Enclosure (2)

Computer Program Performance Specification for

STANDARD AN/UYK-7 EXECUTIVE

In Accordance With TADTASK 2-73

FINAL DRAFT

18 JANUARY 1974

UPDATED TADSO Ltr 159
5-16-74

TABLE OF CONTENTS

SECTION 1

SCOPE

SECTION 2

APPLICABLE DOCUMENTS

SECTION 3

REQUIREMENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

Section 1

SCOPE

1.1 INTRODUCTION

This document specifies the functional performance requirements for the Standard Executive program for the AN/UYK-7(V) computer. Within this document the abbreviation SDEX/7 is used to mean the Standard Executive. The term User Module is used throughout this document to mean the application programs operating in the AN/UYK-7 with SDEX/7.

1.2 FUNCTIONAL SUMMARY

The SDEX/7 shall form the basic mechanism of control for user modules operating in the AN/UYK-7(V) computer. The functions provided by SDEX/7 shall be as follows:

a. Initialization

b. Scheduling

c. Interrupt Management

d. Input/Output Management

e. Error Management

1.2.1 INITIALIZATION

The initialization function shall provide the means to load and set both SDEX/7 and user modules to their initial states during computer start-up or restart. This function shall control all initialization operations and shall not release control for system processing operations until the initialization sequence is complete. The initialization function shall:

a. Initialize all SDEX/7 functions and each AN/UYK-7 central processor in the application configuration.

b. Perform a check to determine the operational memory configuration.

c. Load the designated user modules.

d. Pass Central Processor (CP) control to the user module for initialization of the interface between SDEX/7 and the user modules.

e. Upon completion of initialization release CP control to the scheduling function to begin system processing.

1.2.2 SCHEDULING

The scheduling function shall provide the means for allocation of CP resources among user modules. The scheduling function shall provide for the scheduling of four different types of user module tasks as follows:

a. Successor processing tasks - tasks requiring CP control in response to a user module request.

b. Message processing tasks - tasks requiring CP control to receive and process messages initiated by other tasks.

c. Time-dependent processing tasks - tasks requiring CP control on a time-related basis such that the interval of time between successive executions of the task is never less than a specified amount.

d. Background processing tasks - tasks requiring execution on a time-available basis.

The scheduling function shall recognize a priority of task types as defined by the user at SDEX/7 compile-time. However, in the absence of user selection of task type priorities at SDEX/7 compile-time, the following priority shall be recognized:

a. Successor processing tasks shall receive CP control before any pending message, time-dependent or background tasks.

b. Message processing tasks shall receive CP control only after all pending successor tasks have been honored and before any pending time-dependent or background tasks.

c. Time-dependent processing tasks shall receive CP control only after all pending successor and message tasks have been honored and before any pending background tasks.

d. Background processing tasks shall receive CP control only after all pending successor, message and time-dependent tasks have been honored.

1.2.3 INTERRUPT MANAGEMENT

The interrupt management function shall receive and decode all interrupts within the AN/UYK-7 computer. If the interrupt is associated with executive processing, SDEX/7 shall perform the required processing. For all other interrupts, SDEX/7 shall release CP control to the user module which has registered responsibility for processing the interrupt.

1.2.4 INPUT/OUTPUT MANAGEMENT

The input/output management function shall provide the means whereby user modules can initiate and control computer input/output operations. The input/output management function shall allow the user modules to perform the following operations via SDEX/7:

a. Register responsibility for input and/or output monitor interrupts on an I/O channel basis.

b. Define I/O monitor interrupt processing actions by I/O function on an I/O channel basis.

c. Selectively enable and disable interrupts on I/O channels.

d. Initiate I/O chains in response to user module requests.

1.2.5 ERROR MANAGEMENT

The error management function shall identify hardware and software errors upon occurrence and take action as directed by the user modules. The error management function shall:

a. Allow user modules to selectively register responsibility for processing any or all errors.

b. Identify hardware and software errors and pass the error to the registered user module, or conditionally stop the CP when no user module is registered as responsible for processing the error.

c. Resume processing as directed by the user module or the computer operator.

Section 2

APPLICABLE DOCUMENTS

2.1 PROGRAM DEFINITION DOCUMENTS

None.

2.2 INTER-SYSTEM SPECIFICATIONS

None.

2.3 MILITARY SPECIFICATIONS AND STANDARDS

The following documents are applicable to the extent specified herein:

a.	NAVSHIPS 0967-051-6291,	Specification for Digital Data Computer AN/UYK-7(V) with Addendum.
b.	WS-8506, Revision 1,	Requirements for Digital Computer Program Documentation.
c.	NAVSHIPS 0967-028-0060,	User's Reference Manual for Compiler-Monitor System (CMS-2) for use with AN/UYK-7 Computer.

2.4 MISCELLANEOUS DOCUMENTS

a.	NAVSO -3097	ADP Glossary
b.	PX 6503	AN/UYK-7 NDRO Programs Operating Procedures.

Section 3

REQUIREMENTS

3.1 INTRODUCTION

The set of executive functions provided by SDEX/7 shall be contained within the SDEX/7 program such that it is functionally independent of the user modules. The interface defined between SDEX/7 and all user modules shall be general purpose such that support modules for specific applications can be added without impacting the interface between the SDEX/7 program and other user modules.

3.1.1 GENERAL DESCRIPTION

The SDEX/7 executive functions shall be provided for all AN/UYK-7 computer configurations. This requires that a single copy of the SDEX/7 program be capable of providing the functions in a multiprocessing environment. The maximum configuration within which SDEX/7 will operate shall contain no more than 4 central processors (CPs) and shall have at least one Input/Output Controller (IOC) connected to all CPs. In AN/UYK-7 computer configurations where there is not total memory sharing among processors, it shall be the system designer's responsibility to ensure by design that the processors are properly allocated to the application programs according to SDEX/7 performance criteria as specified herein.

The SDEX/7 program shall operate in the interrupt state of the AN/UYK-7 computer and user modules shall be restricted to operation in the task state. The SDEX/7 program shall utilize the overlap features of the AN/UYK-7 computer to the extent allowed by individual application computer configurations.

The SDEX/7 program shall interface with user modules consisting of one or more separate processing tasks to which SDEX/7 can allocate CP and IOC processing resources according to individual user module requirements.

Each module shall contain at least one instruction segment and may contain one or more data segments. The SDEX/7 program shall dedicate the use of two task base register/storage protection register (SPR) pairs to represent the base, displacement, and memory protection attributes for these segments. Additionally, task base register S7 and storage protection register SPR7 shall be dedicated as a pointer to each user module's Segment Allocation Packet (SAP) for the user module's addressing segments (see paragraph 3.2.4.2). The remaining five task base registers shall be transient and available for user module utilization. The dedicated base and storage protection registers shall be SO/SPRO, S1/SPR1 and S7/SPR7 and the transient base and storage protection registers shall be S2/SPR2 thru S6/SPR6.

A standard interface between SDEX/7 and all user modules shall be provided. All user modules shall be required to utilize this interface for operation with SDEX/7. This interface shall represent the only method for communicating data, user module processing requirements and CP control between modules and

SDEX/7. Through this interface, SDEX/7 shall provide the capability for user modules to register task scheduling requirements and responsibility for processing interrupt and error conditions, as well as, direct the SDEX/7 responses upon occurrence of these conditions.

3.1.2 PERIPHERAL EQUIPMENT IDENTIFICATION

The SDEX/7 program shall interface with a loading device determined by the user's equipment configuration.

3.1.3 INTERFACE IDENTIFICATION

The SDEX/7 program shall interface only with user modules making up the specific application and not directly with other computer systems or other computer programs.

3.2 FUNCTIONAL DESCRIPTION

3.2.1 EQUIPMENT DESCRIPTION

The SDEX/7 program shall operate within the AN/UYK-7(V) computer as specified in paragraph 2.3, item a. Additionally, SDEX/7 shall interface with a peripheral device for initial program loading. This device shall be defined according to user configurations. During system start up, the initialization function shall communicate with this device for the loading of designated user modules. Communication with the device shall be according to the technical specification for the load device selected by the user.

3.2.2 COMPUTER I/O UTILIZATION TABLE

The computer I/O utilization shall be defined for the load device (see paragraph 3.2.1) according to user equipment configurations.

3.2.3 COMPUTER INTERFACE BLOCK DIAGRAM

This paragraph is not applicable for the SDEX/7.

3.2.4 PROGRAM INTERFACES

The SDEX/7 program shall provide a standard interface between it and all user modules. Through this interface, SDEX/7 shall pass CP control to modules for scheduled task processing as well as for processing interrupt and error conditions. Through this interface all user modules will register processing requirements, request SDEX/7 executive functions, and return CP control to SDEX/7 when user module processing is complete. These operations will be accomplished using a set of Executive Service Requests which shall be provided by SDEX/7.

In order to operate with SDEX/7 each user module shall contain a preamble table and a Segment Allocation Packet (SAP). The parameters to be supplied by the user module in each of these tables shall be as specified in the following paragraphs.

3.2.4.1 MODULE PREAMBLE TABLE

The module preamble shall be a fixed length table located at relative address zero of a user module's instruction segment. See Figure 3-1. The fields of this table shall be preserved by the user module at all times. The preamble table shall include the following parameters relating to the user module. Additions or changes to these parameters shall be approved by the procurring agency.

a. Module Name - A three character ASCII name for user module identification. The first character shall be a letter. User module names shall be unique.

RESERVED	M H I / MODULE NUMBER	
PRIORITY NUMBER	MODULE NAME	
MBI	SY ADDRESS OF SUCCESSOR TASK	ENTRY POINT LIST
MBI	SY ADDRESS OF MESSAGE TASK	ENTRY POINT LIST
MBI	SY ADDRESS OF TIME-DEPENDENT TASK	ENTRY POINT LIST
MBI	SY ADDRESS OF BACKGROUND TASK	ENTRY POINT LIST
MBI	SY ADDRESS OF TIME-CRITICAL TASK	ENTRY POINT LIST
MBI	SY ADDRESS OF I/O INTERRUPT TASK	ENTRY POINT LIST
NUMBER OF ENTRIES		SUCCESSOR HISTORY
MAXIMUM TIME		SUCCESSOR HISTORY
TOTAL TIME		SUCCESSOR HISTORY
NUMBER OF ENTRIES		MESSAGE HISTORY
MAXIMUM TIME		MESSAGE HISTORY
TOTAL TIME		MESSAGE HISTORY
NUMBER OF ENTRIES		TIME-DEPENDENT HISTORY
MAXIMUM TIME		TIME-DEPENDENT HISTORY
TOTAL TIME		TIME-DEPENDENT HISTORY
NUMBER OF ENTRIES		BACKGROUND HISTORY
MAXIMUM TIME		BACKGROUND HISTORY
TOTAL TIME		BACKGROUND HISTORY
NUMBER OF ENTRIES		TIME-CRITICAL HISTORY
MAXIMUM TIME		TIME-CRITICAL HISTORY
TOTAL TIME		TIME-CRITICAL HISTORY
NUMBER OF ENTRIES		I/O INTERRUPT HISTORY
MAXIMUM TIME		I/O INTERRUPT HISTORY
TOTAL TIME		I/O INTERRUPT HISTORY

FIGURE 3-1. MODULE PREAMBLE

b. Module number - A unique number by which SDEX/7 shall assign a module index for internal referencing of executive tables and lists. Module numbers shall range from 1 to 255.

c. Priority Number - A priority number to be associated with the user module's successor task. Priority number shall be unique among core-resident active user modules and shall range from 1 (highest priority) to 63 (lowest priority).

d. Entry Point List - A list of sy addresses of the entry points of the module's tasks i.e., successor, message, time-dependent, background, time-critical and I/O interrupt tasks. Each module shall be required to provide a message task and entry point, all others shall be optional.

e. Module Base Indicators (MBI). Bit indicators for each module task indicating the task base registers and associated SPRs that are to be loaded prior to releasing CP control to the task. The valid values for MBI shall be as follows:

   (1) 0 - not used

   (2) 1 - load SO/SPRO

   (3) 2 - load SO/SPRO and S1/SPR1

   (4) 3 - load SO/SPRO thru S2/SPR2

   (5) 4 - load SO/SPRO thru S3/SPR3

   (6) 5 - load SO/SPRO thru S4/SPR4

(7) 6 - load SO/SPO thru S5/SPR5

(8) 7 - load SO/SPRO thru S6/SPR6 (this MBI value shall be invalid for message tasks and if used shall cause S/SPR loading as specified for MBI=6).

Combinations other than the above shall not be allowed by SDEX/7.

f. Module History Indicator (MHI) - A bit indicator which when set indicates that SDEX/7 shall maintain module history for each user module task. When the indicator is not set, no history shall be kept for the user module.

g. Module History Store - A storage area for module history for each user module task, i.e., successor, message, time-dependent, background, time-critical and I/O interrupt task. For each task, storage shall be provided for: number of entries, total run time and maximum run time. If a module does not utilize the history feature, this area may be omitted from the preamble.

3.2.4.2 SEGMENT ALLOCATION PACKET

The Segment Allocation Packet (SAP) shall be a variable length table containing the base address and storage protection attributes for all addressable segments within the user module. The table shall consist of a series of two word items (See Figure 3-2) for each addressable segment. Additions or changes to the SAP dedicated items shall be approved by the procurring agency. The SAP items shall be as follows:

a. Item 0 - Dedicated for S7 and shall contain the beginning address of the user module's SAP, and allows only read access to the SAP by the user module.

b. Item 1 - Dedicated for S0/SPR0 and shall contain the beginning address of the user module's data segment.

c. Item 2 - Dedicated for S1/SPR1 and shall contain the beginning address of the user module's instruction segment.

ITEM				
0	MN	S	V	BA (S7)
	ACCESS			R (SPR 7)
1	MN	S	V	BA (S0)
	ACCESS			R (SPRO)
2	MN	S	V	BA (S1)
	ACCESS			R (SPR 1)
3	MN	S	V	BA (S2)*
	ACCESS			R (SPR 2)*
4	MN	S	V	BA (S3)*
	ACCESS			R (SPR3)*
5	MN	S	V	BA (S4)*
	ACCESS			R (SPR4)*
6	MN	S	V	BA (S5)*
	ACCESS			R (SPR5)*
7	MN	S	V	BA (S6)*
	ACCESS			R (SPR6)*
8	MN	S	V	BA (Si) TRANSIENT
	ACCESS			R (SPRi) TRANSIENT
N	MN	S	V	BA (Si + n) TRANSIENT
	ACCESS			R (SPRi + n) TRANSIENT

*Optional use according to user module setting of the module base indicator in its preamble. When the indicator bits are not set, these areas are transient.

MN	–	Module Number of module containing the addressing segment
S	–	Segment identifier
V	–	Segment Version number
BA	–	Base Address of the addressing segment
ACCESS	–	Defines the Access allowed within the segment (See reference 2.3 item a for SPR Bit allocations)
R	–	Maximum allowed displacement for the addressing segment

FIGURE 3-2. SEGMENT ALLOCATION PACKET

d. Items 3 thru 7 - Transient areas which shall contain the beginning addresses of the user module segments to be loaded into S2/SPR2 thru S6/SPR6 respectively according to the Module Base Indicator value in the user module's preamble for the task to receive CP control. (See paragraph 3.2.4.1, item e).

e. All other Items - Transient areas available for use by the user module.

Whenever a user module is to receive CP control, the task base registers S7, S0, and S1 and the associated SPRs shall be loaded. Additionally, the task base registers S2 thru S6 and their associated SPRs shall also be loaded according to the module base indicator setting for the module task. Task base registers S2 thru S6 and the associated SPRs shall be treated as transient by SDEX/7 and shall be loadable from any SAP item (except item 0) by an executive service request issued by the user module.

3.2.5 FUNCTION DESCRIPTION

The SDEX/7 program shall provide the following functions:

a. Initialization

b. Scheduling

c. Interrupt Management

d. Input/Output Management

e. Error Management

3.2.5.1 INITIALIZATION

The initialization function shall be performed to accomplish computer program start or restart. During computer program start-up, the initialization function shall control all operations which shall include initializing all CPs, loading the initial configuration of user modules defined by the user, passing CP control to each user module for local initialization and registration of processing requirements/ responsibilities, and when the initialization process is complete, CP control shall be passed to the scheduling function to begin system operation. Figure 3-3 illustrates the sequence of the initialization function.

3.2.5.2 SCHEDULING

The scheduling function shall provide the means by which user module tasks are requested and subsequently given CP control for processing. The scheduling function shall distribute the CP processing resources to user modules according to user module task processing requirements. Upon completing any scheduled user module task, CP control shall always be returned to the scheduling function to begin the query for the next task for processing. The order in which the scheduling function queries the individual task scheduling lists shall be selectable at SDEX/7 compile-time. Figure 3-4 illustrates the sequence of the scheduling function.

3.2.5.3 INTERRUPT MANAGEMENT

The interrupt management function shall initially receive and decode all AN/UYK-7 interrupts and perform the required action for executive related interrupts. For all other interrupts, CP control shall be passed to the user module which has registered responsibility for processing the interrupt. Figure 3-5 illustrates the sequence of the interrupt management function.

3.2.5.4 INPUT/OUTPUT MANAGEMENT

The input/output management function shall provide the means by which user modules can initiate and control computer input/output operations. Figure 3-6 illustrates the sequence of the input/output management function.

3.2.5.5 ERROR MANAGEMENT

The error management function shall provide the means by which user modules may selectively register responsibility for any or all error conditions. Upon the occurrence of an error, the error management function shall pass CP control with pertinent error information to the user module which registered responsibility for processing the error. If an error occurs for which no user module has registered responsibility, the error management function shall conditionally stop the CP. Figure 3-7 illustrates the sequence of the error management function.

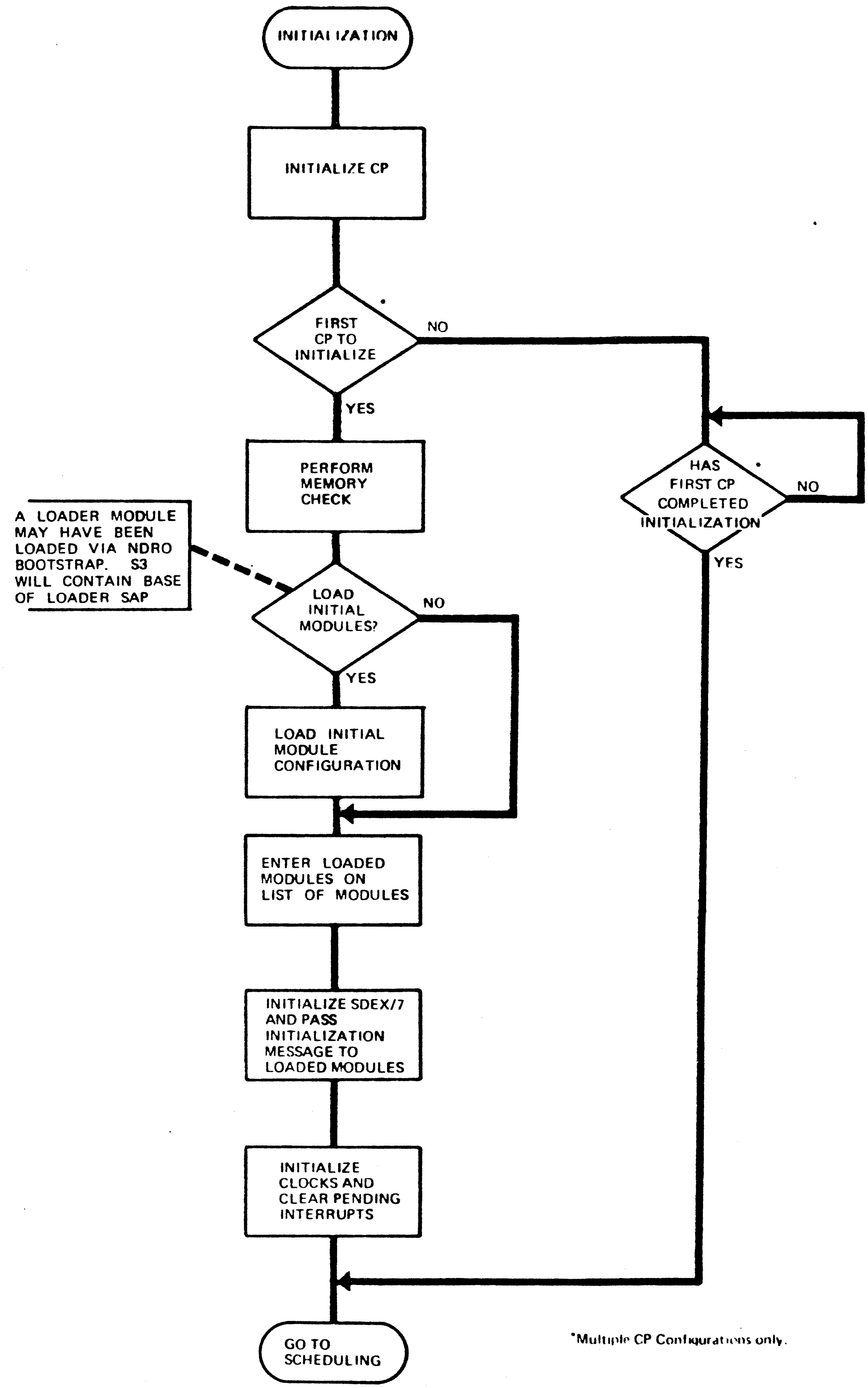
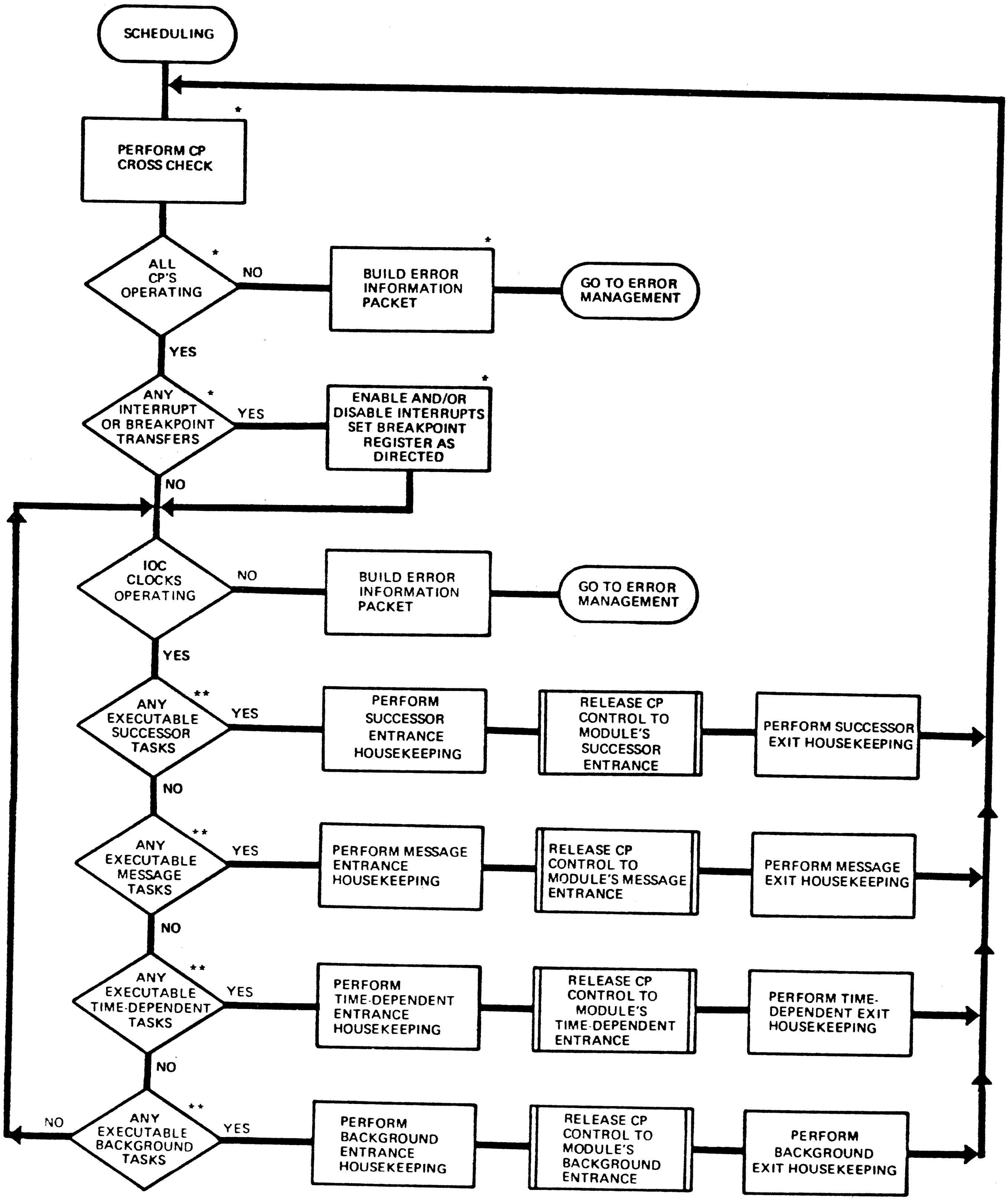


