SUPERVISORY SYSTEM FOR ON-BOARD CHECKOUT AND DATA MANAGEMENT SYSTEM (OCDMS), SATURN/APOLLO APPLICATION PROGRAM

PRC D-1178
Rev A

23 August 1967

Prepared for

National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Huntsville, Alabama

PLANNING RESEARCH CORPORATION
LOS ANGELES, CALIFORNIA WASHINGTON, D.C.

SUPERVISORY SYSTEM FOR ON-BOARD CHECKOUT AND DATA MANAGEMENT SYSTEM (OCDMS), SATURN/APOLLO APPLICATION PROGRAM

PRC D-1178
Rev A

23 August 1967

Prepared for

National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Huntsville, Alabama

Prepared by

D. D. Creed
D. H. Stomberg

PLANNING RESEARCH CORPORATION
LOS ANGELES, CALIF. WASHINGTON, D.C.

This document was prepared by Planning Research Corporation under Contract Number NAS8-20367, "An Airborne Evaluating Equipment Study," for the George C. Marshall Space Flight Center of the National Aeronautics and Space Administration. The work was administered under the technical direction of Quality and Reliability Assurance Laboratory, Marshall Space Flight Center, with Walter T. Mitchell acting as project manager.

Specification No. CG00N8-2036710A
Revision No. A
Release Date 23 August 1967

CONTRACT END ITEM SPECIFICATION
(COMPUTER PROGRAM)

PERFORMANCE/DESIGN AND
PRODUCT CONFIGURATION REQUIREMENTS

(CEI 036710A)

SUPERVISORY SYSTEM FOR ON-BOARD CHECKOUT
AND DATA MANAGEMENT SYSTEM (OCDMS),
SATURN/APOLLO APPLICATION PROGRAM

Approved by CGRiley
(Planning Research Corp.)

Approved by ____________________

Date 23 August 1967

Approval Date ____________________

Contract Number NAS8-20367

PLANNING RESEARCH CORPORATION
SPECIFICATION CHANGE LOG

As Of ____________

Superseding ____________

SCN NO.	ECP NO.	SCN DATE	PAGES AFFECTED	ITEM EFFECTED

PLANNING RESEARCH CORPORATION

SPECIFICATION ISSUE	ECP'S	PRODUCTION EFFECTIVITY
BASIC Date______	ORIGINAL DESIGN ECP'S	*
Date______	INCORPORATED ECP'S	
Date______	INCORPORATED ECP'S	
Date______	INCORPORATED ECP'S	

* (Effectivity of the Specification on Production article or item Serial Number as Applicable).

TABLE OF CONTENTS

Page

1.0 SCOPE . . . . I-1 of 75

2.0 APPLICABLE DOCUMENTS. . . . . I-2 of 75

2.1 Project Documents . . . . I-2 of 75

2.2 Specifications . . . . I-2 of 75

2.3 Other Publications . . . . I-2 of 75

3.0 REQUIREMENTS . . . . I-3 of 75

3.1 Performance . . . . I-3 of 75

3.1.1 System Requirements . . . . I-3 of 75

3.1.2 Operational Requirements . . . . I-3 of 75

3.1.2.1 Function 1: System Management . . . I-4 of 75

3.1.2.1.1 Sources and Types of Inputs for System Management Functions. . . . . I-6 of 75

3.1.2.1.2 Destinations and Types of Outputs for System Management. . . . . I-7 of 75

3.1.2.1.3 Information Processing for Systems Management. . . . . I-8 of 75

3.1.2.2 Function 2: Memory Management. . . I-10 of 75

3.1.2.2.1 Sources and Types of Inputs for Memory Management . . . . I-15 of 75

3.1.2.2.2 Destinations and Types of Outputs for Memory Management . . I-16 of 75

3.1.2.2.3 Information Processing for Memory Management . . . . I-18 of 75

3.1.2.3 Function 3: Language Interpreter . . . I-20 of 75

3.1.2.3.1 Sources and Types of Inputs for the Language Interpreter . . . . . I-20 of 75

3.1.2.3.2 Destinations and Types of Outputs for the Language Interpreter. . . . . I-22 of 75

3.1.2.3.3 Information Processing for the Language Interpreter . . . . I-23 of 75

3.1.2.4 Function 4: Procedure Management . I-25 of 75

TABLE OF CONTENTS
(Continued)

Page

3.1.2.4.1 Sources and Types of Inputs for Procedure Management ...... I-25 of 75

3.1.2.4.2 Destinations and Types of Outputs for Procedure Management.............. I-28 of 75

3.1.2.4.3 Information Processing for Procedure Management ...... I-29 of 75

3.1.2.5 Function 5: Schedule Management... I-30 of 75

3.1.2.5.1 Sources and Types of Inputs for Schedule Management .... I-30 of 75

3.1.2.5.2 Destinations and Types of Outputs for Schedule Management.. I-33 of 75

3.1.2.5.3 Information Processing for Schedule Management .... I-33 of 75

3.1.2.6 Function 6: Interruption Management I-35 of 75

3.1.2.6.1 Sources and Types of Inputs for Interruption Management ..... I-35 of 75

3.1.2.6.2 Destinations and Types of Outputs for Interruption Management .... I-38 of 75

3.1.2.6.3 Information Processing for Interruption Processing ...... I-39 of 75

3.1.2.7 Function 7: Console Communication . I-41 of 75

3.1.2.7.1 Sources and Types of Inputs for Console Communication ...... I-41 of 75

3.1.2.7.2 Destinations and Types of Outputs for Console Communication............ I-44 of 75

3.1.2.7.3 Information Processing for Console Communication ...... I-45 of 75

3.1.2.8 Function 8: CIU/Signal Adapter Communication .... I-47 of 75

3.1.2.8.1 Sources and Types of Inputs for CIU/Signal Adapter Communication............ I-47 of 75

TABLE OF CONTENTS
(Continued)

Page

3.1.2.8.2 Destinations and Types of Outputs for CIU/Signal Adapter Communication . . . . I-49 of 75

3.1.2.8.3 Information Processing for CIU/Signal Adapter Communication . . . . I-50 of 75

3.1.2.9 Function 9: Uplink Communication . . I-51 of 75

3.1.2.9.1 Sources and Types of Inputs for Uplink Communication . . . . I-51 of 75

3.1.2.9.2 Destinations and Types of Outputs for Uplink Communication . I-52 of 75

3.1.2.9.3 Information Processing for Uplink Communication . . . . I-52 of 75

3.1.2.10 Function 10: Downlink Communication . . . . I-53 of 75

3.1.2.10.1 Sources and Types of Inputs for the Downlink Communication Function. . . . . I-53 of 75

3.1.2.10.2 Destinations and Types of Outputs for Downlink Communication . . . . I-54 of 75

3.1.2.10.3 Information Processing for Downlink Communication. . . . . I-55 of 75

3.1.3 Data Base Requirements. . . . . I-57 of 75

3.1.4 Human Performance . . . . I-57 of 75

3.2 CPCEI Definition . . . . I-57 of 75

3.2.1 Interface Requirements . . . . I-57 of 75

3.2.1.1 Interface Block Diagram . . . . I-57 of 75

3.2.1.2 Detailed Interface Definition. . . . . . . I-57 of 75

3.2.1.2.1 OCDMS Support System CPCEI . I-57 of 75

3.2.1.2.2 OCDMS Experiment Procedure CPCEI . . . . I-61 of 75

3.2.1.2.3 OCDMS Computer and Associated Peripheral Equipment . . . . I-61 of 75

TABLE OF CONTENTS
(Continued)

Page

3.2.1.2.3.1 Computer . . . . I-61 of 75

3.2.1.2.3.2 Manufacturer-Supplied Programming Language(s) . I-61 of 75

3.2.1.2.3.3 Control/Display Unit. . . . . I-61 of 75

3.2.1.2.3.4 Bulk Storage . . . . I-61 of 75

3.2.1.2.4 OCDMS Experimental Hardware. I-61 of 75

3.2.1.2.5 Operations Personnel . . . . I-61 of 75

3.2.1.2.6 Ground Stations (PIF, ACE, GOSS) . . . . I-62 of 75

3.2.2 Government-Furnished Property List. . . . I-62 of 75

3.3 Design Requirements . . . . I-62 of 75

3.3.1 Programming Standards . . . . I-62 of 75

3.3.2 Program Design . . . . I-62 of 75

3.3.3 Program Modification . . . . I-62 of 75

3.3.4 CPCEI Testing Facilities . . . . I-62 of 75

3.3.5 CPCEI Expandability . . . . I-64 of 75

4.0 QUALITY ASSURANCE PROVISIONS . . . . I-65 of 75

4.1 Implementation Test Requirements . . . . I-65 of 75

4.1.1 Design and Development Testing . . . . I-66 of 75

4.1.2 Preliminary Qualification Test . . . . I-69 of 75

4.1.2.1 Qualification Test Requirements . . . . I-69 of 75

4.1.2.2 Resources Required for Testing . . . . I-69 of 75

4.1.2.3 Test Schedules and Locations . . . . . . I-69 of 75

4.1.3 Special Test Requirements . . . . I-69 of 75

4.2 Integration Test Requirements . . . . I-69 of 75

4.2.1 General. . . . . I-69 of 75

4.2.1.1 Sequence of Tests. . . . . I-69 of 75

4.2.1.2 Functions To Be Tested. . . . . I-70 of 75

4.2.1.3 Testing Environment . . . . I-70 of 75

4.2.1.4 Support Computer Programs Required . . . . I-70 of 75

TABLE OF CONTENTS
(Continued)

Page

4.2.1.5 Personnel Required . . . . I-70 of 75
4.2.1.6 Equipment Required . . . . I-70 of 75
4.2.2 Acceptance/Qualification Test . . . . I-70 of 75
4.2.2.1 Sequence of Tests. . . . . I-70 of 75
4.2.2.2 Functions To Be Tested. . . . . I-70 of 75
4.2.2.3 Testing Environment. . . . . I-70 of 75
4.2.2.4 Support Computer Programs Required. I-70 of 75
4.2.2.5 Personnel Required . . . . I-70 of 75
4.2.2.6 Equipment Required . . . . I-70 of 75
6.0 NOTES. . . . . I-70 of 75
10.0 APPENDIX . . . . I-71 of 75
10.1 Glossary. . . . . I-71 of 75

LIST OF EXHIBITS

		Page
1.	Functional Operation of OCDMS Supervisory System	I-5 of 75
2a.	System Management (System Initialization)	I-9 of 75
2b.	System Management (System Status/Accounting)	I-11 of 75
2c.	System Management (System Termination)	I-12 of 75
2d.	System Management (Queued Interruption Processing)	I-13 of 75
3.	Memory Management	I-15 of 75
4.	Language Interpreter	I-21 of 75
5a.	Procedure Management	I-26 of 75
5b.	Procedure Control Block	I-27 of 75
6a.	Scheduling Management	I-31 of 75
6b.	SAP Mechanism	I-36 of 75
7.	Interruption Processing	I-37 of 75
8.	Interrupts Allowed	I-40 of 75
9a.	Console Communication (Printer and Keyboard)	I-42 of 75
9b.	Console Communication (Display)	I-43 of 75
10.	CIU/Signal Adapter Communication	I-48 of 75
11.	OCDMS Supervisory System Interfaces	I-58 of 75
12.	OCDMS Support System--Supervisory System Interface Elements	I-60 of 75
13.	Generalized System Operation Flow	I-63 of 75

Specification No. CG00N8-2036710A
Revision No. A
Release Date 23 August 1967

CONTRACT END ITEM SPECIFICATION
(COMPUTER PROGRAM)

PART I

PERFORMANCE/DESIGN REQUIREMENTS

(CEI 036710A)

SUPERVISORY SYSTEM FOR ON-BOARD CHECKOUT AND DATA MANAGEMENT SYSTEM (OCDMS), SATURN/APOLLO APPLICATION PROGRAM

Approved by CS Riley
(Planning Research Corp.)

Date 23 August 1967

Contract Number NAS8-20367

Approved by ____________________

Approval Date ____________________

1.0 SCOPE

This part of this specification establishes requirements for performance, design, test, and qualification of a group of computer programs identified as a Supervisory System for an On-Board Checkout and Data Management System (OCDMS), CEI 036710A. This CPCEI is used to provide a spaceborne-vehicle, computer-based system with the functional capability to do the following:

- Centralize functions controlling the flow of data between computer programs and external subsystems
- Provide a software environment capable of supporting concurrent program execution (i.e., multiprogramming)
- Manage system resources in such a manner as to make for the most efficient use of the available hardware and to assure system response time necessary for proper execution of the application programs
- Continuously monitor and control program and hardware operation

The CPCEI requires, as a basis of its operations, a preprocessed data base, application programs, and verified computer codes. These requirements shall be satisfied in accordance with the specifications set forth by the OCDMS Support System CPCEI, Reference 2.2.2.

