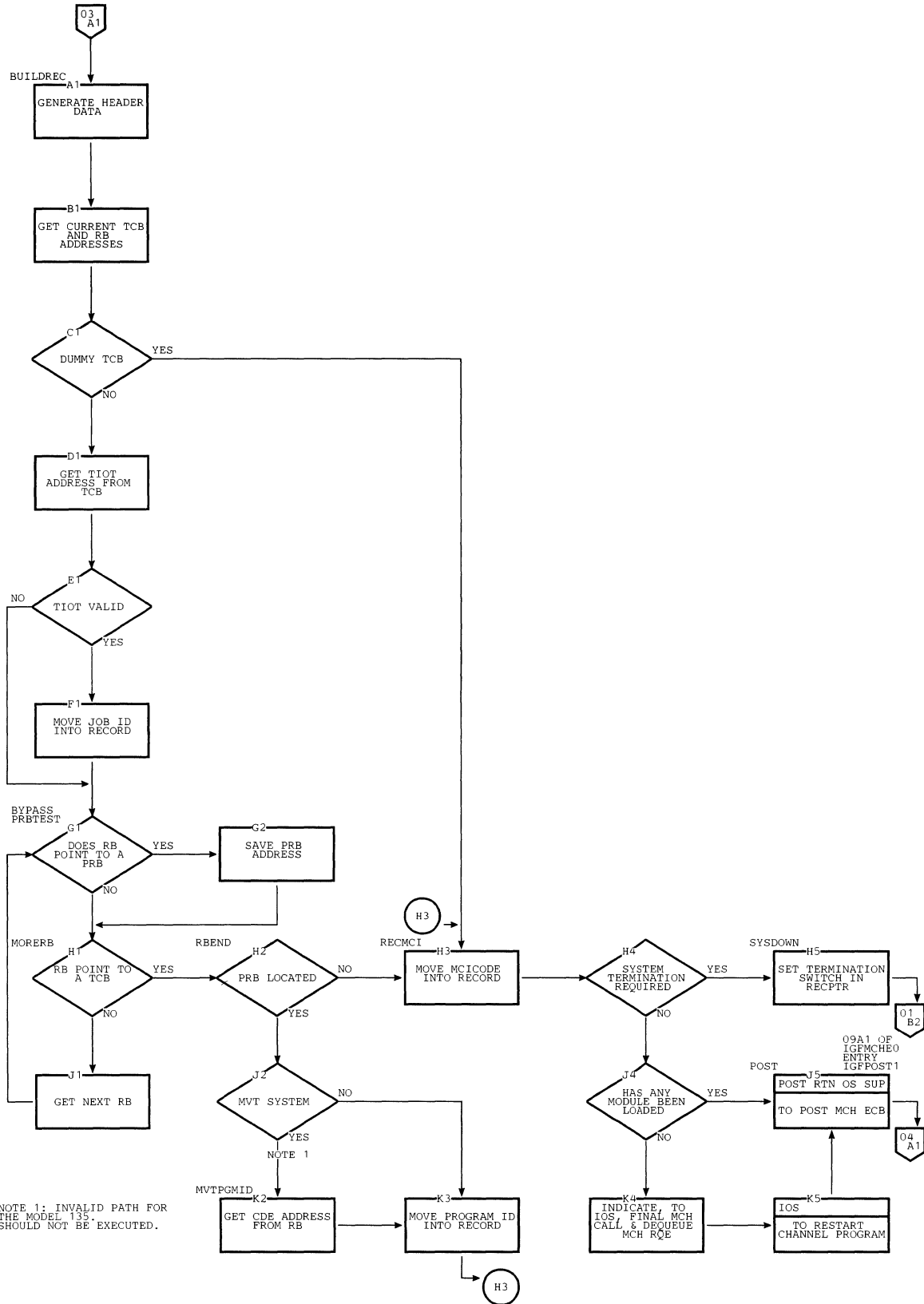
NOTE 1: THIS BLOCK TESTS A BIT IN MCHDAMAG RATHER THAN A BIT IN REPDAR1 (SEE INTRODUCTION TO THESE FLOWCHARTS FOR THIS USE OF REPDAR1). WHEN THE MCHDAMAG BIT INDICATING TERMINATION IS ON, THE SYSTEM (AND NOT JUST A PROBLEM PROGRAM) MUST BE TERMINATED.

NOTE 3: HIR (HARDWARE INSTRUCTION RETRY?) IS THE SAME AS CPU RETRY.

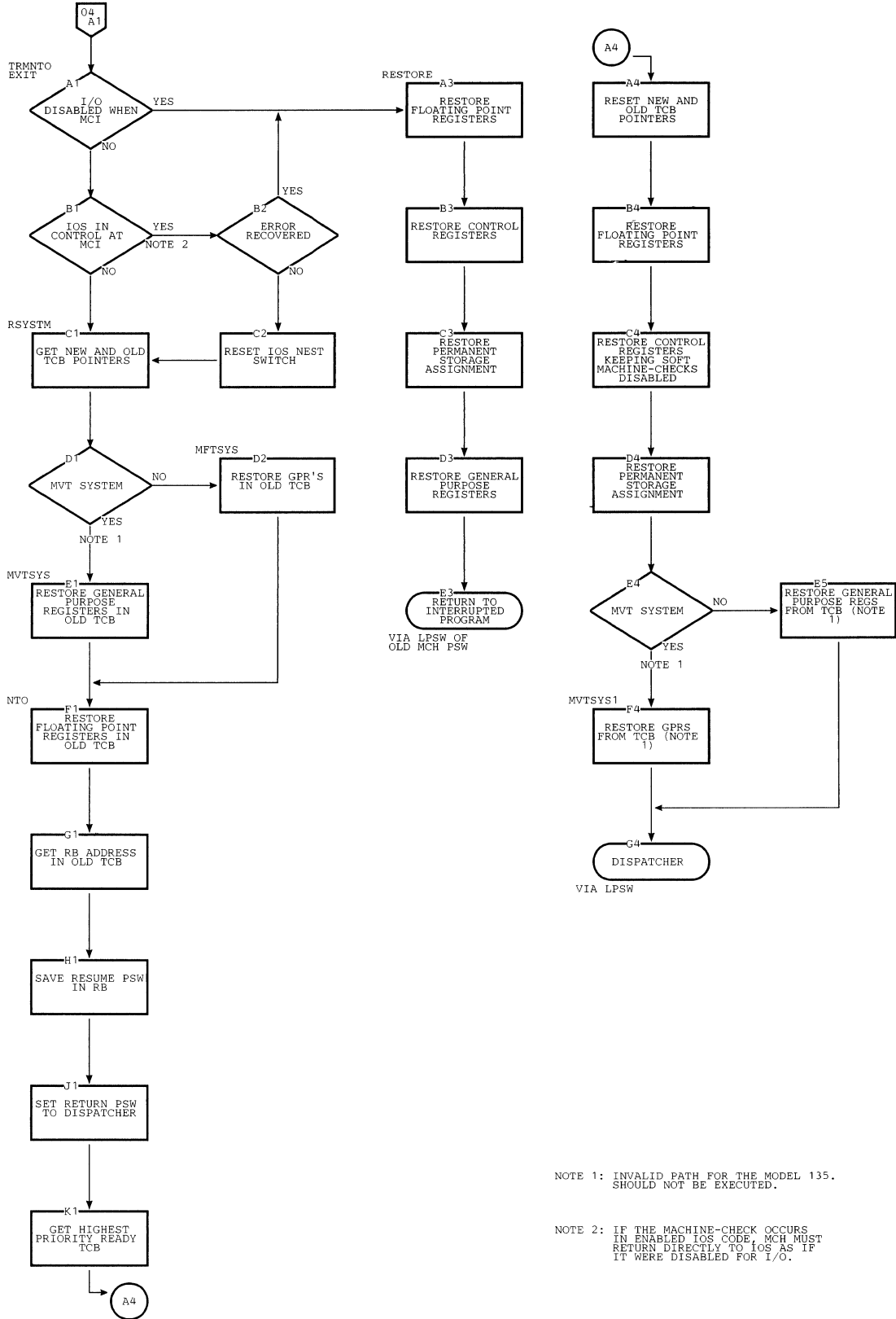
IGFMCH50. Model 135 Soft Machine-Check Handler (Part 2 of 4)



IGFMCH50. Model 135 Soft Machine-Check Handler (Part 3 of 4)



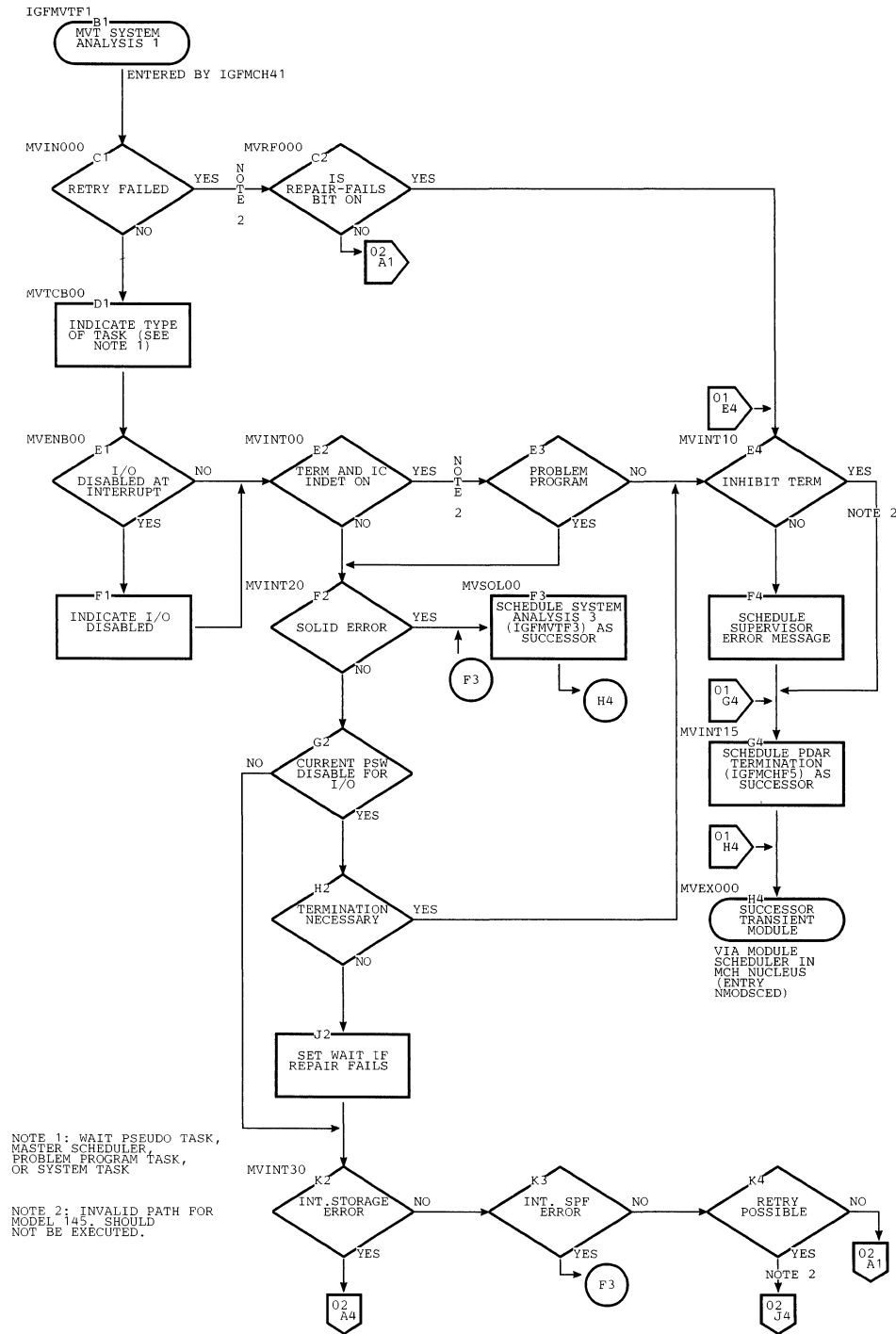
IGFMCH50. Model 135 Soft Machine-Check Handler (Part 4 of 4)



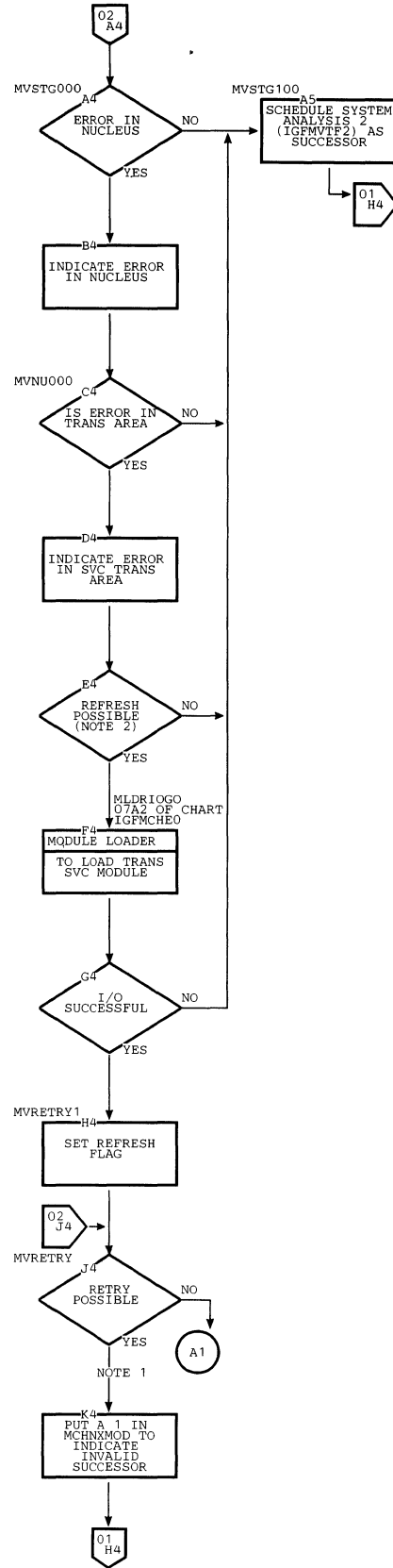
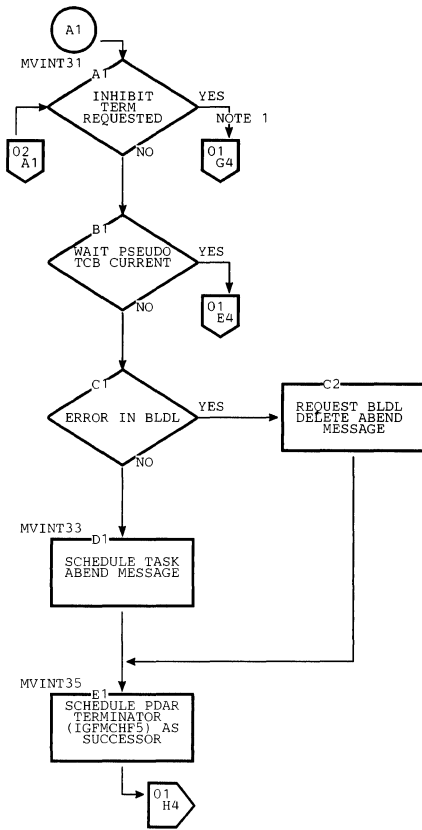
NOTE 1: INVALID PATH FOR THE MODEL 135. SHOULD NOT BE EXECUTED.

NOTE 2: IF THE MACHINE-CHECK OCCURS IN ENABLED IOS CODE, MCH MUST RETURN DIRECTLY TO IOS AS IF IT WERE DISABLED FOR I/O.