FIGURE 3-3. INITIALIZATION FUNCTIONAL PROCESSING



*MULTIPLE CP CONFIGURATIONS ONLY
**ORDER OF QUERY IS COMPILE TIME SELECTABLE

FIGURE 3-4. SCHEDULING FUNCTIONAL PROCESSING

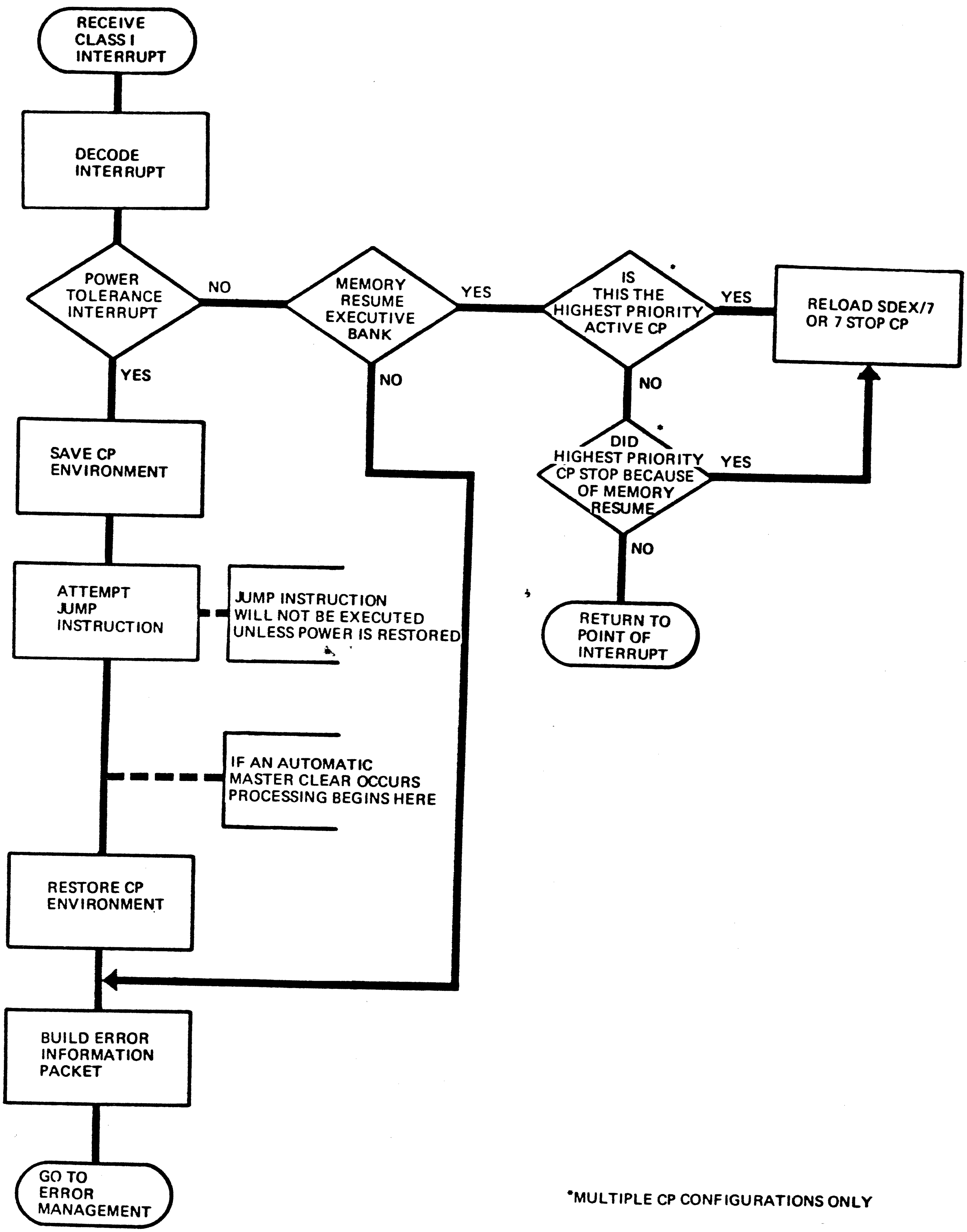


FIGURE 3-5. INTERRUPT MANAGEMENT FUNCTIONAL PROCESSING
(Page 1 of 3)

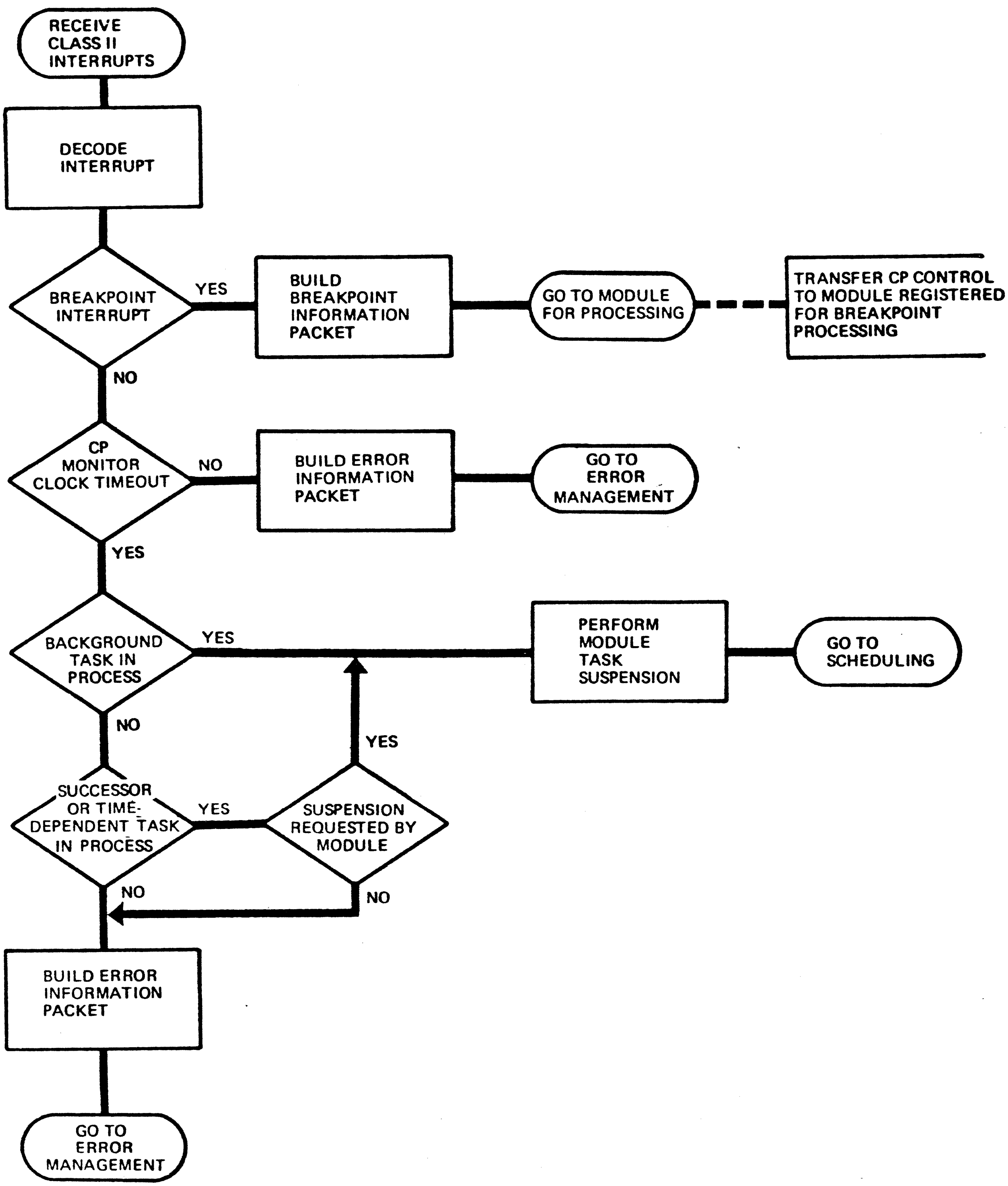


FIGURE 3-5. INTERRUPT MANAGEMENT FUNCTIONAL PROCESSING
(Page 2 of 3)

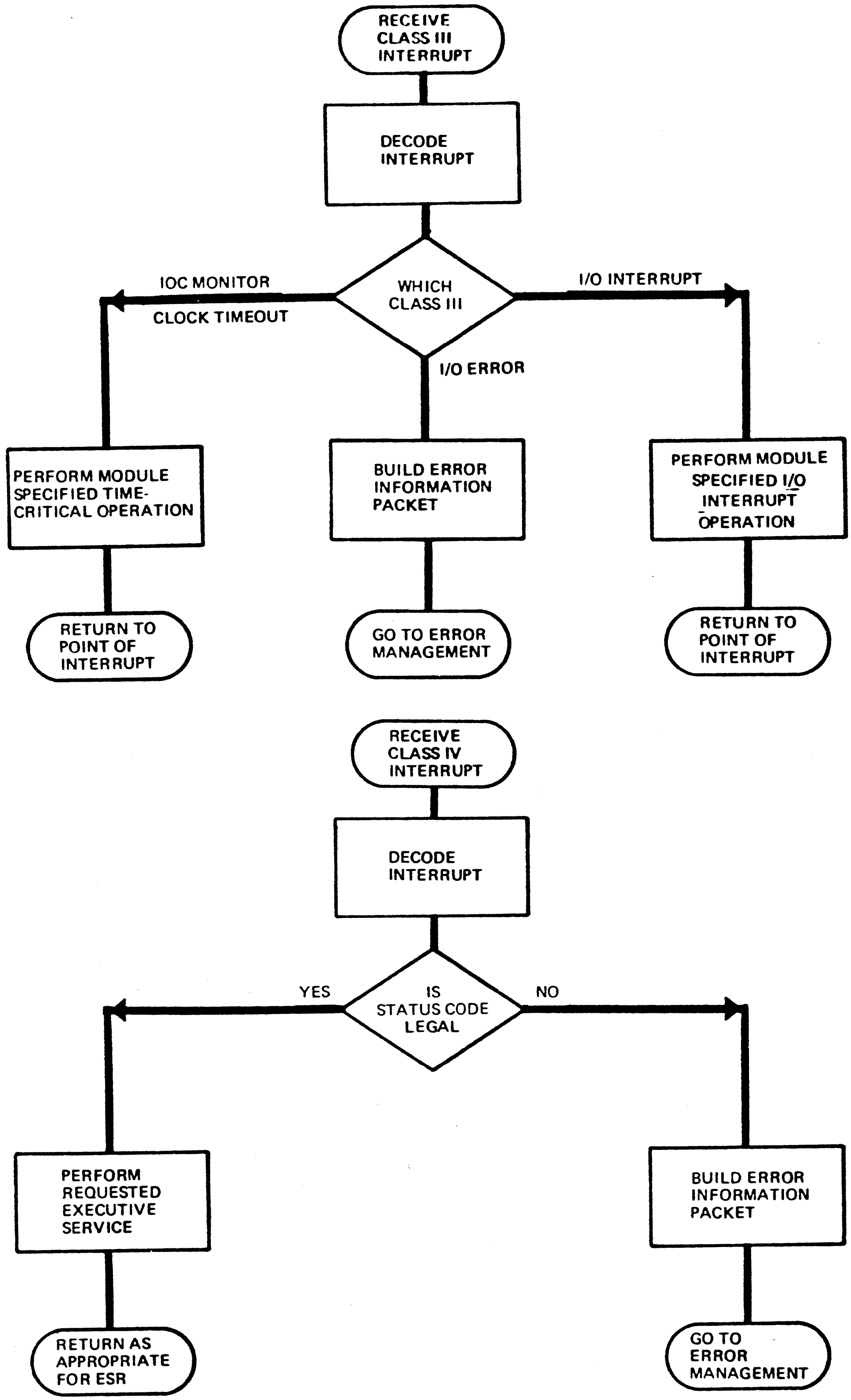


FIGURE 3-5. INTERRUPT MANAGEMENT FUNCTIONAL PROCESSING
(Page 3 of 3)

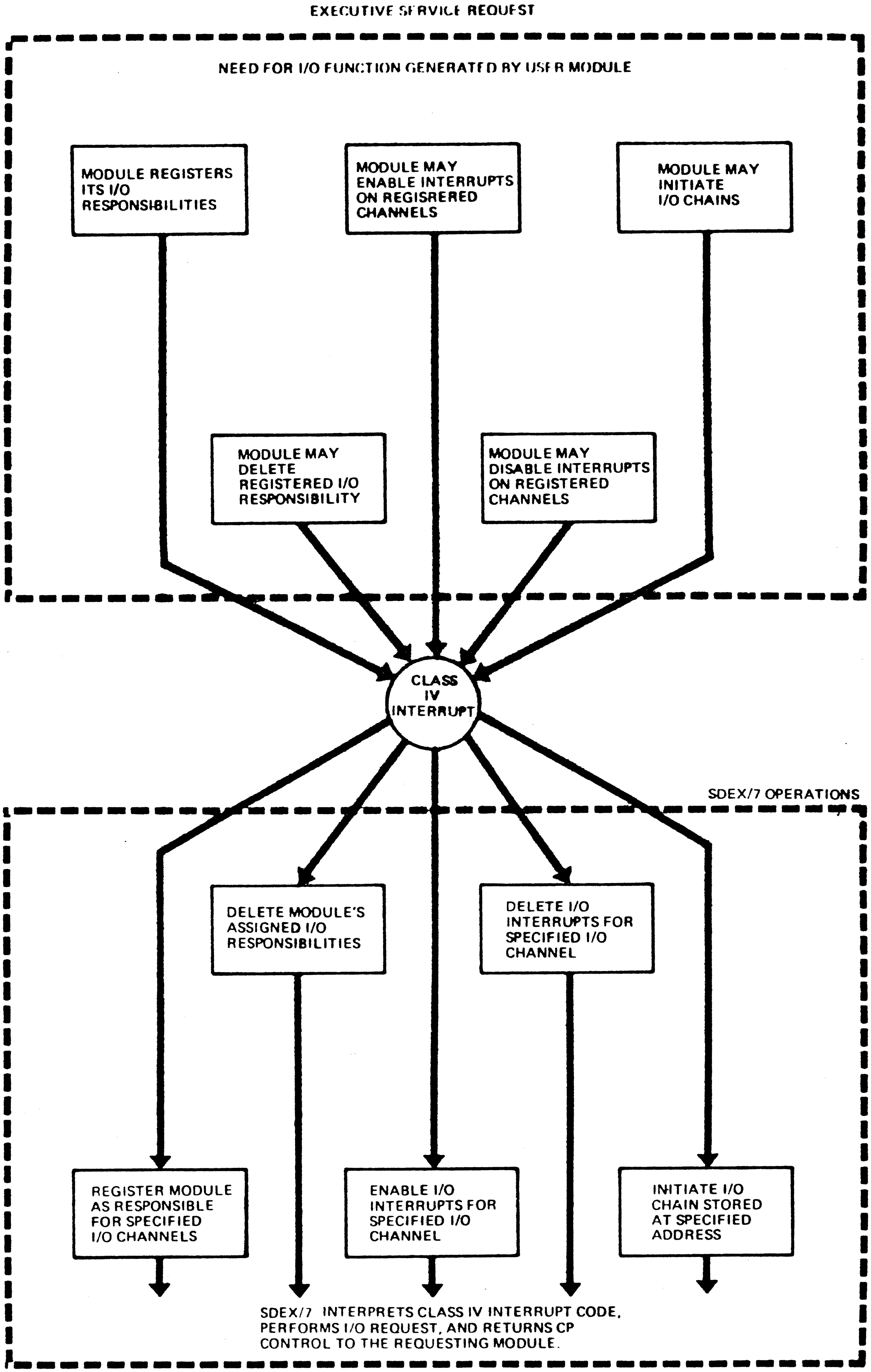


FIGURE 3-6. I/O MANAGEMENT FUNCTIONAL PROCESSING

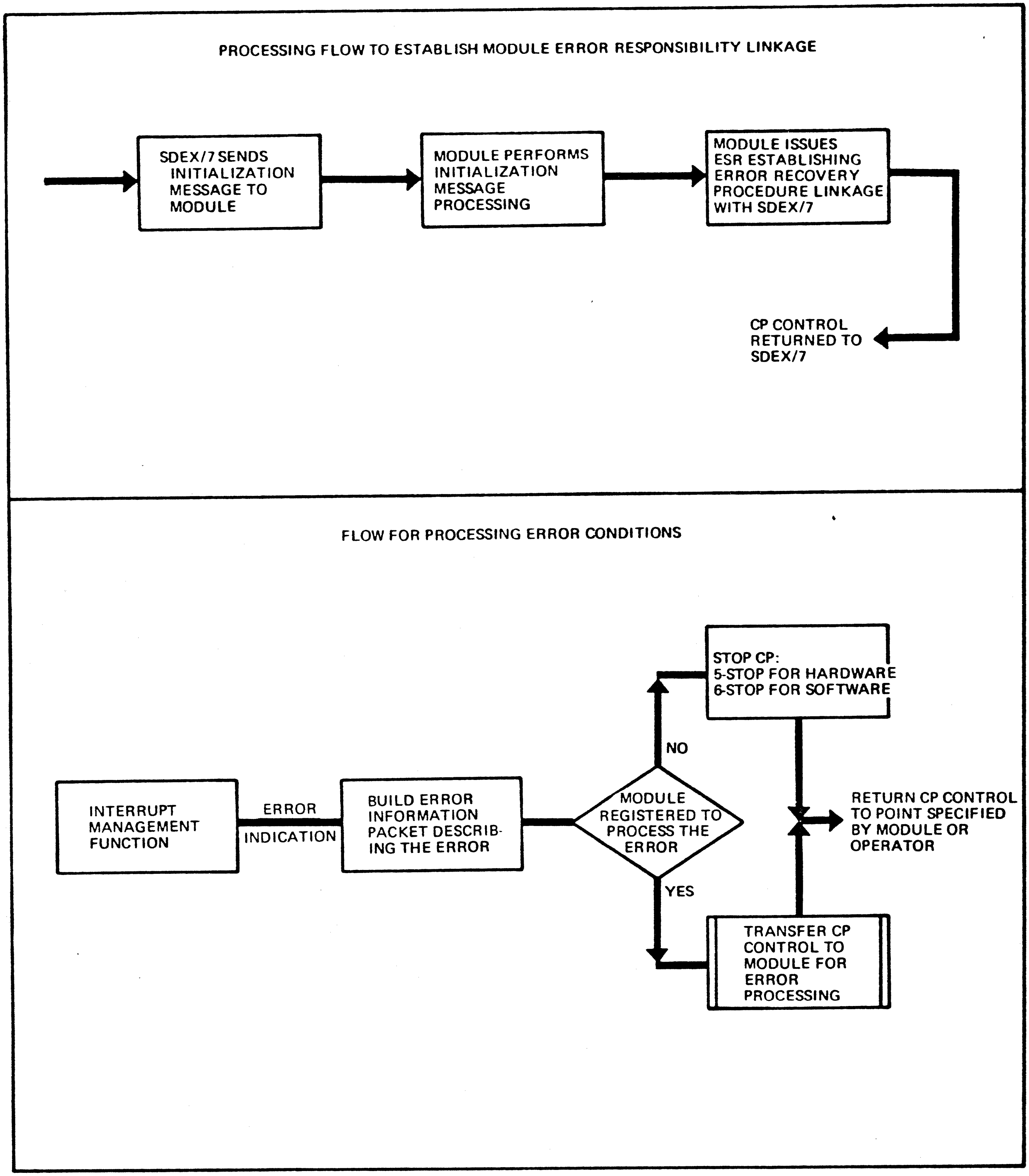


FIGURE 3-7. ERROR MANAGEMENT FUNCTIONAL PROCESSING

3.3 DETAILED FUNCTIONAL REQUIREMENTS

3.3.1 INITIALIZATION INTRODUCTION

The initialization function shall receive CP control after SDEX/7 has been loaded via NDRO bootstrap. The initialization function shall control all processing until initialization is complete, and then release CP control to the scheduling function.

