

A11103 089736

NAT'L INST OF STANDARDS & TECH R.I.C.



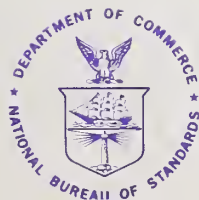
A11103089736

Kelly, John C/UNIVAC 1108 EXEC level 32R  
QC100 .U57 NO.500-, 34, 1978 C.2 NBS-PUB

E & TECHNOLOGY:



# UNIVAC 1108 EXEC LEVEL 32R2 PERFORMANCE HANDBOOK



NBS Special Publication 500-34

U.S. DEPARTMENT OF COMMERCE  
National Bureau of Standards

## NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards<sup>1</sup> was established by an act of Congress March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, and the Institute for Computer Sciences and Technology.

**THE NATIONAL MEASUREMENT LABORATORY** provides the national system of physical and chemical and materials measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; conducts materials research leading to improved methods of measurement, standards, and data on the properties of materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government Agencies; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

Absolute Physical Quantities<sup>2</sup> — Radiation Research — Thermodynamics and Molecular Science — Analytical Chemistry — Materials Science.

**THE NATIONAL ENGINEERING LABORATORY** provides technology and technical services to users in the public and private sectors to address national needs and to solve national problems in the public interest; conducts research in engineering and applied science in support of objectives in these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

Applied Mathematics — Electronics and Electrical Engineering<sup>2</sup> — Mechanical Engineering and Process Technology<sup>2</sup> — Building Technology — Fire Research — Consumer Product Technology — Field Methods.

**THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY** conducts research and provides scientific and technical services to aid Federal Agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal Agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following divisions:

Systems and Software — Computer Systems Engineering — Information Technology.

<sup>1</sup>Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.

<sup>2</sup>Some divisions within the center are located at Boulder, Colorado, 80303.

JUL 13 1978

not acc

00

100

457

no 500.34

1978

C.2

# COMPUTER SCIENCE & TECHNOLOGY:

## UNIVAC 1108 EXEC LEVEL 32R2

### Performance Handbook

---

Capt. John C. Kelly and Lt. Gary P. Route

*Special publication*

Federal Computer Performance Evaluation and  
Simulation Center  
Washington, DC 20330

Sponsored by

U.S. Military Personnel Center  
Alexandria, VA 22332



---

U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, Secretary

Dr. Sidney Harman, Under Secretary

Jordan J. Baruch, Assistant Secretary for Science and Technology

U.S. NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

Issued June 1978

## **Reports on Computer Science and Technology**

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

**National Bureau of Standards Special Publication 500-34**

Nat. Bur. Stand. (U.S.) Spec. Publ. 500-34, 144 pages (June 1978)

CODEN: XNBSAV

**Library of Congress Catalog Card Number: 78-600054**

U.S. GOVERNMENT PRINTING OFFICE

WASHINGTON: 1978



## FOREWORD

Operating systems largely determine the performance of modern computer systems. They control large-scale, multi-programming computers with programmed rules and procedures that may be tailored, within limits, to fit the particular needs of different users in widely differing applications. Proper tuning of operating system control variables is where any serious attempt to achieve efficient computer utilization must begin, and often where its greatest benefits accrue.

This report describes the user-adjustable controls of one such operating system: UNIVAC EXEC 8, level 32 release 2. It presents methods of detecting and diagnosing performance problems, identifies important EXEC control points, and recommends control settings that are believed to produce generally satisfactory results at some installations.

Such information is, of course, site and system specific. That is, it has direct application only to EXEC 8 sites, limited application outside of level 32, and qualified application even to other sites with the identical release. For non-EXEC sites, the document should prove of interest primarily as a format for presenting a kind of information that would be useful in any operating system environment and as a model for analyzing the subtle problem of operating system tuning on any vendor system. For other releases of EXEC 8, resemblances to level 32 are probably sufficient to stimulate creative thinking about specific areas of performance improvement, although the caution against careless or uncritical application of recommendations is especially urgent here. For sites using level 32, the information here should be of enormous value and much of it directly transferable. This class of reader must remember only that operating systems are adjustable precisely because no two installations have quite the same configuration, workload or functional priorities, and that no set of control parameters will suit all implementations even of the same system equally well.

This report was prepared in its original form by the Federal Computer Performance Evaluation and Simulation Center (FEDSIM) under the sponsorship of the U.S. Army Military Personnel Center (MILPERCEN). It is reprinted here with minor editorial changes by the National Bureau of Standards' Institute for Computer Sciences and Technology (NBS/ICST) in the belief that it will have useful application at other installations besides the one for which it was intended.

Publication of this report in no way implies recommendation or endorsement of UNIVAC EXEC 8 systems by the National Bureau of Standards, nor is it to be construed to pass qualitative judgment in any degree on the performance characteristics of the system or any system features described herein.

Special note is made of the fact that this publication is the result of a cooperative interaction between NBS and FEDSIM to make the results of FEDSIM's generally applicable work available to the Federal ADP community. NBS also wishes especially to express its appreciation to MILPERCEN for making the occasion for this collaboration possible.

The authors of this report welcome questions concerning the technical content, which should be addressed to them at: FEDSIM, Washington, DC 20330. NBS/ICST would appreciate very much comments concerning the general utility and need for computer system tuning guides. Of particular interest would be proposed or candidate reports, facilities interested in sponsoring or participating in developing tuning guides, and facilities interested in serving as test beds for evaluating candidate guidelines. Queries and expressions of interests of this nature should be addressed to either John F. Wood or Richard F. Dunlavey, National Bureau of Standards, Building 225, Room A265, Washington, DC 20234 (Area Code 301, 921-3485).

M. Zane Thornton, Acting Director  
Institute for Computer Sciences  
and Technology  
National Bureau of Standards  
U. S. Department of Commerce

## PREFACE

This report is based on a detailed analysis of the UNIVAC 1108 Operating System - EXEC Level 32R2 - as operated by the United States Army Military Personnel Center (MILPERCEN). Since the procedures and guidelines specified in the report strongly depend on the particular equipment type and EXEC level, care should be taken before applying the results to different environments. Questions relating to the subject of this report or to the possibility of extending the results should be addressed to the study's authors at the Federal Computer Performance Evaluation and Simulation Center (FEDSIM).

## ABSTRACT

This is a report prepared for the Army Military Personnel Center (MILPERCEN). It describes a set of hypotheses for evaluating the performance of UNIVAC 1100 Computer Systems. The hypotheses specifically apply to EXEC Level 32R2 operating on UNIVAC 1108 Computer Systems. Attempts to apply the guidelines to different UNIVAC 1100 models or to different levels of the EXEC may lead to erroneous results.

The report contains sections on each major complex within the EXEC. Each section (1) states the performance hypotheses associated with that complex, (2) explains how the available measurement tools may be used to determine if the hypotheses are true, and (3) identifies which performance parameters need to be changed to correct the problem situation. All of the performance hypotheses discussed in the body of the report are collected together in Appendix A. Appendix B summarizes the performance parameters and Appendix C contains an introduction to the Software Instrumentation Package (SIP).

Key words: Computer performance management; measurement tools; operating system performance; performance guidelines; performance measurement; tuning guides; UNIVAC 1108.



## TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
I. INTRODUCTION.....	1
A. Background.....	1
B. Objective.....	1
II. METHODOLOGY.....	3
III. PROCESSOR MANAGEMENT.....	5
A. Introduction.....	5
B. CPU Priorities.....	7
C. CPU Quanta.....	8
D. Test and Set Interrupts.....	9
IV. MEMORY MANAGEMENT.....	11
A. Memory Configuration.....	11
B. Memory Allocation.....	11
C. Demand/Batch Sharing.....	21
D. Function/Segment Activity.....	23
E. EXPOOL.....	25
V. FACILITIES MANAGEMENT.....	29
A. Master File Directory.....	29
B. Mass Storage Rollouts.....	31
C. Granule Tables.....	37
D. Mass Storage Management Practices.....	40
E. Scheduling.....	43
VI. INPUT/OUTPUT.....	47
A. Late Acknowledgments.....	47
B. Seek Time.....	52
C. Block Size.....	55
VII. TEST AND DEBUG.....	59
APPENDICES	
A. PERFORMANCE HYPOTHESES.....	61
B. PERFORMANCE PARAMETERS.....	67
C. INTRODUCTION TO SIP.....	107

## TABLE OF CONTENTS

### TABLES

<u>Table</u>		<u>Page</u>
III-1	Activity Priority Types.....	7
III-2	CPU Quanta.....	8
IV-1	Core Priorities.....	14
IV-2	Demand Priority Levels for CQFACT=800.....	15
IV-3	Demand Priority Levels for CQFACT=3200.....	16
IV-4	Batch Priority Levels.....	17
IV-5	Cost Table.....	21
IV-6	Segment Status Values.....	25
V-1	Mass Storage Thresholds.....	31
V-2	Point Values for TSLR and MTBR.....	36
V-3	Point Values for SIZE.....	36
VII-1	Test and Debug Parameters.....	59
B-1	Performance Parameters.....	70
C-1	SIP Reports (Level 4.0).....	110

### FIGURES

#### Figure

III-1	Nesting of EXEC Functions.....	5
III-2	Major System Queues.....	6
IV-1	Memory Configuration.....	12
IV-2	Demand/Batch Sharing Algorithm.....	22

## I. INTRODUCTION

### A. BACKGROUND

The United States Army Military Personnel Center (MILPERCEN) is responsible for maintaining the personnel records of all enlisted and officer personnel in the Army. In support of this mission, MILPERCEN operates seven UNIVAC 1108 computer systems. During FEDSIM's study, the computer systems were operating under control of the UNIVAC 1100 Operating System, Level 32R2.

Because MILPERCEN recognized the importance of operating system performance on overall system performance, they requested that FEDSIM study how operating system performance could be improved. Their request for assistance was formalized in FEDSIM/MILPERCEN Customer Agreement AY-505-080-ARMY, titled MILPERCEN EXEC-8 ANALYSIS. Additional tasks in support of this project were outlined in Amendment No. 1 to Customer Agreement AY-505-080-ARMY. The funding for the additional tasks was obtained by transferring funds from another project, NA-505-079-ARMY.

### B. OBJECTIVE

The objective of this project was to optimize the performance of the UNIVAC 1100 Operating System for MILPERCEN's unique environment. To accomplish this objective, the project was divided into two parts: (1) the development of a set of general performance guidelines and (2) the application of those guidelines to MILPERCEN. This report summarizes the guidelines developed. A second report describes their application to MILPERCEN.

In support of the project objective, the customer agreement outlined 14 areas of investigation:

1. Library Configuration
2. Swapping
3. EXEC-8 Configuration Parameters
4. Queue Management/Interrupt Processing
5. Internal Priority Management
6. Real-Time Interface Tailoring
7. Dynamic Allocation of Core Memory
8. Management of Buffer Space
9. Dispatching Time Slice Management
10. Data File Rollout/Rollback Management
11. Communications Terminal Network
12. Officer Personnel Utilization System
13. I/O Complex
14. Remote Terminal Emulator

After preliminary analysis, it was found that many of these areas overlapped with one another. To eliminate the redundancies, the following consolidated areas were developed:

- (1) Officer Personnel Utilization System
- (2) Processor Management
- (3) Memory Management
- (4) Facilities Management
- (5) Input/Output
- (6) Test and Debug

The Officer Personnel Utilization System is not discussed in this report. It is treated separately in another report. This report devotes a separate section to each of the remaining areas mentioned above.

## II. METHODOLOGY

A four-step procedure was used to prepare this set of guidelines:

- (1) Learn the fundamentals of how each EXEC complex works.
- (2) Identify the controllable performance parameters associated with each complex.
- (3) Outline the performance hypotheses associated with each complex.
- (4) Determine how SIP can be used to decide if the hypotheses are true.

This procedure was followed for each EXEC complex. Results for each complex are presented in separate sections.

The most difficult problem associated with this project was obtaining adequate documentation. The primary source of information was the actual EXEC program listings. Additional information was obtained from the UNIVAC EXEC Internals Course, the UNIVAC Dump Analysis Course, and the Programmer's Reference Manual.

The hypotheses presented in this report describe conditions which, if true, would have an adverse affect on performance. They are stated as if they were true, though their existence must be proven or disproven by empirical analysis. The hypotheses are not intended to be statements of fact, but rather pointers to potential performance problems.





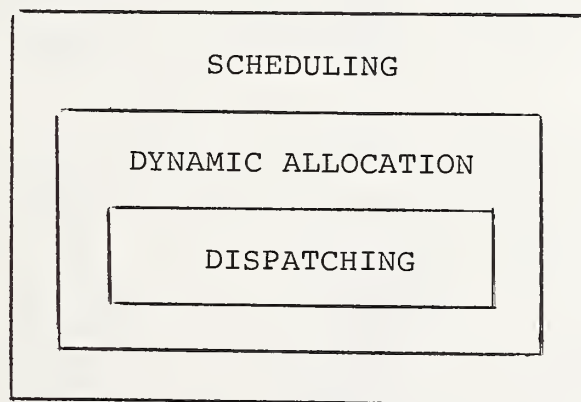
### III. PROCESSOR MANAGEMENT

#### A. INTRODUCTION

Processor management is the responsibility of the Dispatcher. The Dispatcher is the heart of the operating system; it makes the final decision about which activity to execute next. Before a run can be dispatched, it must be processed by other parts of the EXEC. The Course Scheduler (CS) decides which runs should be opened; the Dynamic Allocator (DA) decides which activities associated with those runs should be loaded into main storage; and the Dispatcher (DISP) decides which activity, of those loaded into main storage, should be given control of the CPU. The nested relationship between the EXEC functions is illustrated in Figure III-1. When a run enters the system, it must pass through a series of queues before it eventually completes executing. A queue is associated with each major EXEC complex (see Figure III-2). This section addresses only the Dispatcher Queue, also known as the Switch List (SWL). The other queues are discussed in following sections.

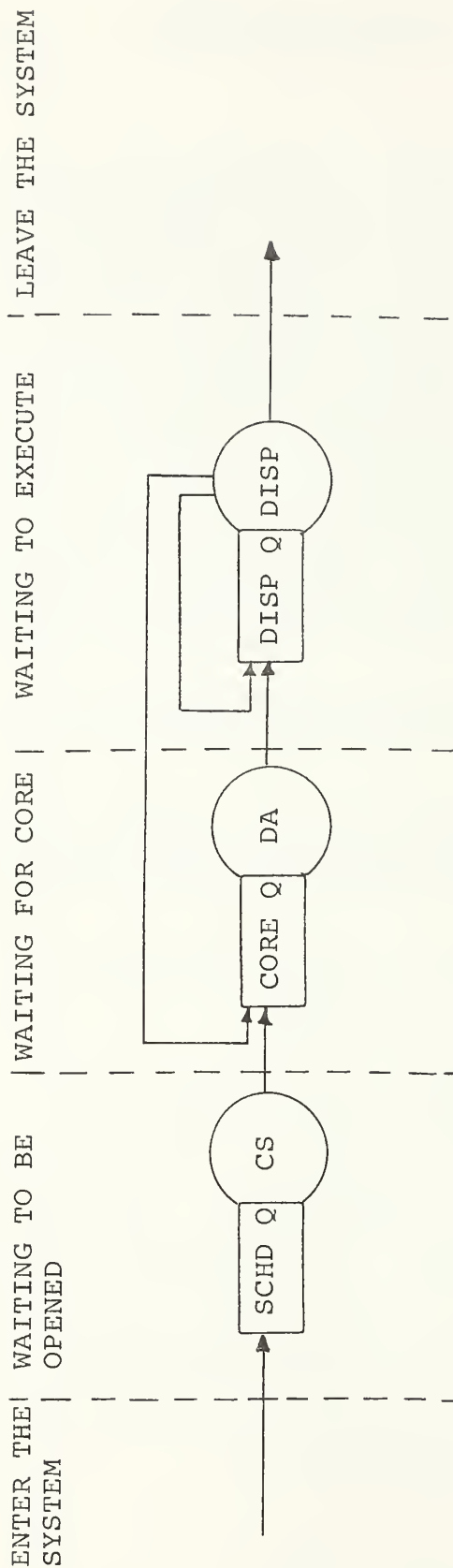
The three main performance hypotheses associated with processor management and activity dispatching are listed below:

- (1) Batch programs that are assigned the same CPU priority as demand programs are causing demand programs to have degraded response times.



NESTING OF EXEC FUNCTIONS

FIGURE III-1



MAJOR SYSTEM QUEUES

FIGURE III-2

(2) CPU quanta that are set either too high or too low are causing the CPU to be inefficiently allocated.

(3) An excessive number of test and set interrupts are causing programs to be delayed until data areas are unlocked.

Each of these problems is discussed in the sections that follow.

## B. CPU PRIORITIES

All activities in the system are controlled by an internal priority referred to as type and level (TAL). TAL is interpreted by the Dispatcher as a CPU priority. Type specifies the priority class of an activity, and level specifies a subpriority within a type. There are seven activity types numbered 0 through 6. The number of allowable levels varies for each type. TAL is denoted by a 3-digit octal number of the form TLL, where T is the type and LL is the level.

All activities are queued on the switch list according to their TAL (CPU priority). Whenever the state of the system changes, the Dispatcher selects the highest priority (lowest TAL) activity on the switch list to execute next.

Activity Types 0, 1, and 3 are reserved for EXEC activities. Type 2 is reserved for real-time activities; all other user activities operate at Types 4, 5, and 6. The activities associated with each type are summarized in Table III-1. Type 4 is reserved for Transaction Interface Package (TIP) programs. Both demand and batch programs run at Type 6.

TYPE	LEVELS	ACTIVITIES
0	00	Interlock Processing (EXEC-0)
1	01-02	High EXEC (EXEC-1)
2	01-35	Real-Time
3	01-35	Low EXEC (EXEC-3)
4	01-07	TIP
5	01-07	Deadline Batch
6	01-07	Demand and Batch

## ACTIVITY PRIORITY TYPES

TABLE III-1

This priority structure can cause performance problems for demand programs. Since demand and batch programs have the same CPU priority, demand programs may be preempted by batch programs. In previous versions of the EXEC, demand programs executed at Type 4; but, when TIP was introduced, they were reduced to Type 6. For installations that do not run TIP, demand programs may be changed back to Type 4 by changing the software parameter DSWTCHTYP. If DSWTCHTYP is changed to 4, batch throughput should be closely monitored since demand programs may lock out batch programs.

### C. CPU QUANTA

The length of time an activity is given access to a system resource is referred to as a quantum. There are two types of quanta: core quanta and CPU quanta. The Dynamic Allocator allocates main storage in terms of core quanta, and the Dispatcher allocates CPU time in terms of CPU quanta. Since an activity must be core-resident before it can be dispatched, it can receive a CPU quantum only if it is currently allocated a core quantum. Core quanta and their performance implications are discussed in a subsequent chapter.

Only Program Types 4, 5, and 6 are assigned CPU quanta; Types 0, 1, 2, and 3 are allowed to execute until completion, or until they are preempted by a higher priority activity. Quantum assignments are made according to the current level of the activity. The quanta assigned to each user level are summarized in Table III-2. These values are hardcoded in the element DISP.

USER LEVEL	QUANTUM (MILLISECONDS)
1	1.4
2	3
3	32
4	64
5	128
6	256
7	512

CPU QUANTA FOR TYPES 4, 5, AND 6

TABLE III-2



An activity initially begins execution at Level 2 and is allowed to execute for the corresponding CPU quantum time. Level 1 is reserved for the processing of I/O interrupts that occur at the completion of an I/O operation (IOIS and IOWIS). If the activity does not complete execution before its quantum expires, the level of the activity is incremented; and the activity is requeued on the switch list. This continues until the activity either completes execution or reaches Level 7. If the activity reaches Level 7, it continues there until it completes execution. One exception to the above procedure occurs when an activity voluntarily gives up control to do an I/O operation. In this case, the activity returns to Level 2, regardless of its current level.

CPU Dispatching is purely preemptive. An executing activity may be preempted by a higher priority (TAL) activity at any time. After each interrupt, the Dispatcher must determine the highest priority activity and give control to it. If the preempted activity has not exceeded its quantum, it will be requeued at the same level. The next time it is given control, it will be allowed to execute for the quantum time it has remaining.

CPU quanta are particularly sensitive to an installation's workload. If they are set too low, the CPU will spend a disproportionate percent of its time switching between activities. Because of the large number of times the Dispatcher is executed, this can represent a substantial amount of EXEC overhead. The overhead will show up in a SIP report as high EXEC 0 percentages. If the quanta are set too high, compute bound programs may degrade the response time of interactive programs. A delicate balance can only be achieved by experimentation, but the process is simplified with Level 33 of the EXEC. The SIP report for Level 33 prints the number of times an activity exceeded the quanta at each level. This information can be used to determine if the quantum assigned to a particular level should be increased or decreased.

#### D. TEST AND SET INTERRUPTS

Test and set interrupts are used to protect common data areas in a multiprocessor configuration. A common data area is one that may be concurrently accessed by more than one activity. Each area to be protected has a test and set word associated with it. Before accessing a protected area, an activity executes a test and set (TS) instruction on the protection word. If the word is clear, it is set; and the

access takes place. If the word is set, a test and set interrupt occurs; and the activity continues to retry until the word is cleared.

SIP counts the number of test and set interrupts for each test and set word. SLEX1, which protects the switch list, normally has the highest counts. Abnormally high test and set counts indicate processor contention and should be investigated.

#### IV. MEMORY MANAGEMENT

##### A. MEMORY CONFIGURATION

The maximum memory space available on UNIVAC 1108's is 262K words. Because memory is limited, it assumes increased importance when evaluating system performance. Drawing a memory configuration is a useful first step in determining how efficiently memory is being utilized. The configuration should show how much memory is reserved for system functions and how much is available for user programs. The dynamic characteristics of memory behavior will determine if it is distributed in the most efficient manner. The general memory configuration for UNIVAC 1108's is illustrated in Figure IV-1. The dynamic characteristics of memory usage are described in the sections that follow.

##### B. MEMORY ALLOCATION

User main storage is dynamically controlled within the EXEC by the Dynamic Allocator (DA). The DA is responsible for allocating main storage (core) to all active users in the system. User requests for core are maintained on the Core Request Queue (CRQ). The DA selects the highest priority request from the CRQ and attempts to assign memory to it in the most economical way possible. The desirability of assigning a particular segment of core to a program is based on a set of internal cost factors.

When programs are initially loaded into main memory, they are assigned a core quantum. A core quantum represents a guaranteed amount of time that a program is allowed to use memory. Once assigned a core quantum, a program cannot be forced to be swapped out by a higher priority activity until its core quantum has expired. Programs that have completed their core quantum are eligible to be swapped out to make room for other activities.

Gaining access to memory does not guarantee access to the CPU. An activity in memory must compete with all other activities in memory on a pure priority basis for access to the CPU. CPU access is also controlled by quanta; however, unlike core quanta, CPU quanta do not guarantee that an activity will not be interrupted by a higher priority activity.

Situations exist wherein a program may be eligible for swap out even though its core quantum has not expired. The most common situation arises when the program is waiting for

Resident EXEC I-Bank	Interrupt locations, permanently resident EXEC instructions
Permanent Segment Area	Transient segments selected for permanent residency
EXEC Segment Overlay Area	Area reserved for execution of transient segments
User Main Storage	User programs and data EXEC segments on a space available basis
EXPOOL Expansion	Expansion area if EXPOOL becomes tight
Resident EXEC D-Bank	Permanently resident EXEC data storage
EXPOOL	EXEC data storage pool Buffers are requested and released as needed

## MEMORY CONFIGURATION

FIGURE IV-1

the completion of Symbiont I/O operation (READ\$ or PRINT\$). In general, whenever a program enters a long wait state, it becomes a candidate for swap out.

Three performance hypotheses arise in connection with memory allocation:

- (1) Inefficiently assigned core priorities prevent the system from properly discriminating among users.
- (2) Core quanta that are set either too low or too high are causing unnecessary system overhead in the form of excessive swapping.
- (3) Fragmented memory is causing blocks of memory to go unused and programs to remain swapped out longer than necessary.

#### 1. Core Priorities

Access to main memory is controlled by core priorities. Two sets of core priorities are defined: in-core priorities and request priorities. That is, a program may have a different core priority while it is waiting on the CRQ than when it is actually loaded into memory. The two sets of priorities are summarized in Table IV-1. The following observations about Table IV-1 are noteworthy:

- (1) Both demand and batch programs have an in-core priority of 07 when they are using their initial quantum. This guarantees that they will be loaded before all other programs that have already received at least one quantum.
- (2) The core request priority of 10 is reserved for batch programs that have been elevated by the Dynamic Allocator Periodic Adjustment Routine (DAPA) to satisfy the demand/batch sharing algorithm.
- (3) Batch programs have a higher in-core priority than request-core priority. This assures that waiting batch programs will not cause in-core batch programs to be swapped.
- (4) EXEC segments residing in user program area have lower priorities than all user programs. Since EXEC segments may reside in user program space, this guarantees that user programs can get the space back when they need it.



(5) Common banks have higher priorities than most other users. This is significant, since system performance may be improved by configuring certain programs as common banks.

---

PRIORITY	REQUEST-CORE PRIORITIES	IN-CORE PRIORITIES
<hr/>		
00	UNUSED	RESIDENT EXEC
01	UNUSED	DOWNED MEMORY
02	REAL-TIME	REAL-TIME
03	COMMON STATIC BANKS	COMMON STATIC BANKS
04	TIP PROGRAMS	TIP PROGRAMS
05	CRITICAL DEADLINE	CRITICAL DEADLINE
06	INCORE MOVE REQUEST	COMMON DYNAMIC BANKS
07	COMMON DYNAMIC BANKS	INITIAL QUANTUM PROG
10	ELEVATED BATCH PROG	DEMAND
11-17	DEMAND	DEMAND
20-27	UNUSED	BATCH/NON-CRIT DL
30-37	BATCH/NON-CRIT DL	UNUSED
40-50	UNUSED	WRONG MEMORY TYPE
51	UNUSED	LONG-WAIT BANKS
52	UNUSED	LOAD-IN-PROGRESS SEG
53	UNUSED	ACTIVE SEGMENTS
54	UNUSED	WAITING SEGMENTS
55	UNUSED	INACTIVE SEGMENTS
56-76	UNUSED	UNUSED
77	UNUSED	AVAILABLE CORE

---

## CORE PRIORITIES

TABLE IV-1

The problems associated with demand and batch core priorities are described below.

a. Demand Core Priorities. Demand core priorities are a function of program size and the parameter CQFACT. If CQFACT is incorrectly specified, the system will not properly discriminate among demand users. The demand core priority (CQHL) is computed as

$$CQHL = 010^1 + \text{Min} (7, L)$$

---

<sup>1</sup>This report follows the UNIVAC convention of writing octal integers with an initial 0.

where L is the program level. The program level is a function of program size S and C=CQFACT, as given by the following equation:

$$L = \left\lfloor \frac{S^2 + C + C/2}{C} \right\rfloor = \left\lfloor \frac{S^2}{C} + 1.5 \right\rfloor$$

The brackets denote that only the integer portion of the result is used. The program size S is specified in core blocks (512 words).

