LY28-1760-0
File No. S370-36

Program Product

MVS/Extended Architecture System Logic Library: System Resource Manager

MVS/System Product:

JES3 Version 2	**5665-291**
JES2 Version 2	**5740-XC6**

IBM

This publication supports MVS/System Product Version 2 Release 2.0, and contains information that was formerly presented in MVS/Extended Architecture System Logic Library Volume 12, LY28-1250-3, which applies to MVS/System Product Version 2 Release 1.7. See the Summary of Amendments for more information.

First Edition (June, 1987)

This edition applies to Version 2 Release 2.0 of MVS/System Product 5665-291 or 5740-XC6 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 Bibliography, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product in this publication is not intended to state or imply that only IBM's product may be used. Any functionally equivalent product may be used instead.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Department D58, Building 921-2, PO Box 390, Poughkeepsie, NY 12602. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

PREFACE

The MVS/Extended Architecture System Logic Library is intended for people who debug or modify the MVS control program. It describes the logic of most MVS control program functions that are performed after master scheduler initialization completes. For detailed information about the MVS control program prior to this point, refer to MVS/Extended Architecture System Initialization Logic. For general information about the MVS control program and the relationships among the components that make up the MVS control program, refer to the MVS/Extended Architecture Overview. To obtain the names of publications that describe some of the components not in the System Logic Library, refer to the section Corequisite Reading in the Master Preface in MVS/Extended Architecture System Logic Library: Master Table of Contents and Index.

HOW THE LIBRARY IS ORGANIZED

SET OF BOOKS

The System Logic Library consists of a set of books. Two of the books provide information that is relevant to the entire set of books:

1. The MVS/Extended Architecture System Logic Library: Master Table of Contents and Index contains the master preface, the master table of contents, and the master index for the other books in the set.

2. The MVS/Extended Architecture System Logic Library: Module Descriptions contains module descriptions for all of the modules in the components documented in the System Logic Library and an index.

Each of the other books (referred to as component books) in the set contains its own table of contents and index, and describes the logic of one of the components in the MVS control program.

ORGANIZATION OF THE COMPONENTS

Most component books contain information about one component in the MVS control program. However, some component books (such as System Logic Library: Initiator/Terminator) contain more than one component if the components are closely related, frequently referenced at the same time, and not so large that they require a book of their own.

A three or four character mnemonic is associated with each component book and is used in all diagram and page numbers in that book. For example, the mnemonic ASM is associated with the book MVS/Extended Architecture System Logic Library: Auxiliary Storage Management. All diagrams in this book are identified as Diagram ASM-n, and all pages as ASM-n, where n represents the specific diagram or page number. Whenever possible, the existing component acronym is used as the mnemonic for the component book. The Table of Book Titles in the Master Preface in MVS/Extended Architecture System Logic Library: Master Table of Contents and Index lists the book titles, the components included in each book (if a book contains more than one component), the mnemonics for the books, and the order number for each book.

HOW TO USE THE LIBRARY

To help you use this library efficiently, the following topics cover

- How to find information using book titles and the master index
- What types of information are provided for each component
- How to obtain further information about other books in the System Logic Library

FINDING INFORMATION USING THE BOOK TITLES

As you become familiar with the book titles, MVS component names and mnemonics, and the book contents, you will be able to use the System Logic Library as you would an encyclopedia and go directly to the book that you need. We recommend that you group the books in alphabetical order for easy reference, or, if you are familiar with MVS, that you to group the books by related functions.

The Table of Book Titles in the Master Preface in MVS/Extended Architecture System Logic Library: Master Table of Contents and Index contains a list of book titles and mnemonics. It provides a quick reference to all the books, and their corresponding components, in the System Logic Library.

FINDING INFORMATION USING THE MASTER INDEX

If you are not sure which book contains the information you are looking for, you can locate the book and the page on which the information appears by using the master index in System Logic Library: Master Table of Contents and Index. For the component books, the page number in an index entry consists of the mnemonic for the component and the page number; for System Logic Library: Module Descriptions, the page number consists of the mnemonic "MOD" and the page number.

For example:

ASM-12 refers to MVS/Extended Architecture System Logic Library: Auxiliary Storage Management, page ASM-12.

MOD-245 refers to MVS/Extended Architecture System Logic Library: Module Descriptions, page MOD-245.

INFORMATION PROVIDED FOR MOST COMPONENTS

The following information is provided for most of the components described in the System Logic Library.

1. An introduction that summarizes the component's function
2. Control block overview figures that show significant fields and the chaining structure of the component's control blocks
3. Process flow figures that show control flow between the component's object modules
4. Module information that describes the functional organization of a program. This information can be in the form of:
   - Method-of-Operation diagrams and extended descriptions.
   - Automatically-generated prose. The automated module information is generated from the module prologue and the code itself. It consists of three parts: module description, module operation summary, and diagnostic aids.

5. Module descriptions that describe the operation of the modules (the module descriptions are contained in System Logic Library: Module Descriptions)

Some component books also include diagnostic techniques information following the Introduction.

FURTHER INFORMATION

For more information about the System Logic Library, including the order numbers of the books in the System Logic Library, see the Master Preface in MVS/Extended Architecture System Logic Library: Master Table of Contents and Index.

CONTENTS

FIGURES

SUMMARY OF AMENDMENTS

**Summary of Amendments
for LY28-1760-0
for MVS/System Product Version 2 Release 2.0**

This publication is new for MVS System Product Version 2 Release 2.0. It contains information that was reorganized from the System Resource Manager (SRM) section in MVS/XA System Logic Library Volume 12 , LY28-1250-3, which applies to MVS/XA System Product Version 2 Release 1.7.

This publication contains changes to support MVS/System Product Version 2 Release 2.0. The changes include:

- Module Description for the following new module:

  IRARMCBS

- The following changed modules:

IRARMEVT	IRARMFIP
IRARMFPG	IRARMSRV
IRARMWLM	IRARMWLS

- Minor technical and editorial changes throughout the publication.

SRM — SYSTEM RESOURCES MANAGER

INTRODUCTION

The system resources manager (SRM) is a component of the base control program (BCP). The SRM is responsible for the efficient management of system resources and has two principal objectives:

- To distribute the use of the system's processor, I/O, and storage resources among individual users (address spaces) in a way that satisfies the installation's response time, turnaround time, and work priority requirements. (The installation specifies these goals in SYS1.PARMLIB member IEAIPSxx.)

- To optimize the use of resources by system users in terms of system throughput.

The SRM achieves its objectives primarily through swapping decisions and dispatching priorities, using a number of algorithms to compute values on which to base its judgements. Swapping is the process of moving address spaces into and out of real storage. Dispatching is the process of initiating the execution of an address space. By making decisions that maximize the number of address spaces in real storage and assigning priorities that maximize the rate of their execution, the SRM contributes a great deal to the ability of the system to do more work.

The process operates in the following way. When an address space is swapped out its entire working set (active user pages) moves to auxiliary storage or extended storage. The SRM uses the real page frames released by the swap for paging activity by other address spaces or for swapping in a swapped-out address space. When swapped in, an address space competes for system resources with all other swapped-in address spaces.

Note: Extended storage and expanded storage are terms that have the same meaning. This book uses the term extended storage.

To control this contention, the SRM assigns priorities to address spaces. These priorities may be fixed or governed by the mean-time-to-wait algorithm. An address space controlled by the mean-time-to-wait algorithm executes with a priority based on relationships between its use of processor and I/O resources. To provide more flexibility, an address space may be time-sliced between two dispatching priorities. The IPS defines the duration of a time slice.

Page replacement and device allocation algorithms in the SRM further support effective use of time and system resources. In the page replacement process, the SRM finds which real frames show less activity than other frames and release the less active frames for different work. In the device allocation process, the SRM selects which devices can be used to store tape and DASD data sets having non-specific volume requests.

The SRM obtains control as a result of the execution of a SYSEVENT macro instruction. The SYSEVENT macro instruction, signalling the occurrence of a user of system-related event, serves as an extended routing function based on the SYSEVENT code generated from the specified operand. Each SYSEVENT code represents a logically distinct interface to the SRM, with its own circumstances, its own input and output conventions, and its own resultant SRM actions.

The SYSEVENT processing function receives control from the interface function to perform processing related to the SYSEVENT. Depending on the SYSEVENT type, either IRARMINT or IRARMFIP obtains control. IRARMINT is a generalized routine that passes control to IRARMEVT, which contains a separate processing routine for each SYSEVENT routed through IRARMINT. IRARMEVT

routes a device allocation SYSEVENT to IRARMEV2, an expansion module of IRARMEVT. IRARMFIP contains separate entry points for SYSEVENTs whose execution time is critical.

The SRM maintains the information needed for its functions primarily in the resource manager control table (RMCT), which is contained in IRARMCNS and located by a pointer in the CVT (CVTOPCTP). The SRM keeps track of user (address space) information in user control blocks (OUCBs), user extension blocks (OUXBs), and user swap blocks (OUSBs).

SRM control is the dispatcher of the SRM, residing in the module IRARMCTL. Along with various other SRM routines, particularly IRARMSWP, it schedules and processes requests for actions and algorithms related to swap analysis and swap processing.

A set of resource-use algorithms manages the use of system-wide resources. Designed to improve overall system resource use, these algorithms calculate values that promote effective storage management, I/O management, and processor management. The resource monitor supplies these algorithms with necessary system information.

The workload manager of the SRM, a collection of subroutines that monitor the rate at which individual address spaces use system resources, supplies the swapping recommendations that SRM control requests. The workload manager also collects data for resource measurement facility (RMF) workload activity reporting.

SRM FUNCTIONS

The SRM consists of a number of modules whose names generally begin with the prefix IRARM. Each module, however, does not directly correspond to a unique SRM function. Rather, separate entry points within the modules identify the various functions that make up the SRM component.

The major functions of the SRM and the modules containing the entry points that perform them are:

- Swap analysis: IRARMSWP uses an algorithm to select which users from which domains to swap in or out. The swap decisions are based upon domain targets and swap recommendation values that the workload manager and resource use algorithms supply.
- Multiprogramming level (MPL) target adjustment: IRARMRMR uses algorithms to monitor system use and raise or lower domain MPLs accordingly.
- Storage management: IRARMSTM and IRARMST2 manage storage using the following algorithms:
  - Unreferenced interval count (UIC) update
  - Page replacement
  - Auxiliary storage shortage prevention
  - Pageable storage shortage prevention
  - SWA storage shortage prevention
  - Storage isolation
  - Storage load balancing
- I/O load balancing: If the installation requests I/O load balancing, IRARMIOM uses an algorithm to assign I/O swap recommendations to address spaces based upon their use of logical paths that are out of balance (underused or overused).
- CPU load balancing: If the installation requests CPU load balancing, IRARMCPM uses algorithms to measure processor utilization and compute processor swap recommendations for address spaces. The CPU swap recommendations complement the I/O swap recommendations that I/O load balancing generates. IRARMCPM contains other algorithms to control changing

priorities (chapping) of address spaces that are in the mean-time-to-wait group or that have changed performance group periods. IRARMCPM also contains functions to activate and deactivate the time-slice algorithm.

- Time-slice: IRARMFIP invokes ASCBCHAP, which changes the priority of an address space on a dispatching queue.
- Workload management: IRARMWLM uses a timed algorithm to periodically compute new workload manager recommendation values for address spaces. IRARMWLM also contains routines for start, stop, suspend, and resume transaction processing and period change processing. IRARMWLS contains the user ready swap-in scheduling routine.
- Workload reporting: IRARMWLS, IRARMWAR, and IRARMFIP perform the workload reporting functions. IRARMFIP contains the subsystem transaction reporting routines. IRARMWLS contains a timed algorithm that manages the transaction reporting queues. IRARMWAR contains routines that update the workload activity measurement table (WAMT) for both MVS (TSO, started task, and batch) transactions and subsystem transactions.

SRM INTERFACE

Most system components use the SYSEVENT macro instruction to communicate with the SRM. SYSEVENTs fall into three classes:

- Address space SYSEVENTs: notify the SRM of a change in status for a particular address space
- System status SYSEVENTs: notify the SRM of a change in system status
- SRM service SYSEVENTs: request a particular SRM support function

The SYSEVENT macro instruction generates either a branch or SVC entry (SVC 95) into the SRM. By enqueueing a time queue element (TQE), the SRM can also invoke itself on a timed basis. When the period of time has passed, the TIME SYSEVENT issued by the timer SLIH drives the SRM. The SRM interface ensures that the SRM executes in 31-bit addressing mode and that the caller's addressing mode is restored on exit.

Entries can be any of three SRM interface modules: IRARMINT, IRARMGLU, or IRARMFIP. IRARMGLU contains 24-bit entry points that pass control to IRARMGLU and IRARMFIP in 31-bit addressing mode. IRARMINT, a generalized routine, passes control to IRARMEVT, which contains a separate processing routine for each SYSEVENT routed through IRARMINT. For the device allocation SYSEVENT (DEVALLOC), the expansion module of IRARMEVT (IRARMEV2) obtains control. IRARMFIP, the fast interface processor, contains entry points for SYSEVENTs that require rapid processing.

Branch entry callers must be in supervisor state or key 0-7, and associated data areas must be page-fixed. Disabled page faults that occur when user data areas are referenced cause the SYSEVENT issuer to be abnormally terminated (abend code X '15F'). Branch entry callers must also pass, in register 13, the address of a 72-byte save area, which can be stored by using the caller's key. The SYSEVENT issuer is responsible for serializing the use of this area (by means of disablement, global lock, or local lock).

SYSEVENT 26 requires no authorization. For other SYSEVENTs, authorization required for SVC entry depends on the SYSEVENT. Programs that issue SYSEVENTs 29 and 2A either must have APF authorization or must be assigned the nonswappable program property (PPTNSWP='1') in the program properties table. All other SYSEVENTs require that the calling program be APF-authorized or in supervisor state or protection keys 0-7;

also, its associated data areas must be page-fixed. Unauthorized use of the SVC entry, like disabled page faults that occur while referencing user data areas, causes the SYSEVENT issuer to be abnormally terminated (abend code X'15F').

The SRM interface passes control to the SYSEVENT processor related to the particular SYSEVENT. Depending upon the SYSEVENT, the SRM might then perform further processing not necessarily related to the invoking SYSEVENT. Thus, many SYSEVENTs serve both as status change notifiers, or service requestors, and as occasions for performing a wide range of SRM processing.

Providing a common point of invocation and simplified access, the SRM interface also processes requests from internal SRM routines that service system components. These include such services as cross memory post, obtaining SQA storage, and issuing a write-to-operator (WTO) message. The service interface routines, each with its own entry point, reside in the module IRARMSRV. Figure 1 lists SRM entry points and identifies the module that contains them. This figure also includes the addressing (AMODE) and the residency (RMODE) attributes for each module. More detailed descriptions of each entry point can be found preceding the method-of-operation diagrams that illustrate each module.

SRM Entry Point Name and Type[1] \ SRM Modules		IEAVNP10	IEEDISPD	IEEMB812	IRARMCPM	IRARMCTL	IRARMERR	IRARMEVT	IRARMFIP	IRARMGLU	IRARMINT	IRARMIOM	IRARMIPS	IRARMOPT	IRARMRMR	IRARMSET	IRARMSRV	IRARMSTM	IRARMST2	IRARMSWP	IRARMWAR	IRARMWLM	IRARMWLS	IRARMEV2	IRARMCHM	IRARMFPG
AMODE[2]		A		31	31	31	31	31	31	A	31	31	31	31	31		31	31	31	31	31	31	31	31	31	
RMODE[2]		24		24	A	A	A	A	A	24	A	A	A	A	A		A	A	A	A	A	A	A	A	A	
CHAP	I				X																					
CKSIGUR	I																	X								
CPLRVSWF	I				X																					
CPUTLCK	I				X																					
CPUWAIT	I				X																					
DEFSWPIN	I																			X						
IEAVNPIO	E	X																								
IEEDISPD	E		X																							
IGC095	E										X															
IRARMACT	E				X																					
IRARMAP1	E				X																					
IRARMAP2	E				X																					
IRARMASM	E					X																				
IRARMCAP	E																			X						
IRARMCED	E					X																				
IRARMCEL	I					X																				
IRARMCEN	I					X																				
IRARMCET	E					X																				
IRARMCHF	E																								X	
IRARMCL0	E				X																					
IRARMCL1	E				X																					
IRARMCL3	E				X																					
IRARMCPF	E																								X	
IRARMCPI	I																			X						
IRARMCPO	I																			X						
IRARMCPS	I								X																	
IRARMCQT	I					X																				
IRARMCRD	I					X																				
IRARMCRL	E					X																				
IRARMCRN	E					X																				
IRARMCRT	I					X																				
IRARMCRY	E					X																				
IRARMCSF	E					X																				
IRARMCSI	E					X																				
IRARMCSO	E					X																				
IRARMCVL	I																			X						
IRARMDBS	I								X																	

[1] E – external entry points (called from outside the module)
I – internal entry points (called from within the same module)
[2] A indicates ANY

Figure 1 (Part 1 of 4). SRM Module/Entry Point Cross Reference

SRM Entry Point Name and Type[1] \ SRM Modules		IEAVNP10	IEEDISPD	IEEMB812	IRARMCPM	IRARMCTL	IRARMERR	IRARMEVT	IRARMFIP	IRARMGLU	IRARMINT	IRARMIOM	IRARMIPS	IRARMOPT	IRARMRMR	IRARMSET	IRARMSRV	IRARMSTM	IRARMST2	IRARMSWP	IRARMWAR	IRARMWLM	IRARMWLS	IRARMEV2	IRARMCHM	IRARMFPG
AMODE[2]		A		31	31	31	31	31	31	A	31	31	31	31	31		31	31	31	31	31	31	31	31	31	
RMODE[2]		24		24	A	A	A	A	A	24	A	A	A	A	A		A	A	A	A	A	A	A	A	A	
IRARMDEL	E							X																		
IRARMEDD	E																								X	
IRARMEVD	E																								X	
IRARMEVU	E																								X	
IRARMEQ1	E				X																					
IRARME05	E									X																
IRARME28	E																							X		
IRARME50	E									X																
IRARME51	E									X																
IRARME53	E									X																
IRARME54	E									X																
IRARME55	E									X																
IRARME60	E									X																
IRARME64	E									X																
IRARMExx	I							X																		
IRARMFMA	E												X													
IRARMFPG	E																									X
IRARMFRE	E												X													
IRARMF05	E								X																	
IRARMF50	E								X																	
IRARMF51	E								X																	
IRARMF53	E								X																	
IRARMF54	E								X																	
IRARMF55	E								X																	
IRARMF56	E								X																	
IRARMF60	E								X																	
IRARMF64	E								X																	
IRARMGMA	E												X													
IRARMHIT	E																						X			
IRARMI00	E									X																
IRARMI01	E										X															
IRARMI02	E																X									
IRARMI03	E																X									
IRARMI04	E																X									
IRARMI05	E																X									
IRARMI06	E																X									
IRARMI07	E																X									
IRARMI08	E																X									
IRARMI09	E																X									
IRARMI10	E										X															
IRARMI11	E																X									
IRARMI16	E																X									
IRARMI17	E																X									
IRARMI20	E																X									
IRARMI21	E																X									
IRARMI48	E										X															
IRARMIL0	E											X														
IRARMIL1	E											X														
IRARMIL3	E											X														
IRARMIL4	E											X														
IRARMIL5	E											X														
IRARMIL6	E											X														

[1] E – external entry points (called from outside the module)
I – internal entry points (called from within the same module)
[2] A indicates ANY

Figure 1 (Part 2 of 4). SRM Module/Entry Point Cross Reference

SRM Entry Point Name and Type[1] \ SRM Modules		IEAVNP10	IEEDISPD	IEEMB812	IRARMCPM	IRARMCTL	IRARMERR	IRARMEVT	IRARMFIP	IRARMGLU	IRARMINT	IRARMIOM	IRARMIPS	IRARMOPT	IRARMRMR	IRARMSET	IRARMSRV	IRARMSTM	IRARMST2	IRARMSWP	IRARMWAR	IRARMWLM	IRARMWLS	IRARMEV2	IRARMCHM	IRARMFPG
AMODE[2]		A		31	31	31	31	31	31	A	31	31	31	31	31		31	31	31	31	31	31	31	31	31	
RMODE[2]		24		24	A	A	A	A	A	24	A	A	A	A	A		A	A	A	A	A	A	A	A	A	
IRARMIPS	E												X													
IRARMLDB	I							X																		
IRARMLPI	E																								X	
IRARMMSP	E																								X	
IRARMMS2	E																		X							
IRARMMS3	E																	X								
IRARMMS4	E																		X							
IRARMMS5	E																		X							
IRARMMS6	E																		X							
IRARMMWR	E												X													
IRARMNOP	E									X							X									
IRARMOPT	E													X												
IRARMPFE	E																X									
IRARMPR1	E																	X								
IRARMPR5	E																	X								
IRARMPR9	E																	X								
IRARMRDR	E	X																								
IRARMRM1	E														X											
IRARMRM2	E														X											
IRARMRPS	E					X																				
IRARMRR1	E						X																			
IRARMRR2	E						X																			
IRARMRVT	E							X																		
IRARMS47	E																			X						
IRARMST	E																X									
IRARMSIC	I															X										
IRARMSOP	I															X										
IRARMSQA	E																		X							
IRARMSRE	E																X									
IRARMSWT	E				X																					
IRARMSWI	E							X																		
IRARMTRC	E										X															
IRARMTRT	E												X													
IRARMUPD	E																								X	
IRARMUXB	E							X																		
IRARMWM0	E																						X			
IRARMWM1	E																					X				
IRARMWM2	E																					X				
IRARMWM3	E																						X			

[1] E – external entry points (called from outside the module)
I – internal entry points (called from within the same module)
[2] A indicates ANY

Figure 1 (Part 3 of 4). SRM Module/Entry Point Cross Reference

SRM Entry Point Name and Type[1] \ SRM Modules		IEAVNP10	IEEDISPD	IEEMB812	IRARMCPM	IRARMCTL	IRARMERR	IRARMEVT	IRARMFIP	IRARMGLU	IRARMINT	IRARMIOM	IRARMIPS	IRARMOPT	IRARMRMR	IRARMSET	IRARMSRV	IRARMSTM	IRARMST2	IRARMSWP	IRARMWAR	IRARMWLM	IRARMWLS	IRARMEV2	IRARMCHM	IRARMFPG
AMODE[2]		A		31	31	31	31	31	31	A	31	31	31	31	31		31	31	31	31	31	31	31	31	31	
RMODE[2]		24		24	A	A	A	A	A	24	A	A	A	A	A		A	A	A	A	A	A	A	A	A	
IRARMWM4	E																									
IRARMWM5	E																				X	X				
IRARMWM7	I																				X	X				
IRARMWM8	E																					X				
IRARMWM9	E																					X				
IRARMWMI	I																				X					
IRARMWMJ	I																				X					
IRARMWMK	E																				X					
IRARMWML	E																					X				
IRARMWMN	E																				X					
IRARMWMO	E																				X					
IRARMWMP	E																				X					
IRARMWMQ	E																				X					
IRARMWMR	E																				X					
IRARMWR1	E																			X						
IRARMWR2	I																			X						
IRARMWR3	E																			X						
IRARMWR4	E																			X						
IRARMWR5	I																			X						
IRARMWR6	E																			X						
IRARMWR7	I																			X						
IRARMWR8	E																			X						
IRARMWR9	E																			X						
IRARMWRA	I																			X						
IRARMXPE	E															X										
IRARMXPS								X																		
IRARMXVT	E							X																		
NEWDP	I				X																					
RMRR2CKQ	I						X																			
RMRR2GST	I						X																			
RMRR2INT	I						X																			
RMRR2PER	I						X																			
RMRR2REQ	I						X																			
RMRR2RTY	I						X																			
RMRR2SPR	I						X																			
RMRR2VFB	I						X																			
RMRR2VLD	I						X																			
STEAL	I																	X								
STMFENCE	I																	X								
STORUSE	I																	X								

[1] E — external entry points (called from outside the module)
I — internal entry points (called from within the same module)
[2] A indicates ANY

Figure 1 (Part 4 of 4). SRM Module/Entry Point Cross Reference

SYSEVENT PROCESSOR

The SYSEVENT processor function (IRARMEVT or IRARMEV2) receives control from the interface function to perform processing related to the SYSEVENT. In a multiprocessing environment, the system may not be able to perform some of these routines immediately because of concurrent SRM processing on another processor. Execution of the requested routines is then deferred until a subsequent invocation of the SRM. SYSEVENTs with critical execution time requirements use a special fast interface, IRARMFIP, rather than the SYSEVENT processor, IRARMEVT. SYSEVENT 1C (device allocation) receives control from IRARMEVT but the code that processes the request resides in the module IRARMEV2.

SYSEVENT processors often invoke other SRM functions. Some of these functions are executed as subroutines of the SYSEVENT processor. Other functions, called actions and algorithms, are treated differently. An action is a user-related function, executed if the SRM lock is held, deferred if the lock is not held. An algorithm is a function executed on behalf of the system instead of the user; its execution is always deferred. SRM control (IRARMCTL) executes deferred actions, algorithms, and timed (periodically invoked) algorithms.

Depending upon what higher level locks are held by the SYSEVENT issuer, SYSEVENT processors might or might not obtain the SRM lock. SYSEVENTs that do not hold the SRM lock return directly to the SRM interface for return to the issuer. SYSEVENT processors that hold the SRM lock generally exit by branching to the SRM control module, IRARMCTL. SRM control exits by returning to the interface module, IRARMINT, whether SRM control is entered via IRARMEVT or IRARMFIP.

Figure 2 lists all SYSEVENTs in alphabetical order with their associated codes (in hexadecimal).

ALTCPREC(21)	ICSCHK(3C)	QSCEFL(12)	SYQSCST(23)
AVQLOW(17)	INITATT(0A)	QSCEST(0C)	TERMWAIT(02)
AVQOK(18)	INITDET(0B)	REQPGDAT(27)	TGETTPUT(22)
BRINGIN(2C)	JOBSELCT(08)	REQSERVC(26)	TIME(05)
CHANNEL(48)	JOBTERM(09)	REQSVDAT(31)	TIMEREXP(01)
CMDEND(40)	MEMCREAT(06)	REQSWAP(2B)	TRANSWAP(0E)
CMDSTART(3F)	MEMDEL(07)	RESETPG(1F)	TRAXERPT(35)
CONFIGCH(1D)	MIGCNSTR(42)	RSMCNSTS(16)	TRAXFRPT(36)
COPYDMDT(28)	MIGPURGE(43)	RSTORCMP(13)	TRAXRPT(37)
DDR(47)	MIGSWAP(44)	SETDMN(25)	UCBCHG(46)
DEVALLOC(1C)	NEWICS(3D)	SOUTSUSP(45)	USERRDY(04)
DIRECTPO(38)	NEWIPS(20)	SQALOW(19)	VERIFYPG(1E)
DONTSWAP(29)	NEWOPT(34)	SQAOK(1A)	WKLDCHG(41)
EASINIT(1B)	NIOWAIT(03)	SWINFL(11)	WKLDCOLL(2E)
ENQHOLD(14)	NOHOLD(33)	SWINSTAT(10)	WKLDINIT(2D)
ENQRLSE(15)	OKSWAP(2A)	SWOUTCMP(0F)	WKLDTERM(2F)
HOLD(32)	QSCECMP(0D)	SYQSCCMP(24)	

Figure 2. SYSEVENT List

Figure 3 lists the SYSEVENTs by hexadecimal code. For each SYSEVENT, the figure identifies the situations that cause the SYSEVENT to be issued, the information passed and returned, the internal SRM routines it might invoke, the functions of the invoked routines, and the actions taken as a result. In addition, this figure indicates whether the SRM interface routine obtains the SRM lock and where control passes after SYSEVENT processing completes. All SYSEVENTs receive the associated codes as input information (in byte 3 of register 0). Bytes 0 and 1 in register 0 pass the ASIDs listed as input. In the following charts, the SYSEVENT names are listed with the hexadecimal code followed by the decimal equivalent.

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
TIMEREXP X'01' 1	TOD clock initialization is complete.	Indication of whether TOD clock initialization occurred (01) or not (00) (register 1, byte 3)	None	Periodic entry point initialization (IRARMWMY)	Resets the time due fields of the time driven queue according to the current time.	When TOD clock initialization occurs, this SYSEVENT schedules SRM's time driven routines. Subsequent scheduling is done through SYSEVENT 05 (TIME). When TOD clock initialization occurs, SRM starts channel measurements.	Yes	SRM control (IRARMCTL)
				Periodic entry point scheduling (IRARMCET)	See Diagram SRM-10.			
TERMWAIT X'02' 2	Issued by TGET or TPUT when a user enters terminal wait.	• ASID • Input (00) or output (80) indication (register 1, byte 0)	None	Control swap-out (IRARMCSO)	See Diagram SRM-12.		Yes	SRM control (IRARMCTL)
NIOWAIT** X'04' 4	Issued by WAIT macro processing when some task in an address space enters long wait.	• ASID	None	Control swap-out (IRARMCSO)	See Diagram SRM-12.		Yes	Invoker via IRARMI01
USERRDY** X'04' 4	An SRB has been scheduled for an address space for which QUIESCE is running, or for a swapped-out address space.	• ASID	None	User ready processing (IRARMHIT)	The ready user is placed on the OUT queue.	User-ready processing is performed through the action request routine.	No	Invoker via IRARMI01

**Note: This SYSEVENT can be invoked while executing in cross memory mode.

Figure 3 (Part 1 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
TIME X'05' 5 (in module IRARMFIP)	SRM timer interval has expired.	None	None	Rotate algorithm (IRARMRT1) Time slice algorithm (IRARMTS1) Channel path status sampling (IRARMCPS) UCB queue length sampling (IRARMDBS) Set timer routine (IRARMI05)	Reorder the ASCB dispatching queue for those priorities due to be rotated. Change the ASCB dispatching priority for those address spaces due to be time sliced up or down. Sample the current channel path activity and increment the channel path counts for active paths. Accumulate the current UCB request queue lengths. Queue a TQE to cause the next SRM timer interrupt.	Invoke the rotate and time slice algorithms if they are active and due to execute. Sample channel measurements if they are active and due to execute. Schedule all other SRM algorithms that are due. Set the time of the next invocation of this SYSEVENT. Start or stop channel measurement if TOD clock status changes.	Yes	SRM control (IRARMCTL) or invoker via IRARMI01
MEMCREAT X'06' 6	An ASID has been associated with a new address space and space has been obtained for an ASCB and OUSB.	• ASID. • START(01)/ LOGON(02)/ MOUNT(03) indication. (register 1, byte 0)	Indication whether or not memory creation should proceed because of a resource shortage. (00–proceed 80–do not proceed). (register 1, byte 0).	Storage request (IRARMI04). User control block repositioning (IRARMRPS)	Obtain storage for an OUCB and OUXB if no resource shortages exist. Place user on "in" queue.	User control block repositioning is performed through the action request routine.	Yes	SRM control (IRARMCTL)
MEMDEL TYPE= NOSWIN X'07' 7	MEMDEL TYPE=NOSWIN is issued by RSM early in address space termination prior to freeing the RSM control blocks.	• ASID • NOSWIN indicator (register 0 byte 2, bit 0)	None	None	NA	Set indicator in OUCB to prevent swapping during address space termination.	Yes	SRM control (IRARMCTL)
MEMDEL X'07 7	Storage associated with an ASCB is about to be freed, and an ASID is disassociated with an address space.	• ASID	Indication that memory delete may proceed. –00 (register 1, byte 3).	OUCB and OUXB delete (IRARMDEL)	Free storage associated with OUCB and an OUXB. Make sure that the rotate and time slice tables have been updated for this user.	OUCB and OUXB delete is performed indirectly, through action request routine.	Yes	SRM control (IRARMCTL)

Figure 3 (Part 2 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
JOBSELCT X'08' 8	An address space has begun using system services, on behalf of a new job, START or MOUNT command, or a TSO session.	Address of the installation control specification parameter list (ICSP) (register 1)	None	Control swapout (IRARMCSO).	Called to swapout an address space if a second level auxiliary page shortage exists or an excess of fixed frames exists.	This SYSEVENT authorizes the accumulation of service for the job. SRM validates the performance group number indicated for the address space. If it is not valid, a default value is assigned.	Yes	SRM control (IRARMCTL)
				Transaction stop routine (IRARMWMO).	Updates the accumulated time and service for a job. Also indicates that the current transaction has ended or been suspended. If workload activity reporting is active, invokes IRARMWR4 to accumulate report information.			
				Assign report and control performance groups. (IRARMFPG)	Assigns the appropriate control and report performance groups specified by the installation control specification.			
JOBTERM X'09' 9	An address space has completed using the system resources on behalf of a job. START or MOUNT command, or a TSO session	• ASID • Address of job-name or user id	None	Transaction stop routine (IRARMWMO)	Updates the accumulated time and service for a transaction. Also indicates that the current transaction has ended. If workload activity reporting is active, invokes IRARMWR4 to accumulate report information.	This SYSEVENT revokes authorization for starting new transactions.	Yes	SRM control (IRARMCTL)

Figure 3 (Part 3 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
INITATT X'0A 10	Whenever an initiator attaches a task	Address of the installation control specification parameter list (ICSP) (register 1)	None	Transaction resume processing (IRARMWMR)	Resumes a suspended transaction, if the performance group number for a new non-TSO job step is the same as for the previous step; otherwise starts a new transaction.	SRM validates the performance group number indicated for the address space. If it is not valid, a default value is assigned. If the input dispatching priority is in the APG, the SRM will follow the IPS specification for this user. Time slice and rotate information is updated.	Yes	SRM control IRARMCTL)
				Assign report and control performance groups. (IRARMFPG)	Assigns the appropriate control and report performance groups specified by the installation control specification.			
				Start new transaction (IRARMWMN).	Indicate the start of a new transaction. If workload activity reporting is active, calls IRARMWR6 to indicate that a transaction has ended.			
				Time slice/ rotate switch (IRARMSWT)	Adjusts the user counts in the time slice and rotate tables.			
				Time slice/ rotate activation (IRARMACT)	Sets the rotating or time slicing status as active or inactive.			
				Change dispatching priority (IRARMI02)	Move ASCB to correct position on dispatcher queue.			

Figure 3 (Part 4 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information: Passed	Information: Returned	Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
INITDET X'0B' 11	Whenever an initiator detaches a task.	• ASID. • Dispatching priority. (register 1, byte 3)	None	Transaction stop routine (IRARMWMO).	Updates the accumulated time and service for a transaction. Also indicates that the current transaction has ended or been suspended. If workload activity reporting is active, invokes IRARMWR4 to accumulate report information.		Yes	SRM control (IRARMCTL)
				I/O load balancing IMCB deletion (IRARML4).	Frees I/O measurements control block (which has been created if the user is a heavy I/O user).	IMCB deletion is performed through action request.		
				Time slice/ rotate switch (IRARMSWT)	Adjusts the user counts in the time slice and rotate tables.			
				Time slice/ rotate activation (IRARMACT)	Sets the rotating or time slicing status as active or inactive.			
				Change dispatching priority (IRARMI02).	Move ASCB to correct position on dispatcher queue.			
QSCEST X'0C' 12	Issued during quiesce processing when the status of all associated tasks has been determined.	• ASID • Long wait indication (00 not in long wait 80 in long wait). (register 1, byte 0)	• Continue with (00) or terminate (08) quiesce processing. (register 1, byte 3).	I/O load balancing user I/O monitoring (IRARMILO). Cross-memory post routine (IRARMI06)	An I/O measurement control block is created for heavy I/O users. The IMCB is updated with channel usage data from the Timing Control Table I/O Table (TCTIOT). (See Diagram SRM-18.) If swap out is turned around, notify the TRANSWAP or REQSWAP issuer.	**Note:** After this SYSEVENT, no further quiesce processing is performed for: • non-swappable users, and • users being swapped because of a long wait, and who are no longer in a long wait status.	Yes	SRM control (IRARMCTL)

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
QSCECMP* X'0D' 13	Issued when the RCT has completed quiesce processing for an address space.	• ASID. • Long wait indicator. (00—not in long wait 80—in long wait). (register 1, byte 0).	• Indication whether the USERRDY SYSEVENT (4) has been received for this task since quiesce start (00—received 80—not received). (register 1, byte 0). • Indication whether to initiate a swapout (00), wait on a QECB for a logically swapped address space (04), or begin restore (08). (register 1, byte 3). • Reg 1 byte 2 contains the swap out reason code.	Transaction Quiesce Processing (IRARMWMQ).	Increments the cumulative service received by a transaction by the amount received during a real storage residence period. Also updates the performance group period indication if a transaction has completed a performance group period. Determines whether to continue the transaction, or to stop or suspend it at this point for the reason that caused the swapout. If workload activity reporting is active, invokes IRARMWR4 to accumulate report information.		Yes	SRM Control (IRARMCTL)
				CPU Load Balancing Profile Adjustment (IRARMCL0).	Updates the mean time to wait indication for use by CPU load balancing (see Diagram SRM-26) and users in the APG.			
				User Control Block Repositioning (IRARMRPS).	Changes the status of the address space to out-of-real-storage and positions it on the correct queue (normally the "out" queue; however, will be the "wait.. queue for users entering long wait, or for users swapped because a resource shortage exists).	User Control Block Repositioning is performed indirectly through action request routine.		
				Page Stealing (IRARMPR9)	Determine swap-in working set size.			
				ASCB Priority Change (IRARMI02)	Change the priority of a logically swapped address space to put it at the bottom of the dispatching queue.			

Notes:

1. After this SYSEVENT no further quiesce processing is performed for:
   - Non-swappable users
   - Users.being swapped because of a long wait and who are no longer in a long wait status
   - Users being logically swapped
2. If a user's think time is low enough, do not physically swap out the address space. Instead, leave the address space in storage, change its dispatching priority to 1, tell QUIESCE to issue a WAIT, and put the OUCB on the WAIT queue. This process is equivalent to a logical swap out. When the address space becomes ready again, the OUCB is moved to the IN queue, and the normal dispatching priority is restored.

Figure 3 (Part 6 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
TRANSWAP X'0E' 14	Issued by the initiator for V=R and nonswappable job steps. Issued by the BTAM open routine to swap out address spaces so that their LSQA and fixed pages can be swapped back into preferred storage rather than reconfigurable storage.	• Current ASID • Address of the ECB to be posted (optional) (register 1)	• Return code: X'00' – The request was honored. X'04' – Another TRANSWAP has already completed.	Control swap-out (IRARMCSO)	Initiates the swap-out of the address space. (See Diagram SRM-12.)	Region control task (RCT) is posted to begin the swap-out. If an ECB was passed as input, the ECB is posted when the address space is next swapped in.	Yes	Dispatcher for an SVC entry of SRM provided that a swap-out was begun; otherwise, SRM control (IRARMCTL)
SWOUTCMP X'0F 15	All I/O required to swap out an address space has completed.	• ASID • Pointer to parameter list (register 1) containing: – number of pages swapped out (word 1, bytes 0 and 1). – working set size, in number of pages to be swapped in (word 1, bytes 2 and 3). – indication of whether the address space is waiting for an unfinished RSM service (Word 2, byte 3, bit 7 on means the address space is waiting for service.)	None	Free OUXB storage (IRARMUXB)	Free the storage associated with an OUXB, when the address space is swapped out.	IRARMUXB is performed indirectly, through action request.	No	Invoker via IRARMI01
				Swap analysis (IRARMCAP)	Swap analysis is requested when a user is voluntarily swapped out.	Swap analysis is invoked through algorithm request rotuine.		
				User ready processing (IRARME04)	See SYSEVENT USERRDY (04)	User ready processing is invoked if the user ready indicator is off, but an indication of an unfinished RSM service is received.		

Figure 3 (Part 7 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
SWINSTAT X'10' 16	By RSM to notify of swap status	• ASID • Code in register 1: 00 — Swap-in is starting. 01 — Frames are allocated for the user's working set pages 02 — Address space is swapped-in	None	Stage one of swap-in processing (IRARMSWI)		IRARMSWI is performed indirectly, through action request.	Yes	Invoker via IRARMI01
SWINFL X'11' 17	Swap-in processing failed to obtain or initialize the LSQA storage for an address space.	• ASID • Indication of swap-in failure (register 1, byte 3): 01 — not enough PCBs 02 — not enough frames available for the working set pages 03 — the swap-in would cause a shortage of pageable frames. 04 — there are no double frame pairs available to back the user's segment table.	None	User control block repositioning (IRARMRPS)	Changes the status of the address space to out-of-real-storage and positions the OUCB on the correct queue (normally the out queue).	User control block repositioning is performed indirectly, through an action routine. The SRM takes one of the following actions depending upon the swap-in failure indication code (register 1, byte 3): 01 — no action. 02 — SRM raises the steal thresholds to obtain enough available frames for this user's working set pages. At the next invocation of swap analysis, the swap-in is retried. 03 — SRM puts the address space in defer status on the WAIT queue. A message is issued to the operator indicating the storage requirements for the address space. 04 — no action.	Yes	SRM control (IRARMCTL)
				Free storage (IRARMI04)		Free OUXB		

Figure 3 (Part 8 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
QSCEFL X'12' 18	The RCT failed to complete quiesce processing because of an abnormal situation.	• ASID	None	User control block repositioning (IRARMRPS)	Changes the current status of the user from out-of-real-storage to in-real-storage.	User control block repositioning is performed indirectly, through an action routine.	Yes	SRM control (IRARMCTL)
RSTORCMP X'13' 19	The RCT has completed restore processing for an address space.	• ASID • Long wait indicator (00—not in long wait 80—in long wait) (register 1, byte 0)	None	Restore completed processing (IRARMWMR)	Invoked so the work-load manager can initialize the fields used for monitoring service during a period of real storage residence.		Yes	SRM control (IRARMCTL)
				User control block repositioning (IRARMRPS)	Changes the current status of the user from out-of-real-storage to in- real-storage.	User control block repositioning is performed indirectly, through an action routine.		
				Control swap-out (IRARMCSO)	Requests that a swap-pable user still in long wait status, or a user with TRANSWAP or REQSWAP pending, will be swapped out.			
ENQHOLD** X'14' 20	A user's execution is delayed because of a request for a resource being held by another user. This SYSEVENT is also issued when an authorized program that issues RESERVE for a shared DASD obtains control of the resource, even if no user's execution is delayed.	• ASID of memory holding resource • Address of QCB for resource	None	None		Users in real storage, holding resources, are given a spurt of non-swappable service equal to the enqueue residence value (ERV). (See Diagram SRM-26.) Users out of storage are marked as holding a requested resource so that swap analysis (IRARMCAP) will initiate a swap-in.	Yes	Invoker via IRARMI01

**Note: This SYSEVENT can be invoked while executing in cross memory mode.

Figure 3 (Part 9 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
ENQRLSE** X'15' 21	A contention or potential contention situation has disappeared because of the release of a resource by a user for whom an ENQHOLD SYSEVENT had previously been received.	• ASID. • Address of QCB for resource	None	None		If the user has freed all resources in contention, eliminate special treatment.	Yes	Invoker via IRARMI01
RSMCNSTS X'16' 22	Real storage has been configured into or out of the system (during system initialization or after a CONFIG storage command or a storage error).	• Indication that the RSM counters are being initialized (register 1, byte 3): 00 – not initialized 04 – initialized	None	Fixed frame threshold calculation (IRARMMS5).	Recalculate thresholds with current RSM frame counts.	IRARMMS5 is scheduled for execution the next time the CONTROL function receives control.	No	Invoker Via IRARMI01
AVQLOW** X'17' 23	The number of available real storage page frames has fallen below the available page queue low limit.	• Indication of cause. (register 1, byte 3): 1– Available queue (AVQ) is below limit. 2– AVQ is 1 when a page fault occurs. 3– AVQ is 0 when a page fault occurs. 4– Number of fixed real frames below 16 megabytes exceeds the limit. 5– Page taken from SQA reserve queue.	None	Main storage occupancy analysis (IRARMMS2)	For level 1, 2, or 3, initiate page stealing. (See Diagram SRM-17.) For level 4, notify the system operator of a pageable storage shortage, inhibit the creation of new address spaces, and attempt to reduce the number of fixed frames by swapping out address spaces. Swap-out the address with significantly more than the average number of fixed pages and identify these address spaces to the system operator. For level 5, go into a 100 millisecond disabled spin loop if there is any outstanding paging I/O.	Because it is important that the main storage occupancy analysis algorithm be executed as soon as possible, an SRB is scheduled after requesting the algorithm; the SRB issues SYSEVENT 30 when it is dispatched, which results in the CONTROL function being invoked. This algorithm (IRARMMS2) will then be executed.	No	Invoker via IRARMI01

**Note: This SYSEVENT can be invoked while executing in cross memory mode.

Figure 3 (Part 10 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
AVQOK** X'18' 24	Enough real storage pages have been freed to alleviate a shortage condition.		None	None		SRM ceases its special efforts to free real storage.	No	Invoker via IRARMI01
SQALOW** X'19' 25	There exists a critical shortage of SQA pages below 16 megabytes.	• Indication of whether shortage is of severity 1 (01) or 2 (02) (register 1, byte 3)	None	SQA shortage message writer (IRARMSQA).	Inform the system operator of the SQA shortage (See Diagram SRM-17.)	The message writer algorithm is scheduled for execution the next pass through the CONTROL function. SRM does not permit the creation of new address spaces when an SQA shortage exists.	No	Invoker via IRARMI01
SQAOK** X'1A' 26	An SQA page shortage has been relieved below 16 megabytes.	Code indicating the level of the relieved shortage (register 1, byte 3): above level 1 (01) above level 2 (02)	None	SQA shortage message writer (IRARMSQA).	Inform system operator of the fact that an SQA shortage has been relieved. (See Diagram SRM-17.)	Issue a message if all SQA shortages are relieved (that is, level 1).	No	Invoker via IRARMI01
EASINIT X'1B' 27	System address space initialization is creating a new address space.	• ASID • Indication of whether the new address space is to be privileged (1) or not (0) (register 1, byte 2, bit 0). • Code indicating dispatching priority of address space (register 1, byte 3): – data only (00) – high (01) – low (02)	None	Find performance group routine (IRARMFPG)	Assigns the appropriate control and report performance groups as specified by the installation control specification.	This SYSEVENT assigns the appropriate performance group, domain, and dispatching priority to the new address space.	Yes	SRM control (IRARMCTL)
				Transaction start routine (IRARMWMN)	Starts a new transaction.	If the time-of-day clock has been initialized, this SYSEVENT starts a transaction for the new address space.		

**Note: This SYSEVENT can be invoked while executing in cross memory mode.

Figure 3 (Part 11 of 26). SYSEVENT Processor

Figure 3 (Part 12 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
DEVALLOC X'1C' 28 (IRARMEV2)	A device allocation choice must be made among two or more candidates.	• ASID • Pointer to a three word list (register 1) containing: – address of a list of candidate UCB addresses (word 1) – address of a list of UCB addresses already allocated to requestor (word 2) – address of a two word return area. (word 3)	• Pointer to same three word list as on entry (register 1), with return area containing: – address of the candidate list entry that was selected (word 1) • Successful (00) or unsuccessful (08) indication (Register 15, byte 3)	None		The UCB is selected by applying the following selection principles in the order indicated: • Avoid contention (reallocating same UCB to same user) for direct access. • Avoid allocation on units with premounted volumes. • Give preference to less heavily utilized logical paths, assuming that each previous allocation for this user has a known projected constant impact on utilization. • Select the tape drive specified in the installation's IEAOPTxx parmlib member. The selection options are: – the next highest device number from the last allocated device (default) – the first device number in the candidate list – the lowest device number in the candidate list – random selection from the candidate list • For direct access devices, pick the one with the lowest average delay, assuming each previous allocation for this user has a known constant impact on delay. • Choose randomly, if more than one candidate remains.	Yes	SRM control (IRARMCTL)

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
CONFIGCH X'1D' 29	A CONFIG command has been issued for a channel or processor.	• ASID. • SMF record describing the change. (pointed to by register 1)	None	None		SRM enables another processor for I/O interruptions. If a vector facility is coming online, SRM checks for users waiting for a vector. If any are found, they are made ready. In addition, updates SRM control information for demand paging thresholds for MPL adjustment. Recalculates all processor execution time thresholds based on the number ofprocessors previously and presently online.	Yes	SRM control (IRARMCTL)
VERIFYPG X'1E' 30	An interpreter has received a performance group number which needs verification.	• Performance group number. (register 1, bytes 2-3)	• Valid (00)/ invalid (02 – TSO user ASID) indication (register 1, byte 2)	None		The IPS is checked for performance group number validity. If the number is invalid, a default is provided.	Yes	SRM control (IRARMCTL)
RESETPG X'1F' 31	The system operator has entered a RESET command for a particular address space.	• ASID. • New performance group number. (register 1, bytes 2-3)	Return code indicating – request honored (00) or –performance group number invalid (04) or –ASID not currently assigned (08). (register 1, byte 2).	Start new transaction (IRARMWMN).	For users in real storage, a new transaction is started. For swapped out users a new transaction is started when the swap in occurs. If workload reporting is active, IRARMWR6 is called to indicate that a transaction has ended.	Starting a new transaction results in the user being associated with the performance objective, domain, and dispatching priority control corresponding to the first period of the performance group definition.	Yes	SRM Control (IRARMCTL)
NEWIPS X'20' 32	The system operator has entered a SET command with the IPS keyword.	• ASID • Pointer to WMST describing new IPS. (register 1)	• Old IPS description	Set to new IPS (IRARMSET).	If workload activity reporting is active for the measurement facility, the reporting is terminated (it is re-established later by the measurement facility. A performance group number for each transaction is assigned and a new transaction is begun.	The IRARMSET routine is called by the IRARMIPS routine in IRARMEVT which is invoked indirectly, through the action request routine.	Yes	SRM Control (IRARMCTL)

Figure 3 (Part 13 of 26). SYSEVENT Processor

Figure 3 (Part 14 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routine Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
NEWIPS Cont'd. X'20' 32				ASCBCHAP standard interface (IRARMAP2) Create or free I/O load balancing control blocks (IRARMLDB)	If necessary, all address spaces currently in storage are given a new dispatching priority. If I/O load balancing is active, the SRM creates control blocks to maintain load balancing data. If load balancing is being turned off, the SRM frees these control blocks.			
ALTCPREC X'21' 33	As a result of an error some processor has had to be configured out of of the system.	• Address of the PCCA for the processor. (register 1).	None	Enable a processor for I/O interrupts (IRARMILG)		If a processor enabled for I/O interruptions is moving offline, the SRM enables another processor for I/O interruptions. In addition, updates SRM control information for demand paging thresholds for MPL adjustment. Recalculates all processor execution time thresholds based on the number of processors previously and presently online.	Yes	SRM control (IRARMCTL)

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
TGETTPUT X'22' 34	A TGET OR TPUT instruction has completed some I/O to a terminal.	• ASID. • TGET (0) or TPUT (1) indication. (register 1, byte 0, bit 0). • (for TGET) entire message transferred indicator. (0—all transferred; 1—at least one more TGET required). (register 1, byte 0, bit 0).	None	Start new transaction (IRARMWMN)	For TGET, indicates the start of a new TSO transaction. If workload reporting is active, IRARMWR6 is called to indicate that a transaction has ended. If the TGETTPUT SYSEVENT was preceeded by a TERM-WAIT condition the IRARMWMN routine is instead called at the time the address space is swapped in.	Starting a transaction results in the user being associated with the first period of his performance group.	Yes	SRM control (IRARMCTL)
SYSQSCT X'23' 35	The system start/stop routine has been entered to stop the system.	None	None	None		The SRM saves the time at which the system was stopped.	No	Invoker via IRARMI01
SYQSCCMP X'24' 36	The system start/stop routine is about to restart the system.	None	None	None		Steps forward transaction starting times by the duration of the system stoppage.	No	Invoker via IRARMI01

Figure 3 (Part 15 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
SETDMN X'25' 37	The operator entered a SETDMN command to change constraint values for a domain.	• Pointer to the SETD parameter list (register 1).	Return code (register 15) 0: Successful 4: Invalid domain 8: Minimum exceeds maximum 12: Invalid AOBJ or DOBJ value. 16: SETDMN input is incompatible with the current IPS.	None		Update the domain descriptor table with the new values in the SETDMN command.	Yes	Invoker via IRARMI01
REQSERVC X'26' 38	Issued by the TSO TIME command, to obtain user related service data.	• ASID. • Address of 3-word return area. (register 1).	• Return area for a TSO user: – Total service (word 1). – Total transaction active time for all transactions (word 2). – Last performance group number (word 3, bytes 0 & 1). – Total number of transactions (word 3, bytes 2 & 3). • Return area for a non-TSO user: – Total service (word 1). – Total transaction active time (word 2). – Last performance group number (word 3, bytes 0 & 1). • Indication whether data was successfully returned (00) or not (04). (register 15, byte 3).	Service calculation routine (IRARMWM1).	Calculates the service accumulated during the current "in real storage" interval. This is added to previous accumulated service to obtain total service.	Accumulated service information is stored in the user's area (while not holding the SRM lock) and under the user's protect key.	Yes	Invoker via IRARMI01

Figure 3 (Part 16 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
REQPGDAT X'27' 39	Issued by SMF during step termination, to obtain user paging data. This SYSEVENT is intended for SMF. Other callers must not set the indicator requesting that the related data fields in the OUSB and OUXB be reset to zero. If reset is requested by another caller, the data is lost to SMF.	• ASID. • Address of 16 word return area. (register 1). • Indicator of whether the related data fields in OUSB and OUXB are to be reset to zero.	• Return area: – Non VIO page-ins (word 1). – Non VIO page-outs (word 2). – Non VIO reclaims (word 3). – VIO page-ins (word 4). – VIO page-outs (word 5). – VIO reclaims (word 6). – Pages swapped in (word 7). – Pages swapped out (word 8). – Swapouts (word 9). – Common area page-ins (word 10). – Common area reclaims (word 11). – Pages stolen (word 12). – LPA page-ins (word 13). – LPA reclaims (word 14). – CPU page-seconds (words 15, 16). • Indication whether data was successfully returned (00) or not (04). (register 15, byte 3).	None		The SRM obtains paging data from SRM control blocks and if requested resets related fields in these blocks to zero.	Yes	SRM Control (IRARMCTL)

Figure 3 (Part 17 of 26). SYSEVENT Processor

Figure 3 (Part 18 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information Passed	Information Returned	Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
COPYDMDT X'28' 40	Issued when a "DISPLAY" command with the keyword "DMN" has been entered.	Pointer to a fixed area to receive a copy of the domain table. (register 1).	Pointer to same area (register 1). Word 0 indicates whether nonswappables are counted in the CMPLs of domains (x'40') and whether the domain table is included (x'80'). Words 1 and 2 contain the time when displayed values were in effect. Words 3 and 4 contain IPS parmlib member name. Words 5 and 6 contain OPT parmlib member name. Words 7 and 8 contain installation control specification parmlib member name. Word 9 and on — count of domains and domain table.	None		Duplicate Domain Information	Yes	Invoker Via IRARMI01
DONTSWAP X'29' 41	Issued to notify SRM that the issuing address space must not be swapped out until either an OKSWAP (2A) or INITDET (0B) SYSEVENT.	• ASID.	• Indication whether request was honored (00), was not honored because it was not for the current address space (04), or was not honored because it was not authorized (08). (register 1, byte 3).	Swap Status Change Request (IRARMWMK).	Determine SRM algorithms applicable to user, and reposition user on SRM swap queue.		Yes	SRM Control (IRARMCTL)

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
OKSWAP X'2A' 42	Issued to notify SRM that issuing address space, which had previously issued a DONTSWAP SYSEVENT, may again be considered for swapping.	• ASID.	• Indication whether request was honored (00) was not for the current address space (04), or was not authorized (08). (register 1, byte 3).	Swap Status Change Request (IRARMWMK).	Same as for DONTSWAP (29).		Yes	SRM Control (IRARMCTL)
REQSWAP X'2B' 43	Issued when a CONFIG storage command has been issued, to swap out the address space that occupies the storage to be taken offline.	• ASID. • Address of ECB to be posted (if dependency exists on requested swap). (register 1).	• Indication whether the request is honored (00), was ignored because the address space is non-swappable (04), or is ignored because the address space is in the process of swapout(0C). (register 1, byte 3).	Control swapout (IRARMCSO).	Initiates the swapout of the address space. (See Diagram SRM-12.)	The region control task (RCT) is posted to begin the swapout. If swap completion notification is requested (by providing an ECB), the ECB is posted when the address space is next swapped in.	Yes	SRM control (IRARMCTL)
BRINGIN X'2C' 44	Issued when the system operator has issued a CANCEL command for a particular job.	• ASID.	• Indication whether request is honored (00), or is not honored because the address space is in the process of being swapped (08). (register 1, byte 3).	Simulate user ready notification (IRARMHIT).	Invokes IRARMWMU to make the address space eligible for swap-in.	Expedite the swap-in of an address space that is swapped-out.	Yes	SRM control (IRARMCTL)

Figure 3 (Part 19 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
WKLDINIT X'2D' 45	Issued by RMF to request that SRM begin collecting workload activity data.	• ASID. • Data collection buffer address. (register 1)	• Indication whether request is honored(00), or is not honored because of incorrect buffer size (08), or data collection is already active (20). (register 15, byte 3).	Workload activity recording initialization (IRARMWR1).	Constructs and initializes the workload activity measurement table (WAMT).		Yes	SRM control (IRARMCTL)
WKDCOLL X'2E' 46	Issued by RMF at the end of a reporting interval, to collect workload activity data.	• ASID. . • Data buffer address. (register 1)	• Indication whether request was honored (00), whether an IPS change has occurred (04), or data buffer had not yet been established (40). (register 15, byte 3).	Workload activity recording data collection (IRARMWR3).	Moves the contents of the WAMT into a collection buffer.		Yes	SRM control (IRARMCTL)
WKLDTERM X'2F' 47	Issued by RMF to terminate workload activity data recording, at RMF termination or when an IPS change has occurred.	• ASID.	• Address of the buffer no longer used by SRM. (register 1). • Indication whether the request was honored (00) or the data collection buffer had not yet been established (40). (register 15, byte 3).			The SRM indicates that workload activity data collection no longer be performed.	Yes	SRM control (IRARMCTL)
X'30' 48	Issued by the SRM when the control function must be invoked immediately (that is, without waiting for the next SYSEVENT issued by another component).	• ASID. • Address of issuing SRB (register 1).	None	SRM control (IRARMCTL).	Performs control mainline processing, in the course of which a scheduled critical function will be performed (See Diagram SRM-7).	Frees up SRM SRB for reuse.	Yes	SRM control (IRARMCTL)

Figure 3 (Part 20 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
REQSVDAT X'31' 49	Issued by SMF during job session termination to obtain user related service data.	• ASID. • Address of 8 word return area (register 1)	• Return area for a user: – Total service (word 1) – Total transaction active time (for TSO, the total of all transactions) (word 2) – Last performance group number (word 3, bytes 0&1) – TSO only: total number of transactions (word 3, bytes 2 & 3) – Session residency time (word 4) – Session processor service (word 5) – Session IOC service (word 6) – Session MSO service (word 7) – Session SRB service (word 8) • Indication whether data was successfully returned (00) or not (04) (register 15, byte 3)	Service calculation routine (IRARMWM1)	Calculates the service accumulated during the current in real storage interval. This is added to previous accumulated service to obtain total service.	Accumulated service information is stored in the caller's area under the caller's protect key.	Yes	Invoker via IRARMI01

Figure 3 (Part 21 of 26). SYSEVENT Processor

Figure 3 (Part 22 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
HOLD X'32' IRARMFIP 50	Issued to notify SRM that the issuing address must not be swapped out until a NOHOLD SYSEVENT (33) is issued.	• ASID	None	None	Not applicable	Increase the HOLD count in OUCB for this user. Note: HOLD/NOHOLD have a function similar to the DONTSWAP (29)/ OKSWAP (2A) SYSEVENTs, but are faster.	No	Invoker
NOHOLD X'33' 51 IRARMFIP	Issued to notify SRM that the issuing address space, which has previously issued a HOLD SYSEVENT, may again be considered for swapping.	• ASID	None	None	Not applicable	Decrease the HOLD count in the OUCB for this user. When the count reaches zero the address space is again considered for swapping.	No	Invoker
NEWOPT X'34' 52	Issued when the system operator has entered a SET command with the OPT keyword.	The address of the parameter list containing the new OPT constants.	None	Set to a new OPT (IRARMSOP). Start or turn off I/O load balancing (IRARMLDB)	Copy the OPT parameters into various SRM control blocks. If I/O load balancing is active, the SRM creates control blocks to maintain load balancing data. If load balancing is being turned off, the SRM frees these control blocks.	Call IRARMSOP to copy the values from the parameter list to the control blocks.	Yes	SRM control (IRARMCTL)

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
TRAXERPT X'35' 53 IRARMFIP	Issued to report the completion of a subsystem transaction for which service data is available.	Pointer to a 72-byte area containing transaction start time, name, userid, class and subsystem name as well as CPU time, SRB time, MSO time and I/O count. (register 1)	Indication of one of the following: • Whether the request was honored (00) • If an immediate retry might succeed (08) • If reporting is temporarily suspended (12) • If reporting is inoperative (16)	None	Not applicable	Calculate transaction elapsed time and save it with input in an extended queue element.	No	Invoker
TRAXFRPT X'36' 54 IRARMFIP	Issued to report the completion of a subsystem action.	Pointer to a 40-byte area containing transaction elapsed time, name, user ID, class, and subsystem name (register 1)	(Same as for TRAXERPT 35)	None	Not applicable	Save input data in a basic queue element.	No	Invoker
TRAXRPT X'37' 55 IRARMFIP	(Same as for event 36)	Pointer to a 40-byte area containing transaction start time, name, user ID, class, and subsystem name (register 1)	(Same as for TRAXERPT 35)	None	Not applicable	(Same as for event 36)	No	Invoker
DIRECTP0 X'38' 56 IRARMFIP	When a page is being removed from real storage.	• ASID or 'FFFF' (common) or (virtual fetch) (register 0, byte 0-1) • Page-out page indicator 0 – unchanged 1 – changed (register 1, byte 0, bit 0) • Page type 1 – page-out 2 – virtual fetch (register 1, byte 3)	• Destination indicator 0 – to extended storage 4 – to auxiliary storage (register 15, byte 3)	None	Not applicable	Direct page-out to auxiliary storage or extended storage on criteria table.	No	Invoker

Figure 3 (Part 23 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information Passed	Information Returned	Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
ICSCHK X'3C' 60 IRARMFIP	To determine if the UADS should be examined to validity check the logon performance group specification.	None	Indication of whether TSO is specified in the installation control specification (00), or not (12).	None	Not applicable	Examine the TSO bit in the installation control specification common section.	No	Invoker
NEWICS X'3D' 61	The operator has requested that a new set of installation control specification parameters be placed into effect.	Address of the new installation control specification control parameters (register 1)	Address of the installation control specification parameter table previously in effect (register 1)	Set new installation control specification parameters into effect (IRARMSIC)	Inserts the address of the new installation control specification parameter table into the RMCT. If workload activity reporting is active, the reporting is terminated.	The new installation control specification parameters become active in both controlling and reporting work.	Yes	SRM control (IRARMCTL)
NEWICS X'3E' 62 continued				IRARMFPG	Assign report and control performance groups using the new installation control specification tables.			
				IRARMWMN	Start a new transaction if the user switched control performance groups.			
CMDSTART X'35' 63	Issued by the TMP when a command processor is to be invoked using either ATTACH or LINK	Address of the command start parameter list	None	IRARMFPG	If not processing a CLIST, assign report performance groups from the installation control specification using the new transaction name.	The process of starting a new transaction results in a user being associated with the first period of the associated performance group.	Yes	For time sharing address spaces, exit to SRM control (IRARMCTL) For non-time sharing address spaces, exit to the caller (IRARMI01)
				Start new transaction (IRARMWMN)	If processing a CLIST and the installation requested that each command be treated as a new transaction, start a new transaction.			