3.3.1.1 INITIALIZATION INPUTS

The inputs to the initialization function shall be as follows:

a. interrupt base register S3 containing either the base address of the SDEX/7 instruction segment or the absolute address of the SAP table of a user supplied loader module.

b. NDRO working storage area containing the IOC and channel number of the bootstrap load device.

3.3.1.2 INITIALIZATION PROCESSING

The SDEX/7 program shall be placed in memory by the AN/UYK-7 NDRO bootstrap or a user-supplied Loader Module which shall release CP control to the SDEX/7 upon completion of the load. As each CP enters the initialization function, S0, S1 and S2 shall be loaded with the base address of the SDEX/7 instruction, shared data and CP dedicated data segment respectively, any pending interrupts shall be cleared and the CP and IOC monitor clocks shall be disabled to prevent any preemption of the initialization function. To facilitate

CP-dedicated data access and CP identification, interrupt index register B1 shall be set to the CP index. The CP index shall be derived from a compile-time parameter equating the CP identification number (bits 20-22 from the ASR) to the user-defined index for the CP. This index (B1) shall not be altered at any time. The SDEX/7 shall then initialize itself, as follows:

a. For the first CP to initialize, Initial Condition Words for Class I, II, III, and IV interrupts shall be loaded such that interrupt processing can be performed within the initialization function. For CPs other than the first, all interrupts shall be locked out and the Initial Condition Words shall be loaded for the interrupt management routines. For all CPs the Class I interrupt routines shall be placed at the proper location specified by the NDRO program selected for the application. The location of the following routines shall be as specified in reference 2.4, item a for the selected NDRO program:

(1) Auto start routine - for auto start after power is resupplied following a power failure.

(2) Normal recovery routine - for memory resume faults when the alternate recovery routine is not available.

(3) Alternate recovery routine - for memory resume faults.

b. The bootstrap mode bit (ASR bit 7) shall be cleared.

c. Interrupt state base registers shall be set to the data and instruction segments of SDEX/7.

d. Executive data dedicated only to the executing CP shall be initialized.

At this point, all CPs other than the first to initialize have completed initialization processing and shall proceed to the scheduling function if the first CP has completed the entire initialization function. Otherwise, the CPs shall wait for a signal from the first CP, indicating completion of its initialization processing, before proceeding to the scheduling function.

The first CP shall release interrupt lockout and check memory to determine the amount and condition of memory available.

As a minimum, the memory check shall be a single pass check testing two consecutive words of each memory bank to determine if the memory bank is available in the configuration. If errors in memory operation are detected, the CP shall be stopped with the address of the erring memory location displayed on the maintenance panel along with the expected and received values of the check. Depression of the START switch by the operator shall cause the memory check to continue with the next sequential bank.

If a memory error is detected, the memory bank shall be declared inoperable and not available for use. Upon completion of the memory check, the contents of interrupt S0 shall be compared with the contents of S3. If the contents are not equal (SDEX/7 has been loaded by a user-supplied loader module) S3 shall be the base address of that module's SAP, which shall be placed in the SDEX/7 list of modules. The loading of designated modules as specified below shall not be performed and the initialization processing shall continue at the point where the loading process is complete.

If the contents of S0 and S3 are equal indicating that SDEX/7 was loaded by NDRO Bootstrap, the user modules identified by the user through the user of SDEX/7 compile-time parameters shall be loaded. The loader contained within the initialization function shall first search the NDRO working storage area to identify the IOC and channel number to which the load device is connected. The IOC and channel number are left in the working storage area by the NDRO Bootstrap routine which is used to initially load SDEX/7. The I/O command chains used by the loader shall then be modified using the IOC and channel numbers to provide correct communication with the load device.

The user shall be provided the capability to specify the absolute allocation of each user module to be loaded through the use of SDEX/7 compile-time parameters. The user modules shall then be loaded according to this allocation unless a failed memory bank was detected which makes the specified allocation impossible. In this case, loading shall be accomplished using the next available memory bank. If the user does not specify the absolute allocation of the user modules, the loading process shall be accomplished such that maximum utilization of the AN/UYK-7 overlap feature is achieved within available memory. During the loading of each user module, the Segment Allocation Packet (SAP) shall be completed with the final allocation of each addressable segment defined by the user module.

Upon completion of the loading process, interrupts shall be locked out and the absolute address of each loaded user module's SAP shall be placed in the SDEX/7 list of modules (task list). The executive data areas shared by all CPs shall then be initialized including the SDEX/7 message packing areas. The packing areas shall be defined such that they overlay the portion of initialization used for module loading (See paragraph 3.3.1.4.3).

Initialization messages shall then be formatted and passed in turn to each loaded module for local user module initialization, including registration of scheduling, interrupt and error responsibilities. After all modules have received and processed the initialization message, i.e., completed local initialization, the IOC monitor clock shall be set for the earliest time-critical interrupt (if any) requested by a user module. If no interrupt has been requested, the IOC monitor clock shall be disabled. The IOC real time clock (RTC) shall be cleared and interrupt lockouts shall be released. Any waiting CPs shall then be signaled that initialization is complete and CP control shall be passed to the scheduling function to begin system processing.

3.3.1.3 INITIALIZATION OUTPUTS

The initializtaion outputs shall be initialization messages to each loaded user module and executive data tables initialized to reflect user module requirements and responsibility registrations.

3.3.1.4 INITIALIZATION SPECIAL REQUIREMENTS

3.3.1.4.1 MULTIPLE IOC CONSIDERATIONS

Only one IOC real time clock and monitor clock shall be used in any system. In multiple IOC configurations, those clocks that are not used shall be disabled. Further, the IOC containing the clocks being used shall be connected to all CPs.

3.3.1.4.2 INITIALIZATION MESSAGE

During initialization, an initialization message shall be passed to each loaded user module via its message entrance and the user module shall process the message according to user specification. The initialization message shall be formatted as specified in paragraph 3.3.2.4.7 and shall consist of a control word indicating message type 01 and a single text word indicating the type of user module initialization as follows:

01 - Initialize indicating a system start-up with no valid local data.

The initialization message shall not be restricted to use by SDEX/7 and may be initiated by any user module. Additional message text parameters may be defined for each separate application depending upon application requirements without impact upon the SDEX/7 initialization function.

3.3.1.4.3 SYSTEM MESSAGE DATA INITIALIZATION

A fixed amount of memory shall be available for use by user modules as message packing areas. The total amount of memory reserved for this purpose shall be a SDEX/7 compile-time parameter defined by the user for his application. The loader portion of the initialization function shall be a part of this area and thus shall be overlayed after user module loading.

The system message packing area shall be divided into segments consisting of a number of fixed length packets. The number of segments, ratio of segment size and packet lengths for each segment shall be SDEX/7 compile-time parameters such that the user can define each for his application. During initialization of executive data areas, the necessary controls for managing the system message packing area shall be according to these user-specified SDEX/7 compile-time parameters. The handling of messages using the system message packing areas is specified in paragraph 3.3.2.4.7.

3.3.2 SCHEDULING INTRODUCTION

The scheduling function shall allocate CP processing time to modules according to user module task scheduling requirements and user module task type priorities. Upon entry into the scheduling function, a check shall be performed to determine if the master IOC monitor and real time clocks are operating. If the IOC real time clock fails to increment in fifty successive passes through the scheduling function, a clock failure error shall be indicated and CP control shall be passed to the error management function. For multiple CP configurations, a CP cross-check shall first be performed to determine if all CPs are operating. In addition, any enable or disable I/O channel interrupt or set breakpoint register operations shall be performed in the executing CP as specified by any other CP which has honored user module requests (ESRs) for these operations.

3.3.2.1 SCHEDULING INPUTS

The inputs for the scheduling function shall consist of ESRs regarding the scheduling of user module tasks, RTC values indicating the time at which each CP last entered the scheduling function, and a flag indicating whether or not any enable or disable channel interrupt or breakpoint operations are required.

3.3.2.2 SCHEDULING PROCESSING

In multiple CP configurations, the executing CP shall first store the current RTC value indicating its time of entry to the scheduling function. The entry times saved by the other CPs in the configuration shall then be compared to the current RTC value. If the time lapsed between the current time and the last time that each other CP entered the scheduling function exceeds a system-defined maximum (SDEX/7 Compile-time parameter), that CP shall be considered inoperable and the error indication shall be sent to the error management function. Otherwise, the executing CP shall perform any indicated interrupt enable and disable or breakpoint register operations that may have been requested by a user module being executed by another CP. Upon completion of these operations (if any), the executing CP shall proceed with user module task scheduling.

The scheduling of user module tasks shall be based upon a priority of task types such that all requested tasks of a specified type shall be allotted CP processing time only after all tasks of a higher priority type are completed and before tasks of a lower priority type are begun. Where there are no tasks located for execution in the lowest priority task type, the scheduling function shall resume its search at the highest priority task type. The priority of task types shall be SDEX/7 compile-time parameters such that the user can define, for his application, the order in which the lists of the requested task types are searched. The scheduling function shall provide scheduling for the following user module task types:

a. Successor tasks

b. Message tasks

c. Time-dependent tasks

d. Background tasks

3.3.2.2.1 SUCCESSOR SCHEDULING

After all pending user module tasks of higher priority types have been honored, the successor indicator list shall be checked. This successor indicator list shall contain an indicator corresponding to the successor entrance for each user module that registers such an entrance, and shall be maintained by SDEX/7. The successor indicators shall be arranged on the list according to the user module priority numbers. The successor indicator list shall be searched starting at the highest priority indicator position looking for a set indicator signifying that the associated user module is to receive CP control at its successor entrance. When a set successor indicator is found, a check shall be made to determine if the task is executable by the executing CP. (See paragraph 3.3.2.4.1 and 3.3.2.4.2 for rules regarding CP dedication and module reentrancy). If not, the search of the successor indicator list shall be resumed, continuing until a successor task is located which can be executed by the executing CP or until the entire list has been searched. If no executable successor task is found, the scheduling function shall proceed to query the list of tasks that is next in priority.

When an executable successor task is found, the successor indicator for the task shall be cleared. The time-slice indicator for the task shall be checked. If set, the task environment saved in the Module's Successor Save area shall be restored. Otherwise, the dedicated base registers for the module shall be loaded from the user module's SAP, and the CP monitor clock shall be loaded with the appropriate value to prevent user module overrun (See paragraphs 3.3.2.4.3 and 3.3.2.4.4). CP control shall then be released to the user module's successor task.

Upon completion of successor task processing, the user module shall execute the Module Exit ESR. Upon acknowledgement of this ESR, CP control shall be returned to the beginning of the scheduling function. When all successor indicators are clear, the scheduling function shall proceed to query the list of tasks that is next in priority.

3.3.2.2.2 MESSAGE SCHEDULING

After all pending user module tasks of higher priority types have been honored, the list of messages awaiting processing shall be checked. The check shall be performed such that a First-in/First-out (FIFO) flow of messages to receiving user modules is maintained. When a message is located for processing, the message hold flag for the receiving user module shall be checked. A user module's message hold flag is set whenever the user module is not executable as specified in paragraphs 3.3.2.4.1 and 3.3.2.4.2. This insures the FIFO order of message processing by each user module). If the message hold flag for the receiving user module is set, the search shall continue by checking the next message in the message task scheduling list. Otherwise, a check shall be made to determine if the receiving user module can be executed by the executing CP (see paragraphs 3.3.2.4.1 and 3.3.2.4.2). If not, the message hold flag shall be set for the receiving user module and the search of the message task scheduling list shall continue.

If the receiving user module can be executed, all message hold flags shall be cleared and task base register S6 shall be loaded with the base address of the message and SPR6 shall be loaded with the appropriate displacement value for the message allowing read access to the message. The dedicated base registers for the receiving user module shall be loaded from the user module's SAP, and the CP monitor clock shall be loaded with the appropriate value to prevent user module overrun (See paragraph 3.3.2.4.3). CP control shall then be released to the receiving user module's message task. After the user module has completed processing associated with the message, the user module shall execute the Module Exit ESR. Upon acknowledgement of this ESR, CP control shall be returned to the beginning of the scheduling function. After all messages have either been processed or are being held due to receiving user module busy or CP assignment conditions, all message hold flags shall be cleared and the scheduling function shall proceed to query the list of tasks that is next in priority.

3.3.2.2.3 TIME-DEPENDENT SCHEDULING

After all pending user module tasks of a higher priority type have been honored, the list of time-dependent tasks shall be searched. There shall be no priority among time-dependent tasks; each task shall be tested for execution strictly on a round-robin basis. Timing parameters associated with time-dependent tasks shall be used by the scheduling function to determine when the tasks are due for execution.

The time-dependent list shall be searched looking for the next task whose time-to-initiate-execution (TE) is less than or equal to the current value of the IOC real time clock. When a time-dependent task is located which is due for execution, a check shall be made to determine if the task is executable by the executing CP. (See paragraph 3.3.2.4.1 and 3.3.2.4.2 for rules regarding CP dedication and module reentrancy). If not, the search of the time-dependent list shall be resumed, continuing until a time-dependent task is located that can be executed by the executing CP or until the entire list has been searched. If no executable time-dependent task is located, the scheduling function shall proceed to query the list of tasks that is next in priority.

When an executable time-dependent task is located, the next TE for the task shall be computed by adding the time interval parameter (supplied by the user module when the time-dependent task was registered) to the current value of the IOC real-time clock. The new TE shall then be placed in the time dependent list. The time-slice indicator for the task shall be checked. If set, the task environment saved in the user module's time-dependent save area shall be restored. Otherwise, the dedicated base resisters for the user module shall be loaded from the user module's SAP, and the CP monitor clock shall be loaded with the appropriate value to prevent module overrun (see paragraphs 3.3.2.4.3 and 3.3.2.4.4). CP control shall then be released to the user module's time-dependent task.

Upon completion of time-dependent processing, the user module shall execute the Module Exit ESR. Upon acknowledgement of this ESR, the time interval parameter shall be checked. If the time interval is zero, the task registration shall be deleted from the time dependent list. Otherwise, CP control shall be returned to the beginning of the scheduling function. When no executable time-dependent task is found, the scheduling function shall proceed to query the list of tasks that is next in priority.

3.3.2.2.4 BACKGROUND SCHEDULING

After all pending user module tasks of a higher priority type have been honored, the list of background tasks shall be searched. Background task scheduling shall provide the capability for user modules to perform processing on a time-available basis. Background tasks shall be time-sliced using the CP monitor clock to periodically return CP control to the scheduling function. If the task does not complete processing within an allotted time, it shall be timed out, and rescheduled for further processing at a later point of time. Once a background task has been started, it shall process on a time-sliced basis until the task is complete. The background task shall then be rescheduled for background task initiation according to the interval between job parameter supplied by the user module.

There shall be no priority among background tasks. Each task shall be tested for execution strictly on a round-robin basis. Timing parameters supplied by the user module shall be used by the scheduling function to determine when the task is due for execution.

The background list shall be searched looking for a task whose TE is less than or equal to the IOC real time clock. When a background task is located which is due for execution, a check shall be made to determine if the task is executable by the executing CP (See paragraph 3.3.2.4.1 and 3.3.2.4.2 for rules regarding CP dedication and module reentrancy). If not, the search of the background list shall be resumed, continuing until a background task is located that can be executed by the executing CP or until the entire list has been searched. If no executable background task is located, the scheduling function shall proceed to query the list of tasks that is next in priority.

When an executable background task is located, the next TE for the task shall be computed by adding the user module-supplied interval between slices to the current value of the IOC real time clock. If the task had been previously time-sliced, the task environment shall be restored from the user module's background save area where it was stored when the task was time-sliced. If not, the dedicated base registers shall be loaded from the user module's SAP, and the CP monitor clock shall be loaded with the time-slice value for the background task. CP control shall then be released to the user module's background task.

When the module completes background processing or when the background task is suspended, CP control shall be returned to the beginning of the scheduling function. When no executable background task is found, the scheduling function shall proceed to query the list of tasks that is next in priority.

3.3.2.3 SCHEDULING OUTPUTS

The outputs of the scheduling function are associated with the particular task being honored. The outputs from the scheduling function for scheduled user module tasks shall be as follows:

a. Dedicated Base Registers - The dedicated base registers shall be set for the user module (see paragraph 3.2.4.2) whenever CP control is released to the user module.

b. Message Pointer - task base register S6 shall be loaded with the base address of the message to be processed and SPR6 shall be loaded to allow read access to the message when CP control is released to a user module's message task.

c. CP Monitor Clock - The CP monitor clock shall be loaded with the appropriate value to prevent user module overrun (see paragraphs 3.3.2.4.3 and 3.3.2.4.4).

3.3.2.4 SCHEDULING SPECIAL REQUIREMENTS

3.3.2.4.1 CENTRAL PROCESSOR DEDICATION

The SDEX/7 program shall provide the capability to dynamically assign CPs to modules by CP numbers such that only the user-selected CPs are allowed to execute the user module tasks associated with successor, message, time-dependent, and background entrances. In a multiple CP configuration, the scheduling function shall always check CP dedication before releasing CP control to a user module's task. If the user module is not executable (due to CP dedication) by the executing CP, the scheduling function shall resume its search for a task to execute according to the task type scheduling criteria.

3.3.2.4.2 MODULE REENTRANCY

The scheduled user module task, i.e., successor, message, time-dependent and background, shall not be reentrant. Whenever the scheduling function locates a user module task for execution, a check shall be made to determine if another CP is currently executing a scheduled task, within that user module. If not, the user module shall be indicated busy and CP control can be released to the user module task. Otherwise, the user module shall be considered busy and the scheduling function shall resume the search for a task to execute according to the task type scheduling criteria. When the user module task completes processing and executes the Module Exit ESR, the module busy indication shall be cleared.

3.3.2.4.3 MODULE OVERRUN PROTECTION

The scheduling function shall provide a special means by which user module timing integrity can be protected against individual user modules overrunning an allotted time. User module overrun times for each task type shall be SDEX/7 compile-time parameters. Whenever a user module is given CP control, the CP monitor clock shall be loaded with a parameter indicating the maximum time to be allowed for processing the user module's task. If the user module fails to complete processing in the allotted time, the CP monitor clock interrupt will occur and SDEX/7 will regain control. The task type that was interrupted shall then be checked. If the task can be suspended (see paragraph 3.3.2.4.4), the task environment shall be saved and CP control shall be returned to the scheduling function. If not, a user module overrun error shall be indicated and CP control shall be passed to the error management function.

3.3.2.4.4 MODULE TASK TIME-SLICING

The scheduling function shall provide the capability for specified user module tasks to be time-sliced. The user module tasks for which time-slicing is provided shall be those associated with the background, time-dependent and successor entrances.

3.3.2.4.4.1 BACKGROUND TASK TIME-SLICING

For a background task, the user module shall use a Register Background Entrance ESR to specify the initial time (value of the real time clock) after which the background task may be executed. In addition, a transient SAP item (see paragraph 3.2.4.2) shall be supplied which identifies the base of the user module's save area for the background task environment in the event that the task is time-sliced. The background save area shall include the following:

a. Interval between time-slices - the minimum time between successive suspensions and resumptions of the time-slice portions of the background task.
b. Interval between jobs - the minimum time between completion of the background task and its next scheduled entry.
c. Time-slice - the CP processing time to be allotted to the task before time-slicing. This value shall be

placed in the CP monitor clock each time the user module is given control for background processing.

d. Subtotal running time - accumulated amount of CP processing time used by the background task. This parameter shall be used only when the user module invokes the module history storage feature (see paragraph 3.3.2.4.6).

e. Register save area - storage area for the eight task state accumulators, seven index registers and eight task base registers and the associated SPRs.

f. Active Status Register - storage area for the Active Status Register (ASR) as set at the time of CP monitor clock interrupt.

g. Task state p-register - storage area for the p-register contents which indicates the address of where processing is to resume when the background task is again given CP control.

3.3.2.4.4.2 TIME-DEPENDENT TASK TIME-SLICING

The scheduling function shall provide a compile-time option for time-dependent user module tasks to be time-sliced on a module-by-module basis. For a time-dependent task, the user module shall use a Register Time-Dependent Entrance ESR to specify the initial time (value of the real time clock) after which the time-dependent task may receive CP control, and the interval between successive executions of the task. In addition, the user module shall provide a parameter indicating if the task may be time-sliced. If so, a transient SAP item (see paragraph 3.2.4.2) shall also be supplied which identifies the base of the user module's save area for the time-dependent task environment in the event the task is time-sliced. The time-dependent save area shall include the following:

a. Time-slice - CP processing time to be allotted to the task before time-slicing. This value shall be placed in the CP monitor clock each time the user module is given CP control for time-dependent processing.

b. Subtotal running time - accumulated amount of processing time used by the time-dependent task. This parameter shall be used only when the user module invokes the module history feature (see paragraph 3.3.2.4.6).

c. Register save area - storage area for the eight task state accumulators, seven index registers and eight base registers and associated SPRs.

d. Active Status Register - storage area for the ASR as set at the time of CP monitor clock interrupt.

e. Task State p-register - storage area for the p-register contents which indicates the address of where processing is to resume when the time-dependent task is again given CP control.

