

Program Logic

IBM System/360 Operating System Input/Output Supervisor

Program Number 360S-CI-505

This publication describes the operation of the I/O supervisor within the IBM System/360 Operating System control program. The I/O supervisor's components, the EXCP supervisor and the I/O interruption supervisor, are discussed in detail to show the internal structure and logic involved in the control of I/O devices and channels.

Program Logic Manuals are intended for use by IBM customer engineers involved in program maintenance and by system programmers involved in altering the program design. Program logic information is not necessary for program operation and use; therefore, distribution of this manual is limited to persons with program maintenance or modification responsibilities.

Restricted Distribution

PREFACE

This publication discusses the operation of the I/O supervisor. It is directed primarily to IBM customer engineers and to IBM system programmers.

The publication is divided into an introduction and five parts to provide both general and detailed presentations of the following areas:

- EXCP supervisor.
- I/O interruption supervisor.
- Error routines.
- SVC transient area routines.
- Control blocks, tables, and queues.

All options that may be added to the primary control program are included in the presentations. The control program options are explained in the publication IBM System/360 Operating System: Storage Estimates, Form C28-6551.

RECOMMENDED PUBLICATIONS

Information required for an understanding of this manual is contained in the following publications:

IBM System/360: Principles of Operation, Form A22-6821

IBM System/360 Operating System: Introduction to Control Program Logic, Program Logic Manual, Form Y28-6605

In addition, the following publication may be found convenient for reference:

IBM System/360 Operating System: System Programmer's Guide, Form C28-6550

Second Edition (April 1967)

This edition, Form Y28-6616-1, is a reprint of Form Y28-6616-0 incorporating TNLS Y28-2155, dated 6/1/66, Y28-2169, dated 8/31/66, dated 2/27/67. Note also that the prefix is changed from Z28- to Y28.

Specifications contained herein are subject to change from time to time. Any such change will be reported in subsequent revisions or Technical Newsletters.

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 publications can be obtained through IBM Branch Offices.

A form for readers' comments appears at the back of this publication. It may be mailed directly to IBM. Address any additional comments concerning this publication to the IBM Corporation, Programming Systems Publications, Department D58, PO Box 390, Poughkeepsie, N. Y. 12602

INTRODUCTION	7	PART II: I/O INTERRUPTION SUPERVISOR.	28
The Input/Output Supervisor.	8	General Operating Procedure.	28
EXCP Supervisor	10	Operation of the Subroutines and	
I/O Interruption Supervisor	11	Modules	29
The Error Routines	11	UCB Lookup Routine.	29
The Purge and Restore Routines	12	I/O Interruption Analysis	29
The Control Blocks, Tables, and Queues	12	Channel Data Check, Channel	
Control Blocks.	12	Control Check, Interface	
Input/Output Block	13	Control Check Bits.	29
Data Control Block	13	Control Unit End Bit	30
Data Extent Block.	13	PCI Bit and PCI Appendage.	30
Event Control Block.	13	Channel End Bit and Channel End	
Unit Control Block	13	Appendage	30
Tables.	14	Device End Bit	30
UCB Lookup Table	14	Attention Bit.	31
Channel Table.	14	Unit Check Bit and Abnormal End	
Device Table	15	Appendage	31
Statistics Table	15	Unit Exception and Incorrect	
Attention Table.	15	Length Bits	31
Request Element Table.	15	Program Check, Protection Check,	
Logical Channel Word Table	16	and Chaining Check Bits	31
Queues.	16	Status Modifier and Busy Bits.	31
Logical Channel Queues	16	Trapcode Modules.	31
Seek Queues.	17	Unit Record and	
Queueing Procedures.	17	Telecommunications Trapcode	
PART I: EXECUTE CHANNEL PROGRAM		Modules	31
SUPERVISOR.	18	Tape Trapcode Module	31
General Operating Procedure.	18	Direct-Access Trapcode Module.	32
Operation of the Subroutines and		Channel Restart Subroutine.	32
Modules	19	Channel Search Modules.	32
EXCP Validity Check Subroutine.	19	Multiplexor Channel with Burst	
GET Request Element Subroutine.	20	Devices	32
Error EXCP Routine.	20	Multiplexor Channel with No	
Test Channel Modules.	21	Burst Devices	33
Selector Channel Test Channel		Selector Channel Included in	
Module.	21	Only One Logical Channel.	33
Multiplexor Channel Test Channel		Selector Channel Included in	
Module.	21	More than One Logical Channel	33
Enqueue and Dequeue Modules	21	Sense Subroutine.	34
Normal Enqueue Module.	22	Interfaces.	34
Priority Enqueue Module.	22	Attention Routine Interface.	34
Normal Dequeue Module.	22	Error Routine Interface.	35
Ordered Seek Enqueue Module.	22	SER Interface.	36
Ordered Seek Dequeue Module.	23	ABTERM Routine Interface	36
Select Subroutine	23	Post Routine Interface	36
Start I/O Modules	24	PART III: IBM-SUPPLIED ERROR ROUTINES	37
Unit Record and		Device-Dependent Routines.	37
Telecommunications Start I/O		General Operating Procedure	38
Module.	24	Device-Dependent Characteristics.	39
Tape Start I/O Module.	24	1052, 2150 Error Routine	39
Direct-Access Start I/O		2540/2821 Error Routine.	41
Module	25	1403/1443 Error Routine.	41
End-of-Extent Appendage.	26	1442, 2501, 2520 Error Routine	42
Start I/O Subroutine.	26	2671/2822 Error Routine.	43
Start I/O Appendage.	26	2400 Tape Series Error Routine	43
Post-Start I/O Subroutine	26	Direct-Access Device Error	
		Routine	45
		2250 Error Routine	48
		2260 Error Routine	48

Common Routines.	48	Event Control Block	62
Error Interpreter Routine	48	Input/Output Block.	62
Write-to-Operator Routine	48	Logical Channel Word Table.	66
Statistics Update Routine	49	Request Element Table	67
I/O Outboard Recording Routines.	50	Statistics Table.	67
The Outboard Recorder	50	UCB Lookup Table.	70
The Statistical Data Recorder	50	Unit Control Block.	71
Module Operation.	50	CHARTS	75
PART IV: SVC TRANSIENT AREA ROUTINES	52	APPENDIX A: LOAD MODULE NAMES	84
SVC Purge Routine	52	APPENDIX B: LINKAGES WITHIN THE	
Purge Complete Subroutine.	54	REQUEST ELEMENT TABLE	84
Restore Routine	54	APPENDIX C: THEORY OF LOGICAL	
DEVTYPE Routine	54	CHANNELS.	86
IOHALT Routine.	54	APPENDIX D: SYSTEM ENVIRONMENT	
PART V: CONTROL BLOCK AND TABLE		RECORDING OUTPUT DATA SET FORMAT.	88
FORMATS	55	Header Record	88
Attention Table	55	Statistical Data Records.	90
Channel Table	56	Record Entries.	91
Data Control Block.	57	INDEX.	93
Data Extent Block	59		
Device Table.	59		

FIGURES

Figure 1. Organization of the Input/Output Supervisor Within Main Storage	7	Figure 12. Device Table Entry Format	59
Figure 2. Relationships Among a Processing Program or the Control Program, an Access Method, and the Input/Output Supervisor	8	Figure 13. Event Control Block Format After Posting	62
Figure 3. Control Flow Within the System and the Input/Output Supervisor.	9	Figure 14. Input/Output Block Format	62
Figure 4. Control Blocks Used by the Input/Output Supervisor	12	Figure 15. Logical Channel Word Format	66
Figure 5. Tables Used by the Input/Output Supervisor	14	Figure 16. Request Element Formats	67
Figure 6. Execution of Ordered Seek Requests.	22	Figure 17. Statistics Table Entry Formats	67
Figure 7. UCB Lookup Algorithm	29	Figure 18. UCB Lookup Table Format	70
Figure 8. Attention Table Entry Format	55	Figure 19. Unit Control Block Format	71
Figure 9. Channel Table Format	56	Figure 20. Seek Queue Control Word Format for FIFO and Priority Requests	74
Figure 10. Data Control Block Fields Used by the Input/Output Supervisor	57	Figure 21. Seek Queue Control Word Format for Ordered Requests	74
Figure 11. Data Extent Block Format.	59	Figure 22. Linkages Within the Request Element Table	84
		Figure 23. Logical Channels.	86
		Figure 24. Physical Channel Paths and Logical Channels.	87
		Figure 25. Header Record Format.	88
		Figure 26. Statistical Data Record Format.	90
		Figure 27. Record Entry Format.	91

CHARTS

Chart 01. Functional Flow of the Input/Output Supervisor	75	Chart 06. Input/Output Interruption Analysis (2).	80
Chart 02. Control Flow Within the EXCP Supervisor.	76	Chart 07. Scheduling of an IBM-Supplied Error Routine.	81
Chart 03. Control Flow Within the Error EXCP Routine.	77	Chart 08. General Operating Procedure for Device-Dependent Error Routines	82
Chart 04. Control Flow Within the Input/Output Interruption Supervisor.	78	Chart 09. OBR/SDR Module Flow.	83
Chart 05. Input/Output Interruption Analysis (1).	79	Chart 10. General Operating Procedure-Graphics.	83

TABLES

Table 1. Modular Areas Within the Input/Output Supervisor	9
Table 2A. Sense Bytes and IOB Flags	40
Table 2B. IOB Error Counts.	40.1
Table 3. Load Module Names	84

The input/output (I/O) supervisor is the portion of the control program that issues the privileged I/O instructions and supervises the resulting I/O operations for any program that requests I/O device activity. The I/O supervisor has two purposes:

- To handle I/O requests, which are requests for the execution of channel programs.
- To handle I/O interruptions, which result from the execution of channel programs and from operator intervention.

To facilitate the handling of the I/O requests and interruptions, the I/O supervisor is divided into two major program sections:

- Execute channel program (EXCP) supervisor.
- Input/output (I/O) interruption supervisor.

These major sections are resident in main storage and provide control program support for the execution of channel programs.

The EXCP supervisor handles all I/O requests issued by means of the EXCP macroinstruction and SVC 15. The I/O interruption supervisor handles all I/O interruptions, provides error recovery supervision, indicates to the user when an I/O request is complete, and restarts available devices and channels.

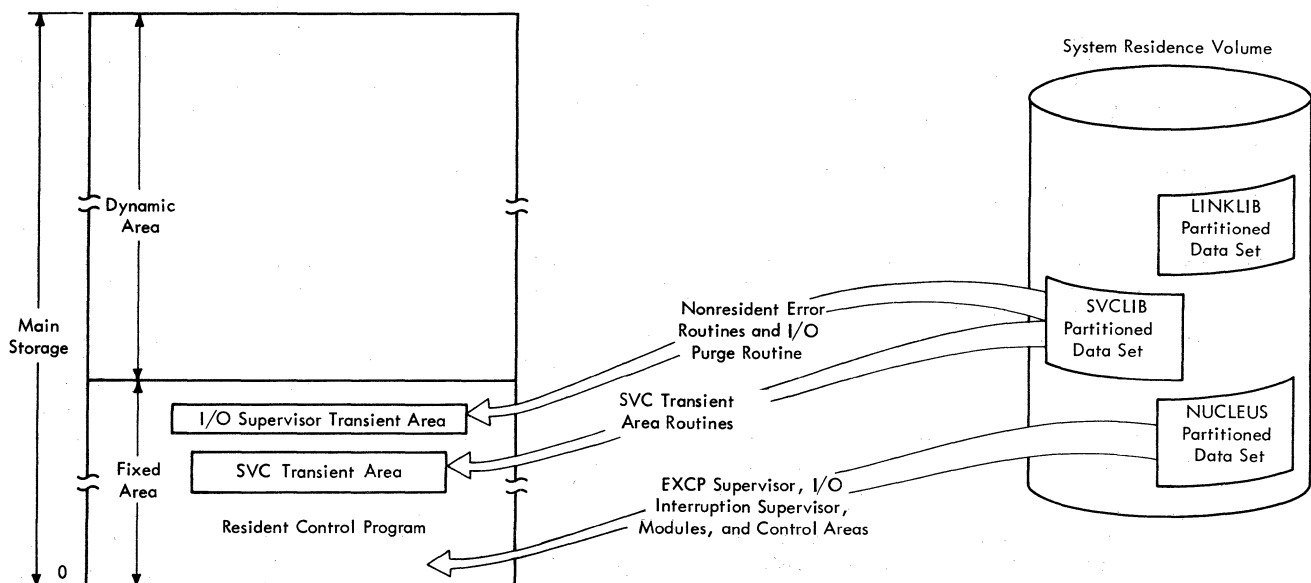
Routines and control areas used with the I/O supervisor are:

- Error routines.
- Purge routine and restore routine.
- DEVTYPE routine.
- IOHALT routine.
- Control blocks, tables, and queues.

Error routines are transient and are called by the I/O supervisor to handle device and channel errors and unusual conditions. The purge and restore routines are also transient and are used, respectively, to remove from the system I/O requests that are active or waiting to be started and to reschedule I/O requests that were previously removed. The DEVTYPE routine passes information concerning I/O device characteristics to the user. The IOHALT routine allows the graphics or telecommunications user to halt I/O activity on any device except a direct access device.

Control blocks, tables, and queues all reside in main storage. Control blocks are created either by the user at assembly time by means of control statements and macroinstructions, or by the system at system generation time. Tables, and also the control fields for queues and the elements to be queued, are created at system generation time. The queues are then maintained by the I/O supervisor.

Figure 1 shows the organization of the I/O supervisor within main storage.



• Figure 1. Organization of the Input/Output Supervisor Within Main Storage

THE INPUT/OUTPUT SUPERVISOR

The I/O supervisor starts and supervises I/O operations for programs that request I/O device activity.

Processing programs and the control program request I/O device activity via a macro-instruction. The type of macro-instruction that is used depends upon the use of an access method.¹

If an access method is used, the expansion of the GET/PUT or READ/WRITE macro-instruction contains a branch to access method routines. These routines construct a channel program, provide control block information, and cause control to be routed to the I/O supervisor via an EXCP macro-instruction.

If an access method is not used to request the execution of channel programs, the processing program or the control program must issue the EXCP macro-instruction. Before doing so, they must provide the necessary channel program and the control block information required for the I/O supervisor.²

Figure 2 illustrates the relationships among a processing program or the control program, an access method, and the I/O supervisor.

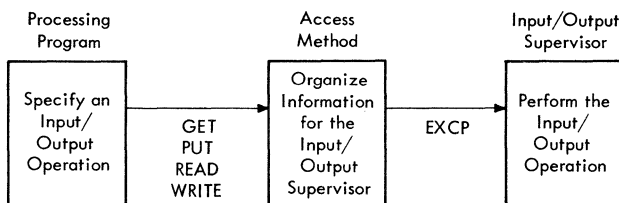


Figure 2. Relationships Among a Processing Program or the Control Program, an Access Method, and the Input/Output Supervisor

¹Access method descriptions are contained in the section "Data Access Methods" of the IBM System/360 Operating System: Data Management publication, Form C28-6537.

²The necessary information is contained in the section "Execute Channel Program (EXCP) Macro-Instruction" of the System Programmer's Guide.

When a channel program must be retried because of an error in its execution, SVC 15 is issued, usually by IBM-supplied error routines.

Execution of the EXCP macro-instruction or SVC 15 causes an interruption. This and all other interruptions in the system are handled by the first level interruption handlers (FLIH).³ After storing the program status word (PSW) and saving the contents of general registers, a FLIH routes control to the routine that performs the functions required by the interruption.

The I/O supervisor receives control from a FLIH after the following interruptions:

- Supervisor call (SVC) interruptions, caused when SVC 0 (EXCP) and SVC 15 are executed.
- I/O interruptions of all types.

An SVC interruption also occurs when SVC 16, 17, 24 or 33, respectively, for the PURGE RESTORE, DEVTYPE or IOHALT macro-instruction is executed. A FLIH gains control; however, since the required purge or restore routine is transient, it must be called into the SVC transient area before control is given to it.

Figure 3 shows the flow of control within the system and within the I/O supervisor from the time an SVC or I/O interruption occurs until the time the interruption is handled and the control program or a processing program is given control.

Within the mainline code of the I/O supervisor, there are five modular areas (these are not load modules). There are three modular areas within the EXCP supervisor and two within the I/O interruption supervisor. The size and function of these areas is dependent upon system configuration. Their inclusion in the I/O supervisor is determined at system generation (SYSGEN) time.

³The FLIHs are described in the IBM System/360 Operating System: Fixed-Task Supervisor, Program Logic Manual, Form Y28-6612.

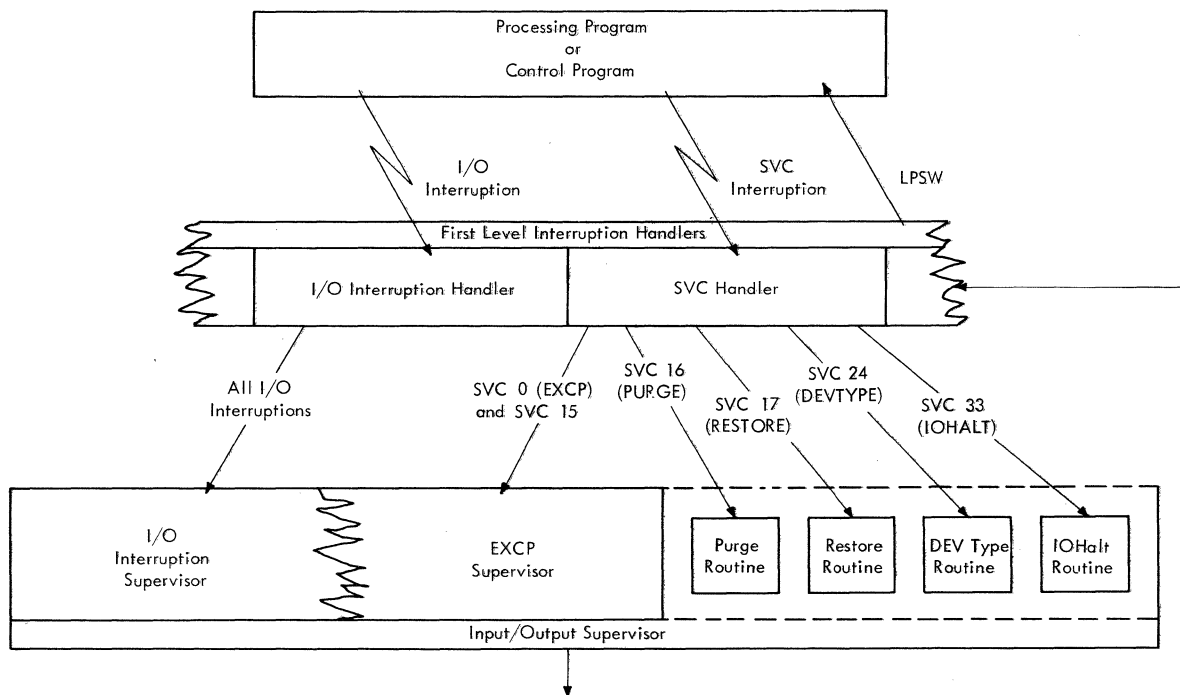


Figure 3. Control Flow Within the System and the Input/Output Supervisor

The modular areas, shown in Table 1, are:

- **Test channel.** A test channel module determines if a channel is available for the device that is to be used for an I/O request.
- **Enqueue and dequeue.** An enqueue module queues I/O requests that cannot be immediately started. After an I/O request has finished, a dequeue module removes the request from the queue.
- **Start I/O.** Start I/O modules provide device information and actions preceding the issuance of a start I/O instruction for the device.
- **Channel search.** When an I/O request has finished such that a channel becomes available, a channel search module determines the next I/O request to be started on that channel by searching queues constructed by enqueue modules.
- **Trapcode.** Trapcode modules establish information for a device after an I/O interruption associated with that device.

Table 1. Modular Areas Within the Input/Output Supervisor

Major Program Section	Modular Area	
	Name	Features
EXCP Supervisor	Test Channel	One module per logical channel; created at SYSGEN time.
	Enqueue and Dequeue	Three modules; selected at SYSGEN time according to enqueueing options.
	Start I/O	One module per device class; selected at SYSGEN time.
I/O Interruption Supervisor	Channel Search	One module per physical channel; created at SYSGEN time.
	Trapcode	One module per device class; selected at SYSGEN time.

The object module for the I/O supervisor is named IEASU00. This load module includes the mainline code and the modular areas of the I/O supervisor and in addi-

tion, includes the dispatcher, the exit effector, the FLIHS, and the communication vector table (CVT) of the fixed-task supervisor. The load module names for the various error routines and the purge and restore routines are summarized in Appendix A.

Routines that supplement the handling of I/O requests and interruptions are appendages to the I/O supervisor. (They do not alter the I/O supervisor program.) These appendages may be supplied by the user of the I/O supervisor.

Appendages are called into main storage by the open routine and are removed by the close routine or at the end of a job step.¹ When the I/O supervisor branches to an appendage, the current PSW has all interruptions except machine check interruptions masked as disabled; it indicates the supervisor state, and a protection key of zero.

There are five exits from the I/O supervisor to an appendage vector table; the table contains addresses, which may or may not route control to an appendage. (The appendage vector table is described in Part V: Data Extent Block.) If control is not routed to an appendage, the I/O supervisor retains control and continues normal processing.²

The five appendage exits are:

- End-of-extent. The provision for an end-of-extent appendage is included within the EXCP supervisor for direct-access device I/O requests. The appendage is given control when the EXCP supervisor determines that execution of the request will violate the extent limits of the data set.
- Start I/O. The provision for a start I/O appendage is included within the EXCP supervisor for all I/O requests. The appendage receives control preceding the execution of the start I/O (SIO) instruction for the request.
- Program-controlled interruption. The provision for an appendage for a program-controlled interruption (PCI) is included within the I/O interruption supervisor. The PCI appendage returns

control to the I/O interruption supervisor where normal processing continues.

- Channel end. The I/O interruption supervisor exits to a channel end appendage after a channel end interruption occurs, before indicating to the user how the I/O request completed. (The channel end appendage is also entered when the channel end interruption is accompanied by a wrong length indication or a unit exception, or both.)
- Abnormal end. The I/O interruption supervisor exits to an abnormal end appendage whenever there has been an error associated with the processing of an I/O request.

EXCP SUPERVISOR

The EXCP supervisor is the portion of the I/O supervisor that starts the channel programs requested by use of the EXCP macro-instructions and SVC 15. It performs the same procedures when initiating the channel programs for both. Chart 01 shows the functional flow of the EXCP supervisor, and also of the I/O interruption supervisor. (All charts precede Appendix A.)

Before accepting a channel program for execution, the EXCP supervisor determines if:

- The control blocks for the I/O request are valid.
- The device for the I/O request is available.
- A physical channel for the device is available.

If the first of these conditions is not met, either a program check results or the task requesting the I/O activity is abnormally terminated. If either the device or the channel is not available, the I/O request is queued for subsequent processing.

When the required conditions are met, the EXCP supervisor issues an SIO instruction, which in turn initiates the commands in the channel program. The following conditions may result:

- The channel program may be accepted and the I/O operations successfully started.

¹The OPEN and CLOSE routines are described in the publication IBM System/360 Operating System: Input/Output Support (OPEN/CLOSE/EOV), Program Logic Manual, Form Y28-6609.

²Additional information regarding the use of appendages can be found in the EXCP section of the System Programmer's Guide.

- An error may occur during initiation of the channel program, in which case, control is given to the I/O interruption supervisor and the reason for the error is determined. If necessary, an error routine is scheduled.
- The control unit for the device may signal busy. In this case, the channel program is not accepted and the I/O request is queued (except in the case of some telecommunications requests).

After the resulting condition is handled, control returns to the FLIH.

I/O INTERRUPTION SUPERVISOR

An I/O interruption occurs after signals from I/O devices have been generated (in response to channel program execution and operator action) and accepted by the central processing unit (CPU). The I/O interruption supervisor analyzes status bits in the channel status word (CSW), which is stored as a result of the I/O interruption, to determine the reasons for the I/O interruption.

The I/O interruption supervisor processes the following, when denoted by the CSW status bits:

- Attentions. The appropriate attention routine is entered.
- Status modifier situations. Status modifier conditions, such as a control unit busy condition that causes an I/O request to be queued, are handled.
- Control unit end, channel end, and device end interruptions. These interruptions indicate normal ending conditions and normal processing continues. Normal completions are indicated to the user.
- Busy conditions. When the selected I/O device or control unit is unavailable, the I/O request is queued.
- Unit check. A sense command is issued to obtain additional information regarding the unit check.
- Errors and unusual conditions. When errors, such as program, channel chaining, and protection checks, and unusual conditions, such as unit exception and wrong length indication, are detected, an error routine is scheduled. An uncorrectable error is indicated to the user as a permanent error condition.
- Program-controlled interruptions. A PCI appendage supplied by the user of the I/O supervisor is entered.
- Catastrophic errors. When catastrophic errors (channel data check, channel control check, and interface control check CSW bits) occur, the system environment is recorded and the CPU is put into the wait state. The system is then considered inoperative and the Initial Program Load procedure must be followed to re-initialize the system. (System environment recording (SER) is a group of programs that collect and record data regarding System/360 machine errors.)

The completion of an I/O request is indicated by a channel end or a channel end and device end interruption. The I/O interruption supervisor indicates this completion to the user.

After the reason for the I/O interruption has been determined and the interruption processing completed, the I/O interruption supervisor determines the state of the channel upon which the interruption was received. Pending I/O interruptions are accepted and handled. If the channel is free, an attempt is made to restart it with a queued I/O request (one that was previously requested, but could not be started immediately). The appropriate queues are searched until an I/O request is started or until it is determined that there are no I/O requests that can be started on the free channel. Control returns to the FLIH.

THE ERROR ROUTINES

Standard IBM-supplied error routines are selected at SYSGEN time for each type of I/O device in the system. Error routines are transient and reside in the SVC library.¹ However, the portions of the direct-access device error routine that are necessary for the system residence device and for frequent unusual conditions (end-of-cylinder, head switching, alternate track procedures, etc.) are resident in main storage. The names of the error routine load modules are in Appendix A.

Whenever an I/O interruption occurs and the I/O interruption supervisor detects an error that requires error recovery procedures, it determines whether or not an IBM-supplied error routine is to be used.

¹The requirements for an error routine are discussed in the EXCP section of the System Programmer's Guide.

An indication of the required error routine and of the channel program that developed the error are passed to the task supervisor's exit effector and an asynchronous exit is requested. The error routine must be scheduled asynchronously because immediate in-line processing of the error would hinder efficient channel scheduling.

The appropriate error routine is entered into the I/O supervisor transient area (providing the error routine is not resident) and, when scheduled, is given control. The error routine determines the type of error and, where possible, issues the ERREXCP macro-instruction in an attempt to retry the channel program and to recover from the error. At completion of error processing (successful or unsuccessful), the error routine issues the ERREXCP macro-instruction so that the I/O request can be properly terminated by the I/O interruption supervisor. The completion is indicated to the user and the error routine returns control to the task supervisor.

THE PURGE AND RESTORE ROUTINES

The SVC purge and restore routines are transient and are called into the SVC transient area whenever a PURGE or RESTORE macro-instruction (SVC 16 and SVC 17, respectively) is issued. The load module name for the SVC purge routine is IGE0001F and for the SVC restore routine is IGE0001G.

The SVC purge routine removes active or queued I/O requests from the system. The SVC restore routine reschedules I/O requests that were previously purged.

There is also an I/O purge routine; the load module name is IGE0025E. This routine is used only by the I/O supervisor and should not be confused with the SVC purge routine, which is called by means of a system macro-instruction. The I/O purge routine removes certain requests from the system whenever there has been a permanent error. It is discussed in Part II: Error Routine Interface.

THE CONTROL BLOCKS, TABLES, AND QUEUES

The I/O supervisor, as well as other portions of the control program, records the status of devices, data sets, and internal routines in control blocks, tables, and queues.

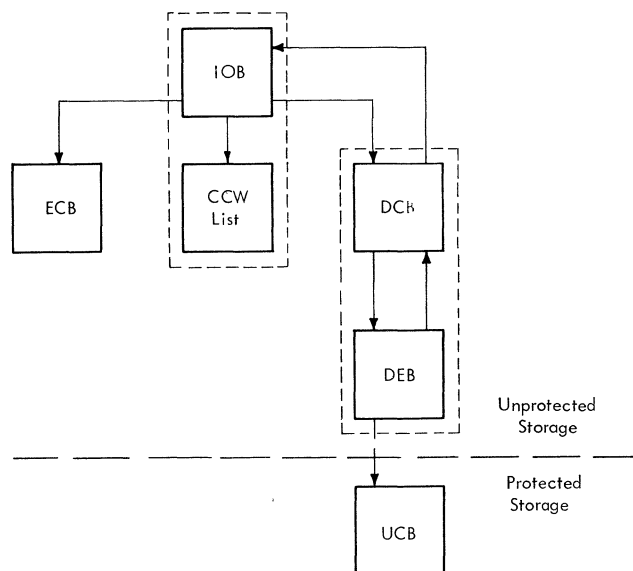
The control blocks supply information to the I/O supervisor regarding I/O requests and devices, and regarding the data sets required for the execution of the requests. The tables point to device- and channel-dependent modules and are used internally by the I/O supervisor. Queues are used by the I/O supervisor for I/O requests that cannot be started immediately. Part V contains detailed formats and explanations of the control blocks and tables.

CONTROL BLOCKS

Control blocks maintain the information that concerns the status and activity of I/O devices, requests, and interruptions. The I/O supervisor is concerned with the following control blocks:

- Input/output block.
- Data control block.
- Data extent block.
- Event control block.
- Unit control block.

These control blocks, and the pointers indicating their relationships, are shown in Figure 4.



• Figure 4. Control Blocks Used by the Input/Output Supervisor

Input/Output Block

An input/output block (IOB) is the communication medium between a routine that requests I/O operations and the I/O supervisor. The address of an IOB is passed to the I/O supervisor when an I/O request is given.

The contents of an IOB include:

- The address of the channel program, which consists of a group of channel command words (CCW), to be executed.
- An indication that the channel program is or is not related to other channel programs. (Related channel programs are all associated with the same data set. Each is dependent upon the successful completion of its preceding related channel programs. If a permanent error occurs that is associated with a related channel program, the I/O purge routine purges the remaining related requests for the data set.)
- A storage area for the CSW, which is stored by the I/O supervisor at the completion of an I/O request.
- The initial seek address for a direct-access device.
- The address of the data control block and the event control block.

Data Control Block

A data control block (DCB) contains information that pertains to the current use of a data set and contains the address of a data extent block. A DCB provides the I/O supervisor with the status of error correction procedures for each data set and indicates whether or not IBM-supplied error routines are to be used for the data set.

Data Extent Block

A data extent block (DEB) is an extension of the information contained in a DCB. Each DEB is associated with a DCB, and the two blocks address each other. A DEB describes the extent of a data set; it indicates the storage medium, the location, and the boundaries of the data set, contains an appendage table address, and provides data protection and priority information.

Event Control Block

An event control block (ECB) is used by the I/O supervisor to indicate to the user when and how the execution of his channel program completed. A normal completion is determined by either a channel end or a channel end and device end interruption. The I/O supervisor indicates a request has completed, successfully or not, by posting the request complete by use of the post routine of the task supervisor.¹

Unit Control Block

A unit control block (UCB) is the means by which the I/O supervisor notes and determines the activity and status of each I/O device attached to the system. A UCB represents the I/O device for the volume upon which a data set resides. It is addressed by a DEB.

There is one UCB for each I/O device or telecommunications line addressed in the system. One UCB per volume is required if a data set resides on two or more volumes that are mounted concurrently upon separate I/O devices. In this case, each of the UCBs is addressed by the same DEB.

The contents of the UCB include:

- The unit address, which is used in the SIO instructions for the device.
- The type of device the UCB represents (e.g., IBM 2311 Disk Storage, IBM 2400 Magnetic Tape Unit, etc.).
- The flag bits, which indicate the current status of the associated I/O device. For example, the UCB busy bit may be set to indicate that the device is currently in operation. Other flag bits may be set to indicate UCB not ready, control unit busy, post, etc.
- An area for sense information, which details the conditions, such as a programming error or equipment malfunction, that cause a unit check interruption. When the I/O supervisor issues a sense command, up to six bytes of sense information from the control unit of the device is read into this area; the number of bytes of information that is meaningful varies with the device type.

¹For information concerning the post routine, refer to the Fixed-Task Supervisor, Program Logic Manual.

- Reference fields, consisting of an indexing value or an address, that refer to tables. When the reference is an indexing value, the I/O supervisor adds it to the starting address of the pertinent table to obtain the address of the proper entry. When the reference is an address, it is of the proper entry.

UCB Lookup Table

The UCB lookup table contains the address of each UCB in the system and the values that are necessary to locate these UCB addresses within the table. The table is divided into channel values, control unit values, and UCB addresses.

TABLES

Tables contain pointers to control blocks and to device- and channel-dependent routines, modules, and queues. They also provide an area for device statistics. The I/O supervisor contains the following tables:

- UCB lookup table.
- Channel table.
- Device table.
- Statistics table.
- Attention table.
- Request element table.
- Logical channel word table.

These tables, and the pointers indicating their relationships, are shown in Figure 5.

Channel Table

The channel table is used by the I/O supervisor to re-enable a channel for additional I/O interruptions and to locate the channel search module for the channel. The table contains one entry for each physical channel attached to the system.