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
CMDEND X'40' 64 IRARMFIP	When a TSO command is completed.	• ASID • Stacking command indicator: – 00 – This command did not change the stack – 08 – This command added one or more mands to the stack	None	None	Not applicable	If the command is not from a stack or did not change the stack, set indicators to stop counting for the command and clear its transaction name.	No	Invoker
WKLDCHG X'41' 65	When takeover is in progress	• ASID	None	Enable a processor for I/O interrupts (IRARMIL6)		Establish fast workload acceptance for the specified address space	Yes	Invoker
MIGCNSTR X'42' 66	When RSM can't find enough extended storage frames to migrate or this shortage is relieved.	• Status indicator (01–shortage 02–relieved) (Register 1, byte 3)	• Override indicator in shortage case 00–override storage isolation 04–don't override (Register 15, byte 3)	None	Not applicable	Determine if storage isolation should be overridden.	Yes	Invoker
MISPURGE X'43' 67	When there is a shortage of extended storage and some needs to be freed.	None	• ASCB or zero • Number of pages to migrate • Number of pages to be converted and freed	None	Not applicable	Look for pages to migrate: • Deferred migration swaps • Address spaces on auxiliary storage with pages on extended storage • Isolated address spaces above maximum.	Yes	Invoker
MIGSWAP X'44' 68	When RSM attempts to swap a primary working set.	• ASID • Type of swap code 0 – reconfig 1 – purge from extended 2 – age • Number of frames being migrated	• Number of frames being migrated (or zero).	Control swap-in (IRARMCSI) OUCB repositioning (IRARMRPS)	Swap the address space in Switch to appropriate queue	If address space is not about to be swapped in, migrate it now or later.	Yes	Invoker

Figure 3 (Part 25 of 26). SYSEVENT Processor

SYSEVENT	When Issued	Information		Routines Invoked	Function of Invoked Routine	SRM Action	SRM Lock Held	Exit To
		Passed	Returned					
SOUTSUSP X'45' 69	When swap-out is suspended because of a shortage of extended storage frames.	• ASID	• Swap-out indicator 00 — defer 04 — terminate	None	Not applicable	Tell RSM to swap to auxiliary or suspend.	No	Invoker
UCBCHG X'46' 70	When a VARY command is issued to place a device online or offline, or a CONFIG command is issued to place a channel path online or offline.	Address of the UCB (register 1)	None	Determine the logical path index for a device (IRARMLPI)	Calculate the index to the logical path block corresponding to the current set of channel paths to each device.	If the device is coming online, build a data collection block for it. If the device is going offline, free its data collection block.	Yes	Invoker
DDR X'47' 71	When a SWAP command is issued.	Address of a parameter list with forward and backward pointers to the UCBs.	None	Determine the logical path index for a device (IRARMLPI)	Calculate the index to the logical path block corresponding to the current set of channel paths to each device.	Update the logical path index in the control block corresponding to each of the two devices.	Yes	Invoker
CHANNEL X'48' 72	When the channel reports an error in the measurement facility.	Address of the channel facility report block (CFRB) (register 1).	None	Stop channel measurements (IRARMMSP)	Turn off the channel measurement facility, signal the operator, and issue an ENF signal.	Stop the collection of channel measurements. If the error is temporary, ensure that measurements are started again after a 16-second delay.	Yes	Invoker

Figure 3 (Part 26 of 26). SYSEVENT Processor

SRM CONTROL BLOCKS AND QUEUES

The main control block table for the SRM is the resource manager control table (RMCT). The RMCT, which resides in the nucleus resident module IRARMCNS, is located by a pointer in the CVT (CVTOPCTP). (See "Control Block Overview" later in this book.) IRARMCNS also contains the following:

- Control tables used by processor, I/O, storage, and resource management routines
- Constants used by the control algorithm to determine the criteria and frequency of swap analysis
- Tables describing the entry points for various SRM routines, algorithms, and event-initiated actions

The state of each address space is described by the user indicators in the corresponding SRM user control block (OUCB). (See "Individual User Indicators" later in this book.) Each OUCB resides on one of four OUCB queues, which correspond to categories of address spaces. When the state of an address space changes, the OUCBGOB bit is set in its OUCB, and the OUCB reposition routine is called to move the OUCB to the queue and position which reflects its new state.

A description of each category of address space, corresponding to the four OUCB queues follows.

- IN - The address space contains ready work. It is swapped in and actively completes for system resources.
- OUT - The address space contains ready work. It does not actively compete for system resources because it is logically swapped out (I/O operations are quiesced) or physically swapped out (resides on auxiliary or extended storage). The address space is eligible for swap-in.
- Logical Swap WAIT - The address space does not contain ready work, and is logically swapped out. The address space is not eligible for swap-in.
- WAIT - The address space is physically swapped out, and is not eligible for swap-in because
  - It contains no ready work, or
  - It was swapped out to relieve a resource shortage, or
  - Swap-in failed too many times.

The SRMDATA format control statement of PRDMP (print dump) and IPCS and the CBFORMAT command of IPCS can format the OUCBs on these four queues. The CBSTAT command of IPCS can also analyze an OUCB for important information. SRMDATA also formats the domain table (DMDT), which indicates the swapping status of the system.

If an address space is being swapped in (bit OUCBGOI is on) or swapped out (bit OUCBGOO is on), or is about to change queues (bit OUCBGOB is on), its position on the OUCB queue may not reflect its current status. When a swap completes (successfully or unsuccessfully), the OUCB is repositioned to reflect its current status.

The primary address-space-related control block is the OUCB, which holds much of the information about the state of an address space. For swapped-in users, other information exists in the SRM user extension block (OUXB). (Note that both the OUCB and the OUXB are mapped in IRARMCNS). If an address space is swapped out, an SRM user swap block (OUSB) built in LSQA saves some of the data that the OUXB of the address space contained. The OUSB is swapped out with the address space.

When the OUCB reposition routine is called, it examines the user indicators for the address space to determine the queue and position which reflects the address space's status. The following tests are made in succession to determine the state and placement of the OUCB.

OUCB	Placement
OUCBOUT='0'B	The address space is swapped in and the OUCB is placed at the end of the IN queue.
OUCBRDY='1'B OUCBAXS='0'B OUCBFXS='0'B OUCBJSAS='0'B OUCBJSFS='0'B OUCBCSFS='0'B	The address space is eligible for swap-in and the OUCB is placed at the end of the OUT queue.
OUCBMGSW='1'B	A migration swap for the address space has been deferred and the OUCB is placed at the beginning of the WAIT queue.
OUCBLSW='0'B	The address space is physically swapped out and the OUCB is placed at the end of the WAIT queue.
OUCBESSW='1'B	A swap-out to extended storage for the address space has been deferred and the OUCB is placed at the beginning of the logical swap WAIT queue.
All others	The address space has been logically swapped by think time and the OUCB is placed at the end of the logical swap WAIT queue.

When repositioning is completed (OUCBGOB bit is turned off), the following queue conditions are met.

- A swapped-in address space can only be on the IN queue.
- A logically-swapped address space can be on the logical swap WAIT queue or the OUT queue.
- An address space that is swapped out can be on the WAIT queue, the logical swap WAIT queue, or the OUT queue.
- An address space for which a migration swap has been requested can be on the WAIT queue or the OUT queue.
- A swapped-out address space can be on the WAIT queue or the OUT queue.
- An address space being swapped in can only be on the OUT queue.

The setting of the swap transition flags for swap-out processing occurs in the following order.

- If swap-out is initiated successfully the OUCBGOO bit is turned on.
- The OUCB is repositioned and the OUCBGOO bit is turned off at quiesce complete for a logical swap-out and at swap-out complete for a physical swap-out.

The setting of the swap transition flags for swap-in processing occurs in the following order.

- If swap-in is initiated successfully, the OUCBGOI bit is turned on.
- The OUCB is repositioned and the OUCBGOI bit is turned off at restore complete for a logical swap-in and at swap-in status 2 for a physical swap-in.

The setting of the swap transition flags for migration swap processing occurs in the following order.

- If the migration swap is initiated successfully, the OUCBGOO bit is turned on.
- At swap-out complete, the OUCBGOO bit is turned off, and the OUCB is repositioned if it is on the WAIT queue.

SRM CONTROL

SRM control is the dispatcher of the SRM. It resides in the module IRARMCTL along with various other SRM routines. Most SYSEVENTs that execute holding the SRM lock exit to SRM control to perform the following functions:

- SRM control executes deferred actions (routines that execute on behalf of a single user). Examples of actions are:
  - Moving a user control block from one SRM queue to another
  - Memory delete processing
- SRM control executes deferred algorithms (routines that execute on behalf of the entire operating system). Examples of algorithms are:
  - Page replacement
  - Real page shortage prevention
- Following the TIMEREXP SYSEVENT, SRM control schedules timed algorithms. Examples of timed algorithms are:
  - Assigning swappable users their current workload level (swappable user evaluation algorithm)
  - Keeping the multiprogramming level (MPL) at its target level in each domain by performing user swaps (swap analysis algorithm)

ACTION/ALGORITHM SCHEDULING

Actions and algorithms can be requested and scheduled by any of the components of the SRM. Request handling subroutines within IRARMCTL deal with these requests. Requests for actions are processed in one of the following ways:

- The action is called inline if the SRM lock is held and if the action was not requested by another action.
- Otherwise, the action is deferred. To indicate that the action was requested, a flag is set in the OUCB.

Requests for algorithms are always deferred, but, to indicate a request, a flag is set in the RMCT. If a deferred action or algorithm is considered critical, the request-handling subroutine schedules an SRM to another entry point, IRARMCED (within IRARMCTL). IRARMCED executes SYSEVENT X'30'. SYSEVENT X'30' exits to SRM control, which then executes the deferred action or algorithm.

Non-critical actions and algorithms, requested but deferred, execute during the next pass through SRM control. This execution normally occurs after the processing of the next SYSEVENT that holds the SRM lock.

SRM control identifies the actions and algorithms to execute by examining bit strings in the OUCB (for actions) and the resource manager control table (RMCT) (for algorithms). Bits set to 1 in the OUCB (OUCBACN field) and in the RMCT (RMCTALA and RMCTALR fields) identify deferred action and algorithm requests respectively.

The actual addresses of the individual routines that process actions and algorithms are located in resource manager entry point elements (RMEPs), which are chained together. One RMEP chain exists for actions and another for algorithms. SRM control compares the bits set to 1 in the bit string (the OUCB or RMCT) with each RMEP in the action/algorithm RMEP chain. When a match occurs, the entry point address in the RMEP identifies the action or algorithm routine that will get control.

As part of routing control to the identified routine, SRM control turns off the bit in the OUCB or RMCT used to find the appropriate RMEP. When all bits in the OUCB and RMCT bit strings are off, SRM control has processed all deferred actions and algorithms and exits to a return point in the SRM interface module IRARMINT. Figure 4 and Figure 5 show in more detail the routines and bit settings used to process algorithms and actions.

	REQUESTING	SCHEDULING			EXECUTING		
	SRM Lock Held	Scheduling Routine	RMEP Chain	Bit String	SRM Lock Held	Executing Routine	RMEP Chain
ACTIONS	NO YES[1]	IRARMCRN	EPDT	OUCBACN[2]	YES	(IRARMCEN, IRARMCRT)	EPDT
ALGORITHM	NO YES	IRARMCRL IRARNCRL	EPAT EPAT	RMCTALA RMCTALR	{YES	(IRARMCEL, IRARMCRT)	EPAT}
TIMED ALGORITHM	YES	IRARMCET	IRACTMQE	RMCTALR	YES	(IRARMCEL, IRARMCRT)	EPAT

Notes:

1. [1]If SRM lock is held when an action is requested, it is not deferred except where an action invokes another action. Control passes to IRARMCRY if the action is IRARMCSI or IRARMCSO or to IRARMCRN for all all other actions and then directly to the action.

2. [2]During execution this field is inspected only in OUCBs that have been queued by the action-scheduling routine (IRARMCRN).

Figure 4. Processing of Actions and Algorithms in IRARMCTL

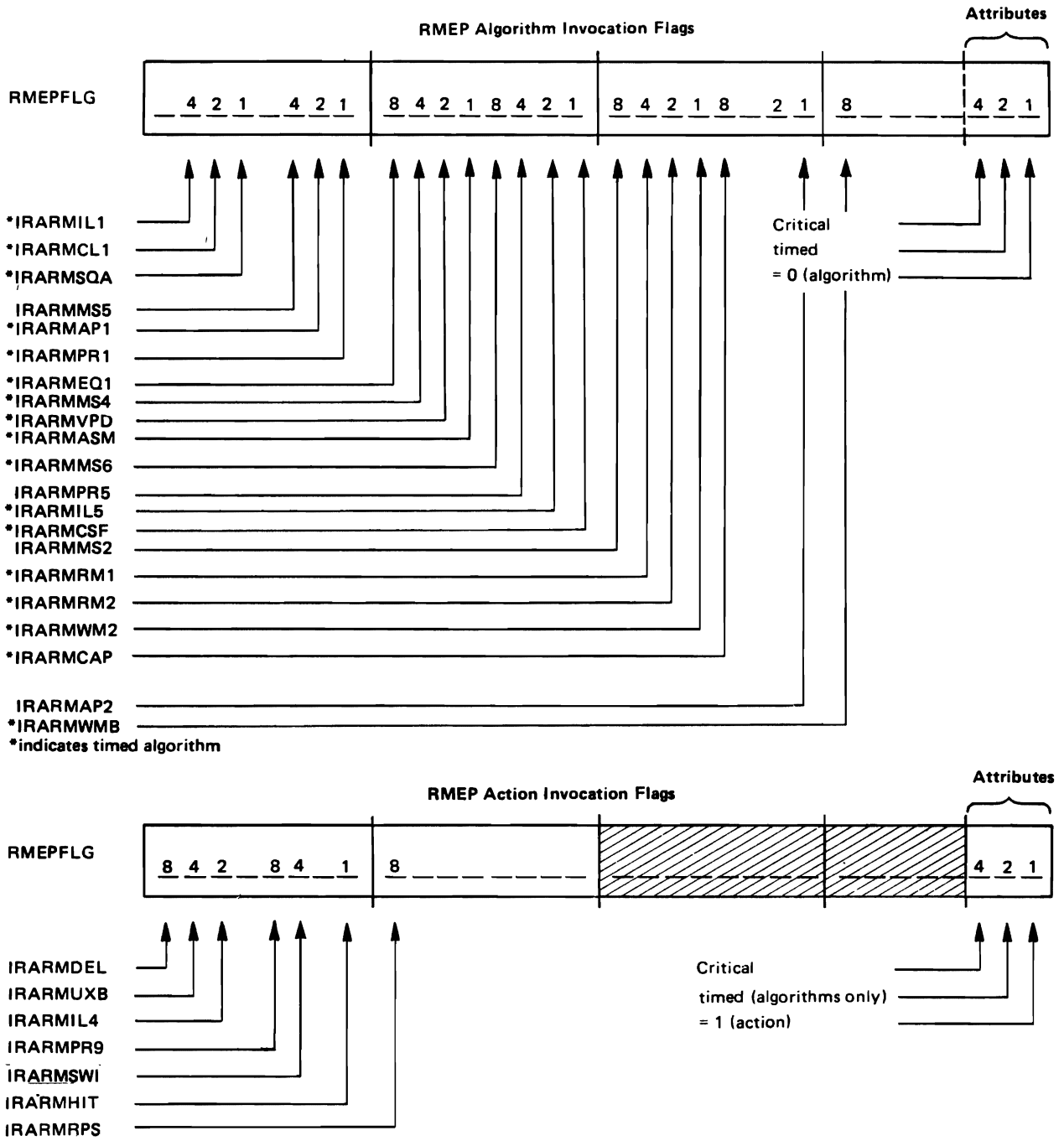


Figure 5. RMEP Algorithms and Action Invocation Flags

SWAP ANALYSIS

The swap analysis algorithm, which resides in the module IRARMSWP, attempts to maintain the multiprogramming level (MPL) at the target value in each domain defined to the system. A domain is a group of user address spaces, defined in the installation performance specification (IPS), that have common execution characteristics; for example, all TSO users might be assigned to one domain. The MPL in a domain is the number of users in that domain who occupy real storage. The target MPL is the number of users in real storage that the SRM resource monitor has determined to be optimal for this domain.

The decision to swap an address spaces is made on the basis of a number of input factors supplied by other SRM functions. The workload manager provides workload levels for each user. The resource-use algorithms point to significant users of system resources. Swap analysis combines the individual recommendations of the workload manager and resource manager into a composite recommendation value (RV).

When a swap-in is considered desirable, IRARMSCI initiates the swap-in by scheduling IARSIN in the real storage manager (RSM) and passing a parameter list containing the address of the ASCB. If not enough available frames exist, IARSIN notifies the SRM through a SYSEVENT (SWINFL) to try the swap-in later. When a swap-out is considered desirable, IRARMCSO initiates the swap-out by posting the region control task (RCT), which calls IARSOUT, the RSM swap-out routine.

The following numbered paragraphs describe the sequential steps in the swap analysis algorithm. During step 2 and step 4 all domains are considered in numerical order. The algorithm terminates at the end of any step that causes at least one swap.

1. Defer swap-in. Based on the amount of storage available, the maximum number of deferred express users is swapped in (an express user is a user enqueued on a resource, a nonswappable user, or a user being cancelled). If enough storage exists for all deferred express users to be swapped in, then as many deferred non-express users as possible are also swapped in. If a deferred user cannot be swapped in and the swap has been evaluated for a period of ten seconds, the address space is placed in defer status on the WAIT queue. The system operator is then notified of the swapping failure and the storage requirements for the address space.

2. Unilateral swap-out. In each domain, the required number of user address spaces are swapped out to lower the MPL to its target value. In each domain, the user with the smallest recommendation values (RVs) are selected for swap-out.

3. Express swap-in. If any express users are swapped out, the express user who has been swapped out for the longest time is swapped in. if not enough storage exists for the swap-in, the steal thresholds are temporarily raised and the swap-in is deferred until the next invocation of swap analysis. If no user is swappable, the MPL in that domain temporarily exceeds its target.

4. Unilateral swap-in. In each domain, the required number of user address spaces are swapped in to raise the MPL to its target value. In each domain, users who have the largest RVs are selected for swap-in. If not enough storage exists for a swap-in, the steal thresholds are temporarily raised and the swap-in is deferred until the next invocation of swap analysis.

5. Exchange swap. For a domain with an MPL that matches its target, an in-real-storage address space and an out-of-real-storage address space are selected for exchange. The user in real storage with the lowest RV and the user out of real storage with the highest RV are selected. The exchange proceeds only is the difference in their recommendation values exceeds a certain limit (specified in RMPTXCHT). If an exchange is justified, the swap-out of the real in-real-storage user is initiated, and the swap-in of the out-of-real-storage user is deferred until a subsequent invocation of swap analysis.

The following method-of-operation diagrams describe important routines within IRARMCTL and IRARMSWP that relate to the swapping of address spaces.

- Swap analysis (IRARMCAP), analyzes users and, if it determines a swap desirable, request it.
- Control swap-out (IRARMCSO), initiates requested user swap-outs.
- Control swap-in (IRARMCSI), initiates requested user swap-ins.

- User selection for swap-in (IRARMCPI), finds the user with the highest recommendation value (RV) in its domain.
- User selection for swap-out (IRARMCPO), finds the user with the lowest recommendation value (RV) in its domain.
- User evaluation (IRARMCVL), calculates a recommendation value (RV) for a specific user.

RESOURCE USE ALGORITHMS

To improve overall system resource utilization, the resource use algorithms provide functions that promote the efficient management of storage, I/O, and processor capabilities. The resource monitor records system information that benefits the performance of these functions.

STORAGE MANAGEMENT

The following functions promote the effective management of storage:

- Page replacement
- Real page shortage prevention
- Auxiliary slot shortage prevention
- SQA shortage prevention
- Pageable real storage shortage prevention
- Storage load balancing

Page Replacement

This function (entry point IRARMPR1 in module IRARMSTM) maintains a current record of the frames that have gone unreferenced for the longest period of real time and the age of the oldest unreferenced frame in the system. Invoking the real storage manager's (RSM) UIC update routine, IARXUU, at periodic real time intervals, accomplishes this function. At each invocation, IARXUU increases the unreferenced interval count (UIC) for those frames unreferenced since the last IARXUU invocation. If IARXUU finds that an allocated frame was referenced during the last interval, it resets the UIC to zero. When the page replacement algorithm completes updating the UICs for all allocated frames, it saves the highest UIC in the system for use by the real page shortage prevention algorithm and the resource monitor algorithm.

Real Page Shortage Prevention

The SRM invokes this function (entry point IRARMMS2 in module IRARMST2) periodically to manage logically swapped users. If the available frame queue is below the logical swap available frame target, eligible logically swapped users are swapped out to meet the target. The function is also invoked when the available frame queue falls below the available queue LOW threshold (RCEAFCLO). When the real page shortage prevention algorithm is notified of a real page shortage, the SRM steals frames from all users and the system pageable area (SPA). It steals the oldest unreferenced allocated frames in the system, starting with those showing the highest UIC (as saved by the page replacement algorithm), until the count of available frames plus the count of the stolen pages exceeds the available frame queue OK threshold RCEAFCOK). If the installation specified real storage isolation in the IPS, the steal process does not violate the target working set size for the common area or for an address space protected by storage isolation. Thus, the SRM could steal a page not protected by storage isolation even when the page has a lower UIC than protected pages.

Auxiliary Slot Shortage Prevention

At periodic intervals, this function (entry point IRARMASM in module IRARMCTL) checks for two levels of auxiliary slot shortages. Reaching the first level threshold causes (1) the prevention of address space creation, (2) the swap-out of the batch user who is acquiring auxiliary storage slots at the greatest rate, and (3) the delay of newly-initiated jobs.

Reaching the second level threshold causes the swap-out of the batch user acquiring auxiliary storage slots at the next greatest rate. Messages to the operator indicate the occurrence of either shortage level. When this function detects an alleviation of auxiliary slot shortage, a message to the operator also indicates that fact. Address spaces can again be created, and those users swapped out to relieve the shortage regain eligibility for swap-in.

SQA Shortage Prevention

The virtual storage manager (VSM) invokes this function (entry point IRARME25 in module IRARMEVT) when it detects a shortage of system queue area (SQA) space. This function prohibits the creation of address spaces for the duration of the SQA shortage and notifies the operator of the existence and severity of the shortage. When VSM informs this function that the SQA shortage has been relieved, the operator receives a message to that effect and the system resumes the creation of address spaces.

Pageable Real Storage Shortage Prevention

The real storage manager (RSM) invokes this function (entry point IRARMMS2 in module IRARMST2) when the percentage of fixed frames (below 16 megabytes) to total frames (below 16 megabytes) exceeds a pre-defined limit. This function can:

1. Prohibit the creation of address spaces
2. Initiate a swap-out of the swappable user who owns the greatest number of fixed frames
3. Physically swap some or all logically swapped users
4. Delay newly-initiated jobs
5. Lower target MPLs for some or all domains

The operator is notified of the existence and severity of the pageable storage shortage and of the identify of the swapped users. When this function detects alleviation of the shortage, a message to the operator indicates that fact. Creation of address spaces can continue, and those address spaces swapped out to remedy the shortage regain eligibility for swap-in.

Storage Load Balancing

This set of routines monitors the real storage use of address spaces specified by the installation. If the routines consider real storage underused or overused, this function recommends address spaces for swap-in or swap-out based on their effect on the resource. Because it is scaled, the storage load balancer recommendation never exceeds one fifth of the highest possible workload level of the IPS currently in effect. This recommendation is multiplied by the MSO resource factor coefficient specified in the IEAOPTxx SYS1.PARMLIB member currently in effect.

I/O MANAGEMENT

The following functions promote the effective management of I/O resources:

- Device allocation
- I/O load balancer swap analysis
- Selective enablement

Device Allocation

This function makes device allocation decisions based on I/O load balancing considerations when a choice must be made among device candidates. IRARME28 in module IRARMEV2 includes this function. The device allocation function applies the following rules (in the order indicated) to the candidates:

1. If the request is for a tape, eliminate all candidates on ready devices (eliminate premounted tape drives) and on devices that contain passed volumes.

2. Choose the candidate on the logical path (the set of physical paths to a device or group of devices) with the lowest use. The level of use takes into account any data sets previously allocated to a device on the logical path.

3. Choose the direct access candidates with the lowest average delay per request. Choose the tape candidates based on the SELTAPE keyword specified in the IEAOPTxx SYS1.PARMLIB member currently in effect. The options are:

   - The next highest device number from the last allocated device (default)
   - The first device number in the candidate list
   - The lowest device number in the candidate list
   - Random selection from the candidate list

4. From a list of equal candidates, choose none at random.

5. Ensure that the selected candidate device has not been previously allocated to the same user.

I/O Load Balancer Swap Analysis

This function consists of a set of routines that monitor the I/O logical path use of certain users and recommend users for swapping based on the extent to which the swap-in or swap-out would correct a detected I/O imbalance. The I/O load balancer recommendation is scaled, never exceeding one-fifth of the highest possible work-load level of the IPS currently in effect. This recommendation is then multiplied by the IOC resource factor coefficient specified in the IEAOPTxx SYS1.PARMLIB currently in effect.

Selective Enablement

To permit the enablement of the fewest processor without incurring delays in I/O interruption activity, this function controls the number of processors enabled for I/O interruptions. Processors are enabled or disabled as necessary to maintain the interruption processing rate between thresholds specified in the IEAOPTxx SYS1.PARMLIB member currently in effect. To make its decision, the SRM divides the number of interruptions processed through the test pending interrupt (PI) instruction by the total number of I/O interruptions and compares the resultant percentage to the thresholds. If the percentage exceeds the upper limit, another processor (if available) will be enabled. If the percentage falls below the lower limit (and more than one processor is enabled), a processor will be disabled. In addition, if an enabled processor is taken offline or fails, this function enables another processor. IRARMIL5 and IRARMIL6 in module IRARMIOM perform this function.

CPU MANAGEMENT

The following functions promote effective processor management:

- Automatic priority group (APG) management
- ENQ/DEQ algorithm
- CPU load balancer swap analysis

Automatic Priority Group (APG) Management

The SRM maintains control over one or more contiguous sets of 16 priorities. Within each set of 16 priorities, the SRM uses two distinct priority schemes, enabling the installation to exercise precise control over the rules governing the dispatching of work. Users fall into one of two priority categories:

- Users with dispatching priorities based on the user's mean execution time before entering a wait state. This wait is defined as any time a task issues a WAIT, goes into page wait, or enters terminal wait, and no other ready tasks occupy this address space. Users who quickly release the processor obtain a high dispatching priority within the set of priorities governed by this scheme.

- Users with fixed priorities, defined by the installation.

ENQ/DEQ Algorithm

This function inhibits the swap-out of users who control (are enqueued on) resources that other system users need. Swap-out is prevented until the user releases the resource or the user has executed for the interval specified in the enqueue residence value (ERV) installation tuning parameter.

CPU Load Balancer Swap Analysis

This function consists of a set of routines that monitor the system-wide processor load. They recommend users for swapping when the system is underused or overused and when users exist who would improve the balance by being swapped in or out. The CPU load balancer's recommendation is scaled so that is never exceeds one-fifth of the highest workload level that the IPS makes possible. This recommendation is then multiplied by the CPU resource factor coefficient specified in the IEAOPTxx SYS1.PARMLIB member currently in effect.

RESOURCE MONITOR

At periodic intervals, this function samples indicators of system resource and domain contention, including response time objective (RTO) domains associated with TSO transactions. The resource monitor uses this information to calculate system and domain contention factors. Based on these factors and thresholds specified in the IPS or the IEAOPTxx SYS1.PARMLIB member, the resource manager might do the following to ensure maximum resource use:

- Adjust domain target MPLs
- Adjust the system MPL
- Adjust the system maximum think time for logical swapping
- Adjust the allocation of double frame pairs to provide an adequate supply on RSM's queue
- Adjust extended storage available thresholds

WORKLOAD MANAGEMENT

The workload manager is a collection of subroutines that perform three main functions:

1. Monitoring the rate at which system resources are being provided to individual address space.

2. Providing the swapping recommendations (based on installation specifications and resource use) that SRM control (IRARMCTL) requests.

3. Collecting data for Resource Measurement Facility (RMF) workload activity reporting.

Subroutines that perform the first two functions reside in the workload manager modules, IRARMWLM and IRARMWLS. The data collecting subroutines that perform the third function reside in the workload activity recording module, IRARMWAR. The workload manager does not control nonswappable address spaces and certain privileged system control program functions.

In providing swapping recommendations to SRM control, the workload manager affects the relative rates at which processing resources are provided to active address spaces. By comparing the resource use (service rate) for an address space with the installation performance specifications, the workload manager computes the workload level (the workload manager recommendation value) for the address space. SRM control uses this value as a swapping recommendation.

The workload activity recording (IRARMWAR) collects data for RMF when the installation requests workload activity reports. The workload manager and the SYSEVENT processor periodically invoke IRARMWAR to collect data for placement in the workload activity measurement table (WAMT). The workload reports are based on the data in the WAMT. An installation can analyze these reports to determine the appropriate installation performance specification parameters to meet its needs.

IRARMWLM and IRARMWLS perform the following tasks to support workload management functions:

- Swappable user evaluation.
- IN queue and OUT queue scanning-evaluating each non-privileged swappable user and assigning a current workload level.
- Individual user evaluation-evaluating a single non-privileged swappable user and assigning a current workload level.
- Transaction reporting for a subsystem-collecting transaction data provided by various subsystems for RMF reporting.
- User ready processing-initializing user control blocks and repositioning the user from the WAIT queue to the OUT queue so that the user becomes eligible for swap-in.

IRARMWAR performs the following tasks to support the data collection functions of the workload manager:

- WAMT initialization-updating the workload activity measurement table (WAMT) with information from the workload manager specification table (WMST).
- WAMT reinitialization-copying the WAMT data to a temporary buffer and then updating service values and workload levels.
- ICSM initialization-building the installation control specification map (ICSM). The ICSM is a mapping of performance group numbers with their corresponding installation control specification names. RMF uses the ICSM to include the names in its workload and transaction reports.

DEBUGGING HINTS

Various system and user indicators supply valuable information for diagnostic and debugging procedures. Useful fields exist in SRM data areas, user control blocks, and system diagnostic work areas.

USEFUL FIELDS

To assist in problem analysis, especially in enabled wait situations, it is helpful to understand how the SRM views the total system, as well as the individual address spaces. The counters and flags that pertain to the entire system are located in fields of data areas in IRARMCNS. The counters and flags that pertain to an individual address space are found in the fields of that user's OUCB.

System Indicators

The SRM control table (RMCT) is located at the start of module IRARMCNS. Its address is found in the field CVTOPCTP of the CVT data area (CVT+X'25C'). Generally, when the SRM is in control, register 2 holds the address of the RMCT. Through its fields, the RMCT points to other fields that contain information about the SRM's current processing. For the anchors and locations of the following fields, see the Debugging Handbook.

Useful system indicators include the following fields:

MCTAVQ1 This bit indicates that the count of available pages has fallen below the available frame queue LOW threshold (RCEAFCLO) value, so the real storage manager (RSM) has called the SRM to steal pages in order to increase the count of available pages. This condition is normal.

MCTSQA1 This bit indicates that the number of available SQA pages is critically low. Message IRA100I is issued to the operator and MCTSMS1 is set to 1 to indicate that the message has been issued.

MCTSQA2 This bit indicates that the number of available SQA pages has fallen below a second, more critical threshold than the one noted in MCTSQA1. Message IRA101I is issued to the operator and MCTSMS2 is set to 1 to indicate that the message has been issued.

MCTASM1 This bit indicates that the SRM has detected that less than 30% of all local auxiliary storage slots are available. The SRM has informed the operator of this fact through message IRA200I, and has taken appropriate action to relieve the shortage.

MCTAMS2 This bit indicates that the SRM has detected that less than 15% of all local auxiliary storage slots are available. The SRM has informed the operator of this critical slot shortage, through message IRA201I and has taken appropriate action to relieve the shortage.

MCTFAVQ This bit indicates that the SRM or the RSM has detected a pageable storage shortage condition. If the MCTLGPSS bit is set to 1, the SRM detected the shortage. That is, the sum of the number of fixed frames and the number of page I/O operations in progress to the page data sets exceeded the threshold set in MCCMAXFX. If the MCTPHPSS bit is set to 1, the RSM detected the shortage. That is, the count of fixed frames (RCETOTFX for all fixed frames, RCEBELFX for fixed frames below 16 megabytes) exceeded the threshold set in RCEMAXFX.

MCTLGAVQ This bit indicates that the SRM has increased the threshold values set in RCEAFCLO and RCEAFCOK (available frame queue thresholds) to initiate the frame stealing necessary to swap in an address space rather than wait for the SRM to issue an AVQLOW SYSEVENT.

MCTTWSS This halfword contains the target working set size for the common area. The SRM attempts to keep at least this minimum number of frames assigned to the common area.

RCVUICA This halfword indicates the average high unreferenced interval count (UIC) that the resource monitor detected at the last interval. Based on this value, the target MPL for a domain might be altered.

RCVCPUA This halfword indicates the average processor utilization that the resource monitor detected at the last interval. Based on this value, the target MPL for a domain might be altered.

RCVDPR This halfword indicates the demand paging rate that the resource monitor detected at the last interval. Based on this value, the target MPL for a domain might be altered.

RCVMSPP This halfword indicates the page delay time (in milliseconds) that the resource monitor detected at the last interval. Based on this value, the target MPL for a domain might be altered.

RCVFXIOP This halfword indicates the average percent of frames that are fixed or used for page I/O.

RCVMFXA This halfword indicates the average percent of frames eligible to be fixed that are actually fixed.

RMCAINUS This halfword indicates the count of address spaces (including non-swappable address spaces) currently residing in storage.

CCVENQCT This halfword indicates the number of address spaces currently residing in storage and marked nonswappable because they are holding ENQ resources that other address spaces want.

LSCTCNT This fullword indicates the number of address spaces currently logically swapped out because of terminal wait.

LSCTCNTW This halfword indicates the number of address spaces currently logically swapped out because of a long or detected wait.

Individual User Indicators

The register conventions generally used by the SRM to process individual user functions can help you locate important SRM control blocks.

Register	Contents
2	Address of the RMCT
3	Address of the RRPA
4	Address of the OUCB
5	Address of the ASCB (if used by the requested SYSEVENT)

The SRM user control block (OUCB) contains flags and counters that provide information about a specific user. There is one OUCB for each address space, pointed to by ASCBOUCB (ASCB+'90').

The following key fields define the characteristics of a specific user:

OUCBLSW — If this bit is set to 1, the address space is logically swapped out.

OUCBESSW — If this bit is set to 1, the user has been or will be swapped to extended storage.

OUCBESSS — If this bit is set to 1, the user is suspended for extended storage swap out.

OUCBMGSW — If this bit is set to 1, the user is being or has been migrated to auxiliary storage.

OUCBMWT — If this bit is set to 1, the SRM has detected that this user has not been dispatched but has been occupying storage for a processor-dependent period. The user is swapped out until the dispatcher informs the SRM that the address space has work to do.

OUCBAXS — If this bit is set to 1, the user has been swapped out of storage because the user's address space was obtaining auxiliary storage slots at the fastest rate in the system when an ASM slot shortage occurred.

OUCBENQ — This bit indicates that a different address space has tried to enqueue on a resource held by this address space. This user is treated as non-swappable for an installation-defined time period.

OUCBYFL — This field indicates how the address space is created:

- Bit 1 (OUCBSTT) - If set to 1, indicates that a START command created the address space.
- Bit 2 (OUCBLOG) - If set to 1, indicates that a TSO LOGON created the address space.
- Bit 3 (OUCBMNT) - If set to 1, indicates that a MOUNT command created the address space.

OUCBCSFS — If this bit is set to 1, the user is being delayed. Either (1) swap-in has failed for this address space because of a shortage of available storage, or (2) the user was swapped out because of a shortage of pageable frames.

OUCBFXS — This bit indicates that the address space is selected for swap-out to relieve a shortage of pageable storage. Either the SRM or the RSM can detect a shortage of pageable storage. If OUCBLGFX is set to 1, the SRM detected the shortage and the address space had more frames allocated to it than did any other swappable address space. If OUCBLGFX is turned off, the RSM detected the shortage and the address space had more fixed frames than an average address space.

OUCBDFSW — If this bit is set to 1, swap-in has been delayed. The RCEAFCLO and RCEAFCOK fields have been increased by the number of frames needed to complete the swap-in.

OUCBJSAS — If this bit is set to 1, there was an auxiliary storage shortage at the time of job select processing for this user. The user's initiation is delayed until the shortage is relieved.

OUCBJSFS If this bit is set to 1, there was a pageable frame shortage at the time of job select processing for this user. The user's initiation is delayed until the shortage is relieved.

OUCBSRC This field contains a code that describes why the user was last swapped out. The possible codes and their meanings are:

01 - Terminal output wait
02 - Terminal input wait
03 - Long wait
04 - Auxiliary input wait
05 - Real storage shortage
06 - Detected wait
07 - REQSWAP SYSEVENT issued
08 - ENQ exchange by swap analysis
09 - Exchange based on recommendation values provided by swap analysis
0A - Unilateral swap-out by swap analysis
0B - Transition swap

OUCBRDY This bit indicates that ready work became available for this address space, which was swapped out because of a wait. The address space, now able to execute, is therefore a candidate for swap-in.

OUCBTWSS This halfword contains the target working set size for the address space. The SRM attempts to keep at least this minimum number of frames assigned to the address space.

OUCBHOLD This fullword contains the number of outstanding HOLD SYSEVENTs issued by the address space. A non zero number results in quiesce processing, which turns the swap-out around and restores the address space.

Other Indicators

The SRM domain descriptor table can be useful in pinpointing a problem involving the SRM's MPL control. Mapping of the table can show causes of erratic response time, data about a user kept out of main storage, and other user and system information.

SRM SDWA CONTENTS

When either of the two functional recovery routines (FRRs) is entered (see "SRM Recovery Considerations" later in this section), the FRR fills in the SDWA fields before scheduling the SVC dump so that the dump matches the SYS1.LOGREC entry. However, in some cases, the FRR changes the abend code or reason code after the dump is scheduled and before the LOGREC entry is written, which means that the code in the LOGREC entry differs from the one in the dump.

The FRR also places problem determination data into the SDWA variable recording area (SDWAVRA) in key-length-data format using standard keys. (See the SDWAVRA area in the Debugging Handbook for a description of the keys). The following fields provide important information:

Key	Contents
VRAETF	The entry point address of either the SRM routine that was in control at the time of the error, or, if a subroutine was in control, the routine that called the subroutine.
VRARRP	A copy of the recovery routine parameter area (RRPA). The RRPA contains status information used on exit from the SRM and during the SRM recovery processing. The low-order byte in the first word of the RRPA contains the SYSEVENT code for the original entry to the SRM. The IRARRPA mapping macro maps the format of the RRPA.
VRAFP	A copy of the RRPA (as in field VRARRP) but with several entries cleared because they can be different for different invocations of the same function. The VRAFP is the footprint area the SRM uses to recognize duplicate problems.
VRALBL	The name of the routine that failed.
VRAOA	The original hexadecimal abend code. The FRR might have changed the code.
VRAAID	The address space identifier of the user for which the SRM was invoked.
VRACA	The caller's address if the SYSEVENT was branch-entered.

GENERAL INFORMATION USEFUL FOR SRM ANALYSIS

Information useful to consider when analyzing the SRM for diagnostic purposes include its recovery routines and serialization procedures.

SRM RECOVERY CONSIDERATIONS

The SRM uses two functional recovery routines (FRRs) that are located in IRARMERR. One FRR (recovery routine 1, RR1) gets control whenever an error occurs after the SRM is branch-entered by a routine that holds a lock higher in the lock hierarchy than the SRM lock. The other FRR (recovery routine 2, RR2) gets control whenever errors occur while the SRM is running with the SRM lock.

If it is suspected that the SRM is entering error recovery and a stop is necessary at the time of error, RMRR2INT, a subroutine common to both RR1 and RR2, obtains control.

Recovery routine 1 (RR1) retries if a retry routine exists. If no retry routine exists, or if the error recurs, RR1 percolates the error.

The specific actions taken for different errors are as follows:

- If the SRM issues the ABEND macro, or the operation has performed the system RESTART function, the error percolates.
- If the error occurs in the SRM workload activity recording routine, the RMF task abends. If the SRM is active in the same address space as the RMF task, the error percolates.
- If a translation or protection exception occurs in SYSEVENT processing, SRM recovery changes the abend code to X'15F'. The FRR validates the queue and status data that the SRM maintains and percolates the error.
- For other SRM errors, the FRR validates the queue and status data that the SRM maintains and retries the SRM routine that failed. If the error repeats and is associated with an action or an algorithm, another retry is attempted to bypass the routine in error. Otherwise, the error percolates.

Recovery routine 2 (RR2) first checks many special situations, such as the following:

- Is RMF active and should it be terminated?
- Is SET IPS active and should an abend code be converted?
- Is the OUCB valid and should an abend code be converted?

RR2 then retries if a retry routine exists. If no retry routine exists, or if the error recurs, RR2 percolates the error. The specific actions taken for errors are the same as those listed for RR1.

SRM (IRARMRR1 and IRARMRR2) suppresses dumps by using DAE (DUMP Analysis and Elimination). SRM does not suppress dumps that have an error code of X'5F', because they are not necessarily duplicate dumps. SRM uses the default DAE criteria as well as the module level (shown in VRALVL) and the routine name (shown in VRALBL) to determine whether a dump is a duplicate.

The issuing of most SYSEVENTs before SRM NIP processing (performed by IEAVNIP10) causes a direct return of control to the issuer; no SRM processing occurs.

SRM SERIALIZATION

All issuers of enabled, branch entry SYSEVENTs must hold the local lock when the SYSEVENT is issued.

The SRM obtains the SRM lock on all SYSEVENT entries except the following:

```
USERRDY X'04
SWOUTCMP X'0F'
RSMCNSTS X'16'
AVQLOW X'17'
AVQOK X'18'
SQALOW X'19'
SQAAOK X'1A'
```



```
SYQSCST X'23'
SYQSCCMP X'24'
HOLD X'32'
NOHOLD X'33'
DIRECTPO X'38'
SOUTSUSP X'45'
```

The issuer of any of these SYSEVENTs, with the exception of HOLD, NOHOLD, and DIRECTPO must be disabled when issuing the SYSEVENT because the SRM uses processor-related save areas while processing the SYSEVENTs. The issuer of HOLD, NOHOLD and DIRECTPO must also be in key 0. Because the SRM must be able to obtain the SRM lock when it is entered by means of a SYSEVENT, issuers of SYSEVENTs not listed above must not hold the SRM lock or any global lock when they issue the SYSEVENT.