To determine the range of program sizes that corresponds to different levels, the above equation can be solved for S:

$$S = \sqrt{C(L-3/2)}$$

Since EXEC Level 32R2 currently has C set equal to 800,

$$S = 20\sqrt{2L-3}$$

The equation for S can be used to generate Table IV-2.

The table demonstrates two important points. First, all demand programs greater than 34K have the same level and consequently the same core priority. Second, the range of program sizes assigned to a given level decreases as the level increases. That is, the level equation is more sensitive to size changes in large programs than it is to size changes in small programs.

LEVEL L	SIZE (BLOCKS) S	SIZE (K)	RANGE (K)
1	0-19	0- 9	9
2	20-34	10-17	7
3	35-44	18-22	4
4	45-52	23-26	3
5	53-59	27-29	2
6	60-66	30-33	3
7	67-	34-	-

DEMAND PRIORITY LEVELS FOR CQFACT = 800

TABLE IV-2

The range of program sizes assigned to a unique level can be increased by properly adjusting CQFACT. To double the range of programs assigned different levels, CQFACT must be increased by a factor of 4. Letting  $C = 4 * 800 = 3200$  produces Table IV-3. When CQFACT = 3200, the system distinguishes among programs in the range 0 to 67. As will be seen shortly, the core priority also affects the computation of core quanta.

LEVEL	SIZE (BLOCKS)	SIZE (K)	RANGE (K)
1	0- 39	0-19	19
2	40- 69	20-34	14
3	70- 89	35-44	9
4	90-105	45-52	7
5	106-119	53-59	6
6	120-132	60-66	6
7	133-	67-	-

DEMAND PRIORITY LEVELS FOR CQFACT = 3200

TABLE IV-3

The Dynamic Allocator Response Time Report from SIP may be used to determine if CQFACT is properly set. This report specifies the number of requests of each core priority that the DA received. If the requests are not distributed across all priorities, CQFACT should probably be adjusted.

b. Batch Core Priorities. Batch core priorities are a function of the run card priority. As was noted earlier, batch programs have different core priorities, depending on whether they are waiting to get in core or whether they are already in core. The computation of batch core priority is similar to that for demand programs:

In core:  $CQHL = 020 + L$   
Waiting for Core:  $CQHL = 030 + L$

where L is the priority level. The priority level is computed directly from the run card priority as follows:

$$L = \left\lfloor \frac{(P - 'A') * 8}{26} \right\rfloor$$

where P denotes the octal representation of the Priority P.

If the equation is evaluated for all values of P, Table IV-4 can be produced.

RUN CARD PRIORITY	LEVEL
A,B,C,D	0
E,F,G	1
H,I,J	2
K,L,M	3
N,O,P,Q	4
R,S,T	5
U,V,W	6
X,Y,Z	7

#### BATCH PRIORITY LEVELS

TABLE IV-4

Table IV-4 suggests that batch users should be assigned run priorities according to their relative importance. If all users default to the same priority, the potential benefit of being able to distinguish between different batch users is obviated. Data about how run card priorities are used may be extracted from the Master Log File.

## 2. Core Quanta

Quantum size has important performance implications. If quanta are set too long, demand response time may be lengthened. If quanta are set too short, excessive swapping overhead may be generated. The first step in determining how core quanta should be assigned is to understand how they are computed.

The core quantum of a program is a function of (1) the RUN CARD priority, (2) the time required to swap the program, (3) the parameter CQAN, and (4) the core priority. The objective is to give the program at least as much core residency time as it takes to swap the program in and out. This is an attempt to cut down on swapping overhead. More specifically, the core quantum is computed from the following formula:

$$2^{|CQHL-CLTAL|} * \left[ \frac{('Z'+25-P)}{25} * \frac{CQAN}{10} * S * SFRATE + SFLATE \right]$$

The core quanta are measured in Standard Units of Processing (SUP's). One SUP corresponds to a 200-microsecond increment. Hereafter, a 200-microsecond increment is referred to as a tic. The maximum quantum size allowed is  $2^{18}-1$  tics = 262,143 tics = 52.4 seconds. The individual terms in the core quantum formula are described below.

a. Swap Time. The basis of the core quantum calculation is the time required to swap a program. The time to swap a program in and out is given by

$$S * \text{SFRATE} + \text{SFLATE}$$

where S is the program size in core blocks (512 words), SFRATE is the time in tics required to transfer two blocks to the SWAP\$FILE device, and SFLATE is the average number of tics required to make two accesses to the SWAP\$FILE device.

The other terms in the formula are adjustments to the swap time.

b. Run Card Priority. The run card priority affects the core quantum multiplicatively. The highest run card priority effectively doubles the quantum size. The adjustment is given by

$$('Z' + 25 - P)/25$$

where P equals the priority from the run card and 'Z' = 037 = 31. For example, if P = 'S', the factor becomes  $(31+25-24)/25 = 1.28$ . The minimum and maximum values are  $(31+25-31)/25 = 1.0$  and  $(31+25-6)/25 = 2.0$ , respectively.

c. CQAN Adjustment. CQAN is the core quantum dynamic adjustment factor. It is currently hardcoded at 15. Since it is divided by 10 in the quantum equation, it increases the core quantum 1.5 times. CQAN is the only external parameter available for adjusting the core quantum size calculation.

d. Core Priority. When a program exceeds (uses up) its core quantum, its core priority is reduced; that is, its priority level is increased by one, up to a maximum of 7. The next time the program is loaded, its quantum size is doubled by multiplying the quantum by

$$\frac{| \text{CQHL} - \text{CLTAL} |}{2}$$



where CQHL is the highest priority allowed for the program and CLTAL is the priority last assigned to the program. The computation of CQHL is described in the section on core priorities (IV.B.1.a). CLTAL is given by

$$CLTAL = CQHL + N$$

where N is the number of quanta that have been used since the program began executing. Since the maximum spread between CQHL and CLTAL is 7, the core priority adjustment lies in the range  $2^0 = 1$  and  $2^7 = 128$ .

The fact that CQHL is used to compute CLTAL demonstrates the importance of correctly specifying the parameter CQFACT. If CQFACT is set too low, large programs will receive a high (low numeric value) initial core priority. This means that the program will have more opportunities to double its quantum size. System performance could be affected if many large programs begin doubling their core quanta.

e. Example. Consider the following parameter values:

$$P = 'S' = 030 = 24$$

$$S = 14 \text{ blocks} = 7K$$

$$CQAN = 15$$

$$SFRATE = 33 \text{ tics/block for } 1782$$

$$SFLATE = 220 \text{ tics for } 1782$$

$$CQHL = 011 = 9$$

$$CLTAL = 017 = 15$$

The core quantum Q is given by

$$\begin{aligned} Q &= 2^{|9-15|} * \left[ \frac{(31+25-24)}{25} * \frac{15}{10} * 14 * 33 + 220 \right] \\ &= 70851 \text{ tics} \\ &= 14.2 \text{ seconds} \end{aligned}$$

f. Problem Detection. Performance problems associated with core quanta may be detected from SIP. The DA Response Time Report specifies the number of requests that exceeded their initial core quantum. If these numbers are high, it may be advantageous to increase CQAN. Performance for important batch users may be improved by properly assigning run card priorities.

### 3. Core Fragmentation

As memory is allocated and released, empty spaces are created in memory that are too small to hold any available program. This memory space goes unused, and memory becomes fragmented. The Memory Fragmentation Report from SIP reports the number of program loads that failed because of memory fragmentation. If memory fragmentation is found to be a problem, it may be necessary to make adjustments to the memory cost allocation table (CTABL).

CTABL is a set of seven parameters the DA uses to optimize the allocation of memory. The cost table is defined with two conflicting objectives in mind: (1) maximize memory utilization and (2) minimize the amount of swapping. Swapping would be eliminated if a program were allowed to remain in memory until it completed execution; however, this would result in very poor memory utilization. The cost table attempts to achieve a balance between these two objectives.

The DA allocates memory in two phases: the first phase (Phase 0) attempts to allocate memory based on a bank's preference for primary or extended storage. Banks are considered in the following order: require primary, require extended, prefer primary, prefer extended, and no preference. The second phase (Phase 1) searches for available memory on an individual, bank-by-bank basis.

A separate set of cost factors is used for each phase. Of the seven factors in the cost tables, three are for maximizing memory utilization and four are for minimizing swapping. A sample cost table is given in Table IV-5. Each factor in the cost table is used as a cost or weight to perform a particular allocation. If  $C_i$  is the cost for factor  $i$  and  $X_i$  is the value of factor  $i$ , the cost to make a particular allocation is given by

$$\text{Allocation Cost} = \sum_{i=1}^7 C_i * X_i$$

Phase 0 cost factors attempt to minimize swapping, while Phase 1 cost factors attempt to maximize memory utilization.

COST FACTOR	PHASE 0 COST	PHASE 1 COST
Core Priority	1	1
Distance from Edge	32	1,024
Blocks not used	1	2,047
Blocks with I/D Conflict <sup>1</sup>	1,024	32
Blocks not in Preferred Storage Type <sup>1</sup>	512	8
Blocks to swap out <sup>1</sup>	1,024	64
Programs to suspend <sup>1</sup>	640	64

<sup>1</sup>Minimize Swapping

#### COST TABLE

TABLE IV-5

#### C. DEMAND/BATCH SHARING

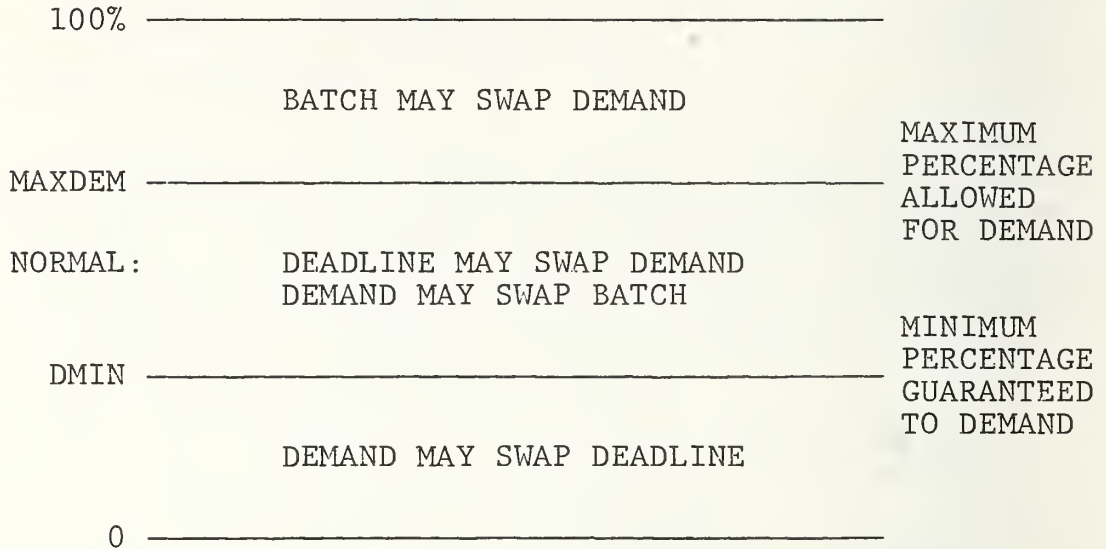
If system resources are not being properly distributed between demand and batch users, it may be necessary to adjust the parameters associated with the demand/batch sharing algorithm. The demand/batch sharing algorithm is driven by the three parameters DMIN, DMAX, and DINC. DMIN specifies the minimum percentage of core block SUP's (CBSUP's) guaranteed to demand users. DMAX is used to compute the maximum percentage of CBSUP's that may be used by demand programs. That is, DMAX specifies the minimum percentage of CBSUP's guaranteed to batch programs.

The DA Periodic Adjustment Routine (DAPA) is responsible for guaranteeing that the required percentages are being satisfied. DAPA computes the maximum allowed demand percentage as

$$\text{MAXDEM} = \text{Min} (\text{DMAX}, \text{DMIN} + \text{DINC} * \text{DOPEN})$$

where DOPEN is the number of demand runs open at a particular instant and DINC is the incremental percentage added to DMIN

for each demand run open. MAXDEM varies dynamically as the number of demand runs open varies. MAXDEM is, however, subject to an upper bound of DMAX. The entire process is illustrated in Figure IV-2.



#### DEMAND/BATCH SHARING ALGORITHM

FIGURE IV-2

DAPA computes the CBSUP's consumed by all demand programs during the previous  $n$  seconds. It then computes what percentage this is of the total CBSUP's available during that time period. If the derived percentage is larger than DMIN and less than MAXDEM, no adjustment is necessary. If the derived percentage is greater than MAXDEM, DAPA adjusts the core priorities of batch programs so that they are higher than demand programs. This will enable batch programs to swap out demand programs and consume a larger percentage of the computer resources available. Similarly, if the derived percentage is less than DMIN, demand programs are not receiving their guaranteed percentage. This is corrected by raising the priority of demand programs above deadline batch priority. SIP may be used to determine how the parameters are set. The System Idle Activity Report lists the current values for DMIN, DMAX, and DINC. It also specifies the average number



of demand runs open (DOPEN). These data can be used to determine if DINC is reasonably set. For example, DINC should be set such that

$$DMIN + DINC * AVG DOPEN < DMAX$$

The Master Log File can be used to determine the number of CBSUP's being accumulated by demand and batch programs. An estimate can be made from SIP by looking at the average program size and the percent of CPU time consumed for each type.

#### D. FUNCTION/SEGMENT ACTIVITY

The main performance problems associated with function/segment activity are the delays caused by excessive segment loads. EXEC segments are transient routines that normally reside on mass storage (disk or drum). If an activity requests the service of a segment that is not in core, that activity must wait until the needed segment is loaded. Excessive loading of high frequency segments may significantly affect system performance.

The Function/Segment Activity Report in SIP may be used to identify which segments are potential bottlenecks. Of those segments with the highest request counts, the segments with the highest load counts should be investigated in more detail. Three hypotheses must be tested:

- (1) The wrong segments have been made permanently resident.
- (2) The size of the segment overlay area is incorrectly specified.
- (3) The sticking powers assigned to the segments are incorrectly specified.

These problems are discussed in the following sections.

##### 1. Resident Segments

The most direct approach to minimizing segment loads is to make the most active segments permanently resident. This approach, however, should be reserved for the most critical situations. For example, if response time delays could be



directly related to the time required to load a particular segment, that segment should be considered a candidate for permanent residency. In many installations, the segment D5CRIT satisfies this requirement.

Segments may be made permanently resident by specifying them in an XXXSEG statement at system generation time. The XXXSEG statement specifies how EXEC segments are to be configured in the system.

## 2. Segment Overlay Area

The next most direct approach for minimizing segment loads is to increase the amount of storage permanently reserved for EXEC segments. When memory is initially configured, the EXEC reserves a fixed block of storage for the execution of EXEC segments. Unless otherwise directed, the EXEC reserves a space equal to the size of the largest segment. If the size of the largest segment is 6K, the segment overlay area may hold as many as 12 1-block (512-word) segments. The size of the segment overlay area may easily be increased by properly specifying the parameter OVSIZE.

Increasing the size of the overlay area does not guarantee that certain segments will always be resident; it only increases the probability that they will be resident. If a segment has a high sticking power (core priority), the probability will be even higher.

Segments do not have to execute in the segment overlay area, and they may (and commonly do) execute in the user program area. However, when executing in user core, they have a lower core priority than any user program. If the space they occupy is required by a user program, they will be overlaid. (EXEC segments are reentrant and are never swapped out. They are simply overlaid and reloaded when needed again).

## 3. Sticking Power

"Sticking power" is another name for a segment's core priority. The term indicates the importance of retaining a copy of the segment in core. Sticking power is a four-digit octal number in the form SSFF, where SS is the segment status and FF is the segment frequency. The order of the numbers determines that status is the most significant component of sticking power. A segment's status changes

dynamically as the state of the segment changes. The four possible states and associated status values are summarized in Table IV-6. Note how these correspond to the core priorities in Table IV-1.

STATUS	STATE
52	Load-In-Progress (LIP)
53	Active
54	Waiting
55	Inactive

#### SEGMENT STATUS VALUES

TABLE IV-6

Frequency is the only part of sticking power that can be explicitly specified. Frequency is defined at system generation time by the parameter `FREQ` in the element `AAPCT`. Frequency is an estimate of the relative expected usage of a particular segment. Since segment usage is very workload dependent, the default values may be incorrect for a particular installation.

Proper frequency values are easy to assign. The segments with the highest request counts should have the highest frequencies. Increasing a segment's frequency should increase the probability of it being in core when it is requested.

When changing segment frequency, the following idiosyncrasy should be noted. Since frequency is a component of core priority, low values correspond to high priorities. In the element `AAPCT`, just the opposite is true. High values correspond to high priorities. Frequencies in the range 00-77 may be specified.

#### E. EXPOOL

`EXPOOL`, the Executive Storage Pool, is a block of main storage reserved for use by the `EXEC`. `EXPOOL` is organized into a set of buffers that are acquired and released as required. All buffers are initially one core block (512 words) in length; however, the `EXEC` may request buffers of size  $2^n$  where  $n$  may range from 2 to 9. When the `EXEC`

requests a buffer smaller than any that already exists, a larger buffer must be successively halved until the required size is available. For example, a request for a four-word EXPOOL buffer may result in a 512 word buffer being halved seven times.

The major performance problems associated with EXPOOL are listed below:

- (1) EXPOOL size is set either too high or too low.
- (2) Not enough EXPOOL expansion blocks are reserved.
- (3) EXPOOL thresholds are improperly specified.

Each of these problems is discussed in the following sections.

### 1. EXPOOL Size

The size of EXPOOL is computed at system generation time by the formula

$$\text{EXPOOLSIZE} = \left( \sum_i P_i * W_i + \text{EXPADJ} \right) * (9/8)$$

where  $P_i$  is the value of parameter  $i$  and  $W_i$  is the number of words of EXPOOL to be configured for each unit value of parameter  $i$ . For example, consider the number of EXPOOL words required to support the maximum allowable number of batch runs. MAXOPN specifies the maximum number of batch runs allowed open, and EXPBCH estimates the number of EXPOOL words needed to support each batch run. Therefore, MAXOPN\*EXPBCH words of EXPOOL will be reserved to support batch runs. Similar requirements are computed in the element E8END for approximately 20 other parameters.

Since the EXPOOL requirement computed from the parameters is an estimate, the computed value may have to be adjusted. EXPADJ is a configuration parameter for this purpose. For extra safety, a 12.5% cushion is added to the final value. This is done by multiplying the final value by 9/8. A final adjustment, not shown in the equation, is to round the EXPOOL size up to an integral multiple of 01000.

Changes to EXPOOL size are made by changes to EXPADJ. The proper size is easily determined by monitoring the EXPOOL Activity Report in SIP. This report specifies the maximum number of EXPOOL words in use during the monitoring period. If this number is continually less than the number

of EXPOOL words allocated (also reported by SIP), EXPADJ should be set to a lower value. This may mean a larger negative number. This is particularly important on a UNIVAC 1108, wherein memory is a critical resource.

The reverse problem is also important; i.e., not enough EXPOOL is allocated. Such a condition will show up as an EXPOOL tight condition or an EXPOOL expansion. In the extreme case, all EXPOOL expansion blocks will be used, and the system will crash with an 070 stop. These problems are discussed in the next two sections.

Before assuming that EXPOOL size is too low, potential problem areas should be investigated. The most common cause of EXPOOL being used up is a program that uses an inordinate number of mass storage granule tables. Another common cause is excessive operator key-ins. Both these conditions use large numbers of EXPOOL words.

## 2. EXPOOL Expansion

Whenever an EXPOOL request is made and there are only two core blocks of EXPOOL space remaining, EXPOOL will expand. EXPOOL expands one core block at a time, up to a maximum of EXPEXP blocks. EXPEXP is a CONFIG parameter that specifies the maximum number of 512-word expansion blocks allowed. Because EXPOOL expands so that the EXPOOL area remains contiguous (see the configuration diagram in Figure IV-1), user programs may have to be swapped out so that memory can be reorganized. Once EXPOOL expands, it does not contract.

EXPEXP is an important parameter. If the system uses all the expansion blocks available and still needs more, the system will crash with an 070 stop. EXPEXP should therefore be set high enough to avoid such a catastrophe.

## 3. EXPOOL Thresholds

Before EXPOOL thresholds can be explained, the priorities associated with EXPOOL requests must be described. In addition to specifying the size of EXPOOL buffer required, each EXPOOL request has a priority associated with it. There are two main priority types: non-immediate and immediate. Non-immediate requests are non-critical requests that may be queued if no EXPOOL buffers are available. Alternatively, non-immediate requests can specify an abnormal



return rather than be queued to wait for an available buffer. Immediate requests, on the other hand, are critical. They must be satisfied immediately, and they cannot be queued. If they cannot be satisfied, the system will crash (070 stop).

Non-immediate requests are further divided into low, medium, and high priorities. If no priority is specified, medium priority is assumed. SIP summarizes the number of EXPOOL requests by both size and priority.

Associated with each non-immediate priority is a threshold. The purpose of each threshold is to guarantee that a certain number of EXPOOL words will be reserved for the next higher priority request. The thresholds are EXTLOW, EXTMED, and EXTHIG for priorities low, medium, and high, respectively. When the EXPOOL in use reaches one of the thresholds, all requests with a priority less than or equal to the priority for that threshold will be queued. For example, when the EXPOOL in use reaches EXTHIG, only immediate requests will be honored. All non-immediate requests will be queued or given abnormal returns.

Whenever EXPOOL in use reaches one of the thresholds, EXPOOL is said to be "tight." SIP reports the number of times EXPOOL became tight and the number of requests that occurred while EXPOOL was tight.

EXTLOW, EXTMED, and EXTHIG are hardcoded in the element E8END. If they are not specified properly, EXPOOL may not become tight soon enough to prevent a crash; or EXPOOL may become tight too soon and cause severe system performance degradation. They are specified as octal percentages of EXPOOLSIZE. For example, the default value for EXTHIG is  $\text{EXPOOLSIZE} * 076/0100$ . The corresponding multipliers for EXTLOW and EXTMED are  $070/0100$  and  $074/0100$ , respectively.

Another threshold used to signal a shortage of EXPOOL space is EXSCH. EXSCH is defined as  $\text{EXTLOW} - \text{EXPBCH}$ , where EXPBCH is the number of EXPOOL words required to support one batch run. If the EXPOOL in use goes above EXSCH, the Course Scheduler (CSN) prevents any additional batch runs from being opened.



## V. FACILITIES MANAGEMENT

### A. MASTER FILE DIRECTORY

The Master File Directory (MFD) contains the control information needed to manage catalogued files. At the heart of the MFD is the directory look-up table. The look-up table contains a one-word pointer to the detail information for each catalogued file in the system. The look-up table is indexed by a hash code computed from the file name. If two file names are similar, there is a chance they will have the same hash code and point to the same entry in the look-up table. If this happens, the look-up table entry must be modified to point to a search item. The search item contains pointers to the detail information for each file with that hash code.

Four performance hypotheses apply to the efficient use of the MFD:

- (1) Too small a look-up table is causing duplicate hash codes.
- (2) Many similar (near duplicate) file names are causing duplicate hash codes.
- (3) Activities attempting to search the MFD at the same time are being delayed because of contention.
- (4) The MFD is improperly located on the mass storage devices.

#### 1. Look-up Table Size

The smaller the size of the directory look-up table, the higher the probability of duplicate hash codes. Duplicate hash codes have two adverse effects: (1) the MFD takes longer to search, because an additional level of pointers is introduced that requires an additional I/O operation; and (2) mass storage space is wasted because a 28-word search item is generated for as few as two file names with the same hash code.

The size of the look-up table may be controlled by the configuration parameter DCLUTS. UNIVAC recommends DCLUTS be assigned a prime number at least twice as large as the

number of catalogued files in the system. Using a prime number reduces the probability of two file names having the same hash code.

## 2. Similar File Names

Similar file names increase the probability of duplicate hash codes. This may be minimized in two ways: (1) use a prime number for DCLUTS and (2) avoid naming conventions that repeat major portions of a file name.

A file's hash code is an index to the file's appropriate look-up table entry. This hash code is obtained from a hashing algorithm that operates on the file's file name and qualifier. The file name and qualifier are combined by the hashing algorithm to form a single 18-bit half-word. This half-word is divided by the parameter DCLUTS, which specifies the look-up table's length. The remainder from this division is the index into the look-up table. The entry corresponding to the file name/qualifier combination is found by adding the remainder to the look-up table's beginning address. Similar file names could therefore cause several files to have the same entry in the look-up table.

## 3. MFD Contention

Access to the MFD look-up table is controlled by a set of lock cells. The number of lock cells is computed from the parameter DLOKSF:

$$N = 2^{DLOKSF}$$

where DLOKSF must be an integer in the range 0 to 4. N, therefore, lies in the range 1 to 16.

The MFD is divided into N equal parts, and each part is controlled by a separate lock cell. When an activity accesses the MFD, only the portion of the MFD that contains its file is locked. The remainder of the MFD is available for concurrent use by another activity. If DLOKSF is set to 0, no concurrent accessing of the MFD may take place.

It would at first appear that DLOKSF should always be set to 4. This is not true. When the EXEC needs to lock the entire MFD, it must set each cell individually. Increasing DLOKSF by 1 doubles the overhead associated with locking the entire MFD.

#### 4. MFD Location

The MFD look-up table should be located on the fastest mass storage device available (usually a drum). The MFD detail information for each file is contained on the same device on which the file resides. This information comprises a device directory. For most mass storage devices, the first directory track begins in Position 0. Three exceptions to this rule are FASTRAND II, FASTRAND III, and 8460 disks.

The parameters POSDIR and POS60 control the positioning of the directory on FASTRAND and 8460 devices, respectively. To minimize arm and head movement, the parameters should be set so that the directory is located near the middle of the allocated file space. Alternatively, if one or two files account for the majority of a device's activity, the directory should be located near those files.

#### B. MASS STORAGE ROLLOUTS

The EXEC manages mass storage as a virtual resource. That is, from a user's viewpoint, an unlimited amount of mass storage is available. When more mass storage is requested than is physically available, the EXEC releases space by rolling out infrequently used files to magnetic tape.

The rollout process is controlled by the five thresholds, MSW1 through MSW5. These thresholds, which specify critical levels of mass storage in tracks, are computed at system generation time (see Table V-1). When the total number of unallocated mass storage tracks drops below MSW1, mass storage is said to be "tight." This tight condition is detected by the facilities allocation element FALL. When a tight condition is detected, FALL calls the element ROLOUT, which actually initiates the unload/rollout operation.

THRESHOLD	DEFINITION	TYPICAL VALUE
MSW1	MSW2+MSW4	1664 tracks
MSW2	MSW3+MSW4	1248 tracks <sub>1</sub>
MSW3	0150*MAXOPN	832 tracks <sub>1</sub>
MSW4	MSW3/2	416 tracks
MSW5	0200	128 tracks
MSTOL	MSW4/4	104 tracks

<sup>1</sup>Assumes that MAXOPN = 8.