2.0 APPLICABLE DOCUMENTS

The following documents of exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between documents referenced herein, the detailed contents of Sections 3, 4, and 10, and the detailed requirements in Sections 3, 4, and 10 shall be considered superseding requirements.

2.1 Project Documents: None

2.2 Specifications

2.2.1 Performance and Design Requirements for the On-Board Checkout and Data Management System, General Specifications for, Specification No. SS2036701A, dated 29 March 1967

2.2.2 Performance/Design and Product Configuration Requirements, Support System for On-Board Checkout and Data Management System (OCDMS) Saturn/Apollo Applications Program, Specification No. CG00N8-2036701, March 1967

2.3 Other Publications

2.3.1 MSFC-PROC-485, Input for Configuration Management Accounting and Reporting System, Preparation of, 28 October 1965

2.3.2 PRC D-1336, On-Board Checkout System Hardware Design, 11 November 1966

2.3.3 PRC D-850, On-Board Checkout System Software Design, 17 November 1966

2.3.4 SR-QUAL-65-48; NASA, MSFC, Directives for Software Development, 24 June 1966

2.3.5 PRC D-1403, Technical Advisement Memorandum No. 171-3 On-Board Checkout and Data Management System: Design Supplement, 31 March 1967

2.3.6 PRC D-1417, Revision A, Technical Advisement Memorandum No. 171-4, On-Board Checkout and Data Management System: Control and Display Unit, 19 July 1967

3.0 REQUIREMENTS

The OCDMS requirements include performance requirements, design and construction requirements, and requirements for functional areas. These requirements define and control a spaceborne checkout and data management system to be used for the Saturn/Apollo Applications (S/AA) Experiment Program or similar extended space missions. Performance and design requirements included herein are allocated from, identical with, or in recognition of, requirements established by the OCDMS General Specification or the OCDMS Support System Specification (References 2.2.1 and 2.2.2).

3.1 Performance

Pertinent performance requirements established by these two documents include the following:

- Stimuli generation and application
- Response measurement
- On-line operation communications
- Uplink/downlink communications
- Operating mode requirements
- Internal operational requirements
- OCDMS hardware/software verification

OCDMS Supervisory System requirements will also be derived from the following specific OCDMS requirements.

- Reliability
- Maintainability
- Useful life
- Human performance
- Safety
- Environmental constraints

3.1.1 System Requirements

The limits and/or capacities of OCDMS Supervisory System performance shall be constrained to operational envelopes including system control, procedures and data management, interface management, and system verification testing. These applications shall in general be restricted to tasks prescribed by the OCDMS Support System CPCEI (Reference 2.2.2) and operations specified on-line through man-machine interface or initiated via uplink communication from the ground.

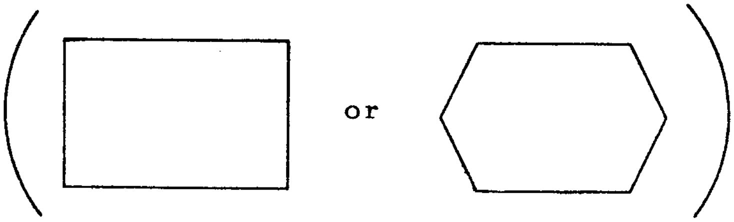
3.1.2 Operational Requirements

The OCDMS Supervisory System CPCEI is illustrated in Exhibit 1. This block diagram is meant to portray the major functional areas of the Supervisory System and the general sphere of responsibility of each of the areas. The diagram itself is not intended to specify structure or levels of hierarchy within the Supervisory System. Overall operational functions are identified in subsequent paragraphs. Where it seemed necessary and/or helpful, functional diagrams further describing the respective operational functions have been included. In these diagrams, the following conventions are used.

- A single line (→) implies a flow of Central Processing Unit (CPU) control.

- A double line (⇒) implies a flow of information.

- An emphasized figure



implies an operation that is an implicit part of the operational function being illustrated.

The identification, descriptions, and relationships expressed (in both prose and diagram) for these CPCEI functions are intended for total systems operations and are not intended as a restrictive design definition of computer program component (CPC) organization or as functional descriptions of particular main programs or subprograms.

3.1.2.1 Function 1: System Management

The system management function shall provide general executive control for the Supervisory System. The particular performance criterion for computer program components (CPC's) covered by this function includes the following:

- Initialization of the CPCEI

- Termination of the CPCEI

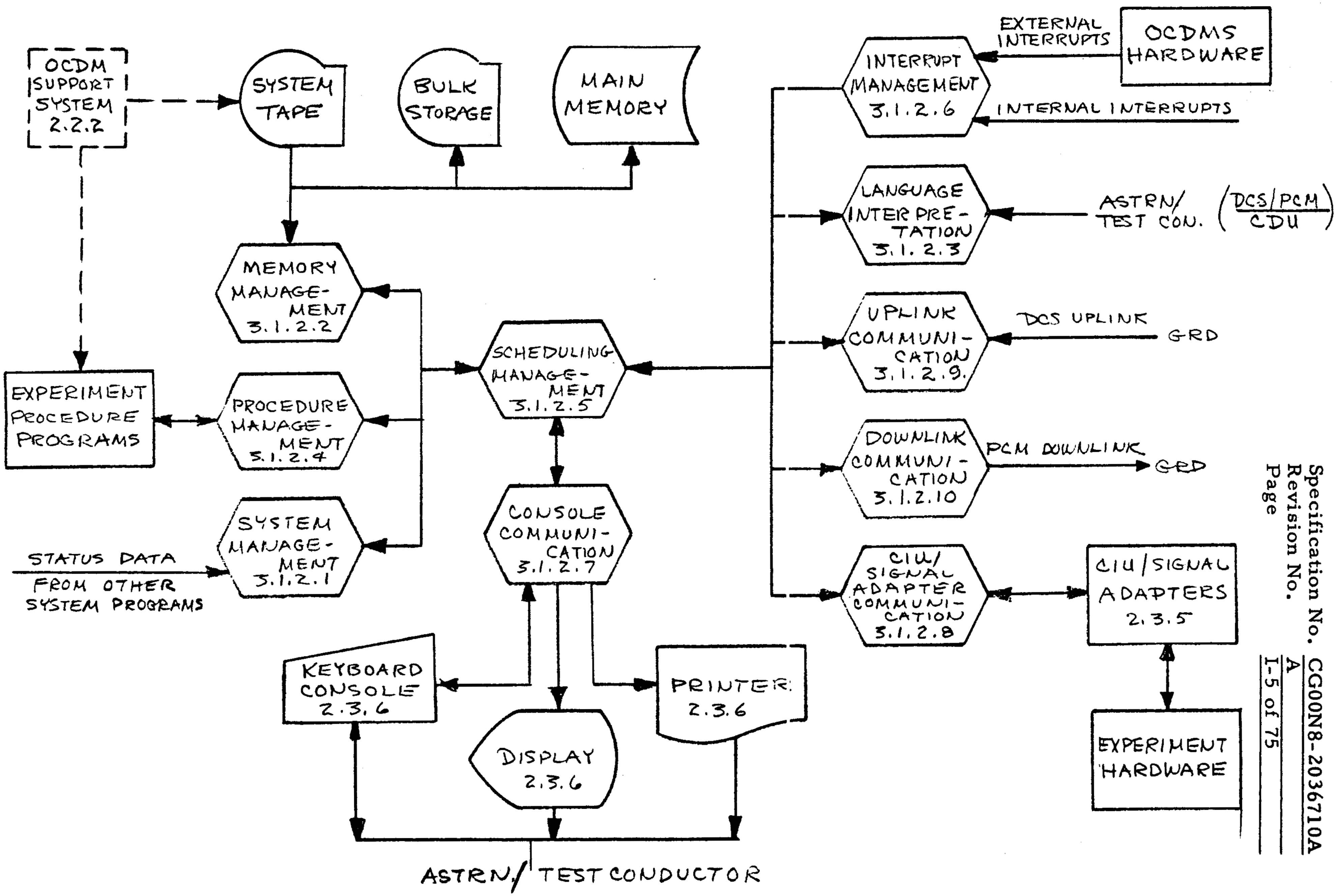

OCDM SUPPORT SYSTEM 2.2.2
SYSTEM TAPE
BULK STORAGE
MAIN MEMORY
EXTERNAL INTERRUPTS
OCDMS HARDWARE
INTERRUPT MANAGEMENT 3.1.2.6
INTERNAL INTERRUPTS
LANGUAGE INTERPRETATION 3.1.2.3
ASTRN/ TEST CON. (DCS/PCM / CDU)
MEMORY MANAGEMENT 3.1.2.2
UPLINK COMMUNICATION 3.1.2.9.
DCS UPLINK
GRD
EXPERIMENT PROCEDURE PROGRAMS
PROCEDURE MANAGEMENT 3.1.2.4
SCHEDULING MANAGEMENT 3.1.2.5
DOWNLINK COMMUNICATION 3.1.2.10
PCM DOWNLINK
GRD
SYSTEM MANAGEMENT 3.1.2.1
STATUS DATA
FROM OTHER SYSTEM PROGRAMS
CONSOLE COMMUNICATION 3.1.2.7
CIU/ SIGNAL ADAPTER COMMUNICATION 3.1.2.8
CIU/SIGNAL ADAPTERS 2.3.5
KEYBOARD CONSOLE 2.3.6
PRINTER 2.3.6
DISPLAY 2.3.6
EXPERIMENT HARDWARE
ASTRN/ TEST CONDUCTOR

EXHIBIT 1 - FUNCTIONAL OPERATION OF OCDMS SUPERVISORY SYSTEM

- Input/output resource allocation (channel selection, etc.)
- Restart provisions (after power-off conditions)
- Error detection and recovery (not otherwise allocated to other system functions)
- Monitoring, accumulation, and report generation of system status data
- Accounting and event trial facilities

3.1.2.1.1 Source and Types of Inputs for System Management Functions

a. Functional inputs shall include, but not be limited to, the following types:

(1) Operations personnel instructions in the form of keyboard and manual control switch signals

(2) System Definition, which will be supplied by the Support System and will consist of a list of those programs and data sets that constitute the resident Supervisory System

(3) Programs and data sets that constitute the resident Supervisory System

(4) A description of the status and configuration of the major elements of the system (System Status and Configuration Dictionary), see Section 3.1.2.1.2.b, below. This consists of an array of status words indicating which programs within the system are active, inactive, or busy; status words for program detected errors; and status words indicating which portions of the hardware system are active, inactive, or have faulted.

(5) Miscellaneous flags and other entries, from the major programming elements of the system, providing the information required for the System Status and Configuration Dictionary

b. Sources of these inputs shall, respectively, include but not necessarily be limited to the following:

(1) Console Communication function (3.1.2.7)

(2) OCDMS Support System (Reference 2.2.2, OCDMS Support System CPCEI)

(3) OCDMS Support System

(4) Maintained as a function of System Management

(5) Other Supervisory System functions (3.1.2.2-3.1.2.10), as appropriate

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Arrival frequency (to be determined)

3.1.2.1.2 Destination and Types of Outputs for System Management

a. Functional output shall include, but not necessarily be limited to, the following types:

(1) Updated description of system status to System Status and Configuration Dictionary

(2) System status displays, reports, messages, and accounting and events records--to be sent to the operations personnel via the display, printer, and/or PCM downlink; or to auxiliary storage, as appropriate

(3) Control signals (discretes, code words, etc.) to the system hardware to implement equipment shutdown

b. Destinations of these outputs shall, respectively, include but not necessarily be limited to, the following:

(1) Carried out as a function of System Management

(2) Console Communication and Downlink Communication functions (Sections 3.1.2.7 and 3.1.2.10)

(3) To be determined

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Output frequency (to be determined)

3.1.2.1.3 Information Processing for Systems Management

a. System Initialization Subprogram Function

The system shall be capable of starting or restarting (after power shutdown) system activity in response to an interruption initiated by the operations personnel via the Control/Display Unit (CDU) or a Digital Command System (DCS) uplink message, or by the wake-up timer (when previously set by the system). When entered, this subprogram will proceed to fetch into core, from bulk storage, those CPC's (load module contents) containing that portion of the Supervisory System necessary for start-up. A definition of the required CPC's shall be provided by header-information on the bulk storage media (system definition provided by Support System), by the operations personnel (if necessary or desired by them), and by the System Status and Configuration Dictionary (in the case of restart after power-off conditions). In addition to loading the necessary modules into main storage, System Initialization will involve the execution of system verification and checking routines to a level necessary to establish confidence in the system. These will include routines that verify system input (used during ground checkout) and that perform self-check and error-detection operations in relation to the Computer Interface Unit (CIU), auxiliary memory devices, CDU, and main memory (parity check). These operations will be carried out as directed by the System Initialization Subprogram and supplementary commands from the operations personnel. Exhibit 2a illustrates, functionally, the operation of this subprogram function.