Control Block Overview

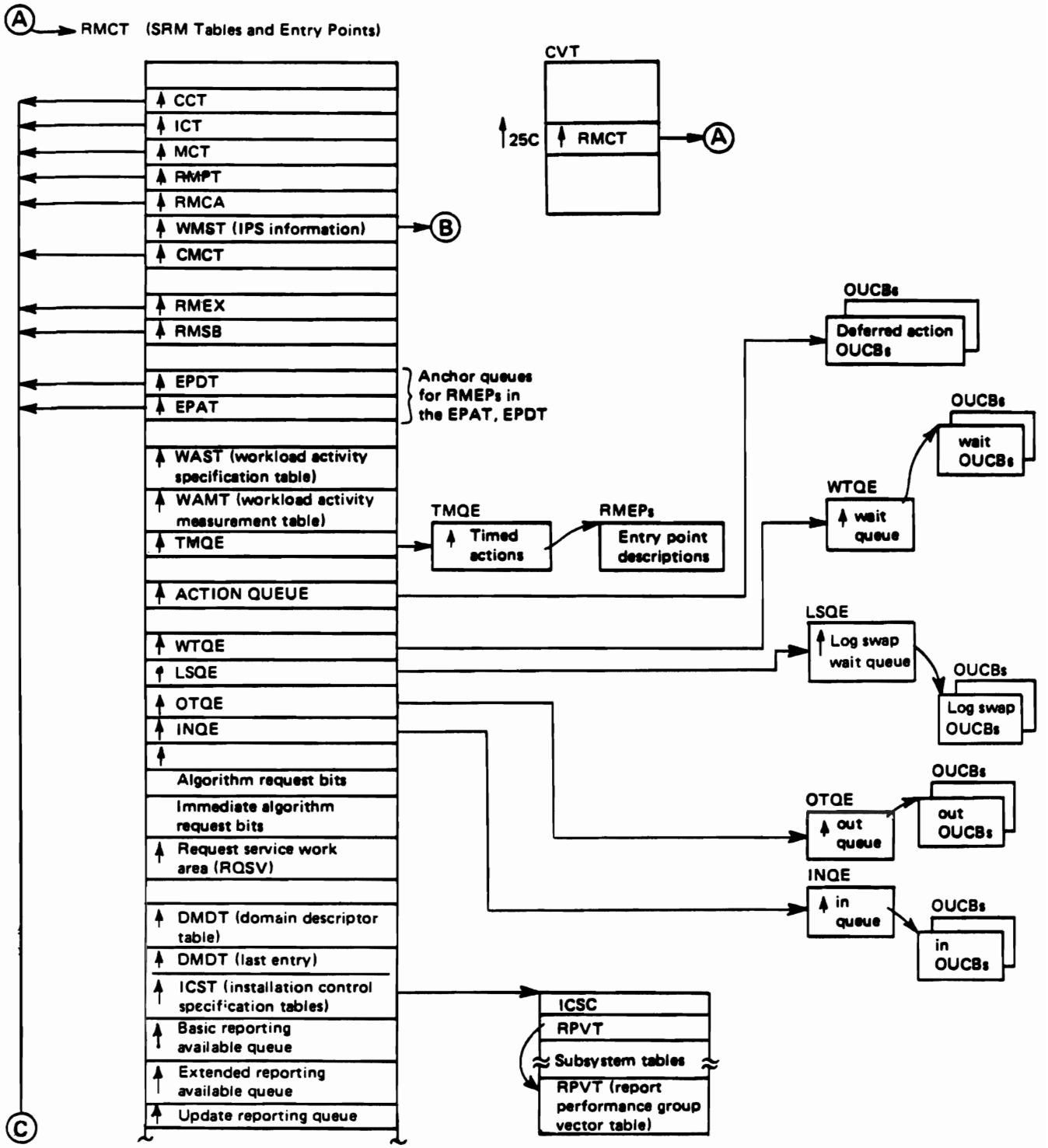


Figure 6 (Part 1 of 3). System Resources Manager (SRM) Control Block Overview

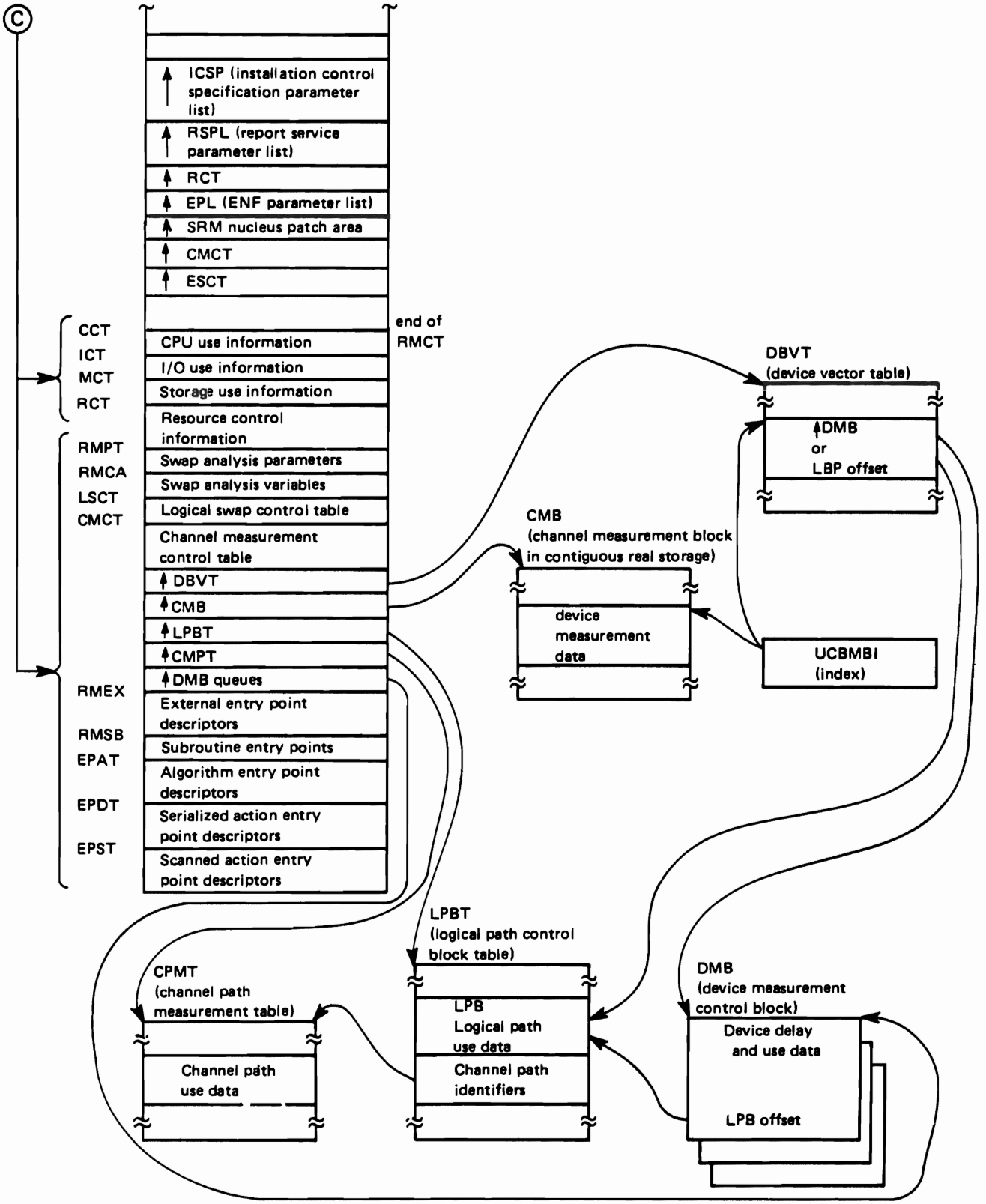


Figure 6 (Part 2 of 3). System Resources Manager (SRM) Control Block Overview

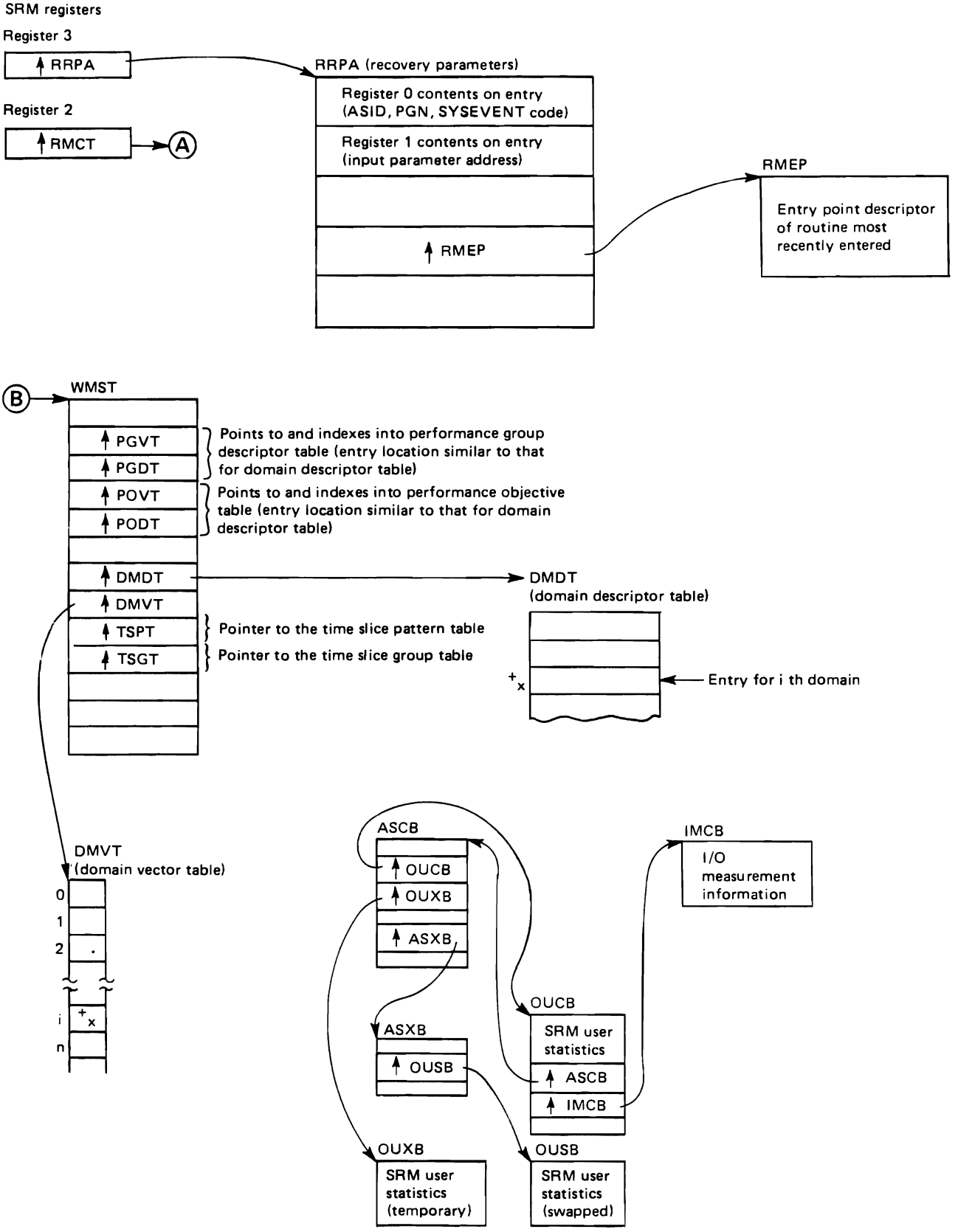


Figure 6 (Part 3 of 3). System Resources Manager (SRM) Control Block Overview

PROCESS FLOW

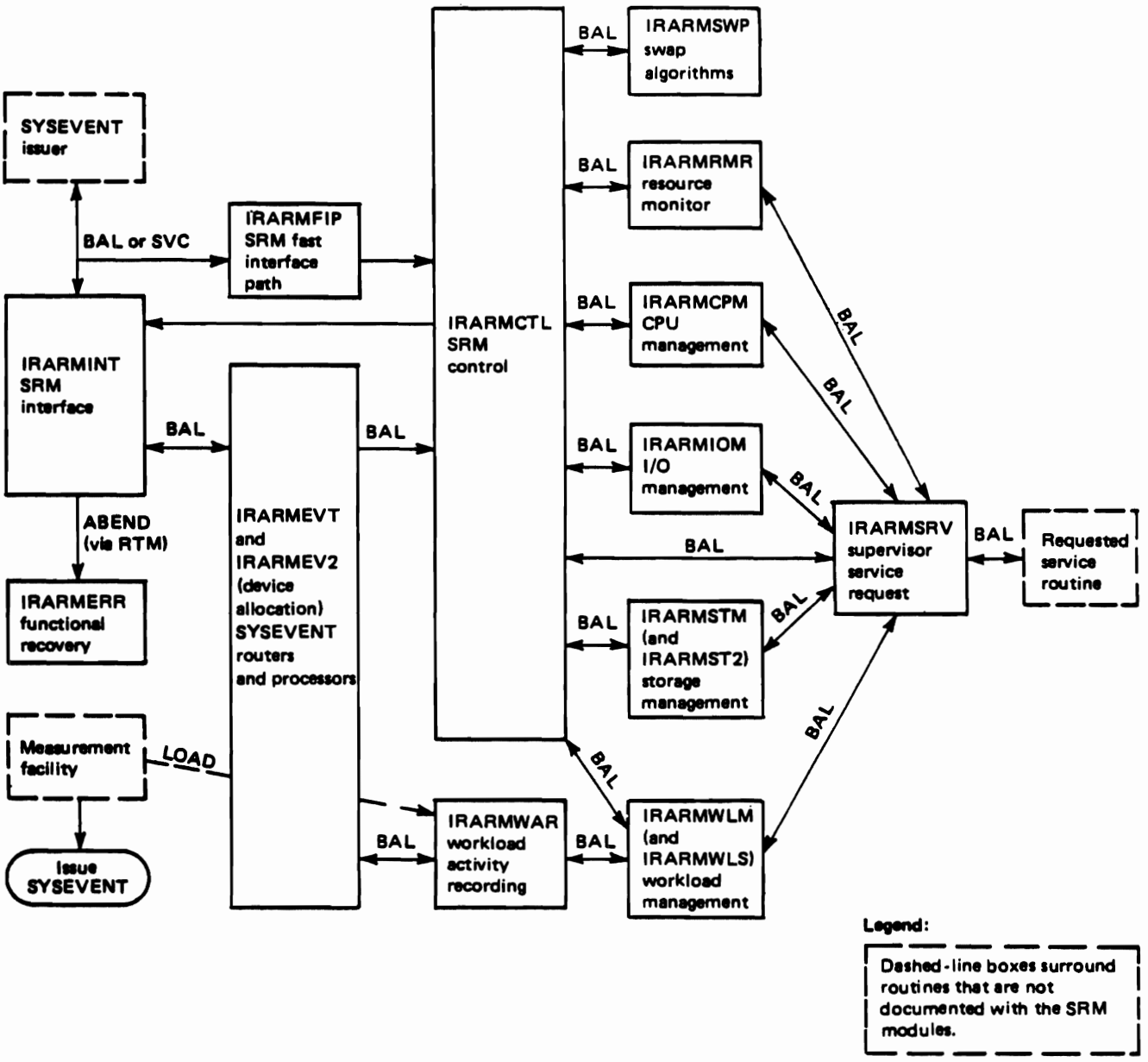


Figure 7. System Resources Manager (SRM) Mainline Processing Module Flow

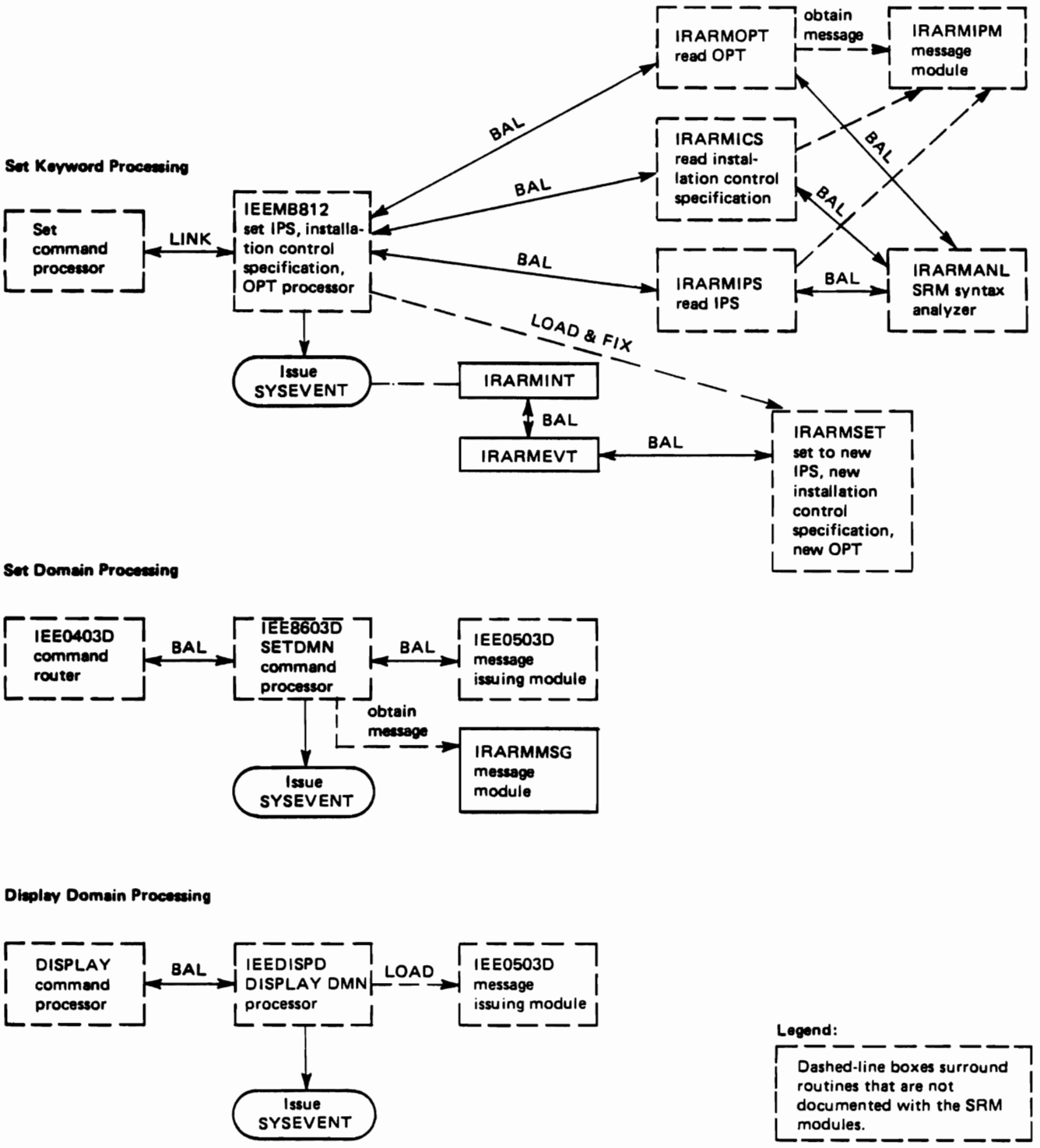


Figure 8. System Resources Manager (SRM) Command Processing Module Flow

METHOD OF OPERATION

This section contains logic diagrams for the modules in this component. The modules, grouped by function, are presented in the following order

Diagrams	Function
SRM-1 and SRM-2	Interface
SRM-3 and SRM-6	Service Routines
SRM-7	Find Performance Groups
SRM-8 through SRM-13	SRM Control
SRM-14 through SRM-17	Swap Analysis
SRM-18	Storage Management
SRM-19 through SRM-23	I/O Management
SRM-24 through SRM-26	Channel Measurement
SRM-27	CPU Management
SRM-28 and SRM-30	Resource Monitor
SRM-31 through SRM-33	Workload Management
SRM-34 and SRM-35	Workload Activity Recording

The diagrams use either hipo format or prologue format.

The following figure shows the symbols used in hipo format logic diagrams. The relative size and the order of fields in control block illustrations do not always represent the actual size and format of the control block.

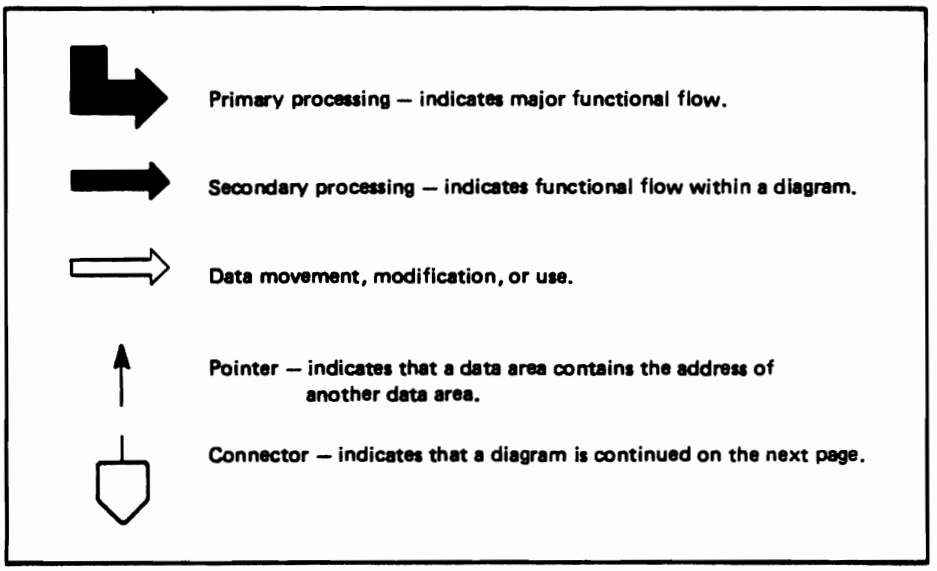


Figure 9. Key to Hipo Logic Diagrams

The prolog format diagrams contain detailed information that is broken down into four different headings. The four headings and the topics they document are:

Module Description, which includes:
- Descriptive name
- Function (of the entire module)
- Entry point names, which includes:
  - Purpose (of the entry point)
  - Linkage
  - Callers
  - Input
  - Output
  - Exit normal
  - Exit error, if any
- External references, which includes:
  - Routines
  - Data areas, if any
  - Control blocks
- Tables
- Serialization

Note: Brief SRM module descriptions are also included in MVS/Extended Architecture System Logic Library: Module Descriptions, which contains module descriptions for all the MVS/Extended Architecture components described in the System Logic Library.

Module Operation, which includes:
- Operation, which explains how the module performs its function.
- Recovery operation, which explains how the module performs any recovery.

Diagnostic aids, which provide information useful for debugging program problems; this includes:
- Entry point names
- Messages
- Abend codes
- Wait state codes
- Return codes for each entry point. Within each entry point, return codes might be further categorized by exit-normal and exit-error.
- Entry register contents for each entry point
- Exit register contents for each entry point

Logic Diagram, which illustrates the processing of the module, the input it uses, the output it produces, and the flow of control. Some modules do not have a logic diagram because the processing is sufficiently explained in the module description, the module operation, and the diagnostic aids sections. -- Fig 'SRM' unknown -- illustrates the graphic symbols and format used in the logic diagrams.

LOGICKEY - Key to the Logic Diagrams **STEP 01**

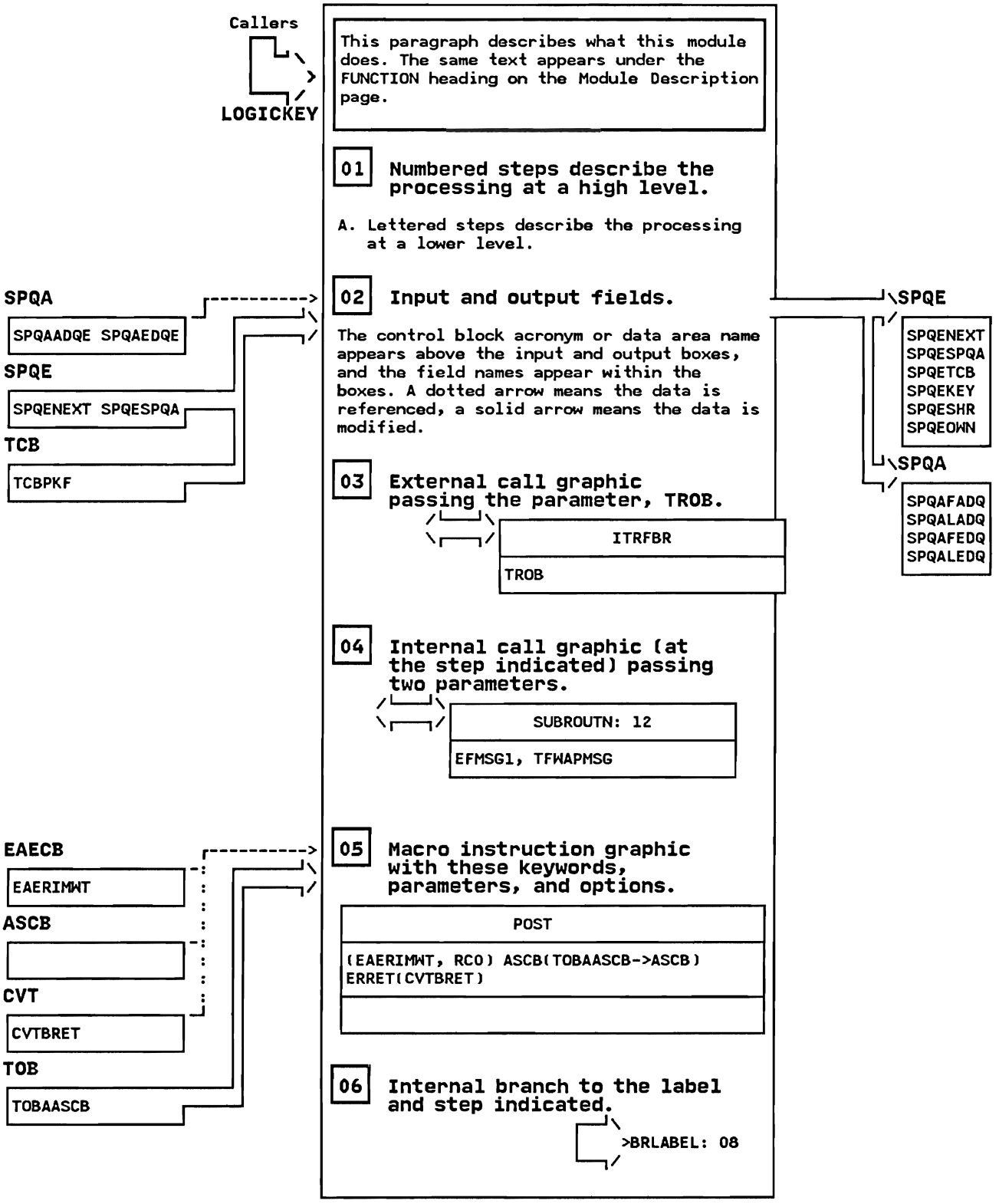


Figure 10 (Part 1 of 2). Key to Prologue Method-of-Operation Diagrams

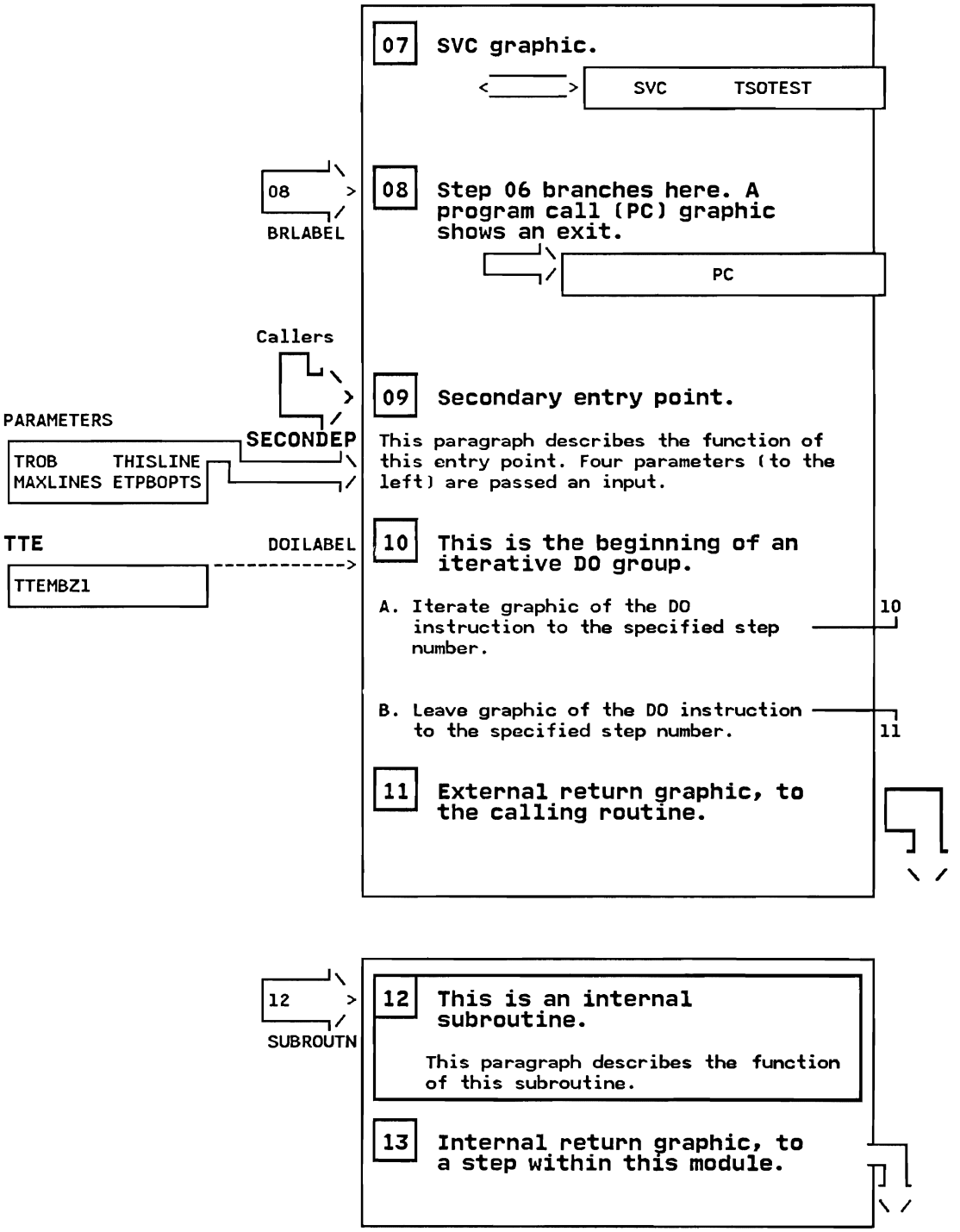


Figure 10 (Part 2 of 2). Key to Prologue Method-of-Operation Diagrams

IRARMINT MODULE ENTRY POINT SUMMARY

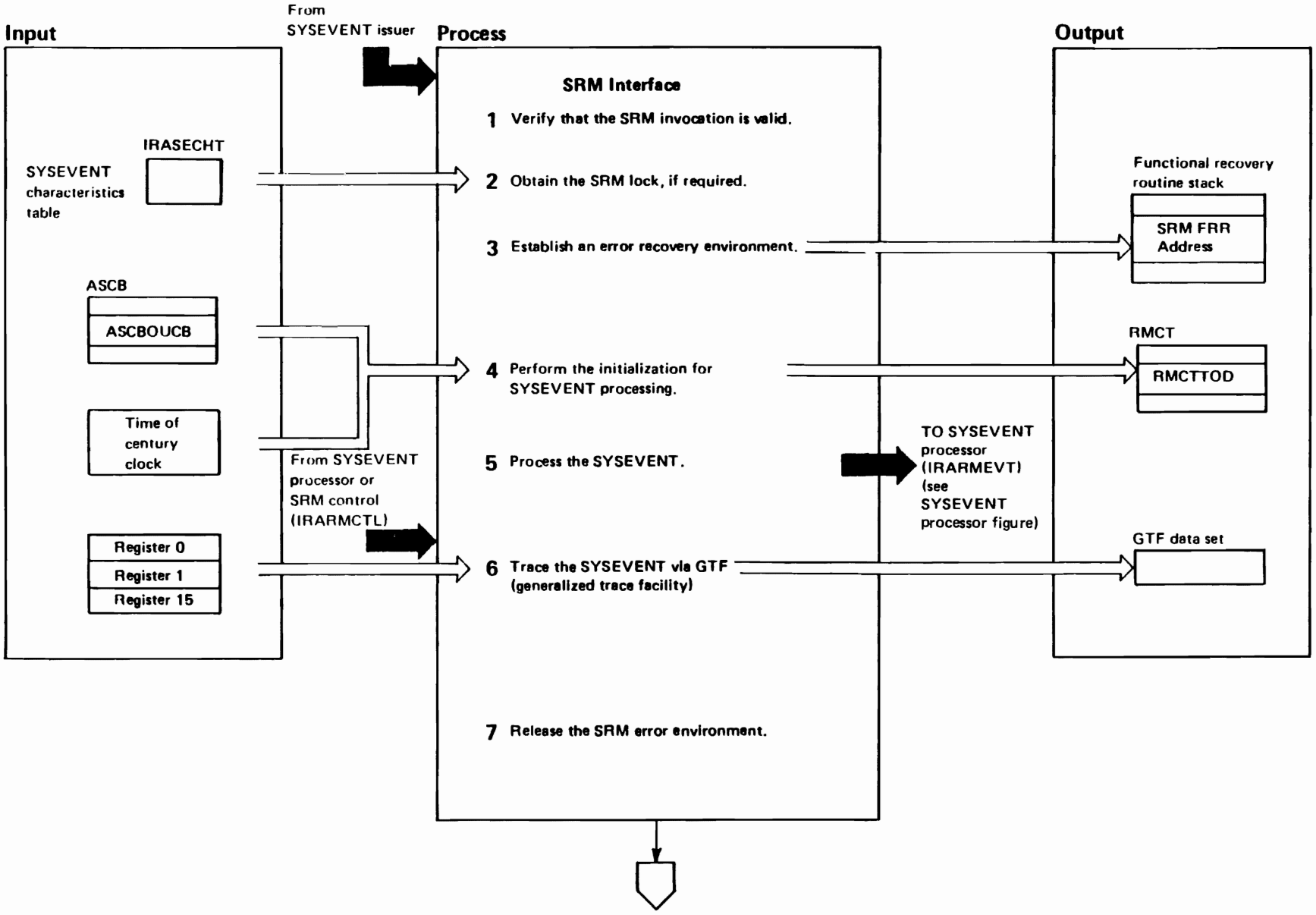
IGC095	SVC entry point to SRM.
IRARMI48	Branch entry point to SRM. Handle the internal SYSEVENT (30).
IRARMI01	Entry point from IRARMEVT or IRARMCTL. Return to the SYSEVENT issuer.
IRARMI10	Entry point to SRM. Abend a SRM user.

IRARMGLU MODULE ENTRY POINT SUMMARY

Fast interface path 24-bit entries:

IRARME05	TIME SYSEVENT
IRARME50	HOLD SYSEVENT
IRARME51	NOHOLD SYSEVENT
IRARME53	TRAXERPT SYSEVENT
IRARME54	TRAXFRPT SYSEVENT
IRARME55	TRAXRPT SYSEVENT
IRARME60	ICSCHK SYSEVENT
IRARME61	CMDEND SYSEVENT

DIAGRAM SRM-1. IRARMINT - SRM Interface (1 of 4)



Extended Description	Module	Label
The SRM interface receives control when a SYSEVENT macro instruction is issued. If the SRM interface receives control by means of a branch entry, IRARMINT or IRARMGLU saves the caller's addressing mode and passes control to IRARMI48 in 31-bit addressing mode. For a supervisor call entry, the SVC FLIH saves the caller's addressing mode and passes control to IRARMINT in 31-bit addressing mode. The SRM interface performs the locking necessary to ensure that SRM functions that must be serialized are not performed simultaneously on more than one processor. SRM requests the SRM lock unconditionally before passing control to the SYSEVENT processor. If the lock is held by another processor, the lock manager will spin waiting for the lock to be released. Otherwise, SRM acquires the lock and continues processing immediately.	IRARMGLU IRARMINT IRARMGLU	IRARMI00 IGC095 IRARME05
Fast interface path 31-bit entries save the caller's address mode and pass control to the corresponding 31-bit entries in IRARMFIP. (See the IRARMFIP diagram).		
1 For all SYSEVENTs that generate supervisor call entries to the SRM (SVC 95), except for SYSEVENT REQSERVC (26), the issuer must be authorized. Programs that issue SYSEVENTs 29 and 2A must have either APF authorization or must be assigned the non-swappable program property (PPTNSWP='1') in the program properties table. For all other SYSEVENTs, the user is considered, authorized only if executing in supervisor state or protection keys 0-7, or if authorized by APF (authorized program facility).	IRARMINT	IGC095
For SYSEVENTs that generate a branch entry to SRM, the issuer must be executing in protection key 0-7 and must be in supervisor state.	IRARMGLU	IRARMI48
2 The SYSEVENT characteristics table indicates, for each SYSEVENT entry, whether or not the SRM lock must be obtained for SRM serialization purposes.	IRARMINT	IRARM000

Extended Description	Module	Label
3 IRARMINT establishes a functional recovery routine to protect SRM from unexpected errors. It specifies the FULLXM option on the SETFRR macro so that, when recovery is necessary, RTM will attempt to restore the cross memory environment before invoking SRM's FRR. The processing performed for an error situation depends upon whether the SRM lock was held. (See Error Processing.)	IRARMINT	RMINT005
4 Before passing control to the SYSEVENT processor, IRARMINT obtains a pointer to the SRM user control block (OUCB) corresponding to the input ASID (address space identifier). For SYSEVENT MEMCREAT, there is not yet an OUCB (an OUCB is obtained by IRARMEVT if no resource shortages exist). This module obtains the current time-of-century and formats it for SRM use. The time-of-day clock value is stored and shifted 22 bits to the right, and SRM uses the rightmost 32 bits of the resulting value. Therefore, SRM constants representing time are in units of 1024 microseconds (approximately 1 millisecond). IRARMINT issues a 75F abend if SRM timer pop does not occur in the expected interval.	IRARMINT	IRARM001
5 The interface invokes the SYSEVENT processor to initiate the appropriate processing.	IRARMEVT	
6 If GTF is active, the SRM interface produces a GTF trace record using the HOOK macro. This record includes: • Register 0 (as input except that the ASID is placed here even when it is not included as input) • Register 1 (as input, with the addition of possible return indicators that might overlay input data) • Register 15 (containing any necessary return code in byte 3)	IRARMINT	
7 IRARMINT removes the address of the SRM FRR from the system FRR stack.	IRARMINT	IRARMI01

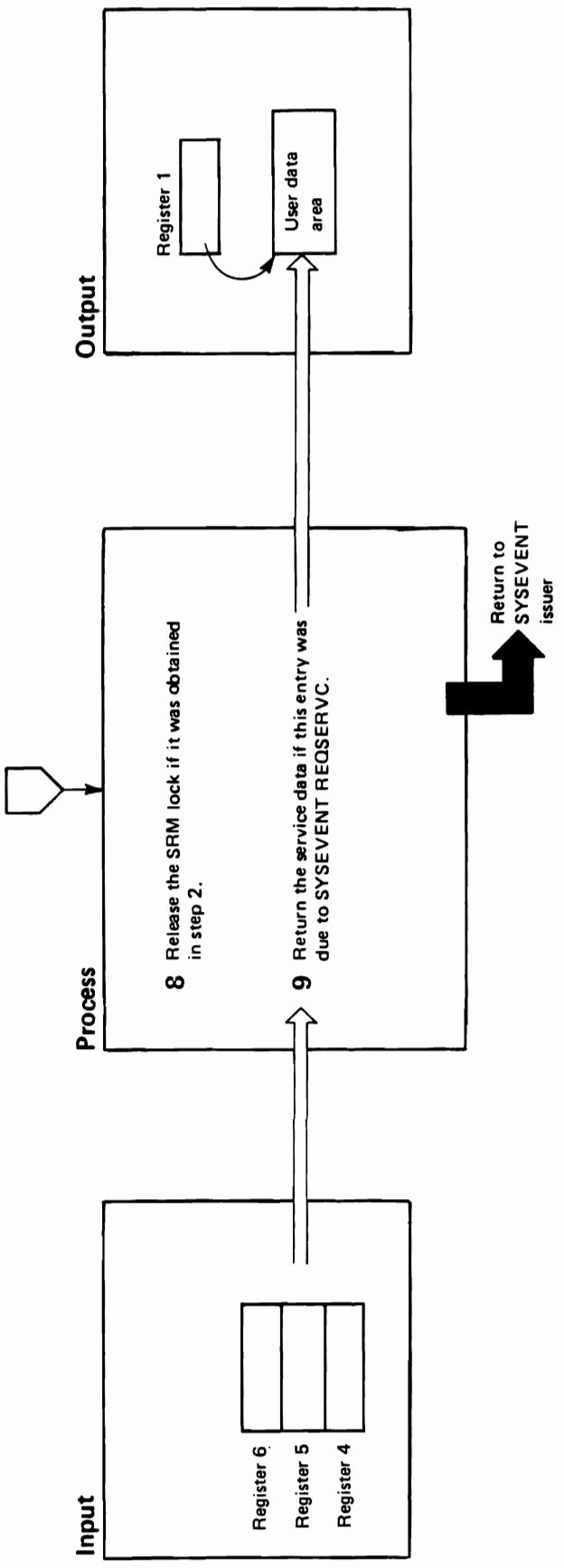


DIAGRAM SRM-1. IRARMINT - SRM Interface (3 of 4)

Extended Description	Module	Label
8 If the invoking routine did not already hold a lock higher in the locking hierarchy than the SRM lock (except for SYSEVENTS SYQSCST and SYQSCCMP), IRARMINT releases the SRM lock.		RMINT010
9 To prevent disabled page faults and an invalid SRM invocation, and to insure system integrity, the service data is stored while not holding the SRM lock, and in the user's protection key.		
If the SRM interface received control by means of a branch entry, IRARMINT restores the caller's addressing mode and returns control.		
Error Processing		
The issuer of a SYSEVENT is abnormally terminated (ABEND code '15F'X) if:	IRARMINT	
• An invalid ASID or SYSEVENT code was supplied (reason code 4).		
• The program was not authorized to issue the SYSEVENT (reason code 8).		
• A page fault occurred in referencing a data area assumed to be fixed (reason code 12).		
• The program did not have the correct storage key for storing into a parameter data area (reason code 16).		
• The SRM lock was held on entry to the SRM (reason code 20).		
A SYSEVENT issuer is terminated (ABEND code '25F') if SRM determines that a system failure has resulted in the loss of data used by the SRM in controlling an address space. Similarly, the set IPS task will be terminated (ABEND code '25F') when SRM receives an error occurring during SRM processing relating to a set to new IPS command.		

Extended Description	Module	Label
A functional recovery routine (FRR) provides the error recovery environment for SRM processing. When an error occurs during SRM processing (or when an error occurring in a routine invoked by SRM has been passed back (percolated) to SRM), the recovery/termination manager establishes the cross memory environment that existed at the time the SRM FRR was created and gives control to the SRM FRR. If SRM was operating without holding the SRM lock when the error occurred, error processing will consist of making one attempt at retrying the failing routine; a second failure will result in the error being passed to the previous routine in the FRR stack. If SRM was operating under the SRM lock when the error occurred, the FRR will perform queue validation before making an attempt at retrying the failing routine; queue validation consists of verifying that the three OUCB queues are properly chained (re-chaining where necessary), and that OUCBs OUXBs (user control block extensions), and OUSBs (user swappable blocks) exist and are valid, where they are required. Likewise, the pointers between the ASCBs and OUCBs is checked. Where it is necessary to create a new OUCB or OUXB, a bit is set in the OUCB to indicate that the data reflected in these newly created blocks may not be valid.	IRARMERR	
		IRARMRR1
		IRARMRR2
		RMRR2VLD
When errors occur during SRM locked processing, retry action depends upon whether the error occurred during SYSEVENT related or non-SYSEVENT related processing. For SYSEVENT-related processing, one retry will be attempted. Subsequent failure will result in the error being passed to the previous routine in the FRR stack. For non-SYSEVENT-related processing (that is, processing which SRM control was driving), one retry of the failing routine will be attempted. A second error will cause an attempt to bypass the twice failing routine. Subsequent errors will result in the error being passed to the previous routine in the FRR stack.		

DIAGRAM SRM-2. IRARMFIP - SRM Fast Interface (1 of 2)

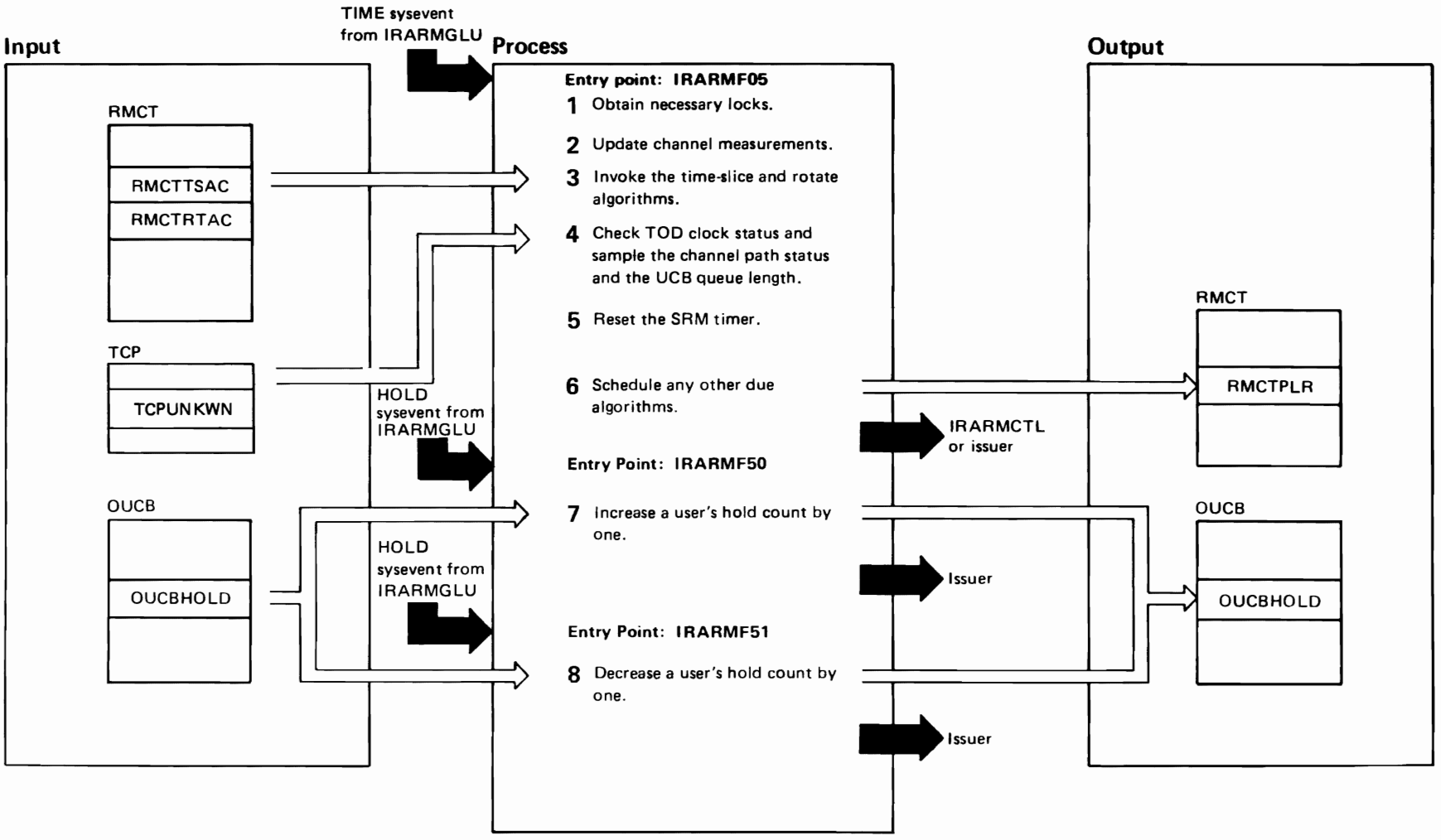


DIAGRAM SRM-2. IRARMFIP - SRM Fast Interface (2 of 2)

Extended Description	Module	Label
The SRM fast interface path receives control when special pre-defined SYSEVENTs are issued. This path does not validity check the issuer, nor does it perform the normal generalized entry processing.		
1 For the TIME SYSEVENT, obtain the SRM lock and establish an FRR.	IRARMFIP IRARMFIP IRARMFIP	IRARMF05 IRARMRT1 IRARMTS1
2 • If the hardware supports the collection of processor controller data and it has been dropped, IRARMFIP updates the channel measurements. If none have been dropped for 10 minutes, fast interface path tries to restart the facility. • If the processor controller data is not provided and sampling is due, IRARMFIP gets data from channel path status and UCB queue lengths.	IEAVEAC0	
3 If time slicing control is active and the algorithms are due, IRARMFIP invokes them and then performs any priority changes required.		
4 If the TOD clock is invalid, IRARMFIP calls IRARMMSP to stop channel measurements. If the TOD clock is valid and channel measurements are not active, mark channel measurements as pending start. If sampling is due, invoke IRARMCPS and IRARMDBS to measure the current channel path activity and the SSCH queue length for the UCBs.	IRARMFIP IRARMFIP	IRARMCPS IRARMDBS
5 Queue a TQE for the minimum of 1 SRM second or the time when either a channel measurement sample or the time slice algorithm is due to execute again.	IRARMSRV	IRARMI05
6 If other SRM algorithms are due, turn on the corresponding bits in RMCTALR and exit to the SRM control routine. If no other algorithms are due, return to the invoker through the SRM standard interface.	IRARMCTL IRARMINT	 IRARMI01

Extended Description	Module	Label
7 For the HOLD SYSEVENT, increase a full word counter in the OUCB via compare and swap. Return to the issuer.	IRARMFIP	IRARMF50
8 For the NOHOLD SYSEVENT, decrease the count of outstanding hold events for a user and return.	IRARMFIP	IRARMF51
Error Processing		
The TIME SYSEVENT establishes an FRR to process errors that occur in the processing of the SYSEVENT.	IRARMERR	IRARMRR2
The recovery routine performs queue verification and then attempts retry. The retry routine sets a new timer value and exits through the standard SRM interface.		RMRR2VLD IRARMXIP IRARMI01
The remaining SYSEVENTs in this module do not establish recovery environments.		

Entry Point Summary

IRARMFIP module – interface routines – entry point summary

IRARMF05 – Invoke time slice algorithm; reset SRM's timer; schedule other algorithms due.

IRARMTS1 – Time-slicing algorithm.

IRARMCPS – Channel path sampling routine.

IRARMDBS – UCB queue length sampling routine.

IRARMXIP – Retry routine for fast SYSEVENTs.

IRARMF50 – Increase a user's HOLD count.

IRARMF51 – Decrease a user's HOLD count.

IRARMSRV MODULE ENTRY POINT SUMMARY

IRARMI02	ASCBCHAP entry point.
IRARMI03	UIC steal processing entry point.
IRARMI04	Obtain or free SQA or ESQA storage.
IRARMI05	Requeue SRM TQE routine.
IRARMI06	Cross-memory post entry point.
IRARMI07	Swap SRB schedule routine.
IRARMI08	I/O enablement SRB schedule routine.
IRARMI09	RECORD entry point.
IRARMI11	ENF signal entry point.
IRARMI16	Cross-memory post routine.
IRARMI17	Post ECB while holding the local lock.
IRARMI20	Fixed frame count routine.
IRARMI21	Obtain or free double frame pair routine.
IRARMSFT	Invoke safety net check.

IRARMERR MODULE ENTRY POINT SUMMARY

IRARMRR1	Functional recovery for globally locked entries (entries to SRM in which the SRM lock could not be obtained). Retry the failing SRM routine when possible, otherwise percolate the error.
IRARMRR2	Functional recovery for non-global locked entries (entries to SRM in which the SRM lock was obtained). Validate queue and cleanup. Copy error-related data into the fixed and variable parts of the SWDA. Issue an SDUMP and record the log record. Retry the failing routine if possible; otherwise, percolate the error. If the failing routine was an action or algorithm routine and retry has been attempted once, bypass the failing routine and retry the control routine algorithm twice if necessary, before percolation.
RMRR2RTY	Return to RTM indicating retry.
RMRR2PER	Return to RTM indicating percolation.
RMRR2INT	FRR initialization.
RMRR2VLD	Validate the control blocks.
RMRR2GST	Release the dispatcher lock in order to call IRARMI04.
RMRR2CKQ	Verify the location of an OUCB.
RMRR1VFB	Verify addresses.
RMRR2REQ	OUCB enqueue routine entry point.
RMRR2SPR	Return with the return code in register 15.

DIAGRAM SRM-3. IRARMSRV - SRM Service Routine (1 of 14)

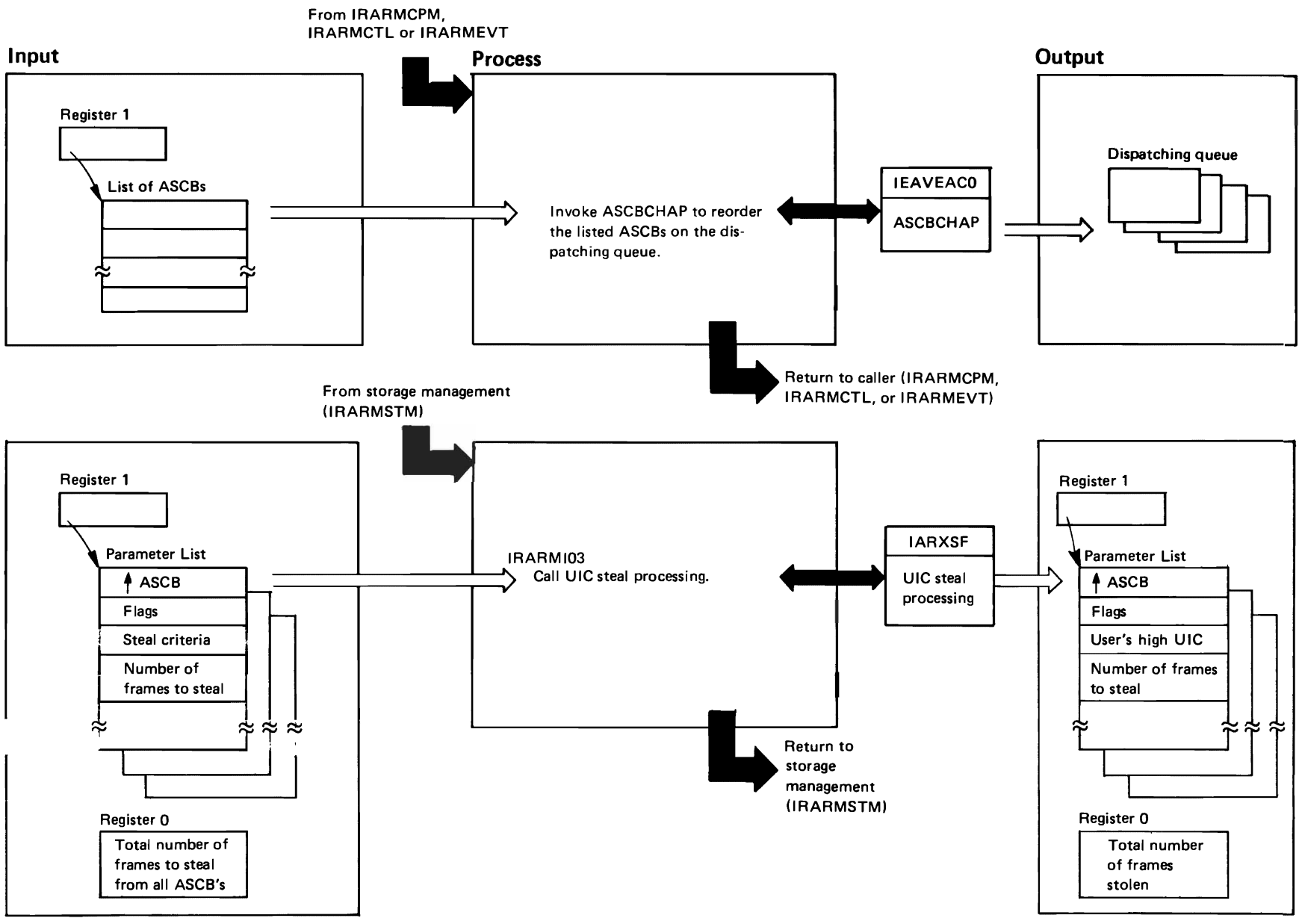


DIAGRAM SRM-3. IRARMSRV - SRM Service Routine (2 of 14)

Extended Description	Module	Label (or Segment)
This module is a collection of several independent routines that act as interfaces between SRM and various system services.	IRARMSRV	IRARMSRV
IRARMI02 Reposition the listed ASCBs in the ASCB dispatching queue to reflect their new dispatching priorities.	IRARMSRV IEAVEAC0	IRARMI02
IRARMI03 Steal pages from users included in the list that have frames with UICs that meet the steal criterion.	IRARMSRV IARXSF	IRARMI03

DIAGRAM SRM-3. IRARMSRV - SRM Service Routine (3 of 14)

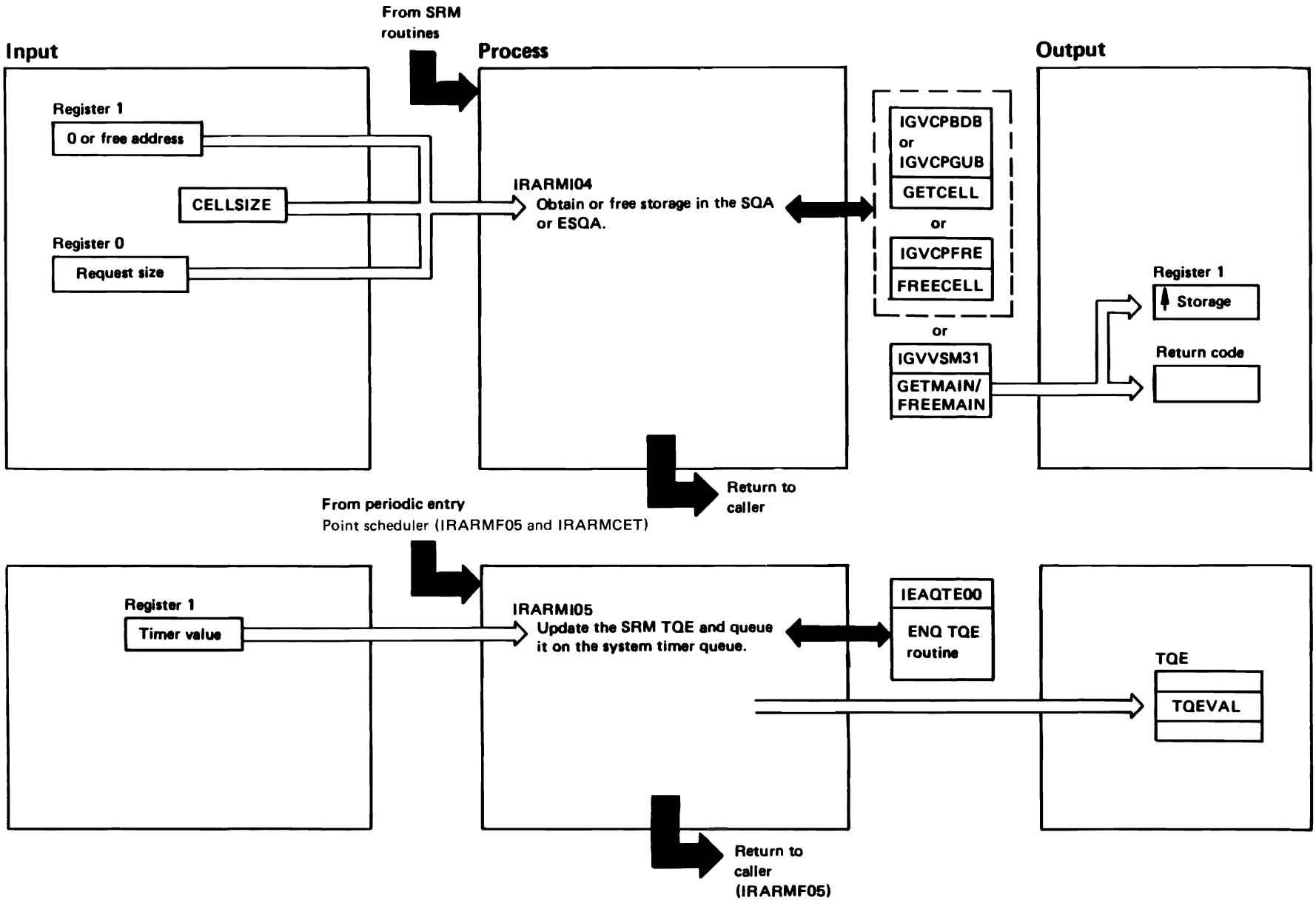
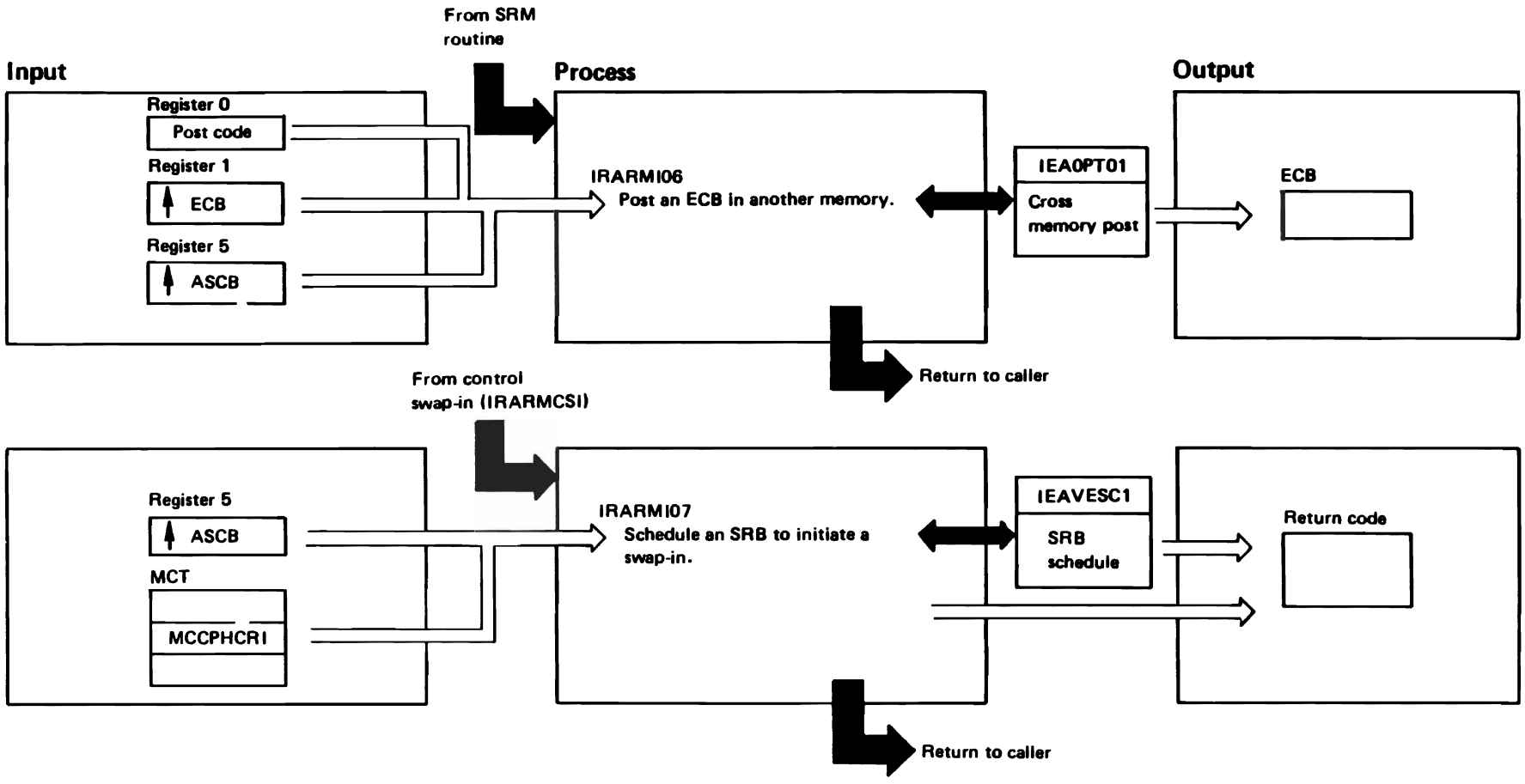