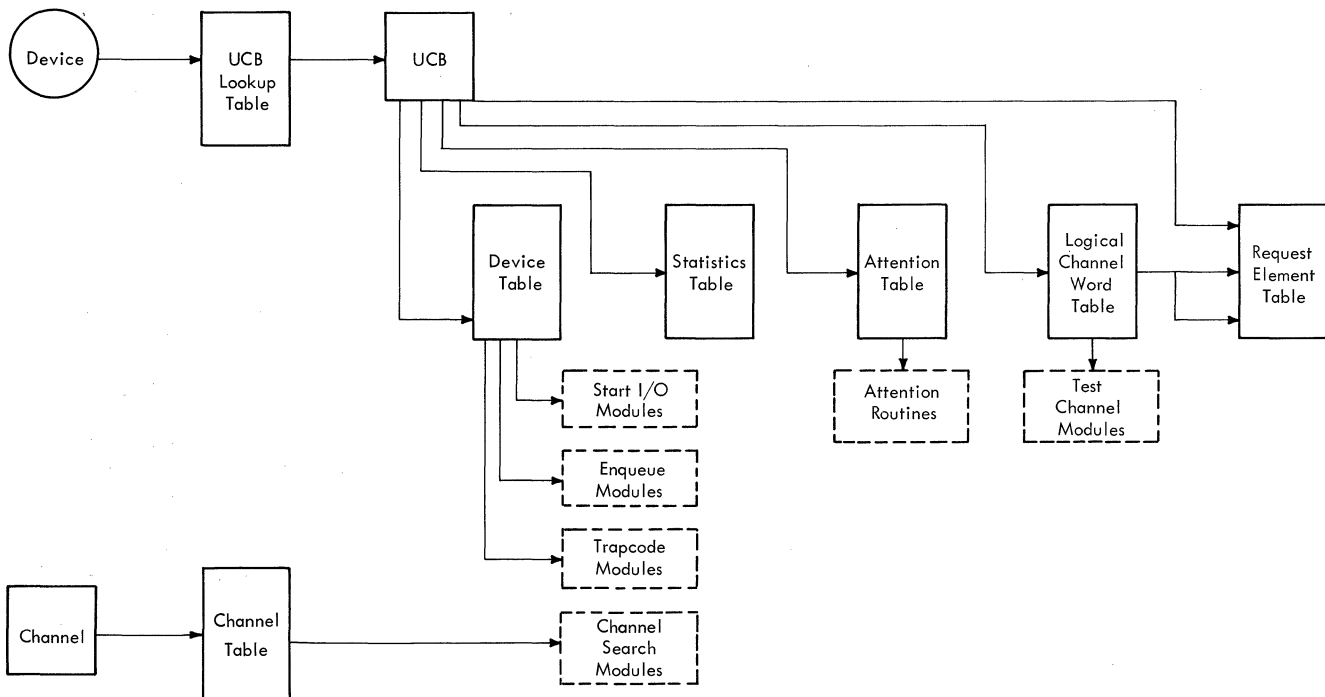


Figure 5. Tables Used by the Input/Output Supervisor

Device Table

The device table is the means by which the I/O supervisor determines the proper device-dependent modules for a device. Each entry in this table contains the address of a start I/O module, an enqueue module, and a trapcode module.

Statistics Table

The statistics table contains one entry, which includes up to eight statistics counters, for each I/O device attached to the system. Each statistics counter is used by the IBM-supplied error routines to count the number of temporary read and write errors, equipment checks, interventions required, and other statistics that depend on the particular device type.

Whenever a statistics counter overflows or a permanent error occurs, the SER program's statistical data recording routine uses the statistics counters to update statistical data records. Statistical data records are expansions of the statistics counters. They reside on the system residence device, so that they are available for recall and analysis.

Attention Table

The attention table is constructed at SYSGEN time to contain the addresses of attention routines. Attention routines are specified by system routines to service attention interruptions.

There are two types of attention interruptions:

- Normal attention interruptions, which are indicated by the attention bit in the CSW status field.
- Unsolicited device end interruptions, which are caused by the transition from the not-ready to the ready state that occurs when the operator presses the ready button.

When a system routine desires that an attention or unsolicited device end interruption is to be handled, it places an indexing value into the UCB for the device. When the indexing value is added to the starting address of the attention table,

the address of the appropriate entry within the table is obtained. The system routine must have placed into this entry, at SYSGEN time, the address of the attention routine for the device. Since the attention table is constructed at SYSGEN time, it is available only to routines that specify their attention routines at SYSGEN time.

When an attention interruption occurs, the I/O supervisor passes control to the attention routine. If execution of the attention routine is not desired, the system routine must place a zero into the UCB instead of an indexing value. The first entry in the attention table contains a branch back to the I/O supervisor; thus, when an attention interruption occurs, the starting address added to zero yields the address of the first entry in the table and the I/O supervisor regains control.

Request Element Table

The request element table¹ consists of a number of request elements that are used to represent I/O requests. The number of elements in the table is determined at SYSGEN time and remains fixed.

The I/O supervisor issues a request element to represent each I/O request given by means of the EXCP macro-instruction. An element is not issued when an I/O request is retried by means of SVC 15. When the element is issued, the I/O supervisor stores information into it, which includes the addresses of the UCB, DEB, and IOB associated with the I/O request.

Each request element can be in one of three states:

- Available.
- Active.
- Queued.

An example of the linkage relationships of the available, active, and queued request elements within the request element table is presented in Appendix B.

Available Request Elements: Available request elements are not in current use; they are either unused or contain information that is no longer needed by the I/O supervisor. These request elements are issued to represent incoming I/O requests.

¹In some documentation, the request element table is referred to as the 12 star (12*) table and its request elements as 12* elements.

The available request elements are linked together by means of a "link field" that is contained in the first two bytes of each element. This linkage of available request elements forms a "freelist". Each element in the freelist points to the next element. Because other request elements in the request element table might be active or queued, members of the freelist are not necessarily contiguous within the table.

The first element in the freelist, referred to as "next available", is always the next request element issued by the I/O supervisor for an incoming I/O request. The "freelist pointer" contains the address of the next available request element. It resides within the I/O supervisor and is addressed by an entry in the communications vector table (CVT).¹

The last element in the freelist is the last available request element and is identified by means of a "dummy address" in its 2-byte link field. The dummy address is a hexadecimal FFFF (i.e., has all bits on). When there are no available request elements, the freelist pointer contains the dummy address to indicate that the freelist is empty.

Active Request Elements: Active request elements correspond to I/O requests that have been started but have not yet been completed. The I/O supervisor indicates that a request element is active by placing the address of the active request element into the UCB that represents the device being used for the I/O request. When an I/O interruption associated with the device occurs, the I/O supervisor finds the active request element, which is then used to find the various control blocks that are used during the processing of the I/O interruption.

Queued Request Elements: Queued request elements correspond to I/O requests that could not be started immediately. The elements belonging to the same queue are linked together by means of the link fields located in the first two bytes of the elements. The last element in each queue is denoted by means of the dummy address within its link field. During the queuing procedure, only the contents of the link fields in the request elements are changed.

¹The CVT is described in the Introduction to Control Program Logic, Program Logic Manual.

Logical Channel Word Table

The logical channel word table contains a logical channel word entry that corresponds with each logical channel in the system. A multiplexor channel has one logical channel; a selector channel may have more than one. (For further information concerning logical channels, refer to Appendix C.)

A logical channel word designates the first and last request elements in the logical channel queue associated with a logical channel. If the queue is empty, the logical channel word contains in its first two bytes the dummy address of hexadecimal FFFF. A logical channel word also contains the address of a channel-dependent test channel module.

QUEUES

The I/O supervisor uses two types of queues for I/O requests:

- Logical channel queues.
- Seek queues.

The logical channel queues consist of request elements that represent I/O requests for the execution of channel programs on either sequential or direct-access devices.

The seek queues contain request elements that represent I/O requests for seek operations to be executed on direct-access devices. When a seek operation has been executed and the access arm of the direct-access device is in the proper position, the representative request element is enqueued in the appropriate logical channel queue.

Logical Channel Queues

I/O operations awaiting execution are held for subsequent processing in a logical channel queue. There is one logical channel queue associated with each logical channel in the system. (See Appendix C.) A multiplexor channel has one logical channel queue; a selector channel may have more than one. The last request element in each logical channel queue is identified by the dummy address in the link field of the element.

Seek Queues

The I/O supervisor uses seek queues in conjunction with logical channel queues to overlap seek operations for non-drum direct-access devices that are attached to the same physical channels. There is one seek queue associated with each non-drum direct-access device in the system. The UCB for a direct-access device contains a seek queue control word that indicates the seek queue for the device. When the seek queue is empty, the seek queue control word contains in its first two bytes the dummy address of hexadecimal FFFF.

When an I/O request is given for a direct-access device, a seek address is indicated by the user in the associated IOB. This seek address is used by the I/O supervisor to issue an initial seek command required for the requested I/O operations.

When the direct-access device necessary to execute the initial seek command is busy, the request element representing the I/O request is enqueued in the seek queue for that device. When the direct-access device becomes available and the seek command has been executed so that the access arm is at the proper address, the request element is dequeued from the seek queue and enqueued in the logical channel queue associated with the device and the physical channel. The remaining I/O operations specified in the channel program for the I/O request are then started when a physical channel and the device are available.

Queueing Procedures

The I/O supervisor effects linkage within the queues of request elements in the request element table by means of three types of queueing procedures: first-in-first-out (FIFO), priority, and ordered. The user selects queueing procedures at SYSGEN time.

In FIFO queueing, a request element is queued according to the order in which its corresponding I/O request is received (pushup queue). The queue is constructed and maintained so that the next request element to be retrieved is the oldest element in the queue.

In priority queueing, a request element is queued according to the dispatching priority of the task associated with its corresponding I/O request. The next request element to be retrieved is that with the highest priority.

In ordered queueing, a request element, representing an I/O request for a direct-access device, is queued according to the numerical order of the given seek address. The seek addresses are ordered from the lowest to the highest. Related requests are queued in FIFO arrangement and are executed where appropriate during the ordered "sweep" of the direct-access arm. Ordered queueing minimizes the movement of the arm.

PART I: EXECUTE CHANNEL PROGRAM SUPERVISOR

The EXCP supervisor receives control initially from the SVC FLIH after either an SVC 0 or an SVC 15 interruption. When the EXCP supervisor is entered, the current PSW has all interruptions except machine check interruptions masked as disabled, indicates supervisor and running states, and has the supervisor protection key of zero.

The EXCP macro-instruction (SVC 0) is used to request normal I/O processing of a channel program. SVC 15 is used by the error routines to request that a channel program whose execution encountered an error be retried and also to indicate a permanent error or a successful retry. The I/O purge routine uses SVC 15 to return control to the I/O supervisor after related requests are purged.

The user of EXCP or SVC 15 is responsible for certain fields within the DCB and the IOB associated with the I/O request. Within the DCB, the user must ensure that:

- DCB exception bits set to 11, indicating a permanent error on the data set, are reset to 00. If this is not done, further related requests for that data set are posted as complete without execution.
- DCB exception bits set to 01, indicating error recovery procedures are in progress, remain at this setting.
- The tape block count remains unchanged until the IOB associated with the request has been posted complete.

Within the IOB, the user must ensure that:

- The block count increment amount for tape block counts is reset if the channel program indicated by the IOB changes.

The field must be zero if block count updating is not desired.

The field must be negative for a backspace.

The field must be zero for a rewind; the DCB block count field must also be zero.

- The unrelated flag is on to indicate the I/O request is unrelated or is off

to indicate the I/O request is related, and must be handled as such.

- The start address contains the address of the first CCW to be executed in the channel program.
- The channel program description indicates the type of chaining in the channel program, as follows:

00 No chaining
10 Data chaining
01 Command chaining
11 Data and command chaining

- The ECB address contains the address of the corresponding ECB.
- The DCB address contains the address of the DCB for the data set on which I/O processing is requested.

When the EXCP supervisor receives control, the address of the IOB that contains the information necessary for the I/O request must reside in general register 1. Processing of the I/O request then takes place.

After the operation of the EXCP supervisor is complete, control returns to the type 1 exit of the SVC FLIH, which determines whether control is to be given to the dispatcher or to the routine giving the I/O request.

Chart 02 shows control flow among the subroutines and the modules within the EXCP supervisor.

GENERAL OPERATING PROCEDURE

For a normal I/O request, the EXCP validity check subroutine is entered directly from the SVC FLIH. This subroutine checks the validity of the control blocks associated with the I/O request. It also initializes certain IOB fields that are to be used in later processing.

For an error retry, the EXCP validity check subroutine is entered from the error EXCP routine. The error EXCP routine ensures that the IOB fields are not initialized, since their content is necessary for the error retry.

After the EXCP validity check subroutine is executed, the get request element subroutine issues a request element for a normal I/O request. It stores address information required by the I/O supervisor into the next available request element. Because a request element is issued for all normal I/O requests, an additional request element is not issued if an error retry for a particular I/O request becomes necessary.

The flag bits within the UCB that denote the status of the I/O device are inspected to determine if the requested device is available for processing.

When the device is available, a test channel module determines if a physical channel in the device's logical channel is available. The correct test channel module is selected by means of the logical channel word table.

If conditions associated with the device prohibit processing or if the test channel module does not find an available physical channel, the request element is queued for subsequent processing of the I/O request. The appropriate enqueue module is selected by means of the device table entry for the particular device.

When a channel is free and the device is available, the select subroutine is entered. This subroutine selects the appropriate start I/O module for the device by means of the device table and gives control to it. The start I/O module performs device-dependent functions and branches to the start I/O subroutine, which in turn issues the SIO instruction for the I/O request.

The condition code and the CSW (if stored) that results from the execution of the SIO instruction indicate whether the channel program for the request has been started, whether the request element must be queued, or whether the SIO instruction for the I/O request should be re-issued. The start I/O subroutine and the post-start I/O subroutine handle these results.

Once the operations for an I/O request are terminated (because of completion, error, or external action on the device), an I/O interruption occurs. Information concerning the operation of the I/O interruption supervisor is contained in Part II: I/O Interruption Supervisor.

OPERATION OF THE SUBROUTINES AND MODULES

The following subroutines, routines, and modules are used to perform the functions of the EXCP supervisor:

- EXCP validity check subroutine.
- Get request element subroutine.
- Error EXCP routine.
- Test channel modules.
- Enqueue and dequeue modules.
- Select subroutine.
- Start I/O modules.
- Start I/O subroutine.
- Post-start I/O subroutine.

EXCP VALIDITY CHECK SUBROUTINE

For both normal (SVC 0) and error retry (SVC 15) entries into the EXCP supervisor, the EXCP validity check subroutine validates the control blocks associated with the request. These control blocks are:

- Input/output block.
- Data control block.
- Data extent block.
- Unit control block.

The addresses of the control blocks are placed into general registers. During this process, the DEB and UCB identification fields are checked, the pointer to the DCB within the DEB is checked, and if applicable, the DEB protection tag is compared against the task control block (TCB) protection tag. If the subroutine finds that the control blocks are on improper boundaries, a program check occurs. On other errors, the ABTERM routine interface of the I/O interruption supervisor is entered and the task is abnormally terminated.

When the EXCP validity check subroutine is used for normal I/O processing, the following IOB fields associated with the last I/O request processed on the currently requested I/O device are reset:

- Error count.
- Flags 1, 2, and 3.
- CSW.
- Condition code.
- Sense.
- ECB code.

When the subroutine is used for error retry, these fields are not reset and control returns to the error EXCP routine.

GET REQUEST ELEMENT SUBROUTINE

After a normal I/O request is validated, a request element is issued to represent it. If an error retry for a particular I/O request becomes necessary, an additional request element is not required because one had been previously issued. Control block information is loaded into the request element, and UCB conditions, which determine if the request element must be queued, are tested.

If there is no available request element (i.e., the freelist pointer contains the dummy address of hexadecimal FFFF), the EXCP supervisor uses an enable/disable loop (pseudodisable) to allow an I/O interruption to occur. The I/O interruption supervisor processes the I/O interruption and, if the I/O interruption signifies completion of an I/O request, returns the associated request element to the freelist. If, however, there is an error associated with the I/O request that is interrupted, the task that initiated the operation in error is abnormally terminated.

The EXCP supervisor uses the request element that is returned to the freelist to represent the I/O request. However, if no request element can be returned to the freelist, or if no I/O interruption occurs, the EXCP supervisor repeats the enable/disable loop.

The next available request element is loaded with the following information:

- Address of the UCB, which represents the I/O device to be used for the requested operations.
- TCB identification, as noted in the TCB for the task requesting I/O operations. (In an operating system in which option 4 is included, the address of the TCB is also placed into the request element.)
- Address of the IOB, which indicates the channel program to be executed.
- Dispatching priority of the request, as noted in the DEB. (If no priority is specified, this field is not significant.)
- Address of the DEB, which indicates the location of the data set to be operated upon.

If the request is related and there is a permanent error, as indicated by the IOB unrelated flag and the DCB permanent error code, the request cannot be executed. (An error has occurred on a previous related

I/O request for the data set.) In this case, the abnormal end appendage exit is taken. For a normal return from the appendage, the request is posted complete with the purge code and the request element is returned to the freelist. Control returns to the task supervisor.

UCB Condition Test: The conditions associated with the I/O device that may prohibit immediate processing are denoted by the following bits contained in the UCB:

- Control unit busy.
- Device busy.
- Not ready.
- Disk arm seeking.
- Error routine in control.

If any of these bits are on, the request element is queued by an enqueue module for subsequent processing of the I/O request. If none are on, the device is available, is able to accept the request, and a test channel module is entered.

ERROR EXCP ROUTINE

The error EXCP routine directs the retry of I/O requests (and the results of each retry) to the I/O supervisor. The routine is entered from the SVC FLIH after the SVC 15 interruption. Chart 03 shows the flow of control in the error EXCP routine.

SVC 15 is provided to simplify error recovery procedures with respect to request elements; an additional request element need not be issued by the get request element subroutine.

IBM-supplied error routines issue SVC 15 to retry channel programs because of an error, and also issue it when the error has been corrected, when another retry is necessary, and when the error is permanent. (See Part III: IBM-Supplied Error Routines.)

The I/O purge routine issues SVC 15 when it has purged all related requests and the abnormal end appendage must be entered. (See Part II: Error Routine Interface.)

The resident portion of the direct-access device error routine returns (it need not issue SVC 15) to the error EXCP routine: when an error is corrected; and when a request element must be queued in a seek queue for error recovery purposes. In the first case, the channel end appendage is entered; in the second, the channel restart subroutine of the I/O interruption supervisor is entered.

The only parameter that the user of SVC 15 must furnish is the address of the request element that represents the I/O request that is to be retried. The request element contains the address of the IOB associated with the I/O request and is loaded into general register 1 for entry to the EXCP validity check subroutine.

The EXCP validity check subroutine checks the validity of the control blocks pertinent to the request and loads their addresses into general registers. The EXCP validity check subroutine then returns to the error EXCP routine; the IOB fields are not reset.

When an error routine is in control (the IOB error correction indicator flag is set) and the request must be retried, the error EXCP routine tests the following UCB condition flags:

- Device busy.
- Not ready.
- Disk arm seeking.
- Disk data transfer.

The EXCP supervisor determines the result of the test, and the request element is either enqueued or a test channel module is entered.

When the error has been corrected and the error routine is no longer in control, the channel end appendage is entered. If the error has not been corrected, the error routine interface of the I/O interruption supervisor is entered, which indicates a permanent error condition and enters either the I/O purge routine or the abnormal end appendage, depending on whether or not the request is related.

TEST CHANNEL MODULES

When the device necessary for the I/O request is available, a test channel module determines whether or not a physical channel in the device's logical channel is available for device operation. If a physical channel is not available, control passes to an enqueue module and the request element is queued.

There are two types of test channel modules: one for a selector channel; the other for a multiplexor channel. A selector channel test channel module exists for each logical channel in the system. One multiplexor channel test channel module exists for the logical channel associated with the multiplexor channel.

Module Selection: The test channel module associated with the device and its logical channel is selected by means of the logical channel word indexing value, which is contained in the UCB, and the logical channel word table. The proper logical channel word contains the address of the test channel module for the device.

Selector Channel Test Channel Module

The selector channel test channel module issues a test channel (TCH) instruction for each physical channel attached to the requested device to determine whether or not a physical channel is available for that device. If an available channel is found, the select subroutine attempts to start the channel program for the I/O request. If one is not found, the request element is queued.

Multiplexor Channel Test Channel Module

Because a multiplexor channel never presents a busy condition to a TCH instruction, one is not issued and control immediately passes to the select subroutine, which attempts to start the I/O request. The multiplexor channel test channel module is included to facilitate channel-independent coding.

ENQUEUE AND DEQUEUE MODULES

If the channel program for an I/O request cannot be immediately started because a device or a physical channel is unavailable, the request element that represents the I/O request is queued. The queueing procedure used depends on the type of device and the type of enqueueing specified by the user at SYSGEN time.

The queueing procedures that are available are: FIFO, priority, and ordered. Logical channel queues are maintained in either FIFO or priority arrangements; seek queues are maintained in FIFO, priority, or ordered arrangements.

The dequeuing procedures depend on the type of enqueue module assigned to a device. The normal dequeue module is used for the dequeuing requirements of all queues maintained in FIFO and priority arrangements. The dequeue module for ordered seeks is used for seek queues maintained in an ordered arrangement.

Module Selection: The address of the appropriate enqueue module is contained in the device table entry for the device class of the particular device. The proper entry in the table is found by use of the device table indexing value contained in the UCB that represents the device.

To determine the dequeue module to be used for a particular type of enqueueing, the enqueue module address that is contained in the device table is incremented by eight; at the resultant address is a branch to the dequeue module.

Normal Enqueue Module

The normal enqueue module queues request elements in a FIFO arrangement. The module is used for the enqueueing requirements of both logical channel queues and seek queues.

This module places the address of the request element that represents an I/O request into the link field of the last request element in the pertinent logical channel queue or seek queue. This request element becomes the "last" element in the queue and the dummy address is placed into its link field. The previous last request element is then updated to point at the new last request element.

Priority Enqueue Module

At SYSGEN time, the user specifies priority queueing for certain logical channel queues and seek queues. The request elements that represent the I/O requests are queued in an order that maintains the priority specified in the DEB for the pertinent task.

Priority queueing is effected by searching the logical channel or seek queue until the priority point at which the request element is to be queued is found. The address of the request element is placed into the link field of the next highest or equal priority request element in the queue. The highest priorities are at the top of the queue. Within a given priority, requests are in a FIFO arrangement.

Normal Dequeue Module

The normal dequeue module is used by the I/O supervisor to dequeue request elements

from queues maintained in FIFO and priority arrangements. A search is made of the queue until the request element to be dequeued is located. The element is unlinked and the queue is relinked by means of the 2-byte link field.

Ordered Seek Enqueue Module

The ordered seek enqueue module queues the request elements for unrelated seek requests into the seek queue for a direct-access device according to seek address. This arrangement causes the stand alone seeks issued by the channel restart subroutine of the I/O interruption supervisor to be executed in the order of seek address, as the arm moves from the lower to the higher cylinder addresses. Ordered arrangement optimizes seeking time by eliminating irregular arm movement. Priority considerations are disregarded in ordered queueing.

Since seek requests are executed only as the arm is moving from the lower to the higher addresses on the cylinders, the cylinder address at the time of the request determines whether the request is to be executed in this pass across the face of the device or during the next pass. (See Figure 6.)

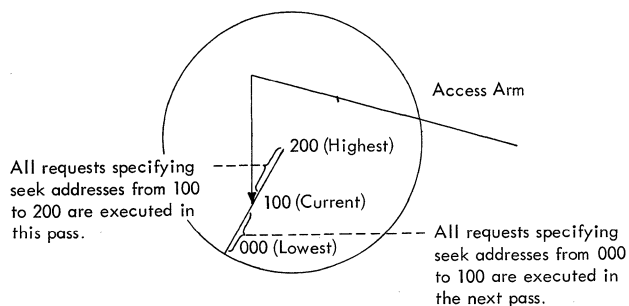


Figure 6. Execution of Ordered Seek Requests

To help determine when the request is to be executed, a seek queue for ordered queueing is arranged into three different queues. The first element in each queue is addressed by the seek queue control word in the UCB for the direct-access device. (See Part V: Unit Control Block.)

The seek queue divisions and their arrangements are:

- Primary queue. The primary queue contains all request elements for requests that specify a seek address greater than or equal to the current cylinder address. The queue is ordered from the lowest seek address, which is the top of the queue, through the highest seek address.
- Secondary queue. The secondary queue contains all request elements for requests that specify a seek address less than the current cylinder address. The secondary queue is ordered the same as the primary queue.
- Related queue. The related queue contains all request elements for related requests. The queue is maintained in a FIFO arrangement. The order of execution of a related request is discussed in "Ordered Seek Dequeue Module."

When an I/O request must be queued for a direct-access device that has ordered queueing specified, the request element is placed into the seek queue according to seek address. (If the seek queue is empty, the element is placed as the first element in the primary queue. If the request is related, it is placed at the end of the related queue.)

The current cylinder address, specified by the last seek address field in the UCB for the device, is compared to the seek address specified for the I/O request in the IOB. If the current cylinder address is lower than the seek address, the primary queue is searched for the point at which the element should be placed to maintain order. If the current cylinder address is higher than the seek address, the secondary queue is searched. When the correct point is found, the element is queued. Control then returns to the task supervisor.

Ordered Seek Dequeue Module

After a stand alone seek has completed for a device that specifies ordered seek queueing, the ordered seek dequeue module dequeues the request element for the stand alone seek request from the seek queue. (Stand alone seeks are explained in "Direct-Access Start I/O Module.") The module removes the request element from the primary queue of the seek queue and, if necessary, switches a related request element or the secondary queue (or both) to become the primary queue. Thus, the channel search module of the I/O interruption

supervisor need refer to only one field within the seek queue control word while obtaining a seek request to restart the channel.

The ordered seek dequeue module first tests to determine if the primary queue is empty. If so, it switches the secondary queue to become the primary queue by replacing the address of the primary queue in the seek queue control word with the address of the secondary queue. Thus, a new secondary queue must be started and the next stand alone seek to be executed causes the arm to move back to a lower cylinder address and to begin a new pass of the face of the device.

After the secondary-to-primary queue switch is made or, if the primary queue was not empty and no switch was made, the seek address for the first request in the related queue is compared with the current cylinder address and with the seek address for the first element in the primary queue. If the related seek address is between the current cylinder address and the primary queue seek address, the module moves the first request from the related queue to the top of the primary queue. In this manner, related requests are executed where appropriate within the ordered pass of the arm.

Control returns to the post-start I/O subroutine.

SELECT SUBROUTINE

The select subroutine selects the appropriate device-dependent start I/O module and then passes control to it, except when an error exists from a previous I/O request.

This subroutine is entered from both major program sections of the I/O supervisor. The EXCP supervisor uses the select subroutine to start an I/O request when the requested device is available and the test channel module has located an available physical channel for the device. The I/O interruption supervisor uses the select subroutine to restart a channel with a previously queued I/O request.

If the I/O request is for a burst device attached to a multiplexor channel, it must be determined if the I/O request can be started without causing overrun on the other devices attached to the multiplexor channel. The procedure that determines this is described in Part II: Multiplexor Channel with Burst Devices Attached.

When it is determined that the I/O request may cause overrun, control is given to an enqueue module and the request element is queued. When the I/O request can be started, control returns to the select subroutine where the proper device-dependent start I/O module is selected.

Errors from previous I/O requests are indicated when the UCB intercept flag is set or when the DCB specifies that a permanent error is associated with a related I/O request.

UCB Intercept Flag: The UCB intercept flag indicates that the IOB associated with the next I/O request on the device is to be intercepted for error correction procedures. (The error occurred after channel end on the previous I/O request for the device. Correction was not attempted then because the request element had been returned to the freelist.) When this flag is set, the select subroutine intercepts the IOB for the current request by storing the CSW status bytes, contained in the UCB for the device associated with the error, into that IOB. The error routine interface of the I/O interruption supervisor is then entered and an error routine is scheduled, so that the current request may be tried for error correction.

Related Request: A related I/O request is indicated when the unrelated flag in the associated IOB is not set. A related request is started only if a previous related I/O request has successfully completed.

The associated DCB is inspected for a permanent error condition. If present, the related I/O request is posted complete with a permanent error indication. All subsequent requests related to this I/O request are posted complete with a purge error indication. (The IOBs for the I/O requests are not held in the DEB purge chain. The request elements are returned to the freelist.) Control returns to the task supervisor.

START I/O MODULES

When a device and channel are both available for an I/O request, a start I/O module is entered. It establishes the prerequisite device-dependent commands (set mode command for tape, set file mask command for direct-access devices, etc.), the CCW address fields for the channel address word (CAW), and the CCWs necessary for the operation of an I/O device. The start I/O subroutine is then entered to issue the SIO instruction.

Module Selection: There is one start I/O module for each device class in the system. The address of the appropriate start I/O module is obtained by the select subroutine. It adds the device table indexing value contained in the UCB that represents the device to the starting address of the device table. The entry in the device table at the resultant address contains the start I/O module address.

Unit Record and Telecommunications Start I/O Module

The unit record and telecommunications start I/O module is used for all I/O requests on unit record equipment and telecommunications devices. It loads a general register with either the IOB start or the IOB restart address field, depending on the setting of the IOB start/restart flag. Control is then given to the start I/O subroutine where the CAW is loaded with the contents of the register.

However, when an immediate operation is required by an error routine, control goes directly to the start I/O subroutine.

Tape Start I/O Module

The tape start I/O module is used for all I/O requests on the 2400 tape series. This module prepares the following command chained CCWs, contained within I/O supervisor storage, for execution by the start I/O subroutine:

- First CCW. A set mode command, which is specified in the modifier byte of the DEB extent by the open routine, is placed in the command code field. This command sets the mode (parity, density, etc.) of operations on the tape unit. (If 9-track, the modifier byte will contain a NOP.)
- Second CCW. The command code field contains a transfer-in-channel (TIC) command. The data address field is loaded with the IOB start or restart address.

Control passes to the start I/O subroutine.

Tape Unit Positioning: When repositioning (backspacing, forward spacing, erasing) is necessary for error recovery procedures, control passes immediately to the start I/O subroutine where a stand alone CCW is

loaded with the repositioning command code. The code is specified in the IOB modifier byte by the 2400 tape series error routine.

Cyclic Redundancy Check Correction: When cyclic redundancy check (CRC)¹ correction is required, the tape start I/O module loads the first CCW command code field with a request track in error operation code (hexadecimal 1B). It also places the third sense byte, which contains the CRC bits, into the CCW for later transfer to the tape adapter unit (TAU).

Direct-Access Start I/O Module

The direct-access start I/O module is used for all I/O requests on direct-access devices. The extent limits are checked; if they will be violated, the end-of-extent appendage exit is taken. If the requested appendage has movable access arms, a stand-alone seek and a subsequent start data transfer seek are used to start the channel programs. If the request is for a device such as the 2301 drum storage unit (which has a fixed access mechanism) the stand-alone seek is not necessary, and only the start data transfer seek is issued.

Stand Alone Seek: The stand alone seek is a preliminary seek command that is necessary to position the access arm before a channel program for a direct-access device can be initiated. The stand alone seek starts the access arm seeking for one I/O request while allowing other I/O operations on the same physical channel.

The stand alone seek is issued by means of a stand alone CCW contained within I/O supervisor storage. The module loads the CCW with the seek command code and with the seek address that is provided by the user within the IOB. The file mask at this time allows all seeks, having been automatically reset following the last operation.

Before issuing the stand alone seek, the direct-access start I/O module ensures that all user-specified seek addresses are within the extent limits indicated by the DEB that is associated with the data set to be operated upon. The module stores the IOB seek address into the UCB for later start data transfer seek procedures and then indexes the extent area of the DEB with the M EXTENT field of the IOB to locate the correct extent.

The module checks the IOB seek address to determine whether or not it is within

¹Information regarding a CRC error is contained in the IBM 2401, 2402, 2403, 2404, and 2816 Model I: Principles of Operation publication, Form A22-6866.

the limits of that extent. If not, an end-of-extent appendage is entered.

If the seek address is within the extent limits, the direct-access start I/O module loads the address of the stand alone CCW into the CAW. Control passes to the start I/O subroutine, which enters the start I/O appendage and then issues the stand alone seek to position the access arm for the direct-access device.

If a CSW is not stored upon return from the start I/O subroutine, the module effects a loop with a test I/O (TIO) instruction until the CSW is stored. The stored CSW indicates that channel end has occurred and that the seek address has been transferred to the head register of the control unit. The module sets the disk arm seeking flag in the UCB to indicate that the direct-access device is involved in a seek operation and, if necessary, enters the dequeue module to dequeue the request element from the seek queue.

The address of the request element for the I/O request is placed into the UCB until device end for the seek operation is received. At this time, the request element is enqueued in a logical channel queue for subsequent data transfer procedures.

Start Data Transfer Seek: Device end for a stand alone seek indicates that the access arm is in position for data transfer operations and that the channel program may be initiated. The start data transfer seek is issued when the channel search module within the I/O interruption supervisor determines that the associated physical channel and direct-access device are available. The start data transfer seek initiates the channel program.

To issue the start data transfer seek, the direct-access start I/O module prepares the following command chained CCWs with the proper command codes and data addresses:

- First CCW. The command code field contains a seek command. The data address field contains the address of the seek address that was stored in the UCB during the stand alone seek operation.