If the time-dependent task is registered such that time-slicing is not allowed, the rules for user module overrun protection as specified in paragraph 3.3.2.4.3 shall apply. If a time-dependent task is time-sliced, the next time-to-initiate execution (TE) for the task shall be reset in the time-dependent scheduling list to the current value of the IOC real time clock. This shall allow the time-sliced task to be activated at the earliest time according to the scheduling criteria specified in paragraph 3.3.2.2.1 thru 3.3.2.2.4.

3.3.2.4.4.3 SUCCESSOR TASK TIME-SLICING

The scheduling function shall provide a compile-time option for user module successor tasks to be time-sliced on a module-by-module basis. For a successor task the user module shall use the Register Successor Entrance ESR to specify whether or not the task may be time-sliced. If so, a transient SAP item (See paragraph 3.2.4.2) shall also be supplied which contains the base of the user module's save area for the successor task environment in the event that the task is time-sliced. The successor save area shall include the following:

a. Time-slice - CP processing time to be allotted to the task before time-slicing. This value shall be placed in the CP monitor clock each time the user module is given CP control for successsor processing.

b. Subtotal running time - accumulated amount of processing time used by the successor task. This parameter shall be used only when the user module invokes the module history feature (see paragraph 3.3.2.4.6).

c. Register save area - storage area for the eight task state accumulators, seven index registers and eight base registers and associated SPRs.

d. Active Status Register - storage area for the ASR as set at the time of CP monitor clock interrupt.

e. Task State p-register - storage area for the p-register contents which indicates the address of where processing is to resume when the successor task is again given CP control.

If the successor task is registered such that time-slicing is not allowed, the rules for user module overrun protection as specified in paragraph 3.3.2.4.3 shall apply. If a successor task is time-sliced, the successor indicator for the task shall be set in the successor indicator list. This shall allow the time-sliced task to be reactivated at the earliest time according to the scheduling criteria specified in paragraph 3.3.2.2.1 thru 3.3.2.2.4.

3.3.2.4.5 TASK SUSPENSION OPTIONS

The scheduling function shall provide the capability for user module tasks to request suspension using the Suspend Task ESR. This capability shall apply for successor and time-dependent tasks that are registered for time-slicing and for any background task. Upon receipt of this ESR, the task environment shall be saved in the module-supplied save area specified when the module registered the task. The task shall be rescheduled according to the parameters supplied with the

ESR (see table 3-1, item 42). CP control shall then be returned to the scheduling function to continue processing. When the task again receives CP control according to the scheduling criteria for the task type, the environment shall be restored and processing shall resume at the point of suspension.

3.3.2.4.6 MODULE SCHEDULED TASK OPTIONS

The scheduling function shall provide the capability for the user to select the types of user module tasks scheduled for his application by SDEX/7 compile-time options. These options shall allow the user to select or omit scheduling for any of the following task types:

a. Successor tasks

b. Time-dependent tasks

c. Background tasks

If the user opts, at SDEX/7 compile-time, to omit any particular task type from scheduling considerations, no SDEX/7 instructions or data pertaining to that task type shall be included in the SDEX/7 program. This constraint shall also apply to all ESRs provided for registration, deletion of registration and requests for scheduling of the task type.

3.3.2.4.7 MODULE HISTORY OPTION

Module history shall be provided by the scheduling function on a module-by-module basis. The preamble for each user module shall contain a parameter which indicates whether or not module history is to be maintained for the user module and a history storage area (see paragraph 3.2.4.1). If the module history is to be maintained, the following information shall be maintained for each user module task:

a. Number of entries

b. Total run-time

c. Maximum run-time

3.3.2.4.8 MESSAGE HANDLING

Any user module receiving CP control may send messages to any other user module. Three types of message handling shall be provided as follows:

a. Immediate message

b. System message

c. Local message

Any module receiving CP control for successor, message, time-dependent or background task processing may initiate an immediate message to the user module designated as the common System module by compile-time parameter. When the Initiate Immediate Message ESR is received by SDEX/7, the environment for the requesting task shall be saved and the CP monitor clock shall be saved and reloaded with the maximum time for a message entrance. The dedicated base registers shall be loaded from the SAP table of the designated Common System module, task base register S6/SPR/6 shall be loaded with the address, displacement, and memory protection attributes for the immediate message and CP control shall be passed to the message entrance of the Common System module. Upon completion of the processing of the immediate message, the saved task environment and the CP monitor clock shall be restored and CP control shall be returned to the module initialing the immediate message.

The immediate message shall be handled when received as specified above. The handling of System and local message shall be as specified in the following paragraphs.

3.3.2.4.8.1 SYSTEM MESSAGE HANDLING

During program loading and initialization an area of memory shall be set aside as a system message packing area. Through SDEX/7 compile-time parameters, the user may determine the length of the area, the number and size ratio of segments within the area and the size of system packets within each segment.

When a user module requires the sending of a message using a system packet it shall execute the Request System Packet ESR. With this ESR the user module shall request a system packet of a specific size and the task base register to be used as the pointer to the packet. If a system packet of the specified size is available, the base register shall be loaded with the base address of the packet and the associated SPR shall be loaded allowing both read and write access to the packet area. If a packet of the specified size is not available, SDEX/7 shall attempt to assign the smallest packet available of sufficient size to meet the request. If a packet cannot be assigned, a system packet unavailable error shall be indicated and CP control shall be passed to the error management function. If CP control is returned from error management, CP control shall be returned to the requesting user module with the ESR status indicating that no packet was available.

Once a packet is assigned, the user module may use it to pack a message. The Send System Message ESR shall then be used to initiate the message to up to four receiving user modules. The module shall be allowed to repeat the ESR to send the message to all desired receiving user modules. Upon receipt of the ESR, the receiving module count in the message control word shall be incremented by the number of receiving user modules specified by the ESR and a pointer to the message packet and the module index for each receiving user module shall be placed in the message scheduling list.

The scheduling of the user modules to receive the message shall be as specified in paragraph 3.3.2.2.2. Whenever a user module completes processing of a system message and executes the Module Exit ESR, the receiving module count in the message control word shall be decremented. When this count reaches zero, the system packet shall be reclaimed and made available for use by other user modules. The message control word required for each message shall include the following parameters:

a. Message type

b. Number of words

c. Reserved quarter word for receiving module count

3.3.2.4.8.2 LOCAL MESSAGE HANDLING

User modules may utilize local data stores for the purpose of sending messages. These messages shall be designated local messages and shall have the same format and be handled in the same manner as system messages specified in paragraph 3.3.2.4.8.1 with the following exceptions:

a. The user module provides the message packing area and no Request System Packet ESR is necessary.

b. The user module is required to include an item in its SAP for each local message packet.

c. The user module is responsible for checking the receiving module count to ensure that the packet is not reused before the last receiving module has processed the message.

d. In order to send the local message, the user module shall execute the Send Local Message ESR. With this

ESR the user module shall supply the SAP index indicating the SAP item containing the base address and size of the message. Also, the module index for each user module to receive the message shall be specified with the ESR.

3.3.2.4.9 MULTIPLE CP LIST/TABLE ACCESS CONSIDERATIONS

Simultaneous access by multiple CPs to any scheduling list shall be prohibited. List access by CPs shall be controlled by a flag which, when set, shall prohibit other CPs from accessing the data contained in the list.

If a CP attempts access to a scheduling list during the course of interrupt processing and finds the access flag set (list access prohibited), the CP shall pause to await clearing of the flag by the CP currently accessing the data. Lists shall be locked out only when necessary to insure the integrity and logic of the list contents. If any CP fails to gain access to a list within an allotted time, a list access error shall be declared and passed to the error management function for resolution. Whenever a CP gains access to a list, it shall set the access flag and proceed with processing. The access flag shall be cleared by the executing CP when processing of the list is complete. Any CP which encounters list lockout during its search of the successor, message, time-dependent, or background scheduling lists shall not wait for list access, but instead shall immediately return to the beginning of the scheduling function.

3.3.2.4.10 SCHEDULING FUNCTION ESRs

Executive services shall be provided in response to the following ESRs:

a. Register Successor Entrance

b. Request Successor Entrance

c. Delete Successor Entrance

d. Register Time-Dependent Entrance

e. Delete Time-Dependent Entrance

f. Register Background Entrance

g. Delete Background Entrance

h. Request System Packet

i. Send System Message

j. Send Local Message

k. Module CP Assignment

l. Module Exit

m. Suspend Task

n. Initiate Immediate Message

3.3.3 INTERRUPT MANAGEMENT INTRODUCTION

3.3.3.1 INTERRUPT MANAGEMENT INPUTS

The inputs to the Interrupt Management function shall consist of the interrupt Status Code (ISC), the Active Status Register (ASR) and the P-register. These are saved by AN/UYK-7 hardware in the appropriate Designate Storage Words (DSWs) at the time of interrupt occurrence.

3.3.3.2 INTERRUPT MANAGEMENT PROCESSING

All AN/UYK-7 interrupts shall be received and decoded by the interrupt management function. The interrupt management function shall respond as directed by modules according to their registration of responsibilities for error processing, time critical tasks and I/O channel interrupt processing. All interrupt processing shall follow two basic rules:

a. The CP monitor clock shall be saved and disabled during interrupt processing.

b. Interrupts representing errors shall be passed to the error management function for processing.

3.3.3.2.1 CLASS I INTERRUPT MANAGEMENT

The following types of Class I interrupts shall be decoded by SDEX/7:

a. CP-operand memory resume.

b. CP-IOC command resume.

c. CP-instruction memory resume.

d. CP-IOC interrupt code resume.

e. IOC-memory resume.

f. Intercomputer timeout.

g. Power tolerance.

With the exception of the Power Tolerance Interrupt, a hard-wired fault analysis routine in non-destructive readout (NDRO) memory initially receives CP control for Class I interrupts. The NDRO program then releases CP control to the alternate recovery routine (if provided). Otherwise, CP conrol is released to the normal recovery routine. Upon entry into either recovery routine, the registers used by the NDRO program shall be restored from the CMR locations in which the NDRO program saved register contents. These registers shall be restored as follows:

a. CMR 161 to class I DSW3

b. CMR 162 and 163 to Interrupt AO

c. CMR 164 to interrupt B1

d. CMR 165 to interrupt SO

The addresses for these routines shall be as specified in reference 2.4, item a for the NDRO program selected for the application.

3.3.3.2.1.1 CLASS I MALFUNCTIONS

The normal recovery routine shall restore the interrupt state registers used by the NDRO program and shall pass the error indication to the error management function. The alternate recovery routine for each CP shall also restore the interrupt state registers used by the NDRO program and shall check to determine if the failure is in the memory bank containing the instructions for SDEX/7. If so, processing shall be as follows:

a. In single or dual CP configuration, the SDEX/7 program shall be reloaded, or the CP shall be 7 stopped. The selection shall be a compile-time option for the user.

b. In configurations having three or more CPs, a check shall be made to determine if the executing CP is the highest priority CP, i.e., has the highest memory priority. If so, a reload indicator shall be set and the SDEX/7 shall be reloaded, or the CP shall be 7 stopped according to the option selected by the user. If not, the reload indicator shall be checked. If set, the SDEX/7 shall be reloaded, or the CP shall be 7 stopped according to the option selected by the user. Otherwise, the interrupt shall be ignored and CP control shall be returned to the point of interrupt.

3.3.3.2.1.2 POWER TOLERANCE INTERRUPT

The Power Tolerance Interrupt is a special Class I interrupt. This interrupt occurs when power falls below specified tolerance limits which might cause data loss. If the power tolerance interrupt occurs, the control memory register contents and the IOC monitor clock shall be saved in a core memory area reserved for this purpose, and then a manual jump instruction shall be executed. If normal power returns, the jump instruction shall transfer CP control to a routine which shall restore control memory contents and the IOC monitor clock (first CP to recover only), indicate power tolerance error and proceed to the error management function. If the power drops below a second tolerance level, an automatic master clear will occur.

If the computer power returns to normal after the automatic master clear, the CP will automatically start at an address specified in the NDRO memory provided that the AUTO START/MANUAL switch on the operator panel is in the AUTO START position. A recovery routine shall be provided at the specified address to process an auto start. This routine shall load the interrupt state base registers and the Initial Condition Words, and restore the remainder of control memory and the IOC monitor clock (first CP to recovery only). The power tolerance error shall be indicated and passed to the error management function.

3.3.3.2.2 CLASS II INTERRUPT MANAGEMENT

The AN/UYK-7 generates a Class II interrupt whenever certain software errors are detected, the CP monitor clock counts down through zero, or an interprocessor or breakpoint interrupt occurs. For the software error, the error shall be indicated and processing shall proceed as defined by the error management function. Floating point errors shall be an exception. Whenever the floating point error occurs, the interrupt status code (ISC) and the address of where the error occurred shall be saved. Through the use of the Return Floating Point Error ESR, the executing user module may request this information at any time. When this request is received, SDEX/7 shall respond as follows:

a. If no error has occurred task A0 shall be set to zero.

b. If a floating point error occurred, the ISC and address of the erring instruction shall be passed to the requesting user module.

The CP monitor clock shall be used for two slightly different but related operations. It shall be used to limit the amount of CP time spent within a user module to a user-defined maximum and for time-slicing specified user module tasks.

When the CP monitor clock interrupt is received, the task type currently being executed shall be determined. If the task is a background type, the task environment shall be saved in the user module's background save area, the module busy indicator shall be cleared, and CP control shall be passed to the scheduling function. If the interrupted task was a successor or time-dependent type which was registered to allow time-slicing, the time slice indicator for the task shall be set, the task environment shall be saved in the user module's successor or time-dependent save area as appropriate and CP control shall be passed to the scheduling function. For successor and time-dependent tasks not allowing time-slicing, and message tasks, the interrupt shall signify a user module time overrun and the module busy indicator shall be cleared and the error indication shall be passed to the error management function.

The interprocessor interrupt shall be handled as an error in single CP versions of SDEX/7. When an interprocessor interrupt is received, the error shall be indicated and processing shall proceed as defined by the error management function.

In multiple CP configurations, the interprocessor interrupt shall indicate that a CP has responded to an IOC monitor clock interrupt from other than the IOC being used for master timing. (See paragraph 3.3.3.2.3.2). Upon receipt of the interprocessor interrupt, the time-to-initiate-execution (TE) for the earliest time-critical task shall be compared to the contents of the IOC real time clock. If the TE is greater than the real time clock, the interprocessor interrupt shall be igonored and CP control shall be returned to the point of interrupt. Otherwise, processing shall proceed as specified for handling an IOC monitor clock interrupt in paragraph 3.3.3.2.3.2.

The breakpoint interrupts shall be used only for program maintenance operations. When a breakpoint interrupt occurs, an information packet consisting of the task environment shall be built and CP control shall be passed to the user module registered for processing breakpoint interrupts. If a breakpoint interrupt occurs and no user module is registered for breakpoint processing the CP shall be 5 stopped with the breakpoint condition displayed on the maintenance panel.

3.3.3.2.3 CLASS III INTERRUPT MANAGEMENT

The Class III interrupts include I/O controller related notifications to the CP. The Class III interrupts shall be categorized as follows for interrupt management:

a. I/O Monitor Interrupts

b. IOC Monitor Clock Interrupts

c. I/O Illegal Instruction Interrupts

3.3.3.2.3.1 I/O MONITOR INTERRUPTS

The I/O monitor interrupts include Class III interrupts which are caused by user modules through initiation of an I/O Controller command. These include the following four interrupt types.

a. IOC external interrupt monitor.

b. IOC external function monitor.

c. IOC input data monitor.

d. IOC output data monitor.

User modules shall have the option of specifying whether, upon interrupt occurrence, CP control is to be passed immediately to the user module for interrupt processing, or that it's successor task be scheduled for processing subject to normal successor scheduling criteria. The user module shall specify the option using the Register I/O Interrupt Entrance ESR. The option shall be selectable for each of the four interrupt types on an I/O channel. User modules shall be allowed to share interrupt responsibility on an I/O channel so long as two user modules do not register for the same interrupt type on the I/O channel.

Upon receipt of an I/O interrupt, the IOC and channel numbers shall be isolated from the interrupt status code in Class III DSW 2. A check shall be made to insure that the interrupt is valid for the executing CP by determining if a disable interrupt operation for this IOC and channel is to be transferred to the executing CP on a subsequent pass through the scheduling function. If the interrupt for this IOC and channel are to be disabled, the interrupt shall be ignored and CP control shall be returned to the point of interrupt. Otherwise, the interrupt shall be processed as directed by the registered user module.

If no user module has registered responsibility for processing the interrupt, an error shall be indicated and CP control passed to the error management function. If a user module has registered to have its successor task scheduled in response to the interrupt, the Interrupt Status Code (ISC) and the current real time clock value shall be stored in the I/O storage table supplied by the user module, (see paragraph 3.3.3.4.2). The number of entries in this table shall be incremented, the user module's successor indicator shall be set and CP control shall be returned to the point of interrupt.

If the user module registered for immediate processing of the interrupt, the interrupted task environment shall be saved, and the dedicated base registers for the responsible user module shall be loaded from the module's SAP. The Class III interrupt status code shall be placed in task accumulator AO and CP control shall be passed to the user module's I/O interrupt entrance with Class III interrupts locked out. When the user module returns CP control using the Module Exit ESR, the saved task environment shall be restored, module history if requested shall be updated and CP control shall be returned to the point of interrupt.

The user module task associated with the I/O interrupt entrance shall not be protected from reentrance by multiple CPs.

Also, CP dedication for the user module shall not apply to the module's I/O interrupt task. In multiple CP configurations where there is not total memory sharing, it shall be the user's responsibility, through selective enabling and disabling of I/O channel interrupts in the CPs, to insure that the CP which answers a given interrupt can execute the user module registered for processing the interrupt.

3.3.3.2.3.2 IOC MONITOR CLOCK INTERRUPTS

The IOC monitor clock shall be used to maintain strict fixed interval timing for module time-critical tasks. Options shall be provided for user modules to perform processing at precise fixed intervals. These options shall be as follows:

a. User modules may acquire immediate CP control at its time-critical entrance when the interrupt occurs.

b. User modules may specify that a predefined I/O chain be initiated in response to the interrupt.

c. User modules may specify that its successor task be scheduled in response to the interrupt.

d. User modules may specify any combination of the above options.

User modules shall register for time-critical interrupts using the Register Time-Critical Entrance ESR. Using this ESR the user module shall specify one of the above options, the first time the interrupt is to occur and the time interval between interrupt occurrences. If the time interval parameters is zero, the interrupt shall occur only once. If not, the interrupt shall be automatically updated by SDEX/7. User modules shall be allowed to have only one time-critical request pending at any time. However, concurrent pending time-critical requests by different modules shall be allowed by SDEX/7. When the time-critical interrupt occurs, the time for the next interrupt shall be computed and saved, the IOC monitor clock shall be reloaded for the next pending time-critical task and the options requested by the registered user module shall be determined. If options b. or c. were requested, the designated I/O chain shall be initiated or the user module's successor indicator shall be set as requested. If option a. was not selected, CP control shall then be returned to the point of interrupt.

If option a. was requested, the interrupted task environment shall be saved, the dedicated base registers for the registered user module shall be loaded from the module's SAP and CP control shall be passed to the user module's time-critical entrance with Class III interrupts locked out. When the user module returns CP control using the Module Exit ESR, the saved task environment shall be restored, module history if requested shall be updated and CP control shall be returned to the point of interrupt.

User modules registered for a time-critical task shall not be protected from reentrance by multiple CPs. If option a. was specified, the user module shall receive immediate CP control even though another CP is currently processing another task within the user module. Also, CP dedication for the user module shall not apply to the module's time-critical task. In multiple CP configurations not having total memory sharing among processors, it shall be the user's responsibility to place those user modules having time-critical tasks in memory banks accessible by all CPs.

If an IOC monitor clock interrupt is received from an IOC other than that IOC being used by SDEX/7, an interprocessor interrupt shall be broadcasted to all CPs. The interprocessor interrupt shall be generated such that is is sent to all CPs except for the CP generating the interrupt. The handling of the interprocessor interrupt shall be as specified in paragraph 3.3.3.2.2.

3.3.3.2.3.3 I/O ILLEGAL INSTRUCTION INTERRUPTS

The AN/UYK-7 computer generates two types of Class III interrupts to indicate illegal use of IOC commands. These interrupts are for:

a. IOC illegal command address register instruction.

b. IOC illegal chain instruction.

For these interrupts, the error shall be indicated and processing shall proceed according to the error management function.

3.3.3.2.4 CLASS IV INTERRUPT MANAGEMENT

The Class IV interrupt results from the execution of an "Enter Executive State" instruction. All Class IV interrupts shall be requests for executive services. The interrupt status code stored in the Class IV DSW 2 shall be used to identify the service being requested by the user modules. This code is provided by the 16 bit sy field of the Enter Executive States instruction. If the interrupt status code cannot be identified an error condition shall be indicated and passed to the error management function. Also, for errors isolated during the processing of an ESR, the user shall have the option of specifying the action to be taken. A SDEX/7 compile-time option shall be provided that allows the user to specify that all ESR errors are to be processed via the error management function, or that ESR errors are to be indicated by a negative status flag set in task accumulator A0 and CP control returned to the requesting user module at the point of ESR request. The set of execution services to be provided and the associated processing of the ESRs shall be as specified in Table 3-1.

3.3.3.3 INTERRUPT MANAGEMENT OUTPUTS

The outputs of the interrupt management function shall consist of the following:

a. Error indications defining each error condition.

b. Task environment for breakpoint interrupts.

c. I/O channel interrupts.

   (1) A successor task is scheduled with the interrupt status code and the value of the IOC real time clock saved in the user module's I/O storage table (see paragraph 3.3.3.4.2).

   (2) CP control is passed to the user module's I/O interrupt entrance with the interrupt status code in a task state accumulator.