TABLE V-1: MASS STORAGE THRESHOLDS

Separate actions are triggered as unallocated mass storage falls below each threshold. An actual rollout to magnetic tape may be initiated when MSW1 is passed. MSW2 simply causes a warning message to be printed on the operator's console. When MSW3 is passed, ROLOUT places a Coarse Scheduler Hold on all demand and batch runs. MSW4 generates another warning message. Finally, if mass storage decreases below MSW5, any user requesting mass storage for file expansion is terminated. However, any EXEC mass storage requests are still honored.

The physical rollout process is not performed by the element ROLOUT. ROLOUT starts a predefined (canned) run stream, which in turn calls the SECURE Processor. The SECURE Processor is responsible for the management of all catalogued files in the system, and rollout is one of its functions.

Four performance hypotheses are associated with the mass storage rollout process:

- (1) Mass storage rollout is occurring either too soon or too late.
- (2) Frequent rollouts are causing unnecessary system overhead.
- (3) Since an excessive amount of mass storage is rolled out during each rollout operation, frequent rollbacks are occurring.
- (4) The SECURE processor is unloading the wrong files and allowing inactive files to occupy mass storage.

#### 1. Rollout Initiation

As described above, the rollout process is based on the three threshold values MSW1, MSW3, and MSW5. Setting these values involves a trade-off. If they are set too high, the rollout process will be initiated while a significant amount of available mass storage remains. If they are set too low, the rollout process will be delayed, and mass storage requests by user programs may be terminated.

MSW1 is the most important threshold because it may initiate an actual rollout of mass storage to tape (see page 34). MSW1 should be set according to the system's file allocation activity. This activity is based on (1) the number of runs entering the system and (2) the mass storage dynamic expansion by existing runs. An additional consideration is the amount of time it takes the operator to mount



the rollout tape. If the file allocation activity causes mass storage to decrease below the MSW5 threshold before a rollout completes, MSW1 should be increased. However, MSW1 should not be so large as to cause the rollout of mass storage with a significant amount of remaining space. This would cause a portion of mass storage to remain unused.

The MSW3 threshold is important for two reasons. First, MSW1, MSW2, and MSW4 are currently computed as a function of MSW3 (see Table V-1). Thus, alterations to MSW3 must consider the effect on the other thresholds. Second, MSW3 determines when a Coarse Scheduler Hold will be placed on demand and batch runs. MSW3 should be set according to the dynamic expansion activity of the currently open runs. If this activity causes mass storage to decrease below MSW5, MSW3 should be increased. As a guideline, MSW3 should be the average file space required for a run multiplied by the average number of runs concurrently active.

Whenever mass storage is reduced to the MSW5 threshold, only EXEC mass storage requests are statisfied. MSW5 should therefore be set large enough to handle these requests.

The principal information source concerning the optimal setting of MSW1, MSW3, and MSW5 is the console warning messages. A unique warning message is displayed when available mass storage falls below each threshold. The rate at which these messages appear serves as a guideline for setting the thresholds. If the MSW1 warning message is quickly followed by the MSW2 and MSW3 messages, the MSW1 threshold should be increased. If the MSW3 message is quickly followed by the MSW4 and MSW5 messages, MSW3 should be increased. The MSW3 message should be displayed only occasionally; the MSW5 message should almost never be displayed.

## 2. Frequent Rollouts

Frequent mass storage rollouts might occur for two reasons: (1) too few tracks are being unloaded during each rollout and (2) a large number of mass storage requests occurs within a short time period. Although little can be done about the second reason, the first can be controlled.

The element ROLOUT calculates the number of tracks to be unloaded from the following formula:

$$NBTRK = \left( \frac{TC}{PERCNT} \right) - AV$$



where TC is the total mass storage capacity in tracks configured in the system, AV is the number of available (unallocated) tracks at the time of the rollout, and PERCNT is a parameter that specifies the fraction of total system mass storage tracks to be made available as result of a rollout. For example, if PERCNT = 20, 1/20th (5%) of all configured mass storage should be available after the rollout is complete. Since AV tracks are currently available, the amount to be unloaded is reduced by this amount. The number of tracks to be unloaded is constrained so that

$$MSW1 \leq NBTRK \leq 030000$$

That is, at least MSW1 tracks, but no more than 030000 tracks, are unloaded for each rollout. This constrained value of NBTRK is plugged into a SECURE Processor command to unload mass storage. The command as it appears in the predefined run stream is

$$UNLOAD\ TRACKS = NBTRK.$$

Unless the number of tracks (NBTRK) is specified, the UNLOAD command releases 3000 tracks.

The SECURE Processor first attempts to satisfy the UNLOAD command by eliminating files which have current copies on magnetic tape. Since these files are already duplicated on tape they are just marked in the MFD as being unloaded and no I/O to tape is done. As a last resort, the SECURE Processor will select files without current copies on tapes for unloading. To unload these files, I/O is required to transfer them to magnetic tape.

The number of rollouts that occurs during a certain time period may be obtained from the Master Log File. SECURE performs the rollout operation under the RUNID ROLOUT. If Master Log Data is unavailable, the SIP I/O Trace report may be used to obtain the approximate number of rollouts that occurred. This report specifies the number of words unloaded for each file. Converting the number of words to tracks and summing them will give the total number of tracks unloaded. Dividing the total number of tracks by NBTRK will yield the approximate number of rollouts.

### 3. Frequent Rollbacks

Frequent rollbacks may occur because too much mass storage is unloaded during each rollout. This problem is easily resolved by increasing PERCNT so that fewer tracks are unloaded each time.

#### 4. Unload Eligibility Factor

The unload eligibility factor (UEF) determines which files are to be unloaded during a rollout operation. If the UEF is improperly specified, files may have to be reloaded (rolled back) shortly after they are rolled out. The SECURE Processor computes a UEF for each catalogued file. Those files with the highest UEF are unloaded first.

The UEF is a function of six factors: (1) time since last reference, (2) mean time between references, (3) size, (4) equipment type, (5) F-Cycle, and (6) type: public or private. Of these factors, only the first three are significant. The UEF is computed from the following formula:

$$\text{UEF} = \text{CURNCY} * P(\text{TSLR}) + \text{FREQCY} * P(\text{MTBR}) + \text{SIZEWT} * P(\text{SIZE})$$

where TSLR equals time since last reference, MTBR equals mean time between references, and SIZE equals file size in tracks. The notation  $P(\text{FACTOR})$  denotes the number of points assigned to the specified factor. CURNCY, FREQCY, and SIZEWT are weights associated with each factor.

The point values assigned to each factor are obtained by a table look-up operation. The table used to look-up the points for TSLR and MTBR is given in Table V-2. The point values for SIZE are given in Table V-3. The UEF is normalized so that it lies in the range 0 to 63. Three UEF values are reserved for special use. A UEF of 0 is reserved for files that are already unloaded; a UEF of 1 is reserved for catalogued tape files and removeable disk files; and a UEF of 2 is reserved for files that are "unload inhibit" (V option).

The optimal point values and weights depend on the unique file characteristics of an installation. The characteristics of each file may be extracted from the MFD and summarized by the program LISTER. In release 19R1 Version 2 of the SECURE Processor, CURNCY is set to zero. This implies that TSLR has no effect on which files are selected for unloading. The other weights have a value of 1 in the same release of SECURE.

The calculation of UEF by SECURE is illustrated by the following example. The file ACOB\*PFPROJECT has the following characteristics:

TSLR = 4 days 15 hours = 4.63 days

Time since catalogued = 104 days

Number of references = 46

HOURS	DAYS	POINTS	HOURS	DAYS	POINTS
1	0.04	1	80	3.33	16
2	0.08	2	98	4.08	17
3	0.13	3	116	4.83	18
4	0.17	4	134	5.58	19
7	0.29	5	152	6.33	20
10	0.42	6	182	7.58	21
13	0.54	7	212	8.83	22
16	0.67	8	242	10.08	23
21	0.88	9	272	11.33	24
28	1.17	10	302	12.58	25
34	1.42	11	344	14.33	26
40	1.67	12	386	16.08	27
50	2.08	13	428	17.83	28
60	2.50	14	470	19.58	29
70	2.97	15	512	21.33	30
			544	23.08	31

TABLE V-2: POINT VALUES FOR TSLR AND MTBR

TRACKS	POINTS	TRACKS	POINTS
20	1	500	16
40	2	550	17
60	3	600	18
80	4	650	19
100	5	700	20
120	6	810	21
140	7	920	22
160	8	1030	23
180	9	1140	24
200	10	1250	25
250	11	1360	26
300	12	1470	27
350	13	1580	28
400	14	1690	29
450	15	1800	30
		1900	31

TABLE V-3: POINT VALUES FOR SIZE

MTBR = 2.26 days

SIZE = 2 tracks

From Tables V-2 and V-3, the point values are

P (TSLR) = 18

P (MTBR) = 14

P (SIZE) = 1

Therefore, the UEF is  $0*18+1*14+1*1 = 15$ .

### C. GRANULE TABLES

Mass storage is allocated in increments called granules. Granules may be specified in either tracks or positions. One track equals 64 sectors or 1792 words; one position equals 64 tracks or 114,688 words. The EXEC sets up buffers called granule tables to keep track of the physical mass storage assigned to each file. Granule tables contain 26-word entries. Each entry in the granule table reserves a physical block (granule) of mass storage. If track granularity is specified, each word in the granule table represents 1792 words of mass storage. If position granularity is specified, each word in the granule table represents 114,688 words of mass storage. If a file occupies non-contiguous mass storage space, the intervening space must also be included in the granule tables. Granule tables are stored on mass storage and brought into EXPOOL buffers when they are needed.

Four performance hypotheses apply to granule table usage:

(1) Users are specifying the wrong file granularity on their assign cards.

(2) The maximum number of granule tables allowed in EXPOOL when EXPOOL becomes tight is inefficiently specified.

(3) The maximum number of mass storage sectors requested at one time for file granule tables is too small.

(4) The default values for the maximum number of granules assigned to a file is excessive.



## 1. File Granularity

A file is allocated mass storage in granules. Thus, if a file with position granularity uses only one track of a position, the entire position is allocated to that file. A complete track is allocated to a file when track granularity is specified.

Specifying track granularity utilizes mass storage more effectively than position granularity. If a file with track granularity uses only part of a track, only a portion of a track is unused. However, if position granularity is specified for the same file, 63 additional tracks would be allocated to the file. These additional 63 tracks will be unused and unavailable for allocation to other files. Since track granularity results in a finer allocation of mass storage, it is usually more efficient than position granularity.

Position granularity should be specified only for large, contiguous files. The main advantage of position granularity is a reduction in main storage usage for PCT and EXPOOL buffers. Position granularity uses a single granule or word of memory to specify 64 contiguous tracks. For the same 64 tracks, track granularity will use 64 granules or words of memory. There is therefore a reduction in the amount of main storage used. The swap file is one case wherein position granularity will reduce the amount of main storage needed for granule tables.

## 2. Granule Tables in EXPOOL

A file's granule tables, which are located in the directory of the file's device, are chained together. Whenever part of a file is accessed, the granule table referencing that part is brought into EXPOOL. If an I/O operation references a granule not addressed by the granule table in EXPOOL, the other granule tables are read in one at a time until the proper granule table is found.

The parameter NGTB specifies the maximum number of a file's granule tables permitted in memory when EXPOOL becomes tight (when the remaining number of EXPOOL buffers has dropped below a critical threshold). If a file has more than NGTB granule tables when EXPOOL becomes tight, granule tables are paged out until only NGTB remain.



Two performance problems are associated with NGTB. First, if NGTB is set too large, valuable EXPOOL space is occupied by excess granule tables. Second, if NGTB is set too small, excessive granule table paging may result. As an example, suppose NGTB were set to one and a random access file were alternatively referencing two granule tables. Each reference would cause one granule table to be written out and the other to be read in. If NGTB were equal to 2 and both granule tables were in EXPOOL, the paging would be eliminated.

NGTB may be determined by deciding the amount of EXPOOL that should be reserved for granule tables when EXPOOL is tight. If AMT denotes this amount, NGTB may be found by the following formula:

$$\text{NGTB} = \frac{\text{AMT}}{\text{NFILE} * 032}$$

where NFILE is the number of concurrent files using granule tables and 032 represents the number of words per granule table.

### 3. Granule Table Mass Storage Sectors

File granule tables are maintained in the device directory, and each granule table occupies one mass storage sector. The hardcoded parameter MAXSEC specifies the maximum number of granule tables (mass storage sectors) that may be requested by the EXEC at one time. The EXEC element FALL determines whether the number of requested sectors is greater than MAXSEC. If the requested number is greater, MAXSEC is used. MAXSEC is also used to determine the required EXPOOL buffer size for the element FASEC. (FASEC allocates sectors of mass storage space for the EXEC.)

The main problem associated with MAXSEC is setting it too small. If a request is for more sectors than MAXSEC, the remaining sectors will be requested again. Overhead is generated because multiple FASEC requests occur when obtaining the required number of mass storage sectors. MAXSEC should be set to the 80th or 90th percentile of the most number of granule tables needed to describe the largest file.

### 4. Default Value for Granules

An option on the facility control statements allows the user to specify the maximum allowable file length in granules.

When this value is unspecified, MAXGRN is substituted. MAXGRN is a configuration parameter that defines the default value for a file's maximum number of granules. MAXGRN or the specified value is used to indicate when the file's length exceeds the maximum number of granules.

The primary reason for specifying maximum file length is to prevent a user from accidentally overallocating his file space. Because this is more likely to occur during the debug phase of a program, a possible value for MAXGRN is the 80th percentile of all file sizes.

#### D. MASS STORAGE MANAGEMENT PRACTICES

Improvement in facility performance is obtainable through mass storage management. An overall review of mass storage utilization should be made before the facility parameters are altered. This would eliminate most of the simple mass storage problems. Since mass storage usage is dynamic, the review should be made on a periodic basis. Depending on the system activity, a biweekly or monthly review may be appropriate.

Three performance hypotheses apply to mass storage management:

- (1) Incorrect file placement is causing excessive I/O times and unit contention.
- (2) Non-contiguously allocated mass storage file space is causing excessive I/O and unused mass storage.
- (3) Mass storage space is being wasted by unused cataloged files.

##### 1. File Placement

Several factors must be considered when attempting to optimize file placement: the size of the file, the frequency of file access, the potential for file contention, and the performance characteristics of the available mass storage devices. In general, the most frequently accessed files should be placed on the fastest devices. If the files are large or the devices are small, this may not be reasonable. Similarly, if many frequently accessed files are placed on the same device, a device contention bottleneck may result.

The following guidelines may be helpful when attempting to optimally place files:

- (a) Place the most frequently accessed files (usually EXEC files) on the fastest device (usually a drum). Contention and its resulting queuing may have to be tolerated if no more high speed devices are available.
- (b) Distribute large, frequently accessed files across several devices to minimize the possible contention.
- (c) Assure that files are allocated contiguous space on the devices to which they are assigned.
- (d) Distribute frequently accessed files across several devices of the same speed.
- (e) If the controller is found to be the source of most queuing (rather than the individual devices), consider installing a dual access controller.

The information needed to make the above decisions may be obtained from a combination of SIP, MFD map (DIRVER), and MFD list (LISTER). The Main Storage Analysis by Unit Report in SIP summarizes the queuing associated with the controllers and the units. DIRVER may be used to identify the largest files; LISTER may be used to identify the most frequently referenced files.

## 2. Non-contiguous Mass Storage Allocations

Non-contiguous mass storage occurs when mass storage is allocated in clusters throughout a device. This causes a newly catalogued file to be allocated in pieces throughout the device. Non-contiguous allocations cause an increase in the number of seeks required to read or write a file. As a result, the I/O times for this file will be greater than for a contiguously allocated file. There will also be an increase in overhead because of the multiple-parting (M-part) of I/O requests, i.e., the user makes I/O requests for a group of logically contiguous words that are actually non-contiguous on mass storage. The EXEC must compute additional access control words (ACW's) in order to handle these non-contiguous blocks.

The most convenient indicator of whether a system's mass storage space is allocated non-contiguously is the number of



multiple-parting I/O requests that occur. The SIP report that deals with the number of times various EXEC areas were entered contains this information. This report provides the number of times the EXEC area dealing with M-parts was entered. If this number is large, non-contiguously allocated mass storage is indicated. Specific, large, frequently accessed catalogued files with non-contiguous allocations can be identified by using (1) a MFD list (LISTER) for the number of accesses since the file was catalogued and (2) a MFD map (DIRVER) for determining if the files were non-contiguous.

Two actions may alleviate non-contiguous mass storage space. First, unused catalogued files should be deleted (as discussed in the following section). This will make more mass storage available and increase the probability of a file being allocated contiguously. Second, the file administrator should periodically review how large, frequently used files are allocated. If a file is identified as being non-contiguously allocated, it should be transferred to tape and rewritten to a contiguous mass storage area. To make a large enough contiguous space, several smaller files may also need reallocation. The device that the file is being allocated to should be considered at the same time that the file is reallocated. If the file is accessed frequently, assigning the file to a higher speed device may be advantageous.

### 3. Unused Catalogued Files

One possible reason for mass storage unavailability is the abundance of unused, mass storage catalogued files. A good user practice is to delete a catalogued file when it is no longer needed. If the users always delete their unused catalogued files, (1) more mass storage space is available, (2) there is less chance of allocating mass storage non-contiguously, and (3) mass storage rollouts will be reduced. If there are users who do not delete their unused files, then a procedure such as archiving might be employed. Archiving is the deletion from mass storage of all catalogued files not referenced for a fixed time period. These deleted files are transferred to tape. At the same time the file is deleted from mass storage, the file's Master File Directory entry should also be eliminated. This will reduce the number of unnecessary directory items. For example, catalogued files not accessed for seven days might be archived to tape at the end of each week. Ideally, archiving will prevent unnecessary EXEC initiated rollouts when mass storage allocation activity is highest. The archiving process should be done during non-production hours.

Another possible reason for mass storage unavailability is that users are creating catalogued files instead of using temporary files. This causes the same situation as would the unused catalogued files. The users should create a catalogued file only when it is necessary.

## E. SCHEDULING

The Course Scheduler handles the EXEC scheduling functions. Batch runs are placed into a schedule queue after the legality of the @RUN control statement is checked. Demand runs are immediately opened. When either a batch or demand run is opened, the Course Scheduler reads the run's first set of facility control statements. If the requested facilities are unavailable, the run is placed in a facilities hold queue. When the run's facilities are available, the run is fully accepted and the Dynamic Allocator is given control.

Two performance hypotheses apply to scheduling:

- (1) The number of demand and batch runs simultaneously open does not effectively utilize computer resources.
- (2) The installation is not taking advantage of the run card priorities to assist in scheduling work.

### 1. Demand/Batch Runs Open

A system's run mix can consist of demand and batch runs. The emphasis that the Course Scheduler places on either run type should be based on the objective of the computer system. If the objective is service to demand users, care should be taken to see that performance is not degraded by batch runs. However, the number of batch runs allowed simultaneously open should be set high enough to utilize those computer resources incompletely used by demand runs. For example, batch runs may use the CPU while demand runs are waiting for I/O to complete.

The two parameters that affect the balance between demand and batch are CDHOLD and MAXOPN. CDHOLD specifies the maximum possible number of simultaneously active demand runs. Similarly, MAXOPN specifies the maximum possible number of batch runs open at one time. CDHOLD should be set to allow the greatest number of active demand runs without



(1) degrading demand response time to an unacceptable level or (2) preventing the minimum required batch throughput. MAXOPN should then be set high enough to allow the greatest number of batch runs to be simultaneously open without degrading demand response time. If MAXOPN is set as large as possible, the Course Scheduler will control the computer utilization. The Course Scheduler will select from the backlog those batch runs that will use the remaining available computer resources.

Data needed for setting CDHOLD and MAXOPN are obtained from two SIP reports. The System Idle Activity report gives the average number of demand and batch runs open; the Response Time Report provides the demand response time. Demand response time is an indicator of the service that users are obtaining. If the average number of open demand runs is close to CDHOLD and demand response time is small, CDHOLD can be increased. The same can be done for batch by adjusting MAXOPN. If demand response time becomes unacceptable, CDHOLD and/or MAXOPN can be reduced until satisfactory service is obtained (assuming some other condition is not at fault).

## 2. Run Priorities

The proper use of run priorities will result in (1) providing equitable system availability to all users and (2) establishing a priority structure that will allow critical runs to receive better service. Whenever possible, user runs should be assigned a priority. This allows the discrimination of various types of runs. For example, a distinction should be made between a run's degree of I/O- versus CPU-proneness. An I/O-prone run should receive a higher priority than a CPU prone run. Of course, the urgency of the run is another factor to be considered.

MPRIOR is one of two configuration parameters that affect run priorities. MPRIOR specifies the maximum allowed priority on a @RUN statement. If the @RUN card priority is higher than MPRIOR, then MPRIOR is substituted. MPRIOR is useful in determining the number of priority classes that may be assigned to a run. Since the possible priorities are "A" through "Z" (with "A" being a higher priority than "Z"), up to 26 different classes (A through Z) are possible. If MPRIOR is set to R, only nine classes are possible.

APRIOR is the second configuration parameter that affects run priorities. APRIOR is the default priority used for @RUN statements with no priority specified. Assuming that most runs have a priority assigned, APRIOR should be set to a lower priority than the majority of runs with priorities assigned. However, if most runs default for priority, the purpose of a priority structure is defeated.

The number of runs at each priority is obtainable from the Master Log File. A report could be written that summarizes the general characteristics of those runs at each priority. From this report, it may be determined if the current priority structure is satisfactory or if runs with common characteristics need assignment to a different run priority.



## VI. INPUT/OUTPUT

### A. LATE ACKNOWLEDGMENTS

A late acknowledgment is the failure of the CPU to respond to a control unit within a critical response time. Every time a word is transferred between the CPU and a control unit, the CPU must respond to a signal from the control unit. For example, when a word is ready for input, the control unit activates its Input Data Request (IDR) line. When the word has been received by the CPU, the CPU sends an Input Acknowledge (IA) signal back to the control unit acknowledging receipt. The control unit can accept another word from the device only after the IA signal has been received from the CPU. Otherwise, the previous word might be over-written before it has been accepted by the CPU. During data output, the CPU must respond to an Output Data Request (ODR) signal with an Output Acknowledge (OA) signal within a critical response time. If a late acknowledgment occurs, the next word is not transferred until the device's next revolution.

Four performance hypotheses apply to late acknowledgments:

- (1) Non-optimal drum interlace factors are resulting in slow effective transfer rates.
- (2) Scatter read and gather write requests on a disk subsystem are causing persistent late acknowledgments.
- (3) Symbiont scatter/gather I/O is causing late acknowledgments.
- (4) Late acknowledgments are occurring on a memory module overloaded by I/O requests.

#### 1. Interlace Factors

Five possible interlace factors are used on the FH-432, FH-1782, and FH-880 drums: 1, 2, 4, 8, and 16. An interlace of 1 implies that logically consecutive data words are stored in contiguous drum locations. If the drum interlace is  $n$ ,  $n-1$  drum locations are skipped between consecutive data words. Doubling the interlace factor halves the drum transfer rate. Interlace factors may be altered either by resetting a switch or by physically changing several integrated circuit boards within the drum's controller. The

configuration parameters DHLACE, DHLACH, and DHLAC8 tell the EXEC what interlace factors are being used for the FH-432, FH-1782, and FH-880 drums, respectively. If they are incorrectly specified, the EXEC will erroneously compute timing information associated with the drums.

The optimum interlace factor involves a trade-off between obtaining the highest data transfer rate and the number of late acknowledgments that occur. Whenever a late acknowledgement occurs, an additional drum revolution (called a missed revolution) is required to read or write the next word. The number of late acknowledgments might therefore be great enough to offset the doubling of the transfer rate by using a lower interlace factor. The optimum interlace factor should be the least possible value of 1, 2, 4, 8, and 16, where the number of late acknowledgments will not decrease the effective transfer rate below the slower transfer rate obtainable at a higher interlace factor.

The SIP Input/Output Activity Report is useful in determining the optimum interlace factor. The following information is obtained from this report:

TIME = Total channel busy time in seconds

WORDS = Number of words transferred

ACW = Number of I/O operations

AVAXST = Time per I/O operation in milliseconds  
excluding data transfer time

AVAXST includes both the access time and the time lost due to missed drum revolutions. The number of missed drum revolutions N may be estimated by

$$N = \frac{ACW * (AVAXST - A)^1}{D}$$

---

<sup>1</sup>This formula depends on two important assumptions that may not always be true: (1) the number of ACW's equals the number of I/O operations and (2) the true average access time is equal to the theoretical. The first assumption is not true for scatter/gather I/O, and the second is not true for a sequence of I/O operations to the same area on drum. Each subsequent I/O request will have missed the next data word and will have to wait another complete revolution to access the word.



where A equals the average access time of the drum and D equals the delay per missed drum revolution. The average access times for the FH-432 is 4.3 ms and the FH-1782 is 17.0 ms. The delay times for a missed revolution are 8.5 ms for the FH-432 and 34.0 ms for the FH-1782.

The effective transfer rate ER (including the missed drum revolutions) is found by the following formula:

$$ER = W / (W/R + N * D)$$

In this formula, W equals the number of words transferred (as recorded by SIP) and R is the maximum transfer rate at the present interlace. The factor (W/R) is the time to transfer W words at R words per second; the factor (N\*D) is the time lost due to missed drum revolutions. If the effective transfer rate ER is less than the transfer rate obtained at the next higher interlace factor, the higher factor should be used. If ER is greater than the next higher interlace factor's transfer rate, the present interlace factor is satisfactory. Another, similar experiment should be performed at the next lower interlace factor to determine if a higher transfer rate is possible on the drum.

## 2. Scatter/Gather I/O Requests

Late acknowledgments may occur on a disk subsystem during scatter read and gather write requests. A scatter read is an I/O technique wherein a contiguous block of data that resides on a peripheral device is read into multiple, non-contiguous main storage buffers. These data are transferred in a single, continuous I/O operation. A gather write is the converse I/O operation--writing from several main storage areas to one mass storage area.

An I/O operation uses an access control word (ACW) to specify the length and location of each I/O buffer in main storage. During a scatter/gather request, the EXEC must switch control between ACW's as the I/O operation transfers data to or from each buffer. This switching of ACW's is a potential cause of late acknowledgments on a disk subsystem with a high transfer rate. Since the I/O operation is continuous, ACW's must be switched within the time interval between the transfer of consecutive data words.

Late acknowledgments may be minimized by the configuration parameter INCORE. INCORE applies only to disk subsystems. When INCORE is set to one, code is generated to

ensure that each ACW specifies a main storage buffer whose length is an integer multiple of a disk's record length. This guarantees that ACW switching will occur on record boundaries. When ACW's are switched on a record boundary, more time is available for changing ACW's because additional time is required to pass over the interrecord gap.