When the command is executed, the seek address overlays the control unit head register. The head register must be reinitialized to the proper seek address because the seek address from the stand alone seek associated with this channel program might have been destroyed by subsequent seek commands to other direct-access devices on that control unit.

- Second CCW. The set file mask specified in the DEB, which describes the types of write and seek commands that

can be initiated on this data set, is loaded into this CCW.

- Third CCW. The command code field contains a TIC command. The data address field is loaded with either the IOB start or the IOB restart field, depending on the setting of the IOB start/restart flag.

When the command is executed, control passes either to the user channel program or to an error recovery channel program.

The direct-access start I/O module sets the data transfer flag in the UCB, and branches to the start I/O subroutine, which initiates the CCW execution. The results of channel program execution are handled by the post-start I/O subroutine.

End-of-Extent Appendage

Entry into the end-of-extent appendage is made by means of the appendage vector table. Upon return, the appendage specifies one of the following:

- To disregard the error and try again.
- To skip execution of the request and have it posted complete.
- To indicate a permanent error, enter the abnormal end appendage, and then post the request complete.

START I/O SUBROUTINE

The start I/O subroutine issues all SIO instructions for the start I/O modules.

When tape and unit record start I/O modules require immediate operations or unit repositioning, the start I/O subroutine places the operation code contained in the IOB modifier byte into a stand alone CCW for execution. It places the address of that CCW into the CAW.

The start I/O appendage exit is then taken before the SIO instruction is issued. This exit is not taken, however, when a start data transfer seek is to be issued or when an error routine is in control. The start I/O subroutine issues the SIO instruction directly.

Execution of the SIO Instruction: The start I/O subroutine issues the SIO instruction after loading the CAW with the

address of the starting CCW and after loading the SIO instruction with the unit address contained in the UCB.

After execution of the instruction, the start I/O subroutine stores the PSW condition code, the instruction length code, and the program mask into the IOB and stores the unit and channel address into the UCB. Also, the UCB busy and post flags are set. When the SIO instruction is accepted or when the CSW is stored, as indicated by condition code settings of 0 and 1, control passes to the post-start I/O subroutine.

When the path to the device is busy, as indicated by a condition code setting of 2, the request element is queued. If the device is unavailable, as indicated by a condition code setting of 3, the request element is dequeued and the task is abnormally terminated. The I/O request is not posted complete.

Start I/O Appendage

A start I/O appendage is given control preceding the execution of the SIO instruction for an I/O request. For direct-access devices, the appendage receives control preceding the execution of the SIO instruction for the stand alone seek. Entrance to a start I/O appendage is by use of the appendage vector table. Its address is placed in the DEB by the open routine.

The start I/O appendage returns control to the EXCP supervisor with the indication to continue with normal processing or to skip processing of the request. If the result is skip, control is given to the dequeue module and the I/O request is not posted complete.

POST-START I/O SUBROUTINE

The post-start I/O subroutine determines the results of the SIO instructions issued by the start I/O subroutine. It analyzes the condition code and, if the CSW is stored, the CSW status bits.

SIO Instruction Accepted: A condition code setting of 0 indicates the acceptance of the SIO instruction. This implies that the channel program is successfully started. The request element is dequeued from the logical channel queue and its address is placed in the UCB.

CSW Stored: When the CSW is stored, indicated by a condition code setting of 1, the CSW status bits may indicate the following:

- Catastrophic channel errors. When catastrophic channel errors (i.e., channel control, channel data, or interface control checks) occur, control goes to the SER interface and the entire system becomes inoperative.
- Busy. If accompanied by control unit end, the I/O request is retried. If not, the request element is enqueued.
- Channel end. The operation was an immediate command. The I/O request is posted complete and, if device end is also on, the UCB busy bit is cleared.
- Program or Protection Check. The abnormal end appendage is entered and, if the check is not reset, the task is abnormally terminated. The I/O request is not posted complete.
- Unit check. The sense subroutine issues a sense command, the abnormal end appendage is entered, and then the error routine interface is entered.
- Attention. An attention interruption for the device may be brought in. The attention routine interface is given control.

PART II: I/O INTERRUPTION SUPERVISOR

The I/O interruption supervisor receives control after an I/O interruption from the I/O FLIH. The I/O interruption occurs either in response to channel program execution or because of operator intervention. The conditions that cause the I/O interruption are indicated in the CSW.

When the I/O interruption supervisor receives control, the current PSW is in the supervisor and running states, has the supervisor protection key of zero, and has all interruptions, except machine check interruptions, masked as disabled.

After operation of the I/O interruption supervisor is complete, control returns to the I/O FLIH, which determines whether control is to be given to the dispatcher or to the interrupted routine.

Chart 04 shows the control flow among the subroutines and modules within the I/O interruption supervisor.

GENERAL OPERATING PROCEDURE

The I/O interruption supervisor finds the address of the UCB that represents the interrupted device by means of the UCB lookup routine. This routine uses an algorithm with the values contained in the UCB lookup table to find the UCB, which in turn contains the address of the active request element for the interrupted I/O request. (In cases where the request element was already returned to the freelist, such as when an I/O interruption occurs because of an error after channel end, there is no active request element.) From the UCB and the request element, the pertinent control blocks, tables, and modules are located.

The I/O interruption supervisor analyzes the CSW status bits to determine the reason for the I/O interruption and then proceeds to service it. For example, when catastrophic channel failures (channel data check, channel control check, interface control check bits) occur, the SER interface is entered so that the system environment can be recorded; the system becomes inoperative. When the PCI bit is set, a PCI appendage is entered. When the attention bit is set, the attention routine interface is entered and control is given to an attention routine. For a unit check condition, the sense subroutine is entered; this

subroutine issues a sense command and reads additional information regarding the I/O interruption into the sense field of the UCB. If an error or unusual condition occurs (the I/O request did not complete normally), the error routine interface is entered; this routine determines whether or not an IBM-supplied error routine should be scheduled for the condition.

When the I/O request completes normally (indicated by a channel end or device end, or both, interruption), a trapcode module is entered. The device table is used to select the correct trapcode module for the type of device. For direct-access devices, the trapcode module determines if enqueueing of the request element on a logical channel queue is necessary. For tape, the trapcode module updates the DCB block count. For other devices, a trapcode module only facilitates device independence.

The I/O request is posted complete by the task supervisor's post routine, which is entered by means of the I/O interruption supervisor's post routine interface. The type of completion (error, normal, etc.) is specified in the ECB.

When the I/O request is complete and the channel is free, the I/O interruption supervisor attempts to restart the channel. The channel is enabled for interruptions, and any I/O interruption that is pending is accepted and processed. If no I/O interruption is pending, a channel search module is entered.

A channel search module for a particular channel is selected by means of the channel address and the channel table. The module searches the seek and logical channel queues associated with the channel for a request element that represents an I/O request that can be started on the channel. The channel search module is re-entered until a request that can be started is found or until it is determined that there are no such requests available.

The I/O FLIH regains control when the channel is either restarted (by the start I/O routines described in Part I) or it is determined from the queues that there is no request that can be started.

OPERATION OF THE SUBROUTINES AND MODULES

The following subroutines, routines, and modules are used to perform the functions of the I/O interruption supervisor:

- UCB lookup routine.
- I/O interruption analysis.
- Trapcode modules.
- Channel restart subroutine.
- Channel search modules.
- Sense subroutine.
- Interfaces.
- I/O purge routine.

UCB LOOKUP ROUTINE

The UCB lookup routine locates the address of the UCB for the device associated with the I/O interruption by use of the 11-bit I/O address, stored in the I/O old PSW, in conjunction with the values in the UCB lookup table. The I/O address consists of the channel, control unit, and device addresses associated with the I/O interruption. The UCB lookup table, consisting of the channel, control unit, and UCB address list portions, is described in detail in Part V: UCB Lookup Table.

The I/O interruption supervisor locates the UCB for all types of I/O interruptions except for catastrophic errors and for the setting of the CSW control unit end bit alone. The algorithm shown in Figure 7 is used to obtain the address of the UCB.

The UCB contains the address of the active request element (representing the I/O request for which the I/O interruption occurred), which in turn contains the addresses of the associated IOB and DEB. The DEB contains the address of the DCB. The addresses of these control blocks are placed into general registers, which are not changed during the processing of the I/O interruption.

If the I/O interruption is such that the I/O request has been previously posted complete, there is no active request element; the element has already been returned to the freelist. This occurs for devices that have channel end and device end status presented separately.

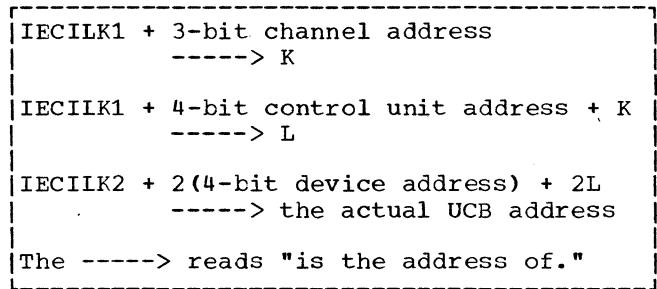


Figure 7. UCB Lookup Algorithm

IECILK1
is the starting address of the UCB lookup table.

IECILK2
is the starting address of the UCB address list portion of the UCB lookup table.

Note: The addresses of both the IECILK1 and the IECILK2 are contained in the CVT.

K
is the index value obtained from the channel portion of the UCB lookup table.

L
is the index value obtained from the control unit portion of the UCB lookup table.

Actual UCB address
is the 2-byte address of the UCB associated with the I/O interruption. It is obtained from the UCB address list portion of the UCB lookup table.

I/O INTERRUPTION ANALYSIS

The I/O interruption supervisor analyzes the CSW status bits, stored as a result of an I/O interruption, to determine the cause of the interruption and the processing procedure that must be followed. The CSW status bits are analyzed in the order in which they are described. Charts 05 and 06 show the details of the I/O interruption analysis.

Channel Data Check, Channel Control Check, Interface Control Check Bits

The CSW status bits may indicate the presence of a channel data check, a channel control check, or an interface control

check. All are catastrophic errors. If one is present, the SER interface is entered and the system environment is recorded. A catastrophic error causes the system to become inoperative.

Catastrophic errors are checked, and thus detected, immediately upon entry into the I/O interruption supervisor. If such an error exists, the other CSW status conditions may not be accurate.

Control Unit End Bit

The setting of the control unit end bit alone, (without a channel end, device end, or unit check bit setting) is a response to the "control unit busy" condition (busy with status modifier) that is presented when the control unit is interrogated while executing an operation. It may occur when a control unit that is shared by two or more I/O devices or channels becomes free. The channel restart subroutine is given control.

PCI Bit and PCI Appendage

If the PCI bit is set, the PCI appendage supplied by the user of the I/O supervisor is entered. Because a PCI condition may ride in with other I/O interruptions, channel status conditions (program, protection, chaining checks) as well as unit status conditions are inspected upon return from the appendage. If no additional status conditions are present, control returns to the I/O FLIH. If status conditions are present, analysis is continued and the conditions are handled.

Channel End Bit and Channel End Appendage

The channel end bit indicates the completion of an I/O request. The I/O interruption supervisor moves the CSW status to the IOB for subsequent examination by the user of the I/O supervisor. The channel end appendage exit is then taken.

The appendage returns control to the I/O interruption supervisor and specifies one of the following:

- Continue normal processing.
- Skip further processing of request.
- Reschedule request.
- Ignore request entirely.

The I/O request is posted complete when the normal processing return is specified.

When the channel end bit is not set, but the UCB post flag is set, it is an indication that channel end was suppressed either because of command chaining in a channel program or because an error occurred at channel end prior to this interruption. The UCB busy and the UCB post bits are turned off, and the CSW status is moved to the IOB for the user. If there are any errors, the sense command is issued and an error routine is scheduled. If no errors are present, but the device end bit is set, the channel end appendage exit is taken, as above. If the device end bit is not set, the channel restart subroutine is entered.

Device End Bit

Device end is caused either by the completion of an I/O operation at the device or by the manual change of the device from the not-ready to the ready state. The following conditions may be present with device end:

- Device end with the UCB busy flag set, which indicates that an I/O device has completed its most current operation. The UCB busy flag is turned off and the operational status of the associated channel is interrogated.
- Device end alone (without the UCB busy or post flags set), which indicates that the I/O interruption is an unsolicited device end caused by the transition from the not-ready to the ready state. The attention routine interface is entered.
- Device end with the UCB post flag set, but with no channel end bit set, which indicates either that channel end has been suppressed by command chaining or that an error occurred at channel end. The handling of these conditions is described under the "Channel End Bit."
- Device end with the UCB busy flag set, with the UCB post flag not set, and with a unit check or unit exception status, which indicates that an error occurred at device end. The UCB intercept flag is set to indicate that the error must be inspected at the next I/O request for the device.

Attention Bit

When the attention bit is set, the attention routine interface is entered.

Unit Check Bit and Abnormal End Appendage

When the unit check bit is set, the sense subroutine is entered so that additional information about the I/O interruption may be obtained. The unit check bit usually occurs with channel or device end bit settings (or both) or during the initial selection of the device.

The I/O interruption supervisor sets the IOB exception flag to indicate an error may be present, and takes the abnormal end appendage exit. The appendage returns to the I/O interruption supervisor and specifies one of the following:

- Continue normal processing.
- Skip further processing of request.
- Reschedule request.
- Enter IBM-supplied error routine.

When an error routine is entered, error correction is attempted; however, if the error routine determines that the error is permanent, the abnormal end appendage is re-entered. The preceding returns are applicable.

Unit Exception and Incorrect Length Bits

The unit exception and incorrect length bits indicate that a device-dependent condition is present. The IOB exception flag is set to indicate that an error may be present, and when a channel end condition exists, the channel end appendage is entered. If the channel end bit is not set, the abnormal end appendage is taken. When control returns to the I/O interruption supervisor, the error routine interface is entered, unless the appendage specifies otherwise.

Program Check, Protection Check, and Chaining Check Bits

When the chaining check status bits are set, the abnormal end appendage exit is taken. Upon return, the error routine interface is entered to schedule an error routine that will inspect the type of

status bit indicated, unless the appendage specifies otherwise.

When the program check or the protection check status bits are set, the abnormal end appendage is entered. If the appendage does not change the setting of the bits and specifies to continue normal processing, an error routine is scheduled. The ABTERM routine interface is subsequently entered and the task associated with the I/O request that developed the error is abnormally terminated.

Status Modifier and Busy Bits

The status modifier and the busy bits are set to indicate a control unit busy condition. (See "Control Unit End Bit.") The UCB is located by means of the UCB lookup routine and the UCB control unit busy flag is then set.

TRAPCODE MODULES

A trapcode module, or the capability for it, exists for each device class. The trapcode modules provide device-dependent functions for channel end and device end interruptions. Upon return, either the channel status is interrogated or the channel end appendage exit is taken, depending on the settings of the UCB post flag.

Module Selection: The appropriate trapcode module is selected by use of the device table, which contains the address of the trapcode module to be used with a particular device class.

Unit Record and Telecommunications Trapcode Modules

The unit record and telecommunications trapcode modules perform no functions. A branch is made back to the I/O interruption analysis.

Tape Trapcode Module

The tape trapcode module updates the block count in the DCB using the block count increment amount specified in the IOB.

Direct-Access Trapcode Module

The direct-access trapcode module determines whether or not a seek request element should be queued in a logical channel queue.

At the completion of a stand alone seek, indicated by a device end interruption, the representative request element is enqueued on the logical channel queue associated with the device. The subsequent start data transfer seek initiates the data transfer operations for the request. However, when either the EXCP supervisor or an error routine is in control and the results of the TIO instruction specify a channel end and device end to indicate completion of a stand alone seek, the start data transfer seek is given immediately.

If a seek end has not occurred, the module returns directly to the I/O interruption analysis. The "both on" or "both off" settings of the disk arm seeking and disk data transfer flags in the UCB for the interrupted device indicate that there is no seek end. When both flags are on, data transfer is taking place.

CHANNEL RESTART SUBROUTINE

The channel restart subroutine attempts to restart the channel after an analysis of the I/O interruption determines that the channel is available for processing.

The available channel is first re-enabled by use of the channel mask contained in the entry for that channel in the channel table. If a pending I/O interruption is accepted, the I/O interruption supervisor handles it. Otherwise, the channel is disabled and a channel search module is entered.

The channel search module returns with the address of either a request element that was previously queued in a logical channel queue associated with the channel, or with the address of a UCB that represents a direct-access device attached to the channel. The UCB contains the seek queue control word, which provides the address of a request element previously queued in the seek queue for a direct-access device.

If the request element contains a dummy address, there is no request element queued for the available channel and the task supervisor regains control. However, if the request element does not contain a dummy address, the associated UCB data

transfer, busy, and not ready flags are inspected. The channel search module regains control if any of the UCB flags are on to indicate that the device is not available.

When the device is available, control is routed to the select subroutine, where start I/O procedures are followed to issue an SIO instruction. The select subroutine returns with a "try again" or "accepted" decision.

If the decision is a try again, the channel search module is re-entered. If it is an accepted decision, the channel is tested for operational status and, depending on whether or not it is busy, control returns to the I/O FLIH or to the channel restart subroutine, respectively.

CHANNEL SEARCH MODULES

The channel search modules supply the addresses of previously queued request elements for a particular channel to the channel restart subroutine until an I/O request is found that can be started on the channel or until it is determined that there is no such I/O request. There is a channel search module for each of the following:

- Multiplexor channel with burst devices attached.
- Multiplexor channel with no burst devices attached.
- Selector channel included in only one logical channel.
- Selector channel included in two or more logical channels.

Module Selection: The channel restart subroutine multiplies the address of the available physical channel by four and then adds it to the starting address of the channel table. The entry at the resultant address contains the address of the channel search module to be used for that channel.

Multiplexor Channel with Burst Devices

Whenever a burst device is operating on a multiplexor channel, that channel is tied up and other devices cannot be serviced. Because of this, those devices may overrun.

When data is transferred to or from an unbuffered control unit, overrun can occur

if the channel fails to respond on time to a request for service from a device. The data (input or output) is considered invalid. Overrun also occurs during command chaining when a device receives a command that is too late for the data.

The devices are classified within the UCB as being burst, overrunable byte, or non-overrunable byte. The following rules are applied to their operation:

- Only one burst device can be operated at one time.
- A burst device and an overrunable byte device cannot operate at the same time.
- More than one overrunable byte device can operate at the same time.
- A non-overrunable byte device can operate at any time.

Burst Device: Whenever a request is for a burst device, or whenever a request element representing a request for a burst device is the first element in the logical channel queue, it must be determined whether or not any other burst devices or an overrunable byte device are operating. If they are, the request is not started and control is returned to the task supervisor. No other request elements in the logical channel queue are inspected until that burst request is started.

The request is started if no other burst device or overrunable byte device is operating. A flag byte is then set to 'FF' to indicate that a burst request is operating. (The flag byte is reset when the request is completed.) Either the channel restart subroutine is entered with the request element for the burst request or control is returned to the select subroutine.

Overrunable Byte Device: Whenever a request element for a request on an overrunable byte device is the first request in the logical channel queue, it must be determined whether or not any burst devices are operating. If none are, the channel restart subroutine is entered and the request is started. A counter is incremented every time an overrunable byte device is started and decremented every time an overrunable byte device terminates its operations.

If a burst device is operating, control is returned to the task supervisor. No other request elements are inspected until that request is started.

Non-Overrunable Byte Device: Whenever the request element for a non-overrunable byte device is the first element in the logical

channel queue, the request is started regardless of the types of devices operating on the multiplexor channel.

Multiplexor Channel with No Burst Devices

The address of the first request element in the logical channel queue associated with the multiplexor channel is passed to the channel restart subroutine. If the request can be started, the select subroutine is given control. Otherwise, the module is re-entered.

When the addresses of all request elements queued in the logical channel queue have been passed to the channel restart subroutine or when none of the I/O requests represented by the request elements could be started, control returns to the task supervisor.

Selector Channel Included in Only One Logical Channel

To start seek overlap, this module passes to the channel restart subroutine the address of a UCB that represents a direct-access device attached to the available physical channel. The UCB contains the seek queue control word, which indicates the first request element in the associated seek queue.

The channel restart subroutine inspects the UCB disk arm seeking flag and, if the arm of the device is busy, returns control to this module for the address of the next direct-access device's UCB. If the arm is not busy, the first seek is initiated. The module is then re-entered until all possible seek requests for the channel within all the seek queues are initiated.

After all seeks have been initiated so that seek overlap is started, a logical channel queue for the channel is searched until a data transfer request is started on the channel, or until it is determined that there are no requests that can be started. Control returns to the task supervisor.

Selector Channel Included in More than One Logical Channel

This module starts seek overlap in the same manner as the module for a selector channel included in only one logical channel, described above. It then starts the

highest priority request in each logical channel queue for the available selector channel. (When priority queueing is not implemented, the address of the first request element is passed to the channel restart subroutine.)

To determine the highest priority request elements, the module places the 2-byte link fields from the first request elements in each logical channel queue for the selector channel into the scratch field of the logical channel word for the associated queue. The scratch field maintains the address of the request element within the queue that is next to be compared for priority. The priority of the request elements are then compared.

The address of the highest priority request element is passed to the channel restart subroutine, which determines whether or not the represented request can be started. If it can, the select subroutine is entered. Otherwise, the address of the next highest priority request element is passed. This procedure continues until a request is started, or until all queues have been inspected and it is determined that no request can be started on that selector channel. Control then returns to the task supervisor.

SENSE SUBROUTINE

The sense subroutine is entered when the analysis of the CSW status bits after an I/O interruption shows that the unit check bit is set. The subroutine issues a sense command to read into the UCB additional information regarding the unit check condition for the device in error.

Issuing the Sense Command: Before a sense command can be issued, the sense subroutine must inspect the UCB busy flag to determine whether or not the device represented by the UCB is busy.

If the device is busy, the sense subroutine sets the IOB sense flag to indicate that an error occurred. The I/O interruption supervisor checks the IOB sense flag at device end for the busy device and, if it is on, "ORs" the CSW field from the IOB (stored there at channel end) to the present CSW bits. Thus, the channel end error appears as a channel end/device end error and the sense command is issued.

When the device is available, a CCW that contains the sense operation code is constructed and the SIO instruction is issued. The sense information is read into the UCB sense area. If the initial response to the

SIO instruction is busy, the SER interface is entered. If it is not available, the ABTERM routine interface is entered.

CSW Inspection: The sense subroutine "spins" on a TIO instruction until the CSW is stored, which indicates that the sense data has been transferred to the UCB. The following procedures are taken for the resultant CSW conditions:

- Catastrophic error. The SER interface is entered.
- Unit check. The ABTERM routine interface is entered.
- CSW busy bit. The SIO instruction is re-issued.
- CSW status modifier bit. The UCB status modifier flag is set and control returns to the I/O interruption supervisor.

All other CSW conditions are disregarded, and the first two bytes of sense information is stored in the IOB. Control is returned to the I/O interruption supervisor.

INTERFACES

Interfaces are routines that are used as common points of departure from, and that direct the transfer of control from, an operating system routine to another system routine. The interface positions data in the form required by the system routine to be entered and also provides a return point. The following interfaces are used by the I/O supervisor:

- Attention routine interface.
- Error routine interface.
- SER interface.
- ABTERM routine interface.
- Post routine interface.

Attention Routine Interface

The attention routine interface routes control to the proper attention routine after an attention interruption occurs. (See "Introduction: Attention Table.")

The appropriate attention routine is indicated by the index value to the attention table that is contained in the ATNTAB field of the UCB that represents the device requiring the attention routine. The user sets the index value when execution of an

attention routine that was specified at SYSGEN time is desired. The attention routine interface adds that index value to the starting address of the attention table. The entry at the resultant address contains either the address of the pertinent attention routine or, if the index value is zero, a branch back to the I/O interruption supervisor.

After the attention routine has been executed, control returns to the I/O supervisor.

Error Routine Interface

The error routine interface controls execution of the error routines required by the I/O supervisor for error recovery procedures. It tests the I/O supervisor error key (contained in the DCB) and determines whether or not error recovery procedures are to be used. If a transient error routine must be scheduled, the error routine interface routes control to the exit effector. In the case of a related request with a permanent error, the error routine interface calls the I/O purge routine, so that the related request elements for the data set will be purged.

I/O Supervisor Error Key Test: Providing that the IOB does not indicate a permanent error, in which case the post routine interface is entered, the I/O supervisor error key is tested to determine when IBM-supplied error routines are to be used. They may be used for:

- All errors. For direct-access devices, the error routine interface branches to the resident portion of the direct-access device error routine, which handles the error. For sequential access devices, the error routine interface passes the request element to the exit effector for subsequent scheduling of the appropriate transient IBM-supplied error routine.
- No errors. The error is posted as permanent.

Scheduling of an Asynchronous Routine: If the I/O supervisor error key indicates that an IBM-supplied error routine is to handle an error, the error routine interface branches to the direct-access device error routine or, if the error routine is transient, causes it to be scheduled asynchronously. Chart 07 shows the scheduling procedure.

For direct-access devices, the resident portion of the direct-access device error

routine is entered, and if necessary, the transient portions are called in.

For sequential-access devices, the error routine interface passes the address of the request element in error to the exit effector of the task supervisor. The exit effector places this element into the asynchronous exit queue for scheduling. The exit effector then completes the name of the required error routine by use of the ERRTAB byte in the UCB. This byte, when appended to the name in the system interrupt request block (SIRB) yields the name of the error routine within the SVC library. (If the freelist is empty, the error routine interface returns the request element in error to the next available, and enters the ABTERM routine interface. The associated task is abnormally terminated because in scheduling the required error routine, the contents supervisor uses the EXCP macro-instruction.)

The contents supervisor uses the 8-character name to search the SVC library for the error routine. When found, the error routine is read into the I/O supervisor transient area, and then given control.

The I/O purge routine is called into the I/O supervisor transient area in the same manner as the transient IBM-supplied error routines. The error routine interface passes the request element for the related I/O request to the exit effector, which places it into the asynchronous exit queue for scheduling of the routine.

The constant that is used to complete the name of an error routine to be scheduled asynchronously, which is normally obtained from the ERRTAB byte in the UCB, resides however, for the I/O purge routine, in a location following the UCB lookup table. The address of a location that reflects this change is placed into the UCB address field of the request element. Thus, this location simulates that of the ERRTAB byte of the UCB, and the exit effector uses it to complete the name of the I/O purge routine within the SVC library. When the I/O purge routine gains control, it replaces the UCB address field in the request element with the actual UCB address for the device so that later processing is not affected.

I/O Purge Routine: The I/O purge routine is transient and is called by the error routine interface whenever a related I/O request develops a permanent error condition. (The name of the I/O purge load module is IEC0025E.) All other related I/O requests for the same data set must be purged because they are dependent upon the successful completion of the preceding

related I/O requests, one of which has failed.

The I/O purge routine removes all queued request elements for the related I/O requests from the logical channel and seek queues that are associated with the particular data set extent(s) defined in the DEB. (The correct queues are found by means of the logical channel word and seek queue control word indicated in the UCB.) The I/O purge routine returns the request elements to the freelist, chains the IOBS associated with them to the DEB system purge chain field, and posts the ECB that corresponds to each IOB complete with a permanent error code.

The I/O purge routine returns control to the I/O supervisor via SVC 15 and the abnormal end appendage exit is taken.

SER Interface

The SER interface is entered whenever the I/O supervisor detects a catastrophic error (i.e., when a CSW channel data check, channel control check, or interface control check bit is set). It routes control to the SER routines.

The SER interface places the address of the request element in error into location 280 and places the unit address at location 58 of the I/O old PSW. The hexadecimal code 0F, which indicates an I/O channel failure, is then placed into location 115 of the machine check (MCH) new PSW. The MCH new PSW indicates the wait state and has all interruptions masked as disabled. The SER interface then loads the MCH new PSW. Since one of the SER routines gets immediate control, hardware indicators can also be inspected.

ABTERM Routine Interface

The ABTERM routine interface routes control to the ABTERM service routine of the task supervisor, which abnormally terminates a task.

The ABTERM routine interface is entered from the I/O supervisor whenever the following occur:

- Device or channel not operational.
- CSW program check bit set.
- CSW protection check bit set.
- Invalid DEB or UCB identification field.
- DCB specification error in the DEB.
- Error routine to be scheduled, but no request element.

The I/O request associated with these conditions is not posted complete.

Post Routine Interface

The post routine interface routes control to the post routine of the task supervisor. The post routine interface prepares the 30-bit completion code that describes the manner in which an I/O request has completed and passes it to the post routine, where the code is posted in the ECB for inspection by the user. It also passes the address of the TCB of the task for which the I/O request has completed.

If a WAIT macro-instruction had been issued, the completion code clears the wait bit in the ECB. The post routine clears the wait bit in the request block.

The post routine interface is also used by the purge complete subroutine to post the completion of an SVC purge request (described in Part IV).

An error routine may be entered following a unit check, a unit exception, a wrong length indication, a program check, a protection check, or a chaining check indication in the CSW status bytes. When such a condition exists, the I/O supervisor passes control to the error routine interface, which tests the I/O supervisor error key in the DCB and branches to or has the appropriate error routine scheduled asynchronously.

There are three types of IBM-supplied error routines:

- Device-dependent routines.
- Common routines.
- I/O outboard recording routines.

Device-dependent routines attempt error recovery and correct errors for particular device types according to the standard IBM error recovery procedures (ERP). All of the device-dependent routines, except a resident portion of the error routine for direct-access devices, use the I/O supervisor transient area for execution.

Three routines commonly used by the device-dependent routines are the error interpreter, which is resident, and the write-to-operator and statistics update routines, which are transient.

I/O outboard recording routines maintain external records on the system residence device regarding the operation of each I/O device. These routines are the statistical data recorder and the outboard recorder and are always provided when IBM-supplied error routines are included in the system.

The device-dependent and common routines determine the type of error. When the error is such that the I/O request can be retried, the error routine issues an SVC 15 to retry the channel program, and then returns to the task supervisor by means of the RETURN macro-instruction. As each retry is executed, control returns to the error routine.

When an error is corrected by an IBM-supplied error routine, the routine issues SVC 15 to return control to the I/O supervisor. The ECB is posted with a normal completion code and normal processing continues.

Also, when the IBM-supplied error routine cannot correct the error, the routine

issues SVC 15. Upon receiving control, however, the I/O supervisor purges any request elements related to the request element in error by use of the I/O purge routine. It points to the purged requests with the system purge field in the DEB for the appropriate data set. The request elements are returned to the freelist and the ECB is posted with a permanent error code.

At the completion of error processing (successful or unsuccessful), the error routine, when necessary, updates the statistics table, writes error messages to the operator, and calls the I/O outboard recording routines for external error recording.

DEVICE-DEPENDENT ROUTINES

The device-dependent routines correct errors according to the standard IBM error recovery procedures. The routines are:

- 1052, 2150 error routine, which handles errors on the console typewriter. (The load module name is IGE0000D.)
- 2540/2821 error routine, which handles card read and punch errors. (The load module names are IGE001C and IGE0101C.)
- 1403/1443 error routine, which handles printer errors. (The load module name is IGE0000G.)
- 1442, 2501, 2520 error routine, which handles card read and punch errors. (The load module name is IGE0000E.)
- 2671/2822 paper tape reader error routine, which handles errors on the paper tape reader. (The load module name is IGE0002?.)
- 2400 tape series error routine, which handles magnetic tape errors. (The load module names are IGE0000I, IGE0100I, and IGE0200I.)
- Direct-access device error routine, which handles errors on direct-access devices and performs alternate track procedures. Most of this error routine is resident; the load module for the resident portion is dependent on device type and selection of the track overflow feature, as follows:

IEC23XXB - 2311 only; without overflow
IEC23XXC - 2311 only; with overflow
IEC23XXD - 2311 and/or 2302, 2303;
without overflow
IEC23XXE - 2311 and/or 2301, 2302,
2303, 2314; with overflow
(overflow is a standard fea-
ture on 2301 and 2314)

The load module name for the transient portion is IGE0000A.

- 2250 error routine, which handles errors on the 2250 display unit. (The load module name is IGE0010A.)
- 2260 error routine, which handles errors on the 2260 display station and associated 1053 printer. (The load module names are IGE0010B and IGE0110B.)

GENERAL OPERATING PROCEDURE

The device-dependent routines, all except the resident direct-access device routine, gain control from the contents supervisor. Although each routine has characteristics pertinent to the device type, the programming techniques used for each routine are similar.

The routines set the IOB error indicator bits, examine the sense and CSW status bits stored in the IOB to determine the type of error, and where possible, attempt to recover the error by retrying the channel program by means of SVC 15.

Chart 08 shows the general operating procedure for all device-dependent routines except the 2250 and 2260 error routines. The general operating procedure for these routines is shown in Chart 10.

Setting Error Indicators: All device-dependent routines upon entry, turn the IOB exception bit and error flag on. These flags indicate to the error routine interface that an error routine is in control, and that control must be returned to the error routine until the error is corrected or is determined to be uncorrectable. An uncorrectable error is a permanent error.

When the error is permanent, the IOB exception bit is left on and all error flags and counts are reset to zero. If the DCB exception bits are set to 01 (i.e., indicate that error correction is in progress), the I/O supervisor resets the bits to 11, indicating the permanent error condition. Until the DCB exception of 11 is cleared, all further related requests for the particular DEB are rejected and posted as a permanent error.

When the error is corrected, the IOB and DCB exception flags and the IOB error flags and counts are reset to zero.