3.3.3.4 INTERRUPT MANAGEMENT SPECIAL REQUIREMENTS

3.3.3.4.1 INTERRUPT MANAGEMENT ESRs

The SDEX/7 shall provide the following ESRs:

EXECUTIVE SERVICE	DESCRIPTION	INPUTS	OUTPUTS	VALIDITY CHECK
1. Enter Module	This ESR shall cause the specified module to be entered on the SDEX/7 list of modules. Upon receipt of this ESR, the specified user module's SAP location shall be placed in the SDEX/7 module list and a unique module index will be assigned for the module, corresponding to the item number of the module list used for the module's SAP location storage.	P1 – Absolute address of the user module's SAP	Status – Indicating completion or lack of completion of the ESR	The module number obtained from the preamble located by the specified SAP entry shall be unique.
2. Activate Module	This ESR shall cause the specified user module to be placed active, i.e., allow it to be eligible for scheduling and I/O functions. Upon receipt of this ESR, the user module shall be indicated active in the executive lists.	P1 – Module Number of the user module to be activated	Status	The specified user module shall have previously been entered on the SDEX/7 module list.
3. Delete Module	This ESR shall cause the specified user module to be dropped from the SDEX/7 module list. Upon receipt of this ESR, the SAP location for the specified user module shall be cleared from the module list and the module index assigned to the specified user module shall be released.	P1 – Module Number of the user module to be deleted	Status	The specified user module shall be inactive on the scheduling lists.
4. Deactivate Module	This ESR shall cause the specified user module to be deactivated on the executive lists. Upon receipt of this ESR, the user module shall be cleared from all scheduling and responsibility lists and placed in an inactive status on the module list.	P1 – Module number of the user module to be deactivated	Status	The specified user module shall be active on the module list.
5. Register Successor Entrance	This ESR shall cause the specified user module to be entered on the successor scheduling list. Upon receipt of this ESR, the specified user module shall be assigned a successor indicator based upon the module's priority obtained from its preamble. The suspendable/non-suspendable attribute for the user module shall be saved, and the SAP index shall be saved as a pointer to the SAP item giving the base address of the successor save area.	P1 – Module number of the user module being registered. P2 – Suspension indicator 0 – non-suspendable 1 – suspendable P3 – SAP index	Status	The priority of the specified user module shall be unique, i.e., no other user module may be registered for the priority. The specified user module shall be active.
6. Request Successor Task	This ESR shall cause the assigned successor indicator to be set for the specified user module	P1 – Module number	Status	The specified user module shall be registered for a successor entrance.
7. Delete Successor Entrance	This ESR shall cause the assigned successor indicator to be deleted for the specified user module.	P1 – Module number	Status	The specified user module shall be registered for a successor entrance.
8. Register Time-dependent Entrance	This ESR shall cause the specified user module to be entered on the time-dependent scheduling list. Upon receipt of this ESR, the first time to execute and the time interval parameter shall be placed on the time-dependent scheduling list and the suspendable/non-suspendable attribute and SAP index giving the save area shall be saved. If this ESR is received additional times for the specified user module, it shall be treated as a change in the registration and not an error.	P1 – Module number P2 – Suspension indicator 0 – non-suspendable 1 – suspendable P3 – first time to execute P4 – time interval P5 – SAP index	Status	The specified user module shall be active
9. Delete Time-dependent Entrance	This ESR shall cause the time-dependent registration to be deleted for the specified user module. Upon receipt of this ESR, the specified user module's entry on the time-dependent scheduling list shall be deleted.	P1 – Module number	Status	The specified user module shall be registered for a time-dependent entrance.

TABLE 3-1. EXECUTIVE SERVICE REQUESTS
(Page 1 of 6)

EXECUTIVE SERVICE	DESCRIPTION	INPUTS	OUTPUTS	VALIDITY CHECK
10. Register Background Entrance	This ESR shall cause the specified user module to be entered on the background scheduling list. Upon receipt of this ESR, the first time to execute shall be placed on the background scheduling list. If this ESR is received additional times for the specified user module, it shall be treated as a change in the registration and not an error.	P1 – Module number P2 – First time to execute P3 – SAP index for SAP item for the background save area	Status	The specified user module shall be active.
11. Delete Background Entrance	This ESR shall cause the background registration to be deleted for the specified user module. Upon receipt of this ESR, the specified user module's entry on the background scheduling list shall be deleted.	P1 – Module number	Status	The specified user module shall be registered for a background entrance.
12. Register Time-critical Entrance	This ESR shall cause the specified user module to be registered for a time-critical interrupt. Upon receipt of this ESR, the first time-to-execute shall be stored in the time-critical interrupt list along with the time-interval and processing options selected for the module. The absolute address of the I/O chain shall be saved if appropriate. If the first time to execute is less than that currently counting down in the IOC monitor clock, the clock shall be reloaded for the new value. Additional requests for the specified user module shall be treated as an update of time-critical requirements, not an error.	P1 – Module number P2 – First time to execute. P3 – Time interval P4 – Time-critical options. P5 – Absolute address of I/O chain (if appropriate).	Status	The specified user module shall be active. For schedule successor option, the module shall have a registered successor entrance.
13. Delete Time-critical Entrance	This ESR shall cause the time-critical registration to be deleted for the specified user module. Upon receipt of this ESR, the specified user module's time-critical parameters shall be deleted from the time-critical interrupt list. If the IOC monitor clock is currently counting down for the specified user module, it shall be reloaded for the next time-critical interrupt following that for the specified user module.	P1 – Module number	Status	The specified user module shall be registered for a time-critical entrance.
14. Register I/O Interrupt Entrance	This ESR shall cause the requesting user module to be registered for processing I/O interrupts on specified IOC/channel functions. Upon receipt of this ESR, the requesting module index shall be saved in the I/O responsibility list for each IOC/channel function found in the module's registration packet. Additionally, the user module's specified interrupt response option for each registered IOC/channel function shall be stored.	P1 – SAP index P2 – Pointer to user module's I/O registration packet P3 – Force indicator	Status	The requesting user module shall be active, and no other user module shall be registered for the specified IOC/channel function(s), unless P3 is set, in which case the previous registration shall be ignored.
15. Delete I/O Interrupt Entrance	This ESR shall cause the requesting user module registration for the IOC/Channel functions specified in the module's I/O registration packet to be deleted. Upon receipt of this ESR, the registration of the requesting user module for the IOC/channel functions specified in the I/O registration packet shall be deleted.	P1 – SAP index P2 – Pointer to module's I/O registration packet	Status	The requesting user module shall be legally registered for the specified IOC/channel functions specified.

TABLE 3-1. EXECUTIVE SERVICE REQUESTS
(Page 2 of 6)

EXECUTIVE SERVICE	DESCRIPTION	INPUTS	OUTPUTS	VALIDITY CHECK
16. Module Exit	This ESR shall cause CP control to be returned to SDEX/7 indicating a completion of the requesting user module's task being executed. Upon receipt of this ESR, exit housekeeping appropriate for the completed task shall be performed. If the exit is from a scheduled task, CP control shall be returned to the scheduling function. If the exit is from an interrupt task, CP control shall be returned to the point of interrupt.	None	None	None
17. Error Exit	This ESR shall cause CP processing to be resumed according to the option specified by the requesting user module (see paragraph 3.3.5).	P1 – Error processing option	Status	The requesting user module shall be legally registered for the error which has been processed.
18. Initiate System Message	This ESR shall cause the message contained in the system packet to be queued for processing by the specified receiving user modules. Upon receipt of this ESR, the system packet index and the specified module indexes shall be placed in the message scheduling list. Further, the number of receiving user modules for the message shall be incremented in the message control word.	P1 – System packet index P2 – Module number of receiving modules-up to four.	Status	The receiving user modules shall be legally entered on the SDEX/7 list of modules.
19. Initiate Local Message	This ESR shall cause the message contained in the requesting user module's data area to be queued for processing by the specified receiving user modules. Upon receipt of this ESR, the sending user modules SAP index, defining the SAP entry indicating the location and size of the message, and the receiving and sending module indexes shall be placed in the message scheduling list. Further, the number of receiving user modules for the message shall be incremented in the message control word.	P1 – SAP index P2 – Module number of receiving user modules - up to four	Status	The receiving modules shall be legally entered on the SDEX/7 list of modules.
20. Request System Packet	This ESR shall cause the specified transient base register and SPR to be loaded with the base address of the message and storage protection attributes allowing read and write access to the packet. Upon receipt of this ESR, the specified base register/SPR shall be loaded with the system packet address and the packet shall be indicated as assigned. If no packet of the requested size is available, a packet of a larger size (if any available) shall be assigned. The system packet index of the assigned packet shall be returned to the user module for identification of the packet when the message is sent.	P1 – Transient base register number P2 – System packet size	1. Status 2. Transient base register/SPR loaded with the base of the packet and read/write access to the packet. 3. System packet index	The user module shall be legally registered on the SDEX/7 list of modules. A system packet of the requested size or larger shall be available.
21. Register Error Responsibility	This ESR shall cause the requesting user module to be registered as responsible for processing the errors specified in the module's error registration packet. Upon receipt of this ESR, the user module index of the requesting user module shall be stored in the error registration list for each error specified in the module's error registration packet.	P1 – SAP index P2 – Pointer to module's error registration packet	Status	The requesting user module shall be active on the SDEX/7 list of modules. No other user module shall be registered for the specified error.

TABLE 3-1. EXECUTIVE SERVICE REQUESTS
(Page 3 of 6)

EXECUTIVE SERVICE	DESCRIPTION	INPUTS	OUTPUTS	VALIDITY CHECK
22. Delete Error Responsibility	This ESR shall cause the requesting user module's error responsibilities to be deleted for the errors specified in the module's error registration packet.	P1 – SAP index P2 – Pointer to module's error registration packet	Status	The requesting user module shall be legally registered for the errors being deleted.
23. Register Breakpoint Responsibility	This ESR shall cause the requesting user module to be registered as responsible for setting and clearing the breakpoint register and processing any breakpoint interrupts.	None	Status	The requesting user module shall be active on the SDEX/7 list of modules. No other user module shall be currently registered for breakpoint processing.
24. Delete Breakpoint Responsibility	This ESR shall cause the requesting user module's breakpoint responsibility to be deleted.	None	Status	The requesting user module shall be registered for breakpoint processing.
25. Enable I/O Interrupts	This ESR shall cause I/O monitor interrupts to be enabled for the specified IOC and channels by the executing CP. Upon receipt of this ESR, the "Allow Enable Interrupt" instruction shall be built using the specified IOC and channel indicators. The instruction shall be executed. If additional CP's are to also enable interrupts on the specified IOC and channels, the instruction shall be saved in a transfer table and an indicator set for the CPs such that the instruction is executed by that CP when it enters the scheduling function.	P1 – IOC number P2 – I/O channel indicators P3 – CP number(s)	Status	The requesting user module shall be registered for a function of the specified IOC/channel(s).
26. Disable I/O Interrupts	This ESR shall cause I/O monitor interrupts to be disabled for the specified IOC and channels by the executing CP. Processing of this ESR shall be as specified in 25. above, except for disable interrupts.	P1 – IOC number P2 – I/O channel indicator. P3 – CP number(s)	Status	The requesting user module shall be registered for a function on the specified IOC/channel(s).
27. Initiate I/O	This ESR shall cause the specified I/O chain to be initiated. Upon receipt of this ESR, the "Initiate I/O" instruction shall be built using the IOC and address parameters. The I/O chain shall then be initiated.	P1 – SAP index P2 – Relative address of the I/O chain P3 – IOC number	Status	The IOC number shall be for one of the IOC's currently operating in the configuration.
28. Delete IOC	This ESR shall cause the specified IOC to be listed inoperable by SDEX/7. Upon receipt of this ESR, all module registrations for interrupt responsibility on the affected channels shall be deleted. If the specified IOC is currently being used for time-critical interrupts and real time clock timing, the indicated IOC shall be used for this purpose.	P1 – IOC number of IOC to be deleted P2 – IOC number of IOC to be used for timing	Status	The requesting user module shall be legally registered for error processing
29. Allow Class III Interrupts	This ESR shall cause Class III interrupts to be enabled in the executing CP.	None	Status	The requesting user module shall be legally registered for error processing.
30. Prevent Class III Interrupts	This ESR shall cause Class III interrupts to be disabled in the executing CP	None	Status	The requesting user module shall be legally registered for error processing.

EXECUTIVE SERVICE	DESCRIPTION	INPUTS	OUTPUTS	VALIDITY CHECK
31. Send Immediate Message	This ESR shall cause the specified message to be passed directly to the common system module (module number defined at compile-time). Upon receipt, the task base registers shall be saved for the requesting user module and the dedicated task base registers shall be loaded for the common system module. CP control shall be passed to the common system module at its message entrance. Upon exit from the common system module, the requesting module's task base registers shall be restored and CP control shall be returned to the requesting user module at the point of the request.	P1 – SAP index giving the message location.	None	Common system module shall be identified by a compile-time parameter and be entered on the SDEX/7 module list.
32. Set Breakpoint	This ESR shall cause the breakpoint register to be loaded. Upon receipt of this ESR, the specified address and breakpoint conditions shall be loaded into the breakpoint register. The specified address and conditions shall be saved for all other CPs allowed to execute the user module such that the register can be loaded when the CP's enter the scheduling function.	P1 – Address P2 – Breakpoint condition	Status	The requesting user module shall be legally registered for breakpoint processing.
33. Clear Breakpoint	This ESR shall cause the breakpoint register to be cleared. Processing of this ESR shall be as specified in 32.above.	None	Status	The requesting user module shall be legally registered for breakpoint processing.
34. Load Transient Base Register	This ESR shall cause the specified base register to be loaded from the SAP entry specified with the ESR.	P1 – SAP index P2 – Transient base register number (S2/SPR2 – S6/SPR6)	1. Status 2. Transient base register loaded as specified.	The specified base register shall be restricted to SR/SPR2 through S6/SPR6.
35. Store Transient Base Register	This ESR shall cause the contents of the specified base register and SPR to be stored in the requesting user module's SAP at the specified entry.	P1 – SAP index P2 – Transient base register number (S2/SPR2 - S6/SPR6)	1. Status 2. Transient base register stored as specified.	The specified base register shall be restricted to S2/SPR2 through S6/SPR6.
36. Return Floating Point Error	This ESR shall return the indication of the most recent floating point error occurring during the current execution of the user module's task.	None	1. Floating point error indication. 0 – no error Not Zero – error indication and address of the instruction causing the error	None
37. Convert Module Number to Index	This ESR shall convert the specified user module number to the SDEX/7 assigned module index corresponding to the module number, and return it to the requesting module.	P1 – Module number	1. Status 2. Module index	The specified user module number shall be that of a module entered on the SDEX/7 module list.
38. Convert Module Number to Name	This ESR shall convert the specified user module number to the name of the user module. Upon receipt of this ESR, the preamble of the specified user module shall be located, the module's name shall be placed in a task state accumulator and returned to the requesting user module.	P1 – Module number	1. Status 2. Module name	The specified user module number shall be that of a module entered on the SDEX/7 module list.

TABLE 3-1 EXECUTIVE SERVICE REQUESTS

EXECUTIVE SERVICE	DESCRIPTION	INPUTS	OUTPUTS	VALIDITY CHECK
39. Dedicate Module to CP	This ESR shall cause the specified user module to be registered as executable by the specified CPs.	P1 – Module number P2 – CP indicators	Status	The specified user module shall be entered on the SDEX/7 module list.
40. Return CP Number	This ESR shall place the number of the executing CP in a task state accumulator and return it to the requesting user module.	None	None	None
41. Return SAP Data	This ESR shall cause the SAP items for each user module, entered on the module list, to be transferred to the requesting user module's area.	P1 – Pointer to module-supplied area P2 – SAP index	SAP items for all entered modules	Requesting module shall be legally registered on the SDEX/7 module list.
42. Suspend Task	This ESR shall cause the requesting task to be suspended at the point of ESR request. Upon receipt of this ESR from a valid successor, time-dependent or background task, the task environment shall be saved in the module-supplied save area, the task shall be rescheduled according to the input parameters and CP control shall be passed to the scheduling function.	P1 – Reschedule Indicator*	Status	The requesting task shall be a successor or time-dependent task registered for time-slicing or a background task.

*P1 – Reschedule Indicator – indicates the rescheduling criteria as follows:

(1) Zero Value – If the requesting task is a successor task, suspend the task and do not set the corresponding successor indicator in the successor scheduling list. If the requesting task is a time-dependent or background task, suspend the task and reschedule the task using the current value of the IOC real time clock.

(2) Non-zero Value – If the requesting task is a successor task, suspend the task and set the corresponding successor indicator. If the requesting task is a time-dependent or background task, suspend the task and reschedule the task at the current value of the IOC real time clock plus the provided non-zero value.

TABLE 3-1. EXECUTIVE SERVICE REQUESTS
(Page 6 of 6)

a. Register Error Responsibility.

b. Delete Error Responsibility.

c. Register Breakpoint Responsibility.

d. Delete Breakpoint Responsibility.

e. Register Timed Critical Entrance.

f. Delete Timed Critical Entrance.

g. Register I/O Interrupt Entrance.

h. Delete I/O Interrupt Entrance.

3.3.3.4.2 MODULE I/O STORAGE AREA

If a user module registers to have its successor task scheduled in response to an I/O channel interrupt, the user module shall supply a SAP index and a relative pointer to a storage area of sufficient size for interrupt data. SDEX/7 shall protect against overflowing the module's storage area and treat pending overflow conditions as an error which shall be passed to the error management function (see paragraph 3.3.5).

3.3.4 INPUT/OUTPUT MANAGEMENT INTRODUCTION

Management of input and output operations shall be provided by the input/output management function. Through this function, user modules shall be provided the capability to register I/O channel responsibilities and initiate I/O operations using ESRs.

3.3.4.1 INPUT/OUTPUT MANAGEMENT INPUTS

Inputs to the input/output management function shall be ESRs by which user modules can request the initiation of

I/O operations and the enabling and disabling of I/O channel interrupts.

3.3.4.2 INPUT/OUTPUT MANGEMENT PROCESSING

Immediate initiation of an IOC command chain shall be provided whenever a user module executes the Initiate I/O ESR. Using this ESR the user module shall specify the IOC number and the location of the IOC command chain to be initiated. Upon receipt of this ESR, the appropriate "initiate I/O" instruction shall be built and executed. CP control shall be then returned to the requesting user module to continue processing.

All user modules responsible for interrupts on a channel shall be required to register their desired response for interrupts. A maximum of four modules shall be allowed to respond to interrupts on one channel, i.e., one for each interrupt type. User modules shall be allowed to register response options for each of the four interrupts for each channel. The response options provided to user modules are:

a. Immediate user module entry through the I/O interrupt entrance or,

b. Queuing of the interrupt and scheduling of the user module's successor entrance.

The I/O interrupt entrance provides immediate response to interrupts, but causes suspension of the interrupted user module and lockout of Class III interrupts until the I/O interrupt task is completed. Use of this response option should be limited only to those conditions requiring immediate processing. The second method of processing I/O inputs allows SDEX/7 to queue a number of consecutive interrupts to await processing through the responsible user module's successor entrance.

User modules registered for I/O channels shall be allowed to selectively enable and disable interrupts on those channels through the use of ESRs. A user module which has registered responsibility for an I/O channel (or channels) shall be allowed to enable or disable interrupts of any or all of those channels. Interrupt enable or disable instructions shall be passed to all CPs except when specifically prohibited by the user. The enable/disable interrupt instructions shall be executed by receiving CPs when entering the scheduling function (See paragraph 3.3.2.2).

3.3.4.3 INPUT/OUTPUT MANAGEMENT OUTPUTS

The outputs for the input/output management function shall be the interrupt status code passed to the responsible user module and the setting of successor indicator bits.

3.3.4.4 INPUT/OUTPUT MANAGEMENT SPECIAL REQUIREMENTS

The special requirements for the input/output management function shall be the following ESRs:

a. Register I/O Interrupt Entrance.

b. Delete I/O interrupt Entrance.

c. Initiate I/O.

d. Enable I/O Interrupts.

e. Disable I/O Interrupts.

3.3.5 ERROR MANAGEMENT INTRODUCTION

The error management function shall handle all error conditions isolated by other executive functions. Through this function a user module may selectively register responsibility for processing any error conditions and hence receive immediate CP control at its message entrance should the error occur. If a specialized user error recovery module is used which is responsible for processing all errors, it shall conform to the standard module interface. Errors identified by SDEX/7 shall include those indicated by AN/UYK-7 interrupts and those isolated by SDEX/7 functions due to invalid data or improperly sequenced executive operation.

3.3.5.1 ERROR MANAGEMENT INPUTS

The inputs to the error management function shall include the following:

a. ESRs registering module error processing responsibility.

b. Error identification information from other SDEX/7 functions.

3.3.5.2 ERROR MANAGEMENT PROCESSING

The error conditions shall be identified either through decoding of an interrupt or by SDEX/7 functional processing inconsistencies. The actions to be taken for error conditions or user module registration shall be as follows:

a. Register user module error responsibility as requested. Only one user module shall be allowed to register for a specific error.

b. Non-registered errors shall cause a 5-Stop for software related errors or a 6-Stop for hardware related errors. At a CP stop the error information shall be displayed at the maintenance panel.

c. The illegal ESR errors shall be passed to the registered user module. If no user module is registered, CP control shall be returned to the module, which requested the ESR, with an error status indication.

d. Concurrent error conditions within a CP, with the exception of the occurrence of a power tolerance error during the processing of another error, shall always force a CP stop with the information for the latest error displayed on the maintenance panel.

e. If a power tolerance error occurs during the processing of another error, the power tolerance error shall be processed immediately.

User modules shall be allowed to register responsibility for individual errors or by groups of errors. The following groups shall be recognized for registration:

a. Class I hardware errors.

b. Class II software errors.

c. Class III IOC software errors.

d. Class IV executive software errors.