DIAGRAM SRM-3. IRARMSRV - SRM Service Routine (4 of 14)

Extended Description	Module	Label (or Segment)
IRARMI04 Obtain or free SQA or ESQA storage either from a cell in SRM's cellpool or from other available SQA or ESQA.	IRARMSRV IGVCPBDB or IGVCPGUB or IGVCPFRE IGVVSM31	IRARMI04
IRARMI05 Store a new timer value in the SRM TQE and queue the TQE on the system timer queue.	IRARMSRV IEAVRTI0 IEAVRTI0	IRARMI05 IEAQTD00 IEAQTE00

DIAGRAM SRM-3. IRARMSRV - SRM Service Routine (5 of 14)



Extended Description	Module	Label (or Segment)
IRARMI06 The swap-out routine (IRARMCSO) uses this entry point to post the region control task. The entry point is also used by SYSEVENTs 0F (SWOUTCMP), 0C (QSECST), and 0D (QSCECMP) to post an address space that issued a TRANSWAP or REQSWAP SYSEVENT. If an error is encountered during the cross memory post, the error routine (IRARMXPE) receives control in 31-bit addressing mode. If the swap-out routine invoked IRARMI06, IRARMXPE sets up an FRR, and attempts to clean up. If the swap-out routine did not invoke IRARMI06, IRARMSRV issues an abend with completion code X'55'. Processing this abend results in an SDUMP and termination of the waiting address space.	IRARMSRV IEAVSY50 IRARMSRV	IRARMI06 IEA0PT01 IRARMXPE
IRARMI07 The swap-in routine (IRARMCSI) uses this entry point to obtain and schedule RSM routines in the master scheduler's address space. For a suspended swap-out redrive, it schedules RSM routine IARSURST. For a swap-in or migration swap, it schedules RSM routine IARSIN. To enable the RSM to determine if swapping the user in might cause a pageable frame shortage, IRARMI07 also passes the shortage threshold (MCCPHCRI) to RSM in the SRB.	IRARMSRV IEAVESPM IAVSESC0	IRARMI07 IEAVSPM1 IEAVESC1

DIAGRAM SRM-3. IRARMSRV - SRM Service Routine (7 of 14)

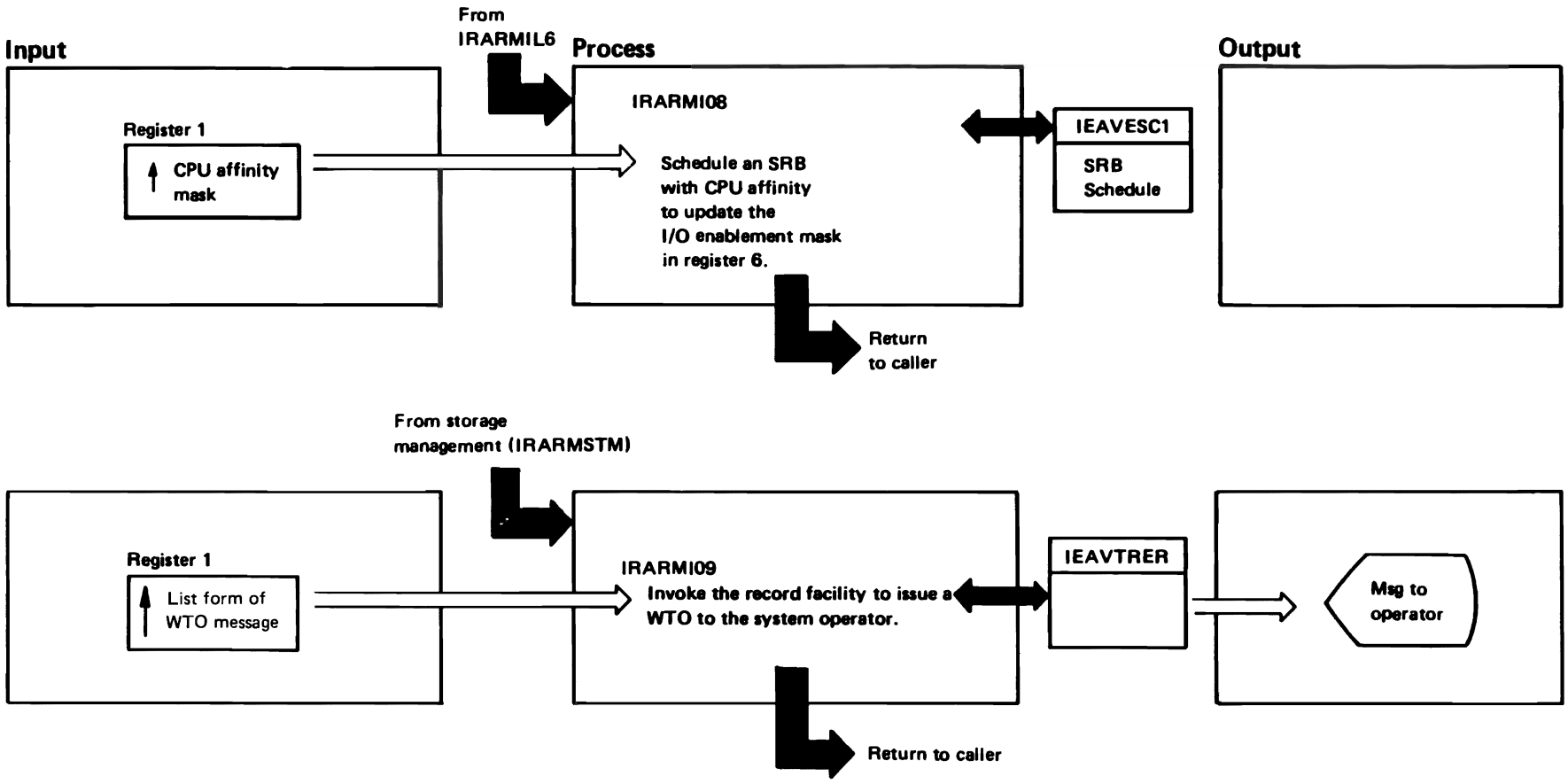
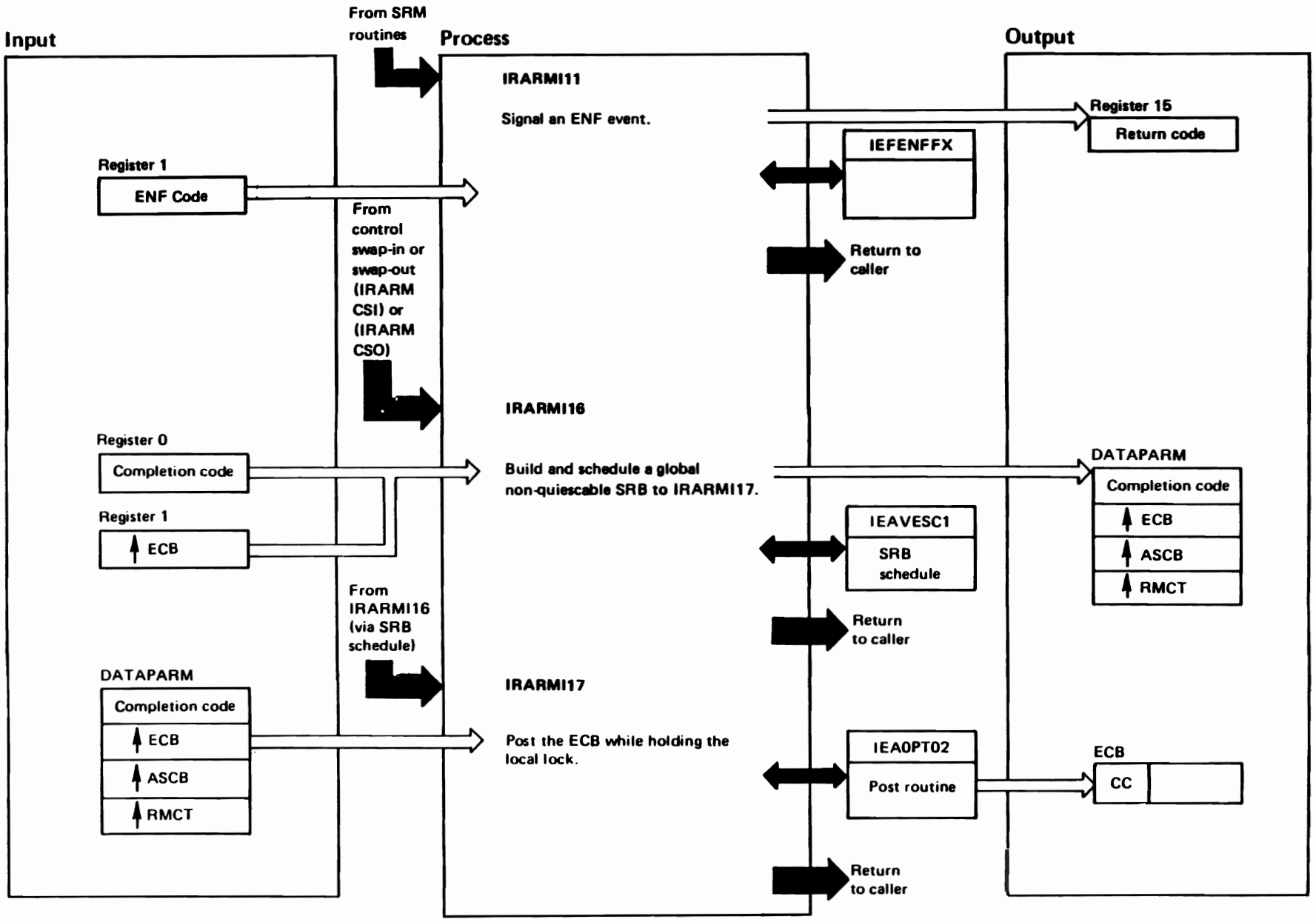


DIAGRAM SRM-3. IRARMSRV - SRM Service Routine (8 of 14)

Extended Description	Module	Label
IRARMI08 Schedule an SRB with CPU affinity to cause IOS to change the I/O enablement mask in control register 6. IOS uses the information found in the PCCA.	IRARMSRV IOSRISCE	IRARMI08 IOSRISCE
IRARMI09 Invoke the RECORD facility to issue a WTO to the system operator console, because the requesting SRM routine holds the SRM lock and cannot issue a WTO.	IRARMSRV IEAVTRER	IRARMI09 IEAVTRER

DIAGRAM SRM-3. IRARMSRV - SRM Service Routine (9 of 14)



Extended Description	Module	Label
IRARMI11 This entry point is an interface to ENF (event notification facility) to signal an event. IRARMI11 issues the signal based on the ENF code passed to IRARMI11 in register 15. The return code from ENF is passed through to the caller of IRARMI11 in register 15.	IRARMSRV IEFENFFX	IRARMI11 IEFENFFX
IRARMI16 Swap-in or swap-out control uses this entry point to post the quiesce ECB. The routine accomplished a cross-memory post by reposting the ECB, if no wait or post has been done, or by scheduling an SRB to run in the address space to be swapped in. The SRB entry point is IRARMI17. The SRB and the data area containing the post parameters are built in an unused area of the OUXB (OUXBFLDS).	IRARMSRV IEAVESC0 IRARMSRV	IRARMI16 IEAVESC1 IRARMI17
IRARMI17 This entry point receives control under an SRB scheduled by IRARMI16. The routine calls POST to post the indicated ECB and then restores the OUXB fields.	IRARMSRV IEAVSY50	IRARMI17 IEA0PT02

DIAGRAM SRM-3. IRARMSRV - SRM Service Routine (11 of 14)

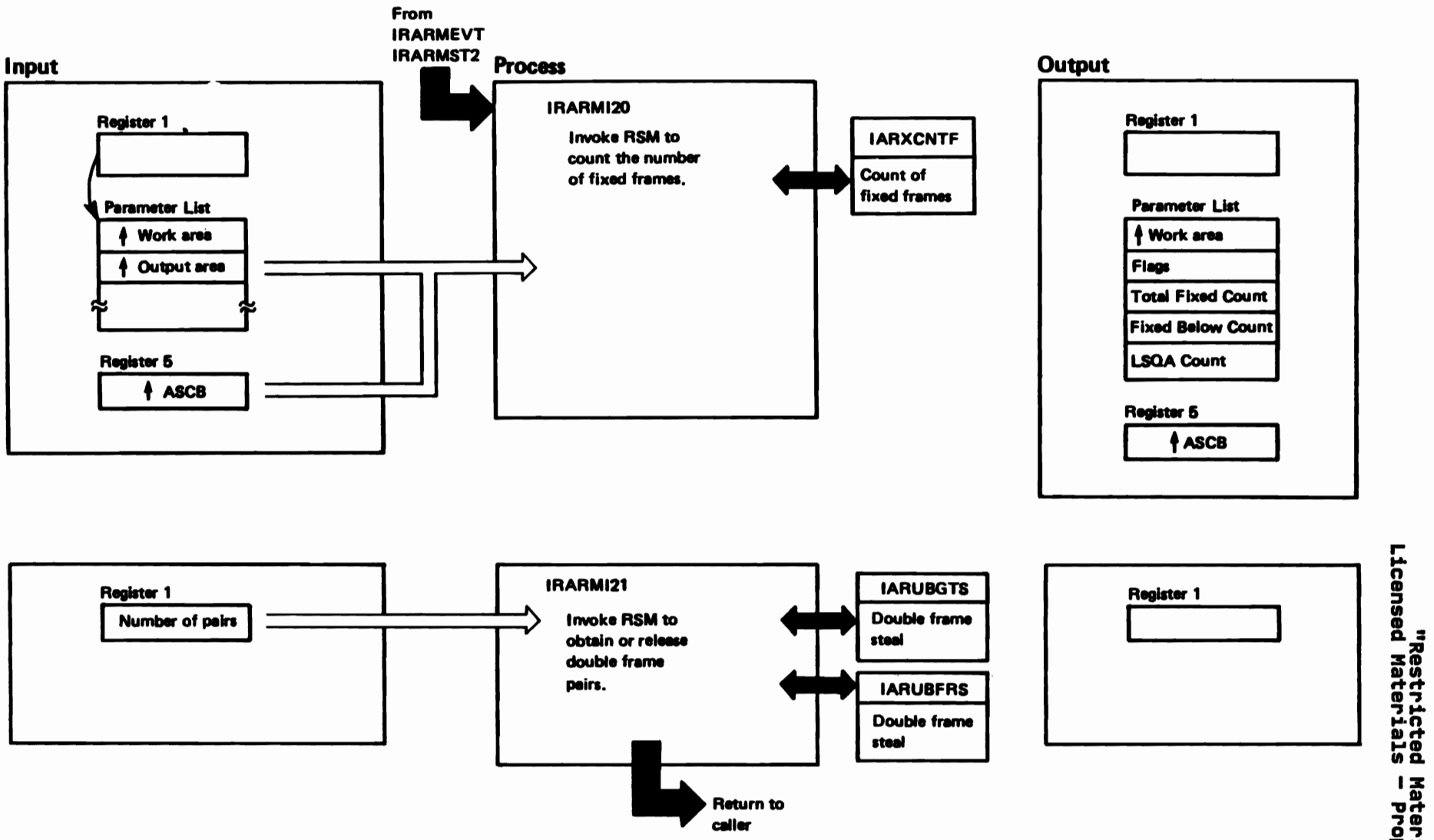


DIAGRAM SRM-3. IRARMSRV - SRM Service Routine (12 of 14)

Extended Description	Module	Label
IRARMI20		
Invoke RSM to obtain the fixed frame count for an address space or the common storage area (CSA).	IRARMSRV IARXC	IRARMI20 IARXCNTF
IRARMI21		
Invoke RSM to manage the double frame queue that segment table allocation uses. Set register 1 to a negative number if the call requests that the RSM free frames. Set register 1 to a positive number if the call requests that the RSM obtain frames.	IRARMSRV IARUB	IRARMI21 IARUBFRS IARUBGTS

DIAGRAM SRM-3. IRARMSRV - SRM Service Routine (13of 14)

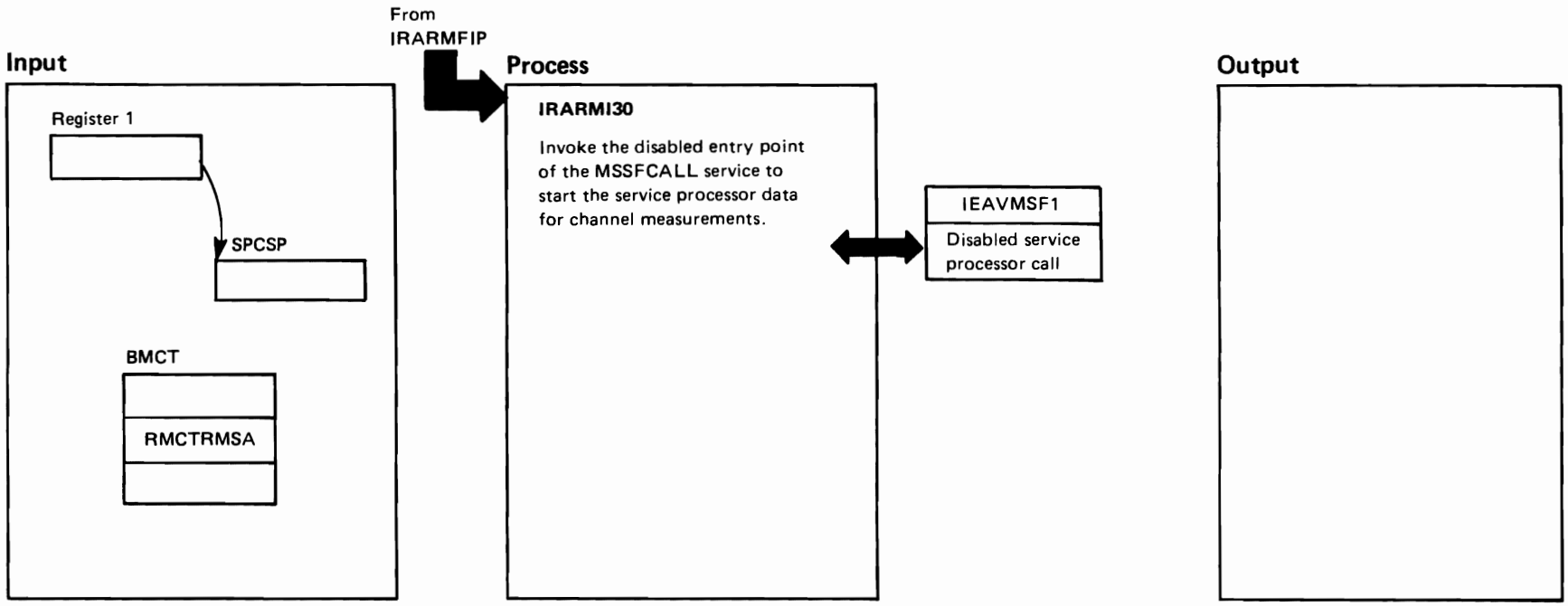


DIAGRAM SRM-3. IRARMSRV - SRM Service Routine (14 of 14)

Extended Description	Module	Label
IRARMI30		
Once started, the service processor collects data and periodically updates the SADB with the data. SRM obatins the	IRARMSRV IRARMSRV	IRARMI30
Once started, the service processor collects data and periodically updates the SADB with the data. SRM obtains the control block, SADB.	IRARMSRV IEAVMSF	IRARMI30 IEAVMSF1

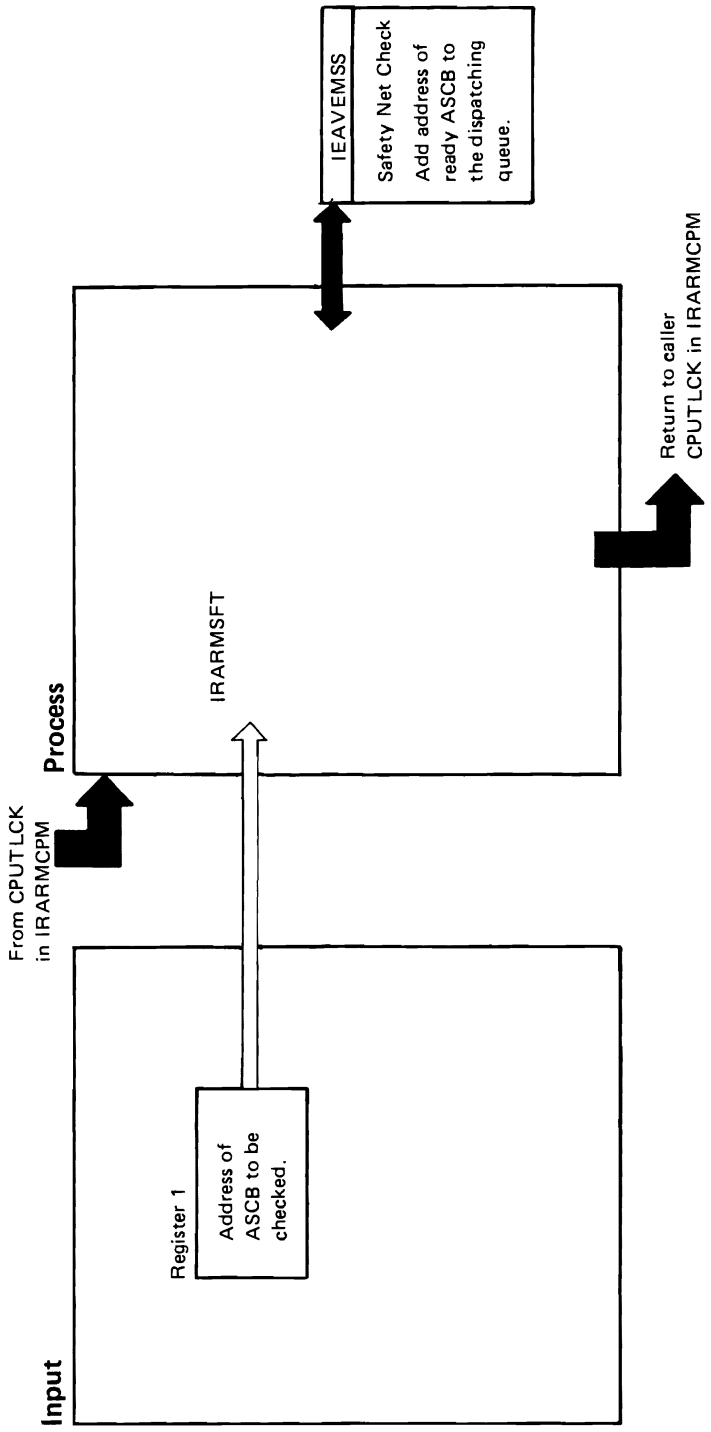


DIAGRAM SRM-4. IRARMSFT - Invoke Safety Net Check (1 of 2)

DIAGRAM SRM-4. IRARMSFT - Invoke Safety Net Check (2 of 2)

Extended Description	Module	Label
The CPUTLCK subroutine in IRARMCPM determines that either: • A ready address space is not on the dispatching queue or • a processor is not fully utilized which may indicate that the dispatcher cannot locate the ready address space. The CPUTLCK subroutine invokes IEAVEMSS, entry point in IRARMSFT. IEAVEMSS locates and adds the ready address space to the dispatching queue. Control returns to the CPUTLCK subroutine.		

IRARMEVT MODULE ENTRY POINT SUMMARY

IRARMRVT	SYSEVENT processor. Begin to process the indicated SYSEVENT.
IRARMXVT	SYSEVENT retry. Prepare a retry of SYSEVENT that incurred a system error.
IRARMDEL	Synchronize memory delete processing.
IRARMAPG	Check the automatic priority group (APG) status and reposition the user on the ASCB dispatching queue.
IRARMSWI	Place the OUCB of the newly swapped-in user on the IN queue and preform related processing.
IRARMLDB	Create or free device-related control blocks for I/O load balancing.
IRARMUXB	Synchronize OUXB deletion at swap-out completion time.

IRARMFPG MODULE ENTRY POINT SUMMARY

IRARMFPG finds the performance group to associate with a unit of work.

DIAGRAM SRM-5. IRARMFPG - Find Performance Groups (1 of 7)

FUNCTION:
This routine is called whenever SRM needs to determine the proper performance groups to associate with a unit of work. The unit of work may be a batch jobstep, a TSO user, TSO command, started task, or a transaction from the transaction report SYSEVENTs. IRARMFPG determines the proper performance groups according to the Installation Control Specification (ICS) PARMLIB member currently in effect. The subsystem, userid, transaction name, transaction class, and account number associated with the unit of work are supplied by the caller as input parameters.

ENTRY POINT: IRARMFPG

PURPOSE: See function

LINKAGE:
- REG 0 - address of output area FPGO
- REG 1 - ICSP ICS input parameter list
- REG 2 - RMCT address
- REG 3 - RRPA address (recovery parameters)
- REG 4 - OUCB address of memory with which to associate the SYSEVENT
- REG 5 - ASCB address of memory with which to associate the SYSEVENT
- REG 6 : REG 12 - irrelevant
- REG 13 - address of current stack frame
- REG 14 : REG 15 - irrelevant

CALLERS:
- IRARME08 - JOBSELCT SYSEVENT PROCESSOR EVT
- IRARME10 - INITATT SYSEVENT PROCESSOR EVT
- IRARME27 - EASINIT SYSEVENT PROCESSOR EVT
- IRARME63 - CMDSTART SYSEVENT PROCESSOR EVT
- IRARMWM8 - TRANSACTION REPORTING ALGORITHM WLS
- IRARMSIC - "SET ICS" ROUTINE SET

INPUT:
- Register 1 points to either
  1) The parameter list (ICSP) which contains subsystem name, transaction name, USERID, transaction class, and (optionally) flags, a performance group number, and an account number. OR
  2) A shorter parameter list identical to the first four fields of the ICSP, but not containing the flag, optional performance group, or the account number. This short list is used by transaction reporting to obtain RPGNs only. (Callers IRARMWM8 and IRARME63.)

  If a complete ICSP is present, the high order bit of register 1 is turned on. This also shows that a control performance group and a report performance group is requested.

- Register 0 points to the area mapped by the FPGO which will contain the CPGN and RPGNs. The caller must clear the RPGN fields prior to calling FPG if checking for valid data is desired on return.

- Tables pointed to by RMCTICST->ISCC->ICSS-ICSE

- OUCBACNT indicates account number on JOB statement

OUTPUT:
- Register 15 contains the return code.
- The output area (pointed to by register 0 on input) contains the performance groups found by FPG, if any.

DIAGRAM SRM-5. IRARMFPG - Find Performance Groups (2 of 7)

IRARMFPG - MODULE DESCRIPTION (Continued)

The return code tells which have been found.

EXIT NORMAL:

EXIT ERROR: No error exits

EXTERNAL REFERENCES:

ROUTINES: None

CONTROL BLOCKS:
- ICSP - Installation Control Specification Parameters
- ICSC - Installation Control Specification Common
- ICSS - Installation Control Specification Subsystem
- ICSE - Installation Control Specification Entry
- OUCB - SRM User Control Block, one per address space
- RMCT - SRM Control Table, anchor for all SRM

SERIALIZATION: SRM LOCK

DIAGRAM SRM-5. IRARMFPG - Find Performance Groups (3 of 7)

IRARMFPG - MODULE OPERATION

This routine scans the tables that describe the Installation Control Specification (ICS), looking for a match on the subsystem name in the input parameters. Finding this, if Report Performance Groups Numbers (RPGN) were
requested, FPG looks for a match on the transaction names, userids, transaction classes, and account number for that subsystem and assigns a report performance group from the tables for every match found. If a Control Performance Group Number (CPGN) was requested, the CPGN corresponding to the highest matching level name is used. If no match is found on the transaction name, account number, userid or class, a subsystem default CPGN is always assigned. RPGN fields are neither set to zero nor defaulted if no specific match is found.

RECOVERY OPERATION: IRARMFPG participates in component wide recovery.

DIAGRAM SRM-5. IRARMFPG - Find Performance Groups (4 of 7)

IRARMFPG - DIAGNOSTIC AIDS

ENTRY POINT NAME: IRARMFPG

MESSAGES: None

ABEND CODES: None

WAIT STATE CODES: None

RETURN CODES:

- 0 SUBSYSTEM MATCH FOUND.
  REQUESTED PGNS RETURNED.
- 4 SUBSYSTEM MATCH FOUND.
  NO REPORT PERFORMANCE GROUPS FOUND.
- 8 SUBSYSTEM MATCH NOT FOUND.

REGISTER CONTENTS ON ENTRY: Irrelevant

REGISTER CONTENTS ON EXIT: Irrelevant

DIAGRAM SRM-5. IRARMFPG - Find Performance Groups (5 of 7)

IRARMFPG - Find Performance Group Subroutine

STEP 01

- IRARME08 - JOBSELCT SYSEVENT PROCESSOR EVT
- IRARME10 - INITATT SYSEVENT PROCESSOR EVT
- IRARME27 - EASINIT SYSEVENT PROCESSOR EVT
- IRARME63 - CMDSTART SYSEVENT PROCESSOR EVT
- IRARMWM8 - TRANSACTION REPORTING ALGORITHM WLS
- IRARMSIC - "SET ICS" ROUTINE SET

IRARMFPG

This routine is called whenever SRM needs to determine the proper performance groups to associate with a unit of work. The unit of work may be a batch jobstep, a TSO user, TSO command, started task, or a transaction from the transaction report SYSEVENTs. IRARMFPG determines the proper performance groups according to the Installation Control Specification (ICS) PARMLIB member currently in effect. The subsystem, userid, transaction name, transaction class, and account number associated with the unit of work are supplied by the caller as input parameters.

ICSCEND ICSPSUBO
ICSSNAME ICSSSUBS
ZERO

01 Scans the tables that describe the Installation Control Specification (ICS) looking for a match on the subsystem name in the input parameters.

ICSSPTR

ICSPSUBO ICSSNAME
RCNOSYBS RCNORPGN

02 If no subsystem match is found, set return code.

RTNCOD
FLAGS

INPREG ICSPACTL
OUCBACNT ICSSACTN

03 Depending on caller's request and the current environment, sets flags indicating which control and/or report performance groups are needed.

RPGNTRXN
RPGNUSRD
RPGNCLS
FPGOCPGN
CPGNTRXN
CPGNUSRD
CPGNCLS
CPGNACTN
RPGNACTN

ICSSRPG
ICSSRPGN RCSUCCES

04 Set the subsystem RPGN if there is one.

FPGOSRPG
RTNCOD

05 Searches each table until the requested PGNs are found. When a report performance group is found, resets the RPGN flag. When a control performance group is found, reset all CPGN flags, thus terminating the search of all remaining tables for a CPGN.

A. Obtains performance group numbers from the transaction name table.

FINDTRXN

DIAGRAM SRM-5. IRARMFPG - Find Performance Groups (6 of 7)

IRARMFPG - Find Performance Group Subroutine **STEP 05B**

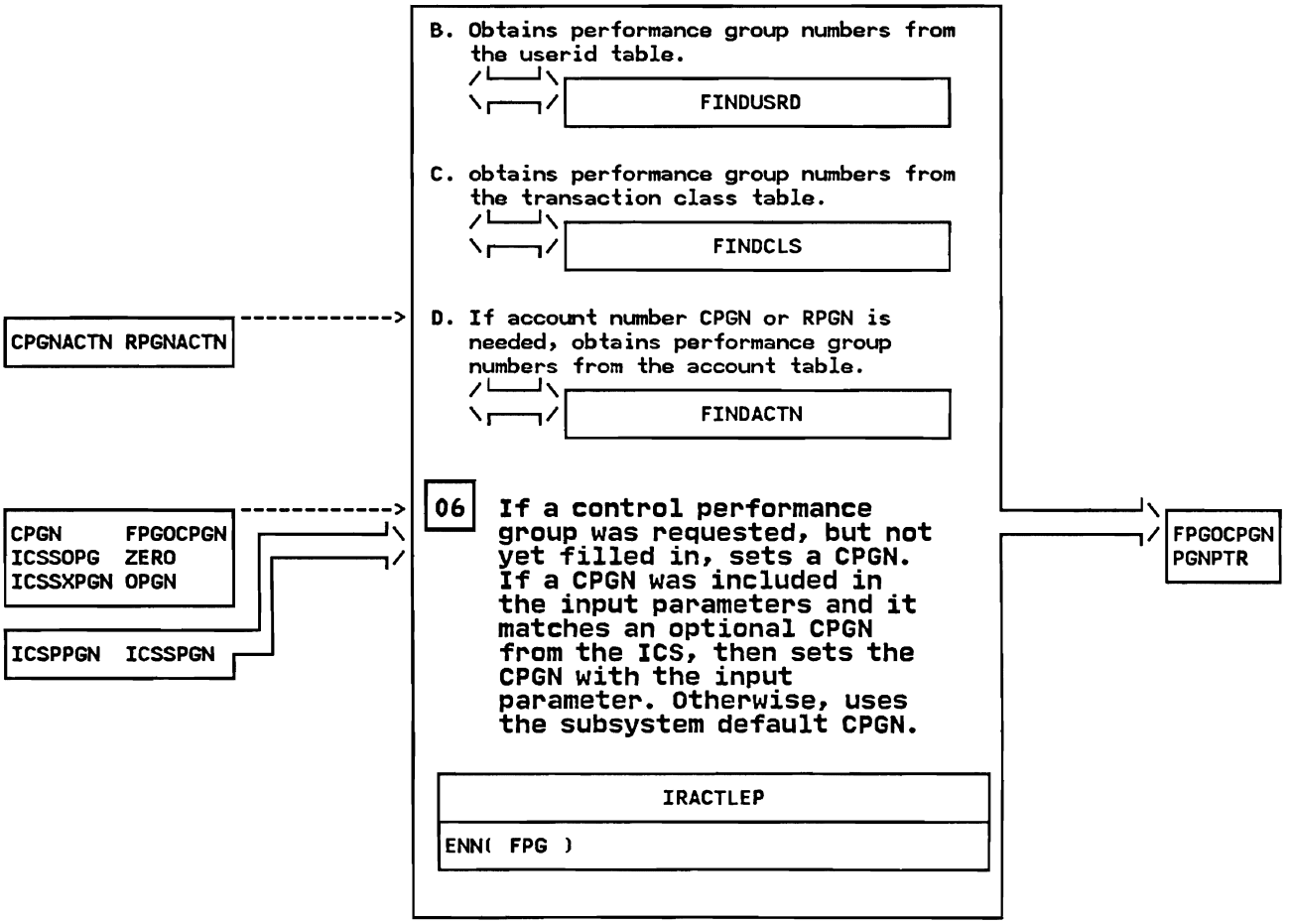
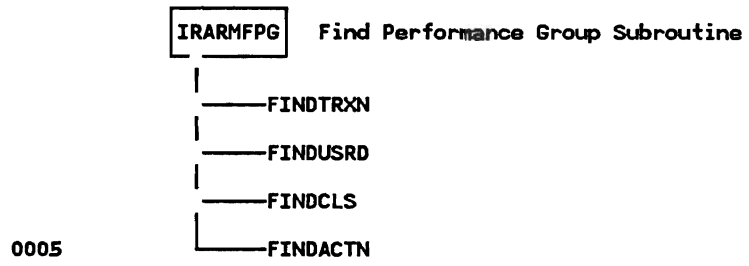


DIAGRAM SRM-5. IRARMFPG - Find Performance Groups (7 of 7)



IRARMCTL MODULE ENTRY POINT SUMMARY

IRARMASM	Auxiliary stage shortage monitoring. Monitors the extent of auxiliary storage allocation. Informs the operator if there is a shortage of available auxiliary pages and directs the swap out of users currently acquiring slots at the faster rate.
IRARMCTL	Mainline control processing. Transfers to deferred user action processing (IRARMCEN) and then to the algorithm request routine (IRARMCEL).
IRARMCEN	Deferred user action processing. Examines the OUCBACN field of the OUCB to determine the users on the action queue and routes control to all routines whose request bits have been set in the OUCBACN field. Dequeues each OUCB after its indicated actions have been performed.
IRARMCEL	Algorithm request routine. Examines the RMCTALR and RMCTALA fields in the RMCT and routes control (via IRARMCRT) to each algorithm whose request bit has been set in either of the two fields. Resets the individual bit after each algorithm completes.
IRARMCET	Periodic entry point scheduler. A timer initialization, schedules the algorithm currently due for execution and queues the SRM timer element to permit interrupts when the next algorithm is due for execution.
IRARMCED	SRB dispatched original entry processor. Receives control under an SRB scheduled by the dispatcher and sets up an entry to the mainline of SRM (IRARMCEN) by issuing SYSEVENT(30).
IRARMCQT	Periodically invoked entry point rescheduler. Accepts a request to reschedule the execution of a periodically invoked algorithm and requeues the corresponding RMEP block on the timed entry queue.
IRARMCRD	SRB scheduling routine. Accepts a request to schedule the SRM SRB, which if available, is scheduled to obtain the SRM lock.
IRARMCRL	Algorithm scheduling routine. Accepts requests for an algorithm to be run. Turns on the bit associated with the algorithm in the RMCTALA or RMCTALR.
IRARMCRN	Action request routine. Accepts requests for an action which must run under the SRM lock. If the SRM lock is held, control passes immediately to the action via a routing routine. If the SRM lock is not held, the bit set in the OUCBACN field of the OUCB associated with the requesting user identifies that the action requested is deferred.
IRARMCRT	Entry point table scanner. Accepts an invocation bit pattern and an entry point table address. Compares the bit pattern to invocation flags in the entry point table entries. Invokes the routine identified by the entry point when a match is found between the bit pattern and the invocation flags.
IRARMCRY	User swap request receiving routine. Accepts a request for a user swap and checks to see if such a swap is already in progress. Routes control to IRARMSCO or IRARMCSI if a swap is not in progress and the SRM lock is held.
IRARMCSF	Deferred swap analysis. Scans the WAIT queue for the deferred users who could not be swapped in because of a storage shortage If a swap-in appears possible, the user is taken out of the deferred state and placed on the OUT queue to be considered by swap analysis.
IRARMCSI	User swap-in request. Accepts a swap-in request, allocates an OUXB for the user and initiates the swap-in
IRARMCSO	User swap-out request. Accepts a swap-out request and posts the region control task's quiesce routine to initiate the swap-out.
IRARMRPS	OUCB repositioning routine. Dequeues an OUCB and requests it on the queue specified by its user indicators.

DIAGRAM SRM-6. IRARMCTL - SRM Control (1 of 2)

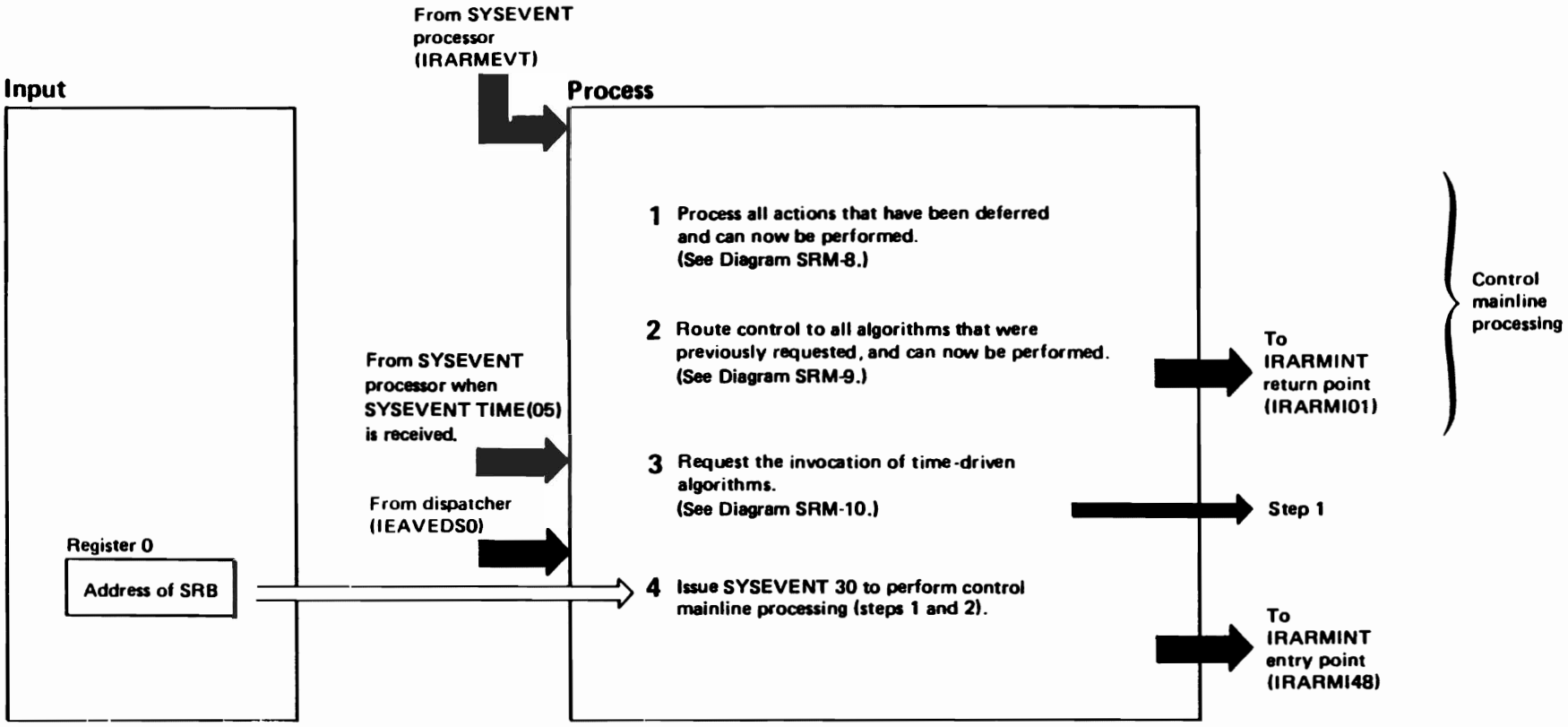


DIAGRAM SRM-6. IRARMCTL - SRM Control (2 of 2)

Extended Description	Module	Label
SRM Control routes control to actions and algorithms that have been requested and also to timed algorithms that have come due.	IRARMCTL	
1 Route control to actions that have been requested but deferred. Actions are SRM functions performed on behalf of a single user.	IRARMCTL	IRARMCEN
2 Route control to algorithms that have been requested. Algorithms are SRM functions performed on behalf of the system.	IRARMCTL	IRARMCEL
3 Request the invocation of time-driven algorithms that are now due. The queue of time-driven algorithms is scanned, and all algorithms that are due are requested by turning on representative bits in RMCTLAR. SRM Control now branches to step 1. Continuing with step 2, SRM Control will route control to those time-driven algorithms that were requested.	IRARMCTL	IRARMCET
4 This SRM Control entry point receives control under an SRB scheduled by another SRM component. The SRB was scheduled on behalf of routines not holding the SRM lock to execute critical actions and algoithms. Upon receiving control under the SRB, SRM Control makes a branch entry into the interface module, IRARMINT, to execute SYSEVENT 30. The SYSEVENT processor will in turn branch to SRM Control at step 1. Control will then be routed to the critical actions and algorithms that were requested.	IRARMCTL	IRARMCED

DIAGRAM SRM-7. IRARMCEN - Deferred Action Processor (1 of 2)

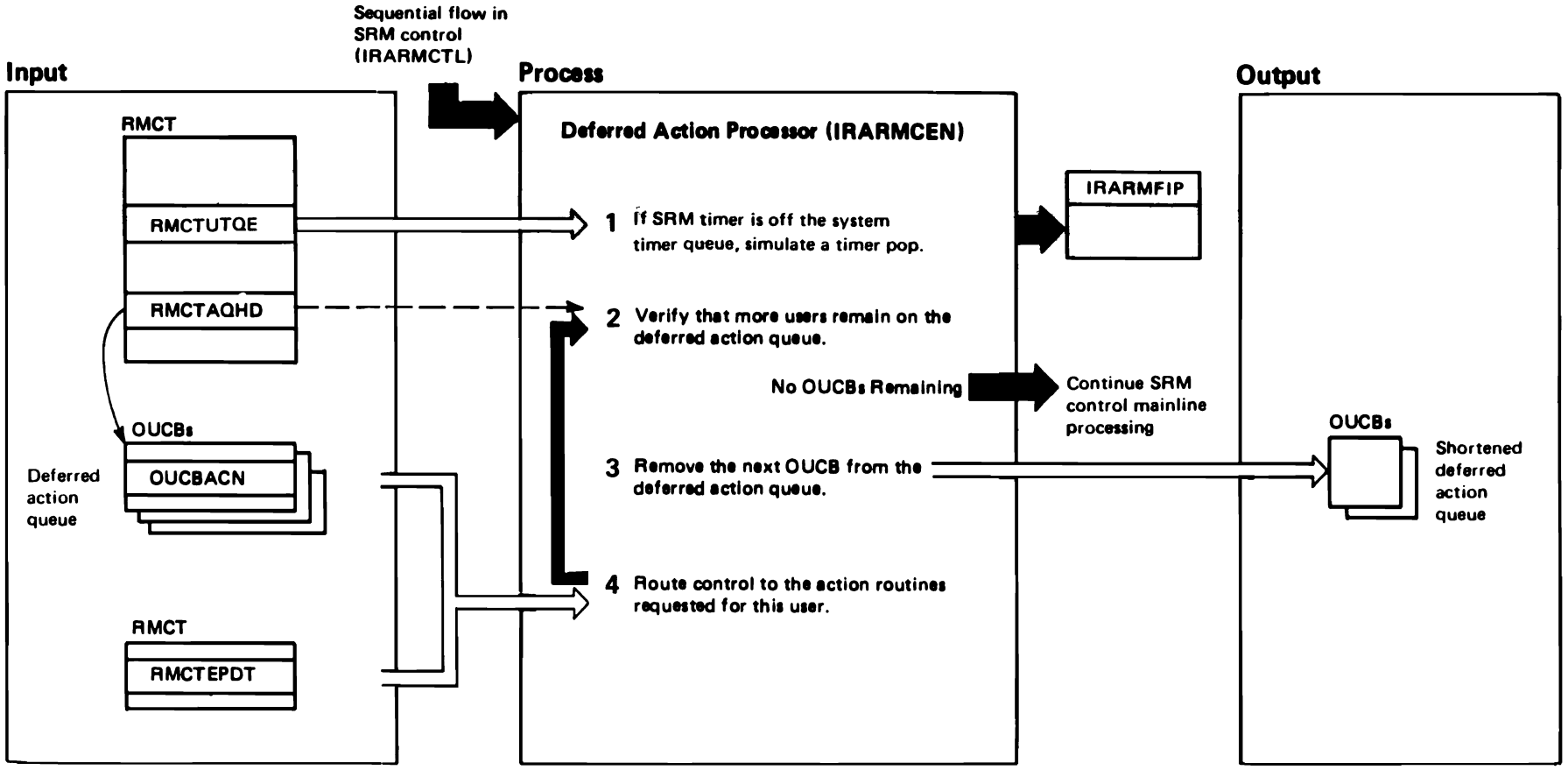


DIAGRAM SRM-7. IRARMCEN - Deferred Action Processor (2 of 2)

Extended Description	Module	Label
The Deferred Action Processor routes control to each requested routine for all OUCBs on the deferred action queue. The entry point descriptors for all possible action routines are contained in RMCTEPDT.	IRARMCTL	IRARMCEN
1 If SRMTQE is not on the system timer queue, pass control to IRARMFIP to simulate a timer pop.	IRARMCTL	IRARMCEN IRARMFIP
2 If the action queue header is pointing to the dummy pre-assembled OUCB (that is, RMCTAQHD=RMCTOUCB), then the action queue is empty.	IRARMCTL	IRARMCEN
3 The top OUCB is dequeued via compare-and-double-swap, to prevent multi-processing interaction problems. OUCBACT is set to "no".	IRARMCTL	IRARMCEN
4 IRARMCRT scans the EPDT entry point table looking for entry point blocks (RMEPs) whose invocation flags match "one" bits in the input bit pattern. For each successful match, the corresponding entry point is invoked. The invocation bit of each routine invoked is set to zero in the input bit pattern. It is possible for an action routine to call another action routine. In this case, the new routine request is inserted into the OUCBACN field, to be picked up during the processing of this OUCB.	IRARMCTL	IRARMCRT

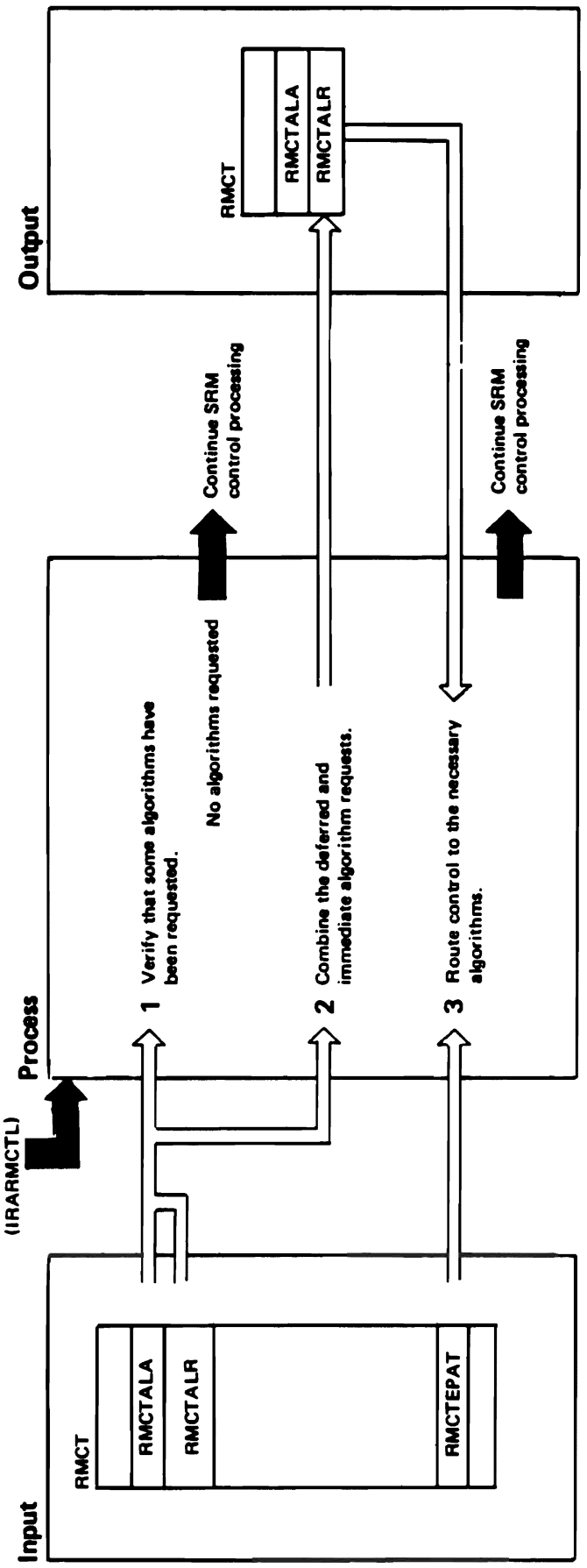


DIAGRAM SRM-8. IRARMCEL - Algorithm Request Processor (1 of 2)

DIAGRAM SRM-8. IRARMCEL - Algorithm Request Processor (2 of 2)

Extended Description	Module	Label
The algorithm request processor routes control to all algorithms that have been requested and can now be executed. The entry point descriptors for all possible algorithm routines are contained in RMCTEPAT.	IRARMCTL	IRARMCEL
1 Algorithms are requested if RMCTALA and RMCTALR are not both zero. Algorithm requests are stored in RMCTALR by SRM locked routines, and in RMCTALA by SRM unlocked routines.	IRARMCTL	IRARMCEL
2 Compare and swap logic is used to insure that all current requests are obtained for a multiprocessing environment.	IRARMCTL	RMCELL1
3 IRARMCRT scans the EPAT entry point table looking for entry point blocks (RMEPs) whose invocation flags match "one" bits in the input bit pattern. For each successful match, the corresponding entry point is invoked. For each algorithm called, the invocation bit is set to "zero" in the request bit pattern. Input parameters: • reg. 1 – address of first entry point block (RMEP) in the EPAT chained table • reg. 6 – address of input Lit pattern (RMCTALR)	IRARMCTL	IRARMCRT

DIAGRAM SRM-9. IRARMCET - Periodic Entry Point Scheduling (1 of 2)