IOB Channel Program Description Bit Test:

In all error routines, except the direct-access device, 2671/2822, 1052, 2150, 2250, and 2260 error routines, the IOB channel program description bits are inspected for the type of chaining within a channel program in error so that the proper restart procedure is taken. The direct-access device, 2671/2822, 1052, 2150, 2250, and 2260 error routines always retry the channel program in error from the top of the CCW list (at the first CCW); thus, they handle any type of channel program and chaining is not significant.

The procedure taken for chained channel programs, shown in Chart 08, is as follows:

- Data and Command Chaining. The error routines do not retry any channel programs that have mixed chaining (IOB channel program description bits set to 11); the routines write an error message to the operator and indicate a permanent error by means of the DCB exception bits. SVC 15 is issued so that the request element will be returned to the freelist, and then control is returned to the task supervisor via the RETURN macro-instruction.

- Command Chaining. When a channel program is command chained (IOB channel program description bits set to 01), it must be restarted at the proper CCW address. When a channel program has been retried (indicated by the setting of the entry flag) and the error is corrected, a subsequent error may occur on a different CCW. This error is detected when the address of the erroneous CCW does not equal the restart address in the IOB. In this case, the channel program is restarted at the erroneous CCW address. However, if an error occurs at the execution of the SIO instruction for this retry, the accuracy of the CCW is unpredictable. This error is indicated by the condition code setting of 01. The channel program is again started at the address of the erroneous CCW, which is the restart address in the IOB.

- Data Chaining or No Chaining. When the channel program has data chaining or no chaining (IOB channel program description bits set to 10 or 00, respectively), the channel program is restarted from the top of the CCW list, which is the start address in the IOB.

Determining the Type of Error: The type of error is indicated by the setting of the CSW status bits, and in the case of a unit check condition, the sense bits. The error routines (except the 2250 and 2260 error routines) examine the setting of these bits by means of the error interpreter routine. Instructions within the 2250 and 2260 error routines test the bits; the other device-dependent routines branch to the error interpreter with a list of codes and address constants following the branch instruction.

Each entry in the list is 1 byte in length. The order of the entries depends upon IBM standard error recovery procedures, upon the type of device-dependent routine in control, and upon the meaning the CSW status or sense bits have for the type of device.

Each entry contains a code that corresponds to either a CSW

status bit or a sense bit, and an indexing factor that indicates the point at which the error interpreter is to return to the error routine after determining the type of error.

The error interpreter routine compares each code to the CSW status or sense bit. If the particular status or sense bit is set, the error routine regains control at the point indicated by the address constant.

The error routine then handles the condition denoted by the particular status or sense bit. The following procedures are taken:

- Permanent error. A permanent error is one that either cannot be corrected or fails to be corrected after standard error recovery procedures. A permanent error is indicated by the IOB exception bit. The statistics update and the I/O outboard recording routines are called for error recording and then SVC 15 is issued.

After the I/O supervisor routines have posted the request complete with a permanent error, have entered the ABTERM routine interface in the case of a program or protection check, have purged any requests related to the request in error, and have entered the abnormal end appendage, control returns to the error routine. The error routine uses SVC 3 to return to the task supervisor.

- Operator Intervention. When it is determined that operator intervention is required, the write to operator routine is entered for the writing of the "intervention required" message. SVC 15 is issued, then the RETURN macro-instruction is issued to return control to the task supervisor.
- Recoverable errors. The channel program is retried by means of the ERREXCP macro-instruction, and if the error is corrected, the IOB exception and error flags are set to zero and normal processing continues.

If the error is uncorrectable, the procedure for a permanent error is taken.

Statistics Update: In unit check cases, the statistics update routine is always called to update the error counters in the statistics table and, if a statistics counter overflows, the statistical data recorder is called for external error recording.

DEVICE-DEPENDENT CHARACTERISTICS

Since each device type has different characteristics, the meaning of the CSW unit exception bit, the sense bits, IOB flags 1, flags 2, and flags 3 fields, and the IOB error count field vary according to the type of device. The CSW status bit interpretation and their order of inspection is shown in the coding for the IBM-supplied error routines. The sense bit meanings are shown in Table 2 and are explained in the subsequent error routine sections. The IOB flags 1, 2, and 3 fields and the IOB error count field are also shown in Table 2.

1052, 2150 Error Routine

Special considerations are necessary when the console typewriter has a permanent read or write error. On a permanent read error, the write-to-operator routine is called; it writes a message indicating typewriter failure to read input messages. (See "Common Routines.") When operator intervention is required and a permanent write error has occurred, the console bell is rung by use of the console alarm routine.

The console alarm routine issues a TIO instruction until the console device is free and then issues the SIO instruction that rings the console bell. After doing so, the console alarm routine loads the PSW to put the machine in the wait state.

The bits in the first sense byte (byte 0) have the meanings described below.

- Bit 0: Command Reject
This bit is set when an invalid command code is detected.
- Bit 1: Intervention Required
This bit is set when one of the following occurs:
 - Console is out of forms.
 - Not-ready key is depressed.
- Bit 2: Bus-Out Check
This bit is set when a parity error is detected on a command or a data byte.
- Bit 3: Equipment Check
This bit is set when one of the following occurs:
 - Keyboard parity error.
 - Keyboard-printer compare check.
 - Printer failed to take a mechanical cycle.

Bits 4 - 7: Unused.



Table 2A. Sense Bytes and IOB Flags

Sense Byte Meanings									
Bit Device	Byte 0								
	0	1	2	3	4	5	6	7	
1052, 2150	Cmd Rej	Int Req	Bus Out	Eq Chk					
2540/2821	Cmd Rej	Int Req	Bus Out	Eq Chk	Data Chk		Unusual Cmd		
1403/1443	Cmd Rej	Int Req	Bus Out	Eq Chk	Type Bar	Type Bar		Ch 9	
1442, 2501, 2520	Cmd Rej	Int Req	Bus Out	Eq Chk	Data Chk	Over-run			
2671/2822	Cmd Rej	Int Req	Bus Out	Eq Chk	Data Chk				
2400	Cmd Rej	Int Req	Bus Out	Eq Chk	Data Chk	Over-run	Wrt Cnt Zero	Data Cnvr Chk	
2311/2841, 2302/2303, 2314	Cmd Rej	Int Req	Bus Out	Eq Chk	Data Chk	Over-run	Trk Cond Chk	Seek Chk	
2250	Cmd Rej	Should Not Occur	Bus Out	Should Not Occur	Data Chk	Should Not Occur	Buffer Running	Should Not Occur	
2260	Cmd Rej	Int Req	Bus Out	Eq Chk	Should Not Occur	Should Not Occur	Should Not Occur	Should Not Occur	
2301/2820	Cmd Rej	Int Req	Bus Out	Eq Chk	Data Chk	Over-run		Inval Addr	
Byte 1									
2400	Noise	Unit Status Avail	Unit Status Busy	7 Trk	At Load Point	Wrt Stat	File Prot	Tape Ind	
2311/2841, 2302/2303, 2314	Data Chk Fld	Trk Over-run	End of Cyl	Inv Seq	No Rec Fnd	File Prot	Missing Addr Mrkr	Over-flow Inc	
2250	Light Pen Detect	End Order Seq	Char Mode						
2301/2820	Data Chk In Count	Trk Over-run	End of Cyl	Inval Seq	No Rec Found	File Prot	Svc Over-run	Over-flow Inc	
Byte 2									
2400							Multi Err	Trk Exists	
2311/2841, 2302/2303, 2314	Unsafe		Serial-izer Chk		ALU Chk	Unsel Status			
2250				Buffer Address Register					
2301/2820	Unsafe	Shift Reg Chk	Skew Fail	Ctr Chk	Comp Chk				
Byte 3									
2400	R/W VRC	LRCR	Skew	CRC	Skew Req VRC		Bkwd Status	Compare	
2311/2841	Rdy	On Line	Unsafe	Wrt Safety	On Line	End of Cyl		Seek-Incpl	
2260				Buffer Address Register					
2301/2820	LRC Bit 0	LRC Bit 1	LRC Bit 2	LRC Bit 3	Bit 4	Bit 3	Bit 2	Bit 1	
2302/2841	Rdy	Op	Rd Safety	Wrt Safety	On Line			CE Cyl	
2303/2841	Rdy	Op	Rd Safety	Wrt Safety	On Line				
2314/2841	Busy	On Line	Unsafe	Wrt Safety	Pack Change	End of Cyl		Seek Incmpl	

IOB Flags								
Bit Device	Flags 1							
	0	1	2	3	4	5	6	7
1052, 2150	Channel Program Description	Err Flag	Mode 1			IOB Ex		Restrt Flag
2540/2821	Channel Program Description	Err Flag				IOB Ex		Restrt Flag
1403/1443	Channel Program Description	Err Flag				IOB Ex		Restrt Flag
1442, 2501, 2520	Channel Program Description	Err Flag				IOB Ex		Restrt Flag
2671/2822	Channel Program Description	Err Flag				IOB Ex		Restrt Flag
2400	Channel Program Description	Err Flag	Tape Reposition	CRC Bit	IOB Ex			Restrt Flag
2311/2841, 2301/2820, 2302/2303, 2314	Channel Program Description	Err Flag				IOB Ex		Restrt Flag
2250	Channel Program Interruption	Err Flag				IOB Ex		Restrt Flag
2260	Channel Program Description	Err Flag				IOB Ex		Restrt Flag
Flags 2								
2301/2820, 2302/2303, 2311/2314, 2321/2841				Read Home Address	No End of Ext Test	CCHH Update		
2540/2821								QSAM Acc Method
2400	Halt					Wrt UEX		
Flags 3								
1052, 2150			Wrt Err Cnt	Bus Out Cnt	Cntrl Bit M1	Msg Code	Cntrl Bit M0	Log Out
2540/2821	Ind 1	Entry Flag	Rd Err	Bus Out Cnt	Punch Retry	Msg Code	QSAM Cnt	Log Out
1403/1443			Parity Error	Bus Out Cnt		Msg Code		Log Out
1442, 2501, 2520		Entry Flag	Rd Err Cnt	Bus Out Cnt	Data Chk	Msg Code	Over-run	Log Out
2671/2822			Eq Chk Count	Bus Out Cnt		Msg Code		Log Out
2400	Ind 1	Entry Flag	Tape Clean Bit	Cont Flag		Msg Code	Count Unit Busy	Ind 2, Log Out
2311/2841, 2301/2820, 2302/2303, 2314		Trk Cond	No Rec Fnd	Bus Out Cnt	Re-store	Msg Code		Log Out
2250						Msg Code		Log Out
2260						Msg Code		Log Out
2321/2841	Ind 1	Trk Cond	No Rec Fnd	Pick Flag	Re-Store	Msg Code	Sweep Flag	Log Out

Table 2B. IOB Error Counts

Byte 4								
2400	Echo Err	Rej Tape Unit	Rd Clock Err	Wrt Clock Err	Dly Cntr Err	Seq Ind C	Seq Ind B	Seq Ind A
2301/2820	Seq Ind 0	Seq Ind 1	Seq Ind 2	Seq Ind 3	Seq Ind 4	Seq Ind 5	Seq Ind 6	Seq Ind 7
Byte 5								
2311/2841 2302/2303 2314	Command in progress when overflow incomplete occurs or Zero							
2301/2820	X	X	X	X	X	X	X	X

IOB Error Counts

Device	Bit								Bit							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
2400	Bus Out	Overrun		Chain Check		Control Count			Wrt Ind	Read Data Check			Write			
2311/2841 2301/2820	Seek Err Count			Data Check Count					Wrt Ind	Overrun Cnt	Chain Chk Cnt		No Record Found Count			
2540/2821	X	X	X	X	X	X	X	Data Chk	X	X	X	X	X	X	X	X
2250	Bus Out 1	Bus Out 2	Data Chk 1	Eq Chk	2840 In Chk	2840 Out Chk	ABEND App	X	X	X	X	X	X	X	X	X
2260	Bus Out 1	Bus Out 2	Bus Out 3	Err Rtn Ch Pgm	X	X	ABEND App	Eq Chk	X	X	X	X	X	X	X	X

2540/2821 Error Routine

When a card jam causes a punch error indication on the 2540 punch, operator restart procedures are necessary.

No recovery is attempted under program control on a punch error (equipment check) unless the queued sequential access method (QSAM) is in use and requests recovery.

The bits in the first sense byte (byte 0) have the following meanings for the 2540 error routine:

Bit 0: Command Reject

The setting of this bit indicates one of the following permanent errors:

- A read backward or write command was issued to the reader.
- A read backward control other than NOP or a read feed and stacker select command was issued to the punch without the PFR feature.
- A read backward or control command was issued to the punch with the PFR feature.
- An incorrect sequence of instructions was issued.
- A command was received which the device was not designed to execute.

Bit 1: Intervention Required

The setting of this bit indicates a not-ready condition due to one of the following:

- Cards are not at each station (not EOF for the Reader).
- Stacker is full.
- Hopper is empty (not EOF for the Reader).
- Stop key is depressed.
- Chipbox is full or removed.
- Card is jammed.

A message is typed, and the operator must perform the right action. Upon correction, the last operation which was interrupted is repeated.

Bit 2: Bus-Out Check

The setting of this bit indicates a parity error on bus-out on a command or data byte during either initial selection or command execution.

When this error is detected, no punching occurs. The operation is retried once and if it occurs a second time, it is considered a permanent error and a message is written.

Bit 3: Equipment Check

The setting of this bit occurs after one of the following:

- Hole count error.
- Buffer parity error.
- Translate check.
- Address check.

This error applies to the previous card punched for which data was transmitted; it is considered permanent.

Bit 4: Data Check

The setting of this bit indicates an invalid card code was detected. The operation is retried once, and upon recurrence, is denoted as a permanent error.

Bit 5: Unused.

Bit 6: Unusual Command Sequence

This bit is set due to a read following a read with no intervening feed. This error is considered permanent.

Bit 7: Unused.

1403/1443 Error Routine

The bits in the first sense byte (byte 0) have the meanings described below.

Bit 0: Command Reject

This bit is set when one of the following occurs:

- A read backward command is received.
- A carriage instruction to space more than three lines is received.
- A carriage instruction to skip to channel 0, 13, 14, or 15 is received.
- A command is received that the device has not been designed to execute.

These errors are permanent.

Bit 1: Intervention Required

This bit indicates a not-ready condition due to one of the following:

- Forms have run out or jammed.
- Stop key is depressed.
- Cover interlock is open.
- Synch. check.

A message is typed and the operator must correct the condition. When the interruption occurs due to the correction of the error, i.e., the unit goes from not-ready to ready status, the last operation is repeated.

Bit 2: Bus-Out Check

The setting of this bit indicates incorrect parity on bus-out during command-out time during selection or at service-out during data transfer. This error is considered permanent.

Bit 3: Equipment Check

The setting of this bit indicates a program resettable malfunction was detected in the printer or in its controls. The errors are buffer or hammer checks, and are reset by the next write or control command.

Bits 4 and 5: Type Bar Selection (1443 only)

These bits are used in combination and indicate the following:

- 52 character set (00)
- 13 character set (01)
- 39 character set (10)
- 63 character set (11)

These bits can be changed only by repositioning the type bar character indication switch.

Bit 4: Data Check (1403 only)

If the universal character set feature is installed, the setting of this bit indicates that a code in data storage did not match any code generator storage. This is considered a permanent error.

If the universal character set feature is not installed, this bit should not be set.

Bit 5: Code Generator Storage Parity Error (1403 only)

If the universal character set feature is installed, this bit indicates that a parity error was detected on data being written into or read from code generator storage. If it occurs while loading generator storage, a retry is done. If it occurs at any other time, it is treated as a permanent error.

If the universal character set feature is not installed, this bit should not be set.

Bit 6: Unused.

Bit 7: Channel 9

This bit is set when a hole is sensed in channel 9 of the carriage control tape during execution of the last write or control command.

1442, 2501, 2520 Error Routine

The bits in the first sense byte (byte 0) have the meanings described below.

Bit 0: Command Reject

This bit is set when a read backward command is issued or a command is received which the device has not been designed to execute. The error is considered permanent.

Bit 1: Intervention Required

This bit is set when one of the following occurs:

- Power is off.
- Cards are not registered at the read station.
- Stacker is full.
- Feed check light is on.
- Stop key is depressed.
- Chip box is full or removed.
- No cards are in hopper and end of file is not on.
- Cover interlock is open.

A message is typed, and the operator must correct the condition. Upon correction, the last operation that was interrupted is repeated.

Bit 2: Bus-Out Check

This bit is set when there is a parity check on bus-out during command or data bytes, but not during the time period of address-out or command-out proceed.

On a command byte, one retry is made; if upon return the error is not corrected, it is considered permanent.

On a data byte, the card is already punched; the error is considered permanent.

Bit 3: Equipment Check

This bit is set at the following situations:

- Punch operations - the punch echo is checked for invalid card codes; i.e., rows 1 through 7 have more than one punch in one column.
- Read or punch operations - equipment check detects an error from the 9 bit match. The invalid card code part of equipment check is not set during card image punch.

Bit 4: Data Check

The setting of this bit indicates an invalid card code was detected during a read operation; i.e., rows 1 through 7 have more than one punch in any one column. (This is not indicated when

the read command is in data mode 2.) This error pertains to the card for which data was sent.

A message is typed, and the operator must reposition the cards for one retry. If the error persists, it is considered permanent.

Bit 5: Overrun

This bit is set during a read operation if the interface has not trans-

mitted the last previous character by the time the next character is to be read into the register.

The operator must reposition the cards and the operation is retried once. If the error persists, it is considered permanent.

Bits 6 and 7: Unused.

2671/2822 Error Routine

Because the paper tape reader cannot be backspaced or repositioned under program control, all unit check errors except a bus-out check are considered permanent. The bus-out check is retried once, but if not corrected, is then considered permanent.

The sense bits in the first sense byte (byte 0) have the meanings described below.

Bit 0: Command Reject

This bit is set for the 2671 Paper Tape Reader when a read backward or a write command is received at the tape reader or valid commands have invalid modifier bits.

A parity error on the command prevents these conditions from being detected.

Bit 1: Intervention Required

This bit is set when the device is not-ready. This is best described by stating the conditions for ready. The device is considered ready when:

- Power is on.
- Tape is properly loaded.
- Start key is pressed.

The 2671 Paper Tape Reader becomes not-ready immediately after a read operation has been initiated when:

- There is an overstep condition (the tape did not stop on one character) on a normal or abnormal tape stop.
- Tape jamming occurs in the take-up stacker (special feature). The tape transport mechanism is immediately released.
- The end of tape is detected at the reading station (with EOF light off).
- The stop key has been depressed during the read operation and a normal or abnormal completion has occurred.

As soon as the device is restored from the not-ready conditions back to a ready condition, a device end interruption is generated to inform the program of this transition.

Bit 2: Bus-Out Check

This bit is set for the 2671 Paper Tape Reader when a parity error has been detected for a command issued by System/360. A transmission error or a command byte is thus separated from a

programming error as an invalid command.

Bit 3: Equipment Check

This bit indicates that a time-out has been detected in the tape reader since the last data byte was sensed at the tape read station. This time-out of approximately 1 second can be caused by:

- A failure of the photo cells or the light projection system.
- A tape drive motor failure.
- An unusual tape creeping condition.
- A length of tape without punching.

Bit 4: Data Check

This bit is set for 2671 Paper Tape Reader when an invalid character is read. Parity of the data to be read can be checked if the parity switch is set to either the odd or even position. Regardless of the tape code used, the data sent to the system is in odd parity.

2400 Tape Series Error Routine

Minimum tape record lengths are 12 bytes for a read operation and 18 bytes for a write operation. This minimum length has been established to distinguish noise records from data records. There is no check for noise unless there is a data check.

The maximum block size on tape is limited to 32,000 bytes.

The bits in the first sense byte (byte 0) have the meanings described below.

Bit 0: Command Reject

The setting of this bit is caused by:

- A write, write tape mark, or erase command is addressed to a file protected tape unit.
- A "data converter on" mode set command is recognized on a control unit without the data converter feature.

This error is considered permanent.

Bit 1: Intervention Required

This bit is set when the addressed tape unit is not-ready or when the addressed tape unit is nonexistent.

The status of the addressed unit is determined by the setting of bits 1 and 2 in sense byte 1. If set, a message is typed out to the operator

indicating the type of error. A nonexistent tape address is considered as an unrecoverable error. A console message is typed, and control is given to the error routine interface.

Bit 2: Bus-Out Check

This bit is set when uneven parity appears on bus-out. The error may be caused:

- During the initial selection sequence (command out response).
- During a write or control operation when a byte of data is offered from the channel (on bus-out in response to service-in).

If the cause is the latter case, the tape is repositioned and the operation retried. In the first case, the operation is retried without any repositioning of the tape. In both cases, if the error persists, it is considered to be a permanent error.

Bit 3: Equipment Check

This bit is set due to a machine malfunction. Data transfer and tape motion are indeterminate for all equipment checks, and no attempt is made to recover. Control is passed to the I/O outboard recording routines, the error is indicated as permanent, and control returns to the error routine interface.

The equipment check may occur for the following:

- A difference in overall parity when operating seven-track feature with the translator off (Bit 7, Byte 3).
- A difference in overall parity among three-byte characters on a nine-track write operation with tape control having the seven-track feature and data converter (Bit 7, Byte 3).
- Echo error indicating that the tape unit failed to generate an echo pulse in response to a write pulse (Bit 0, Byte 4).
- Incorrect parity is recognized in the R/W register during a write operation on the 2403/2803 (tape control) without seven-track feature (Bit 7, Byte 3).
- Selected tape unit failed to respond to a set read or set write status when instructed or became not-ready during the execution of a tape motion operation (Bit 1, Byte 4).

- A clocking error occurs (Bit 2, 3, and 4 of Byte 4).

- The tape control failed to keep the proper sequence of controls in operation (Bit 5, 6, and 7 of Byte 4).

Bit 4: Data Check

This bit indicates a parity error on data during a read, read backward, or write operation. It is set at the following situations:

- The R/W register during a read or read backward operation contains incorrect parity (the bit cannot be set after an overrun or receipt of a stop command). (Byte 3, Bit 0).
- Longitudinal parity of a record does not agree with the check character of that record. Occurs during a read, read backward, or write operation (Byte 3, Bit 1).
- Excessive skew encountered during a write type operation (Byte 3, Bit 2).
- Cyclic redundancy check computed during a read or read backward does not agree with the computed check mode created when the record was written (Byte 3, Bit 3).
- Character containing incorrect parity has been read into skew register during a write, write tape mark, or ERG (Byte 3, Bit 4).

On a read or read backward with data check, the record in error is examined for noise (a record less than 12 bytes), unless data chaining is used, in which case the records are not checked for noise. If a noise record is detected, the noise bit (sense byte 1, bit 0) is checked. If it is off, the record is true noise, is ignored, and the operation continues.

If the noise bit is on, the read record is not a noise record, the tape is repositioned, and the operation is re-executed in the same direction as the original read. If the error persists, the operation is retried up to nine times for a total of ten reads per record, at which point the tape is positioned past the tape cleaner (unless the tape is at load point or contains previous noise records), repositioned to its original position, and an attempt is again made to read the tape.

The above process is repeated until the error is corrected, or is repeated

to a maximum of 10 loops or 100 reads. If the error still persists, the affected record is defined as a permanent read error, counted as such, and return is made to the error routine interface after the appropriate operator message is typed. (For nine-track tape operations on read errors, the CRC is used in the attempt to recover.)

If the data check is associated with a write operation, the tape being written is backspaced, one record erased, and then the write operation repeated. This procedure is repeated until the write is corrected or a total of fifteen rewrites have been executed. If the error still persists, it is recorded as a permanent error. The operator is notified, and control passes to the error routine interface.

Bit 5: Overrun

This check can occur only during a read, read backward, or write operation. The bit is set when the control unit requests service but finds the servicing lines busy. This implies that the data has come too fast for the system or channel to handle it. The data transfer is stopped.

When overrun is detected, the tape is repositioned and the command is re-executed until the error is either corrected or retried five times. If the error persists, it is recorded as a permanent error.

Bit 6: Word Count Zero

This bit is set when the command is halted (by the halt I/O instruction) after initial selection but before the transfer of data has taken place. This implies no tape movement. It is also set when a write is issued with a WC of zero.

Bit 7: Data Converter Check

This bit is set during a read operation using a data converter option upon reading module 4 with a remainder other than zero.

Direct-Access Device Error Routine

There are four direct-access device error routines available to the user. The one used depends on the devices in the user's system. The routines are:

1. IEC23XXB - 2311/2841 error routine without overflow.
2. IEC23XXC - 2311/2841 error routine with overflow.

3. IEC23XXD - 2301/2820 (2302, 2303, 2311, 2314/2841) error routine without overflow.

4. IEC23XXE - 2301/2820 (2302, 2303, 2311, 2314/2841) error routine with overflow.

The resident direct-access device error routines attempt to restart the channel program in error from its original starting point, the first CCW. This eliminates the problem of repositioning required by disk, and allows the handling of channel programs that include command and data chaining. Any errors within errors are also handled. For example, if a data check occurs and upon retry a seek check is encountered, the error counts for both the data check and seek check are incremented and recovery is still possible. An unrecoverable error is not indicated until the error is retried the standard number of times.

The handling of track condition check is an exception to the general practice of restarting the channel program at the original start location. In this case, the home address and R0 record on the track is read to determine if the track is a defective or alternate track and to locate the corresponding alternate or defective track.

If it is an alternate track, the address of the defective track, plus one, is used in a seek command. (This is found in the ID field of the R0 record.) The operation is resumed after a search to the desired track position.

If it is a defective track, the address of the alternate track is used in a seek command. (This is found in the ID field of the R0 record.) The operation is resumed after a search to the desired track.

The direct-access device error routine also handles both cylinder and head switching for sequential and split cylinder.

On a no-record-found error, no console message is issued if the arm has sought the correct track. Instead, a permanent error is flagged in the IOB.

When the overflow incomplete bit is set, a CCW for the remainder of the record must be constructed, because only a partial record has been either read or written. The command code for the CCW to be constructed is taken from sense byte 5. Since it is required to construct the command outside of the original CCW list and then TIC to the original channel program, the data chaining must not extend beyond this CCW. Data chaining that includes a TIC may result in an uncorrectable overrun condition.

Whenever an overflow incomplete, in conjunction with a file mask violation, end-of-cylinder, or track condition check is detected, the direct-access device error routine assumes that the continuation of the overflow record on the following track has the Record 1 in its ID field. This assumption is necessary because the error routine must orient the interrupted channel program to the first data record on the following track.

Sense Byte 0: The bits in the first sense byte have the meanings described below.

Bit 0: Command Reject

This bit indicates that the 2841 has received an invalid operation code. It is also turned on when a seek check, invalid sequence of commands, or file protect condition is detected.

Bit 1: Intervention Required

This bit indicates that the specified file is:

- Not physically attached to the system.
- Physically attached to the system but not available for use because the file motor is not on, a cover interlock is open, etc.

Bit 2: Bus-Out Parity

This bit indicates that a data parity error has been detected during the transfer of information from the channel to the 2841. This check is an odd redundancy check and occurs on both control and write operations. This check is done by the 2841. A parity error detected during a command transfer is a bus-out check and not a command reject.

The sequence is repeated ten times if the error continues to occur. On the eleventh occurrence of the error condition, a message is written to the operator and the control unit is considered inoperative.

Bit 3: Equipment Check

This bit indicates that an unusual condition has been detected in the control unit or device. A message is written to the operator.

The conditions that cause the bit to be set are:

1. Unsafe (Byte 2, Bit 0) - This bit is used to indicate that a file malfunction has been detected. Some of the malfunctions are:
 - More than one head has been selected.

- The file is trying to read and write at the same time.
- The write gate is off, and the write driver is on.
- The write gate is on, and the write driver is off.
- The erase driver is off, and the erase gate is on.
- The erase driver is on, and the erase gate is off.
- More than one file module has been selected.
- One of the DC file voltages has been lost (2311 only).

2. Serializer/Deserializer Check (Byte 2, Bit 2) - This bit indicates that a bit has been either lost or gained when the parallel channel data is converted to serial file data during a write operation.

3. Channel Tag Line Check (Byte 2, Bit 3) - This bit indicates that the 2841 has detected the presence of two concurrent tag lines, i.e., command out and service out.

4. 2841 Arithmetic Logic Unit (ALU) Check (Byte 2, Bit 4) - This bit indicates that more than one condition has been detected during a search operation, i.e., during a search equal operation both the equal and high latches are on.

5. Unselected Status (Byte 2, Bit 5) - Status line from files is on with no file selected.

A message is written to the operator.

Bit 4: Data Check

This bit indicates that a data error has been detected in the information received from the file. Error detection in the 2841 is performed by the division of the message polynomial by a generator polynomial, using Mod 2 arithmetic. In the case of the 2841, the polynomial is $X^{16} + 1$.

Bit 5: Overrun

This bit indicates that either a service-out signal was not received in the 2841 within the specified time allowed after service-in or that a chained CCW was issued but that it was received too late to be properly executed. Detection of an overrun during reading or writing causes an immediate stop of data transmission. When writing, the remaining portion of the record area is padded out with valid zeros. On reading, the remaining portion of the record continues to be

read into the 2841 and the check is generated.

The original sequence of operations is repeated ten times if the error continues to occur. On the eleventh occurrence of the error, a message is written and the task is abnormally terminated.

Bit 6: Track Condition Check

This bit does not apply to the 2301. It is set for other direct-access devices whenever the following conditions are detected.

- Defective track - (track condition bit B6=1). A track condition check is generated whenever a search, read or write command is attempted on a flagged defective track. Exception: home address and descriptor record (R0) commands. A seek to the alternate track is made. (Address of alternate track in ID field of R0 count area.)
- Alternate track - (track condition bit B7=1). A track condition check is generated when command-chaining and multiple track mode signals indicate that operations are to continue on the next high ordered track. The track condition check indication inhibits the incremental head switching. A seek to the defective track plus one is made. (Address of defective track in ID field of R0 count area.)

Bit 7: Seek Check

This bit indicates that the file has been unable to successfully complete a seek due to:

- Failure of a home address compare after automatic head switching on a multi-track operation.
- Failure of hardware which results in the access mechanism going to either the inner or outer crash stop.
- An invalid address (2301 only): the address in a seek command is more than 199.

Sense Byte 1: The bits of the second sense byte have the following meanings.

Bit 0: Data Check in Count Field

This bit indicates that a data error has been detected in a count field read from the file.

Bit 1: Track Overrun

This bit indicates that writing has not been completed by the time index

point is detected. This type of error is created during a write-R0 or write-count, key, or data operation.

Bit 2: End-of-Cylinder

This bit indicates that the CCW command chain has not been completed, but that end-of-cylinder has been detected.

Bit 3: Invalid Sequence

This bit indicates that an attempt has been made to execute an invalid sequence of CCWs. Invalid sequences are normally related to write operations. It will also occur if two set file mask CCWs are attempted to be performed in the same chain of CCWs. Command reject (Byte 0, Bit 0) is also turned on when an invalid sequence is encountered.

Valid sequences are defined in the individual command description. A summary of valid sequences follows:

1. Set File Mask
Write HA
2. Set File Mask
Write HA
Write R0
Write CKD
3. Search HA
TIC
Write R0
Write CKD
4. Search ID EQ (R0)
TIC
Write CKD
Write CKD
Write Special CKD
5. Search ID EQ (Previous ID)
TIC
Write CKD
6. Search Key EQ (Previous Key)
TIC
Write CKD (Rn)
Search ID (Rn)
TIC
Read KD (Rn)
Write CKD (Rn+1)
7. Search ID EQ
TIC
Write KD or Write Data
8. Search Key EQ
TIC
Write Data

Bit 4: No Record Found

This bit indicates that two index points were detected while executing a chain of CCWs, a read home address, or a read R0, with no intervening read or write operation on the data field of any record. (This could be an expected condition on search command chains.)

Bit 5: File Protected

This bit indicates that a seek, write CCW, or M/T read or search command was issued that violates the file mask. The write file mask, if violated, also causes the command reject bit to be set.

Bit 6: Service Overrun (2301) or Missing Address Markers (2841)

The service overrun bit is turned on when a service out signal is not received in the 2301/2820 within the specified time allowed after the service in signal. The overrun bit (bit 5 of sense byte 0) is also set.

The incident of a missing address marker is detected during the execution of any command or chain of commands which operate successive count fields on a track.

The detection is accomplished by identifying that two successive records on a track have equal bit conditions in bit 0 of the flag bytes. In normal conditions, bit 0 of the flag byte is always a zero for all even numbered records and is always a one for all odd numbered records.

Upon detection of a missing address marker, bit 6 of sense byte one (missing address marker) is turned on for all commands or chained commands except search ID CCWs. The search ID CCW may be used to pass over the missing address marker so that the remaining data on the track can be retrieved.

In addition to the flag byte check described above, a missing address marker will be detected if two index points are passed with no intervening address marker recorded on the track.

Bit 7: Overflow Incomplete

This bit is used with the optional record overflow feature. It is set as follows:

- Overflow to a bad track - overflow incomplete and track condition check are set.
- Overflow from an alternate track - overflow incomplete and track condition check are set.
- Overflow to a file protected boundary - overflow incomplete and file protected are set.
- Overflow to wrong track-head number compares unequal - overflow incomplete and seek check are set.

2250 Error Routine

The bits in the first sense byte have the meanings described below:

Bit 0: Command Reject

The setting of this bit indicates that there was an invalid modifier bit in a command, or an invalid command sequence. This is considered a permanent error.