b. System Status and Configuration Management Subprogram Functions

The system shall dynamically maintain the System Status and Configuration Dictionary (SCDY) based on information transmitted to the System Status and Configuration Management subprogram by the various elements of the Supervisory System. This information will be of the nature required for the SCDY to contain a profile of the system indicating processing load, memory utilization, peripheral equipment utilization (input/output load), and hardware status. This together with the Unit Control Blocks (Sections 3.1.2.2, 3.1.2.7 through 10) and Procedure Control Blocks (Sections 3.1.2.4, Procedure Management) shall provide a complete description of the status and configuration of the onboard system. This information will be periodically summarized into a form suitable for output to the operations personnel and/or saved on auxiliary storage as an accounting/events record. This summary may take place on the basis of one or more of the following:

(1) As a normal scheduled function of System Management

(2) As a result of a request for such by the operations personnel

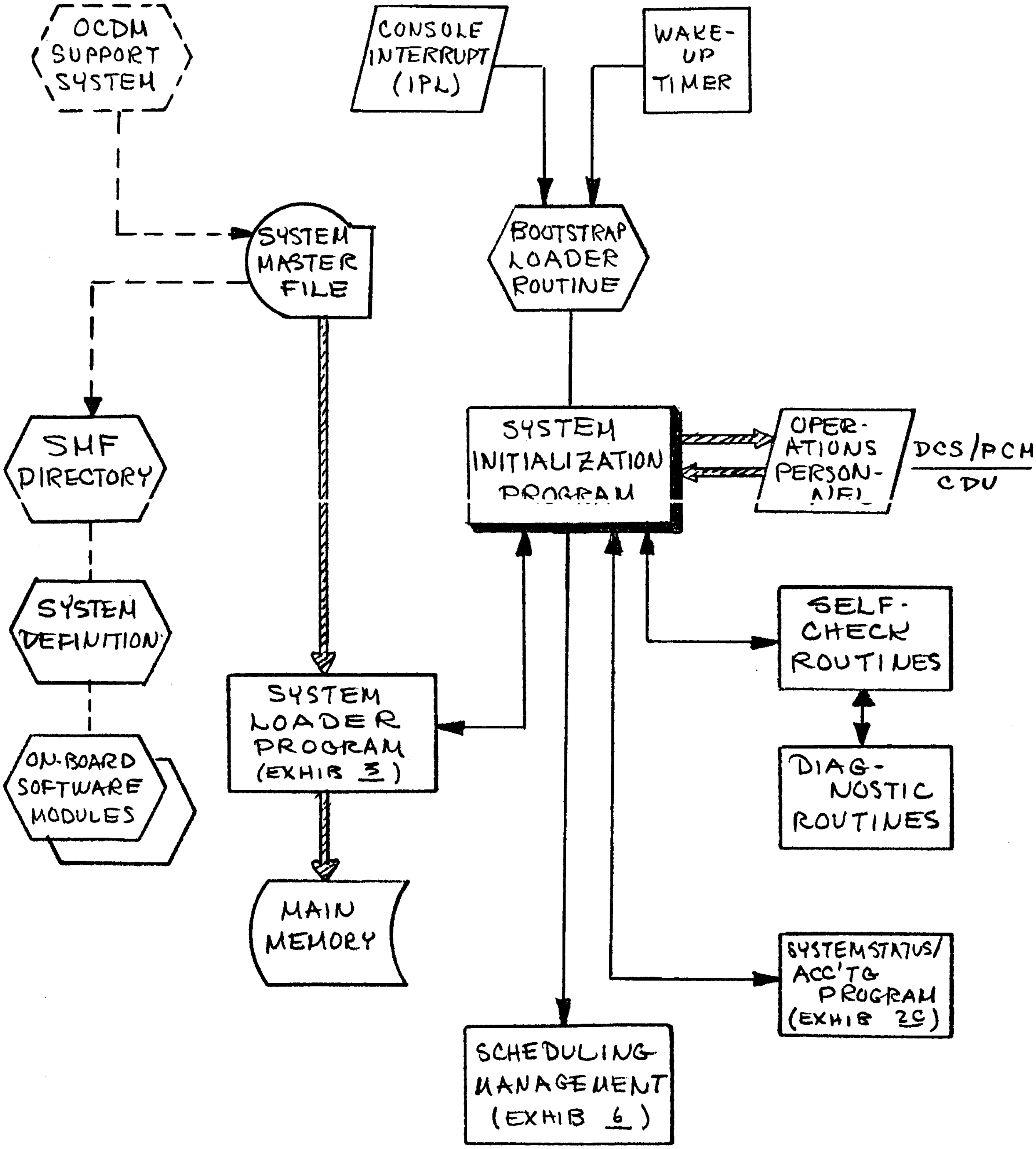


EXHIBIT 2a - SYSTEM MANAGEMENT (SYSTEM INITIALIZATION)

(3) As a result of an error or fault in the system

(4) In response to a macro, calling for such activity

Additionally, subprograms shall respond to certain hardware interruptions indicating system error conditions. In these cases, action may cause a reconfiguration of the system to attempt a "work around" to the source of error (e.g., failure of an input/output device or a device or a section of memory), or it may necessitate a complete shutdown. The particular response will be determined by reference to predefined severity codes. Exhibit 2b illustrates, functionally, the operation of this subprogram function.

c. System Shutdown Subprogram Functions

System shutdown subprograms will be entered as a result of completing normal programmed activities, or by direction of the operations personnel. When entered. the routines shall provide for system restart by reading necessary status and index information and program registers onto bulk storage, and making sure that the System Status and Configuration Dictionary reflects the latest configuration of the system prior to shutdown. They shall also issue messages to the operations personnel informing them of pending action, and finally initiate the applicable control signals to power-down the system. The operations personnel shall have the option to override this normal program sequence in the event of emergencies, in order to either effect or prevent a shutdown.

Routines implementing this system function shall also be responsible for setting the wake-up timer when appropriate, prior to system shutdown.

Exhibit 2c illustrates the functional operations of this subprogram.

d. Queued-Interruptions Processing Subprogram Functions

One of the functions of Interruption Management (see Section 3.1.2.6) is to recognize and queue nonimmediate interruption signals. The function of this subprogram shall be to examine this queue and call and/or release control to the routine(s) required to process the interruption signal(s). This subprogram will receive control on a regular basis as a function of Schedule Management (see Section 3.1.2.5).

Exhibit 2d illustrates the functional operation of this subprogram.

3.1.2.2 Function 2: Memory Management

The memory management function shall provide for dynamic allocation and recovery of storage units from the storage "pool."

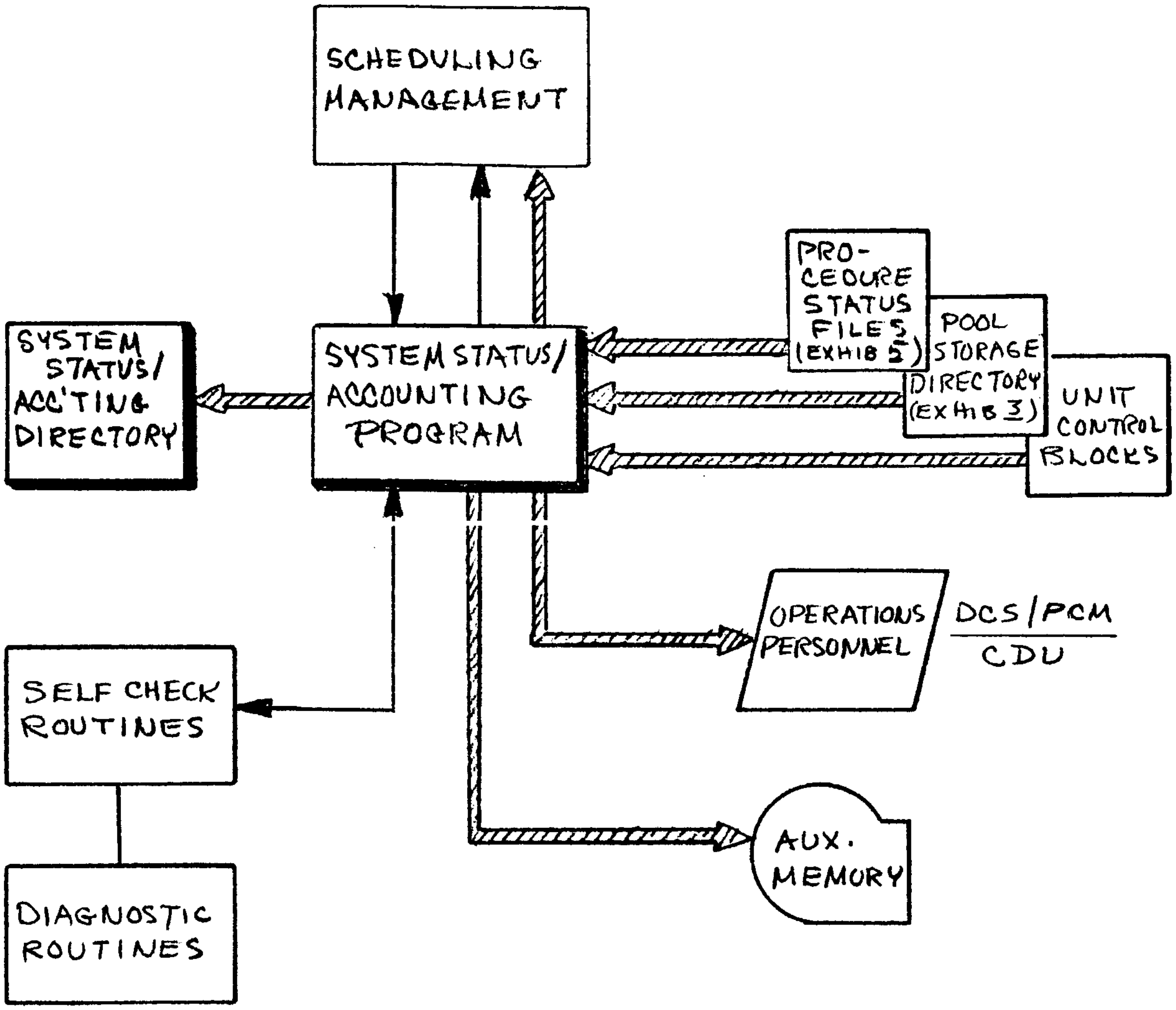

SCHEDULING MANAGEMENT
SYSTEM STATUS/ ACC'TING DIRECTORY
SYSTEM STATUS/ ACCOUNTING PROGRAM
PRO-CEDURE STATUS FILES (EXHIB 5)
POOL STORAGE DIRECTORY (EXHIB 3)
UNIT CONTROL BLOCKS
OPERATIONS PERSONNEL
DCS/PCM
CDU
SELF CHECK ROUTINES
DIAGNOSTIC ROUTINES
AUX. MEMORY

EXHIBIT 2b - SYSTEM MANAGEMENT (SYSTEM STATUS/ACCOUNTING)