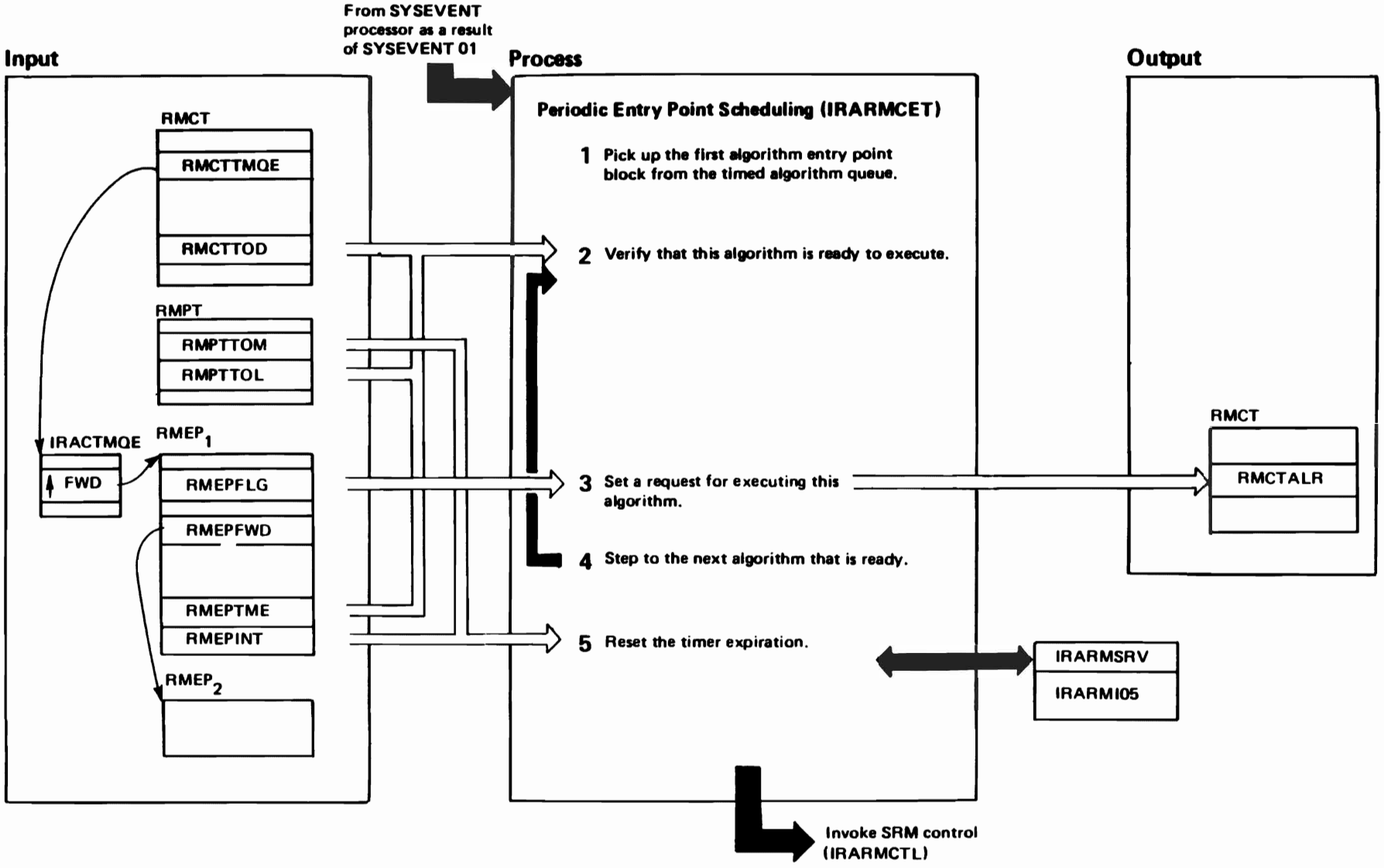


DIAGRAM SRM-9. IRARMCET - Periodic Entry Point Scheduling (2 of 2)

Extended Description	Module	Label
Periodic entry point scheduling is invoked following timer initialization. It sets up requests for all SRM periodically scheduled algorithms that are then due. It also requests the setting of the first SRM TQE to cause an interruption when next required. After the initial setting of the TQE, the TIME (05) SYSEVENT initiates the SRM timer queue expiration processing.	IRARMCTL	IRARMCET
1 The timer algorithm queue is ordered by the RMEPTME value of the RMEP blocks on the queue.	IRARMCTL	IRARMCET
2 An algorithm on the time-driven queue is due if the RMEPTME value is less than the current time (RMCTTOD) + an allowable tolerance (RMPTTOL).	IRARMCTL	IRARMCET
3 The algorithm request field is set up for later action by algorithm control routing (IRARMCEL).	IRARMCTL	IRARMCET
4 The next RMEP block is obtained from the queue.	IRARMCTL	IRARMCET
5 A new timer interruption is requested for the greater of: the minimum scheduling period (RMPTTOM), and the smallest time due of a scheduled routine. (See Diagram SRM-1.)	IRARMSRV	IRARMI05

DIAGRAM SRM-10. IRARMCSI - Control Swap-In (1 of 2)

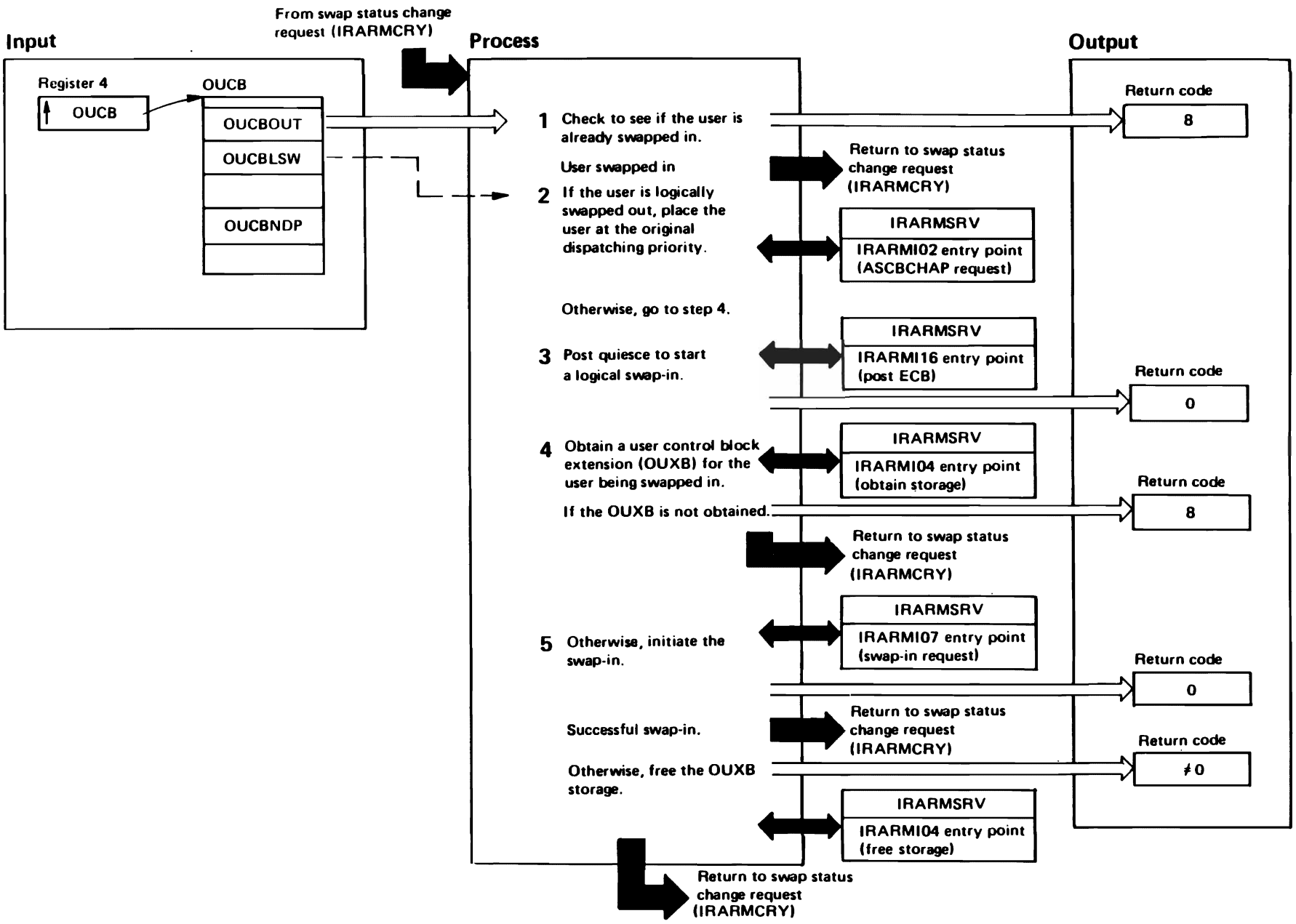


DIAGRAM SRM-10. IRARMCSI - Control Swap-In (2 of 2)

Extended Description	Module	Label (or Segment)
Control swap-in accepts a request that an address space be swapped in. If the address space is already swapped in, a return code is set; if not, control swap-in initiates a swap-in of the address space.	IRARMCTL	IRARMCSI
1 Control swap-in returns to the calling routine with a return code of 8 if the user for which a swap-in has been requested has already been swapped-in. Otherwise, control goes to step 2.	IRARMCTL	IRARMCSI
2 To expedite quiesce processing, request that the user's address space dispatching priority be reset to the dispatching priority associated with it before it was logically swapped.	IRARMSRV	IRARMI02
3 IRARMI16 will accomplish a cross-memory post by scheduling an SRB in the address space to be posted.	IRARMSRV	IRARMI16
4 The user control block extension (OUXB) is obtained. It remains in existence as long as the user is swapped in and is released at swap-out.	IRARMSRV	IRARMI04
5 If the swap-in is successfully initiated (return code from IRARMI07 equals 0), the OUXB is cleared, the address of the OUXB is placed into the ASCB (ASCBOUXB), and the OUCB going-in bit is set (OUCBGOI).	IRARMSRV	IRARMI07
Otherwise, the storage for the OUXB is freed.	IRARMSRV	IRARMI04
Error Processing:		
If an attempt to obtain storage for an OUXB fails (step 4), or an attempt to initiate a user swap-in fails (step 5), the user remains on the OUT queue, and control swap-in returns to the caller with an error return code.	IRARMCTL	IRARMCSI

DIAGRAM SRM-11. IRARMCSO - Control Swap-Out (1 of 4)

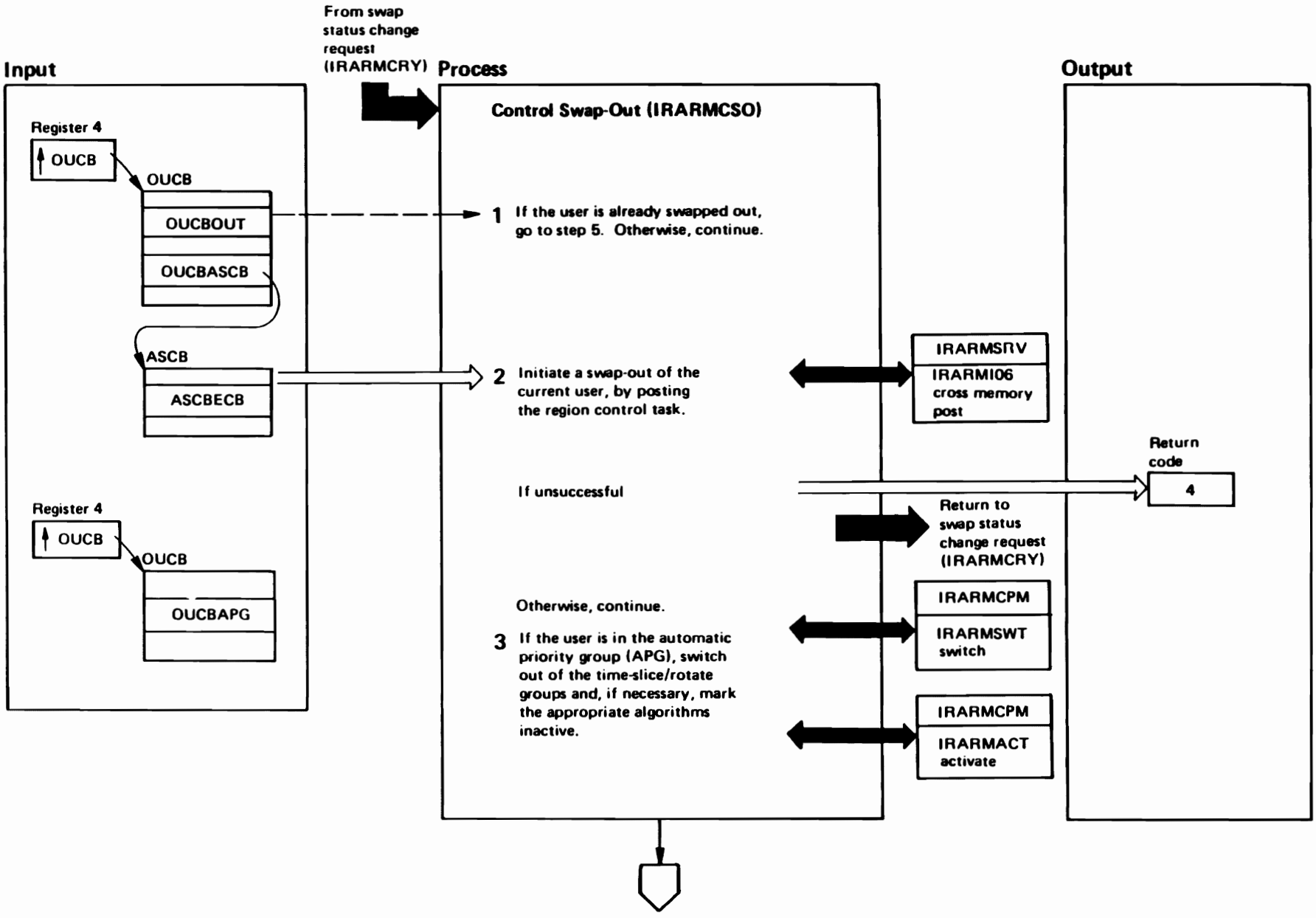


DIAGRAM SRM-11. IRARMCSO - Control Swap-Out (2 of 4)

Extended Description	Module	Label
Control Swap-Out accepts a request that an address space be swapped out. If the address space is already swapped-out, a return code is set; if not, control swap-out initiates the swap-out of the address space.	IRARMCTL	IRARMCS0
1 Control swap-out skips to step 5 if the user for which a swap-out has been requested has already been swapped-out, as indicated by the OUCBOUT bit. Otherwise, control goes to step 2.	IRARMCTL	IRARMCS0
2 The supervisor service request routine requests the initiation of quiesce processing for the user to be swapped out. This request results in the posting of an ECB for the indicated address space, so that the RCT will begin quiesce processing. A successful return from IRARMI02 indicates that the post of quiesce processing has been scheduled for the address space. The process of quiesce processing will be indicated to the SRM by future SYSEVENTs (typically, quiesce started, followed by quiesce completed, followed by swap-out complete).	IRARMSRV	IRARMI06
3 Users are removed from time-slice group for the duration of swap-out.	IRARMCPM	IRARMSWT IRARMACT

DIAGRAM SRM-11. IRARMCSO - Control Swap-Out (3 of 4)

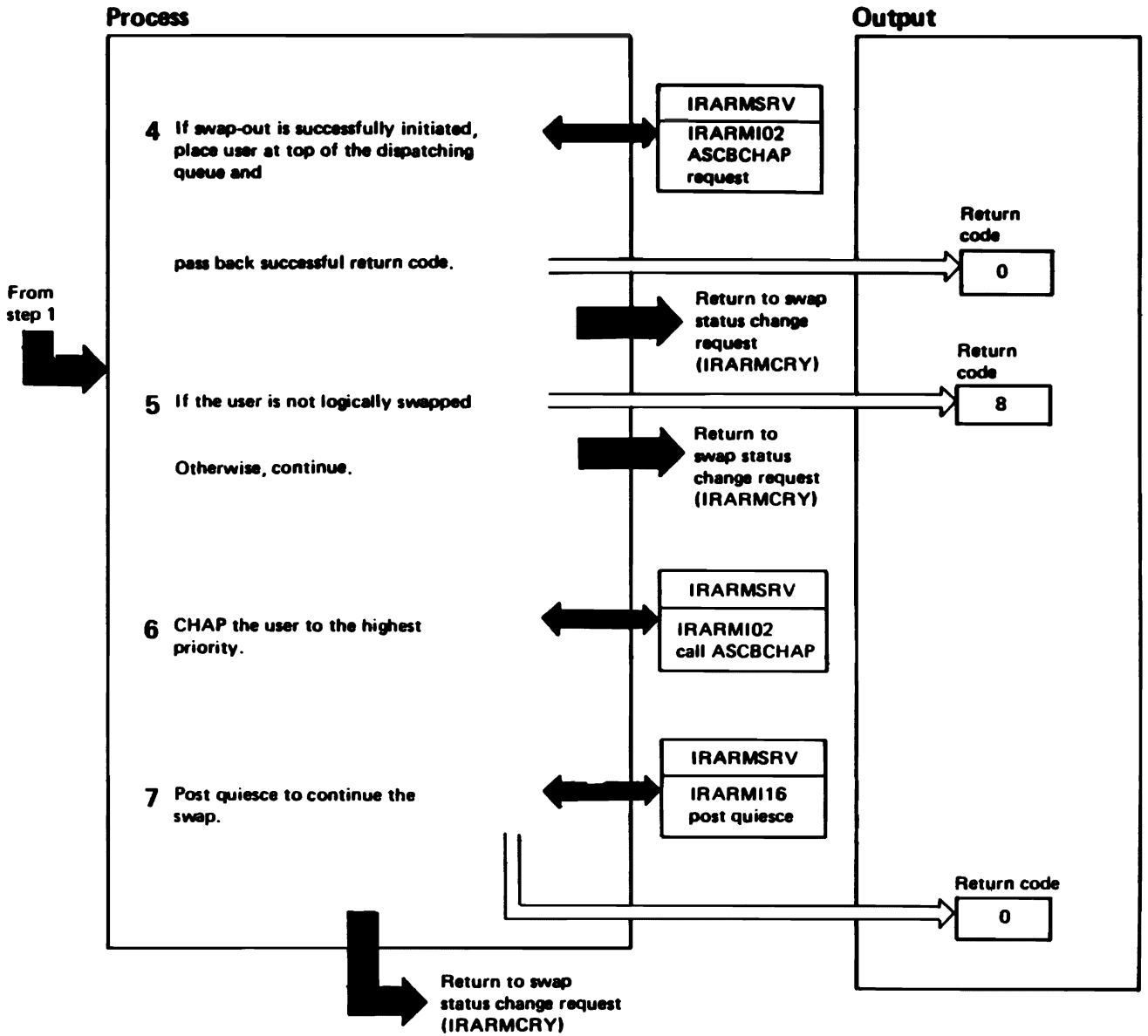


DIAGRAM SRM-11. IRARMCSO - Control Swap-Out (4 of 4)

Extended Description	Module	Label
4 To expedite quiesce processing, request that the user's ASCB be moved to the highest priority.	IRARMSRV	IRARMI02
5 Test swapped-out users to ensure that they are logically swapped. Users already physically swapped cannot be reswapped. A return code of 8 indicates this error.		
6 To expedite processing, request that the user's ASCB be moved to the highest priority.	IRARMSRV	IRARMI02
7 Call IRARMI16 to post the quiesce ECB by preposting it or by scheduling an SRB into the address space. The swap-out complete SYSEVENT is issued when the swap-out is finished.	IRARMSRV IRARMSRV	IRARMI16 IRARMI17

IRARMSWP MODULE ENTRY POINT SUMMARY

IRARMCAP	Swap analysis algorithm. Attempts to keep the multiprogramming level (MPL) at its target level in each domain by performing user swaps.
IRARMCPI	Select swap-in candidate subroutine. Scans the OUT queue for the user in a particular domain with the highest recommendation value.
IRARMCPO	Select swap-out candidate subroutine. Scans the IN queue for the user in a particular domain with the lowest recommendation value.
IRARMCVL	User swap evaluation routine. Computed a numerical value representing the recommendation of a user to be swapped in. This recommendation value is the sum of the user's workload level and the recommendations of the I/O and CPU resource managers.
IRARMSA7	Deferred migration swap redrive algorithm. Reattempts any migration swaps deferred by storage shortages when enough storage is available.

DIAGRAM SRM-12. IRARMCAP - Swap Analysis (1 of 2)

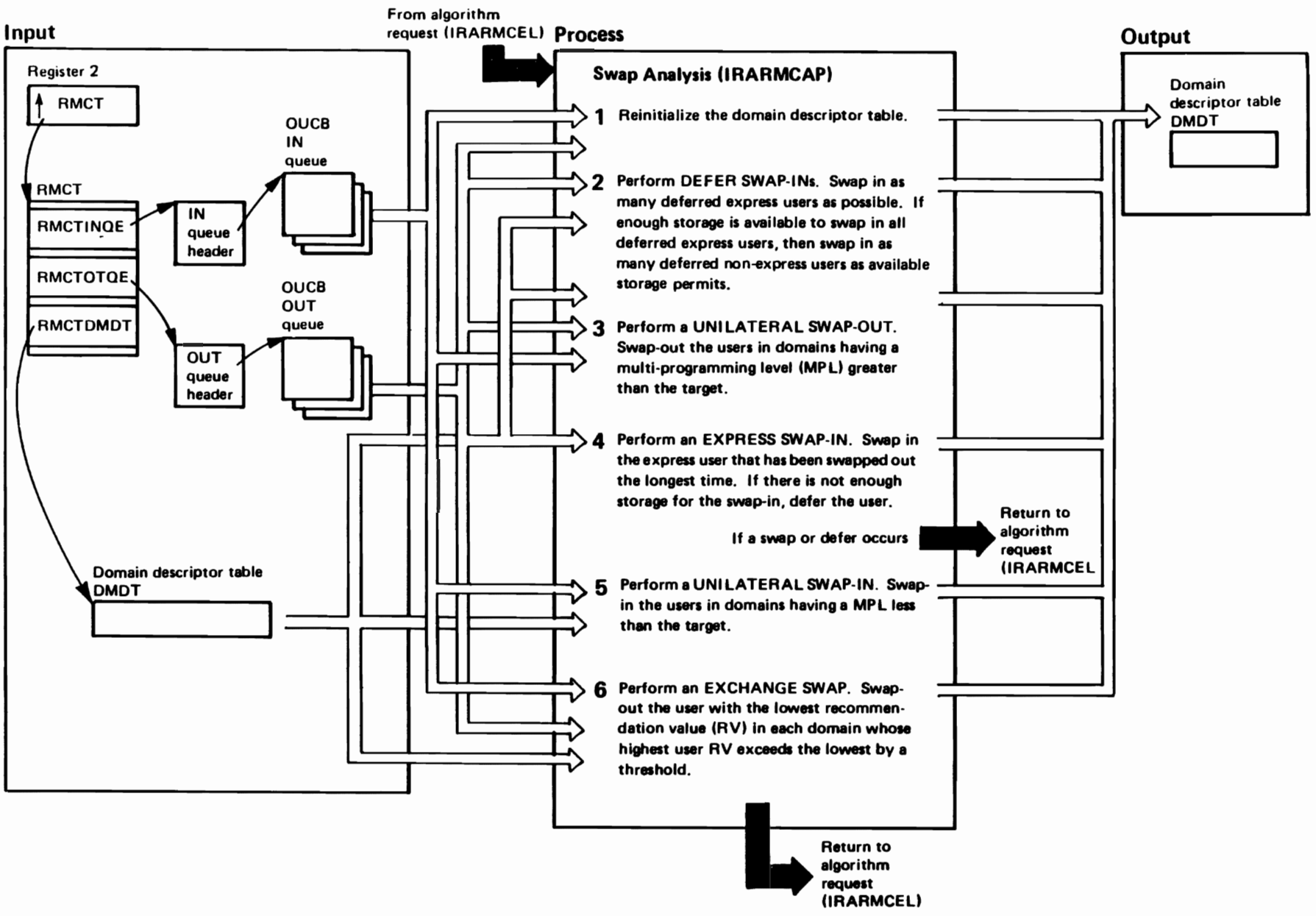


DIAGRAM SRM-12. IRARMCAP - Swap Analysis (2 of 2)

Extended Description	Module	Label (or Segment)
Swap analysis is performed on a time-driven basis. It is an algorithm activated by IRARMF05. It is also activated by the processing of three SYSEVENTS: USERRDY (04), SWOUTCMP (0F), and QSCECMP (0C).	IRARMSWP	IRARMCAP
1 The domain descriptor table has one entry for each domain defined by the IPS. Each OUCB on the IN and OUT queues is examined. Swappable, valid users on the IN queue which are not in the process of being swapped out or moving from one SRM queue to another are counted in the current multiprogramming level (MPL) for a domain, as well as users on the OUT queue which are being swapped in. An installation option allows non-swappable users to be counted in the current MPL for a domain. Fields in each domain descriptor table entry are reinitialized with the above MPL count information.	IRARMSWP	IRARMCAP
Note: A terminating address space that is swapped out is not counted as a swapped out user in the domain table and is not considered for swap-in.		
2 Find the users that were previously picked for swap-in, but were deferred because of insufficient storage. (When a swap-in is deferred, SRM temporarily steals storage at a faster rate until enough storage is available to complete the swap-in). Swap-in as many deferred express users as will fit into storage. (An express user is a user that is enqueued on a resource and is either non-swappable or is being canceled). If enough storage is available to swap in all deferred express users, then swap in as many deferred non-express users as available storage permits. For deferred swap-ins, IRARMCAP chooses first the users that have been swapped out the longest time.		
3 Search the domain descriptor table entries for a domain with an MPL higher than the target value and swap-out the user with lowest recommendation value (RV). Repeat until the MPL reaches the target value in every domain.	IRARMSWP IRARMCTL	IRARMCPO IRARMCSO

Extended Description	Module	Label (or Segment)
3 *(continued)* If at least one swap is performed in this step, swap analysis ends here. Otherwise, continue at step 4.		
4 If there are any express users on the OUT queue, attempt to swap in the express user that has been swapped out the longest time. If there is not enough storage for the swap-in, defer the user. If the MPL in that domain is less than the target, swap that user in. If the MPL is not less than the target, make room for the user by a swap-out of the user with the lowest RV. Repeated calls to swap analysis may be necessary to reduce the MPL below the target value to allow the express user to be swapped in. If there is no swap or defer, continue to step 5. Otherwise, swap analysis ends here.	IRARMCTL IRARMSWP IRARMCTL	IRARMCSI IRARMCPO IRARMCSO
5 Search the domain descriptor table entries for a domain with an MPL less than target and swap-in the user with the highest RV. Repeat until the MPL (plus users in the process of being swapped out) reaches the target in each domain. If there is not enough storage for a swap-in, defer the swap-in until the next invocation of swap analysis. (See step 2). If at least one swap or defer is done in this step, swap analysis ends here.	IRARMSWP IRARMCTL	IRARMCPI IRARMCSI
6 Search the domain descriptor table entries for a domain with an MPL that equals the target for that domain. In this domain, find the out-of-storage user with the largest RV to come in, and the in-storage user with the smallest RV to remain in. If the difference of their RVs exceeds a threshold (RMPTXCHT), swap out the user with the lower RV. This exchange swap is only done for one domain.	IRARMSWP IRARMCTL	IRARMCPO IRARMCPI IRARMCSO
Error Processing: • IRARMERR handles all unexpected errors. • Any non-zero return codes from called routines cause swap analysis (IRARMCAP) to end without finishing its processing.		

DIAGRAM SRM-13. IRARMCPI - Select User for Swap-In (1 of 2)

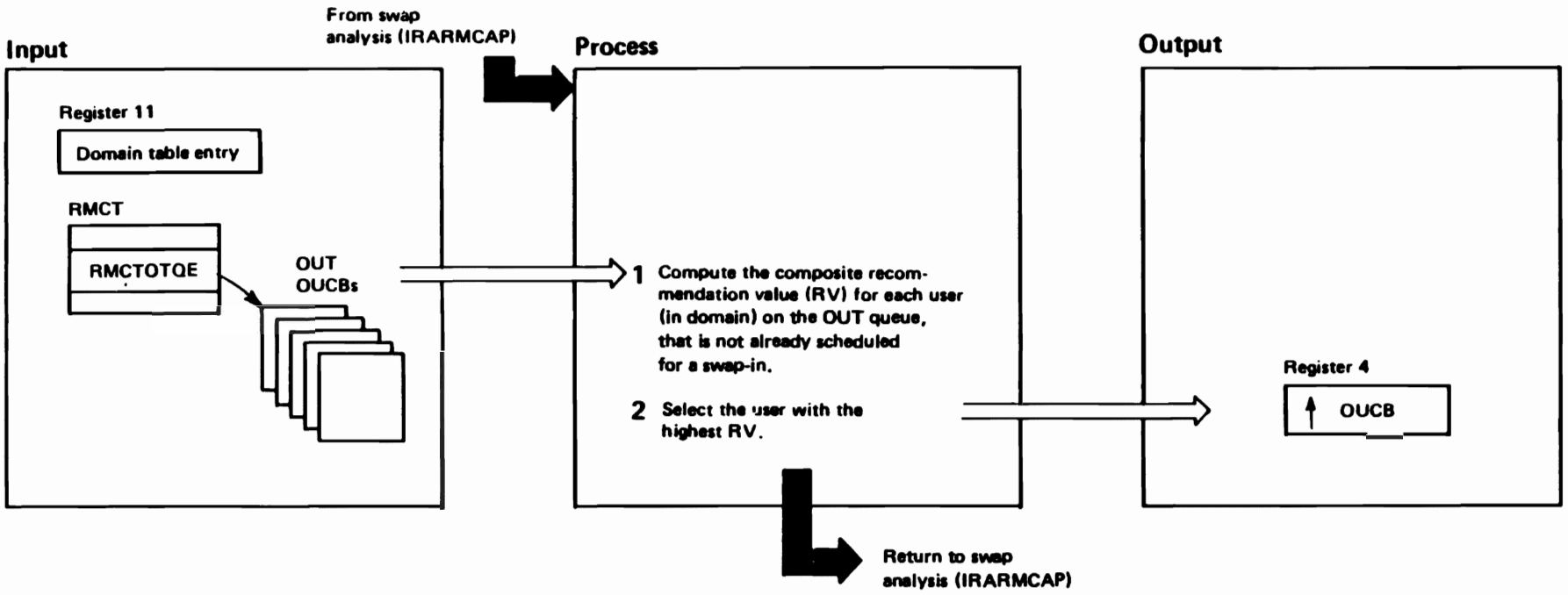


DIAGRAM SRM-13. IRARMCPI - Select User for Swap-In (2 of 2)

Extended Description	Module	Label
This routine chooses the user with the highest RV in a particular domain on the OUT queue. If one of the users represented by an OUCB in this domain is assigned to a different domain, for example, because of a period change, return a code of zero indicating no user found. In this case, swap analysis (IRARMCAP) is rescheduled to ensure that the domain descriptor table is initialized to reflect this domain change. The following two steps are performed in a loop until all OUCBs on the OUT queue have been evaluated.	IRARMSWP	IRARMCPI
1 Examine each OUCB on the OUT queue for users in the specified domain. Use the user evaluation subroutine to compute the composite RV for each user.	IRARMSWP	IRARMCPI
	IRARMSWP	IRARMCVL
2 Compare the computed RV to that of the highest RV found up till now. Save this OUCB as the best candidate for a swap-in if its RV is greater. Otherwise, continue until all OUCBs on the OUT queue in this domain have been evaluated.	IRARMSWP	IRARMCPI

DIAGRAM SRM-14. IRARMCPO - - Select User for Swap-Out (1 of 2)

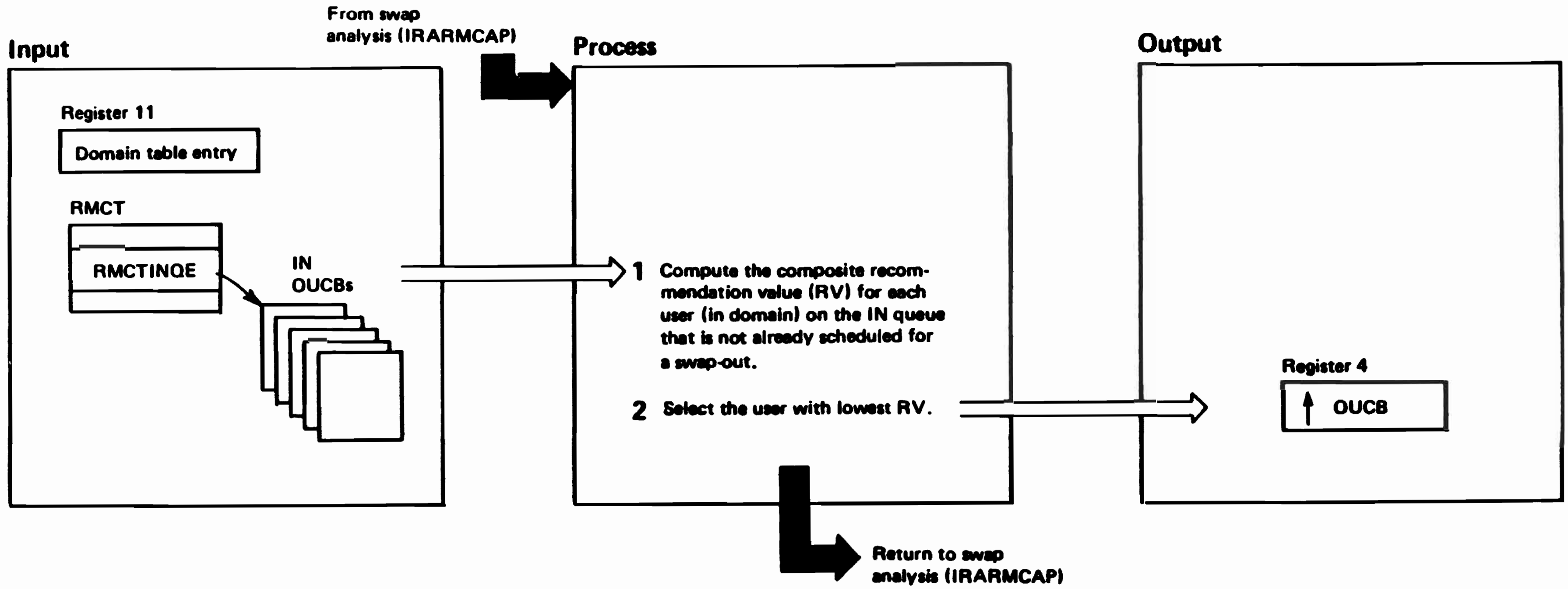


DIAGRAM SRM-14. IRARMCPO - - Select User for Swap-Out (2 of 2)

Extended Description	Module	Label
This routine chooses the user with the lowest RV in a particular domain on the IN queue. If one of the users represented by an OUCB in the domain is assigned to a different domain, for example, because of a period change, return a code of zero indicating no user found. In this case swap analysis (IRARMCAP) is rescheduled to ensure that the domain descriptor table is initialized to reflect this domain change.	IRARMSWP	IRARMCPO
The following two steps are performed in a loop until all OUCB's on the IN queue have been evaluated.		
1 Examine each OUCB on the IN queue for users in the specified domain. Use the user evaluation subroutine to compute the composite RV for each user.	IRARMSWP	IRARMCPO
	IRARMSWP	IRARMCVL
2 Compare the computed RV to that of the lowest RV up till now. Save this OUCB as the best candidate for a swap-out if its RV is lower.	IRARMSWP	IRARMCPO
Otherwise, continue until all OUCBs in this domain on IN queue have been evaluated.		

DIAGRAM SRM-15. IRARMCVL - User Evaluation (1 of 2)

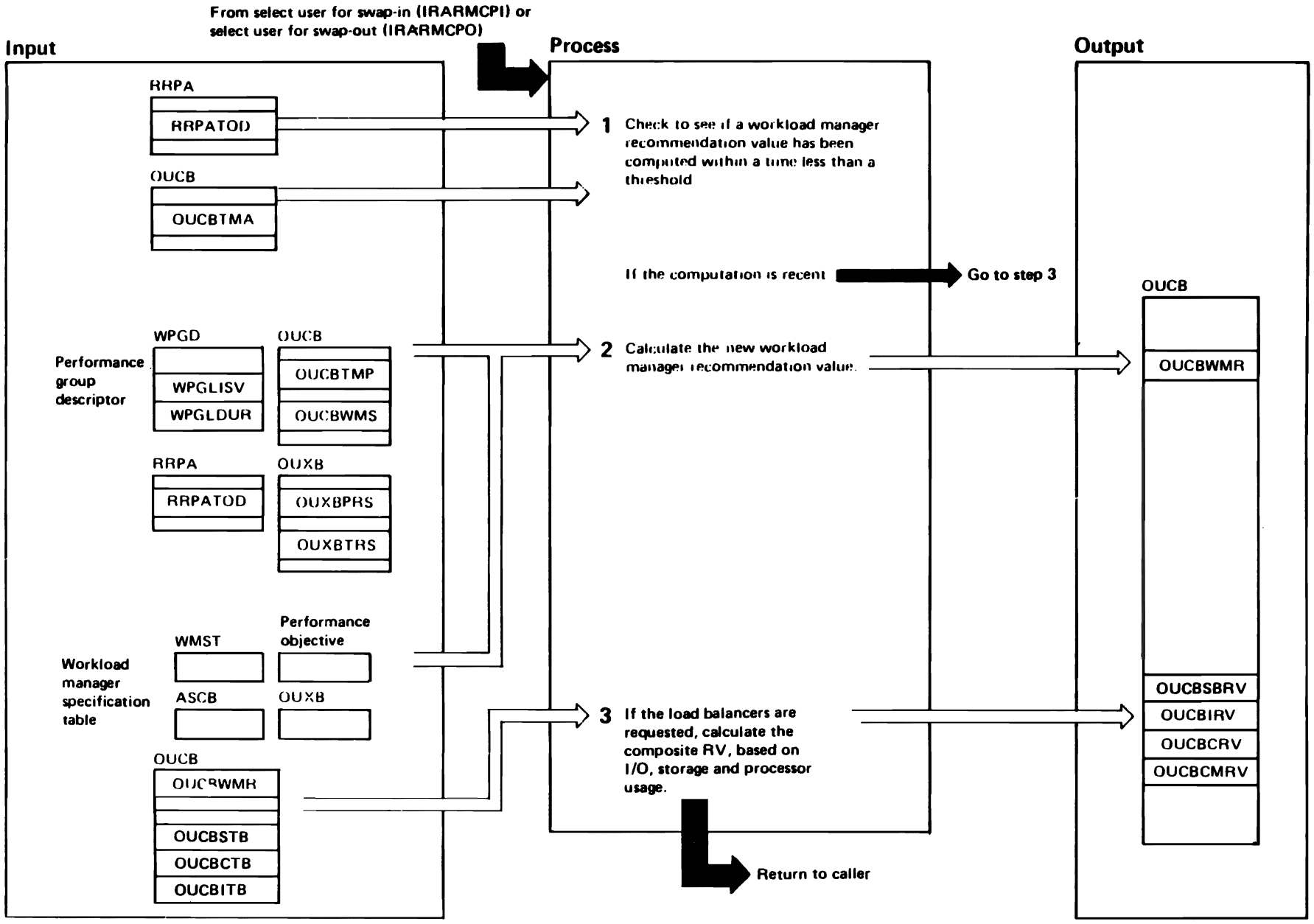


DIAGRAM SRM-15. IRARMCVL - User Evaluation (2 of 2)

Extended Description	Module	Label
User evaluation computes a recommendation value (RV) for one user based on its workload manager recommendation value and its I/O, storage and processor recommendation values.	IRARMSWP	IRARMCVL
1 A new value is calculated for the workload manager RV only if sufficient time has elapsed since its previous calculation. (Swap analysis evaluating threshold RMPTSAET).	IRARMSWP	IRARMCVL
2 Compute the workload manager recommendation value (the normalized workload level) representing the desirability of a swap of this user. This value is based on the rate at which the user has recently been receiving service and on the IPS	IRARMWLM	IRARMWM3
3 If the applicable RTB value is 1, add to the workload level an I/O manager recommendation value (for significant users of I/O), add a processor manager recommendation value (for significant users of the processor resource), and add a storage manager recommendation value (for significant users of storage). A positive RV favors the swap-in of a user to correct a storage or I/O imbalance and a negative RV favors the swap-out of a user.	IRARMIOM	IRARMIL3
	IRARMCPM	IRARMCL3
	IRARMCPM	IRARMMS3

IRARMSTM MODULE ENTRY POINT SUMMARY

*CKSIGUR	Storage load balancing. Monitor significant users of real storage.
IRARMMS3	Storage load balancing swap evaluation. Produce a numerical recommendation value that reflects the eligibility of a user to be swapped (based on the use of real storage and the contention for real storage).
IRARMPR1	Page replacement normal processing. Examine each user in main storage and the system pageable area and call RSM to update UICs for each user.
IRARMPR5	Page replacement real page shortage force steal. Steal as many frames as required to relieve a real page frame shortage. The steal decision is made at entry IRARMMS2. The oldest unreferenced system-wide frames are stolen first.
IRARMPR9	Swap out trim routine. Trim user logical swaps by think time and for swap-outs to auxiliary storage and extended storage.
*STEAL	Internal IRARMSTM steal subroutine. Add users to interface list until full, then call RSM steal routine (IARXSF via IRARMI03) and record the number of pages stolen.
*STMFENCE	Storage isolation. Calculate the target working set sizes for address spaces and the common area if controlled by the page-in rate.
*STORUSE	Storage load balancing. Monitor real storage utilization.

---
*Internal subroutine

IRARMST2 MODULE ENTRY POINT SUMMARY

IRARMMS2 Real page shortage prevention. As frames are needed, finish deferred swaps to extended storage and physically swap users that have been logically swapped for too long. After this, if a real page shortage exists, schedule IRARMPR5. If a pageable frame shortage exists, inform the operator of the shortage and of the users that might have caused the shortage condition and the amount of fixed and real storage pages allocated to these users. Direct the swap-out of these users.

IRARMMS4 Threshold adjustment routine for storage load balancing and pageable storage shortage processing. Increase or decrease the threshold used to determine which address spaces are considered significant users of real storage. Calculate the percentage of users that are eligible for storage load balancing and that are considered significant users of real storage with the existing threshold. The threshold is increased to keep the percentage of significant users at a constant percentage of the eligible users.

Increase or decrease the threshold used as the median fixed frame count for all ready address spaces, and the threshold for TSO address spaces swapped out for terminal wait. The medians are used by pageable storage shortage processing and logical swapping to assess a user's impact on pageable storage.

IRARMMS5 Pageable storage shortage calculation. Calculates the thresholds for pageable storage detection and relief for use by RSM and SRM.

IRARMMS6 Main storage occupancy long wait detection. Discover user that have gone into long wait without notifying SRM. Swap out such users, if they are swappable.

IRARMSQA SQA shortage message writer. Inform the operator of system queue area shortages. Signal SQA shortage using event notification facility (ENF).

DIAGRAM SRM-16. IRARMSTM, IRARMST2 - Storage Management (1 of 14)

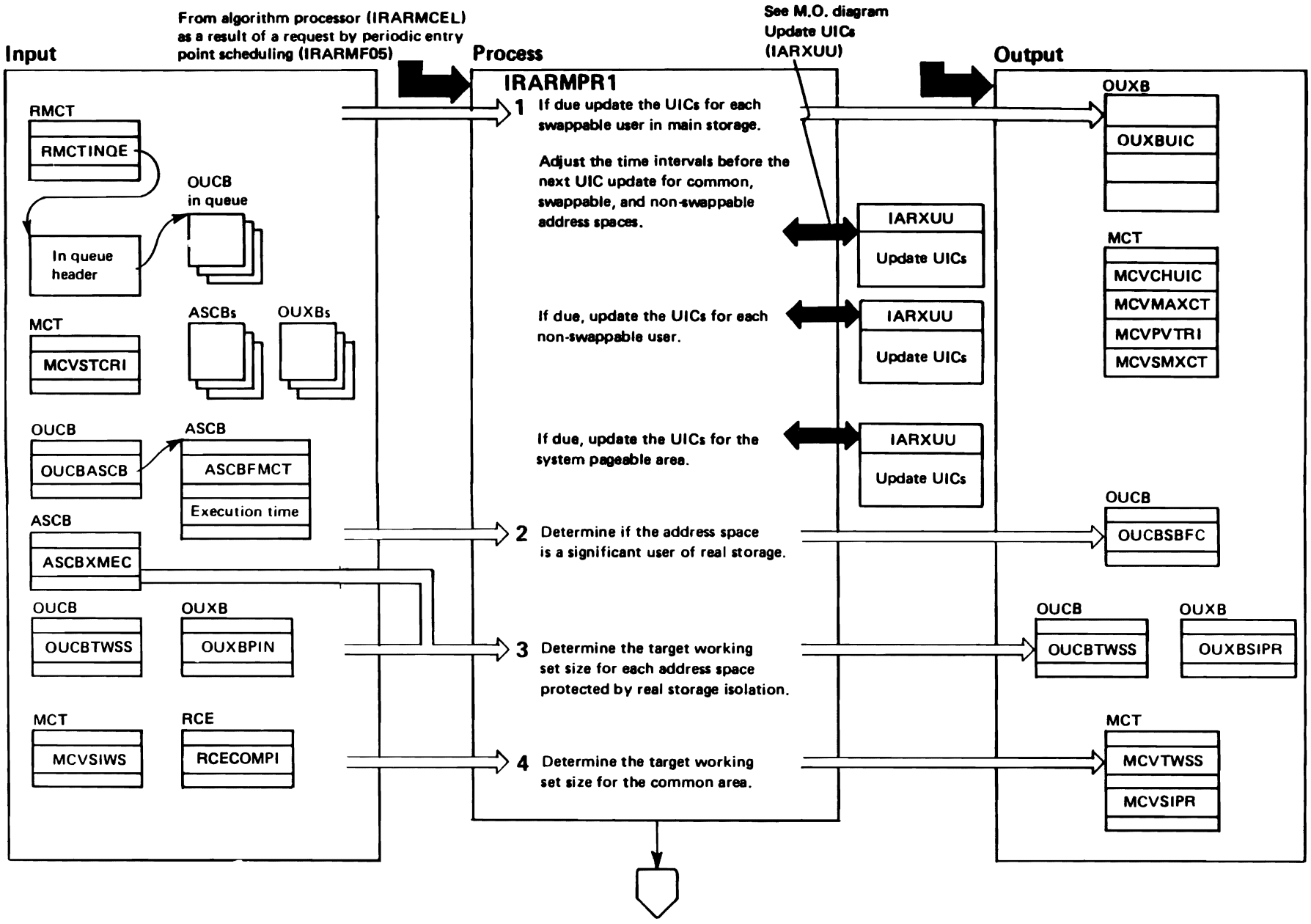
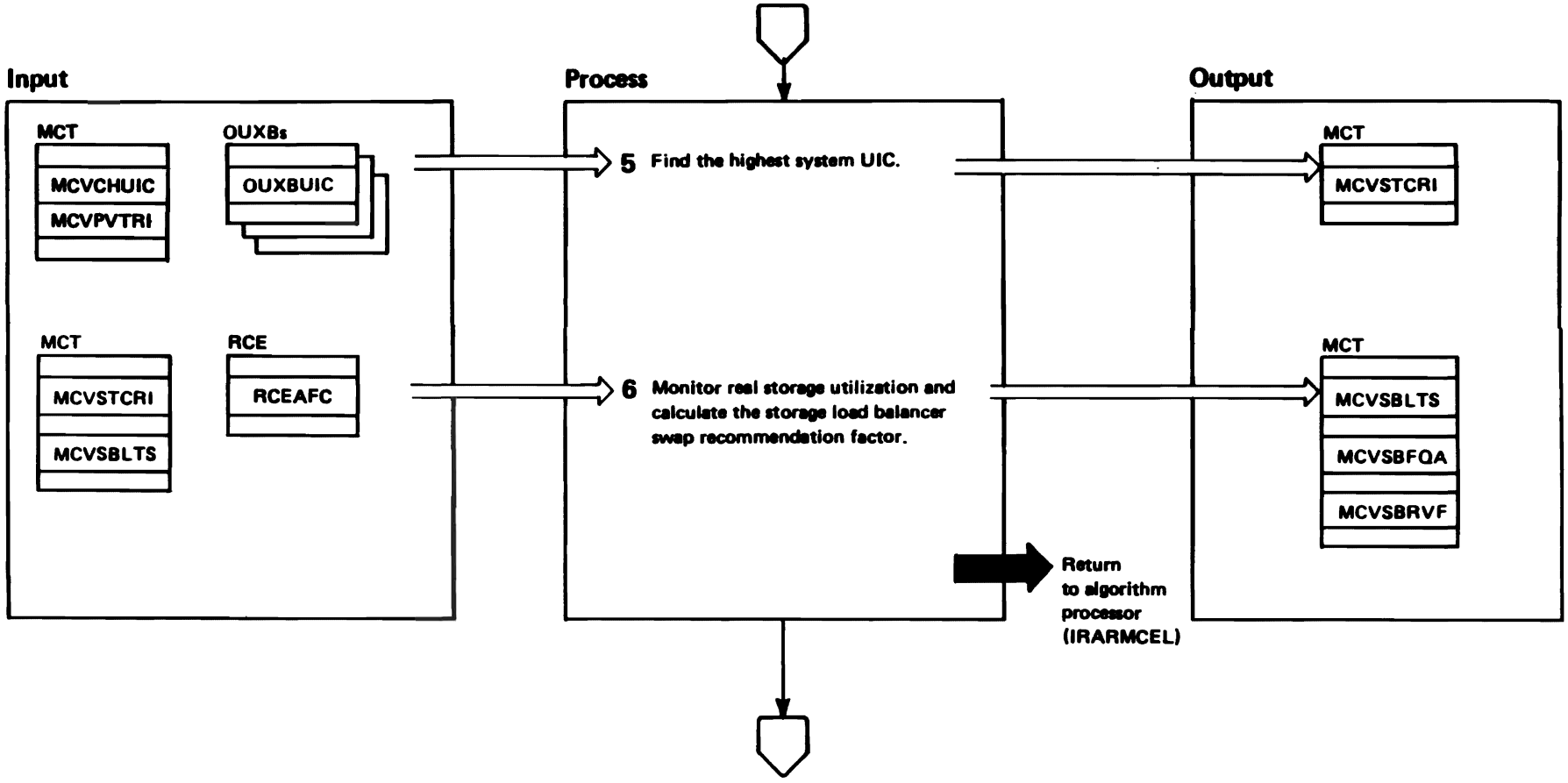


DIAGRAM SRM-16. IRARMSTM, IRARMST2 - Storage Management (2 of 14)

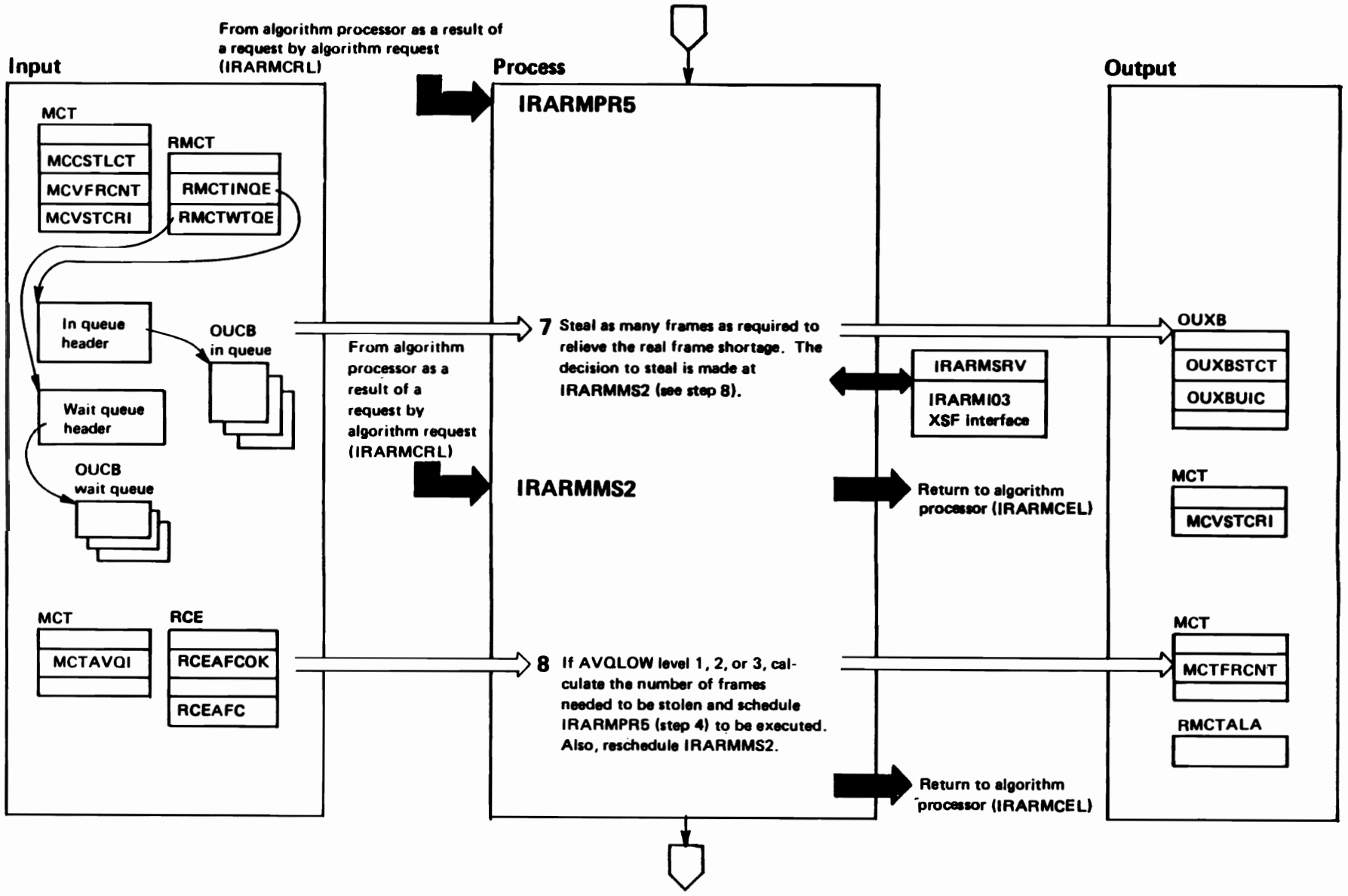
Extended Description	Module	Label
Storage management consists of essentially independent routines that are invoked by SRM control or by the SYSEVENT processor to control the usage of main and auxiliary storage by all users. For non-swappable users (users whose fixed storage must remain in real storage), the mechanism of page stealing (freeing non-fixed pages for other use) is used for storage management control. For swappable users, both page-stealing and swapping provide the necessary control.	IRARMSTM	
1 The frequency of updatng UICs for the common area, swappable, and non-swappable address spaces will correspond to every *nth* invocation of IRARMPR1, where *n* is calculated based on storage usage, which is indicated by the highest UIC of any page in the system. IRARMSTM schedules an SRB to update the UICs. This is done separately from other SRM processing to prevent SRM from disabling the system for excessive periods of time.	IRARMSTM	IRARMPR1
IRARMSTM enters the RSM UIC update module (IARXUU) to update the UICs for swappable and non-swappable users of storage and for the common area. IARXUU updates the UICs for each address space passed in the parameter list. A zero ASCB address in the parameter list indicates the common area. RSM examines the UICs associated with each of the user's (or the common area's) pages. If the page reference bit is on, the UIC for this page is set to zero, and the reference bit reset. If the reference bit is off, the corresponding UIC is increased by the value specified in the parameter list. RSM then saves the highest UIC count for this user (or the common area) in the corresponding entry in the parameter list. The highest UIC for the system pageable area is also saved in the corresponding entry in the parameter list. The SRM saves the highest UIC for each user in the corresponding OUXB and the highest UIC for the common area in MCVCHUIC. UIC updating for common and non-swappable address spaces occurs on every *nth* invocation of IRARMPR1. *n* is recalculated based on storage usage, which is indicated by MCVSTCRI and the constants MCCNCLIM, MCCNCDEL, and MCCNCMAX. UIC updating for swappable address spaces occurs on every *nth* invocation of IRARMPR1. *n* is recalculated based on MCVSTCRI and the constants MCCSWLIM, MCCSWDEL, and MCCSWMAX.	IARXUU	IARXUU
2 If storage load balancing is active and applicable to this address space, and the address space has accumulated enough execution time since it was last swapped in, IRARMSTM saves the number of frames owned by the address space in OUCBSBFC and determines if the address space meets the criteria for a significant user of storage. If it does, IRARMSTM indicates that it is a significant user (OUCBSBSU=1).	IRARMSTM	CKSIGUR
3 If real storage isolation is applicable to this address space and the page-in rate control is in effect, IRARMSTM adjusts the target working set size based on (1) the minimum and maximum working set sizes specified in the IPS and (2) the page-in rate for the address space. The page-in rate for an address space is defined as the average number of private area page-ins per second of execution time, except for address spaces executing in cross memory mode. In a cross memory environment only address spaces executing as the home address space accumulate execution time. Except at initilization, an address space that can be entered using a program call (PC) from another address space (ASCBXMEC=1) rarely executes as the home address space and, therefore, accumulates little or no execution time. The page-in rate for this type of address space is based instead on elapsed time and is defined as the average number of page-ins per second of transaction residency time.		STMFENCE
4 If real storage isolation is applicable to the common area and the page-in rate control is in effect, IRARMSTM adjusts the target working set size based on (1) the minimum and maximum working set sizes specified in the IPS and (2) the page-in rate for the common area. The page-in rate for the common area is based on elapsed time and is defined as the number of CSA and PLPA page-ins per second of elapsed time.		IRARMPR1