Bit 1: Should Not Occur

If any of the "Should Not Occur" bits are set, a catastrophic error is assumed. The Log Out bit in the IOB is set, and the permanent error procedure is performed.

Bit 2: Bus-Out Check

This bit is turned on when a parity error is detected on a command or data byte during bus-out. The operation is retried once, and on the second occurrence of the error condition a message is written to the operator.

Bit 3: Should Not Occur

Bit 4: Data Check

The setting of this bit indicates that a buffer parity error was detected either during a read operation or during image regeneration. The operation is retried once, and on the second occurrence of the error condition a message is written to the operator.

Bit 5: Should Not Occur

Bit 6: Buffer Running

This bit is set when orders are being executed in the buffer.

Bit 7: Should Not Occur

2260 Error Routine

The bits in the first sense byte (byte 0) have the meanings described below:

Bit 0: Command Reject

The setting of this bit indicates that there was an invalid modifier bit in a command, or an invalid command sequence. This is considered a permanent error.

Bit 1: Intervention Required

The Intervention Required bit is set if a Write 1053 Printer command is encountered, and the 1053 Printer is not ready.

Bit 2: Bus-Out Check

This bit is turned on when a parity error is detected on a command or data byte during bus-out. There are three cases:

1. Bus-Out Check Initial Selection, command or data byte parity for 2260 or 1053.
2. Bus-Out Check Data Transmission, data byte parity for 2260 following Write Line Address or Erase command.
3. Bus-Out Check Data Transmission, data byte parity for 2260 following Write Buffer Storage command.

For 1 and 2, the operation is retried once, and on the second occurrence of the error condition a message is written to the operator.

For 3, an Erase Buffer Storage command is issued, and the Write command is reissued; on the second occurrence of the error condition a message is written to the operator.

Bit 3: Equipment Check

The setting of this bit indicates that a parity error was detected during a read operation.

When this condition occurs, the 2260 error routine transfers control to its second load (load module IGE0110B), which erases the message in error. The operator is then requested to re-enter the message; if the error is encountered a second time, it is considered permanent.

Bits 4-7: Should Not Occur

If any of the "Should Not Occur" bits are set, a catastrophic error condition is assumed. The Log Out bit in the IOB is set, and the permanent error procedure is performed.

COMMON ROUTINES

Common routines are used by the device-dependent error routines to analyze the type of error, to write console messages, and to update the statistics tables. These routines are:

- Error interpreter, which tests the sense bits and the CSW status bits to determine the type of error. This routine is resident; its load module name is IECINTRP.
- Write-to-operator routine, which constructs an "error" or "intervention required" message and writes that message to the operator. This routine is transient; its load module name is IGE0025C.
- Statistics update routine, which keeps the counters in the statistics tables up-to-date with the temporary and permanent errors determined by the device-dependent error routines. This routine is transient; its load module name is IGE0025D.

ERROR INTERPRETER ROUTINE

The error interpreter routine determines the types of errors for which error correction is required. The address of the routine is contained in the communication vector table (CVT).

The error interpreter "ANDs" the codes in the code and address constant list of the device-dependent routine to the corresponding sense or status bit. (See Part III: General Operating Procedure.) If the result is not zero, the bit is on and correction procedures for the specified sense or status condition must be taken. The address constant is added to the linkage register and an address within the device-dependent routine is obtained. Control returns to the device-dependent routine at that address.

WRITE-TO-OPERATOR ROUTINE

The write-to-operator routine is used by the device-dependent routines to format an error message and to write that message to the operator by use of the WTO macro-instruction.

The write-to-operator routine constructs two message types: an intervention required message and an error message.

The standard message code, message text, channel and unit address, command code, CSW status bits, and the first two sense bytes are written into the message formats. For direct-access devices the cylinder and track addresses are also written.

The standard message code consists of an internal component name, serial number, and action code.

The internal component name identifies the system component of Operating System/360 responsible for the format and content of the message. The operating system component for the I/O supervisor is IEA. The three characters are alphabetic and are identical to the internal code assigned to the system component.

The serial number is assigned to each message within a program.

The action code is a single character used to indicate the nature of the operator action required.

Intervention Required Message: This message is used when operator intervention is required. Its message format is:

IEA000A INT REQ aaa bb cccc dddd

where:

IEA000A is the standard message code for the machine operator. The internal component name is IEA. The serial number is 000, and the action code is A, meaning attention. Operator action is required before the device can continue its operation.

INT REQ is the message text, indicating intervention required.

The following a-b-c-d information is typed in hexadecimal:

aaa is the unit address.

bb is the command code as specified in the failing CCW in the channel program.

cccc is the status bytes of the CSW specified in the IOB.

dddd is the first two sense bytes specified in the IOB.

It is the operator's responsibility to take the correct action, such as: make the unit ready; feed more cards to the reader or punch; clear card jam; clear full stacker; clear full chip box; insert paper to the printer or console, etc.

Error Message: This message is used when a permanent error has occurred. Its message format is:

IEA000I I/O ERR aaa bb cccc dddd eeee ffff

where:

IEA000I is the standard message code for the operator. The internal component name is IEA, the serial number is 000, and the action code is I, meaning information, immediate operator action is not required.

I/O ERR is the message text, indicating the occurrence of an I/O error.

The following a-b-c-d-e-f information is typed in hexadecimal.

aaa is the unit address.

bb is the command code as specified in the failing CCW in the channel program.

cccc is the status bytes of the CSW as specified in the IOB.

dddd is the first two sense bytes as specified in the IOB.

The following e and f fields apply only to direct-access devices:

eeee is the cylinder address.

ffff is the track address.

STATISTICS UPDATE ROUTINE

The statistics update routine updates the 4-bit counters contained in the statistics table for the device-dependent routines. (See Introduction: Statistics Table.)

The work area of the statistics table entry for a particular device indicates the

counters that must be updated. (See Part V: Statistics Table.) The first byte of the work area corresponds to the bits in sense byte 0. The second work area byte corresponds to sense byte 1, or, if sense byte 2 is significant for the device type, to bits from both sense bytes 1 and 2. The order of correspondence is device-dependent and indicated by the counts maintained in the statistics table entries shown in Part V.

The first (low-order) bit of the work area corresponds to the first 4-bit counter in a statistics table entry. This correspondence continues through the counters and the last bit of the work area.

When determining the counters to be updated, the statistics update routine inspects the work area starting at the last work area bit (high-order). The table entry counters are thus updated from the right side of the table to the left side starting from the bottom. However, if a data check is indicated, the read and write counters are updated first.

If an overflow occurs in the counters, the statistical data recorder is entered for external error recording. (See following section.)

I/O OUTBOARD RECORDING ROUTINES

The I/O outboard recording routines, the outboard recorder (OBR) and the statistical data recorder (SDR), are a portion of the system environment recording (SER) routines. They are provided as an extension of the IBM-supplied error routines. The OBR and the SDR use the I/O supervisor transient area.

The OBR and the SDR collect environmental and statistical data regarding errors that occur in I/O devices and control units. These errors are termed I/O outboard errors. The OBR and the SDR then format the data and enter it as an external error record on the system residence device.

THE OUTBOARD RECORDER

The OBR is called by an IBM-supplied error routine whenever an I/O device has a permanent error. The OBR constructs a record entry that contains a description of the device environment at the time of failure and writes it on the system residence device.

OBR Interface: An error routine places into location 280 (decimal) the address of the request element associated with the error; from it, the OBR can obtain the address of the UCB, which indicates the device in error, and of the IOB. The address of the statistics table entry for the device is placed in location 284 (decimal), and the last four digits of the module name for the OBR are placed in a register. Control is routed to the task supervisor. The supervisor uses the digits in the register to complete the name of the OBR, which is then loaded from the systems residence device into the I/O supervisor transient area and given control.

The I/O supervisor transient area cannot be used by any other error routines until the OBR has completed its processing; this action guarantees that no other requests are made to the OBR such that the data at location 280 and 284 is overlaid. Upon completion of processing, the OBR routine calls in the SDR routine.

THE STATISTICAL DATA RECORDER

The SDR is called by an IBM-supplied error routine whenever any one of the 4-bit error statistics counters within the statistics table contains a count of 15. The SDR adds that 4-bit counter plus the partial counts within the other counters to an expanded statistics counter on the system residence device. Thus, a summary list of the various device failures are available for inspection.

SDR Interface: An error routine places the address of the statistics table entry at location 284 and the address of the request element in error at location 280. (These locations are decimal.) The routine is called in the manner described above in "OBR Interface." The SDR resets the statistics table counters for future use, and return is made to the supervisor by use of SVC 3.

Whenever the error routine detects that a statistics counter has overflowed at the same time that a permanent error has occurred, it calls the OBR which performs the outboard recording. The OBR then calls the SDR, which performs the statistical update.

MODULE OPERATION

Six modules, all of which use the I/O supervisor transient area, make up the I/O outboard recording routines:

- The OBR consists of three modules. The load module names are IGE0025F, IGE0125F, and IEG0225F.
- The SDR consists of two modules. The load module names are IGE0525F, and IGE0625F.
- The error message module is used by both the OBR and the SDR. Its load module name is IGE0425F.

The three OBR modules and the two SDR modules are executed whenever a permanent error occurs; the two SDR modules are executed whenever a counter reaches the value of 15. If an error occurs during module execution, the error message module is called.

After an error routine determines that I/O outboard recording is required, it places the address of the request element in error and of the statistics table entry for the device into the machine log out area, loads the name of either the OBR or the SDR module for the transfer control (XCTL) routine of the contents supervisor, and then branches to the XCTL routine. (See the OBR and SDR interface descriptions.)

After loading, the first module of either the OBR or the SDR gains control from the supervisor routine FINCH, which masks all interruptions as disabled. The module that gains control is executed and then brings the next required module into the I/O supervisor transient area by use of the XCTL routine, and so forth, until all required modules are executed. The last module to be executed issues an SVC 3 (EXIT) and control is given back to the supervisor.

Depending upon the type of system residence device and the number of UCBs in the system, the SER routine requires a data set extent, on the system residence device, of at least two tracks. These tracks are formatted and initialized by the SYS1.LOGREC initialization program.¹

The extent is termed the SER output data set and contains in contiguous positions:

- Header record.
- Statistical data records.
- Record entries.

A detailed description of these records may be found in Appendix D.

¹For information concerning this program, refer to the IBM System/360 Operating System: Utilities, Program Logic Manual, Form Z28-6614.

All modules use the machine log out area for operations upon the SER output data set. The machine log out area (decimal locations 128 through 383 of main storage) is used because no other feasible storage locations are available. If a machine check or catastrophic channel error occurs, which causes hardware recording in the log out area to overlay any information put there by the I/O outboard recording routines, a higher level diagnostic SER routine (SER0 or SER1) gains control and the occurrent errors are handled from that level.

To read and write the header record, statistical data records, and record entries, the OBR and the SDR modules use the EXCP macro-instruction. The IOB is loaded with the DCB, ECB, and CCW list addresses. For error messages, the error message module uses the WTO macro-instruction. Before these macro-instructions are issued, however, the system is enabled.

The channel programs necessary for operations upon the SER output data set consist of the following CCWs:

- Read home address (required only by the SDR).
- Search on ID or key equal for the appropriate record.
- TIC to the first CCW until the record is found.
- Read or write the record.

After execution of the channel program and when the I/O request is complete, the modules inspect the ECB completion code, which provides the status of the execution as follows:

- Normal. The record has been read into the log out area or written onto the system residence device with no errors.
- Intercepted. The IOB for the request was intercepted for other error recovery procedures because of an error at a previous device end; the EXCP macro-instruction is re-issued.
- Fail. The request has failed because either the data set is not within the extent specified or a permanent I/O error has occurred; an error index value and the load name for the error message module is set and an EXIT macro-instruction is issued.

Note: An EXIT macro-instruction is also issued when the header safety byte is zero.

The function of each module follows; the flow among them is shown in Chart 09.

First OBR Module: The first OBR module, upon gaining control, reads the header record from the system residence device into the machine log out area and starts constructing a record entry.

The module obtains and formats the following fields within the machine log out area (build area):

- Record entry label.
- Record entry type.
- Device type.
- Date.
- Program ID.

The name of the second load module of OBR is loaded into the XCTL register and the XCTL macro-instruction is issued.

Second OBR Module: The second OBR module completes the construction of the record entry. It inserts the following fields:

- First CCW.
- Failing CCW.
- CSW.
- Sense information.
- Channel and device address.
- Last seek address.
- Volume label.
- Time.

The name of the third load module of the OBR is loaded into the XCTL register and the XCTL macro-instruction is issued.

Third OBR Module: The third OBR module writes the header record and the record entry onto the system residence device. If the records do not fit in the extent, the error message module is called. The name

of the first load module of the SDR or the name of the error message module is loaded into the XCTL register and the XCTL macro-instruction is issued.

First SDR Module: The first SDR module reads the statistical data record for the particular device from the system residence device and obtains the internal statistical counters which are to be added to the external counters in the statistical data record. The second SDR module is then called.

Second SDR Module: The second SDR module adds the appropriate statistical table entry counter (or counters) to the external statistical data record counters and then writes the record back onto the system residence device. If necessary, the error message module is called; otherwise, the machine log out area is cleared, and SVC 15 is issued so that the request element for the routines will be returned to the freelist. Finally, the RETURN macro-instruction is issued for a return to the task supervisor and then to the I/O supervisor.

Error Message Module: This module handles all errors occurring within the I/O out-board recording modules and writes a message to the operator regarding them. If the OBR routine was in control, the error message module calls in the first module of the SDR; if the SDR was in control, it returns to the task supervisor.

The following error messages may be written to the operator:

1. IFB001I I/O ERROR DURING SER
2. IFB003I SER RECORDING AREA FULL
3. IFB004I SER DISK AREA FORMAT ERROR

The SVC transient area routines are type 3 SVC routines that are called into the SVC transient area in response to system macro-instructions. The macro-instruction names, load module names, and SVC numbers are:

- PURGE: Load module IGC0001F (SVC 16)
- RESTORE: Load module IGC0001G (SVC 17)
- DEVTYPE: Load module IGC0002D (SVC 24)
- IOHALT: Load module IGC0003C (SVC 33)

SVC PURGE ROUTINE

The purge routine removes (purges) request elements that are associated with a task or a data set. Request elements may be purged from the logical channel and seek queues of the I/O supervisor, from the asynchronous exit queue of the task supervisor, or from the request blocks that are chained to the TCB. The request elements are purged according to the options specified in the purge input parameters. These options are:

- Purge specified DEB. Purge only the specified DEB or purge the complete DEB chain.
- Post Complete. Post the purged I/O requests complete or do not.
- Halt I/O. Issue a halt I/O instruction for active requests or allow them to quiesce.
- Purge related. Purge only related requests or purge all requests.
- Purge only requests associated with a specified TCB on the DEB. Indicated by a unique code in the parameter list.

After the request elements are purged, the SVC purge routine returns the elements to the freelist and chains the IOBs that are addressed by each of the purged request elements onto the chain field specified in the purge input parameters.

The queues are purged in the following order:

1. Asynchronous exit queue.
2. Request blocks for the given TCB.
3. Logical channel and seek queues of the I/O supervisor that are used for I/O requests associated with either the DEB or the TCB specified.
 - a. If the DEB is specified, the SVC purge routine examines its extents for the device (or devices) upon which the data set resides and purges the associated logical channel and seek queues.
 - b. If the TCB is specified, the SVC purge routine examines the request element table and removes each active request element associated with the TCB from its corresponding queue.

The SVC purge routine contains four subroutines that purge the queues according to the options specified. These subroutines are:

- Asynchronous exit queue (AEQ) purge subroutine. Purges the asynchronous exit queue by use of the check subroutine.
- Request block (RB) subroutine. Purges the RB list by use of the check subroutine.
- Check subroutine. Purges the AEQ and RB list according to the options specified by the PURGE macro-instruction and returns the purged request elements to the freelist.
- UCB purge subroutine. Purges the logical channel and seek queues of the request elements associated with either the DEB or TCB specified in the purge input parameters.

Purge Complete Subroutine

Whenever a PURGE macro-instruction is given without the halt I/O option specified, the SVC purge routine must allow the active I/O requests to quiesce.

The SVC purge routine places a count of the active I/O requests into the second four bits of the ECB for the purge request. These requests must finish before the ECB can be posted. The SVC purge routine sets the IOB purge flag for each active I/O request to indicate that it is awaiting the completion of that I/O request.

The purge complete subroutine is given control when the I/O interruption supervisor, at the completion of an I/O request, determines that the IOB purge flag is set. The purge complete subroutine decrements the count in the ECB for the purge request and when the count is not zero, returns control to the I/O interruption supervisor. When the count is zero, the subroutine posts the ECB for the purge request complete via the post routine interface.

RESTORE ROUTINE

The restore routine re-introduces purged requests to the system. Because the requests are exclusively for I/O operations, the EXCP macro-instruction is issued for each request element. The request elements are indicated by the IOBs in the chain created by the SVC purge routine.

The RESTORE macro-instruction requires that the IFLGS field in the DCB be initialized in the same way as for the EXCP macro-instruction.

DEVTYPE ROUTINE

When SVC 24 is executed, the DEVTYPE routine is loaded into the SVC transient area and receives control. The expansion of the DEVTYPE macro-instruction places parameters in general registers; the DEVTYPE routine uses the parameters to identify the device, and to place device information in an output area.

The DEVTYPE routine looks at the parameter that specifies the address of a field that contains a DD name. Using that DD name, the routine finds the corresponding TIOT entry, and uses it to locate the proper UCB. The DEVTYPE routine then moves the UCB device code field to the high order word of the output area specified by the area parameter.

If the DEVTAB parameter is specified in the DEVTYPE macro-instruction, and the device is a direct access device, the output area must be 5 words long. The DEVTYPE routine locates the device characteristics table via the communications vector table, moves the appropriate entry into the low order three words of the output area, and moves the blocksize field to the second word of the output area.

If the DEVTAB parameter is not specified, the output area must be 2 words long. The DEVTYPE routine determines the maximum block size from the device characteristics table (if the device is a direct access device) or from internal constants, and moves it to the second word of the output area.

When processing is complete, the DEVTYPE routine places a return code in general register 15. An error return code of 04 indicates one of the following conditions:

- No output area specified: the area parameter is missing from the DEVTYPE macro-instruction.
- DD name not found: there is no TIOT entry that corresponds to the DD name supplied.
- Invalid UCB unit type field: the UCB unit type field (byte 4 of the device code field) does not specify a direct access, tape, or unit record device.

If the request is completed satisfactorily, the DEVTYPE routine places a return code of 00 in general register 15, and exits via SVC 3.

IOHALT ROUTINE

When the IOHALT routine (SVC 33) receives control, it is given the address of the UCB associated with the device to be stopped. It checks that address for validity, then inspects the UCB device code field to make sure that the device is not a direct access device.

If the device is a byte mode device, and there is a burst mode device running on the same channel, the routine enters a wait loop until the burst mode device stops.

The HALT I/O instruction is then executed, and the condition codes and CSW status bits are inspected to determine the success of the operation.

The appropriate condition code is placed in general register 15. If the operation was successful, the IOHALT routine places a post code (X'48') in the ECB code field of the IOB and issues SVC 3 (EXIT).

The I/O supervisor uses control blocks and tables to communicate with itself, with the rest of the control program, with processing programs, and with I/O devices.

This section of the publication describes in detail the characteristics of the control blocks and of the tables used by the I/O supervisor.

Figures 8 through 21 show control block and table formats. Where pertinent, a discussion of the fields follows the figures.

For convenience, the following control blocks and tables are presented in alphabetical order:

- Attention Table.
- Channel Search Table.
- Data Control Block.
- Data Extent Block.
- Device Table.
- Event Control Block.
- Input/Output Block.
- Logical Channel Word Table.
- Request Element Table.
- Statistics Table.
- UCB Lookup Table.
- Unit Control Block.

ATTENTION TABLE

The attention table is used by the I/O supervisor to obtain the addresses of the attention routines required to service the I/O devices attached to the system.

The attention table has the following characteristics:

- Creation. The table is created at system generation time.
- Storage Area. The table resides, as a permanent part of the resident supervisor, in protected resident storage (when protection is available).
- Size. The table contains one 4-byte entry per attention routine, up to a maximum of 64 entries.
- Means of Access. The ATNTAB byte, supplied by the user in the UCB, is added to the starting address of the attention table to obtain the proper attention routine entry in the table for the device.
- Format. The format of an attention table entry is shown in Figure 8.

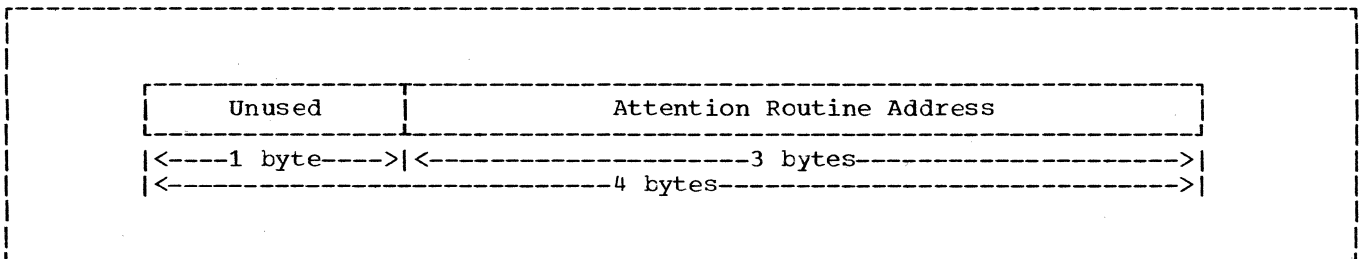


Figure 8. Attention Table Entry Format

CHANNEL TABLE

The I/O supervisor uses the channel table to find the proper channel search module and to re-enable for additional I/O interruptions.

The channel table has the following characteristics:

- Creation. The channel table is created at system generation time.
- Storage Area. The table resides, as a permanent part of the resident supervisor, in protected resident storage (when protection is available).
- Size. The table contains one 4-byte entry for each physical channel attached to the system.
- Means of Access. The address of a particular physical channel is multiplied by 4 and added to the starting address of the channel table to obtain the address of the entry for that physical channel.
- Format. The format of the channel table is shown in Figure 9.

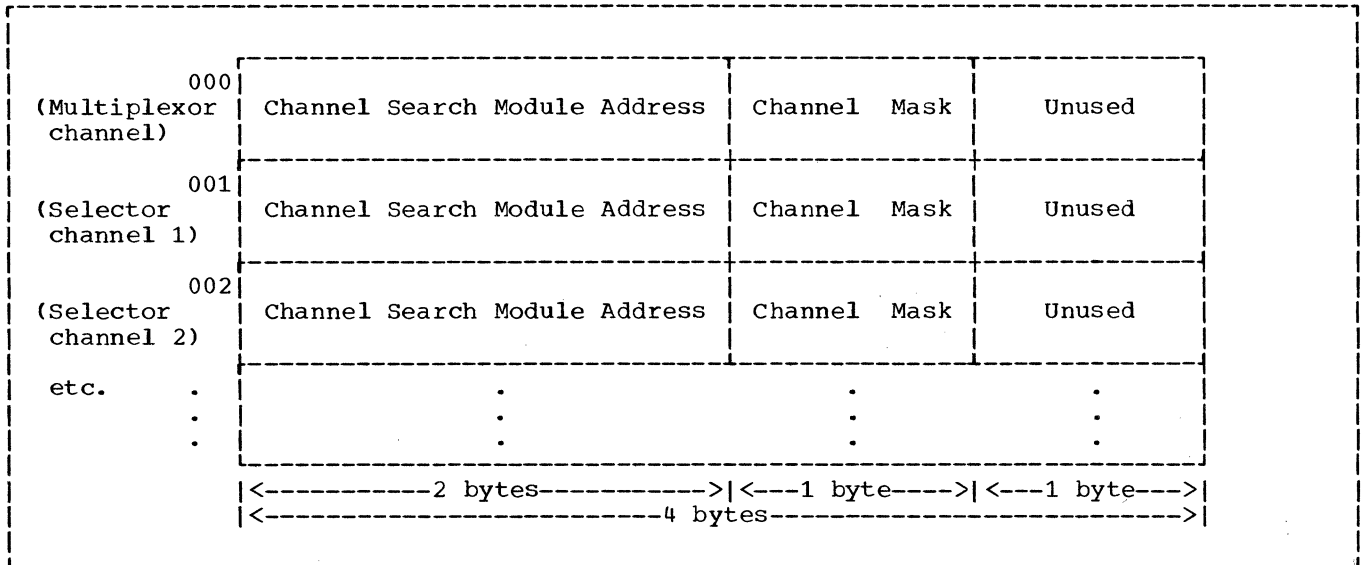


Figure 9. Channel Table Format

Channel Search Module Address (2 bytes)
 This 2-byte field contains the address of the channel-dependent channel search module associated with the physical channel. A channel search module exists for each physical channel and it determines the next request

to be started on that channel when it becomes available.

Channel Mask (1 byte)
 This byte contains the channel re-enable mask.

DATA CONTROL BLOCK

A data control block (DCB) is used primarily by the open and close routines of data management, but the I/O supervisor is concerned with several of its fields.

The characteristics of a DCB are as follows:

- Quantity. There is one DCB for each data set.
- Creation. Space for the DCBs is reserved at problem program assembly time. The DCB is completed at open time.
- Storage Area. All DCBs reside in unprotected problem program storage.
- Size. Refer to the Introduction to Control Program Logic, Program Logic Manual for the size and description of the entire DCB.
- Format. The DCB fields used by the I/O supervisor are shown in Figure 10.

FIELD	CONTENT	SIZE	SECTION
IFLGS	I/O Supervisor error flags	1 byte	Foundation Block
DEBAD	DEB address	3 bytes	
BLKCT	Tape block count	4 bytes	Tape

Figure 10. Data Control Block Fields Used by the Input/Output Supervisor

IFLGS (1 byte)

These are flag bits used by the I/O supervisor in communicating error conditions and in determining corrective procedures. They decode as follows:

Bits 0 and 1: DCB Exception

These two bits are used during error recovery procedures when accessing sequential files where positioning is important. The flags indicate the error status for the pertinent data set, and must always be present. They have the following value:

- 00 Normal conditions are present.
- 01 Error correction is in progress.
- 11 A permanent error condition exists.

Bits 2 and 3: Printer Overflow

These bits are set by error routines

when printer overflow occurs. They have the following value:

- 10 Channel 9 printer overflow.
- 01 Channel 12 printer overflow.

Bits 4 and 5: I/O Supervisor Error Key

This key is mandatory and indicates whether or not an IBM-supplied error routine should be entered. The key is checked by the error routine interface. The flags have the following value:

- 00 Always use IBM-supplied error routines.
- 11 Never use IBM-supplied error routines.

Bits 6 and 7: Reserved

DEBAD (3 bytes)

This DEB address field contains the actual address of the data set's associated data extent block. The DEB contains the location and boundaries of that data set. The DEB address must always be present in the DCB foundation block after open time.

BLKCT (4 bytes)

This device-dependent area is used by the I/O supervisor to accumulate the tape block count for a data set on tape that is either being read or written. The block count designates physical unit position, not relative data set position. At the normal completion (channel end/device end) of all I/O requests, the IOB block count increment amount is added to this DCB block count field to form the updated tape block count.

Notes: An incorrect block count field results when an error occurs in the middle of a command chained CCW list. In this case, it is the user's responsibility to adjust the DCB block count field. He must determine which CCW had the error and then decrement the block count so that it reflects the true count.

When an uncorrectable error occurs, the block count field is updated as usual.

The effect of a halt I/O request by the purge routine on the block count field is unpredictable, since the I/O interruption condition is determined by the disposition of the device and control unit, which are variable.

DATA EXTENT BLOCK

A data extent block (DEB) describes the location and boundaries (extent) of a data set and points to the pertinent UCB(s) for the device(s) upon which the data set resides. The characteristics of a DEB are as follows:

- Quantity. There is one DEB for each DCB.
- Creation. The DEBs are created by the open routine.
- Storage Area. All DEBs reside in the dynamic area.
- Size. Refer to the Introduction to Control Program Logic, Program Logic Manual for the size and description of the entire DEB.
- Format. The DEB format is shown in Figure 11.

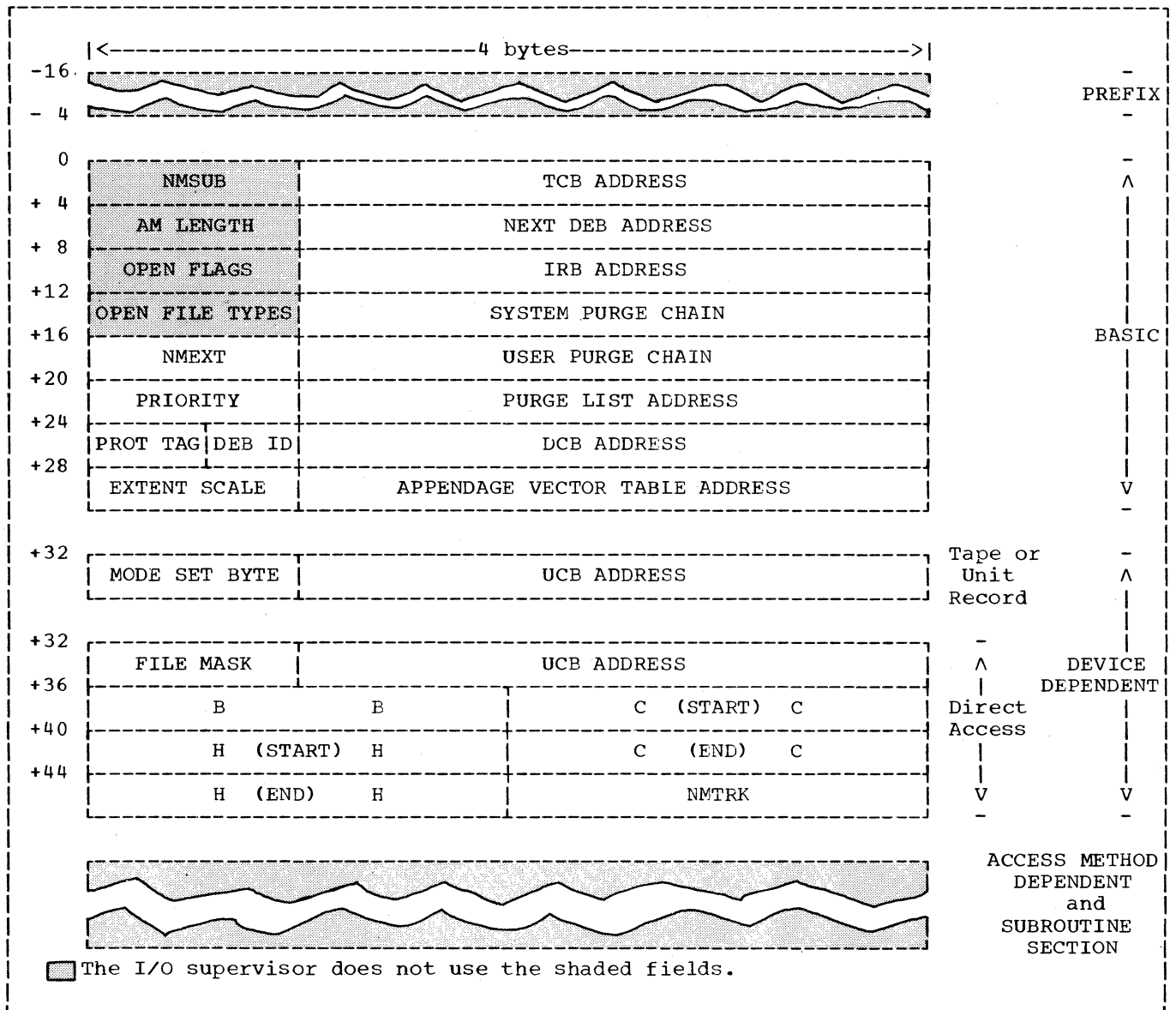


Figure 11. Data Extent Block Format

TCB ADDRESS (3 bytes)
This field contains the address of the TCB associated with this DEB.

NEXT DEB ADDRESS (3 bytes)
This field contains the address of the next DEB in a chain of DEBs running under the same task.

IRB ADDRESS (3 bytes)
The interruption request block (IRB) address for system error exits is contained here.

SYSTEM PURGE CHAIN (3 bytes)
This field contains the address of the first IOB in the chain of IOBs for I/O requests purged by the I/O purge routine.

NUMBER OF EXTENT ENTRIES (1 BYTE)
This byte contains the number of extent entries specified in this DEB.

USER PURGE CHAIN (3 bytes)
This field contains the address of the first IOB in the chain of IOBs for I/O requests purged by the purge routine for SVC 16.

PRIORITY (1 byte)
This field contains a binary number, assigned by the open routine, which indicates the dispatching priority of the task associated with an I/O request.

PURGE LIST ADDRESS (3 bytes)
This field contains the address of a parameter list for an SVC purge request. The I/O supervisor uses this field when purging to locate the appropriate purge ECB to post complete.

PROTECTION TAG (4 bits)
These bits contain the protection tag for the associated task.

DEB ID (4 bits)
These four bits identify this block as a DEB by means of a hexadecimal "F".

DCB ADDRESS (3 bytes)
The address of the DCB corresponding to this DEB is contained here.

EXTENT SCALE (1 byte)
This byte contains a shift factor used to determine the size of the extent entries in the DEB. This field contains 4 for a direct-access device and 2 for a non-direct-access device.

