SY27-7239-2
File No. S360/S370-37

Systems

OS/VS
Recovery Management Support Logic

OS/VS1 Release 3
OS/VS2 Release 1.6

IBM

Second Edition (January 1974)

This edition applies to release 3 of OS/VS1 and release 1.6 of OS/VS2 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. This edition makes SN27-1407 obsolete. Changes are continually made to the specifications herein; before using this publication in connection with the operation of IBM systems, consult the IBM System/360 and System/370 Bibliography, GA22-6822, and the current SRL Newsletter for the editions that are applicable and current.

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.

Copies of this and other IBM publication can be obtained through IBM Branch Offices.

A form for reader's comments appears at the back of this publication. Address any additional comments concerning the contents of this publication to: IBM Corporation, Programming Publications, Department 636 Neighborhood Road, Kingston, New York, 12401

PREFACE

This publication describes the operation and design of recovery management support programs that are part of OS/VS (virtual storage):

- Machine-Check Handler (MCH)
- Channel-Check Handler (CCH)
- Dynamic Device Reconfiguration (DDR)

These programs are designed to keep the system operational in the event of a processor, storage, channel, or device failure. (An additional recovery management support program, Alternate Path Retry, is described in the OS/VS I/O Supervisor PLM.)

This publication is divided into three parts: MCH, CCH, and DDR. Each part is independent of the others and may be used as a separate manual, although the parts share a table of contents and an index. Each part of the manual is divided into the following sections:

Section 1: Introduction summarizes the functions and operations of the program. This section also contains definitions needed to understand the rest of the part.

Section 2: Method of Operation describes the functions of the program in detail and relates the functions to the coding. The functions are described mainly through method of operation diagrams, which also refer to the coding.

Section 3: Program Organization describes the program from a module or routine point of view. Overall flow diagrams show the relationship of the modules; module descriptions show the logic of each module.

Section 4: Directory is a guide to the named areas of coding.

Section 5: Data Areas describes the fields of information used by the program.

Section 6: Diagnostic Aids contains techniques and information to use in determining the source of problems that arise in the program.

Appendix A: Missing Interruption Checker contains an explanation and method of operation diagram showing how recovery management checks for missing device interruptions.

This publication is intended to be used as a guide to the program listings. The intended audience is system maintenance personnel, including customer engineers, programming support representatives, and system programmers.

This publication assumes an understanding of OS/VS, including paging and I/O operation. Information on these subjects can be found in introductory OS/VS publications as well as in:

OS/VS Supervisor Logic

OS/VS I/O Supervisor Logic

CONTENTS

OPERATION DIAGRAMS

LIST OF ABBREVIATIONS

APR	Alternate Path Retry
CCC	channel control check
CCH	Channel-Check Handler
CCW	channel command word
CDC	channel data check
CPU	central processing unit
CSECT	control section
CUA	channel and unit address
CVT	communications vector table
DSS	Dynamic Support System
ECC	error checking and correction
ECSW	extended channel status word
EREP	IGCEREP0 utility
ERP	error recovery procedure
ERPIB	error recovery procedure interface block
EXCP	execute channel program
HIO	halt input/output
HIPO	hierarchy and input-processing-output (diagram)
HIR	hardware instruction retry
ICC	interface control check
IOB	input/output block
IOS	Input/Output Supervisor
IPL	Initial Program Loading routine
MCH	Machine-Check Handler
MOD	method of operation (diagram)
NIP	Nucleus Initialization Program
OBR	Outboard Recorder
OS/VS	Operating System/Virtual Storage
PQA	paging queue area
PSW	program status word
RQE	request queue element
RVT	recovery vector table
SEREP	System Environment Recording and Editing Program
SHUT	Special Handler for Unusual Termination
SIO	start input/output
SMCH	Soft Machine-Check Handler
SPF	storage protect feature
SQA	system queue area
SVC	supervisor call
SYSRES	system residence
TIO	test input/output
TSO	time-sharing option
UCB	unit control block
WTO	Write-to-Operator

PART 1: MACHINE-CHECK HANDLER

PURPOSE OF MCH

MCH (Machine-Check Handler) minimizes the computing time lost due to machine malfunctions. MCH does this by correcting certain malfunctions and by producing diagnostic records and messages to help system maintenance personnel find the cause of the problem.

OVERVIEW OF MCH

A machine malfunction can originate from the CPU, real storage, or control storage. When any of these fails to work properly, the machine attempts to correct the malfunction (see Figure MCH-1). If the machine corrects the malfunction, it notifies MCH by a machine-check interruption. MCH records the fact that the machine has failed to operate properly (except when the machine is in quiet mode; see "Recording and Quiet Modes" in this section.) When the machine-check interruption occurs, the CPU logs out fields of information in real storage detailing the cause and nature of the error. The model-independent data is stored in the fixed logout area, and the model-dependent data is stored in the extended logout area. MCH uses these fields to analyze the error and to produce an error record.

If the machine fails to correct the malfunction through hardware recovery facilities, the machine still causes a machine-check interruption and logs out. In this case, however, the fixed logout contains an interruption code indicating an unsuccessful recovery attempt. MCH then analyzes the data and attempts to keep the system as fully operational as possible. The location and cause of the malfunction determine what action MCH takes: (1) resumes system operations; (2) resumes system operations at the expense of the task that was interrupted; (3) places the system in the wait state. MCH always records as much information about the error as possible.

SYSTEM/370 RECOVERY FEATURES

Because the operation of MCH depends on certain recovery actions taken by the machine and on the information given to it by the machine, some of the features of the machine are described here. For a more complete description, see System/370 Principles of Operation.

Automatic Recovery Features

System/370 has two built-in methods of recovering from machine malfunctions: HIR (hardware instruction retry) and ECC (error checking and correction). Whenever possible, HIR and ECC recover from machine malfunctions without assistance from MCH. The HIR 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. ECC detects and often corrects invalid data fetched from real and control storage.

Hardware Instruction Retry

HIR microprogram routines automatically retry CPU errors. These routines save source data before it is altered by an operation. When an error is detected, a microprogram 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 by HIR, the error is considered permanent.

Error Checking and Correction

ECC routines check the validity of data fetched from real and control storage. Data enters and leaves real or control storage through a storage adapter unit. This unit checks each doubleword for correct parity in each byte. If a single-bit error is detected, it is corrected. The corrected doubleword is then sent back into real or control storage and on to the CPU (except on the Model 145). MCH is notified by a machine-check interruption and finds the address of the error in the fixed logout.

ECC does not correct multiple-bit errors, but notifies MCH by a machine-check interruption. MCH finds the address of the error in the fixed logout and attempts to recover from the error.

Fixed Storage Areas

There are four fixed areas of real storage for System/370:

- Permanent storage assignment (decimal locations 0-159)

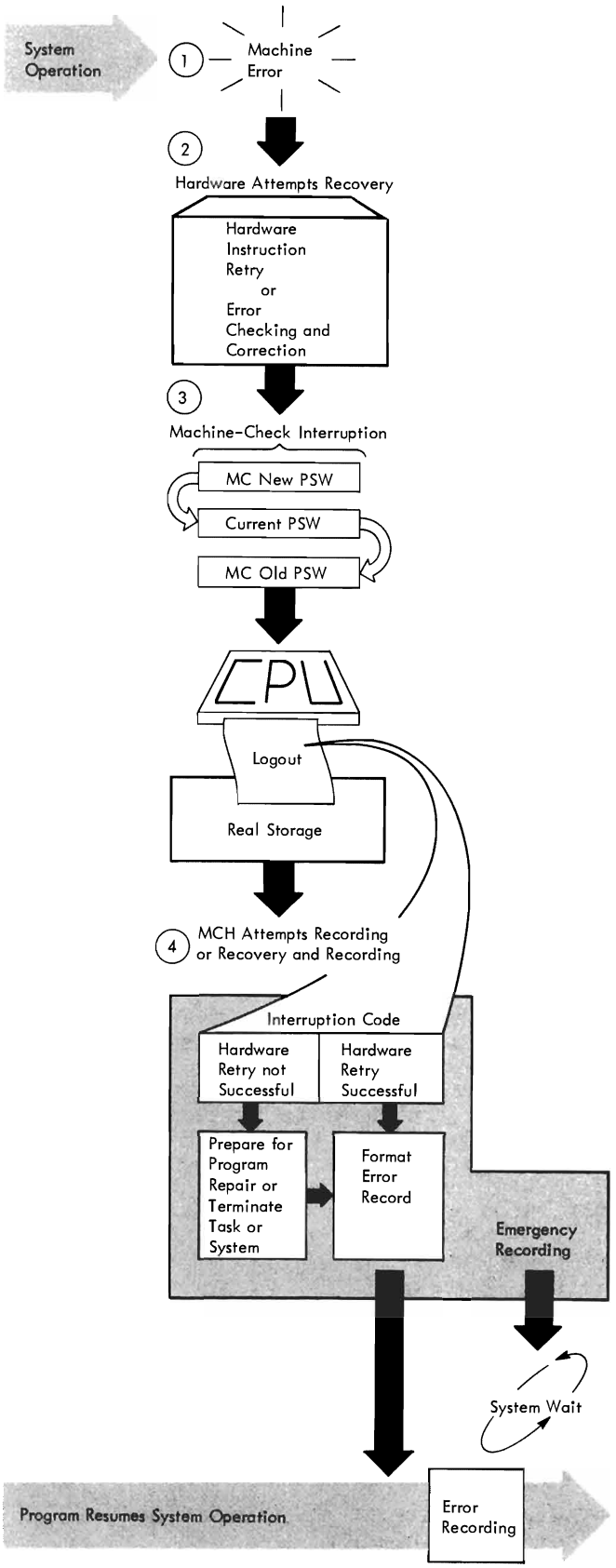


Figure MCH-1. MCH overview

- I/O communications area (decimal locations 160-191)
- Fixed logout area (decimal locations 232-511)
- Extended logout area (begins at location 512 unless the address in control register 15 specifies otherwise)

The fixed storage areas are described in MCH Section 5.

Fixed Logout Area

Data is put into the fixed logout area when any machine-check interruption occurs. This area contains data that is model independent, although not all models use every field. The fixed logout area contains the machine-check interruption code, which indicates the reason for the interruption; system status information; and the contents of the general-purpose, floating-point, and control registers at the time of the machine-check interruption. This data is processed by MCH.

Extended Logout Area

The extended logout area contains data that is model dependent. It begins at address 512 or at the address specified in control register 15. The length of the extended logout area is different for each model. If the extended logout mask bit (MCEL bit) in control register 14 is enabled, data is written into the extended logout area for all types of machine-check interruptions. This data is recorded by MCH in the SYS1.LOGREC data set.

The size of the logout areas for each CPU model is shown in Figure MCH-2.

Control Registers

MCH uses two control registers for loading and storing control information. Control register 14 contains mask bits that (1) indicate whether certain conditions can cause machine-check interruptions and (2) control conditions under which an extended logout can occur. Control register 15 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 is used to move information from real storage to control registers; STORE CONTROL is used to move information from control registers to real storage.

	Model 135	Model 145	Model 155II	Model 158	Model 165II	Model 168
Fixed Logout	280 bytes	280 bytes	280 bytes	280 bytes	280 bytes	280 bytes
Extended Logout	none	192 bytes	672 bytes	672 bytes	1416 bytes	1416 bytes

Figure MCH-2. Fixed and extended logout area sizes

Figure MCH-3 shows control register 14. System/370 Principles of Operation contains a detailed description of the use of the control registers and shows the masks used to prevent logouts and machine-check interruptions.

Recording and Quiet Modes

Hardware instruction retry and error checking and correction can operate in two modes: recording and quiet. In recording mode, a machine-check interruption occurs for each malfunction. In quiet mode, a machine-check interruption occurs only for malfunctions that were not corrected by HIR or ECC. Recording mode is the normal mode of operation for Models 155 and 165II. All other supported CPU models normally run in recording mode for HIR and quiet mode for ECC. Quiet mode is usually used when there is a large number of corrected errors. When in recording mode, a threshold value of 12 is used by MCH to define the number of interruptions allowed for corrected errors. When the number of corrected errors exceeds this threshold, quiet mode is entered automatically, and no more interruptions occur for corrected errors. The operator can use the MODE command to switch from recording mode to quiet mode or from quiet mode to recording mode. The current mode can be determined by using the STATUS operand of the MODE command.

MCH ERROR RECOVERY

Machine-check interruptions can be classified as either soft or hard. A soft machine-check interruption indicates to MCH that the hardware has already corrected the error. MCH makes a record of the event and the circumstances under which it occurred. A hard machine-check interruption indicates to MCH that the hardware has not corrected the malfunction and MCH must try to recover from it.

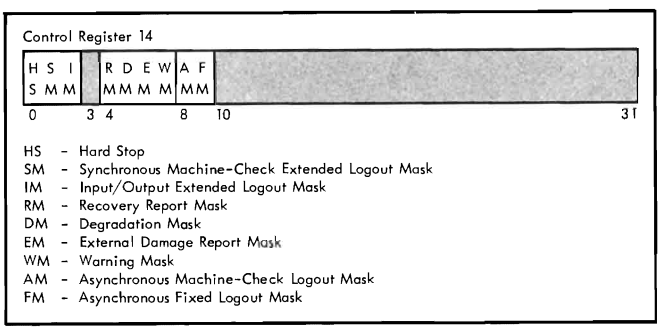


Figure MCH-3. Control register 14

Soft Interruptions

After the machine has recovered from a machine check (assuming that it is operating in recording mode), a machine-check interruption occurs. From the machine-check interruption code, MCH determines that the error has been corrected, and therefore recovery by MCH is not necessary. MCH must, however, check whether a change to quiet mode is necessary and prepare a recovery report.

Hard Interruptions

When HIR or ECC facilities have not been able to correct an error, MCH tries to enable the system to continue processing. A hard machine-check interruption occurs on any of the following conditions:

- System damage
- Instruction processing damage
- Timing facility damage
- Buffer damage
- High resolution timer damage (location 80)
- External damage (when associated with timing damage)
- Power warning.

Depending on the type of malfunction that caused the interruption and the status of the system at the time of the interruption, MCH (1) resumes system operations; or (2) resumes system operations at the expense of the task that was interrupted; or (3) places the system in the wait state.

Levels of Error Recovery

Recovery from machine malfunctions can be divided into four categories:

Task recovery
System recovery
System-supported restart
System repair

Task recovery is most acceptable; system repair is least acceptable.

Task Recovery

Task recovery is transparent to the interrupted program. This type of recovery can be made by either HIR or ECC, or by MCH. Task recovery by MCH is made by correcting SPF (storage protection feature) keys and intermittent errors in real storage.

System Recovery

System recovery is attempted when task recovery is impossible. System recovery is the continuation of system operations at the expense of the task (and subtasks if attached) in which the error appeared. The task containing the error is terminated either by abnormally ending the task or by marking it nondispatchable. System recovery can take place only if the task in question is not required for system operation.

System-Supported Restart

System-supported restart is tried after task and system recovery have failed or when they could not be tried. This occurs when a task required for system operation contains an error. The operator is notified that the system cannot continue to operate and has been placed in a wait state. He must then reload the system.

Note: System-supported restart is a warm-start operation; that is, the remaining information on the job queue is usable. The job queue does not have to be rebuilt.

System Repair (Manual)

System repair also occurs when the system has been placed in a wait state, but it requires the services of maintenance personnel and takes place at the discretion of the operator. Usually, the operator will have already tried to recover by system-supported restart one or more times without success. System repair is required, for example, if a hard error occurs so frequently that system-supported restart is not successful.

This section describes the functions of MCH and how MCH performs these functions. Many references are provided to material in other sections of this publication and to the MCH program listing. Two types of information are presented:

- Text that discusses the design of MCH.
- Method of operation diagrams that show the functions performed by MCH in the order that they occur. The module name and label of each processing step are included in the implementation notes for each diagram. These diagrams can be used for quick reference when reviewing MCH functions. Diagram MCH-1 shows how the method of operation diagrams for MCH are organized.

THE LOGIC OF MCH

MCH has two basic paths of execution: one for malfunctions that have been corrected by HIR or ECC (soft interruptions), and one for malfunctions that have not been corrected by those facilities (hard interruptions).

For uncorrected failures, MCH goes through four major operations:

Preservation of the system environment
Analysis of the malfunction
Program damage recovery
Recording and termination

For malfunctions that have been corrected by HIR or ECC, the program damage recovery step is omitted. In addition to the four steps mentioned above, MCH controls whether the machine operates in recording or quiet mode.

COMMUNICATIONS

In order to process a machine-check interruption, MCH must have certain data describing the failure. For this purpose, the CPU provides a logout which gives MCH the information needed to properly analyze the error and determine how to handle it. After the model-independent logout is produced, MCH moves it into the MCH record buffer, where it is used by the transient modules of MCH to communicate with each other. (Since MCH uses an overlay scheme, it is necessary to store information in an area that will not be overlaid by the next module brought into the transient area.) The MCH Nucleus also uses the common area to store and retrieve data about the failure. The common area is described in MCH Section 5.

PRESERVATION OF THE SYSTEM ENVIRONMENT

MCH normally receives control through a machine-check interruption. If another machine-check interruption were to occur while MCH was being executed, control would be returned to the beginning of MCH. Thus the first machine-check interruption would never be processed, since the information about it in the logout areas 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 the following emergency provisions:

- It disables for machine-check interruptions which have already been corrected by HIR or ECC, so that only serious errors can interrupt MCH. If the error being handled has not been corrected, there is no need to interrupt MCH to report a less serious error. If the error being handled has been corrected, there is no need for one corrected error to have priority over another.
- It saves the contents of the fixed logout area in the MCH record buffer. If a hard machine-check interruption occurs now, the original data is not overlaid. Also, extended logouts are prevented by setting a mask in control register 14, so that the extended logout area is not overlaid.
- It saves the machine-check old PSW. Then, if a second error occurs, causing the current PSW to replace the old PSW, control can be returned to the program that was interrupted first (provided that the error was corrected with the original system status intact).
- It alters the address in the machine-check new PSW and the program-check new PSW to point to the MCH SHUT (Special Handler for Unusual Termination) routine. Then, if a second error occurs, control is passed to the SHUT routine, rather than to the beginning of MCH. This is done because a second machine-check interruption indicates that the

Error Condition	Special Circumstances	MCH Response
Hard on Soft	Error within MCH	Determines the severity of the error. For any error other than system damage, attempts 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	Attempts to record the original error and returns control to the system.
	Recovery not made from original error	Places the system in a disabled wait state and notifies the operator.
Note: MCH is not enabled for soft errors at any time during the processing of soft or hard errors. MCH is only reenabled for soft errors after the recorder gets control.		

Figure MCH-4. MCH responses to error-on-error conditions

error occurred during MCH execution. MCH would go into a loop if the machine-check new PSW were not altered and the second error occurred. Also, when an error occurs within MCH, MCH is operating in a degraded state and might not be able to recover from the original error.

- It enables hard machine-check interruptions to give them priority over soft machine-check interruptions.

There is always the danger of the machine failing immediately after MCH is entered, while the system is disabled for interruptions. If this happens, the machine comes to a hard stop, no instructions are executed, and no interruptions occur. System operation can be continued only by means of a system reset or IPL. Figure MCH-4 shows MCH responses to various error-on-error conditions.

MCH saves the fixed logout, extended logout, and the machine-check old PSW to protect them from a second interruption. Once the system has been reenabled for interruptions, MCH saves the permanent storage assignment. The four data areas are saved in the following locations:

- Fixed logout: address contained in the RVTINLOG field of the RVT.
- Extended logout: address found in control register 15.
- Machine-Check old PSW: MCHRPSW in the MCH common area.
- Permanent storage assignment: MCHPSA in the MCH common area.

Figures MCH-5 and MCH-6 show the MCH environment before and after initialization.

ANALYSIS OF THE FAILURE

To accurately assess the extent of the damage at the time of the machine-check interruption, MCH analyzes the machine-check interruption code. MCH identifies the type of error, where it occurred, and under what special circumstances it occurred, if any.

Figure MCH-7 shows the fields of the interruption code and which fields are used by each System/370 model. See MCH Section 5 for a complete description of the machine-check interruption code.

Some of the major fields of the machine-check interruption code are:

- Machine Check Subclasses: Bits 0 through 8, the subclasses, indicate the machine-check condition causing the interruption. On each interruption, at least one of these bits must be set. If more than one error has occurred, several bits may be set.
- Tense: This field indicates the timing of the interruption. For example, if bit 14 is set to 1, the machine-check old PSW contains the address of the instruction in which the error occurred. If bits 14 and 15 are set to 0, the machine-check old PSW contains the address of an instruction beyond the point of error.
- Storage Errors: This field indicates that the error was in real storage.

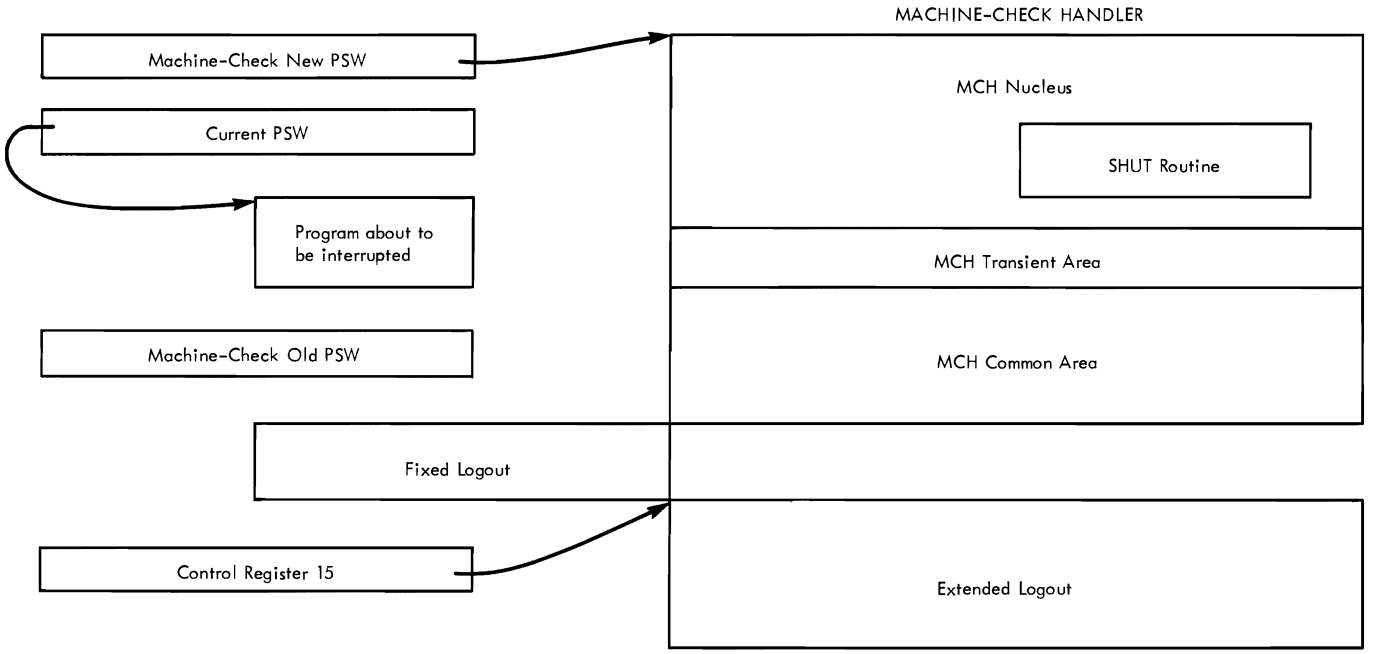


Figure MCH-5. MCH and environment before a machine-check interruption

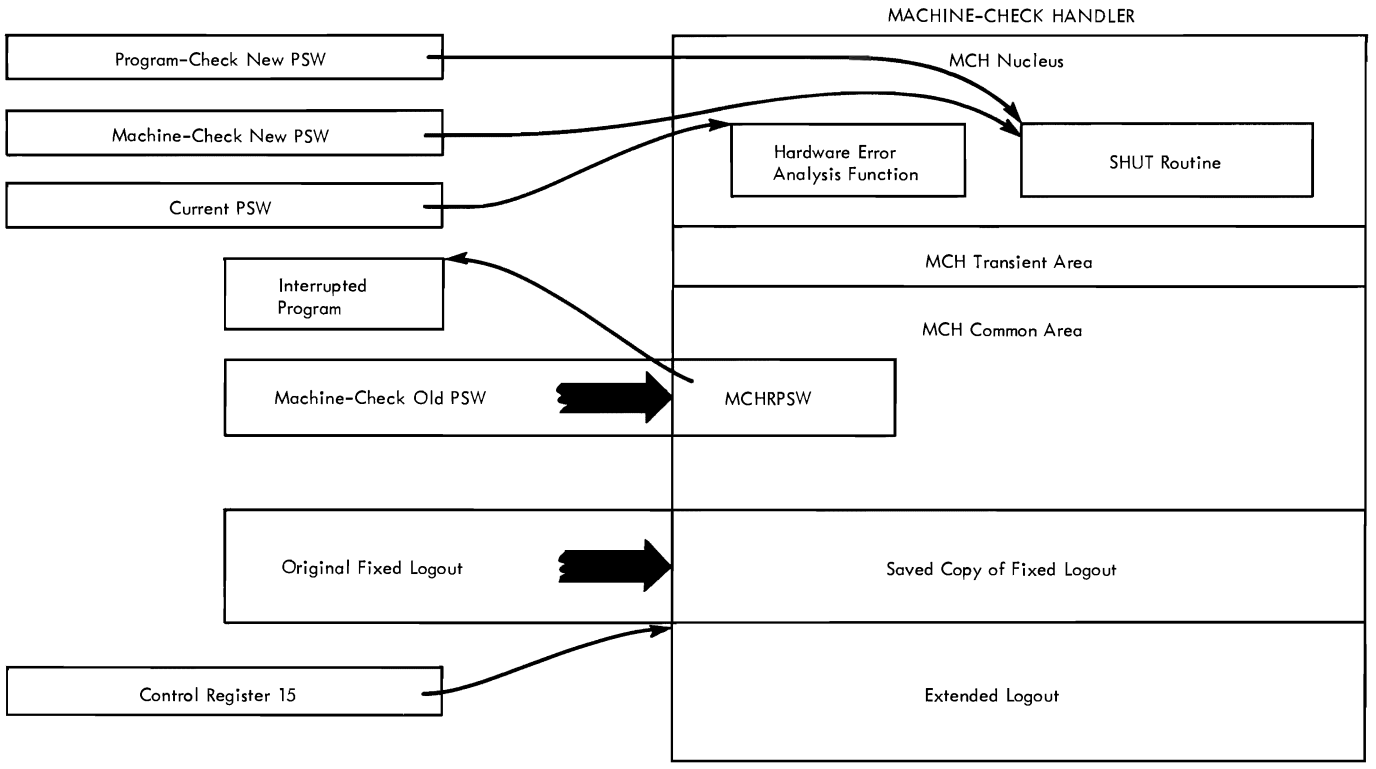


Figure MCH-6. MCH after the environment has been saved following a machine-check interruption

Machine-Check Interruption Code

Model	SD (0)	PD	SR	TD	CD	ED	DG (7)	W	B (14)	D	SE (16)	SC	KE	WP (20)	MS	PM	IA	FA	RC	FP (27)	GR	CR	LG	ST (31)	CT (46)	CC	Machine-Check Extended Logout Length (48–63)
135	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•		•	•	•	•	•			
145	•	•	•	•	•			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			
155	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
158	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
165	•	•	•		•	•	•			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
168	•	•	•		•	•	•	•		•	•	•	•	•	•	•	•	•		•	•	•	•		•	•	

Bits	Field
0-8	Subclass
14-15	Time of interruption occurrence
16-18	Storage errors
20-25, 27-31, 46, 47	Validity
6, 9-13, 19, 26, 32-45	Not assigned

Note: The fields of the machine-check interruption code are fully explained in MCH Section 5.

Figure MCH-7. Fields of the machine-check interruption code and which fields are used by each System/370 model

- Validity: The validity bits represent the various fields stored during machine-check interruption. Any bit that is 0 indicates that the associated data (general registers, condition code, etc.) has been affected by the error.

- Extended Logout Length: This field indicates the length in bytes of the extended logout area.

Types of Failures

The following types of failures can be identified 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 unretryable or unsuccessfully retried, or the damage resulted from a multiple-bit failure in real storage or an SPF key error.

- Hardware retry successful (soft error): The CPU instruction was successfully retried.

- ECC successful (soft error): A single-bit storage error was corrected by ECC.

- Time facility 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.

- Power warning: Loss of power to the system may be impending.

System Damage

System damage occurs when the circuitry or the microcode in the CPU has failed. Multiple-bit errors in control storage are included in this category. By presenting the failure 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 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 running. The first step is to determine from the interruption code the type of error that occurred.

There are three types of failures that are classified as instruction processing damage. Bit one, the instruction processing damage bit, is set to one in all cases. Related bit settings indicate the type of instruction processing damage that occurred.

- Processor damage: 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 retried the instruction but was not successful.

- Multiple-bit error in real storage: MCH 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 location. If another machine-check interruption occurs, the error is solid. Otherwise it is considered 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 a routine that services the expected interruption. The result of this analysis is placed in the MCH common area. Later, MCH uses this information to assess the damage to the task occupying that failing section of real storage.

- 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 routine that will service the expected machine-check interruption. A succession of fetches and stores using all 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 failure, MCH determines the effect of the failure on the operating system. The effect of the failure primarily depends on whether it occurred in the processor, real storage, or an SPF key. If there was a failure, of any type, on a model not supported by MCH, the path of execution that is taken is always the one described here under "Processor Failures."

Processor Failures

For a processor failure (or a failure in a model not supported by MCH) either MCH places the system in the wait state, or the ABTERM routine terminates the job-step task. MCH places the system in the wait state for one of two reasons:

- The failure affected an indispensable system task. If, in the machine-check old PSW, the system was disabled for I/O interruptions while in supervisor state at the time of the interruption, MCH assumes that critical system activity was in progress and that to continue system activity would only cause further damage to the system.

- The status of the system at the time of the failure is unknown. If the wait TCB is active, MCH cannot analyze further and must assume that the integrity of any further processing cannot be maintained.

If the task is not indispensable and the status of the system is known, MCH determines whether the task in control alone is damaged, in which case the task could be terminated without affecting system operations.

Real Storage Failures

Just as for processor failures, if the storage failure appears in an indispensable system task or if the status of the system is unknown, the system must be terminated. The system must also be terminated if:

- The failure was anywhere in the VS nucleus.

- The failure was in the SQA (system queue area), PQA (paging queue area), or LSQA.

If the failure was in none of these, MCH determines if it occurred in a pageable area, nonpageable area, or in a fixed page, and indicates in the page table that the page is invalid or unavailable. Later, when that page is referred to, a paging exception occurs and the paging supervisor brings in the back-up copy of the page.

This procedure occurs only if MCH determines that the page containing the failure has not been changed. If it has been changed, the task must be terminated by ABEND. If it has not been changed, the job is reassigned a new real page, and the instruction is automatically reexecuted. However, MCH still must determine what to do with the page containing the failure. For solid failures, MCH dequeues the real page from the table of available pages, making it permanently unavailable. For intermittent failures, MCH takes no further action.

SPF Key Failures

SPF keys that fail intermittently are reset to their original value to permit the system to continue operation. For solid errors, the same conditions that apply to storage errors apply to SPF key errors.

Figure MCH-8 shows the conditions and corresponding actions taken by MCH.

CONDITION	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Type of error																												
Storage	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x													
Storage Protect Key																x	x	x	x	x	x	x	x	x	x			
Processor																										x	x	x
Extent																												
Solid	x	x	x	x	x	x	x	x								x	x	x	x	x	x							
Intermittent						x	x		x	x	x	x	x	x	x							x	x	x	x			
Location																												
Pageable area	x	x	x	x	x					x	x	x	x	x		x	x	x				x	x					
Nucleus/SQA/PQA/LSQA						x													x				x	x				
Fixed page (nonpageable) area							x	x	x						x					x	x		x		x			
Status of PSW at MCI																												
Supervisor mode	x	x	x	x						x	x	x	x			x	x									x	x	
Problem program mode	x	x	x	x						x	x	x	x			x	x									x		x
Wait mode			x		x							x		x				x										
I/O masking																												
Enabled																										x		
Disabled																											x	x
Task nature																												
Critical							x													x							
Non-critical								x	x						x						x							
Storage state																												
Changed	x	x			x					x	x			x														
Unchanged			x	x								x	x															
Recursion in MCH								x													x							
ACTION TAKEN																												
Make real page permanently unallocatable	x	x	x	x	x											x	x	x										
Invalidate Page Table Entry	x	x	x	x	x					x	x	x	x	x		x	x	x										
Repair SPF key																						x	x	x	x			
Set task non-dispatchable								x													x							
Enter disabled wait					x	x	x							x				x	x	x							x	
ABEND affected task	x	x							x	x	x				x	x	x											
ABEND current task		x		x							x		x				x						x			x		x
Execute Emergency Recorder					x	x	x							x				x	x	x							x	
Queue record for Recorder	x	x	x	x				x	x	x	x	x	x		x	x	x				x	x	x	x	x	x		x
Continue system operation	x	x	x	x				x	x	x	x	x	x		x	x	x				x	x	x	x	x	x		x

Note: When more than one condition of a type is indicated (encircled x's), the action is the same no matter which condition represents the situation. For example, in column 6 both solid and intermittent are indicated to show that any storage error (solid or intermittent) in the Nucleus, SQA, PQA, or LSQA results in MCH placing the system in a disabled wait state after using the Emergency Recorder.

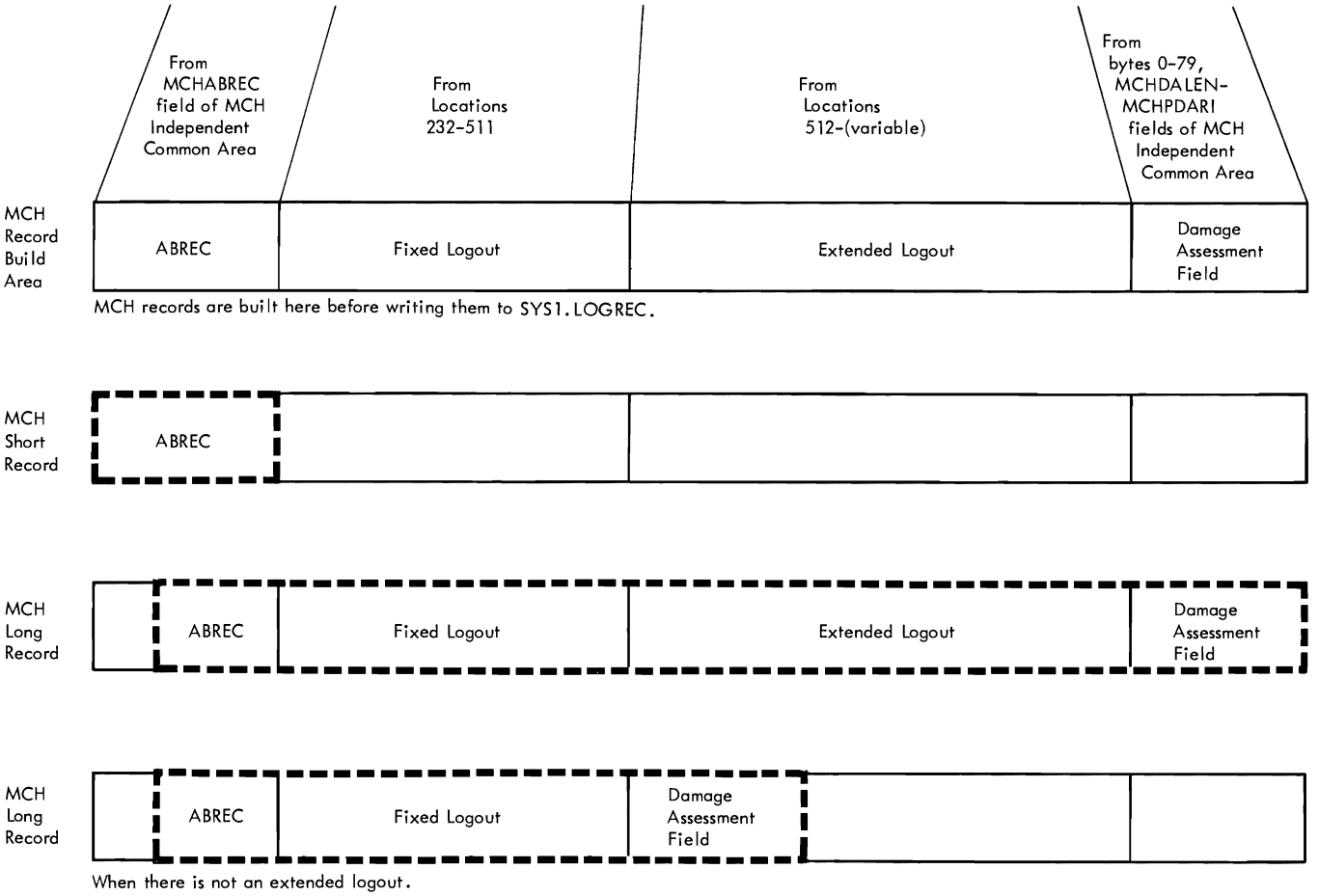
Figure MCH-8. Actions taken by MCH

RECORDING AND TERMINATION

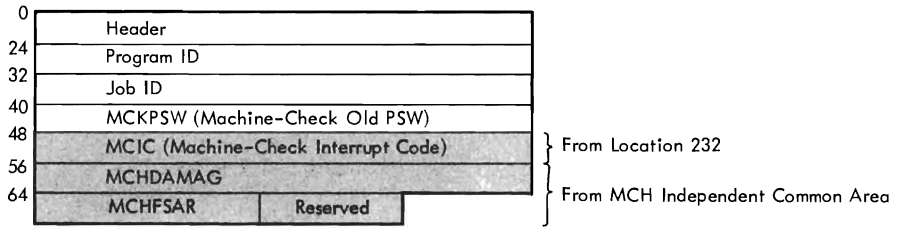
The recording function of MCH has two parts. The first is formatting an error record and later writing it into the SYS1.LOGREC data set. The second is emergency recording; that is, the recording MCH does when it has determined that system operation cannot continue.

Error Recording

MCH produces two kinds of error records: the short record and the long record; both are shown in Figure MCH-9. MCH formats the error record before it terminates its operations. Writing the record into SYS1.LOGREC takes place after MCH terminates to decrease the chances of a second machine-check interruption occurring while MCH is



A full ABREC is 70 bytes long and consists of:



The full ABREC is included in an MCH short record, but only the first 48 bytes are included in an MCH long record because the MCIC is part of the fixed logout, and MCHDAMAG and MCHFSAR are part of the damage assessment field.

Figure MCH-9. MCH record format

being executed. Further, the system is in quiet mode from the time MCH exits until the recording occurs.

MCH does the following to put a record into the SYS1.LOGREC data set:

- Formats the complete error record.
- Activates the system communications task in order for it to load the MCH Error Recorder.
- Terminates itself by relinquishing control to the operating system dispatcher or the interrupted program.

If the dispatcher gets control it dispatches the next ready task, which should be the system communications task. It brings in the MCH Error Recorder, which then writes the MCH records into the SYS1.LOGREC data set.

If another hard machine-check interruption occurs before the error record is written, the MCH Error Recorder writes the short record of the first interruption and the complete 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.

When more than three hard machine-check interruptions occur before the MCH Error Recorder has a chance to write the records into SYS1.LOGREC, the record having the lowest priority is overlaid. The "lost summary" field in the record is updated to show the location and severity of the lost record. The priority of machine checks is:

1. Hard unrecovered (task terminated)
2. Hard recovered (task recovered)
3. Soft

The MCHDAMAG field of the MCH error record reflects the error analysis and action taken by MCH. The field is model independent, and some bits do not apply to every model. Specifically, the buffer and control storage bits are not set for all models.

In addition, some recovery management action bits in this field are not set for all models. The repair bit is set if MCH has repaired an SPF key failure. The storage reconfigure bit is set if MCH has performed some type of real storage reconfiguration. The buffer reconfigure bit is set if MCH has performed some type of buffer reconfiguration. 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 error. For example, 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 nonretryable.

The recovery management information area of the field indicates hardware logging of the error.

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

Emergency Recording

Emergency recording is necessary when the system cannot continue to operate, because of either a CCH catastrophic condition or a machine-check interruption unrecoverable condition. Instead of giving control to the operating system to write the error record, since the system is known to be unreliable, MCH writes it. The MCH Emergency Recorder determines whether there is a CCH record to be written and if so writes the record. MCH then 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 attempt 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

Either of the following causes MCH to receive control from the Channel-Check Handler (CCH):

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

When CCH is entered, it sets RVTWSMCC in RVTWSFL0 of the recovery vector table. This indicates to MCH, when a machine-check interruption occurs, that CCH is the affected program and that the system must be terminated.

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

1. Constructs a full channel inboard record and places the address of this record in RVTCCHRC.

2. Sets RVTWSCIO in RVTWSFL0 to indicate that CCH has created a record to be written and that the system must be terminated because of a channel error.

3. Sets wait state code in MCHIBUF.

4. Uses the address in RVTSHUT to pass control to SHUT.

MCH places the system in a wait state with a code passed in MCHIBUF.

MCH METHOD OF OPERATION DIAGRAMS

These Method of Operation Diagrams show the functions of MCH and relate these functions to the exact modules and entry points in the code where they are performed.

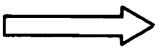
These particular diagrams are known as HIPO diagrams because they show Hierarchy as well as Input, Processing, and Output. The first diagram is a visual table of contents. The rest of the diagrams are arranged in a hierarchy according to the level of detail.

Read the diagrams left to right, input-processing-output. The processing steps are numbered sequentially; these numbers also refer to the implementation notes at the bottom of most of the diagrams. The notes contain additional information about the processing steps.

The arrows are used as follows:


Shows control flow to or from another program.


Primary processing flow. Shows the path followed to accomplish the principal function of the body of code.


Data transfer. Indicates that data is moved from one location to another.


Control information transfer. Indicates the setting or changing of switches or pointers that will be used to determine the course of future processing.


Pointer. Indicates that a field in one data area contains an address that points to another field or data area.


Data reference. Indicates that the contents of a data area are tested or read in order to determine the course of subsequent processing.