IGFMVTF1. MVT System Analysis 1 (Part 1 of 2)

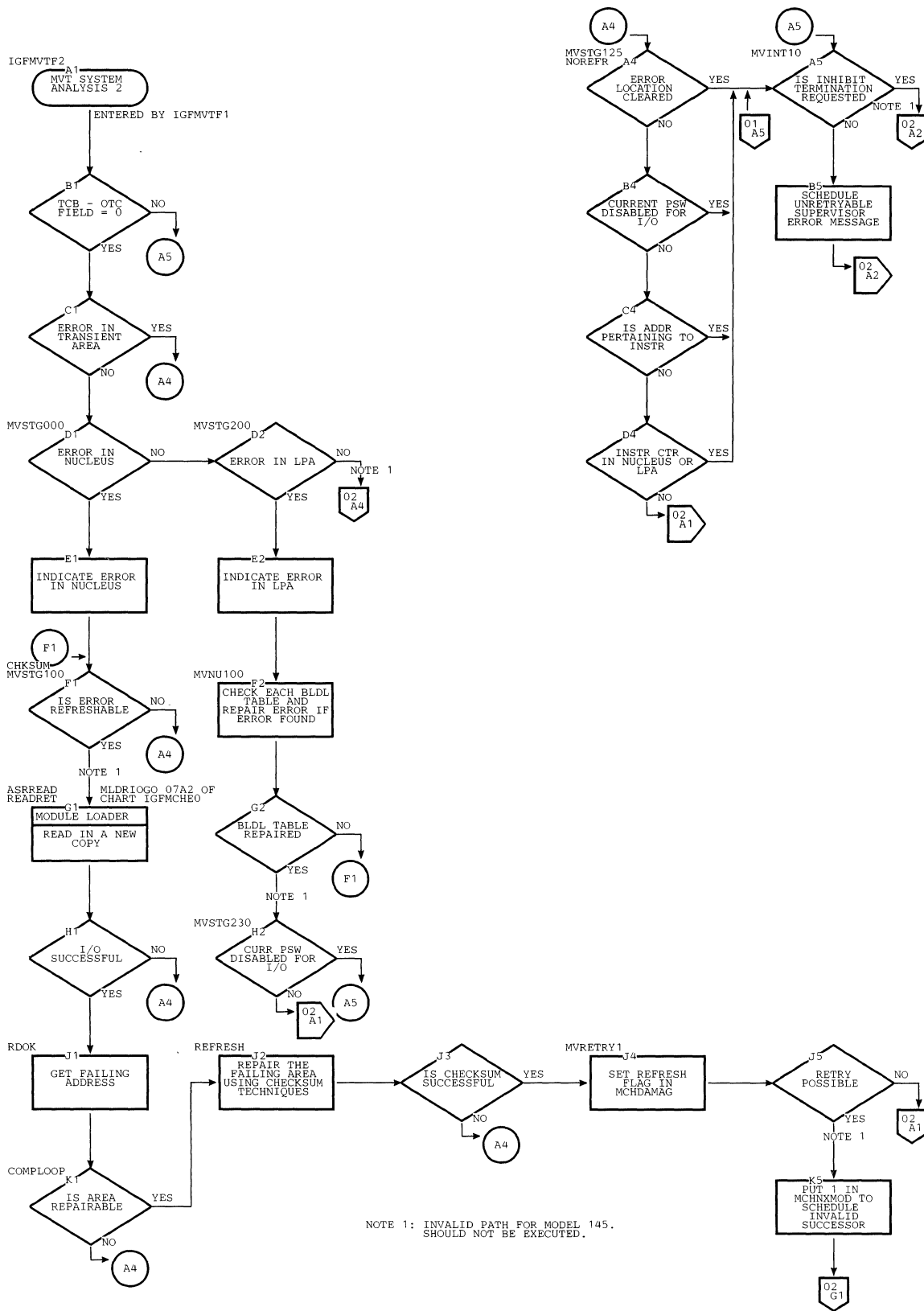


IGFMVTF1. MVT System Analysis 1 (Part 2 of 2)

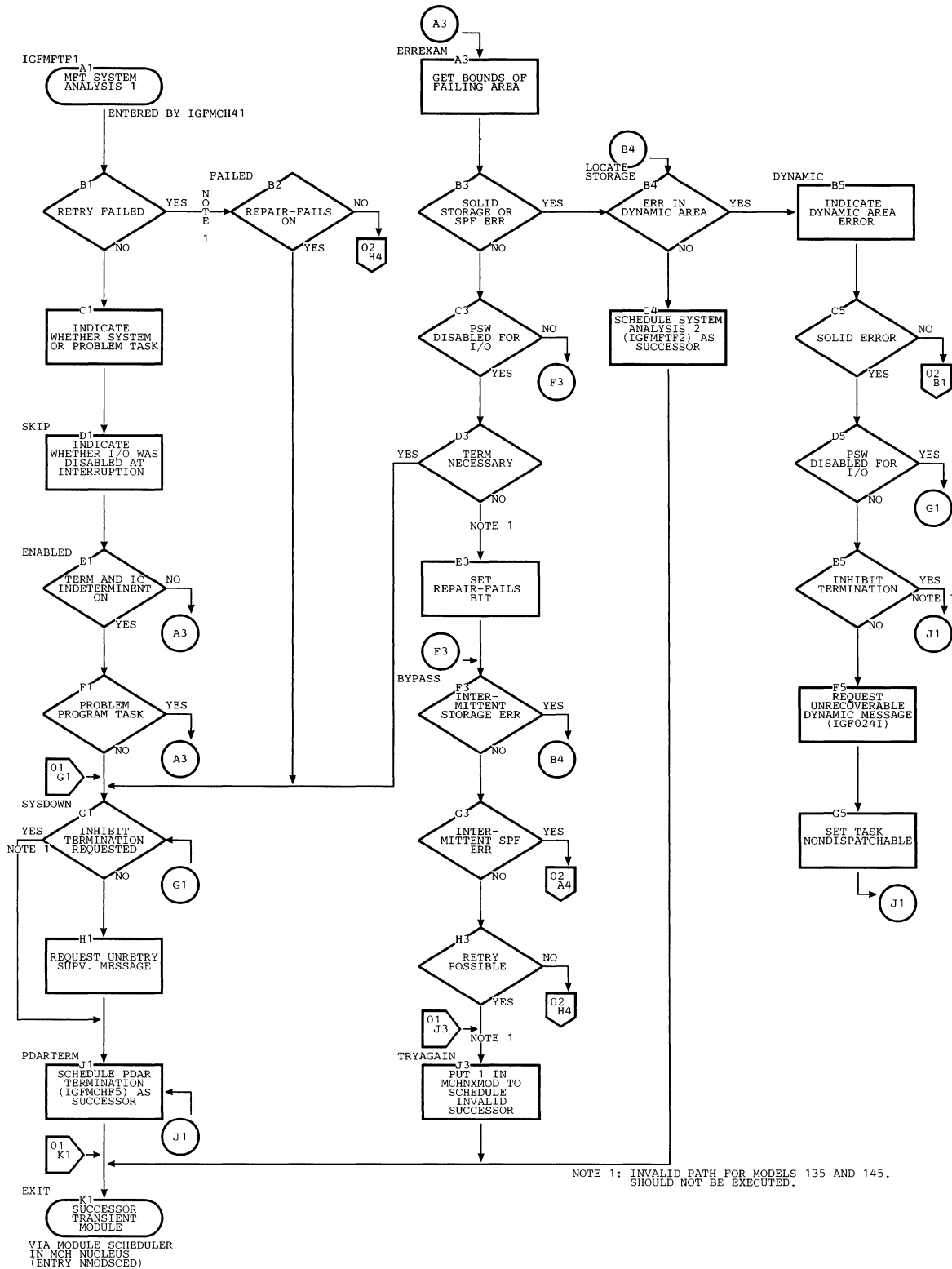


NOTE 1: INVALID PATH FOR MODEL 145. SHOULD NOT BE EXECUTED.
 NOTE 2: SVC TRANSIENT AREA IS REFRESHED WHEN POSSIBLE, BUT THE AFFECTED TASK IS STILL TERMINATED BY MCH.

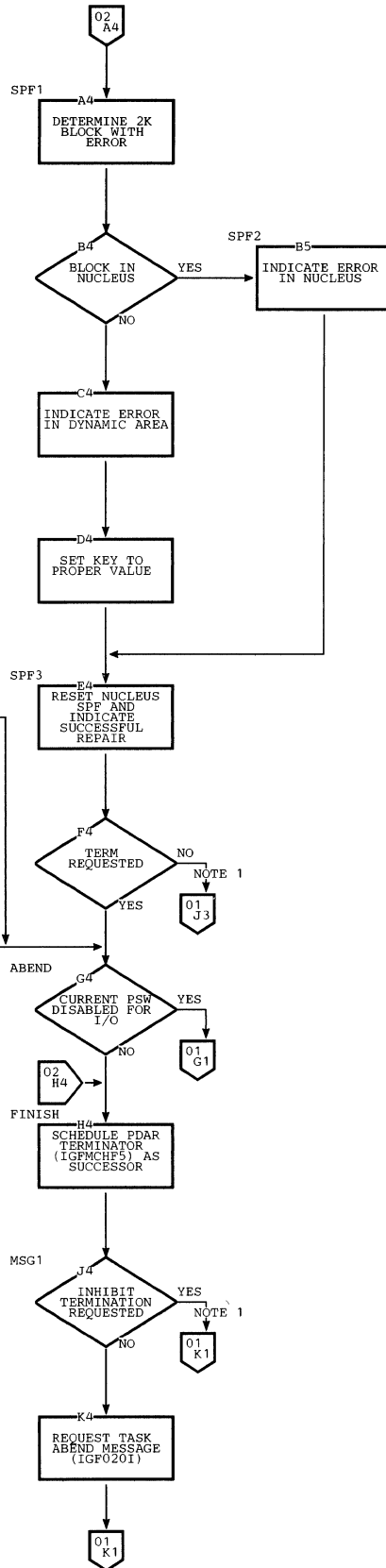
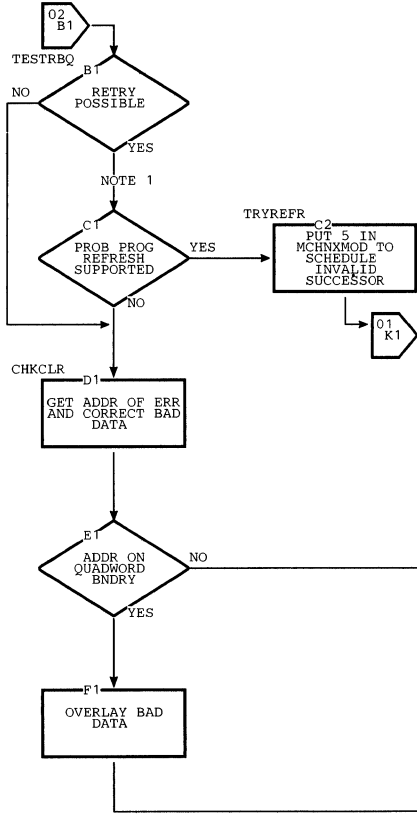
IGFMVTF2. MVT System Analysis 2 (Part 1 of 2)



IGFMFTF1. MFT System Analysis 1 (Part 1 of 2)

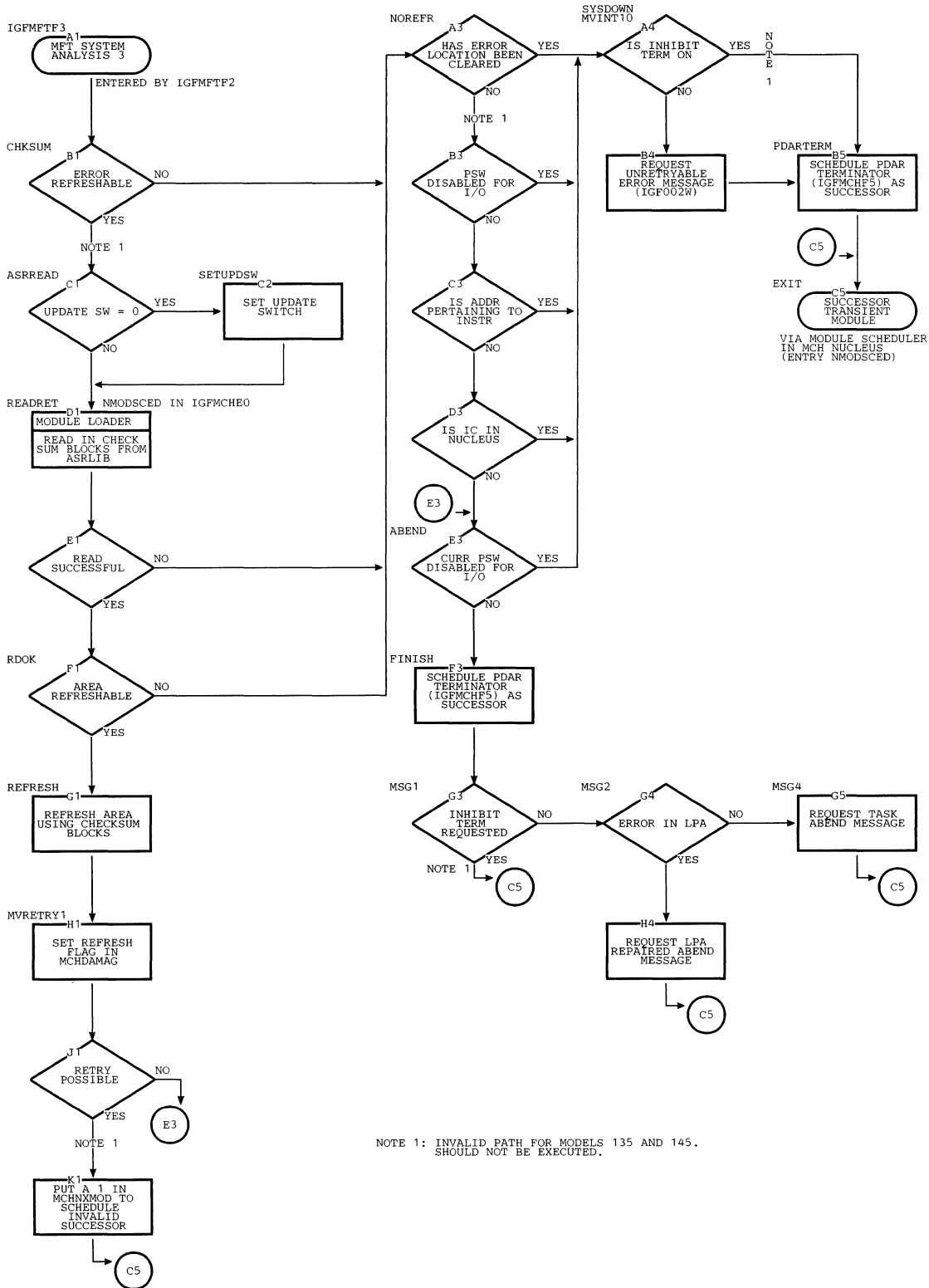


IGFMFTF1. MFT System Analysis 1 (Part 2 of 2)

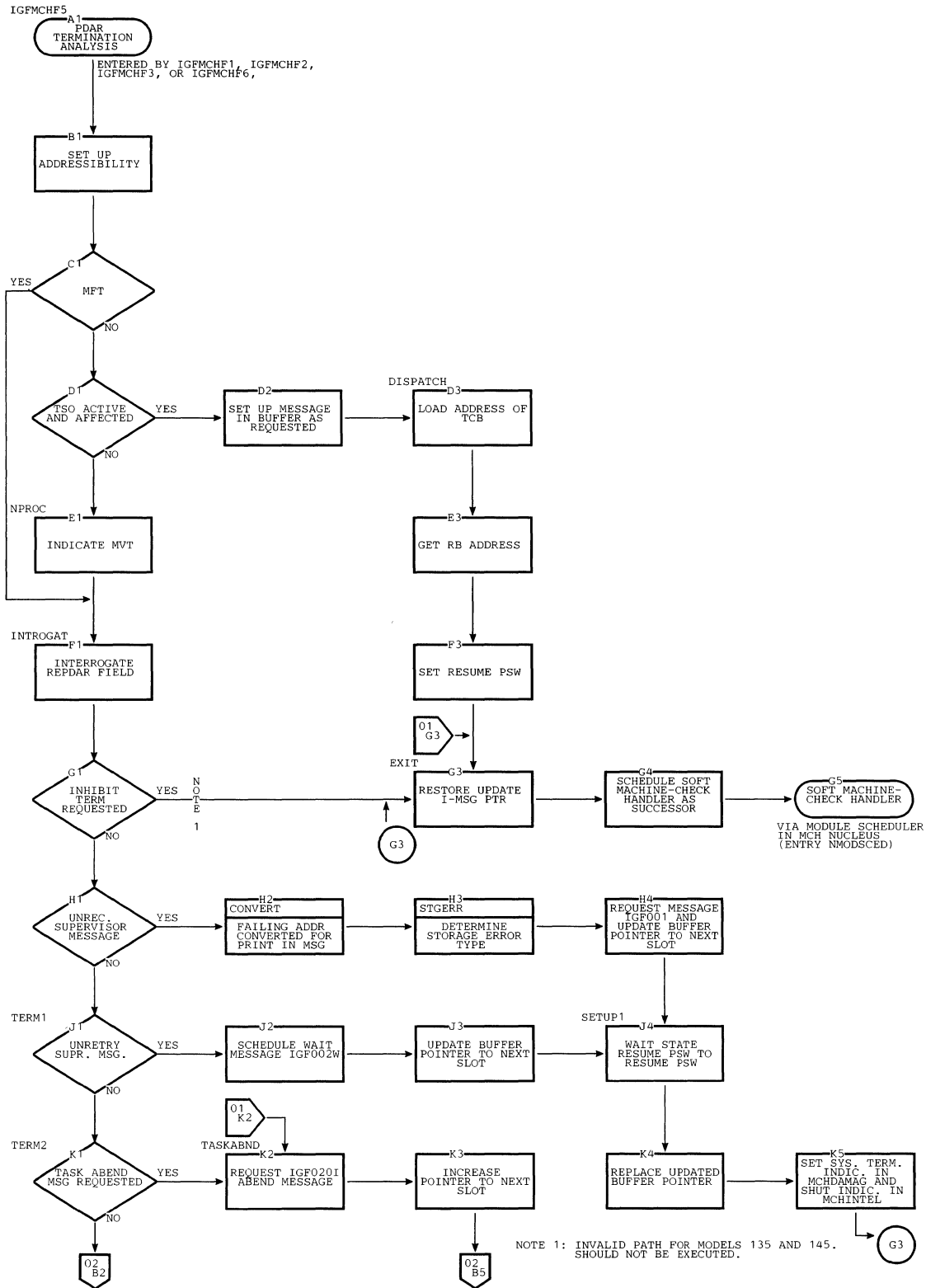


NOTE 1: INVALID PATH FOR MODELS 135 AND 145. SHOULD NOT BE EXECUTED.