APPENDAGE VECTOR TABLE ADDRESS (3 bytes)
This field contains the address of the appendage vector table that is con-

structed at open. The appendage table consists of address constants (adcons) as follows:

0	DC A(EOE)	End of Extent
4	DC A(SIO)	Start I/O
8	DC A(PCI)	PCI
12	DC A(CE)	Channel End
16	DC A(XCE)	Abnormal End
<-4 bytes->		

The I/O supervisor uses the respective adcon as a branch address to the five types of appendages. If an appendage is not used, the respective adcon points to a branch back to the I/O supervisor "normal" processing return.

When no appendages are required, this table contains five adcon's pointing back to the "normal" processing return.

FILE MASK or MODE SET (1 byte)
This field contains either the file mask for a direct-access device or a mode set byte for tape.

UCB ADDRESS (3 bytes)
This field contains the address of a UCB associated with this DEB.

BB (2 bytes)
This field contains the bin number of a direct-access volume (data cell drive).

CC START (2 bytes)
This field contains the cylinder address for the start of an extent on a direct-access volume.

HH START (2 bytes)
This field contains the read/write head number for the start of an extent on a direct-access volume.

CC END (2 bytes)
This field contains the cylinder address for the end of an extent on a direct-access volume.

HH END (2 bytes)
This field contains the read/write head number for the end of an extent on a direct-access volume.

NMTR (2 bytes)
This field contains the number of tracks allocated to a given extent on a direct-access volume.

DEVICE TABLE

The device table is used by the I/O supervisor to locate the device-dependent start I/O, enqueue, or trapcode module that pertains to the device in use.

The following characteristics are pertinent:

- Creation. The device table is created at system generation time.
- Storage Area. The device table resides, as a permanent part of the resident supervisor, in protected resident storage (when protection is available).
- Size. The device table may have from one to a maximum of 42 entries, each entry being 6 bytes in length. There is always one entry per device type per enqueueing option specified for the device types at system generation time. For example, direct-access devices using priority enqueueing would have one entry, direct-access devices using FIFO enqueueing would have one entry, etc.
- Means of Access. The DEVTAB byte in the unit control block (UCB), when added to the starting address of the device table, gives the address of the proper entry in the table for the device.
- Format. The format of a device table entry is shown in Figure 12. The possible entries in the device table and their order, are:

```

Unit Record
  Priority
  FIFO
Tape
  Priority
  FIFO
Direct-access
  Ordered
  Priority
  FIFO
Telecommunications
  Priority
  FIFO
  
```

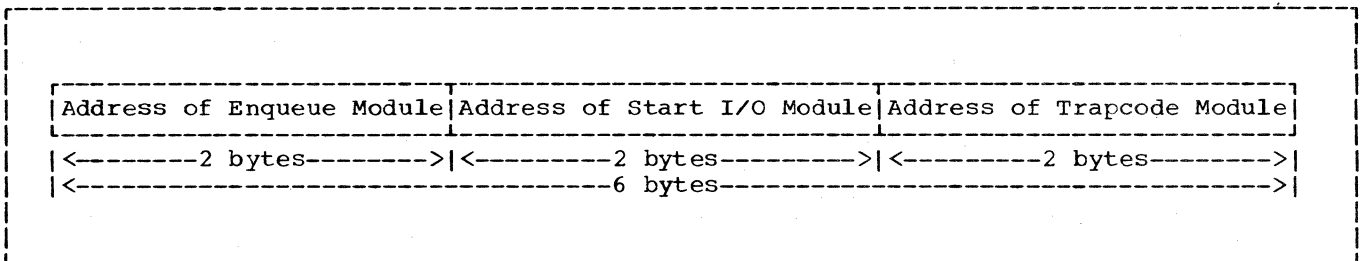


Figure 12. Device Table Entry Format

EVENT CONTROL BLOCK

An event control block (ECB) is used to post the completion of an I/O request. It indicates whether or not the request has completed successfully.

The following characteristics are pertinent:

- Quantity. There is one ECB for each I/O request.
- Creation. The ECBs are created by the user of the I/O supervisor.
- Storage Areas. All ECBs reside in unprotected problem program storage.
- Size. An ECB is 4 bytes in length.
- Format. The ECB format, after it is posted complete, is shown in Figure 13.

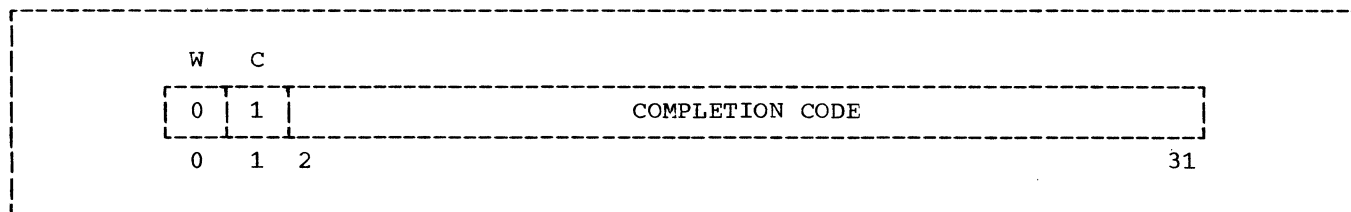


Figure 13. Event Control Block Format After Posting

<p>Bit 0: W This is the wait bit and is set to 0.</p>	<p>000100 IOB intercept code. Whenever an error occurs after a channel end interruption for a device, the I/O request for that device has already been posted complete and the request element returned to the freelist. To handle the error, the I/O supervisor sets the UCB intercept flag to indicate that the next I/O request for that device must be intercepted. When intercepted, the IOB for the new I/O request and the CSW and sense data for the error are passed to the error recovery procedures for the device. If a permanent error exists, the ECB for the intercepted IOB is posted complete with the IOB intercept code.</p>
<p>Bit 1: C This is the complete bit and is always set to 1 when the I/O request has completed, regardless of the success of the event.</p>	
<p>Bits 2-31: Completion Code (30 bits) This code, along with the complete bit, is the ECB code as contained in the IOB. The completion code indicates to the user in what manner the I/O request has completed. When the first bit of the completion code is off, the request was not successfully completed. The next five bits indicate the cause and the remaining 24 bits are 0. There are six different completion codes; the six significant bits for these codes are:</p>	
<p>111111 Normal completion (no errors).</p>	<p>001000 Not started or purged. This code signifies either that the I/O request has not been started or that it has been purged.</p>
<p>000001 I/O permanent error code.</p>	
<p>000010 Extent permanent error code. This code indicates that the seek address specified in the IOB is out of the extent specified in the DEB.</p>	<p>001111 Error could not be retried. This code signifies that the home address and/or R0 could not be read during error recovery procedures.</p>

INPUT/OUTPUT BLOCK

An input/output block (IOB) contains information that pertains to a requested I/O operation and is used by the I/O supervisor. The address of an IOB is passed as the only parameter of the EXCP macro-instruction.

The characteristics of an IOB are as follows:

- Quantity. There must be one unique IOB for each I/O request.
- Creation. The IOBs are created by the user of the I/O supervisor.
- Storage Area. All IOBs reside in unprotected problem program storage.
- Size. An IOB is 32 or 40 bytes in length.
 1. The 32-byte field is always present for all devices.
 2. The 32-byte field plus 8 additional bytes are present for direct-access and telecommunications devices.
- Format. The IOB format is shown in Figure 14.

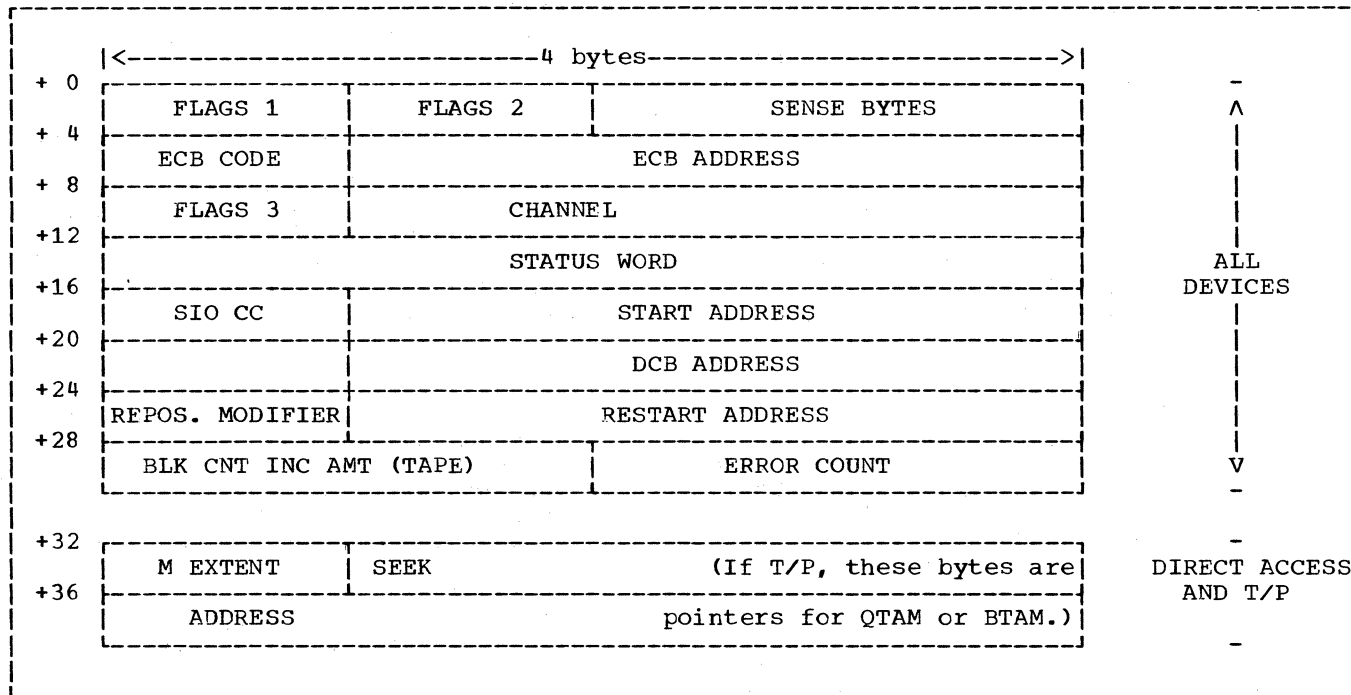


Figure 14. Input/Output Block Format

FLAGS 1 (1 byte)

Bits 0 - 1: Channel Program Description
 These two bits indicate the construction of the CCW list, and are set by the user of the I/O supervisor. They decode as follows:

- 00 No chaining
- 10 Data chaining only
- 01 Command chaining only
- 11 Data and command chaining (mixed)

Bit 2: IOB Error Correction Indicator

This bit is set by an IBM-supplied error routine to indicate that the I/O request associated with the IOB is under control of the error routine.

Bit 3: Modifier Flag 1

This bit is set by an error routine to indicate that the device in use must be repositioned by use of the command code specified by an IBM-supplied

error routine in the IOB reposition modifier byte. (See Part I: Tape Start I/O Module.)

Bit 4: Modifier Flag 2

This bit is set by an IBM-supplied error routine to indicate that a cyclic redundancy check (CRC) is required for the tape. (See Part I: Tape Start I/O Module.)

Bit 5: IOB Exception

This bit is set to indicate that an error occurred during the I/O request associated with this IOB. The bit is turned off when the error is determined to be corrected; if uncorrected, the exception bit is left on.

Bit 6: Unrelated

This bit is set by the user of the I/O supervisor and identifies all I/O requests (for use of a data set) that are not related to any other request. Requests that are related are executed in a first-in first-out (FIFO) arrangement and no related request is started until all prior related requests have completed. (Related is 0, unrelated is 1.)

Bit 7: Start/Restart

This bit is set by an IBM-supplied error routine to indicate that the restart address in the IOB is to be used for the I/O request; when 0, the start address is used.

FLAGS 2 (1 byte)

Bit 0: Halt I/O Issued Flag

This bit is set by the SVC purge routine to indicate that it has issued a halt input/output (HIO) instruction for the device in use for the I/O request that is associated with the IOB and is being purged.

Bit 1: Sense Flag

This bit indicates that a sense command should be issued by the sense subroutine because of an error at the previous channel end. At device end, the device is free and the sense command is issued.

Bit 2: Purge Flag

This bit is turned on by the SVC or I/O purge routine when awaiting completion of the device and allowing the I/O request to quiesce.

Bit 3: Read HA R0 Flag

This bit is turned on by the 2311 error routine when no seek is required.

Bit 4: No Test for Out-of-Extent Flag

This bit is turned on by the 2311 error routine when an alternate track is in use.

Bit 5: CCHH Update Flag

This bit is turned on by the direct-access device error routines when updating the seek address because of a cylinder end or file mask violation condition.

Bit 6: Reserved

Bit 7: IOB QSAM Flag

This bit is turned on by the Queued Sequential Access Method routines when error recovery (using three buffers) is to be provided for a 2540 card punch.

SENSE BYTES (2 bytes)

The sense subroutine loads these bytes with the first 2 bytes of sense information read into the UCB when a sense command is given.

ECB CODE (1 byte)

The I/O supervisor places the completion code for the I/O request associated with the IOB into this area. This code is also placed into the event control block (ECB) when the request is posted complete. (For additional information, see "Event Control Block".)

ECB ADDRESS (3 bytes)

These bytes are set by the user of the I/O supervisor to contain the address of the event control block (ECB) that corresponds to the IOB and is to be posted complete at the completion of the I/O request described in the IOB.

FLAGS 3 (1 byte)

This byte is used by IBM-supplied error routines and is defined in Part III: Device-Dependent Characteristics.

CHANNEL STATUS WORD (7 bytes)

These bytes contain the low-order seven bytes of the channel status word (CSW), as stored by the I/O supervisor at channel end.

SIO CC (1 byte)

The low-order two bits of this byte are set by the start I/O subroutine to contain the condition code resulting from issuance of the SIO instruction for the associated I/O request.

START ADDRESS (3 bytes)

These three bytes are supplied by the

user of the I/O supervisor and contain the address of the first CCW that is to be executed in the channel program.

DCB ADDRESS (3 bytes)

These bytes are supplied by the user of the I/O supervisor to contain the address of the data control block (DCB) that corresponds to the IOB.

REPOSITION MODIFIER (1 byte)

This byte is loaded with a positioning command code by IBM-supplied error routines that are used for device repositioning procedures when the IOB modifier flag 1 is on.

RESTART ADDRESS (3 bytes)

This address is used as the CCW address in the channel program for an error restart by IBM-supplied error routines during error correction procedures.

BLOCK COUNT INCREMENT AMOUNT (2 bytes)

This is the tape block count increment amount set by the user of the I/O supervisor and is used by the I/O supervisor at normal channel end (completion of the I/O request) to

update the tape block count (BLKCT) field in the DCB associated with the IOB. The field is signed and may contain an increment or decrement up to ± 32768 . When an uncorrectable error occurs, the I/O supervisor uses this field to update the DCB block count as usual.

ERROR COUNT (2 bytes)

These bytes contain flags and counts used by the IBM-supplied error routines and are explained in Part III: Device-Dependent Characteristics.

M EXTENT (1 byte)

This byte is set by the user of the I/O supervisor to indicate the number of the DEB extent that is to be used for the request. (The first extent is indicated by the character 0, the second by 1, the third by 2, etc.)

SEEK ADDRESS (7 bytes)

The user supplies the initial seek address BBCCHHR required for the I/O request for the direct-access device. For teleprocessing devices, this field contains QTAM and BTAM pointers.

LOGICAL CHANNEL WORD TABLE

The logical channel word table consists of the logical channel words that control the logical channel queues. It is used by the I/O supervisor and the I/O purge and SVC purge routines.

The logical channel word table has the following characteristics:

- Creation. The table is created at system generation time.
- Storage Area. It resides, as a permanent part of the resident supervisor, in protected resident storage (when protection is available).
- Size. The table contains one 8-byte logical channel word per logical channel queue.
- Means of Access. The LCHTAB byte in the UCB, when multiplied by 8 and added to the starting address of the logical channel word table, gives the proper logical channel word for the device and its associated logical channel queue.
- Format. The format of a logical channel word is shown in Figure 15.

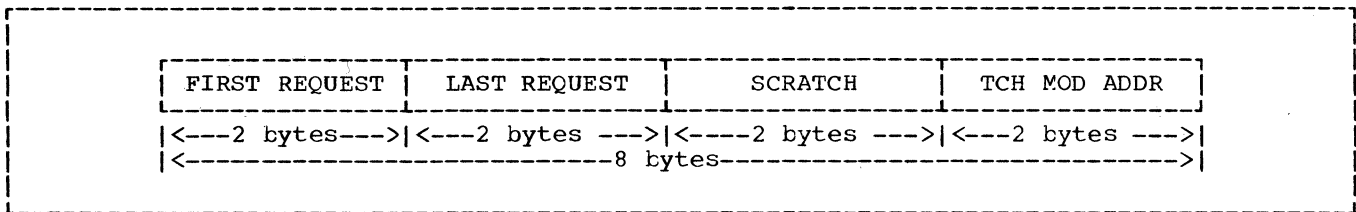


Figure 15. Logical Channel Word Format

FIRST REQUEST (2 bytes)

These two bytes contain either an address or an index value to the first request element in the logical channel queue.

LAST REQUEST (2 bytes)

These two bytes contain either an address or an index value to the last request element in the logical channel queue.

Notes:

1. When a logical channel queue is void, the "first request" field contains a dummy link address of hexadecimal FFFF and the "last request" field contains the address of that logical channel word.

2. When there is only one request element in the queue, both "first request" and "last request" contain the address of that element.

SCRATCH (2 bytes)

This field is used as a temporary storage area for an address or index value. It is used when more than one logical channel queue for a physical channel is searched in order to find the highest priority I/O request with which to restart the channel.

TCH MOD ADDR (2 bytes)

This field addresses the device-dependent test channel module.

REQUEST ELEMENT TABLE

The elements in the request element table are used by the I/O supervisor to represent active or queued I/O requests. The unused elements in the table are available for incoming I/O request representation.

The request element table has the following characteristics:

- Creation. The table is created at system generation time.
- Storage Area. It resides, as a permanent part of the resident supervisor, in protected resident storage (when protection is available).
- Size. The total number of request elements in the table is defined at system generation time. The request element table for a system in which option 4 is excluded contains a 12-byte request element for the maximum number of I/O requests expected at any one time; and for a system in which option 4 is included, a 16-byte request element.
- Means of Access. The active request elements are addressed by the LAST REQUEST field in the associated UCB. The available request elements are contained in the freelist, which is addressed by the freelist pointer in the CVT. The queued request elements are within the particular logical channel queue or seek queue referenced by the logical channel word and seek queue control word, respectively.
- Format. The I/O supervisor is concerned with all information in a request element. The format of a 12-byte and 16-byte request element is shown in Figure 16.

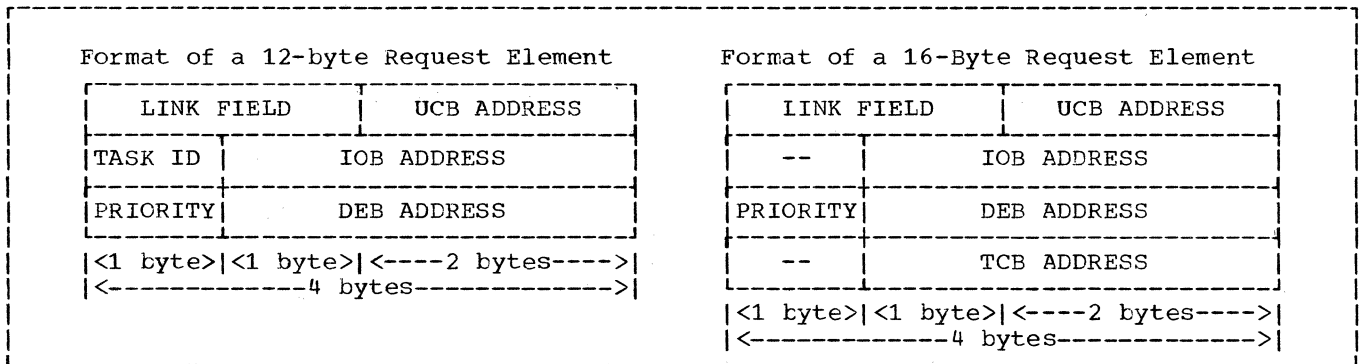


Figure 16. Request Element Formats

LINK FIELD (2 bytes)

This is a 2-byte link field used to link the request elements that are members of a particular queue or belong to the freelist.

UCB ADDRESS (2 bytes)

This field addresses the UCB associated with the queued I/O request.

TASK ID (1 byte)

This byte contains the task control block identification of the task that originally initiated the I/O request.

IOB ADDRESS (3 bytes)

This field contains the address of the IOB associated with the I/O request.

PRIORITY (1 byte)

This byte contains the priority of the I/O request represented by this request element. The priority is assigned at open time according to the priority of the associated task. If priority enqueueing is not specified, this byte is not significant.

DEB ADDRESS (3 bytes)

This field contains the address of the DEB associated with the data set for this I/O request.

TCB Address (3 bytes)

This field contains the address of the task control block for the task that initiated the I/O request.

STATISTICS TABLE

The statistics table contains the statistical data required to maintain statistical data records. It is used by IBM-supplied error routines and the statistical data recorder (SDR) of SER.

The statistics table has the following characteristics:

- Creation. The statistics table is created at system generation time.
- Storage Area. The table resides, as a permanent part of the resident supervisor, in protected resident storage (when protection is available).
- Size. The statistics table contains from 0 to 128 ten-byte entries; there is one entry for each device attached to the system.
- Means of Access. The STATAB byte in the UCB contains a factor, which when multiplied by ten and added to the starting address of the statistics table, provides the address of the proper entry for a device.
- Format. The format of each entry in the table varies with the type of device. The first 8 bytes of an entry contain statistical data; the last 2 bytes are a work area used by error routines. The device type formats are shown in Figure 17; note that the work area is not shown.

	Unit Record Equipment		2400 Tape Series	
1052	Temporary	Temporary	Temporary	Temporary
2150	Read Failures	Write Failures	Read Failures	Write Failures
1015				
1285	Intervention	Bus Out	Intervention	Bus Out
1402	Required	Check	Required	Check
1442				
1404	Equipment		Equipment	Overrun
2201	Check	Overrun	Check	
1403				
1443			Word	Data
2501	Device-Dependent	Device-Dependent	Count 0	Converter Check
2520				
2671			R/W Vertical	Longitudinal
2701			Redundancy Check	Redundancy Check
2702				
7770			Skew	Cyclic
7772				Redundancy Check
2250				
2260			Skew Reg VRC	Noise
1053				
	<-----1 byte----->		<-----1 byte----->	

● Figure 17. Statistics Table Entry Formats

(Continued)

2841 Control Unit		2820 Control Unit	
Temporary Read Failures	Temporary Write Failures	Temporary Read Failures	Temporary Write Failures
Intervention Required	Bus Out Check	Intervention Required	Bus Out Check
Equipment Check	Overrun	Equipment Check	
Track Condition	Seek Check	Track Condition Check	
Unsafe			Track Overrun
Serializer/Deserializer	Control Unit Tag Line		
Arithmetic Logical Unit		No Record Found	
Missing Address Marker			
<-----1 byte----->		<-----1 byte----->	

Figure 17. Statistics Table Entry Formats (Continued)

UCB LOOKUP TABLE

The UCB lookup table is used by the I/O interruption supervisor to obtain the address of the UCB associated with an I/O interruption.

The UCB lookup table has the following characteristics:

- Creation. The table is created at system generation time.
- Storage Area. The table resides, as a permanent part of the resident supervisor, in protected resident storage (when protection is available).
- Size. The size of the table is dependent upon the number and the unit addresses of I/O devices, control units, and physical channels attached to the system.
- Means of access. The table values are used in the algorithm routine described in Part II: UCB Lookup Routine. The table is addressed by the CVT.
- Format. The UCB lookup table is segmented as shown in Figure 18.

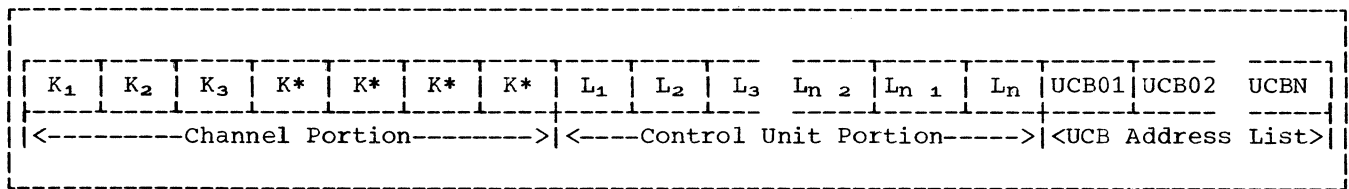


Figure 18. UCB Lookup Table Format

K_n (1 byte)

The channel portion contains index values that are relative to the starting address of the entire UCB lookup table.

index values that are relative to the starting address of the UCB address list.

L_n (1 byte)

The control unit portion contains

UCB_n (2 bytes)

The UCB address list contains the addresses of the UCBs in the system.

UNIT CONTROL BLOCK

Every device is represented by a unit control block (UCB). A UCB indicates device activity and provides the linkage to device-dependent code and tables. The properties of a UCB follow:

- **Creation.** The UCBs are created at system generation time.
- **Storage Area.** All UCBs are located below 32K in protected resident storage (when protection is available), and are a permanent part of the resident supervisor.
- **Size.** A UCB is from 24 to 100 bytes.
- **Quantity.** There is one UCB per unique, physical device address defined for the system. Each telecommunications line on a transmission control unit and a device attached to a shared control unit or a tape unit switch has only one UCB.
- **Format.** The UCB format is shown in Figure 19.

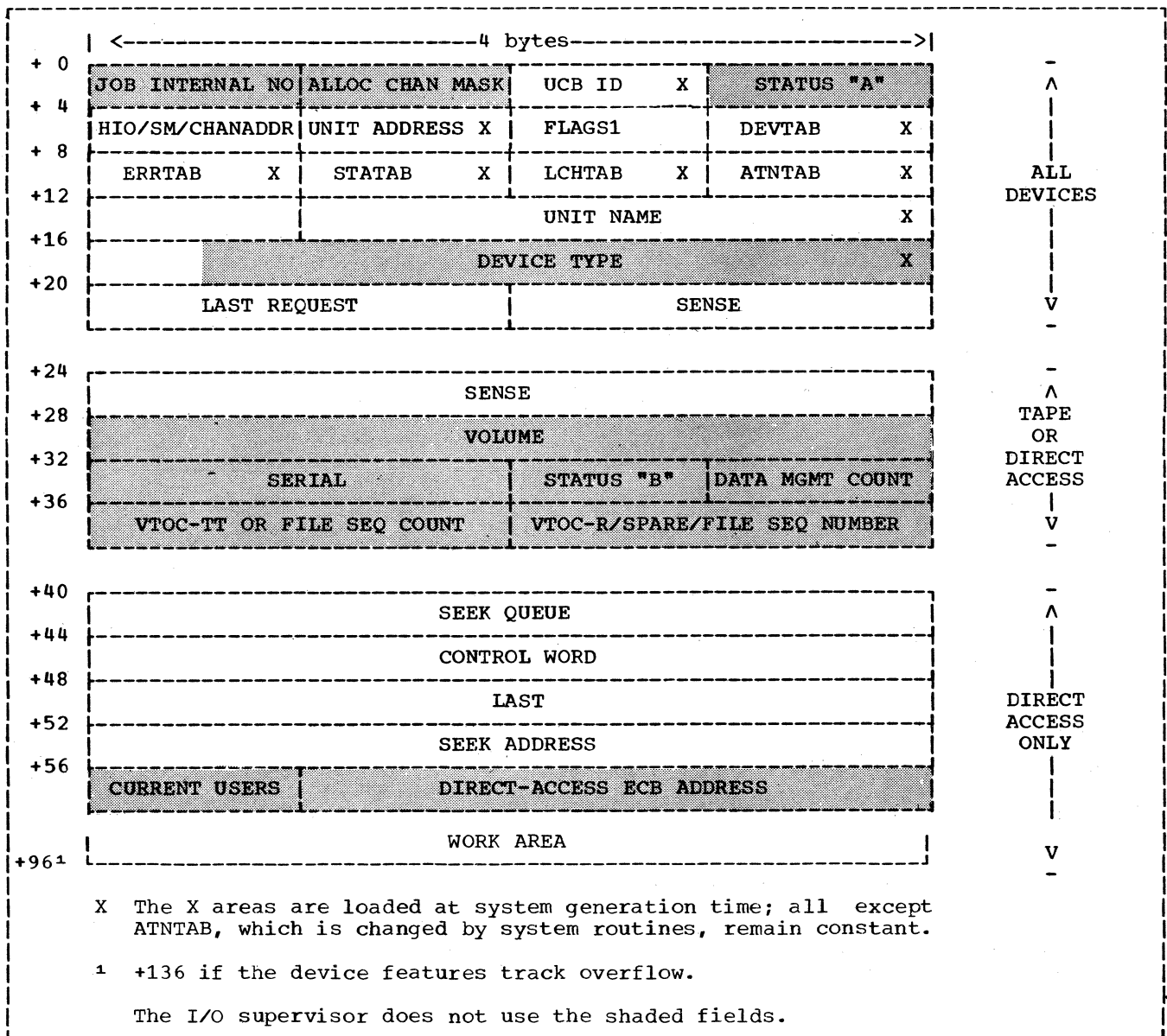


Figure 19. Unit Control Block Format

UCB ID (1 byte)

This is an encoded field of hexadecimal "FF" that identifies this block as a UCB.

HIO (1 bit)

This bit is turned on when a halt I/O (HIO) instruction is issued for the device represented by this UCB.

SM (1 bit)

This bit indicates that a sense command was attempted for the represented device, but that the CSW status modifier flag was set in response to the sense command. When the device is free, the sense is re-issued.

Bits 2, 3, and 4: Not Used.

CHANNEL ADDRESS (3 bits)

Bits 5 to 7 contain the physical channel address of the last physical channel used for the device represented by this UCB.

UNIT ADDRESS (1 byte)

This byte gives the absolute device and control unit address that pertains to the device represented by this UCB. The address of the physical channel to be used by the device is calculated when a physical channel becomes available. The total I/O address, consisting of the unit address and the channel address, is used in the start I/O (SIO) instruction.

Note: A device accessed through a tape unit switch must always have the same control unit address.

FLAGS 1

Bit 0: UCB Busy

This bit is turned on by the I/O supervisor when the represented device is started and remains on until device end.

Bit 1: UCB Not Ready

This bit is turned on when the I/O supervisor determines a device not-ready condition and indicates that operator intervention is required. A message is displayed to the operator by the error routine handling the condition.

Bit 2: Post

This bit indicates either that channel end is awaited, or that an error occurred at device end which will be passed when the next SIO instruction is issued for this device.

Bit 3: UCB Intercept

This bit indicates an error has occurred after channel end for the last request processed on the device and that an error routine must be called when the next request for the device is given. (Refer to IOB intercept code in "Event Control Block".)

Bit 4: Control Unit Busy

This bit is turned on whenever a bit setting of busy and status modifier are detected in the CSW after issuance of a SIO instruction. The setting of the control unit busy bit indicates that requests issued for the represented device by means of the EXCP macro-instruction cannot be started. This bit does not, however, prevent queued requests for the device from being started by the channel restart subroutine. This bit is necessary because the control unit end CSW indication sometimes presents itself to the channel with a different address than was presented with control unit busy.

Bit 5: Disk Data Transfer

This bit indicates, when the UCB represents a direct-access device, that the device is currently involved in a data transfer operation.

Bit 6: Disk Arm Seeking

This bit indicates, when the UCB represents a direct-access device, that the device is currently involved in a seeking operation.

Bit 7: Error Routine in Control

This bit is turned on by an error routine to indicate that error recovery procedures are in progress for the device and that any related requests must not be started.

DEV TAB (1 byte)

This field provides the indexing value (in bytes) from the beginning of the device table to the significant device class entry in the table for the device represented by the UCB. The device table entry contains the addresses of the device-dependent enqueue, start I/O, and trapcode modules for the device type. (For additional information, refer to "Device Table.")

ERRTAB (1 byte)

This byte contains a numeric constant that is used by the exit effector to complete the 8-byte name of an IBM-supplied error routine within the SVC library for the represented device.

STATAB (1 byte)

At SYSGEN time, a device number ranging from 0 to 128 is assigned to each I/O device attached to the system. The number for the represented device is stored in this byte. The statistics table contains an 8-byte data area, in which information regarding the device's operation is maintained, and a 2-byte work area for each device. The I/O supervisor locates each data area by using the product of the STATAB byte and ten as an indexing value. (Refer to "Statistics Table.")

LCHTAB (1 byte)

This field, when multiplied by eight, gives the indexing value (in bytes) from the beginning of the logical channel word table to that logical channel word associated with the represented device. The logical channel word indicates the status of the particular logical channel queue to be used when the I/O supervisor must queue a request for the device. (For additional information, refer to "Logical Channel Word Table.")

ATNTAB (1 byte)

This field contains the indexing value (in bytes), set by system routines, from the beginning of the attention table to one of its 4-byte entries. The entry contains the actual address of the attention routine required for the represented device. (For additional information, refer to "Attention Table.")