Diagram MCH-1. MCH Overview and Table of Contents

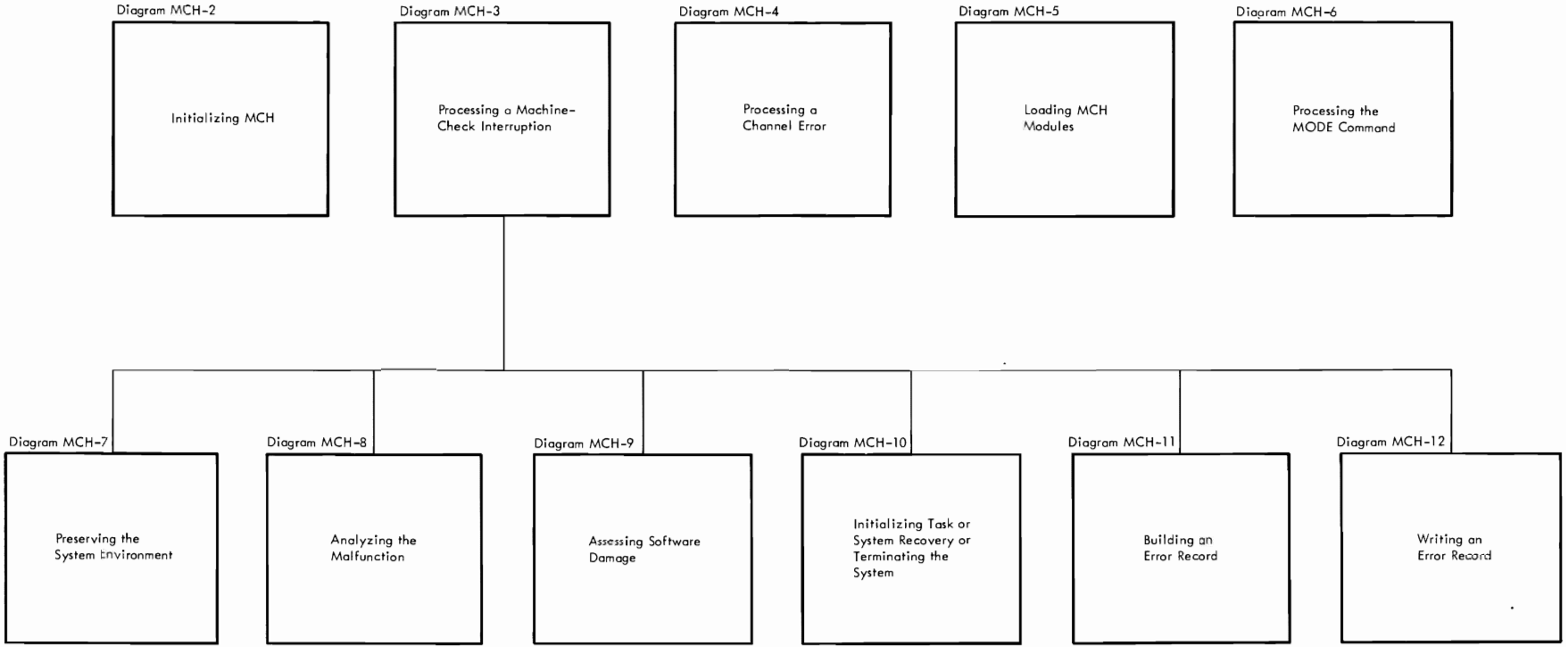
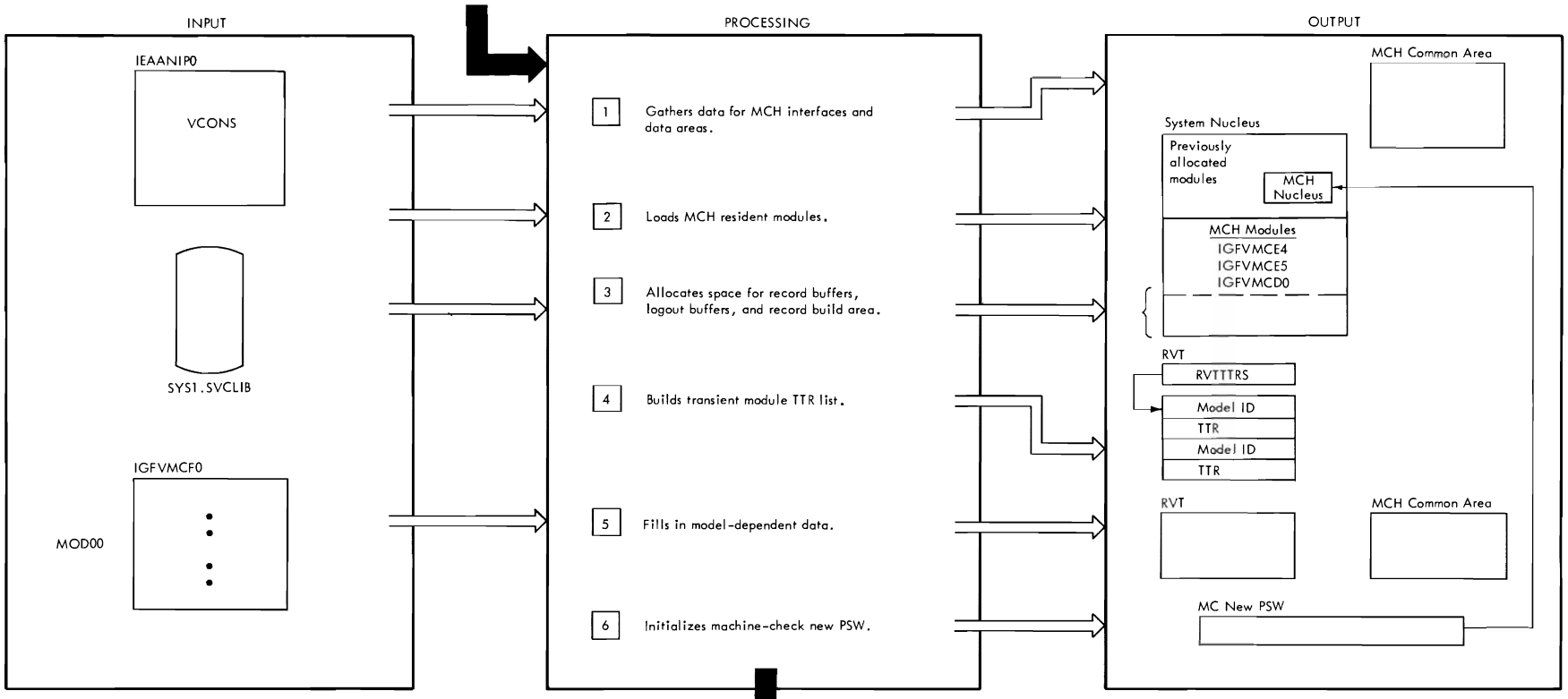


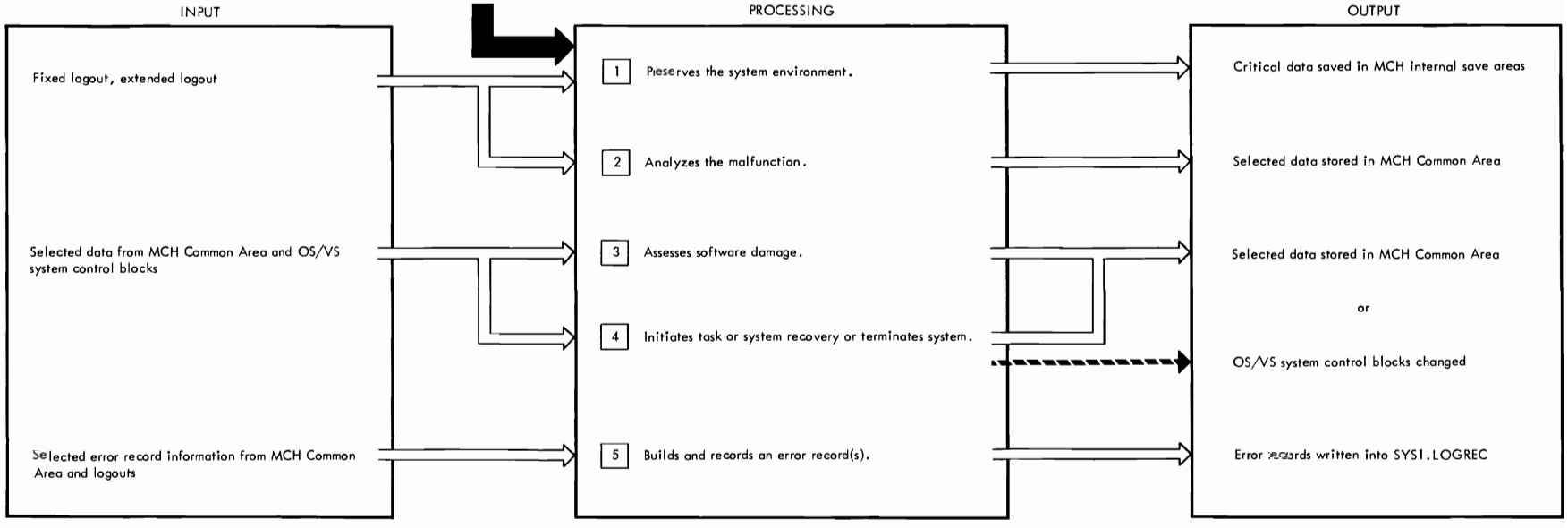
Diagram MCH–2. Initializing MCH



Implementation Notes for Diagram MCH-2

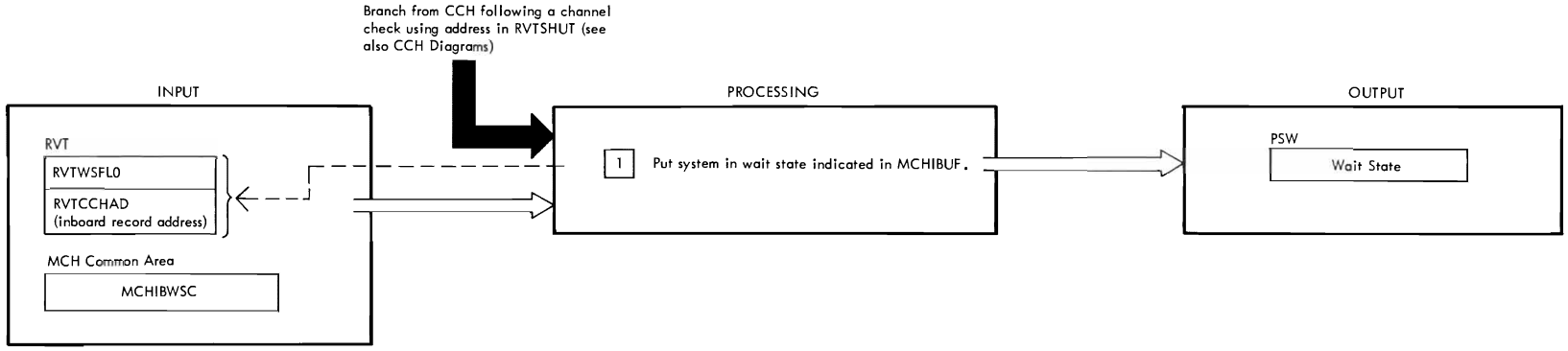
		Module	Label
1	NIP passes pointers to IGFVMCF0.	IEANPRMS (VS1) IEAVNP06 (VS2)	IGFCCEND
2	The NIP subroutine is used to load modules at the end of the system nucleus (VS1 only).	IGFVMCF0	IGFVMCF0
3	MCH updates the system nucleus and pointer to allocate the necessary space. MCH then fills in the RVT pointers.	IGFVMCF0	INITRVT
4	MCH executes BLDLs and constructs a table of TTRs; it then puts a table pointer in RVTTTRS.	IGFVMCF0	LOOPA
5	MCH sets up MCHLOGIC with the model number index.	IGFVMCF0	MOD00
6	MCH turns off the wait bit and points to IGFVMCE0. For the 168, MCH New PSW contains the address of IGFVMCB1, which analyses MCIC and gives control to the MCH nucleus.	IGFVMCF0	MODEXIT
7	MCH branches to NIP with a return code.	IGFVMCF0	EXIT01

Diagram MCH–3. Processing a Machine-Check Interruption



Note: Steps 3 and 4 do not apply to malfunctions that have been corrected by HIR or ECC.

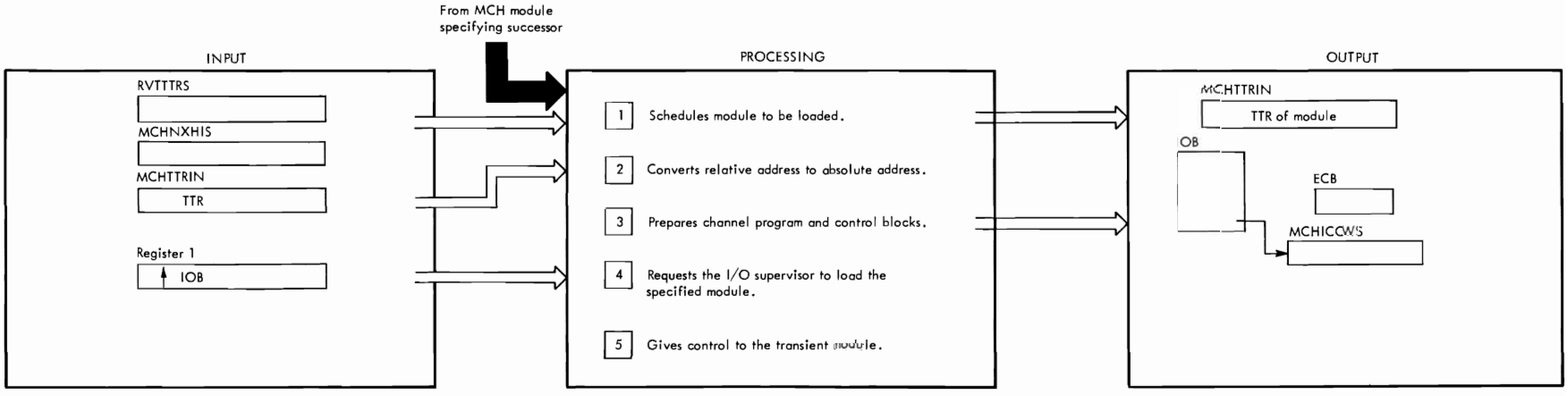
Diagram MCH–4. Recording a Channel Error



Implementation Notes for Diagram MCH-4.

		Module	Label
1	MCH collects data after CCH has passed control via a Branch. RVTCCHAD contains the inboard record address and RVTWSFL0 indicates to MCH that a catastrophic channel error has occurred. The wait state to code is held in MCHIBWSC.	IGFVMCE0	IGFERR0
	The MCH SHUT routine routes control to a portion of the MCH Nucleus, which places the system in the wait state.	IGFVMCE0	NOCONSL

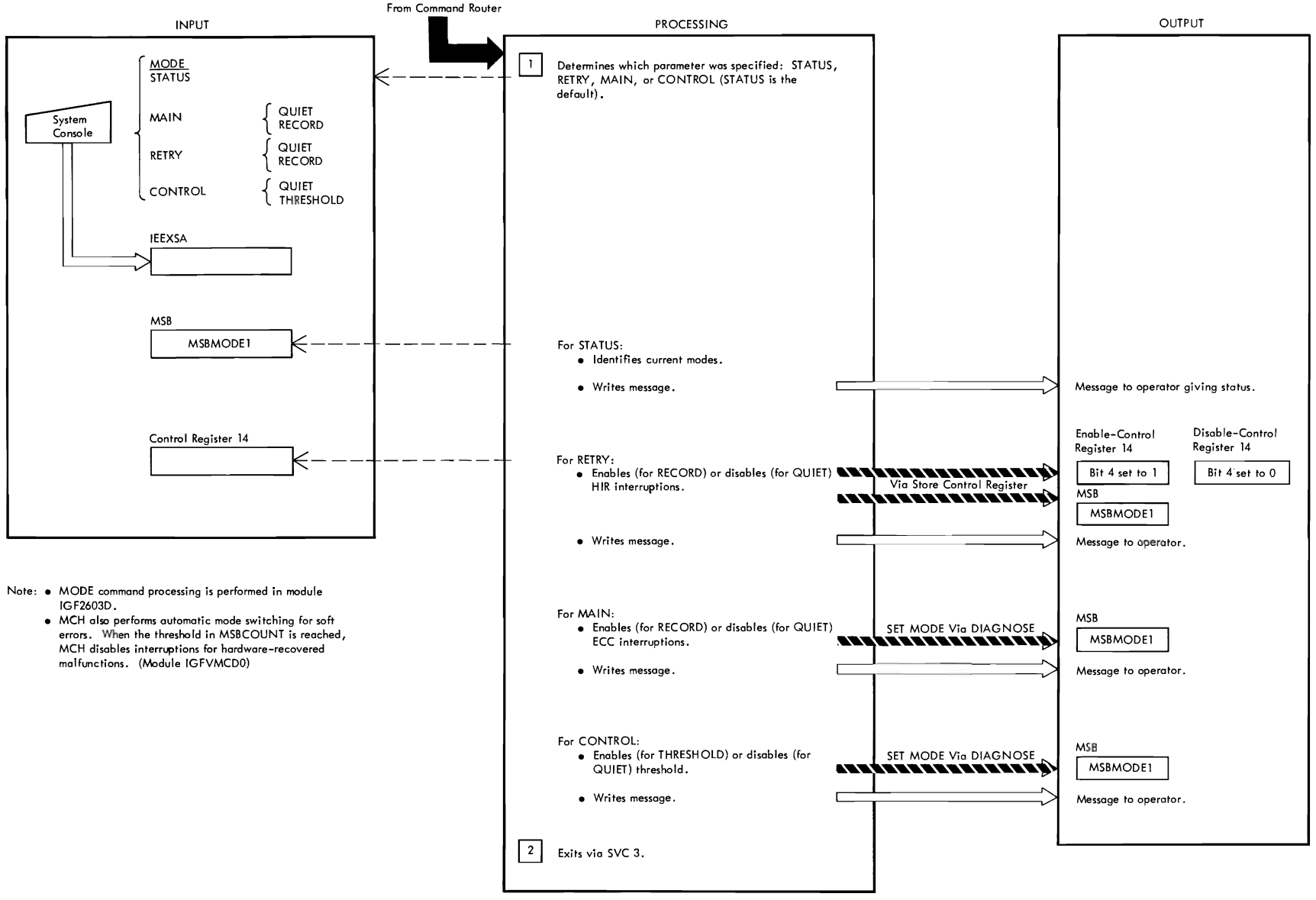
Diagram MCH–5. Loading MCH Transient Modules



Implementation Notes for Diagram MCH–5

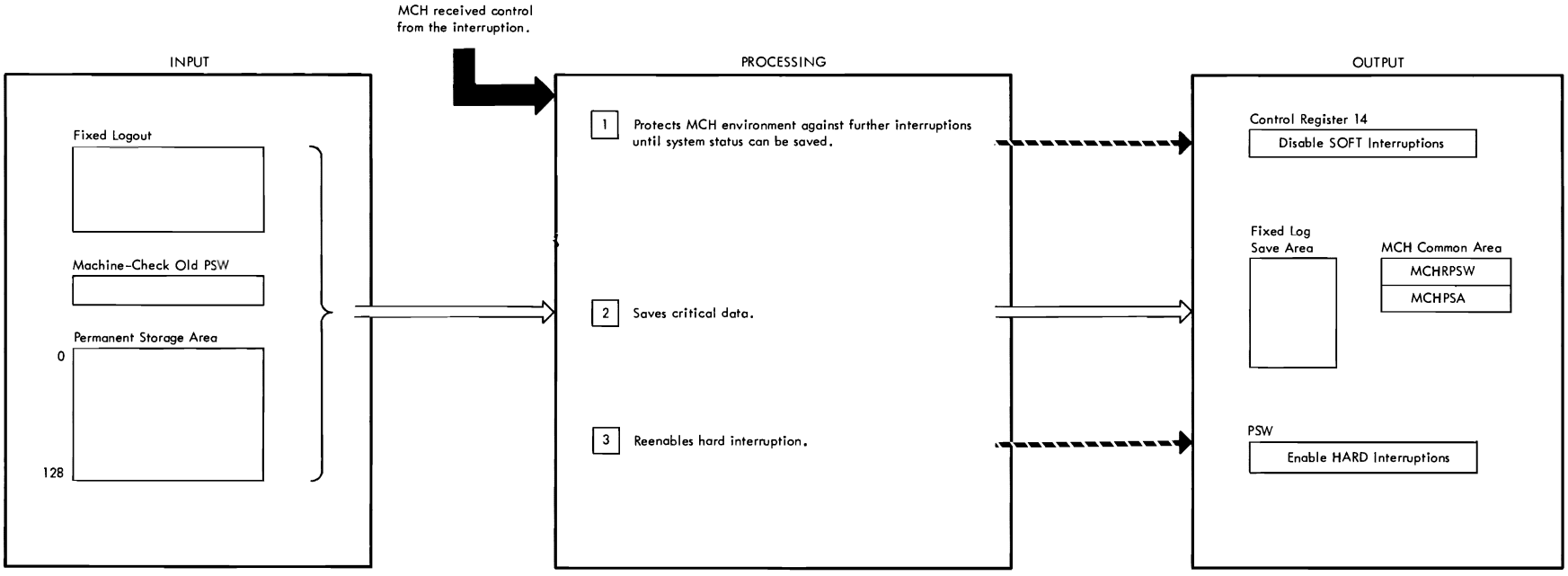
		Module	Label
1	Any transient module places the successor's index value in MCHNXHIS.	any transient module	
2	MCH converts the TTR to MBBCCHHR (physical track address).	IGFVMCE4	IGFLOAD
3	MCH puts the CCW chain address in the IOB, sets up the ECB address, and puts the SEEK MBBCCHHR in the IOB.	IGFVMCE4	NTTRC1
4	MCH passes IOS the address of the IOB and ECB, and sets up the MCH I/O First Level Interrupt Handler.	IGFVMCE4	NEXT
5	MCH converts the TTR to MBBCCHHR (physical track address).	IGFVMCE4	NMODSKD9

Diagram MCH–6. Processing the MODE Command



Note:
- MODE command processing is performed in module IGF2603D.
- MCH also performs automatic mode switching for soft errors. When the threshold in MSBCOUNT is reached, MCH disables interruptions for hardware-recovered malfunctions. (Module IGFVMCD0)

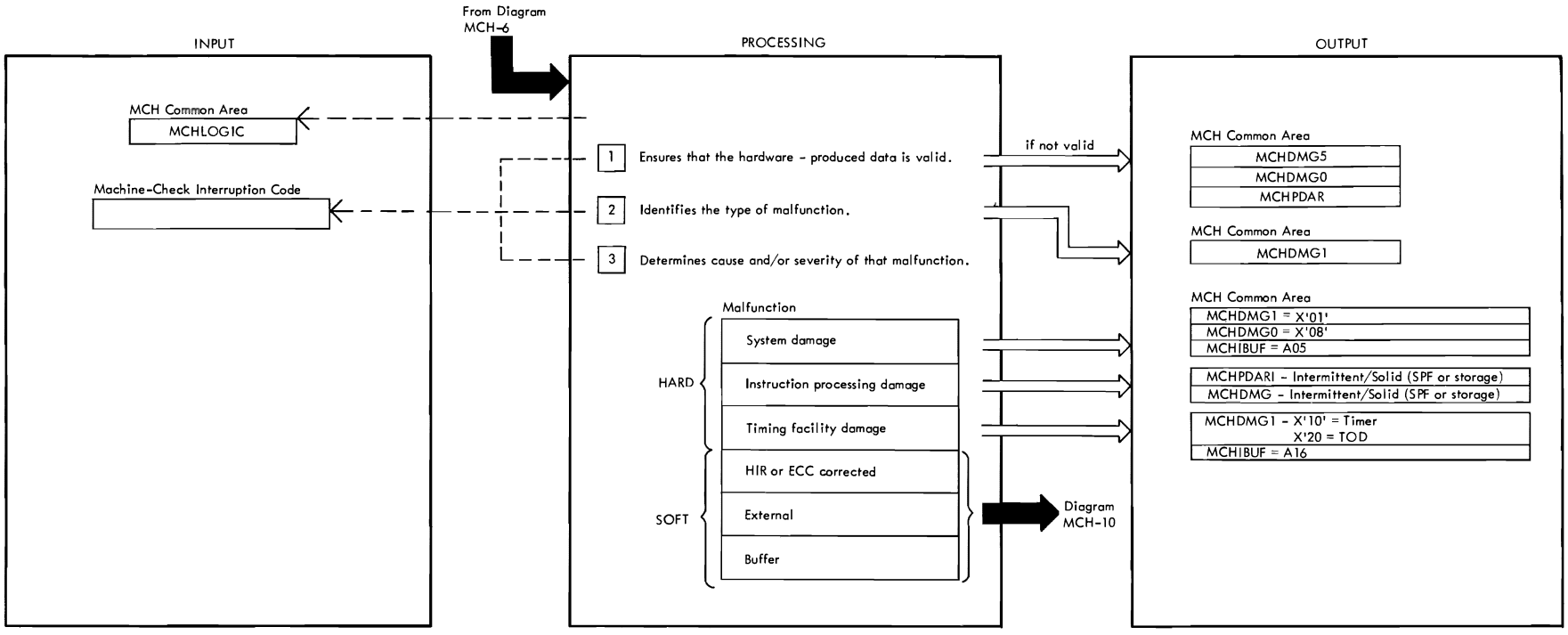
Diagram MCH–7. **Preserving the System Environment**



Implementation Notes for Diagram MCH-7

		Module	Label
1	Control is received with I/O, machine-check, and external interruptions disabled and in supervisor state. Buffer disabled in the Models 158 and 168.	IGFVMCE0	IGFN0000
2	Critical data is saved.	IGFVMCE0	NCHERTST
3	Hard machine-check interruptions are enabled in the PSW.	IGFVMCE0	NCHERTST

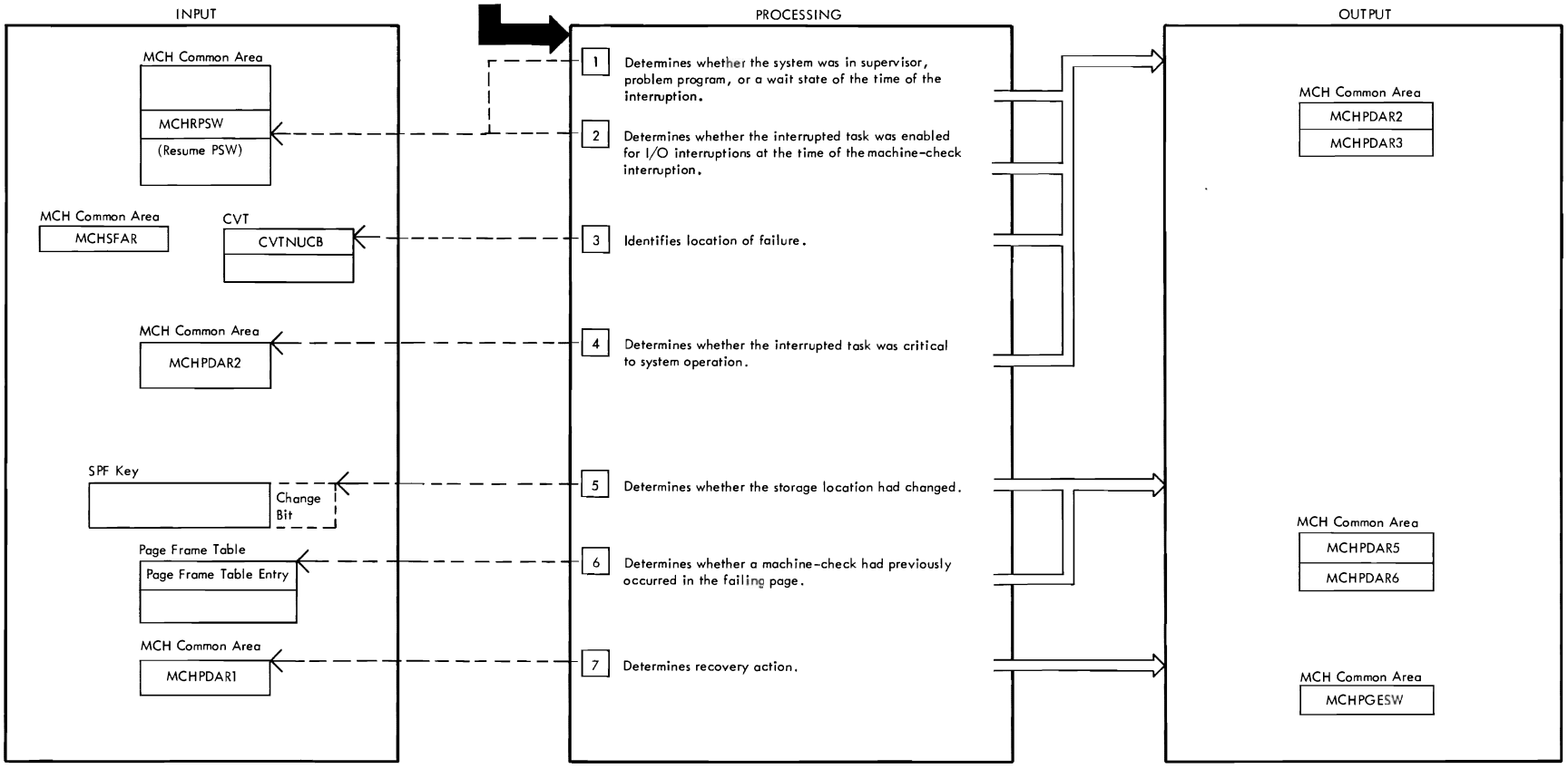
Diagram MCH–8. Analyzing the Malfunction



Implementation Notes for Diagram MCH-8

	Module	Label
1 The MCH Nucleus insures that the interruption code, PSW, and control registers are valid.	IGFVMCE0	MCICCK MCICVALD
2 The machine-check interruption code is examined to determine the type of failure.	IGFVMCE0	ANALYSIS TESTSYSD TSTTIME TESTIPD
3 The severity of the malfunction is determined by identifying whether the interruption was hard or soft and, in the case of real storage or SPF key damage, whether the failure was solid or intermittent.	IGFVMCD1	EXERCISE

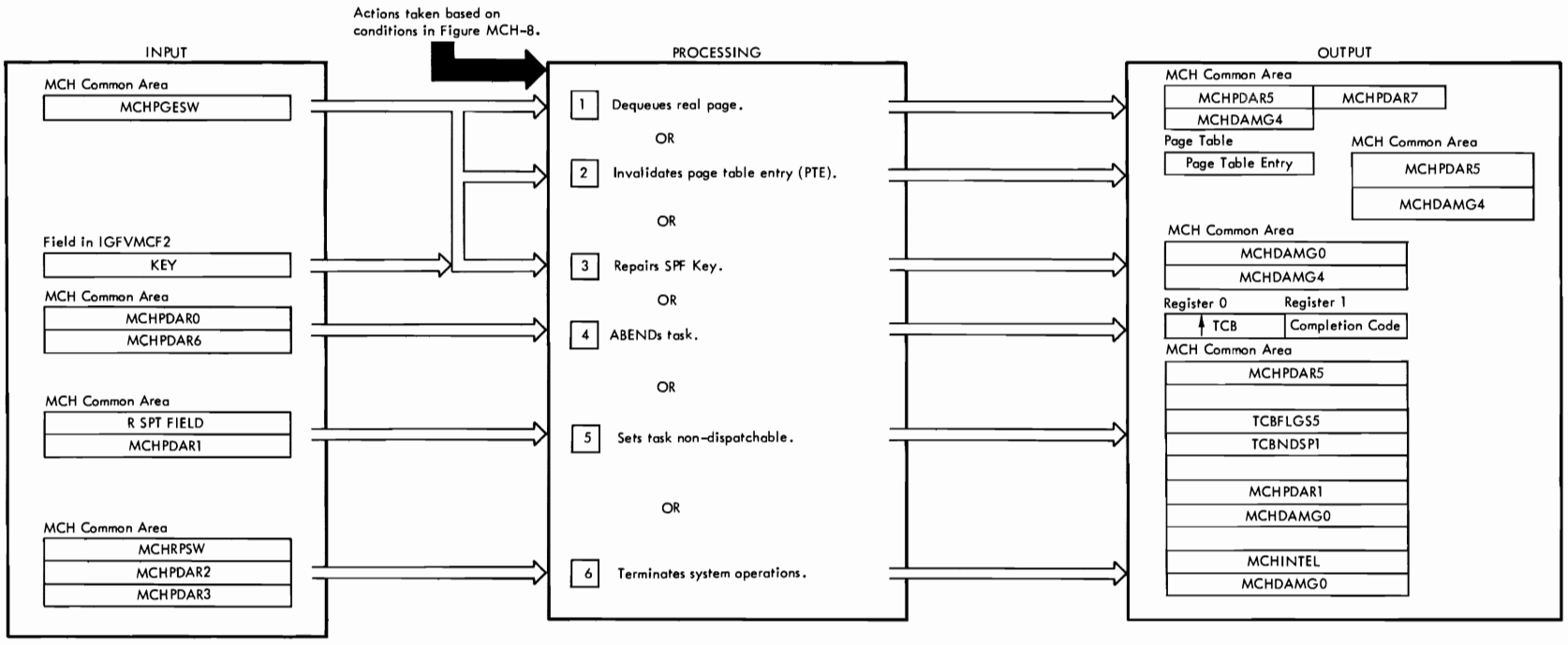
Diagram MCH–9. Analyzing Software Damage



Implementation Notes for Diagram MCH–9

		Module	Label
1	The fields in the resume PSW are tested to determine the system status.	IGFVMCF1	AOSWAIT
2	The resume PSW is checked again.	IGFVMCF1	AOSDIS
3	The CVTNUCB, page frame table entry, and main storage supervisor boundary box are checked.	IGFVMCF2	AOSNUC1 AOSFIX
4	The problem program, supervisor state, and wait indicators are tested.	IGFVMCF2	AOSRECU1 AOSWAIT
5	The SPF key change bit is tested.	IGFVMCF2	AOSCHAN
6	A flag in the page frame table entry is checked.	IGFVMCF2	STATUS
7	MCHPDAR1 is tested for the type of failure.	IGFVMCF2	AOSKEY2 XPTEKEY AOSSTOR2

Diagram MCH–10. Initiating Task or System Recovery or Terminating the System

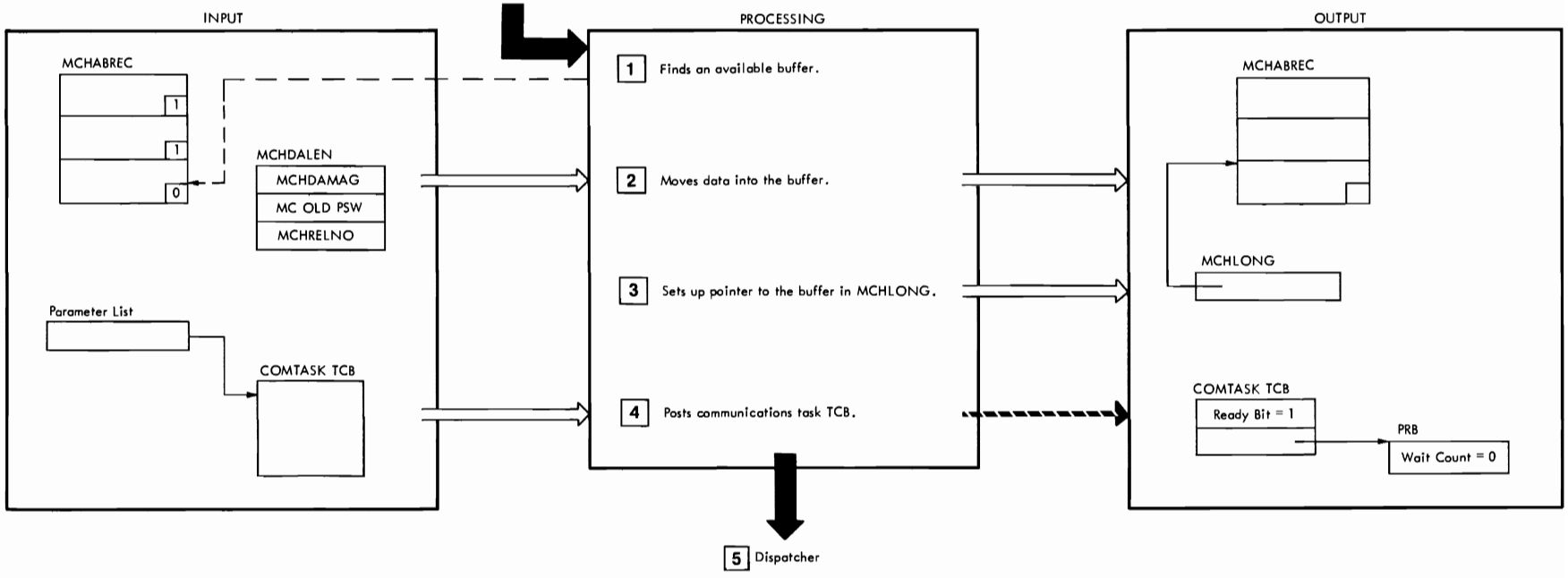


Implementation Notes for Diagram MCH–10

		Module	Label
1	The real page is dequeued through the DEQUEUE routine (RSTDQX00).	IGFVMCF2	PAGEIPTE
2	The "invalid" flag in the page table entry is set and the SPF key is stored in the entry.	IGFVMCF2	PAGEIPTE
3	The SPF key is restored by putting the failing storage address on a 2K boundary and executing the SSK instruction.	IGFVMCF2	AOSKEY
4	The decision to ABEND the task is based on the conditions shown in Figure MCH–7.	IGFVMCF2	AOSTERM1 AOSWAIT
	Initiation of ABEND procedures.	IGFVMCF4	AOSABEND

		Module	Label
5	The decision to set the task non-dispatchable is based on the conditions shown in Figure MCH–7.	IGFVMCF2	STATUS
	The non-dispatch bit is set in the TCB.	IGFVMCF4	AOSDISP
6	The decision to terminate system operations is based on the conditions shown in Figure MCH–7.	IGFVMCF2	AOSNUC2 AOSRECUR PAGESUP BADTRANS
	The system is terminated by placing it in the wait state through an LPSW.	IGFVMCF4 IGFVMCE0	AOSMSG1 IGFERR0 MSGWRITE NOCONSL

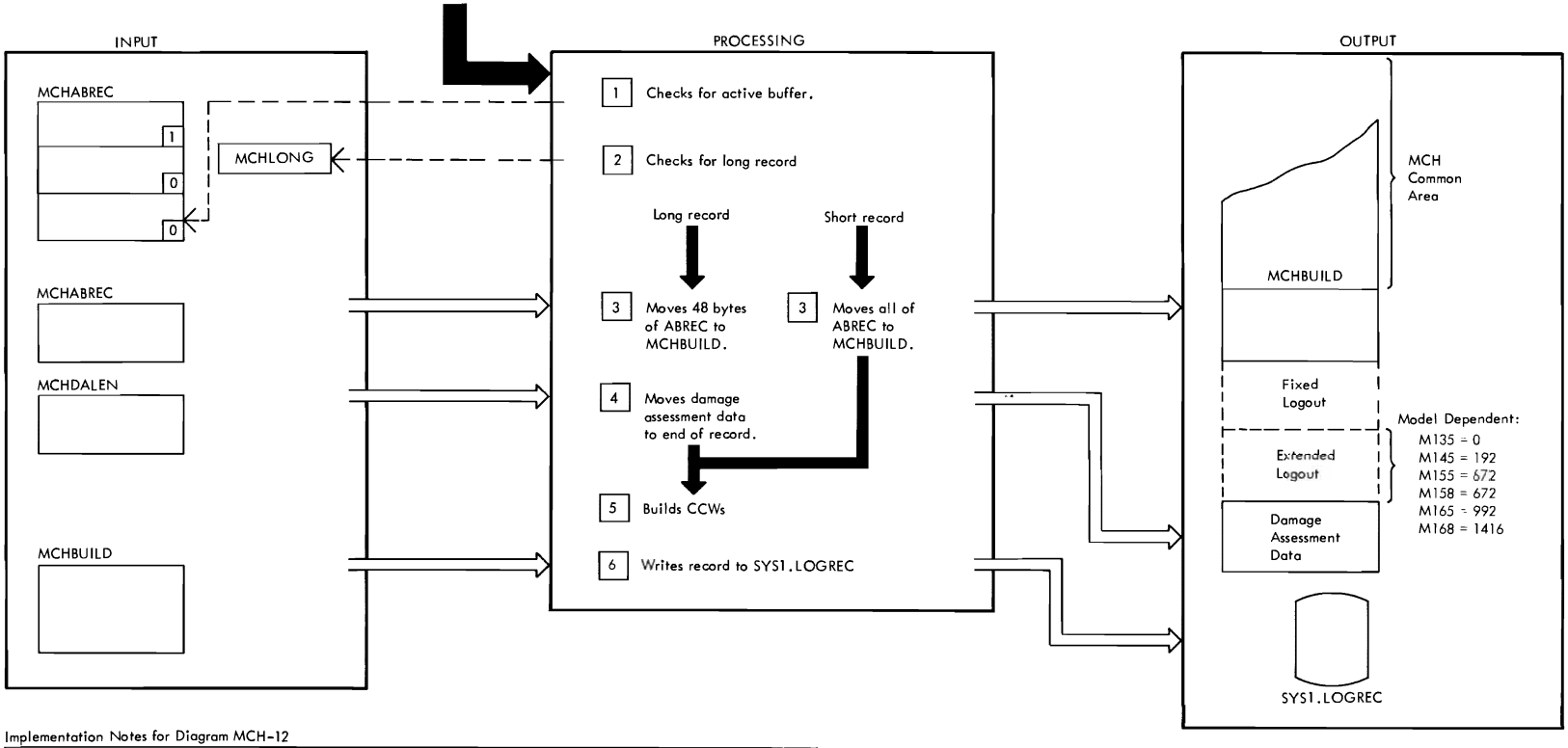
Diagram MCH–11. Building an Error Record



Implementation Notes for Diagram MCH-11

		Module	Label
1	The last byte in each of the three ABREC buffers is tested. A zero in the high-order bit indicates that the buffer is available.	IGFVMCE5	BUFMGMT
2	The MCHDAMAG field, the machine-check old PSW, and the MCHRELNO field of the MCH Common Area are moved into the buffer.	IGFVMCE5	BUILDREC
3	The address of the current buffer is placed in the MCHLONG field of the MCH Common Area.	IGFVMCE5	RECFRMT
4	MCH branches to the system post routine, passing the address of the Communications Task TCB and ECB in the UCM.	System Post Routine	IEA0PT01
5	If I/O is disabled, control is passed to the dispatcher or the interrupted program.	IGFVMCE0	IGFLPSWE

Diagram MCH–12. Writing an Error Record



Implementation Notes for Diagram MCH-12

		Module	Label
1	The MCH Error Recorder searches for an active buffer (from the bottom up). 1=active.	IGFVMCE2	TEST1
2	When the active buffer is found, it is compared to the address in MCHLONG to determine if it is a long record.	IGFVMCE2	TEST2
3	If it is a long record, the contents are moved to the MCHBUILD field to be contiguous with the logouts.	IGFVMCE2	BEGIN
4	The damage assessment data is made contiguous also.	IGFVMCE2	MOVEDA
5	MCH constructs CCWs to write to SYS1.LOGREC.	IGFVMCE2	COMBUILD
6	An SVC 0 (EXCP) is issued to have IOS write to the SYS1.LOGREC data set.	IGFVMCE2	READY

PHYSICAL CHARACTERISTICS OF MCH

Overlay Structure of MCH

MCH has an overlay structure; some parts of the program, called resident modules, remain in real storage and other parts, called transient modules, are in real storage only when they are being used. The transient modules are stored in the SYS1. SVCLIB data set.

Loading Transient Modules in a System with 192K or More Bytes of Real Storage

In a VS1 system with a real storage size equal to or greater than 192K bytes, or in any VS2 system, the 1K MCH transient area is a fixed part of the nonpageable portion of storage.

When MCH is not being used, the Soft Machine-Check Handler (SMCH) occupies the transient area. SMCH is a module that handles the model-dependent portions of soft machine-check interruptions. Having SMCH reside in the transient area eliminates the need to bring in modules from auxiliary storage when a soft machine-check interruption occurs.

When a hard machine-check interruption occurs, and the MCH Nucleus determines that transient modules are needed to continue processing the machine-check interruption, the MCH Nucleus passes control to the MCH I/O Control module (IGFVMCE4) to bring in the necessary transient modules from SYS1. SVCLIB. The first module brought into the transient area then overlays SMCH.

When a transient module finishes execution, it determines which transient module will succeed it, and specifies the successor module to the MCH I/O Control module. The MCH I/O Control module then transfers control to the I/O supervisor, which loads the next module into the transient area. After all processing has been completed, SMCH is loaded into the transient area. Except for system termination, SMCH 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.

Loading Transient Modules in a System with Less Than 192K Bytes of Real Storage

In a VS1 system with a real storage size of less than 192K bytes, the MCH transient area and the MCH I/O Control module reside on a 2K page instead of in real storage. SMCH is the transient area.

When a machine-check interruption occurs, and the MCH Nucleus determines which transient modules are needed to continue processing the machine-check interruption, control is given to a paging interface module which is part of the Paging Supervisor. This module fixes the 2K page with the MCH transient area and the MCH I/O Control module.

Execution continues as described above for a larger system, except that SMCH (IGFVMCD0) schedules the record build module (IGFVMCE5) to be loaded instead of branching to it. When execution is complete, the 2K page is freed to allow its use by the system.

To accomplish this difference in loading transient modules, the following changes appear in system structure:

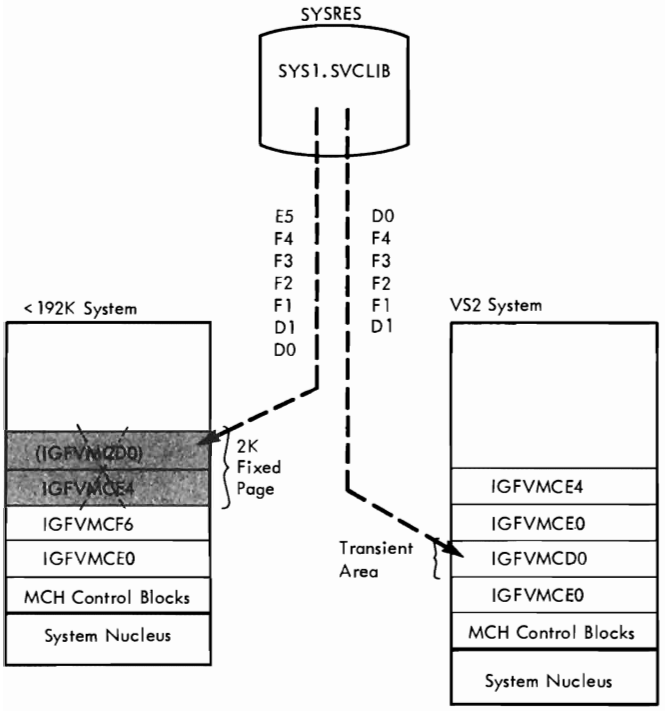
- In a system ≥ 192K: IGFVMCD0 contains SMCH and acts as the transient area. IGFVMCE5 is contained in the nucleus and contains Error Record Build.

- In a system < 192K: IGFVMCD0 is used as SMCH and acts as the transient area on a 2K page. IGFVMCE5 is loaded somewhere other than resident storage by IGFVMCD0. IGFVMCF6 is added as a paging interface (owned by the Paging Supervisor).

Figure MCH-10 illustrates the transient loading technique.

MCH GENERAL PROCESSING

Figures MCH-11 and MCH-12 illustrate the modules used in general processing of soft and hard errors.