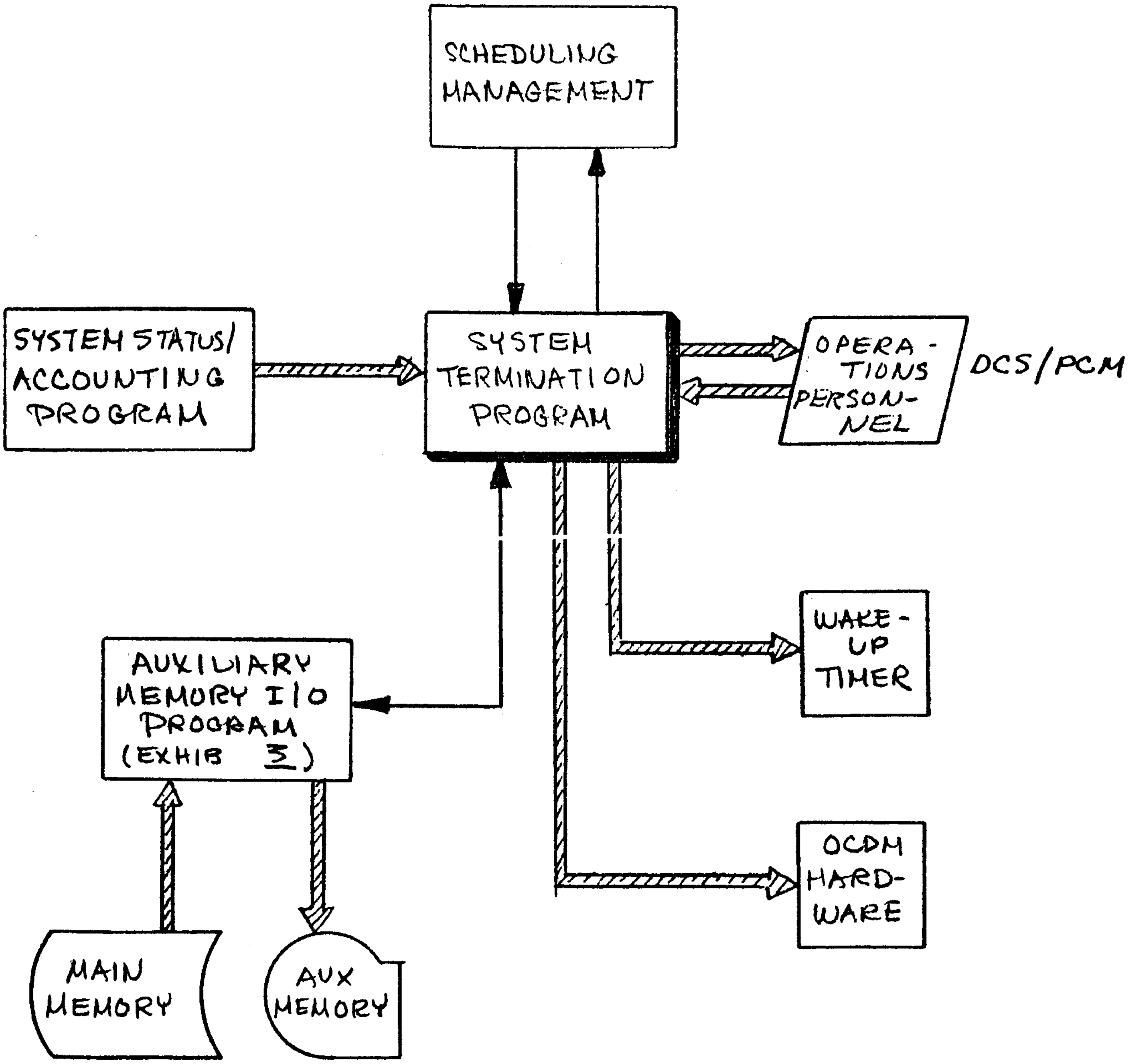

SCHEDULING MANAGEMENT
SYSTEM STATUS/ ACCOUNTING PROGRAM
SYSTEM TERMINATION PROGRAM
OPERATIONS PERSONNEL
DCS/PCM
WAKE-UP TIMER
AUXILIARY MEMORY I/O PROGRAM (EXHIB 3)
OCDM HARDWARE
MAIN MEMORY
AUX MEMORY

EXHIBIT 2c - SYSTEM MANAGEMENT (SYSTEM TERMINATION)

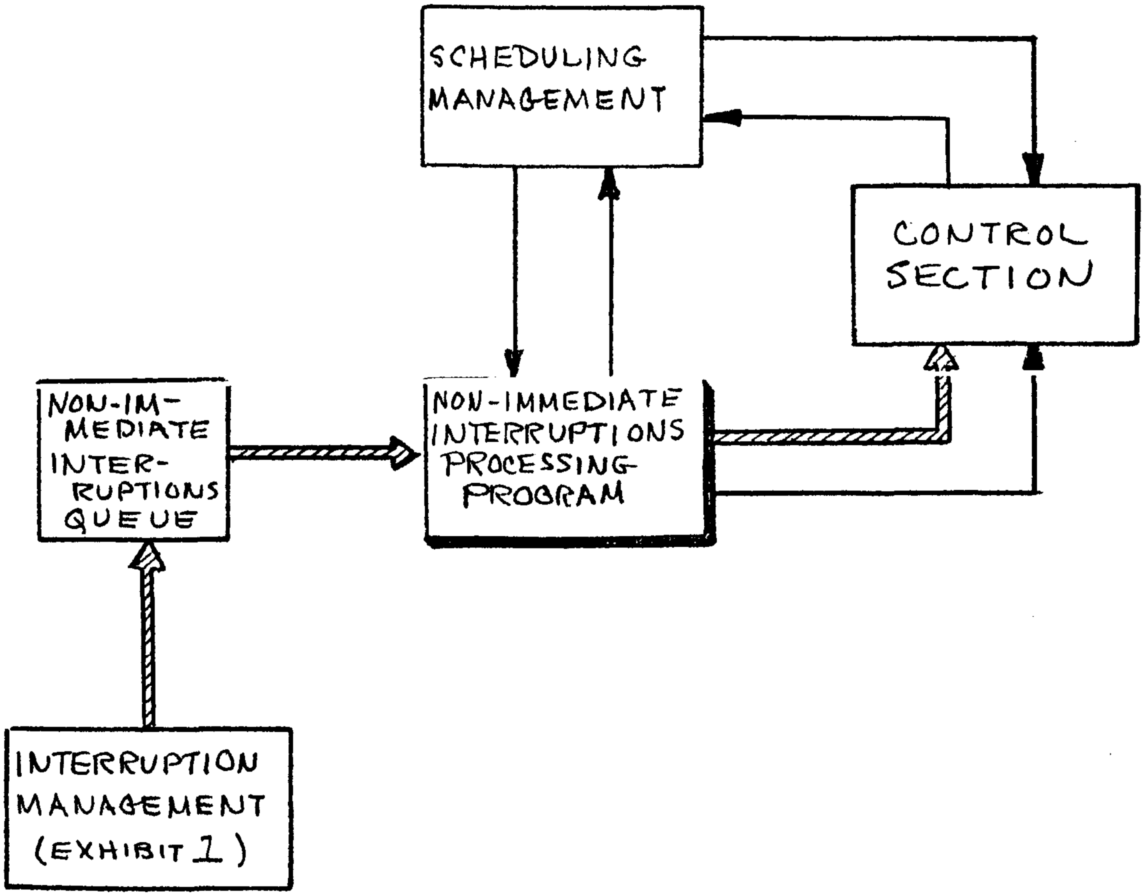

SCHEDULING MANAGEMENT
CONTROL SECTION
NON-IM-MEDIATE INTER-RUPTIONS QUEUE
NON-IMMEDIATE INTERRUPTIONS PROCESSING PROGRAM
INTERRUPTION MANAGEMENT (EXHIBIT 1)

EXHIBIT 2d - SYSTEM MANAGEMENT (QUEUED INTERRUPTION PROCESSING)

Program and data set transfers between main and bulk storage are also actions to be performed in conjunction with this function. Exhibit 3 illustrates the operations associated with this function.

3.1.2.2.1 Source and Type of Inputs for Memory Management

a. Functional inputs shall include, but not necessarily be limited to, the following types:

(1) Task-requests from other system programs and procedure programs to:

- o GET/RETURN storage units from/to the storage pool
- o Load/relocate load modules
- o READ/WRITE from/on auxiliary storage

(2) Request-compliance information from other system programs concerning the results of their processing of task-requests generated by Memory Management routines

(3) Description of the contents of main storage (Main Storage Directory--MSDY), which will be an array consisting of, but not necessarily limited to, the label, absolute location, and size of the following:

- o Every Procedure Control Block in core
- o Every resident Supervisory System program
- o Every nonresident Supervisory System program in core
- o Every data set in core (supervisory reference tables, UCB's, allocated storage units, etc.)

(4) Description of the contents of auxiliary storage (Auxiliary Storage Directory--ASDY), which will be an array consisting of, but not necessarily limited to, the label and relative position on the auxiliary storage media of every load module on the media

(5) Auxiliary storage device Unit Control Block(s), which will be prepared by the Support System and

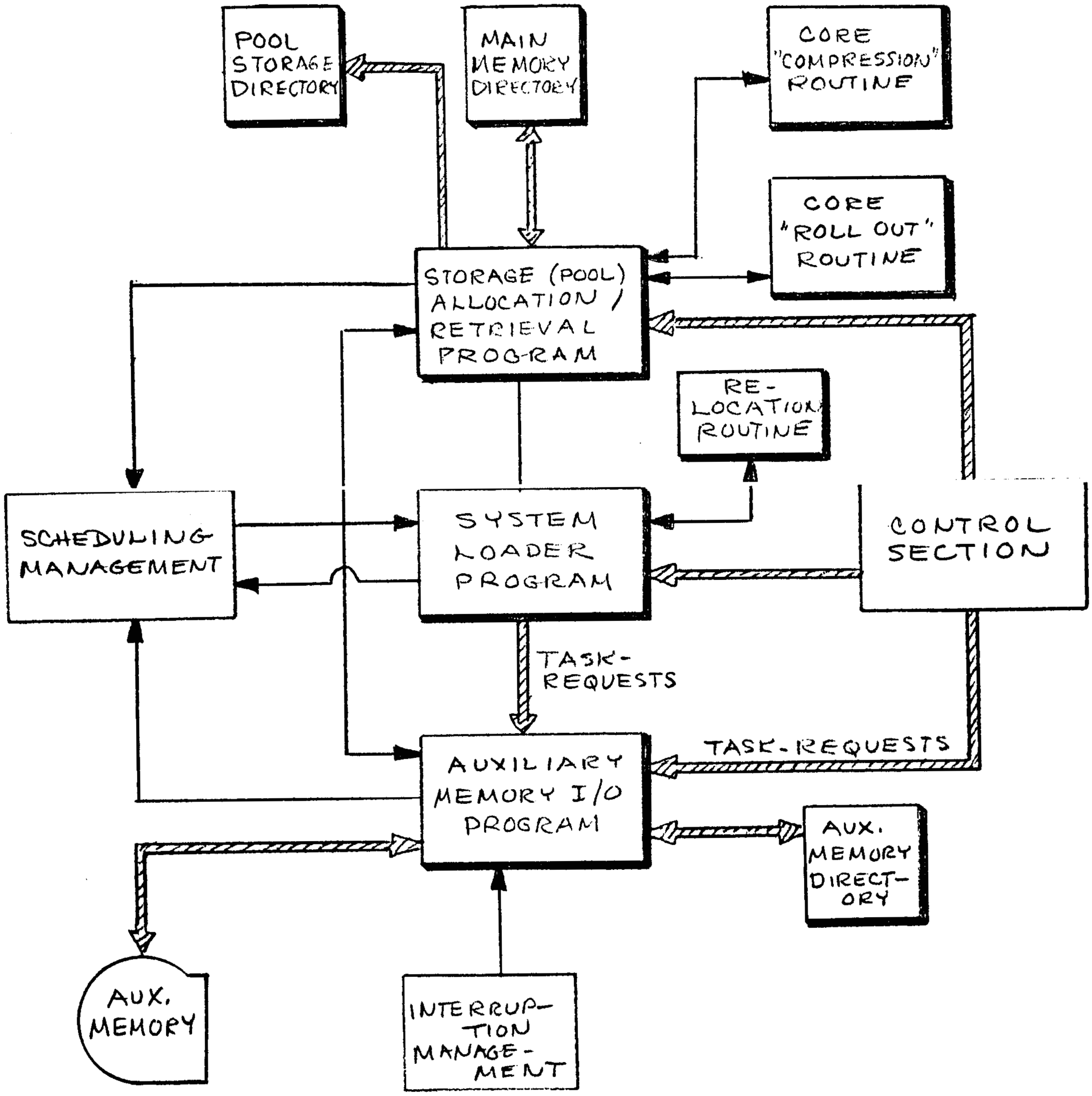


EXHIBIT 3 - MEMORY MANAGEMENT

will contain such control information necessary to access the device efficiently (i.e., channel address, device address(es), timing data, current point of device media relative to device read/write heads, etc.)

(6) Definition of the storage "pool" (Storage Pool Directory--SPDY), which will be an array indicating the location and size of all storage units in the "pool" and will designate which have been allocated and which are available for allocation

(7) Control signals and information transfers (load modules) from auxiliary storage device(s)

b. Sources of these inputs shall, respectively, include but not necessarily be limited to, the following:

(1) Other Supervisory System functions (Sections 3.1.2.2 through 3.1.3.10), as appropriate

(2) Other Supervisory System functions (Sections 3.1.2.2 through 3.1.2.10), as appropriate

(3) Maintained as a function of Memory Management

(4) Initially provided by the OCDMS Support System and thereafter maintained as a function of Memory Management

(5) Provided by the OCDMS Support System

(6) Initially provided by the OCDMS Support System and thereafter maintained as a function of Memory Management

(7) To be determined

c. Units of measure (to be determined)

d. Limits/range (to be determined)

e. Accuracy/precision (to be determined)

f. Arrival frequency (to be determined)

3.1.2.2.2 Destinations and Types of Output for Memory Management

a. Functional outputs shall include, but not necessarily be limited to, the following types:

(1) Allocation and transfer of control of storage "pool" units to various programs, and "loading" of specified program segments or data sets for various programs

(2) Modifications, as necessary, to the Main Storage Directory, Auxiliary Storage Directory, and Storage "Pool" Directory

(3) Status information for the system Status/Configuration Directory, which will include such information as will indicate the extent of utilization of main and auxiliary storage device workload, etc.