For the errors in each class a unique code shall be defined to identify the individual error. Table 3-2 provides a list of

ERROR CONDITIONS	CLASS	CODE
CP-Operand Memory Resume	001	000
CP-IOC Command Resume	001	001
CP-Instruction Memory Resume	001	002
CP-IOC Interrupt Code Resume	001	003
IOC Memory Resume	001	012
Intercomputer Timeout	001	013
Power Tolerance	001	017
Interprocessor Interrupt	002	000
CP Illegal Instruction Error	002	002
Priviledged Instruction Error	002	003
Operand Read or Indirect Addressing	002	006
Operand Write	002	011
Operand Limit	002	012
Instruction Execute	002	015
Instruction Limit	002	016
CP Monitor Clock	002	017
IOC Illegal CAR Instruction	003	000
IOC Clock Failure	003	001
IOC Illegal Chain Instruction:	003	004
External Interrupt	003	004
External Function	003	005
Output	003	006
Input	003	007
IOC Monitor Clock Interrupt	003	012
IOC External Interrupt Monitor	003	014
IOC External Function Monitor	003	015
IOC Output Data Monitor	003	016
IOC Input Data Monitor	003	017
Illegal ESR	004	001
Illegal I/O Request	004	002
Illegal Message Transfer Request	004	003
Illegal Module Entry on Task List	004	004
Overflow Scheduling List	004	005
Overflow Message List	004	006
Invalid ESR Data	004	007
System Packet Unavailable	004	010
Test and Set Time Out	005	001

TABLE 3-2. ERROR CONDITIONS

error conditions by class and code which shall be available for user module responsibility registration.

Error information shall be communicated to the registered module through an error packet built by the error management function and passed to the module. The error packet shall contain a message control word (message type - 00) and the following data:

a. Error Class (see tables 3-2).

b. Error code (see table 3-2).

c. IOC number affected as appropriate.

d. Memory bank or I/O channel as appropriate.

e. P-register as set at the time of the error.

f. CP executing when the error occurred.

g. Module number of the module being executed when the error occurred.

h. State (interrupt or task) of the CP when the interrupt occurred.

i. Module task being executed when the error occurred.

j. Scheduling list affected if the error is a list overflow or list access time-out error.

k. Task state environment including accumulators, index registers and base registers/SPRs.

l. DSWs of the interrupt class when the error occurred.

m. ESR in which the error was located (Class IV, code 7 only).

When an error occurs, the error packet shall be built, the dedicated base registers shall be loaded from the registered module's SAP and the base of the error packet shall be loaded into task base register S6. CP control shall be passed to the user module at its message entrance in the task state. When the user module has completed processing of the error, CP control shall be returned to SDEX/7 using the Error Exit ESR. Using this ESR the user module can direct SDEX/7 processing according to the following options:

a. Return to point of error, i.e., ignore the error and continue processing.

b. Set erring user module's successor indicator and return to point of error.

c. Return CP control to the scheduling function.

When no user module is registered for processing the error, a CP stop shall occur. The error class, code, executing user module number, and active module task when the error occurred shall be displayable on the maintenance panel from task state accumulators. The operator may select one of the above error exit options in interrupt accumulator A0 and depress the computer start switch to resume processing.

3.3.5.3 ERROR MANAGEMENT OUTPUTS

The Error Management outputs shall include the following:

a. The error packet shall be built and CP control passed to the message entrance of the registered user module.

b. The error class, code, module identification, and active task type for display from task acumulators at the maintenance panel.

3.3.5.4 ERROR MANAGEMENT SPECIAL REQUIREMENTS

The following ESRs shall be provided:

a. Register Error Responsibility.

b. Delete Error Responsibility.

c. Error Exit.

3.4 ADAPTION

3.4.1 SDEX/7 COMPILE-TIME CONFIGURABILITY

The SDEX/7 capabilities and features shall be tailorable for specific applications through the use of compile-time options and parameters. If an option is not selected by the user, no instructions or data pertaining to the option shall be included in the resulting SDEX/7. The following compile-time options shall be provided.

a. CP/IOC configuration selection

b. Successor task scheduling selection

c. Time-dependent task scheduling selection

d. Background task scheduling selection

e. Task type scheduling priority selection

f. Recovery option for memory resume errors in executive memory banks

g. Error management option for ESR errors

h. Successor and time-dependent time-slicing

3.4.1.1 CP CONFIGURATION SELECTION

The SDEX/7 functions shall be provided for all possible AN/UYK-7 configurations with up to a maximum of 4 CPs. The SDEX/7 program shall be structured such that the program for single CP configurations does not contain any instructions or data pertaining to multiple CP/IOC configurations. Further, in the program for multiple CP/IOC configurations only instructions and data shall be included as necessary to support the SDEX/7 functions for the specific number of CP/IOCs.

3.4.1.2 SUCCESSOR TASK SCHEDULING SELECTION

The user shall have the option at SDEX/7 compile-time of selecting whether or not the scheduling function shall include the successor task scheduling feature. If not selected, all SDEX/7 instructions, and data pertaining to successor task scheduling including ESRs shall be omitted from the SDEX/7 program for the specific number of CP/IOCs.

3.4.1.3 TIME-DEPENDENT TASK SCHEDULING SELECTION

The user shall have the option at SDEX/7 compile-time of selecting whether or not the scheduling function shall include the time-dependent task scheduling feature. The constraints shall be as defined in paragraph 3.4.1.2, except for time-dependent task scheduling.

3.4.1.4 BACKGROUND TASK SCHEDULING SELECTION

The user shall have the option at SDEX/7 compile-time of selecting whether or not the scheduling function shall include the background task scheduling feature. The constraints shall be as defined in paragraph 3.4.1.2, except for background task scheduling.

3.4.1.5 TASK TYPE SCHEDULING PRIORITY SELECTION

At SDEX/7 compile-time the user shall have the option of specifying the order in which the scheduling task lists are searched. This option shall include all possible combinations of the four task scheduling types defined and any lesser set depending upon user selection of task scheduling options (see paragraphs 3.4.1.2 through 3.4.1.4).

3.4.1.6 RECOVERY OPTION FOR MEMORY RESUME ERRORS

The user shall have the option at SDEX/7 compile-time of selecting the processing for memory resume interrupt indicating a failure of one of the memory banks containing the SDEX/7 program. The user may select one or both of the following:

a. 7-stop the CP whenever the memory failure occurs.

b. Process a reload of the SDEX/7 program in an alternate memory bank.

When both options are selected, SDEX/7 shall first test for key 7 set. If set, the CP will be 7-Stopped. Otherwise, processing shall proceed with a reload of the SDEX/7 program in an alternate memory bank.

3.4.1.7 ERROR MANAGEMENT OPTION FOR ESR ERRORS

The user shall have the option at SDEX/7 compile-time of selecting the processing for errors occurring during the course of an ESR. The option shall be either of the following:

a. The error shall be returned to the user module requesting the ESR with a status indicator set negative in task accumulator A0. CP control shall be returned to the requesting user module at the point of request.

b. The error shall be handled via the error management function for processing by the user module which registered responsibility for processing the error.

3.4.1.8 SUCCESSOR AND TIME-DEPENDENT TIME-SLICING

The user shall have the option as SDEX/7 compile-time of selecting the capability to allow successor and time-dependent task time-slicing. This option shall allow the user to register successor and time-dependent tasks as suspendable with processing as specified in paragraph 3.3.2.4.4.2 and 3.3.2.4.4.3.

3.4.1.9 SDEX/7 COMPILE TIME PARAMETERS

The compile-time parameters provided by SDEX/7 shall include the following:

a. Scheduling list capacities.
b. System message packing area size.
c. System message packing area segment size ratios.
d. System packet sizes for each segment.
e. Assigned CP indexes.
f. Module overrun parameters by module task type.
g. Common system module number.
h. Initial module numbers.
i. Maximum time CP is allowed away from the scheduling function.
j. Scheduling list capacities.

3.4.2 SDEX/7 CAPACITIES

The design of SDEX/7 shall be such that its capacities are adaptable to individual system requirements through the use of compile-time parameters (see paragraph 3.4.1.6). These capacities shall be constrained by the following:

a. SDEX/7 shall interface with a maximum of 63 core resident modules.

b. SDEX/7 shall handle up to 255 queued messages.

c. All other capacities shall be as specified by the user through SDEX/7 compile-time parameters.

3.4.3 MAXIMUM SDEX/7 MEMORY REQUIREMENT

The total size of the SDEX/7 program for single CP/IOC configurations shall not exceed 3500 (decimal) words. For multiple CP/IOC configurations the SDEX/7 program size shall not exceed 3500 words plus 750 words for each additional CP and 64 words for each additional IOC.

These size estimates assume that compile-time capacities include at least one item in each scheduling list and a system message packing area sufficient for support of 63 user modules.

3.4.4 MAXIMUM TIMING REQUIREMENTS

Table 3-3 lists the maximum SDEX/7 executive overhead times which shall be allowed for passing CP control to scheduled and interrupt tasks. Table 3-4 lists the maximum SDEX/7 executive overhead for the ESRs.

SCHEDULING FUNCTION	OVERHEAD TIME FOR SCHEDULER TASKS IN μSEC		
	SINGLE CP EXECUTIVE	MULTIPLE CP EXECUTIVE	FOR MODULE HISTORY ADD:
Successor Entrance	140	+20	+25
Message Entrance	180	+60	+25
Time-Dependent Entrance	175	+30	+25
Background Entrance	175	+30	+17
Time-Critical Interrupt — Automatic I/O Initiation	50	+ 9	+ 0
Time-Critical Interrupt Set Successor Indicator	50	+18	+ 0
Time-Critical Interrupt Module Entrance	340	+30	+22
I/O Interrupt Set Successor Indicator	120	+18	+ 0
I/O Interrupt Module Entrance	295	+20	+22

The time for scheduling a successor, message, time-dependent or background entrance is the time spent in SDEX/7 starting at the top of the scheduling loop, entering and exiting the user module, and returning to the top of the scheduling loop. The time for interrupt processing is the executive overhead time beginning at the time the interrupt is received by the CP and ending with control returned to the interrupted program. Also the times shown assume no memory access conflicts among CPs and IOCs in the configuration.

TABLE 3-3. SDEX/7 SCHEDULING OVERHEAD TIME

EXECUTIVE SERVICE REQUEST	OVERHEAD TIME IN μSEC
1. Enter Module	120
2. Activate Module	40
3. Delete Module	80
4. Deactivate Module	250*
5. Register Successor Entrance	95
6. Request Successor Task	60
7. Delete Successor Task	100*
8. Register Time-Dependent Entrance	50
9. Delete Time-Dependent Entrance	70*
10. Register Background Entrance	75
11. Delete Background Entrance	70*
12. Register Time-Critical Entrance	90*
13. Delete Time-Critical Entrance	75*
14. Register I/O Interrupt Entrance	150*
15. Delete I/O Interrupt Entrance	150*
16. Module Exit	80
17. Error Exit	80
18. Initiate System Message	95
19. Initiate Local Message	100
20. Request System Packet	60
21. Register Error Responsibility	175*
22. Delete Error Responsibility	175*
23. Register Breakpoint Responsibility	120
24. Delete Breakpoint Responsibility	120
25. Enable I/O Interrupt	80
26. Disable I/O Interrupt	80
27. Initiate I/O	50
28. Delete IOC	120
29. Allow Class III Interrupts	50
30. Prevent Class III Interrupts	50
31. Send Immediate Message	75
32. Set Breakpoint	70
33. Clear Breakpoint	70
34. Load Transient Base Register	35
35. Store Transient Base Register	35
36. Return Floating Point Error r	35
37. Convert Module Number to Module Index	50
38. Convert Module Number to Module Name	70
39. Dedicate Module to CP	95
40. Return CP Number	35
41. Return SAP Data	500*
42. Suspend Task	70

*Times listed are maximums assuming a multiple CP/IOC configuration with no memory access conflicts. Time is an average estimate, time varies according to processing number of data items in the affected data table.

TABLE 3-4. SDEX/7 EXECUTIVE OVERHEAD TIMES FOR ESRS

Section 4

QUALITY ASSURANCE

4.1 INTRODUCTION

The SDEX/7 computer program shall be tested at two levels: 1) computer program testing; and 2) acceptance tesing. Computer Program Test Plans (CPTPL) and Computer Program Test Procedures (CPTPR) shall be provided for each level of testing. The CPTPL shall specify the hardware configuration and computer programs required to accomplish each level of testing. The contractor/activity producing SDEX/7 shall be responsible for providing all test programs and hardware, and for the successful accomplishment of all test specified herein.

4.2 COMPUTER PROGRAM TEST REQUIREMENTS

Computer program tests shall be performed using a CMS-2 object tape of the SDEX/7 program that is free of run-time patches. These tests shall verify that all segments of the computer program satisfy the performance requirements specified in section 3 of this document. The procuring agency shall approve the CPTPL and CPTPR before computer program testing is performed.

The computer program tests shall be performed in an AN/UYK-7 configuration having at least two CPs and two IOCs. In addition to verifying the performance requirements specified in section 3, the computer program tests shall include timing tests for the scheduling function and all ESRs.

4.3 ACCEPTANCE TEST REQUIREMENTS

Acceptance testing shall be performed using a CMS-2 object tape of the SDEX/7 program that is free of run-time patches. All executive functions shall be demonstrated by this level of testing. Acceptance testing shall be performed in an AN/UYK-7 configuration having at least two CPs and two IOCs and total memory sharing. At least two versions of the SDEX/7 computer program shall be demonstrated. These versions shall include the following:

a. single CP/single IOC program

b. two CP/two IOC program

The acceptance tests shall, at a minimum, demonstrate each SDEX/7 function as specified in this document. Further, an exhaustion test shall be performed wherein a dynamic test program requiring at least 70 percent of the available processing time and utilizing all aspects of the scheduling, interrupt management and I/O management functions is run for a period of not less than twelve hours. Further, the task load shall be under operator control via some interactive device. The operator shall be provided the capability to vary the processing load at will using directives to the test program via the interactive device. During all acceptance testing the criteria defined in paragraphs 4.3.1 through 4.3.5 shall be used in test verification. The performance of the acceptance test shall be indicated by specific data recorded by the test program. This data shall be extracted by check-points built into the test program. The SDEX/7 program shall not be altered in any way for the purpose of testing.

4.3.1 INITIALIZATION FUNCTION

The Initialization function shall be tested for loading and start-up operations. The test shall exercise the function for a range of from 1 to a maximum of 63 loaded user modules. Within this test, the SDEX/7 program shall load and initialize each user module such that each user module receives and processes an initialization message from SDEX/7.

4.3.2 SCHEDULING FUNCTION

The scheduling function shall be tested for each defined user module task type. The tests shall include registration, deletion and scheduling of all user module tasks. The test shall exercise the scheduling function at minimum and full scheduling loads and be representative of all possible scheduling combinations for single and multiple CP/IOC configurations. Within this test the following shall be verified:

a. User module tasks receive CP control according to the priority of task type scheduling.

b. Successor tasks shall receive CP control based upon module priority.

c. Message tasks shall receive CP control based on a strict first-in/first-out flow of messages.

d. Time-dependent tasks shall receive CP control according to the time-to-execute parameters for the registered tasks and the specified scheduling criteria.

e. Background tasks shall receive control according to the time-to-execute parameters for the registered tasks and specified scheduling criteria.

4.3.3 INTERRUPT MANAGEMENT

The interrupt management function shall be tested for each class of AN/UYK-7 interrupts. The tests shall include nested interrupt conditions and representative module task preemption combinations during program operation. Within the test each ESR shall be tested for validity. Also, each AN/UYK-7 interrupt shall be processed and action taken as specified herein for each interrupt. The test shall reflect proper isolation of errors as well as the use of the non-error interrupt types. Module interrupt tasks shall be given CP control as specified such that for time-critical tasks, strict intervals are maintained for successive executions and I/O interrupts are processed as specified.

4.3.4 INPUT/OUTPUT MANAGEMENT

The Input/Output Management function shall be tested for representative combinations of input/output operations. The test shall be performed for both single and multiple CP/IOC combinations such that interrupt enable, disable and module interrupt response are tested in a representative set of combinations. Within the test, I/O control shall be provided for up to two IOCs with 16 channels each. In the multiple CP configuration, interrupt enable and disable request shall be accomplished as specified in this document.

4.3.5 ERROR MANAGEMENT

The error management function shall be tested for all errors listed in Table 3-3. The tests shall verify each recovery option provided and shall demonstrate the SDEX/7 program's ability to recover from errors. Within the test, the SDEX/7 program shall properly isolate each error, pass the error to the module registered for processing the error, and subsequently resume processing as directed by the user module.

Section 5

NOTES

5.1 INTRODUCTION

The purpose of the SDEX/7 computer program is to provide a fundamental set of executive functions for computer programs operating in the AN/UYK-7 computer. The following paragraphs present the design rationale followed in developing the performance requirements for SDEX/7 and discuss the capabilities and constraints of SDEX/7 on a fundamental basis.

5.2 DESIGN RATIONALE

The design of the SDEX/7 functions is based upon the requirement for SDEX/7 to be functionally independent of the program (modules) using the executive functions. In order to achieve this functional independence it is necessary to define a common interface between SDEX/7 and all modules. Every module must conform to this interface regardless of the module's functions. There are no exceptions.

The common module interface defines the rules for functional interaction between SDEX/7 and modules, and includes the physical data areas supplied by each module in order to use the executive functions. The following paragraphs define the functional and physical portions of the common interface.

5.2.1 FUNCTIONAL INTERFACE

The functional interface consists of the method by which modules communicate processing requirements to SDEX/7 and the manner in which SDEX/7 implements this processing. The functional interface is illustrated by Figure 5-1. In order to begin processing, all modules must first register their processing requirements using SDEX/7 ESRs (usually during initialization, but registration can be done at any time). The processing requirements may include the following:

a. Task scheduling requirements and criteria.

b. Interrupt processing responsibilities.

c. Error processing responsibilities.

The registration of scheduling requirements consists of each module registering each of its scheduled task entrances for processing using the provided registration ESRs. Message tasks are an exception and do not require registration because all modules are assumed to have a valid message task for processing at least the initialization message.

The registration of interrupt processing responsibilities consists of each module registering any time-critical, I/O channel monitor and breakpoint interrupt processing responsibilities (if any) using the registration ESRs. With these ESRs the modules define their responsibilities and specify the type of processing to be performed when the interrupt occurs.

The registration of error processing responsibilities consists of each module registering to process errors should the error occur. The registration is done such that modules can register to process only selected errors or a single module may register to process all errors.

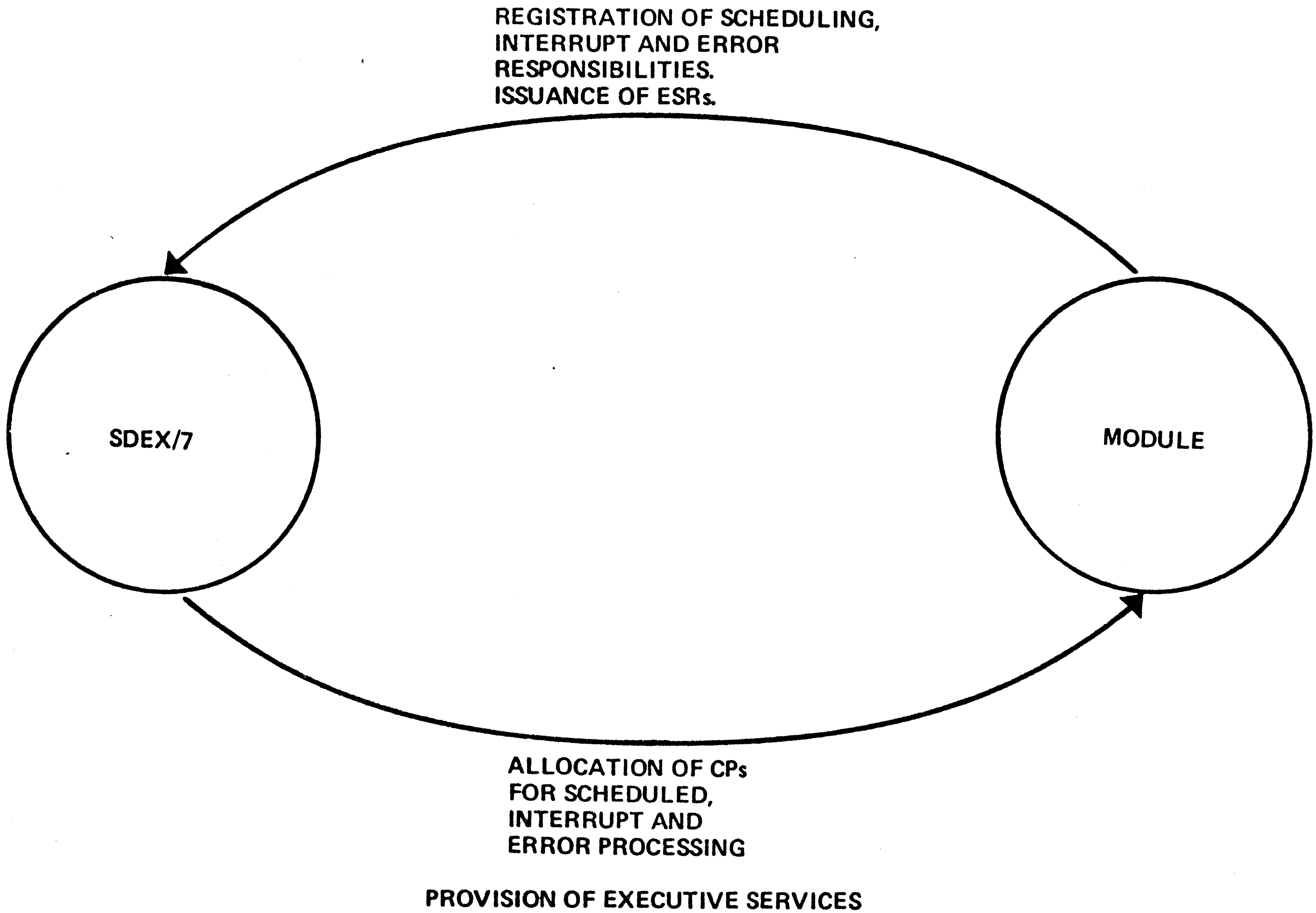


FIGURE 5-1. FUNCTIONAL INTERFACE BETWEEN SDEX/7 AND MODULES

Using the above registrations, the modules control SDEX/7 functional processing. From the set of registered module tasks for scheduling, the scheduling function allocates CP processing time to modules according to the scheduling criteria for the type of task. When a time-critical, I/O monitor or breakpoint interrupt occurs, the interrupt management function handles the interrupt according to the registration of the module responsible for processing the interrupt. When an error occurs, the error management function formulates a packet of error information and passes it to the module which registered to process the error. In this manner, the processing within the computer is controlled by the executive functions according to the registration of the module for processing.