Legend:

- Status of system after IPL
- Brought in on occurrence of error in < 192K system
- Transient Area loading, showing sequence of IGFVMC__ modules
- Removed by IGFVMCF6 at exit to free page

Figure MCH-10. MCH transient module loading

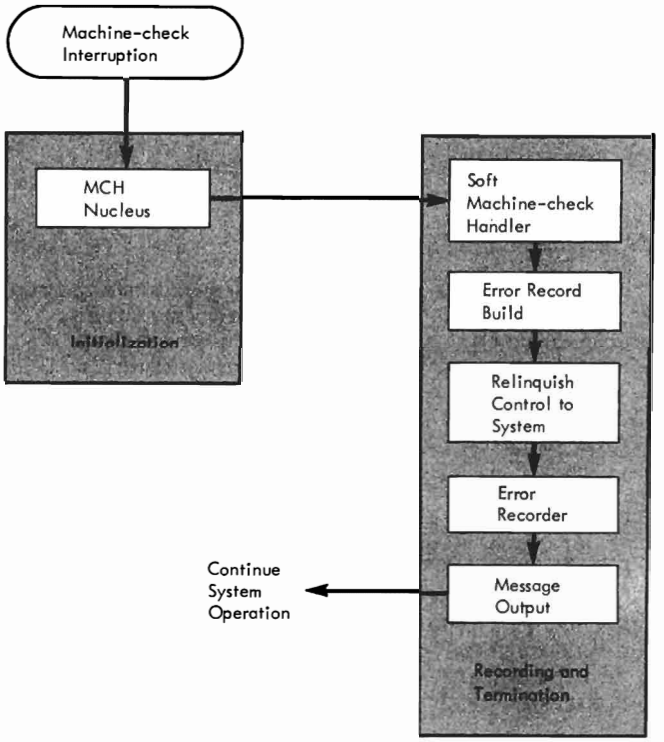


Figure MCH-11. General processing of a soft machine-check interruption

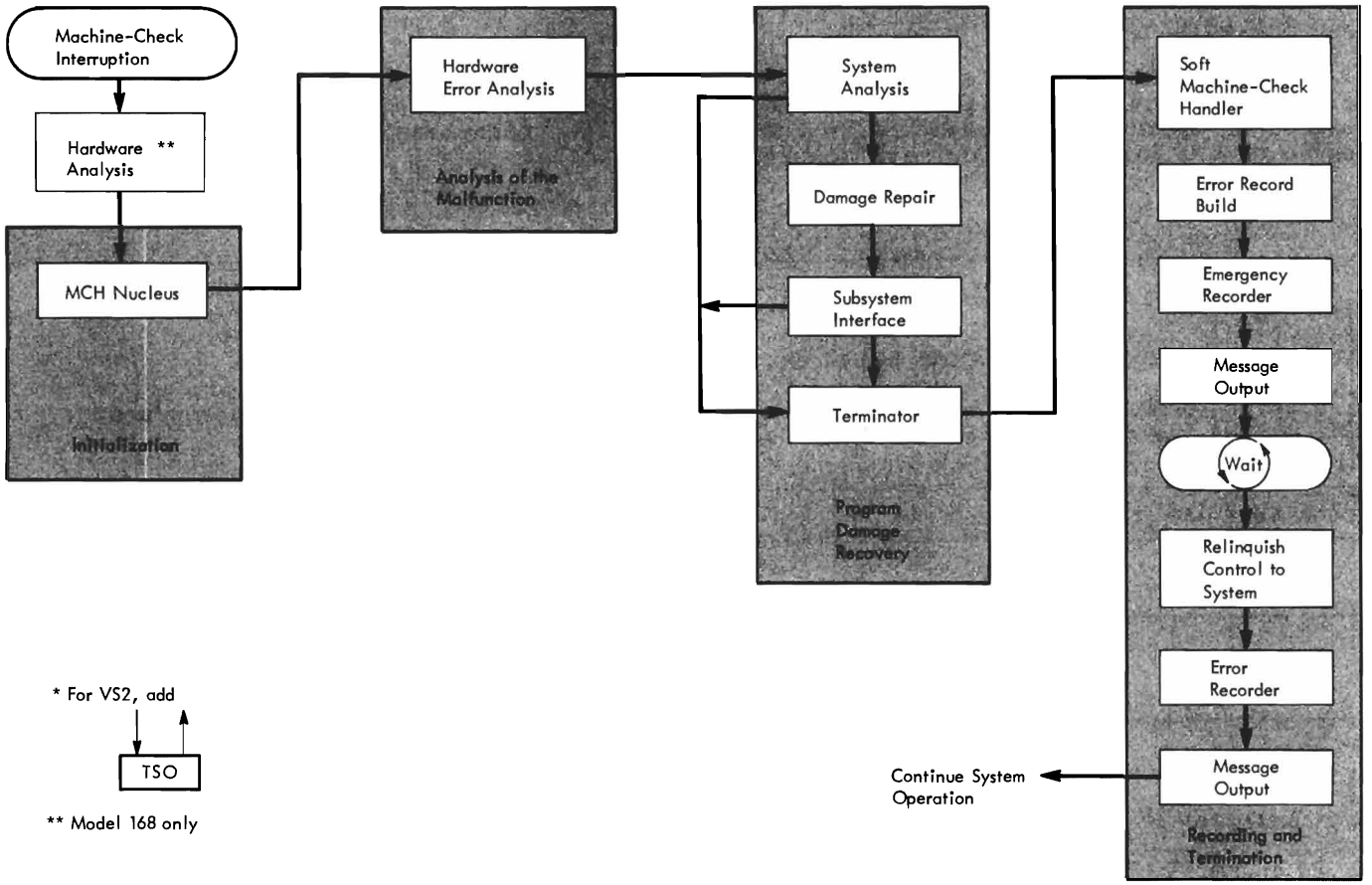


Figure MCH-12. General processing of a hard machine-check interruption

MCH MODULE DESCRIPTIONS

The following module descriptions summarize the functions of MCH that are done by each of the MCH modules. Additional information concerning MCH modules can be found in the method of operation diagrams and in the prologues for each module. Prologues can be found on the microfiche for each module.

IGFVMCD0 - Soft Machine-Check Handler

- Handles high speed buffer failures.
- Switches mode of the processor from record to quiet when the threshold for soft errors has been reached.

Entry Point	Entered From	Exits To
IGFVMCD0	IGFVMCE0 IGFVMCF4	IGFVMCE5 to build the error record.

IGFVMCD1 - Hardware Error Analysis

- Analyzes and gathers data about the malfunction causing the interruption.
- Determines the severity of real storage failures.
- Handles machine-check interruptions occurring as a result of storing and fetching from the damaged location.
- Determines the severity of the SPF key failure (all models).
- Handles machine-check interruptions resulting from key exercises.

Entry Point	Entered From	Exits To
IGFVMCD1	IGFVMCE0	IGFVMCF1 to continue MCH recovery procedures. IGFVMCD0 when the type of malfunction cannot be identified through the machine-check interruption code.

IGFVMCE0 - Machine-Check Handler Nucleus		
• Inhibits interruptions. • Saves critical data. • Examines hardware-produced data for cause of interruption. • Handles machine-check interruptions occurring while another machine-check interruption is being processed.		
Entry Point	**Entered From**	**Exits To**
IGFVMCE0	Machine-check interruption	IGFVMCD0 when there is an HIR- or ECC-corrected malfunction. IGFVMCD1 for instruction processing damage.
IGFERR0 (SHUT)	Machine-Check interruption Channel-Check Handler	The wait state: when a hard machine-check interruption occurs during the processing of a hard machine-check interruption and the original interruption has not yet been recovered from. when system damage has occurred. when the I/O module (IGFVMCE4) is unable to load a transient module. when entry was from the wait state interface. (DSS, Paging Supervisor, or CCH.)
IGFPOST1	IGFVMCE5	Dispatcher or interrupted program.

IGFVMCE1 - Console Write		
• Issues messages to the system console through an SVC 35.		
Entry Point	**Entered From**	**Exits To**
IGFVMCE1 (IGCR207B)	IGFVMCE2	Communications Task Router.

IGFVMCE2 - Error Recorder		
• Writes MCH error records into the SYS1.LOGREC data set.		
Entry Point	**Entered From**	**Exits To**
IGFVMCE2 (IGCR107B)	System communications task router	IGFVMCE1 (IGCR207B) to issue associated messages.

IGFVMCE3 - Emergency Recorder

- Writes CCH and MCH error records into the SYS1.LOGREC data set.

Entry Point	Entered From	Exits To
IGFVMCE3	IGFVMCE4	IGFERR0 (SHUT routine)

IGFVMCE4 - MCH I/O Control

- Loads transient MCH modules.
- Handles I/O operations for the Emergency Recorder (IGFVMCE3).

Entry Point	Entered From	Exits To
NMODSCED	IGFVMCE0 IGFVMCD1 IGFVMCF1 IGFVMCF2 IGFVMCF3 IGFVMCF4 IGFVMCE3 IGFVMCE5 IGFVMCD0	Any of the following transient modules: IGFVMCD0 IGFVMCD1 IGFVMCF1 IGFVMCF2 IGFVMCF3 IGFVMCF4 IGFVMCF5 IGFVMCE3
IGFLOAD	Module Scheduler subroutine of IGFVMCE4	IGFVMCE0 if the loading operation is not successful.
IGFIORTN	I/O Initialization subroutine of IGFVMCE4	

IGFVMCE5 - Error Record Build

- Prepares error records to be recorded in SYS1.LOGREC.
- Posts the RMS ECB to permit the MCH Error Recorder to run under the communications task TCB.
- Terminates the resident operations of MCH.

Entry Point	Entered From	Exits To
IGFVMCE5	IGFVMCE0	IGFVMCE0 for returning control to the system.
	IGFVMCD0	IGFVMCE3 to put MCH record out if system cannot continue.

IGFVMCF1 - System Analysis

- Determines which portion of the operating system is associated with the malfunction.
- Determines whether it is feasible to attempt to repair the damage to the software.

Entry Point	Entered From	Exits To
IGFVMCF1	IGFVMCE0	IGFVMCF2 when it has been determined that software repair is possible.
	IGFVMCD1	IGFVMCF4 when software repair is either not required or not possible.

IGFVMCF2 - Damage Repair

- Determines repair actions that can be taken in respect to the type of error that occurred: repairing damaged SPF key, isolating damaged real storage locations so they cannot be reused, and forcing damaged pages into new locations in real storage so they can be reexecuted successfully.

Entry Point	Entered From	Exits To
IGFVMCF2	IGFVMCF1 IEAPTRV IEAPFP2 (VS2) IEAPRLS3 (VS2) RSTDQX00 (VS1) IGC07902 (VS2)	IGFVMCF4 when repair has been successful. IGFVMCF3 when repair attempt has been unsuccessful to allow the subsystem interface to attempt recovery. IEAPTRV to translate the real failing storage address to a virtual address. IEAPFP2 to find the PTE. IEAPRLS3 to dequeue the page from the PFT (VS2). RSTDQX00 to dequeue the page from the PFT (VS1). IGC07902 to set the job step TCB (JSTCB) and all subtasks nondispatchable (VS2).

IGFVMCF3 - Subsystem Interface

- Determines whether the active subsystem contains recovery support.
- Calls the recovery module of the subsystem.
- Passes data to the subsystem to permit it to carry out its recovery activities.

Entry Point	Entered From	Exits To
IGFVMCF3	IGFVMCF2 Subsystem Recovery module	A subsystem recovery module. IGFVMCF4 after subsystem has attempted recovery.

IGFVMCF4 - Terminator		
• Schedules termination of tasks or operating system based on conclusions of previously executed modules. • Schedules appropriate message, either "task non-dispatchable" or "wait state" message.		
Entry Point	Entered From	Exits To
IGFVMCF4	IGFVMCF1 IGFVMCF2 IGFVMCF3	IGFVMCD0 to initiate error recording.

IGC2603D - MODE Command Module		
• Informs the operator of the status of the processor upon his request. • Switches the mode of the processor from record to quiet or quiet to record.		
Entry Point	Entered From	Exits To
IGC2603D (IGFVMCD4 alias)	IEE0403D (Command Router)	Supervisor (SVC 3) IGC0503D (for error messages).

The directory provides a quick reference from this publication to the pertinent coding, or from the coding back into the publication. The directory contains the following information:

Module and CSECT Name: The name of the object module and CSECT containing the coding indicated by the label.

Label: Names taken from the listing for entry points, significant sections of coding, and data areas.

Library: The place where the module resides.

Name: The descriptive name used to refer to the module.

Diagram: The method of operation diagram containing information about the coding identified by the label.

Module and CSECT Name	Label	Library	Name	Diagram
IGC2603D (IGFVMCD4 alias)	CKSYNTAX STATRTN MAINRTN RETRYRTN	SYS1.SVCLIB SYS1.LPALIB (VS2)	MODE Command Module	
IGFVMCD0	MODESWCH	SYS1.SVCLIB, SYS1.LINKLIB (VS1) SYS1.SVCLIB, Nucleus (VS2)	Soft Machine-Check Handler	6
IGFVMCD1		SYS1.SVCLIB	Hardware Error Analysis	8
IGFVMCE0	ANALYSIS IGFERR0 IGFN0000 MCICCK MCICVALD MSGWRITE NOCONSL SETCHMSG TESTIPD TSTTIME TSTTODC TESTSYSD	Nucleus	MCH Nucleus	8 4 7 8 8 4 4 4 8 8 8
IGFVMCE1		SYS1.SVCLIB SYS1.LPALIB (VS2)	Console Write	
IGFVMCE2	BEGIN COMBUILD MOVEDA READY TEST1 TEST2	SYS1.SVCLIB SYS1.LPALIB (VS2)	Error Recorder	12 12 12 12 12 12
IGFVMCE3	ACTCHECK IORTN	SYS1.SVCLIB	Emergency Recorder	4 4

Module and CSECT Name	Label	Library	Name	Diagram
IGFVMCE4		SYS1.LINKLIB (VS1) Nucleus (VS2)	MCH I/O Control	5
IGFVMCE5	BUFMGMT BUILDREC RECFRMT	SYS1.LINKLIB \ VS1 SYS1.SVCLIB / SYS1.SVCLIB \ VS2 Nucleus /	Error Record Build	11 11 11
IGFVMCF1		SYS1.SVCLIB	System Analysis	9
IGFVMCF2	IGFVMCF2	SYS1.SVCLIB	Damage Repair	10
IGFVMCF3		SYS1.SVCLIB	Subsystem Interface	
IGFVMCF4		SYS1.SVCLIB	Terminator	10
IGFVMCF6		SYS1.LINKLIB	<192K interface to paging	

This section contains descriptions of the principal data areas used by MCH:

- Fixed storage areas
- Machine-check interruption code
- MCH independent common area
- MCH recovery vector table
- Machine status block
- MCH long record
- MCH testing vector table

The symbolic names shown in individual fields represent the displacement, in bytes, from the beginning of a specific data area to the field. Access is gained to a specific field by using an instruction in which the beginning address of the area (usually contained in a register) is the base address, and the symbolic field name represents the displacement. Use of the fields can be traced in the MCH listings by first locating the symbolic field names in the cross-reference table at the back of the listings and then noting where the names are used.

The field headings, used for other than the fixed areas, and their meanings are:

Displacement: The numeric address of the field relative to the beginning of the data area. The first number is in decimal; the second (in parentheses) is the hexadecimal equivalent.

Bytes and Bit Patterns: The size (number of bytes) of the field and the bit settings of flag fields; that is, the state of bits in a byte. When the column is used to show the state of bits, it is shown as follows:

....	The 8 bit positions (0-7) in a byte. For ease of scanning, the high-order (left-hand) four bits are separated from the low-order four bits.
1...	A reference to bit 0.
.... ..xx	Bits 6 and 7 are reserved.

Bit settings that are significant are shown and described. Bit settings that are not presently significant are described as reserved bits.

Field Name: A name that identifies the field.

Field Description: The use of the field.

Figure MCH-13 is provided to assist in locating MCH fields. This figure contains an alphabetic list of the fields with their displacements.

Field	Displacement	Field	Displacement	Field	Displacement	Field	Displacement
MCKETLO*	328 (148)	MCHINTEL	104 (68)	MCHSKADR	604 (25C)	RVTINLOG	32 (20)
MCKFXLO*	48 (30)	MCHINTL0	104 (68)	MCHSPARE	2 (2)	RVTIOREG	112 (70)
MCKPSW*	40 (28)	MCHINTL1	105 (69)	MCHSPR1	93 (5D)	RVTIOSEP	56 (38)
MCHABREC	619 (26B)	MCHIOB	332 (146)	MCHSPR2	370 (174)	RVTIOSIH	60 (3C)
MCHABRNO	618 (26A)	MCHIOBSK	362 (16C)	MCHSRCH	536 (218)	RVTIOSNS	80 (50)
MCHBYREM	514 (202)	MCHIODCB	352 (160)	MCHSTATD*	(end of long record)	RVTLEVEL	100 (64)
MCHCCHAD	612 (264)	MCHIOECB	336 (150)			RVTLOADA	28 (1C)
MCHCCHLN	616 (268)	MCHIPTR	402 (194)	MCHSUBA	316 (13C)	RVTLNG	102 (68)
MCHCCW	544 (220)	MCHLEVEL	320 (140)	MCHSUBAW	88 (58)		or 140 (8C)
MCHCCWRD	390 (188)	MCHLOGC0	108 (6C)	MCHTCBAF	80 (50)	RVTMSB	64 (40)
MCHCCWSR	374 (178)	MCHLOGC1	109 (6D)	MCHTCBCU	76 (4C)	RVTNIPRM	24 (18)
MCHCCWTC	382 (180)	MCHLOGC2	110 (6E)	MCHTFLAG	611 (263)	RVTNUCAD	0 (0)
MCHCLRLN	94 (5E)	MCHLOGIC	108 (6C)	MCHTIC	544 (220)	RVTPAGES	84 (54)
MCHDALEN	0 (0)	MCHLONG	492 (1EC)	MCHTKPCY	525 (20D)	RVTPDISW	124 (7C)
MCHDAMAG	8 (8)	MCHLSTTR	518 (206)	MCHTMSVE	112 (70)	RVTPGESW	120 (78)
MCHDEVCD	529 (211)	MCHLWLIM	498 (1F2)	MCHTRCAP	516 (204)	RVTPOST1	8 (8)
MCHDMG0	8 (8)	MCHMAINT	835 (343)	MCHTTRIN	324 (144)	RVTPRIMY	48 (30)
MCHDMG1	9 (9)	MCHMCSRT	612 (264)	MCHUCB	328 (148)	RVTRESUM	108 (6C)
MCHDMG2	10 (A)	MCHMLSAV	252 (FC)	MCHUPLIM	502 (1F6)	RVTSBSYS	96 (60)
MCHDMG3	11 (B)	MCHNXHIS	111 (6F)	MCHWORK	496 (1F0)	RVTSECND	52 (34)
MCHDMG4	12 (C)	MCHNXMOD	92 (5C)	MCHWRNCT	527 (20F)	RVTSHUT	44 (2C)
MCHDMG5	13 (D)	MCHPDAR	56 (38)	MCHWRNSW	534 (216)	RVTSIRBA	68 (44)
MCHDMG6	14 (E)	MCHPDARI	76 (48)	MCHWRNTR	530 (212)	RVTSIRBE	128 (80)
MCHDMG7	15 (F)	MCHPDAR0	56 (38)	MSBBUFER	16 (10)	RVTSIRFQ	136 (88)
MCHERR	400 (192)	MCHPDAR1	57 (39)	MSBBUFTH	20 (14)	RVTSMCH	4 (4)
MCHEXCCW	576 (240)	MCHPDAR2	58 (3A)	MSBCOUNT	8 (8)	RVTSVF	116 (74)
MCHEXCSW	568 (238)	MCHPDAR3	59 (3B)	MSBCPUID	0 (0)	RVTTP1EX	132 (84)
MCHEXDCB	580 (244)	MCHPDAR4	60 (3C)	MSBBUFD1	44 (2C)	RVTTRANS	24 (18)
MCHEXECB	600 (258)	MCHPDAR5	61 (3D)	MSBBUFD2	56 (38)	RVTTRS	40 (28)
MCHEXEPT	564 (234)	MCHPDAR6	62 (3E)	MSBBUFE1	40 (28)	RVTVCOMM	12 (C)
MCHEXIOB	560 (230)	MCHPDAR7	63 (3F)	MSBBUFE2	48 (30)	RVTWSFLG	88 (58)
MCHEXLEV	55 (37)	MCHPGESW	92 (5C)	MSBECCDS	36 (24)	RVTWSFL0	88 (58)
MCHEXSEK	592 (250)	MCHPGWPC	124 (7C)	MSBECCD2	72 (48)	RVTWSFL1	89 (59)
MCHFCTL	398 (190)	MCHPSA	124 (7C)	MSBECCEN	32 (20)	RVTWSFL2	90 (5A)
MCHFSAR	16 (10)	MCHPSWAD	100 (64)	MSBECCE2	64 (40)	RVTWSFL3	91 (5B)
MCHFSAV	64 (40)	MCHPSWCP	98 (62)	MSBLEVEL	80 (50)	RVTWSMSG	16 (10)
MCHFSEAV	68 (44)	MCHPSWM	96 (60)	MSBLNG	92 (5C)	TVTLEVEL	28 (1C)
MCHFSTBY	535 (217)	MCHPSWSP	99 (63)	MSBLOGSZ	6 (6)	TVTMCN	8 (8)
MCHHDREC	496 (1F0)	MCHPSWWM	97 (61)	MSBMODE	24 (18)	TVTPCN	4 (4)
MCHHISTY	24 (18)	MCHRCTL	399 (191)	MSBMODEL	4 (4)	TVTRPSW	24 (18)
MCHIBLNG	408 (19A)	MCHRECST	507 (1FB)	MSBMODE1	24 (18)	TVTRTN	16 (10)
MCHIBTXT	410 (19C)	MCHRELNO	106 (6A)	MSBSERNO	1 (1)	TVTSIM	0 (0)
MCHIBUF	406 (198)	MCHRES	401 (193)	MSBSPARE	84 (54)	TVTSOLID	0 (0)
MCHIBWSC	406 (198)	MCHRPSW	96 (60)	MSBTHRLD	12 (C)	TVTSPARE	29 (1D)
MCHICBSP	356 (164)	MCHSBSF1	85 (55)	RVTBUILD	36 (24)	TVTTEST	0 (0)
MCHICCWS	344 (158)	MCHSBSF2	86 (56)	RVTBUFAD	92 (5C)	TVTWAIT	12 (C)
MCHIOCSW	340 (154)	MCHSBSF3	87 (57)	RVTCCHRC	20 (14)	TVTWPSW	20 (14)
MCHINT	398 (190)	MCHSBSID	84 (54)				

Note: Field names with an * are in the MCH Long Record. Otherwise, all field names beginning with MCH are in the MCH Independent Common Area, all field names beginning with MSB are in the Machine Status Block, all field names beginning with RVT are in the Recovery Vector Table, and all field names beginning with TVT are in the Testing Vector Table.

Figure MCH-13. Locations of MCH fields

FIXED STORAGE AREAS USED BY MCH

Location	Length in Bytes	Field Description
0 (0)	8	IPL PSW
8 (8)	8	IPL CCW 1
16 (10)	8	IPL CCW 2.
24 (18)	8	External old PSW.
32 (20)	8	SVC old PSW.
40 (28)	8	Program-check old PSW.
48 (30)	8	Machine-check old PSW.
56 (38)	8	I/O old PSW.
64 (40)	8	Channel status word
72 (48)	4	Channel address word
76 (4C)	4	Reserved.
80 (50)	4	Interval timer.
84 (54)	4	Reserved.
88 (58)	8	External new PSW.
96 (60)	8	SVC new PSW.
104 (68)	8	Program-check new PSW.
112 (70)	8	Machine-check new PSW.
120 (78)	8	I/O new PSW.
128 (80)	5	Reserved.
133 (85)	1	External ILC.
134 (86)	2	External interrupt code.
136 (88)	1	Reserved.
137 (89)	1	SVC ILC.
138 (8A)	2	SVC interrupt code.
140 (8C)	1	Reserved.
141 (8D)	1	Program-check ILC.
142 (8E)	2	Program-check interrupt code.
144 (90)	1	Reserved.
145 (91)	3	Translation exception address.
148 (94)	1	Monitor class number.
149 (95)	1	PER (Program Event Recording) code.
150 (96)	3	Reserved.
153 (99)	3	PER (Program Event Recording) address.
156 (9C)	1	Reserved.
157 (9D)	3	Monitor code.
160 (A0)	8	Reserved.
168 (A8)	4	Channel ID.
172 (AC)	4	I/O extended logout pointer.
176 (B0)	4	Limited channel logout (ECSW).
180 (B4)	4	Reserved.
184 (B8)	4	I/O address.
188 (BC)	44	Reserved.
232 (E8)	8	Machine-check interrupt code.
240 (F0)	8	Reserved.
248 (F8)	4	Failing storage address.
252 (FC)	4	Region code.
256 (100)	96	Reserved.
352 (160)	32	Floating-point register save area.
384 (180)	64	General purpose register save area.
448 (1C0)	64	Control register save area.
512 (200)	Variable	CPU extended logout area.

MACHINE-CHECK INTERRUPTION CODE

The machine-check interruption code is two full words long (64 bits), and it 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.

Bit	Meaning
0 (SD)	System Damage: 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.
1 (PD)	Instruction Processing Damage: Set when the extent of the damage is limited to an executed instruction or its associated operands.
2 (SR)	System Recovery: Errors were detected but have been successfully recovered without loss of system integrity.
3 (TD)	Timer Damage: Damage has occurred to either the timer or to location 80.
4 (CD)	Time-of-Day Clock Damage: Damage has occurred to the time-of-day clock.
5 (ED)	External Damage: 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.
6	Reserved.
7 (DG)	Degradation: Continuous degradation of system performance, more serious than normal system recovery has occurred. Degradation may be reported when system recovery conditions exceed a hardware pre-established threshold or when unit deletion has occurred.
8 (W)	Warning: Damage is impending to some part of the system; for example, loss of power or loss of cooling.
9-13	Reserved.
14 (B)	Backup: 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 0, a valid instruction address points to an instruction beyond the error.
15 (D)	Delayed: 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.
16 (SE)	Storage Error Uncorrected: A reference to storage resulted in the detection of damaged data that could not be corrected.
17 (SC)	Storage Error Corrected: A reference to storage resulted in the detection of an error that was subsequently corrected.
18 (KE)	Key in Storage Error Uncorrected: A key in storage contains invalid CBC.

Bit	Meaning
19	Reserved.
20 (WP)	PSW Validity: Bits 12-15 of the machine-check old PSW are valid.
21 (MS)	PSW Masks and Key Validity: All PSW bits other than interruption code, ILC, AMWP, IA, CC, and program mask of the machine-check old PSW are valid.
22 (PM)	Program Mask and Condition Code Validity: The program mask and condition code in the machine-check old PSW are valid.
23 (IA)	Instruction Address Validity: 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.
24 (FA)	Failing-Storage Address Valid: The failing-storage address in the failing save area is valid.
25 (RC)	Region Code Valid: A valid region code has been stored.
26	Reserved.
27 (FP)	Floating-Point Registers Valid: the contents stored in the floating-point register save area are the same as the contents of the registers at the point of interruption.
28 (GR)	General Registers Valid: The contents stored in the general register save area are the same as the contents of the registers at the point of interruption.
29 (CR)	Control Register Validity: The contents stored in the control register save area accurately reflect the condition of the control registers at the time of interruption.
30 (LG)	Log Valid: The CPU extended log information was correctly stored.
31 (ST)	Storage Logical Validity: The contents of those storage locations that are modified by execution were restored to their contents at the point of interruption.
32-45	Reserved.
46 (CT)	CPU Timer Valid: The CPU timer is not in error and the contents stored in the CPU timer save area (location 216) reflect the correct state of the CPU timer at the time the interruption occurred.
47 (CC)	Clock Compartor Valid: The clock comparator is not in error and the contents stored in the clock comparator save area (location 224) reflect the correct state of the clock comparator.
48-63	CPU Extended Log Length: 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 0.

MCH INDEPENDENT COMMON AREA

The MCH independent common area occupies 1,024 bytes in the MCH resident area. It is used by the MCH modules to communicate with each other and to store data to be included in the error records that are written into the SYS1.LOGREC data set.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
0 (0)	2	MCHDALEN	Length of damage assessment field.
2 (2)	6	MCHSPARE	Reserved.
8 (8)	8	MCHDAMAG	Damage assessment data.
8 (8)	1	MCHDMG0	System status:
	1...		Hardware recovery.
	.1..		Software recovery.
	..1.		Task aborted.
	...1		Task nondispatchable.
	.... 1...		Operating system termination.
	.... .1..		Quiet mode in effect.
	xx		Reserved.
9 (9)	1	MCHDMG1	Damage area:
	1...		Real storage.
	.1..		Buffer.
	..1.		Control storage.
	...1		Interval timer.
	.... 1...		Processor.
	.... .1..		Channel error.
	.... ..1.		Time-of-day clock.
	1		System damage.
10 (A)	1	MCHDMG2	Reserved.
11 (B)	1	MCHDMG3	Error type:
	1...		Intermittent.
	.1..		Solid.
	..1.		Data.
	.... .1..		Protect.
	...x x.xx		Reserved.
12 (C)	1	MCHDMG4	RMS action data:
	1...		Loop time out.
	.1..		Repair.
	..1.		Storage reconfigure.
	...1		Buffer reconfigure.
	 xxxx		Reserved.
13 (D)	1	MCHDMG5	RMS information status:
	.1..		Invalid machine-check interrupt code.
	..1.		Invalid failing storage address.
	...1		Program check in MCH.
	x... xxxx		Reserved.
14 (E)	1	MCHDMG6	Reserved.
15 (F)	1	MCHDMG7	Reserved.
16 (10)	4	MCHFSAR	Real address location of the failure.
20 (14)	4		Reserved.
24 (18)	32	MCHHISTY	Table of which MCH modules have executed and in what sequence.
54 (36)	1	MCHNXHIS	First byte following the MCH history table, containing the ID of the module last loaded.
55 (37)	1	MCHEXLEV	Second byte following the MCH history table, containing the change level of the last loaded module.
56 (38)	8	MCHPDAR	Program damage assessment and repair data.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
56 (38)	1	MCHPDAR0	PDAR action:
	1...		Termination of current task.
	.1..		Retry possible.
	..1.		Bypass (affected task) terminator.
	...1		Repair unsuccessful.
	.... 1...		Indeterminate instruction counter.
	xxx		Reserved.
57 (39)	1	MCHPDAR1	Failure type:
	1...		Solid storage data error.
	.1..		Intermittent storage data error.
	..1.		Solid SPF key error.
	...1		Intermittent SPF key error.
	 xxxx		Reserved.
58 (3A)	1	MCHPDAR2	Task in control when error occurred:
	1...		Wait pseudo task.
	.1..		Paging supervisor task.
	..1.		Master scheduler task.
	...1		System task.
	.... 1...		Problem program task.
	.... .1..		Current PSW disabled for I/O.
	xx		Reserved.
59 (3B)	1	MCHPDAR3	Location of failure:
	1...		Nucleus.
	.1..		SQA area.
	..1.		LSQA area.
	...1		PQA (VS1 Only).
	.... 1...		Pageable area.
	.... .1..		Fixed page area.
	.... ..1.		V=R address area.
	1		Critical area.
60 (3C)	1	MCHPDAR4	Reserved.
61 (3D)	1	MCHPDAR5	Requested operator awareness message:
	1...		Supervisor damage.
	.1..		Task abnormal termination.
	..1.		Damaged page now unavailable.
	...1		Damaged page now deleted.
	.... 1...		Task nondispatchable.
	.... .1..		Software recovery.
	xx		Reserved.
62 (3E)	1	MCHPDAR6	Footprints:
	1...		Change bit on in storage protect key of the affected page.
	.1..		Key in external page table.
	..1.		Page recursion.
	...1		Page fix active on low end.
	.... 1...		Translate error.
	.... .1..		No page exists.
	.... ..1.		Relocate off.
	1		Low end interface active.
63 (3F)	1	MCHPDAR7	Footprints - interfaces:
	1...		ABTERM interface.
	.1..		Translate interface.
	..1.		Find page interface.
	...1		Page dequeue interface.
	.... 1...		Page enqueue interface.
	.... .1..		Status function interface.
	.... ..1.		V=R release interface active.
	1		Post interface active.
64 (40)	4	MCHFSAV	Beginning failing location's virtual address.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
68 (44)	4	MCHFSEAV	Ending failing location's virtual address.
72 (48)	4	MCHPDARI	Instruction address at failure.
76 (4C)	4	MCHTCBCU	TCB pointer of interrupted task.
80 (50)	4	MCHTCBAF	Pointer to TCB of the affected task.
84 (54)	1	MCHSBSID	ID of subsystem (TSO=Hex '91').
85 (55)	1	MCHSBSF1	Flags used by the subsystem:
	1...		Subsystem module was in control at the time of the interruption.
	..1.		Subsystem ID flag.
	.x.x xxxx		Reserved.
86 (56)	1	MCHSBSF2	Flags used by the subsystem:
	1...		A solid storage failure exists.
	.xxx xxxx		Reserved.
87 (57)	1	MCHSBSF3	Flags used by the subsystem:
	1...		No action required by MCH.
	.1..		Subsystem has handled current task.
	..1.		Subsystem has handled the affected task.
	...x xxxx		Reserved.
88 (58)	1	MCHSUBAW	
	1...		TSO flag.
	.xxx xxxx		Reserved.
89 (59)	3		Reserved.
92 (5C)	1	MCHPGESW	Contains switches for paging supervisor: 1111 0000 Set page not available. 0000 1111 Key failure. 0000 0000 Set page invalid. 1111 1111 Translation needed.
93 (5D)	3	MCHSPR1	Reserved.
92 (5E)	2	MCHCLRLN	Length cleared by IGFVMCE0 upon initial entry (MCH FLIH).
96 (60)	8	MCHRPSW	Machine-check resume PSW.
96 (60)	1	MCHPSWM	PSW masks:
	.1..		(PER) program event recording mask.
	.... .1..		Translate mask.
	.... ..1.		I/O mask.
	1		External interruption mask.
	x.xx x...		Reserved.
97 (61)	1	MCHPSWWM	
	.... .1..		Machine-check interruption enabled.
	.... ..1.		Wait mask.
	1		Problem program state.
	xxxx x...		Reserved.
98 (62)	1	MCHPSWCP	Condition code and program mask.
99 (63)	1	MCHPSWSP	Segment protection.
100 (64)	4	MCHPSWAD	Address field (second word of PSW).
104 (68)	2	MCHINTEL	Field of indicators used by SHUT routine in the MCH Nucleus.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
104 (68)	1	MCHINTL0	
	1...		Multiple machine-check recursion.
	..1.		Put system down with scheduled message.
	...1		Error record written successfully.
	.... .1..		I/O interface active.
	.... ..1.		System damage detected.
	.x.. x..x		Reserved.
105 (69)	1	MCHINTL1	Reserved for SHUT routine information.
106 (6A)	1	MCHRELNO	Release number.
107 (6B)	1	MCHTMEVL	Time value for I/O Loop.
108 (6C)	3	MCHLOGIC	
108 (6C)	1	MCHLOGC0	
	...1		Model 135 indicator.
	.... 11..		Model 145 indicator.
	.... 1...		Model 155II/158 indicator.
	.... 1...		Model 165II/168 indicator.
	....		Unknown processor model code.
	xxx. ..xx		Reserved.
109 (6D)	1	MCHLOGC1	Reserved.
110 (6E)	1	MCHLOGC2	
111 (6F)	1	MCHNXMOD	Index value for TTR of next module to be loaded into transient area. IGFRVT-RVTTRS contains pointers to TTRS.
112 (70)	12	MCHTMSVE	Save area for timing.
124 (7C)	128	MCHPSA	Permanent storage assignment.
252 (FC)	64	MCHMLSAV	Register save area for transient modules.
316 (13C)	4	MCHSUBA	Subsystems running under the operating system.
320 (140)	4	MCHLEVEL	Level of macro source.
324 (144)	4	MCHTTRIN	Input TTR of specified transient module.
328 (148)	4	MCHUCB	Address of UCB for I/O operations.
332 (14C)	4	MCHIOB	First byte flags of I/O block.
336 (150)	4	MCHIOECB	Pointer to RMSECB.
340 (154)	8	MCHIOCSW	Last seven bytes = CSW.
344 (158)	4	MCHICCWS	Pointer to channel program.
352 (160)	4	MCHIODCB	Address of DCB.
356 (164)	8	MCHICBSP	Unused field in IOB.
362 (16C)	8	MCHIOBSK	SEEK field (equal to MBBCCHHR of module to be loaded).
370 (174)	4	MCHSPR2	Spare.
374 (178)	8	MCHCCWSR	Search CCW.
382 (180)	8	MCHCCWTC	TIC CCW.
390 (188)	8	MCHCCWRD	Read CCW.
398 (190)	8	MCHINT	Fields used by Error Recorder and Console Write routines.
398 (190)	1	MCHFCTL	Functional control byte.
	11..		
	1...		Invoke Error Recorder.
	.1..		Invoke Console Write routine.
	..1.		Emergency Recorder is running.
	...x xxxx		Reserved.
399 (191)	1	MCHRCTL	Record control byte.
	1...		CCH record to be written.
	.1..		No room for long record.
	..xx xxxx		Reserved.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
400 (192)	1	MCHERR	Error control byte.
	1...		LOGREC FULL message needed.
	.1..		I/O error message needed.
	..1.		Format error message needed.
	...1		MCHIBUF overlaid.
	.... 1...		LOGREC NEARLY FULL message.
	xxx		Reserved.
401 (193)	1	MCHRES	Reserved.
402 (194)	4	MCHIPTR	Pointer to message buffer.
406 (198)	84	MCHIBUF	MCH message buffer.
406 (198)	2	MCHIBWSC	Wait state code - first byte of MCHIBUF. Use of MCHIBUF for wait state messages only.
408 (19A)	2	MCHIBLNG	Length of text.
410 (19C)	80	MCHIBTXT	Text of message.
492 (1EC)	4	MCHLONG	Pointer to long record.
496 (1F0)	122	MCHWORK	Work area for error recording modules.
496 (1F0)	2	MCHHDREC	First field of MCHWORK used as identifier of LOGREC header.
498 (1F2)	4	MCHLWLIM	Starting CCHH of the extent of LOGREC.
502 (1F6)	4	MCHUPLIM	Ending CCHH of the extent of LOGREC.
506 (1FA)	1		Reserved.
507 (1FB)	7	MCHRECST	BBCCHHR of start of recording area.
514 (202)	2	MCHBYREM	Number of bytes remaining on track.
516 (204)	2	MCHTRCAP	Number of bytes which can be written.
518 (206)	7	MCHLSTTR	BBCCHHR of last record written on a track.
525 (20D)	2	MCHTKPCY	Tracks per cylinder.
527 (20F)	2	MCHWRNCT	Number of bytes remaining on early warning message track when LOGREC is 90% full.
529 (211)	1	MCHDEVCD	Device type code.
530 (212)	4	MCHWRNTR	CCHH of track representing LOGREC 90% FULL.
534 (216)	1	MCHWRNSW	
	1...		LOGREC 90% FULL message issued.
	.xxx xxxx		Reserved.
535 (217)	1	MCHFSTBY	Valid LOGREC header record.
536 (218)	8	MCHSRCH	CCW used for searches.
544 (220)	8	MCHTIC	CCW used to continue searches.
552 (228)	8	MCHCCW	CCW for READ or WRITE.
560 (230)	4	MCHEXIOB	IOB used by Error Recorder.
564 (234)	4	MCHEXEPT	Pointer to ECB.
568 (238)	8	MCHEXCSW	CSW field.
576 (240)	4	MCHEXCCW	Pointer to CCW.
580 (244)	4	MCHEXDCB	Pointer to DCB.
584 (248)	8		Reserved.
592 (250)	8	MCHEXSEK	MBBCCHHR for SEEK command
600 (258)	4	MCHEXECB	ECB for Error Recorder.
604 (25C)	7	MCHSKADR	BBCCHHR for SEEK.
611 (263)	1	MCHTFLAG	Switch for EOF WRITE.
612 (264)	4	MCHMCSRT	Return address for MCS systems.
612 (264)	4	MCHCCHAD	Address of CCH record.
616 (268)	2	MCHCCHLN	Length of CCH record.
618 (26A)	1	MCHABRNO	Number of ABREC buffers.
619 (26B)	216	MCHABREC	Abbreviated record buffer.
835 (343)	53	MCHMAINT	Reserved.

MCH RECOVERY VECTOR TABLE

The RVT (recovery vector table) is an MCH data area containing addresses and flags used during MCH processing. The address of the RVT is in the CVT (communications vector table) at the symbolic location CVTRMS.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
0 (0)	4	RVTNUCAD	Pointer to MCH Nucleus.
4 (4)	4	RVTSMCH	Pointer to Soft MCH.
8 (8)	4	RVTPOST1	Pointer to MCH Post routine.
12 (C)	4	RVTVCOMM	Pointer to MCH common area.
16 (10)	4	RVTWSMSG	Pointer to MCH Nucleus wait state entry.
20 (14)	4	RVTCCHRC	Pointer to CCH inboard record.
24 (18)	4	RVTNIPRM	Fields passed to NIP for systems with less than 192K bytes.
24 (18)	4	RVTTRANS	Pointer to MCH transient area.
28 (1C)	4	RVTLOADA	Pointer to Module Loader.
32 (20)	4	RVTINLOG	Pointer to model-dependent logout area. RVTINLOG is initialized by NIP.
36 (24)	4	RVTBUILD	Pointer to record buffer build area. This field is initialized by NIP.
40 (28)	4	RVTTTRS	Pointer to MCH TTR table. This field is initialized by NIP.
44 (2C)	4	RVTSHUT	Pointer to MCH SHUT routine in MCH Nucleus.
48 (30)	4	RVTPRIMY	Pointer to primary logout area. This field is initialized by NIP.
52 (34)	4	RVTSECND	Pointer to secondary logout area. This field is initialized by NIP and used only for the Model 165II and 168.
56 (38)	4	RVTIOSEP	Pointer to IOS entry point for MCH/IOS interface.
60 (3C)	4	RVTIOSIH	Pointer to IOS First-Level Interrupt Handler.
64 (40)	4	RVTMSB	Pointer to machine status block (MSB).
68 (44)	4	RVTSIRBA	Address of SIRB.
72 (48)	8		Reserved.
80 (50)	4	RVTIOSNS	Pointer to IOS nest switch.
84 (54)	4	RVTPAGES	Address of low end system (less than 192K) page-in module.
88 (58)	4	RVTWSFLG	Wait state interface flags.
88 (58)	1	RVTWSFL0	First byte of RVTWSFLG - CCH wait state interface flag:
	1...		Machine-check in CCH.
	.1..		Unrecoverable channel error.
	..xx xxxx		Reserved.
89 (59)	1	RVTWSFL1	System wait state interface flag.
	1...		Paging supervisor request.
	.1..		Dynamic support system request with no message.
	..1.		Dynamic support system request with message.
	...x xxxx		Reserved.

	Displacement	Bytes and Bit Patterns	Field Name	Field Description
	90 (5A)	1	RVTWSFL2	MCH wait state interface flag.
		1...		MCH in process.
		.xxx xxxx		Reserved.
	91 (5B)	1	RVTWSFL3	Reserved wait state interface flag.
	92 (5C)		RVTBUFAD	Pointer to wait state message buffer.
	96 (60)	4	RVTSBSYS	Subsystem interface flags.
	100 (64)	4	RVTTVT	Address of test vector table.
	104 (68)	4	RVTLEVEL	Latest level of source macro.
	108 (6C)	4	RVTLNG	Length of RVT.
	108 (6C)	4	RVTRESUM	Address of resume PSW.
	112 (70)	4	RVTIOREG	Address of I/O FLIH register save area.
VS1 only	116 (74)	4	RVTSVF	Type-1 SVC switch.
VS1 only	120 (78)	4	RVTPGESW	IOS page exception switch.
VS1 only	124 (7C)	4	RVTPDISW	Pseudo disable switch.
VS1 only	128 (80)	4	RVTSIRBE	SIR branch entry address.
VS1 only	132 (84)	4	RVTTP1EX	Type-1 exit entry point.
VS1 only	136 (88)	4	RVTSIRFQ	Long TCB queue.
VS1 only	140 (8C)	4	RVTLNG	Length of RVT.

MACHINE STATUS BLOCK

The machine status block contains hardware counters and values that are needed to keep track of the status of the processor.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
0 (0)	1	MSBCPUID	Target of STORE CPU ID instruction.
1 (1)	3	MSBSERNO	CPU serial number.
4 (4)	2	MSBMODEL	Model number.
6 (6)	2	MSBLOGSZ	Size of the extended logout.
8 (8)	4	MSBCOUNT	Soft error counter.
12 (C)	4	MSBTHRLD	Soft error threshold value.
16 (10)	4	MSBBUFER	Buffer failure counter.
20 (14)	4	MSBBUFTH	Buffer failure threshold counter.
24 (18)	4	MSBMODE	Quiet/record mode control flags.
24 (18)	1	MSBMODE1	
	1...		Record mode for instruction retry.
	.1..		Buffer enabled.
	..1.		Quiet mode for main storage.
	...1		Threshold mode for control storage.
	.... 1...		Diagnose instruction for placing main storage in record mode not issued.
	.... .1..		Error frequency limit overflow condition.
	xx		Reserved.
25 (19)	3		Reserved.
32 (20)	8*	MSBECCEN	Enable the Model 158 ECC.
40 (28)	8*	MSBECCDS	Disable the Model 158 ECC.
48 (30)	8*	MSBBUFE1	Enable the Model 158 buffer.
56 (38)	8*	MSBBUFD1	Disable the Model 158 buffer.
64 (40)	8	MSBBUFE2	Enable the Model 168 buffer.
72 (48)	8	MSBBUFD2	Disable the Model 168 buffer.
80 (50)	8	MSBECCE2	Enable the Model 168 ECC.
88 (58)	8	MSBECCD2	Disable the Model 168 ECC.
96 (60)	4	MSBLEVEL	Change level of IGFMSB macro instruction.
100 (64)	8	MSBSPARE	Reserved.
108 (6D)	2	MSBLNG	Length of MSB.

*These fields are 4 bytes long followed by 4 blank bytes for doubleword alignment.