The SIP Input/Output Activity Report may be used to determine if late acknowledgments are occurring on a disk subsystem. The following information is obtained from this report:

ACW        = Number of I/O operations

AVAXST    = Time per I/O operation in milliseconds  
            excluding data transfer time

The number of late acknowledgements N may be estimated by

$$N = \frac{ACW * (AVAXST - A)}{D}$$

Since AVAXST includes both the access time and time lost due to missed disk revolutions, the disk's average access time A is subtracted from AVAXST. The value D equals the delay per missed disk revolution. If scatter/gather requests are used often and N is large, INCORE should be set to 1.

As general guidelines, UNIVAC recommends that INCORE be set to 1 on (1) any 1106 system or (2) an 1108 system with 8440 disks configured. If persistent late acknowledges occur on an 1108 or 1106 II system, INCORE should also be set to 1. INCORE should be set to 0 on (1) 1108 systems configured with 8414, 8424, or 8425 disks on a high priority channel or (2) any 1110 system.

### 3. Symbiont Scatter/Gather Requests

Late acknowledgements may also occur during symbiont scatter/gather I/O requests. The symbiont complex maintains its data in chains of 28-word buffers within core. During an I/O operation, symbiont data are read into or written from this chain of eight buffers. A total of 224 words is transferred at one time. These words are transferred directly to or from these eight buffers during a scatter/gather I/O. Since the I/O operation is continuous, the EXEC must be able to change buffers between the transfer of consecutive words. If the EXEC cannot switch to the next buffer fast enough, a late acknowledgment occurs.

Symbiont late acknowledgments may be reduced by setting the configuration parameter SCGA to 0. Symbiont scatter/gather is disabled when SCGA equals 0. Instead of transferring data directly to or from the eight 28-word buffers, a contiguous 224-word EXPPOOL buffer is used. During a write operation, the data of the eight 28-word buffers are transferred into a 224-word EXPPOOL buffer. The actual I/O operation will then write the data from the 224-word buffer to a peripheral device. Conversely, the data read from a peripheral device will be transferred into the 224-word buffer and then later dispersed into a chain of eight 28-word buffers.

UNIVAC states that SCGA must equal 0 if U20 tapes exist on an 1106, 1108, 1108A or 1100/20 system. The U20 tapes are not supported because of their high transfer rate of about 53,333 words per second.

#### 4. Memory Module Overloading

Late acknowledgments, also called data overruns, may occur when a memory module becomes saturated by I/O accesses. This will occur when a memory module's transfer rate is less than the combined transfer rate of the subsystems active on the memory module. The number of late acknowledgments (data overruns) can be reduced by setting the configuration parameter IOLM to 1. IOLM equated to 1 causes the assembly of code for the EXEC's I/O Load Monitor.

I/O Load Monitor reduces late acknowledgments by adjusting the I/O activity on memory modules that are nearly saturated with I/O requests. I/O Load Monitor does this by maintaining the available I/O transfer rate (LMLVL) for each memory module. LMLVL is the module's maximum I/O transfer rate with the sum of subsystem transfer rates currently active on the module subtracted out. Before a request goes to the device handler, I/O Load Monitor compares this additional subsystem's transfer rate with LMLVL of the request's memory module. If the request's subsystem transfer rate is greater, the request is queued. If not, the request's subsystem transfer rate is subtracted from LMLVL, and the request goes to the device handler. When a request's I/O completes, the memory module's LMLVL is increased by the amount of the completing request's subsystem transfer rate. Another adjustment is made to LMLVL, depending on whether a data overrun occurred during the completed request's I/O. If an overrun occurred, LMLVL is decreased by at least 12.5% of LMLVL. This will reduce the number of future data overruns. If the request completed normally and a queue exists, LMLVL is increased by at least 1.56% of LMLVL.



Information about the number of late acknowledgments (data overruns) that are occurring can be obtained from the SIP report I/O Load Monitor Activity. If nonrecoverable data overruns are occurring, the percentages used to adjust a memory module's LMLVL might be altered.

IOLM may be set to 0 (turned off) when the sum of the subsystem transfer rates is less than a memory module's maximum I/O transfer rate. For example, a system could have up to five FH-1782 drums transferring data to one memory module before the sum of the subsystem transfer rates would equal a memory module's maximum transfer rate. However, if data overruns begin to occur, the monitor should be turned on (IOLM = 1).

#### B. SEEK TIME

The time to access a disk record is the sum of seek time and latency time. Seek time is the time needed to move the disk's read/write head to the required cylinder. Latency time is the time the disk rotates before the desired position comes under the read/write head. Under pre-seeking, the EXEC initiates disk seeks for requests that are queued. When those requests are serviced, the disk will be on position and no seek time will be required.

Four performance hypotheses apply to pre-seeking:

- (1) Pre-seeking is disabled on 8440 disk drives.
- (2) The parameters used for selecting the next I/O requests for pre-seeking are set wrong.
- (3) Angular addressing on the 8440 disk subsystem is disabled.
- (4) Full disk pack seeks are degrading disk performance.

#### 1. 8440 Disk Pre-seeking

When UNIVAC first introduced the 8440 disks, a hardware error was associated with the 8440 pre-seeking function. As a result, the parameter NOSEEK was placed in the element IO. This parameter prevented 8440 pre-seeking until the problem was corrected. Currently, NOSEEK is equated to 1-STPAUL. If NOSEEK equals 1 (STPAUL = 0), 8440 pre-seeking is disabled. NOSEEK equated to 0 (STPAUL = 1) enables 8440 pre-seeking.

The hardware error associated with 8440 pre-seeking was corrected by Field Change Order (FCO) Number 037 (applied to the disk controller). Field Support Bulletin Number 252, dated 22 August 1974, details the application of pre-seeking to the 8440 disks. If a site has this hardware error corrected, then NOSEEK should be equated to 0 in order to enable pre-seeking. This problem does not apply to such other disks as the 8414's and 8460's.

The amount of improvement obtained by pre-seeking depends on the computer system's configuration and the system's workload. Pre-seeking should provide significant improvement on an 8440 disk subsystem if (1) there are more than two disks per control unit or more than three disks per dual access controller, (2) the workload is I/O-prone, and (3) the file activity is distributed evenly between disk units and throughout each disk unit. The SIP Input-Output Activity report is useful for determining whether pre-seeking is advantageous. Pre-seeking should decrease the percent of time a disk subsystem's channel is busy. The value of AVAXST should also decrease (the average time per I/O operation, excluding data transfer time).

## 2. On-Position Requests

The EXEC element IO attempts to minimize disk access time (read/write head movement) by servicing on-position requests before other requests. An on-position request is a request that references the cylinder where the read/write head of the disk is currently positioned. The parameter MAXOPR limits the number of on-position I/O requests that will be serviced consecutively. After MAXOPR on-position requests have been serviced, the disk's position is advanced one position. The disk I/O handler moves the read/write head in one direction until all requests in that direction have been checked. When an edge is reached, the scan direction is reversed. If no on-position requests are found when searching the request queue, the request closest to the current disk position (in the scan direction) is selected for service. Since the disk only scans in one direction at a time, it is possible for the previous on-position request to be closest and again given service.

To prevent a request from being permanently locked out by on-position requests, the parameter MAXLLC is used. MAXLLC specifies the maximum number of times that a request may be passed over. A request is considered passed over if, after having been selected as the closest request, it is



then rejected because either an on-position request or a closer request was found. A request may be rejected only once each time the request queue is searched. When IO detects a request that has been rejected MAXLLC times, that request is serviced immediately.

MAXLLC should be set larger than MAXOPR; otherwise, the purpose of MAXOPR will be defeated. The value of MAXOPR should be based on the length of time that requests may be locked out because of on-position requests. The time allowed depends on whether the computer's workload is demand or batch. If the workload is demand, a series of on-position requests should be allowed to monopolize a device for only a short period of time (say, one-half second). In a batch environment, the potential lockout time may be longer (say, one second). After the allowed lockout time is established, MAXOPR can be calculated from the SIP Input-Output Activity report. For each disk subsystem, the average I/O time per request is obtained by dividing the channel busy time by the number of ACW's. MAXOPR for the subsystem is then obtained by dividing the allowed lockout time by the average I/O time per request. If several disk subsystems exist, potential values of MAXOPR should be calculated for each subsystem. The actual value assigned to MAXOPR should be based on the most active subsystem.

For example, assume the maximum allowed lockout time in a demand environment is established at 0.5 seconds. From SIP, the following data may be obtained:

8414 Channel Busy Time = 567,544 seconds

Number of 8414 ACW's = 17,119

Assuming one I/O request per ACW, the average I/O time per request is 0.033 seconds. MAXOPR is calculated to be 15 requests (0.5/0.033).

### 3. Angular Addressing

Angular addressing, also called rotational position sensing, is another feature of the 8440 disk subsystem. Like pre-seeking, Field Change Order (FCO) Number 037 must be implemented on the 5033 Control Unit before the 8440 disks may use angular addressing. Angular addressing is controlled by the parameter ANGLOF.

When an I/O request is initiated, a command is given to the control unit specifying an angular position on a track. The control unit will then determine if the disk is more than a given distance away from the addressed position. If it is, the channel is logically disconnected from the operation. Just before the disk rotates to the position where the read or write will occur, the channel is logically reconnected and the read or write occurs. This assumes that the channel was not busy transferring another unit's data when reconnection was supposed to take place. If the channel was busy, the procedure is repeated during subsequent disk revolutions. In these cases, the service time will actually be longer than it would be without angular positioning.

#### 4. Full Pack Seeks

A performance problem occurs if a disk pack has frequently used files allocated at the extremes of the pack. A simplified example is one file located on the pack's outer cylinder and another file located on the inner cylinder. This would generate full pack seeks and would result in an increase in access times.

A special case of this problem is treated by the configuration parameter FASTSK. When the EXEC initially allocates disk space to a file, the inner cylinder is allocated first. Once the inner cylinder is full, the EXEC returns to the outer cylinder to continue allocation. Since this process often splits a frequently accessed system file, frequently full pack seeks may occur. By setting FASTSK to 1, the inner cylinder is marked allocated, and file fragmentation is prevented.

One disadvantage associated with using FASTSK is a reduction in the amount of space available for allocation. If an 8414 disk is prepped at 112 words per record, UNIVAC states that 36 out of 3072 tracks will be unavailable for allocation. A general solution to the seeking problem is to force frequently accessed files to be placed close together. This would eliminate frequent full pack seeks and not reduce the amount of mass storage available for allocation.

#### C. BLOCK SIZE

I/O block size is the number of words transferred in one I/O operation. Block size may degrade performance in either of two ways:

(1) Block sizes that are not integer multiples of the physical disk record size are causing frequent read before write operations.

(2) Small average block sizes are resulting in low effective transfer rates and high EXEC I/O overhead.

Each of these problems are discussed below.

## 1. Read Before Writes

Read before write I/O operations are generated by the EXEC when the I/O block size is not an integer multiple of the physical record size. Since the smallest unit of I/O is the physical record size, a block which is smaller than the record size must be written out as if it were a full block. Since this might destroy good information at the end of the physical record, the EXEC (1) reads the physical record into core, (2) changes the part of the record that is being written into, and (3) writes the entire record back to disk.

Read before write operations may be detected by SIP I/O Trace. If a large number of them are occurring, two approaches may be used to reduce their frequency: (1) require programs to specify an I/O block size that is an integer multiple of the physical record size or (2) change the physical record size to agree with frequently used I/O block sizes.

In general, disks may be prepped at one, two, or four sectors per record. Since there are 28-words per sector, disk record sizes may be 28, 56, or 112 words per record. The only exception to the above rule is that a disk used for EXEC files must be prepped at 28 or 112 words per record. Since most disks are prepped at 112 words per record, I/O block sizes should be multiples of 112 words. Since some EXEC I/O is done in 28-word blocks, it may be advantageous to prep the system disk at 28 words per record.

## 2. Small Block Sizes

I/O block size is the number of words transferred in one I/O operation. When this number is small, the effective transfer rate for a large volume of data is substantially reduced. This is because access times are typically much longer than transfer times. Consider the following example

on a 1782 drum with a transfer rate of 240,000 words per second and an average access time of 17 msec. A file of 10,000 words blocked at 100 words per block would require 1.74 seconds to transfer. Of this time, only 0.04 seconds is devoted to the actual transfer of data; the remaining time is due to access time. The same file transferred in blocks of 1000 words would require only 0.21 seconds to transfer, an eightfold improvement.

Besides reducing I/O time, large block sizes substantially reduce EXEC overhead. Fewer interrupts are generated, resulting in less I/O processing. The trade-off is the increased memory size required to hold the buffers.





## VII. TEST AND DEBUG

The EXEC contains parameters that control internal error checking and the maintenance of trace entries. If the system is unstable and crashes frequently, these parameters are a valuable asset in debugging the system. Most of this debugging information is found in a PANIC dump. On a stable system, however, these parameters can be turned off and the number of trace entries can be reduced.

Two performance hypotheses that relate to test and debug parameters are:

- (1) EXEC instructions are being executed to perform unneeded error checking and event tracing.
- (2) Critical memory space is being used for storing trace entries and test and debug program instructions.

Table VII-1 lists the test and debug parameters and identifies how each parameter adversely affects EXEC overhead. For the purposes of this table, the cutoff for low and high time is based on the frequency that the code must be executed; the cutoff used for low and high memory usage is 200 words. Each of these parameters is described in Appendix B.

---

<u>ADVERSE AFFECT ON EXEC OVERHEAD</u>		
<u>PARAMETER</u>	<u>TIME</u>	<u>SPACE</u>
DADBUG	Low	Low
DATRAC	High	Low
EXPTRACE	High	Low
FCDBSZ	High	High
FNCCHK	High	Low
IODEBUG	High	High
SIPDAT	High	Low
SIPIO	High	Low
STPAUL	High	High
SWPCKO	High	Low
UNIVAC	Low	High

---

### TEST AND DEBUG PARAMETERS

TABLE VII-1



APPENDIX A  
PERFORMANCE HYPOTHESES



## I. PROCESSOR MANAGEMENT

1. CPU Priorities. Batch programs that are assigned the same CPU priority as demand programs are causing demand programs to have degraded response times.
2. CPU Quanta. CPU quanta that are set either too high or too low are causing the CPU to be inefficiently allocated.
3. Test and Set Interrupts. An excessive number of test and set interrupts are causing programs to be delayed until data areas are unlocked.

## II. MEMORY ALLOCATION

1. Core Priorities. Inefficiently assigned core priorities prevent the system from properly discriminating among users.
2. Core Quanta. Core quanta that are set either too low or too high are causing unnecessary system overhead in the form of excessive swapping.
3. Core Fragmentation. Fragmented memory is causing blocks of memory to go unused and programs remain swapped out longer than necessary.
4. Demand/Batch Sharing. System resources are not being properly distributed between demand and batch users.

## III. FUNCTION/SEGMENT ACTIVITY

1. Resident Segments. The wrong segments have been made permanently resident.
2. Segment Overlay Area. The size of the segment overlay area is incorrectly specified.
3. Sticking Power. The sticking powers assigned to the segments are incorrectly specified.

## IV. EXPOOL

1. EXPOOL Size. EXPOOL size is set either too high or too low.
2. EXPOOL Expansion. Not enough EXPOOL expansion blocks are reserved.
3. EXPOOL Thresholds. EXPOOL thresholds are improperly specified.



## V. MASTER FILE DIRECTORY

1. Look-up Table Size. Too small a look-up table is causing duplicate hash codes.
2. Similar Filenames. Many similar (near duplicate) filenames are causing duplicate hash codes.
3. MFD Contention. Activities attempting to search the MFD at the same time are being delayed because of contention.
4. MFD Location. The MFD is improperly located on the mass storage devices.

## VI. MASS STORAGE ROLLOUTS

1. Rollout Initiation. Mass storage rollout is occurring either too soon or too late.
2. Frequent Rollouts. Frequent rollouts are causing performance degradation.
3. Frequent Rollbacks. Since an excessive amount of mass storage is rolled out during each rollout operation, frequent rollbacks are occurring.
4. Unload Eligibility Factor. The SECURE Processor is unloading the wrong files and allowing inactive files to occupy mass storage.

## VII. GRANULE TABLES

1. File Granularity. Users are specifying the wrong file granularity on their assign cards.
2. Granule Tables in EXPOOL. The maximum number of granule tables allowed in EXPOOL when EXPOOL becomes tight is inefficiently specified.
3. Granule Table Sectors. The maximum number of mass storage sectors requested at one time for file granule tables is too small.
4. Default Value for Granules. The default value for the maximum number of granules assigned to a file is excessive.

## VIII. MASS STORAGE PRACTICES

1. File Placement. Incorrect file placement is causing excessive I/O times and unit contention.

2. Non-Contiguous Mass Storage Allocation. Non-contiguously allocated mass storage space is causing excessive I/O and unused mass storage.

3. Unused Catalogued Files: Mass storage space is being wasted by unused catalogued files.

#### IX. SCHEDULING

1. Demand/Batch Runs Open. The number of demand and batch runs simultaneously open does not effectively utilize computer resources.

2. Run Priorities. The installation is not taking advantage of the run card priorities to assist in scheduling work.

#### X. LATE ACKNOWLEDGMENTS

1. Interlace Factors. Non-optimal drum interlace factors are resulting in slow effective transfer rates.

2. Scatter/Gather I/O Requests. Scatter read and gather write requests on a disk subsystem are causing persistent late acknowledgments.

3. Symbiont Scatter/Gather Requests. Symbiont scatter/gather I/O is causing late acknowledgments.

4. Memory Module Overloading. Late acknowledgments are occurring on a memory module overloaded by I/O requests.

#### XI. SEEK TIME

1. 8440 Disk Pre-seeking. Pre-seeking is disabled on 8440 disk drives.

2. I/O Request Selection. The parameters used for selecting the next I/O requests for pre-seeking are set wrong.

3. Angular Addressing. Angular addressing on the 8440 disk subsystem is disabled.

4. Full Pack Seeks. Full disk pack seeks are degrading disk performance.

#### XII. BLOCK SIZE

1. Read Before Writes. Block sizes that are not integer multiples of the physical disk record size are causing frequent read before write operations.

2. Small Block Sizes. Small average block sizes are resulting in low effective transfer rates and high EXEC I/O overhead.

### XIII. TEST AND DEBUG

1. Execution Time. EXEC instructions are being executed to perform unneeded error checking and event tracing.

2. Memory Space. Critical memory space is being used for storing trace entries and test and debug program instructions.

APPENDIX B  
PERFORMANCE PARAMETERS





This appendix summarizes those parameters identified as important for tuning system performance. The appendix begins with a table (Table B-1) that alphabetically lists all the parameters and identifies the EXEC area they affect. Detailed descriptions of each parameter, organized into a section for each EXEC area, follow the table. Each parameter has the following title line associated with it:

TYPE, ELEMENT, LINE NO, VALUE

where

TYPE	=	The parameter type: CP if configuration and SW if software (hardcoded)
ELEMENT	=	Name of EXEC element where the parameter is defined
LINE NO	=	Line number within the element where the parameter is defined on System 7 as of 9 June 1976
VALUE	=	Value on System 7 as of 9 June 1976

EXEC AREA AFFECTED							
	C	M	E	F	S	I	T
	P	E	X	A	C	/	E
	U	M	P	C	H	O	S
		O	O	I	E		T
		R	O	L	D		/
		Y	L	I	U		D
				T	L		E
				I	I		B
				E	N		U
				S	G		G
PARAMETER	C	M	E	F	S	I	T
ANGLOF						I	
APRIOR					S		
BAWAIT		M					
CDHOLD					S		
CHGDED					S		
SKSM				F			
CLRCOR		M					
COR\$MAX			E				
COUNT	C						
CQAN		M					
CQFACT		M					
CTABL		M					
CURNCY				F			
DADBUG							T
DATRAC							T
DCLUTS				F			
DEDLIN					S		
DHLACE						I	
DHLACH						I	
DHLAC8						I	
DINC		M					
DLOKSF				F			
DMAX		M					
DMIN		M					
DSWTCHTYP	C						
EXPADJ			E				
EXPEXP			E				

PERFORMANCE PARAMETERS  
TABLE B-1

EXEC AREA AFFECTED							
	C	M	E	F	S	I	T
	P	E	X	A	C	/	E
	U	M	P	C	H	O	S
		O	O	I	E		T
		R	O	L	D		/
		Y	L	I	U		D
				T	L		E
				I	I		B
				E	N		U
				S	G		G
PARAMETER	C	M	E	F	S	I	T
EXPRIM			E				
EXPSPL			E				
EXPTRACE							T
EXSCH			E				
EXTHIG			E				
EXTLOW			E				
EXTMED			E				
FASTSK						I	
FCDBSZ							T
FNCCHK							T
FREQ		M					
FREQCY				F			
INCORE						I	
IODEBUG							T
IOLM						I	
LOGEOF				F			
LOGTYP				F			
LWAIT	C						
MAXCRD					S		
MAXCRQ					S		
MAXDEM					S		
MAXF4						I	
MAXF8						I	
MAXF17						I	
MAXGRN				F			
MAXLLC				F			
MAXOPN					S		

PERFORMANCE PARAMETERS  
TABLE B-1

EXEC AREA AFFECTED							
	C	M	E	F	S	I	T
	P	E	X	A.	C	/	E
	U	M	P	C	H	O	S
		O	O	I	E		T
		R	O	L	D		/
		Y	L	I	U		D
				T	L		E
				I	I		B
				E	N		U
				S	G		G
PARAMETER	C	M	E	F	S	I	T
MAXOPR						I	
MAXPAG					S		
MAXRTC	C						
MAXSEC				F			
MAXTIM					S		
MAXTUX					S		
MAXTWAIT	C						
MPRIOR					S		
MSTOL				F			
MSW1				F			
MSW2				F			
MSW3				F			
MSW4				F			
MSW5				F			
NGTB				F			
NHOLDT					S		
NOSEEK						I	
OVSIZ		M					
PERCNT				F			
PMIN		M					
POOLSIZE						I	
POSDIR				F			
POS60				F			
PRIEXP			E				
QTM TBL	C						
RECBIN			E				
RSINCT					S		

PERFORMANCE PARAMETERS  
TABLE B-1



EXEC AREA AFFECTED							
	C	M	E	F	S	I	T
	P	E	X	A	C	/	E
	U	M	P	C	H	O	S
		O	O	I	E		T
		R	O	L	D		/
		Y	L	I	U		D
				T	L		E
				I	I		B
				E	N		U
				S	G		G
PARAMETER	C	M	E	F	S	I	T
SBLEV1							T
SBLEV2							T
SBLEV3							T
SBLEV4							T
SBLEV5							T
SCGA						I	
SIPDAT							T
SIPIO							T
SIZEWT				F			
SSDENS				F			
SSEQPT				F			
STORE		M					
STPAUL							T
SWPCKO							T
SWAP		M					
SYMALL						I	
SYSRP				F			
TFEXP				F			
TFMAX				F			
TRMXCO					S		
TRMXPO					S		
TRMXT					S		
UEFTRK				F			
UEFLUP				F			
UNIVAC							T
WHLDRM				F			
XXXSEG		M					

PERFORMANCE PARAMETERS  
TABLE B-1

## PROCESSOR PARAMETERS

COUNT	SW, DISP, 134, EQU 01000
DSWTCHTYP	SW, DAILR, 13, EQU BARH
LWAIT	SW, AAPCT, 3041, EQU 50
MAXRTC	SW, DISP, 132, EQU 35000
MAXTWAIT	SW, DISP, 133, EQU 30001
QTMTBL	SW, DISP, 3677, See Below

## PROCESSOR PARAMETERS

COUNT SW, DISP, 134, EQU 01000

COUNT specifies the number of words to transfer for the Block Transfer instruction in the Dispatcher IDLE Loop.

DSWTCHTYP SW, DAILR, 13, EQU BATCH

DSWTCHTYP makes the demand switching type equal to the Batch Type (6). This was done to reserve Type 4 for TIP processing. If no TIP processing is being done, DSWTCHTYP may be reevaluated.

LWAIT SW, AAPCT, 3041, EQU 50

LWAIT is the length of wait time in milliseconds. It is used by routines that execute an ER TWAIT\$ to compute the desired length of wait in msec. It is also used by the Dispatcher as a threshold to determine if the segment long wait bit (SGLWAT) should be set. If the wait time is greater than or equal to (5\*LWAIT), set SGLWAT.

MAXRTC SW, DISP, 132, EQU 35000

MAXRTC is the maximum real time clock value in 200  $\mu$ sec increments. The value 35000 is equivalent to 7 sec. MAXRTC is used as the "infinite" quantum size for Activity Types 1 through 3.

MAXTWAIT SW, DISP, 133, EQU 30001

MAXTWAIT is the maximum time wait value in milliseconds. The value 30001 is equivalent to 30 sec. MAXTWAIT is the longest length of time an activity will be allowed to wait on the TWAIT queue.

QTMTBL SW, DISP, 3677, See Below

QTMTBL is a 7-word table that holds the CPU quanta for Levels 1 through 7 for

PROCESSOR PARAMETERS (Cont'd)

Activity Types 4 through 6. The values in milliseconds for the 1108 are:

---

LEVEL	QUANTUM (msec)
1	1.4
2	3.0
3	32.0
4	64.0
5	128.0
6	256.0
7	512.0

---

## MEMORY PARAMETERS

BAWAIT	SW, DAPAM, 18, EQU 21
CLRCOR	CP, AACONFIG, 732, EQU 1
CQAN	SW, DA, 173, +15
CQFACT	SW, ADAEQT, 17, EQU 800
CTABL	SW, DA, 280, See Below
DINC	CP, AACONFIG, 789, EQU 2
DMAX	CP, AACONFIG, 802, EQU 50
DMIN	CP, AACONFIG, 812, EQU 20
FREQ	SW, AAPCT, 4996, See Below
OVSIZ	CP, AACONFIG, 1068, EQU 0
PMIN	SW, ADAEQT, 21, EQU 5
STORE	CP, AACONFIG, 2193, See Below
SWAP	CP, AACONFIG, 2903, See Below
XXXSEG	CP, AACONFIG, 2891, See Below



## MEMORY PARAMETERS

BAWAIT SW, DAPAM, 18, EQU 21

BAWAIT specifies the number of minutes a batch program will be allowed to wait on the Core Request Queue (remain swapped out) before DAPAM will requeue the core request at a priority higher than all other demand batch requests (priority 010)

CLRCOR CP, AACONFIG, 732, EQU 1

If CLRCOR is nonzero, core is always cleared before loading a program. Core is cleared regardless of the use of the B option at collection.

CQAN SW, DA, 173, +15

The core quantum adjustment factor, CQAN, is one of several factors used to compute a program initial core quantum. Besides CQAN, core quantum is a function of (1) RUN card priority, (2) swap time, and (3) core priority. The initial core quantum is computed by SETCQU in DACCC in 200  $\mu$ sec increments.