(4) Request-compliance information to other system programs; concerning processing of task-requests directed by them to Memory Management Routines

(5) Updated auxiliary storage device Unit Control Block(s) (e.g., new status, new position, etc.)

(6) Control signals and information transfers to auxiliary storage devices

b. Destinations for these outputs shall, respectively, include but not necessarily be limited to, the following:

(1) Managed as a function of Memory Management

(2) Managed as a function of Memory Management

(3) System Management function (3.1.2.1)

(4) Other Supervisory System functions (3.1.2.1, 3.1.2.3 through 3.1.2.10), as appropriate

(5) Managed as a function of Memory Management

(6) To be determined

c. Units of measure (to be determined)

d. Limits/range (to be determined)

e. Accuracy/precision (to be determined)

f. Output frequency (to be determined)

3.1.2.2.3 Information Processing for Memory Management

The Memory Management function shall be implemented by routines which perform as follows.

a. Loading

During System Initialization, the loader will be used implicitly by the System Initialization subprogram; otherwise it will be used explicitly by various system programs via scheduled execution of task-requests. When the loader receives control, it will first respond to all request-compliance information received by it from the Storage Management subprogram (indicating that a requested module has been read in, or that a requested module cannot be read in because of mechanical trouble), thereby eliminating redundant load requests. Actions involved in the loader function include the following:

(1) Adjusting location-dependent quantities within a module, using relocation information provided in the module's header file

(2) Updating the Main Storage Directory with the label, size, and location of the loaded module

(3) Issuing request-compliance information to the program which requested the module. This shall include, as a minimum, the label and location of the module. If a module cannot be located, the information will indicate the cause (e.g., tape read error, undefined module)

The loader shall next select one task-request from its queue and process it. This will involve the following:

(1) Obtaining storage space for the module, if none has been specified by the task-request

(2) Sending a task-request to the auxiliary Storage Management subprogram for the appropriate module. This request will specify the label of the module, the location into which it should be read, and a return pointer to the loader, indicating where appropriate request-compliance information is to be stored

(3) Return of CPU control to the Scheduling Management program

The loader will also be capable of accepting and queuing task-requests from other programs.

b. Main Storage Management Subprogram Functions

The main storage management CPC's will handle necessary interface between main memory and other elements of the system requesting use of it. These CPC's will be entered as a function of Scheduling Management, and will be capable of the following operations:

(1) Accepting, queuing, and responding to task-requests from other system programs concerning the allocation, return, or transfer of storage "pool" units in main memory

(2) Allocating storage areas from a predefined storage pool (defined at system generation time by the OCDMS Support System), to the requesting program. If the storage pool does not contain the required storage area, this subprogram will perform a limited amount of core compression (packing the already allocated storage areas in lowest available core to make use of any gaps of unallocated core between them). This will require use of the relocation capabilities of the loader

(3) Returning allocated storage area to the pool either on request from an applicable program or on the basis of a supervisory cleanup activity

(4) Updating of the Storage Pool Directory

(5) Providing request-compliance information to other applicable programs indicating either the location of an allocated storage area, or that there is none available at the time

(6) Providing the System Management function with information concerning the utilization of main and auxiliary storage, and the status of main and auxiliary storage hardware

c. Auxiliary Storage Management Subprogram Functions

The routines implementing the functions of auxiliary Storage Management shall be capable of the following operations:

(1) Accepting, queuing, and responding to task-request from other programs

(2) Transmitting/receiving control signals and data transfers to and from auxiliary storage devices

(3) Maintaining an Auxiliary Storage Dictionary indicating relative location of logical records and amount of storage space available

(4) Providing request-compliance information to other applicable programs indicating the result of the processing of their requests

(5) Reformatting data on auxiliary storage devices in order to utilize the devices more efficiently

(6) Accessing the appropriate Unit Control Block(s) for information required to communicate with the auxiliary storage device(s)

(7) Updating information, as necessary, in the Unit Control Block(s)

3.1.2.3 Function 3: Language Interpreter

The language interpretation function shall be implemented to provide translation and execution of a limited set of on-line statements, commands, and other symbolic codes entered as consequence of other major CPCEI functional operations. Associated with this function are the requirements for routines which shall perform checkout and data management action sequences for all procedural source inputs to the system (both for supervisory control sections and macro-instructions and/or macro-calls from application programs). This function is illustrated in Exhibit 4.

3.1.2.3.1 Sources and Types of Inputs for the Language Interpreter

a. Functional inputs shall include, but not necessarily be limited to, the following types:

(1) Functions, entered on-line, to be interpreted, including:

- o Basic elements (letters, digits, identifiers, numbers, and strings)
- o Expressions (variables, arithmetic and Boolean operators; and Computer Interface Unit and Control/Display Unit function key codes)
- o Statements (procedural, conditional, assignment, and GO-TO types)--to the extent of parameter modification

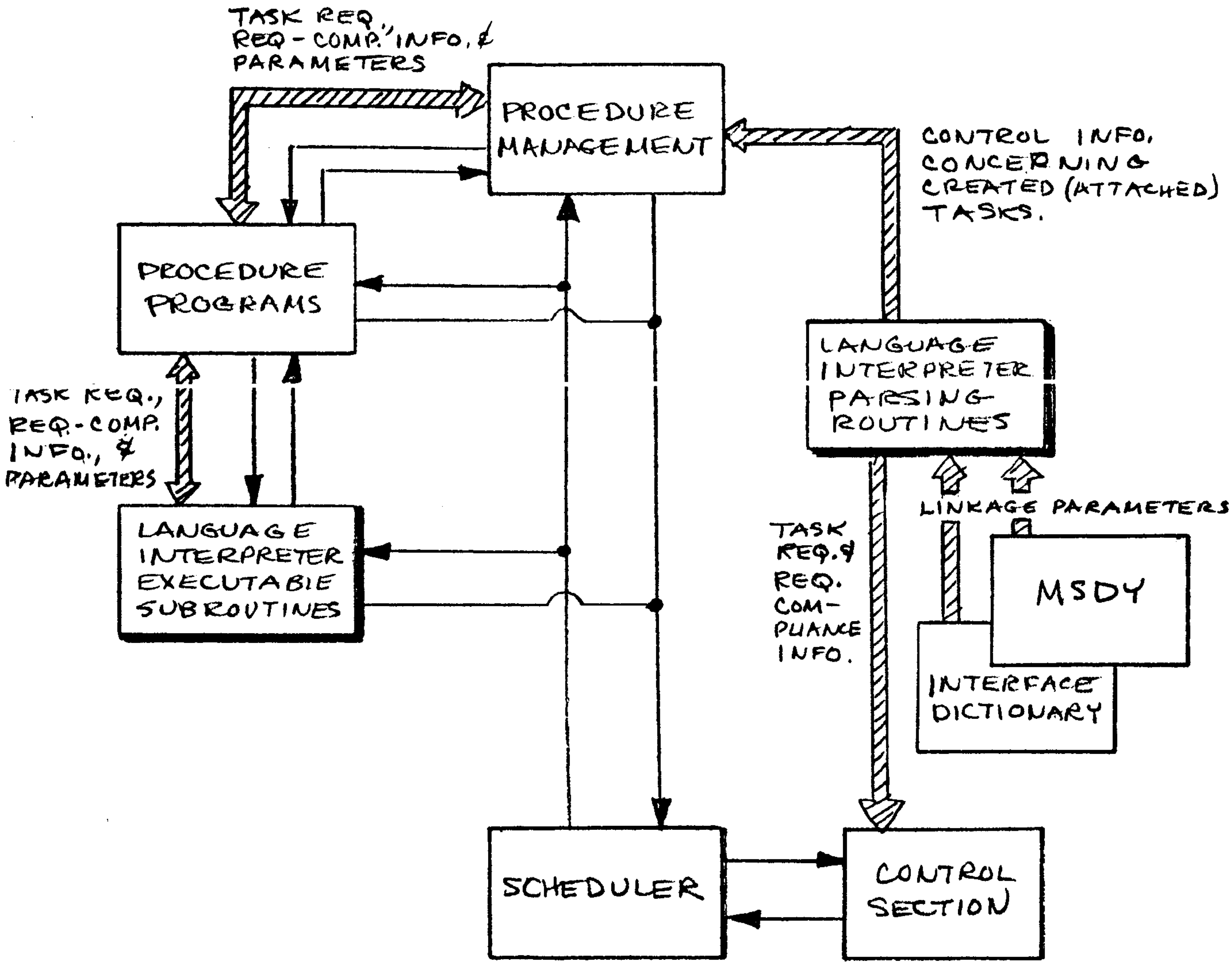

TASK REQ., REQ-COMP. INFO, & PARAMETERS
PROCEDURE MANAGEMENT
CONTROL INFO. CONCERNING CREATED (ATTACHED) TASKS.
PROCEDURE PROGRAMS
TASK REQ., REQ-COMP. INFO., & PARAMETERS
LANGUAGE INTERPRETER PARSING ROUTINES
LINKAGE PARAMETERS
LANGUAGE INTERPRETER EXECUTABIE SUBROUTINES
TASK REQ. & REQ. COM-PLIANCE INFO.
MSDY
INTERFACE DICTIONARY
SCHEDULER
CONTROL SECTION

EXHIBIT 4 - LANGUAGE INTERPRETER

(2) Task-requests from procedure programs: the content of these will depend entirely on the nature of the particular executable subroutine (see 3.1.2.3.3b, below) to which they are directed

(3) Request-compliance information pertaining to task-requests generated by the Language Interpreter

(4) Linkage parameters concerning the locations of executable subroutines and Unit Control Blocks--obtained from the Main Storage Directory

(5) Addressing, switching, and other control parameters (bit masks, etc.), and an index of labeled procedure steps, substeps, and other procedure hold-points--obtained from an On-Line Interface Directory, supplied by the Support System

b. Sources of these inputs shall, respectively, include but not necessarily be limited to, the following:

(1) Console Communication and Uplink Communication functions (3.1.2.7 and 3.1.2.9)

(2) Procedure Management function (3.1.2.4) and procedure programs supplied by OCDMS Support System

(3) Other Supervisory System functions; probably limited to Memory Management (3.1.2.2)

(4) Memory Management (3.1.2.2)

(5) OCDMS Support System

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Arrival frequency (to be determined)

3.1.2.3.2 Destinations and Types of Output for the Language Interpreter

a. Functional output shall include, but not necessarily be limited to the following types:

(1) Tasks (substeps, steps), created and attached to existing procedures, in the form of linkage and data transfer macro-instructions, and generated command/control routine calling sequences

(2) Supervisory System executive table entries

(3) Task-requests, as necessary, to other system functions (probably limited to Memory Management)

(4) Definition of data control blocks, task control blocks, and the linkage necessary for access to given fields within these blocks

b. Destinations of these outputs shall, respectively, include but not necessarily be limited to, the following:

(1) Procedure Management function (3.1.2.4)

(2) System Management function (3.1.2.1)

(3) Other Supervisory System functions, but probably limited to Memory Management (3.1.2.2)

(4) Procedure Management function (3.1.2.4)

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Output frequency (to be determined)

3.1.2.3.3 Information Processing for the Language Interpreter

The programmed processes of the Language Interpretation function shall essentially consist of two sets; those coded functions implementing the interpretation capability (thereafter referred to as the Interpreter), and those implementing the verbs of the symbolic language, the Execution subroutines.

a. Interpreter

The Interpreter will receive the expressions, statements, and basic elements from the various sources. When given CPU control, as a function of Schedule Management, it will perform the operations necessary to translate these inputs into a form (calling sequences, executable code sets) that may be handled by the Supervisory System in the same manner as the procedure programs which are produced by the Support System. As a result, the following will be true.

(1) The language processing criteria shall be in accordance with Paragraph 3.1.2.1.3, Reference 2.2.2 (the OCDMS Support System specification).

(2) The interpreter expression decomposition logic shall be in accordance with Paragraph 3.1.2.1.3.3, Reference 2.2.2.

(3) Supervisor parameter and table update generation logic shall be in accordance with Paragraph 3.1.2.1.2.4, Reference 2.2.2.

(4) The interpreter code conversion and execution sequence shall employ, to the maximum extent possible, the same algorithms and appropriate computer coding developed in accordance with Paragraphs 3.1.2.1.2.5 and 3.1.2.1.3.6, Reference 2.2.2.