MCH LONG RECORD

See Figure MCH-9 for an explanation of the various MCH records.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
0 (0)	24	Header	
0 (0)	1	Record type	Record ID. X'10' = MCH record.
1 (1)	1	Operating	System ID:
	111.	system	000-OS. 001-DOS. 010-OS/VS. 011-CP67.
	...1		Release level.
	 xxxx		Reserved.
2 (2)	6	Switches	
2 (2)	1		Record independent information.
	1...		Multiple record.
	.1..		System/370 machine.
	.... 1...		Time macro instruction used (HHMMSS).
	.... .1..		Multiprocessing.
	.... ..1.		BC mode version 2.
	1		EC mode.
	..xx		Reserved.
3 (3)	1		Record dependent information.
	1...		Short form of record.
	.1..		Record incomplete.
	..1.		MCH terminate system.
	...1		First record of two record recording.
	.... 1...		Channel record included.
	.... .1..		Portion of data overlaid.
	.... ..1.		External machine check.
	1		Model 67, Mod 2.
4 (4)	2		Reserved.
6 (6)	1		Record count:
	1111		Sequence number of physical record.
	 1111		Total number of physical records in this logical record.
7 (7)	1		Reserved.
8 (8)	8		Date and time.
16 (10)	8		CPU ID.
16 (10)	1		Reserved.
17 (11)	3		CPU serial number.
20 (14)	2		CPU model number.
22 (16)	2		Maximum machine-check extended logout.
24 (18)	8	Program ID	Program ID.
32 (20)	8	Job ID	Job ID.
40 (28)	8	MCKPSW	Machine-check old PSW
48 (30)	280	MCKFXLO	Fixed logout.
328 (148)	Variable	MCKETLO	Extended logout.
Variable	80	MCHSTATD	Damage assessment.

MCH TESTING VECTOR TABLE

The TVT (testing vector table) is used in MCH for testing only. It indicates whether simulation of machine conditions that cannot be artifically created on a real machine are allowed. It also indicates when MCH testing of the simulation mode is allowed. The address of the TVT is in the RVT (recovery vector table) at the symbolic location RVTTVT.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
0 (0)	4	TVTTEST	TVT flag word.
	1...	TVTSOLID	Solid MCI simulation active.
	.1..	TVTSIM	MCI full simulation active.
4 (4)	4	TVTPCN	Address of MCH program check wait state new PSW.
8 (8)	4	TVTMCN	Address of MCH MCI wait state new PSW.
12 (C)	4	TVTWAIT	Address of wait state PSW instruction.
16 (10)	4	TVTRTN	Address of return PSW instruction.
20 (14)	4	TVTWPSW	Address of wait state PSW.
24 (18)	4	TVTRPSW	Address of return PSW.
28 (1C)	.... ..1.	TVTLEVL	TVT level indication.
29 (1D)	3	TVTSPARE	Reserved.

This section is intended to aid in locating errors in MCH. Discussions on register conventions, problems that may exist when the IGF910W message appears, the MCH history table, and messages and wait state codes are included.

REGISTER CONVENTIONS

Figure MCH-14 shows how MCH uses its registers. Three modules are exceptions to these conventions: the Error Recorder, Console WRITE, and the MODE Command modules. They follow the conventions of the operating system.

Register	Used by MCH as
0-8	Work registers
9	Address of the RVT
10	Pointer to the communications vector table
11	Pointer to the MCH common area
12	Nucleus base register
13	Address of the save area in the I/O interface
14	Return address into the I/O interface from transient modules
15	Transient module base register

Figure MCH-14. Register conventions

POSSIBLE PROBLEMS INDICATED BY MESSAGE IGF910W

When the IGF910W message appears, the following can be done to isolate the cause of the error:

1. Verify the interruption code. If the interruption code is invalid, the error was caused by a hardware malfunction.

2. Check whether 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 so, and if 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 the error is a program check in MCH. The history table in the MCH independent common area can then be checked to determine in which MCH module the program check occurred. The contents of the registers at the time the program check occurred can be found at location X'268' for VS1, and X'260' for VS2.

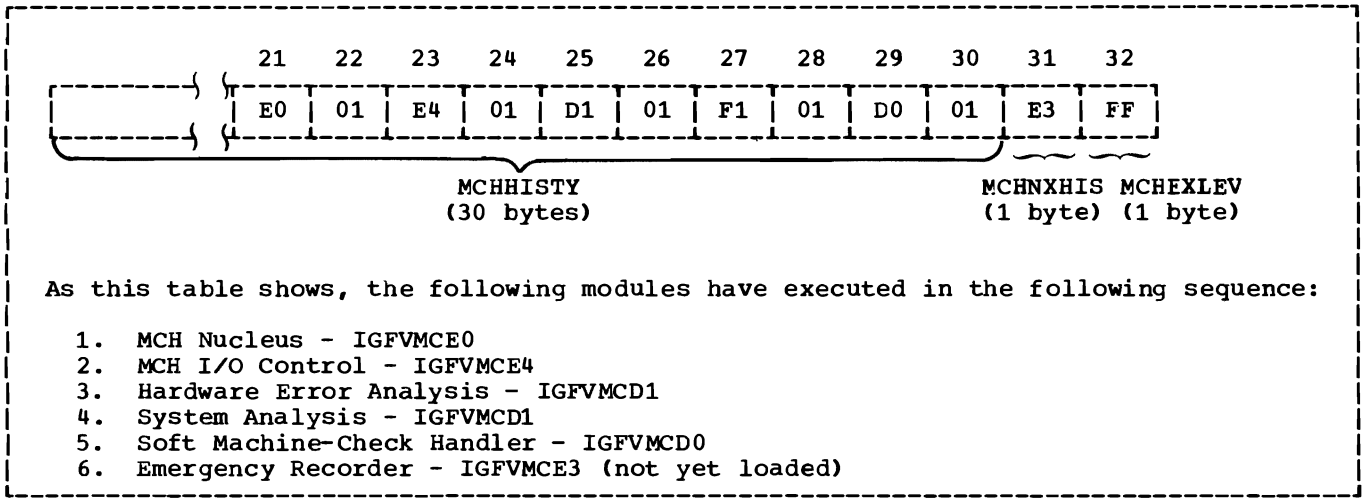
MCH HISTORY TABLE

The MCH history table (MCHHISTY in the MCH independent common area) can be used to determine which modules have been executed since the time of the machine-check interruption and the sequence in which they were executed. The modules are identified by their IDs and level numbers.

When MCH is initially entered, the Nucleus puts its own ID and level numer in the two bytes following the table (MCHNXHIS and MCHEXLEV). When a successor module is specified, the I/O interface module moves this data into the last two bytes of the table, bytes 29 and 30. The ID of the successor module is put into MCHNXHIS and a hexadecimal 'FF' is put into MCHEXLEV. After the module is successfully loaded, the successor module places its level number into MCHEXLEV. This process is continued. As each module is loaded, all the previous IDs and level numbers are moved two bytes (toward a lower address). The I/O interface module always places the ID of the successor module into MCHNXHIS and the successor module always overlays the 'FF' of MCHEXLEV with its own level number. Figure MCH-15 shows the use of the MCH history table.

MCH MESSAGES AND WAIT STATE CODES

Figure MCH-16 lists the messages that are produced by MCH and the modules that request each message. The code, where shown, is the wait state code informing the operator of an error condition that caused the system to be placed in the wait state. The message will be lost if an I/O error occurs while writing to the console or if the control blocks for a graphic console have been paged out.



As this table shows, the following modules have executed in the following sequence:

1. MCH Nucleus - IGFVMCE0
2. MCH I/O Control - IGFVMCE4
3. Hardware Error Analysis - IGFVMCD1
4. System Analysis - IGFVMCD1
5. Soft Machine-Check Handler - IGFVMCD0
6. Emergency Recorder - IGFVMCE3 (not yet loaded)

Figure MCH-15. The MCH history table shows the order in which MCH modules have been executed.

Message	Code	Requested By	Cause of Wait State Code
IGF900W SUPERVISOR DAMAGE, [RUN SEREP,] RESTART SYSTEM	A05 A05	IGFVMCE0 IGFVMCF4	MCI in supervisor MCI in supervisor
IGF910W SYSTEM INTEGRITY LOST, [RUN SEREP,] RESTART SYSTEM	A01 A03 A11	IGFVMCE0 IGFVMCE0 IGFVMCE0	Hard on hard error Program check in MCH Erroneous MCIC
IGF920I TASK jobname - ABNORMAL TERMINATION SCHEDULED		IGFVMCF4	
IGF924E TASK jobname - SET NON-DISPATCHABLE		IGFVMCF4	
IGF930W CATASTROPHIC I/O FAILURE, [RUN SEREP,]	A04 0A0C	IGFVMCE0 IGFVMCF6	I/O error in MCH Unable to page module in, VS1 system is less then 192K bytes.
IGF940I TSO [USER] ABNORMAL [SUBSYSTEM] TERMINATION SCHEDULED		IKJVAM00 (TSO Subsystem)	
IGF944I SUCCESSFUL RECOVERY BY [HARDWARE] [SOFTWARE]		IGFVMCE5	
IGF950W TIMING FACILITY FAILURE, [RUN SEREP,] RESTART SYSTEM	A16 A17	IGFVMCE0 IGFVMCE0	Time of day clock, clock comparator, or CPU timer Interval timer error
IGF952I BUFFER BLOCKS DELETED		IGFVMCD0	

Figure MCH-16 (Part 1 of 2). MCH messages and wait state codes

Message	Code	Requested by	Cause of Wait State Code
IGF953I STATUS: INSTRUCTION RETRY-[QUIET] [RECORD], MAIN STORAGE-[QUIET] [RECORD], CONTROL STORAGE-[QUIET] [THRESHOLD], BUFFER-[BLOCKS DELETED]		IGFVMCD4 (IGF2603D)	
IGF954E SYS1.LOGREC DATA SET [FORMAT ERROR] [RECORDING FAILURE] [NEARLY FULL] [IS FULL]		IGFVMCE2	
IGF955I [CONTROL STORAGE] [INSTRUCTION RETRY] [MAIN STORAGE] NOW IN [QUIET/THRESHOLD] [QUIET/RECORD] MODE		IGFVMCD0 IGC2603D (IGFVMCD4)	
IGF961I DAMAGED PAGE NOW [UNAVAILABLE] [DELETED]		IGFVMCF4	
	0A0A	IGFVCCHC	Catastrophic CCH error.

Figure MCH-16 (Part 2 of 2). MCH messages and wait state codes

PART 2: CHANNEL-CHECK HANDLER

PURPOSE OF CCH

CCH (Channel-Check Handler) aids the I/O supervisor in recovering from channel errors and informs the operator or system maintenance personnel of the occurrence of channel errors.

CCH provides the device-dependent ERPs (error recovery procedures) of the I/O supervisor with the information needed to set up for retrying a channel operation that has failed; without the information provided by CCH, the ERPs would have to consider most channel errors as system termination conditions.

CCH provides the operator or system maintenance personnel with information about channel errors that enables them to keep statistics about the channel or helps them to bring about recovery from system termination conditions.

HOW CCH ACHIEVES ITS PURPOSE

CCH receives control from the I/O supervisor when a channel data check, channel control check, or interface control check occurs. CCH produces an ERPIB (error recovery procedure interface block) for the ERP and a record to be written in the SYS1.LOGREC data set for the operator or system maintenance personnel. A copy of the record may be obtained by using the IFCEREP0 utility program. A message describing the channel error is issued each time a record is written into SYS1.LOGREC.

Channels Supported by CCH and How CCH Uses the Information Provided by These Channels

CCH supports the 2860, 2870, 2880, 135, 145, and 155II channels. The 2860, 2870, and 2880 stand-alone channels can be attached to System/370 Model 165II. Each of these channels produces a logout at a specific location in main storage (see Figure CCH-1) when a channel error occurs. This logout contains information about the cause of the channel error and the conditions under which it occurred. CCH analyzes the logouts of these channels and places the results of its analysis into the last three bytes of the ERPIB.

System/370 Models 135, 145, 155II, and 158 have integrated channels. The circuitry analyzes channel errors and presents the results of its analysis in the ECSW (extended channel status word). The ECSW contains the same information that CCH produces for the stand-alone channels. For the integrated channels, CCH moves the last three bytes of the ECSW into the last three bytes of the ERPIB.

The ERPIB

Besides the analysis of the logout or the copy of the ECSW, the ERPIB contains

Channel	Logs Out at: Location	Logs Out at: Location Pointed to by	Logs Out for ____ Bytes	CSW at	ECSW at	Unit Address at
2860	304		24	64		
2870	304		24	64		
2880		172	112	64		
135	256	172	24	64	176	184
145		172	96 maximum	64	176	184
155II/ 158	155II/158 channels do not logout			64	176	184

Note: All locations are decimal.

Figure CCH-1. Channel logout locations and other pertinent locations referred to by CCH

program flags and the address of the unit control block. The ERPIB is illustrated in CCH Section 5.

The Channel Error Inboard Record

The inboard record contains a copy of the last half of the ERPIB, information about the status of the system at the time of the channel error, and a copy of the channel logout. The inboard record is illustrated in CCH Section 5.

I/O SUPERVISOR AND MCH PROCESSING OF CHANNEL ERRORS

CCH performs no error recovery; it does not retry any operations, make any changes to the system, or even write its own records or message. Recovery from channel errors is performed by the routines of the I/O supervisor. Termination of the system because of channel errors is performed by the routines of the Machine-Check Handler.

Figure CCH-2 is a simplified illustration of the flow of control between the Channel-Check Handler, the Machine-Check Handler, and certain I/O supervisor routines. The following discussion provides some background on the routines that work with CCH. They include I/O supervisor routines that handle normal I/O operations and channel errors, and the ERPs that process channel errors and use the information CCH provides in the ERPIB.

I/O Supervisor Normal I/O Processing

When an application or a system program requests an I/O operation, the I/O supervisor executes a Start I/O instruction which causes a chain of channel commands to be executed. After execution of all the commands, an I/O interruption occurs. If the I/O supervisor determines that there were no errors, it notifies the requester that the I/O was successful. The I/O supervisor then passes control to the Dispatcher.

I/O Supervisor Channel Error Processing

As illustrated in Figure CCH-2, if a channel error occurs during execution of a command, or on the interruption following command execution, normal processing is suspended while the error is handled. The I/O supervisor passes control to CCH to analyze the error and to construct an ERPIB. The I/O supervisor then schedules an ERP.

The ERP determines whether the operation can be retried. If it cannot (because it is unretryable or because it has already been retried the specified number of times), the ERP passes control to the WTO (Write-to-Operator) routine, which issues a message declaring a permanent error. The WTO routine then passes control to the Statistics Update routine.

If the operation can be retried, the error recovery procedure prepares for the retry and returns control to the I/O supervisor, which retries the operation. After the retry, the ERP again gains control. If the retry was successful, the ERP passes control to the Statistics Update routine. If the retry was unsuccessful, but the operation can be retried again, the ERP returns control to the I/O supervisor. The Statistics Update routine updates the statistics table for the device and passes control to the Outboard Recorder.

The first load of the Outboard Recorder checks whether it was entered because of a permanent error. If so, it formats an outboard record to record the error as permanent. It updates the statistics for SYS1.LOGREC data set by adding the counts recorded in the statistics table to the statistical values already in SYS1.LOGREC. It passes control to the second load of the Outboard Recorder.

The second load of the Outboard Recorder writes any outboard records and updates the statistics in SYS1.LOGREC. It then determines whether it was entered because of a channel error. If so, it writes into SYS1.LOGREC all inboard records produced by CCH and again updates statistics on SYS1.LOGREC. The Outboard Recorder then issues message IFB002I to inform the operator that a channel error has occurred, and clears the inboard record area. When finished (or if there were no channel inboard records to be written), the Outboard Recorder gives up control via SVC 3 (EXIT).

How an ERP Handles a Channel Error

If the ERP determines that it was entered because of a channel error, it searches for the ERPIB that CCH constructed for that error. It searches the ERPIB table for a UCB address that matches the UCB address in the RQE (request element). (The UCB address is in the second, third, and fourth bytes of the ERPIB.) If a matching UCB address is not found, the error must be treated as permanent, because the ERP cannot set up for the retry without an ERPIB (except for channel data checks, for which the ERP does not require such information to set up for a retry). If the ERPIB is found, the ERP makes certain tests, evaluates the termination and sequence codes in the ERPIB, and verifies the channel status word. These steps are illustrated in Figure CCH-3.

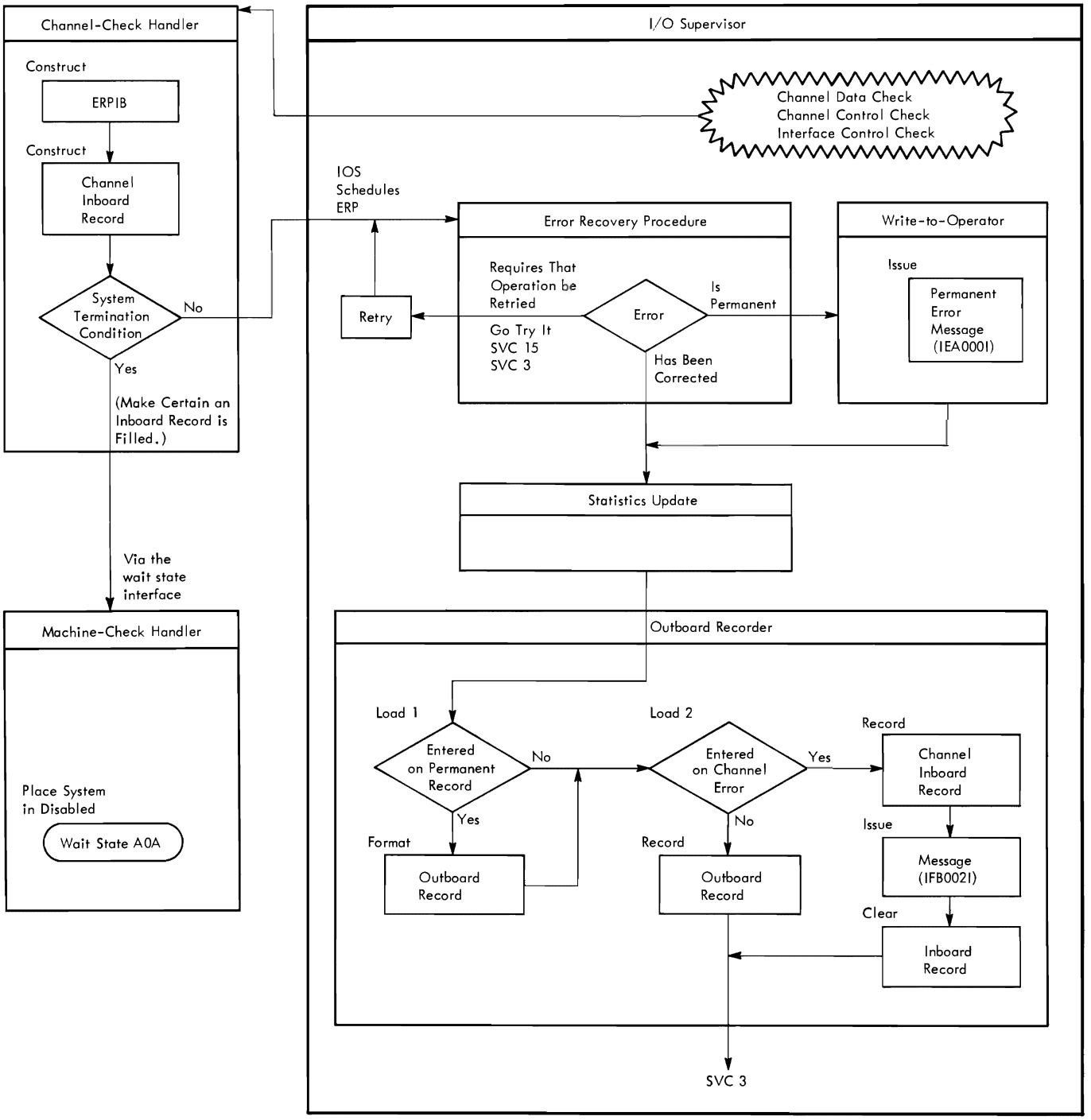


Figure CCH-2. Overview of CCH-MCH-IOS processing of I/O errors

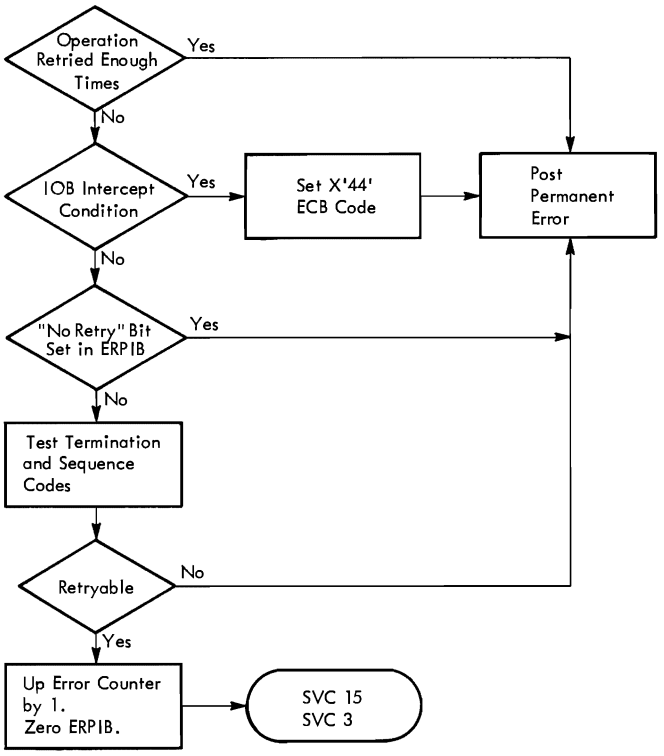


Figure CCH-3. Standard tests made by an ERP when entered following a channel error

It is important to remember that CCH is channel dependent, whereas the ERPs are device dependent. CCH may analyze the error, determine that the channel error can be recovered from, and pass its information to the ERP in the ERPIB. When the ERP examines the information in the ERPIB, however, taking into consideration the device that is involved in the error, it may determine that a retry cannot be attempted. Or, after several retries, the ERP may determine that the required number of retries for that device has been made without success and indicate that a permanent error exists. (CCH may have determined already that retry is impossible; it would have (1) not put the ERPIB in the ERPIB table or (2) set the "no retry" bit in the ERPIB.)

Each ERP has three exits:

- For errors that are permanent, it passes control to the Write-to-Operator routine to indicate to the operator that a permanent error has occurred.
- For errors for which operations must be retried or for which repositioning is necessary, it causes SVC 15 (ERREXCP) and SVC 3 (EXIT) to be issued.
- For errors that have been corrected, it passes control to the Statistics Update routine.

Before the ERP passes control, it clears the ERPIB to 0's, regardless of the result of the retry attempt. This frees the ERPIB and prevents the possibility of two identical UCB addresses existing in the ERPIB table.

CCH FIXED STORAGE REQUIREMENTS

Following nucleus initialization, the CCH modules reside in the nucleus. The amount of storage required by CCH depends on the type of channel that is included in the system. Figure CCH-4 illustrates the amount of storage needed for the possible channel configurations.

Channels		Number of Bytes Needed to Support Each Channel			
		CCH Central Module (IGFVCCHC)	+ Analysis Routine	+ Logout Area	+ Inboard Record Area
135		2100	105	24	351
145		2100	164	96	567
155II and 158		2100	62	0	279
Model 165II and 168	2860	2100	1026	24	339
	2870		1026		
	2880		1026	112	603

Therefore:

If the system includes:	The total bytes needed for CCH are:
135 channels	2100+105+24+351 = 2580
145 channels	2100+164+96+567 = 2927
155II/158 channels	2100+62+0+279 = 2441
2860 or 2870 channel	2100+1026+24+339 = 3489
2860 and 2870 channels	2100+2(1026)+24+339 = 4515
2880 channel	2100+1026+112+603 = 3841
2860 & 2880 channels or 2870 & 2880 channels	2100+2(1026)+24+112+603 = 4891
2860 & 2870 & 2880 channels	2100+3(1026)+24+112+603 = 5917

- Only 24 bytes of logout area are needed for use by the 2860 and/or the 2870.
- Only 339 bytes of inboard record area are needed for use by the 2860 and/or the 2870.
- If the 2880 channel is in a system with a 2860 or 2870 or both, only 603 bytes of inboard record area are needed, but the 2880 needs its own 112 bytes of logout area in addition to the 24 bytes for the 2860 and/or 2870.

Figure CCH-4. CCH fixed storage requirements

CCH SECTION 2: METHOD OF OPERATION

This section describes the functions of CCH and how CCH performs these functions. Many references are provided to material in other sections of this publication and to the CCH program listing. Two types of information are presented:

- Text that discusses the design of CCH.
- Method of operation diagrams that show the functions performed by CCH in the order that they occur. The module name and label of each processing step are included in the implementation notes for each diagram. These diagrams can be used for quick reference when reviewing CCH functions. Diagram CCH-1 shows how the method of operation diagrams for CCH are organized.

FUNCTIONS OF CCH

The Channel-Check Handler has two primary functions: analyzing the channel failure and recording the results of this analysis for use by an ERP and the operator or system maintenance personnel.

CCH first analyzes the status of the channel, as shown in Diagram CCH-3. Using the information in the channel logout or the extended channel status word, and considering the type of channel error and which I/O supervisor routine was executing when the channel error occurred, CCH must determine whether the channel error is recoverable. If CCH determines that the channel error is recoverable, it returns control to the I/O supervisor at the correct displacement to allow the I/O supervisor to schedule an ERP. If CCH determines that the channel error is not recoverable, but that operations may continue (for example, when a channel error occurs in the Sense routine), it returns control to the I/O supervisor at the "no retry" displacement or with the "no retry" indicator set in the ERPIB. If the error is asynchronous and cannot be associated with a specific task, CCH returns control to the I/O supervisor at the "alternate return" displacement, so that the I/O supervisor can restart the channel, essentially ignoring the error.

If CCH determines that a system termination condition exists, it ensures that an inboard record is complete and then passes control to the Machine-Check Handler to write the record and to place the system in a disabled wait state.

Also, in the process of analyzing the channel error, CCH must decide whether to place the ERPIB in the ERPIB table for the ERP to use in setting up for retrying the operation. CCH does not place the ERPIB in this table if the channel error is a system termination condition or if the ERP can set up for retrying the operation without an ERPIB.

CCH passes the results of its analysis to the I/O supervisor in the ERPIB; the results of the analysis determine the point in the I/O supervisor to which control is returned. The I/O supervisor can then decide whether to schedule an ERP. The ERP, using the information in the ERPIB, makes the final decision as to whether the device is in a recoverable position. Diagram CCH-4 shows the steps for constructing an ERPIB.

Whatever the results of its analysis, CCH always attempts to provide the operator or maintenance personnel with a record of the channel error. Diagram CCH-5 shows the steps for constructing a channel error inboard record.

CCH ERROR ANALYSIS

The first step in analyzing the channel error is to determine what bit was set in the channel status word -- bit 44 (channel data check), bit 45 (channel control check), or bit 46 (interface control check). Using this information, CCH checks for a system termination condition by determining if (1) the channel is supported by CCH, (2) the logout or ECSW was stored completely, and (3) the UCB address is valid, as well as by using the information in the logout or the ECSW. Figure CCH-5 illustrates the differences in CCH execution for the various channels and channel checks.

CCH always builds a work ERPIB. The ERPIB table has space for five ERPIBs. The work ERPIB is intended to be placed in the ERPIB table for use by an ERP and in the inboard record for use by the operator or system maintenance personnel. In some cases, however (system termination conditions, nonretryable conditions, or when the error recovery procedure does not need an ERPIB to set up retrying the operation), the ERPIB is not placed in the ERPIB table, but is built only for inclusion in the inboard record.

Actions Taken by CCH or Channel	2860			2870			2880			135			145			155II/158		
	CDC	CCC	ICC	CDC	CCC	ICC	CDC	CCC	ICC	CDC	CCC	ICC	CDC	CCC	ICC	CDC	CCC	ICC
Channel logs out.		x	x		x	x	x	x	x		x	x		x	x			
CCH analyzes the logout.		x	x		x	x		x	x									
CCH moves the ECSW.											x	x		x	x		x	x
CCH constructs a work ERPIB.	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
CCH places the ERPIB in a list for the ERP (unless a system termination condition exists or the ERP does not need an ERPIB to set up for retrying the operation).		x	x		x	x		x	x		x	x		x	x		x	x
CCH creates the ERPIB to be placed in the inboard record only.	x			x			x			x			x			x		
CCH attempts to construct an inboard record.	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
CCH places the logout in the inboard record.		x	x		x	x	x	x	x		x	x		x	x			

CDC (channel data check) is indicated by bit 44 of the channel status word.
CCC (channel control check) is indicated by bit 45 of the channel status word.
ICC (interface control check) is indicated by bit 46 of the channel status word.

Figure CCH-5. Differences in CCH execution for the various channels and channel checks

CCH also always attempts to build an inboard record. The inboard record area has room for three inboard records. Under certain conditions, however, space for an inboard record is not available. When space for one is available (or when CCH clears a space for use because the channel error is a system termination condition and the Machine-Check Handler must have a record to write into SYS1.LOGREC), CCH copies into it the channel logout (for those channels that log out), the last four bytes of the ERPIB, and other information of interest to the operator or system maintenance personnel.

Passing Control to CCH from the I/O Supervisor

CCH receives control from the I/O supervisor through the System Environment Recording Interface routine (SERR04). The channel error could have been intercepted at any one of seven points within five routines of the I/O supervisor. To allow CCH to determine where the channel error was intercepted, the executing routine of the I/O supervisor places a value into a field labled CATAPP before an interception point is reached. If no channel error occurs, the value is overlaid by the value of the next routine that gains control; if a channel error does occur, the proper value is in the byte when CCH gains control. This procedure is illustrated in Figure CCH-6.

Returning Control to the I/O Supervisor

The UCB Search routine of IGFVCCHC returns control to a point in the I/O supervisor. The appropriate CCH interception-dependent routine, upon receiving control from the ERPIBSET routine of IGFVCCHC, has set a displacement in register 12; the displacement depends on the information collected by the CCH ERPIBSET routine and the appropriate analysis routine. The UCB Search routine adds this displacement to the I/O supervisor base return address in register 14 before returning control to the I/O supervisor. The return displacements are as follows:

Name	Value	Return for
CCHRTHIO	0	Halt I/O
CCHRTSIO	0	Start I/O or Test I/O
CCHRTINT	4	Interruption
CCHRTSNN	12	Sense, no retry
CCHINT2	16	CCH received control on an asynchronous interruption for which no UCB or RQE could be found (the interrupt may have been for an

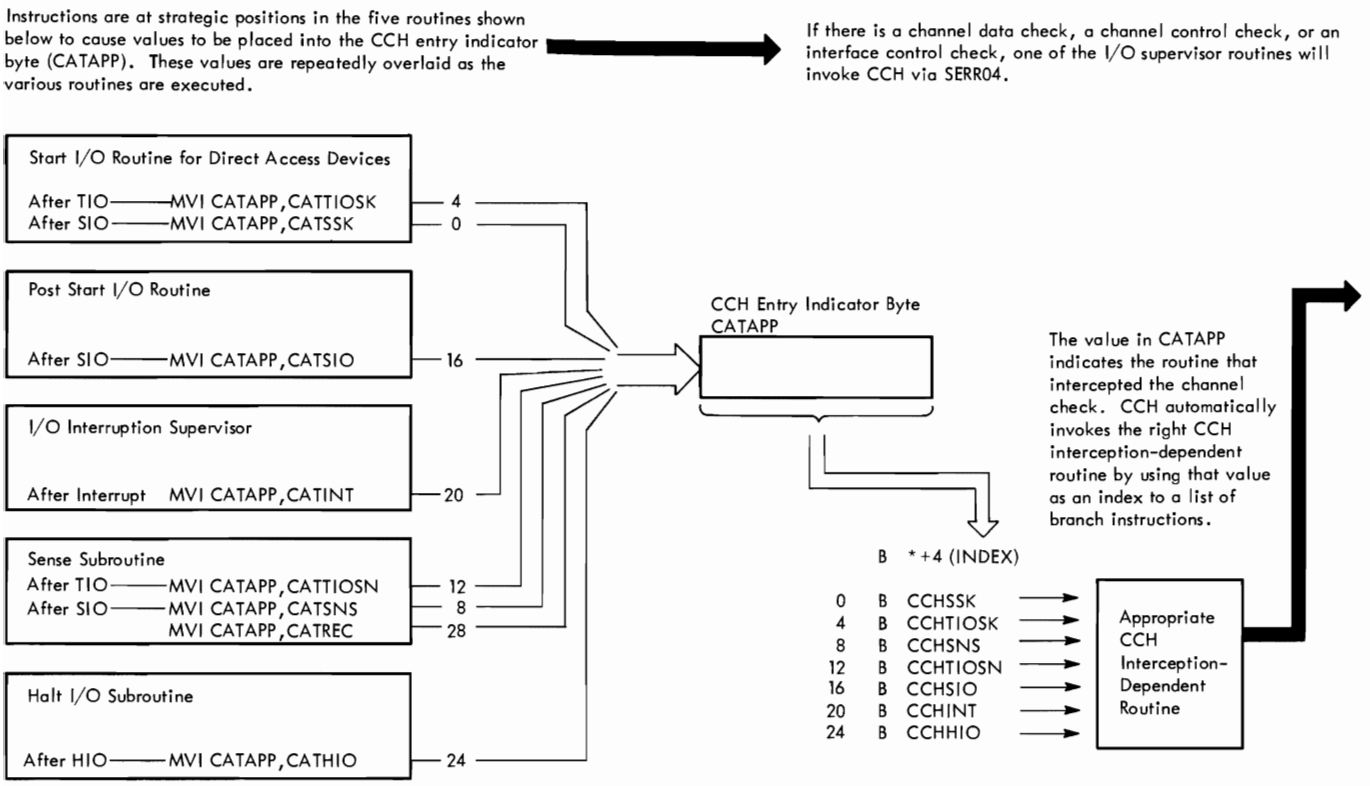


Figure CCH-6. How CCH knows from what I/O supervisor routine it was entered

address that does not exist). This return allows the I/O supervisor to restart the channel without passing control to an ERP; the interruption is essentially ignored.

Interfacing with the Machine-Check Handler

As soon as CCH is entered following a chanel error, it sets bit 0 of byte 0 of the RVTWSFLG field in the RVT (recovery vector table) to 1 to identify itself to the Machine-Check Handler should a machine check occur during CCH exeuction. The UCB Search routine of IGFVCCHC resets the bit to 0 just before it releases control.

If the UCB Search routine of IGFVCCHC finds that CCH has determined the channel error to be a system termination condition, it sets bit 1 of byte 0 of RVTWSFLG to 1 to indicate the situation to the Machine-Check Handler, and sets bit 0 of byte 0 to 0. The UCB Search routine then places a X'0F' in the machine-check new PSW (location 115) for SEREP and branches to the Machine-Check Handler using the address in RVTSHUT. It places the system in a disabled wait state.

The address of the inboard record is contained in RVTCCHRC. Figure CCH-7 illustrates the CCH-MCH interface in the RVT.

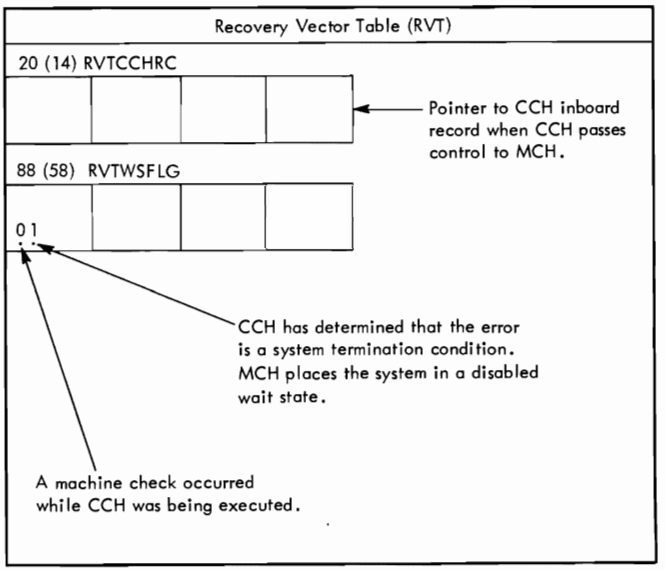


Figure CCH-7. CCH-MCH interface in the RVT

Determining Whether the Channel Has Logged Out or Stored the ECSW Successfully

The CCH initialization module (IGFVCCIN) sets the logout area to all 1's at nucleus initialization. When entered, the appropriate CCH analysis module checks the logout area. If any word of the logout area contains all 1's the channel has not logged out successfully. After analyzing the logout, the CCH analysis module restores the logout area to all 1's so that the test can be repeated if another channel error occurs.

Similarly, the CCH initialization module sets the first byte of the ECSW area to all 1's. When entered, the CCH analysis module checks the first bit of the ECSW; if it is 1, the ECSW has not been stored successfully. After moving the ECSW to the ERPIB, the CCH analysis module restores the first byte to all 1's so that the test can be repeated.

Why There Might Not Be an ERPIB Available

Channel errors could be occurring faster than error recovery procedures can be scheduled; therefore, all five ERPIBs could be in use. If all five ERPIBs are in use, CCH searches the error logical channel queue for a UCB address in one of its RQEs that matches the UCB address in one of the ERPIBs. If CCH does not find matching UCB addresses, CCH frees one of the five ERPIBs by clearing it to 0's. Each of the five ERPIBs is thus compared and freed if possible until all ERPIBs that can be freed are freed. Finally, CCH searches the ERPIB table for the last available ERPIB to use.

There is also the possibility that there is no ERPIB available because the ERP neglected to clear the ERPIB when it was finished processing the error; this should never occur.

Why There Might Not Be an Inboard Record Available

When the Data Collection routine of IGFVCCHC finds that all three inboard records are full, it sets the "no record available" switch to indicate that an inboard record will not be produced for this error. Only if the "system termination" switch is set does the Data Collection routine clear one of the inboard records and use it.

This situation (of all records being full) occurs when CCH is entered four or more times for what is essentially the same channel error. For an operation that must be retried a maximum of ten times by an ERP, for example, CCH can be entered 11 times for the same kind of channel error on the same failing operation. Each time, a new ERPIB is created, used, and cleared. However, since inboard records are pending until all retries of the operation have been made, and since there are only three inboard records available to CCH for recording any one error, only the first three channel errors (the first error and errors on the first two retries) are recorded as inboard records.

CCH METHOD OF OPERATION DIAGRAMS

These Method of Operation Diagrams show the functions of CCH and relate these functions to the exact modules and entry points in the code where they are performed.

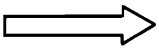
These particular diagrams are known as HIPO diagrams because they show Hierarchy as well as Input, Processing, and Output. The first diagram is a visual table of contents. The rest of the diagrams are arranged in a hierarchy according to the level of detail.

Read the diagrams left to right, input-processing-output. The processing steps are numbered sequentially; these numbers also refer to the implementation notes at the bottom of most of the diagrams. The notes contain additional information about the processing steps.

The arrows are used as follows:

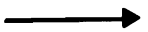
Shows control flow to or from another program.

Primary processing flow. Shows the path followed to accomplish the principal function of the body of code.



Data transfer. Indicates that data is moved from one location to another.

Control information transfer. Indicates the setting or changing of switches or pointers that will be used to determine the course of future processing.



Pointer. Indicates that a field in one data area contains an address that points to another field or data area.

Data reference. Indicates that the contents of a data area are tested or read in order to determine the course of subsequent processing.