The maximum core quantum allowed is  $2^{18}-1 = 262,143$  tics = 524 secs (1 tic = 200  $\mu$ sec). The core quantum is computed from

$$2^{|CQHL-CLTAL|} * \left[ \frac{'Z'+25-P}{25} * \frac{CQAN*SWAPTIME}{10} \right]$$

where CQHL = Highest allowed priority

CLTAL = Current priority

CQAN = Core quantum adjustment factor

P = RUN card priority

CQFACT SW, AADAEQT, 17, EQU 800

CQFACT is used by CQLVL in DACCC and CCR in DA to compute CQHL, the core priority for demand programs. Demand core priorities lie in the range 011 to 017. They are computed from

## MEMORY PARAMETERS (Cont'd)

$$CQHL = 010 + \text{MIN}(7, L)$$

where L is the program level. The program level is a function of program size and C = CQFACT, as given by the following equation:

$$L = \left[ \frac{S^2}{C} + 1.5 \right]$$

The brackets denote that only the integer portion of the result is used. The program size is specified in core blocks (512 words).

CTBAL

SW, DA, 280, See Below

The cost table (CTABL) is a set of seven parameters used to optimize the allocation of memory. The seven parameters are (1) core priority, (2) distance of area from edge of user space, (3) blocks within area not in use, (4) blocks with I/D bank conflict, (5) blocks not in preferred memory type, (6) blocks that would have to be swapped out, and (7) programs that would have to be suspended. Each factor  $X_i$  listed above has a cost  $C_i$  associated with it. The cost of allocating a program to a given area is then given by

$$\text{Allocation Cost} = \sum_{i=1}^7 C_i * X_i$$

The area with the lowest cost is then selected for allocation.

DINC

CP, AACONFIG, 789, EQU 2

DMAX

DC, AACONFIG, 802, EQU 50

DMIN

CP, AACONFIG, 812, EQU 20

DMIN, DMAX, and DINC are the basic inputs to the demand/batches sharing

## MEMORY PARAMETERS (Cont'd)

algorithm in DAPA. DMIN specifies the minimum percent of computer time guaranteed to demand programs. DMAX and DINC are used to compute MAXDEM:

$$\text{MAXDEM} = \text{MIN} (\text{DMAX}, \text{DMIN} + \text{DINC} * \text{DOPEN})$$

where DOPEN is the number of demand runs open at any instant in time.

MAXDEM specifies the maximum amount of time allowed for demand programs. This guarantees that batch programs will receive at least  $(100 - \text{MAXDEM})$  percent of the system resources.

FREQ

SW, AAPCT, 4996, See Below

Frequency (FREQ) is a component of segment sticking power. Sticking power is a four-digit octal number of the form SSFF, where SS is the segment status and FF is the frequency. Status may have one of four values, depending on a segment's current state: 52 = load-in-progress, 53 = active, 54 = waiting, and 55 = inactive. Frequencies are in the range 00 to 077 and are defined at system generation time in the element AAPCT. Sticking power is important because it is issued as a segment's core priority.

OVSIZ

CP, AACONFIG, 1068, EQU 0

OVSIZ defines the function/segment overlay size. OVSIZ is the number of core blocks devoted exclusively to transient EXEC segments. Increasing this value may increase throughput but will decrease the amount of core available to user programs. A value of 0 will cause the system to assign the minimum amount of overlay core needed to avoid core lock up, i.e., an amount equal to the size of the largest segment.

## MEMORY PARAMETERS (Cont'd)

PMIN SW, AADAEQT, 21, EQU 5

PMIN specifies the number of minutes a common static bank will be locked in memory if unused. DAPAM scans the EXEC Bank Descriptor Table (BDT) and marks swappable all common static banks that have not been referenced for PMIN minutes.

STORE CP, AACONFIG, 2193, See Below

STORE is a label in a System Generation Statement (SGS) that defines the main storage configuration. It has the form:

STORE FIRST, LENGTH, 'TYPE', INTLV

where

FIRST = First address of area  
LENGTH = Length of area in 01000's  
TYPE = Memory type, e.g., '7005' for 1108  
INTLV = 1 for no interleave and  
          2 for odd/even interleave

The fields are repeated for each area to be defined. For example:

STORE 0,0100, '7005', 2

SWAP CP, AACONFIG, 2903, See Below

SWAP is a label in an SGS that defines the sizes and allocation of the program swap file SWAP\$FILE. It has the form:

SWAP MAX, 'GRN', 'DEVICE', GRAN

where

MAX = Maximum number of granules that may be assigned to the file.

'GRN' = 'TRK' for track granularity  
          'POS' for position granularity

## MEMORY PARAMETERS (Cont'd)

'DEVICE' = Any Fastrand formatable device type that may be used on an ASG card, e.g., 'F17' = FH-1782 drum

GRAN = Number of granules to be reserved on this device

'DEVICE', GRAN may be repeated as many times as needed.

Example: SWAP 25, 'POS', 'F17', 14, 'F2', 1

XXXSEG

CP, AACONFIG, 2891, See Below

XXXSEG is a label in an SGS that specifies how EXEC segments are to be configured in storage. It has the form:

XXXSEG, COND 'SEG1', 'SEG2',...

where

XXXSEG = XRESEG if segments are to be made permanently resident for execution in storage.

XXXSEG = XTPSEG if transient segments are to be executed in primary storage. Does not apply to 1106/1108.

XXXSEG = XRPSEG if segments are to be executed in primary storage. Does not apply to 1106/1108.

COND = An expression which, if it evaluates to zero, indicates that the XXXSEG statement should be ignored.

'SEGN' = A list of segments to which the XXXSEG statement applies.

Example: XRESEG, NOMUX 'DSCRIT'



### EXPOOL PARAMETERS

COR\$MAX	CP, AACONFIG, 739, EQU 32
EXPADJ	CP, AACONFIG, 370, EQU 076000
EXPEXP	CP, AACONFIG, 375, EQU 10
EXPRIM	CP, AACONFIG, 868, EQU 0
EXPSPL	CP, AACONFIG, 889, EQU 'H'
EXSCH	SW, E8END, 285, EQU EXTLOW-EXPBCH
EXTLOW	SW, E8END, 275, EQU EXTLNG*070/0100
EXTMED	SW, E8END, 277, EQU EXTLNG*074/0100
EXTHIG	SW, E8END, 279, EQU EXTLNG*076/0100
PRIEXP	CP, AACONFIG, 1127, EQU 0
RECBIN	SW, EXPOOL, 19 EQU 0

## EXPOOL PARAMETERS

COR\$MAX	CP, AACONFIG, 739, EQU 32	COR\$MAX is the maximum number of log entries that is kept in core at any one time. Each log entry uses a 32-word EXPOOL buffer. A trade-off exists between EXPOOL usage and logging overhead.
EXPADJ	CP, AACONFIG, 370, EQU 076000	The EXEC estimates the amount of EXPOOL required (EXPREQ) based on (1) the maximum number of demand and batch jobs that may be open simultaneously and (2) the particular hardware configuration. Since EXPREQ may over or under estimate the amount of EXPOOL actually required, EXPREQ can be adjusted by specifying an appropriate positive or negative value for EXPADJ. The number of words eventually reserved for EXPOOL is given by EXPOOLSIZE and is computed in E8END.
EXPEXP	CP, AACONFIG, 375, EQU 10	EXPEXP is the number of 512-word EXPOOL expansion blocks allowed. EXPOOL does not contract once an expansion takes place.
EXPRIM	CP, AACONFIG, 868, EQU 0	EXPRIM is the percent of EXPOOL in Primary Storage (does not apply to 1106/8).
EXPSPL	CP, AACONFIG, 889, EQU "H"	EXPSPL is the EXPOOL storage priority after which EXPOOL gives buffers in extended storage. This allows for fine tuning by shifting allocation among PS and ES (does not apply to 1106/8).
EXSCH	SW, E8END, 285, EQU EXTLOW-EXPBCH	EXSCH is used to signal to the system that EXPOOL is approaching a tight condition. EXSCH is equated to the low EXPOOL threshold

## EXPOOL PARAMETERS (Cont'd)

less the estimated number of EXPOOL words required for one batch run. EXPBCH is hard-coded in E8END to be 850. EXSCH is used by the Coarse Scheduler (CSN) to prevent any additional runs from being opened, it is used in the UOM change in DAPA to indicate an EXPOOL hold on the operator's console, and it is used by FIMAIN to determine if a facility's hold should be placed on a run.

EXTLOW            SW, E8END, 275, EQU EXTLNG \* 070/0100

EXTMED           SW, E8END, 277, EQU EXTLNG \* 074/0100

EXTHIG           SW, E8END, 279, EQU EXTLNG \* 076/0100

Threshold values are used to determine when EXPOOL is tight for the three types of non-immediate priority requests. That is, if a low priority EXPOOL request is made when the EXPOOL storage in use is greater than EXTLOW, EXPOOL is said to be tight for that request, and that request will be queued until the EXPOOL in use drops below EXTLOW. EXTLNG is the length EXPOOL storage in Extended Storage; on the 1108, this is just EXPOOLSIZE.

PRIEXP           CP, AACONFIG, 1127, EQU 0

PRIEXP is the number of primary EXPOOL expansion blocks allowed (does not apply to 1106/8).

RECBIN           SW, EXPOOL, 19, EQU 0

RECBIN is used as a conditional assembly flag to turn EXPOOL recombination code on and off. Its possible values have the following meaning: 1 = complete recombining and 0 = do not complete recombining.

## FACILITIES PARAMETERS

CKSM	CP, AACONFIG, 332, EQU 1
CURNCY	SW, SECURE (MAIN), 737, +0
DCLUTS	CP, AACONFIG, 765, EQU 2049
DLOKSF	CP, AACONFIG, 793, EQU 2
FREQCY	SW, SECURE (MAIN), 739, +1
LOGEOF	CP, AACONFIG, 488, EQU 0
LOGTYP	CP, AACONFIG, 463, EQU 0
MAXGRN	CP, AACONFIG, 992, EQU 128
MAXSEC	SW, FALL, 83, EQU 125
MSTOL	CP, AACONFIG, 1014, EQU MSW4/4
MSW1	CP, AACONFIG, 1012, EQU MSW2+MSW4
MSW2	CP, AACONFIG, 1011, EQU MSW3+MSW4
MSW3	CP, AACONFIG, 1009, EQU 0150*MAXOPN
MSW4	CP, AACONFIG, 1010, EQU MSW3/2
MSW5	CP, AACONFIG, 1013, EQU 0200
NGTB	SW, FALL, 82, EQU 5
PERCNT	SW, ROLOUT, 17, EQU 10
POSDIR	CP, AACONFIG, 1099, EQU 191
POS60	CP, AACONFIG, 1107, EQU 135
SIZEWT	SW, SECURE (MAIN), 738, +1
SSDENS	CP, AACONFIG, 611, EQU 'V'
SSEQPT	CP, AACONFIG, 614, EQU '16N'
SYSPRP	CP, AACONFIG, 621, EQU 4

FACILITIES PARAMETERS (Cont'd)

TFEXP	CP, AACONFIG, 633, EQU 60
TFMAX	CP, AACONFIG, 1327, EQU 360
UEFLUP	SW, SECURE (SETUP), 2425, See V.B.4
UEFTRK	SW, SECURE (SETUP), 2430, See V.B.4
WHLDRM	CP, AACONFIG, 659, EQU 1



## FACILITIES PARAMETERS

CKSM CP, AACONFIG, 332, EQU 1

CKSM activates the checksum feature of all Master Bit Tables. The Master Bit Table associated with each device will be lengthened by one word because of the addition of a Checksum Word at the end. CKSM is also used as a conditional assembly flag for code that (1) adjusts each Master Bit Table's length, (2) generates the checksum, and (3) determines the validity of the checksum.

CURNCY SW, SECURE (MAIN), 737, +0

CURNCY is the weight assigned to time since last reference (TSLR) in the computation of the unload eligibility factor (UEF). A weight of 0 implies that TSLR does not affect which files are unloaded. This setting applies to Level 19R1 VERS 2 of SECURE.

DCLUTS CP, AACONFIG, 765, EQU 2049

DCLUTS specifies the size in words of the Directory Loop-Up Table. The Directory Look-Up Table is part of the Master File Directory (MFD) and is used to index into the other tables in the directory. The MFD contains the identification and characteristics of each catalogued file.

DLOKSF CP, AACONFIG, 793, EQU 2

DLOKSF determines the number of MFD lock cells and is called the MFD lock scale factor. Each MFD lock cell prevents a portion of the directory look-up table from being used. DLOKSF must be an integer from 0 to 4. The number of MFD lock cells (one 36 bit word per cell) configured will be:

$$2^{\text{DLOKSF}}$$

## FACILITIES PARAMETERS (Cont'd)

Thus, the number of lock cells can range from 1 to 16.

FREQCY SW, SECURE (MAIN), 739, +1

FREQCY is the weight assigned to mean time between references (MTBR) in the computation of the unload eligibility factor UEF. This setting applies to Level 19R1 VERS 2 of SECURE.

LOGEOF CP, AACONFIG, 488, EQU 0

LOGEOF, in conjunction with LOGTYP, controls the conditional assembly of code that specifies action taken on log tape during autorecovery. LOGEOF is valid only when LOGTYP equals 1. If LOGEOF equals 0, an EOF will not be written; and the log tape will not be re-wound on autorecovery. The EXEC will continue writing on the extended tape. If LOGEOF equals 1, an EOF will be written on the extended log tape; a rewind will be done; and the next F-cycle assigned.

LOGTYP CP, AACONFIG, 463, EQU 0

LOGTYP controls conditional assembly flags. Code is assembled, depending on whether the Master Log is written on a FASTRAND device or on tape. LOGTYP equals 0 specifies that the Master Log is written on a FASTRAND device. LOGTYP equal to 1 specifies that the Master Log is to be written on tape. LOGTYP is also used in conjunction with LOGEOF to specify action taken during autorecovery.

MAXGRN CP, AACONFIG, 992, EQU 128

MAXGRN specifies the maximum number of granules assigned to a file when no maximum is provided.

## FACILITIES PARAMETER (Cont'd)

MAXSEC SW, FALL, 83, EQU 125

MAXSEC specifies the maximum number of mass storage sectors containing file granule tables that may be requested at one time. FALL checks if the number of requested sectors is greater than MAXSEC. If the requested number is greater than MAXSEC, MAXSEC is used instead. MAXSEC is used in the same manner to determine the required EXPOOL request buffer size for the element FASEC. FASEC is the EXEC element that allocates sectors of mass storage space for EXEC use.

MSW1 CP, AACONFIG, 1012, EQU MSW2+MSW4

MSW2 CP, AACONFIG, 1011, EQU MSW3+MSW4

MSW3 CP, AACONFIG, 1009, EQU 0150\*MAXOPN

MSW4 CP, AACONFIG, 1010, EQU MSW3/2

MSW5 CP, AACONFIG, 1013, EQU 0200

MSTOL CP, AACONFIG, 1014, EQU MSW4/4

MSW1 through MSW5 are thresholds used in the rollout process to determine the criticality of mass storage availability. A console message is displayed whenever mass storage availability goes below each threshold, and a different message is displayed when the availability exceeds each threshold. MSTOL makes the amount of available mass storage appear less than it actually is when mass storage availability increases. Thus, MSTOL prevents "togglng" between console messages when mass storage oscillates around a threshold.

NGTB SW, FALL, 82, EQU 5

NGTB specifies the maximum number of granule tables that a catalogued file may have in

## FACILITIES PARAMETERS (Cont'd)

EXPOOL when EXPOOL becomes 'tight'. EXPOOL is 'tight' when the available EXPOOL Storage drops below EXTLOW words (an EXPOOL parameter) threshold. When a low-priority EXPOOL request is refused during the addition of another granule table, the number of granule tables in EXPOOL is reduced to NGTB.

PERCNT

SW, ROLOUT, 17, EQU 10

PERCNT specifies the fraction of total system mass storage tracks to be made available as a result of a rollout. A rollout is initiated when the number of available (unassigned) mass storage tracks drops below the threshold MSW1. If the amount of mass storage available at the time of the rollout is denoted by AV, the number of tracks (NBTRK) to be unloaded to tape is given by

$$\text{NBTRK} = \frac{\text{TC}}{\text{PERCNT}} - \text{AV}$$

where TC is the total mass storage capacity in tracks configured on the system.

NBTRK is constrained so that it will be in the interval

$$[\text{MSW1}, 30000_8]$$

That is, at least MSW1 tracks, but no more than 30,000<sub>8</sub> tracks, will be unloaded for each rollout.

POSDIR

CP, AACONFIG, 1099, EQU 191

POS60

CP, AACONFIG, 1107, EQU 135

POSDIR specifies the position of the first directory track on a FASTRAND II. It also specifies the position of a FASTRAND III's first directory track. For a FASTRAND III, POSDIR\*3/2 is used. POS60 specifies the

## FACILITIES PARAMETERS (Cont'd)

position of the first directory track on a F8460. For all other devices, the first directory track is located in position 0.

SIZEWT SW, SECURE (MAIN), 738, +1

SIZEWT is the weight assigned to file size in tracks in the computation of the unload eligibility factor (UEF). This setting applies to LEVEL 19R1 VERS 2 of SECURE.

SSDENS CP, AACONFIG, 611, EQU 'V'

SSDENS specifies the density in flux changes per inch (FPI) of tapes assigned by the system. The codes for SSDENS are the same as those used on the @ASG control card. In this case, a density of 1600 FPI is being used.

SSEQPT CP, AACONFIG, 614, EQU '16N'

SSEQPT specifies the equipment type all tapes assigned by the system will use. In this case, a UNISERVO 16 nine-track tape is specified.

SYSPRP CP, AACONFIG, 621, EQU 4

SYSPRP defines the system prep factor for the EXEC resident on the system disk unit. SYSPRP is the number of FASTRAND (28 word) sectors that will constitute one disk record. SYSPRP should be set to either 1 or 4 if disks are configured. This will give either 28 or 112 words per record.

TFEXP CP, AACONFIG, 633, EQU 60

TFMAX CP, AACONFIG, 1327, EQU 360

TFEXP is a systems standard retention period for tape files in days. TFMAX is the maximum retention period for tape files in days. Both TFEXP and TFMAX are limited to a maximum specified value of 2047 because they are stored in 12-bit fields.



## FACILITIES MANAGEMENT (Cont'd)

UEFLUP                    SW, SECURE (SETUP), 2425, See V.B.4

UEFLUP is a look-up table for obtaining (1) the point values assigned to time since last reference (TSLR) and (2) the mean time between references (MTBR) in the unload eligibility factor (UEF) computation.

UEFTRK                   SW, SECURE (SETUP), 2430, See V.B.4

UEFTRK is a look-up table for obtaining the point values assigned to file size in tracks. It is used in the computation of the unload eligibility factor (UEF).

WHLDRM                   CP, AACONFIG, 659, EQU 1

When WHLDRM is set to 1, word addressable, unit-granular assignments will be made. WHLDRM is a conditional assembly flag that turns on code to handle unit-granular assignments.

## SCHEDULER PARAMETERS

APRIOR	CP, AACONFIG, 285, EQU 'S'
CDHOLD	SW, LOCTAB, 53, +20
CHGDED	SW, LOCTAB, 2, EQU 3
DEDLIN	CP, AACONFIG, 363, EQU 1
MAXCRD	CP, AACONFIG, 506, EQU 500
MAXCRQ	SW, LOCTAB, 1, EQU 2
MAXDEM	SW, AAPCT, 1419, EQU RSICNT
MAXOPN	CP, AACONFIG, 522, EQU 8
MAXPAG	CP, AACONFIG, 527, EQU 300
MAXTIM	CP, AACONFIG, 531, EQU 10
MAXTUX	CP, AACONFIG, 536, EQU 30
MPRIOR	CP, AACONFIG, 540, EQU 'A'
NHOLDT	CP, AACONFIG, 1016, EQU 16
RSICNT	CP, AACONFIG, 595, EQU 80
TRMXCO	CP, AACONFIG, 642, EQU 0
TRMXPO	CP, AACONFIG, 646, EQU 1
TRMXT	CP, AACONFIG, 650, EQU 1

## SCHEDULER PARAMETERS

APRIOR

CP, AACONFIG, 285, EQU 'S'

APRIOR is the default priority for @RUN statements that have no priority specified. APRIOR may be any alphabetic character selected from the set A through Z. The nearer APRIOR is to the beginning of the alphabet, the higher the default priority will be.

CDHOLD

SW, LOCTAB, 53, +20

CDHOLD specifies the maximum number of active demand runs. When the @RUN statement is received from a terminal, it is determined whether this additional demand run will cause the number of active demand runs to be greater than CDHOLD. If the number of active demand runs will be greater than CDHOLD, (1) the user is informed that a system hold exists on the number of demand runs and (2) the information about the run is released.

CHGDED

SW, LOCTAB, 2, EQU 3

CHGDED specifies in minutes when a deadline run's scheduling priority is changed to critical as the run's deadline time approaches. The run's priority is changed to one of the priorities 2 through 5. The priority is changed to 'n' when the run's deadline time is CHGDED\*n minutes away, where n can equal 2, 3, 4 or 5.

DEDLIN

CP, AACONFIG, 363, EQU 1

Allow deadline runs if DEDLIN is non-zero.

MAXCRD

CP, AACONFIG, 506, EQU 500

MAXCRD is the default for the maximum number of cards punched for batch runs, if not specified in the @RUN statement.

## SCHEDULER PARAMETERS (Cont'd)

For a demand run without a maximum number of punched cards specified, 131071 will be used. Whether the maximum is specified or not, it is limited to 262143.

MAXCRQ            SW, LOCTAB, 1, EQU 2

MAXCRQ specifies the minimum number of core queue entries that the Coarse Scheduler should provide to the Dynamic Allocator. It is defined, but apparently never used.

MAXDEM            SW, AAPCT, 1419, EQU RSICNT

MAXDEM is a conditional assembly parameter for turning on demand code when MAXDEM is greater than 0. MAXDEM is also used in determining the swap file size. RSICNT is described below.

MAXOPN            CP, AACONFIG, 522, EQU 8

MAXOPN is the maximum number of batch runs that may be open at one time.

MAXPAG            CP, AACONFIG, 527, EQU 300

If the estimated page count is unspecified in the @RUN statement, the default for batch runs is MAXPAG. The value used will be limited to 262143 ( $2^{18}-1$ ). If the P-option on the @RUN card is specified, a run is terminated when its page count is exceeded. Otherwise, the operator is notified and given the option of terminating the run.

MAXTIM            CP, AACONFIG, 531, EQU 10

If the estimated run time is unspecified in the @RUN statement, the default for batch runs is MAXTIM in minutes. If the T-option on the @RUN card is specified, a run is terminated when its estimated run time is exceeded. Otherwise, the operator is notified and given the option of terminating the run.

## SCHEDULER PARAMETERS (Cont'd)

MAXTUX	CP, AACONFIG, 536, EQU 30
	MAXTUX is the number of seconds before reaching the estimated run time at which the MAXTIM contingency will be honored.
MPRIOR	CP, AACONFIG, 540, EQU 'A'
	MPRIOR is the maximum allowed priority on a @RUN statement. If the priority specified on the @RUN card is higher than MPRIOR, MPRIOR is substituted. MPRIOR may be any alphabetic character selected from the set A through Z. The nearer MPRIOR is to the beginning of the alphabet, the higher the maximum allowed priority will be.
NHOLDT	CP, AACONFIG, 1016, EQU 16
	NHOLDT is the maximum number of attempts to open a run from the Hold Queue before the operator is notified. Runs are placed in the Hold Queue if facilities are unavailable. NHOLDT must be less than or equal to 077.
RSICNT	CP, AACONFIG, 595, EQU 80
	RSICNT is the maximum number of terminals allowed to interface simultaneously with the Remote Symbiont Interface (RSI). When a terminal signs on, a check is made to determine if the number of active terminals is greater than RSICNT. If the number of active terminals is greater, the line is terminated. RSICNT is also used as a conditional assembly flag to turn on demand code when RSICNT is greater than 0.
TRMXCO	CP, AACONFIG, 642, EQU 0
	If non-zero, TRMXCO terminates a run when the punched card estimate is exceeded. TRMXCO has no effect on demand runs. It overrules the "C" option on the @RUN card.



## SCHEDULER PARAMETERS (Cont'd)

TRMXPO

CP, AACONFIG, 646, EQU 1

If non-zero, TRMXPO terminates a run when the page estimate is exceeded. This has no effect on demand runs. It overrules the 'P' option on @RUN card.

TRMXT

CP, AACONFIG, 650, EQU 1

If non-zero, TRMXT terminates a run when the estimated run time is exceeded. This has no effect on demand runs. It overrules the 'T' option on @RUN card.

## I/O PARAMETERS

ANGLOF	SW, AAPCT, 933, EQU 1
DHLACE	CP, AACONFIG, 668, EQU 0
DHLACH	CP, AACONFIG, 671, EQU 1
DHLAC8	CP, AACONFIG, 674, EQU 0
FASTSK	CP, AACONFIG, 379, EQU 0
INCORE	CP, AACONFIG, 416, EQU 1
IOLM	CP, AACONFIG, 451, EQU 1
MAXF4	CP, AACONFIG, 510, EQU 200
MAXF8	CP, AACONFIG, 513, EQU 10000
MAXF17	CP, AACONFIG, 516, EQU 1000
MAXLLC	SW, IO, 3561, EQU 30
MAXOPR	SW, IO, 3560, EQU 12
NOSEEK	SW, IO, 80, EQU 1-STPAUL
POOLSIZE	CP, AACONFIG, 1093, EQU 25
SCGA	CP, AACONFIG, 688, EQU 0
SYMALL	SW, AASMTAGS, 3598, EQU 036

## I/O PARAMETERS

ANGLOF                   SW, AAPCT, 933, EQU 1

ANGLOF determines whether the angular addressing capability (also known as rotational position sensing, RPS) on the 8440 disk subsystem will be used. ANGLOF is a conditional assembly parameter that enables angular addressing if equated to 0. Otherwise, angular addressing is disabled when ANGLOF is set to 1. Field Change Order (FCO) number 037 must be implemented on the 5033 Control Unit before the 8440 disks may use angular addressing.

DHLACE                   CP, AACONFIG, 668, EQU 0

DHLACH                   CP, AACONFIG, 671, EQU 1

DHLAC8                   CP, AACONFIG, 674, EQU 0

DHLACE, DHLACH, and DHLAC8 are the interlace factors for 432, 1782, and 880 drums. The possible values are 1, 2, 4, 8 and 16; these values should correspond to the interlace factor at which the drum is actually set. An interlace of 1 implies that consecutive data words are stored in consecutive drum locations. If the drum interlace is n, n-1 drum locations are skipped between consecutive data words. Doubling the interlace factor halves the drum transfer rate.

FASTSK                   CP, AACONFIG, 379, EQU 0

If FASTSK equals 1, all subsequently prepped packs will have their last simulated FASTRAND position marked as allocated. That the inner cylinder remains unused results in a reduction of full pack seeks.

INCORE                   CP, AACONFIG, 416, EQU 1

INCORE specifies the method of processing a scatter read or gather write request on a disk subsystem. If INCORE equals 1, the

## I/O PARAMETERS (Cont'd)

EXEC insures that access control words (ACW's) fall on record boundaries. If ACW's do not all fall on record boundaries, the code generated by INCORE will build a new set of ACW's that do fall on record boundaries.