5.2.2 PHYSICAL INTERFACE

The physical interface consists of the data tables supplied by each module in order to operate with SDEX/7. Figure 5-2 illustrates the physical interface required for each module. These are the Segment Allocation Packet (SAP) and the preamble. In addition, there are secondary components required only when the module uses certain features of SDEX/7. These are special storage areas and packing areas for local messages.

The SAP is a variable length table consisting of a set of items, where each item contains the absolute base address, displacement, and memory protection attributes for an addressable

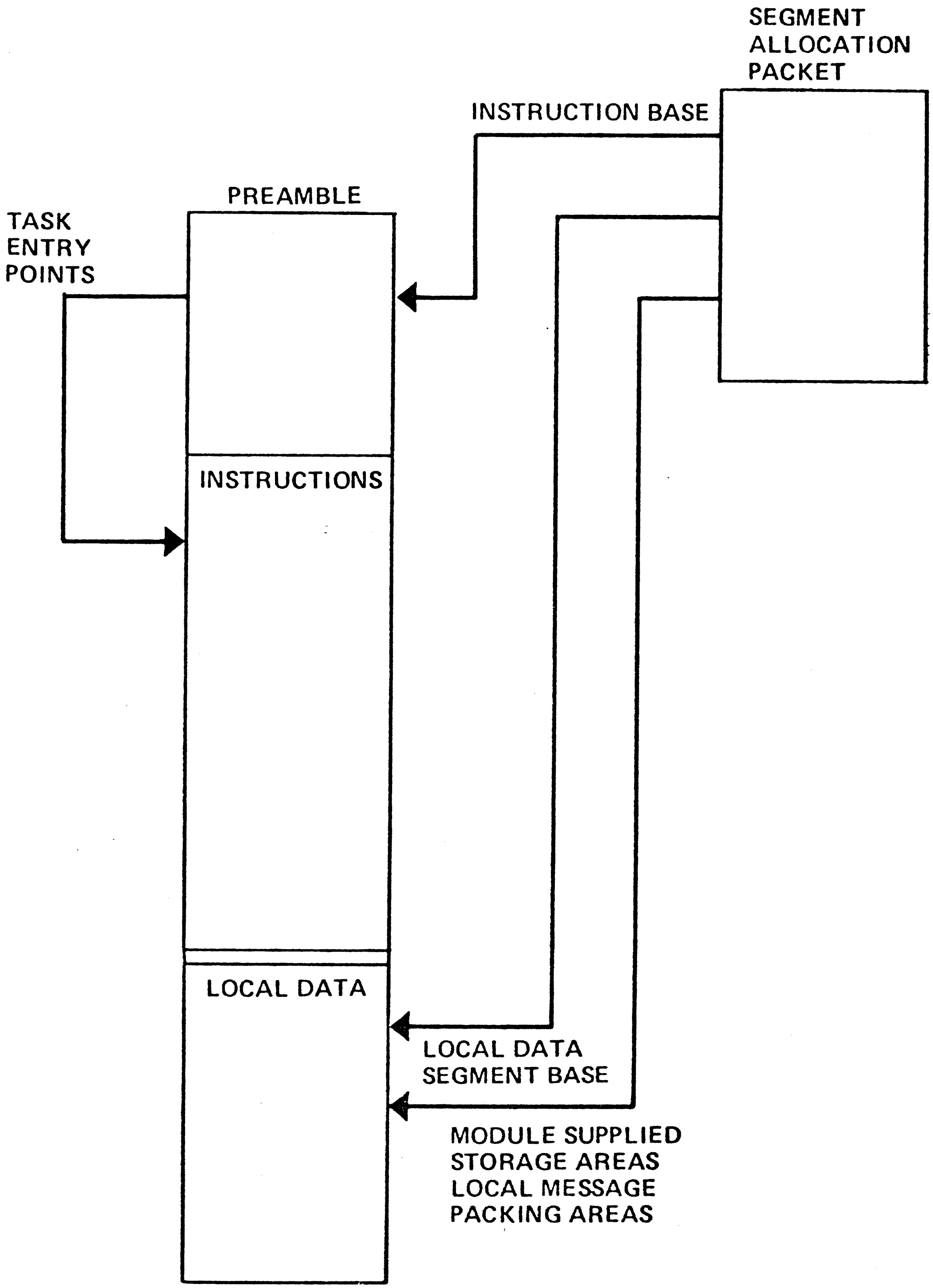


FIGURE 5-2. MODULE PHYSICAL INTERFACE WITH SDEX/7

segment of the module. Each item consists of two words formatted such that they can be directly loaded into a base register and SPR respectively. Using the SAP items, addressable segments can be quickly referenced. SAP items consist of items dedicated for specific SDEX/7 use and items defined for module use (transient use). The dedicated items include the base and displacement for the module's SAP, instruction segments, and local data segments. The transient items are available for module use as necessary. For example, local message packing areas and save areas for suspendable task must be defined by separate SAP items. See paragraph 3.2.4.2 for definition of the dedicated and transient use of the SAP items.

The secondary storage areas are supplied by the module only when specific features of SDEX/7 are used. These areas consist of storage areas for the task environment if certain module tasks are registered as suspendable, and local message packing areas if local messages are issued by the module.

The preamble table consists of data identifying the module and its included tasks. It contains the module name, number, priority number and sy addresses
to each scheduled module task or interrupt processing task, as well as storage for module history if required.

5.3 FUNCTIONAL CONSIDERATIONS

The following paragraphs define each executive function in terms of use, capability and constraints.

5.3.1 INITIALIZATION FUNCTION

The initialization function is defined as a separate function intended to control all operations during computer loading and program startup. The initialization function provides initialization of the CP and SDEX/7 (after loading of SDEX/7 by the AN/UYK-7 NDRO bootstrap), a check of available memory, the loading of an initial set of user modules, and the passing of an initialization message to each module for local module initialization.

The initial set of user modules to be loaded is defined at SDEX/7 compile-time. Using this feature, individual applications can define system loading and initialization to meet the requirements of the system. For example, an application in which all modules are core-resident at all times would require one-time program loading with no dynamic reconfiguration or operator direction. This application would define all modules as the initial configuration and would not require a loader capability other than that provided by SDEX/7.

Another application may consist of a large set of modules of which only a subset are core-resident at any one time. Further, the system may require dynamic reconfiguration while the system is operating. For this application, the user could define a Dynamic Loader Module and a System Control Module as the modules to be initially loaded and initialized by SDEX/7. Once loaded and initialized these two modules could be used to select the dynamic configuration for system operation. Using the features of SDEX/7 initialization, the system initialization can be as simple or complex as the application requires. An additional feature is provided by SDEX/7 in the form of the Return SAP Data ESR. After all modules are loaded, any module may execute this ESR and will receive, from SDEX/7 the contents of the SAP for each module entered on the module list. This data can then be used to formulate a printout or display of a memory map showing the final memory allocation for each module.

5.3.2 SCHEDULING FUNCTION

The scheduling function uses a multi-tier scheduling philosophy to achieve a high degree of flexibility for applications using SDEX/7. The multi-tier scheduling philosophy is one where all module tasks are categorized by type (each type corresponds to a tier in the scheduling function) such that a different scheduling criteria can be applied in each tier. Further, the scheduling function is organized such that there is a priority among task types. In this manner, all pending tasks of a priority type may be honored only after all tasks of higher priority types are honored and before any pending tasks of lower priority types. The module task types for the scheduling function are as follows:

a. successor - asynchronous, one-shot tasks
b. message - asynchronous message processing tasks
c. time-dependent - synchronous tasks executed periodically after a specified time interval has passed.

d. background - synchronous tasks executed periodically and whose processing can be automatically time-sliced by SDEX/7.

The scheduling criteria used in each tier is further discussed in paragraphs 5.3.2.1 thru 5.3.2.4.

The individual scheduling tiers of the scheduling function are compile-time selectable by the user for his specific application. This allows the user to select any or all of the scheduling tiers at SDEX/7 compile-time. When a particular tier is not selected, the SDEX/7 instructions, data and ESRs pertaining to that tier are omitted in the resulting SDEX/7 program. This means that the user does not pay in time and memory overhead for an executive capability that he does not use.

The priority of the scheduling tiers is also selectable at SDEX/7 compile-time. This feature allows the user to define the tier priorities within the scheduling function for his application. For example, one application might select the scheduling tiers with successor scheduling in the highest priority tier followed respectively by message, time-dependent, and background task scheduling in lower priority tiers. Another application having critical data exchange between modules may define the scheduling tier priorities as message, time-dependent, successor, and background task scheduling respectively.

The selectability of the scheduling tiers and the respective priority ordering allows the user to configure the scheduling function to suit his particular application. The following paragraphs discuss each of the defined task types from the standpoint of module registration, scheduling criteria and associated run-time options available when using a particular task type in an application.

The scheduling function is designed to prevent concurrent execution of a scheduled task by two or more CPs. This is done to eliminate the need for user modules to be coded reentrantly. However, in some applications reentrant module implementation may be a requirement. To accomplish this capability under SDEX/7 the user module must be designed with at least two SAP tables and two preambles each containing a different module number and identical task entrances. By registering the module twice using the different preambles, the module appears to SDEX/7 as two different modules to which it can allocate CP resources. This method thus allows reentrance user modules to be utilized without forcing the requirments of reentrancy on all user modules.

5.3.2.1 SUCCESSOR SCHEDULING

Modules requiring the scheduling of a successor task must first register the task using the Register Successor Entrance ESR. When this ESR is received by SDEX/7 the specified module is assigned a successor indicator in the successor scheduling list based upon the module priority found in its preamble. A module so registered is eligible to have the successor task scheduled. The assigned successor indicator can be set in the following ways:

a. by use of the Request Successor Task ESR issued by the registered module or any other module.

b. automatically by SDEX/7 in response to a time-critical interrupt if the module has registered for a time-critical interrupt and has specified the requesting of the successor task whenever the interrupt occurs (see paragraph 5.3.3.2).

c. automatically by SDEX/7 in response to an I/O monitor interrupt if the module has registered for I/O channel responsibility and has specified the requesing of the successor task whenever the I/O monitor interrupt occur (see paragraph 5.3.3.1).

The scheduling criteria used within the successor tier is a priority search of the successor scheduling list looking for the highest priority module whose assigned successor indicator has been set. Upon locating the highest priority requested successor task that can be executed (see paragraph 5.4.1 and 5.4.2) the successor indicator is cleared and the module is passed CP control at its successor entrance for task processing. Upon completion of the processing, the module must execute the Module Exit ESR to return CP control to SDEX/7 for further processing.

When registering a successor task, it may be specified as suspendable. This run-time option allows the user to define tasks requiring excessive amounts of processing time as successor tasks to take advantage of the priority scheduling criteria and have the task time-sliced (suspended) by SDEX/7 to periodically return CP control to the scheduling function to process higher priority tasks. In order to utilize this feature, a successor task save area must be supplied by the module (a SAP item is dedicated as a pointer to the area). This save area provides storage for the task environment should the task be time-sliced (suspended) and also includes the time-slice value indicating the amount of time to be allotted to the task before time-slicing.

When a suspendable successor task is passed CP control, the CP monitor clock is loaded with the time-slice value. If the CP monitor clock interrupt is received signifying the end of the time-slice, the task state environment will be saved in the save area, and the module's successor indicator will again be set by SDEX/7 before returning to the scheduling function. When the task is again located by the scheduling function, the saved task environment is restored, the CP monitor clock is set for another time-slice and CP control is passed to the point where the task has been suspended by the CP monitor clock interrupt. This process will continue until the task processing is completed and the Module Exit ESR is executed by the module.

For applications using successor scheduling where a successor task can be requested in more than one way, caution must be exercised because successor tasks are considered to be one-shot tasks receiving CP control for processing only when requested. When the indicator for a specific successor task has been set, additional requests for the task will have no effect. The scheduling function will honor the task only once regardless of the number of requests since the last time the task received CP control. In cases where a successor task is suspended (the successor indicator is set) and a request for another execution of the task is received, the task will still be resumed at its suspended point. In applications where this case is a possibility the user must design a method of queuing the additional requests until the previous execution of the successor task is complete.

5.3.2.2 MESSAGE SCHEDULING

Modules which receive and process messages do not require specific registration because each module is required to provide a message task for processing the initialization message.

When a message is initiated by a module to another module, the receiving module's index (assigned when the module is entered on the module list) and the location of the message is placed in a first-in/first-out (FIFO) message scheduling list. The scheduling criteria used for passing the messages to the receiving modules is based on the FIFO flow of messages such that messages are processed in the order in which they are sent.

Modules sending a message may use either a system message packet or a packing area within the sending module's local data area. When a message is sent it is handled the same in both cases by the scheduling function. Further, both types of messages follow the same format consisting of a message control word followed by message text words. The difference between the two types of messages, i.e., system or local, is the packing area used. The use of the two types of messages is discussed in the following paragraphs.

5.3.2.2.1 SYSTEM MESSAGES

An area of memory is set aside during SDEX/7 initialization for system message packing. The total amount of memory for system message packing, the ratio of areas for different packet sizes and the packet sizes are SDEX/7 compile-time parameters. Figure 5-3 illustrates the system message packing

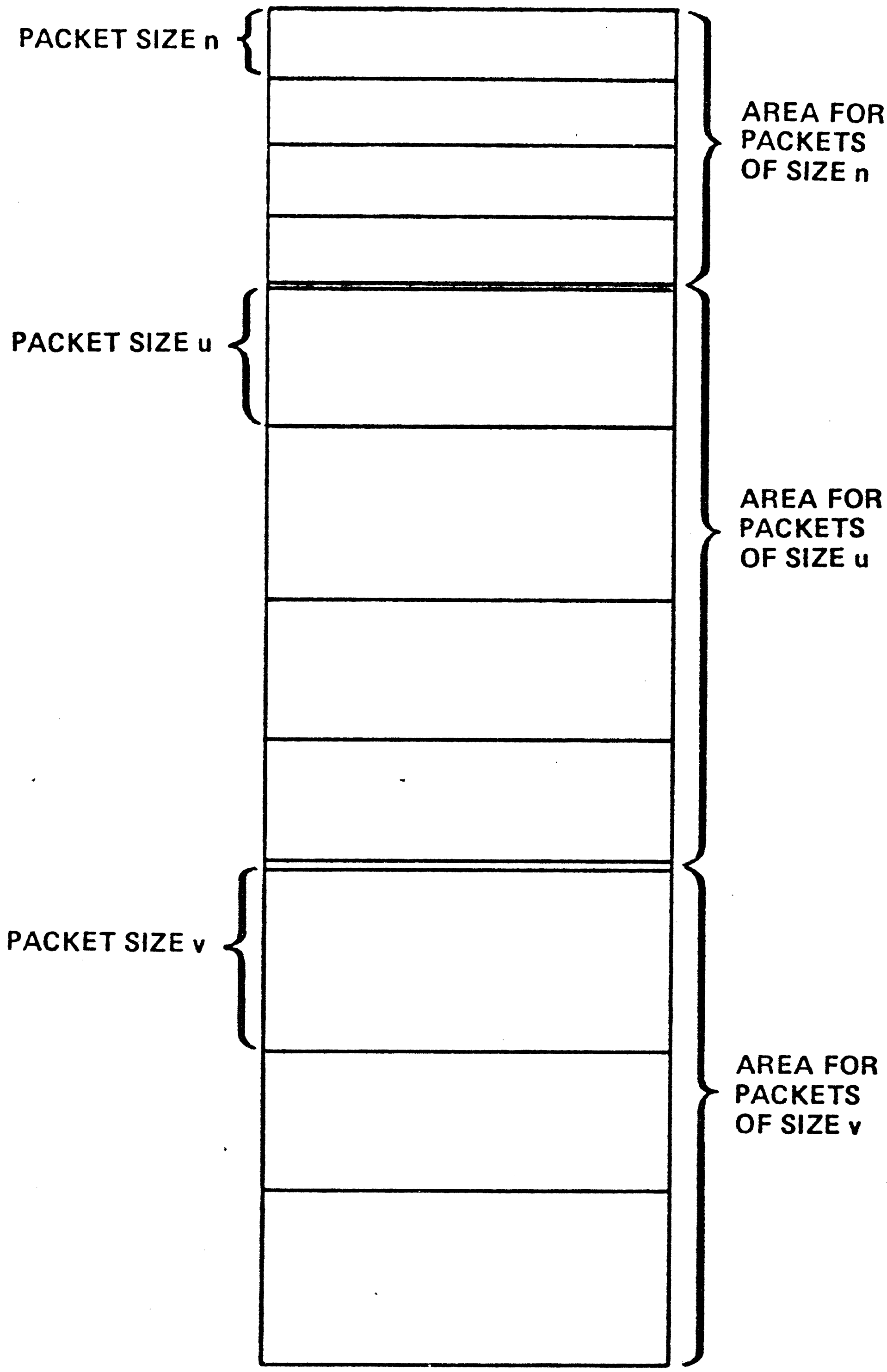


FIGURE 5-3. SYSTEM MESSAGE PACKING AREAS

area. During initialization, the area is initialized to contain packets of the sizes specified by the SDEX/7 compile-time parameters.

When a module requires the use of a system packet it must request a packet using the Request System Message Packet ESR. Upon receipt of this ESR, SDEX/7 will allocate a packet of the specified size, or a larger size if the specified size is not available, to the module by placing the address of the assigned packet in a task base register specified by the requesting module. The corresponding SPR is loaded to allow read and write access to the assigned packet.

Once a module is assigned a system packet, the packet may be used to pack a message. The message can then be sent to up to four modules with a single execution of the Send System Message ESR. This ESR can be repeated until the message is sent to all desired modules. Each time the message is sent, SDEX/7 increments the receiving module count field in the message control word, and places a pointer to the message and the receiving module index in the message task scheduling list.

Each time the scheduling function passes a message to a receiving module for processing and processing is completed, the receiving module count in the message control word is decremented by 1. When the count reaches zero, the system packet is reclaimed by SDEX/7 and made available for another assignment.

System packets can be used for message packing by any module task without restriction. However, for each separate message the module wishes to send, a separate system packet must be requested. If a module requests a system packet and then does not use the packet to send a message, the packet will automatically be reclaimed by SDEX/7 when the module executes the Module Exit ESR.

System packets may be used for purposes other than sending messages. For example, a system packet could be used as a temporary scratch pad area for a module. When the module receives CP control for processing, it could request a system packet for use as a scratch pad area until its processing was complete. The restriction on this type of use is that the quarter word of the message control word used for the receiving module count by SDEX/7 must be zero when the module exits.

5.3.2.2.2 LOCAL MESSAGES

Modules may use areas in their local data segments for message packing. In order to utilize this feature, the module must include a SAP item for each local packing area it intends to use. The SAP index items must contain the base addresses of the packing areas and SPR quantities which allow only read access to the area. (Because the packing areas are in the local data segment, the sending module has read and write access to the areas via S1/SPR1).

Local messages are sent by a module executing the Send Local message ESR. Using this ESR the message can be sent to up to four modules with a single request. This ESR can be repeated until the message is sent to all desired modules. Upon receipt of the ESR, SDEX/7 places a pointer (SAP index) and the receiving module index(es) in the message task scheduling list and increments the receiving module count in the message control word.

Each time the scheduling function passes the message to a receiving module for processing and processing is completed, the receiving module count is decremented by 1. It is the sending module's responsibility to monitor the receiving module count field in order to determine when the local packing area can be reused, i.e., the receiving module count reaches zero.

The local message feature can be used for processing other than the exchange of messages. For example, module data or instruction areas can be linked between two or more modules. This is accomplished by a module defining a SAP item containing the base address and SPR value for the area to be shared. The module can send this area to other modules as a local message. When the local message is received by other modules, the base address and SPR values are in a transient task base register/SPR pair. The receiving module can then execute the Store Transient Base Register ESR causing the base register and SPR values to be stored in its own SAP. Any time thereafter the module can cause a transient base register to be loaded from that SAP item to accomplish linkage to the shared area.

5.3.2.3 TIME-DEPENDENT SCHEDULING

Modules requiring the scheduling of a time-dependent task must first register the task using the Register Time-Dependent Entrance ESR. When this ESR is received by SDEX/7, the specified first time-to-initiate-execution (TE), the time interval between executions and the specified module's module index is saved in the time-dependent task scheduling list. The specified module is then available for time-dependent scheduling.

The scheduling criteria used within the time-dependent tier is a round-robin search for a TE less than or equal to the current value of the IOC real time clock. When the condition is met, a pointer to the next item in the list is saved and will be used as the starting point for the next search of the time-dependent task list. If the task can be executed (see paragraph 5.4.1 and 5.4.2), the next TE for the module will be computed by adding the time-interval parameters for the task to the current value of the IOC real time clock. The resulting value is stored as the next TE for the task in the time-dependent task scheduling list. The module is then passed to CP control for processing. When processing is complete, the module must execute the Module Exit ESR to return CP control to SDEX/7 for further processing.

When registering a time-dependent task, it may be specified as suspendable. This run-time option allows the user to define tasks that require excessive amounts of processing time as time-dependent tasks to take advantage of the time based scheduling criteria which allows repeated executions of the task. In order to utilize this feature, a time-dependent save area must be supplied by the module (a SAP item is dedicated as a pointer to this area). This save area provides storage for the task environment should the task be time-sliced and also includes the time-slice value indicating the amount of time to be allotted to the task before time slicing.

When a suspendable time-dependent task is passed CP control, the CP monitor clock is loaded with the time-slice value. If the CP monitor clock interrupt occurs signifying the end of the time-slice, the task state environment will be saved in the module's time-dependent save area. The next TE for the task is then reset to the current value of the IOC real time clock and CP control is passed to the scheduling function. When the task is again located for execution, the saved task state environment is restored, and the CP monitor clock is set with the time-slice value. The next TE is computed by adding the time-interval to the IOC real time clock value and stored in the time-dependent task scheduling list. CP control is then passed to the point where the task was previously suspended. This process will continue until the task is completed and the Module Exit ESR is executed by the module.

5.3.2.4 BACKGROUND SCHEDULING

Modules requiring the scheduling of a background task must first register the task using the Register Background Entrance ESR. When this ESR is received by SDEX/7, the specified first TE and the specified module index shall be placed on the background task scheduling list. The specified module is then available for background scheduling.