Diagram CCH–1. CCH Overview and Table of Contents

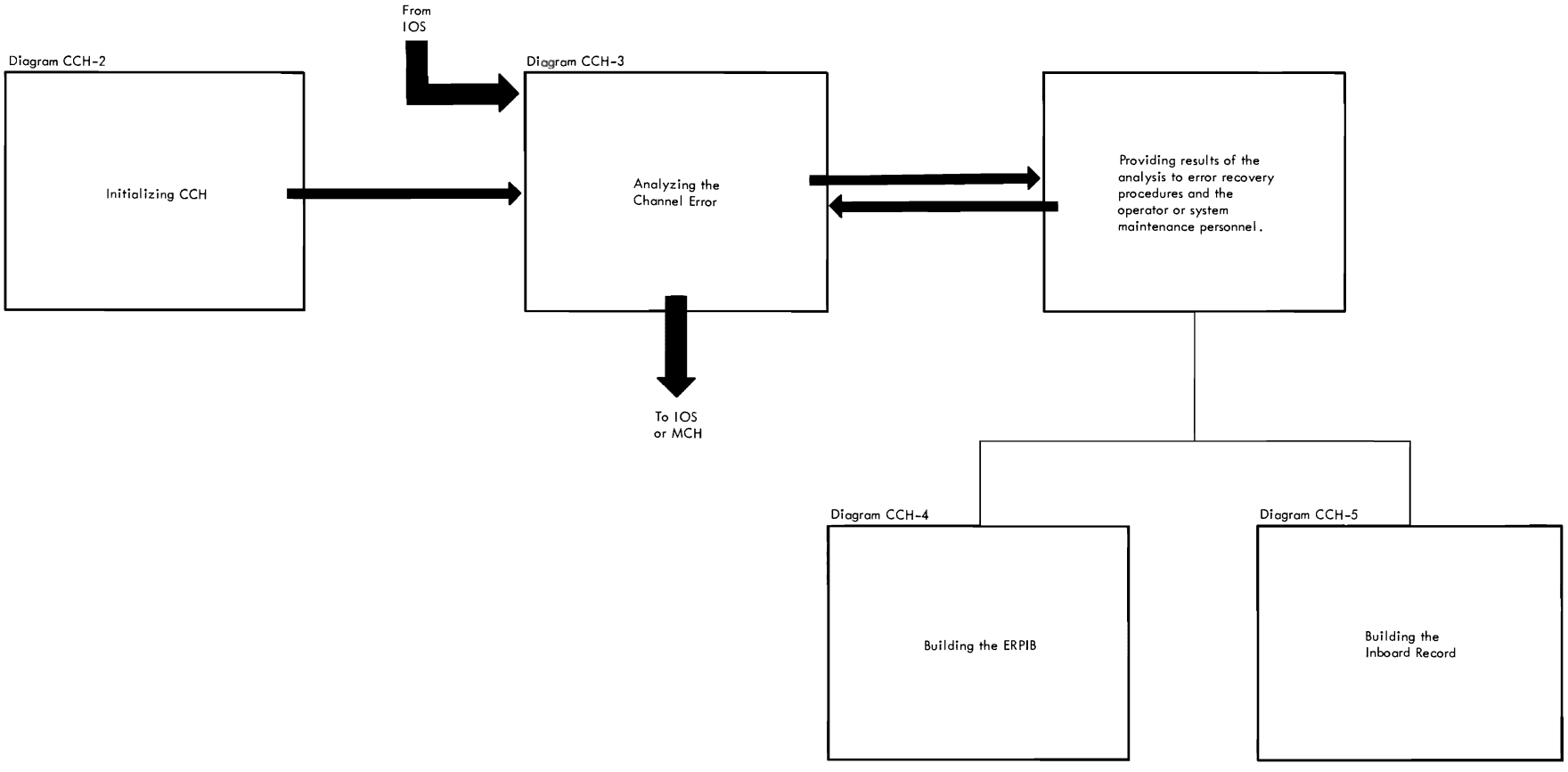
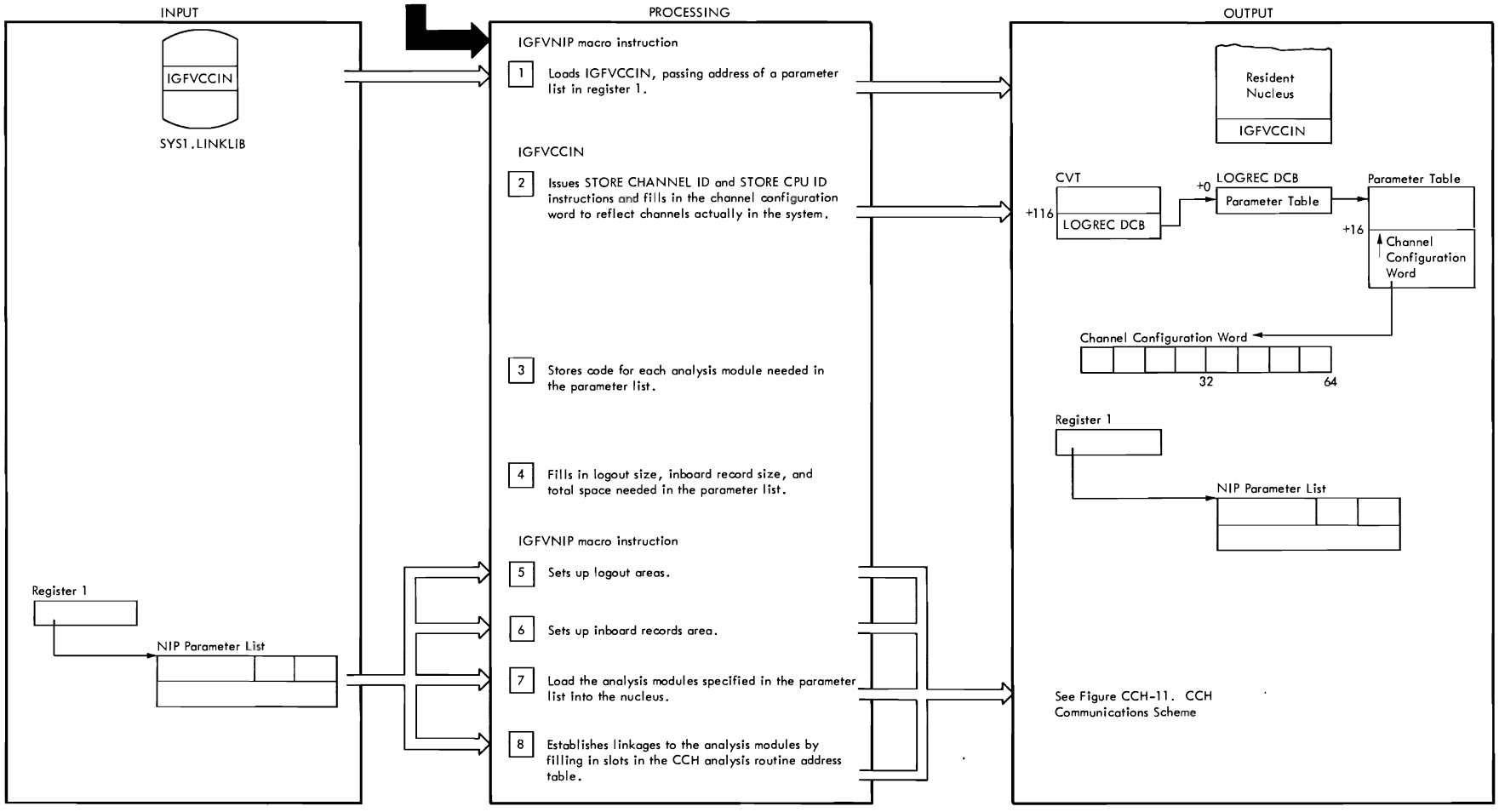


Diagram CCH–2. Initializing CCH

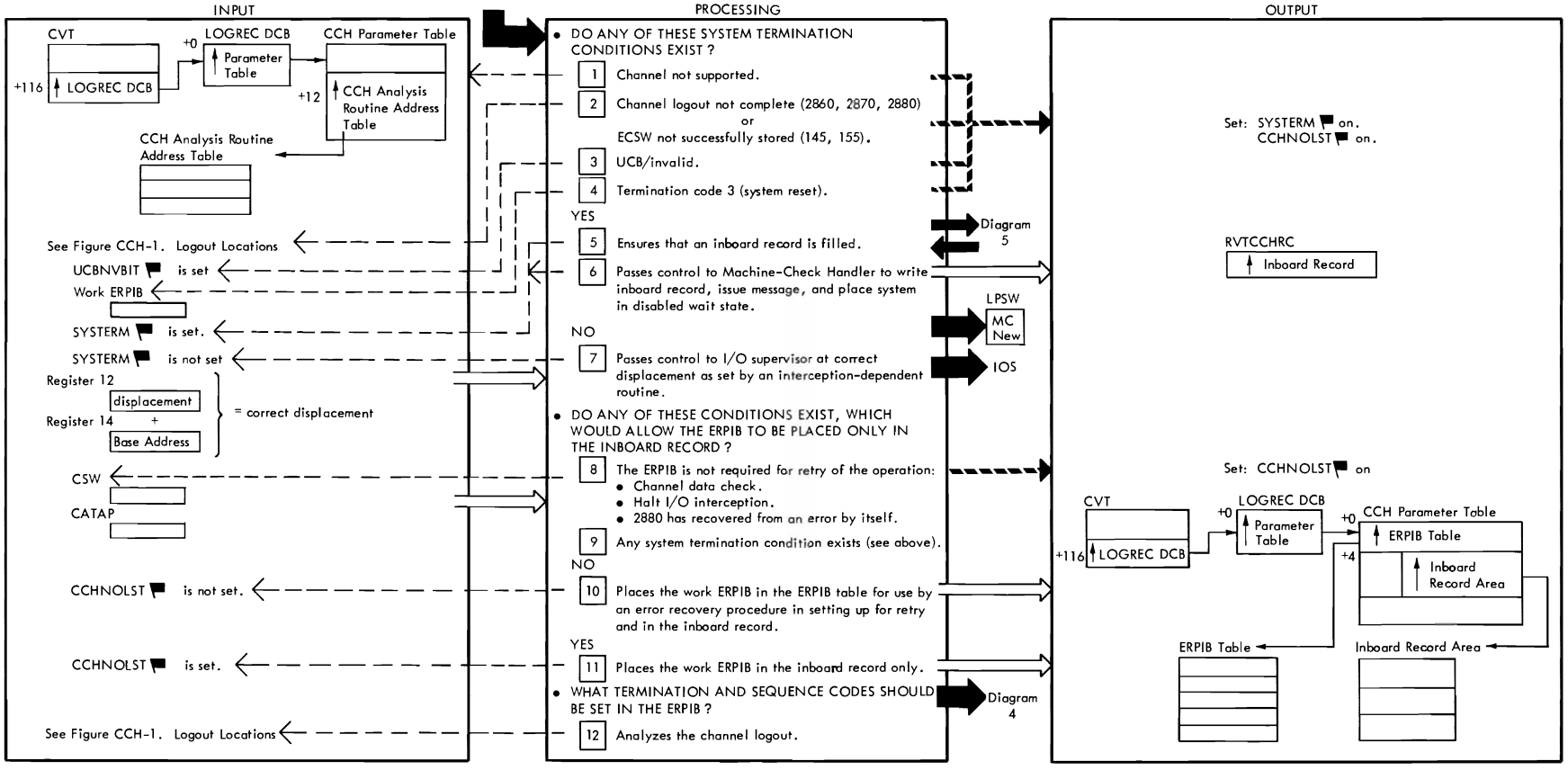


Implementation Notes for Diagram CCH-2

	Module	Label
1 IGFVNIP is a recovery management support macro instruction used by NIP to initialize CCH and MCH.		
2	IGFVCCIN	STOREID
3 IGFVCCIN first indicates the channel analysis routines needed in MASTRBYT, then transfers the information to the NIP parameter list.	IGFVCCIN	CHANSUPT

	Module	Label
4 5 6 7 8 } See OS/VS IPL and NIP Logic PLM for further discussion of IGFVNIP.	IGFVCCIN	GETRESIZ

Diagram CCH–3. Analyzing the Channel Error



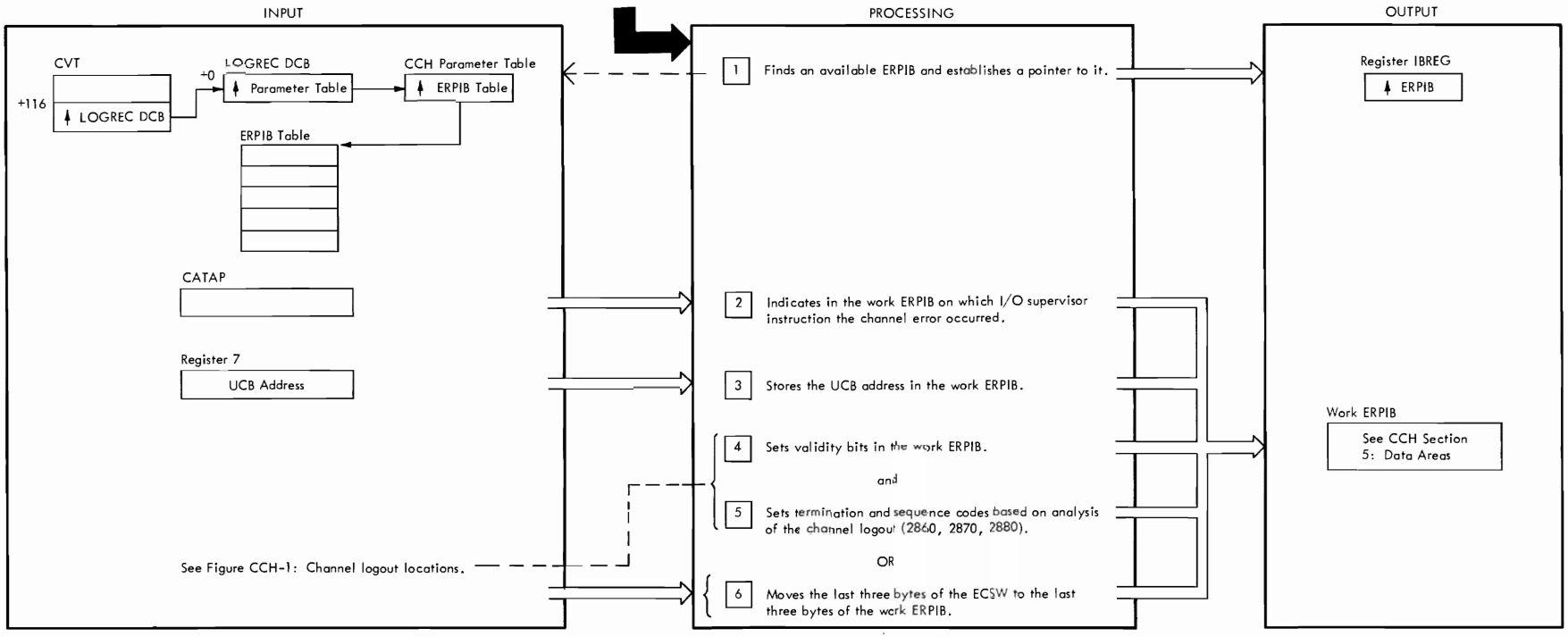
Implementation Notes for Diagram CCH-3

		Module	Label
1	If there is no entry in the channel analysis routine address table for the analysis routine to support the channel, then the channel is not supported.	IGFVCCHC (ERPIBSET)	CON80
2	See discussion on determining whether the channel has logged out or stored the ECSW successfully.	Correct analysis routine	CCH029A CCHLOGOK
3	There will be zeros in register 7 if IOS found no valid UCB.	IGFVCCHC (ERPIBSET)	NORMCONT
4		IGFVCCHC (ERPIBSET)	NORMCONT
5	If the inboard record has not been built, the "CCHNORE" flag will be set, so a recursion is made through IGFVCCHC with the "SYSTERM" flag set.	IGFVCCHC (UCB Search)	

		Module	Label
6		IGFVCCHC (UCB Search)	CCHQUIT
7	See discussion on returning control to the I/O Supervisor.	IGFVCCHC (UCB Search)	CCHGETRG
10		IGFVCCHC (ERPIBSET)	CCH101 CCH102
11		IGFVCCHC (ERPIBSET)	CCH101 CCH102
12	Also part of the function of building the ERPIB.		

Note: All switches or flags (⚑) are located in the CCH internal switches, which follow the work ERPIB in the coding of IGFVCCHC. See Figure CCH-14.

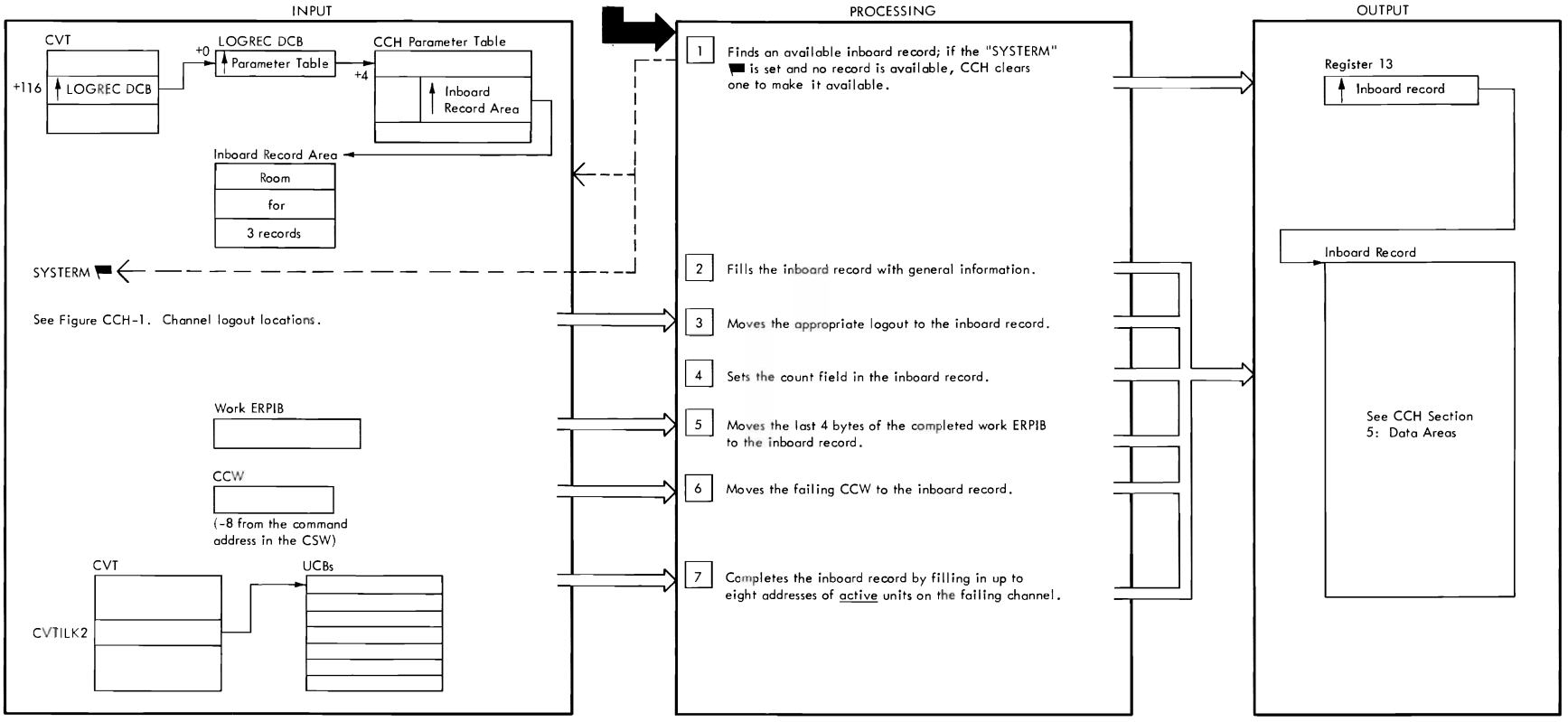
Diagram CCH–4. Building the ERPIB



Implementation Notes for Diagram CCH–4

		Module	Label
1	Five in table. Each is cleared by an ERP. See discussion of why an ERPIB might not be available. CCH always scans the ERPIB table from the bottom to the top, so the fifth ERPIB will be filled for the first error and the fourth ERPIB will be filled for the second error and so on.	IGFVCCHC (ERPIB Entry Find)	CCH000
2	See Figure CCH-6.	IGFVCCHC (Correct interception-dependent routine)	CCHSNS CCHTIOSN CCHSIO CCHSSK CCHTIOSK CCHINT CCHHIO (One of these)
3	This is the UCB address from the I/O Supervisor	IGFVCCHC (ERPIBSET)	CON80
4		IGFVCCHC (ERPIBSET)	CCHCNTU
5		Correct analysis routine	CCH02860 CCH02870 CCHLOGOK
6	CCH always sets byte 0 of the ECSW to X'FF' after the ECSW has been copied to the work ERPIB.	Correct analysis routine	CCH145 CCH00A48

Diagram CCH–5. Building the Inboard Record



Implementation Notes for Diagram CCH-5

		Module	Label
1	Recursion.	IGFVCCHC (Data Collect)	CCH007
2	CCH gets the jobname from the TIOT if it exists and is fixed in resident storage; otherwise, CCH gets the jobname from the CSCB if it exists and is fixed. If CCH cannot get the jobname from the TIOT or the CSCB, it leaves the field filled with zeros.	IGFVCCHC (Data Collect)	CCHSETFL
3	CCH always sets the logout to ones after it has copied the logout into the inboard record.	Correct analysis routine	CCHRETRN TESTRESW MOVELOG MOVE
4		Correct analysis routine or IGFVCCHC on channel data checks for all but the 2880 channel.	
5		IGFVCCHC (ERPIBSET)	CCH102
6	If the CCW is not in real storage, a paging exception may result. CCH checks if the CCW is in the non-pageable area; if not, CCH calls IEAPTRV in the IEANUC01 load module to XLATE the real CCW address to a virtual CCW address and tests the return code to see if the address is valid.	IGFVCCHC (ERPIBSET)	CCHMOVE
7		IGFVCCHC (UCB Search)	UCBSCH

ORGANIZATION OF CCH

CCH consists of one initialization module (IGFVCCIN), one central processing module (IGFVCCHC), and six channel error analysis modules (IGFVCC60, IGFVCC70, IGFVCC80, IGFVCC35, IGFVCC45, IGFVCC55).

The initialization module is brought into the system by nucleus initialization and is deleted after initialization; its function is to aid NIP in initializing CCH by indicating what CCH channel error analysis routines are to be brought into the nucleus and what logout areas are to be set up. The central module includes (1) routines that locate and fill part of the ERPIB and the inboard record as they analyze the channel error, (2) interfaces with the I/O supervisor and the Machine-Check Handler, and (3) data areas. The analysis modules analyze the channel logout or move the ECSW to complete the ERPIB and also move the logout to the inboard record.

The execution of IGFVCCHC, following some initial functions, begins at one of four interception-dependent routines. The interception-dependent routine passes control to and receives control back from the ERPIBSET routine. The ERPIBSET routine in turn passes control to and receives control back from an analysis module. Finally, the interception-dependent routine, having reviewed the information collected by the ERPIBSET routine and the analysis module, passes control to the UCB Search routine. The UCB Search routine checks flags set by the other CCH routines to decide whether to return control to the I/O supervisor (and at what displacement, using a value set by the interception-dependent routine) or to pass control to the Machine-Check Handler.

Figure CCH-8 illustrates the processing done by each of the CCH routines to accomplish the functions shown in the method of operation diagrams.

INCLUDING CCH IN THE SYSTEM-DYNAMIC LOADING

CCH uses a technique called dynamic loading to save space in the nucleus. Dynamic loading involves bringing into the nucleus only what is needed for any single system configuration.

During system generation, IGFVCCHC is placed in the resident nucleus, and IGFVCCIN and all six analysis modules are placed on SYS1.LINKLIB. During nucleus initialization, NIP, using the IGFVNIP macro instruction, brings IGFVCCIN into the resident nucleus. IGFVCCIN determines which analysis modules are needed for the system configuration by issuing the STORE CHANNEL ID and STORE CPU ID instructions. The channel configuration word is filled in to show which channels are presently in the system; it is illustrated in Figure CCH-9. IGFVNIP then brings the indicated analysis modules into the resident nucleus. Finally, IGFVNIP deletes IGFVCCIN from the resident nucleus.

This loading technique is illustrated in Figure CCH-10. For a more detailed discussion of CCH initialization, see OS/VS IPL and NIP Logic.

CCH COMMUNICATIONS SCHEME

In order to establish pointers to the analysis routines, the ERPIB table, and the inboard record area, a CCH communications area is produced before and during CCH initialization. During system generation, the address of the first word of the SYS1.LOGREC data control block is placed in the communications vector table at displacement 116. The first word of the SYS1.LOGREC DCB contains the address of a five-word parameter table, which is the CCH portion of the I/O recovery management support communications area (IORMSCOM).

The areas referred to in the parameter table all reside in the resident nucleus before CCH is initialized. During nucleus initialization, the inboard record area and the analysis modules are appended to the nucleus by the IGFVNIP macro instruction, and their addresses are saved in the CCH communications scheme.

The CCH communications scheme is illustrated in Figure CCH-11.

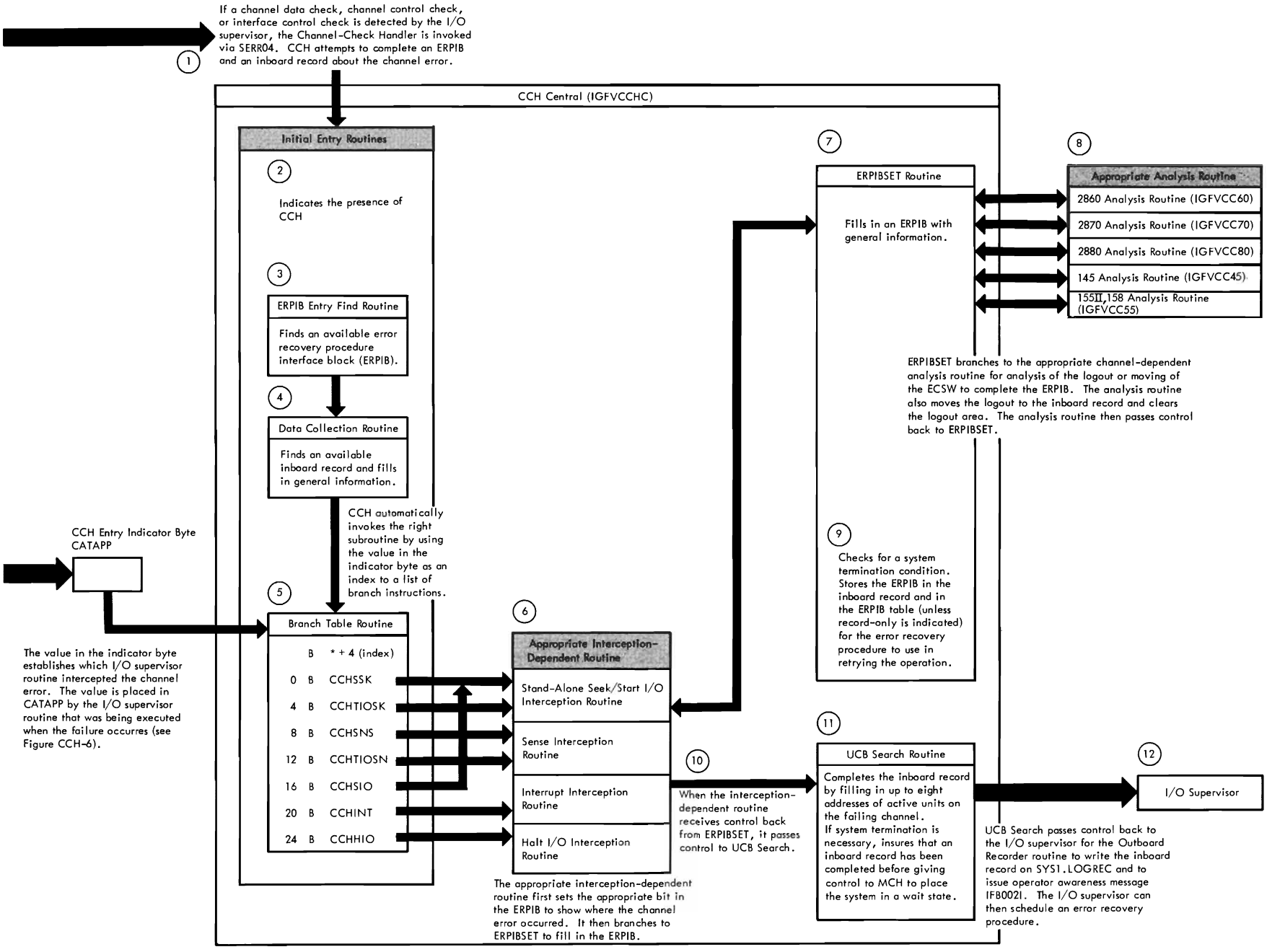
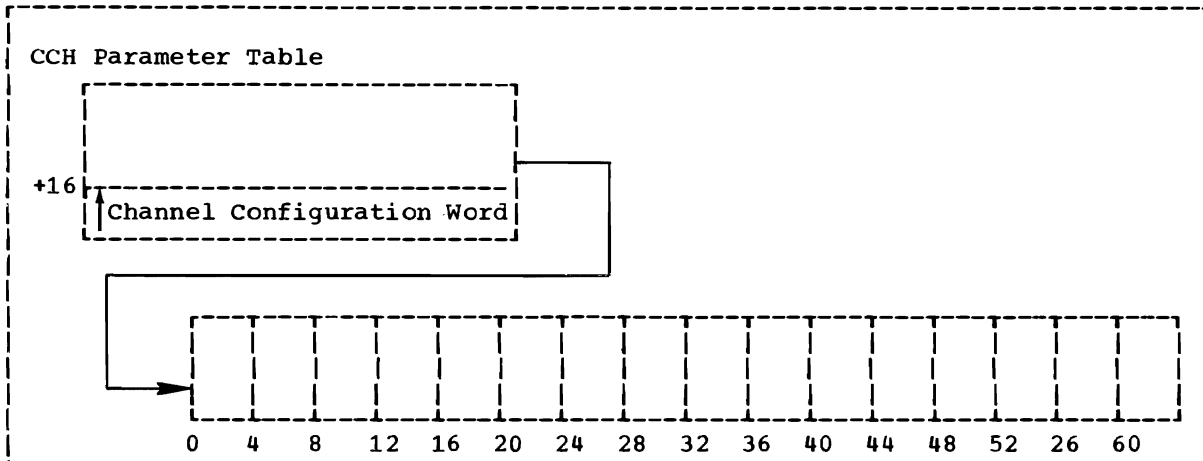


Figure CCH-8. Overview of processing by CCH routines



The channel configuration word is 64 bits long. Each half-byte represents one channel. Each channel is represented by a four-bit hexadecimal digit as follows:

Four Bit Hex Count	Channel Assignment
0	No channel attached
1	Integrated multiplexer channel
2	Integrated selector channel
3	Integrated block multiplexer channel
4	Reserved
5	2860 selector channel
6	2870 multiplexer channel
7	2880 block multiplexer channel
8	2880 selector channel
9	Reserved
A	Integrated file adapter
B-E	Reserved
F	Unrecognized channel attached

Figure CCH-9. Channel configuration word

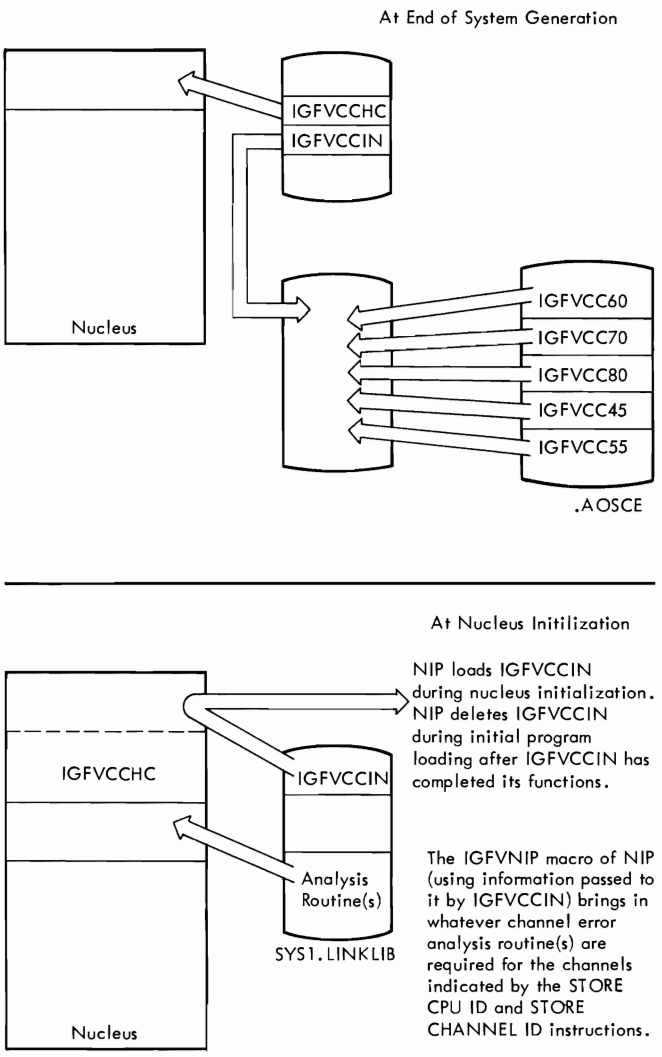


Figure CCH-10. CCH dynamic loading

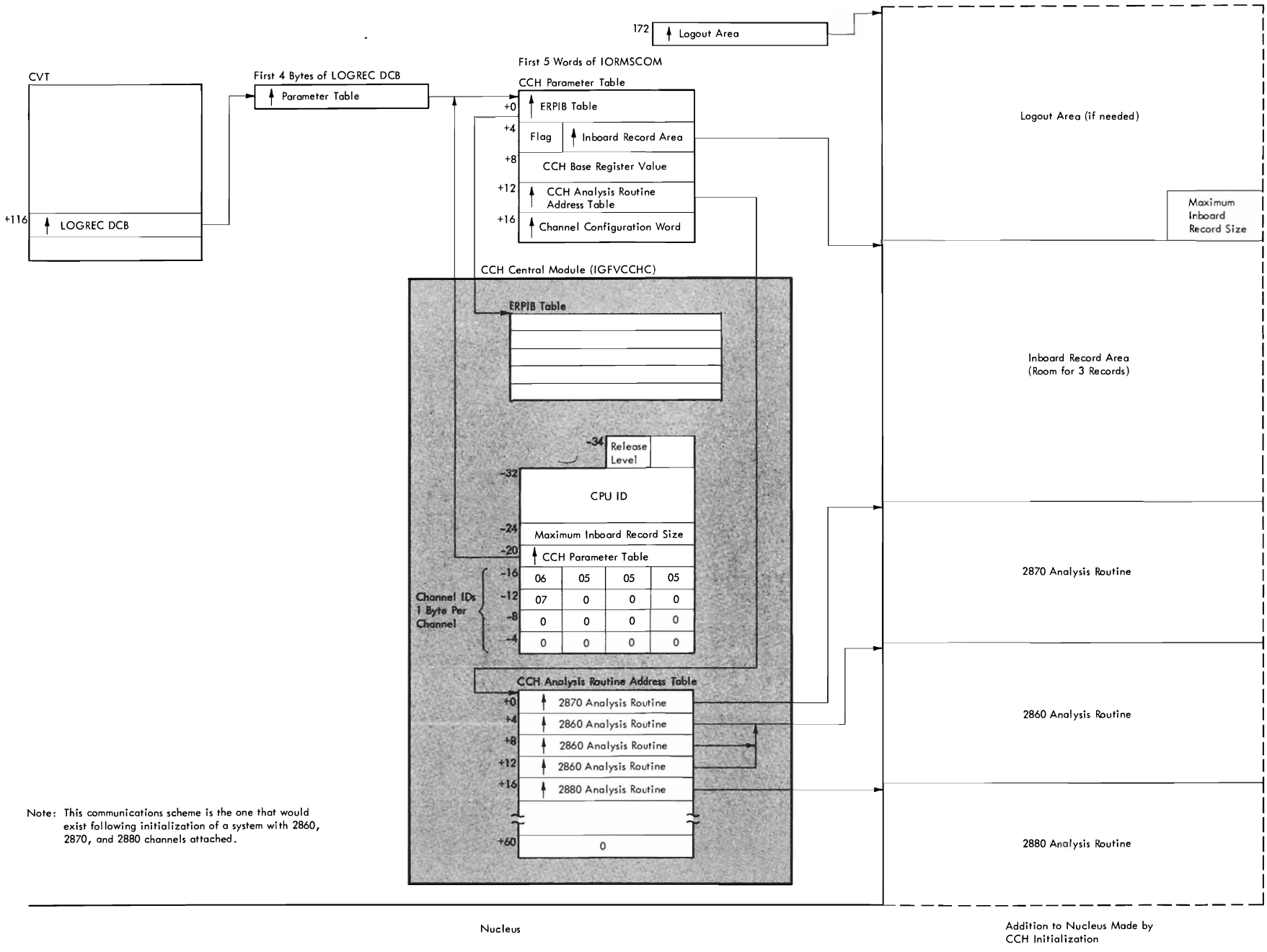


Figure CCH-11. CCH communications scheme following a possible initialization

CCH MODULE DESCRIPTIONS

The following module descriptions summarize the functions of each CCH module. Additional information about CCH modules can be found in the method of operation diagrams and in the prologues for each module. Prologues can be found on the microfiche.

IGFVCCIN - CCH Initialization

- Determines which analysis modules are needed for the channels in the system.
- Informs IGFVNIP which analysis modules are needed (for IGFVNIP to bring into the nucleus).
- Informs IGFVNIP how much real storage is needed for the inboard records and logouts, and indicates the sizes of the logouts and the size of one inboard record.

Entry Point	Entered From	Exits To
IGFVCCIN	IGFVNIP	IGFVNIP with the parameter list filled.

IGFVCCHC - CCH Central Processing

- Sets a bit to signify that CCH had control in case a system termination condition exists.
- Checks for a valid UCB address.

ERPIB Entry Find routine

- Determines the address of an available ERPIB.

Data Collection routine

- Determines the address of an available inboard record.
- Places general information in the inboard record.

Branch Table routine

- Branches to the correct interception-dependent routine using the value set by the I/O supervisor in CATAP.

Interception-Dependent routines

Start I/O and Test I/O

- Sets the SIO or TIO bit in the work ERPIB.
- Passes control to the ERPIBSET routine to complete the ERPIB.
- Sets the SIO or TIO displacement for return to the I/O supervisor.
- Passes control to the UCB Search routine.

IGFVCCHC - CCH Central Processing (Cont'd)

Interception-Dependent routines (Cont'd)

Sense

- Sets the SIO or TIO bit in the work ERPIB.
- Indicates that the ERPIB is for the inboard record only.
- Passes control to the ERPIBSET routine to complete the ERPIB.
- Sets the "no retry" displacement for return to the I/O supervisor and the "no retry" bit in the ERPIB.
- Passes control to the UCB Search routine.

Interruption

- Sets the interruption bit in the work ERPIB.
- Passes control to the ERPIBSET routine to complete the ERPIB.
- Sets the interruption or alternate return displacement for return to the I/O supervisor.
- Passes control to the UCB Search routine.

Halt I/O

- Sets the HIO bit in the work ERPIB.
- Indicates that the ERPIB is for the inboard record only.
- Passes control to the ERPIBSET routine to complete the ERPIB.
- Sets the HIO displacement for return to the I/O supervisor.
- Passes control to the UCB Search routine.

ERPIBSET routine

- Stores the UCB address in the work ERPIB.
- Checks to ensure that the channel is supported.
- Places general information in the ERPIB.
- Passes control to the correct analysis module to complete the ERPIB (bypasses the analysis module for channel data checks, except on the 2880 channel).
- Examines the bits set by the analysis module and sets appropriate flags.
- Moves the ERPIB to the ERPIB table and to the inboard record or to the inboard record only.
- Returns control to the correct interception-dependent routine.

IGFVCCHC - CCH Central Processing (Cont'd)

UCB Search routine

- Completes the inboard record with up to eight addresses of active units on the failing channel.
- If system termination is indicated, returns to IGFVCCHC, if necessary to complete an inboard record. Passes control to MCH.
- If system termination is not indicated, passes control to the I/O supervisor.

Entry Point	Entered From	Exits To
IGFVCCHC	I/O supervisor	Machine-Check Handler via a branch to the address contained in RVTSHUT if a system termination condition exists. I/O supervisor via register 14 if a system termination condition does not exist.

IGFVCC60 - CCH 2860 Channel Error Analysis

- Indicates system termination if the logout is not complete.
- Examines combinations of bits set in the channel logout and then sets bits in the work ERPIB to indicate:

  Validity of the unit address, device status, channel program address, and count.

  Source of error.

  Termination code.

  Sequence code.
- Moves the channel logout to the inboard record.
- Sets the logout area to 1's.
- Puts the record length in the inboard record count field.

Entry Point	Entered From	Exits To
IGFVCC60	IGFVCCHC (ERPIBSET)	IGFVCCHC (ERPIBSET)

IGFVCC70 - CCH 2870 Channel Error Analysis

- Indicates system termination if the logout is not complete.
- Examines combinations of bits set in the channel logout and then sets bits in the work ERPIB to indicate:

  Validity of the unit address, device status, channel program address, and count.

  Source of error.

  Termination code.

  Sequence code.
- Moves the channel logout to the inboard record.
- Sets the logout area to 1's.
- Puts the record length in the inboard record count field.

Entry Point	Entered From	Exits To
IGFVCC70	IGFVCCHC (ERPIBSET)	IGFVCCHC (ERPIBSET)

IGFVCC80 - CCH 2880 Channel Error Analysis

- Indicates system termination if the logout is not complete.
- Examines combinations of bits set in the channel logout and then sets bits in the work ERPIB to indicate:

  Validity of the sequence code, unit address, device status, and channel program address.

  Source of error.

  Termination code.

  Sequence code.
- Moves the channel logout to the inboard record.
- Sets the logout area to 1's.
- Puts the record length in the inboard record count field.

Entry Point	Entered From	Exits To
IGFVCC80	IGFVCCHC (ERPIBSET)	IGFVCCHC (ERPIBSET)

IGFVCC35 - CCH 135 Channel Error Analysis

- Indicates system termination if the ECSW is not complete.
- Moves the last three bytes of the ECSW to the last three bytes of the work ERPIB.
- Moves the hardware-stored unit address to the inboard record.
- Moves the channel logout to the inboard record.
- Sets the logout area and the first byte of the ECSW to 1's.
- Puts the record length in the inboard record count field.

Entry Point	Entered From	Exits To
IGFVCC35	IGFVCCHC (ERPIBSET)	IGFVCCHC (ERPIBSET)

IGFVCC45 - CCH 145 Channel Error Analysis

- Indicates system termination if the ECSW is not complete.
- Moves the last three bytes of the ECSW to the last three bytes of the work ERPIB.
- Moves the hardware-stored unit address to the inboard record.
- Moves the channel logout to the inboard record.
- Sets the logout area and the first byte of the ECSW to 1's.
- Puts the record length in the inboard record count field.

Entry Point	Entered From	Exits To
IGFVCC45	IGFVCCHC (ERPIBSET)	IGFVCCHC (ERPIBSET)

IGFVCC55 - CCH 155II/158 Channel Error Analysis

- Indicates system termination if the ECSW is not complete.
- Moves the last three bytes of the ECSW to the last three bytes of the work ERPIB.
- Moves the hardware-stored unit address to the inboard record.
- Sets the first byte of the ECSW to 1's.
- Puts the record length in the inboard record count field.

Note: Also provides minimum support for unrecognized channels.

Entry Point	Entered From	Exits To
IGFVCC55	IGFVCCHC (ERPIBSET)	IGFVCCHC (ERPIBSET)

CCH SECTION 4: DIRECTORY

The directory provides a quick reference from this publication to the pertinent coding, or from the coding back into the publication. The directory contains the following information:

Module and CSECT Name: The name of the object module and CSECT containing the coding indicated by the label.

Label: Names taken from the listing for entry points, significant sections of coding, and data areas.

Library: The place where the module resides before initialization.

Name: The descriptive name used to refer to the module.

Diagram: The method of operation diagram containing information about the coding identified by the label.

Module and CSECT Name	Label	Library	Name	Diagram
IGFVCCIN	CHANSUPT	SYS1.LINKLIB	CCH Initialization	2
	GETRESIZ			2
	STOREID			2
IGFVCCHC	CCHCNTU	Nucleus	CCH Central	4
	CCHGETRG			3
	CCHHIO			4
	CCHINT			4
	CCHMOVE			5
	CCHQUIT			3
	CCHSETFL			5
	CCHSIO			4
	CCHSNS			4
	CCHSSK			4
	CCHTIOSK			4
	CCHTIOSN			4
	CCH000			4
	CCH007			5
	CCH101			3
	CCH102			3,5
	CON80			3,4
	UCBSCH			5
IGFVCC60	CCHRETRN	SYS1.LINKLIB	CCH 2860 Channel Error Analysis	5
	CCH02860			4
	CCH029A			3
IGFVCC70	CCHRETRN	SYS1.LINKLIB	CCH 2870 Channel Error Analysis	5
	CCH02870			4
	CCH029A			3
IGFVCC80	CCHLOGOK	SYS1.LINKLIB	CCH 2880 Channel Error Analysis	3,4
	TESTRESW			5
IGFVCC35	CCH35	SYS1.LINKLIB	CCH 135 Channel Error Analysis	4
	MOVELOG			5
IGFVCC45	CCH45	SYS1.LINKLIB	CCH 145 Channel Error Analysis	4
	MOVE			5
IGFVCC55	CCH00A48	SYS1.LINKLIB	CCH 155II Channel Error Analysis	4

This section contains descriptions of the principal data areas used by CCH:

- Channel-Check Handler parameter table
- Error recovery procedure interface block (ERPIB)
- Channel error inboard record

The symbolic names shown in individual fields represent the displacement, in bytes, from the beginning of a specific data area to the field. Access is gained to a specific field by using an instruction in which the beginning address of the area (usually contained in a register) is the base address, and the symbolic field name represents the displacement. Use of the fields can be traced in the CCH listings by first locating the symbolic field names in the cross-reference table at the back of the listings and then noting where the names are used.

The field headings and their meanings are:

Displacement: The numeric address of the field relative to the beginning of the data area. The first number is in decimal; the second (in parentheses) is the hexadecimal equivalent.

Bytes and Bit Patterns: The size (number of bytes) of the field and the bit settings of flag fields; that is, the state of bits in a byte. When the column is used to show the state of bits, it is shown as follows:

....	The eight bit positions (0-7) in a byte. For ease of scanning, the high-order (left-hand) four bits are separated from the low-order four bits.
1...	A reference to bit 0.
.... ..xx	Bits 6 and 7 are reserved.

Bit settings that are significant are shown and described. Bit settings that are not presently significant are described as reserved bits.

Field Name: A name that identifies the field.

Field Description: The use of the field.

Figure CCH-12 is provided to assist in locating CCH fields. This figure contains an alphabetic list of the fields with their displacements.

Field Name	Displacement	CCH Parameter Table	ERPIB	Inboard Record
ACTIO	40 (28)			x
CCHANTAB	12 (C)	x		
CCHBASE	8 (8)	x		
CCHCFGWD	16 (10)	x		
CCHPTTAB	0 (0)	x		
CHANID	80 (50)			x
CHNLOG	92 (5C)			x
COUNT	0 (0)			x
CPUIDFD	24 (18)			x
CSWLWB	64 (40)			x
CUADDR	81 (51)			x
DEVTP	76 (4C)			x
ECSW	72 (48)			x
ERPIBAD	0 (0)	x		
FAILCCW	56 (30)			x
FLAG	4 (4)	x		
IGBLAME	5 (5)		x	
IGPRGLFG	4 (4)		x	
IGTERMSQ	7 (7)		x	
IGVALIDB	6 (6)		x	
IGUCB	1 (1)		x	
JOBID	32 (20)			x
KEY	8 (8)			x
RECORD1	5 (5)	x		
RELENGTH	6 (6)			x
SWITCHES	10 (0A)			x
TIME	16 (10)			x
TIOADDR	88 (58)			x

Figure CCH-12. Locations of CCH fields

CHANNEL-CHECK HANDLER PARAMETER TABLE

This data area is located in the first five words of the IORMSCOM (input/output recovery management support communications area). Its use in the Channel-Check Handler communications scheme is illustrated in Figure CCH-11.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
0 (0)	20	CCHPTTAB	
0 (0)	4	ERPIBAD	Address of error recovery procedure interface block table.
4 (4)	1	FLAG	Flag indicating if there are any inboard records to be written.
5 (5)	3	RECORD1	Address of inboard record area.
8 (8)	4	CCHBASE	CCH base register value.
12 (C)	4	CCHANTAB	Address of CCH channel error analysis routine pointer table.
16 (10)	4	CCHCFGWD	Address of channel configuration word.