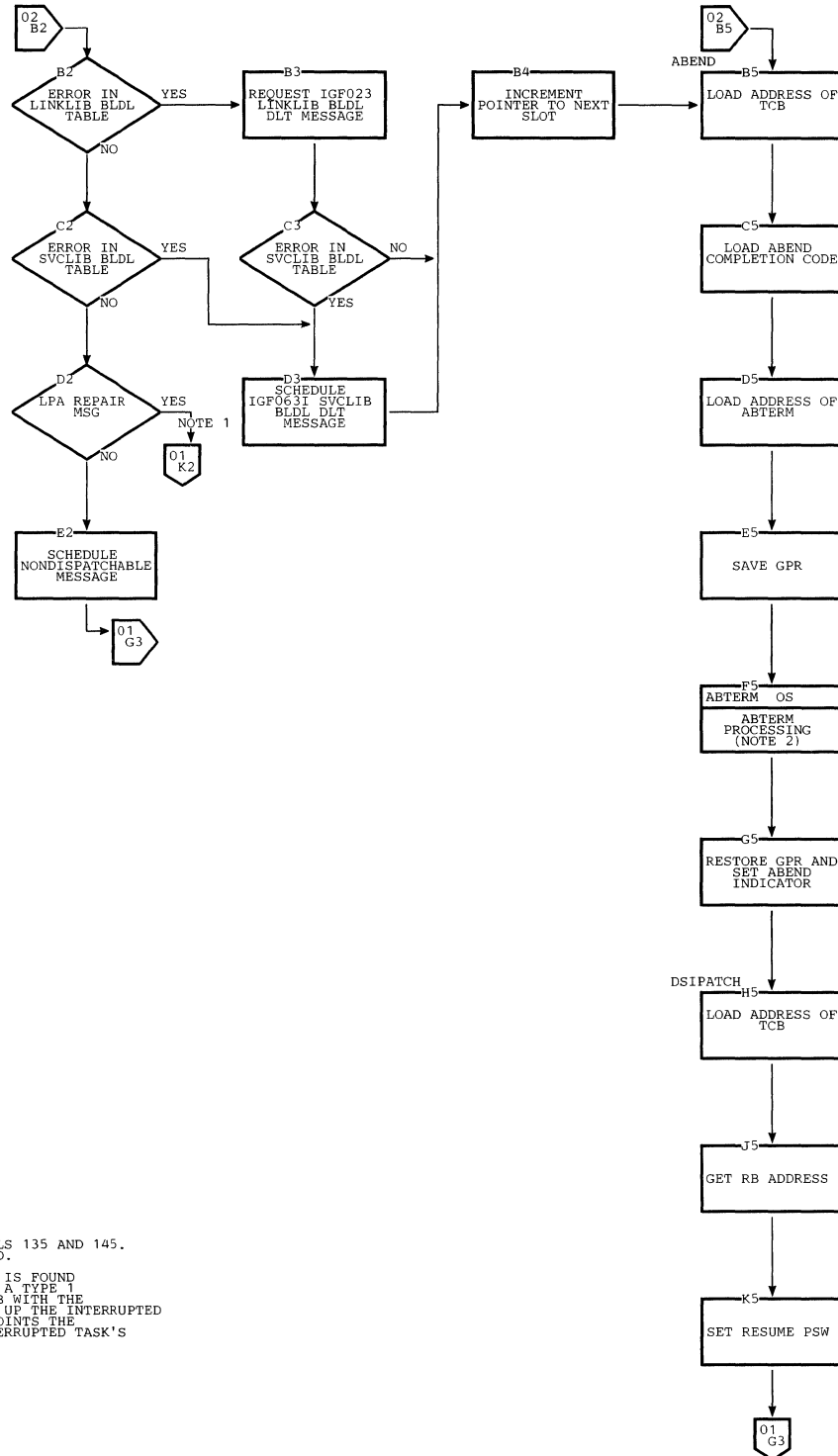
IGFMFTF3. MFT System Analysis 3



IGFMCHF5. PDAR Terminator (Part 1 of 2)



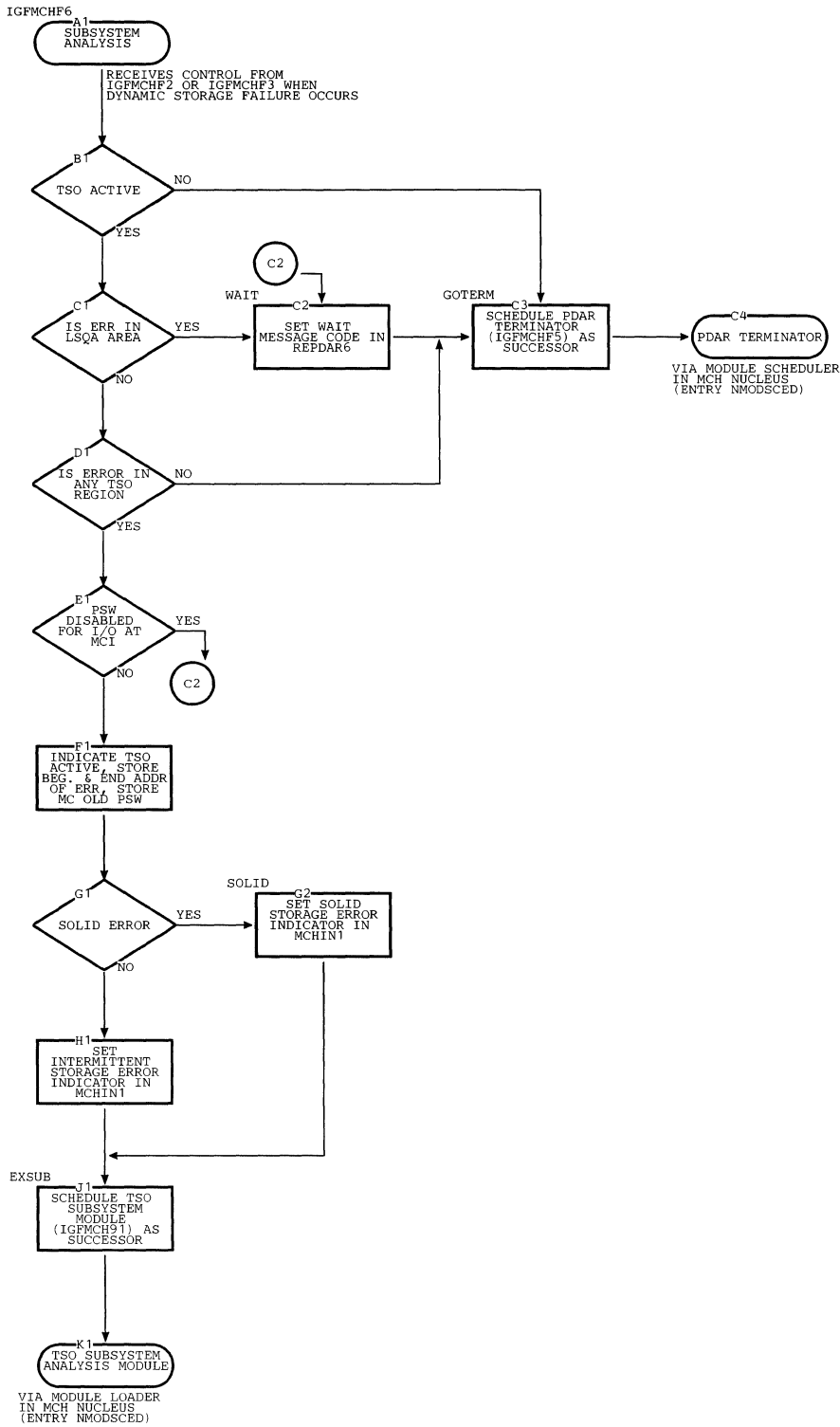
IGFMCHF5. PDAR Terminator (Part 2 of 2)



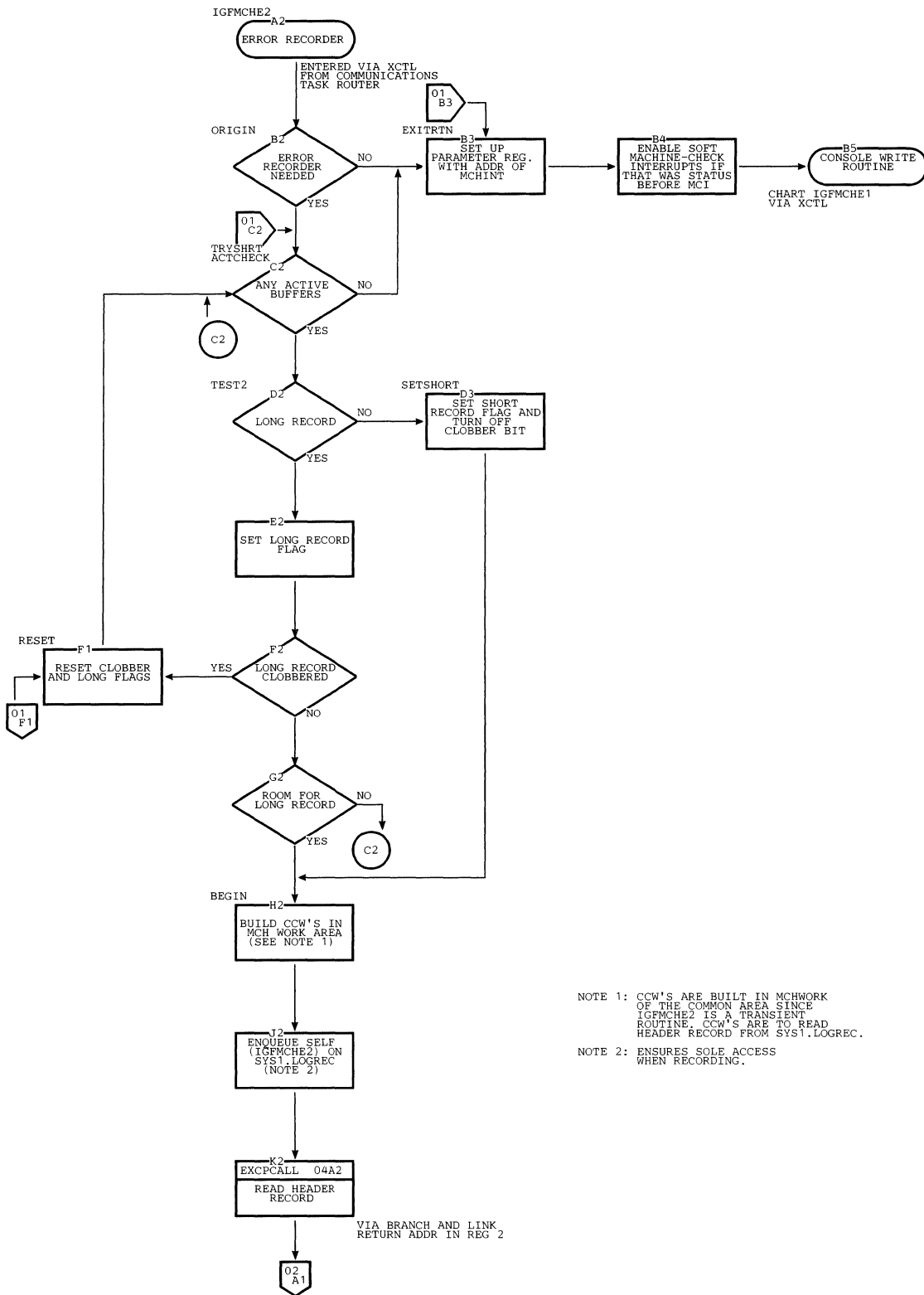
NOTE 1: INVALID PATH FOR MODELS 135 AND 145. SHOULD NOT BE EXECUTED.

NOTE 2: THE ADDRESS OF ABTERM IS FOUND IN THE CVT. ABTERM IS A TYPE 1 SVC THAT POSTS THE TCB WITH THE COMPLETION CODE, SETS UP THE INTERRUPTED TASK FOR ABEND, AND POINTS THE RESUME PSW OF THE INTERRUPTED TASK'S RB TO THE ABEND SVC.

IGFMCHF6. TSO Subsystem Analysis



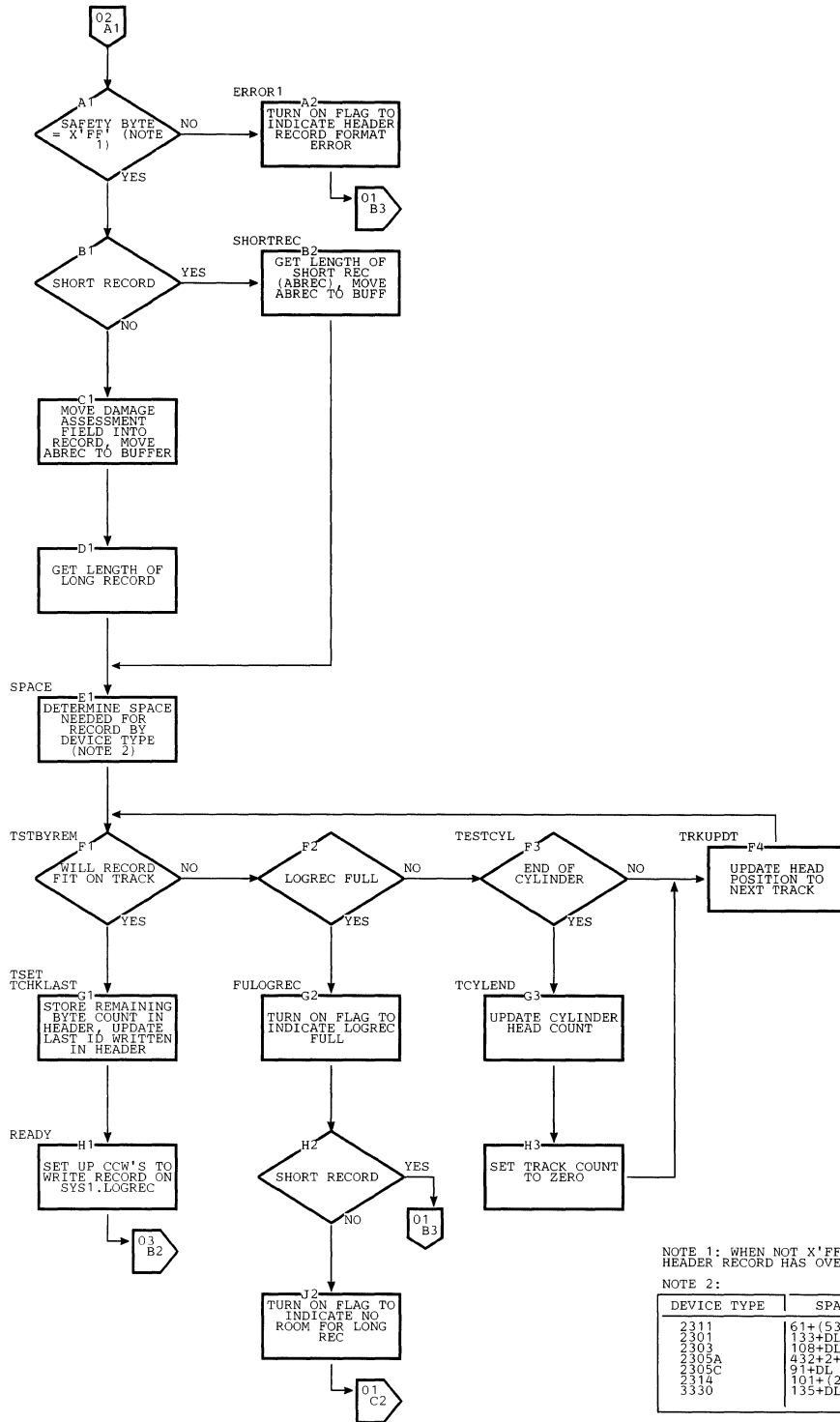
IGFMCHE2. Error Recorder (Part 1 of 4)



NOTE 1: CCW'S ARE BUILT IN MCHWORK OF THE COMMON AREA SINCE IGFMCHE2 IS A TRANSIENT ROUTINE. CCW'S ARE TO READ HEADER RECORD FROM SYS1.LOGREC.