UNIT NAME (3 bytes)

This field contains a symbolic name in EBCDIC characters for the device represented by the UCB. This name is used for device allocation and also by error routines in messages to the operator concerning this device.

DEVICE TYPE (4 bytes)

The contents of this field describe the type of device and pertinent characteristics of it. The first 4 bits of the first byte are flags that decode as follows:

Bit 0: Unassigned.

Bit 1: Overrunable Device
When on, this bit indicates that the device is overrunable.

Bit 2: Burst Device

When on, it indicates that the device operates in burst mode; when off, the device operates in byte mode.

Bit 3: Data Chaining

When this bit is on data chaining is allowed for the device. If it is off, data chaining is not allowed.

The remaining bits and bytes are not used by the I/O supervisor; they are discussed in the Introduction to Control Program Logic, Program Logic Manual.

LAST REQUEST (2 bytes)

The I/O supervisor issues a request element to represent each I/O request for a device. When the I/O request is started, the address of the request element is placed into this field (LAST REQUEST) which then indicates the last I/O request for the device. The information in the request element is used during interruption processing. (For additional information, refer to "Request Element Table.")

SENSE (2 to 6 bytes)

When the I/O supervisor issues a sense command, up to 6 bytes of sense information from the device's control unit is read into this area of the UCB. The number of bytes of sense information that is read varies with the device type; however, all information significant to the use of the device is normally provided within the first two bytes. The I/O supervisor then places these two bytes into the IOB.

SEEK QUEUE CONTROL WORD (8 bytes)

This area contains the seek queue control word which gives the status of the seek queue for a direct-access device. The format of this field is shown in Figure 20.

When used for ordered seek enqueueing and dequeuing, the format of the seek queue control word is shown in Figure 21.

LAST SEEK ADDRESS (8 bytes)

This area contains the seek address MBBCCHHR for the last seek on the direct-access device represented by the UCB.

WORK AREA (40 bytes)

This area is used by IBM-supplied error routines for the CCWS necessary for the alternate track procedures and the head and cylinder switching procedures for direct-access devices.

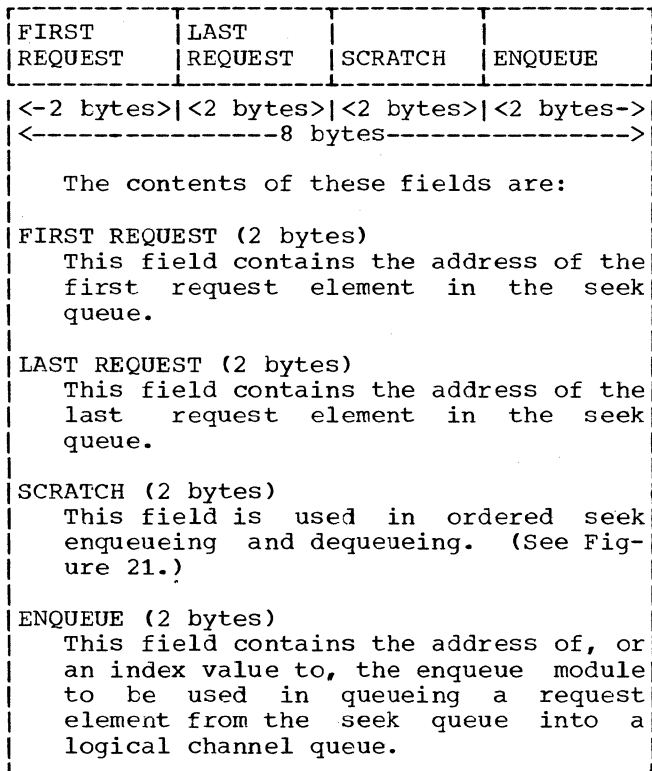


Figure 20. Seek Queue Control Word Format for FIFO and Priority Requests

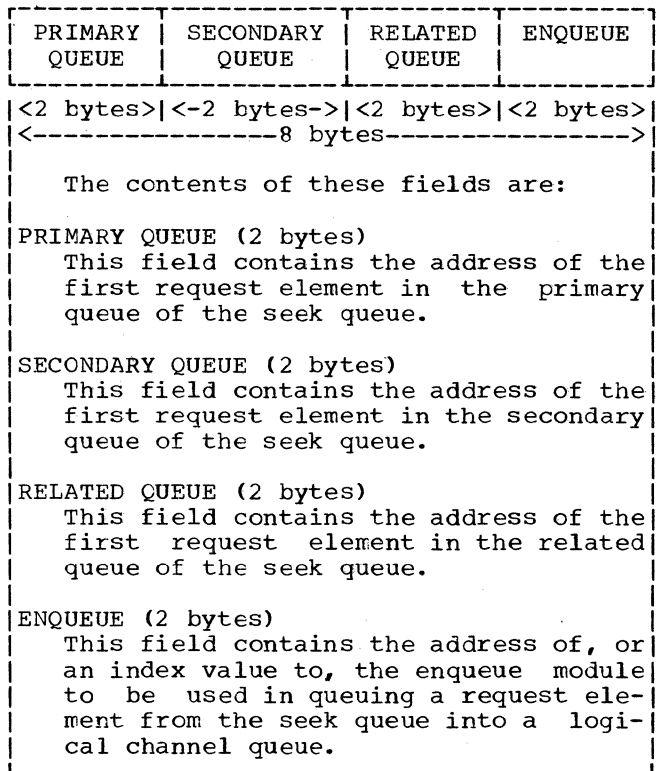
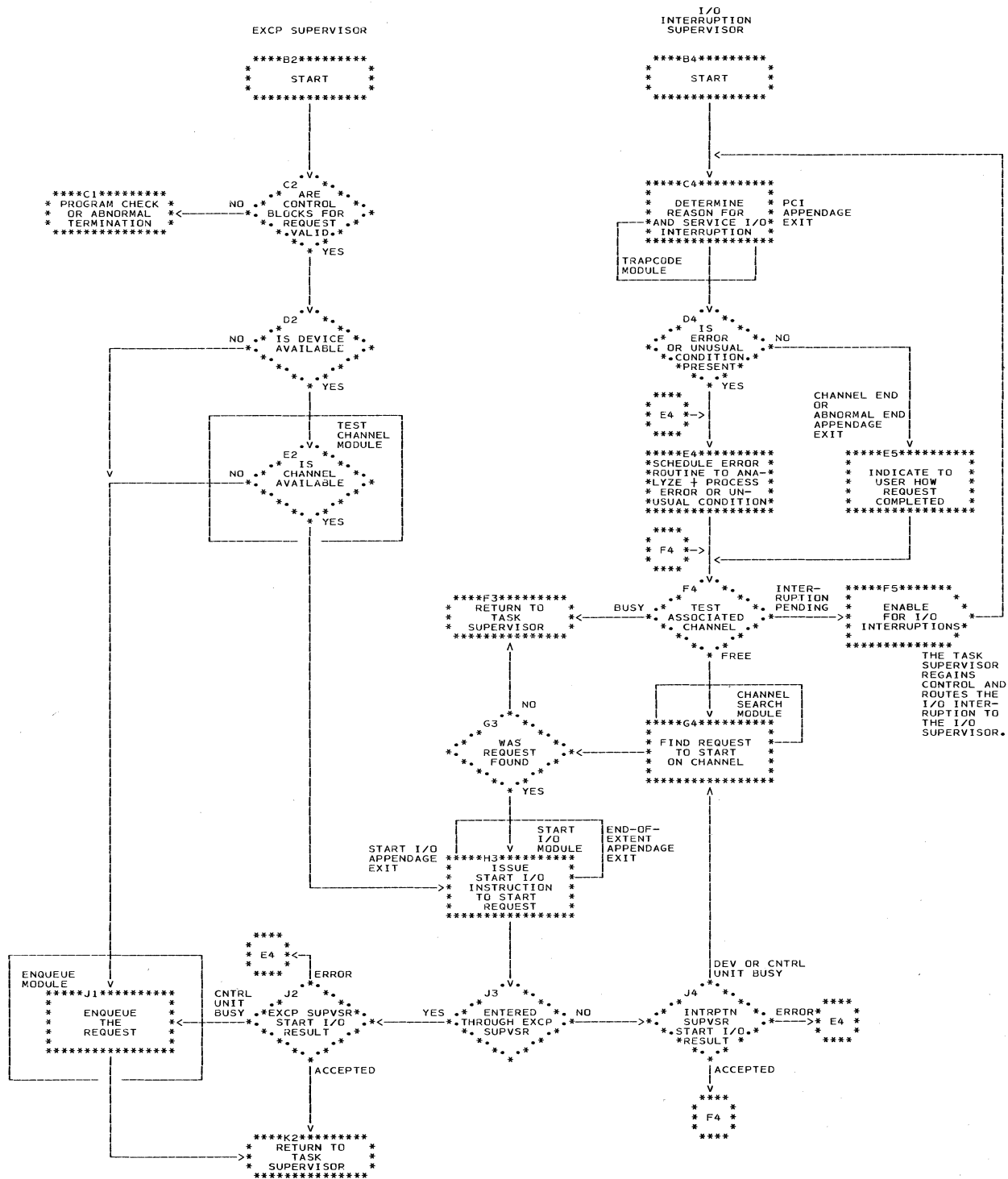
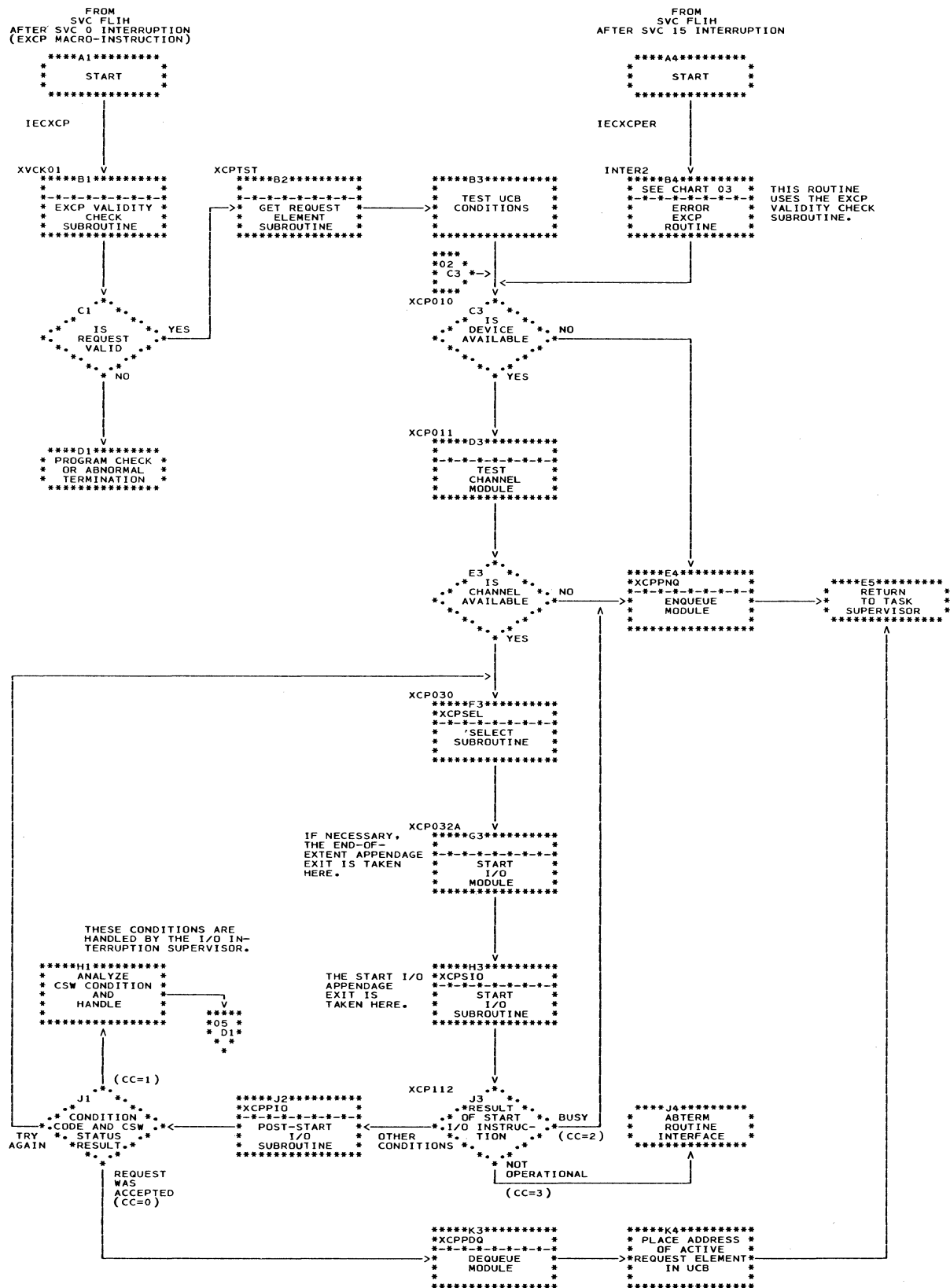


Figure 21. Seek Queue Control Word Format for Ordered Requests

Chart 01. Functional Flow of the Input/Output Supervisor



• Chart 02. Control Flow Within the EXCP Supervisor



• Chart 03. Control Flow Within the Error EXCP Routine

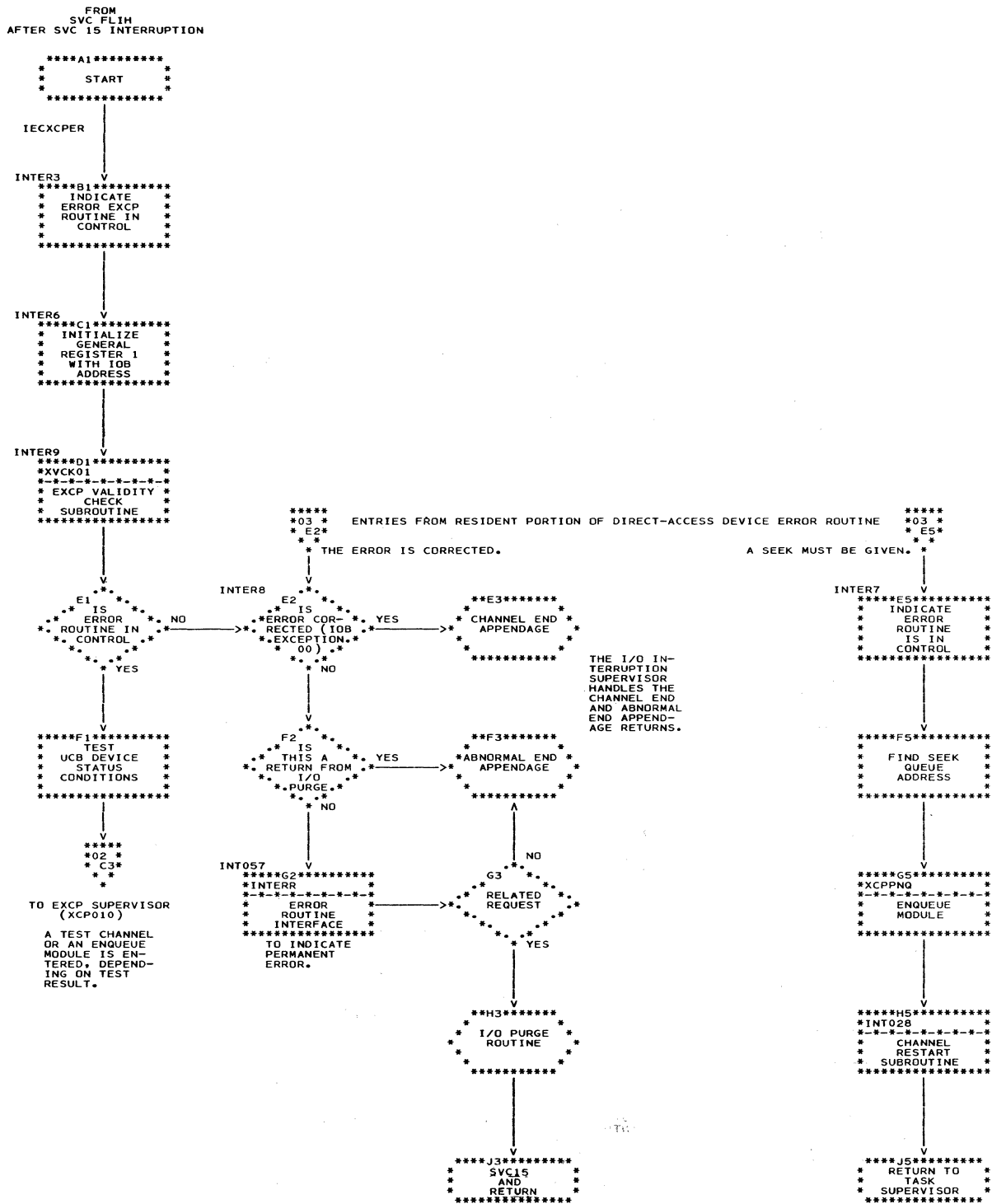


Chart 04. Control Flow Within the Input/Output Interruption Supervisor

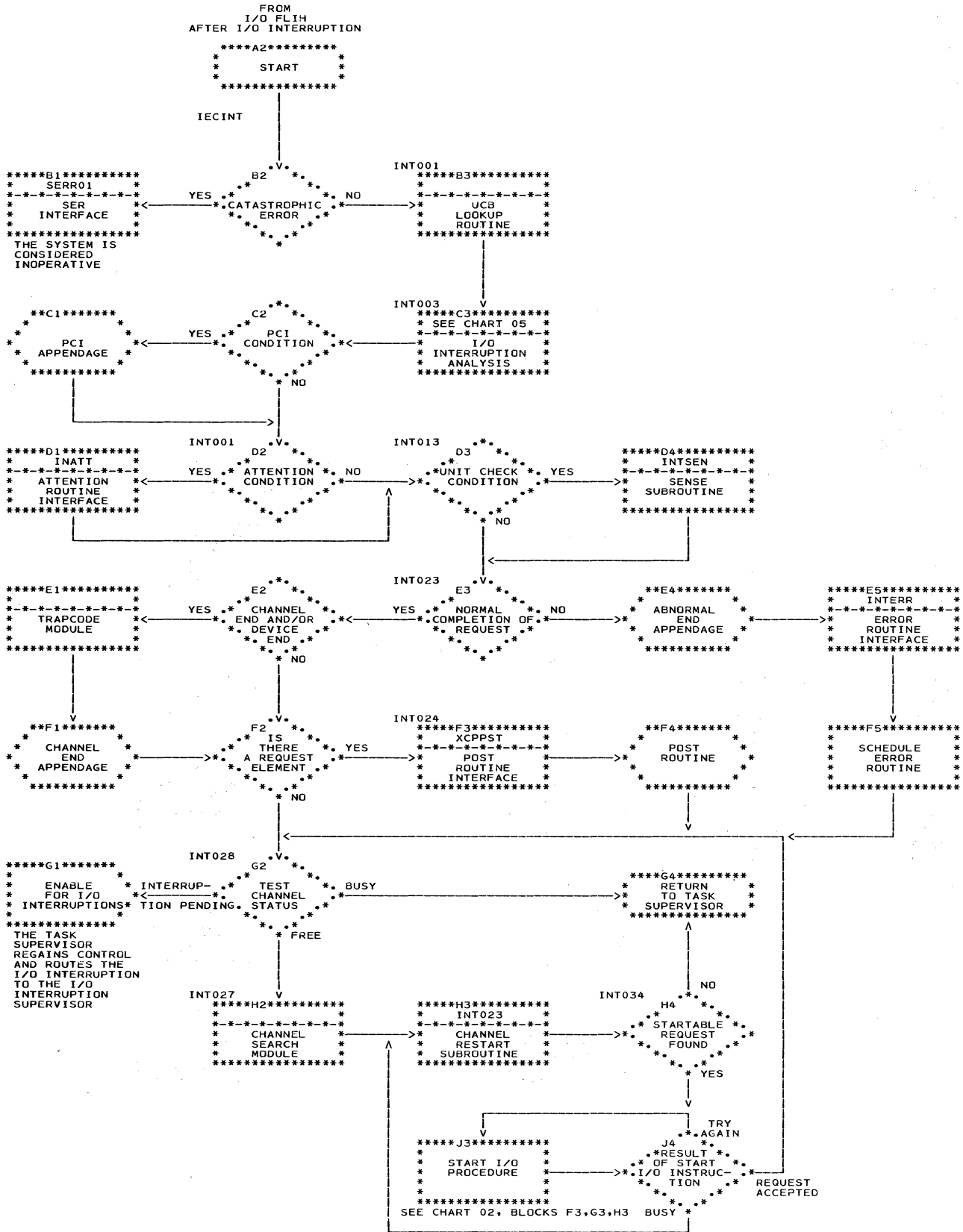
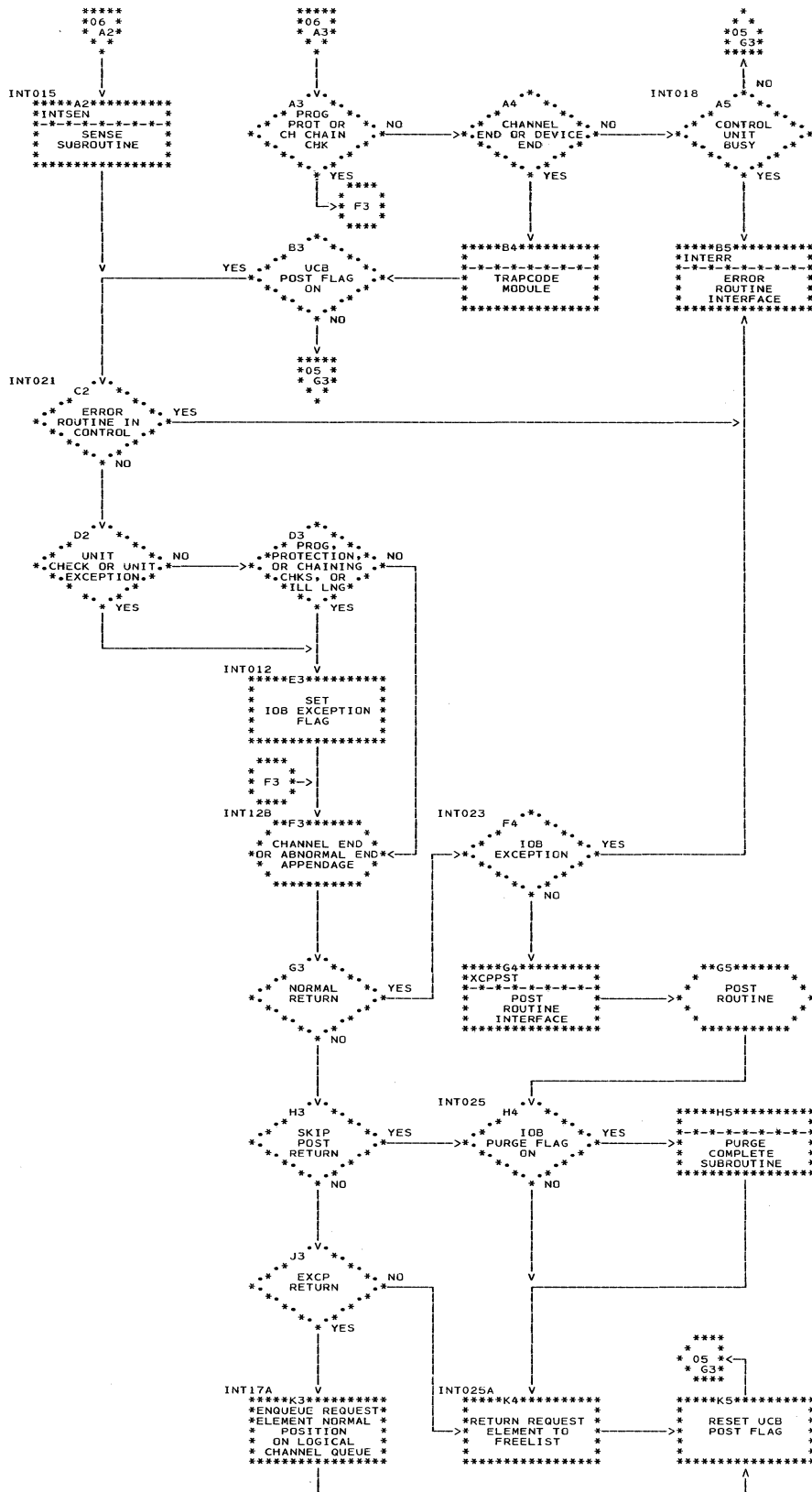


Chart 06. Input/Output Interruption Analysis (2)



• Chart 08. General Operating Procedure for Device-Dependent Error Routines

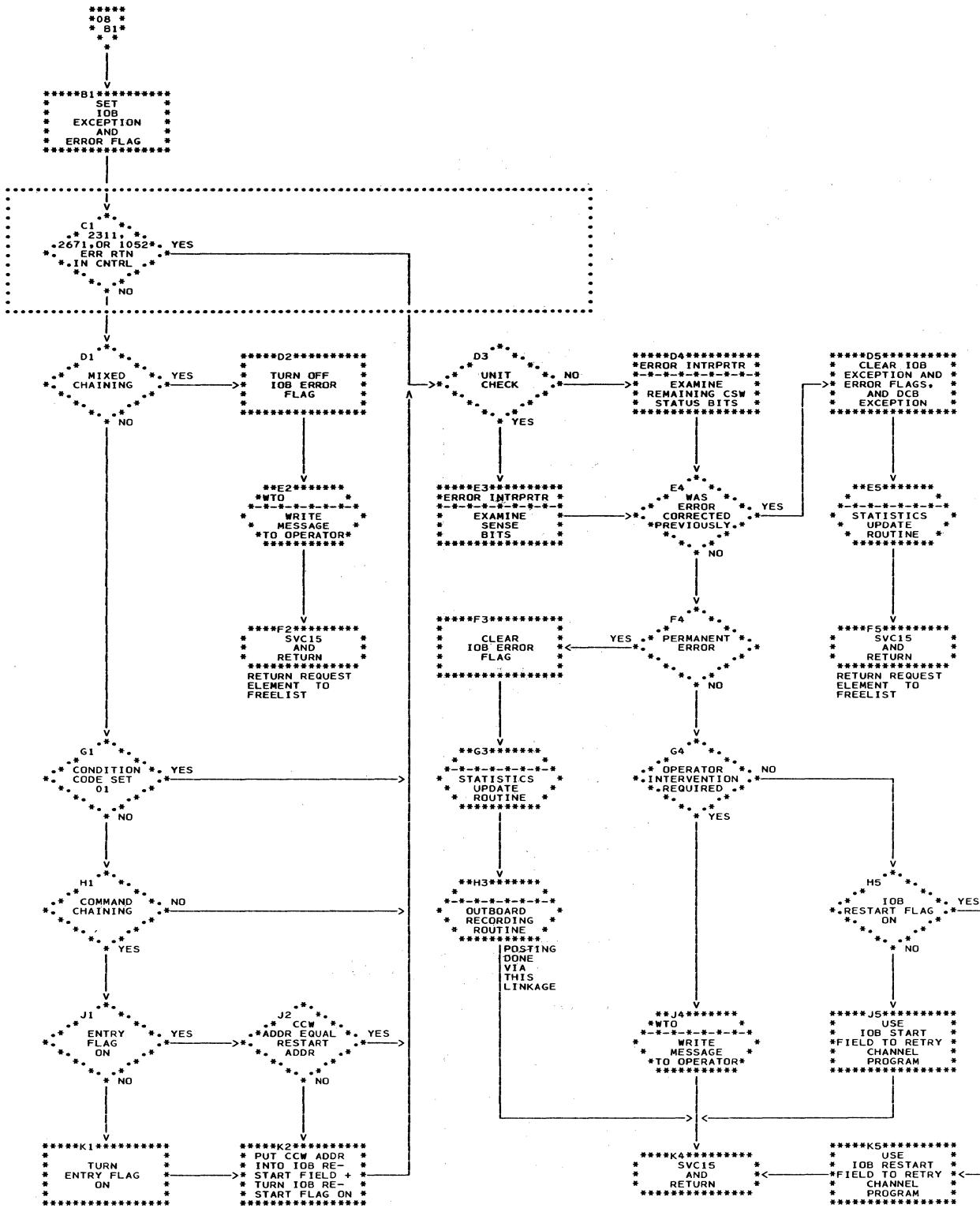
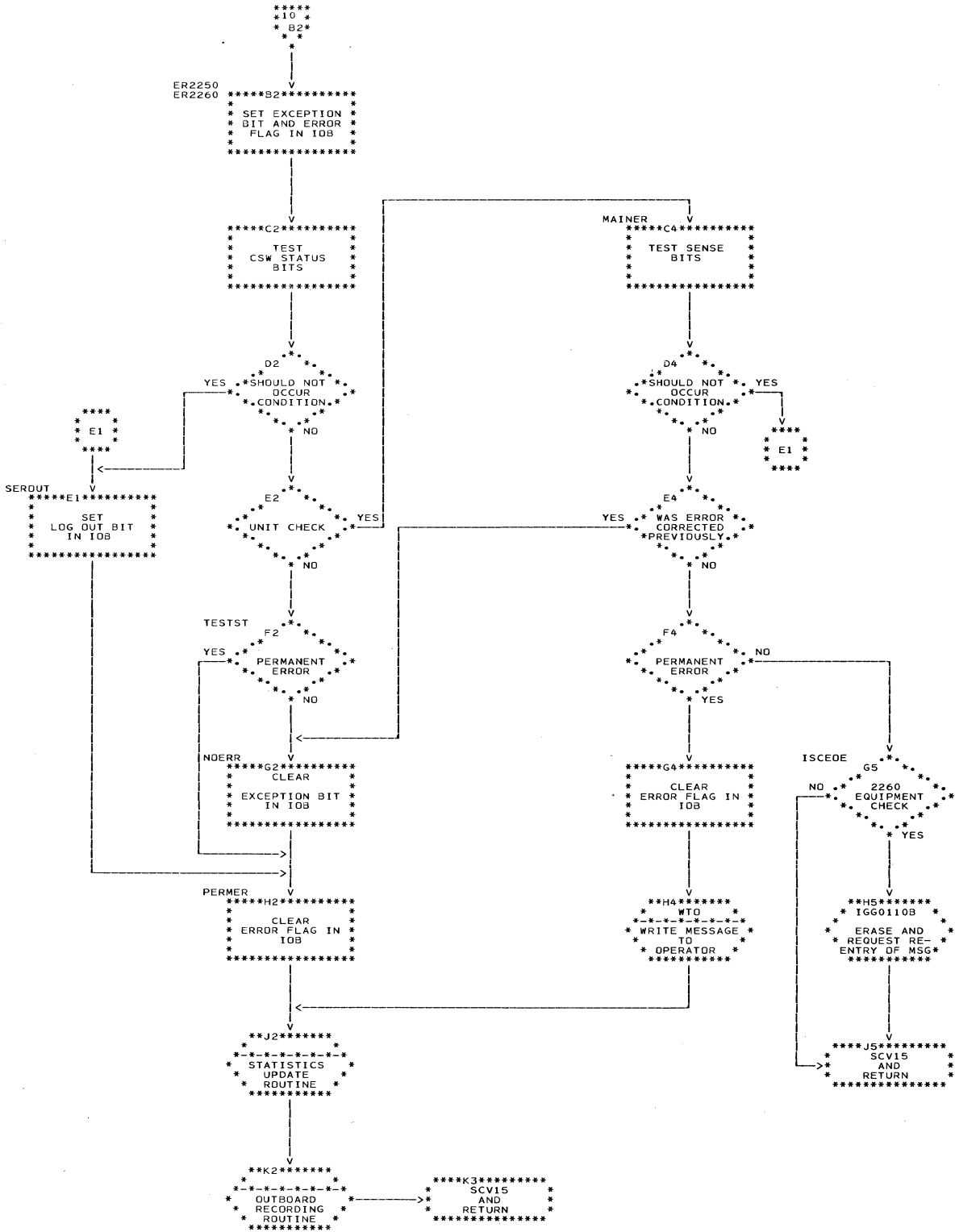


Chart 10. General Operating Procedure-Graphics



APPENDIX A: LOAD MODULE NAMES

The load module names for the I/O supervisor, the error routines, and the purge and restore routines are shown in Table 3.

Table 3. Load Module Names

LOAD MODULE NAME	ROUTINE NAME
IEAASU00	Resident I/O Supervisor and Fixed-Task Supervisor
IEC23XXB	Resident Error Routine for 2311 only (without overflow)
IEC23XXC	Resident Error Routine for 2311 only (with overflow)
IEC23XXD	Resident Error Routine for 2311 and/or 2302, 2303 (without overflow)
IEC23XXE	Resident Error Routine for 2311 and/or 2301, 2302, 2303, 2314 (with overflow)
IGE0000A	Transient Direct-Access Device Error Routine
IGE0001C	Load 1 of 2540/2821 Error Routine
IGE0101C	Load 2
IGE0000D	1052,2150 Error Routine
IGE0000E	1442,2501,2520 Error Routine
IGE0000G	1403/1443 Error Routine
IGE0002?	2671/2822 Error Routine
IGE0000I	Load 1 of 2400 Tape Series Error Routine
IGE0100I	Load 2
IGE0200I	Load 3
IGE0010A	2250 Error Routine
IGE0010B	Load 1 of 2260 Error Routine
IGE0110B	Load 2
IECINTRP	Resident Error Interpreter Routine
IGE0025C	Write-to-Operator Routine
IGE0025D	Statistical Update Routine
IGE0025E	I/O Purge Routine
IGE0025F	Load 1 of OBR Routine
IGE0125F	Load 2
IGE0225F	Load 3
IGE0425F	Error Message Module
IGE0525F	Load 1 of SDR Routine
IGE0625F	Load 2
IGC0001F	SVC Purge Routine
IGC0001G	SVC Restore Routine
IGC0002D	SVC DEVTYPE Routine
IGC0003C	SVC IOHALT Routine

APPENDIX B: LINKAGES WITHIN THE REQUEST ELEMENT TABLE

The request element table contains request elements that represent active I/O requests, available request elements that form a freelist, and queued request elements that are part of either a logical channel queue or a seek queue. The following example provides a detailed explanation of these elements and their linkages within the request element table.