ERROR RECOVERY PROCEDURE INTERFACE BLOCK

The ERPIB (error recovery procedure interface block) is the eight-byte field in which CCH places the results of its analysis of the channel error for use by the appropriate ERP.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
0 (0)	1		Reserved.
1 (1)	3	IGUCB	Address of the unit control block for the device in use when the channel failure occurred. The ERP uses this address to locate the correct ERPIB for the current error.
4 (4)	1	IGPRGLFG	Program flags. Indicates the selection or interruption sequence when the CSW was stored:
	1...		CSW was stored after a Start I/O instruction was executed.
	.1..		CSW was stored after an I/O interruption.
	..1.		CSW was stored after a Test I/O instruction was executed.
	...1		CSW was stored after a Halt I/O instruction was executed.
	.... .1..		Sense data was stored.
	.... ..1.		Count in CSW is valid.
	1		No retry is to be attempted under any conditions.
	.... x...		Reserved.
5 (5)	1	IGBLAME	Probable source of error. The determination is made by the channel on the basis of the type of error check, the location of the checking station, the information flow path, and the success or failure of transmission through previous check stations. Normally, only one bit is set in this field. However, when communication cannot be resolved to a single unit, such as when the interface between units is at fault, multiple bits (normally two) may be set in this field. When a reasonable determination cannot be made, all bits in this field are set to 0. If the detect and source fields indicate different units, the interface between them can also be considered suspect.
	1...		CPU error.
	.1..		Channel error.
	..1.		Storage control unit error.
	...1		Storage error.
	.... 1...		Control unit error.
	xxx		Reserved.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
6 (6)	1	IGVALIDB	Validity indicators. When the designated field is stored by the channel with the correct contents, the validity bit is 1. When the designated field is stored by the channel with unpredictable contents, the validity bit is 0. The validity bits for nonstored fields are meaningless.
	1...		Interface address is valid.
	...1		Sequence code is valid.
	.... 1...		Unit status is valid.
	.... .1..		Command address is valid. The CSW contains a valid command address.
	.... ..1.		Channel address is valid.
	1		Device address is valid.
	.xx.		Reserved.
7 (7)	1	IGTERMSQ	Termination and sequence (retry) codes:
	11..		Termination code. Specifies the termination signals used on the I/O interface after the channel detected the error. This field has meaning only when a channel control check or an interface control check is indicated in the CSW. When neither of these is indicated, no termination has been forced by the channel. 00 Interface disconnect. 01 Stop, stack, or normal termination. 10 Selective reset. 11 System reset.
	..xx		Reserved.
	.... 1...		I/O error alert. The indicated unit has signaled the I/O error on the I/O interface. The channel performs a selective reset and causes an interface control check to be set in the CSW.
	.... .111		Sequence code. The sequence codes have different meanings for the stand-alone and integrated channels, as follows: For 2860, 2870, and 2880 channels: Indicates when the channel detected the error. Meaningful only for channel control checks or interface control checks. Not to be used by error routines involved with unit checks. 000 The error occurred during execution of a Test I/O instruction. A pending interruption in the channel may have been cleared.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
7 (7)	(Continued)	IGTERMSQ	001 The channel has sent a Command Out signal but has not yet examined the unit status. May indicate unpredictable movement on devices with control units that start movement at the Command Out signal. 010 The device accepted the command but no data was transferred. Magnetic tape devices do not begin tape motion for a write operation until the first byte of data has been transferred. 011 At least one byte of data was transferred over the interface. The command address in the CSW is eight bytes higher than the address of the command causing the data transfer. In general this means device movement may have taken place if an Interface Disconnect signal was issued. 100 The command was not accepted by the control unit, or a Command Out signal has not been sent to the control unit for the command at the address eight bytes higher than the address in the CSW. No device movement has taken place. 101 The command was accepted but data transfer may not have taken place. If the termination code is "interface disconnect," activity can be retried on most devices. It cannot be retried if the command was a Write command for a magnetic tape device. 110 Reserved. 111 No other codes apply. If any other sequence codes are retryable, this code is retryable. For 135, 145, and 155II channels: Identifies the I/O sequence in progress at the time of the error. It is meaningless if stored during the execution of Halt I/O or Halt Device.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
7 (7)	(Continued)	IGTERMSQ	For all cases, the channel program address, if validly stored and nonzero, is the address of the current CCW+8. When a Test I/O is issued to a channel with a pending channel logout condition, the Test I/O causes the channel to store a CSW, perform a logout, and reset the condition without regard for the device address associated with the Test I/O. The I/O address that pertains to the logout condition is stored in locations 185-187. A logout resulting from an error that occurred during the execution of a Test I/O instruction can be differentiated from a logout cleared by a Test I/O by examining the sequence code, because sequence code 0 can only occur in the first situation. 000 A channel-detected error occurred during the execution of a Test I/O instruction. 001 Command Out with a nonzero command byte on Bus Out has been sent by the channel, but device status has not yet been analyzed by the channel. This code is set with a Command Out response to Address In during initial selection. 010 The command has been accepted by the device, but no data has been transferred. This code is set by a Service Out or Command Out response to Status In during an initial selection sequence, if the status is either channel end alone, or channel end and device end, or channel end, device end, and status modifier, or all 0's. 011 At least one byte of data has been transferred over the interface. This code is set with a Service Out response to Service In and, when appropriate, may be used when the channel is in an idle or polling state.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
7 (7)	(Continued)	IGTERMSQ	100 The command in the current CCW has either not yet been sent to the device or else was sent but not accepted by the device. This code is set when one of the following occurs: • When the command address is updated during command chaining or a Start I/O. • When Service Out or Command Out is raised in response to Status In during an initial selection sequence with the status on Bus In including attention, control unit end, unit check, unit exception, busy, status modifier, (without channel end and device end) or device end (without channel end). • When a short, control-unit-busy sequence is signaled. • When command retry is signaled. • When the channel issues a Test I/O command rather than the command in the current CCW. 101 The command has been accepted, but data transfer is unpredictable. This code applies from the time a device comes on the interface until the time it is determined that a new sequence code applies. It may thus be used when a channel goes into the polling or idle state and it is impossible to determine that code 2 or 3 applies. It may also be used at other times when a channel cannot distinguish between codes 2 and 3. 110 Reserved. 111 Reserved.

CHANNEL ERROR INBOARD RECORD

The inboard record is the place where CCH stores information about the channel error to be written into the SYS1.LOGREC data set for use by the operator and system maintenance personnel.

Displacement	Bytes and Bit Patterns	Field Name	Field Description
0 (0)	6	COUNT	Count information for the Outboard Recorder.
6 (6)	2	RELENGTH	Record length. The length of the record minus 10 bytes.
8 (8)	2	KEY	Record ID. The first byte contains X'20' to identify the record as CCH inboard record. The second byte contains: Bits 0,1,2 -- 000=OS system 010=OS/VS system 3-7 -- release level (binary)
10 (0A)	2	SWITCHES	
10 (0A)	1	SWBYTE0	
	1...		Operator action message.
	.1..		System/370 machine - date and time from System/370 clock.
	..1.		EC mode.
	...x xxxx		Reserved.
11 (0B)		SWBYTE1	
	1...		Message required.
	.1..		Record incomplete.
	..1.		CCH terminated system.
	...1		Channel not supported.
	 xxxx		Reserved.
12 (0C)	4		Reserved.
16 (10)	8	TIME	Time and date record was made, from STORE CLOCK instruction.
24 (18)	8	CPUIDFD	Model and serial numbers of the CPU on which the error occurred.
32 (20)	8	JOBID	Name of the job being executed when the failure occurred.
40 (28)	16	ACTIO	Addresses of I/O devices active at the time of the failure on the failing channel (maximum of eight).
56 (30)	8	FAILCCW	Failing CCW. The last real CCW executed before the failure.
64 (40)	8	CSWLWB	Contents of the CSW stored at the time of the failure.
72 (48)	4	ECSW	The last four bytes of the ERPIB. For the 135, 145, and 155II channels, the last three bytes of the ERPIB are equivalent to the last three bytes of the extended channel status word.
76 (4C)	4	DEVTP	The device type, taken from the UCB for the failing device.
80 (50)	1	CHANID	Channel type ID taken from the channel configuration word (produced during system generation).

Displacement	Bytes and Bit Patterns	Field Name	Field Description
81 (51)	3	CUADDR	Address of the channel and unit being used when the failure occurred. For a 2314, the address is the one initially associated with the cell experiencing the error.
84 (54)	4		Reserved.
88 (58)	4	TIOADDR	Unit address as stored by computer in location 184. See "Note" for field CHNLOG.
92 (5C)	Variable length	CHNLOG	Logout information sent by the channel. The length of this field varies with the type of channel:

Channel	Length
2860	24 bytes
2870	24 bytes
2880	112 bytes
135	24 bytes
145	96 bytes
155II/158	0 bytes

Note: For the 2860, 2870, and 2880 channels, there is no TIOADDR field in the inboard record; the CHNLOG field therefore begins at displacement 88 for those three channels.

This section contains information to aid in locating errors in CCH. Figure CCH-13 shows the contents of the general purpose registers expected upon entry to each CCH module. Figure CCH-14 shows the internal switches (or flags) used by CCH and discussed in other parts of this book.

		Register Input Expected by Each CCH Module						
Register Name	Register Number	IGFVCCIN	IGFVCCHC	IGFVCC60	IGFVCC70	IGFVCC80	IGFVCC45	IGFVCC55
R0	0							
INDEX RVTREG	1	Address of Parameter List						
IOBREG	2		IOB Address					
RTNADRG WORK3	3			Base Address	Base Address	Base Address	Base Address	Base Address
LOGBASE WRK4	4							
IBREG	5							
CUAREG UAREG	6		Unit Address	Unit Address	Unit Address	Unit Address	Unit Address	Unit Address
UCBREG	7		UCB Address (or 0)					
WKERPIB	8			Base Address for WORKERPIB and SWITCHES	Base Address for WORKERPIB and SWITCHES	Base Address for WORKERPIB and SWITCHES	Base Address for WORKERPIB and SWITCHES	Base Address for WORKERPIB and SWITCHES
CVTREG WKREG0 WORK5 WORKREG9	9			CVT Address	CVT Address	CVT Address		
LINKREG2 WKREG1	10			Return to IGFVCCHC	Return to IGFVCCHC	Return to IGFVCCHC	Return to IGFVCCHC	Return to IGFVCCHC
LOGPTR RESZREG WKREG2 WORKREGB	11							
MOVEREG R12 WKREG3 WORKREGC	12							
REREG	13			Inboard Record Address	Inboard Record Address	Inboard Record Address	Inboard Record Address	Inboard Record Address
LINKREG1 R14 WORK4	14	Return to NIP	Return to I/O Supervisor					
BASE	15		Base Address					

Figure CCH-13. Register input expected by the modules of CCH

The internal switches used by CCH follow the WORKERPIB in the code of IGFVCCHC.

Name	Equated To	Meaning	Indicates	Set By	Cleared By	Interrogated By
CCHCMDRG	X'80'	Command register parity valid	Valid parity in 2860 logword for command register.	IGFVCC60	IGFVCCHC (Initial Entry)	IGFVCC60
CCHNORE	X'40'	No inboard record available	There is no inboard record available to record this error.	IGFVCCHC (Initial Entry)	IGFVCCHC (Initial Entry)	IGFVCCHC (all interception-dependent routines except CCHHIO; ERPIBSET; UCBSCH) IGFVCC60 IGFVCC70
CCHERFRE	X'20'	ERPIB freed from active list	An ERPIB has been freed for use. This occurs when it is necessary to search ERRLCH and the SIRB to free an ERPIB.	IGFVCCHC (Initial Entry)	IGFVCCHC (Initial Entry)	IGFVCCHC (Initial Entry)
CCHNOLST	X'10'	Inboard record only-no ERPIB to list	The completed ERPIB is not to be moved to the ERPIB Table to be passed to the ERP. The ERPIB has been created for use only in the inboard record.	IGFVCCHC (CCHHIO; CCHSNS; Initial Entry; ERPIBSET) IGFVCC60 IGFVCC70 IGFVCC80	IGFVCCHC (Initial Entry)	IGFVCCHC (ERPIBSET)
CCHHAND	X'08'	Device end or attention presented	Recovery is allowed on asynchronous device ends and attentions.	IGFVCCHC (Initial Entry)	IGFVCCHC (Initial Entry)	IGFVCCHC (CCHINT)
CCHIBC	X'04'	ERPIB already created	The device-in-error had a previous channel error, and an ERPIB has already been created.	IGFVCCHC (Initial Entry)	IGFVCCHC (Initial Entry)	IGFVCCHC (CCHSNS)
UCBNVBIT	X'02'	UCB invalid	No UCB address was passed to IGFVCCHC by IOS.	IGFVCCHC (Initial Entry; ERPIBSET)	IGFVCCHC (Initial Entry)	IGFVCCHC (Initial Entry; CCHINT)
SYSTERM	X'80'	Signal for system termination	Upon completion of the inboard record, give control to the Machine-Check Handler to record the inboard record and place the system in the wait state.	IGFVCCHC (ERPIBSET) IGFVCC60 IGFVCC70 IGFVCC80	IGFVCCHC (Initial Entry, UCBSCH)	IGFVCCHC (Initial Entry; ERPIBSET; UCBSCH)
INTRN2	X'40'	Alternate return		IGFVCCHC (ERPIBSET)	IGFVCCHC (Initial Entry)	IGFVCCHC (CCHINT)
NOMODEL	X'20'	No model found for logout		IGFVCC80	IGFVCCHC (Initial Entry) IGFVCC80	IGFVCCHC (ERPIBSET) IGFVCC80
NOLOG	X'10'	No channel logout	The 2860 channel error analysis routine has set the SYSTERM switch because the 2860 logout area indicated that no log is present. Therefore, this might not have been a 2860 channel logging out, but a 2880 operating as a selector channel.	IGFVCC60	IGFVCCHC (Initial Entry) IGFVCC80	IGFVCCHC (ERPIBSET) IGFVCC80

Figure CCH-14. CCH internal switches

PART 3: DYNAMIC DEVICE RECONFIGURATION

PURPOSE OF DDR

DDR (Dynamic Device Reconfiguration) enables the system to attempt to bypass the source of various I/O errors by allowing a demountable volume to be moved from one device to another. A request to move (or swap) a volume may be initiated by either the system or the operator.

DESCRIPTION OF DDR

DDR is a standard feature of OS/VS. DDR operates under its own task control block and has its own request queue element, I/O block, data control block, data extent block, channel end appendage, and abnormal end appendage. It is executed in the supervisor state.

The system requests DDR after a permanent I/O error has occurred. The operator may request DDR at any time by issuing the SWAP command.

A swap involves two exchanges: the exchange of information between the UCBs (unit control blocks) for the two devices and the physical exchange of the volume from one device to another. The exchange of UCB information is performed by DDR using SVC 15. The operator is then requested by DDR to move the volume to the new device.

DDR SUPPORT

Volumes can only be swapped between devices of the same type; for example, a volume can be moved from one 2314 drive (the FROM device) to another 2314 drive (the TO device). Only the following devices are supported:

- Direct access devices: 2314, 2319, 3330, 3330-1 and 3340.
- Tape drives: 2400 series and 3400 (2400 equivalent) series.
- Unit-record devices: 1403 printer, 1404 printer, 1442 card read punch, 2501 card reader, 2520 card read punch, 2540 card reader, 2540 card punch, and 3211 printer, 3505 card reader, and 3525 card punch.

A volume can be moved from one device to another, or a volume can be swapped to itself. A volume is swapped to itself if:

- The volume demounted and then remounted on the same device.
- Or, in the case of a 2314, 2319, 3330, 3330-1, or 3340, both the volume and the address plug are moved to another drive.

Other DDR restrictions include the following:

- Teleprocessing devices are not supported.
- Emulator 7-track tapes in original, second generation, format are not supported.
- The system residence volume cannot be specified as the TO device.
- Swap requests for unit-record devices can only be initiated by the operator and then only during intervention-required conditions.
- A shared volume can only be swapped to itself.
- Tape volumes can be swapped only if the following conditions are met:
  - The swap is between tapes with the same number of tracks and the same densities (for example, 9-track 800 BPI to 9-track 800 BPI).
  - If the tape does not have standard labels, the user must supply a program to reposition it after the swap.
  - The block count cannot equal 0.
  - The user is keeping an accurate block count, when the EXCP macro instruction is being used.
  - The TO device is not allocated.
- An OPEN, CLOSE, or EOV must not be active for a device involved in a swap.
- An SVC DUMP must not be active for a device involved in a swap.
- A permanently resident volume cannot be swapped.

In addition to the above restrictions, DDR processes system-initiated swaps for only these conditions:

- A unit check with one of the following indicated in the sense information: bus-out, equipment check, data check, overrun, seek check, unsafe, ALU check, unselected status, read safety, end of cylinder, seek incomplete, or any condition indicated in the data check field.
- One of the following indicated in the channel status word: channel data check, channel control check, interface control check, or chaining check.

When a permanent error is caused by the channel program (such as command reject), DDR is not requested by the system. Other conditions that may set permanent I/O error status, but that are not eligible for DDR, include: wrong length record, no record found, unit exception, program check, protection check, IOB intercept condition, and backing into load point on tape. Also, for tape, a SWAP occurs if "INTERVENTION REQ" or "NOT READY" status exists.

GENERATING DDR

The exact configuration of DDR depends upon the options specified during system generation. Unless otherwise specified, DDR is included in VS1. It is always included in VS2. For VS1, DDR is generated to execute swap requests for the system residence volume (SYSRES) unless otherwise specified. DDR under VS2 does not support swap requests for SYSRES.

If DDR is to process tapes that do not have standard labels, DDRNSL must be specified in the OPTIONS parameter of the CTRLPROG macro instruction, and the user must provide a routine to verify the tape labels. See "Repositioning Tapes without Standard Labels" in DDR Section 2 for details.

This section describes the functions of DDR and how DDR performs these functions. Many references are provided to material in other sections of this publication and to the DDR program listing. Two types of information are presented:

- Text that discusses the design of DDR.
- Method of operation diagrams that show the functions performed by DDR in the order that they occur. The module name and label of each processing step are included in the implementation notes for each diagram. These diagrams can be used for quick reference when reviewing DDR functions. Diagram DDR-1 shows how the method of operation diagrams for DDR are organized.

OVERALL OPERATION

DDR performs three major functions when processing a swap request:

- Verifies the request.
- Controls the execution of the request.
- Executes the request.

DDR Verification

The verification routines first receive control when a DDR request is initated. The request can be initiated either by the system, if a permanent I/O error is encountered, or by the operator. DDR verifies the request and then initiates DDR execution by posting the event control block for DDR. Diagram DDR-2 shows how DDR requests are verified and how execution is initiated for system-initiated swaps. Diagram DDR-3 shows the same operations for operator-initiated swaps.

DDR Control

The control routines route control to the routines that process a DDR request and that terminate DDR execution. Diagram DDR-4 shows how DDR controls both system-initiated and operator-initiated swap requests.

DDR Execution

The execution routines are those routines that process a DDR request. The DDR execution routines:

- Gain control of the devices involved in the swap.
- Swap UCB information.
- Request and monitor the exchange of the volumes.

Diagram DDR-5 shows the execution of operator-initiated swaps (except for SYSRES). Diagram DDR-6 shows the execution of system-initiated swaps (except for SYSRES). Diagram DDR-7 shows the execution of a swap (both operator- and system-initiated) involving SYSRES.

DDR COMMUNICATIONS AREA

A data area (IORMSCOM) is used to maintain communications between the routines that make up DDR and to establish interfaces between DDR and other parts of the operating system. IORMSCOM contains the addresses of pertinent control blocks and information about the status of DDR. A detailed description of IORMSCOM can be found in DDR Section 5.

ACTIVATING DDR

ECB for DDR

The DDR Resident routine (IGFDDRMF for VS1, IGFDDRMV for VS2) resides in the nucleus. If DDR is not processing a DDR request, this routine waits on DDRECB. DDRECB is the ECB (in IORMSCOM) that is posted by one of the DDR verification routines IGE0660A or IGF2503D) to initiate the processing of a verified DDR request.

DDR Control Flow

If a DDR request is operator-initiated, the SWAP command is passed to the SWAP Command Processor (IGC2503D) by the system Command Router (IGC0403D). IGC2503D verifies the command and, if the request can be processed, posts DDRECB.

If DDR is initiated by the system, the Outboard Recorder (OBR) passes control to DDR Central (IGE0660A). IGE0660A verifies the request. If the request is valid, the RQE associated with the request is placed on the DDR wait queue, and DDRECB is posted.

The above discussion does not apply to the VS1 system residence volume. If the

swap request for SYSRES is operator-initiated, DDR processes it as any other other operator-initiated request until IGC0108E gains control. Upon determining that the request is for SYSRES, IGC0108E returns control to IGFDDRMF. IGFDDRMF then branches to IGFDDRSR to process the request.

The system initiates a DDR request for SYSRES as follows. Each time a permanent error is encountered involving a fetch from SYS1.SVCLIB, the disk ERP IEC23xxF (error recovery procedure) branches to IGFDDR05 (in IGFDDRSR). After five unsuccessful attempts by the system to fetch a module from SYS1.SVCLIB, IGFDDR05 determines whether DDR is to handle the error, and then sets a return code and exits to the disk ERP. If DDR is to handle the error, the disk ERP returns control to IOS to post the error and to gain control of SYSRES for DDR. Control is then returned to IGFDDRSR to process the request. Figure DDR-1 illustrates this procedure.

SVC 85

When DDRECB is posted, module IGFDDRMF/MV issues SVC 85 to start processing the swap request. SVC 85 consists of 10 loads (modules IGC0008E - IGC0908E). For VS1, these modules reside in SYS1.SVCLIB; for VS2, these modules reside in SYS1.LPALIB.

Each module of SVC 85 determines its successor and uses SVC 7 (XCTL) to pass control to the next module. When SVC 85 has completed execution, load 0 (IGC0008E) issues SVC 3 to return control to IGFDDRMF/MV.

DDR WAIT QUEUE

A wait queue is used to hold requests for system-initiated swaps. (Operator-initiated requests cannot be queued; they are rejected if DDR is processing a request). Since only one request can be processed at one time by DDR, system-initiated requests are queued until the current request has been processed. The RQEs of failing operations are queued by IGE0660A. When DDR has completed processing one request, it checks the wait queue for any waiting requests.

Since RQEs may be removed from the wait queue by the SVC purge routine of the I/O supervisor, DDR checks the wait queue and the wait queue purge indicator at key points in its execution. If the RQE is no longer on the wait queue or if the purge bit is set, the RQE has been purged. DDR then stops processing that request and checks for another request on the queue. DDR also checks to see that certain user control blocks are still valid.

INTERACTION WITH DEVICE ALLOCATION

Because of a possible conflict between DDR and Device Allocation, from the time Device Allocation determines that a device is eligible for allocation until it marks the device allocated, concurrent execution of DDR and Device Allocation is prevented, as follows:

- Before Device Allocation can allocate a device, it issues a WAIT on ALLOCECB (an ECB in IORMSCOM). If DDR is not processing a request, ALLOCECB is posted, so that Device Allocation can proceed.
- If Device Allocation is not being executed when DDR is activated, DDR marks ALLOCECB waiting.
- If Device Allocation is being executed for a task not under TSO when DDR is activated, DDR sets the task nondispatchable.
- If Device Allocation is being executed for a task under TSO, but TSO is not ready, DDR marks ALLOCECB waiting.
- If Device Allocation is being executed for a task under TSO that is in real storage, DDR issues the TSEVENT macro instruction with a code of 45, causing the task to be swapped out. If the task is swapped back in before DDR has been executed, the task is set nondispatchable.
- If Device Allocation is being executed for a task under TSO that is not in real storage, DDR sets the TJBDDRND bit in the task's terminal job block. If the the task is swapped back in before DDR has been executed, the task is set nondispatchable.
- After DDR has processed a request, it posts ALLOCECB and sets the task (if any) for which Device Allocation is being executed dispatchable.

Note: TSO (time-sharing option) pertains only to VS2.

I/O HANDLING -- IOS-DDR INTERFACE

An IOS-DDR interface handles I/O for DDR. With its IOB error flags off, DDR uses its RQE to issue SVC 15. Control is passed to DDR's channel end appendage (IGFDDR05 if DDR is being executed for

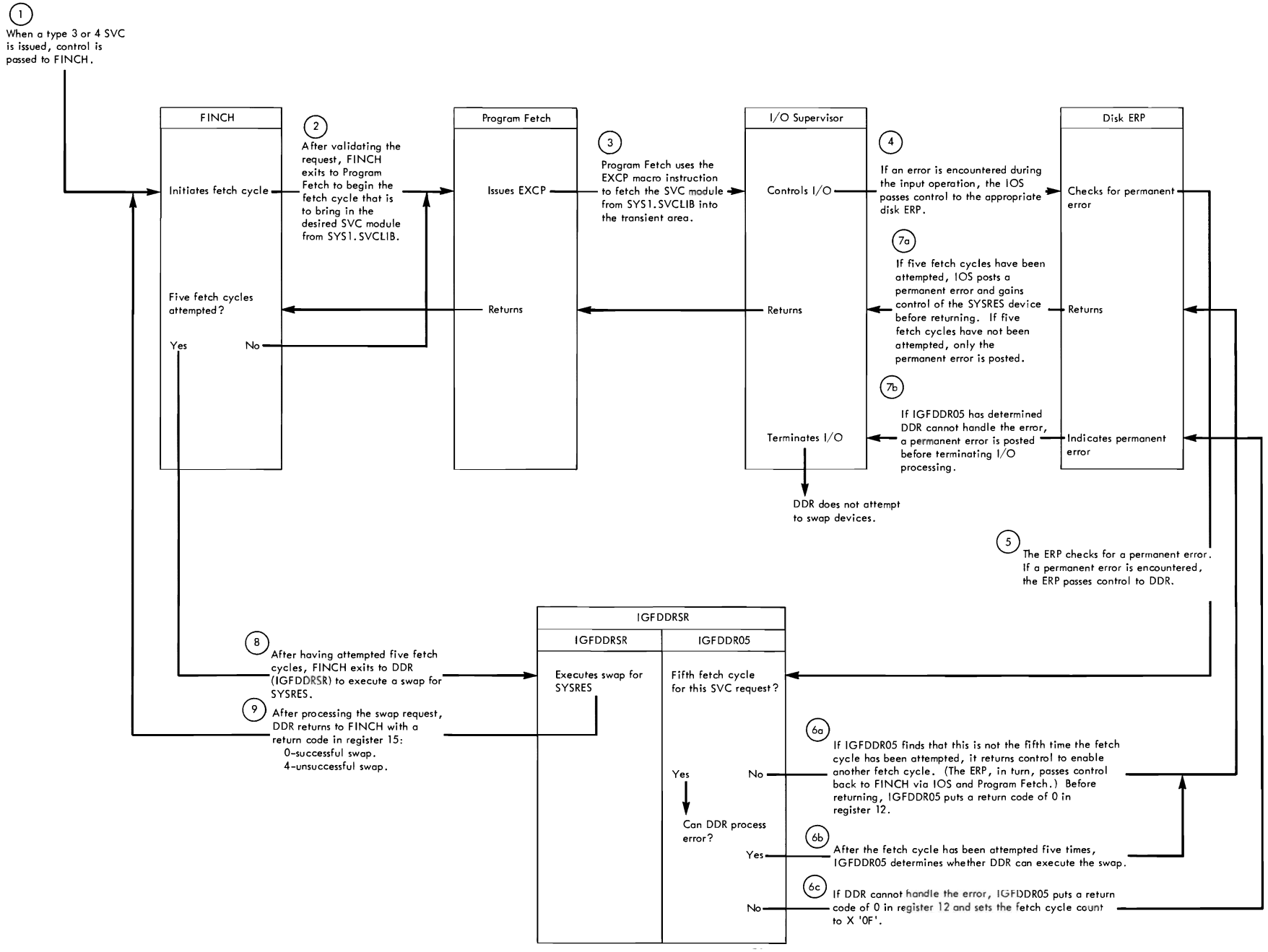


Figure DDR-1. How DDR SYSRES (IGFDDRSR) gains control to process a SWAP request from the system for SYSRES (VS1 only)

SYSRES, otherwise IGFVDDR2). Using a pointer (IOSVT in IORMSCOM) to the IOS-DDR vector table, and the proper displacement, the appendage branches to the part of IOS that performs the needed function. The appendage uses the DDR flags in IORMSCOM to determine the proper displacement; these flags are set by the DDR routine issuing SVC 15.

See "IOS-DDR Vector Table" in DDR Section 5 for a description of the vector table.

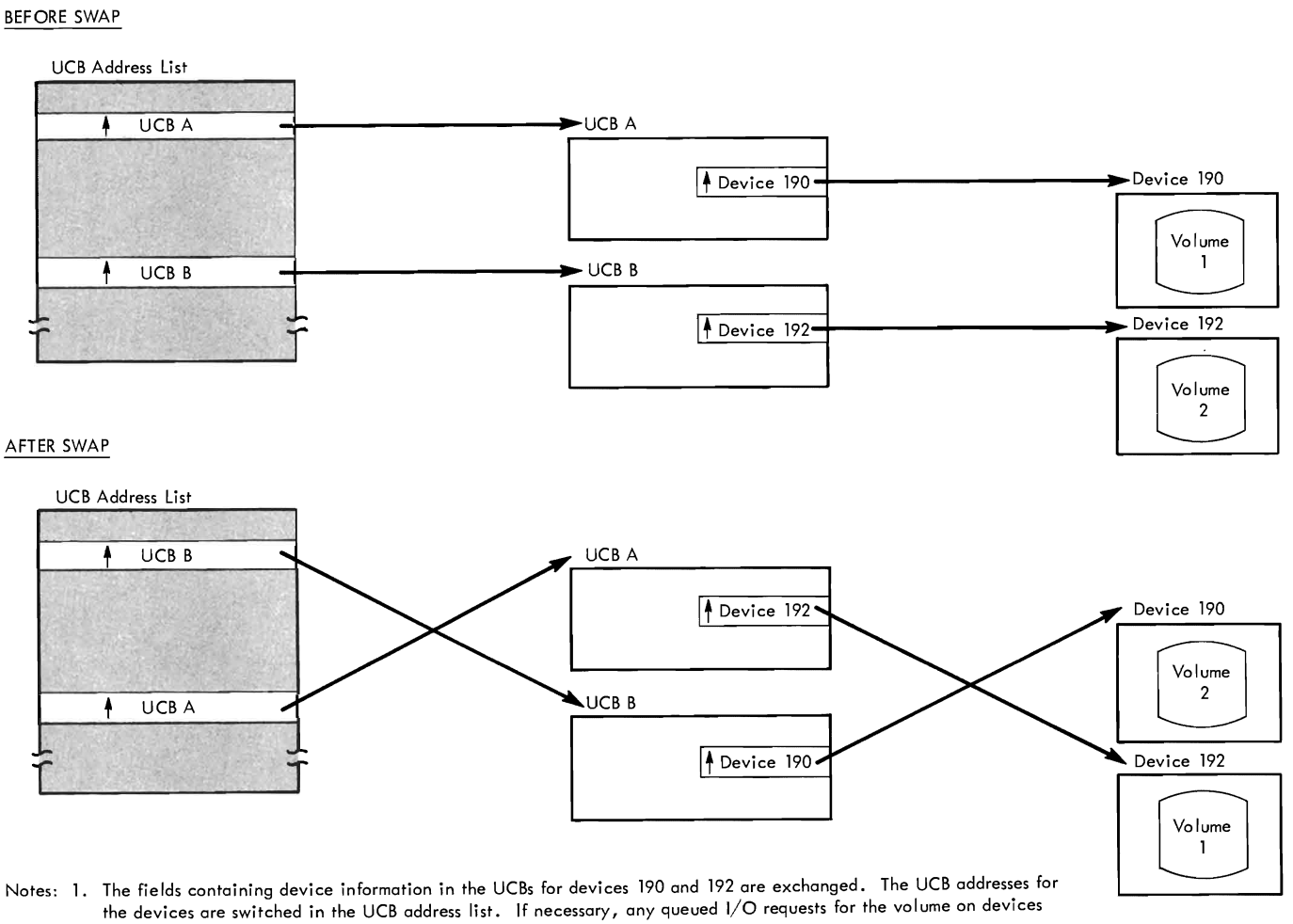
EXCHANGING UCB INFORMATION

When a volume is moved from one device to another, the device retains its UCB so that control blocks containing UCB addresses need not be changed. The UCB addresses for the two devices are switched in the address portion of the unit control block look-up table. See Diagram DDR-5 for those fields that are exchanged in the UCB during a swap.

If the exchange is between logical channels, any queued I/O requests for the volumes involved in the exchange are also interchanged so that they are on the correct logical channel. UCBNRY is set in both UCBs. Figure DDR-2 shows the results of DDR execution.

REPOSITIONING TAPES WITHOUT STANDARD LABELS

At the completion of a swap involving tape data sets, DDR attempts to verify and reposition the volumes. DDR processes swap requests only for tapes with standard labels or no labels, unless the user provides his own routine to verify and reposition tapes that do not have standard labels.



Notes: 1. The fields containing device information in the UCBs for devices 190 and 192 are exchanged. The UCB addresses for the devices are switched in the UCB address list. If necessary, any queued I/O requests for the volume on devices 190 and 192 are exchanged so as to be on the correct queue.

2. The DDR "PROCEED WITH SWAP" message is issued, and the operator exchanges the volumes on devices 190 and 192.

Figure DDR-2. Effects of DDR on system pointers

The user-written routine must be named NSLREPOS and must be placed in SYS1.SVCLIB, for VS1, or SYS1.LPALIB for VS2. When DDR is ready to verify and reposition a tape without standard labels, it issues an XCTL macro instruction and passes control to NSLREPOS.

Upon entry to NSLREPOS, register 0 contains the address of a two-word parameter list. The first word is the address of a 48-byte field containing the label to be verified; the second word is the address of the UCB for the device. Register 2 contains the address of an XCTL list for NSLREPOS to use to return to DDR. Register 5 contains the address of the SVRBs extended save area.

When NSLREPOS completes its task, it should pass control (via the execute form of the XCTL macro instruction) to IGC0408E with one of the following codes in register 0:

Code	Explanation
0	Volume verification is complete. A tapemark follows the label; therefore the DDR tape reposition routine should forward space to the tapemark and clear the block count it had accumulated before repositioning.
4	More information is needed by NSLREPOS for volume verification. When the DDR tape reposition routine receives this code, it reads the first 48-bytes of the next record and returns to NSLREPOS.
8	The wrong volume has been mounted. When the DDR tape reposition routine receives this code, it sends a message to the operator indicating the error and requesting that the correct volume be mounted.
C	Volume verification is complete. No tapemark follows the label so the DDR tape reposition routine should reposition the volume using the block count it has accumulated.
10	Volume verification is complete. The tapemark following the label has already been reached; the DDR tape reposition routine clears the block count it has accumulated and repositions the volume.

See the Tape Labels publication for additional details on writing the NSLREPOS program.

RETRY PROCEDURE BY DDR AFTER SWAP

For non-SYSRES devices, DDR sets up a request for the I/O operation that caused the swap to be initiated. SVC 15 is then issued for that request. If the I/O operation fails again, the ERP is reentered. This entry is treated as a new error, and the ERP performs its normal functions. If the ERP determines that the error is permanent, DDR is requested again.

For SYSRES devices, a permanent error was posted before DDR was initiated. The retry is a new attempt to perform the I/O operation. If this retry fails, DDR does not try another recovery since the device is now judged not to be the cause of the failure.

DDR METHOD OF OPERATION DIAGRAMS

These Method of Operation Diagrams show the functions of DDR and relate these functions to the exact modules and entry points in the code where they are performed.

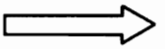
These particular diagrams are known as HIPO diagrams because they show Hierarchy as well as Input, Processing, and Output. The first diagram is a visual table of contents. The rest of the diagrams are arranged in a hierarchy according to the level of detail.

Read the diagrams left to right, input-processing-output. The processing steps are numbered sequentially; these numbers also refer to the implementation notes at the bottom of most of the diagrams. The notes contain additional information about the processing steps.

The arrows are used as follows:

Shows control flow to or from another program.

Primary processing flow. Shows the path followed to accomplish the principal function of the body of code.



Data transfer. Indicates that data is moved from one location to another.

Control information transfer. Indicates the setting or changing of switches or pointers that will be used to determine the course of future processing.

Pointer. Indicates that a field in one data area contains an address that points to another field or data area.

Data reference. Indicates that the contents of a data area are tested or read in order to determine the course of subsequent processing.

Diagram DDR–1. DDR Overview and Table of Contents

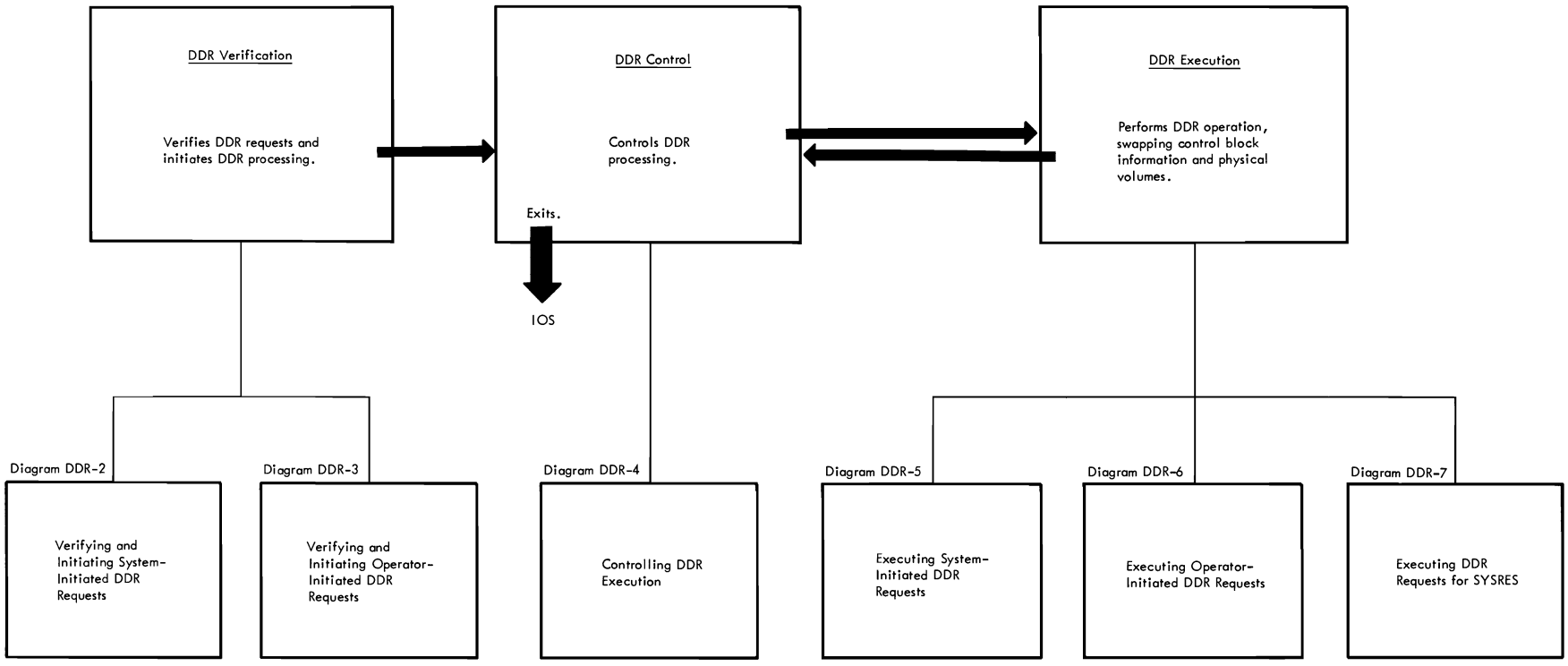
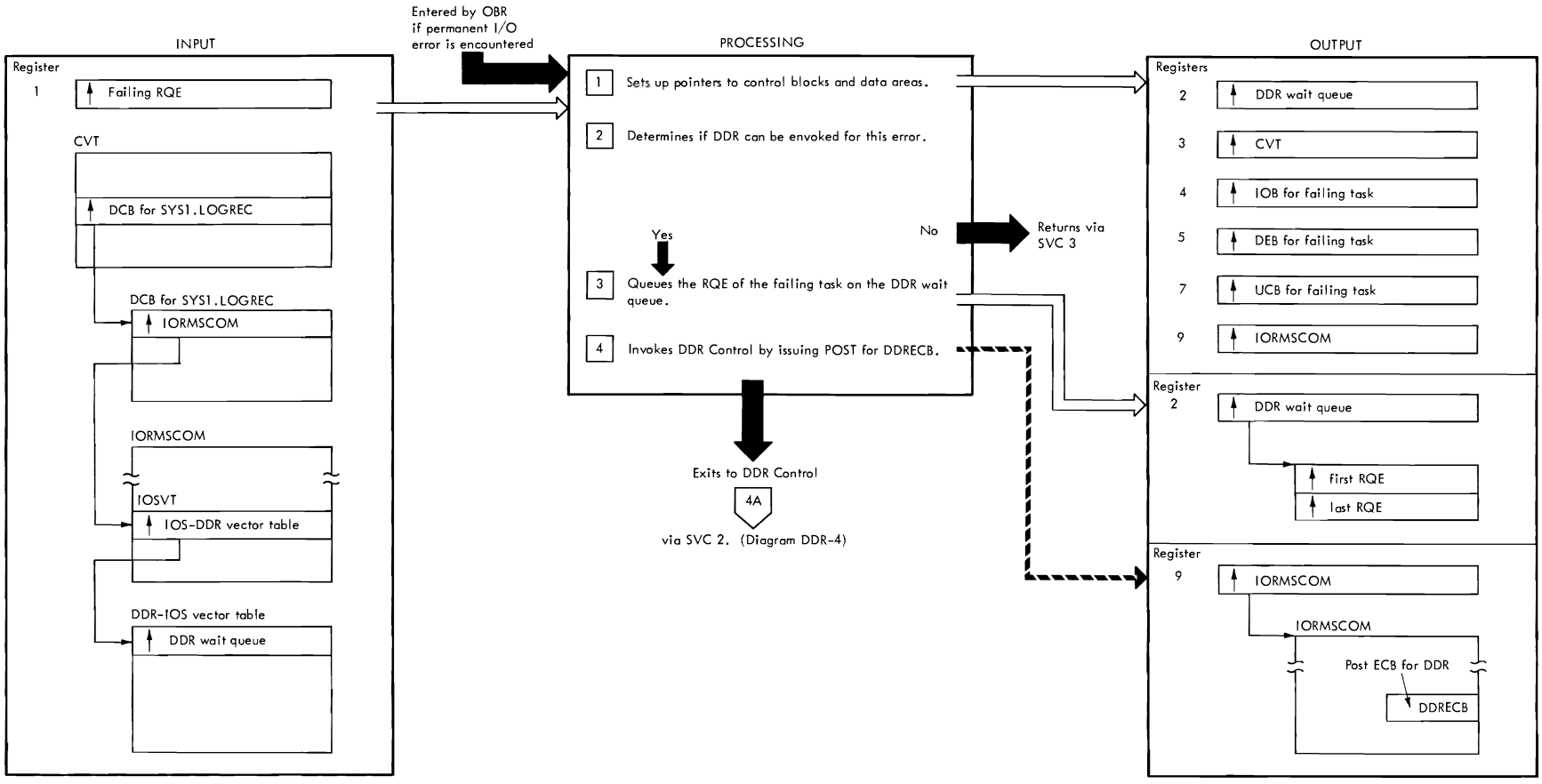


Diagram DDR-2. Verifying and Initiating System-Initiated DDR Requests

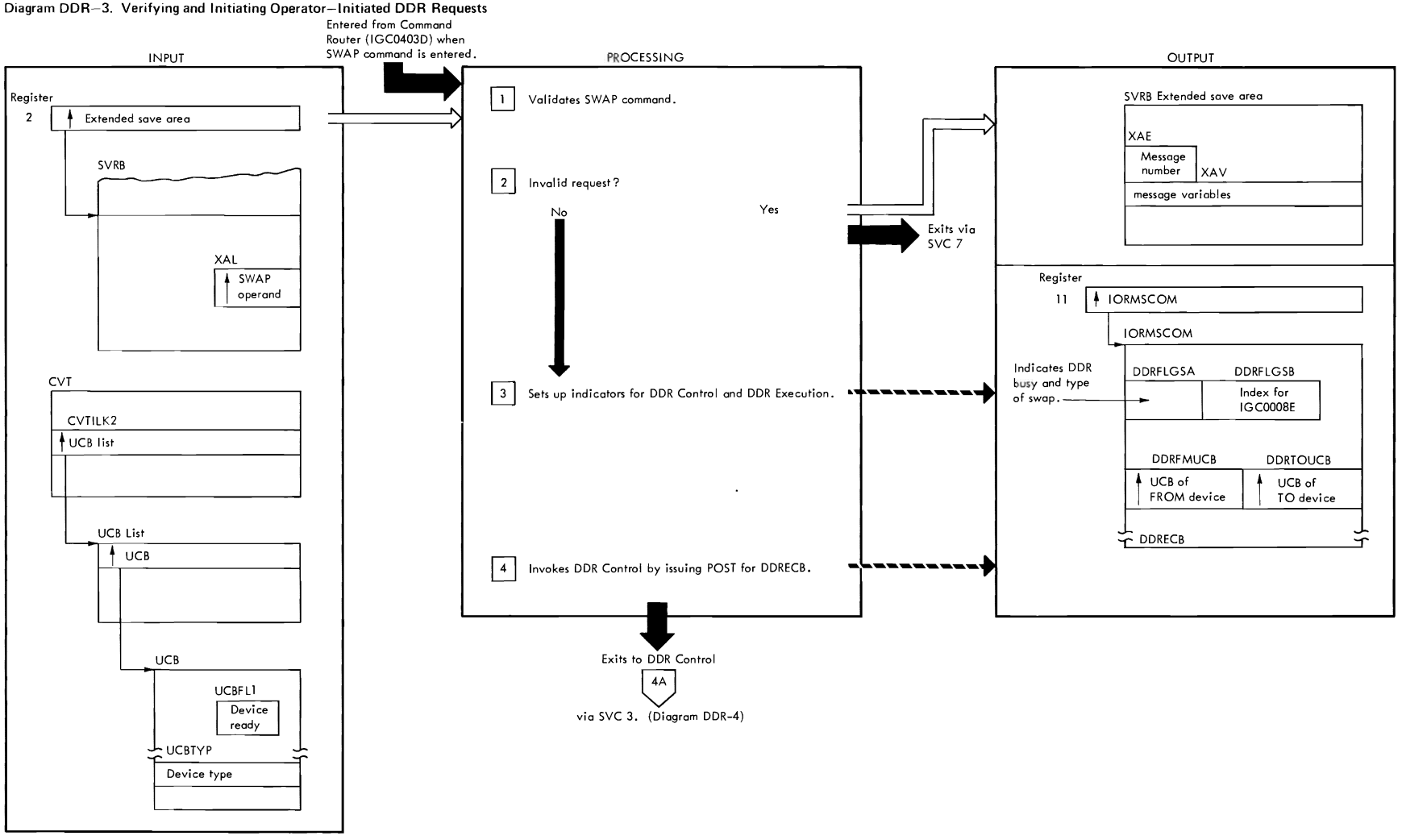


Implementation Notes for Diagram DDR-2

Note (VS1 only): This diagram does not illustrate verification of system-initiated swaps involving the system residence volume (SYSRES). Such SYSRES requests are passed by a disk ERP to IGFDDR05 (in IGFDDRSR). IGFDDR05 verifies the request and returns to caller. A return code of 4 is placed in register 12 if DDR can handle the request; a return code of 0 is placed in the register if DDR cannot handle it. (See Diagram 7 for details on SYSRES processing.)