IOLM CP, AACONFIG, 451, EQU 1

IOLM is a conditional assembly parameter that controls the I/O Load Monitor code generation. I/O Load Monitor adjusts the I/O load on memory modules that may be overloaded by I/O requests. If a memory module's maximum I/O transfer rate is less than the combined transfer rate of the subsystems active on the memory module, a late acknowledgement (data overrun) can occur.

MAXF4 CP, AACONFIG, 510, EQU 200

MAXF8 CP, AACONFIG, 513, EQU 10000

MAXF17 CP, AACONFIG, 516, EQU 1000

MAXF17 specifies the number of FH-432 (MAXF4), FH-880 (MAXF8), and FH-1782 (MAXF17) tracks available to the symbiont complex for writing. This FASTRAND-formatted mass storage allocation for user symbiont files is accomplished by a quota system based on the number of user symbiont files. A file will obtain a certain track allocation quota on each type of drum. The quotas will be used by a file in the order (1) FH-432, (2) FH-1782, and (3) FH-880. If the quotas on these three drums are already used, the remaining amount of the user symbiont file will be allocated on the device specified by the parameter SYMALL.

MAXLLC SW, IO, 3561, EQU 30

MAXLLC is the maximum number of times a queued I/O request may be locked out by on-position requests.

## I/O PARAMETERS (Cont'd)

MAXOPR SW, IO, 3560, EQU 12

MAXOPR is the maximum number of consecutive I/O requests that will be satisfied for a disk unit's current position before the position will be forcibly changed.

NOSEEK SW, IO, 80, EQU 1-STPAUL

NOSEEK is a conditional assembly parameter that enables 8440 disk unit pre-seeking. NOSEEK equated to 1 (STPAUL = 0) turns 8440 pre-seeking off and NOSEEK equated to 0 (STPAUL = 1) enables 8440 pre-seeking. Field Change Order (FCO) Number 037 must be implemented on the disk controller before 8440 pre-seeking is enabled by setting NOSEEK to 0.

POOLSIZE CP, AACONFIG, 1093, EQU 25

POOLSIZE specifies the number of 15-word I/O queue item buffers (IQ items) maintained within the EXEC element IO. Each IQ item represents an I/O request that is queued. If POOLSIZE equals 0, no pooling of IQ items will occur and EXPOOL buffers will be used instead. Use of I/O queue buffers within the element IO reduces EXPOOL overhead, but buffers will be assigned to EXPOOL if required.

SCGA CP, AACONFIG, 688, EQU 0

SCGA specifies whether scatter read and gather write I/O will be used by the EXEC symbiont element SYMBIO. SCGA greater than zero will enable the use of scatter/gather I/O by SYMBIO. No scatter/gather I/O will be done if SCGA equals 0.

SYMALL SW, AASMTAGS, 3598, EQU 036

SYMALL specifies the equipment code of the mass storage device used for user symbiont file allocation after the file's quotas on



I/O PARAMETER (Cont'd)

FH-432, FH-1782, and FH-880 are used. The file's quotas for FH-432, FH-1782, and FH-880 are specified by the parameters MAXF4, MAXF17, and MAXF8, respectively. In this case, 036 specifies that the FASTRAND-formatted 8440 disk subsystem will be used.

## TEST AND DEBUG PARAMETERS

DADBUG	SW, AADAEQT, 3, EQU STPAUL
DATRAC	SW, AADAEQT, 9, EQU 30*UNIVAC
EXPTRACE	CP, AACONFIG, 896, EQU 130
FCDBSZ	CP, AACONFIG, 922, EQU 50
FNCCHK	CP, AACONFIG, 401, EQU 0
IODEBUG	CP, AACONFIG, 972, EQU 1
SBLEV1	CP, AACONFIG, 1199, EQU 1
SBLEV2	CP, AACONFIG, 1206, EQU 1
SBLEV3	CP, AACONFIG, 1210, EQU 1
SBLEV4	CP, AACONFIG, 1214, EQU 1
SBLEV5	CP, AACONFIG, 1218, EQU 1
SIPDAT	CP, AACONFIG, 1222, EQU 1
SIPIO	CP, AACONFIG, 1229, EQU 1
STPAUL	CP, AACONFIG, 280, EQU 0
SWPCKO	SW, AADAEQT, 5, EQU UNIVAC+2*STPAUL
UNIVAC	CP, AACONFIG, 654, EQU 1

## TEST AND DEBUG PARAMETERS

DADBUG	SW, AADAEQT, 3, EQU STPAUL
	DADBUG specifies whether additional Dynamic Allocator internal error detection is done. When DADBUG is equated to 1, error detection occurs.
DATRAC	SW, AADAEQT, 9, EQU 30*UNIVAC
	DATRAC specifies the size of the swaplock trace area. It must be an even number whose value is $2^n - 2$ , where n is any positive integer.
EXPTRACE	CP, AACONFIG, 896, EQU 130
	EXPTRACE is the number of EXPPOOL requests and release trace entries. Each entry consists of two words.
FCDBSZ	CP, AACONFIG, 922, EQU 130
	FCDBSZ specifies the number of FNCCNT actions traced. Each action is traced by a five-word trace entry.
FNCCHK	CP, AACONFIG, 401, EQU 0
	FNCCHK determines whether function validation is performed whenever a nonresident function is read from drum. If FNCCHK equals 0, no checking is done. The function's keyword is checked when FNCCHK is 1; a checksum is computed when FNCCHK equals 2.
IODBUG	CP, AACONFIG, 972, EQU 1
	When IODBUG equals 1, all debug aids within the I/O complex are enabled.
SBLEV1	CP, AACONFIG, 1199, EQU 1
SBLEV2	CP, AACONFIG, 1206, EQU 1
SBLEV3	CP, AACONFIG, 1210, EQU 1
SBLEV4	CP, AACONFIG, 1214, EQU 1

TEST AND DEBUG PARAMETER (Cont'd)

SBLEV5 CP, AACONFIG, 1218, EQU 1

These parameters turn on and off the various levels of SIP. The SIP levels are discussed in detail in the appendix on SIP.

SIPDAT CP, AACONFIG, 1222, EQU 1

The SIP Dynamic Allocator Trace is enabled and written to the SIP SYSBAL\$LOG\$ file when SIPDAT equals 1.

SIPIO CP, AACONFIG, 1229, EQU 1

The SIP I/O Trace is enabled when SIPIO is equated to 1.

STPAUL CP, AACONFIG, 280, EQU 0

Extensive internal EXEC error checking is done. STPAUL is set to 0 for all sites that are not STPAUL sites.

SWPCKO SW, AADAEQT, 5, EQU UNIVAC+2\*STPAUL

When SWPCKO is non-zero, program bank swap checksums are calculated. Two words per block are summed if SWPCKO equals 1; and, sixteen words per block are summed if SWPCKO is 2. If SWPCKO is 3 or more, every word in the bank is used in calculating the checksum.

UNIVAC CP, AACONFIG, 654, EQU 1

When UNIVAC is non-zero, the primary EXEC error checking is enabled.

APPENDIX C  
INTRODUCTION TO SIP





SIP is an event-driven software monitor. That is, SIP counts and times the occurrence of significant events within the operating system. The data collection code is integrated into the operating system. Special instructions are included in the executive programs to count the frequency of certain events or to time the duration of certain activities. These instructions are preceded by conditional assembly flags. The conditional assembly flags are turned on or off by system generation parameters. When the flags are off, the SIP data collection code is not generated as part of the operating system. Therefore, if SIP is not going to be used, the SIP overhead may be avoided.

SIP data collection is hierarchically defined. Five levels of detail may be specified at system generation time. The lowest level generates the least overhead and collects only a basic set of data. Each higher level of SIP includes an additional degree of detail. The levels are divided into two classes: user levels and development levels. User levels collect basic data of general interest; development levels collect detailed information needed for operating system tuning. There are two user levels: User-A and User-B; and three development levels: Development-A, Development-B, and Development-C. Each level has a separate conditional assembly parameter associated with it. These parameters determine which SIP code is or is not assembled when the operating system is generated. The data collected by each level are summarized in Table C-1. Each item in Table C-1 corresponds to a standard report.

Three programs support the SIP data collection code: SYSBAL, SYSSIP, and SIPDRP. SYSBAL contains the storage areas needed to hold the data collected by SIP. SYSBAL also contains numerous subroutines that support the data collection process. SYSBAL must be permanently core-resident when SIP is on. SYSSIP is a nonresident element that provides the interface among the user, the operator, and the system. SYSSIP is capable of responding to a large number of operator key-ins. SIPDRP is the SIP Data Reduction Program; it summarizes the SIP data and produces a set of standard reports. SIPDRP is a FORTRAN program that may be run whenever it is convenient.

The operator key-ins provide a substantial amount of flexibility in controlling the operation of SIP. Of particular interest is the SB CODES S1S2S3S4S5S6 key-in (all SIP operator key-ins begin with SB). The codes' key-in may be

SIP REPORT	SIP LEVELS				
	1	2	3	4	5
System Balance Summary	1				
Memory Utilization		2			
Exec Activity for Each CPU		2			
Dist. of Prog. Size for Run Type		2			
Processor Activity for Each CPU	1				
Idle Time by SIP Collection		2			
I/O Activity for Each CPU	1				
Mass Storage SS Queuing by Type	1				
Mass Storage Analysis by Unit	1				
I/O Queuing by SS	1				
Pre-Seeking and Pre-Positioning	1				
SWAP\$FILE Expansions	1				
Facilities Hold Queue Statistics	1				
I/O Load Monitor Activity	1				
Common Bank Usage		2			
CTMC Comm. Line Usage	1				
EXPOOL Requests and Releases			3		
EXPOOL Counts by Priority				4	
Times EXPOOL Became Tight	1				
EXPOOL Usage	1				
Register Saves and Restores	1				
Segment Requests and Loads			3		
Segment Initial/Active Requests				4	
System Idle Activity	1				
ER Counts	1				
Response Times (Abbreviated)		2			
Response Times (Complete)			3		
Core Fragmentation			3		
Suspend/Swap Statistics			3		
Exec Frequency Counts			3		
Test and Set Summary					5
SIP Overhead	1				

# SIP REPORTS (LEVEL 4.0)

TABLE C-1

used selectively to turn on (SI = 1) or off (SI = 0) six areas of SIP data collection:

- S1 - User level A data
- S2 - Distribution of user program sizes
- S3 - Memory-in-use distributions
- S4 - EXPOOL statistics
- S5 - Function/Segment Activity
- S6 - Most of the rest

The status of each of these codes is reported on the first page of the SIP output. Within the operating system, the SIP data collection code is still present; however, if the appropriate status code is set, a branch instruction bypasses the SIP instructions.

The status codes are very helpful in reducing the amount of overhead produced by SIP. Since SIP is event-driven, SIP overhead is proportional to the overall activity of the system. The overhead generated by SIP is displayed on the last page of the SIP output. The overhead associated with program size calculations is also reported. Since program size calculations often account for 50% or more of the total SIP overhead, it is usually a good idea to turn off the S2 status code. Another way to minimize the amount of SIP overhead is to run SIP periodically for short periods rather than continuously. Experience has shown that 10- or 15-minute periods are long enough to observe steady state behavior. The starting and stopping of SIP are easily controlled with operator key-ins.

The remainder of this appendix is a set of annotated SIP reports. Special comments have been typed on these reports to aid in their interpretation. The reports are in the same sequence that they were printed by SIP. Detailed information about SIP may be found in the SIP documentation manual.

SIP EXEC 8 32R2B:P3-5 SYSTEM BALANCE SUMMARY

SIP RELEASE 4.0

SYSEAL\$LOG\$ IS ON TAPE AND THIS IS BLOCK NUMBER 15

SEPTER 19, 1977

SYSEAL WENT OFF HERE FOR THE 1 TIME

THIS IS A COMPLETE SUMMARY AND THERE WAS A TOTAL OF 15 BLOCKS OF OUTPUT

AMOUNT OF CORE IS 262K

SIZE OF EXEC 8 IS 94K

SIP IS IN THE 'AUTO' MODE WITH A MINIMUM OF 1 RUNS ACTIVE

THE SIP CODE STATUS AT OUTPUT TIME IS:

S1 - ON	SIP USER LEVEL 1
S2 - ON	USER PROGRAM SIZE
S3 - ON	TOTAL CORE USAGE
S4 - ON	EXP00L STATISTICS
S5 - ON	FUNCTION COUNTS
S6 - ON	MOST OF THE REST

TIME ON 14:50:37 TIME OFF 14:59:37

TOTAL ELAPSED TIME 0:08:59.958 ( 539.958 SECONDS)



DISTRIBUTION OF MEMORY UTILIZATION  
( PERCENT OF MEMORY IN USE SEEN FROM THE BCP CHAIN )

	TALLY	TIME ( SEC )
32 BLØCKS	0%	.000
64 BLØCKS	0%	.000
96 BLØCKS	0%	.000
128 BLØCKS	0%	.000
160 BLØCKS	0%	.000
192 BLØCKS	0%	.000
224 BLØCKS	0%	.000
256 BLØCKS	0%	.000
288 BLØCKS	0%	.000
320 BLØCKS	0%	.000
352 BLØCKS	0%	.000
384 BLØCKS	0%	.000
416 BLØCKS	0%	.759
448 BLØCKS	2%	9.680
480 BLØCKS	32%	170.676
512 BLØCKS	66%	358.843
-----	-----	-----
TOTAL	1243	539.959

Always in use by EXEC

AMOUNT OF TOTAL MEMORY IN USE ON THE AVERAGE IS 249K 95%

TALLY: TALLY COUNT OF OCCURRENCES OF THIS RANGE OF UTILIZATION  
 TIME: RUNNING SUM OF TIME THAT THE MEMORY UTILIZATION WAS WITHIN THIS RANGE  
 BLØCKS: NUMBER OF 512-WORD GROUPS  
 K: UNIVAC STANDARD OF 1000 DECIMAL  
 AVE USE: SUM OF MULTIPLYING EACH 16K INTERVAL BY ITS PERCENT

EXECUTIVE ACTIVITY FOR CPU 0

EXEC TYPE	TIME	% OF TOTAL TIME
EXEC0	85.490	16%
EXEC1	.504	0%
EXEC3	29.485	5%

Most EXEC 0 time is for I/O handling.

EXEC 3 is for EXEC support functions; the DA is a big EXEC 3 user. It is common in a demand environment to see EXEC time equal 50% of the CPU active time.

DISTRIBUTION OF USER PROGRAM SIZES FOR CPU 0

PROGRAM TYPE = ESI

INTERVAL SIZE	TALLY COUNT	TIME	% RESIDENT
8K	0	.000	0
16K	0	.000	0
24K	0	.000	0
32K	0	.000	0
40K	0	.000	0
49K	0	.000	0
57K	0	.000	0
65K	0	.000	0
98K	0	.000	0
131K	0	.000	0
196K	0	.000	0
>196K	477	.098	0

ACTIVE TIME FOR ESI IS:

.098

% OF TOTAL TIME IS: 0

PROGRAM TYPE = RTIME

INTERVAL SIZE	TALLY COUNT	TIME	% RESIDENT
8K	0	.000	0
16K	0	.000	0
24K	0	.000	0
32K	0	.000	0
40K	0	.000	0
49K	0	.000	0
57K	0	.000	0
65K	0	.000	0
98K	0	.000	0
131K	0	.000	0
196K	0	.000	0
>196K	9	.001	100

ACTIVE TIME FOR RTIME IS:

.001

% OF TOTAL TIME IS: 0

PROGRAM TYPE = DEMAND

User Level - B

A

INTERVAL SIZE

B

TALLY COUNT

C

TIME

D

% RESIDENT

A: The size of a program includes all banks (except common banks and PCT banks) whether or not the banks are resident.  
B: Tally counts the number of times execution was switched to a program whose size lies in the specified interval.

C: Time is the amount of time spent in programs whose size lies within the specified interval.  
D: % Resident is the average percentage of the total program size that is resident for programs whose total size lies within this interval.

ACTIVE TIME FOR DEMAND IS:

38.708

% OF TOTAL TIME IS: 7

PROGRAM TYPE = DBATCH

INTERVAL SIZE TALLY COUNT TIME % RESIDENT

8K	0	.000	0
16K	0	.000	0
24K	195	.028	100
32K	0	.000	0
40K	0	.000	0
49K	0	.000	0
57K	0	.000	0
65K	0	.000	0
98K	0	.000	0
131K	0	.000	0
196K	0	.000	0
>196K	0	.000	0

ACTIVE TIME FOR DBATCH IS:

.028

% OF TOTAL TIME IS: 0

PROGRAM TYPE = BATCH

INTERVAL SIZE TALLY COUNT TIME % RESIDENT

8K	163	.506	1
16K	31	.006	72
24K	16758	86.775	94
32K	0	.000	0
40K	0	.000	0
49K	0	.000	0
57K	0	.000	0
65K	0	.000	0
98K	0	.000	0
131K	0	.000	0
196K	0	.000	0
>196K	0	.000	0

ACTIVE TIME FOR BATCH IS:

87.288

% OF TOTAL TIME IS: 16

User Level - A

PROCESSOR ACTIVITY ON CPU 0

TYPE OF INTERRUPT

TALLY

INT/SEC

ESI INTERPROCESSOR	0	.0
ISI INPUT MONITOR	1353	2.5
ISI OUTPUT MONITOR	625	1.2
ISI FUNCTION MONITOR	0	.0
ISI EXTERNAL	19002	35.2
ISI INPUT MONITOR (S/G)	2437	4.5
ISI OUTPUT MONITOR (S/G)	2565	4.8
ESI INPUT MONITOR	0	.0
ESI OUTPUT MONITOR	0	.0
ESI EXTERNAL	0	.0
DAY CLOCK	89	.2
REAL TIME CLOCK	21815	40.4
INTERPROCESSOR	45	.1
ILLEGAL INSTRUCTION	1	.0
GUARD MODE	0	.0
FLOATING POINT OVERFLOW	0	.0
FLOATING POINT UNDERFLOW	0	.0
DIVIDE FAULT	0	.0
EXEC TEST AND SETS	11104	20.6
USER TEST AND SETS	0	.0

Should correlate with EXEC 0 time.

Since day clock interrupts occur every 6 seconds, this value should equal the number of elapsed SIP seconds divided by 6.

These are broken out in a later report.

PROCESSOR WENT IDLE	45775 TIMES	
LEAPED IN IDLE ACTIVITY	2807379 TIMES	
IDLE TIME	298.356 SECONDS BY SIP COLLECTION	
IDLE TIME	299.279 SECONDS BY DRP CALCULATION	
IDLE PERCENTAGE	55% BY SIP COLLECTION	
IDLE PERCENTAGE	55% BY DRP CALCULATION	
SYS IDLE LOOP CT	2540045	CPU IDLE LOOP CT 267334
ABNORMAL ENTRIES TO SYS IDLE	10112	
RI OUTPUT ACTIVITY TIME -	.001 SECONDS	

Actual idle time as measured by SIP.  
Estimated idle time based on AIET. These numbers should not differ by more than 2 or 3 points.

INPUT	(A)		(B)		(C)		(D)		(E)		(F)		(G)	
	TIME (SEC)	%	WORDS	ACW	AVRDS	AVAXST(MS)	0	1	2	3	0	1	2	3
CHANNEL 0 FH1782	97.825	18	7050792	2551	2763	26	71%	18%	6%	3%	71%	18%	6%	3%
CHANNEL 1 DI8414	34.058	6	712205	1583	449	13	92%	7%	0%	0%	92%	7%	0%	0%
CHANNEL 2 DI8460	221.559	41	1662499	3451	481	57	62%	33%	4%	0%	62%	33%	4%	0%
CHANNEL 3 TAPDR	.000	0	0	0	0	0	100%	0%	0%	0%	100%	0%	0%	0%
CHANNEL 4 ZILTC	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 5 ZILTC	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 6 ZILTC	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 7 ZILTC	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 8 FH1782	26.781	5	1386557	904	1533	24	58%	20%	11%	5%	58%	20%	11%	5%
CHANNEL 9 DI8414	3.678	1	80514	173	465	15	81%	18%	0%	0%	81%	18%	0%	0%
CHANNEL 10 DI8460	165.493	31	1595886	3273	487	44	43%	40%	13%	1%	43%	40%	13%	1%
CHANNEL 11 TAPDR	.117	0	1864	2	932	58	96%	3%	0%	0%	96%	3%	0%	0%
CHANNEL 12 ZILTC	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 13 CTMCW	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 14 AX9200	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 15 C8NS0L	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%

OUTPUT	(A)		(B)		(C)		(D)		(E)		(F)		(G)	
	TIME (SEC)	%	WORDS	ACW	AVRDS	AVAXST(MS)	0	1	2	3	0	1	2	3
CHANNEL 0 FH1782	42.005	8	3659477	948	3860	28	71%	18%	6%	3%	71%	18%	6%	3%
CHANNEL 1 DI8414	9.492	2	184800	395	467	16	92%	7%	0%	0%	92%	7%	0%	0%
CHANNEL 2 DI8460	10.715	2	84906	156	544	62	62%	33%	4%	0%	62%	33%	4%	0%
CHANNEL 3 TAPDR	.056	0	2150	1	2150	55	100%	0%	0%	0%	100%	0%	0%	0%
CHANNEL 4 ZILTC	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 5 ZILTC	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 6 ZILTC	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 7 ZILTC	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 8 FH1782	10.513	2	673259	256	2629	33	58%	20%	11%	5%	58%	20%	11%	5%
CHANNEL 9 DI8414	.628	0	13832	22	628	28	81%	18%	0%	0%	81%	18%	0%	0%
CHANNEL 10 DI8460	9.057	2	122005	138	884	58	43%	40%	13%	1%	43%	40%	13%	1%
CHANNEL 11 TAPDR	1.601	0	37362	128	291	4	96%	3%	0%	0%	96%	3%	0%	0%
CHANNEL 12 ZILTC	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 13 CTMCW	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 14 AX9200	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%
CHANNEL 15 C8NS0L	.000	0	0	0	0	0	0%	0%	0%	0%	0%	0%	0%	0%

NOTE: THE Q % IS FOR TOTAL I/O (INPUT AND OUTPUT COMBINED PER CHANNEL).  
 THE CHANNEL ACTIVE TIMES FOR MULTIPLE HIGH SPEEDPRINTER CHANNELS ARE ONLY THE TIMES REQUIRED TO XFFR  
 THE PRINT BUFFER TO THE CONTROL UNIT AND DO NOT INCLUDE TIME REQUIRED TO PRINT THE LINE.

A: Total channel active time separated between input and output.

B: Percent of total time that SIP was active.

C: Total number of words transferred by channel.

D: Number of ACW's used to transfer (C) words.

E: Average number of words transferred per ACW.

F: Time per I/O in milliseconds excluding the actual data transfer time.

G: Percent of busy time the queue depth was 0, 1, 2, or 3. Input and output are combined for each channel.

$$E = C/D$$

$$F = (A - \frac{C}{R}) * \frac{1000}{D} \text{ where } R \text{ is the transfer rate in words/second for the device.}$$



## EXECUTIVE ACTIVITY FOR CPU 1

EXEC TYPE	TIME	% OF TOTAL TIME
EXEC0	91.006	17%
EXEC1	.506	0%
EXEC3	33.043	6%

## DISTRIBUTION OF USER PROGRAM SIZES FOR CPU 1

PROGRAM TYPE = ESI

INTERVAL SIZE	TALLY COUNT	TIME	% RESIDENT
8K	0	.000	0
16K	0	.000	0
24K	0	.000	0
32K	0	.000	0
40K	0	.000	0
49K	0	.000	0
57K	0	.000	0
65K	0	.000	0
98K	0	.000	0
131K	0	.000	0
196K	0	.000	0
>196K	5838	1.226	0

ACTIVE TIME FOR ESI IS:

% OF TOTAL TIME IS: 0

PROGRAM TYPE = RTIME

INTERVAL SIZE	TALLY COUNT	TIME	% RESIDENT
8K	0	.000	0
16K	0	.000	0
24K	0	.000	0
32K	0	.000	0
40K	0	.000	0
49K	0	.000	0
57K	0	.000	0
65K	0	.000	0
98K	0	.000	0
131K	0	.000	0
196K	0	.000	0
>196K	20	.003	100

ACTIVE TIME FOR RTIME IS:

.003

% OF TOTAL TIME IS: 0

PROGRAM TYPE = DEMAND

INTERVAL SIZE	TALLY COUNT	TIME	% RESIDENT
8K	2064	2.199	86
16K	116	.193	97
24K	13722	26.435	93
32K	372	3.330	98
40K	724	1.680	97
49K	205	.568	95
57K	450	1.611	93
65K	0	.000	0
98K	0	.000	0
131K	0	.000	0
196K	0	.000	0
>196K	0	.000	0
ACTIVE TIME FOR DEMAND IS:			% OF TOTAL TIME IS: 7
PROGRAM TYPE = DEATCH			
			36.015

INTERVAL SIZE	TALLY COUNT	TIME	% RESIDENT
8K	0	.000	0
16K	0	.000	0
24K	228	.036	100
32K	0	.000	0
40K	0	.000	0
49K	0	.000	0
57K	0	.000	0
65K	0	.000	0
98K	0	.000	0
131K	0	.000	0
196K	0	.000	0
>196K	0	.000	0
ACTIVE TIME FOR DBATCH IS:			% OF TOTAL TIME IS: 0
PROGRAM TYPE = BATCH			
			.036

INTERVAL SIZE	TALLY COUNT	TIME	% RESIDENT
8K	174	.586	7
16K	8	.001	72
24K	16779	79.879	94
32K	0	.000	0
40K	0	.000	0
49K	0	.000	0
57K	0	.000	0
65K	0	.000	0
98K	0	.000	0
131K	0	.000	0
196K	0	.000	0
>196K	0	.000	0
ACTIVE TIME FOR BATCH IS:			% OF TOTAL TIME IS: 15
			80.466

# PROCESSOR ACTIVITY ON CPU 1

TYPE OF INTERRUPT	TALLY	INT/SEC
ESI INTERPROCESSOR	3646	6.8
ISI INPUT MONITOR	1162	2.2
ISI OUTPUT MONITOR	458	.8
ISI FUNCTION MONITOR	0	.0
ISI EXTERNAL	19126	35.4
ISI INPUT MONITOR (S/G)	1472	2.7
ISI OUTPUT MONITOR (S/G)	1812	3.4
ESI INPUT MONITOR	1350	2.5
ESI OUTPUT MONITOR	4053	7.5
ESI EXTERNAL	2468	4.6
DAY CLOCK	0	.0
REAL TIME CLOCK	18848	34.9
INTERPROCESSOR	192	.4
ILLEGAL INSTRUCTION	2	.0
GUARD MODE	0	.0
FLOATING POINT OVERFLOW	0	.0
FLOATING POINT UNDERFLOW	0	.0
DIVIDE FAULT	0	.0
EXEC TEST AND SETS	10095	18.7
USER TEST AND SETS	0	.0