The requirement for fairly extensive man-machine communication in all but the most limited sense (e.g., specification, of an operation, that can be accomplished by pressing a single function key), of action by the Interpreter will require that, in general, the Interpreter will be utilized by the operations personnel in the semiautomatic mode. Because of this, much of the data and programmed processes of the Interpreter could be kept in auxiliary storage rather than in main memory.

As stated in Section 3.1.2.3, the set of expressions, statements and basic elements will be limited (depending on the amount of storage available, the sophistication built into this overall function, and the requirements for such by the nature of each mission). This set may be altered from mission to mission.

b. Execution Subroutines

These routines will serve to carry out the operations specified by the expressions of the procedure programs (i.e., they function as the verbs of the symbolic language in which the procedure programs are written). They will accept task-requests from the procedure programs and receive control as a function of Schedule Management. (In some cases, where the routine is sufficiently short, control may be received directly from a procedure program.) Wherever possible, they will be designed so as to be completely reenterable. In certain cases, those that cannot be made reenterable may have to be duplicated in order for the system to handle cyclic programs (see Schedule Management, 3.1.2.5). This will be entirely dependent on the nature of the procedure programs.

This set of routines will include, but not necessarily be limited to, ones of the following nature.

- Discrete out
- Discrete in
- Analog out
- Analog in
- Read internal parameter
- Limit test
- Program branch
- Display
- Print
- Record time
- Set delay
- Execute subprogram
- Transmit task-request
- Compute

3.1.2.4 Function 4: Procedure Management

The processes implementing Procedure Management shall provide supervisory services and controls for the procedure programs preprocessed by the OCDMS Support System. Program schedules, procedure program intercommunication (for transfer of data, control information, etc.), and the coordination of all mechanics prerequisite to interpretation and execution of application tasks shall be within the jurisdiction of this system function.

Exhibits 5a and 5b illustrate the operations involved in this function.

3.1.2.4.1 Source and Types of Inputs for Procedure Management

a. Functional inputs shall include, but not necessarily be limited to, the following types.

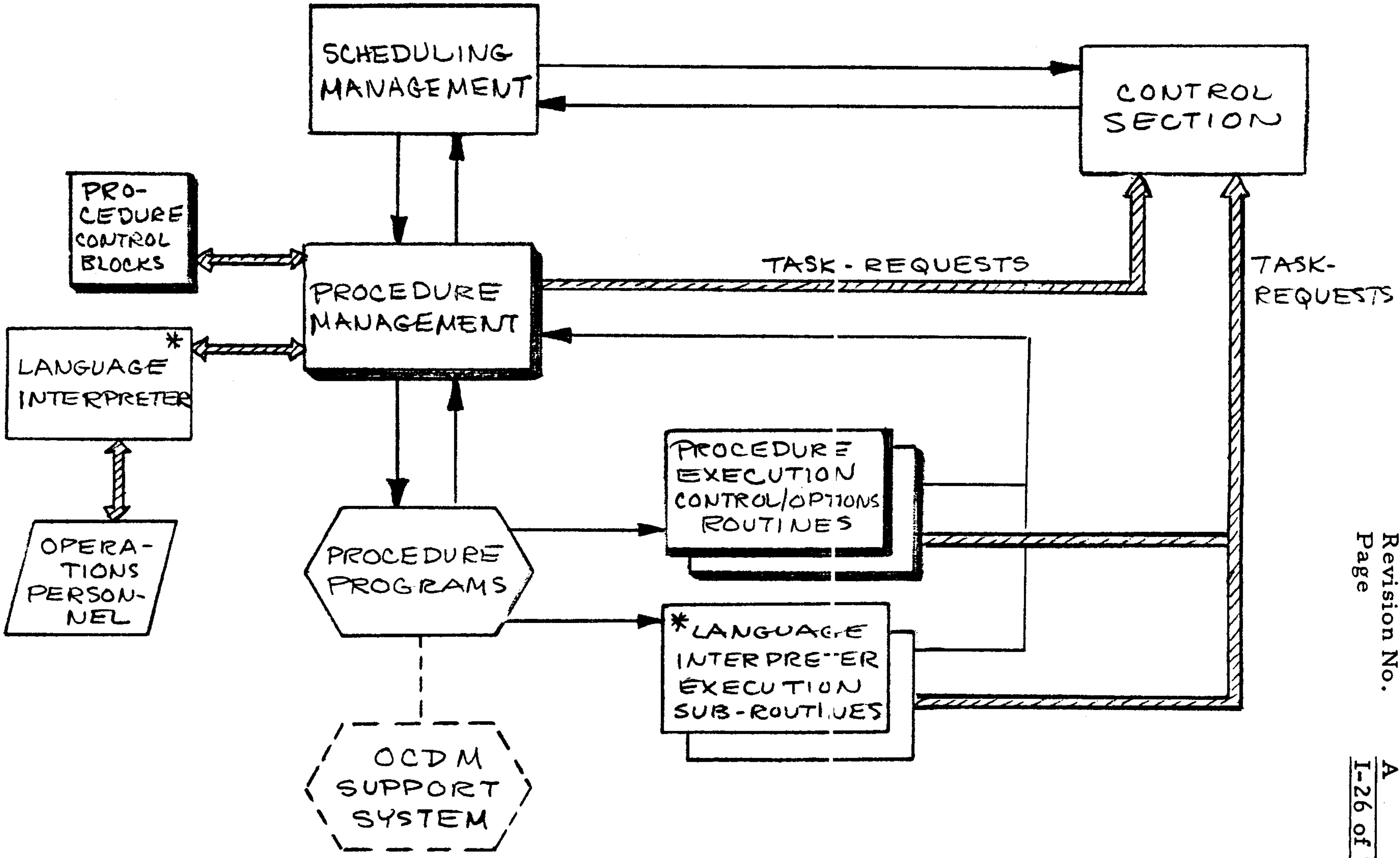


EXHIBIT 5a - PROCEDURE MANAGEMENT

<table>
<tr><td colspan="2">PROCEDURE I/O</td></tr>
<tr><td>SEGMENT I/O</td><td>PROCEDURE STATUS</td></tr>
<tr><td colspan="2">SEGMENT ENTRY POINT</td></tr>
<tr><td colspan="2">SEGMENT RE-ENTRY POINT (CURRENT)</td></tr>
<tr><td colspan="2">TIMING PARAMETER (FOR CYCLIC PROCEDURES)</td></tr>
<tr><td colspan="2">SEGMENT COMPATIBILITY CODE</td></tr>
<tr><td>POINTER TO COMMON</td><td>SIZE OF COMMON</td></tr>
<tr><td colspan="2">SEVERITY CODES</td></tr>
<tr><td colspan="2">POINTER(S) TO UNIT CONTROL BLOCKS } VARIABLE IN SIZE</td></tr>
<tr><td>POINTER(S) TO ALLOCATED STORAGE UNITS } VARIABLE IN SIZE</td><td>SIZES OF ALLOCATED STORAGE UNITS } VARIABLE IN SIZE</td></tr>
<tr><td colspan="2">SEGMENT RELOCATION DICTIONARY } VARIABLE IN SIZE</td></tr>
</table>

EXHIBIT 5b - PROGRAM CONTROL BLOCK

(1) Procedure programs, consisting of calling sequences and sets of executable code as described by Paragraph 3.1.2.1.3.5 and Section 3.1.2.3 of Reference 2.2.2, OCDMS Support System CPCEI

(2) Procedure Control Blocks, consisting of sets of control data relating to the status and identification (Exhibit 5b illustrates a PCB and its contents)

(3) Procedure Schedule File (precedence list), consisting of an array indicating the conditions (i.e., range time, vehicle position, hardware, status, preceding programs, etc., as applicable) prerequisite for turn-on and execution of a given procedure. This information will be prepared by the Support System and updated on-line as necessary, as a function of Procedure Management

b. Sources for these inputs shall, respectively, include, but not necessarily be limited to, the following:

(1) OCDMS Support System

(2) OCDMS Support System and Language Interpreter function (3.1.2.3)

(3) OCDMS Support System

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Arrival frequency (to be determined)

3.1.2.4.2 Destination and Type of Output

a. Functional outputs shall consist of, but not necessarily be limited to, the following:

(1) Procedure status data to the Procedure Control Blocks and Scheduling Management function, indicating whether the procedure program is active or inactive; if the procedure is active, the current procedure step identifier will be maintained

(2) Messages to the operations personnel involving query and response adaptive activities and option

selection, prompting messages to operations personnel concerning activation of procedures

b. Destinations of these outputs shall, respectively, include, but not necessarily be limited to, the following:

(1) Procedure Management and Schedule Management functions (3.1.2.4 and 3.1.2.1)

(2) Console Communication and Downlink Communication Functions (3.1.2.7 and 3.1.2.10)

c. Units of measure (to be determined)

d. Limits/range (to be determined)

e. Accuracy/precision (to be determined)

f. Output frequency (to be determined)

3.1.2.4.3 Information Processing for Procedure Management

The processes implementing Procedure Management shall perform logical information actions defined by control routines that perform the following activities:

a. Scheduling/prompting routines shall periodically monitor the system for the existence of those conditions specified by the Procedure Schedule File as prerequisites for procedure turn-on/turn-off. When conditions are satisfied, a prompting message shall be issued to the operations personnel identifying the procedure to be activated. The operations personnel will be given the option of activating or delaying execution of the procedure.

b. Procedure call/activation: If it shall be decided to initiate execution of a given procedure, then routines will perform the following operations:

(1) Determine if the present scheduling load will allow execution of that procedure; communications with the operator in conjunction with Paragraph 3.1.2.7 shall provide a choice of preferred activities in the event of schedule conflicts

(2) "Call" the procedure, via the loader, and initialize it; the latter action involves updating the applicable Procedure Control Block, generating an appropriate SAP Table element, and transferring it to the Schedule Management routine to be posted in the SAP Table

c. Execution control and accounting: During execution of a procedure, routines shall perform the following operations:

(1) Manage the Procedure Control Blocks; that is, update, as necessary, that information contained therein, pertaining to allocated-storage locations, current position within procedure (point of resumption), and program status (active, waiting, inactive)

(2) Recognize specified hold points (end of step, substep, block, segment, etc.)

(3) Recognize the requirement for, and call for, additional program segments

(4) Receive control as a function of scheduling management and subsequently activate the proper procedure

d. Execution options: During initialization and at predefined hold points of a procedure, the operations personnel shall be given the opportunity to execute the following options:

(1) Repeat, insert, and/or delete a step within a procedure

(2) Modify selected parameters within a step (number of repetitions, limits, etc.)

3.1.2.5 Function 5: Schedule Management

Schedule management is the OCDMS Supervisory System function that sequentially distributes available CPU processing time among those programs having tasks to perform. The programmed processes of Schedule Management shall provide the mechanics to commutate through a list (SAP Table) of control sections available in the system, allocating to each in turn, as required, a unit of CPU control to perform its task. This unit will not exceed a specified maximum value (to be determined during detailed development of the system). This constraint will be effected by programming convention (i.e., design of each individual control section), not by hardware timer interruption. Exhibit 6a illustrates the operation of this function.