DIAGRAM SRM-16. IRARMSTM, IRARMST2 - Storage Management (3 of 14)



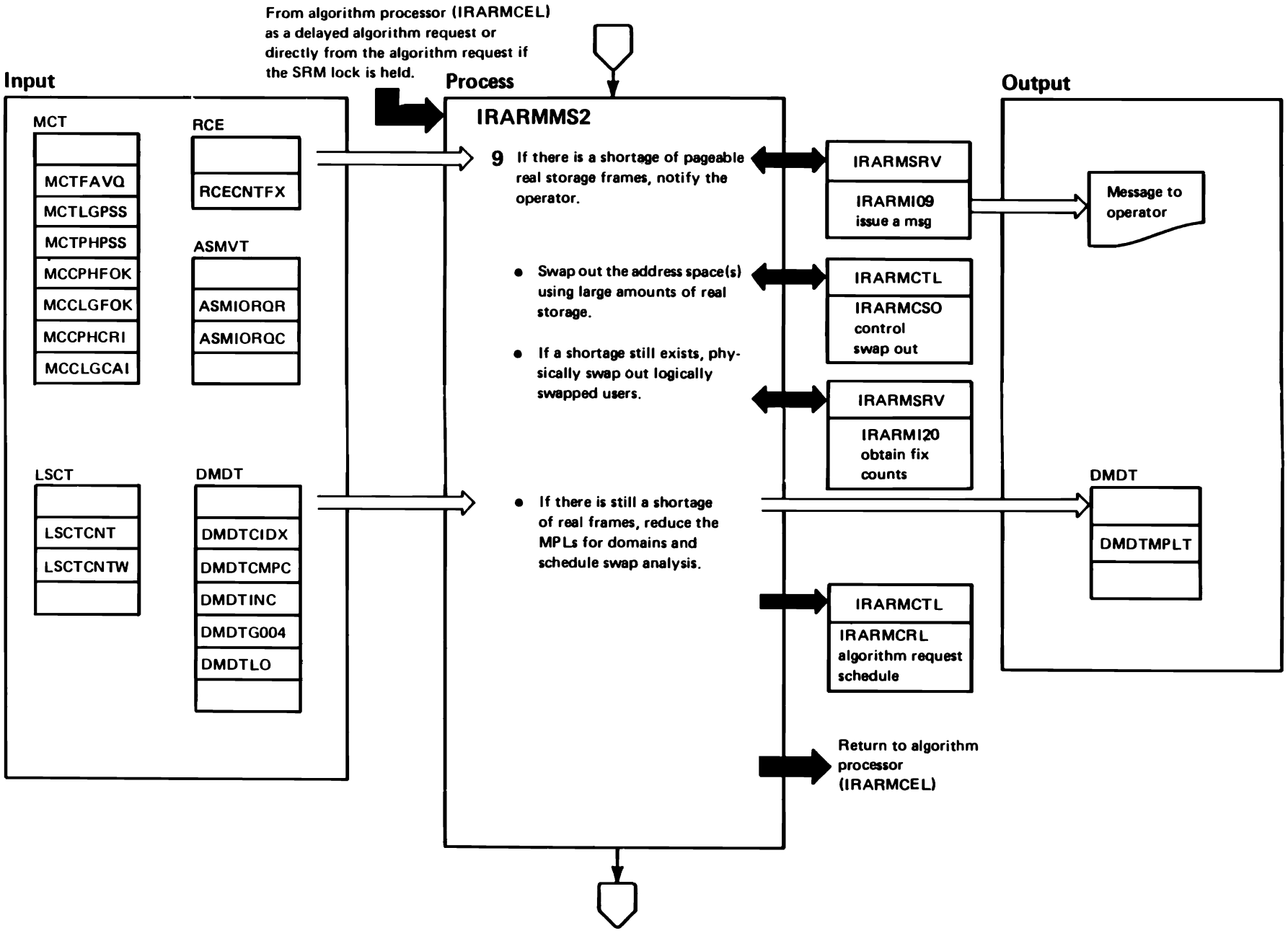
Extended Description	Module	Label
5 The highest UIC found in any address space or system pageable area that is not protected by real storage isolation is identified and saved in MCVSTCRI. This value will be used by the force steal routine (Step 4) as the criterion by which RSM will begin stealing pages.	IRARMSTM	IRARMPR1
6 If storage load balancing is active, monitor storage utilization and calculate the storage load balancer swap factor based on the long term average maximum UIC and the available frame count average.		STORUSE

DIAGRAM SRM-16. IRARMSTM, IRARMST2 - Storage Management (5 of 14)



Extended Description	Module	Label
7 The real page frame shortage force steal routine is a co-operative effort between the SRM and RSM. SRM calls the RSM routine, IARXSF, passing a parameter list consisting of the following:	IRARMSTM	IRARMPR5
• An in-storage user's ASCB address		STEAL
• The number of frames needed from this address space		CKSIGUR
• The type of frames required		STORUSE
• A steal criterion number, MCVSTCRI	IRARMSRV	IRARMI03
• Destination indicators for changed and unchanged pages.	IRARMSTM	IRARMPR5
The SRM also passes to the RSM, in register 0, the total number of frames required.		
If real storage isolation is specified in the IPS for any address spaces or the common area, locate all such address spaces. If any address space or the common area exceeds its maximum working set size, frames are stolen until either the shortage is relieved or the working set sizes of the address space or the common area are equal to its maximum working set size. Also examine each swappable address space to determine if it is a significant user of storage (step 2).		
If a real page frame shortage still exists after trimming all such address spaces and the common area to their maximum working set sizes, frames will be stolen from logically swapped address spaces that were allowed to keep their standard working set pages and pages that were recently referenced. These users will be trimmed to the frames backing their working set. If a real page frame shortage still exists after trimming, frames will be stolen from the common area and all address spaces on the SRM IN queue.		
Since MCVSTCRI is the largest UIC in the system, the steal criteria starts at this UIC. Frames with a UIC greater than or equal to the steal criteria are stolen from each user. During this processing, if real storage isolation is applicable to a user or the common area, frames will not be stolen if it would reduce the user's or the common area's working set below its target working set size (assuming that enough frames can be stolen to relieve the shortage without violating this rule). If frames are still required after all eligible user's frames have been stolen at this criteria, the steal criterion will be decreased and the process repeated until no further frames are required.		
7 *(continued)*		
The result of this procedure is that the oldest unreferenced system-wide frames are stolen first. Frames in the common system area and link pack area are not associated with any specific user. RSM examines these frames when SRM page replacement calls it with an ASCB address of zero.		
The order of stealing at any criterion level is as follows:		
a) Address spaces that are capable of executing		
b) Common system area		
If a shortage still exists, additional frames are stolen in the following order:		
a) Working sets of logically swapped users		
b) Frames protected by storage isolation in UIC order		
Storage utilization is monitored (see step 6) after all steal processing is completed.		
Normally, SRM directs changed and unchanged pages to auxiliary or extended storage using the appropriate criteria table entries. Changed pages are either sent to extended storage or auxiliary storage, and unchanged pages are either sent to extended storage or discarded. If there is no extended storage online, or extended storage migration is constrained or overdue, changed pages are directed to auxiliary storage and unchanged pages are discarded. If frames are needed synchronously for a swap-in, changed pages are directed to extended storage and unchanged pages are directed by the criteria table.		
8 The SRM will have received an AVQLOW SYSEVENT if there is a shortage of real page frames. If the invocation is due to AVQLOW level 1, 2 or 3 (real page frame shortage), and an AVQOK SYSEVENT has not been issued, calculate the number of frames needed to get the available frame queue back to the OK threshold and invoke the forced steal algorithm via the algorithm request mechanism. IRARMMS2 will be rescheduled for execution via the algorithm request mechanism.	IRARMST2 IRARMCTL	IRARMMS2 IRARMCRL

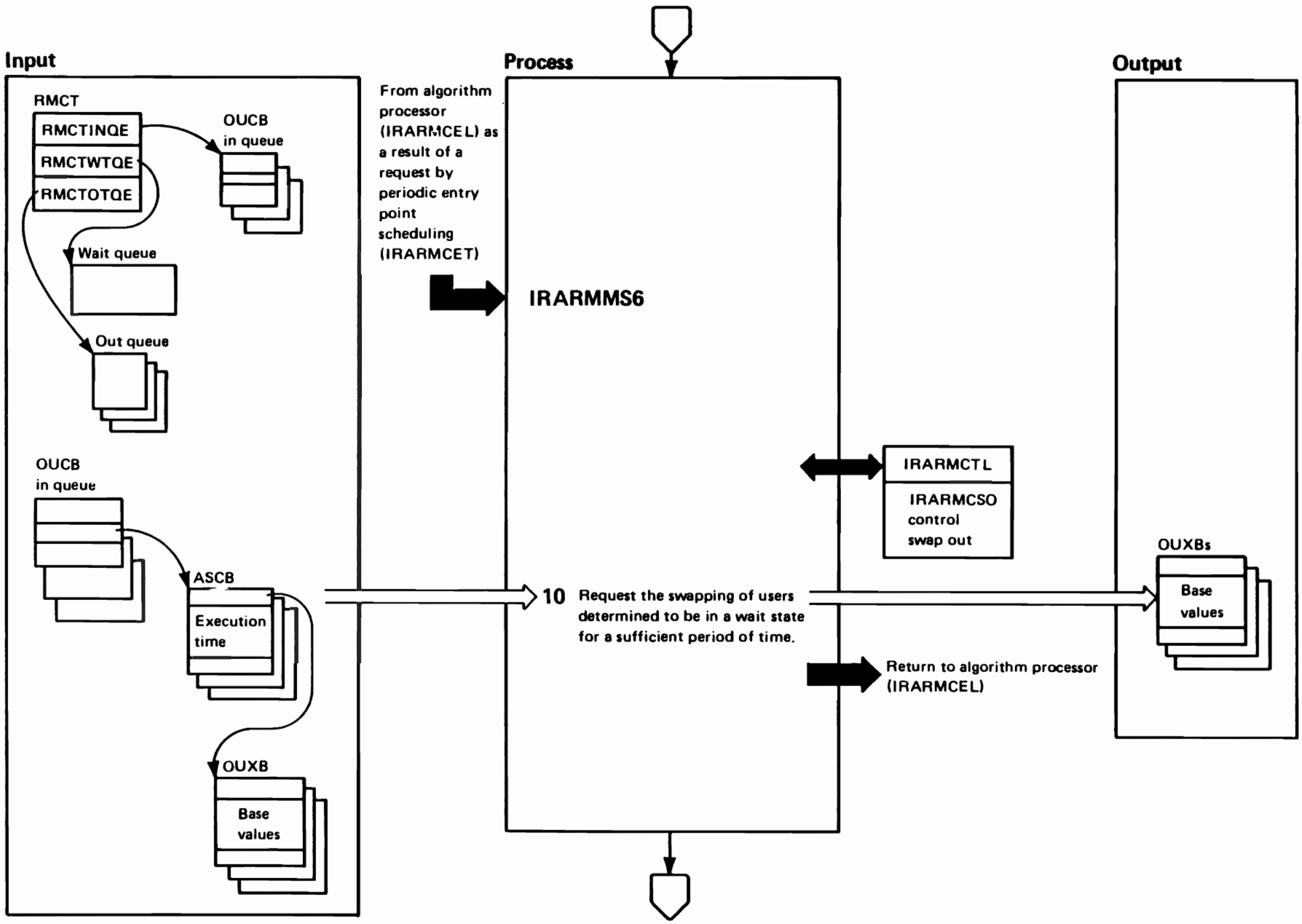
DIAGRAM SRM-16. IRARMSTM, IRARMST2 - Storage Management (7 of 14)



Extended Description	Module	Label
IRARMMS2 is periodically scheduled to calculate the logical swap available frame queue target, and if there are less available frames than the target, to process logically swapped users on the logical swap WAIT and OUT queues until the target is reached or there are no more such users. As frames are needed, users who are suspended for swap-out to extended storage are redriven. Users who are targeted for extended storage are trimmed, built (secondary WS), and swapped out. Users who are logically swapped by think time are physically swapped if their think time exceeds the system think time criterion.		
9 IRARMMS2 is scheduled to relieve a pageable storage shortage:		
• By IRARMMS5 routine when the pageable storage shortage thresholds have changed and a shortage exists with the new thresholds.	IRARMST2	IRARMMS2
• When a pageable real frame shortage is indicated by SYSEVENT ANQLOW le'. .I 4 from RSM.		
• When SRM detects a pageable real frame shortage.		
The shortage processing checks for two levels of shortages, an initial shortage condition and a critical shortage condition. A message describing the current shortage level is written to the console. When the pageable frame shortage is relieved, an additional message is written to indicate the alleviation of the shortage.	IRARMSRV	IRARM109

Extended Description	Module	Label
9 *(continued)* • If the shortage is due to fixing, the address spaces using large amounts of fixed storage are swapped out. If the shortage is due to paging, the address spaces using the largest amounts of real storage are swapped out. A message is issued to the system operator indicating the amount of real storage allocated to the address space and the subset of this amount that is fixed. • If the shortage was due to fixing and enough fixed storage has not been reclaimed, some or all of the logically swapped address spaces are swapped out. • If a shortage due to fixing still exists after all of the logically swapped address spaces have been swapped out, the multiprogramming level(s) (MPLs) for the domain(s) with the lowest contention index is reduced. Swap analysis is then scheduled to swap out the address spaces with the lowest recommendation values.		

DIAGRAM SRM-16. IRARMSTM, IRARMST2 - Storage Management (9 of 14)



Extended Description	Module	Label
10 Users who issue a long wait macro instruction will be detected by SRM when the wait macro processor issues the NIOWAIT SYSEVENT. Users who do not issue a long wait macro instruction to notify SRM that they will be in the wait state for a long time will be detected by IRARMMS6 when they satisfy the following conditions: • they have gone without executing for a sufficient period and • they have no ready work or they have vector work and no processors with a Vector Facility currently online. SRM might allow these users to be logically swapped in case the user becomes active again. The decision to logically swap or not is based on the current storage usage.	IRARMST2	IRARMMS6

DIAGRAM SRM-16. IRARMSTM, IRARMST2 - Storage Management (11 of 14)

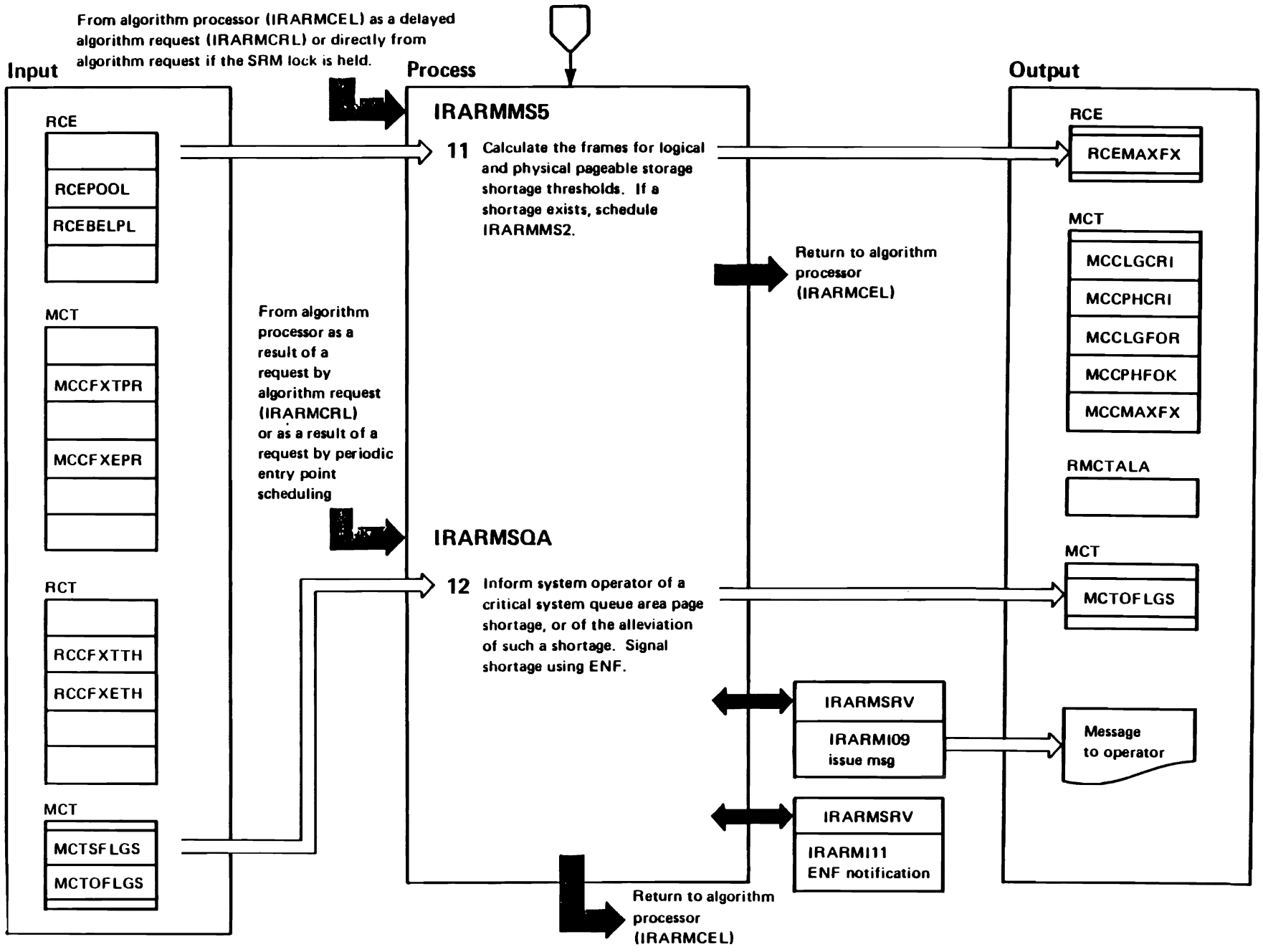


DIAGRAM SRM-16. IRARMSTM, IRARMST2 - Storage Management (12 of 14)

Extended Description	Module	Label
11 IRARMMS5 is called by RSM via SYSEVENT 16 (RSMCNSTS) when the real storage configuration is initialized or changed and by SET OPT processing (IRARMSOP) when the pageable storage shortage threshold percentages might have changed.		
This routine calculates the logical and physical shortage thresholds and the shortage-relieved thresholds from the online frame counts (RCEPOOL and RCEBELPL) and the threshold percentages. If a shortage exists with the new thresholds, IRARMMS2 is scheduled to relieve the shortage.	IRARMST2	IRARMMS5
	IRARMCTL	IRARMCRL
12 The system queue area message writer is invoked by SYSEVENT SQALOW or SQAOK to write shortage messages to the system operator. Messages are also written when SQA expands into CSA, warning the operator of possible SQA fragmentation. The messages cannot be written directly by the invoking routines because the SRM lock must be held. The SRM does not permit the creation of new address spaces when an SQA shortage exists. IRARMSQA issues an ENF signal (event code 4) whenever the routine is entered because a shortage situation has occured or the timer interval for the routine has expired.	IRARMST2	IRARMSQA
	IRARMEV2	IRARME25
	IRARMEV2	IRARME26

DIAGRAM SRM-16. IRARMSTM, IRARMST2 - Storage Management (13 of 14)

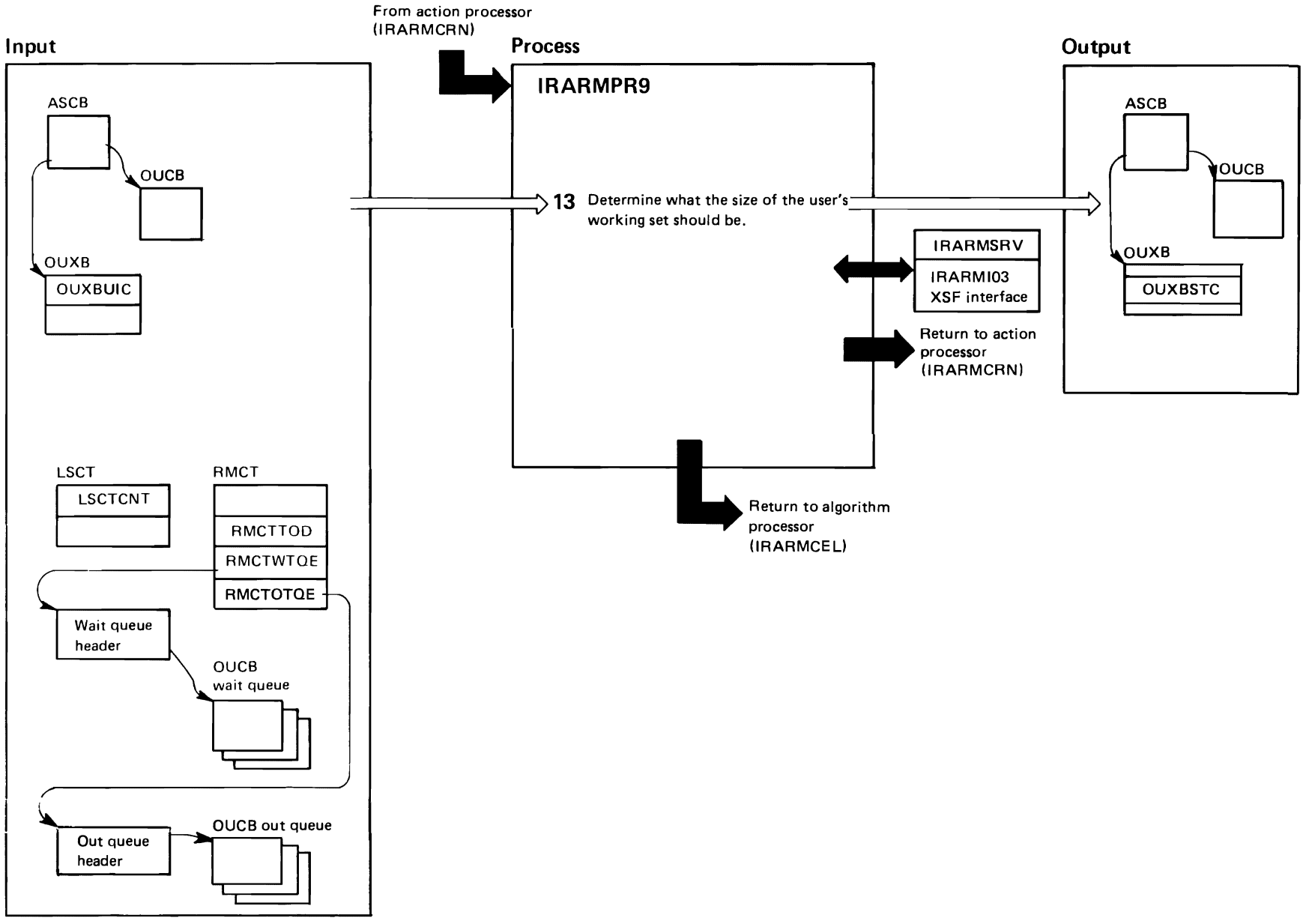


DIAGRAM SRM-16. IRARMSTM, IRARMST2 - Storage Management (14 of 14).

Extended Description	Module	Label
13 This routine is called to trim users who are being logically swapped by think time, swapped to auxiliary storage, or swapped to extended storage. Trim pages are directed to extended storage or auxiliary storage by the swap trim entries in the criteria table. Users being logically swapped by think time are trimmed to a level based on storage contention and, for storage-isolated users, not below their minimum working-set size. Users being swapped to auxiliary storage are trimmed to remove all unreferenced pages which are not protected by a minimum working-set size, if any. Users being swapped to extended storage are trimmed and secondary working set pages are built using a target provided by the caller (MCVPR9TG) as the maximum number of pages to steal. The user is trimmed to a level based on storage contention, and, for storage-isolated users, not below its minimum working-set size. If more pages are to be stolen, secondary working set is built on extended storage for the remaining pages. If the user's secondary working set is built, bit OUCBSWSB is set on. For an address space not logically swapped or not protected by storage isolation, IRARMPR9 leaves the determination of the working set to the RSM. RSM places all pages that have the reference bit set to 1 in the swap-in working set. For logically-swapped address spaces, the user's working set size is trimmed to a level based on storage contention. The logically swapped users that are protected by storage isolation are not trimmed below their minimum working set size.	IRARMSTM IRARMSRV	IRARMPR9 STEAL IRARMI03
13 *(continued)* For an address space that is physically swapped and is protected by storage isolation, the SRM must ensure that the RSM does not trim a user below the minimum working set size for the address space. The SRM trims the pages to the minimum working set size and then notifies RSM to include the remaining pages in the working set. For a logically swapped address space that is (1) not protected by storage isolation and (2) now being physically swapped, the SRM trims the user's working set so that all the pages that have the reference bit set have a UIC of 0.		

IRARMIOM MODULE ENTRY POINT SUMMARY

IRARMIL0	I/O Load Balancing User I/O Monitoring. Compute I/O use profile for all swappable problem state users based on their use of each logical path.
IRARMIL1	Identification of Significant I/O Users. If I/O load balancing is active and I/O is out of balance, identify significant I/O users and ensure that their use of the logical paths is being monitored.
IRARMIL3	I/O Load Balancing User Swap Evaluation. Compute numerical recommendation value which reflects desirability of swapping a user based on its logical path utilization.
IRARMIL4	I/O Load Balancing IMCB Deletion Routine. Clean up control blocks used in monitoring a heavy I/O user at the end of the user job step.
IRARMIL5	Selective Enablement Monitoring. Periodically determine the percent of I/O interruptions currently being processed by the Test Pending Interrupt (TPI) path in IOS and determine if this percent is within the thresholds specified in the IEAOPTxx member of SYS1.PARMLIB.
IRARMIL6	Modify Processor I/O Enablement. Search for a processor whose I/O enablement status can be switched to the state specified in the input parameters to the routine.

DIAGRAM SRM-17. IRARMIL0 - I/O Load Balancing User I/O Monitoring (1 of 4)

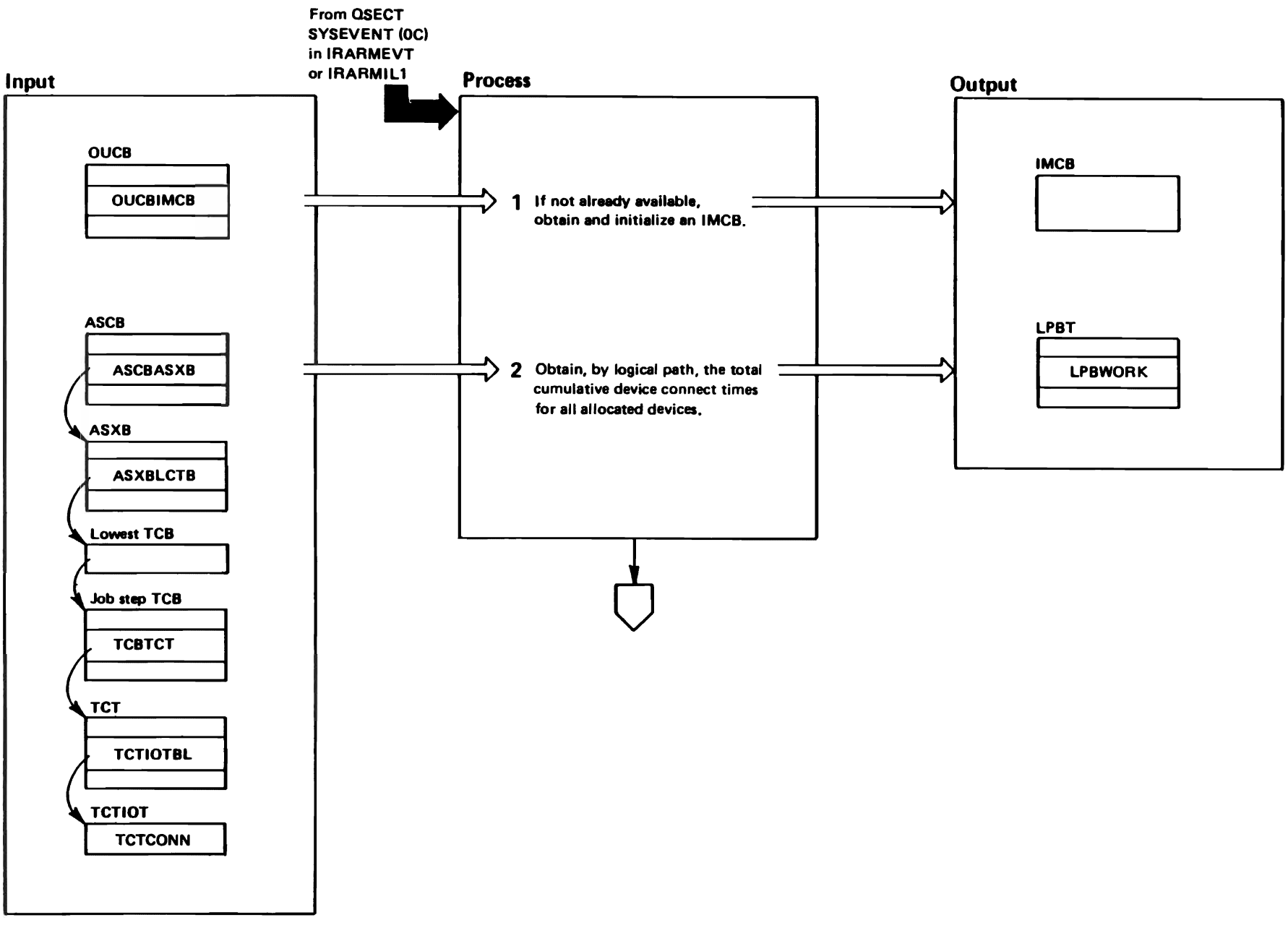


DIAGRAM SRM-17. IRARMIL0 - I/O Load Balancing User I/O Monitoring (2 of 4)

Extended Description

I/O load balancing user I/O monitoring maintains detailed information on logical path (LPB) utilization for heavy I/O users. This LPB information is used by other I/O load balancing functions to influence swapping decisions when heavy users are using logical paths that are not out of balance (either over used or under used).

This monitoring is done during processing of the quiesce-started SYSEVENT (SYSEVENT 0C) or during a periodic check for significant I/O users (IRARMIL1). IRARMIL1 issues a cross memory macro instruction to access the I/O timing control table (TCTIOT). The TCTIOT is located in the user's private area and contains monitoring source data. (See Diagram SRM-19).

1 If I/O load balancing is active and the user does not have an I/O measurement control block (IMCB), obtain an IMCB if the user's total I/O percent connect time is high enough (that is, higher than ICCMNIOR).

2 Access TCTIOT, looking at all user data set declarations, and access all devices allocated to each data set. Through the UCB, associate the device with a logical path, and sum the user's device connect time by logical path using LPBWORK as a working variable for the summation.

Module	Label
IRARMIOM	IRARMIL0

DIAGRAM SRM-17. IRARMIL0 - I/O Load Balancing User I/O Monitoring (3 of 4)

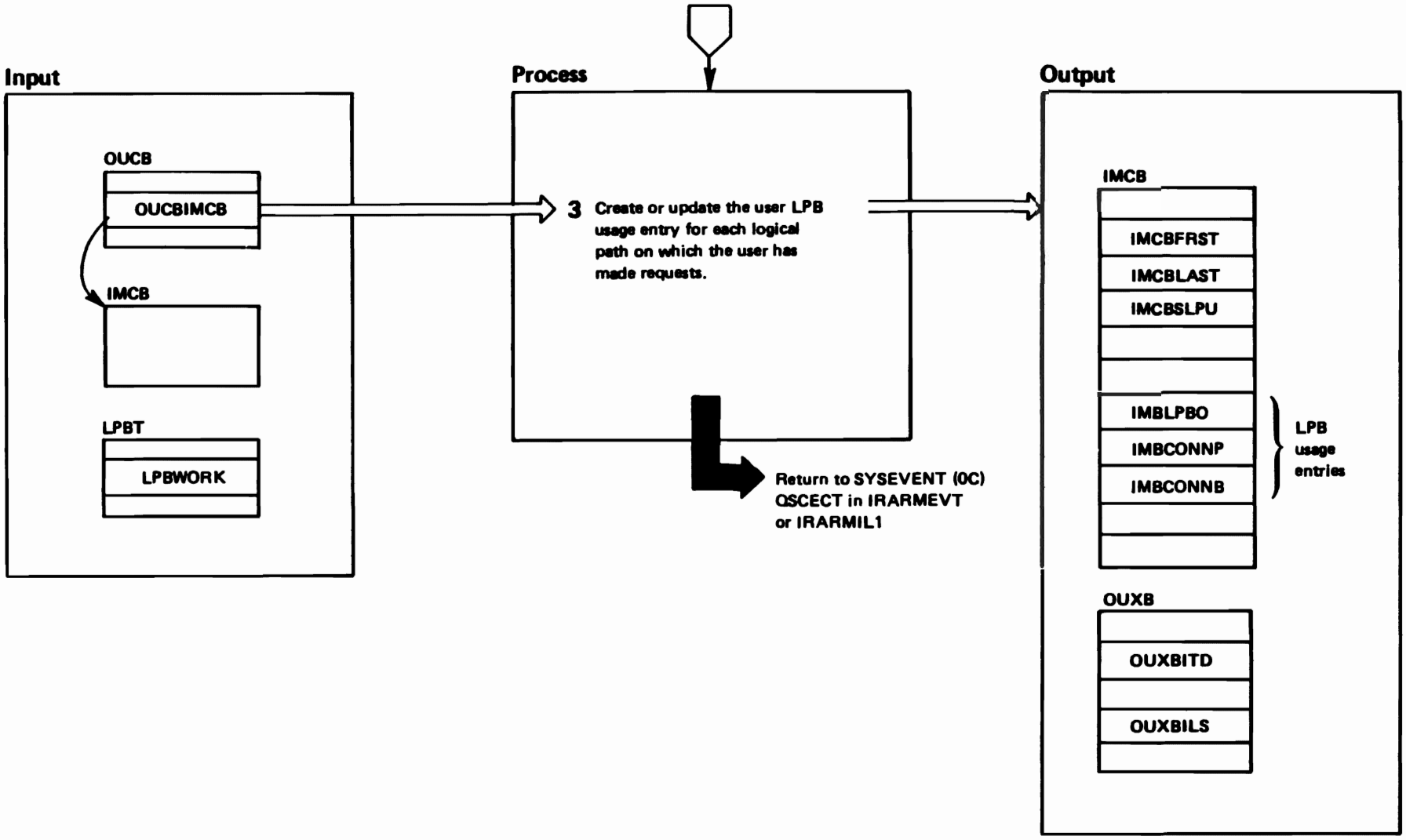


DIAGRAM SRM-17. IRARMIL0 - I/O Load Balancing User I/O Monitoring (4 of 4)

Extended Description	Module	Label
3 Step through all logical paths (LPBT entries), and determine if the user has been monitored for LPB utilization. That is, determine if an LPB entry is established in the IMCB. Update various utilization fields. If an IMCB LPB use entry is not established, make room for an entry by relocating other LPB use entries so that entries are kept in LPBT order. Various IMCB fields are initialized for the new entry.	IRARMIOM	IRARMIL0

Fields updated or created in the LPB use entries are:

- IMBLPBO -- offset to the associated LPB within the LPBT
- IMBCONNP -- percent of connect time for the LPB because of I/O activity for this user
- IMBCONNB -- I/O connect time base value

Fields updated in the OUXB are:

- OUXBITD -- I/O load balancing base time
- OUXBILS -- I/O service base connect time

Summary flags maintained in the IMCB are:

- IMCBSLPU -- indicates the user is active on at least one LPB that is either overused or underused.
- IMCBOVLP -- indicates the user is active on at least one overused LPB.
- IMCBUNLP -- indicates the user is active on at least one underused LPB.

DIAGRAM SRM-18. IRARMIL1 - Identification of Significant I/O Users (1 of 2)

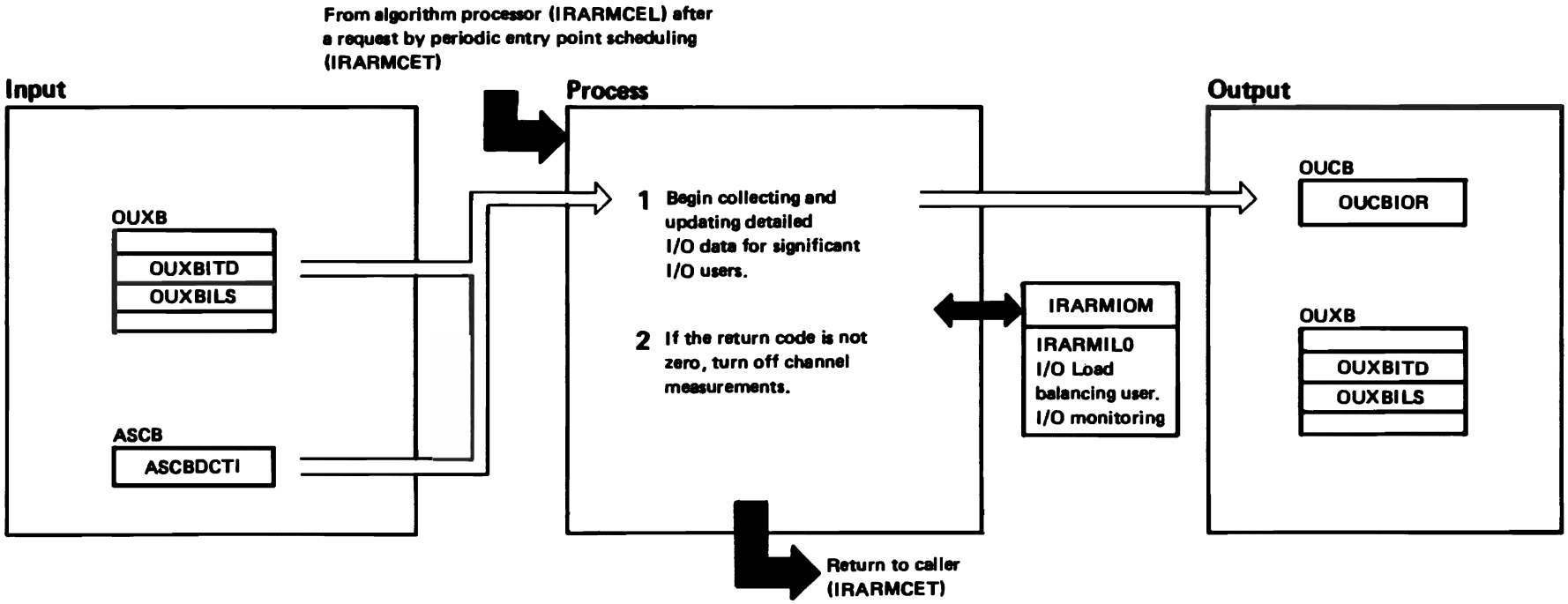


DIAGRAM SRM-18. IRARMIL1 - Identification of Significant I/O Users (2 of 2)

Extended Description	Module	Label
IRARMIL1 identifies significant I/O users when I/O is out of balance, ensures that their use of logical paths is monitored, and calls IRARMIL0 to update I/O use information.		
1 If I/O load balancing is active and I/O is out of balance, IRARMIL1 scans the users in real storage and calculates the recent rate of I/O requests to determine the significant users of I/O and ensures that their use of logical paths is being monitored. If the rate of use is high enough or if the user has been previously monitored, IRARMIL1 calls IRARMIL0 to update the I/O use data for this user in the IMCB of this user's address space. To update the I/O use data, IRARMIL0 checks the UCBMBI, which is an index to the channel measurement block (CMB) and the device vector table (DBVT).	IRARMIOM	IRARMIL1
	IRARMIOM	IRARMIL0
2 If an error condition has occurred, that is, the IRARMIL0 passes a return code of 4 to IRARMIL1, which calls IRARMMSP to turn off channel measurements.		

DIAGRAM SRM-19. IRARMIL3 - I/O Load Balancing User Swap Evaluation (1 of 2)

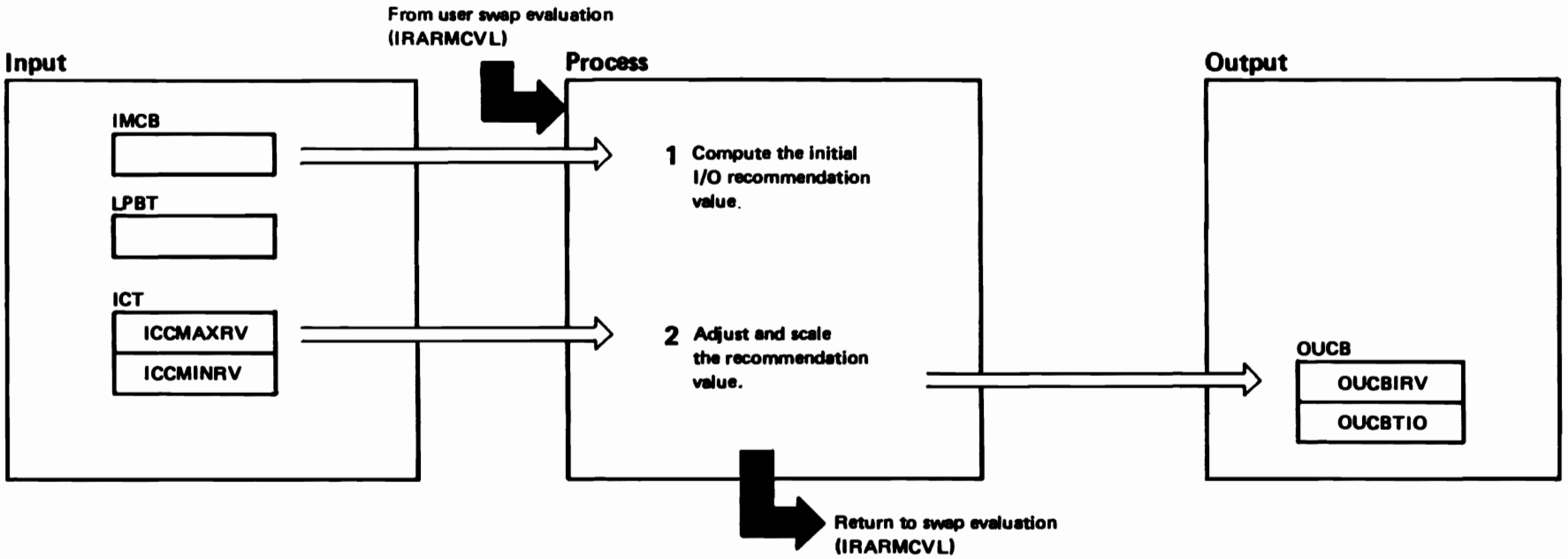


Diagram SRM-20. I/O Load Balancing User Swap Evaluation (IRARMIL3) (Part 2 of 2)

Extended Description	Module	Label
The user swap evaluation routine determines the I/O recommendation value for a user if: • I/O load balancing is in effect, as specified in the user's performance group period. • The IOC coefficient in the IEAOPTxx SYS1.PARMLIB member is non-zero. The recommendation value depends on the degree to which each LPB is out of balance and the user's contribution to the utilization of each LPB.		
1 An initial recommendation value is computed for each entry in the user's I/O measurement control block (IMCB) using the LPB recommendation value and the percent of LPB use due to the user. IRARMUPD, the channel measurement update routine, periodically calculates the LPB recommendation value and flags the LPB if necessary, as being overused or underused depending on the threshold specified in the IEAOPTxx member of SYS1.PARMLIB.	IRARMIOM	IRARMIL3
2 The initial recommendation is adjusted to produce a value that is not greater than twenty percent of the largest workload level.	IRARMIOM	IRARMIL3

DIAGRAM SRM-20. IRARMIL5 - Selective Enablement Monitoring (1 of 2)

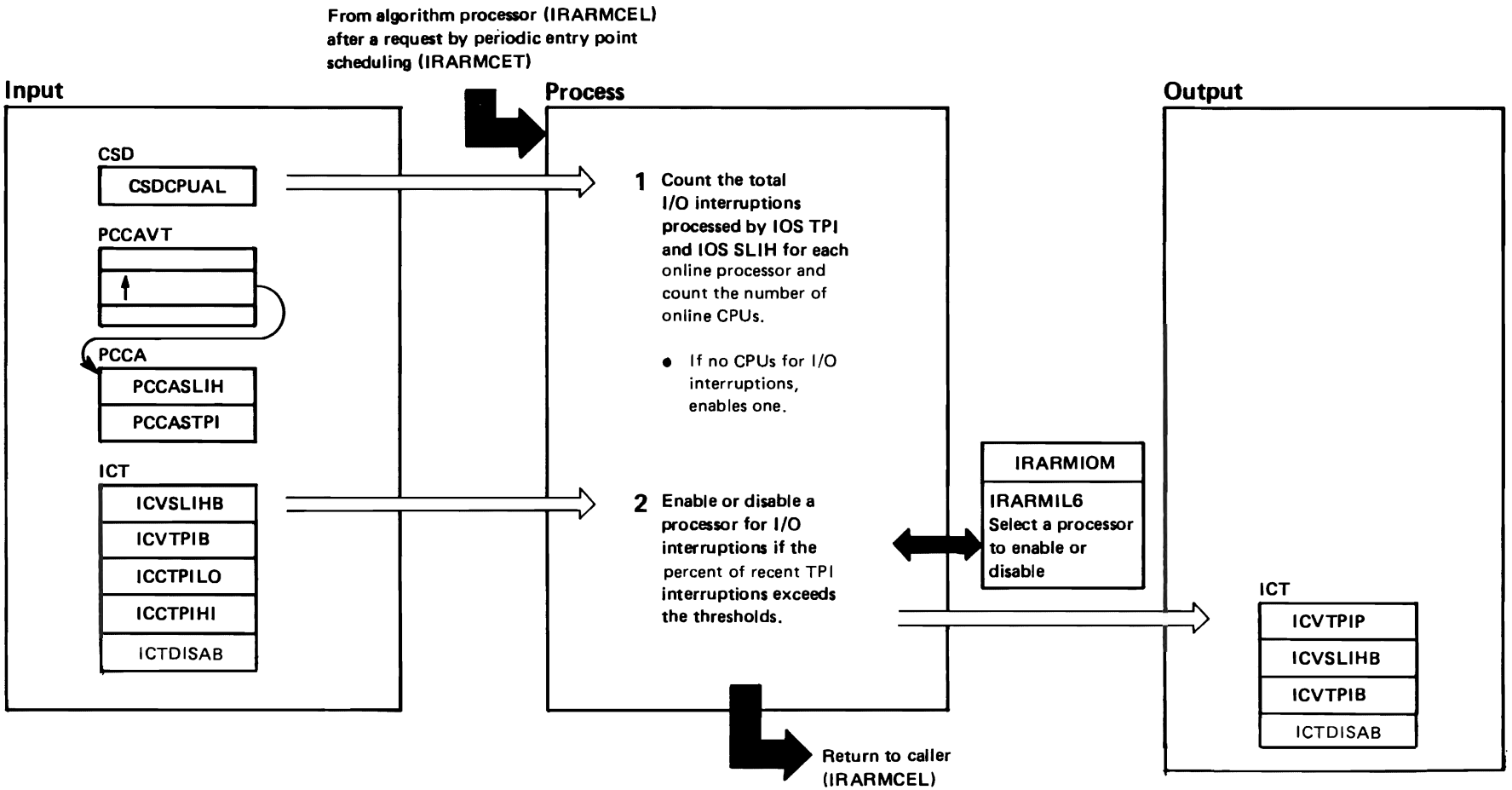


DIAGRAM SRM-20. IRARMIL5 - Selective Enablement Monitoring (2 of 2)

Extended Description	Module	Label
IRARMIL5 periodically checks to see if the number of processors enabled for I/O interruptions can effectively process the I/O workload.		
1 IRARMIL5 determines the number of I/O interruptions taken as interruptions (SLIH) and the number taken by the Test Pending Interrupt (TPI) instruction for each online processor. In addition, IRARMIL5 determines the number and type of processors that are online and enabled for interruptions. If there are no CPUs enabled for I/O interruptions, IRARMIL5 enables one.	IRARMIOM	IRARMIL5
2 IRARMIL5 calculates the changes in the TPI and the SLIH counts since this routine last ran. The routine then computes the percent of recent TPI interruptions compared to the total interruptions. If the percent exceeds the thresholds, call IRARMIL6 to enable or disable another processor. Otherwise, return to caller.	IRARMIOM	IRARMIL5

DIAGRAM SRM-21. IRARMIL6 - Modification of Processor I/O Enablement (1 of 2)

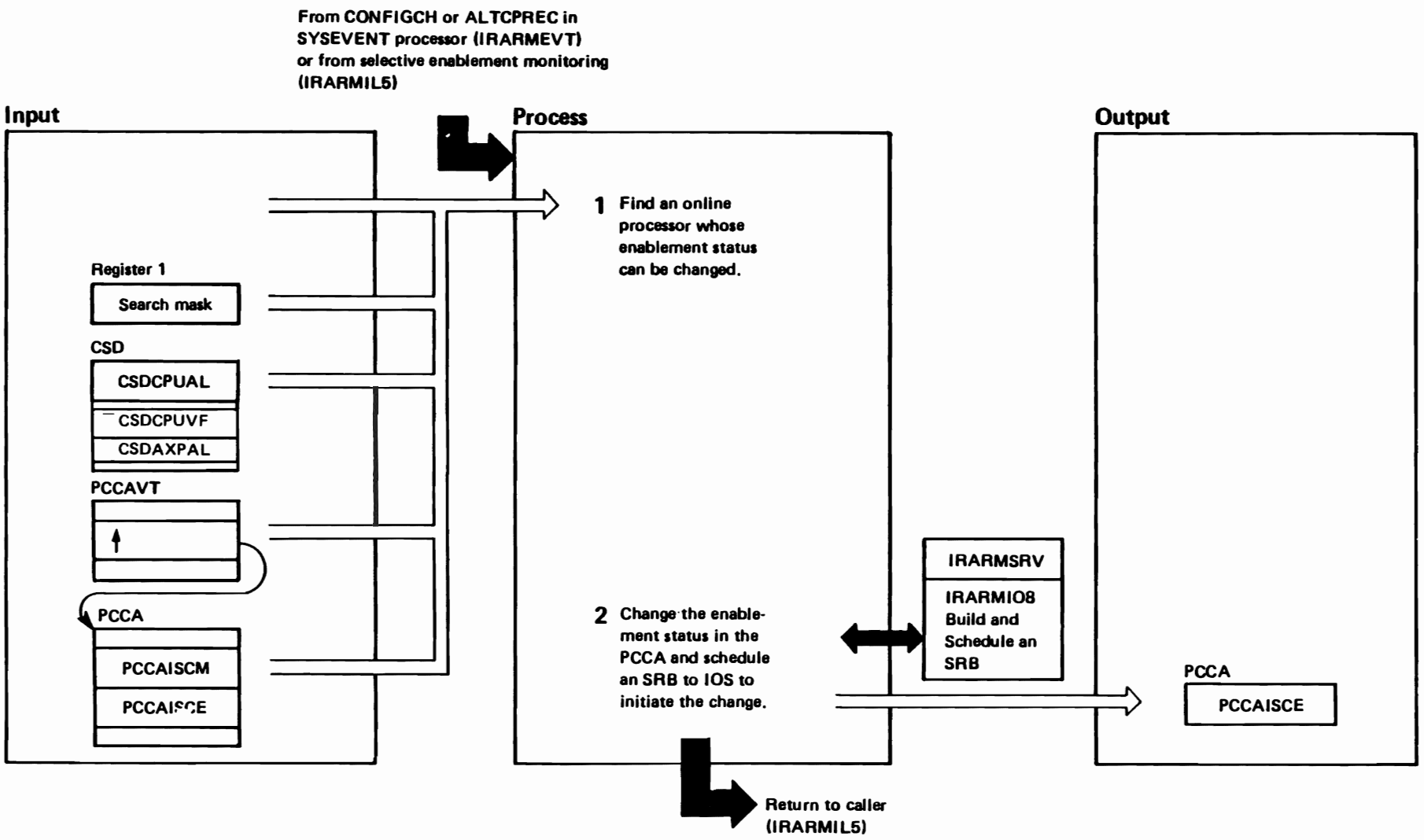


DIAGRAM SRM-21. IRARMIL6 - Modification of Processor I/O Enablement (2 of 2)

Extended Description	Module	Label
A change to the I/O enablement status of a processor is required for one of the following reasons: • The selective enablement monitoring routine determined that the percentage of interruptions is not within the acceptable thresholds. • The CONFIGCH or ALTCPREC SYSEVENT indicates that an enabled processor is being taken offline. The search mask input parameter indicates whether a processor can be enabled or disabled.		
1 Using the CPU affinity mask and the search mask as input, IRARMIL6 examines the PCAAs of each online processor to search for a processor with an I/O enablement status that can be changed as required by the selective enablement monitoring routine (IRARMIL5) or the CONFIGCH or ALTCPREC SYSEVENTs.	IRARMIOM	IRARMIL6
2 If it finds an eligible processor, IRARMIL6 changes the enablement state in PCCAISCE and passes control to IRARMI08. IRARMI08 builds an SRB and schedules the SRB to IOS. IRARMIL6 will choose the best candidate processors for I/O Enablement based on processor type and address. For enablement, non-vector processors are preferred over vector processors.	IRARMIOM IRARMSRV	IRARMIL6 IRARMI08

IRARMCHM MODULE ENTRY POINT SUMMARY

IRARMCHF	MSCH cleanup. Free SRB, SCHIB, and IOSB storage.
IRARMCPF	Measurement facility failure. Issue CHANNEL SYSEVENT when the channel reports an error in the measurement facility.
IRARMEDD	DDR ENF exit. Issue DDR SYSEVENT when a swap command is issued.
IRARMEVD	Vary device ENF exit. Issue UCBCHG SYSEVENT when a device is varied or boxed.
IRARMEVU	Vary path ENF exit. Issue UCBCHG SYSEVENT when a path is varied.
IRARMUPD	Channel Measurement Update Routine. Update DMBs, LPBs, and CPMT if the associated facilities are active. If measurements are "START PENDING" and 16 seconds have elapsed, resume channel measurements.
IRARMMSP	Channel Measurement Monitor Stop Routine. If device connect time interval (DCTI) measurement is active but is not available, issue a message to the operator and mark the control block unavailable. If the CMB monitor facility is active, issue a SCHM to turn CMB measurements off. Write LOGREC entry with reason code 65F. Set the monitor active bit off and mark the DMB and the LPB device data as invalid.
IRARMLPI	Channel Measurement Logical Path Block Index Routine. Calculate an LPB identifier from the channel path in the logical path mask of the UCB. Scan the LPBT for a match. If no match exists, build a new LPB at the end of the LPBT table.