PROCESSOR WENT IDLE	45211 TIMES
LOOPEd IN IDLE ACTIVITY	2555395 TIMES
IDLE TIME	297.638 SECONDS BY SIP COLLECTION
IDLE TIME	291.599 SECONDS BY DRP CALCULATION
IDLE PERCENTAGE	55% BY SIP COLLECTION
IDLE PERCENTAGE	54% BY DRP CALCULATION

SYS IDLE LOOP CT	2284487	CPU IDLE LOOP CT	270908
ABNORMAL ENTRIES TO SYS IDLE	7241		
RT OUTPUT ACTIVITY TIME =	.000 SECONDS		

# INPUT-OUTPUT ACTIVITY ON CPU 1

INPUT	TIME (SEC)	%	WORDS	ACW	AVRDS	AVAXST(MS)	0	1	2	3
CHANNEL 0	85.998	16	6255156	2361	2649	25	76%	14%	5%	3%
CHANNEL 1	30.639	6	629921	1433	439	13	91%	8%	0%	0%
CHANNEL 2	218.717	41	1583585	3328	475	59	63%	27%	8%	1%
CHANNEL 3	.000	0	0	0	0	0	100%	0%	0%	0%
CHANNEL 4	.000	0	0	0	0	0	0%	0%	0%	0%
CHANNEL 5	.000	0	0	0	0	0	0%	0%	0%	0%
CHANNEL 6	.000	0	0	0	0	0	0%	0%	0%	0%
CHANNEL 7	.000	0	0	0	0	0	0%	0%	0%	0%
CHANNEL 8	27.790	5	1351776	954	1416	23	52%	27%	12%	7%
CHANNEL 9	3.481	1	69790	160	436	15	80%	17%	1%	0%
CHANNEL 10	176.587	33	1666684	3461	481	44	41%	42%	13%	2%
CHANNEL 11	.048	0	1800	1	1800	48	100%	0%	0%	0%
CHANNEL 12	.000	0	0	0	0	0	0%	0%	0%	0%
CHANNEL 13	.000	0	0	0	0	0	0%	0%	0%	0%
CHANNEL 14	.182	0	392	28	14	0	100%	0%	0%	0%
CHANNEL 15	.000	0	0	0	0	0	0%	0%	0%	0%

OUTPUT	TIME (SEC)	%	WORDS	ACW	AVRDS	AVAXST(MS)	0	1	2	3
CHANNEL 0	39.248	7	3098128	893	3469	30	76%	14%	5%	3%
CHANNEL 1	8.837	2	171472	360	476	16	91%	8%	0%	0%
CHANNEL 2	15.294	3	141736	205	691	69	63%	27%	8%	1%
CHANNEL 3	.056	0	2150	1	2150	56	100%	0%	0%	0%
CHANNEL 4	.000	0	0	0	0	0	0%	0%	0%	0%
CHANNEL 5	.000	0	0	0	0	0	0%	0%	0%	0%
CHANNEL 6	.000	0	0	0	0	0	0%	0%	0%	0%
CHANNEL 7	.000	0	0	0	0	0	0%	0%	0%	0%
CHANNEL 8	8.455	2	588293	240	2451	26	52%	27%	12%	7%
CHANNEL 9	.804	0	15904	31	513	25	80%	17%	1%	0%
CHANNEL 10	10.057	2	111544	138	808	65	41%	42%	13%	2%
CHANNEL 11	1.715	0	45638	115	396	6	100%	0%	0%	0%
CHANNEL 12	.000	0	0	0	0	0	0%	0%	0%	0%
CHANNEL 13	.000	0	0	0	0	0	0%	0%	0%	0%
CHANNEL 14	.473	0	1320	55	24	0	100%	0%	0%	0%
CHANNEL 15	.000	0	0	0	0	0	0%	0%	0%	0%

NOTE: THE Q % IS FOR TOTAL I/O (INPUT AND OUTPUT COMBINED PER CHANNEL),  
 THE CHANNEL ACTIVE TIMES FOR MULTIPLE HIGH SPEED/PRINTER CHANNELS ARE ONLY THE TIMES REQUIRED TO XFER  
 THE PRINT BUFFER TO THE CONTROL UNIT AND DO NOT INCLUDE TIME REQUIRED TO PRINT THE LINE.

SUBSYSTEM	MASS STORAGE SUBSYSTEM QUEUEING BY TYPE										
	SS #	RT	QUEUE	EXISTS	AVE RT Q	EXEC Q	EXISTS	AVE EXEC Q	USER Q	EXISTS	AVE USER Q
1782DRUM	2	0			(A) .00	(A) 3567	(B) 1.72	(B) 1.00	554		1.12
8414 DISC	4	0			.00	77	1.00	1.00	844		1.02
8460 DISC	7	0			.00	2643	1.03	1.03	14968		1.46
U16 TAPE	8	0			.00	0	.00	.00	1		1.00

MASS STORAGE ANALYSIS BY UNIT

SUBSYSTEM	SS #	UNIT	TOTAL REFERENCES	REF/SEC	TIMES QUEUED	% REQ Q'D	TIMES ASSIGNED	TOTAL ASG	TIME
1782DRUM	2	0	4130	(C) 7.6	987	(D) 23			
1782DRUM	2	1	1734	3.2	598	34			
1782DRUM	2	2	3243	6.0	1386	42			
8414 DISC	4	C	576	1.8	72	7	0		.00
8414 DISC	4	1	0	.0	0	0	0		.00
8414 DISC	4	2	596	1.1	33	5	0		.00
8414 DISC	4	3	0	.0	0	0	0		.00
8414 DISC	4	4	0	.0	0	0	0		.00
8414 DISC	4	5	1449	2.7	136	9	0		.00
8414 DISC	4	6	1043	1.9	549	52	0		.00
8414 DISC	4	7	93	.2	0	0	0		.00
8460 DISC	7	0	6255	11.6	3552	56			
8460 DISC	7	1	1587	2.9	1048	66			
8460 DISC	7	2	5598	11.1	2629	43			
8460 DISC	7	3	310	.6	173	55			
U16 TAPE	8	C	14	.0	0	0	0		.00
U16 TAPE	8	1	3	.0	0	0	1		131.93
U16 TAPE	8	2	0	.0	0	0	0		.00
U16 TAPE	8	3	0	.0	0	0	0		.00
U16 TAPE	8	4	0	.0	0	0	0		.00
U16 TAPE	8	5	0	.0	0	0	0		.00
U16 TAPE	8	6	231	.4	0	0	1		127.34

Blank fields indicate non-removable devices.

[Blank fields indicate non-removable devices.]

Note: An EXEC queue exists more often than a user queue on the 1782 which is to be expected since the 1782 contains the system files. Just the reverse is true for the 8460 and the 8414.

- A: The number of times that an EXEC request was queued for this device when an I/O request was completed. This is not a count of the total number of requests queued.
- B: The average number of requests in the queue when a queue existed.
- C: Total references divided by the SIP active time.
- D: Times queued by total references.



User Level - A

This report specifies the reasons that requests were queued.

SUMMARY OF IO QUEUEING BY SUBSYSTEM

SUBSYSTEM	SS #	SS BUSY	SS BUSY UNIT	SS FREE
1782DRUM	2	(A) 442	(B) 996	(C) 1533
8414 DISC	4	16	31	743
8460 DISC	7	3359	3053	990

PRE-SEEKING AND PRE-POSITIONING ON MOVABLE HEAD MASS STORAGE

Applies to disk

Applies to FASTRAND

# OF SEKS OR PRE-POSITIONS AVE SOFTWARE FAILURES TO FIND REQS NO HEAD POSITION OR SEEK MAX HIT FOR ON-POSITIONS MOVED REQUEST FOR UNIT MOVEMENT NEEDED ONLY REQUESTS POSITION REQUESTS

SUBSYSTEM	(D)	(E)	(F)	(G)	(H)	(I)
3 DI8414	115	33	3371	106	115	4
4 DI8414	1320	32	28350	784	1320	61
6 DI8460	1620	88	23889	3705	1620	277
7 DI8460	3526	107	18559	1733	3526	118
9 TAPU16	0	0	0	0	10	0

THERE WERE

0 SWAP\$FILE EXPANSIONS

THE MAXIMUM SWAP\$FILE ASSIGNED DURING MEASUREMENT WAS

831 TRACKS

The total number of requests queued for a device (A+B+C) should equal the total requests queued from the Mass Storage Analysis Report.

A: All SS's are busy with some other unit. If this number is high for a single access controller, a dual access controller may be needed. If it is high on a dual access controller, more channels may be needed.

B: No free controller, and the desired unit is busy.

C: There is a SS free, but the desired unit is busy. High numbers may indicate poor file placement.

Note: Numbers are only "high" when compared to the total number queued to that device.

Note: SIP does not report on the condition where both SS and unit are free since no queuing occurs in that situation.

A request may be naturally on-position or the EXEC may do a pre-peek to get it on position.

D: The number of requests for which a pre-peek was necessary. "n" should approximately equal "H".

E: Positions moved per seek operation.

F: This count is incremented if there is no request in the queue for the next unit in the cycle.

G: The number of requests that were naturally on position.

D+G: Total number of requests.

H: The number of seek only requests; see "D".

I: The number of times MAXOPR requests were available for consecutive servicing. MAXOPR specifies the maximum number of on-position requests that may be serviced consecutively before requests for other positions are serviced. The following relation must hold:

$$F * MAXOPR \leq D + G$$

User Level - A

FACILITIES HOLD QUEUE STATISTICS

TYPE OF HOLD		OCCURRENCES	TIME SPENT IN HOLD CONDITION		%	Percent of total time that SIP was active.
			Total time in seconds			
PUT IN A TAPE HOLD BATCH DEMAND		0 0 0	.000		0	
PUT IN A MASS S HOLD BATCH DEMAND		1 1 0	38.269		7	
(A) IN 'CS H' MODE		1	134.258		25	
(B) IN 'CS H RUNID' MODE		0	.000		0	
(C) IN 'CS HD' MODE		0	.000		0	
(D) IN 'CS HT' MODE		0	.000		0	
(E) CATALOGUE FILE HOLD ON		1	38.259		7	
<div><div>A: Coarse Scheduler Master Hold, all demand and batch runs are held. May be done by the EXEC or by the operator.</div><div>B: Coarse Scheduler Hold on a specific RUNID.</div><div>C: Coarse Scheduler Hold on demand runs.</div></div>						
<div><div>D: Coarse Scheduler Hold on demand terminals.</div><div>E: Runs held because of unavailable cataloged files.</div><div>F: The amount of time in seconds that the number of runs in the ranges were being held because of inavailability of high speed mass storage. The value does not include hold times due to cataloged files.</div></div>						
# OF RUNS BEING HELD:			1 - 2	3 - 4	5 - 8	9 & UP
HOLD TIME/EQUIP GROUP:						
ANY 7-TRACK TAPES			.000		.000	.000
ANY 9-TRACK TAPES			.000		.000	.000
HIGH SPEED MASS STORAGE			.000		.000	.000
MED HIGH SPEED MASS S			.000		.000	.000
MED LOW SPEED MASS S			.000		.000	.000
LOW SPEED MASS STORAGE			.000		.000	.000

\*\*NOTE\*\*

HIGH SPEED = U.C.S. & F432  
MED LOW = 8414, 8424/8425 DISC  
MED HIGH = F880, F1782, 8440 DISC  
LOW SPEED = F2/F3/8460 DISC.  
ALL TIME IS IN SECONDS.

User Level - A

I/O LOAD MONITOR ACTIVITY

SUBSYSTEM DATA OVERRUNS REQUESTS QUEUED BY LM

LOAD MONITOR QUEUEING BY MEMORY MODULE

MODULE NO.	REQUESTS QD	AVE Q SIZE
0	0	.00
1	0	.00
2	0	.00
3	0	.00

0 NON-RECOVERABLE DATA OVERRUNS

0 RECOVERABLE DATA OVERRUNS.

THERE WERE

This report specifies the number of requests that were queued by I/O Load Monitor because they could cause a data overrun (exceed the transfer rate of a memory module).

A: Average queue size given that a queue exists; therefore, the numbers are greater than or equal to one.

BANK NAME	OPTIONS	COMMON		BANK	ACTIVITY	(C) LOADS	(D)		REQ/SEC
		(A)	(B)				ZERO USE	COUNT	
CHANK	D NPREF	0				0	0		.0
CENTBK	D NPREF	75				10	43		.1
DEFBK	D NPREF	5				0	0		.0
IABK	D NPREF	995				42	11		1.8
QUEBK	D NPREF	5				0	0		.0
RWBK	D NPREF	358				68	502		.7
PLIBK	D NPREF	0				0	0		.0
AUX1BK	D NPREF	5				0	0		.0
AUX2BK	D NPREF	27				8	14		.1
AUX3BK	D NPREF	1278				0	0		2.4
ERRORS	D NPREF	0				0	0		.0
RFRS	D NPREF	0				0	0		.0
RFRS	D NPREF	0				0	0		.0
ASM	D NPREF	0				0	0		.0
SCANNERS	D NPREF	0				0	0		.0
COMPEX\$	D NPREF	0				0	0		.0
UTILITY\$	D NPREF	0				0	0		.0
MAIL	D NPREF	0				0	0		.0
FEMAIL	D NPREF	0				0	0		.0
DMATH\$	D NPREF	0				0	0		.0
ASM	D NPREF	0				0	0		.0
ED\$	D NPREF	33				23	0		.1
FURPR\$	D NPREF	22				5	1		.0
CTSS1	D NPREF	0				0	0		.0
BSCAN\$	D NPREF	0				0	0		.0
BFRS1	D NPREF	0				0	0		.0
BCBS1	D NPREF	0				0	0		.0
ACBS1	D NPREF	0				0	0		.0
ACBS2	D NPREF	0				0	0		.0
ACBS3	D NPREF	0				0	0		.0
ACBS4	D NPREF	0				0	0		.0
ACBS5A	D NPREF	0				0	0		.0
ACBS5L	D NPREF	0				0	0		.0
ACBS6	D NPREF	0				0	0		.0
CSACS	D NPREF	0				0	0		.0
CSANS	D NPREF	0				0	0		.0
CSCFH	D NPREF	0				0	0		.0
CSCGV	D NPREF	0				0	0		.0
CSDIAG	D NPREF	0				0	0		.0
CSFORM	D NPREF	0				0	0		.0
CSIBM	D NPREF	0				0	0		.0
CSINITIAL	D NPREF	0				0	0		.0
CSLABEL	D NPREF	0				0	0		.0
CSLIAN	D NPREF	0				0	0		.0
CSMASS	D NPREF	0				0	0		.0
CSMCS	D NPREF	0				0	0		.0
CSRPW	D NPREF	0				0	0		.0
CSDF	D NPREF	0				0	0		.0
CSISAM	D NPREF	0				0	0		.0
CSSTRING	D NPREF	0				0	0		.0
CSYMBIGNT	D NPREF	0				0	0		.0

Continued from previous page

CSTAPE	D	NPREF	0	0	0	0
TOTALS			2803	571	156	5.2

A: The abbreviations used are as follows:

S = Static  
 D = Dynamic  
 PCORE = Prefer Core  
 PWIRE = Prefer Wire  
 RQCORE = Require Core  
 RQWIRE = Require Wire  
 NPREF = No Preference

These options are taken from the SYSGEN parameters and may not be correct in this report. The options on the MAP card override the SYSGEN parameters and are the ones actually used.

B: Number of requests for the banks. Under certain conditions not all requests are counted. The number should be interpreted with caution.  
 C: The actual number of bank loads.  
 D: The number of times that the bank's use count drops to zero and it is eligible to be overlaid.



THE FOLLOWING STATISTICS DESCRIBE CTMC COMMUNICATION LINE USAGE

DURING THIS MONITORING PERIOD THE MAXIMUM NUMBER OF SIMULTANEOUSLY ACTIVE LTGS WAS 23  
 \*\*\* NOTE: ALL LINES ASSIGNED TO THE EXECUTIVE AT SYSTEM GENERATION ARE INCLUDED IN THIS COUNT \*\*\*\*

LINE	ACTIVE LINE TIME (SEC)	CHARACTER COUNT	CHARACTER/SEC WHILE THIS LINE IS ACTIVE
00001	539.959	0	.00
00002	539.959	0	.00
00003	539.959	0	.00
00004	539.959	0	.00
00011	539.959	0	.00
00012	539.959	0	.00
00013	539.959	0	.00
00014	539.959	0	.00
00021	539.959	0	.00
00022	539.959	0	.00
00023	539.959	0	.00
00024	539.959	0	.00
00025	539.959	0	.00
00026	539.959	924	1.71
00031	539.959	513	.95
00032	539.959	0	.00
00033	539.959	0	.00
00034	539.959	0	.00
00037	264.158	1878	7.11
00040	264.158	183	.69
00041	539.959	1491	26.84
00042	539.959	3983	7.38
00043	539.959	0	.00
00044	539.959	0	.00
00045	539.959	9723	18.01
00046	539.959	92	.17
00047	375.620	996	2.65
00050	375.611	126	.34
00051	539.959	0	.00
00052	539.959	0	.00
00053	539.959	588	1.09
00054	539.959	176	.33
00055	539.959	0	.00
00056	539.959	0	.00
00057	539.959	69	.13
00060	539.959	37	.07
00061	539.959	14828	27.46
00062	539.959	3507	6.49
00063	539.959	48204	89.27
00064	539.959	9745	18.05
00065	539.959	6240	11.56
00066	539.959	370	.69
00067	539.959	13053	24.17
00070	539.959	0	.00
00071	539.959	0	.00
00072	539.959	0	.00

Will not be accurate for short monitoring periods (< 15 minutes).

Not very useful for dial-up lines since no relationship exists between a line and a terminal.

# EXP00L ACTIVITY

BUFFER SIZE	(A)		(B)	
	REQUESTS	ITERATED REQUESTS	RELEASES	ITERATED RELEASES
4 WORDS	4213	0	4207	0
8 WORDS	5906	111	5915	110
16 WORDS	568	152	577	156
32 WORDS	10736	22	10719	18
64 WORDS	2892	241	2893	223
128 WORDS	1541	71	1539	64
256 WORDS	204	35	204	31
512 WORDS	0	13	1	12
	-----	-----	-----	-----
TOTALS	26080	645	26055	614
	48.3/SEC	1.2/SEC	48.3/SEC	1.1/SEC

Develop.  
Level A

## NON-IMMEDIATE-LOW PRIORITY REQUESTS

4 WORDS	4
8 WORDS	0
16 WORDS	0
32 WORDS	86
64 WORDS	0
128 WORDS	2
256 WORDS	0
512 WORDS	0
	-----
TOTAL	92

Non-immediate requests do not have to be satisfied immediately. If EXP00L is tight, they may be queued or deferred by taking an abnormal return. Activities are queued by setting the Program Long Wait, Segment Long Wait, and EXP00L Wait Bits.

## NON-IMMEDIATE-MED PRIORITY REQUESTS

4 WORDS	1575
8 WORDS	1845
16 WORDS	131
32 WORDS	5232
64 WORDS	872
128 WORDS	1090
256 WORDS	170
512 WORDS	0
	-----
TOTAL	10915

"Medium" is the default priority if none is specified on the EXP00L request. These requests commonly account for 50% of all EXP00L requests.

## NON-IMMEDIATE-HIGH PRIORITY REQUESTS

4 WORDS	0
8 WORDS	37
16 WORDS	0
32 WORDS	63
64 WORDS	0
128 WORDS	8
256 WORDS	6

A: Requests that caused buffers to be divided.

B: Releases that caused buffers to be combined.

Note: (A) and (B) may be controlled by the parameter RECBIN.

512 WCRDS

0  
-----  
TOTAL 114

IMMEDIATE PRIORITY REQUESTS

4 WCRDS 2634  
8 WCRDS 4024  
16 WCRDS 457  
32 WCRDS 5355  
64 WCRDS 2020  
128 WCRDS 441  
256 WCRDS 28  
512 WCRDS 0

-----  
TOTAL 14959

Develop.  
Level B

Immediate requests must be satisfied immediately. If they cannot be satisfied, the system will halt with a 077 STOP. These requests commonly account for 50% of all EXP00L requests.

EXP00L Thresholds:

EXP00L CCUNTS BY PRIORITY	EXTLOW LOW	EXTMED MED	EXTHIGH HIGH
ABNORMAL RETURNS FROM EXP00L	0	0	0
REQUESTS QUEUED BY EXP00L	0	0	0
STORAGE TYPE CHANGES BY EXP00L	0	0	0
EXP00L BECAME TIGHT	0 TIMES		

Develop.  
Level B

Non-immediate requests that could not be satisfied because EXP00L was tight.  
Primary requests that were forced into extended memory.  
EXP00L being tight implies that the EXP00L in use exceeded one of the EXP00L thresholds.  
A separate threshold is applied against each request priority.

THERE WERE 0 REQUESTS FOR EXP00L WHILE EXP00L WAS TIGHT

THE CURRENT EXP00L USAGE IS (A) 20228 WORDS, OF (B) 28672 WCRDS  
MAXIMUM EXP00L USED THIS BLOCK WAS (C) 20920 WDS  
MAXIMUM EXP00L ALLOCATION WAS (D) 28672 WORDS

User  
Level A

CURRENTLY ALLOCATED (Derived from EXPADJ.)

B-C = the minimum amount of EXP00L that went unused.  
D-B = the number of words that EXP00L expanded. The maximum number of EXP00L expansion blocks is given by EXPEXP.

	TALLY	PER SEC
MINOR REGISTER SAVES	115934	214.71
MAJOR REGISTER SAVES	39694	73.51
MINOR REGISTER RESTORES	127595	236.31
MAJOR REGISTER RESTORES	44301	82.05

D-X = Development Level - X

FUNCTION-SEGMENT ACTIVITY

Computed by DRP

D-A	D-A REQUESTS	D-A INT REQ	D-A NAME	D-A LOADS	D-B ACTIVE REQUESTS	D-B ACT REQ CLUSTERS	D-A AVE # REQ/LEADS	D-B PRØB ØF ACT REQ	D-B AVE # REQ
1.	79	45	EXITFO	44	0	0	1.80	.00	.00
2.	21	7	ERRFO	5	3	2	4.20	.14	1.50
3.	9	9	UREYIN	9	0	0	1.00	.00	.00
4.	10	3	SNUPQ	3	0	0	3.33	.00	.00
5.	7	3	SNXTP	3	1	1	2.33	.14	1.00
6.	25	8	SYMBKP	9	0	0	2.78	.00	.00
7.	6	3	SNALTS	4	0	0	1.50	.00	.00
8.	163	35	SNADØP	40	1	1	4.07	.01	1.00
9.	64	32	SNADCL	23	0	0	2.78	.00	.00
10.	580	269	PRØRDS	132	7	3	4.39	.01	2.33
11.	3	1	SNPRCN	1	0	0	3.00	.00	.00
14.	4	4	SNCSI	4	0	0	1.00	.00	.00
16.	4	2	BRØTRM	3	0	0	1.33	.00	.00
17.	991	943	DSØRIT	0	991	0	1.00	1.00	991.00
18.	62	62	RSØMSG	47	2	1	1.32	.03	2.00
19.	51	17	VALA	21	16	1	2.43	.31	16.00
20.	11	5	UCØMIS	6	0	0	1.83	.00	.00
21.	10	7	SNØTRM	8	0	0	1.25	.00	.00
22.	105	90	SYMPØB	43	16	7	2.44	.15	2.29
23.	9	7	SYMCØS	6	0	0	1.50	.00	.00
33.	30	2	SYMPØS	2	0	0	15.00	.00	.00
36.	32	16	REFØRK	16	0	0	2.00	.00	.00
38.	406	205	IGØFCS	51	4	3	7.96	.01	1.33
39.	289	248	FALL	135	38	15	2.14	.13	2.52
40.	64	62	FREL	15	0	0	4.27	.00	.00
41.	19	18	FØSEC	13	0	0	1.46	.00	.00
42.	8	8	FØSEC	6	0	0	1.33	.00	.00
43.	493	473	DØC	247	0	0	2.00	.00	.00
46.	4	4	IGNØDT	4	0	0	1.00	.00	.00
48.	219	197	DØLØAD	90	36	14	2.43	.16	2.57
49.	840	779	DØSI	153	432	175	5.49	.51	2.47
50.	122	122	DØQS	95	4	2	1.28	.03	2.00
53.	25	23	SØCCNT	19	0	0	1.32	.00	.00
54.	223	7	ACCØTG	11	0	0	20.27	.00	.00
55.	255	253	LØGM	126	0	0	2.02	.00	.00
56.	37	17	LØGIT	21	0	0	1.76	.00	.00
58.	8	8	NØME	5	0	0	1.60	.00	.00
59.	1514	491	FØNØIN	160	2	2	9.46	.00	1.00
60.	631	517	FØØSG	251	0	0	2.51	.00	.00
61.	212	212	FØØQ	64	0	0	3.31	.00	.00
62.	1	1	FØØCT	1	0	0	1.00	.00	.00
69.	106	94	PØP	44	4	4	2.41	.04	1.00
70.	29	4	CSN	3	0	0	9.67	.00	.00
71.	16	7	CSØCI	8	0	0	2.00	.00	.00
72.	12	4	CØNØEM	4	0	0	3.00	.00	.00
73.	212	53	CSØ	81	4	1	2.62	.02	4.00
74.	126	45	DØILR	60	2	1	2.10	.02	2.00
75.	710	710	CSØ1	150	12	7	4.73	.02	1.71
76.	656	656	CSØ12	218	56	18	3.01	.09	3.11
77.	3	3	FØØLD	3	0	0	1.00	.00	.00
79.	1542	514	CSØ	115	75	31	13.41	.05	2.42
81.	32	20	RØNID	16	0	0	2.00	.00	.00
83.	6	6	CSØ	6	0	0	1.00	.00	.00

86.	1	1	RELBAK	1	0	0	1.00	.00	.00
87.	89	63	DAPA	0	0	0	1.41	.00	.00
88.	9	9	DAPAM	0	0	0	1.00	.00	.00
91.	13	6	MSCEN	10	0	0	1.30	.00	.00
92.	48	22	DANCOR	26	0	0	1.85	.00	.00
94.	392	243	FIFREE	110	0	0	3.56	.00	.00
97.	6	4	CGMT	4	0	0	1.50	.00	.00
98.	4	4	COMPOL	4	0	0	1.00	.00	.00
100.	2	2	CMS	2	0	0	1.00	.00	.00
101.	10	6	CEMILM	4	0	0	2.50	.00	.00
104.	283	283	FCUI	64	47	38	4.42	.17	1.24
105.	1	1	TSWAP	1	0	0	1.00	.00	.00
106.	4	2	TSWMSH	4	0	0	1.00	.00	.00
107.	1	1	TSWFIL	1	0	0	1.00	.00	.00
121.	37	13	LIST	14	0	0	2.64	.00	.00
124.	2	2	TEST1	2	0	0	1.00	.00	.00
126.	11	10	SYSSIP	10	0	0	1.10	.00	.00
132.	407	260	FIBRK1	263	0	0	1.55	.00	.00
133.	2	2	FICKAC	2	0	0	1.00	.00	.00
134.	275	255	FICKAF	242	0	0	1.14	.00	.00
135.	2	2	FICKAT	2	0	0	1.00	.00	.00
136.	3	3	MISCNR	3	0	0	1.00	.00	.00
137.	268	162	FICRIT	175	0	0	1.53	.00	.00
138.	10	6	XKEY	7	0	0	1.43	.00	.00
140.	46	45	CGYCD	35	2	1	1.21	.04	2.00
142.	5821	5686	RESSEG	0	0	0	1.21	.04	2.00
TOTALS	13022	8764			1755	328	3.55	.13	5.35
	24.1/SEC	16.2/SEC			3.3/SEC				

THIS IS THE # OF CALLS TO RESIDENT EXEC AND IS NOT INCLUDED IN THE TOTALS BELOW.