	Module	Label
1 Whether DDR can process the error depends upon the device, the I/O operation, the access method, and the error itself. If DDR cannot process the error, SVC 15 is issued to permit IOS to post the error. After the error is posted, IGE0660A exits via SVC 3.	IGE0660A	IGE0660A DDR07 DDR01 PERREXIT
3 The RQE is queued via SVC 15. If the queue is empty, the DDRBUSY flag in DDRFLGSA is set and the branch index in DDRFLAGSE is initialized to X'08'. The X'08' is used by IGC0008E to route control during DDR execution to IGC0208. Both DDRFLGSA and DDRFLGSB are in IORMSCOM.	IGE0660A IGC0008E	DDR06 DDRENQ BRSTART
4 DDRECB is posted only if DDR is not active. DDRECB is posted so that IGFDDRMF/MV can gain control.	IGE0660A IGFDDRMV	DDR06 IGFDDRMF/MV

Diagram DDR–3. Verifying and Initiating Operator–Initiated DDR Requests



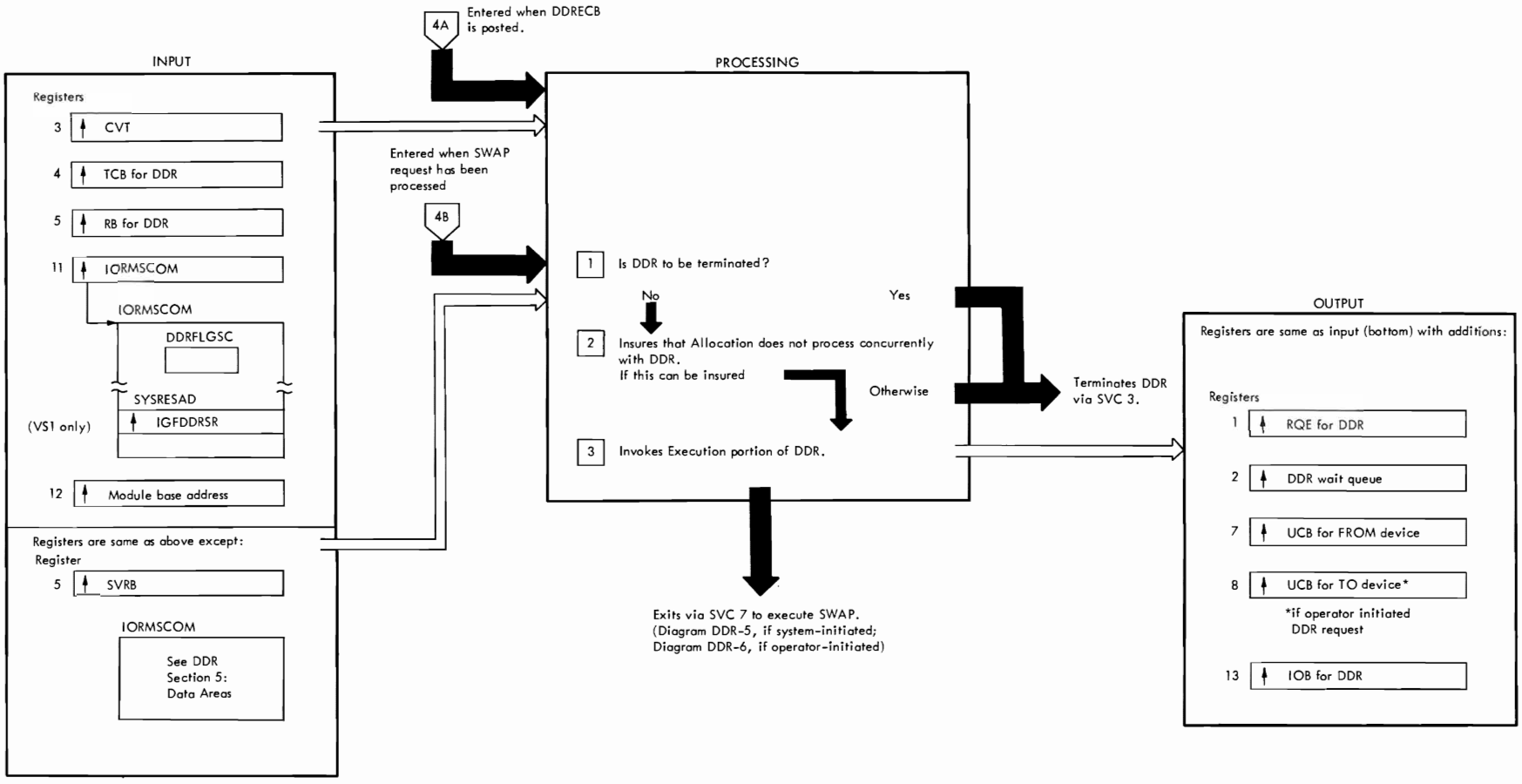
Implementation Notes for Diagram DDR-3

Verification of operator-initiated requests are processed by IGC2503D. IGC2503D receives control from the SVC 34 module IGC0403D when a SWAP command is entered.

		Module	Label
1	If MCS is in the system, only a SWAP command from an authorized console is accepted. In addition, if the format of the command is invalid, if DDR is currently active, if the SYRES device is specified, or if invalid devices are specified. A branch index is set up to allow the router to branch to the appropriate module.	IGC2503D	SWAP00 SWAP01 SWAP50 SWAP07

		Module	Label
2	if DDR is not to be invoked, message parameters are set up in the SVRB extended save area. Control is then returned to IOS.	IGC2503D IGG2103D IGC0503D	SWAP70 IGG2103D IGC0503D
4	DDRECB is posted so that IGFDDRMV can gain control	IGC2503D IGFDDRMV	SWAP22 IGFDDRMV

Diagram DDR–4. Controlling DDR Execution



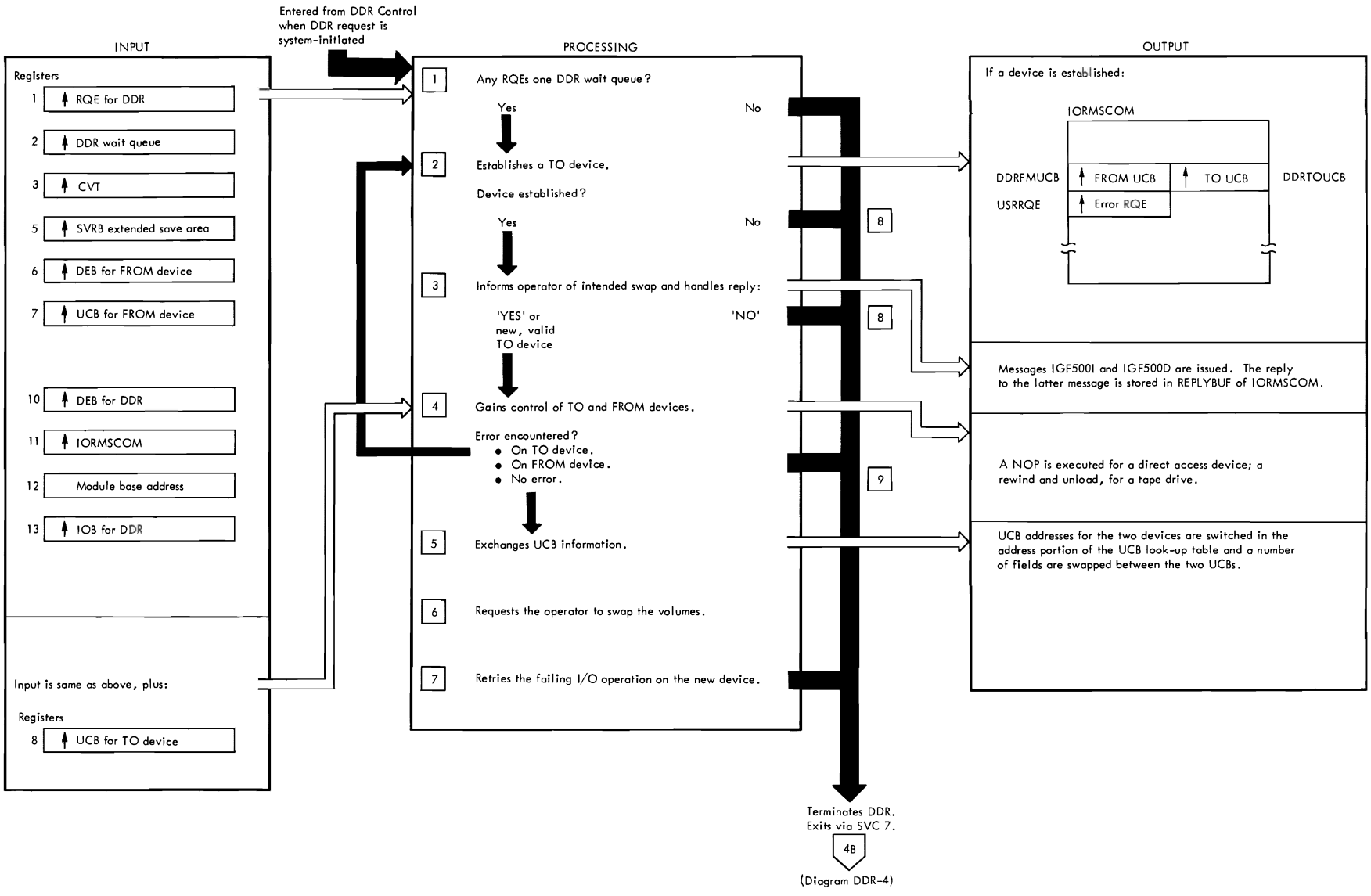
Implementation Notes for Diagram DDR-4

The two modules involved in DDR Control are IGFDDRMV and IGC0008E.

		Module	Label
1	IGFDDRMF/MV issues a WAIT macro instruction for DDRECB (in IORMSCOM) when all current requests for DDR have been processed. DDRECB is posted when a DDR request is initiated.	IGFDDRMV	IORMSWT
		IGE0660A IGC2503D	DDR06 SWAP22
	IGC0008E handles the control functions indicated by notes 2 through 5. IGC0008E receives control from IGFDDRMV when DDR is first initiated, passes control to other DDR routines to perform the requested swap, regains control when these other routines are complete, and terminates DDR execution.	IGC0008E	IGC0008E
	After IGC0008E is entered by IGFDDRMV, all DDR termination is handled by IGC0008E. Before terminating DDR, IGC0008E clears the following flags in IORMSCOM: DDRFLGSA (all but DDRON and NSLAVAIL), DDRFLGSB, C, D, E, F. It also clears the DDRECB to prepare for any additional requests. In addition, ALLOCECB is posted and the task (if any) for which Allocation is processing is set dispatchable (see note 3 below).	IGFDDRMV	IORMSWT ENDDDR
		IGC0008E	CHKWAITQ
	DDR is normally terminated when all requests for DDR have been processed: that is, when the DDR wait queue is empty. If SVC 85 is issued by any task except the DDR TCB, IGC0008E issues on ABEND macro instruction with a code of 155.		SINK

		Module	Label
2	To eliminate any conflict between Device Allocation and DDR, these two functions are not permitted to operate concurrently. DDR takes precedence over Allocation. How DDR restricts Allocation execution depends on whether Allocation is active when IGC0008E gains control, and if active, whether the task using Allocation is under TSO (VS2 only). See "Interaction with Device Allocation" in DDR Section 2ffor details on how DDR controls Allocation.	IGC0008E	CHKDAR CHKTSO ZEROECB CHKENQ
	A DDR request is not executed if Allocation is executing for a task that is queued for WTO or WTOR write queue elements (WQEs).		SYSTMTYP COMPTCB RESET
	Before issuing SVC 3, IGC0008E zeros the DDR ECB (DDRECB). SVC 3 returns control to IGFDDRMF/MV which in turn waits on the ECB to be posted.	IGC0008E IGFDDRMV	CHKWAITQ IORMSWT
3	DDRFLGSB in IORMSCOM contains a branch index code indicating the routine that is to execute the first DDR request. The code is set up by either IGE0660A or IGF2503D. The code depends on who initiated the request and the type of device involved in the swap.	IGC0008E	GETCODE
		IGE0660A IGF2503D IGC0108E IGC0208E IGC0408E	DDRENQ SWAP22 IGC0108E IGC0208E IGC0408E
	Additional queued requests are always passed to IGC0208E, because queued requests can only be system-initiated.		
	When DDR Execution is completed for each RQE, control is returned to IGC0008E via the XCTL macro instruction.		

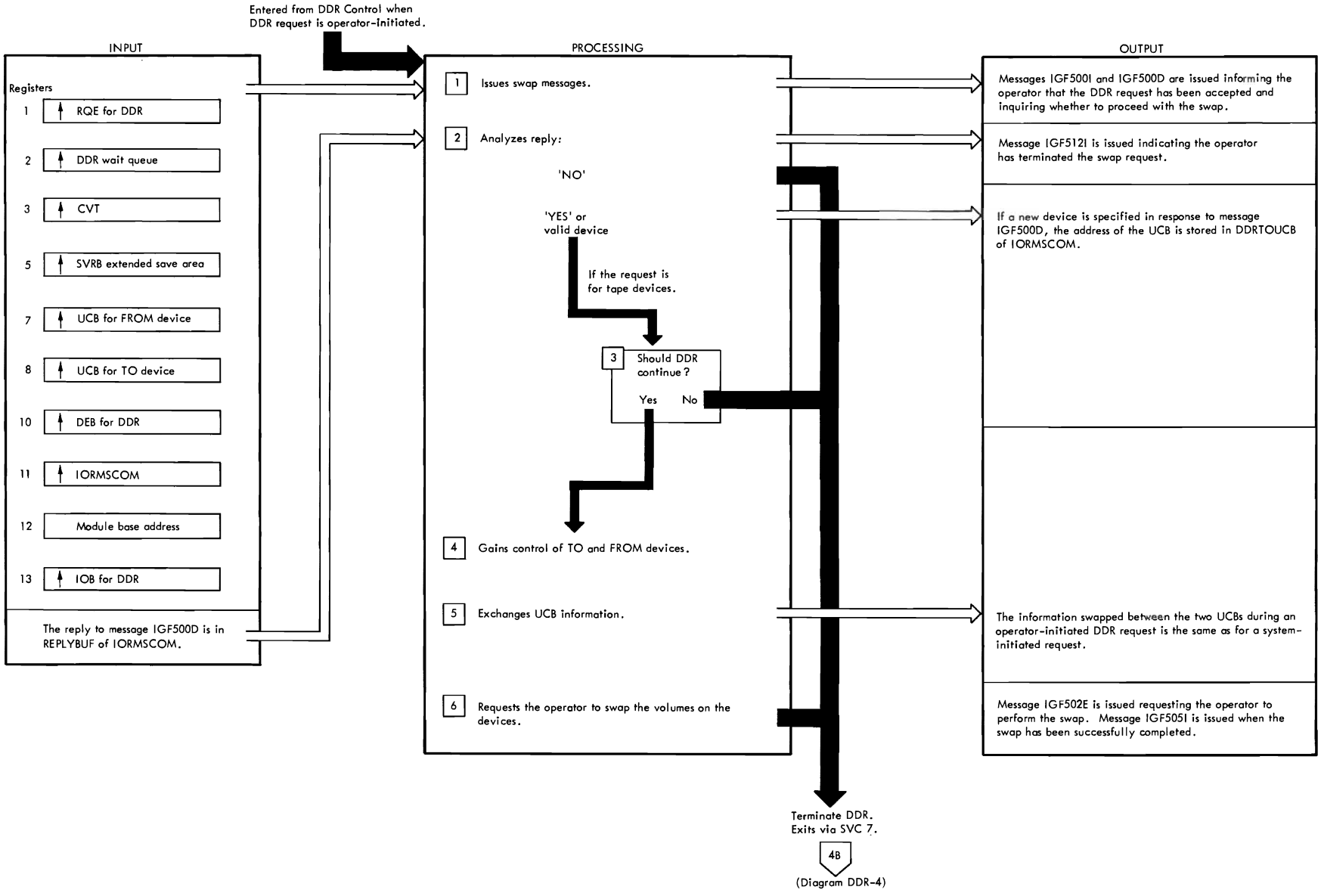
Diagram DDR–5. Executing System–Initiated DDR Requests



System-initiated swap requests are executed by obtaining a suitable device with which to perform the swap, swapping control information relating to the devices, and requesting the operator to perform the physical swap.

		Module	Label
1	The controlling routine during the execution of a system-initiated swap is IGC0208E. This routine gains control from IGC0008E when a system-initiated request is encountered. IGC0208E attempts to process all DDR requests on the DDR wait queue before returning to IGC0008E. The termination flag is set in IORMSCOM before control is returned to IGC0008E.	IGC0208E IGC0008E IGC0208E	IGC0208E GOLOAD2 IGC0208E ENDDDR
2	IGC0208E first attempts to find a suitable device to use as a replacement by scanning available UCBs. A suitable device must meet the following specifications: • Device type is the same as the failing device. • The device is online. • The device is not allocated. • No mount message is pending for the device. • For tape devices, the optional features and densities must match those of the failing device.	IGC0208E	DEVICECK
	If a suitable device cannot be found, IGC0508E is invoked to issue message IGF509I followed by message IGF500D. These messages permit the operator to indicate a suitable device or to cancel the DDR operation. Replies to message IGF500D are validated by IGC0208E. If the reply is invalid, IGC0508E is reinvoked to issue invalid-reply message IGF513I followed by messages IGF509I and IGF500D. If a valid device is indicated by the operator, processing is as for 'YES' in the next step.	IGC0508E IGC0208E IGC0508E IGC0208E	NOTOMSG REPLYMSG CHKREPLY CHKNOTO CHKSWAP SWAPON DEVICECK
3	IGC0208E invokes IGC0508E to issue the required messages. IGC0208E then exits to IGC0308E to proceed with the actual swap. If the reply is invalid, processing is as indicated in note 2, above.	IGC0208E IGC0308E	ASKSWAP SWAPON IGC0308E
4	Control is gained by issuing SVC 15.	IGC0308E	PREPIO
5	The swap of UCB information is performed via SVC 15 in IOS. If the devices are 3330s, IGC0708E is invoked to read the buffered log and to set up a record of the information before UCB data is swapped.	IGC0308E IGC708E	SWAPTHEM TESTMERL GOLOAD7 GETOUT
6	IGC0508E is invoked to issue message IGF502E.	IGC0308E IGC0508E	SWAPTHEM DDRMSG1 CHKPROD
7	Following the volume swap, IGC0508E returns control to IGC0308E for direct access devices or passes control to IGC0408E for tape devices.	IGC0508E IGC0308E IGC0408E	GOXCTL CHKMERL IGC0408E
	For direct access devices: The failing I/O operation is retried via SVC 15.	IGC0308E	SETRETRY
	If the device is a 3330, IGC0708E is invoked to write the record containing the error statistics before the I/O operation is retried.	IGC0708E	CHKMERL RECORDNG
	For tape devices: IGC0408E clears DDRFLGSB in IORMSCOM, then exits to IGC0608E. When the label has been verified, control is returned to IGC0408E to reposition the tape and return control to IGC0008E.	IGC0408E IGC0608E IGC0408E IGC0008E	IGC0408E CHKLABEL GOPOSITN CHKFILE IGC0008E
8	If a swap request is terminated without completing the swap, SVC 15 is issued to dequeue the RQE and to post a permanent error. The DDR request is then terminated by passing control to IGC0008E via IGC0808E. This message routine issues message IGF512I before exiting to IGC0008E.	IGC0208E IGC0008E IGC0808E	CHKREPLY IGC0008E CHKEND
9	If DDR is to be terminated, IGC0008E receives control via IGC0808E.	IGC0308E IGC0008E IGC0808E	IO IGC0008E IGC0808E

Diagram DDR–6. Executing Operator–Initiated DDR Requests



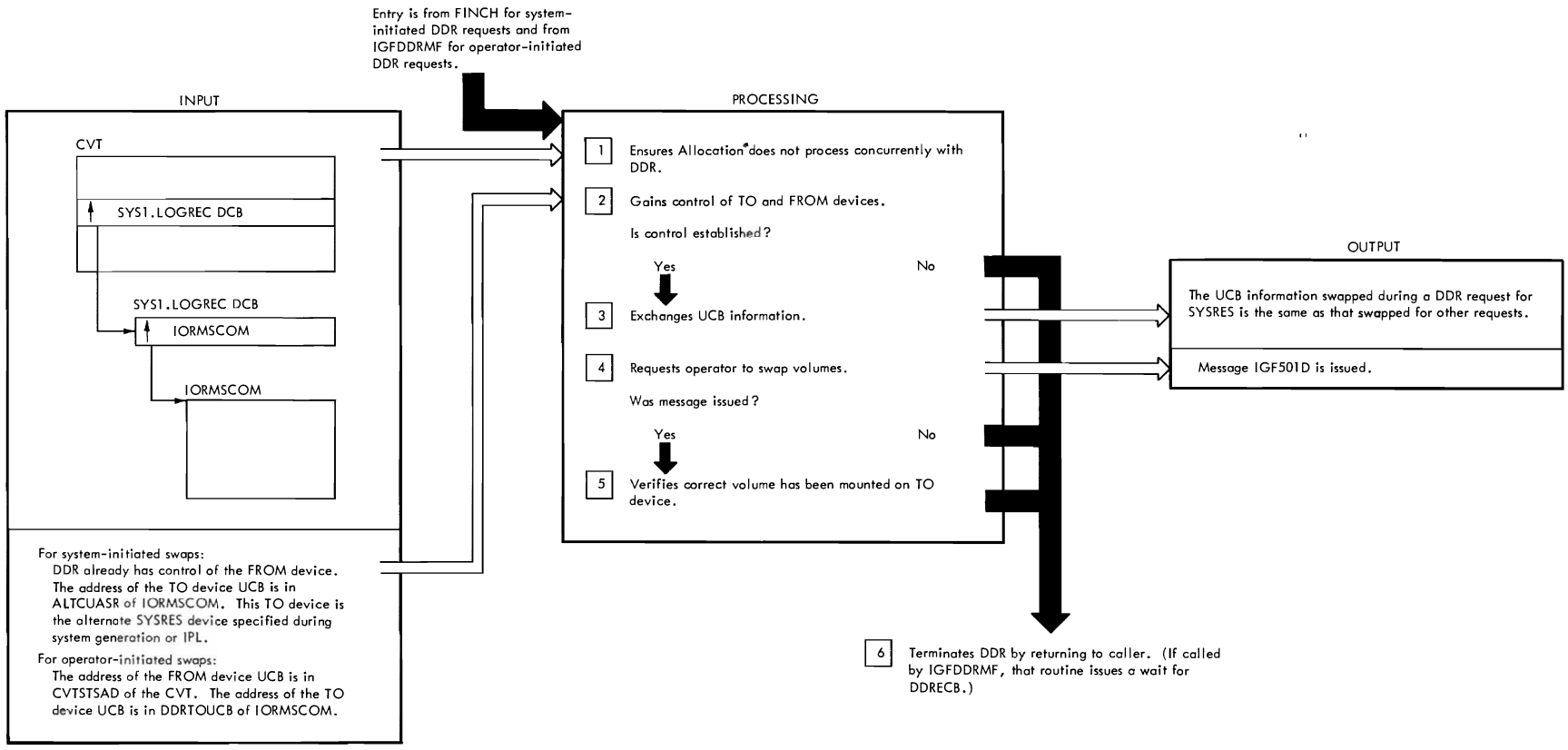
Implementation Notes for Diagram DDR-6

The primary routines involved in executing operator-initiated swap requests are IGC0108E (for direct access and unit-record devices) and IGC0408E (for tape devices).

NOTES FOR DIRECT ACCESS AND UNIT-RECORD DEVICES	Module	Label
1 IGC018E gains control from IGC0008E when operator-initiated swap requests involving direct access or unit record devices are encountered. IGC0108E invokes IGC0508E to issue message IGF500I followed by message IGF500D.	IGC0108E IGC0008E IGC0108E IGC0508E	IGC00108E GOLOAD1 CHKSWAP THISLOAD CHKSWAP REPLYMSG
2 IGC0108E regains control to analyze the reply to message IGF500D. A 'NO' reply causes IGC0108E to terminate processing by exiting to IGC0008E via IGC0808E. This message routine issues message IGF512I before passing control to IGC0008E. If the reply is an invalid device, IGC0508E is invoked to issue message IGF513I followed by messages IGF500I and IGF500D again.	IGC0108E IGC0108E IGC0008E IGC0808E IGC0108E IGC0508E	CHKREPLY TERMDDR IGC0408E CHKEND CHKNUMBR BADDEV CHKSWAP SWAPON
4 SVC 15 is issued to gain control of the devices. This SVC 15 is issued to execute a no-op CCW. (Step 5 is skipped if the device involved is a unit-record device.) If an error is encountered during the no-op, processing is as indicated for invalid replies in note 2 above.	IGC0108E	INITBLKS DASDEXTN SWAPIT DASDEXTN BADDEV
5 The UCB information is swapped using SVC 15. If the devices are 3330s, IGC0708E is invoked to read the buffered log and to set up a record of the information before UCB data is swapped.	IGC0108E IGC0708E	CHKFIRST SWAPTHEM TSTCLASS IGC0708E GETOUT
6 IGC0508E is invoked to issue message IGF502E. Following the volume swap, IGC0508E returns control to IGC0108E. If the devices are 3330x, IGC0108E invokes IGC0708E to write the data record before exiting to IGC0008E.	IGC0108E IGC0508E IGC0108E IGC0708E IGC0108E	THISLOAD CHKPROD RCRMERL RECORDNG ENDDDR
NOTES FOR TAPE DEVICES		
1 IGC0408E gains control from IGC0008E when operator-initiated swap requests involving tape devices are encountered. IGC0408E invokes IGC0508E to issue message IGF500I followed by message IGF500D.	IGC0408E IGC0008E IGC0408E IGC0508E	IGC0408E GOLOAD4 THISLOAD GOXCTL CHKSWAP REPLYMSG
2 IGC0408E regains control to analyze the reply to message IGF500D. A 'NO' reply causes IGC0408E to terminate processing by exiting to IGC0008E via IGC0508E. This message routine issues message IGF512I before passing control to IGC0008E.	IGC0408E IGC0408E IGC0008E IGC0808E	CHKREPLY ENDDDR IGC0008F CHKEND

	Module	Label
If the reply is a new device, IGC0908E is invoked to verify the device. IGC0908E puts a return code in register 0 before returning to IGC0408E: 0=valid device; 4=invalid device.	IGC0408E IGC0908E	GOXCTL CHKOUT TODEV00 COMPTST2
If the reply is an invalid device, IGC0508E is invoked to issue message IGF513I followed by messages IGF500I and IGF500D again.	IGC0408E IGC0508E	CHKCODE INVALON CHKSWAP SWAPON
3 If DDR processing is to be terminated for this request, control is returned to IGC0008E via IGC0808E. This message routine issues message IGF512I before passing control to IGC0008E. DDR is terminated if any of the following conditions are encountered: • No task is using the FROM device. • Non-standard labels are used, but no user routine has been supplied to reposition the tape. • The task for which DDR has been invoked is using the EXCP macro but is not keeping an accurate block count. • An open, close, or EOV was in process for one of the devices involved in the DDR request. • The CNTRL macro was in process for one of the devices involved in the DDR request and the block count in the associated DCB is zero.	IGC0408E IGC0008E IGC0808E IGC0408E	ENDDDR IGC0008E CHKEND GETTCB CHKNSL CHKEXCP CHKOPEN
4 SVC 15 is issued to gain control of the devices. This SVC 15 is issued to execute a rewind and unload. If an error is encountered when executing SVC 15 for the TO device, processing is as indicated for an invalid reply in note 2 (for tape devices). If an error is encountered when executing SVC 15 for the FROM device, processing is terminated as indicated in note 4.	IGC0408E	INITBLKS IOSETUP NOCTRL
5 UCB information is swapped via SVC 15. If the error recovery procedures are in process for the devices, this SVC 15 is not issued; instead, IGC0408E terminates this DDR request by returning control to IGC0008E via IGC0808E. IGC0808E issues message IGF504I before passing control to IGC0008E.	IGC0408E IGC0008E IGC0808E	SWAPTHEM DDRMSG2 IGC0008E CHKERP
6 Message IGF502E is issued by IGC0508E. This routine is invoked by IGC0408E; after issuing the message, this message writer exits to IGC0608E. IGC0608E reads and verifies the labels on the swapped tapes. Standard labels are verified directly by IGC0608E. If the labels are not standard, control is passed to the user-supplied NSLREPOS routine to verify them. If labels cannot be verified, message IGF511A is issued by IGC0808E, and IGC0608E rewinds and unloads the tape via SVC 15. Following the rewind, IGC0608E reads the label from the new tape and begins verification again. After labels have been verified, the tape is spaced forward and control is passed to IGC0408E to reposition it. IGC0408E exits to IGC0008E (via IGC0808E) after repositioning the tape. IGC0808E issues message IGF505I.) If an error is encountered during repositioning, the tape is rewound and unloaded, and IGC0508E is invoked to issue message IGF503I followed by message IGF500I and IGF500D again. Processing then returns to step 2 (for tape devices).	IGC0508E IGC0408E IGC0608E IGC0808E IGC0608E IGC0408E IGC0008E IGC0808E IGC0408E IGC0508E	CHKRPOD SWAPTHEM DDRMSG1 READLAB CHKLABEL COMPLABL TESTNSI WRONGONE CHKWRONG SETRUN READLAB MOVETAPE CHKFILE READIO RETRYIT IGC0008E IGC0808E READIO ERRORRUN CHKERROR

Diagram DDR—7. Executing DDR Requests for SYSRES



Implementation Notes for Diagram DDR-7

Note: This diagram pertains only to DDR under VS1.

This diagram describes the execution of a DDR request involving the system residence volume (SYSRES). A swap for SYSRES can be initiated either by the operator or by the operating system. The system initiates a DDR request for SYSRES only if the error necessitating the swap is associated with SYS1.SVCLIB. Most of DDR Execution for SYSRES is controlled by IGFDDRSR. IGFDDRSR is entered by FINCH in IOS if the request is system-initiated; it is entered by IGFDDRMF if the request is operator-initiated.

		Module	Label
1	To eliminate any conflict between Device Allocation and DDR, these two functions are not permitted to operate concurrently. DDR takes precedence over Allocation. If Allocation is executing when DDR gains control, the task for which it is executing is set non-dispatchable. If Allocation is not executing, ALLOCECB is set to zero. (ALLOCECE is the ECB for Allocation.)	IGFDDRSR	IGF101 IGF102A
2	Control is gained by issuing SVC 15. This SVC 15 issues a no-op for the devices. If the DDR request is operator-initiated, both the FROM and the TO devices have SVC 15s issued for them. If the request is system-initiated, only the TO device requires the SVC. (DDR already has control of the FROM device.	IGFDDRSR	IGF102 IGF103 IGF104 IGF111 IGF108
	If SVC 15 is not successful or if no alternate SYSRES device was specified (and if SYSRES cannot be swapped to itself), the request is terminated and control is returned to the caller.		IGF204A IGF215 IGF303
3	The UCB information is swapped via SVC 15.	IGFDDRSR	IGF108
4	Message IGF501D is issued requesting that the operator swap the volumes. If the message cannot be issued, the DDR request is terminated.	IGFDDRSR	IGF114 IGF207 IGF204
5	A read is issued for the new SYSRES, and the label is verified. If the volume swap was correct, Allocation is re-activated (see note 1 above) and DDR is successfully terminated with a return to caller.)	IGFDDRSR	IGF300 IGF104 IGF302 IGF304
	If the wrong volume is mounted, message IGF511A is issued and volume verification is retried.		IGF307
	If a permanent error is encountered on the read attempt, DDR is terminated after the UCB information is swapped back. (For operator-initiated requests, message IGF507A is issued to give the operator the opportunity of moving SYSRES to another device.)		IGF300 IGF306 IGF308 IGF308B
6	Before IGFDDRSR terminates. ALLOCECB is posted and the task for which Allocation is executing is set dispatchable. (See note 1 above.)	IGFDDRSR	IGF302

NOTES:

- Messages are issued by IGFDDRSR via SVC 15.
- IGFDDR05 is a subroutine in IGFDDRSR that gains control from a disk ERP when an error is encountered while performing an SVC fetch. IGFDDR05 determines if DDR can handle the swap.
- The channel-end appendage for SVC 15 issued by IGFDDRSR is IGF400. IGF400 is in IGFDDRSR.

This section describes the DDR routines and the relationships between them.

Figure DDR-3 shows the relationship of the DDR routines to the DDR functions discussed in DDR Section 2. Figures DDR-4 and DDR-5 show the flow of control between the various routines during the execution of DDR. Figure DDR-4 depicts DDR processing of system-initiated requests. Figure DDR-5 depicts DDR processing of operator-initiated requests.

LOCATING DDR

For VS1

The following DDR modules reside in the nucleus: IORMSCOM, IGFDDRSR, IGFDDRMF, IGFVDDR2, and IGFVDDR3. These modules are permanently resident and are loaded during initial program load. The rest of the DDR modules are loaded into the SVC transient area as needed. IGC0008E - IGC0808E, which make up SVC 85, reside in SYS1.SVCLIB. IGE0660A and IGC2503D also reside in SYS1.SVCLIB.

Note: IGC2503D resides in SYS1.SVCLIB as part of SVC 34; its module name is IGC2503D.

For VS2

All DDR modules are permanently resident in the link pack area (SYS1.LPALIB), except those located in the nucleus - IORMSCOM, IGFDDRMV, IGFVDDR2, and IGFVDDR3.

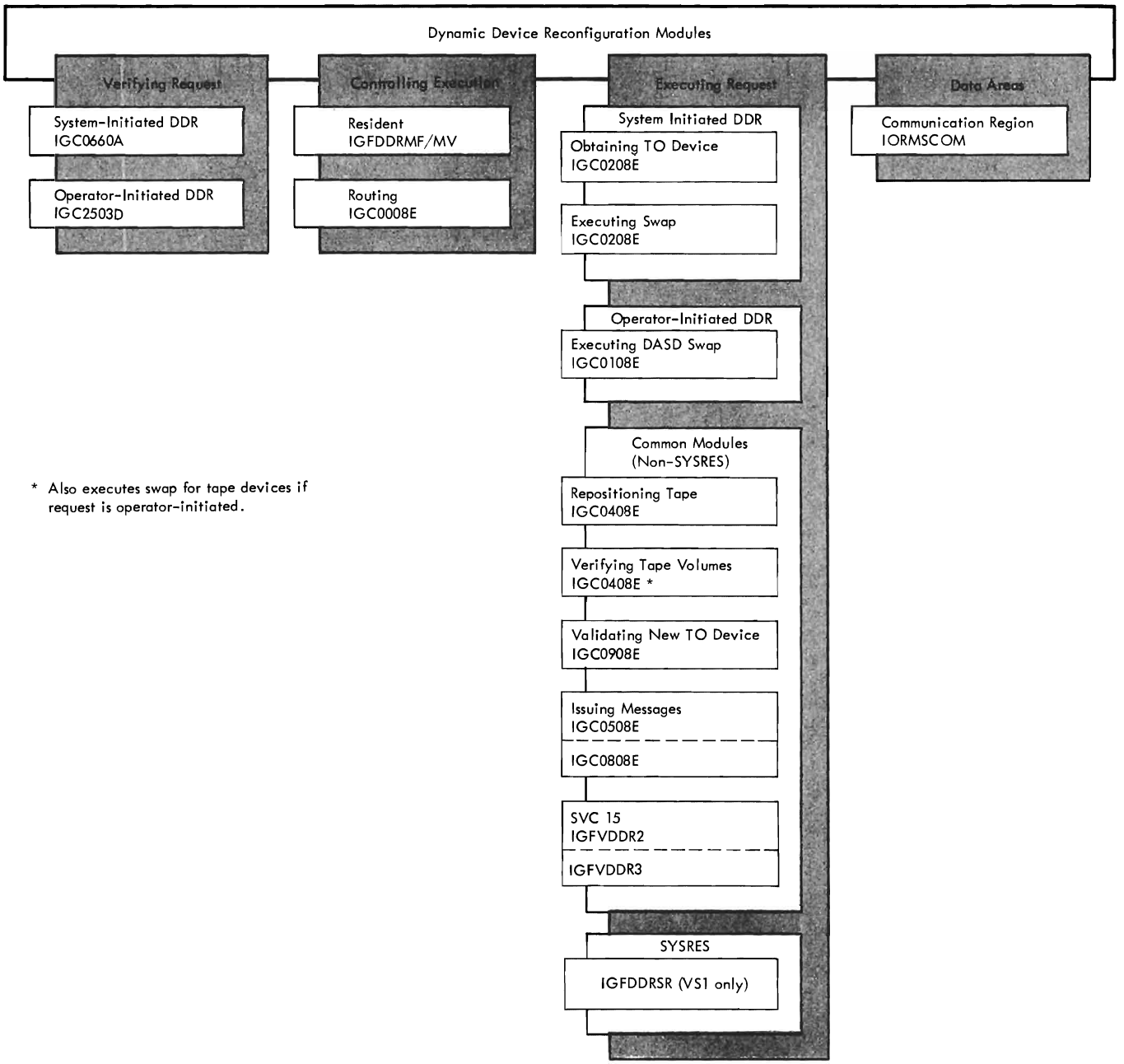


Figure DDR-3. Relationship of DDR functional areas to modules

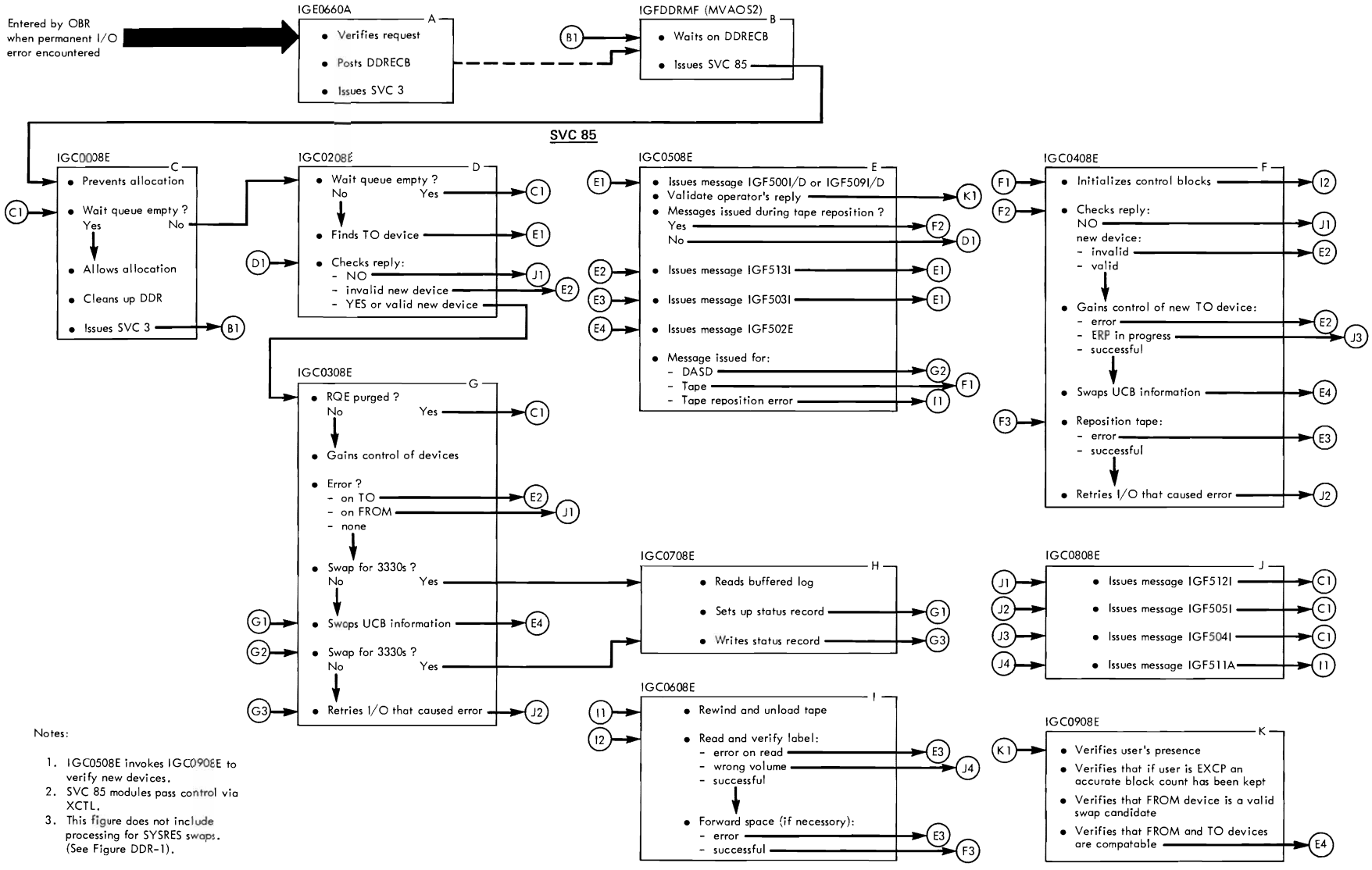
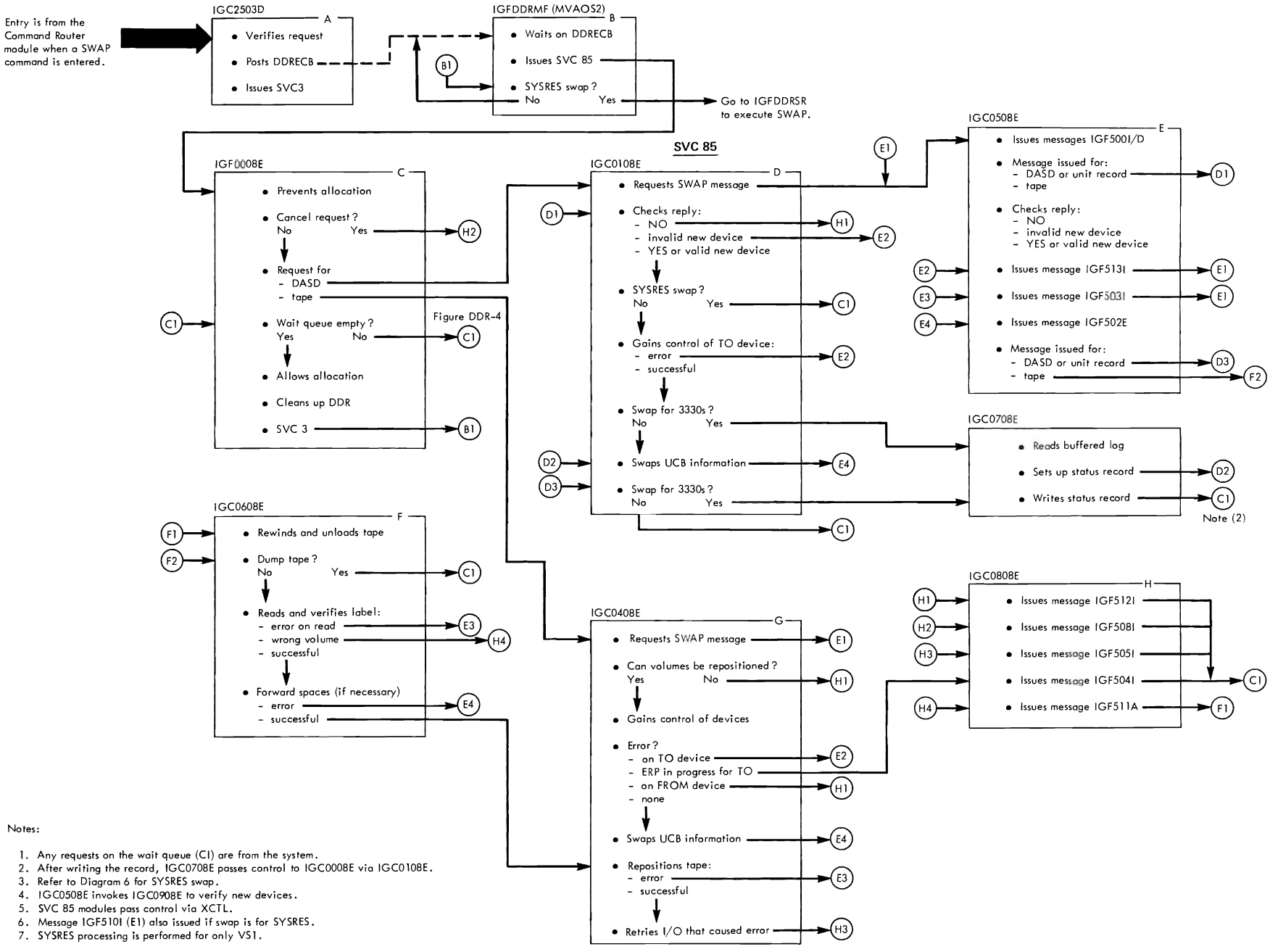


Figure DDR-4. Control flow during system-initiated swap



Notes:

1. Any requests on the wait queue (C1) are from the system.
2. After writing the record, IGC0708E passes control to IGC0008E via IGC0108E.
3. Refer to Diagram 6 for SYSRES swap.
4. IGC0508E invokes IGC0908E to verify new devices.
5. SVC 85 modules pass control via XCTL.
6. Message IGF510I (E1) also issued if swap is for SYSRES.
7. SYSRES processing is performed for only VS1.