DIAGRAM SRM-22. IRARMUPD - Channel Measurement Update Routine (1 of 4)

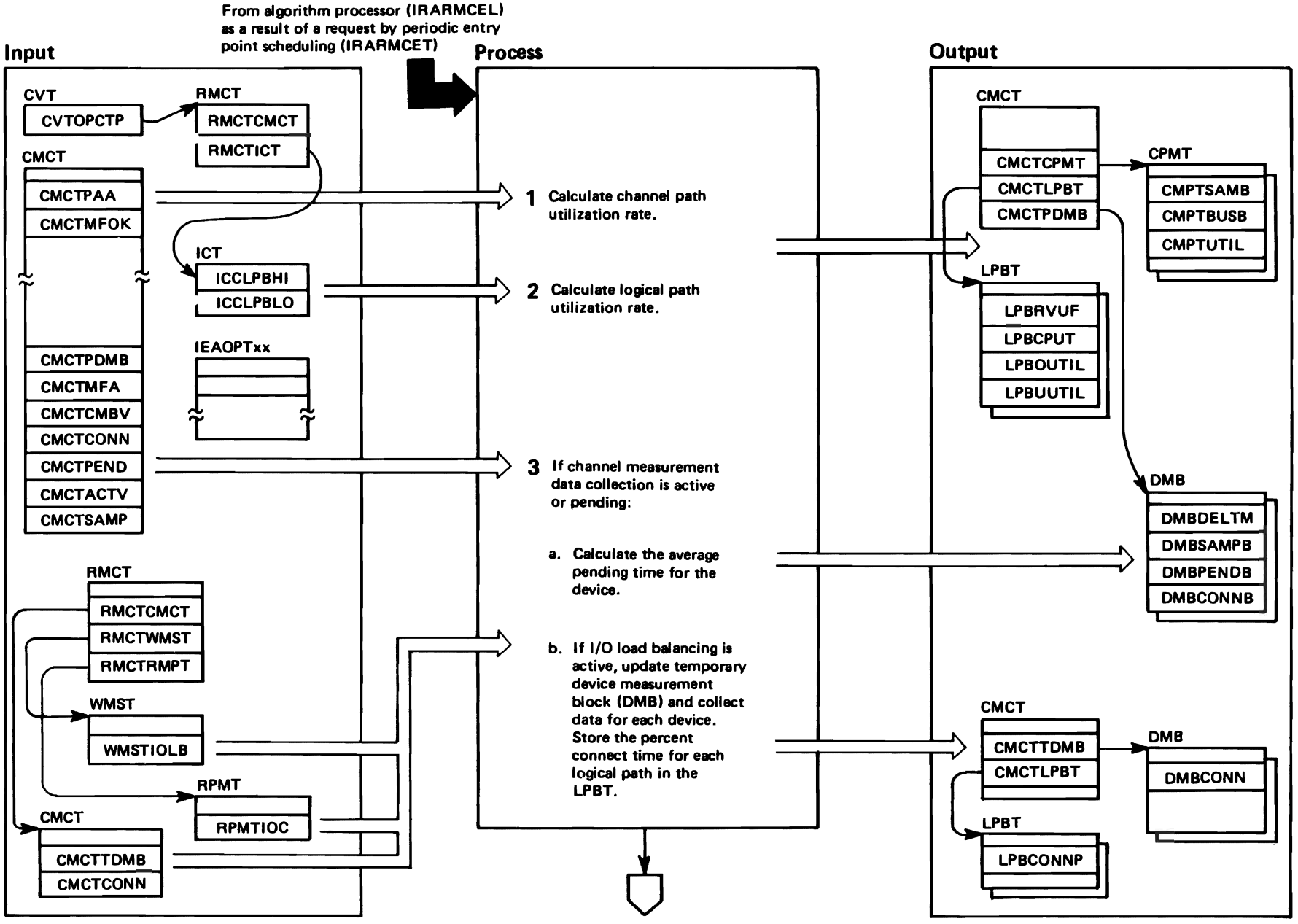


DIAGRAM SRM-22. IRARMUPD - Channel Measurement Update Routine (2 of 4)

Extended Description	Module	Label
When channel measurement is active, the channel measurement routine, IRARMUPD, executes every 3 seconds to update SRM channel path measruements.	IRARMCHM	IRARMUPD
1 The CMCTMFOK field indicates that the channel measurement block (CMB) entries contain current data, and the CMCTPAA field indicates that channel path measurement is active. When these fields in the channel measurement control table (CMCT) are set, IRARMUPD processes the channel path measurement table (CPMT) and calculates the utilization rate for all online channel paths. If a channel path is not online, IRARMUPD s 's the utilization rate to a negative number.	IRARMUPD	CPMTVPDT
2 IRARMUPD scans the logical path control block array and calculates the logical path utilization rate. IRARMUPD compares the logical path utilization rate with the thresholds defined in IEAOPTxx to determine if the device is either overused or underused.	IRARMUPD	LPBUPDT
3 a. If channel measurement data collection is active (CMCTMFA = 1) or pending (CMCTMFSP = 1), IRARMUPD scans the permanent device measurement control blocks (DMBs) and collects data for each device. IRARMUPD calculates the average pending time for each device. Device measurement control blocks (DMBs) are either permanent or temporary. Device allocation uses permanent DMBs; I/O load balancing uses temporary DMBs. b. If I/O load balancing is active, IRARMUPD scans the temporary DMBs and collects data for each device. IRARMUPD stores the percent connect time for each logical path in the logical path control block table (LBPT).		

DIAGRAM SRM-22. IRARMUPD - Channel Measurement Update Routine (3 of 4)

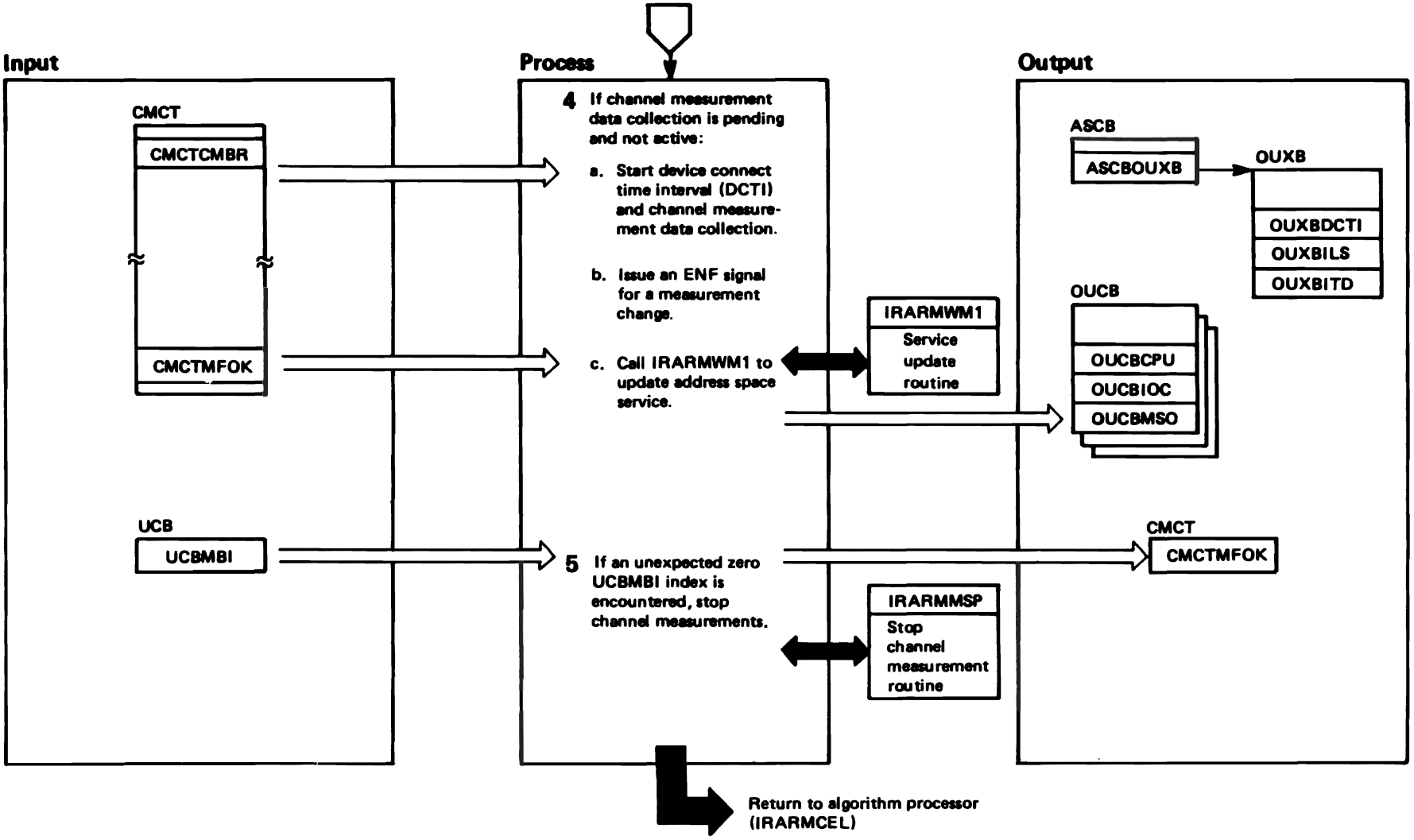


DIAGRAM SRM-22. IRARMUPD - Channel Measurement Update Routine (4 of 4)

Extended Description	Module	Label
4		
a. If channel measurement data collection is pending and not active (as long as no errors occurred during channel measurement initialization), IRARMUPD issues a set channel monitor instruction (SCHM) to start the device connect time interval (DCTI) and channel measurement data collection.	IRARMUPD	
b. IRARMUPD then issues an ENF signal to inform listeners that the SRM channel measurements are active. IRARMUPD sets the CMB (CMCTMFA) and the DCTI (CMCTDCA) collection bits to 1.	IRARMSRV	
c. IRARMUPD calls IRARMWM1 to update service for address spaces that have active transactions and to set base values for DCTI measurements.	IRARMWM1	
5 If an unexpected zero UCBMBI index is encountered, IRARMUPD calls IRARMMSP to stop channel measurements.	IRARMMSP	

DIAGRAM SRM-23. IRARMMSP - Channel Measurement Monitor Stop Routine (1 of 2)

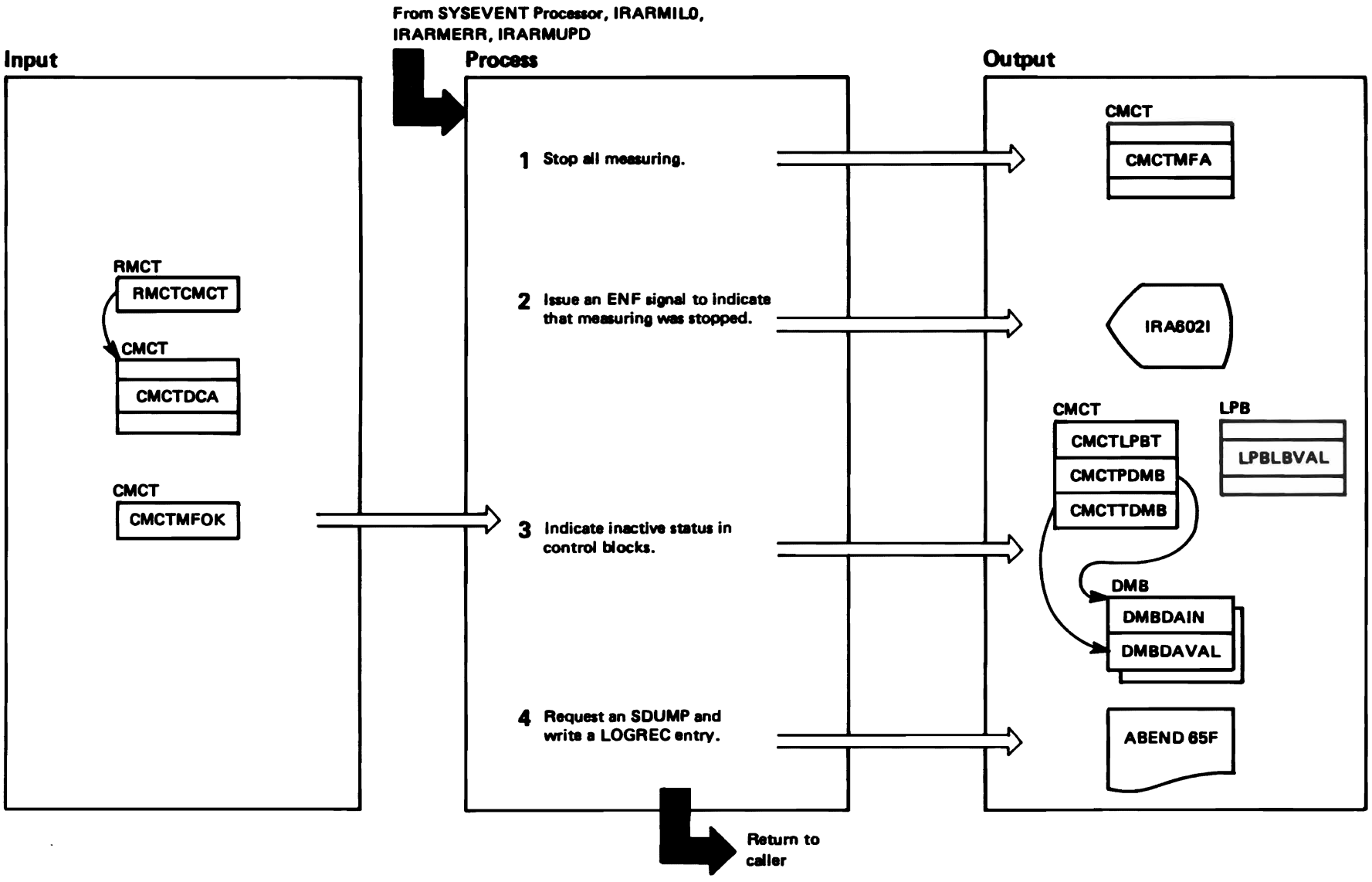


DIAGRAM SRM-23. IRARMMSP - Channel Measurement Monitor Stop Routine (2 of 2)

Extended Description	Module	Label
The channel measurement stop routine, IRARMMSP is invoked to terminate channel measurements.		
1 Terminate channel measurement by turning off CMCTMFA in the CMCT. If device connect time measurement is active, it remains active. However, if the TOD clock is invalid or the channel measurement control blocks are invalid, terminate device connect time measurement by means of a set channel monitor instruction (SCHM).	IRARMCHM	IRARMMSP
2 Issue an ENF signal to inform listeners of a change in the measurement status. A message is issued to the system programmer explaining why measurements were suspended.	IRARMSRV	IRARMI11
3 Update measurement control blocks to reflect the inactive status, and terminate I/O load balancing. If the channel measurement blocks (CMBs) contain valid data, terminate device allocation evaluation and logical path use evaluation.	IRARMCHM	IRARMMSP
4 If IRARMMSP is invoked because of an unexpected zero UCBMBI, a channel EXCP error, or insufficient virtual storage, IRARMMSP produces an SDUMP with a 65F ABEND code and writes a LOGREC entry.	IRARMERR IRARMCHM	IRARMMSP

DIAGRAM SRM-24. IRARMLPI - Measurement Logic Path Block Index Routine (1 of 2)

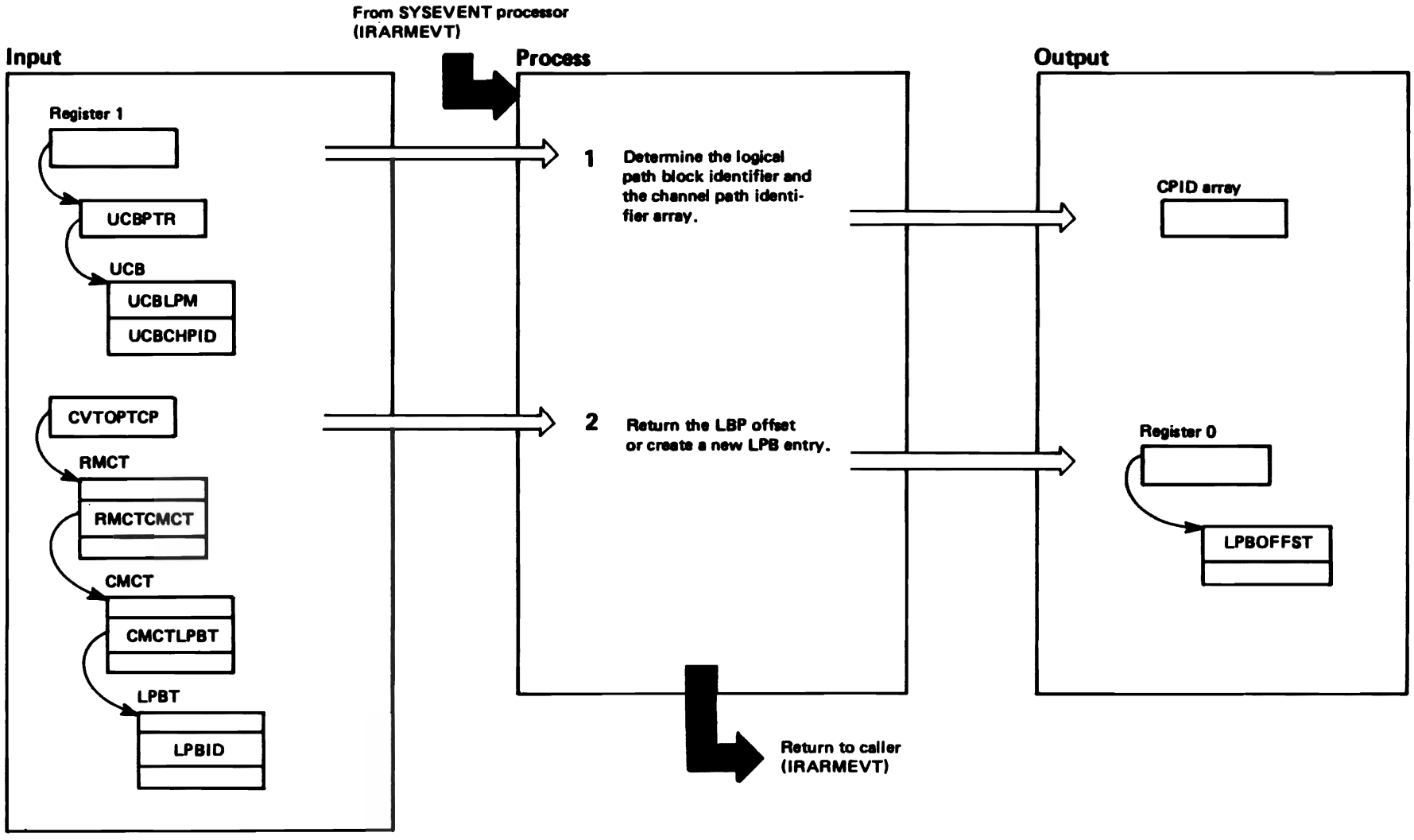


DIAGRAM SRM-24. IRARMLPI - Measurement Logic Path Block Index Routine (2 of 2)

Extended Description	Module	Label
The channel measurement LPB index routine, IRARMLPI, creates an LPB offset in the DMB when a DDR SYSEVENT (47) is issued or when a device is brought online or taken offline.	IRARMCHM	IRARMLPI

1 The LPB offset routine scans all the channel path ids associated with a device, looking for valid channel path identifiers. IRARMLPI arranges the valid channel paths identifiers in ascending order in a channel path array.

2 The LPB offset routine searches the logical path control block table (LPBT) looking for an LPB with a matching array. If no matching LPB exists, IRARMLPI builds a new LPB. The LPB offset is returned to the caller.

IRARMCPM MODULE ENTRY POINT SUMMARY

IRARMACT	Activate dispatching priority control. Examine user counts for time slice groups. Based on these counts, mark the time slice algorithm as active or inactive.
IRARMAP1	Automatic priority group recorder processing. Recompute dispatching priorities for all APG users in main storage.
IRARMAP2	ASCBCHAP standard interface routine. Search the IN queue for APG users whose dispatching priority needs updating.
IRARMEQ1	ENQ/DEQ algorithm ENQ time monitoring. Stop giving extra processor service to users with ENQHOLD SYSEVENTs outstanding who have already received their guaranteed processor service.
IRARMCL0	CPU load balancing user swap processing. Compute user processor use profile at QSCECMP SYSEVENT.
IRARMCL1	CPU utilization monitoring. Compute processor utilization variables for processor load balancing and resource management algorithms. Adjust workload skewing threshold.
IRARMCL3	CPU load balancing user swap evaluation. Produce a numerical recommendation value that reflects the desirability of swapping user based on processor use.
IRARMSWT	Switch to new dispatching priority control. Update the time slice tables to reflect a recent change in a specified user's dispatching priority control.
CHAP	IRARMCPM internal chapping subroutine. Build a parameter list and call ASCBCHAP.
CPLRVSWF	IRARMCPM internal wait factor computation subroutine. Compute system wait factor for processor load balancing recommendation value computation.
CPUWAIT	IRARMCPM internal wait time and CPU utilization compute subroutine. Compute accumulated system wait time total for processors and compute recent processor utilization.
CPUTLCK	IRARMCPM internal CPU utilization checking routine. Insure that the computed processor utilization percentage falls between 0 and 100 percent. If it is 100 percent and any user has not been dispatched, set the percent to 101. If a ready, dispatchable address that should be on the dispatching queue is not, invoke a safety net check routine.
NEWDP	IRARMCPM internal APG computation routine. Compute mean time to wait and new dispatching priorities for the APG user.

DIAGRAM SRM-25. IRARMCPM - CPU Management (1 of 8)

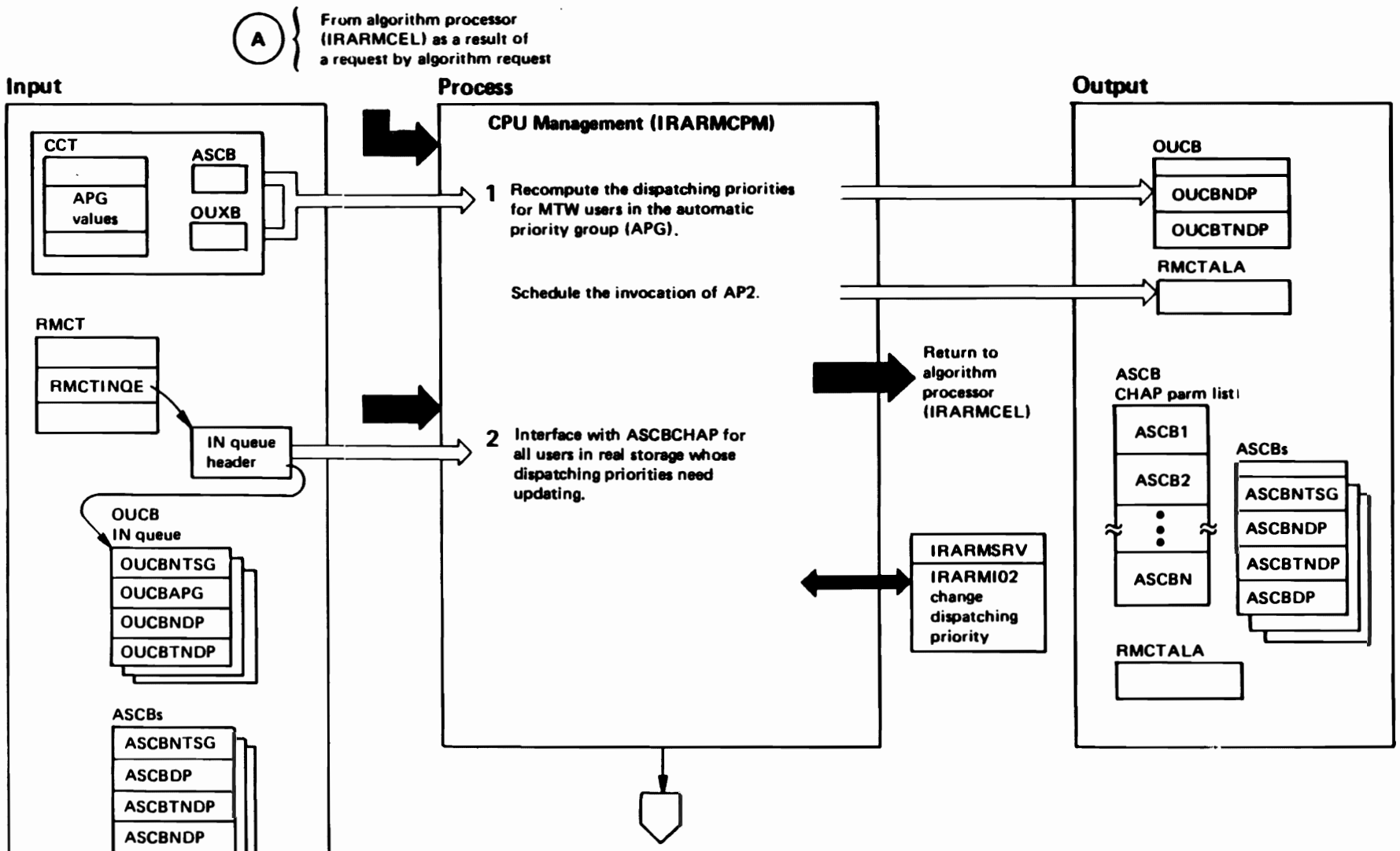
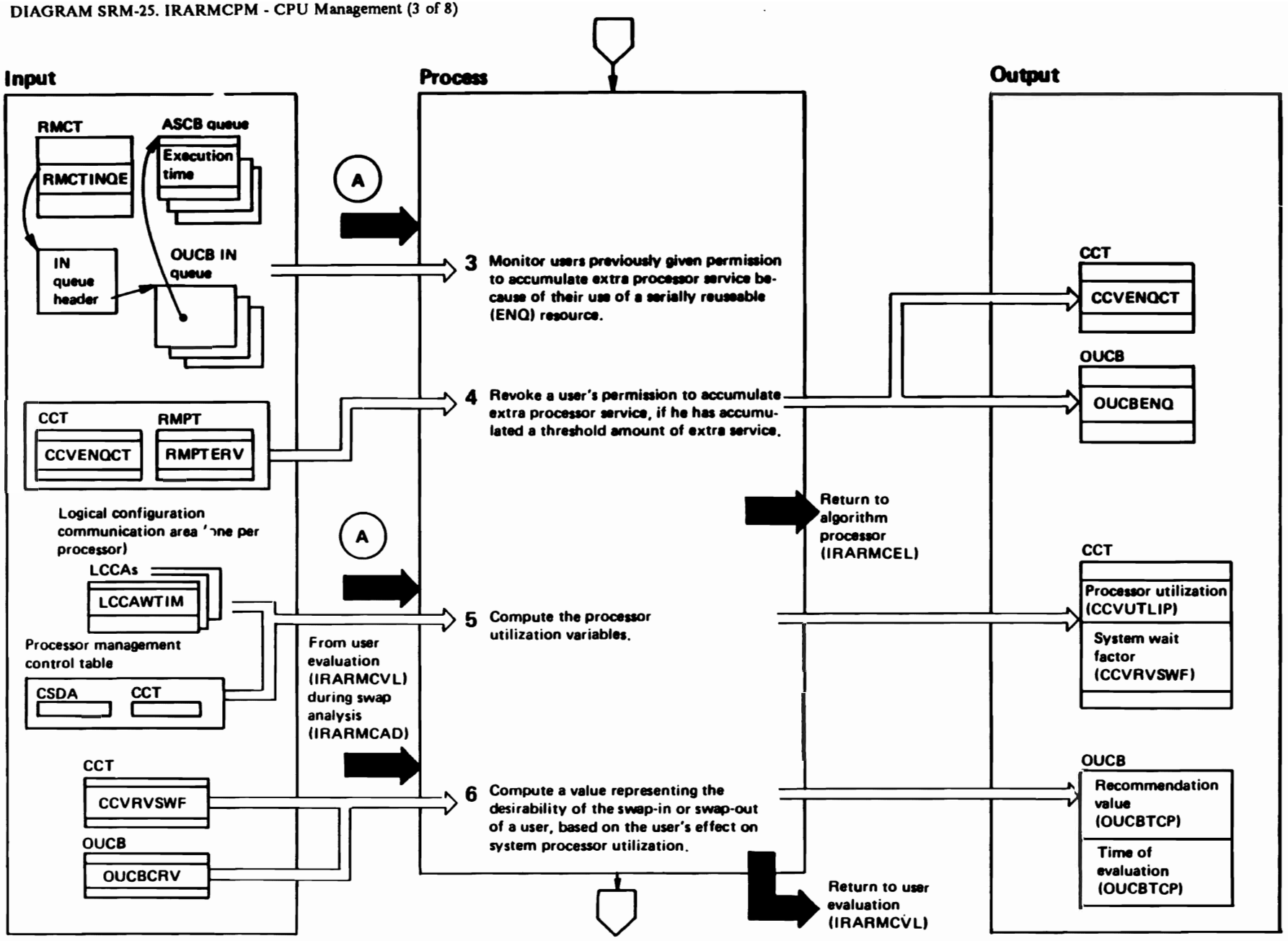


DIAGRAM SRM-25. IRARMCPM - CPU Management (2 of 8)

Extended Description	Module	Label (or Segment)
CPU management consists of a set of routines that monitor the system-wide processor load. They recommend users for swapping when the system is underused or overused.	IRARMCPM	
Processor load balancing system profile adjustment is performed with the SRM receives a QSCECMP SYSEVENT (0D).	IRARMCPM	IRARMCL0
1 IRARMAP1 searches for all users that are in real storage whose dispatching priority places them in a mean-time-to-wait group. AP1 then examines each user's execution time to ensure that the user has executed for more than a threshold value of time since the last computation. If the user has executed sufficient amount of time (CCAPMET), AP1 recalculates the base time slice dispatching priorities for the address space. The new priorities are calculated based on the user's mean-time-to-wait, that is, the average time the user was in execution before entering the wait state. The lower the mean-time-to wait, the higher the user's priority within the mean-time-to-wait group. AP1 then schedules the execution of IRARMAP2 (see step 2).	IRARMCPU	IRARMAP1 NEWDP
2 IRARMAP2 is the standard interface routine with ASCBCHAP. A parameter list is built for every in-storage APG user whose current dispatching priority does not equal its new dispatching priority. IRARMI02 is then invoked to interface with ASCBCHAP The invocation of IRARMACT is scheduled to ensure the timely activation or deactivation of time slicing (see step 8).	IRARMCPU IRARMSRV	IRARMAP2 IRARMI02

DIAGRAM SRM-25. IRARMCPM - CPU Management (3 of 8)



Extended Description	Module	Label (or Segment)
3 A user is given permission to accumulate extra processor service when an ENQHOLD SYSEVENT (14) is received by the SRM, indicating that the user holds a critical resource. The mechanism for giving the user this extra service is the prevention of his swap out by the SRM because of service rate considerations.	IRARMCPM	IRARMEQ1
4 The enqueue residence value (ERV), an OPT parameter, specifies the length of the privileged spurt of service for a user for whom an ENQHOLD SYSEVENT (14) has been issued (see 3). When this time is exceeded, the user is made eligible for swap-out, and his OUCB is so flagged. The individual user evaluation routine is called to assign a current workload manager recommendation value to this user.	IRARMCPM IRARMWLM	IRARMEQ1 IRARMWM3
5 The processor utilization is the average percentage of time any processor in the system was not in the wait state. It is computed by the following formula:	IRARMCPM	IRARMCL1

$$\text{utilization} = 100 - \left[\frac{(\text{sum of wait routines on each processor}) * 100}{(\text{elapsed time since last entry}) * (\text{number of processors})}\right]$$

Processor utilization is artifically set to 101% if actual utilization is 100% and one or more users have not been dispatched. This allows the processor to be considered overused even if the processor threshold is 100%. The system wait factor is calculated for use in determining the swap recommendation value for a user (see step 6); it is a multiple of the square of the difference between a threshold value and the utilization, with the sign indicating the direction of the imbalance (overused or underused). If processor use falls between the high and low thresholds, the factor equals zero.

CPUWAIT
CPLRVSWF

Extended Description	Module	Label (or Segment)
6 The processor swap recommendation value for a significant processor user varies with the degree to which the processor load is out of balance. The recommendation value cannot be greater than one-fifth the highest workload level. For insignificant processor users, the recommendation value is zero.	IRARMCPM	IRARMCL3
The time of this evaluation and the swap recommendation are saved in the OUCB. The user swap evaluation routine, IRARMCVL, then multiplies the recommendation value by the processor resource factor coefficient (RMPTCPU) to produce the final CPM swap recommendation value.	IRARMCTL	IRARMCVL

DIAGRAM SRM-25. IRARMCPM - CPU Management (5 of 8)

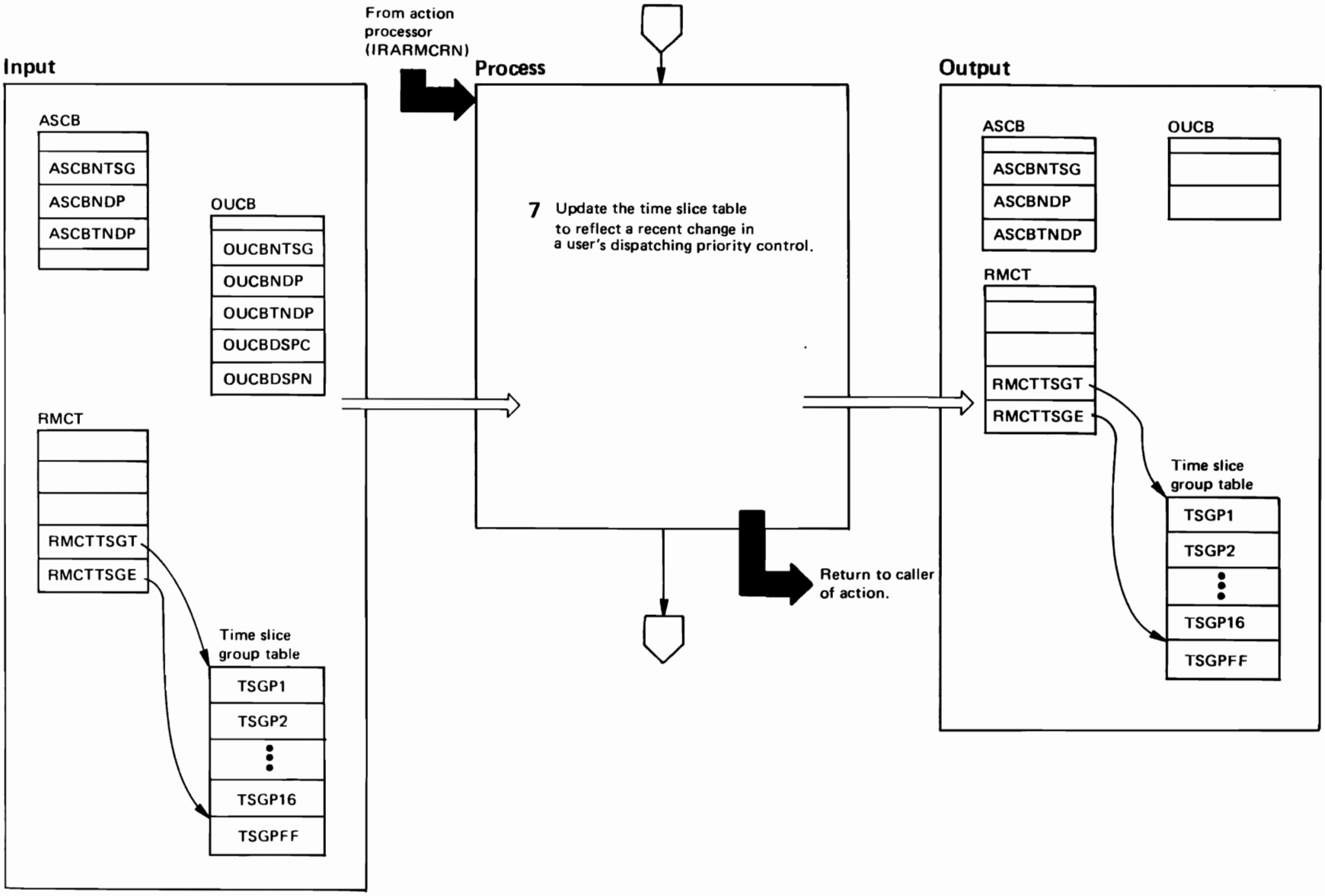


DIAGRAM SRM-25. IRARMCPM - CPU Management (6 of 8)

Extended Description	Module	Label (or Segment)
7 This segment updates the time slice table to reflect a recent change in this user's dispatching control. The user counts for the old dispatching control are decreased and counts for the new control are increased. The new dispatching control indicator is moded from the OUCB to the ASCB.	IRARMCPM	IRARMSWT

DIAGRAM SRM-25. IRARMCPM - CPU Management (7 of 8)

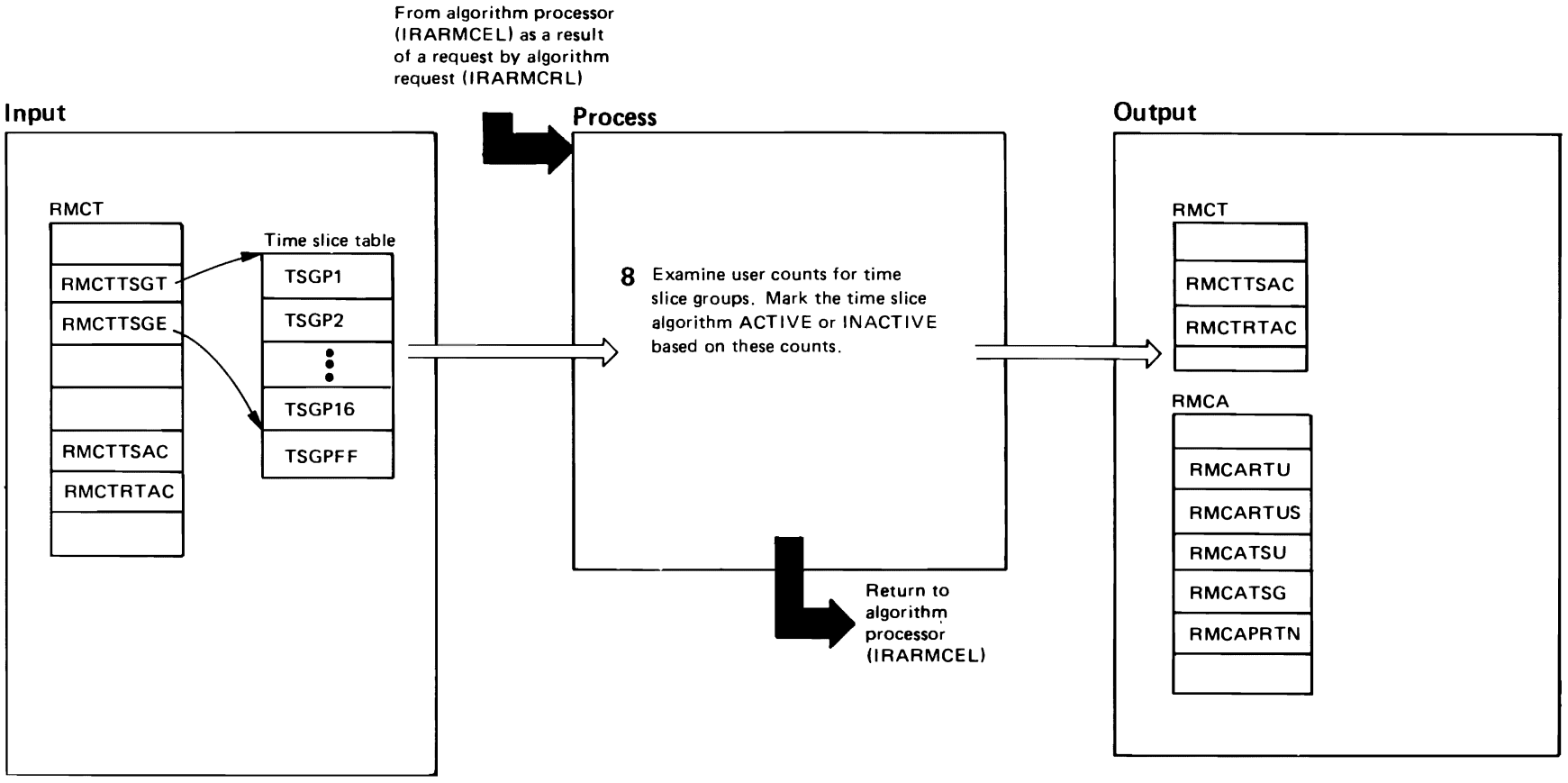


DIAGRAM SRM-25. IRARMCPM - CPU Management (8 of 8)

Extended Description	Module	Label (or Segment)
8 Next, examine the user counts for each time slice group. If any users are found to be in a time slice group, ensure time slicing is active. If no users are found, deactivate time slicing.	IRARMCPM	IRARMACT

IRARMRMR MODULE ENTRY POINT SUMMARY

IRARMRM1 Resource monitor periodic monitoring. Accumulate several system resource contention indicators and the number of ready users for each domain at periodic sample intervals.

IRARMRM2 Resource monitor MPL adjustment processing. Compute the average system resource utilization and determine if the system MPL should be raised or lowered. Adjust system maximum think time for demand swap. Ensure an adequate supply of double frame pairs on RSM's queue. For RTO domains, calculate the average elapsed time for TSO transactions ending in the first period. When enough samples have been accumulated, recalculate extended storage thresholds.

DIAGRAM SRM-26. IRARMRM1 - Resource Monitor Periodic Monitoring (1 of 2)

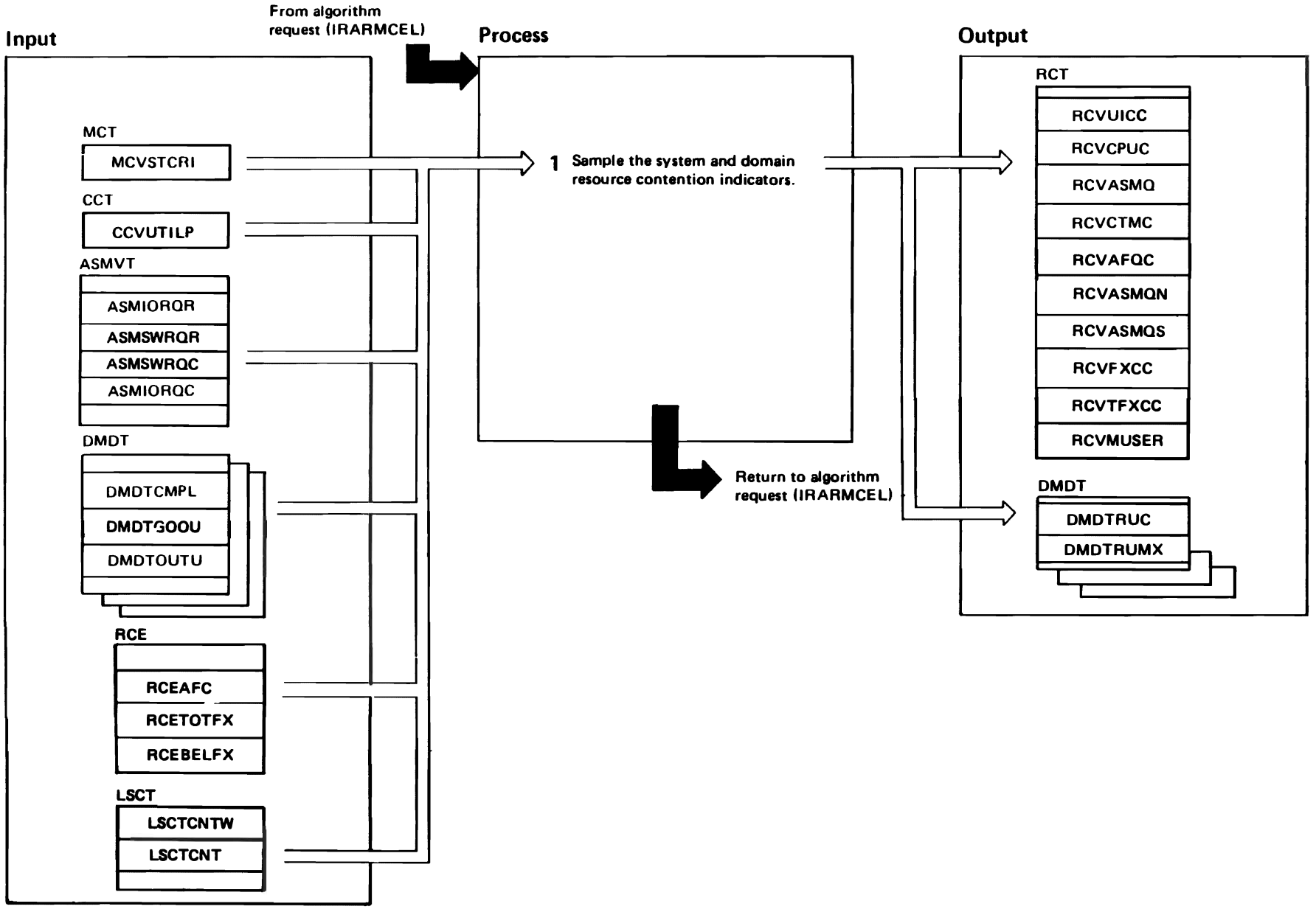
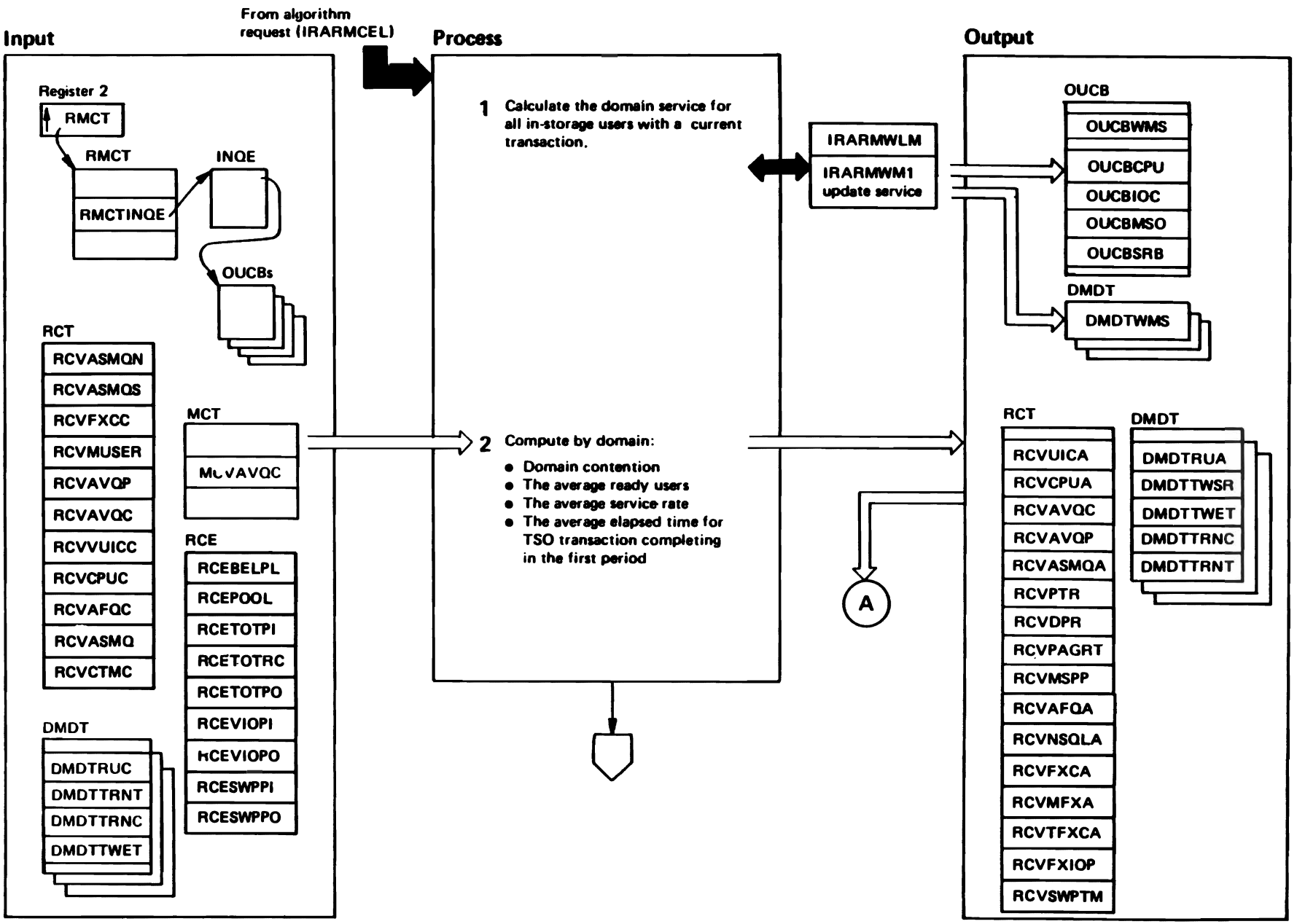


DIAGRAM SRM-26. IRARMRM1 - Resource Monitor Periodic Monitoring (2 of 2)

Extended Description	Module	Label
This routine is invoked at one second intervals and accumulates:	IRARMRMR	IRARMRM1
• The highest system unreferenced frame interval count (MCVSTCRI)		
• The current processor utilization (CCVUTILP)		
• The number of real storage frames available on the frame queue (RCEAFC)		
• The number of uncompleted ASM requests for both swap and nonswap queues (ASM requests minus ASM completed requests)		
• The number of fixed frames below 16 megabytes		
• The total number of fixed frames		
• The minimum number of available frames of extended storage		MCVAECMN
• The number of ready users (the number of users on the 'in' queue plus the number of users of the 'out' queue) for each domain		
Additionally,the maximum ready user count and the maximum in-storage user count since the last invocation of IRARMRM2 are calculated for each domain.		

DIAGRAM SRM-27. IRARMRM2 - Resource Monitor MPL Adjustment Processing (1 of 4)



Extended Description	Module	Label
This routine is invoked at 106 SRM-second intervals. It processes the data accumulated by IRARMRM1 to compute the following: • The average unreferenced frame interval count (RCVUICA) • The average available frame queue count • The number of AVQLOWs over the last RM2 interval (RCVAVQC, for tracking only) • The average percentage of storage below 16 megabytes that is fixed (RCVMFXA) • The average percentage of all storage that is fixed or allocated for nonswap paging (RCVFXIOP) • The average ASM queue length (RCVASMQA) • The system page fault rate per second (RCVPTR) • The system demand paging rate per second (RCVDPR) • The system total paging rate per second (RCVPAGRT) • The average time (in milliseconds) to process a non-swap paging request (RCVMSPP) • The average time (in milliseconds) to process a swap paging request (RCVSWPTM, for tracking only) If enough samples have accumulated, raise or lower the extended storage thresholds based on the percent of time that extended storage is exceeded. (RCEGROUP, RCEAECOK, RCEAECLO). • The time weighted average number of ready users for each domain (DMDTRUA) • The time weighted average service rate for each domain (DMDTTWSR) The time-weighted elapsed time for TSO transactions completing in the first period is calculated for each RTO (response time objective) domain (DMDTTWET). The above system and domain contention factors are used to adjust the domain target MPLs as follows:	IRARMRMR	IRARMRM2
1 To ensure that domain service is up-to-date, IRARMWM1 is called to update user service and to add the new increment to the appropriate domain. **2** DMDTRUA is calculated to one hexadecimal place and is a weighted average of the current and old values. DMDTTWSR and DMDTTWET are similarly weighted averages.	IRARMWLM	IRARMWM1

DIAGRAM SRM-27. IRARMRM2 - Resource Monitor MPL Adjustment Processing (3 of 4)

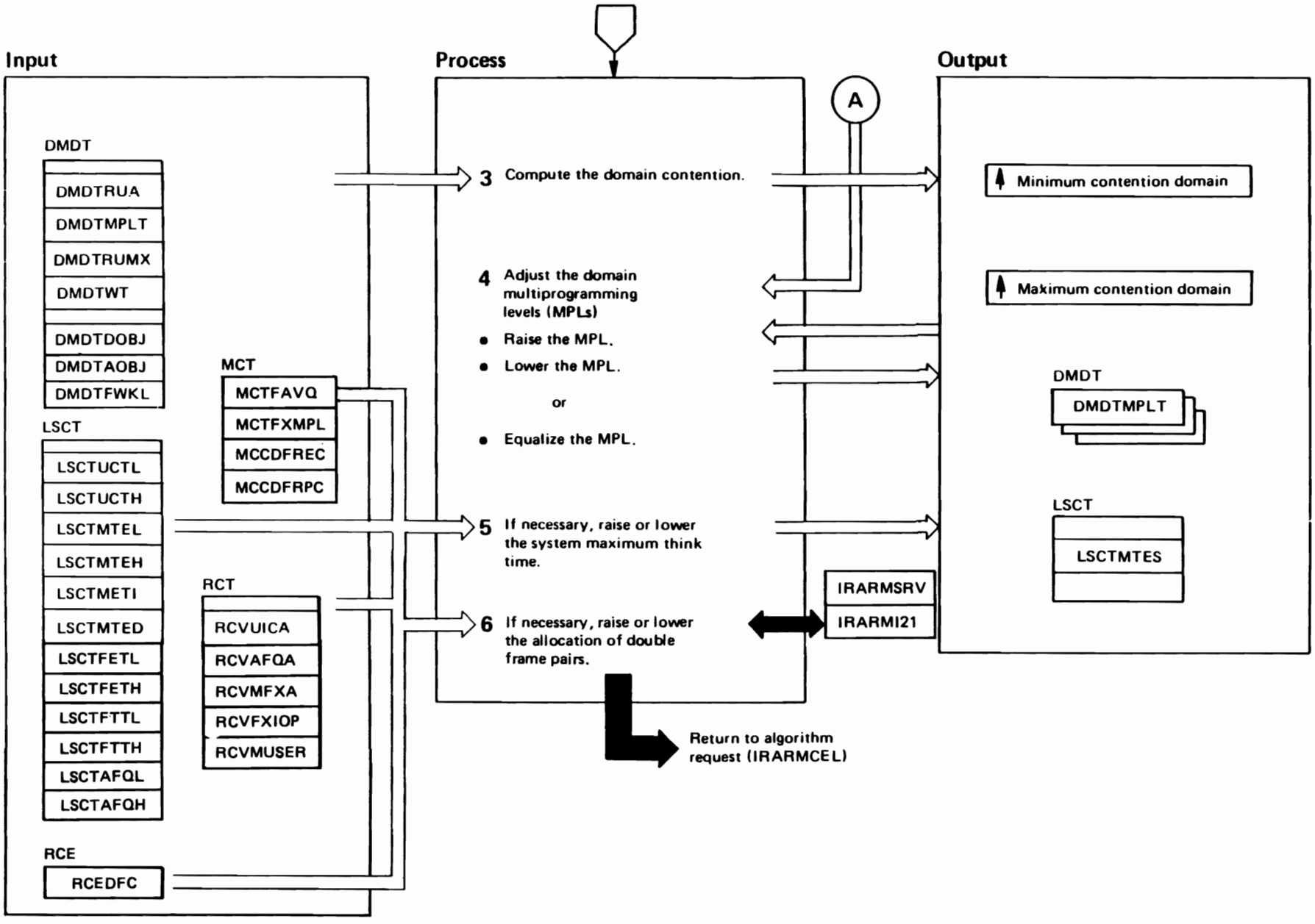


DIAGRAM SRM-27. IRARMRM2 - Resource Monitor MPL Adjustment Processing (4 of 4)

Extended Description	Module	Label

3 Each used domain contention index is computed by the formula:

$$\frac{\text{average ready users x weight}}{\text{max (current target MPL or one)}}$$

This yields a measure of contention for this domain weighted by the user specified importance factor (weight) for the domain

The following values are used if an IPS specifying improved target adjustment is in control.

- For domains with AOBJ – workload level determined by IRARMWM4 from DMDTTWSR. — IRARMWLM — IRARMWM4
- With AOBJ – workload level from IRARMWM4 using DMDTTWSR divided by DMDTRUA.
- With FWKL – use the value for FWKL.

4 The resource monitor will then determine if the system MPL should be raised or lowered by comparing the system contention indicators against predefined limits (limits can be specified in the IEAOPTxx parmlib member). All positive conditions must be met to increase the MPL and only one condition need be met to force a decrease.

DEFAULT LIMITS

	INCREASE MPL	*DECREASE MPL*
UIC (unreferenced interval count)	GT 4	LT 2
Processor utilization	LT 98%	GT 100.9%
Page faults	LT 1000/sec*	GT 1000/sec*
ASMQ	LT 100* requests	GT 100* requests
Demand paging	LT X** or (Processor utilization LT 95% and page process time LT 100 milliseconds)	GT y** and (Processor utilization GT 98% or page process time GT 130 milliseconds)
Fixed frames below 16 megabytes	LT 82%	GT 88%
Frames fixed or allocated to non-swap paging I/O	LT 66%	GT 72%

Extended Description	Module	Label

*These default values effectively eliminate the corresponding conditions from the comparison.

**x and y are processor dependent thresholds based on constants in IRARMCPU. For example, the values for x and y on 168UP are 42 pages/second and 52 pages/second.

- If any domain is unused (the average of the ready user and the highest ready user value less than target minus one) that domain's MPL is decreased by one if the decrease does not drop it below the minimum MPL or one.
- If the system MPL should be raised, the resource monitor selects the domain that has the highest contention index and has not yet reached its maximum MPL and increases this domain's MPL by one.
- If the system MPL should be decreased, the resource monitor selects the domain with the lowest contention index which has not yet reached its minimum MPL and decreases this domain's MPL by one.
- If the system MPL should not be increased or decreased, the resource monitor attempts to equalize the domain's contention index; such that if the highest domain contention index is greater than the lowest, the resource monitor increases the MPL for the high contention domain and decreases the MPL for the lowest contention domain.