NOTE 2: ENSURES SOLE ACCESS WHEN RECORDING.

IGFMCHE2. Error Recorder (Part 2 of 4)



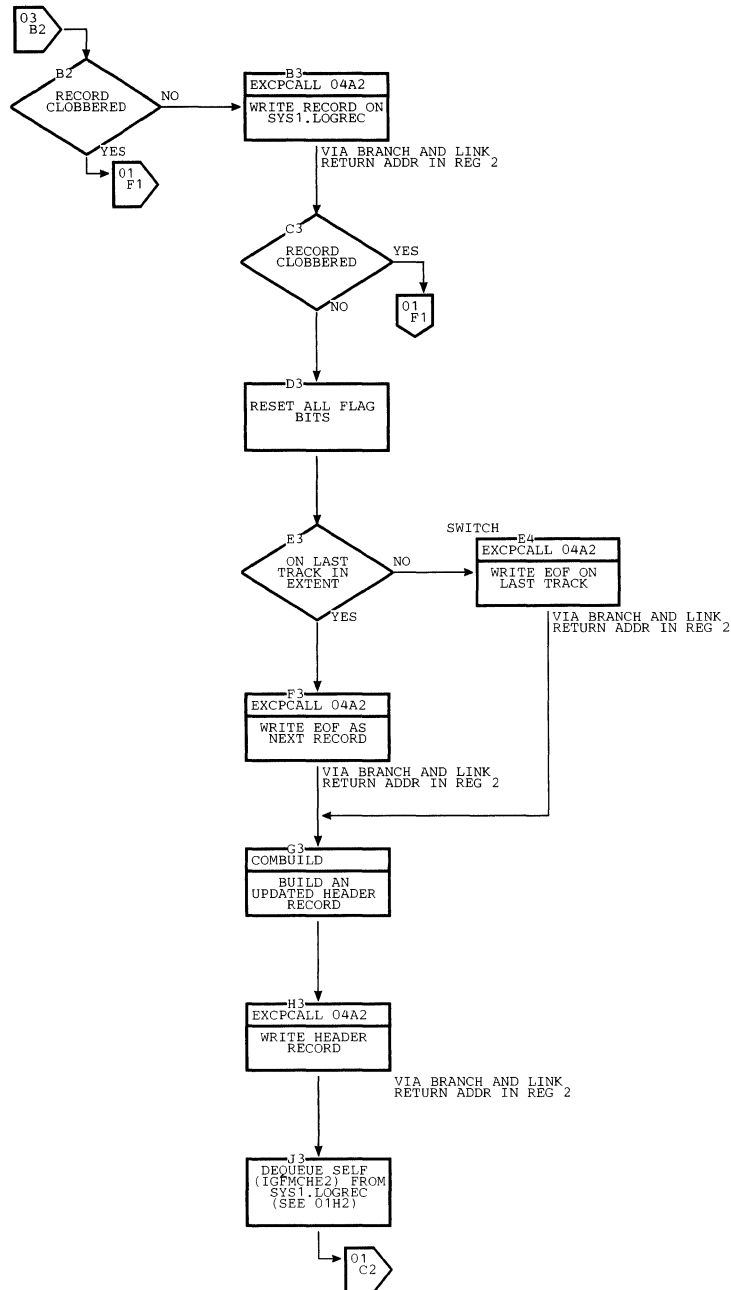
NOTE 1: WHEN NOT X'FF' HEADER RECORD HAS OVERRUN

NOTE 2:

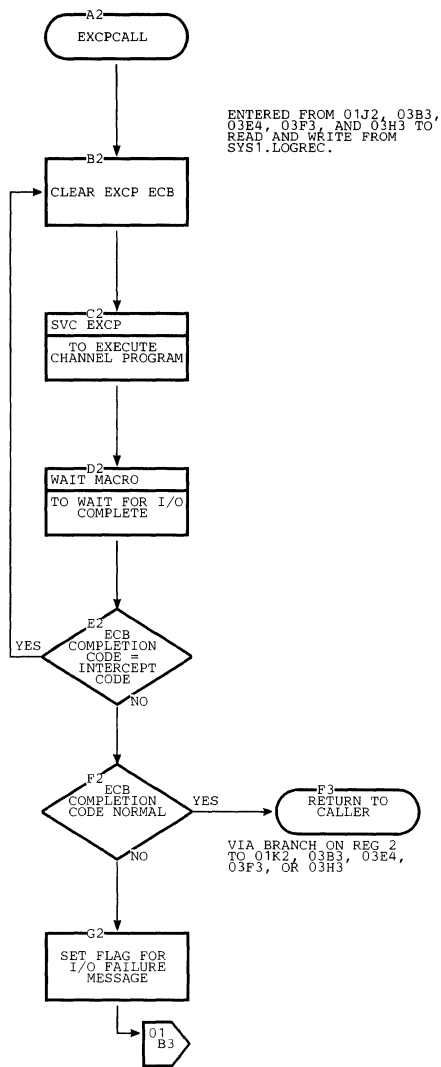
DEVICE TYPE	SPACE NEEDED
2311	61+(537(DL)/512)
2301	133+DL
2303	108+DL
2305A	432+2+DL
2305C	94+DL
2314	101+(2137(DL)/2048)
3330	135+DL

WHERE DL INDICATES DATA LENGTH (RECORD WITHOUT KEY)

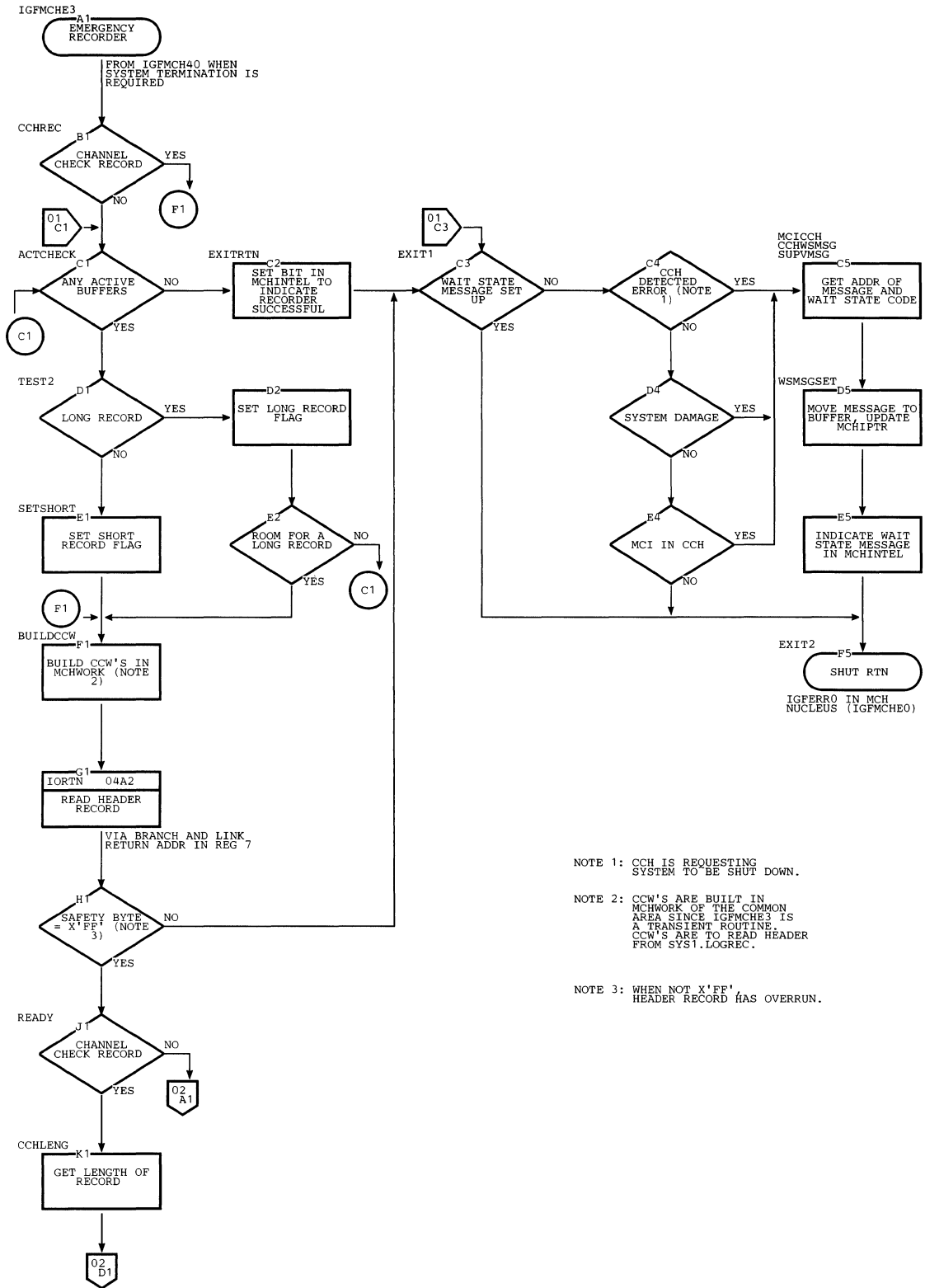
IGFMCHE2. Error Recorder (Part 3 of 4)



IGFMCHE2. Error Recorder (Part 4 of 4)



IGFMCHE3. Emergency Recorder (Part 1 of 3)

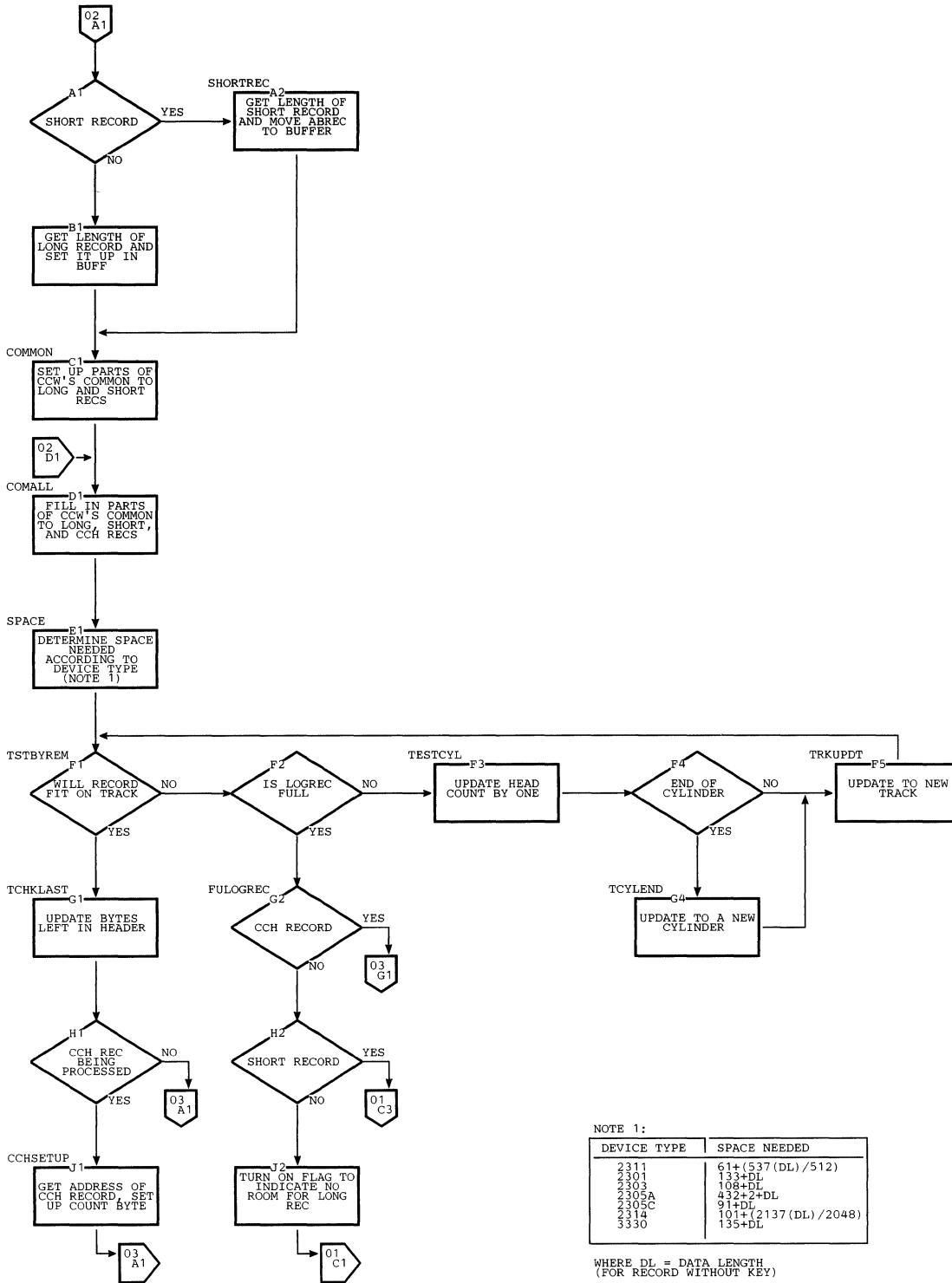


NOTE 1: CCH IS REQUESTING SYSTEM TO BE SHUT DOWN.

NOTE 2: CCW'S ARE BUILT IN MCHWORK OF THE COMMON AREA SINCE IGFMCHE3 IS A TRANSIENT ROUTINE. CCW'S ARE TO READ HEADER FROM SYS1.LOGREC.

NOTE 3: WHEN NOT X'FF' HEADER RECORD HAS OVERRUN.

IGFMCHE3. Emergency Recorder (Part 2 of 3)

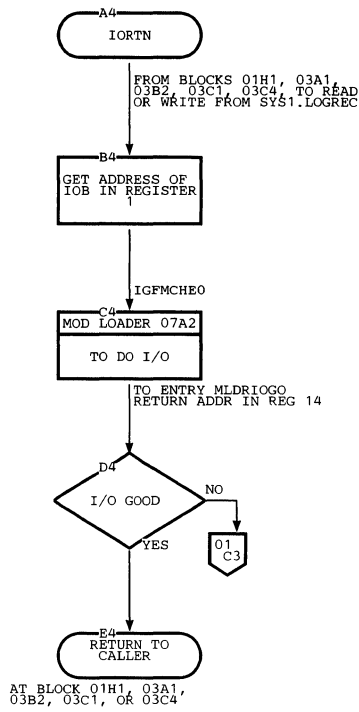
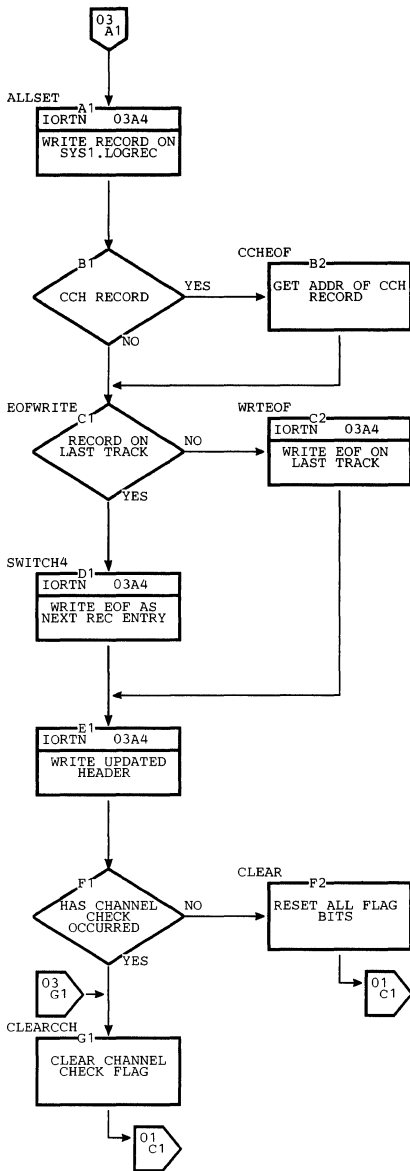


NOTE 1:

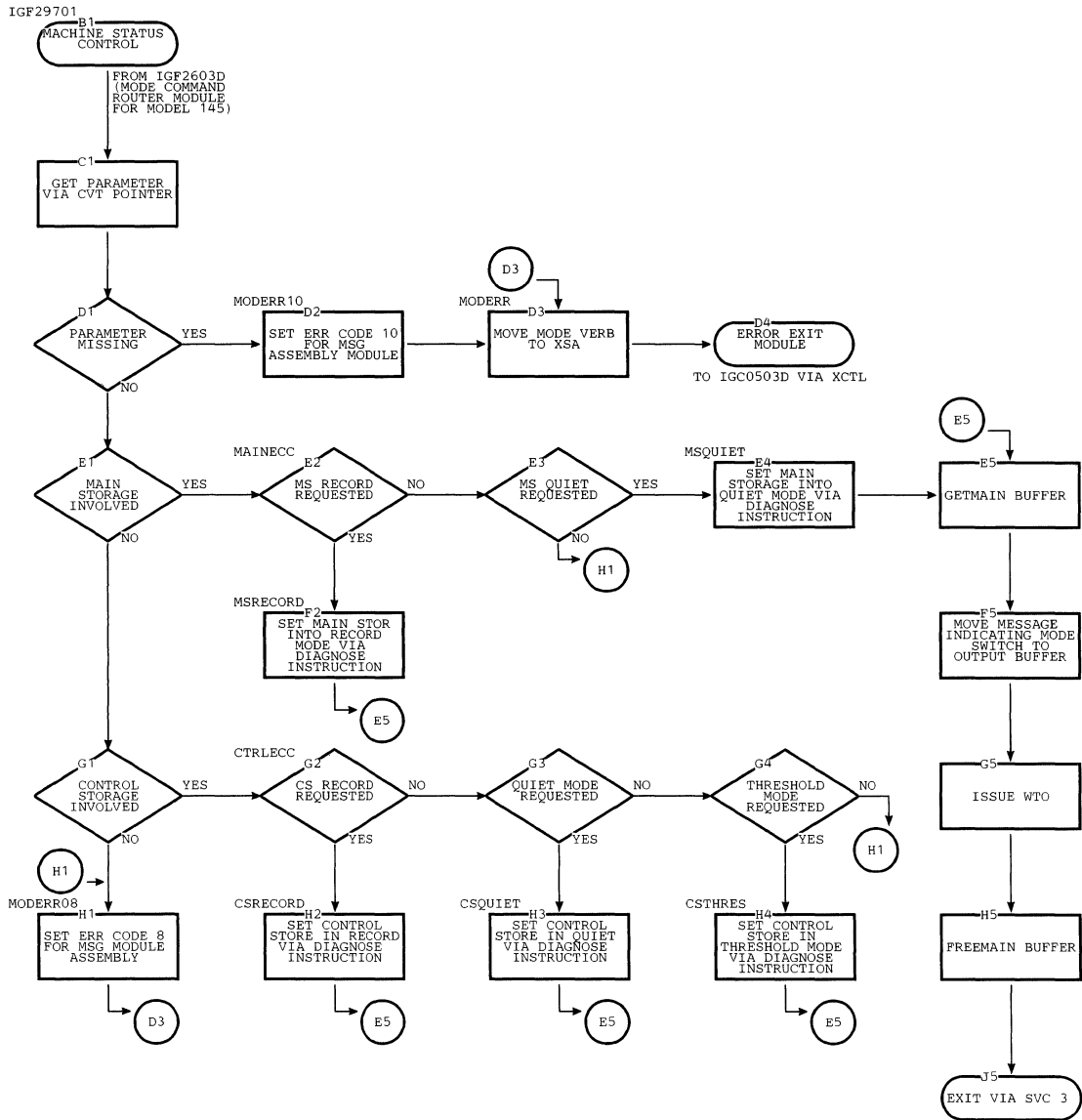
DEVICE TYPE	SPACE NEEDED
2311	61+(537(DL)/512)
2301	133+DL
2303	108+DL
2305A	432+2*DL
2305C	91+DL
2314	101+(2137(DL)/2048)
3330	135+DL

WHERE DL = DATA LENGTH (FOR RECORD WITHOUT KEY)

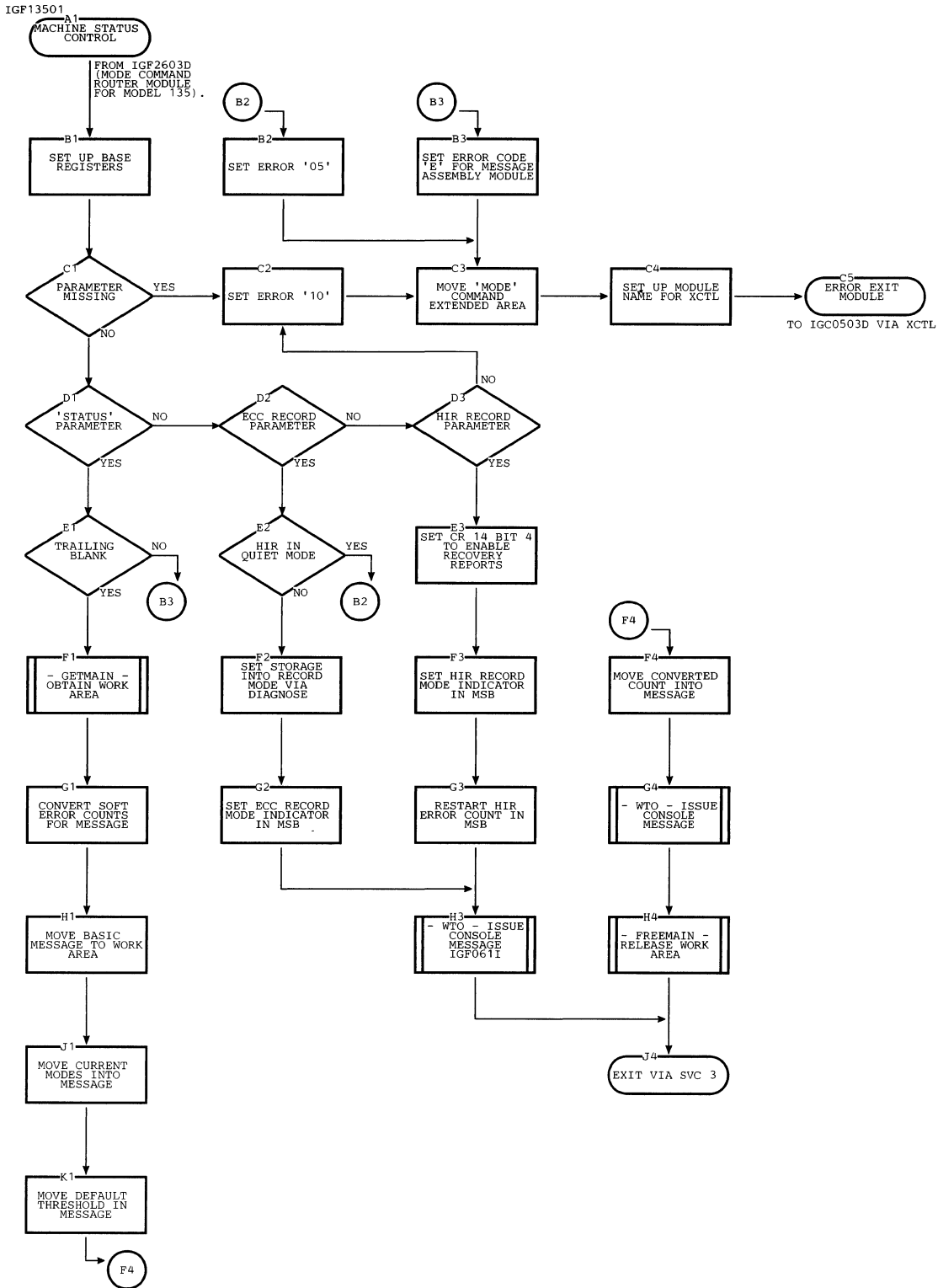
IGFMCHE3. Emergency Recorder (Part 3 of 3)



IGF29701. Model 145 Machine Status Control



IGF13501. Model 135 Machine Status Control



SECTION 4: MCH DATA AREAS

This section contains descriptions of the data areas:

Model Dependent Common Area

MCH Independent Common Area

Record Buffer Build Area

Fixed Logout

Extended Logout

Damage Assessment Field Buffer Area

Subsystem Data Area

Machine Status Block

Figure 3 shows the location of these data areas relative to the MCH program and one another.

Note: The Record Buffer Build Area, Fixed Logout, Extended Logout, and Damage Assessment Field Buffer Area together constitute the basic MCH record written for each machine-check interruption. The techniques of buffering and writing these areas are discussed in this section under Record Buffer Build Area.

MODEL DEPENDENT COMMON AREA

The Model Dependent Common Area occupies 8 bytes in the MCH Resident Area:

Bytes 0-5 are reserved

Bytes 6-7 contain the length of the damage assessment field (see MCHDAMAG in Figure 13).

MCH INDEPENDENT COMMON AREA

The MCH Independent Common Area occupies 1024 bytes in the MCH Resident Area. It is used by the MCH modules to communicate with one another and to store data to be entered into the environmental record. Each field in the Common Area is described below, in alphabetic order, with the fields displacement in decimal and hexadecimal, from the beginning of the Common Area. A storage map of the Common Area follows the description (see Figure 23).

MCHABREC 737 (2E1)

Three byte pointer to ABREC records, TTR table, and successor list.

MCHABRNO 736 (2E0)

One byte containing the number (in hexadecimal) of abbreviated records.

MCHASRNO 354 (162)

One byte containing the number (in hexadecimal) of checksum records.

MCHASRTR 352 (160)

One word containing the record number (in hexadecimal) of the first checksum record on SYS1.ASRLIB.

MCHBLDL 344 (158)

A pointer to the LINKLIB BLDL table.

MCHBUILD 720 (2D0)

A one-word pointer to the MCH error record build area.

MCHCVT 308 (134)

A pointer to the CVT.

MCHDAMAG 8 (8)

Two-word field containing error information to be used as part of the environmental record. (Figure 12 illustrates the contents of this field.)

MCHDCB 400 (190)

A pseudo DCB containing one word for use in loading records.

MCHDEB 404 (194)

A pseudo DEB containing twelve words the first 6 of which are all zeros. Fields included in the pseudo DEB are MCHDEDCB, MCHDEXSC, MCHDEBXT, MCHSTEXT, MCHENEXT, and MCHNMTRK. Figure 24 describes the fields of MCHDEB.

MCHDEBXT 428 (1AC)

One word containing the file mask and the address of the UCB.

MCHDEDCB 420 (1A4)

One word containing the protect key, the DEB ID, and a pointer to the associated DCB. This field is used by IOS to read checksums from SYS1.ASRLIB.

MCHDEXSC 424 (1A8)

One word containing an exit scale for direct access devices and an address appendage table.

MCHDISPL 356 (164)

One byte containing the displacement into successor table of the successor ID.

Dec	Hex								
0	0	MCHDPADR 4 (See note)				MCHLOGIC 4			
8	8	MCHDAMAG 8							
16	10	MCHLSUM 8							
24	18								
32	20	MCHHISTY 28							
40	28								
48	30					REP DAR1	REP DAR2	REP DAR3	REP DAR4
56	38	REP DAR5	REP DAR6	REP DAR7	REP DAR8	REP DARF1 4			
64	40	REP DARF2 4				REP DARI 4			
72	48	REMCOPSW 8							
80	50	REP DAR 16							
88	58								
96	60	MCHTTRS 4							
		MCHPSA 128							
224	E0								
232	E8	MCHMLSAV 64							
288	120					MCHNXIDS 4			
296	128	MCHTCB 4				MCHTTRIN 4			
304	130	MCHSHUT 4				MCHCVT 4			
312	138	MCHSIRB 4				MCHERXNT 4			
320	140	MCHPSTAD 4				MCHLDADR 4			
328	148	MCHEXCP 4				MCH2NDRY 4			
336	150	MCHIOENT 4				MCHERIOB 4			
344	158	MCHBLDL 4				MCHMSB 4			
352	160	MCHASRTR	MCHRELNO	MCHASRNO	MCHNXMOD	MCHDISPL	MCH1STDS	MCHINTEL 2	
360	168	MCHSUBA 2		MCHSUBF 2		MCHSUBP 4			
368	170	MCHNEST 4				MCHSVCEL 4			

Figure 23. MCH independent common area (Part 1 of 2)

Dec	Hex				
376	178	MCHSPARE 20			
396	18C	MCHUCB 4		MCHDCB 4	
404	194				
412	19C	MCHDEB 16			
420	1A4	MCHDEDCB 4		MCHDEXSC 4	
428	1AC	MCHDEBXT 4		(SPARE) 2	MCHSTEXT 2
436	1B4	MCHSTEXT 2 (Cont)		MCHENEXT 4	MCHNMTRK 2
444	1BC	MCHIOB 4		MCHIOECB 4	
452	1C4	MCHIOCSW 8			
460	1CC	MCHICWS 4		MCHIODCB 4	
468	1D4	MCHIOBSP 8			
476	1DC	MCHIOBSK 8			
484	1E4	SPARE 4		MCHINT 4	
492	1EC	MCHIPTR 4			
		MCHIBUF 100			
588	24C				
596	254				
		MCHWORK 120			
716	2CC	MCHLONG 4		MCHBUILD 4	
724	2D4	MCHINLOG 4		MCHREMCH 4	
732	2DC	MCHENQ 4		MCHABRENO 1	MCHABREC 3

Figure 23. MCH independent common area (Part 2 of 2)

Byte	Field Name	Length (bytes)	Contents
0-23	MCHDEB	24	Pseudo field containing zeros
24-27	MCHDEDCB	4	Protect key, pointer to DCB, DEB ID
28-31	MCHDEXSC	4	Exit scale and appendage table
32-35	MCHDEBXT	4	File mask
36-37		2	Reserved
38-41	MCHSTEXT	4	Starting CCHH of extent
42-45	MCHENEXT	4	Ending CCHH of extent
46-47	MCHNMTRK	2	Number of tracks

Figure 24. Fields of MCHDEB

MCHDPADR 0 (0)

One word containing the address of the dependent Common Area.

MCHENEXT 438 (1B6)

The ending CCHH of the extent containing the specified transient module.

MCHENQ 732 (2DC)

Reserved.

MCHERIOB 340 (154)

The address of the IOB for the MVT error transient area.

MCHERXNT 316 (13C)

A pointer to the MFT error transient area.

MCHEXCP 328 (148)

Entry into MCH/IOS interface.

MCHHISTY 24 (18)

Seven words containing the type and order of modules called during error processing. (See Section 5: "Diagnostic Aids.")

MCHIBUF 496 (1F0)

A 25-word message buffer.

MCHICBSP 440 (1B8)

Reserved

MCHICCWS 460 (1CC)

Address of the channel program used to service all MCH I/O requests.

MCHINLOG 724 (2D4)

A one-word pointer to the model independent logout save area.

MCHINT 488 (1E8)

One word used to inform the Emergency Recorder whether a CCH record must be recorded. This field also tells the Console Write routine the condition of SYS1.LOGREC. It is used to interface between MCH and recording and console write routines.

MCHINTEL 358 (166)

A two-byte field containing indicators used by the Secondary Error Handler. Figure 25 describes the significant bits in MCHINTEL.

MCHIOB 444 (1BC)

A flag word used by the Module Loader. It also denotes the entire group of fields in Figure 26.

MCHIOBSK 476 (1DC)

Two words containing the SEEK address (MBCCCHHR) of the module to be loaded. The Emergency Recorder uses this field to read and write header records and write error records.

MCHIOBSP 468 (1D4)

IOB displacement space.

MCHIOCSW 452 (1C4)

Two words containing the last seven bytes of the CSW.

MCHIODCB 464 (1D0)

The address of the DCB of SYS1.SVCLIB for module loading or the address of the DCB of SYS1.LOGREC for emergency recording.

Byte	Bit	Meaning
0	0	System down if unexpected error changes system data
	1	Continue MCH to termination-schedule emergency recorder (put system down)
	2	Put system down with scheduled message
	3	Reserved
	4	Reserved
	5	Terminator scheduled from Secondary Handler
	6	Status change has occurred
1	7	Error record recorded
	0	Indicates Module Loader used
	1-7	Reserved

Figure 25. Fields of MCHINTEL

Byte	Field Name	Length (bytes)	Content
0		1	Flag for command chaining
1-3	MCHIOB	3	Not used
4-7		4	Pointer to MCHCECB
8		1	I/O Error Flags (IOBFLAG3)
9-15	MCHIOCSW	7	Last 7 bytes of CSW
16-19	MCHICWS	4	Address of channel program
20-23	MCHIODCB	4	Address of DCB
24-31		8	Reserved
32-39	MCHIOBSK	8	SEEK field (MBBCCHHR)

Figure 26. Fields of MCHIOB

MCHIOECB 448 (1C0)

A pointer to the MCH ECB.

MCHIOENT 336 (150)

Entry point address of IOS.

MCHIPTR 492 (1EC)

A one-word pointer to the WTO message buffer.

MCHLDADR 324 (144)

Entry point of MCH Module Loader.

MCHLOGIC 4 (4)

One word containing control bits used by the MCH Nucleus to determine which model-dependent MCH module should be designated as successor.

MCHLONG 716 (2CC)

A one-word pointer to the long error record. (See "Error Recording" in Section 2.)

MCHLSUM 16 (10)

Two words denoting the type and number (in hexadecimal) of records lost due to being overlaid by new error records. The field is formatted as shown in Figure 27.