The scheduling criteria used within the background tier is similar to that defined for the time-dependent tier. The major difference between the two scheduling tiers is that the background task is always suspendable and the registration of the task always requires module provision of a background save area, while suspendability of a time-dependent task is a run-time option.

A second difference between the two tiers is in the computation of TE values. When a background task is time-sliced, its next TE is computed by adding the user-specified interval-between-slices to the current value of the IOC real time clock, while the corresponding TE for a time-dependent task would be set equal to the current value of the IOC real time clock. The significance of this difference is that the interval-between-slices for the background tasks can be used to distribute the background processing load across time to ensure that ample portions of time, if available, are shared among a set of background tasks.

A convenient method of breaking up tasks that require excessive amounts of processing time is to employ the current time-slicing suspension feature of SDEX/7 to periodically return to the scheduling function to process any higher priority pending tasks. The capability is also provided which allows the user to define tasks that can be arbitrarily suspended based upon current task processing. For example, if a task requires data to be input or output to a peripheral device before continuing its processing, there is a convenient method currently under SDEX/7 to suspend the task and resume processing of the task when the I/O operation is completed. In general, SDEX/7 provides the capability where a scheduled task can be suspended to process other scheduled tasks and later resumed at the point of suspension.

This capability is achieved by utilizing the Suspend Task ESR within SDEX/7. This ESR is valid only for background tasks and successor and time-dependent tasks which are registered as suspendable. Upon receipt of this ESR, the current task environment saved is the module-supplied save area, the requesting task is rescheduled on an optional basis for processing, and CP control is returned to the scheduling function. After processing any pending tasks of higher priority, the suspended task can then be resumed. The Suspend Task ESR includes the following input parameter:

a. Reschedule Indicator - indicates the rescheduling criteria as follows:

(1) Zero Value - If the requesting task is a successor task, suspend the task and do not set the corresponding success or indicator in the successor scheduling list. If the requesting task is a time-dependent or background task, suspend the task and reschedule the task using the current value of the IOC real time clock.

(2) None-Zero Value - If the requesting task is a successor task, suspend the task and set the corresponding successor indicator. If the requesting task is a time-dependent or background task, suspend the task and reschedule the task at the current value of the IOC real time clock plus the provided non-zero value.

Using the above capability a user module could register its successor task as suspendable and register I/O interrupt responsibility for an I/O channel using the schedule successor option provided by SDEX/7. In the course of processing the successor task an I/O operation could be initiated followed by a Suspend Task ESR with a reschedule indicator of zero. This would cause the task to be suspended and not rescheduled. When the requested I/O operation is completed and the I/O interrupt occurs, the task would be rescheduled and the task processing would be resumed at the point of suspension according to the scheduling criteria for successor tasks. The time during which the I/O operation was taking place would have been used for other task processing.

Similar utility is achieved by using the Suspend ESR for time-dependent and background task suspension whenever the task processing results in the scheduling of higher priority tasks. In this manner, the time-dependent or background task could be suspended to process successor or message tasks and resumed when the higher priority tasks have been completed.

5.3.3 INTERRUPT MANAGEMENT FUNCTION

The purpose of the interrupt management function is to initially decode all interrupts and determine the appropriate action to be taken in processing the interrupts. The set of AN/UYK-7 interrupts can be characterized as one of two types: error conditions, or the occurrence of an event for processing. The following paragraphs discuss the handling of the four classes of AN/UYK-7 interrupts and highlight the processing features of SDEX/7.

5.3.3.1 CLASS I INTERRUPTS

The AN/UYK-7 Class I interrupts all signify some sort of hardware failure in the computer which must be handled as an error by the error management function. When a Class I interrupt occurs the error is indicated by passing the associated interrupt status code to the error management function for handling (see paragraph 5.3.4). The memory resume interrupt is an exception to this general rule. If a memory resume interrupt occurs signifying the failure of a memory bank(s) containing SDEX/7 and the executing CP has the highest priority memory access, a reload indicator for all CPs is set and the executing CP is 7-stopped or SDEX/7 is reloaded depending upon the user's option selection. If the executing CP is not the highest priority CP and the reload indicator is set the executing CP is 7 stopped or SDEX/7 is reloaded depending upon other user's option selection. Otherwise, the executing CP will ignore the interrupt and return to the point of interrupt to continue processing.

5.3.3.2 CLASS II INTERRUPTS

The Class II interrupts are all software related. In general, Class II interrupts that signify errors are identified and passed to the error management function for handling. The following Class II interrupts are handled as special cases:

a. Floating Point Error - this interrupt is saved when it occurs and CP control is returned to the point of interrupt to continue processing. If the executing module subsequently executes the Return Floating Point Error ESR, Class II interrupt status code and the address of the instruction causing the error are passed to the requesting module.

b. Breakpoint Interrupts - when this interrupt occurs the task state environment and the Class II Designator Storage words are passed to the module registered for processing breakpoint interrupts.

c. CP Monitor Clock Interrupt - this interrupt signifies that the module task currenly being executed has exceeded its allotted run time. If the task is a suspendable successor or time-dependent task or a background task, the task state environment is saved, the task is rescheduled for another time-slice and CP control is passed to the scheduling function. Otherwise, a module overrun error is indicated and passed to the error managment function.

5.3.3.3 CLASS III INTERRUPTS

The Class III interrupts are I/O related and fall into three processing categories: 1) IOC errors, 2) I/O channel monitor interrupts, and 3) IOC monitor clock interrupts (time-critical interrupts). The Class III interrupts signifying errors are always passed to the error management function for handling. The processing of I/O channel monitor and time-critical interrupts are discussed in the following paragraphs.

5.3.3.3.1 I/O CHANNEL MONITOR INTERRUPTS

Modules requiring CP control for processing monitor interrupts on a particular I/O channel must register this processing responsibility using the Register I/O Interrupt Entrance ESR. The I/O channel monitor interrupts can be one of four types:

a. external interrupt

b. external function

c. input

d. output

A module can register for any or all of the above interrupt types on any or all I/O channels as long as no other module is registered for the same channel and interrupt type.

For each interrupt type on each channel the module can register one of two response options for processing should the interrupt occur. These options are as follows:

a. pass CP control and the interrupt status code to the module immediately when the interrupt occurs.

b. set the module's successor indicator and save the interrupt status code and the value of the IOC real time clock in the module-supplied I/O storage area when the interrupt occurs.

When an I/O channel interrupt occurs, the interrupt management function checks channel responsibility registration. If no module is registered to process the interrupt, CP control is passed to the error management function for handling of the interrupt as an error. If a module is registered for the interrupt, the registered processing option shall be checked. If the module registered to receive immediate CP control to process the interrupt, the current task environment is saved and CP control is passed to the registered module with Class III lockout set, and with the Class III interrupt status code in a task state accumulator. When the module completes processing and executes the Module Exit ESR, the interrupted task state environment is restored and CP control is returned to the point at which the interrupt occurred.

If the module registered to have its successor indicator set, the assigned indicator is set and the Class III interrupt status code and IOC real time clock value are stored in the module's I/O storage area. CP control is then returned to the point of interrupt.

There are two cautions to be observed for I/O interrupt processing by a module. First, if a module is registered for immediate processing of an I/O monitor interrupt, the amount of processing time should be limited, by design, to less than a millisecond because Class III interrupts are locked out during module processing. Second, if a module registers to have its successor indicator set, the module's successor task must have been previously registered and the I/O storage area must be sufficiently large to accomodate the queued I/O interrupts.

5.3.3.3.2 TIME-CRITICAL INTERRUPTS

The IOC monitor clock is used to generate time-critical interrupts for modules having tasks that require fixed intervals between subsequent executions. In order to utilize this feature, modules must register for time-critical interrupts using the Register Time-Critical Entrance ESR. Using this ESR the module specifies:

a. the first time a time-critical interrupt is to occur.

b. the interval to be maintained between interrupts. If the interval is zero, the time-critical interrupt will occur only once.

c. the processing options to be followed when the interrupt occurs as follows:

   (1) set the responsible module's successor indicator

   (2) initiate the specified I/O chain

   (3) pass CP control immediately to the module's time-critical task

   (4) any combination of the above

d. The absolute address of the I/O chain if option (2) above is specified.

When modules register for time-critical interrupts the first time-to-initiate-execution (TE) is placed on a time-critical interrupt list. This list is maintained by SDEX/7 such that it is time ordered, i.e., the TEs are arranged beginning with the smallest at the top sequentially downward to the largest. The difference between the smallest TE and the IOC real time clock is always used as the count down value for the IOC monitor clock to generate the interrupt at the appropriate time. If the smallest TE is less than or equal to the IOC real time clock, a zero value is placed in the clock to cause the occurrence of an immediate interrupt.

When a time-critical interrupt occurs, the next TE for the interrupt is computed by adding the specified interval between interrupts to the current TE and placing the resulting TE back in the time critical list. The IOC monitor clock is then reloaded, as discussed above, for the next interrupt. If it is loaded with zero the interrupt does not affect the executing CP because Class III interrupts are locked out due to the current processing of the IOC monitor clock interrupt. In this case the resulting interrupt remains queued until another CP answers the interrupt or until Class III lockout is released in the executing CP.

Following the reloading of the IOC monitor clock, the registered options are performed. If the module did not register for immediate CP control, the CP is returned to the point of interrupt following the completion of the options.

If the module registered for immediate CP control, the task state environment is saved and CP control is passed to the registered module's time-critical task with Class III interrupts locked out. When the module executes the Module Exit ESR, the saved task state environment is restored and CP control is returned to the point where processing was interrupted.

The options provided for time-critical interrupt processing offer some significant capabilities. First, the user can cause an I/O operation to be automatically initiated by SDEX/7 on a strict time interval basis without incurring the executive overhead of scheduling a module task. Second, the option to schedule a successor task allows periodic processing to be performed without slippage because the tardiness of any execution is corrected each time the time-critical interrupt occurs. Third, the option for immediate CP control allows strict time-interval critical processing to be performed.

There are two cautions to be observed when using the time-critical feature. First, immediate time-critical task processing time should be held, by design, to less than one millisecond because Class III interrupts are locked out during this time. Second, if the set successor indicator option is used, the module's successor task must have been registered prior to the time-critical registration.

5.3.3.4 CLASS IV INTERRUPTS

The Class IV interrupts are caused by modules executing the "Enter Executive State" instruction to request an ESR.

When a Class IV interrupt is received, the interrupt management function validates the interrupt status code as a proper ESR request. If it is not a proper ESR request, it is passed to the error management function as an illegal ESR. Otherwise, CP control is passed to the appropriate executive service routine.

In the processing of an ESR, certain validity checks are performed. If a validity check shows a condition where the ESR cannot be completed, a status indication (negative flag in a task accumulator) is set and CP control is returned to the requesting module. It is the module's responsibility to monitor this ESR status. In other instances certain executive conditions such as a scheduling list overflow, prevent the ESR from being completed. These are considered error conditions and passed to the error management function for handling.

When a requested ESR process has been completed, CP control is returned to the requesting module at the point where the Class IV interrupt occurred. In this case, the status indicator is cleared indicating completion of the ESR. The ESRs requesting module exits are exceptions and result in CP control being passed to the scheduling function.

5.3.4 INPUT/OUTPUT MANAGEMENT FUNCTION

The purpose of the input/output management function is to provide executive supervision of I/O channel untilization. This supervision consists of providing for I/O channel interrupt registration and providing ESRs for enabling and disabling I/O interrupts on selected channels, as well as for the initiation of module-supplied I/O chains.

Modules to be responsible for processing I/O channel monitor interrupts must register their responsibility as discussed in paragraph 5.3.3.3.1. Once registered for at least one interrupt type on a channel, the module can control the enabling and disabling of interrupts on that channel using the provided ESRs.

In applications using multiple CP configurations, the enabling and disabling of I/O channel interrupts can be selective among CPs through module specification using the Enable and Disable I/O interrupt ESRs. This feature allows the module to enable I/O interrupts for the channel(s) in certain CPs and disable I/O interrupts for the channels in other CPs. In this manner, the user can arbitrarily distribute I/O interrupt processing among the CPs in the configuration.

Any module can request the initiation of a specified I/O chain any time it has CP control. No registration is required for this process. If a channel is being shared by more than one module, it is the user's responsibility to ensure valid operation of the I/O channel. This can be accomplished either by strict module design or by providing one module which performs all I/O operations for the application.

5.3.5 ERROR MANAGEMENT FUNCTION

The purpose of the error management function is to

provide handling of the defined set of errors. The error m na
ment function is designed to allow error processing to be performed as required for the application. This is accompl sh by allowing modules to register responsibility for processi any or all errors using the Register Error Responsibility ES

When an error occurs, the error management function che ks module registration for processing the error. If no module has registered for the error, the CP will be conditionally stopped (5-stop, if the key is set, for Class I errors, 6-st , if the key is set for all others). If the stop occurs, the error information is displayable from task state accumulators using the maintenance panel. The operator may then select one of the recovery options listed below and depress the start switch to resume processing. If the stop key is not set, the occurrence of an unregistered error will be ignored and processing will continue.

If an error occurs, and a module has registered for processing the error, the module is passed an error packet containing the information specified in paragraph 3.3.5.2. CP control is released to the module at its message entrance. The error can then be processed as appropriate for the application. Upon completion of the processing, CP control must be returned to SDEX/7 using the Error Exit ESR. Using this ESR the following processing options can be specified:

a. ignore the error and continue processing
b. set the erring module's successor indicator and return to the point at which the error occurred

c. return CP control to the scheduling function

Upon receipt of the Error Exit ESR, SDEX/7 will resume processing according to the error exit option.

The design of the error management function allows specific applications to define error recovery processing that is as sophisticated or as simple as required. The application may use a single error recovery module, distribute error recovery functions across several modules or elect not to utilize automatic error recovery at all and make it purely an operator function.

5.4 SPECIAL CONSIDERATIONS

The preceding discussions in this section have been devoted to the structure and design rationale used for SDEX/7, as well as a description or the use and limitations of the executive functions from a users standpoint. In addition, to these capabilities, there are two other features of SDEX/7 to be considered, when it is used in multiple CP/IOC configurations. These are CP dedication and module reentrancy.

5.4.1 CP DEDICATION

The SDEX/7 program for multiple CP/IOC configurations allows CPs to be selectively dedicated for the processing of scheduled module tasks using the Dedicate CP ESR. This feature allows the user to specify that the successor, message, time-dependent and background tasks of a module are only to be executed by certain CPs. In this manner CP processing loads can be distributed among the modules making up the application.

In order to implement this feature, the scheduling function is designed to check CP dedication for a task before releasing CP control to the task. For example, a CP is currently executing in the successor scheduling tier and has located a successor task for execution. If the executing CP is not allowed to execute the task because of CP dedication, the search of the successor task scheduling list is continued until the entire list has been searched. This same general philosophy is followed in each of the scheduling tiers.

It should be noted that CP dedication is not extended to interrupt processing tasks, i.e., time-critical or I/O interrupt tasks. These tasks, because of the premptive nature of the interrupts, are executed by whatever CP answers the interrupt. In configurations where there is not total memory sharing among CPs some caution must be exercised.

For example, immediate time-critical tasks must be allocated in a memory bank that is shared by all CPs. For immediate I/O interrupt tasks, the allocation is not important if the I/O channel interrupts are disabled such that a CP cannot respond to an I/O interrupt requiring processing by a task in a memory bank to which the CP does not have access.

5.4.2 MODULE REENTRANCY

The SDEX/7 program is design to inhibit simultaneous or recursive execution of scheduled tasks. This feature is implemented in the scheduling function much in the same manner as CP dedication. Prior to releasing CP control to a module task, a check is made to ensure that the module is not busy, i.e., another CP is not currently executing a scheduled task in the same module. If the module is busy, the scheduling search will continue. If not, the module is indicated as busy and CP control is passed to the task.

In the case of the message scheduling tier an additional consideration must be made because of the possibility of more than one message having been sent to a given module. In this case, it would be possible to check the first message and find that the receiving module was busy. In the checking of the next message (hypothetically for the same module), the task being executed in the module could be completed removing the busy condition. If the second message is allowed to be passed to the module it would break the FIFO flow of messages to the module. Because of this possibility, when a receiving module is found busy, during the course of message task scheduling, a hold flag is set for the duration of the message task list search. When a message task is located that can be executed or if the list is completely searched, all message hold flags for all modules are cleared.

It should be noted that reentrant protection is not extended to immediate interrupt tasks, i.e., time-critical or I/O interrupt tasks. For this reason, in multiple CP configurations immediate interrupt module tasks must be designed to accommodate simultaneous execution by multiple CPs.

5.5 SUMMARY

The SDEX/7 program has been designed to provide basic executive functions for a range of applications that use the AN/UYK-7(V) computer. Flexibility has been provided through the compile-time features and run-time options of the SDEX/7 functions. These features allow the SDEX/7 program to be specifically adapted to individual applications using the basic set of executive functions. The significant features are summarized as followed.

a. SDEX/7 is designed to operate in any AN/UYK-7(V) configuration.

b. SDEX/7 is functionally independent of the application in which it operates.

c. The initialization function allows the user to define system initialization processes specifically for his application.

d. The scheduling function uses a tiered philosophy that allows the user to configure the scheduling function based upon the specific appliation requirements.

e. The interrupt management function allows the application programs to assume responsibility for the processing of interrupts according to the application requirements.

f. The input/output management function supervises I/O channel utilization but allows the application to define how the program is to communicate with the external environment.

g. The error management function provides handling of errors such that the application determines the error recovery procedures according to the specific application requirements.

From the above features, it can be seen that SDEX/7 does not place specific requirements on the applications using it. Instead it provides the basic executive functions for the application program using the various features to allow the application to determine how the executive functions are used. This is the primary attribute of the SDEX/7 design.

APPENDIX A

GLOSSARY

ASR	Active Status Register in the AN/UYK-7 computer.
background entrance	the user module entrance from which processing begins when the scheduling function passes CP control to the module for background processing.
background job	the processing performed by a user module which begins at the background entrance and ends upon exit from the bakcground task.
bootstrap	see NDRO
compile-time parameter	a constant value set before the SDEX/7 is compiled.
CP dedicated data	see dedicated data
CP shared data	see shared data
dedicated data	data which pertains to and is accessed by one CP.
dedicated task state base registers	those base registers always set for the user modules by the scheduling function prior to any module entry. Task state base registers 0 and 1 are dedicated for each task module.
DSW	Designator Storage Words in the AN/UYK-7 computer control memory
entrance	the entry point to a user module task.
ESR	Executive Service Request which is an operation performed by the SDEX/7 in response to a Class IV interrupt.
executive tables	those data tables used by the SDEX/7 for maintaining logical control over its functions and processing.

initial condition words (ICW)	those control memory locations in an AN/UYK-7 to which control is passed upon the occurrence of any interrupt. There is one ICW for each class of interrupts.
initialization message	an intermodule message sent to each module during the initialization process.
initialization (SDEX/7)	the process of setting all flags, indicators, counts, and data areas to their start-up configurations. The initialization function includes sending of initialization messages to all modules in the system so that they may register their entrances with the scheduling function.
intermodule message	any data passed from one user module to another.
interrupt	a single, external or internal to the AN/UYK-7, which causes suspension of CP processing and the passing of CP control to the address contained in one of the Initial Condition Words.
interrupt entrance	the time-critical or I/O channel interrupt entrance of a user module.
interrupt state	the operating state of the AN/UYK-7 CP in which the privileged instruction set may be executed. The interrupt state is entered whenever an interrupt occurs in the AN/UYK-7.
interrupt status code (ISC)	a status code stored in specific locations by the AN/UYK-7 hardware which defines the reason for and condition of an interrupt.
IOC	Input/Output Controller of the AN/UYK-7.
I/O chain initiation request	a request made by a module at the time it registers its time-critical entrance. Upon occurrence of the timed interrupt, the SDEX/7 will automatically execute an Initiate I/O instruction and activate the module's I/O chain.

I/O interrupt entrance	the user module entrance which receives CP control when an I/O interrupt for which the user module has registered responsibility occurs.
list of modules	a SDEX/7 list containing the absolute address of each user module's SAP.
local data	that data which is referenced by a single subprogram or user module.
memory resume interrupt	the Class I interrupt which occurs when a CP or IOC attempts to access a memory location which does not acknowledge the request from the CP or IOC.
message	see intermodule message
message control word	the first word of each message containing the message type, number of words, and other message control data.
message entrance	the module entrance through which CP control is received for processing of a message.
message hold flag	a status flag set to inhibit message receipt by a user module when the user module cannot process its first message in the queue.
module history	a history of user module activity including the number of times entered, the total time spent within the module, and the longest time spent within the module.
Module List	see list of modules
NDRO	Non destructive read-out memory of the AN/UYK-7. It is hard-wired and contains the bootstrap loader and fault analysis routine.
preamble table, module	a table of data preceding each task module instruction segment which holds vital data concerning the user module and its entrances. It is used during module initialization and whenever the user module is entered.
privileged instruction	an AN/UYK-7 instruction which may be executed only when the CP is operating in the interrupt state.

SAP	segment allocation packet supplied by each user module which defines the base address and displacement for each separate addressing segment contained in the module.
scheduling	the process of allocating CP control among various user modules based upon priorities and an established scheduling algorithm.
shared data	data common to and accessed by all CP's in a system.
successor indicator	a bit in the Successor Indicator List which, when set, directs the scheduling function to give CP control to the user module's successor task.
successor indicator list	a table whose individual bits are associated with user module successor tasks.
task state	one of two operational conditions of the AN/UYK-7 hardware. It is the state in which all task processing is performed. See interrupt state.
time-critical entrance	the user module entrance through which CP control is given when a time-critical interrupt signals the start of processing required of the user module.
time-dependent entrance	the user module entrance from which processing begins when the scheduling function passes CP control to the user module for time dependent processing.
time slice	a length of time during which user module task processing may continue before being suspended in favor of other processing.
time-to-initiate-execution	a time at which a user module task becomes eligible for processing.
transient base register	a base register which is available for use by a user module but is not protected by SDEX/7.
user module	any task state module not provided as part of the SDEX/7.