5 The system maximum think time is adjusted for logical swap. This function is unrelated to mainline MPL adjustment processing but must be performed at the same interval. — IRARMRMR — IRARMRM2

6 Maintain RSM's double frame queue to ensure an adequate supply of pairs to back the segment tables. — IRARMRMR — IRARMRM2; IRARMSRV — IRARMI21

7 If enough samples have accumulated, raise or lower the extended storage thresholds based on the percent of time that extended storage is completely allocated. — RCEGROUP; RCEAECOK

IRARMWLM MODULE ENTRY POINT SUMMARY

IRARMWM1 Workload manager service calculator routine. Calculates the amount of service provided to a user since the beginning of the current workload manager measurement interval for that user. Service is calculated using the following equation: Service =(MP)/K+(CT)/K+EI+(SB)/K

where:
- M = The MSO service coefficient scaled by 1/50.
- P = The number of page seconds (the product of the number of frames owned by the user and the amount of TCB time consumed) used by the user.
- K = The time required to execute 10,000 instructions. (Dependent on the processor model.)
- C = The processor service coefficient.
- T = The job step time elapsed in the current interval.
- E = The I/O service for this interval.
- I = The I/O service coefficient.
- S = The SRB time elapsed in the current interval.
- B = The SRB service coefficient.

This routine calculates each of the three service factors and the total service for the user for the interval. It also updates the domain service for the user and adds to the corresponding domain the service accumulated since IRARMWM1 was last called for this user. If the total service for the domain is likely to overflow the field, this routine calls IRARMWML to start a new transaction.

IRARMWM2 Swappable user evaluation routine.Scans the IN (in-storage) queue and the OUT (out-of-storage-but-ready) queue, evaluates each swappable user and assigns each his current workload level.

IRARMWM5 Workload manager update performance group period subroutine. Tests whether a user has accumulated enough service/time to be assigned to a new performance group period. If so, IRARMWM5 adjusts the pointers that indicates the performance group period, performance objective, and domain applicable to the transaction current for the user. Note that the frequency (resolution) at which the test period end is made depends on how often IRARMWM5 is called for any given user.

IRARMWM7 WLM recommendation calculation routine. Calculates a workload manager recommendation value for a user based on the service received and on the performance objective currently associated with the user. Users that have not yet received an amount of service equal to their interval service value (ISV) specification while in storage are given a recommendation value boost. The boost gives preferential treatment to users in their ISV as compared to users in their ISV and users between job steps.

IRARMWMI Workload manager in storage interval change subroutine. Updates the transaction accumulators with the service and the time received by the user during the preceding in-storage interval.

IRARMWMJ Routine to determine the scope of applicability of analysis processing to a user. Examines the current swap status and the performance specification for a user. It indicates if the resource manager algorithms are applicable to this user.

IRARMWMK WLM dontswap/okswap user analysis routine. Calculates the current service and ensures that the user is in the correct performance group period. Applicable algorithm indicators are set based on the new swap status of the user.

IRARMWMN Workload manager transaction start routine. The IRARMWMN routine receives control as the result of a SYSEVENT that has been defined by the workload manager to signify that a new transaction should be started for that user. If the user is not in storage, a flag is set to cause the IRARMWMN routine to be reentered during the swap-in of the user. Otherwise, any existing transaction is stopped by calling IRARMWMO, and the user transaction fields are reset to reflect the new transaction being started.

IRARMWMO Workload manager transaction stop routine. The IRARMWMO routine receives control as the result of a SYSEVENT that has been specified by the workload manager as defining the end of any current user transaction. If a new transaction is to be created for the user, IRARMWMO indicates the end of the current transaction. If the next user event is not known, IRARMWMO leaves the transaction accumulated values for later resumption of the transaction. In any case, IRARMWMO causes the preceding time and service to be properly recorded for the current transaction.

IRARMWMP Calculate page seconds routine. Calculates page seconds and page residency time for an address space.

IRARMWMQ Workload manager quiesce completed SYSEVENT processing routine. The IRARMWMQ routine receives control when a user has stopped executing and is being swapped out. The workload manager records the service given that user while he was in storage. The workload manager determines if a user event caused the swap-out, and flags the user to indicate whether such previous service is to be considered when the user is next swapped-in.

IRARMWMR Workload manager restore completed SYSEVENT processing routine. The IRARMWMR routine receives control when a user has been swapped in, and is ready to begin executing. The workload manager sets up the fields used to calculate the service rate the user is to receive during the forthcoming in-storage residency period.

IRARMWLS MODULE ENTRY POINT SUMMARY

IRARMHIT	Workload manager user ready SYSEVENT swap-in scheduling routine. Receives control as the result of a decision to apply swap-in processing to a ready user. It repositions the ready user from the WAIT queue to the OUT queue. For a TSO user controlled by the RTO parameter, IRARMHIT determines whether or not to delay the transaction and how long the delay should be.
IRARMWML	Restart long running transaction routine. Starts a new transaction for an address space whose service totals may overflow the fields where they are stored.
IRARMWM0	Workload manager service calculator routine. Calculates the amount of service used by a transaction or set of transactions. This routine is called by IRARMWM8 (update processing for subsystem transaction reporting). IRARMWM0 performs the same function as IRARMWM1 except it uses a TRQE as input and an RSPL as output.
IRARMWM3	Individual user evaluation routine. Evaluates a swappable user on the IN or OUT queue and assigns a current workload level.
IRARMWM4	Workload manager workload level calculator subroutine. Accepts a service rate, performance objective, and calculates the corresponding workload level.
IRARMWM5	Workload manager update performance group period subroutine. Tests whether a user has accumulated enough service/time to be assigned to a new performance group period. If so, IRARMWM5 adjusts the pointers that indicates the performance group period, performance objective, and domain applicable to the transaction current for the user. Note that the frequency (resolution) at which the test period end is made depends on how often IRARMWM5 is called for any given user.
IRARMWM7	WLM recommendation calculation routine. Calculates a workload manager recommendation value for a user based on the service received and on the performance objective currently associated with the user. Users that have not yet received an amount of service equal to their interval service value (ISV) specification while in storage are given a recommendation value boost. The boost gives preferential treatment to users in their ISV as compared to users in their ISV and users between job steps.
IRARMWM8	Transaction reporting queue storage control and data update algorithm. Obtains and releases storage as necessary for transaction reporting queue elements. At timed intervals, it reports service values for any queue elements on the update queues and returns the elements to the available queues. The elements are put on the update queues by the transaction reporting SYSEVENT (SYSEVENT 35, 36, and 37).
IRARMWM9	(Alias of IRARMW8) Transaction reporting queue storage control and data update algorithm. Performs the same function as IRARMWM8. The IRARMWM9 alias is used when the function is called as a subroutine (for example, by SET command processing).

DIAGRAM SRM-28. IRARMWM2 - Swappable User Evaluation (1 of 4)

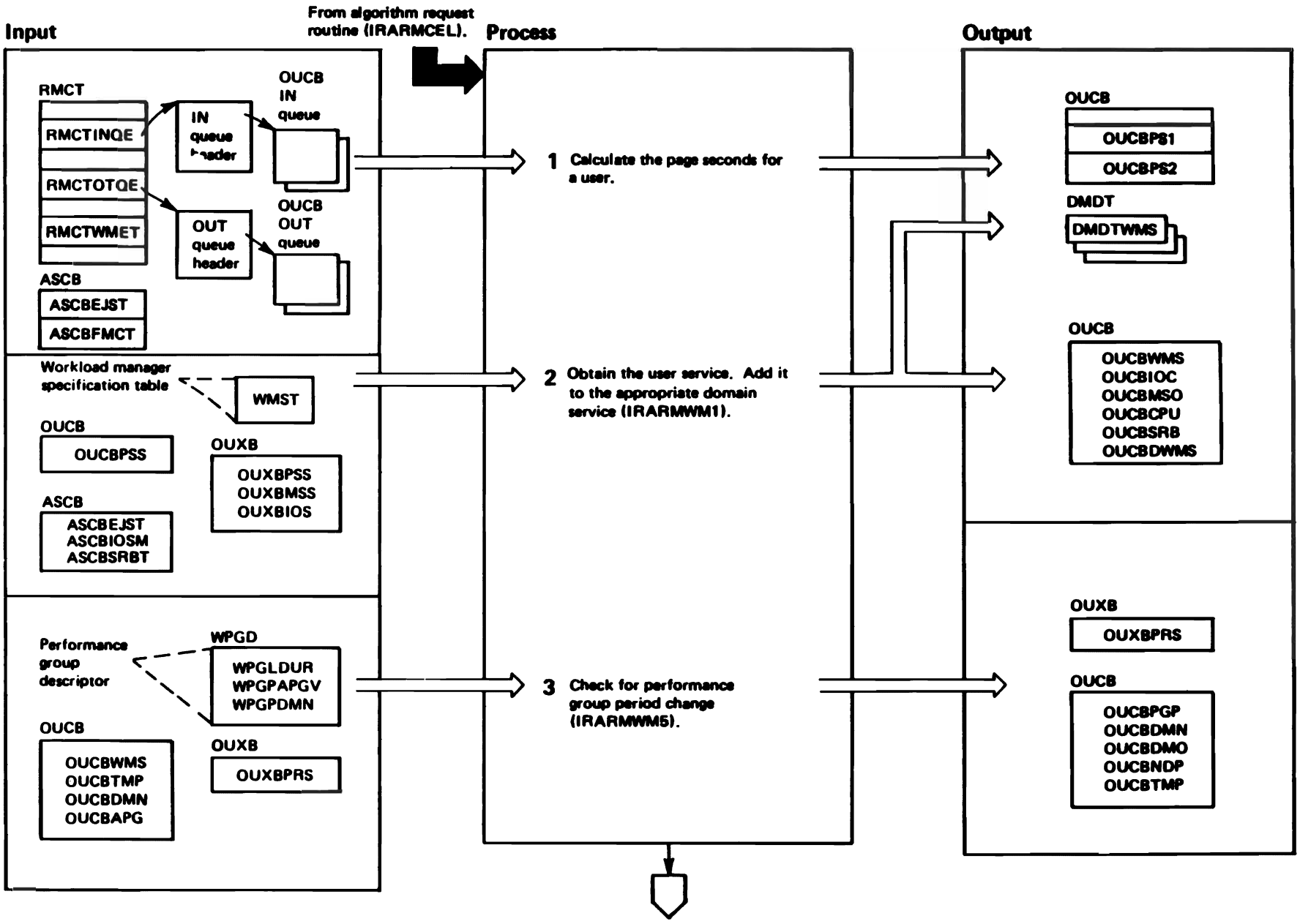


DIAGRAM SRM-28. IRARMWM2 - Swappable User Evaluation (2 of 4)

Extended Description	Module	Label
The IRARMWM2 routine is invoked by the periodic entry point scheduler, via IRARMCEL (the algorithm request routine) approximately every SRM second (that is, one second adjusted according to processor speed) to calculate the page seconds. If a user has gone a minimum of three SRM seconds on one real second without being evaluated, the user is evaluated.		
1 IRARMWM2 scans both the IN and OUT queues and calculates the page seconds (the product of the number of frames owned by the user and the amount of TCB time consumed) for each user.	IRARMWLM	IRARMWM2
2 WM1 is invoked to calculate the service accumulated during the in-real-storage interval for users currently in storage. The service accumulated since the last invocation of IRARMWM1 for this user is added to the corresponding domain's service.	IRARMWLM	IRARMWM1
3 Depending on the unit code in the IPS (service units or time units), the transaction's accumulated service or time units are checked to determine whether a period has ended. If a period has ended, the current period indication is updated. If workload reporting is also active, IRARMWR4 is invoked to communicate the period change. If, in switching period, the user also changes domains, the user is repositioned at the end of the appropriate queue. The user dispatching priority is also updated, if applicable.	IRARMWLM	IRARMWM5
	IRARMWAR	IRARMWR4
	IRARMCTL	IRARMRPS

DIAGRAM SRM-28. IRARMWM2 - Swappable User Evaluation (3 of 4)

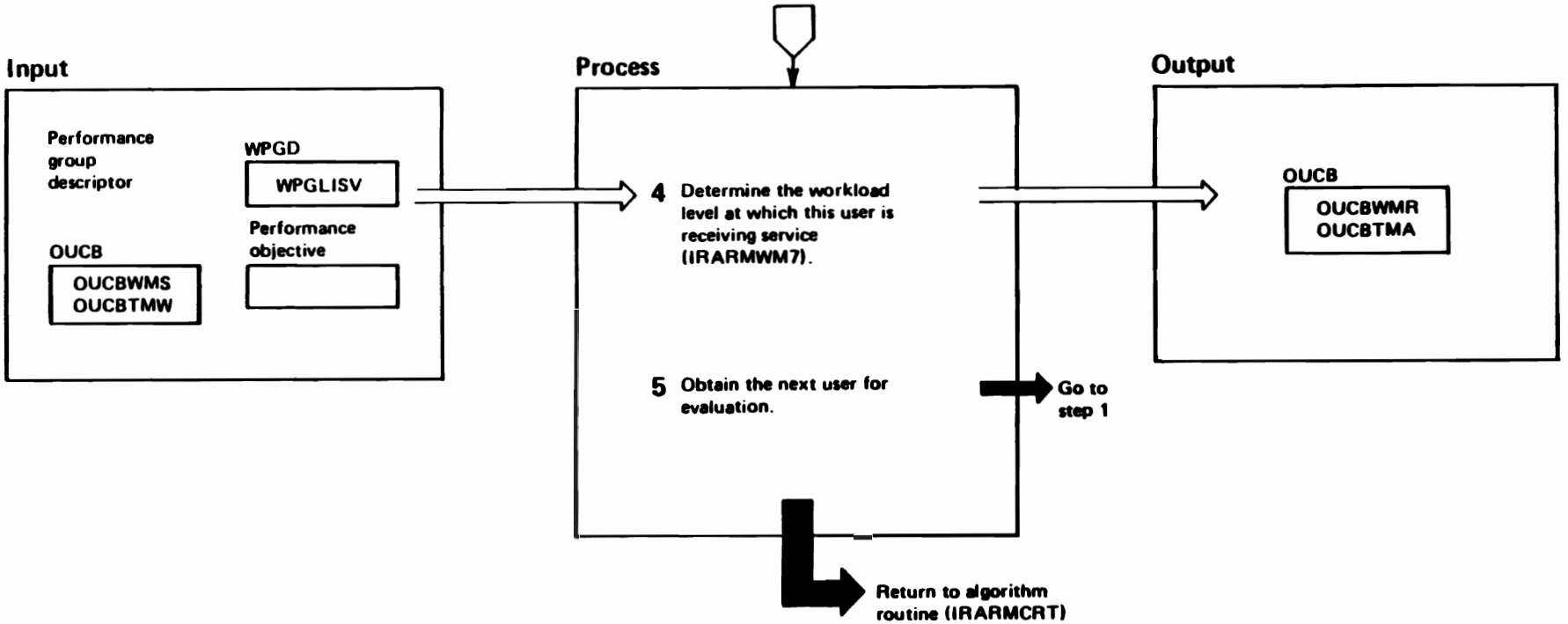


DIAGRAM SRM-28. IRARMWM2 - Swappable User Evaluation (4 of 4)

Extended Description	Module	Label
4 The workload level is a means of comparing users to other users in the same domain. If a user has not yet received enough service to be controlled by the workload manager (that is, his service is less than his interval service value-ISV) or if the user is between job steps, a workload level corresponding to a zero service rate is returned. In calculating his recent service rate, a user's accumulated service is reset to zero when he is swapped in; his accumulated time is reset to zero when he is swapped-out. This accumulated time is saved in the OUCBTMA field of the OUCB.	IRARMWLM	IRARMWM7 IRARMWM4
5 Processing continues until all users on the IN and OUT queues are evaluated.		

DIAGRAM SRM-29. IRARMWM3 - Individual User Evaluation (1 of 2)

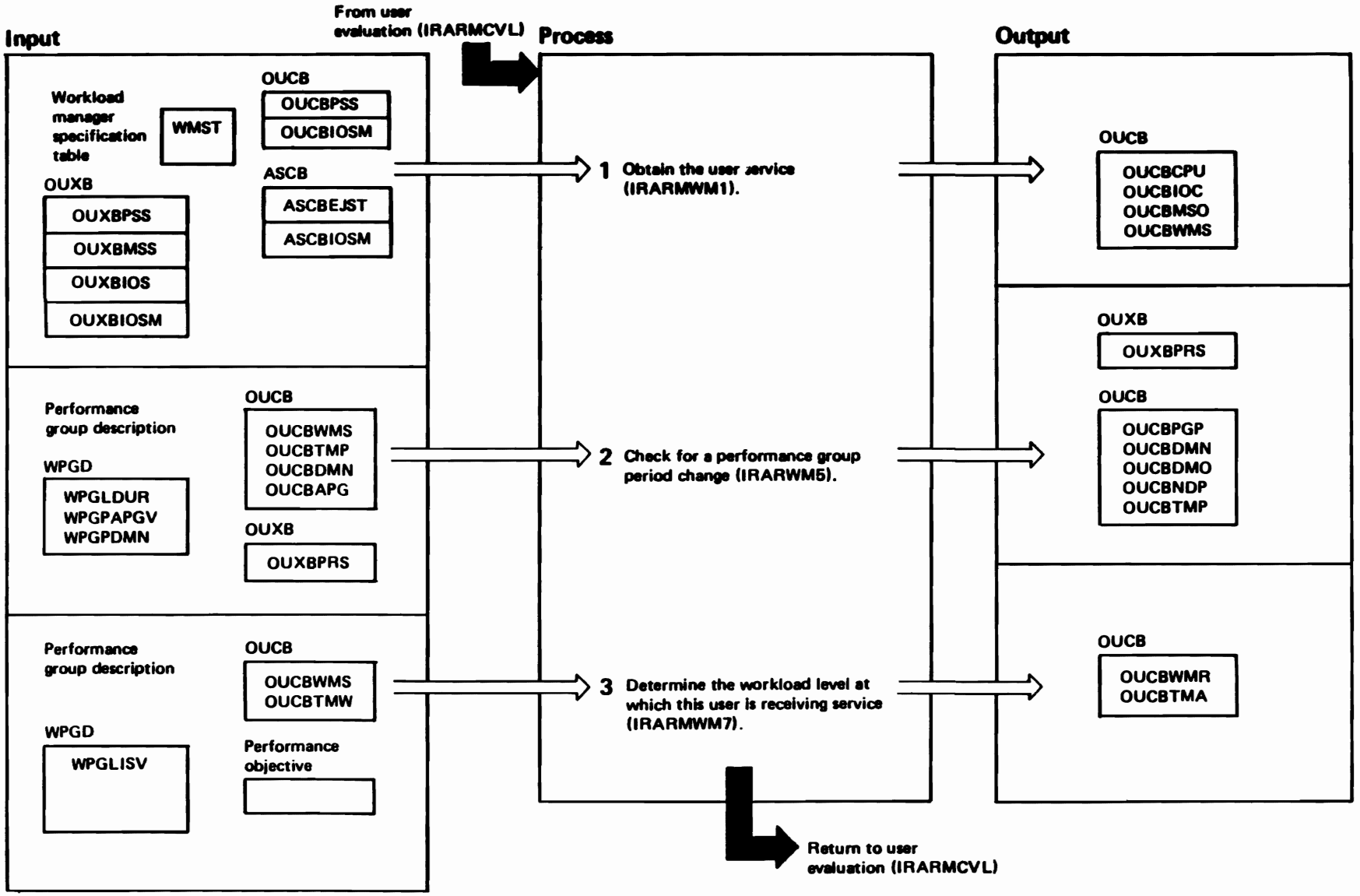


DIAGRAM SRM-29. IRARMWM3 - Individual User Evaluation (2 of 2)

Extended Description	Module	Label
The IRARMWM3 routine is invoked by the user evaluation routine (IRARMCVL) during analysis of users in a particular domain. The major output of the routine is the workload level (recommendation value) of the user being evaluated. Non-swappable and privileged users are not evaluated.		
1 WM1 is invoked to calculate the service accumulated during the in-real-storage interval for users currently in storage.	IRARMWLM	IRARMWM1
2 Depending on the units code in the IPS (service units or time units), the transaction's accumulated service or time units is checked to determine whether a period has ended. If a period has ended, the current period indication is updated. If workload reporting is also active, IRARMWR4 is invoked to communicate the period change. If, in switching periods, the user also changes domains, the user is repositioned at the end of the appropriate queue. The user dispatching priority is also updated, if applicable.	IRARMWLM IRARMWAR IRARMCTL	IRARMWM5 IRARMWR4 IRARMRPS
3 The workload level is a means of comparing users to other users in the same domain. If a user has not yet received enough service to be controlled by the workload manager (that is, the service is less than the interval service value-ISV) or if the user is between job steps, a workload level corresponding to a zero service rate is returned. In calculating a user's recent service rate, a user's accumulated service is reset to zero when swapped-in; the user's accumulated time is reset to zero when swapped-in; the user's accumulated time is reset to zero when the user is swapped-out. This accumulated time is saved in the OUCBTMA field of the OUCB.	IRARMWLM	IRARMWM7 IRARMWM4

DIAGRAM SRM-30. IRARMHIT - User Ready Processing (1 of 2)

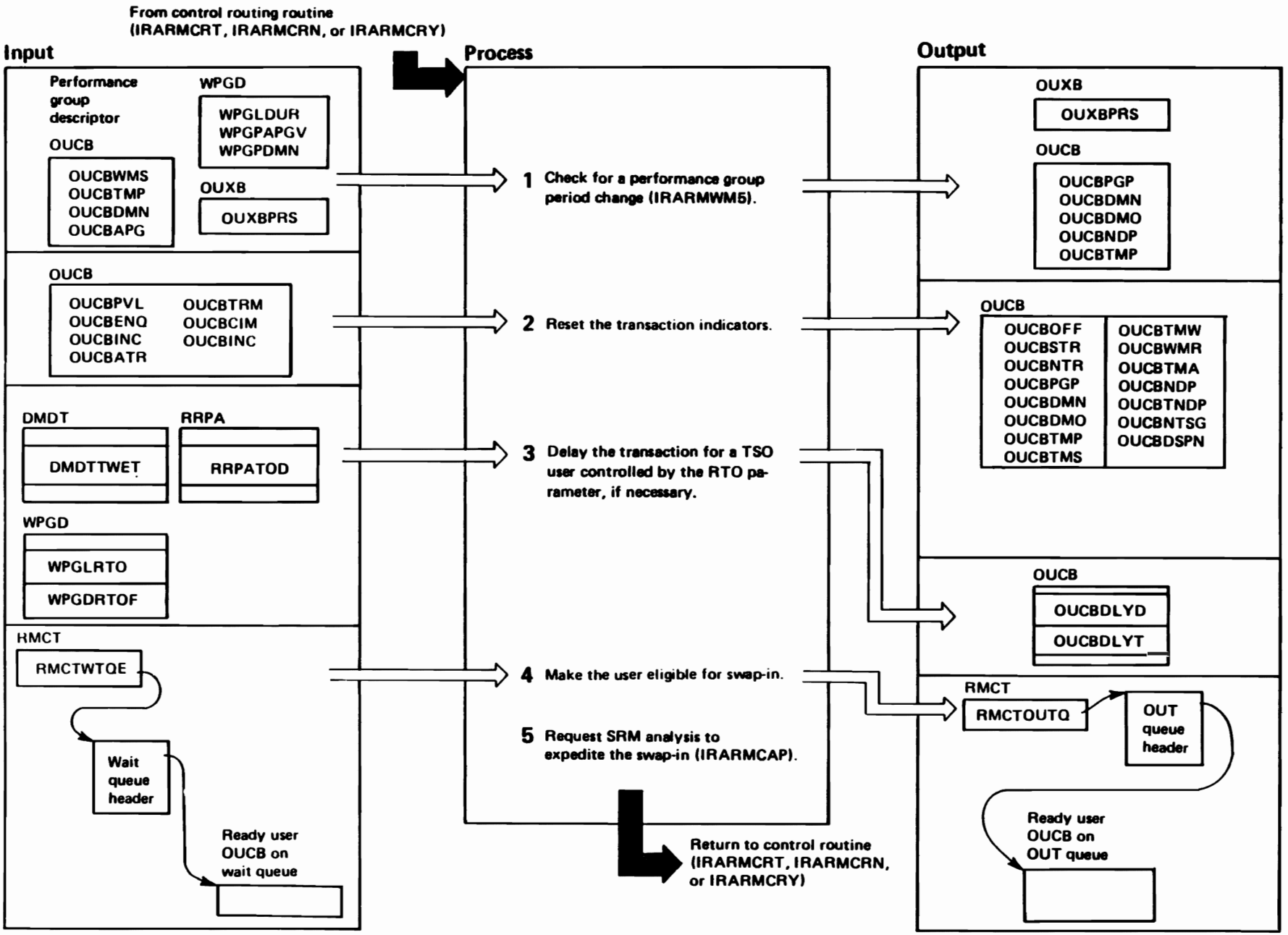


DIAGRAM SRM-30. IRARMHIT - User Ready Processing (2 of 2)

Extended Description	Module	Label
IRARMHIT is requested by IRARMEVT when it receives a user ready SYSEVENT (04) from the dispatcher. The major function of this routine is to make users eligible for swap-in by repositioning them from the WAIT queue to the OUT queue.	IRARMWLS	IRARMHIT
1 Depending on the units code in the IPS (service units or time units), the transaction's accumulated service or time units are checked to determine whether a period has ended. If a period has ended, the current period indication is updated. If workload reporting is also active, IRARMWR4 is invoked to communicate the period change. If, in switching periods, the user also changes domains, the user is repositioned at the end of the appropriate queue. The user dispatching priority is also updated, if applicable.	IRARMWLM	IRARMWM5
	IRARMWAR	IRARMWR4
	IRARMCTL	IRARMRPS
2 The transaction indicators are reset based on the type of user and the user's transaction status when swapped-out. That is:		
a) OUCBs for users between job steps remain effectively unchanged.		
b) OUCBs for terminal wait users are updated to reflect the start of a transaction. Indicators are set to the first period characteristics.		
A workload level is assigned which is equal to the first period objective "zero point".	IRARMWLM	IRARMWM4
c) OUCBs for users that have suspended transactions (may be due to issuing "long wait") are updated so that they look as if the swap-out had just ended.		
3 If the response time objective (RTO) parameter is coded in the IPS for this TSO user and the user becomes ready following a terminal input wait, IRARMHIT delays the swap-in of the user such that the elapsed time of a first period TSO transaction equals the RTO. The amount of time to delay this user is computed based on a time-weighted average maintained in the domain descriptor table (DMDT) for the applicable domain. It is the average of the non-delay elapsed time for TSO transactions ending in the first period in that domain. The statistics for this average are collected when a transaction ends. The average is computed by IRARMRM2 at timed intervals.	IRARMWLM	IRARMHIT
	IRARMRMR	IRARMWMO IRARMRM2
4 The "ready" user OUCB is repositioned from the WAIT queue to the end of the OUT queue.	IRARMCTL	IRARMRPS
5 The SRM analysis function is requested in order to have the user swapped in as soon as possible.	IRARMCTL	IRARMCAP

DIAGRAM SRM-31. IRARMWM8 - Transaction Reporting Algorithm (1 of 4)

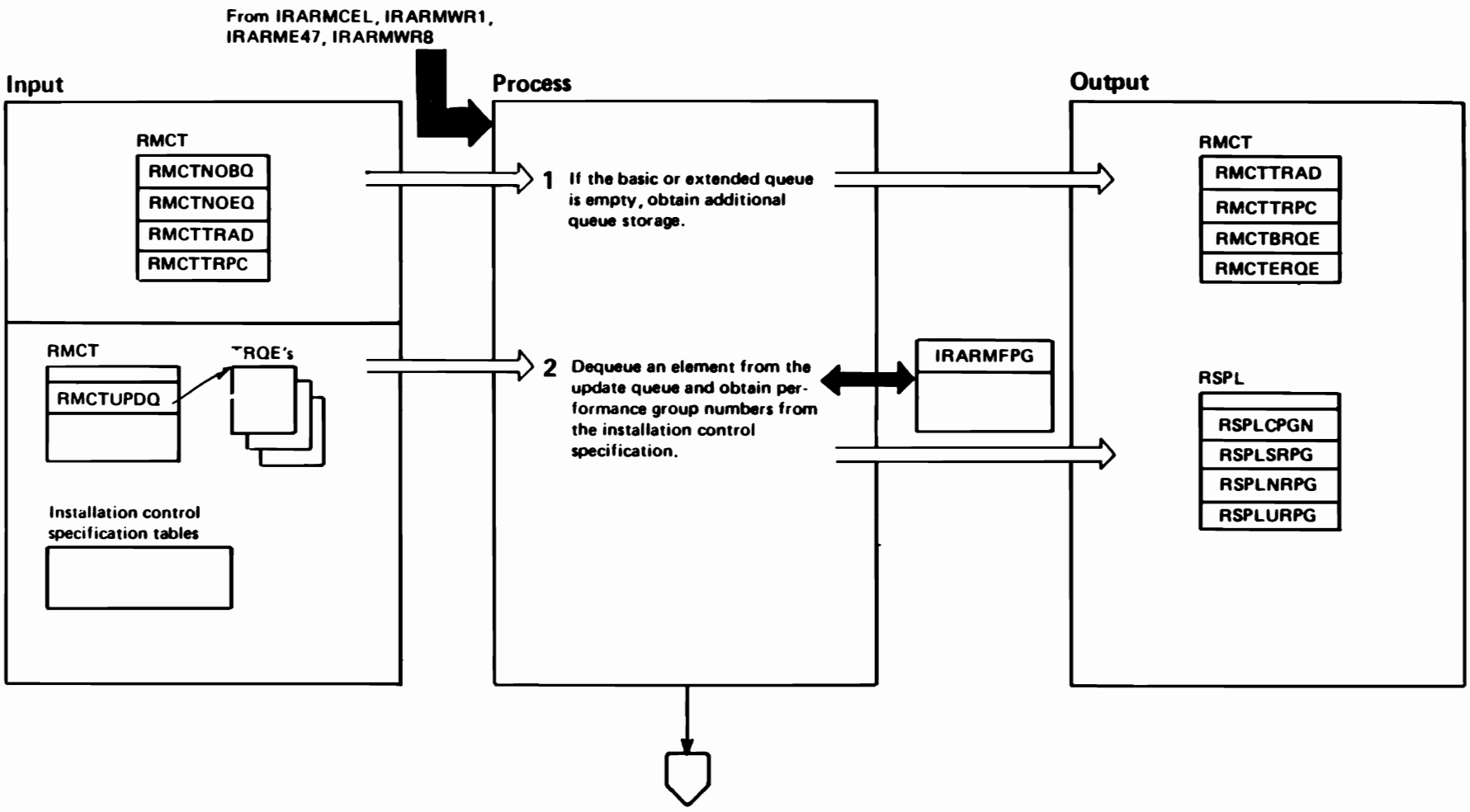


DIAGRAM SRM-31. IRARMWM8 - Transaction Reporting Algorithm (2 of 4)

Extended Description	Module	Label
This routine provides algorithm processing for subsystem transaction reporting. IRARMWM8 is called by IRARMCEL (timed algorithm scheduling routine) at processor-dependent intervals. It is called using the alias IRARMWM9 by IRARMWR1 (workload initialization), IRARMWR3 (workload collection), IRARME47 (workload termination) and IRARMWR8 (set processing). IRARMWM8 processes data accumulated on the transaction update queue. The data is added to the appropriate WAMP entry in the RMF workload activity measurement table (WAMT).	IRARMWLS	IRARMWM8
1 Transaction reporting SYSEVENTs (TRAXRPT and TRAXFRPT) place data in elements from the basic queue. Extended transaction reporting SYSEVENTs (TRAXERPT) place data in elements from the extended queue. Both types are then placed on the update queue. If no queue elements are available for use by a SYSEVENT, one of two bits is set in the RMCT (RMCTNOBQ or RMCTNOEQ) which signals IRARMWM8 to obtain more storage. This storage is partitioned into queue elements and chained to the appropriate queue (basic or extended).		
2 IRARMWM8 serially processes each element on the update queue (steps 2 through 5) until the update queue is empty. IRARMWM8 then calls IRARMFPG to assign one or more report performance groups using the transaction data (subsystem name, transaction name, transaction class and user ID).	IRARMFPG	IRARMFPG

DIAGRAM SRM-31. IRARMWM8 - Transaction Reporting Algorithm (3 of 4)

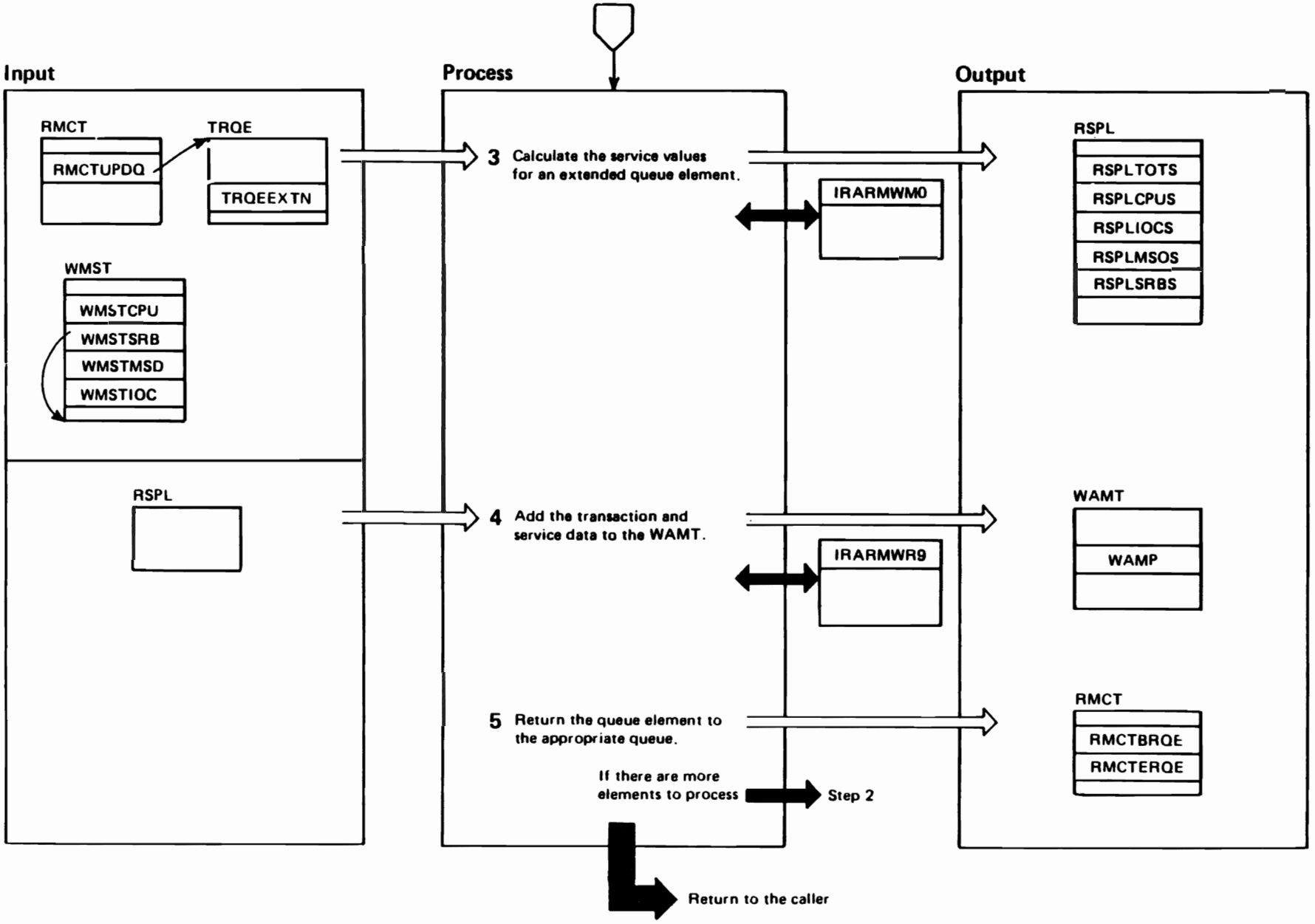


DIAGRAM SRM-31. IRARMWM8 - Transaction Reporting Algorithm (4 of 4)

Extended Description	Module	Label
3 Extended queue elements contain service statistics. IRARMWM8 uses this data to calculate the service rate for the transaction based on the service coefficients in the workload manager specification table (WMST).	IRARMWLS	IRARMWM0
4 IRARMWM8 adds the elapsed time for the transaction, the elapsed time squared, and the service data to the WAMP entry of the WAMT that corresponds to the report performance group obtained in step 2.	IRARMWAR	IRARMWR9
5 This routine returns the element to the appropriate queue and continues processing at step 2 if there are more elements to process, otherwise control is returned to the caller.		

IRARMWAR MODULE ENTRY POINT SUMMARY

IRARMWR1 Workload activity recording initialization subroutine. Constructs and initializes the workload activity measurement table (WAMT) in the buffer (storage from SQA) obtained by RMF.

IRARMWR2 Workload activity recording WAMT initialization subroutine. Builds the WAMT in a format suitable for updating by the SRM.

IRARMWR3 SRM workload activity recording data collection subroutine. Moves the contents of the WAMT into a collection buffer capable of containing the data. Note: The buffer is obtained by RMF from LSQA, storage key 0, and must be fixed in storage.

If the IPS has not been changed, then adds to the collected data the transaction data for the current in-storage interval for each in-storage address space with an active transaction, re-initializes the data collection buffer for the next collection interval, and calculates the workload level for each performance group period that contains transaction data.

IRARMWR4 SRM workload activity recording transaction data update subroutine. Adds the service and transaction active time to the appropriate WAMT performance group period accumulator in the data collection interval.

IRARMWR5 SRM workload activity recording workload level calculation subroutine. Calculates the workload level for each WAMT performance group period entry in which transaction data has been accumulated during the last data collection interval.

Note: For those WAMT entries in which the calculated service rate can be associated with multiple workload levels or is zero even though at least one transaction has been active during the data collection interval, the negative value of the workload level will be calculated to indicated an estimated value to RMF.

IRARMWR6 SRM workload activity recording transaction end update subroutine. Adds to the appropriate WAMT performance group period accumulator the transaction elapsed time and counts the number of transactions that terminated during the current data collection interval.

IRARMWR7 SRM workload activity recording WAMT entry determination subroutine. Obtains addressability to the WAMT performance group period entry in which to accumulate user transaction information.

IRARMWR8 SRM workload activity recording. Terminates workload activity data collection whenever an IPS change occurs.

IRARMWR9 Symbolic workload recording processor. Called by IRARMWM8 each time an extended TRQE is processed. It adds service data to the WAMT for RMF.

IRARMWRA Symbolic performance recording. Constructs the ICSM (in a format that workload activity recording can reference) for symbolic performance recording.

DIAGRAM SRM-32. IRARMWR1 - Initialize for RMF (1 of 2)

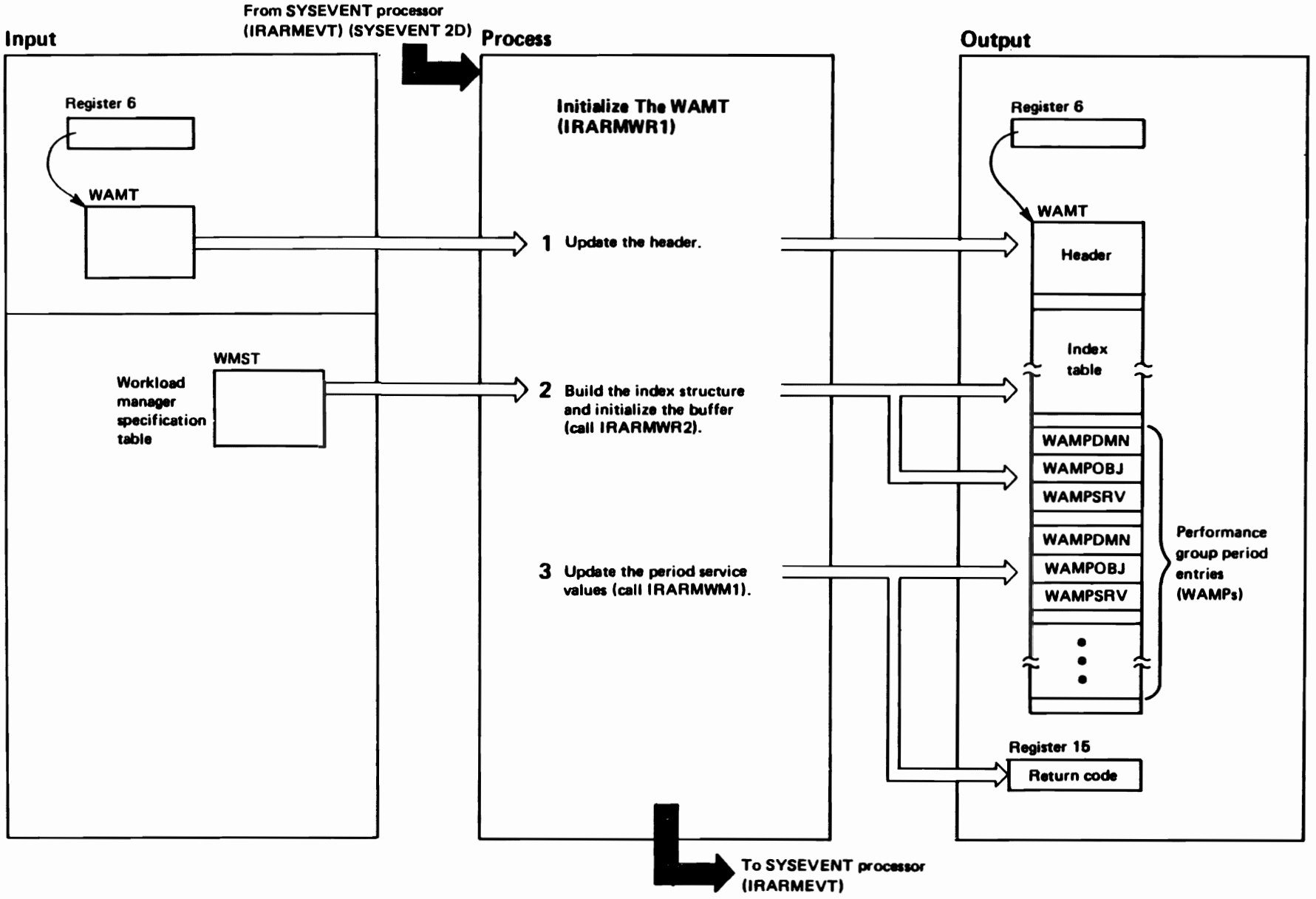


DIAGRAM SRM-32. IRARMWR1 - Initialize for RMF (2 of 2)

Extended Description	Module	Label
IRARMWR1 constructs and initializes the workload activity measurement table (WAMT) in the buffer (storage from SQA) obtained by RMF and input with Sysevent 2D.	IRARMWAR	IRARMWR1
2 The index is used to locate the period entries in WAMT that correspond to a particular performance group. The period entries are updated with their respective domain and performance objective numbers. All other period entry values are zeroed.	IRARMWAR	IRARMWR2
3 Service values in the period entries are initialized such that service already received by active user transactions is not included in the RMF interval service totals.	IRARMWLM	IRARMWM1

Return Codes in Register 15 byte 3

X'00' – Data area accepted and initialized.

X'08' – Length of data area incorrect.

DIAGRAM SRM-33. IRARMRW3 - Collect Data for RMF (1 of 2)

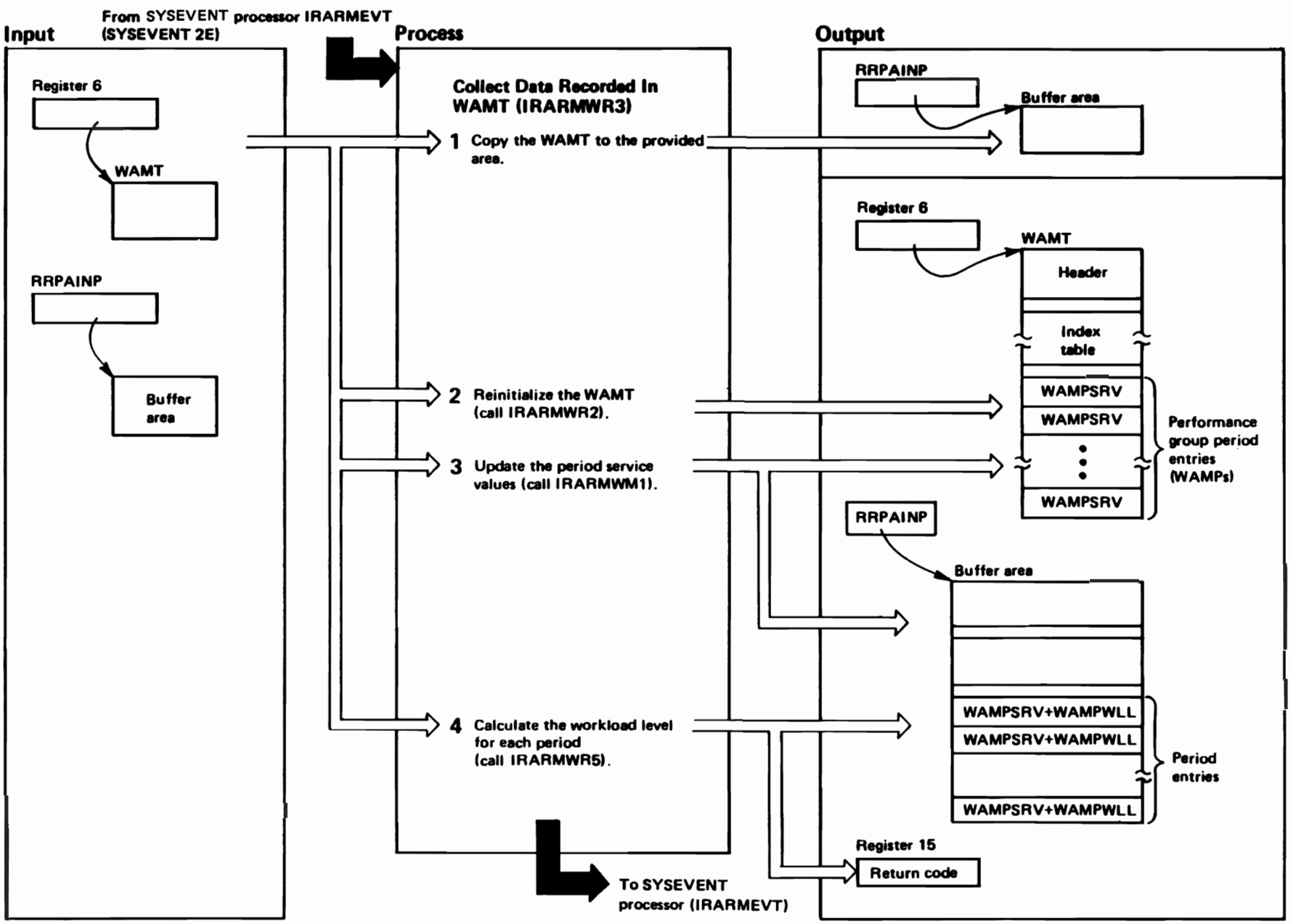


DIAGRAM SRM-33. IRARMRW3 - Collect Data for RMF (2 of 2)

Extended Description	Module	Label
IRARMWR3 copies the contents of the WAMT into a collection buffer. The buffer is obtained by the measurement facility from LSQA and is fixed in storage.	IRARMWAR	IRARMWR3
1 The WAMT is copied into the buffer.		
2 If a set to a new IPS occurs, workload collection is terminated and the WAMT is updated to reflect the statistics at that point in time. If the IPS has not been changed, the WAMT is updated for a new collection interval.	IRARMWAR	IRARMWR2
3 Both the WAMT and the collection buffer are updated to reflect the actual service (SRV) received within each respective interval.	IRARMWLM	IRARMWM1
4 The workload levels (WLL) are updated in the collection buffer for the measurement facility.	IRARMWAR	IRARMWR5

Return codes in Register 15 byte 3

X'00' – Successful data collection

X'04' – Successful data collection, but an IPS change occurred terminating workload collection

X'40' – Data collection is not active, or the data buffer is non-existent, or the copy of the buffer is an incorrect size.

IRARMCBS MODULE ENTRY POINT SUMMARY

IRARMCBS	SRM status for the address space represented

DIAGRAM SRM-34. IRARMCBS - SRM Control Block Status Exit (1 of 5)

IRARMCBS - MODULE DESCRIPTION

DESCRIPTIVE NAME: SRM Control Block Status Exit.

FUNCTION:
Analyzes an OUCB (SRM User control block) to give
SRM status for the address space that it represents

ENTRY POINT: IRARMCBS

PURPOSE: See FUNCTION

LINKAGE: None

CALLERS: BLSACBSS (Through exit services router)

INPUT:
Two parameters are passed:

ZZ2 - The IPCS task variable. Contains parameters that are passed on to the IPCS routines.

CBSP - The Control Block Status Parameter List. This contains information about the ASCB which has been accessed that contains the address of the OUCB that's to be tested here.

OUTPUT: One or more messages, see MESSAGES, below.

EXIT NORMAL:

EXTERNAL REFERENCES:

ROUTINES:
BLSRACCQ -- IPCS dump access service. Used to obtain a copy of a control block in the dump. This version specifies that the dump routine should print messages if it fails to access the control block. Called indirectly through ADPLESRV.

BLSRACCN -- IPCS dump access service. Used to access a single field within a control block instead of the whole thing.

BLSUPPR2 -- IPCS expanded print service. Used to print the messages. Called indirectly through ADPLESRV.

ADPLESRV -- IPCS exit services router.

IRARMCBV -- OUCB control block validity check routine. Called indirectly through the ?BLSRESSA service.

CONTROL BLOCKS:
ASCB -- Address Space Control Block
ESR -- Equate Symbol Record
CBSP -- Control Block Status Parameter List
OUCB -- SRM User Request Block
RMQH -- SRM Queue Header Block

DIAGRAM SRM-34. IRARMCBS - SRM Control Block Status Exit (2 of 5)

IRARMCBS - MODULE OPERATION

The IPCS OUCB validity check routine is called to determine whether the OUCB pointed to by the ASCB that was passed as input in a buffer is valid. The OUCB contains SRM information pertaining to an address space to be analyzed.

If the OUCB is valid, it is obtained through a call to the IPCS dump access service. If, however, the validity check routine returned a code indicating that the OUCB was damaged seriously enough to prevent further processing, message IRA10101I is generated.

The OUCB is on one of four circular double-threaded, single-headed queues, one for each possible state of an address space: IN, OUT, WAIT, or LOGICAL SWAP WAIT. For each of these queues, a pointer in the RMCT points to a Resources Manager Queue Header (RMQH). This is a block that is chained into the queue with the OUCB's and contains a field -- RMQHNAME -- which identifies the queue.

Message IRA10102I states which queue the OUCB is on, if it can be determined. In order to determine which queue the OUCB is on, a chain of OUCB's is followed to its anchor block, the RMQH.

If this analysis routine encounters some difficulty in following the forward pointers (OUCBFWD) through the OUCB chain (in the form of a circular queue or an OUCB that cannot be accessed), the backward pointers are followed.

If the backward search fails (for similar reasons), then if the OUCB was not in a transitioning state (OUCBGOB = 0, OUCBGOI = 0, and OUCBGOO = 0) message IRA10112I is generated to give the OUCB's status according to queue indicator bits in the block (OUCBOUT, OUCBOFF, and OUCBLSW).

If, when the search fails, OUCBGOB, OUCBGOO, or OUCBGOI indicate that the address space was transitioning (meaning that bits OUCBOFF, OUCBOUT, and OUCBLSW give the destination queue, not the present one), message IRA10114I is generated to say that the queue identity cannot be determined. This message appears before any messages describing how this routine failed to follow the queue or read the RMQHNAME field.

If the OUCB queue was determined successfully, and the address space was not transitioning, and the bits in the OUCB indicate a queue that's different from the one the OUCB was found on, message IRA10113I is generated.

If the address space represented by the OUCB is transitioning to another queue, IRA10103I states which it's moving to.

If the address space is swapped out or in the process of being swapped out, message IRA10104I is generated to give the reason for swap-out.

If the address space is in the process of being

DIAGRAM SRM-34. IRARMCBS - SRM Control Block Status Exit (3 of 5)

IRARMCBS - MODULE OPERATION (Continued)

swapped out, message IRA10105I gives the last process started.

If the address space has issued a DONTSWAP SYSEVENT followed by a TRANSWAP SYSEVENT, message IRA10106I states that it is waiting completion of a DONTSWAP/OKSWAP sequence.

The highest code returned from an IPCS access routine called here is returned from this module.

RECOVERY OPERATION: Covered by the IPCS ESTAE

DIAGRAM SRM-34. IRARMCBS - SRM Control Block Status Exit (4 of 5)

IRARMCBS - DIAGNOSTIC AIDS

ENTRY POINT NAME: IRARMCBS

MESSAGES:

The following messages may be generated (via the BLSUPPR2 exit service):

IRA10101I Unable to produce SRM analysis for this address space.

IRA10102I This address space is on the SRM xxx queue
where xxx is IN, OUT, LOGICAL SWAP WAIT or WAIT

IRA10103I It is moving to the SRM xxx queue
where xxx is IN, OUT, LOGICAL SWAP WAIT or WAIT

IRA10104I The reason for swap-out is yyy
where yyy is text associated with this swap out reason code:
input terminal wait (1).
output terminal wait (2).
long wait (3).
auxiliary storage shortage (4).
real pageable storage shortage (5).
detected wait (6).
requested swap (7).
enqueue exchange (8).
exchange on recommendation value (9).
unilateral swap (10).
transition to nonswappable (11).

IRA10105I In the swap-out process, zzz
where zzz is the current state of the swap-out process:
RCT has been posted.
QUIESCE has started.
QUIESCE has completed.
QUIESCE has failed.

IRA10106I A TRANSWAP SYSEVENT is waiting for completion of a DONTSWAP/OKSWAP sequence.

IRA10107I The OUCB for this address space is on a queue whose anchor block has a queue name that does not contain a valid value.

IRA10108I The OUCB for this address space is on an SRM queue with a forward pointer that is not valid.

IRA10109I The OUCB for this address space is on an SRM queue with a circular forward loop.

IRA10110I The OUCB for this address space is on an SRM queue with a backward pointer that is not valid.

IRA10111I The OUCB for this address space is on an SRM queue with a circular backward loop.

IRA10112I According to the fields OUCBOUT, OUCBOFF, and OUCBLSW, this address space is on the SRM xxx queue.

IRA10113I The fields in the OUCB: OUCBOUT, OUCBOFF, and OUCBLSW indicate that this address space is on the SRM xxx queue; however, the address space is on the SRM yyy queue.

IRA10114I Unable to determine the identity of the SRM queue for this address space.

DIAGRAM SRM-34. IRARMCBS - SRM Control Block Status Exit (5 of 5)

IRARMCBS - DIAGNOSTIC AIDS (Continued)

ABEND CODES: None

WAIT STATE CODES: None

RETURN CODES:

EXIT NORMAL:

Register 15 contains zero on return if the the routine completed successfully and has generated status.

EXIT ERROR:

Register 15 contains the highest return code set by a failing service.

REGISTER CONTENTS ON ENTRY:

Standard Entry

REGISTER CONTENTS ON EXIT:

EXIT NORMAL:

Standard Exit

EXIT ERROR:

Standard Exit

INDEX

K

L

M

O

P

R

S

MVS/Extended Architecture System Logic Library: System Resource Manager

LY28-1760-0

S370-36

Printed in U.S.A.

MVS/Extended Architecture
System Logic Library:
System Resource Manager

READER'S
COMMENT
FORM

LY28-1760-0

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note: ***Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.***

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

_____

_____

_____

_____

What is your occupation? _____

How do you use this publication? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

MVS/Extended Architecture System Logic Library: System Resource Manager

LY28-1760-0 S370-36

Reader's Comment Form

Cut or Fold Along Line

Fold and tape Please Do Not Staple Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department D58, Building 921-2
PO Box 390
Poughkeepsie, New York 12602

Fold and tape Please Do Not Staple Fold and tape

Printed in U.S.A.

IBM®

LY28-1760-00