3.1.2.5.1 Source and Type of Inputs for Schedule Management

a. Functional inputs shall include, but not necessarily be limited to, the following types:

(1) Scheduling algorithm parameter table: This will be prepared by the OCDMS Support System at system

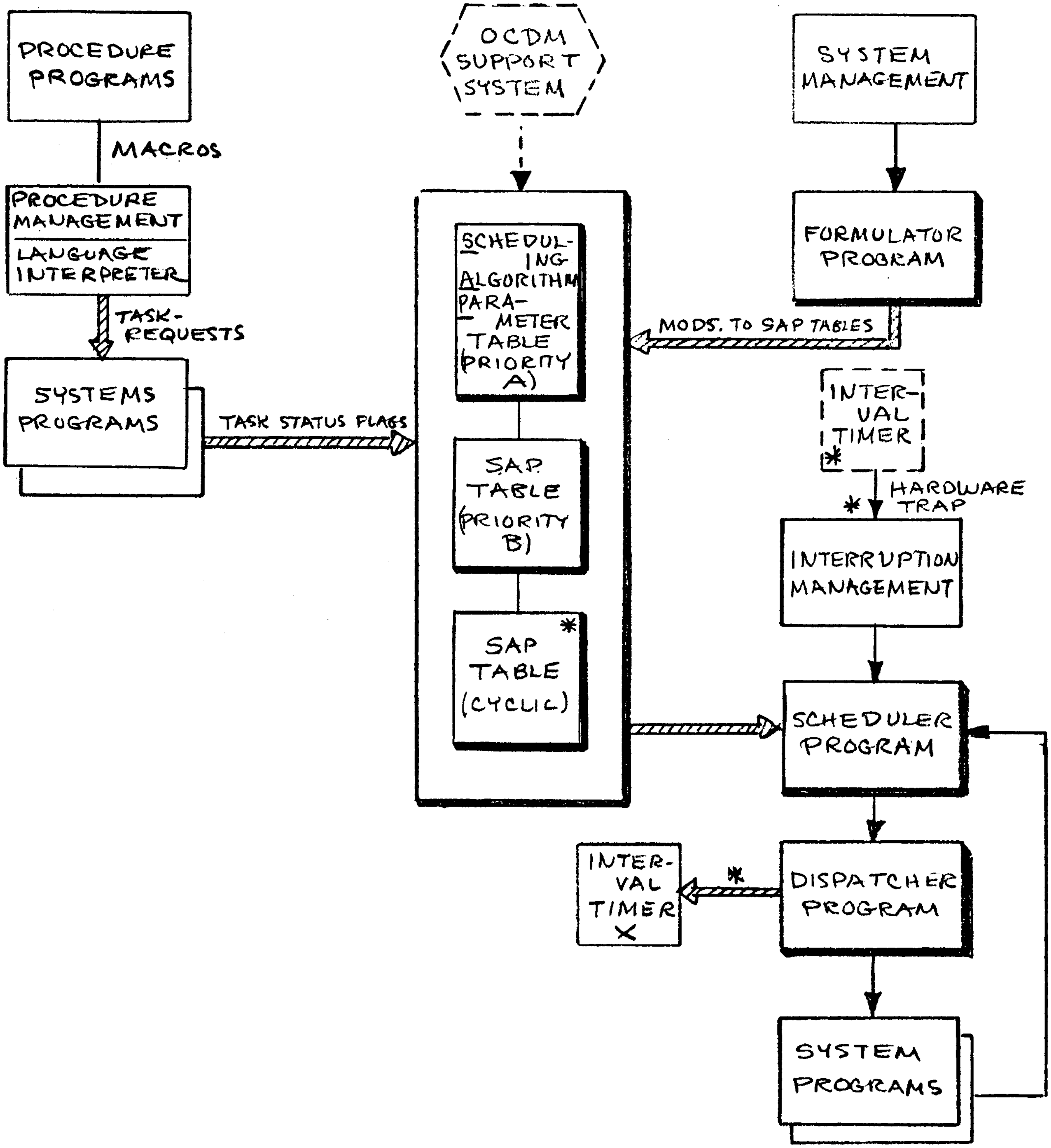


EXHIBIT 6a - SCHEDULING MANAGEMENT

generation time. It will consist of an array of Scheduling Algorithm Parameters (SAP's), and will provide the means by which the Schedule Management function communicates with the operational programs. Presence on this table, of a related SAP, will be a necessary condition for a control segment (logical portion of a procedure or system program) to receive an increment of CPU control. There will be a unique SAP for each system program (or routine) in core that receives control as a function of Schedule Management. These SAP's will contain at least the following parameters:

o A task-status flag that will indicate whether or not the related program has an active task to perform

o A pointer (absolute location) to the related program

o The label associated with the program

o A time-value field for those SAP's pertaining to time-dependent, cyclic programs

(2) Control signals (task-status flag settings): These will be provided by the appropriate programs, which will set/reset the task-status flag within their related SAP to indicate that they have or do not have an active task to perform; therefore, should or should not be given an increment of CPU control.

b. Sources of these inputs shall, respectively, include, but not necessarily be limited to, the following:

(1) OCDMS Support System

(2) Other, scheduled, Supervisory System functions (3.1.2.1 through 3.1.2.4 and 3.1.2.6 through 3.1.2.10)

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Arrival frequency (to be determined)

3.1.2.5.2 Destinations and Types of Outputs for Schedule Management

a. Functional outputs shall consist of, but not necessarily be limited to, the following types:

(1) Modifications to the SAP Table

(2) SAP address: This will be the absolute address of a given SAP and will be transmitted to its related program (so that it may be accessed by it) when the Schedule Management function has had occasion to create a new SAP

(3) Control signals to the interval timers: The general format for an input/output instruction word for the computer interface unit (CIU) may be found in Reference 2.3.5 (OCDMS: Design Supplement, TAM No. 171-3)

(4) Transfer of CPU control to a selected program: This is a functional operation (a branch) and does not involve any physical "output" by the Schedule Management function

b. Destinations of these outputs shall, respectively, include but not necessarily be limited to the following:

(1) Managed implicitly as a function of Schedule Management

(2) Procedures created on-line (via the Language Interpreter)

(3) To be determined

(4) Scheduled, Supervisory System functions (control segments)

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Output frequency (to be determined)

3.1.2.5.3 Information Processing for Schedule Management

The set of programmed processes making up the Schedule Management function can be considered as two major subprograms: the Formulator and the Sequencer subprograms.

a. The Formulator subprogram shall contain the entry points within the Schedule Management program to which other sections of the system will control after having been previously given it as a function of Schedule Management. One group of entry points shall exist for return from programs "periodic" (those requiring control at regular intervals), and another group shall be available for "nonperiodic" programs. For mission requirements dictating both priority and normal scheduling, additional entry point groups may be defined:

(1) "Periodic" entry: When a "periodic" program returns control to the Formulator, it shall indicate to the Formulator that an interval timer must be set with the appropriate time value, in order that the program may receive control within the necessary time period

(2) SAP Table modifications: These modifications will consist of the creation, in the SAP Table, of a new SAP. They will be caused by (1) the formulation, by the operations personnel via the Language Interpreter function, of a task to be executed on a scheduled basis, distinct from an active procedure, and (2) the demand by the operations personnel to increase the rate of execution of a particular program (e.g., a PCM dump routine, see 3.1.2.10, Downlink Communication). The latter would result in the duplication of the existing SAP, so that during one SAP Table polling cycle, that program would then receive two increments of CPU control, instead of only one.

b. The Sequencer subprogram shall process inputs and perform operations as follows:

(1) It shall use the periodic status information saved from the previous sequences to determine the point at which to resume polling the SAP Table

(2) Poll the table unit until it finds an SAP with its task-status flag set (see note below)

(3) Save the cycle status (point of resumption)

(4) Set the proper interval timer with maximum time allowed for task execution

(5) Dispatch control to the program referenced by the selected SAP. The dispatching interface with operational programs shall generally be embodied as a macro-instruction. The action called for shall be deferred rather than immediate. For this reason, a "tracer" mechanism shall be provided to determine the status of action items.

Note: The polling algorithm and the structure of the SAP Table will be such as to reflect a difference in priority level between sets of SAP's. Those residing at a higher priority level would be polled more frequently than those at a lower level. The exact relation between priority levels and the content of each is most properly determined during a higher phase of system development than the document covers.

Exhibit 6b illustrates the use of the SAP element, which is the key to the intelligence mechanism of Schedule Management.

"Periodic" programs will not be handled in the same manner as nonperiodic programs will be (i.e., not on a strictly scheduled basis). Rather, they will be given CPU control on an interrupt basis; the mechanism for which will be one or more interval timers set to appropriate values.

3.1.2.6 Function 6: Interruption Management

The Interruption Management function shall provide the CPCEI capability to recognize the occurrence of, and to identify, the action to be taken in regard to hardware/software interruption signals. Interruptions shall serve either to notify the system that a requirement for immediate action exists or to inform the system that a particular asynchronous event requires future action. OCDMS interruptions will also indicate that data from a low-rate input source is available, that a buffered input/output operation has been completed, that a preselected period of time has elapsed, that some unusual external condition exists, or that an unusual internal condition exists. Exhibit 7 illustrates the operations associated with this function.

3.1.2.6.1 Source and Type of Inputs for Interruption Management

a. Functional inputs shall include, but not necessarily be limited to, interruption signal words. These will be computer words containing binary codes corresponding to the particular interruption signal recognized. They will be provided to the Interruption Executive in standard computer memory locations in the case of hardware-induced branching, and in calling sequences in the case of programmed instruction-induced branching. These signal codes will reflect the occurrence of one or more of the following types of interruptions:

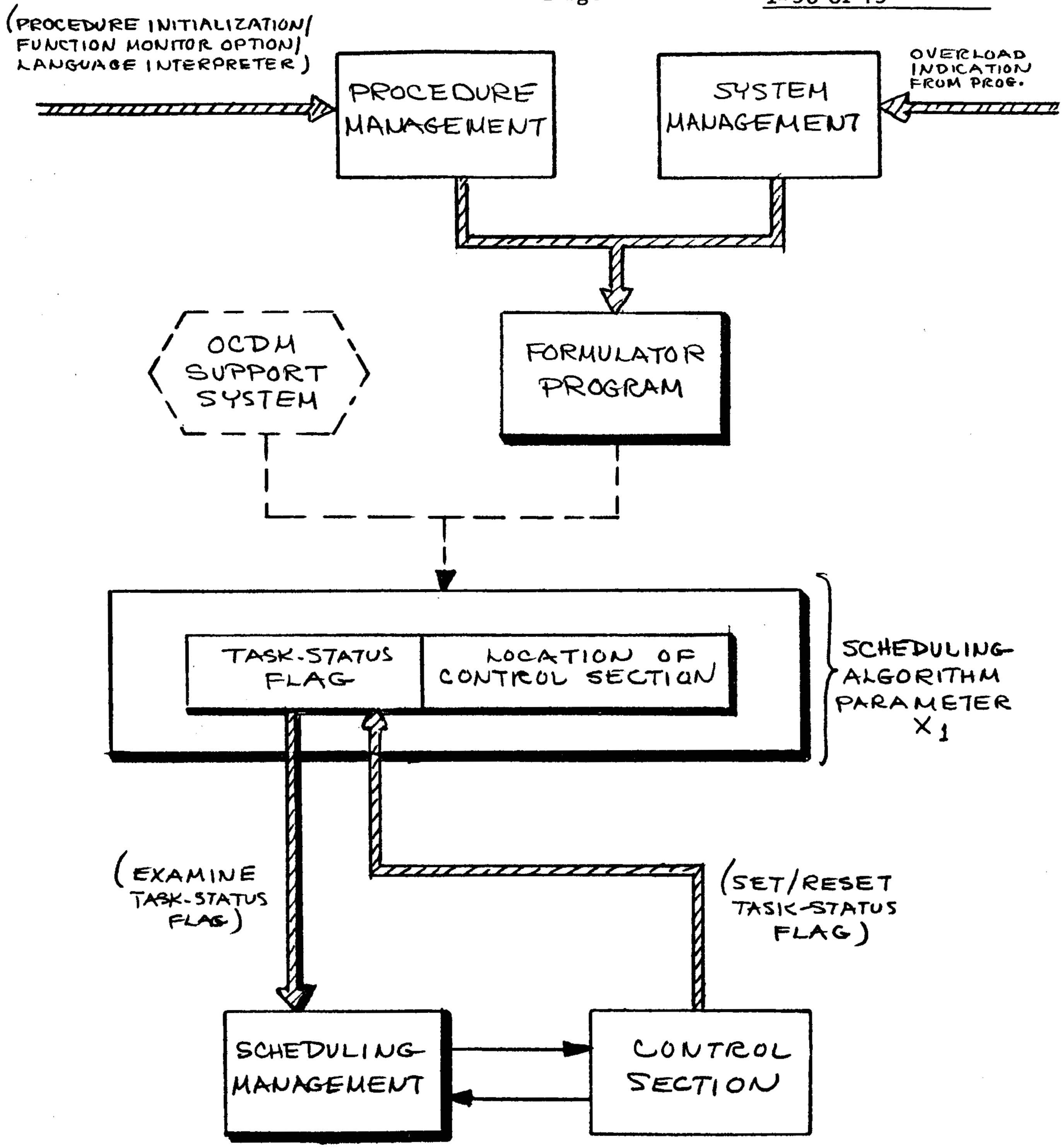

(PROCEDURE INITIALIZATION/
FUNCTION MONITOR OPTION/
LANGUAGE INTERPRETER)
PROCEDURE MANAGEMENT
SYSTEM MANAGEMENT
OVERLOAD INDICATION FROM PROG.
OCDM SUPPORT SYSTEM
FORMULATOR PROGRAM
TASK-STATUS FLAG
LOCATION OF CONTROL SECTION
SCHEDULING ALGORITHM PARAMETER X_1
(EXAMINE TASK-STATUS FLAG)
(SET/RESET TASK-STATUS FLAG)
SCHEDULING MANAGEMENT
CONTROL SECTION