In Figure 22, the request element table contains fourteen elements: four available request elements are in the freelist, two active request elements represent active I/O requests, five queued request elements are in a logical channel queue, and three queued request elements are in a seek queue.

- Available request elements. The freelist pointer contains location 2132, which is the address of the next available request element (first element in the freelist). The request element at location 2132 points to the element at location 2144, which is linked through location 2060, the address of the last available element in the freelist. The last available element is identified by the dummy address of hexadecimal FFFF in its link field.
- Active request elements. The unit control blocks that represent the two devices engaged in active I/O operations contain the addresses of the request elements that represent the I/O requests for the operations. Unit Control Block 1 identifies the request element at location 2108 as the representative of one active I/O request. Unit Control Block 2 identifies the request element at location 2036 as the representative of an active I/O request operating on a direct-access device; Unit Control Block 2 also contains the seek queue control word for that direct-access device.
- Logical channel queue. The logical channel word for the logical channel queue identifies the first and last request elements in the queue. The first request element in the queue resides at location 2024 and this ele-

ment is linked through location 2084, the address of the last element in the queue. The last request element is identified by the dummy address FFFF contained in its link field.

- Seek queue. The seek queue control word, located in Unit Control Block 2, identifies the first and last request elements in the seek queue. The first request element in the seek queue resides at location 2120 and is linked through location 2156, the address of the last element in the seek queue. The last element in the queue is identified by the dummy address FFFF contained in its link field.

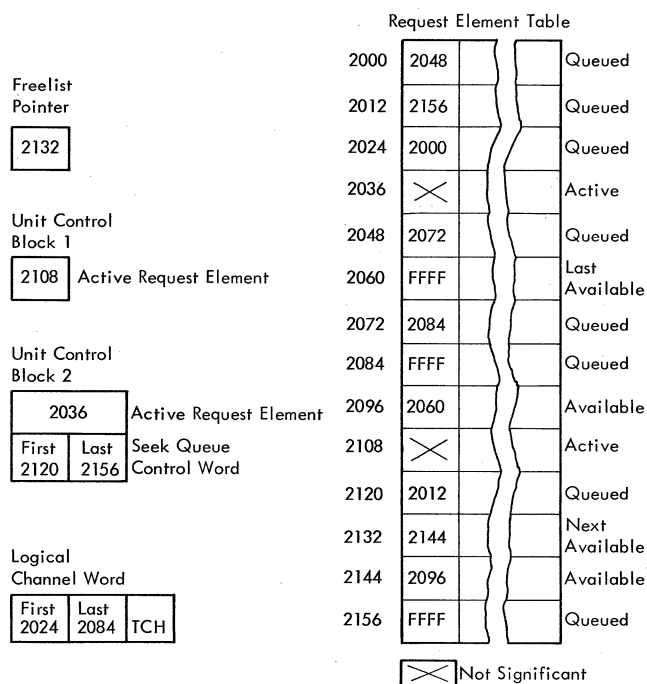


Figure 22. Linkages Within the Request Element Table

APPENDIX C: THEORY OF LOGICAL CHANNELS

Data that is transferred from the central processing unit to an I/O device follows a path through a physical channel. An I/O device can be switched among several physical channels, each of which provides a separate path to the device. The unique collection of all physical channel paths to a particular device is called a logical channel. If more than one device is included in the same set of physical channel paths, those devices are included in the same logical channel.

A physical channel is included in one or more logical channels, although a particular I/O device is included in only one logical channel. In Figure 23, for example, two logical channels (A and B) are shown. Logical channel A consists of the paths through physical channels 1 and 2 to device A, and logical channel B consists of the path through physical channel 2 to devices B and C. The devices and physical channel 1 are included in only one logical channel, whereas physical channel 2 is included in both logical channels.

The physical channel path to a particular I/O device may vary because of channel switching and subchannel considerations. Channel switching occurs when a device is switched, for operation purposes, from one physical channel to one of the other physical channels to which it is attached. In Figure 23, device A may be switched between physical channels 1 and 2. Channel switching does not hold for a multiplexor channel.

A subchannel is the physical channel facility required for a single data transfer operation between the CPU and an I/O device.

A selector channel has only one subchannel and the attached devices share the subchannel. A selector channel is used for only one data transfer operation at a time; however, other I/O devices attached to the selector channel can be simultaneously executing control operations. In Figure 23, if physical channel 2 was a selector channel, device B could maintain a data transfer operation while device C was executing a control operation, and vice versa.

A multiplexor channel, although having only one logical channel, has more than one subchannel, and the devices attached to it can sustain concurrently one I/O operation for each subchannel. In Figure 23, if physical channel 2 was a multiplexor channel, devices B and C could operate simultaneously. Note that device A could not operate because channel switching does not apply for a multiplexor channel, and because a multiplexor channel has only one logical channel.

Various physical channel paths and their logical channels are shown in Figure 24 and are described in the following list. Assume that physical channel 0 is a multiplexor channel and that physical channels 1 through 5 are selector channels.

- Logical Channel A is defined for the paths from physical channel 0 and its subchannels to the tape units on con-

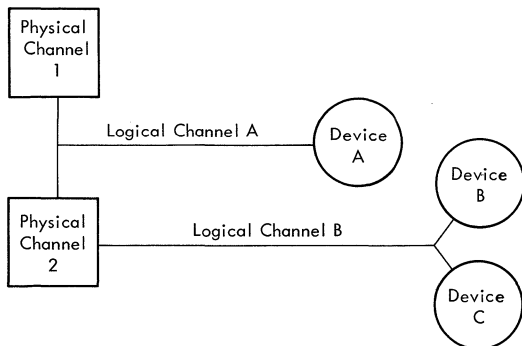


Figure 23. Logical Channels

trol unit 0 (the tape units share a subchannel and operate one at a time); to the reader, punch, and printer, respectively, on the integrated control unit (the reader, punch, and printer each use a separate subchannel and operate simultaneously); to two of the telecommunications lines on the transmission control unit (each telecommunications line uses a separate subchannel).

- Logical Channel B is defined for the path from physical channel 1 to the I/O devices on control unit 1.
- Logical Channel C is defined for the paths from physical channels 1 and 2 to the I/O devices on control units 2 and 3. Because control units 2 and 3 use the same physical channel paths, each device attached to these control units is in the same logical channel.

- Logical Channel D is defined for the paths from physical channels 1 and 3 to the I/O devices on control unit 4.

- Logical Channel E is defined for the paths from physical channels 3, 4, and 5 to the tape units attached to the tape unit switch.

Note: All control units on a tape unit switch must be in the same relative position on their respective physical channels (e.g., position 5 in Figure 24). This is the only unit address assignment requirement. The I/O supervisor requires such positioning for efficient interruption handling.

- Logical Channel F is defined for the path from physical channel 3 to the I/O devices on control units 6 and 7. Because the same physical channel paths are used for control units 6 and 7, each device attached to these control units is in the same logical channel.

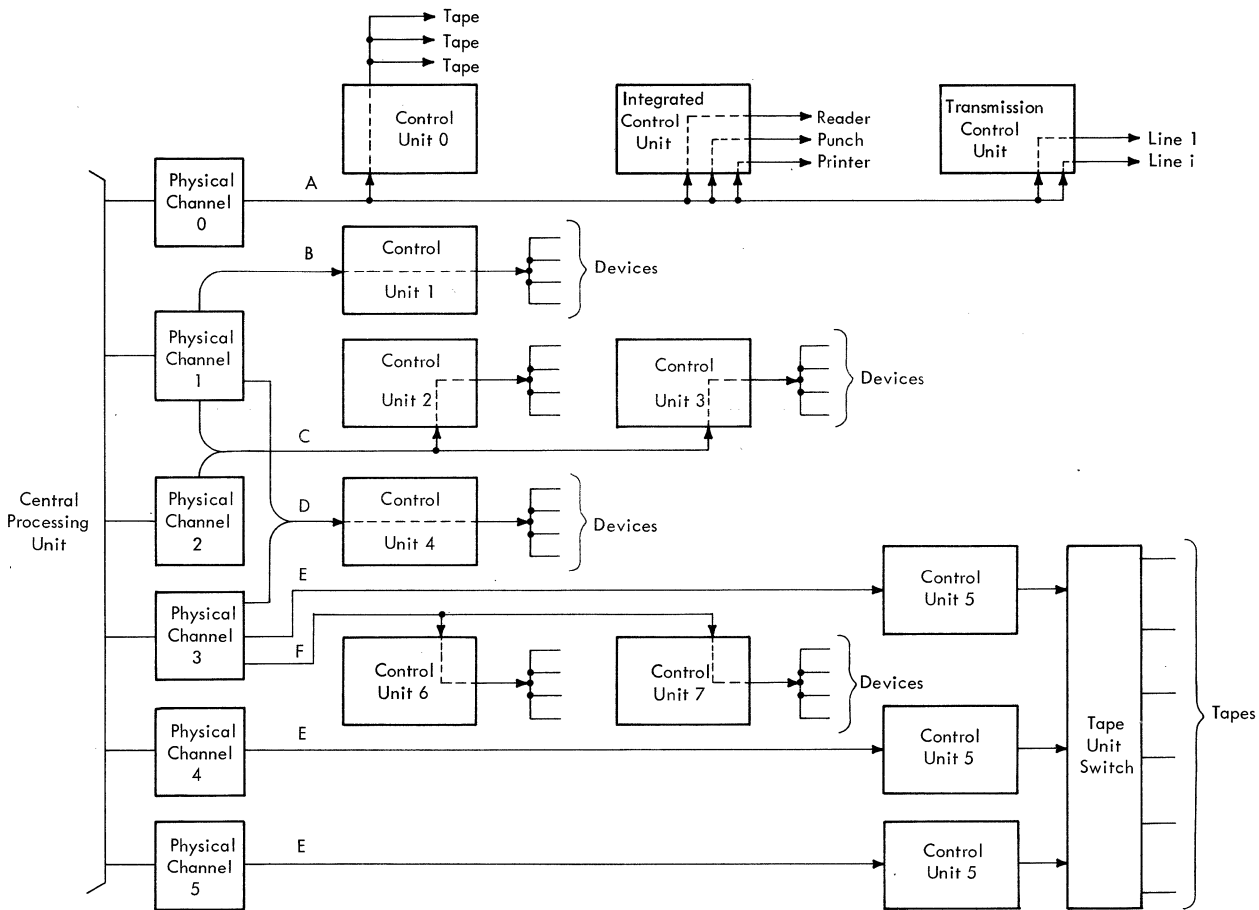


Figure 24. Physical Channel Paths and Logical Channels

Extent Limits (8 bytes)
contains the addresses (CCHH) of the first and last tracks of the data set.

Highest Address (1 byte)
contains the highest address of a track on a cylinder for the particular system residence device.

First Record Entry (7 bytes)
contains the address (BBCCHHR) and ID of the first track of the record entry area. Initially, the ID is zero.

Remaining Bytes (2 bytes)
contains the number of bytes remaining on the track upon which the last record entry was written. Initially, this field contains the track capacity.

Track Capacity (2 bytes)
contains the number of bytes per track for the particular system residence device.

Last Record Entry (7 bytes)
contains the address (BBCCHHR) and ID of the track of the last record entry written in the extent.

UCB Count (2 bytes)
contains the number of UCBs in the system.

Extent Size (2 bytes)
contains the number of tracks in the data set extent.

Device Type (1 byte)
contains an encoded field that designates the type of system residence device as follows:

<u>Device</u>	<u>Code</u>
2311	1
2301	2
2314	8

Safety Field (1 byte)
should contain hexadecimal FF. If an overrun condition, caused by a machine check or channel inboard failure, occurred while the header record was being written, the header record is padded with zeros from the point of overrun. Thus, if this field is zero, critical data has been destroyed and safety precautions are taken to halt and prevent further recording attempts.

STATISTICAL DATA RECORDS

The statistical data records follow the header record. Each is preceded by a 2-byte key. The keys are sequentially numbered, starting with zero, and are the index values to the statistics table entries associated with the particular statistical data record.

The statistical data records have the following characteristics:

- Creation. The extent is formatted by the SYS1.LOGREC initialization program and each record is created and written by SDR.
- Storage Area. The statistical data records reside on the system residence device.
- Quantity. There is one statistical data record for each UCB in the system.
- Size. Each statistical data record is 38 bytes in length and is preceded by a 2-byte key.
- Format. The format of a statistical data record is shown in Figure 26.

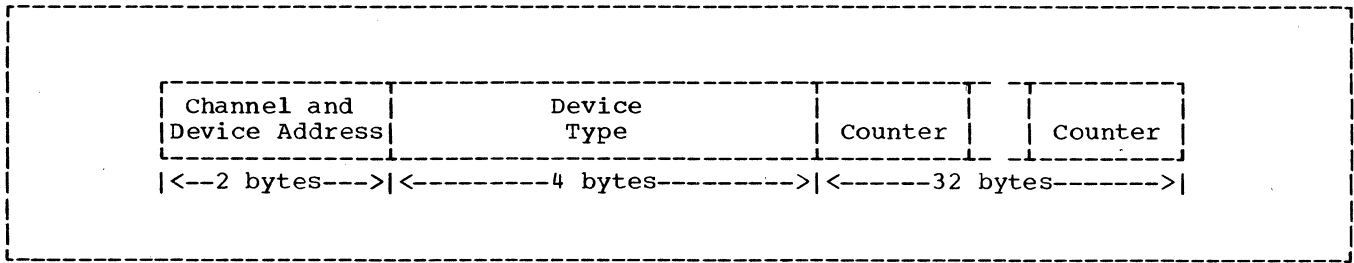


Figure 26. Statistical Data Record Format

Channel and Device Address (2 bytes)
contains the address of the unit associated with this particular statistical data record.

Device Type (4 bytes)
contains an encoded device type field for the device associated with the statistical data record. The field is the same as the UCB device type field.

Counters (32 bytes)
contains sixteen 2-byte counters. Each 4-bit counter within the statistics table of the I/O supervisor is expanded to a 2-byte counter on the statistical data record.

These sixteen counters are the summaries of the 4-bit counters and provide a device error history.

RECORD ENTRIES

The record entry area starts on the next consecutive track after the last statistical data record. The OBR writes into it I/O outboard fixed-length record entries containing information pertinent to the type of device in error.

A record entry has the following characteristics:

- Creation. The area is allocated by the SYS1.LOGREC initialization program and each record is created and written by OBR.
- Storage Area. The record entries reside on the system residence device.
- Size. The length of the record entry area is fixed for each type of system residence device. It depends upon the number of tracks required to receive approximately 100 I/O outboard record entries.
- Format. The format of a record entry is shown in Figure 27.

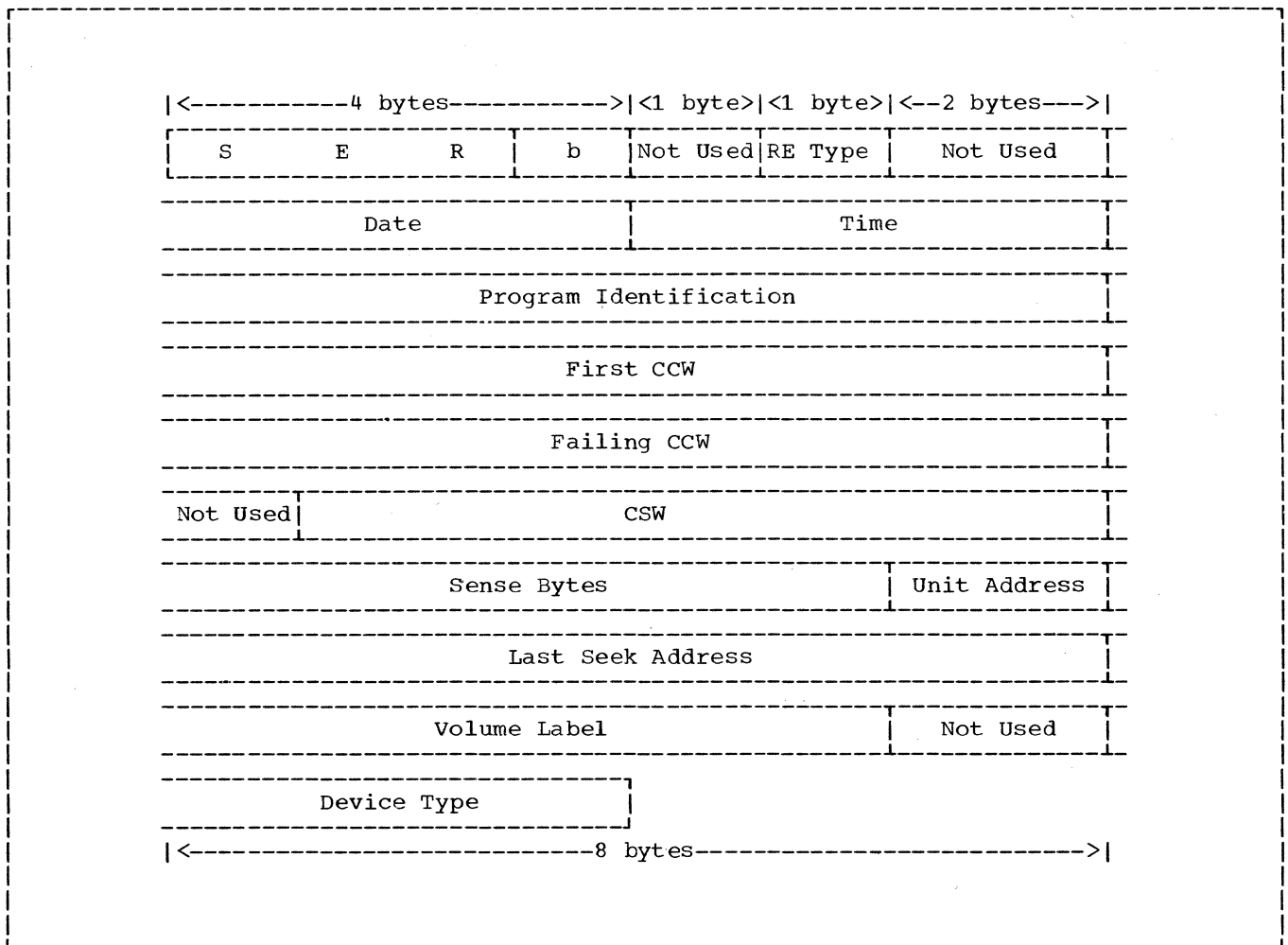


Figure 27. Record Entry Format

Record Entry Label (3 bytes)
contains the record entry ID field of the EBCDIC characters "SER." The label identifies the record as a dynamic error environment output record of OBR, SER0, or SER1.

System Identifier (1 byte)
contains a blank. This field identifies the SER routine (OBR) that wrote the record. The identifier for SER0 is 0, and SER1 is 1.

Record Entry Type (1 byte)
contains an EBCDIC character identifying the type of error that caused the record to be written. The characters are:
CPU machine check (C)
Channel inboard error (I)
I/O outboard error (O)

Date (4 bytes)
contains the current date in packed decimal format (00YYDDZ) as specified in the CVT.

Time (4 bytes)
contains the current time of day after conversion of the pseudo-clocks to binary format.

Program ID (8 bytes)
contains the program identification as specified in the request block for the program in progress at the time of the error.

First CCW (8 bytes)
contains the first CCW in the user's channel program as specified in the

IOB addressed by the request element provided by the IBM-supplied error routine.

Failing CCW (8 bytes)
contains the failing CCW within the channel program as computed from the CSW command address stored in the IOB.

CSW (8 bytes)
contains seven bytes of the CSW (the first byte is not available), stored by the I/O supervisor in the IOB for the I/O request.

Sense Bytes (6 bytes)
contains the sense information stored by the I/O supervisor in the UCB representing the device in error.

Channel and Device Address (2 bytes)
contains the channel and device address for the failing device as specified in the UCB representing that device.

Last Seek Address (8 bytes)
contains the last seek address, as specified in the IOB, for a failing direct-access device.

Volume Label (6 bytes)
contains the volume label when the failing device is tape or direct-access.

Device Type (4 bytes)
contains the device type field as specified in the UCB for the device in error.

- Abnormal termination
 - see ABTERM routine interface
- ABTERM routine interface 36
- ABTERM service routine 36
- Access arm 16-17
 - movement of 22-25
- Access methods 8
- Action code 49
- Active request element 16
 - definition of 29
 - example of 85
- Adcon
 - see address constants
- Address constants
 - in appendage vector table 60
 - in error interpreter 38-39,48
- Address markers 48
- AEQ
 - see asynchronous exit queue
- AEQ purge subroutine 53
- Algorithm
 - see UCB lookup algorithm
- Alternate track procedures 45,47
- ALU check 46
- Appendage vector table 10,60
- Appendages 10
 - abnormal end 31
 - channel end 30
 - end-of-extent 26
 - PCI 30
 - start I/O 26
- Assembly time 7,57
- Asynchronous 35
- Asynchronous exit queue 53
- ATNTAB
 - definition of 73
 - use of 34,55
 - see also attention table; unit control block
- Attention
 - interruptions 15
 - routines 15,34-35
 - address of 55
 - table 15,
 - format of 55
- Available request element 15-16
 - example of 85
- Bell, console 39
- BLKCT 57-58
- Block count
 - definition of 57-58
 - updating 18,31
- Block count increment amount
 - definition of 65
 - resetting 18
 - use 31,58
- Block size, tape 43
- Build area 52
- Burst device 32-33
 - indication of 73
 - request for 23
- Bus out check 39-46
- Busy, CSW status bit
 - see channel status word
- Byte device 33
- Catastrophic errors
 - definition of 11,29-30
 - handling of 36,29-30
 - occurrence of 27,34,51
- CAW
 - see channel address word
- CCW
 - see channel command word
- CCW list 12-13,63
 - retry 38
 - see also channel program
- Chaining 38,63,82
- Channel
 - address 72
 - available 32
 - enabling of 32,56
 - end 11,30,27
 - restart 11,32-33
 - search 6,32
 - Channel address word 24-26
 - Channel command word 24-26,34
 - in error 38,45,58
 - valid sequences 47
 - Channel mask 32,56
 - Channel 9 42,57
 - Channel program 10
 - chained 38,45,63
 - completion 11,28-30
 - execution 26-27
 - initiation 10-11,23-26
 - interruption 28
 - purged 35-36,37
 - related 13,24,36,64
 - retry 37-38
 - Channel program description bits 18,38
 - definition of 63
 - Channel search module address 56
 - Channel status word 29-31,34,64,79-80
 - status bits 38-39,79-80
 - attention
 - see attention
 - busy 27,31
 - chaining check 31,58
 - channel control check 11,29-30
 - channel data check 11,29-30
 - channel end 11,30,27
 - control unit end 11,27,29-30
 - device end
 - see device
 - incorrect length 31
 - interface control check 11,29-30
 - program check 27
 - program-controlled interruption 11
 - protection check 27
 - status modifier 11,31,34,72
 - unit check 11,27,31,34,39
 - unit exception 31

Channel switching 86-87
 Channel table 14
 format of 56
 Channel tag line check 46
 Channel 12 57
 Check subroutine 53
 Command chaining
 see chaining
 Command reject 39-46
 Communications vector table 16
 Complete bit 62
 Completion
 see channel program
 Completion code 36,62
 see also ECB
 Condition code 26,38,64
 Console alarm routine 39
 Console bell 39
 Console typewriter errors 37,39
 Contents supervisor 35,38
 Control unit busy flag 31,72
 Control unit end
 see channel status word
 Counters
 see statistics counters
 CRC
 see cyclic redundancy check
 Current cylinder address 22-23
 CVT
 see communications vector table
 Cyclic redundancy check 25,44,45,64
 Cylinder address 22-23,60
 Cylinder switching 45,73

 Data chaining 73
 see also chaining
 Data check 41-46
 Data control block 12-13,57
 Data converter check 45
 Data extent block 12-13,57,58,59
 Data record 43
 Date 52,92
 DCB
 see data control block
 DEB
 see data extent block
 DEBAD 57-59
 Defective track procedures 45,47
 Dequeuing 21-23
 Device
 address 72
 available 32
 end
 for seek 25,32
 unsolicited 15,30
 type 73
 table 14,15
 format of 61
 DEVTAB 61,72
 Disk arm seeking flag 25,32-33,72
 Disk data transfer flag 26,32,72
 Dispatcher 28
 Dispatching priority
 see priority
 Dummy address
 definition of 16-17
 example of 85
 uses of 20,22,32

 ECB
 see event control block
 ECB code 64
 Enable/disable 16
 End-of-cylinder 47
 End-of-extent 25-26
 Enqueue module address 61
 Enqueuing 21-23
 Entry flag 38,82
 Equipment check 39-46
 ERREXCP macro-instruction 18,20,38
 Error counts 65
 Error flags 57-58
 Error indicators 38
 Error interpreter 38-39
 Error key 35
 Error messages
 see messages
 Error recovery procedures 24-25,35,37-52
 Error routines 11-12,20,37-52
 entry to 35,58
 scheduling 36,81
 ERRTAB
 definition of 73
 use of 35-36,81
 Event control block 13
 format of 62
 Exception bits
 DCB 18,38
 definition of 57
 IOB 31,38,39
 definition of 64
 EXCP macro-instruction 18
 exit effector 35,81
 extent 25,60,65

 FIFO queuing 17,21,22
 File mask 25,60
 File protected 48
 First request 66
 Flag byte check 48
 FLIH 8,9,18,28
 Freelist 16,85
 Freelist pointer 16,85

 Halt I/O 53,58,64,72
 Header record 51-52
 Head register 25
 Head switching 45,47,73

 IEA000A 49
 IEA000I 49
 IECILK1 29
 IECILK2 29
 IFB001I 52
 IFB003I 52
 IFB004I 52
 IFLGS
 see error flags
 Immediate operations 26,27
 Initial seek address 13,17
 Input/output block 13
 format of 63-65
 Intercept flags
 UCB 24,72
 IOB 62
 Internal component name 49
 Intervention required 39-46

Invalid sequence 47
 IOB
 see input/output block
 IOB flags 39-40
 I/O address 14,29,72
 I/O FLIH 28
 I/O interruption 11
 analysis 29-31,79,80
 pending 32
 I/O purge 35-36
 I/O purge routine 20
 I/O request
 see channel program
 I/O supervisor transient area 7,12

 K 29,70

 L 29,70
 Last available 16
 Last request 66,73
 Last seek address field 23,73
 LCHTAB 66,73
 Link field 16,67
 Log out 51-52
 Logical channel 86-87
 Logical channel queue 16,22
 example of 85
 Logical channel word 16,66
 Logical channel word table 14,16,66

 Machine log out area 51-52
 Messages 39,48-49,52
 message text 49
 Mixed chaining
 see chaining
 Modifier byte 24,26,63-65
 Modules 8-9,84
 Multiplexor channel 16,21,32-33,86-87

 Next available 16,20
 example of 85
 9-track tape 24,45
 Noise 43-44

 OBR 50-52
 OBR interface 50
 Open routine 26,57
 Operator intervention 28,39,49
 Option 4 20,67
 Ordered queuing 17,21,22-23
 Ordered seeks 22
 Outboard error 50-52
 Outboard record 91
 Overflow incomplete 45-46,48
 Overflow record 46
 Overrun 23,32-33,42,45-46
 Overrunable 33,73

 Paper tape reader 43
 PCI 11
 Permanent error 36,39
 Post 72
 Post routine 36
 Posting 13,28,38
 Primary queue 23
 Printer overflow 57
 Priority 20,22,34,60,67
 Priority queuing 17,21,22

 Program check
 see channel status word
 Protection tag 60
 Pseudodisable 20
 PSW 18,28,36
 Purge
 completion 36
 options 53
 posting 54
 Purge chain
 DEB 24,25,53,60
 system 36,37,60
 user 60
 Purge complete subroutine 54
 Purge flag 54,64
 PURGE macro-instruction 53-54
 Pushup queue 17

 QSAM 40
 Queued request element 16
 example of 85
 Queuing 17,21-23
 Quiesce 54,64

 Record entries 51-52,89,91
 Related queue 23
 Related request
 see channel program
 Repositioning
 disk 45
 tape 24,26,44
 Request block 53
 Request block subroutine 53
 Request element 15-16,20
 format of 67
 Request element table 14,15-16
 example of 67
 format of 67
 Restart address 38,65
 RESTORE macro-instruction 53-54

 Safety field 89
 Scratch field 34,66
 SDR 50-52
 SDR interface 50
 Secondary queue 23
 Seek 25-26
 end 32
 initiation 25,33
 overlap 33
 stand alone 25,32
 start data transfer 25-26,32
 Seek address 22-23,17,65
 Seek check 45,47
 Seek overlap
 see seek
 Seek queue 16-17,22
 example of 85
 Seek queue control word 17,22,32,33
 format of 73-74
 Selector channel 16,21,32-33,86-87
 Sense 34
 Sense area 64
 Sense bytes 39-48
 Sense command 11,14,34
 Sense flag 34,64
 Sequential cylinder 45
 SER 11,30,36,50,88

SER disk area 52
 SER output data set 52
 format of 88-89
 SER recording area 52
 SER0 52,92
 SER1 52,92
 Serial number 49
 Serializer/deserializer check 46
 Set file mask command 24,25
 Set mode command 24
 SIO instruction 26,34,39
 with error 38
 SIRB 35
 Split cylinder 45
 Stand alone CCW 24,25
 Standard message code 49
 Start address 18,64
 Start I/O module address 61
 Start/restart flag 64
 STATAB 69,73
 Statistical data records 15,39,51-52
 format of 90
 Statistics counters 15,39,49-50
 format of 68-69
 Statistics table 14,15,49-50
 format of 68-69
 work area 50
 Statistics update 34
 Status bits
 see channel status word
 Status modifier
 see channel status word
 Subchannels 86-87
 SVC FLIH 18
 SVC interruptions 8,9
 SVC library 35
 SVC transient area 7,8,12,54
 SYS1.LOGREC initialization program 52,88

Tape block count
 see block count
 Tape record 43
 Tape unit switch 71-72,87
 TAU 25
 TCB identification 20,67
 TCH instruction 21
 Temporary errors 48,68-69

TIC command 24,26
 TIO instruction 25,34,39
 Track condition check 45,47
 Track overrun 47
 Transient 35-36
 Transient areas 7,8
 Trapcode module address 61
 Try again 32
 Type bar 42
 Type 3 SVC 53

UCB
 see unit control block
 UCB busy 72
 UCB condition test 20
 UCB ID 72
 UCB intercept flag 24,30,62,72
 UCB lookup 28-29
 UCB lookup algorithm 29
 UCB lookup table 14
 format of 70
 use of 29
 UCB not-ready 72
 UCB purge subroutine 53
 Unit address 72
 Unit check
 see channel status word
 Unit control block 13-14
 address of 29,60
 format of 71-74
 Unit name 73
 Unrelated flag 18,24
 definition of 64
 Unselected status 46
 Unsolicited device end interruption 15,30
 Unusual conditions 11,28

Validity checking 19

Wait bit 36,62
 WAIT macro-instruction 36
 Wait state 11,39
 Word count zero 45
 WTO macro-instruction 48

12*
 see request element table



International Business Machines Corporation
Data Processing Division
 112 East Post Road, White Plains, N.Y. 10601
 [USA Only]

IBM World Trade Corporation
 821 United Nations Plaza, New York, New York 10017
 [International]

READER'S COMMENTS

Title: IBM System/360 Operating System
Input/Output Supervisor
Program Logic Manual

Form Y28-6616-1

Is the material:	Yes	No
Easy to Read?	___	___
Well organized?	___	___
Complete?	___	___
Well illustrated?	___	___
Accurate?	___	___
Suitable for its intended audience?	___	___

How did you use this publication?
___ As an introduction to the subject ___ For additional knowledge
Other _____ fold

Please check the items that describe your position:

___ Customer personnel	___ Operator	___ Sales Representative
___ IBM personnel	___ Programmer	___ Systems Engineer
___ Manager	___ Customer Engineer	___ Trainee
___ Systems Analyst	___ Instructor	Other _____

Please check specific criticism(s), give page number(s), and explain below:

- ___ Clarification on page(s)
- ___ Addition on page(s)
- ___ Deletion on page(s)
- ___ Error on page(s)

Explanation:

CUT ALONG LINE

fold

fold

<p>FIRST CLASS PERMIT NO. 81 POUGHKEEPSIE, N.Y.</p>

<p>BUSINESS REPLY MAIL NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.</p>
--

POSTAGE WILL BE PAID BY
 IBM CORPORATION
 P.O. BOX 390
 POUGHKEEPSIE, N. Y. 12602

ATTN: PROGRAMMING SYSTEMS PUBLICATIONS
 DEPT. D58

fold



International Business Machines Corporation
 Data Processing Division
 112 East Post Road, White Plains, N.Y. 10601
 [USA Only]

IBM World Trade Corporation
 821 United Nations Plaza, New York, New York 10017
 [International]