MCHMLSAV 228 (E4)

Sixteen words used by I/O and transient routines to save registers.

MCHMSB 348 (15C)

A pointer to the Machine Status Block.

Byte	Meaning
0	Number of tasks terminated due to CPU errors.
1	Number of tasks recovered from CPU errors.
2	Number of errors recovered by CPU retry.
3	Number of tasks terminated due to storage errors.
4	Number of tasks recovered from storage errors.
5	Number of errors recovered by ECC.
6	Reserved.
7	Reserved.

Figure 27. Fields of MCHLSUM

MCHNEST 368 (170)

A pointer to the IOS nest switch.

MCHNMTRK 442 (1BA)

Number of tracks in the extent of the data set being referenced.

MCHNXIDS 292 (124)

A pointer to an eight word field containing the name of each MCH module and a list of the IDs of its successor modules. The format of this field is:

Name (ID)	Number of Successors	ID of Successor 1	ID of Successor n

where each ID occupies one byte

MCHNXMOD 355 (163)

One byte denoting functional successor (set up by transient modules for use by Module Scheduler).

MCHPSA 100 (64)

Thirty-two words to save the Permanent Storage Allocation (0-128 decimal) at the time of the interruption.

MCHPSTAD 320 (140)

The address of the MCH posting routine.

MCHRELNO 353 (161)

Operating system release number (in hexadecimal).

MCHREMCH 728 (2D8)

A one-word pointer to the damage assessment area.

MCHRPSW (see REMCOPSW)

MCHSHUT 304 (130)
A word containing an address used by the Emergency Recorder to return to the SHUT routine.

MCHSIRB 312 (138)
A pointer to the SIRB.

MCHSPARE 376 (178)
Reserved for future use.

MCHSTEXT 434 (1B2)
The starting CCHH of the extent of the data set being referenced.

MCHSUBA 360 (168)
This field denotes the subsystems running under the operating system. TSO is indicated by hexadecimal 80.

MCHSUBF 362 (16A)
This field denotes subsystems for which no MCH support is provided.

MCHSUBP 364 (16C)
A pointer to the subsystem data area (MCHSUB).

MCHSVCBL 372 (174)
A pointer to the SVCLIB BLDL table.

MCHTCB 296 (128)
One word used by the PDAR modules to store the address of the current TCB.

MCHTTRIN 300 (12C)
One word containing the TTR of the next transient module to be loaded. The TTR must be converted to an absolute address.

MCHTTRS 96 (60)
A one-word field containing a pointer to the field containing the TTRs, IDs, and displacements into the successor table of all MCH transient modules.

MCHUCB 396 (18C)
A one-word field containing the UCB address for I/O routines.

MCHWORK 596 (254)
A 120-byte work area for the error recorder.

MCH1STDS 357 (165)
A one-byte field containing the displacement to the successor ID's of IGFMCHE0 for scheduling of the first successor module.

MCH2NDRY 332 (14C)
A pointer to the second MCH Independent Common Area.

REMCOPSW 72 (48)
A doubleword for saving the machine check old PSW.

REPDAR 80 (50)
A 16-byte field containing the job name and step name of the interrupted program.

REPDARF1 60 (3C)
One word containing the starting address of a failing location.

REPDARF2 64 (40)
A one-word location containing the end of a failing location.

REPDARI 68 (44)
A one-word location containing the instruction address at the time of the failure.

REPDAR1 52 (34)
A one-byte field containing action(s) taken by PDAR for a specific failure.

REPDAR2 53 (35)
A continuation of REPDAR1.

REPDAR3 54 (36)
A one-byte field describing operating system status.

REPDAR4 55 (37)
A one-byte field used by PDAR to indicate error location.

REPDAR5 56 (38)
A one-byte field indicating instruction location at the time of the error.

REPDAR6 57 (39)
A one-byte field indicating the scheduling of messages to the operator denoting action taken by PDAR modules.

REPDAR7 58 (3A)
A reserved byte; not used

REPDAR8 59 (3B)
A reserved byte; not used

Figure 28 describes the contents of REPDAR1 through REPDAR8.

RECORD BUFFER BUILD AREA

The Record Buffer Build Area occupies 80 bytes following the Independent Common Area (see Figure 3). The first 2 bytes are unused except for boundary alignment. The next 8 bytes, labeled CTFIELD, contain control information; and the last 70 bytes contain the short form of the MCH record, called the ABREC.

REPDAR1		PDAR Action
Bit	Content	Indicates
0	1	Termination necessary
1	1	Repair/retry failed
2	1	Retry possible (Unused)
3	1	Indeterminate instruction counter
4	1	Instruction involved
5	1	Operand involved
6	1	System wait if Refresh/repair/retry fails
7	1	Inhibit termination (Unused)
REPDAR2		PDAR Action
Bit	Content	Indicates
0	1	Solid storage data error
1	1	Intermittent storage data error
2	1	Solid SPF key error
3	1	Intermittent SPF key error
4	1	Refresh/repair successful
5	1	Storage data error location cleared
6	1	Storage block failure
7	1	Storage unit failure
REPDAR3		Operating System Status
Bit	Content	Indicates
0	1	Wait pseudo task
1	1	Master scheduler task
2	1	System task
3	1	Problem program task
4	1	Current PSW disabled for I/O and external interrupts
5-7		Unused
REPDAR4		Location of Error
Bit	Content	Indicates error in
0	1	Nucleus
1	1	SVC transient area
2	1	Error transient area

Figure 28. PDAR control and action bytes (Part 1 of 2)

REPDAR4 (Cont'd)		Location of Error
3	1	Refreshable nucleus CSECT
4	1	Dynamic area
5	1	Link pack area
6	1	Resident type III SVC
7	1	BLDL table
REPDAR5		Location of Instruction When Failing Address Involves Operand
Bit	Content	Indicates instruction is in
0	1	Nucleus
1	1	Dynamic area
2	1	Link pack area
3-7		Reserved for future use
REPDAR6		Messages
Bit	Content	Meaning
0	1	Schedule unrecoverable supervisor error message
1	1	Schedule unretryable supervisor error message
2	1	Schedule unrecoverable error in dynamic area message
3	1	Schedule task ABEND message
4	1	Schedule LINKLIB BLDL deleted and task ABEND message
5	1	Schedule SVC BLDL deleted and task ABEND message
6-7		Unused
REPDAR7 and REPDAR8		Reserved for future use

Figure 28. PDAR control and action bytes (Part 2 of 2)

When writing a short record, MCH moves the current ABREC from MCHABREC of the Independent Common Area to the 70 bytes reserved for it in the Record Buffer Build Area. MCH then writes the short record from this buffer to the SYS1.LOGREC.

When writing a full record, MCH increments its pointer to CTFIELD by 22 bytes and moves the first 48 bytes of the current ABREC from MCHABREC to the area following the new CTFIELD location. MCH can then write a full record from a contiguous area containing the:

1. First 48 bytes of ABREC
2. Fixed Logout

3. Extended Logout

4. Damage Assessment Field Buffer Area

The contents of the short MCH record or ABREC are shown in Figure 29. The 48 bytes of ABREC used in writing a long MCH record are Key through Machine-Check Interruption Code. All of the ABREC is used when writing a short MCH record.

MCHBUILD in the Independent Common Area points to CTFIELD in the Record Buffer Build Area. CTFIELD is filled in by MCH recorders, IGFMCHE2 and IGFMCHE3, and contains:

Dec	Hex				
0	0	Key	OS Release Number	Switches	2
4	4	Reserved			4
8	8	Date			4
12	C	Time			4
16	10	CPU ID			8
24	18	Program ID			8
32	20	Job ID			8
40	28	Machine-Check Old PSW			8
48	30	Machine-Check Interruption Code			8
56	38	Damage Assessment Data			8
64	40	Lost CPU ABEND Counter	Lost CPU Recovered Counter	Lost CPU Soft Counter	Lost ECC ABEND Counter
68	48	Lost ECC Recovered Counter	Lost ECC Soft Counter	Flag (See Figure 22)	Reserved

Figure 29. Fields of ABREC

1. The CCHHR (5 bytes) of the next record entry available on SYS1.LOGREC.
2. The key length (1 byte) of the MCH record.
3. The data length (2 bytes) of the MCH record.

Note: It is possible that there not be an Extended Logout. In this case, MCH sets up the Damage Assessment Field Buffer Area to immediately following the Fixed Logout.

Figure 30 shows the possible kinds of MCH error records that can be written.

FIXED LOGOUT

The Fixed Logout occupies main storage locations 176 through 511 (decimal). When MCH is first entered, it moves Fixed Logout data from locations 232 through 511 to the

area immediately following the ABREC in the Record Buffer Build Area (see Figure 3).

Fixed Logout locations 176 through 231 are not used by MCH to create the full MCH record.

Figure 31 shows the contents of the Fixed Logout in storage locations 176-511.

EXTENDED LOGOUT

Producing an Extended Logout is attempted for all machine-check interruptions when allowed by the Machine-Check Extended Logout Mask in control register 14 (when bit 1=1). The Extended Logout information (see Figures 32 and 33) is placed into one of the following main storage locations:

- On the Model 145, into the location addressed by the Machine-Check Extended

Logout Pointer contained in control register 15.

- On the Model 135, into a save area within the Fixed Logout beginning at decimal location 256.

See Figure 3 for the location of the extended logout.

Extended Logout for the Model 145: The 112 bytes of the extended logout shown in Figure 32 are positioned in the first 112 bytes of the 192-byte area reserved for the extended logout. The last 80 bytes of this area are unused. When an extended logout is not produced by the hardware, the Damage Assessment Field Buffer Area immediately follows the Fixed Logout in the main storage locations used to build a full MCH record.

Extended Logout for the Model 135: The extended logout, shown in Figure 33, occupies 14 bytes of a scratch area within the Fixed Logout of the Model 135. Also within this scratch area, following the MCH extended logout and a six-byte reserved area, is a CCH logout of 4 bytes. Since the extended logout is completely contained within the Fixed Logout, the Damage Assessment Field Buffer Area immediately follows the Fixed Logout in main storage locations used to build a full MCH record.

DAMAGE ASSESSMENT FIELD BUFFER AREA

This field of the MCH record occupies 74 bytes of main storage immediately following the Extended Logout. If there is no Extended Logout, this field follows the Fixed Logout¹. Figure 13 shows the damage assessment data in this area (beginning with byte 520 of the MCH error record).

MCH moves data into this portion of the MCH error record from the Independent Common Area. Figure 13 indicates which portions of the Independent Common Area are used.

¹Since the Extended Logout of the Model 145, even when present, consists of only 112 bytes of data, the Damage Assessment Field immediately follows that 112 bytes of data rather than following the entire 192-byte area reserved for the Extended Logout. However, an entire 74 bytes of main storage is reserved for the Damage Assessment Field following the 192 bytes for the Extended Logout.

SUBSYSTEM DATA AREA (MODEL 145 ONLY)

The Subsystem Data Area occupies 64 bytes of main storage following the Damage Assessment Field Buffer Area (see Figure 3). When Subsystem Analysis (IGFMCHF6) determines that the error occurred in TSO, it stores information in the Subsystem Data Area for use by TSO and passes control to a TSO Analysis Module IGFMC91.

See the IBM System/360 Operating System Time Sharing Option (TSO) Control Program, GY27-7199, for further information.

MACHINE STATUS BLOCK

The Machine Status Block (MSB) is a system control block used by the Soft Machine-Check Handler and the Machine Status Control routines for recording and mode switching.

The MSB is initialized during NIP processing by IGFMC0 (MCH Initialization). The CVTRMS field of the Communications Vector Table in the resident nucleus contains the address of the MSB.

The fields of the MSB differ for the Models 135 and 145. Below are the fields of the MSB for each model. Displacements from the address in the CVTRMS field are shown as decimal (hexadecimal).

Model 135 Machine Status Block

- MSBCOUNT 8 (8)
One word used as a counter for soft errors.
- MSBCR14 12 (C)
One word used to store control register 14.
- MSBHDCPY 28 (1C)
One word used as a pointer to the UCB of any display device attached to the Model 135.
- MSBMCW 0 (0)
Two words containing the status of the Model 135.
- MSBMODE 20 (14)
One byte used to indicate whether the Model 135 is in recording or quiet mode.
- MSBMSCON 24 (18)
One word used as a pointer to the UCB of the master console.
- MSBTHRLD 16 (10)
One word used to hold the soft error threshold value.

Model 145 Machine Status Block

MSBHDCPY 44 (2C)

One word used as the pointer to the UCB of any display device attached to the Model 145.

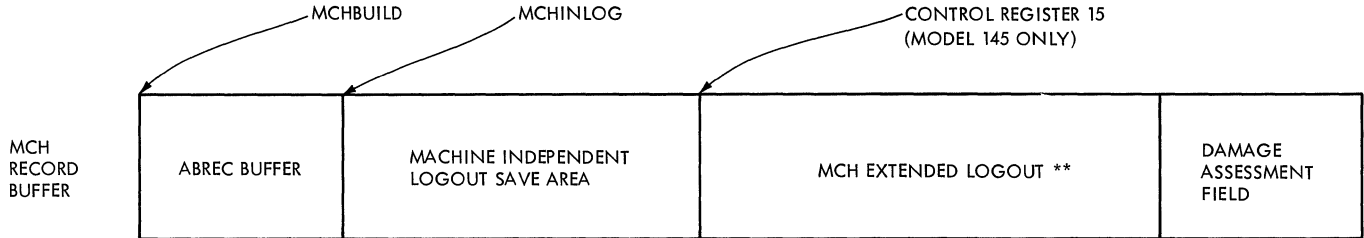
MSBMSCON 40 (28)

One word used as a pointer to the UCB of the master console.

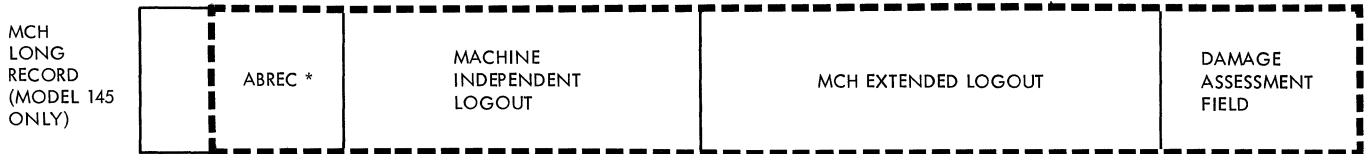
MSBNODSW 0 (0)

A byte of switches used as follows:

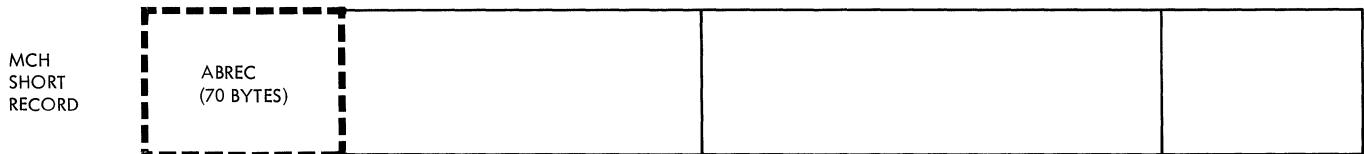
0xxx xxxx Main storage in quiet mode.
 1xxx xxxx Main storage in record mode.
 x1xx xxxx Control storage in record mode.
 xx1x xxxx Control storage in quiet mode.
 x00x xxxx Control storage in threshold mode.



Area - MCH records are built here before writing them to SYS1.LOGREC.



When there is not an Extended Logout.



* Only 48 bytes of ABREC are recorded for MCH LONG RECORD.
 ** Extended Logout is contained within the Independent Logout for the Model 135.