↑  
would like to see  
this number > 10.

↑  
An increasing trend  
here indicates that  
DA overhead should  
be investigated.

- A: Include INT REQ and Loads.  
B: Do not count return links.  
C: Included in total request count.  
D: Requests while segment was active.  
E: Increment only when a request for an inactive function is followed by a request when the function is active. Counts once for string of requests to an active segment.

- F: Requests/Loads  
High is good and low is bad.  
G: Segments with a high request count and a low average request per load are candidates for residency or increased sticking power.  
H: Activity Requests/Requests  
Measures the burst nature of requests.  
I: Activity Requests/Act. Req. Clus.  
Average number of requests made while function was active.



[User Level - A]

These statistics are accumulated by the SIP idle activities.

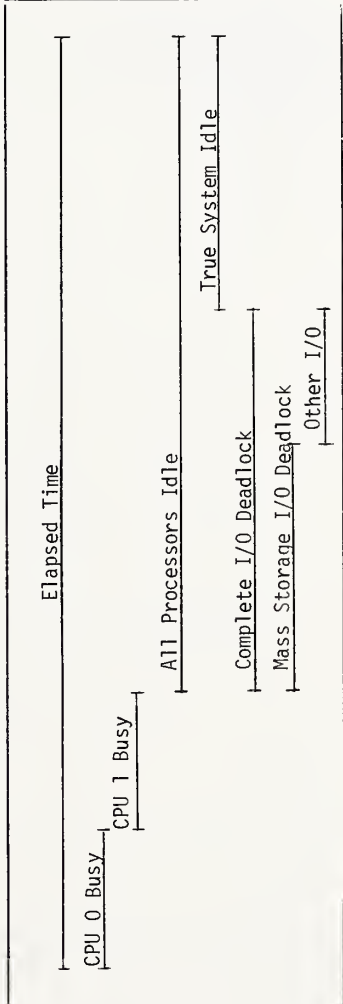
Updated every 6 seconds

Compare to core size divided by Average Program Size.

Identifies what subsystems the CPU's were waiting on.

SYSTEM IDLE ACTIVITY				DEMAND	
DESCRIPTION OF SYS INFO	TOTALS	BATCH			
AVERAGE RUNS OPEN	17.4	2.0		15.4	
AVERAGE RUNS IN CORE	5.8	1.6		4.2	
AVERAGE RUNS IN BACKLOG	2.8	2.8			
TOTAL RUNS THAT OPENED	23	5		18	
TOTAL RUNS THAT FIN'D	7				
SHARING PERCENTAGES -- MIN %: 70 MAX %: 90 INCR %: 5 [Printout of DMIN, DMAX, DINC.]					
TRUE SYSTEM IDLE	.01	0 %	(ALL PROCESSORS IDLE WITH NO I/O BEING DONE AT ALL)		
COMPLETE I/O DEADLOCK	188.58	35 %	(ALL PROCESSORS IDLE WITH SOME SUBSYSTEM BUSY)		
MASS ST I/O DEADLOCK	188.58	35 %	(ALL PROCESSORS IDLE WITH SOME MASS STORAGE SUBSYSTEM BUSY)		
ALL PROCESSORS IDLE	(A) 188.59	→ 35 %	(ALL PROCESSORS IDLE AT THE SAME TIME)		
Percent of total time that processors were idle while SIP was active.					
SYSTEM WENT IDLE	(B) 42131 TIMES,	AVERAGING	(C) .004 SECONDS PER TIME		
SUBSYSTEM ACTIVITY WHILE SYSTEM IDLE					
SUBSYSTEM	ALL PROCESSORS IDLE	% BUSY WHILE COMPLETE I/O DEADLOCK			
1 FH1782	12	12			
2 FH1782	50	50			
3 DI8414	1	1			
4 DI8414	16	16			
6 DI8460	78	78			
7 DI8460	92	92			
9 TAPU16	16	16			

C = A/B



THIS SECTION GIVES THE NUMBER OF ERS BY TYPE FOR ONLY THOSE REQUESTED

	#	ER	TALLY	ER RATE/SEC.	ER GROUP #
BASIC EXEC ER'S:					
	1.	IOS	3128	5.8	2
	2.	IOIS	89	.2	2
	3.	IEWS	21540	39.9	2
	6.	WAITS	1907	3.5	3
	7.	WANY\$	937	1.7	3
	8.	COM\$	66	.1	1
	9.	EXITS	2052	3.8	6
	11.	FORKE	253	.5	6
	13.	READS	1072	2.0	4
	14.	PRINT\$	2531	4.7	4
	15.	CSF\$	561	1.0	7
	16.	IOAXIS	128	.2	2
	18.	DATF\$	57	.1	1
	19.	TIMES	165	.3	1
	21.	IOXIS	38	.1	2
	23.	IIS	4	.0	6
	26.	ITEM\$	23	.0	7
	33.	MCT\$	1	.0	1
	35.	MCRES	13	.0	1
	38.	CMT\$	2	.0	5
	39.	CMIS	2	.0	5
	40.	CMOS	1380	2.9	5
	43.	CMSA\$	2723	5.0	5
	44.	TDATE\$	144	.3	1
	48.	TWAITS	34856	64.6	3
	51.	OPT\$	16	.0	1
	52.	PCT\$	27	.1	1
	53.	SETCS	4	.0	1
	56.	APRNT\$	51	.1	4
	65.	IALL\$	37	.1	6
	66.	TREAD\$	1	.0	4
	68.	PEIS	9	.0	7
	69.	PFS\$	4	.0	7
	72.	PFWL\$	9	.0	7
	73.	LOAD\$	147	.3	1
	76.	FACIL\$	111	.2	7
	85.	MSCONS	15	.0	7
	87.	SWICH\$	47061	87.2	6
	92.	AWAIT\$	5	.0	3
	93.	TSWAP\$	1	.0	7
	95.	PRICNS	3	.0	4
	96.	ACSF\$	15	.0	7
	99.	FACIT\$	149	.3	7
	101.	PNCHAS	146	.3	4
	102.	NAMES	8	.0	6
	103.	ACT\$	3248	15.3	6
	104.	DACT\$	7587	14.1	6
	106.	CRELS	2	.0	5
	111.	PSRS	5	.0	1
	112.	BANK\$	9	.0	1
	113.	ADEDS	12	.0	6
	118.	AREADS	2	.0	4
	120.	ATREAD	45	.1	4
	121.	LINK\$	8	.0	6
	123.	EXLNK\$	4	.0	6
	124.	UNLNE\$	3	.0	6
	125.	RLIST\$	8	.0	6
	TOTAL		137624	254.88	

	ER GROUP	TALLY	ER RATE/SEC.	ER GROUP #
BASIC EXEC ER'S:				
	ALL EXEC ER'S	137624	254.9	1
	IO RELATED ER'S	24923	46.2	2
	ER'S THAT CAUSE ACTIVITIES TO WAIT	37705	69.8	3
	ER'S INVOLVED WITH SYMBIONT ACTIVITY	3851	7.1	4
	COMMUNICATION RELATED ER'S	4309	8.0	5
	ER'S THAT START & STOP ACTIVITIES	65285	120.9	6
	FACILITY INVENTORY RELATED ER'S	897	1.7	7

Abbreviated Version: User Level - B  
Complete Report: Development Level - A

# RESPONSE TIMES

PERCENTAGE OF TOTAL NUMBER OF REQUESTS WHO WAITED FROM :

CORE-QUANTA  
INITIAL  
QUANTA  
EXCEEDED

SECONDS

AVERAGE WAITING TIME	TOTAL NUMBER OF REQUEST	REQ PER SEC	ZERO TO %	.50 TO %	1.00 TO %	1.50 TO %	2.00 TO %	3.00 TO %	5.00 TO %	10.00 TO %	15.00 TO %	30.00 TO %
2.495	130	.2	50.8	6.2	7.7	4.6	14.6	6.9	4.6	2.3	.8	1.5
ACCUMULATIVE %			50.8	56.9	64.6	69.2	83.8	90.8	95.4	97.7	98.5	100.0

Time between a carriage return or XMIT until the first output back to the terminal. May not be the complete response time.

TOTAL SYSTEM RESPONSE TO DEMAND USERS

Represents a major component of terminal response time.

DYNAMIC ALLOCATOR RESPONSE TIMES

The time a request waited on the Core Request Queue (CRQ) before it was loaded by DA.

Demand Priority Levels

CORE-QUANTA

TIP	0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
DEND-11	.449	104	.2	78.8	11.5	3.8	4.8	.0	.0	1.0	.0	.0
DEND-12	.385	12	.0	91.7	8.3	.0	.0	.0	.0	.0	.0	.0
DEND-13	.494	64	.1	68.8	23.4	3.1	3.1	1.6	.0	.0	.0	.0
DEND-14	.796	34	.1	29.4	44.1	17.6	8.8	.0	.0	.0	.0	.0
DEND-15	.590	21	.0	28.6	61.9	9.5	.0	.0	.0	.0	.0	.0
DEND-16	.706	23	.0	47.8	30.4	13.0	4.3	.0	.0	.0	.0	.0
DEND-17/C1-3	.753	140	.3	27.9	53.6	12.9	2.9	1.4	1.4	.0	.0	.0
DEND-17/C4-5	.000	0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
DEND-17/C6-7	.592	29	.1	27.6	37.9	13.8	3.4	13.8	3.4	.0	.0	.0
TOTAL DEMAND	.639	427	.8	49.4	34.9	9.1	3.7	1.9	.7	.2	.0	.0
ACCUMULATIVE %				49.4	84.3	93.4	97.2	99.1	99.8	100.0	100.0	100.0

A

B → 203

A: Divided based on initial program size where 1 is smallest and 7 is largest.

B: Indicator as to whether core quantum size should be increased.

PERCENTAGE OF TOTAL NUMBER OF REQUESTS WHO WAITED FROM :

	AVERAGE WAITING TIME	TOTAL NUMBER OF REQUEST	REQ PER SEC	ZERO T <sub>0</sub>		4.00 T <sub>0</sub>		240.00 T <sub>0</sub>		***** ON		INITIAL QUANTA EXCEEDED
				%		%		%		%		
CRIT. DEADLN BATCH	.000 .000	0 185	.0 .3	.0 75.1	.0 24.9	.0 24.9	.0 24.9	.0 24.9	.0 24.9	.0 24.9	.0 24.9	159
TOTAL BATCH	.900	185	.3	75.1	24.9	24.9	24.9	24.9	24.9	24.9	24.9	159
ACCUMULATIVE %				75.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	

NON-INITIAL QUANTA EXCEEDED

DEMAND LEVEL	COUNTS
12	52
13	40
14	38
15	19
16	10
17	9
20	59

THE FOLLOWING REPORT GIVES AN ANALYSIS OF CORE FRAGMENTATION ACCUMULATED AS EACH TASK FAILS

TOTAL LOAD ATTEMPTS TOTAL FAILS SUCCESSFUL LOADS TIMES UCIMPR FOUND SET  
TO START DA OVER

1161 Total number of Task Loads. (A) 454 A Task Load may fail more than once. 517 Loads where 1 or more of banks were common banks. (B) 229

BATCH TASKS AVE BANK SIZE # OF BANKS NON-BATCH TASKS AVE BANK SIZE

CORE PREF

0-REQUIRE WIRE  
1-REQUIRE CORE  
2-PREFER WIRE  
3-PREFER CORE  
4-NO PREFERENCE

0  
0  
0  
233  
2503

0 K  
0 K  
0 K  
2 K  
3 K

0  
0  
0  
244  
2144

0 K  
0 K  
0 K  
4 K  
3 K

\*\*THESE TASKS FAILED BECAUSE LOWER PRIORITY CORE WAS FRAGMENTED\*\*

# BATCH TASKS AVE SIZE AVE AVAIL CORE NON-BATCH TASKS AVE SIZE AVE AVAIL CORE

(C) 124

34 K

46 K

(D) 122

39 K

57 K

\*\*THESE TASKS FAILED BECAUSE THERE WAS NOT ENOUGH LOWER PRIORITY CORE\*\*

(E) 109

41 K

27 K

(F) 99

41 K

28 K

THE OVERHEAD ACCUMULATED AT TASK FAIL TIME IS .108 SECONDS ← Overhead to produce the above report.

THE FOLLOWING GIVES AN ANALYSIS OF CORE EFFICIENCY AND FRAGMENTATION ACCUMULATED WHEN THE DA DACTS

THE FOLLOWING COMES FROM ORDERING THE TASKS IN THE CORE Q BY SIZE AND TRYING TO FIT INTO UNUSED CORE (G)  
THE SMALLEST ONE FIRST, THEN THE NEXT SMALLEST, ETC. UP TO A MAXIMUM # OF ORDERED TASKS (MAXFIT = 6)  
TOTAL FIT ATTEMPTS TOTAL THAT FIT AVE # THAT FIT AVE SIZE TIMES THE # OF ORDERED TASKS = MAXFIT  
(WHERE THEY ALL FIT) WAS: 20 0

ACTSC DA ACTS TOTAL DA DACTS NON-FUNCT DACTS # OF TIMES QUE EXISTS TOTAL # TASKS IN Q AVE # TASKS PER Q

902

4369

957

466

556

1.1%

TOTAL MARKED FAILS

AVE FAILS PER QUE

AVE TASK SIZE

AVF BANK SIZE

410

.89

31 K

12 K

Indicator of delays caused by DA.

THE OVERHEAD ACCUMULATED AT LA DEACT TIME IS .032 SECONDS

A = C+D+E+F

B: User Core Improvement Flag

The core picture has changed so start the scan over again at the beginning of CRQ.

G: SIP tries to see how many tasks (up to MAXFIT) could have fit in core if it had not been fragmented.



THE FOLLOWING TABLE DISPLAYS AT WHICH DA PHASES THE 707 SUCCESSFUL LOADS WERE MADE

PHASE #	SUCCESSFUL LOADS
1	636
2	32
NO SECTOR NEEDED	39

THERE WERE (A) 51 (113%) OF (B) 45 TASKS THAT WERE SUSPENDED AT LEAST ONCE, WITH A TOTAL OF (C) 568 SUSPENDS (WHERE A TASK MAY BE SUSPENDED MORE THAN ONCE), RESULTING IN 4387.44 SECS OF SUSPENDED TIME.

AVERAGING:

T = Time SIP was active (seconds).  
A task corresponds to an activity--  
one entry on the CRQ.

86.03 SECS/TASK SUSPEND  
7.72 SECS/SUSPEND  
97.50 SECS/TASK EXECUTED  
5.67 TASK SUSPENDED/MIN  
.09 TASK SUSPENDED/SEC  
63.12 SUSPENDS/MIN  
1.05 SUSPENDS/SEC

--D/A  
--D/C  
--D/B  
--A/T/60  
--A/T  
--C/T/60  
--C/T

Computed in DRP

THERE WERE (E) 560 SWAP OUTS (NOT SUSPENDS) TO THE SWAPFILE WHICH MAY NOT BE RELATED TO THE NUMBER OF TASKS, FOR ONE SWAP MAY BE FOR N NUMBER OF TASKS OR FOR M NUMBER OF RANKS FOR ONE TASK.

THERE WERE (F) 729 I/O'S NEEDED TO PERFORM ALL THE SWAPS, WITH A TOTAL OF (G) 7715 K OF MEMORY SWAPPED OUT.

AVERAGING:

1.30 I/O'S/SWAP  
13.78 K OF CORE/SWAP  
10.58 K OF CORE/I/O  
62.23 SWAPS/MIN  
1.04 SWAPS/SEC  
857.29 K SWAPPED/MIN  
14.29 K SWAPPED/SEC

--F/E  
--G/E  
--G/F  
--E/T/60  
--E/T  
--G/T/60  
--G/T

Computed in DRP

DISTRIBUTION OF SWAPS PER SIZE

SIZE RANGE	% OF SWAPS	CUMULATIVE %
0 TO 4 K	37.3	37.3
4 TO 8 K	3.6	40.9
8 TO 12 K	2.0	42.9
12 TO 16 K	2.0	44.8
16 TO 20 K	40.7	85.5
20 TO 28 K	9.6	95.2
28 TO 36 K	2.1	97.3
36 TO 43 K	2.1	99.5
43 TO 51 K	.4	99.8
51 TO 599 K	.2	100.0

Swapping is done on a bank basis.  
Not all banks associated with an activity are necessarily swapped at once.

NOTE: A SUSPENDED TASK IS ONE WHICH IS REMOVED FROM BEING A CANDIDATE FOR EXECUTION BY THE DA. THIS USUALLY INVOLVES AT LEAST ONE BANK ACTUALLY BEING SWAPPED. BUT IT IS POSSIBLE UNDER CERTAIN CONDITIONS WHERE NONE OF THE BANKS ARE SWAPPED OUT.

THE FOLLOWING IS A LIST OF THE # OF TIMES VARIOUS EXEC AREAS WERE ENTERED

EXEC AREA	# OF TIMES ENTERED	PER SEC	DESCRIPTION OF COUNTERS	EXEC ELEMENT
CCR	795	1.5	CHAIN CORE REQ. IN CRQ	DA
RCR	793	1.5	REMOVE CORE REQ. IN CRQ	DA
PSOIOQ	1	.0	I/O DELAYED FOR PGM SWAPD	I/O
MPARTS	163	.3	I/O REQS ACROSS FILE GAP	I/O
IILQDS	45	.1	FIRST LOAD FOR EACH TASK	DALOAD
INITAL	66	.1	ANY LOAD FROM PROGRAM F	DASI
RELQAD	725	1.3	ANY LOAD FROM THE SWAP F	DASI
REL+IN	2	.0	LOADS FROM SWAP&PRGRM F	DASI
LOADS	779	1.4	LOADS FROM ANYWHERE	DASI
INBNKS	278	.5	BANKS FROM PROGRAM FILES	DAILR
RLBNKS	936	1.7	BANKS FROM SWAP FILE	DACCC
DASGIO	236	.4	DA S/G I/O-NOT LARGE BANK	DACCC
DAIO	1474	2.7	COUNT OF ANY DA I/O	DACCC
DISPI	330901	612.8	TOP OF THE DISPATCHER	DISP
NEXT	224930	416.6	FIND NEXT TASK&EXECUTE IT	DISP

DA: The control program of the Dynamic Allocator complex. Responsible for selecting and allocating memory to the banks of a program.

DALOAD: Performs the actual bank loads for the DA complex.

DASI: Swap-In. Handles reload of user and common banks from SWAP\$FILE and reactivates suspended activities waiting for the reload.

DAILR: Initial Load Routine. Builds initial Bank Descriptor Table (BDT) and Bank Load Packet (BLP).

DACCC: Core Contents Control. Called by DA after decision has been made for allocation of storage. Performs actual swapping out or suspending of programs. Calls DALOAD and DASI to get programs loaded.

DISP: The central control point of the operating system. The Dispatcher is responsible for giving control to the proper EXEC routines and allocating time slices to user activities.

IO: The I/O controller. Handles the interface between the user and the hardware via ER's to IO\$, IOW\$, IOWI\$, IOARB\$, and IOAXI\$.

# TEST AND SET SUMMARY

EXEC T/S TOTAL = 21199      USER T/S TOTAL = 0  
 ↖ For both processors.

SLEX1	=	18433
TSSWL	=	15
O,X	=	32
IE	=	198
TSASA	=	1
JK	=	1
MB	=	13
PCFIS	=	908
KDACT	=	1
SLRUNT	=	23
TSSLC	=	33
TSACS	=	2
TSCCIL	=	5
TSCMAP	=	21
TSNAM1	=	12
ESITER	=	292

TOTAL TRACED T/S = 19990

## Common Test and Set Cells

SLEX1:	Protects the switch list queues.
TSSWL:	Protects information in the switch list entry.
SLRUNT:	Protects total accumulated run time in the PCT.
IE:	Protects ER to CPOOL\$ and CJOIN\$.
PCFIS:	Protects granule item.
ESITER:	Protects ESI Activity Termination Queue.
TSCMAP:	Protects demand/batch sharing information.

DURING THIS TEST PERIOD THE SIP PACKAGE CREATED AN OVERHEAD OF	15.446 SECONDS
DURING THIS PERIOD THE CALC OF PGM SIZES CREATED AN OVERHEAD OF	224.241 SECONDS
DURING THIS PERIOD CORE FRAGMENTATION CREATED AN OVERHEAD OF	.140 SECONDS
DURING THIS PERIOD THE IO TRACE PACKAGE CREATED AN OVERHEAD OF	.364 SECONDS
DURING THIS PERIOD THE SIP OUTPUT ROUTINE CREATED AN OVERHEAD OF	.049 SECONDS
	-----
THE GRAND TOTAL IS	240.740 SECONDS

May be eliminated by  
setting S2 off.



U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO.  NBS SP 500-34	2. Gov't Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE  COMPUTER SCIENCE & TECHNOLOGY: UNIVAC 1108 EXEC LEVEL 32R2 Performance Handbook		5. Publication Date  June 1978	
		6. Performing Organization Code	
7. AUTHOR(S)  Capt. John C. Kelly and Lt. Gary P. Route		8. Performing Organ. Report No.	
9. PERFORMING ORGANIZATION NAME AND ADDRESS  NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234		10. Project/Task/Work Unit No.	
		11. Contract/Grant No.  AY-505-080-Army	
12. Sponsoring Organization Name and Complete Address (Street, City, State, ZIP)  U.S. Army Military Personnel Center Alexandria, VA 22332		13. Type of Report & Period Covered  Final	
		14. Sponsoring Agency Code	
15. SUPPLEMENTARY NOTES  Study performed and report prepared by the Federal Computer Performance Evaluation and Simulation Center, Washington, DC 20330.			
16. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)  This is a report prepared for the Army Military Personnel Center (MILPERCEN). It describes a set of hypotheses for evaluating the performance of UNIVAC 1100 Computer Systems. The hypotheses specifically apply to EXEC Level 32R2 operating on UNIVAC 1108 Computer Systems. Attempts to apply the guidelines to different UNIVAC 1100 models or to different levels of the EXEC may lead to erroneous results.  The report contains sections on each major complex within the EXEC. Each section (1) states the performance hypotheses associated with that complex, (2) explains how the available measurement tools may be used to determine if the hypotheses are true, and (3) identifies which performance parameters need to be changed to correct the problem situation. All of the performance hypotheses discussed in the body of the report are collected together in Appendix A. Appendix B summarizes the performance parameters and Appendix C contains an introduction to the Software Instrumentation Package (SIP).			
17. KEY WORDS (six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons) Computer performance management; measurement tools; operating system performance; performance guidelines; performance measurement; tuning guides; UNIVAC 1108.			
18. AVAILABILITY  <input checked="" type="checkbox"/> Unlimited  <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS  <input checked="" type="checkbox"/> Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Stock No. SN003-003  <input type="checkbox"/> Order From National Technical Information Service (NTIS) Springfield, Virginia 22151		19. SECURITY CLASS (THIS REPORT)  UNCL ASSIFIED	21. NO. OF PAGES.  144
		20. SECURITY CLASS (THIS PAGE)  UNCLASSIFIED	22. Price  \$3.25

**ANNOUNCEMENT OF NEW PUBLICATIONS ON  
COMPUTER SCIENCE & TECHNOLOGY**

**Superintendent of Documents,  
Government Printing Office,  
Washington, D. C. 20402**

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Bureau of Standards Special Publication 500-.

Name \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_

(Notification key N-503)

# NBS TECHNICAL PUBLICATIONS

## PERIODICALS

**JOURNAL OF RESEARCH**—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology, and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers each issue contains complete citations to all recent NBS publications in NBS and non-NBS media. Issued six times a year. Annual subscription: domestic \$17.00; foreign \$21.25. Single copy, \$3.00 domestic; \$3.75 foreign.

Note: The Journal was formerly published in two sections: Section A "Physics and Chemistry" and Section B "Mathematical Sciences."

### DIMENSIONS/NBS

This monthly magazine is published to inform scientists, engineers, businessmen, industry, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on the work at NBS. The magazine highlights and reviews such issues as energy research, fire protection, building technology, metric conversion, pollution abatement, health and safety, and consumer product performance. In addition, it reports the results of Bureau programs in measurement standards and techniques, properties of matter and materials, engineering standards and services, instrumentation, and automatic data processing.

Annual subscription: Domestic, \$12.50; Foreign \$15.65.

## NONPERIODICALS

**Monographs**—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

**Handbooks**—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

**Special Publications**—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

**Applied Mathematics Series**—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

**National Standard Reference Data Series**—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a world-wide program coordinated by NBS. Program under authority of National Standard Data Act (Public Law 90-396).

NOTE: At present the principal publication outlet for these data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St. N.W., Wash., D.C. 20056.

**Building Science Series**—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

**Technical Notes**—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

**Voluntary Product Standards**—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The purpose of the standards is to establish nationally recognized requirements for products, and to provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

**Consumer Information Series**—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order **above** NBS publications from: Superintendent of Documents, Government Printing Office, Washington, D.C. 20402.

Order **following** NBS publications—NBSIR's and FIPS from the National Technical Information Services, Springfield, Va. 22161.

**Federal Information Processing Standards Publications (FIPS PUB)**—Publications in this series collectively constitute the Federal Information Processing Standards Register. Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

**NBS Interagency Reports (NBSIR)**—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Services (Springfield, Va. 22161) in paper copy or microfiche form.

## BIBLIOGRAPHIC SUBSCRIPTION SERVICES

The following current-awareness and literature-survey bibliographies are issued periodically by the Bureau:

**Cryogenic Data Center Current Awareness Service.** A literature survey issued biweekly. Annual subscription: Domestic, \$25.00; Foreign, \$30.00.

**Liquefied Natural Gas.** A literature survey issued quarterly. Annual subscription: \$20.00.

**Superconducting Devices and Materials.** A literature survey issued quarterly. Annual subscription: \$30.00. Send subscription orders and remittances for the preceding bibliographic services to National Bureau of Standards, Cryogenic Data Center (275.02) Boulder, Colorado 80302.

**U.S. DEPARTMENT OF COMMERCE**  
**National Bureau of Standards**

Washington, D.C. 20234

OFFICIAL BUSINESS

Penalty for Private Use, \$300

POSTAGE AND FEES PAID  
U.S. DEPARTMENT OF COMMERCE  
COM-215



SPECIAL FOURTH-CLASS RATE  
BOOK

---