Figure DDR-5. Control flow during operator-initiated swap

DDR MODULE DESCRIPTIONS

The following module descriptions summarize the functions of each DDR module. Additional information about DDR modules can be found in the method of operation diagrams and in the prologues for each module. Prologues can be found on the microfiche.

Note: Routines marked with * and listed under "Entered From" are routines that may enter via one of the DDR message routines. For example, one of the routines that enters IGC0008E is IGC0108E. Referring to routines that IGC0108E exits to, note that it passes control to IGC0008E via IGC0508E or IGC0808E.

IGC0008E - DDR Router		
Controls the execution of DDR: • Ensures that Device Allocation does not operate concurrently with DDR. • Routes control to the proper DDR routine to process the swap request. • Terminates DDR execution.		
Entry Point	Entered From	Exits To
IGC0008E	IGFDDRMF/MV (via SVC 85) IGC0108E* IGC0208E* IGC0308E* IGC0408E* IGC0508E IGC0608E IGC0808E	IGFDDRMF/MV (via SVC 3) when DDR is terminated. IGC0208E if DDR request is system-initiated. IGC0108E if DDR request is for direct access storage or unit-record devices and is operator-initiated. IGC0408E if DDR request is for tape devices and is operator-initiated. IGC0808E if the OPERATOR SWAP CANCELLED BY SYSTEM message is to be issued.

IGC0108E - Operator-Initiated DASD DDR		
Executes operator-initiated DDR requests for direct access storage and unit-record devices: • Processes reply to SWAP message. • Gains control of devices. • Swaps UCB information. • Invokes routine to issue PROCEED WITH SWAP message.		
Entry Point	**Entered From**	**Exits To**
IGC0108E	IGC0008E IGC0508E IGC0708E	IGFDDRMF/MV (via SVC 3) if the swap is for SYSRES. IGC0008E (via IGC0508E or IGC0808E) to terminate DDR execution. IGC0508E to issue messages. IGC0708E if 3330s are being swapped. IGC0808E to issue messages.

IGC0208E - System-Initiated DDR I		
• Determines TO device for a system-initiated request.		
Entry Point	**Entered From**	**Exits To**
IGC0208E	IGC0008E IGC0308E* IGC0508E	IGC0008E (via SVC 7) if no RQEs are on the DDR wait queue. IGC0008E (via IGC0808E) if the operator requests that DDR be terminated. IGC0308E if processing for the current request is to continue. IGC0508E to issue messages. IGC0808E if DDR is to be terminated.

IGC0308E - System-Initiated DDR II

Completes execution of system-initiated swaps:

- Gains control of devices for a system-initiated swap.
- Swaps UCB information.
- Invokes routine to issue PROCEED WITH SWAP message.

Entry Point	Entered From	Exits To
IGC0308E	IGC0208E IGC0508E IGC0708E	IGC0008E (via SVC 7) if no RQEs are on the DDR wait queue. IGC0008E (via IGC0808E) if DDR is to be terminated. IGC0208E (via IGC0508E) if a new TO device must be specified by the operator. IGC0408E (via IGC0508E) if operator is to exchange volumes. IGC0508E to issue messages. IGC0708E if 3330s are being swapped. IGC0808E if DDR is to be terminated.

IGC0408E - Operator-Initiated Tape DDR

Repositions tapes after swap and:

- For operator-initiated DDR requests involving tape devices, gains control of devices, swaps UCB information, and invokes routine to issue PROCEED WITH SWAP message.
- For system-initiated DDR requests involving tape devices, initializes system control blocks.

Entry Point	Entered From	Exits To
IGC0408E	IGC0008E IGC0308E* IGC0508E IGC0608E* NSLREPOS	IGC0008E (via IGC0808E) if DDR is to be terminated. IGC0508E to issue messages. IGC0608E to reposition the tape. IGC0808E if DDR is to be terminated.

IGC0508E - Message Writer I		
• Issues messages and exits to routine specified in DDRWORK + 12 (in IORMSCOM).		
Entry Point	Entered From	Exits To
IGC0508E	IGC0108E	IGC0008E after issuing PROCEED WITH SWAP message for other than 3330s. IGC0108E after issuing PROCEED WITH SWAP message for 3330s or after issuing SWAP message.
	IGC0208E	IGC0208E always.
	IGC0308E	IGC0208E after issuing INVALID FOR SWAP message. IGC0308E after issuing PROCEED WITH SWAP message for direct access storage or unit-record devices. IGC0408E after issuing PROCEED WITH SWAP for tape devices.
	IGC0408E	IGC0408E after issuing SWAP or ERROR messages. IGC0608E after issuing PROCEED WITH SWAP message.
	IGC0608E	IGC0408E after issuing ERROR message.
	IGC0908E	IGC0908E if a new TO device is specified by the operator.
Note: See DDR Section 6 for a list of messages.		

IGC0608E - Tape Label Verifier		
• Verifies tape labels after a swap has been completed.		
Entry Point	Entered From	Exits To
IGC0608E	IGC0408E* IGC0508E IGC0808E	IGC0008E if swap is for DDR dump tape. IGC0408E (via SVC 7) if reposition is successful. IGC0408E (via IGC0508E) if an I/O error is encountered. IGC0508E to issue ERROR message. IGC0808E if the wrong volume is mounted. NSLREPOS if the tape does not have standard labels.

IGC0708E - Recording

- Records 3330 buffer sense information on SYS1.IOGREC. (Records are handled via SVC 76.)

Entry Point	Entered From	Exits To
IGC0708E	IGC0108E IGC0308E	Caller always.

IGC0808E - Message Writer II

- Issues messages and exits to routine specified in DDRWORK + 12 (in IORMSCOM).

Entry Point	Entered From	Exits To
IGC0808E	IGC0008E IGC0108E IGC0208E IGC0308E IGC0408E IGC0608E	IGC0008E if entered by other than IGC0608E (to terminate execution). IGC0608E if entered by IGC0608E (to continue execution.)

IGC0908E - Tape Validation

- Validates operator's reply to a tape swap.

Entry Point	Entered From	Exits To
IGC0908E	IGC0508E	IGC0408E to continue tape processing. IGC0508E to issue the swap or invalid message. IGC0808E to issue terminate message.

IGE0660A - DDR Central

Sets up DDR for processing of system-initiated requests:

- Verifies system-initiated requests for DDR.
- Places failing RQE on DDR wait queue.
- Posts DDRECB.

Entry Point	Entered From	Exits To
IGE0660A	IGE0125F (OBR)	Caller always.

IGFDDRMF - DDR Resident (VS1)
IGFDDRMV - DDR Resident (VS2)

Initiates execution of DDR:

- Waits for DDR to be activated; that is, waits for DDRECB to be posted.
- Initiates the processing of a DDR request.

Note: If a program check occurs in a DDR routine, the DDR STAE routine is entered at IGFVSTAE from the system STAE. The DDR STAE routine is an integral part of IGFDDRMV; it is only present for VS2.

Entry Point	Entered From	Exits To
IORMSSVC	Dispatcher	IGC0008E (via SVC 85) IGFDDRSR if the swap request is for SYSRES (VS1).

IGFDDRSR - DDR SYSRES (VS1 only)

Handles all DDR functions for SYSRES:

- Determines if DDR should be invoked for an error involving SYSRES.
- Prohibits Device Allocation from operating while DDR is being executed.
- Determines the TO device if the swap is system-initiated.
- Gains control of the devices.
- Swaps UCB information.
- Issues message requesting operator to exchange devices.
- Verifies that correct volume was mounted on the TO device.

Entry Point	Entered From	Exits To
IGFDDRSR	FINCH IGFDDRMF	Caller always.
IGF400	IOS	IOS (via IOS-DDR vector table) always.
IGFDDR05	Disk ERP	Caller always.

Note: IGFDDR05 is entered whenever a permanent error is encountered in an I/O operation involving SYS1.SVCLIB on SYSRES. IGCDDR05 determines whether DDR SYSRES may be invoked.

IGF400 is the channel end appendage for SVC 15s that are issued by DDR SYSRES.

IGFDDRSR performs all other DDR functions related to a SYSRES swap.

IGFVDDR2 - Channel End Appendage		
Controls DDR I/O operations (SVC 15): • Determines what I/O operation is requested when a DDR routine issues SVC 15. • Takes the appropriate branch into IOS using the IOS-DDR vector table.		
Entry Point	Entered by	Exits To
IGFDDR02	IOS	IOS (via the vector table) always.
Note: The following DDR routines use this routine via SVC 15. IGC0008E, IGC0108E, IGC0208E, IGC0308E, IGC0408E, IGC0608E, and IGC0708E.		

IGFVDDR3 - Abnormal End Appendage		
• Processes error conditions encountered in I/O operations initiated by DDR.		
Entry Point	Entered From	Exits To
IGFDDR03	IOS	IOS via IOS-DDR vector table if permanent error is detected, otherwise via return to caller.

IGC2503D - SWAP Command Processor		
Sets up DDR for operator-requested swaps: • Verifies the SWAP command and determines whether DDR can be invoked for an operator-initiated request. • Determines whether operator-specified tape devices are eligible for swapping. • Posts DDRECB.		
Entry Point	Entered From	Exits To
IGC2503D	IGC0403D (SVC 34 command module)	Caller if DDR is to process the request. IGG2103D (IOS message module) if DDR is not to process the request.

The directory provides a quick reference from this publication to the pertinent coding, or from the coding back into the publication. The directory contains the following information:

Module and CSECT Name: The name of the object module and the CSECT containing the coding indicated by the label.

Label: Names taken from the listing for entry points, significant sections of coding, and data areas.

Description: A synopsis of the function of the coding indicated by the label.

Library: The place where the module resides. (For VS2, all DDR modules reside in the link pack area, except those located in the nucleus - IORMSCOM, IGFDDRMV, IGFVDDR2, and IGFVDDR3.)

Name: The descriptive name used to refer to the module.

Diagram: The method of operation diagram containing information about the coding identified by the label.

Module and CSECT Name	Label	Description	Library	Name	Diagram
IEANUC01 IGF201	IGF201	Contains addresses and status indicators	Nucleus	DDR Communications Area	
IGC0008E	IGC0008E	Controls DDR execution	SYS1.SVCLIB (VS1) SYS1.LPALIB (VS2)	DDR Router	4
IGC0108E	IGC0108E	Processes operator-initiated requests for direct access and unit-record devices	SYS1.SVCLIB (VS1) SYS1.LPALIB (VS2)	Operator-Initiated DASD DDR	4
IGC0208E	IGC0208E	Determines TO device for system-initated request	SYS1.SVCLIB (VS1) SYS1.LPALIB (VS2)	System-Initated DDR I	5
IGC0308E	IGC0308E	Executes swap for system-initiated request	SYS1.SVCLIB (VS1) SYS1.LPALIB (VS2)	System-Initiated DDR II	5
IGC0408E	IGC0408E	Processes operator-initiated requests for tape devices; repositions tapes	SYS1.SVCLIB (VS1) SYS1.LPALIB (VS2)	Operator-Initiated Tape DDR	5,6
IGC0508E	IGC0508E	Issues DDR messages	SYS1.SVCLIB (VS1) SYS1.LPALIB (VS2)	Message Writer I	5,6
IGC0608E	IGC0608E	Verifies that correct tapes are mounted	SYS1.SVCLIB (VS1) SYS1.LPALIB (VS2)	Tape Label Verifier	5,6
IGC0708E	IGC0708E	Records error statistics for 3330s	SYS1.SVCLIB (VS1) SYS1.LPALIB (VS2)	Recording	5,6
IGC0808E	IGC0808E	Issues DDR messages	SYS1.SVCLIB (VS1) SYS1.LPALIB (VS2)	Message Writer II	5,6
IGC0908E	IGC0908E	Validate operator's reply to a tape swap	SYS1.SVCLIB (VS1) SYS1.LPALIB (VS2)	Tape Validation Routine	
IGE0660A	IGE0660A	Verifies system-initiated requests	SYS1.SVCLIB (VS1) SYS1.LPALIB (VS2)	DDR Central	2
IGFDDRMF	DDRRB	RB for DDR	Nucleus	DDR Resident	4
	IGFDDRFC	Marks end of DDR save area			4
	IGFDDRMF	Initiates execution of DDR request (VS1)			4
	IGFDDRND	Marks end of DDR save area			4
	IGFDDR01	Alternate entry point name			
	IGFDDR11	Alternate entry point name			
	IORMSSVC	Issues SVC 85 (VS1)			4
	IORMSWT	Issues WAIT for DDRECB (VS1)			4

Module and CSECT Name	Label	Description	Library	Name	Diagram
IGFDDRMV	IGFDDRMV	Initiates processing of DDR request (VS2)	Nucleus	DDR Resident	4
	IGFVSTAE	Terminates DDR when a program check occurs			
	IORMSSVC	Issues SVC 85 (VS2)			4
	IORMSWT	Issues WAIT for DDRECB (VS2)			4
IGFDDRSR	IGFDDRSR	Handles DDR requests involving SYSRES	Nucleus	DDR SYSRES	7
	IGFDDR05	Determines if the system should invoke DDR for an error on SYSRES			
	IGF0DEB	DEB for DDR SYSRES			
	IGF0IOB	IOB for DDR SYSRES			
	IGF400	Determines proper interface between DDR SYSRES and IOS when an SVC 15 is issued by IGFDDRSR			
IGFVDDR2	IGFVDDR2	Determines proper interface between DDR and IOS when an SVC 15 is issued	Nucleus	Channel End Appendage	
IGFVDDR3	IGFVDDR3	Processes error conditions during DDR I/O	Nucleus	Abnormal End Appendage	
IGF2503D	IGF2503D	Verifies operator-initiated requests	SYS1.SVCLIB (VS1) SYS1.LPALIB (VS2)	SWAP Command Processor	3
IORMSCOM	DDRECB	ECB for DDR	Nucleus	DDR Communications Area	

This section contains descriptions of the principal data areas used by DDR:

- DDR communications area (IORMSCOM)
- IOS-DDR vector table

The symbolic names shown in individual fields represent the displacement, in bytes, from the beginning of a specific data area to the field. Access is gained to a specific field by using an instruction in which the beginning address of the area (usually contained in a register) is the base address, and the symbolic field name represents the displacement. Use of the fields can be traced in the DDR listings by first locating the symbolic field names in the cross-reference table at the back of the listings and then noting where the names are used.

The field headings and their meanings are:

Displacement: The numeric address of the field relative to the beginning of the data area. The first number is in decimal; the second (in parentheses) is the hexadecimal equivalent.

Bytes and Bit Patterns: The size (number of bytes) of the field and the bit settings of flag fields; that is, the state of bits in a byte. When the column is used to show the state of bits, it is shown as follows:

....	The eight bit positions (0-7) in a byte. For ease of scanning, the high-order (left-hand) four bits are separated from the low-order four bits.
1...	A reference to bit 0.
.... ..xx	Bits 6 and 7 are reserved.

Bit settings that are significant are shown and described. Bit settings that are not presently significant are described as reserved bits.

Field Name: A name that identifies the field.

Field Description: The use of the field.

DDR COMMUNICATIONS AREA (IORMSCOM)

IORMSCOM is the DDR communications area. It provides the major interface between the DDR routines, as well as interfaces between DDR and other parts of OS/VS.

The base address of IORMSCOM is contained in the first word of the SYS1.LOGREC DCB. In addition, the address can usually be found in one of the registers if DDR is being executed.

Note: The first 20 bytes (0-19) of IORMSCOM are used by CCH (Channel-Check Handler). For a description of this area, refer to CCH Section 5 of this publication.

Figure DDR-6 is provided to assist in locating DDR fields in IORMSCOM. This figure contains an alphabetic list of the fields with their displacements.

Field	Displacement	Field	Displacement	Field	Displacement
ALLOCECB	20 (14)	DDRFLGSB	40 (28)	IOSVTA	45 (2D)
ALLOCTCB	24 (18)	DDRFLGSC	44 (2C)	OURRQE	66 (42)
ALLOTJID	28 (1C)	DDRFLGSD	48 (30)	REPLYBUF	84 (54)
ALTCUASR	60 (3C)	DDRFLGSE	74 (4A)	RMSTCB	40 (28)
BLKCNT	68 (44)	DDRFLGSF	75 (4B)	RMSTCBA	41 (29)
BRET	260 (104)	DDRFMUCB	56 (38)	SYSRESAD	36 (24)
CNSLID	52 (34)	DDRIOB	152 (98)	USRDCB	53 (35)
COREADDR	88 (58)	DDRTOUCB	58 (3A)	USRDEB	32 (20)
DDRBUFR	280 (118)	DDRWORK	104 (68)	USRRQE	64 (40)
DDRCOUNT	92 (5C)	FILECNT	72 (48)	USRTCB	48 (30)
DDRDEB	212 (D4)	IOSECB	80 (50)	USRTCBA	49 (31)
DDRECB	76 (4C)	IOSVT	44 (2C)	XCTLLIST	136 (88)
DDRFLGSA	36 (24)				

Figure DDR-6. Locations of fields in IORMSCOM

Displacement	Bytes and Bit Patterns	Field Name	Field Description
20 (14)	4	ALLOCECB	ECB for Device Allocation. Used to keep Device Allocation from being executed currently with DDR.
24 (18)	4	ALLOCTCB	If Device Allocation is active, this field contains the address of the TCB for the task for which Device Allocation is being executed.
28 (1C)	2	ALLOTJID	OS/VS2 - TJID of the task for which device allocation is being executed.
30 (1E)	2		Reserved.
32 (20)	4	USRDEB	Address of the DEB for the data set being swapped.
36 (24)	0	SYSRESAD	(See displacement 37.)
36 (24)	1	DDRFLGSA	DDR flags.
	1...	SYSRESOP	This version of DDR can process swap requests for SYSRES (VS1 only).
	..1.	DDRBUSY	DDR request being processed.
	...1	DDRON	System-initiated requests permitted.
	.... 1...	VALIDATE	Verify the TO device.
	.... .1..	NSLUSER	The volume being processed does not have standard labels.
	.... ..1.	NSLAVAIL	An NSLREPOS routine has been provided.
	1	OPINIT	The request being processed is operator-initiated.
	.x..		Reserved.
37 (25)	3		Address of IGFDDRSR (VS1 only).
40 (28)	0	RMSTCB	(See RMSTCBA, displacement 41).
40 (28)	1	DDRFLGSB	DDR flags.
	1...	SWAPMSG	Issue SWAP message.
	.1..	INVALID	Issue INVALID FOR SWAP message.
	..1.	SYSRES	Issue SYSRES RESIDENT ON warning message (VS1 only).
	...1	DDREND	Issue TAPE DDR TERMINATED message.
	.... 1...	PROCEED	Issue PROCEED WITH SWAP message.
	.... .1..	SHRDASD	The device involved in the swap is a shared direct access device.
	.... ..1.	NOTOUCB	A TC device was not found for a system-initiated swap request.
	1	FIRSTIO	SVC 15 has been issued to gain control of a device.
41 (29)	3	RMSTCBA	Address of TCB for DDR.
44 (2C)	0	IOSVT	(See IOSVTA, displacement 45.)

Displacement	Bytes and Bit Patterns	Field Name	Field Description
44 (2C)	1	DDRFLGSC	DDR flags.
	1...	TERMINATE	Terminate DDR.
	.1..	STAE	The DDR STAE routine has intercepted a program check (VS2 only).
	..1.	CANCEL	Issue OPERATOR SWAP CANCELLED BY SYSTEM message.
	...1	ERROR	Issue ERROR message.
	.... 1...	SWAPUCB	SVC 15 has been issued to swap UCB information.
	.... .1..	REPOSITN	A tape that has been swapped must be repositioned.
	.... ..1.	IONOCTRL	SVC 15 has been issued to re-EXCP without control.
	1	ERPINPRO	Issue AN ERP IS IN PROCESS message.
45 (2D)	3	IOSVTA	Address of IOS-DDR vector table.
48 (30)	0	USRTCB	(See USRTCBA, displacement 49.)
48 (30)	1	DDRFLGSD	DDR flags.
	1...	SYSRESBR	Branch to IGFDDRSR. An operator-initiated swap specified SYSRES as the FROM device (VS1 only).
	.1..	DDRMSG	Issue WRONG VOLUME MOUNTED message if this bit is off, and bit 0 of DDRFLGSD is on.
	..1.	DUMPTAPE	The tape device involved in the current DDR operation has a dump tape mounted.
	...1	READ	Read a record.
	.... 1...	STADLABEI	The tape volume being processed has standard labels.
	.... .1..	UNLABELD	The tape volume being processed is unlabeled.
	.... ..1.	FORSPACE	Forward space file.
	1	IOCONTRL	SVC 15 has been issued to re-EXCP with control.
49 (31)	3	USRTCBA	Address of the TCB for the task for which DDR is being executed.
52 (34)	1	CNSLID	Save area for Console Identification; used if operating under MCS.
53 (35)	3	USRDCB	Address of the DCB for the data set involved in a swap.
56 (38)	2	DDRFMUCB	Address of the UCB for the FROM device.
58 (3A)	2	DDRTOUCB	Address of the UCB for the TO device.
60 (3C)	4	ALTCUASR	Address of alternate SYSRES device (VS1 only).
64 (40)	2	USRRQE	Pointer to RQE for I/O operation for which DDR is being executed. The DDR request was initiated by the system.
66 (42)	2	OURRQE	Pointer to RQE for DDR.
68 (44)	4	BLKCNT	Block count for magnetic tape.
72 (48)	2	FILECNT	File count for magnetic tape.

Displacement	Bytes and Bit Pattern	Field Name	Field Description
74 (4A)	1	DDRFLGSE	DDR flags.
	1...	RETRY	Retry the I/O operation that caused the system to invoke DDR.
	.1..	WRONGVOL	Issue WRONG VOLUME MOUNTED message.
	..1.	LABLPROC	Verify that the correct volume (tape) has been mounted.
	...1	READFROM	Read 3330, 3330-1, or 3340 buffer.
	.... 1...	FMERROR	Error was encountered reading FROM 3330, 3330-1, or 3340 buffer.
	.... .1..	TOERROR	Error was encountered reading TO 3330, 3330-1, or 3340 buffer.
	.... ..1.	RECRDING	Record 3330, 3330-1, or 3340 buffer.
	1	READTO	SVC 15 was issued to record a 3330, 3330-1, or 3340 buffer.
75 (4B)	1	DDRFLGSF	DDR flags.
	1...	COMPLETE	Issue SWAP COMPLETE message.
	.1..	REENTRY	The DDR abnormal end appendage was entered. Used by IGC0508E, when calling IGC0908E, to validate a device.
	..1.	REINIT	Reinitialize DDR block count.
	...1	LABELERR	An error was encountered while attempting to verify and reposition a tape after a swap.
	.... 1...	DDRINIT	Initialize DDR.
	.... .1..	OPCANCEL	Cancel operator-initiated swap.
	.... ..1.	NEGCOUNT	Initialize block count to -1.
	1	DDRFIX	User LCB page has been fixed.
76 (4C)	4	DDRECB	ECB for DDR Resident routine (IGFDDRMV).
80 (50)	4	IOSECB	ECB for return from IOS. Used to determine when an SVC 15 has been completed.
84 (54)	4	REPLYBUF	Reply buffer for DDR messages.
88 (58)	4	COREADDR	Pointer to main storage obtained for recording 3330, 3330-1, or 3340 sense buffer information.
92 (5C)	4	DDRCOUNT	Reserved.
96 (60)	2		Contains CNOP 4,8 instruction.
98 (62)	2		Reserved.
100 (64)	4		Block count.
104 (68)	28	DDRWORK	Work area. DDRWORK + 12 is used by the DDR load modules of SVC 85 to indicate successor routines.
136 (88)	16	XCTLLIST	XCTL list for NSLREPOS routine.
152 (98)	60	DDRIOB	IOB used by DDR.
212 (D4)	48	DDRDEB	DEB used by DDR.
260 (104)	4	BRET	BR 14 instruction.
264 (108)	4	FIXECB	ECB for user's fixed page DCB.
272 (110)	8	FIXLIST	Parameter list for page fixed/freed.
280 (118)	80	DDRBUFR	Buffer for DDR I/O.

IOS-DDR VECTOR TABLE

DDR uses the IOS-DDR vector table to perform I/O operations. The vector table is in IOS. See "I/O Handling -- IOS-DDR Interface" in DDR Section 2 for details on how the table is used. The following is a description of the vector table:

Displacement	Meaning
0	Address of DDR wait queue.
4	Branch to re-EXCP with control of device. Turns on the UCB error flag (UCBERR in UCBFL1), and stores a pointer to DDR's RQE in UCBLTS. The RQE is then queued on the logical channel queue, and Channel Restart is entered.
8	Branch to re-EXCP without control of the device. Resets the UCB not-ready flag (UCBNRY in UCBFL1), queues the RQE on the logical channel queue, and enters Channel Restart.
12	Branch to POST without control of device. Resets the UCB error flag, enters IOS-POST interface to post the user's request, and then enters Channel Restart.
16	Branch to POST with control of device. Turns on the UCB error flag. The IOS-POST interface is entered to post the user's request, then Channel Restart is entered.
20	Branch to retry user's request. Dequeues the user's RQE from the DDR wait queue and queues it on the appropriate logical channel queue. The UCB error flag is then turned on, and a pointer to the user's RQE is stored in UCBLTS. Pointers to the IOB, DEB, and DCB are set up. The DCB error flags are reset, and control is routed to re-EXCP the user's I/O request.
24	Branch to swap UCB information. Exchanges device information between the TO and the FROM devices, but leaves volume information alone. Channel Restart is then entered.
28	Branch to cancel DDR. Dequeues the user's RQE from the DDR wait queue, sets the UCB error flag, and posts the user with a permanent I/O error. (If the RQE is not on the wait queue upon entry, IOS exits to Channel Restart.)
32*	Branch to cancel DDR SYSRES. (This branch is taken only when an operator-initiated swap for SYSRES is interrupted by a system-initiated swap request for SYSRES.) Dequeues the DDR SYSRES RQE from the logical channel queue and enters Channel Restart.
36*	Branch to re-EXCP for SYSRES DDR on condition code 3. (This branch is taken only when DDR is being executed for SYSRES.) Prevents a reseek on the I/O operation by setting the UCBASK bit in UCBFL1 of the UCB (except for devices with rotational position sensing). Then uses re-EXCP to perform the I/O request.

*These entries exist only for VS1.

This section contains information to aid in locating errors in DDR. Discussions on register usage, messages, and SYSRES processing are included.

REGISTER USAGE

Figure DDR-7 indicates the significant contents of general purpose registers upon entry to the specified DDR modules.

MESSAGES

DDR messages can be issued by three different modules:

- IGFDDRSR if a swap for SYSRES is being executed (VS1 only).
- IGC0508E or IGC0808E if a swap is not for SYSRES.

Figure DDR-8 shows the messages, the modules that issue them, and the modules that request each message to be issued.

IGFDDRSR PROCESSING (VS1 ONLY)

The DDR SYSRES routine (IGFDDRSR) is segmented into a number of subroutines. Entry to these subroutines is usually via a BAL instruction, while exit from them is usually via a branch to the address specified by the BAL instruction. (The exit address specified by the BAL instruction is usually contained in LINKRG, register 8.)

Figures DDR-9 and DDR-10 are provided as an aid to understanding the relationship between the subroutines and the rest of the module. These figures indicate the events that occur in IGFDDRSR during swapping of SYSRES.

The figures are alike. The action column is a summary of the operations performed in IGFDDRSR. The sequence number column indicates the sequence in which these operations are performed. The label column specifies where in IGFDDRSR these operations are performed.

First, select the appropriate figure; Figure DDR-9 pertains to system-initiated swaps for SYSRES; Figure DDR-10 pertains to operator-initiated swaps for SYSRES. Next find number 1 in the sequence column of the selected figure. Then use the sequence numbers, starting with 1, to trace the sequence of events. (Figures DDR-9 and DDR-10 show the sequence of events for a successful SYSRES swap only; error paths are not shown.)

Note: SYSFLG1 is used by IGFDDRSR to indicate what processing is to be done. SYSFLG1 is in IGFDDRSR, and it is set and reset by that routine throughout DDR SYSRES processing.

	Register Input Expected by Each DDR Module												
Register No.	IGC0008E	IGC0108E	IGC0208E	IGC0308E	IGC0408E	IGC0508E	IGC0608E	IGC0708E	IGC0808E	IGC0908E	IGE0660A	IGFVDDR2	IGFVDDR3
0													
1	DDR Wait Queue Address	Address of RQE for DDR	Address of RQE for DDR	Address of RQE for DDR	Address of RQE for DDR	Address of RQE for DDR	Address of RQE for DDR	Address of RQE for DDR	Address of RQE for DDR	Address of RQE for DDR	Address of RQE for Failing I/O Operation	Address of RQE for DDR	Address of RQE for DDR
2		DDR Wait Queue Address	DDR Wait Queue Address	DDR Wait Queue Address	DDR Wait Queue Address	DDR Wait Queue Address	DDR Wait Queue Address	DDR Wait Queue Address	DDR Wait Queue Address	DDR Wait Queue Address		Address of IOB for DDR	Address of IOB for DDR
3	CVT Address	CVT Address	CVT Address	CVT Address	CVT Address	CVT Address	CVT Address	CVT Address	CVT Address	CVT Address		Address of DEB for DDR	Address of DEB for DDR
4	Address of TCB for RMS	Address of TCB for DDR	Address of TCB for DDR	Address of TCB for DDR	Address of TCB for DDR	Address of TCB for DDR	Address of TCB for DDR	Address of TCB for DDR	Address of TCB for DDR	Address of DCB for DDR		Address of DCB for DDR	Address of DCB for DDR
5	Address of SVRB Extended Save Area or of SVRB	Address of SVRB Extended Save Area	Address of SVRB Extended Save Area	Address of SVRB Extended Save Area	Address of SVRB Extended Save Area	Address of SVRB Extended Save Area	Address of SVRB Extended Save Area	Address of SVRB Extended Save Area	Address of SVRB Extended Save Area	Address of SVRB Extended Save Area			
6			Address of DEB for Failing I/O Operation	Address of DEB for Failing I/O Operation		Address of DEB for Failing I/O Operation if System-Initiated Request		Address of DEB for Failing I/O Operation if System-Initiated Request	Address of DEB for Failing I/O Operation if System-Initiated Request	Address of DEB for Failing I/O Operation			
7	Address of FROM UCB	Address of FROM UCB	Address of FROM UCB	Address of FROM UCB	Address of FROM UCB	Address of FROM UCB	Address of FROM UCB	Address of FROM UCB	Address of FROM UCB	Address of FROM UCB		Address of UCB for DDR	Address of UCB for DDR
8	Address of TO UCB if Operator-Initiated Request	Address of TO UCB			Address of TO UCB	Address of TO UCB if Operator-Initiated Request	Address of TO UCB	Address of TO UCB if Operator-Initiated Request	Address of TO UCB if Operator-Initiated Request	Address of TO UCB			
9							Address of DCB for Failing I/O Operation			Address of UCB for Failing I/O Operation			
10	Address of DEB for DDR	Address of DEB for DDR	Address of DEB for DDR	Address of DEB for DDR	Address of DEB for DDR	Address of DEB for DDR	Address of DEB for DDR	Address of DEB for DDR	Address of DEB for DDR	Address of DEB for DDR			
11	Address of IORMSCOM	Address of IORMSCOM	Address of IORMSCOM	Address of IORMSCOM	Address of IORMSCOM	Address of IORMSCOM	Address of IORMSCOM	Address of IORMSCOM	Address of IORMSCOM	Address of IORMSCOM			
12	Address of IGC0008E	Base Address	Base Address	Base Address	Base Address	Base Address	Base Address	Base Address	Base Address	Base Address	Address of IGE0660A		
13		Address of IOB for DDR	Address of IOB for DDR	Address of IOB for DDR	Address of IOB for DDR	Address of IOB for DDR	Address of IOB for DDR	Address of IOB for DDR	Address of IOB for DDR	Address of IOB for DDR			
14													
15												Base Address	Base Address

* If entered at IGFDDR05: 2 - Address of IOB for SYSRES
3 - Address of DEB for SYSRES
7 - Address of UCB for SYSRES
12 - Address of IGFDDR05
13 - Return Address

If entered at IGF400, contents are the same as for IGFVDDR2.

Figure DDR-7. Register input expected by the modules of DDR

Message Number	Issued By	Requested By
IGF500I SWAP xxx TO yyy IGF500D REPLY 'YES', DEVC, OR 'NO' REPLY 'YES', OR 'NO'	IGC0508E	IGC0108E IGC0208E IGC0308E* IGC0408E
IGF501D SWAP SYSRES FROM xxx TO yyy	IGFDDRSR	IGFDDRSR
IGF502E PROCEED WITH SWAP xxx TO yyy	IGC0508E	IGC0108E IGC0308E IGC0408E
IGF503I ERROR ON yyy, SELECT NEW DEVICE	IGC0508E	IGC0408E IGC0608E
IGF504I AN ERP IS IN PROCESS FOR xxx	IGC0808E	IGC0408E
IGF505I SWAP FROM xxx TO yyy COMPLETE	IGC0808E	IGC0408E
IGF507A VOLUME ON yyy UNIDENTIFIABLE, SWAP SYSRES TO zzz	IGFDDRSR	IGFDDRSR
IGF508I OPERATOR SWAP CANCELLED BY SYSTEM	IGC0808E	IGC0008E
IGF509I SWAP xxx IGF509D REPLY DEVC, OR 'NO'	IGC0508E	IGC0208E
IGF509D REPLY DEVC, OR 'NO'	IGC0508E	IGC0408E IGC0608E
IGF510I SYSRES RESIDENT ON xxx	IGC0508E	IGC0108E
IGF511A WRONG VOLUME MOUNTED ON yyy	IGC0808E	IGC0608E
	IGFDDRSR	IGFDDRSR
IGF512I DDR TERMINATED	IGC0808E	IGC0108E IGC0208E IGC0308E IGC0408E
IGF513I yyy INVALID FOR SWAP**	IGC0508E	IGC0108E IGC0208E IGC0308E IGC0408E

*Messages IGF500I/D are issued for IGC0308E only if the request is for message IGF513I.

**Messages IGF500I/D are always issued following a request for message IGF513I.

Figure DDR-8. DDR messages

Label	Sequence Number					Action
IGF101	1					Disable Allocation routine
IGF111A	2					Find alternate SYSRES device
IGF103	3					Prepare to gain control of device
IGF104	4		16		25	Prepare for I/O
IGF105A		10				
			17			Branch LINKRG
IGF105B	5	11			26	Issue SVC 15
		12				Branch LINKRG
IGF106	6			20	27	Issue WAIT
IGF106B				21	28	Branch LINKRG
IGF108	7					Set LINKRG1=IGF114
IGF109	8					Prepare to swap UCB information
IGF110	9					BAL LINKRG (to IGF105A)
		13				Branch LINKRG1 (to IGF114)
IGF114		14				Prepare to issue message IGF501D
IGF207		15				BALR LINKRG (to IGF104)
			18			Issue EXCP to write message
			19			BAL LINKRG (to IGF106)
				22		Check for I/O errors
IGF300				23		Prepare to read volume ID
				24		BAL LINKRG (to IGF104)
					29	Check for I/O errors
					30	Ensure correct volumes swapped
IGF215A					31	Clean up DDR
IGF301A IGF302					32	Reenable Allocation routine
IGF303					33	Return to FINCH

Figure DDR-9. Execution sequence in IGFDDRSR for system-initiated swap of SYSRES

Label	Sequence Number						Action
IGF101	1						Disable Device Allocation routine
IGF102	2						Set LINKRG=IGF107
	3						Prepare to gain control of FROM UCB
IGF103	4	12					Prepare for I/O
IGF104				27		36	
IGF105	5	13					
IGF105A			21				
				28			Branch LINKRG
IGF105B	6	14	22			37	Issue SVC 15
			23				Branch LINKRG
IGF106	7	15			31	38	Issue WAIT
	8	16			32	39	Branch LINKRG
IGF107	9						Check for I/O errors
	10						Prepare to gain control of TO UCB
IGF108	11						BAL LINKRG (to IGF103)
		17					Check whether I/O complete
IGF109		18					Set LINKRG1=IGF114
		19					Prepare to swap UCB information
IGF110		20					BAL LINKRG (to IGF105A)
			24				Branch on LINKRG1 (to IGF114)
IGF114			25				Prepare to issue message IGF501D
IGF207			26				BAL LINKRG (to IGF104)
				29			Issue EXCP to write message
				30			BAL LINKRG (to IGF106)
					33		Check for I/O errors
IGF300					34		Prepare to read volume ID
					35		BAL LINKRG (to IGF104)
						40	Check for I/O errors
						41	Ensure correct volumes swapped
IGF215A						42	Clean up DDR
IGF301A IGF302						43	Reenable Allocation routine
IGF303						44	Return to IGFDDRMF

Figure DDR-10. Execution sequence in IGFDDRSR for operator-initiated swap of SYSRES

APPENDIX A: MIC (MISSING INTERRUPTION CHECKER)

WHAT MIC IS

MIC (Missing Interruption Checker) checks to see that expected I/O interruptions occur within a specified time period. If the interruption does not occur, the operator is notified so that action can be taken to avoid a wait that eventually involves the system. MIC is not a cure for missing interruptions, only a way of decreasing their impact on the entire system.

HOW MIC FUNCTIONS

For VS1, MIC (IGFTMCHK) is initiated by the operator with a START command and runs in a virtual partition. In VS2, MIC is a subtask of the Master Scheduler. It polls active I/O devices to assure that device and channel ends, or mount requests are fulfilled within a specified time period. This time period can be altered by reassembling the IGFINTVL CSECT with the desired interval, or it may be allowed to default to three minutes. If an I/O operation does not complete in the specified time period, a message is sent to the operator.

In VS2, IGFTMCHK is pageable in the Master Scheduler and is automatically invoked after the Master Scheduler is initialized. Diagram MIC-1 shows how MIC performs its function.

MIC'S TABLE USAGE

MIC uses the CVTILK2 field of the CVT (Control Vector Table), and the following UCB (Unit Control Block) fields:

FIELD	BITS
UCBFL5	UCBTICBT
	UCBNALOC
	UCBALTPH
UCBNAME	
UCBTBYT3	UCB3COMM
	UCB3TAPE
	UCB3DACC
UCBSTAT	UCBONLI

The UCB fields are referenced through the IOSGEN macro instruction to check device and channel ends, and mount conditions. IOSGEN is also used to set bit UCBTICBT.

MESSAGES AND CODES

If IGFTMCHK encounters a missing interruption the following error message is issued:

```
          ┌ MOUNT                 ┐
IGF991E   │ DEVICE AND            │  PENDING FOR
          │ CHANNEL END           │   DEVICE ddd
          │ CHANNEL END           │
          └ DEVICE END            ┘
```

ddd=channel, unit, or address for the device for which the condition is pending.

Operator Action:

MOUNT
ready the named unit. If unit is ready, issue the VARY ONLINE command to generate a pseudo device end so that mount processing can continue.

DEVICE AND CHANNEL END or CHANNEL END
a hardware malfunction has occurred. Terminate the affected job. The device named and the channel in use should receive maintenance.

DEVICE END
examine the named unit for hardware malfunctions such as a hang condition with the select light on and no tape motion, or a disk with the select lock on. Check control or switching units for proper connection. If the device was just rewound or mounted, issue a VARY ONLINE command to generate a pseudo device end. A device end at any other time is inadvisable, since the integrity of the file may be in doubt if the interruption is simulated.

The following multiple console support (MCS) routing codes are used:

tape - 1 and 3
disk - 1 and 4
unit record - 1 and 7

In all cases a descriptor code of 3 (eventual action) is issued.

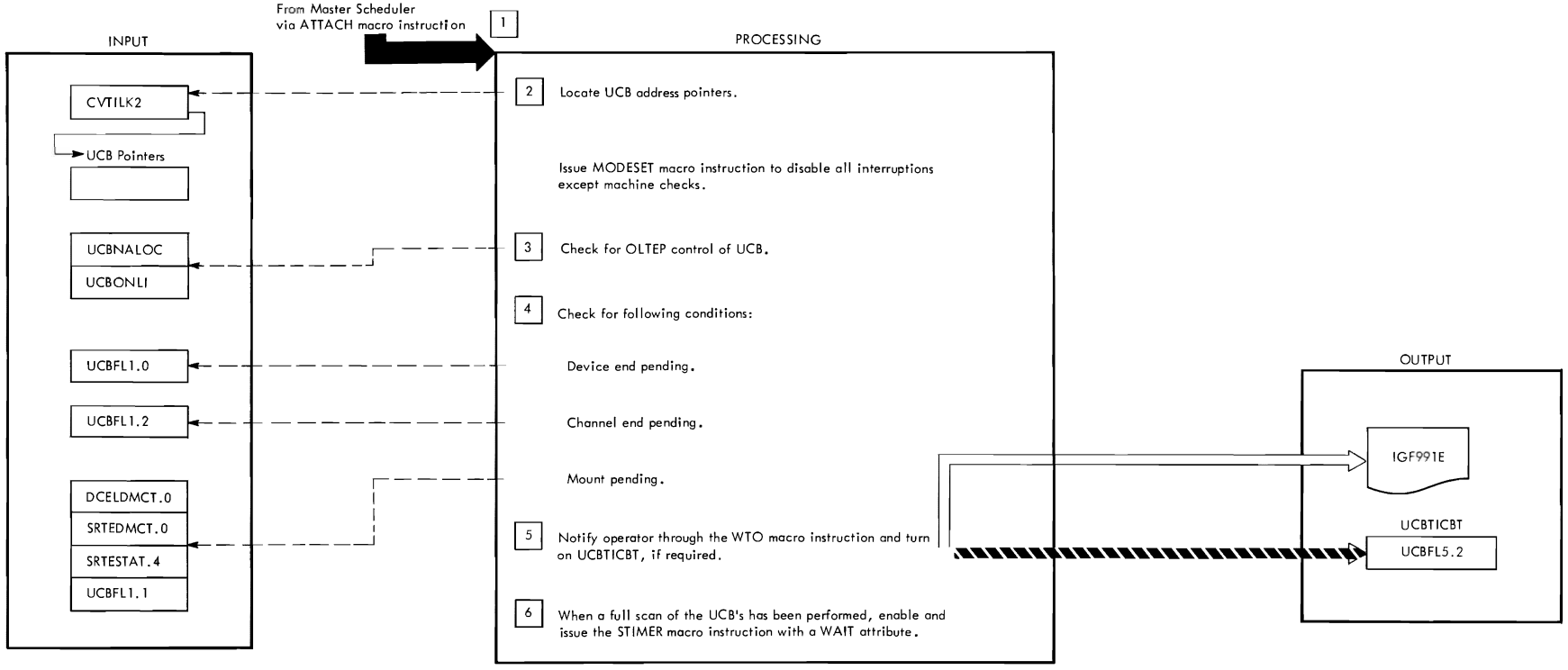
HOW TO VARY THE TIME INTERVAL

In large installations the three minute time interval may not be long enough to allow device mounts. This interval can be changed by specifying the desired interval through the following procedure.

```
//REPLACE   JOB   MSGLEVEL=1
//ASMLK     EXEC  ASMFCL,PARM.LKED='REF,
//          LET,LIST,NCAL,RENT'
//ASM.SYSIN DD
IGFINTVL    CSECT
            DC    CL8'00TT0000'
            END
/*
//LKED.SYSLMOD  DD   DSN=SYS1.LINKLIB,
//              DISP=OLD
//LKED.SYSIN    DD   *
                INCLUDE SYSLMOD(IGFTMCHK)
                ENTRY IGFTMCHK
                NAME IGFTMCHK(R)
/*
```

'TT', in the DC statement, is used to specify the desired time interval in minutes. A 0, or non-numeric characters cause the default value of three to be used.

Diagram MIC—1. Missing Interruption Checker



1 The Master Scheduler sets the time-of-day clock and enters IGFTMCHK through the ATTACH macro instruction. IGFTMCHK is given the following attributes: Key 0, Supervisor State, Enabled, Jobstep, without Shared Subpool 0. Its dispatching priority is just below that of the Master Scheduler.

2 Before processing continues, MIC checks to see that the located UCB is not for a teleprocessing device. If it is, another UCB is selected.

3 UCBNALOC is checked to see if OLTEP is in control and UCBONLI is checked to see if the device represented by the UCB is online (OLTEP uses offline devices). If OLTEP is in control, another UCB is selected.

4 The following matrix shows the possible conditions and actions taken for online, non-TP devices.

ACTION TAKEN	A	A	B	B
non-TP device	X	X	X	X
device or channel end pending	X		X	
mount pending		X		X
UCBTICBT on	X	X		

Action	Description
A	Notify operator of a missing device end.
B	Turn on UCBTICBT

5 The only message issued by IGFMCHK is IGF991E. There are two cases when the operator is not notified: when an offline device is involved (to avoid intersection with OLTEP), and when a teleprocessing device is involved (any measures the operator can take are not applicable to teleprocessing devices). UCBTICBT (UCBFL5, bit 2) indicates that channel or device end, or a mount condition are pending. The IOSGEN macro instruction is used to set this bit.

6 The default time interval for the STIMER macro instruction is three minutes. When the timer interval expires, the UCB scan is resumed.

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

OS/VS RMS Logic (File No. S360/S370-37) Printed in U.S.A. SY27-7239-2

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)

OS/VS
Recovery Management Support
Logic

READER'S COMMENT FORM

SY27-7239-2

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?

- ☐ As an introduction
- ☐ As a reference manual
- ☐ For another purpose (explain)_____
- ☐ As a text (student)
- ☐ As a text (instructor)

_____

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

IBM

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

OS/VS RMS Logic (File No. S360/S370-37) Printed in U.S.A. SY27-7239-2