Figure 30. Possible MCH error records

<u>Dec</u>	<u>Hex</u>	
176	B0	ECSW
180	B4	Reserved (Channel ID)
184	B8	I/O Extended Log Pointer
188	BC	Reserved
232	E8	Machine Check Interrupt Code
240	F0	Reserved
248	F8	Failing-Storage Address
252	FC	ECC Bits Control Word Address (Region Code)
256	100	Fixed Logout Area
352	160	Floating-Point Register Save Area
384	180	General Register Save Area
448	1C0	Control Register Save Area
508	1FC	

Figure 31. Fields of the fixed logout

<u>Dec</u>	<u>Hex</u>	
0	0	Retry Counts
4	4	Machine Check Register A
8	8	Machine Check Register B
12	C	ABRTY Register
16	10	SPTLB Register
20	14	HMRTY Register
24	18	CPURTY Register
28	1C	Control Word
32	20	System Register
36	24	I Register of expanded local storage
40	28	U Register of expanded local storage
44	2C	W Register of expanded local storage
48	30	V Register of expanded local storage
52	34	X Register of local storage
56	38	R Register of local storage
60	3C	Y Register of local storage
64	40	Q Register of local storage
68	44	IBU Register of expanded local storage
72	48	TR Register of expanded local storage
76	4C	SPARE
80	50	SN Register of expanded local storage
84	54	PN Register of expanded local storage
88	58	WK Register of expanded local storage
92	5C	NP Register of expanded local storage
96	60	DM Register of local storage
100	64	DW Register of local storage
104	68	CPU Register (Mode Register)
108	6C	PSWCTL Register

Figure 32. Fields of the extended logout for the Model 145

Displacements within the Fixed Logout

Dec Hex

256	100	CPU Checks 0		CPU Checks 1	
260	104	BAR		Zone in Error	0-0's* SAR**
264	108	SAR (continued)***		Retry Threshold	Retry Flag Retry Count
268	10C	Interrupt Status Latches		Reserved	
272	110	Reserved			
276	114	ICA Check Byte	Select Channel Checks 2	IFA Check Byte	Select Channel Checks 3

<----- 4 bytes ----->

- * 5 bits in length
- ** 3 bits (Total SAR field is 19 bits long)
- *** 16 bits

Figure 33. Fields of the Extended Logout for the Model 135

SECTION 5: DIAGNOSTIC AIDS

This section is intended to assist in diagnosing problems in the Machine-Check Handler program. Included are: register conventions, common machine-check interruption codes, and possible problems that may exist when the "unexpected error" message appears.

REGISTER CONVENTIONS

Figure 34 shows how MCH uses its registers. There are two modules that are exceptions to these conventions; the Error Recorder (IGFMCHE2) and Console Write (IGFMCHE1). Since these modules operate more as part of normal operating system processing rather than MCH processing, they follow the operating system's conventions.

COMMON INTERRUPTION CODE SETTINGS

Figure 35 illustrates the machine-check interruption code.

More than one error can be presented in the interruption code. For example, the SD and PD bits may both be on. In this case MCH would handle the most serious error--SD.

Interruption codes in which no error type is indicated (bits 0 through 8 are zero) are considered invalid by MCH and cause a disabled wait state and message IGF015W.

Register	Used by MCH as
0-9	Work registers
10	Pointer to the Communications Vector Table
11	Pointer to the MCH Common Area
12	Nucleus base register
13	Used to hold the address of the save area in the I/O interface
14	Contains the return address into the MCH Nucleus from transient modules
15	Transient module base register

Figure 34. Register conventions

UNEXPECTED ERRORS

When the Unexpected Error message appears, the following can be done to isolate the cause of the error:

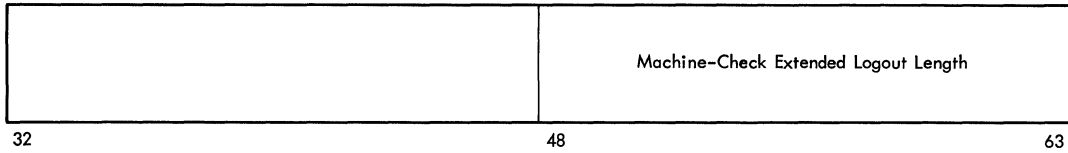
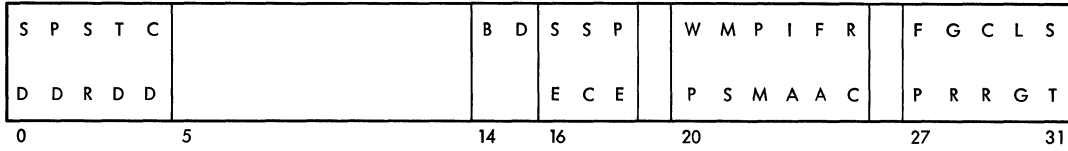
1. Check the interruption code for validity (see Figure 35). If the interruption code is invalid, the error was caused by a hardware malfunction.
2. Check to see if the Fixed Logout represents the same machine check as the Extended Logout.
3. Check the storage dump to see if a program check occurred. If a program check has occurred, and the instruction address portion of the program check new PSW is the same as the instruction address portion of the machine check new PSW, the probable cause of error is a program check in the Machine-Check Handler. The History Table in the MCH Common Area (MCHHISTY) can then be checked to determine in which MCH module the program check occurred.

MCH HISTORY TABLE

The MCH History Table (MCHHISTY in the MCH Common Area) can be used to determine which modules have been executed since the time of the machine-check interruption. The table also shows the sequence in which they were executed. The modules are identified by their IDs and level numbers.

When MCH is entered, the MCH Nucleus places its own ID and level number in the last two bytes of the MCH History Table. When a successor module is specified, the module loader subroutine in the Nucleus moves the data in the table two positions (bytes) toward the beginning of the table. The ID of the successor module is then put into the next to last byte in the table and hexadecimal "FF" is put into the last byte of the table. When the successor module is successfully loaded and given control, it places its level number into the last byte of the table, overlaying the X'FF' and signaling that it has received control.

As each module is loaded by the module loader, this process is repeated such that the most recently executing module has its ID and level number in the last two bytes of the MCH History Table; the next most recently executed MCH module has its ID and



0-4 Subclass
 14-15 Time of interruption (tense)
 16-18 Storage errors
 20-25, 27-31 Validity
 5-13, 19, 26, 32-47 Not assigned, stored as zero

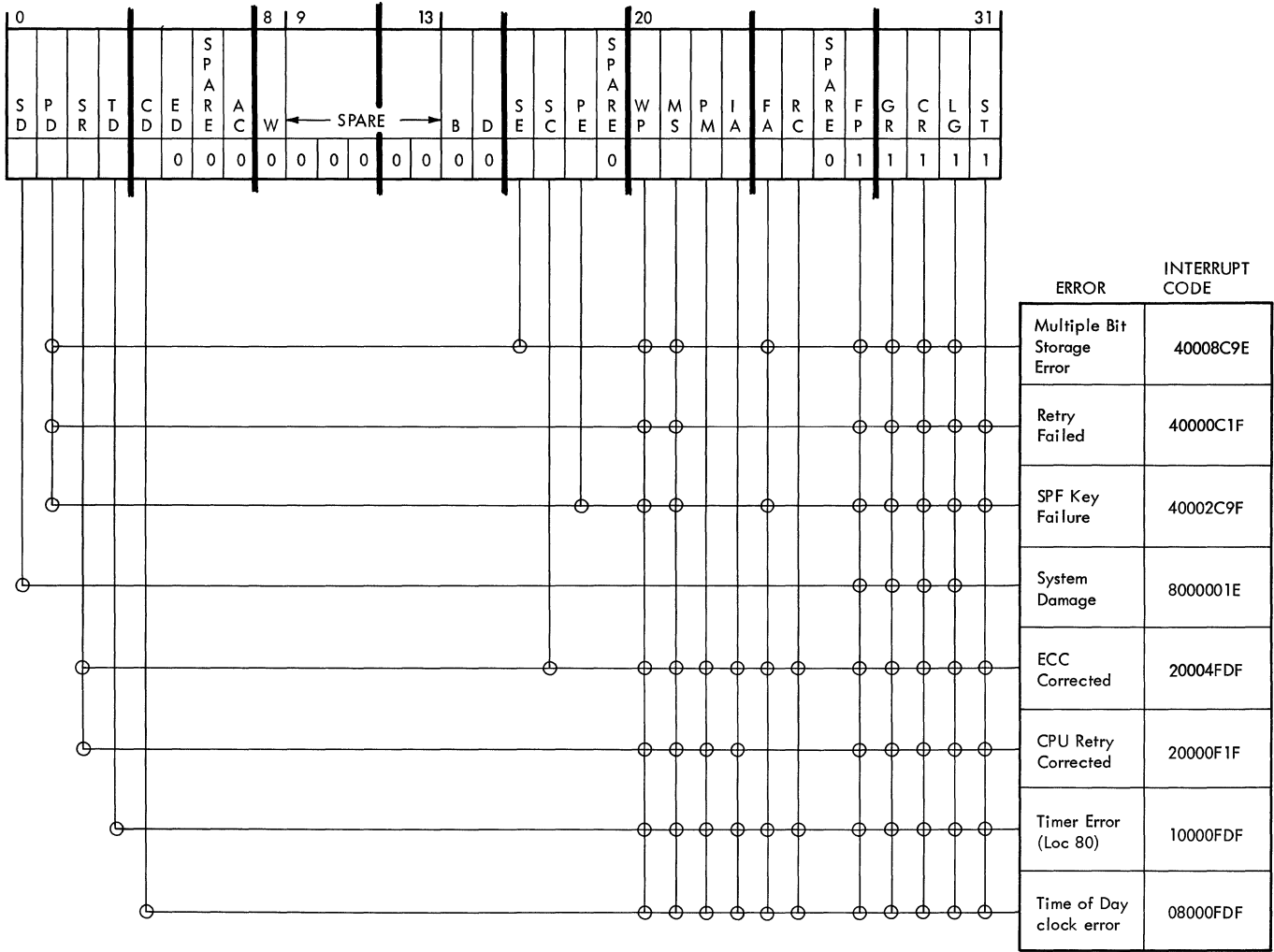


Figure 35. Sample machine-check interruption codes for the Model 145

level number in the previous two bytes of the table, etc.

Figure 36 shows the use of the MCH History Table in recording the following sequence of module execution:

1. MCH Nucleus - IGFMCHE0
2. Preliminary Error Analysis - IGFMCH41
3. MFT System Analysis 1 - IGFMFTF1
4. Soft Machine-Check Handler - IGFMCH40
5. Emergency Recorder - IGFMCHE3

MCHHISTY	ID	LEVEL	ID	LEVEL
0	0			
4	4			
8	8			
C	12			
10	16		E 0	0 1
14	20	4 1	0 1	F 1 0 1
18	24	4 0	0 1	E 3 F F

28 bytes

Figure 36. MCH history table

SECTION 6: MCH MODULE DIRECTORY

The module directory is a guide to named areas of code in the program listing. The module names are listed in the table below in alphabetic order. The other columns contain the module's descriptive name, major functions, entry point, and library residence. The module name also serves as a flowchart identification and a microfiche reference.

Note: The name within parentheses (in the Module/CSECT name column) identifies the module as it is cataloged in SYS1.SVCLIB. If there is no name in parentheses for an entry, the regular module name identifies the module in SYS1.SVCLIB.

Module/ CSECT Name	Module Name/Major Functions	Entry Point	Library
IGFMCHE0	MCH Nucleus Initializes MCH. Handles unexpected interruptions. Interfaces with IOS for loading operations.	IGFMCHE0	SYS1.LINKLIB
IGFMCHE1 (IGCR207B)	Console Write Interface with system WTO routine.	IGFMCHE1	SYS1.SVCLIB
IGFMCHE2 (IGCR107B)	Error Recorder Write error records to SYS1.LOGREC.	IGFMCHE2	SYS1.SVCLIB
IGFMCHE3	Emergency Recorder Writes error records when the system is unable to continue.	IGFMCHE3	SYS1.SVCLIB
IGFMCHF0	MCH Initialization Loads and initializes the MCH Nucleus during IPL/NIP.	IGFMCHF0	SYS1.LINKLIB
IGFMCHF1	PDAR 1 Analyzes program damage and passes control to appropriate PDAR module.	IGFMCHF1 IGFMVTF1 IGFMFTF1	SYS1.SVCLIB SYS1.SVCLIB SYS1.SVCLIB
IGFMCHF2	PDAR 2 Determines which part of the system has been affected by intermittent main storage errors and indicates appropriate action.	IGFMCHF2 IGFMVTF2 IGFMFTF2	SYS1.SVCLIB SYS1.SVCLIB SYS1.SVCLIB
IGFMCHF3	PDAR 3 Determines which part of the system has been affected by solid main storage errors or SPF key errors and indicates appropriate action.	IGFMCHF3 IGFMVTF3 IGFMFTF3	SYS1.SVCLIB SYS1.SVCLIB SYS1.SVCLIB
IGFMCHF5	PDAR Terminator Prepares the task to be either abnormally terminated or set nondispatchable.	IGFMCHF5	SYS1.SVCLIB
IGFMCHF6	TSO Subsystem Analysis To determine whether the machine check occurred in the subsystem area, and to terminate or assist recovery of the subsystem as circumstances warrant.	IGFMCHF6	SYS1.SVCLIB

Module/ CSECT Name	Module Name/Major Functions	Entry Point	Library
IGFMCH40	Soft Machine-Check Handler (145) Contains mode-handling function. Prepares recovery report for SYS1.LOGREC data set. Terminates MCH.	IGFMCH40	SYS1.SVCLIB
IGFMCH41	Preliminary Error Analysis Determines the recovery strategy for MCH based on the interruption code and MCH Common Area.	IGFMCH41	SYS1.SVCLIB
IGFMCH50	Soft Machine-Check Handler (135) Contains mode-handling function. Prepares recovery report for SYS1.LOGREC data set. Terminates MCH.	IGFMCH50	SYS1.SVCLIB
IGF13501	Machine Status Control (135) Permits the operator to control the mode of recording soft machine-check interruptions and to display the current mode status.	IGF13501	SYS1.SVCLIB
IGF29701	Machine Status Control (145) Permits operator to control the mode of recording soft machine-check interruptions in main and control storage.	IGF29701	SYS1.SVCLIB

APPENDIX A: MCH MESSAGE TABLE AND WAIT STATE CODES

Figure 37 lists the messages that are produced by MCH and the module that issues each message. The code, where shown, is a wait state code informing the operator of an error condition that caused the system to be placed in a wait state.

Message	Code	Scheduled By	Issued By
No Message - An MCI occurred while MCH was handling an MCI in MCH.	A02		
IGF001W/S { P } { B } aaaaaa { U }	A06	IGFMCHF5	IGFMCHE0
IGF002W/S SUPVR MC	A05	IGFMCHF5 IGFMCHE3	IGFMCHE0
IGF004W [S] PGM CHK [CPU ID { 1 } { 2 }]	A03	IGFMCHE0	IGFMCHE0
IGF006W/S I/O ERR C	A04	IGFMCHE0	IGFMCHE0
IGF012W/S MCI-CCH C	A0B	IGFMCHE3	IGFMCHE0
IGF013W/S CHAN ERR C	A0A	IGFMCHE3	IGFMCHE0
IGF015W/S UNEX ERR	A14	IGFMCHE0	IGFMCHE0
IGF020I ABEND		IGFMCHF5	IGFMCHE1
IGF023I LINK BLDL DLT		IGFMCHF5	IGFMCHE1
IGF024I xxxxxx SET NONDISPATCHABLE		IGFMCHF5	IGFMCHE1
IGF040I { TSO } SYS { NDISP } { TCAM } { ABEND }		IGFMCHF5	IGFMCHE1
IGF044I ECC SUCCESSFUL		IGFMCH40 IGFMCH50	IGFMCHE1
IGF045I MULTIPLE BIT ERROR		IGFMCH41	IGFMCHE1
IGF048I HIR SUCCESSFUL		IGFMCH40 IGFMCH50	IGFMCHE1
IGF050W/S TOD ERROR	A16	IGFMCHE0	IGFMCHE0
IGF051I HRT ERROR		IGFMCHE0	IGFMCHE1
IGF053I MODE STATUS-ECC { QUIET } HIR { QUIET } { RECORD } { RECORD } COUNT-nn THRESHOLD		IGF13501	IGF13501

Figure 37. MCH message table (Part 1 of 2)

Message	Code	Scheduled By	Issued By
IGF054E SYS1.LOGREC FULL		IGFMCHE2 IGFMCHE3	IGFMCHE1
IGF055I QUIET MODE		IGFMCH40	IGFMCHE1
IGF055I QUIET MODE ECC		IGFMCH50	IGFMCHE1
IGF055I QUIET MODE ECC, HIR		IGFMCH40 IGFMCH50	IGFMCHE1
IGF056I I/O ERROR IN RECORDER		IGFMCHE2	IGFMCHE1
IGF057E DISK FORMAT ERROR		IGFMCHE2	IGFMCHE1
IGF060E SYS1.LOGREC NEAR FULL		IGFMCHE2	IGFMCHE1
IGF061I MODE SWITCH COMPLETE		IGF29701 IGF13501	IGF29701 IGF13501
IGF063I SVC BLDL DLT		IGFMCHF5	IGFMCHE1

Figure 37. MCH message table (Part 2 of 2)

APPENDIX B: MACHINE-CHECK INTERRUPTION CODE

The bit positions of the indicators in the machine-check interruption code are illustrated in Figure 35. The machine-check interruption code includes information about the type and severity of the error, the validity of the various fields that are stored, and the validity and length of the extended logout. The following describes the contents of the machine-check interruption code.

Bit 0 - SD

System Damage -- This bit is set whenever interruptions may have been lost or damage has occurred that cannot be isolated to one or more of the less severe machine check damage types, either internal or external.

Bit 1 - PD

Instruction Processing Damage -- This bit is set when the extent of the damage is limited to an executed instruction or its associated operands.

Bit 2 - SR

System Recovery -- This bit indicates that errors were detected but have been successfully recovered without loss of system integrity. See Bit 17.

Bit 3 - TD

Timer Damage -- Damage has occurred to either the timer or to location 80.

Bit 4 - CD

Time-of-Day Clock Damage -- Damage has occurred to the time-of-day clock.

Bit 5 - ED

External Damage -- (Not used for Model 145.) Indicates that a channel, channel controller, switching unit or other unit external to the CPU or to a storage unit has been damaged during operations not directly associated with the CPU. ED is used to report damage of this type only when the more conventional reporting procedures, such as I/O interruption, are unavailable or are impractical.

Bit 6 - Reserved

Bit 7 - AC

Automatic Configuration -- (Not used for Model 145.) A buffer page in the CPU has been disabled by the hardware. Operations will continue, but with decreased performance.

Bit 8 - W

Warning -- (Not used for Model 145.) Damage is impending to some part of the system; for example, loss of power or loss of cooling.

Bits 9 through 13 - Reserved

Bit 14 - B

Backup -- The backup bit indicates that the machine state at the point of interruption has been restored to a hardware checkpoint state prior to the occurrence of error; that is, the PSW, registers, and storage reflect a valid state either at the beginning of the instruction in error or some prior instruction. If the backup bit is zero, a valid instruction address points to an instruction beyond the error.

Bit 15 - D

Delayed -- This bit indicates that some or all of the information stored as a result of this interruption was delayed in being reported because the interruption type was masked off for the duration of one or more instructions.

Bit 16 - SE

Storage Error Uncorrected -- Indicates that a reference to storage resulted in the detection of damaged data that could not be corrected.

Bit 17 - SC

Storage Error Corrected -- Indicates that a reference to storage resulted in the detection of an error that was subsequently corrected. Bits 2 and 17 are set on when the frequency of error corrections for single bit errors exceeds the limit set by hardware (Error Frequency Limit Overflow).

Bit 18 - PE

Protection Storage Error Uncorrected -- Indicates that a reference to the Storage Protection Key resulted in the detection of an uncorrectable error in the key in storage. The keys in storage are not checked for errors during storage references when the PSW key is zero.

Bit 19 - Reserved

Bit 20 - WP

PSW Validity -- Indicates that bits

12-15 of the machine check old PSW are valid.

Bit 21 - MS

PSW Masks and Key Validity -- Indicates that all PSW bits other than Interruption Code, ILC, AMWP, IA, CC, and Program Mask of the machine check old PSW are valid.

Bit 22 - PM

Program Mask and Condition Code Validity -- Indicates that the program mask and condition code in the machine check old PSW are valid.

Bit 23 - IA

Instruction Address Validity -- Indicates that the instruction address in the machine check old PSW accurately reflects the point in the instruction sequence at which the interruption occurred. Note that the instruction location at interruption and the instruction location at the time of the error may not be the same. If backup has been indicated, a valid instruction address will point to the instruction in error or prior to the error. If backup is not indicated a valid instruction address will point to an instruction following the error.

Bit 24 - FA

Failing-Storage Address Valid -- Indicates that the failing-storage address is valid.

Bit 25 - RC

Region Code Valid -- (Not used for the Model 135.) Indicates that a valid region code has been stored.

Bit 27 - FP

Floating-Point Registers Valid --

Indicates that the contents stored in the floating-point register save area are the same as the contents of the registers at the point of interruption.

Bit 28 - GR

General Registers Valid -- Indicates that the contents stored in the general register save area are the same as the contents of the registers at the point of interruption.

Bit 29 - CR

Control Register Validity -- Indicates that the contents stored in the control register save area accurately reflect the condition of the control registers at the time of interruption.

Bit 30 - LG

Log Valid -- (Not used for the Model 135.) Indicates that the CPU extended log information was correctly stored.

Bit 31 - ST

Storage Logical Validity -- Indicates that the contents of those storage locations that are modified by execution were restored to their contents at the point of interruption.

Bits 32 through 47 - Reserved

Bits 48 through 63

CPU Extended Log Length -- (Not used for the Model 135.) This field indicates the length in bytes of the information stored in the extended log area, starting at the location specified by the CPU extended log pointer in control register 15. On a machine-check interruption when no logout occurs, this field is set to zero.

Indexes to program logic manuals are consolidated in the publication IBM System/360 Operating System: Program Logic Manual Master Index, GY28-6717. For additional information about any subject listed below, refer to other publications listed for the same subject in the Master Index.

Where more than one reference is given, the major reference is first.

- abbreviated record (ABREC) 20,109,111-112
- ABEND 6,57-58
- Abnormal End Appendage routine 50
- ABREC (abbreviated error record) 20,109,111-112
- allocation
 - of auxiliary storage 7-8
 - of data areas by NIP 47-48
- altering PSWs
 - I/O new 17
 - machine-check new 14,19
 - program new 14
- analysis of the hardware error
- assessment of damage 19
- automatic recovery features 1
- auxiliary storage requirements 7-8

- buffer build area, record 109,111-112
- buffering and formatting 51-53

- CCH, interface to and recording for 23-24
- Channel-Check Handler, interface 23-24
- channel error record 24
- clock, time-of-day 17
- codes, machine-check
 - interruption 17,125-126
- common area 104-109,7,14
- communications 10
 - area 3,10
 - between modules 8,10
- Console Write routine, IGFMCH1
 - flowchart 98
 - module description 59
- control, passing of by transient modules 18,8,9,14,17
- control module 7
- control register 14 3
- control register 15 3,7
- control registers 3
- control storage malfunctions 5,17
- CPU
 - malfunctions 4,5
 - retry 1,10,17,19
 - successful 17
- CTFIELD 109

- damage assessment 19
 - field buffer area 113
- damage report (hard machine-check interruption) 1
- degraded state of operation 14
- dependent common area 104
- disabled for interruptions 10-14

- ECC (Error Checking and Correction) 1,3,10,17
 - successful 17
 - validity checking 3
- Emergency Recorder, IGFMCH3
 - flowchart 99-101
 - module description 59
- emergency recorder module (IGFMCH3) 59
- emergency recording 20-23
- enable hard errors 14
- error-on-error conditions 10-14
- error record
 - CCH 24
 - MCH 114,21-22
- Error Recorder, IGFMCH2
 - flowchart 94-97
 - module description 58-59
- error recording 20-23
- error recovery
 - levels of 6-7
 - types 1,6-7
- errors, unexpected 118
- exercising a location 19
- extended logout 112-113,3,7,10,14,17
 - mask bit 3

- fixed area (Permanent Storage Allocation) 3
- fixed logout 112,7,10,14
- fixed storage locations 3
- flowcharts 61-103
- formatting the error record 20-23

- hard errors 1,3-4,11,14
- hard machine check 1,3-4,11,14
- hard stop 14
- hardware
 - error analysis 17-19
 - malfunctions, types 17
 - recovery features of Models 135 and 145 1
- high-resolution timer 17
- history of executed modules 118-120

- IGFIORTN, Module Loader 49
- IGFLOAD, I/O Initialization 49
- IGFMCH0, MCH Nucleus
 - flowchart 64-72
 - module description 48-50
- IGFMCH1, Console Write routine

- flowchart 98
- module description 59
- IGFMCH2, Error Recorder
 - flowchart 94-97
 - module description 58-59
- IGFMCH3, Emergency Recorder
 - flowchart 99-101
 - module description 59
- IGFMCHF0, MCH Initialization
 - flowchart 62-63
 - module description 47-48
- IGFMCHF1 (see IGMVTF1 or IGMFTF1)
- IGFMCHF2 (see IGMVTF2 or IGMFTF2)
- IGFMCHF3 (see IGMVTF3 or IGMFTF3)
- IGFMCHF5, PDAR Terminator
 - flowchart 91-92
 - module description 57-58
- IGFMCHF6, TSO Subsystem Analysis
 - flowchart 93
 - module description 58
- IGFMCH40, Model 145 Soft Machine-Check Handler
 - flowchart 73-76
 - module description 51-53
- IGFMCH41, Preliminary Error Analysis
 - flowchart 81
 - module description 53-54
- IGFMCH50, Model 135 Soft Machine-Check Handler
 - flowchart 77-80
 - module description 50-51
- IGFMFTF1, MFT System Analysis 1
 - flowchart 87-88
 - module description 56
- IGFMFTF2, MFT System Analysis 2
 - flowchart 89
 - module description 56-57
- IGFMFTF3, MFT System Analysis 3
 - flowchart 90
 - module description 57
- IGMVTF1, MVT System Analysis 1
 - flowchart 82-83
 - module description 54
- IGMVTF2, MVT System Analysis 2
 - flowchart 84-85
 - module description 54-55
- IGMVTF3, MVT System Analysis 3
 - flowchart 86
 - module description 55
- IGF13501, Model 135 Machine Status Control
 - flowchart 103
 - module description 59-60
- IGF29701, Model 145 Machine Status Control
 - flowchart 102
 - module description 59
- independent common area 104-109
- inhibit termination on 61
- inhibit termination requested 61
- initialization
 - by MCH nucleus 10-17,47-48
 - by NIP 47-48
- instruction
 - processing damage 17-19
 - retry 1,17
 - unretryable 1,19
- interface with CCH 23
- intermittent errors 19
- interruption code, machine-check 125-126
 - common settings 118-119
 - location 3
 - some major fields 17
- interruptions disabled 10-14
- I/O
 - and module loading 9,14-18
 - communications area 3
 - Initialization, IGFLOAD 49
 - interruption 17
 - new PSW 17
 - Supervisor 14-17,8
- job termination 6-7,19-20,50-53
- key, damage to SPF 17,19
- levels of error recovery 6-7
- Link Pack Area, error occurs in 19
- loading
 - successor modules 8,9,14,17,18
 - transient modules 8,9,14,17,18
- logic of MCH 10
- logout 10
- lost-record counter 52
- machine-check
 - interruption code 125-126
 - location of 3
 - some major fields 17
 - new PSW 14,19
 - old PSW 10-14
 - subclasses 17
- machine circuitry 10,12,13,17
- machine malfunctions
 - sources of 1,2,17
 - types of 17
- Machine Status Control (Model 135), IGF13501
 - flowchart 103
 - module description 59-60
- Machine Status Control (Model 145), IGF29701
 - flowchart 102
 - module description 59
- main storage
 - malfunctions 5,17,19
 - requirements 7
- malfunctions, types of hardware 17
- MCH
 - common area 7,11
 - error record 114,112,21-22
 - error recovery 6
 - history table 118-120
 - independent common area 104-109
 - Initialization, IGMCHF0
 - flowchart 62-63
 - module description 47-48
 - Nucleus, IGMCH20
 - flowchart 64-72
 - module description 48-50
 - nucleus area 17,7
 - resident area 7
 - transient area 7,9,14,17,18
- MCHDEB, fields of 107

MCHINLOG 14,107
MCHINTEL, fields of 107
MCHIOB, fields of 108
MCHLSUM, fields of 108
MCHNXIDS table 49,108
MCHTRS table 49,109
message table 123-124
MFT System Analysis 1, IGFMTF1
 flowchart 87-88
 module description 56
MFT System Analysis 2, IGFMTF2
 flowchart 89
 module description 56-57
MFT System Analysis 3, IGFMTF3
 flowchart 90
 module description 57
MODE command 5-6,59-60
mode, automatic switching of 5-6,4,50
mode handling for the Model 135 50,5-6
mode handling for the Model 145 51,6
model-dependent common area 104
modes, control of 3-5
modes of recovery operation 4-5
module
 directory 121-122
 loader 48-49,9,16-18
 loading 8,9,16-17,18
 scheduler 49
multiple-bit storage errors 3,17,19
multiple errors 17
MVT System Analysis 1, IGFMTF1
 flowchart 82-83
 module description 54
MVT System Analysis 2, IGFMTF2
 flowchart 84-85
 module description 54-55
MVT System Analysis 3, IGFMTF3
 flowchart 86
 module description 55

NIP (Nucleus Initialization Program)
 initialization of MCH 47-48
Normal End Appendage routine 50
normal recording procedures 19-20
Nucleus, MCH (IGFMCH0) 10-17,48-50
nucleus area 7

operation diagrams 25-46
operator communications 5-6,10
overlay routines, area for 7,8,9,14-17
overlay structure of MCH 9

parity checking 1-3
PDAR
 control and action bytes 110-111
 modules 19
 (see also MFT and MVT System
 Analysis)
 Terminator, IGFMFH5
 flowchart 91-92
 module description 57-58
physical characteristics 7
Preliminary Error Analysis (PEA),
IGFMCH41 13
 flowchart 81

 module description 53-54,17
 priority of machine-check
 interruptions 10-14
 program-check new PSW 14
 program damage
 assessment and repair modules (PDAR) 19
 recovery 19
 procedures 19
 program termination 19-20,6
PSA (Permanent Storage Allocation or fixed
area) 3,14
PSW
 altering 14,19
 I/O new altered 17
 saving the old 14
purpose of the Machine-Check Handler 1

quiet mode 3-6

receiving control 10
record buffer build area 109,112
recording
 channel-check records 23-24
 emergency 20-23
 error records 20,21-22,24
 mode 3-4
recording and termination 19-23
recovery 6,19
 design for Models 135 and 145 1
 report (soft machine-check
 interruption) 1,8
register
 control 3
 conventions 118
repair, system 7,19
resident area 7
restart, system-supported 6
retry
 CPU 1,17,19
 fails 1,19
 successful 17
 unsuccessful 17-19
retry on 61

saving the environment 14,47-48
scheduling successor
 modules 48-49,18,14-17
second errors 10-14
sequence of executed modules 118-120
short error record 114,112,20
SHUT (Special Handler for Unusual Termina-
tions) routine 14,48
single-bit errors 1-6
soft errors 27,1-6,10,12-13
soft machine check 12-13
Soft Machine-Check Handler (IGFMCH40)
 flowchart 73-76
 module description 51-53
Soft Machine-Check Handler (IGFMCH50)
 flowchart 77-80
 module description 50-51
solid errors 11,4,19
solid, single-bit errors 1-4
SPF key 17-19
storage errors 17

Storage Protect Feature (SPF) 17,19
storage requirements for MCH 7
subclasses of machine-checks 17
subsystem data area 113,7
successor modules, scheduling and
loading 8,9,14-17,18,48-50
supervisor area, error occurs in 19
switching modes, automatic 3-6
system
 damage 17,6,19
 disabled for interruptions 10-14
 recovery 6-7,19
 repair 6-7,19
 termination 6-7,17,19
System Analysis
 of MFT modules 56-57
 of MVT modules 54-55
system-supported restart 6
SYS1.LOGREC 7,19-20
SYS1.SVCLIB 7

task termination 6,19
TCB (Task Control Block), setting it
 nondispatchable 19
tense 17
Termination 46,53,57-58

termination, task and system 6-7,19
termination necessary 61
threshold mode 3-6
Time of Day Clock Damage 17
Timer Damage 17
transient area 7,9,14-18
transient modules 18,8-9,14-17
TSO (Time Sharing Option) Subsystem
 Analysis, IGF MCHF6
 flowchart 93
 module description 58
types
 of error recovery 1-3,6-7
 of hardware malfunctions 17

unexpected errors 118
unretryable instruction 17
unsuccessful retry 17-19

validity bits 17,125-126

wait state 17,19
 codes 123-124
warm start (system-supported restart) 6

IBM System/360 Operating System
Machine Check Handler for the
IBM System/370 Models 135 and 145

Order No. GY27-7237-1

**READER'S
COMMENT
FORM**

Your views about this publication may help improve its usefulness; this form will be sent to the author's department for appropriate action. Using this form to request system assistance or additional publications will delay response, however. For more direct handling of such requests, please contact your IBM representative or the IBM Branch Office serving your locality.

How did you use this publication?

- | | |
|--|---|
| <input type="checkbox"/> As an introduction | <input type="checkbox"/> As a text (student) |
| <input type="checkbox"/> As a reference manual | <input type="checkbox"/> As a text (instructor) |
| <input type="checkbox"/> For another purpose (explain) _____ | |

Please comment on the general usefulness of the book; suggest additions, deletions, and clarifications; list specific errors and omissions (give page numbers):

Cut or Fold Along Line

What is your occupation? _____

Number of latest Technical Newsletter (if any) concerning this publication: _____

Please include your name and address in the space below if you wish a reply.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Your comments, please . . .

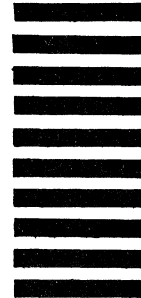
This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Fold

Fold

**First Class
Permit 40
Armonk
New York**

Business Reply Mail
No postage stamp necessary if mailed in the U.S.A.



Postage will be paid by:

**International Business Machines Corporation
Department 636
Neighborhood Road
Kingston, New York 12401**

Fold

Fold



**International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)**

**IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)**

Cut or Fold Along Line

MCH for IBM S/370 Models 135 and 145 Printed in U.S.A. GY27-7237-1



**International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)**

**IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)**