EXHIBIT 6b - SAP MECHANISM

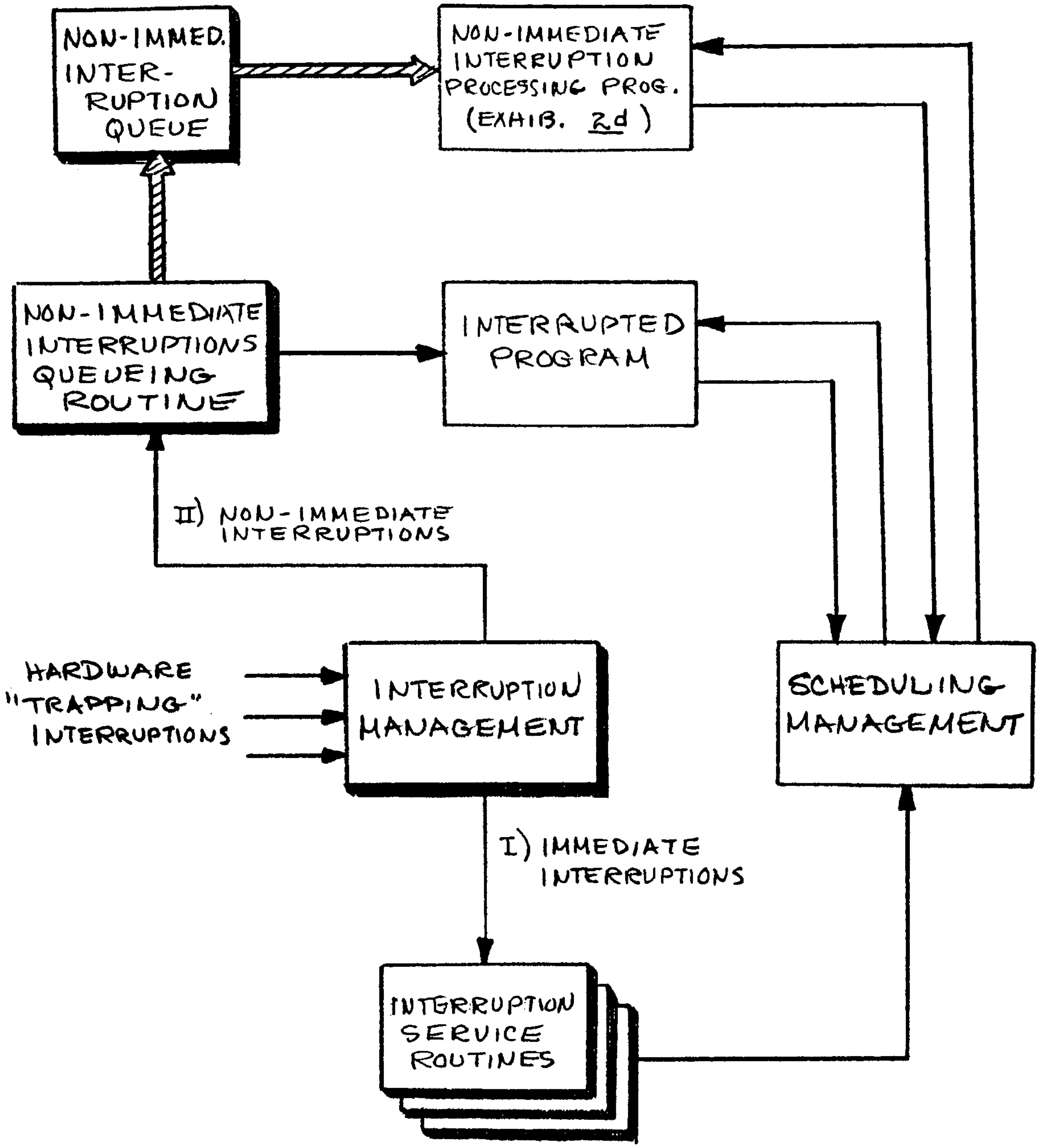

NON-IMMED. INTER-RUPTION QUEUE
NON-IMMEDIATE INTERRUPTION PROCESSING PROG. (EXHIB. 2d)
NON-IMMEDIATE INTERRUPTIONS QUEUEING ROUTINE
INTERRUPTED PROGRAM
II) NON-IMMEDIATE INTERRUPTIONS
HARDWARE "TRAPPING" INTERRUPTIONS
INTERRUPTION MANAGEMENT
SCHEDULING MANAGEMENT
I) IMMEDIATE INTERRUPTIONS
INTERRUPTION SERVICE ROUTINES

EXHIBIT 7 - INTERRUPTION PROCESSING

(1) Input/output: fault or status conditions of input/output devices on channels

(2) Program overflow, underflow, etc.

(3) Supervisor call: instruction(s) used to transfer control to the Supervisory System

(4) External: fault or status conditions of internal timer, console interrupt key, and external hardware units

(5) Machine check: computer hardware fault

(6) Priority conditions

b. Sources (to be determined)

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Arrival frequency (to be determined)

3.1.2.6.2 Destinations and Types of Outputs for Interruption Management

a. Functional outputs shall consist of, but not necessarily be limited to, the following:

(1) Interrupts enabled

(2) Interrupts allowed

(3) Interrupts disabled

(4) Interrupts not allowed

(5) Interruptions completely serviced

(6) Interruptions partially serviced

b. Destinations (to be determined)

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Output frequency (to be determined)

3.1.2.6.3 Information Processing for Interruption Processing

a. Exhibit 8 diagramatically shows which type of interrupts shall be enabled and allowed for a routine currently being executed.

b. Interrupt service shall be either of the following two kinds:

(1) Complete Service

- o SAVE program counter and registers, as required
- o ISOLATE interrupt type
- o TRANSFER to interrupt routine for action preference or priority
- o SERVICE interrupt request
- o RETURN interrupt state to READY
- o RESTORE program counter, etc.

(2) Partial Service

- o SAVE program counter and registers, as required
- o ISOLATE interrupt type
- o TRANSFER to interrupt routine for action preference or priority
- o ENTER interrupt type in a queue list
- o RETURN interrupt state to WAITING
- o RESTORE program counter, etc.

The particular service to be selected will be decided on the basis of whether the conditions require immediate or nonimmediate handling (see Paragraph 4.3.5, Reference 2.2.3).

Type of Interrupt Enabled and Allowed	Program or Routine Being Executed					
	Multi-plexed Sequence Control	SVC Interrupt Processor	Program Interrupt Processor	I/O Interrupt Processor	External Interrupt Processor	Priority Interrupt Processor
SVC	Y	N	N	N	N	N
PI	Y	(Y)	(Y)	(Y)	(Y)	(Y)
I/O	Y	Y*	Y*	N	N	N
External	Y	Y	Y	Y	Y	N
Priority	Y	Y	Y	Y	Y	Y_h

( ) If interrrupt occurs, it is treated as a machine check interrupt.

* Partly enabled

Y_h Interrupt allowed if of higher priority than the interrupt currently being processed.

Y Yes

N No

EXHIBIT 8 - INTERRUPTS ALLOWED

3.1.2.7 Function 7: Console Communication

The console communication function shall be implemented to provide proper operation of OCDMS controls and displays. Operations personnel must be able to effectively change systems characteristics, and in order to do so, shall be provided with adequate information concerning the functioning of the computer and the experiment apparatus.

Exhibits 9a and 9b illustrate the operations involved for the printer/keyboard and the display, respectively.

3.1.2.7.1 Sources and Types of Inputs for Console Communication

a. Functional inputs shall include, but not necessarily be limited to, the following types:

(1) Keyboard entry commands for procedure control and test-point monitor requests

(2) Keyboard annotations of manual experiment results, system status, and message-to-tag particular data, for later analysis

(3) On/off switch control signals for unique system modes or performance functions (e.g., Automatic, Semiautomatic, or Manual Mode, and sense switches with mission/configuration dependent meanings)

(4) Query and response operations between the OCDMS console operator and Supervisory System

(5) Task-requests to display or print specified sets of data

(6) Request-compliance information concerning allocated storage units

b. Sources of these inputs shall, respectively, include, but not necessarily be limited to, the following:

(1) Control Display Unit hardware

(2) Control Display Unit hardware

(3) Control Display Unit hardware

(4) Control Display Unit hardware

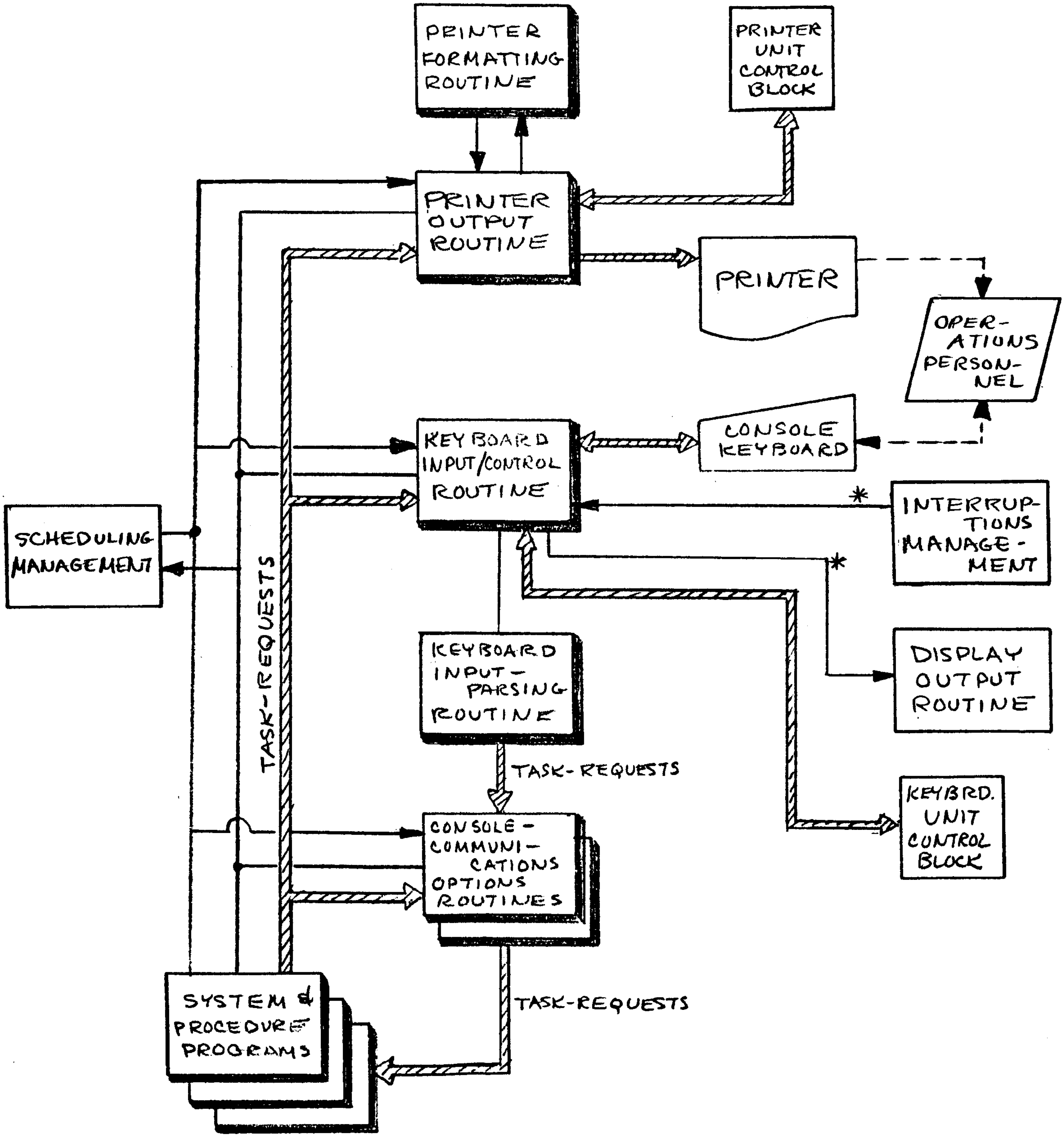

PRINTER FORMATTING ROUTINE
PRINTER UNIT CONTROL BLOCK
PRINTER OUTPUT ROUTINE
PRINTER
OPERATIONS PERSONNEL
KEYBOARD INPUT/CONTROL ROUTINE
CONSOLE KEYBOARD
INTERRUPTIONS MANAGEMENT
*
*
SCHEDULING MANAGEMENT
TASK-REQUESTS
DISPLAY OUTPUT ROUTINE
KEYBOARD INPUT-PARSING ROUTINE
TASK-REQUESTS
KEYBRD. UNIT CONTROL BLOCK
CONSOLE-COMMUNICATIONS OPTIONS ROUTINES
TASK-REQUESTS
SYSTEM & PROCEDURE PROGRAMS

EXHIBIT 9a - CONSOLE COMMUNICATION (PRINTER AND KEYBOARD)

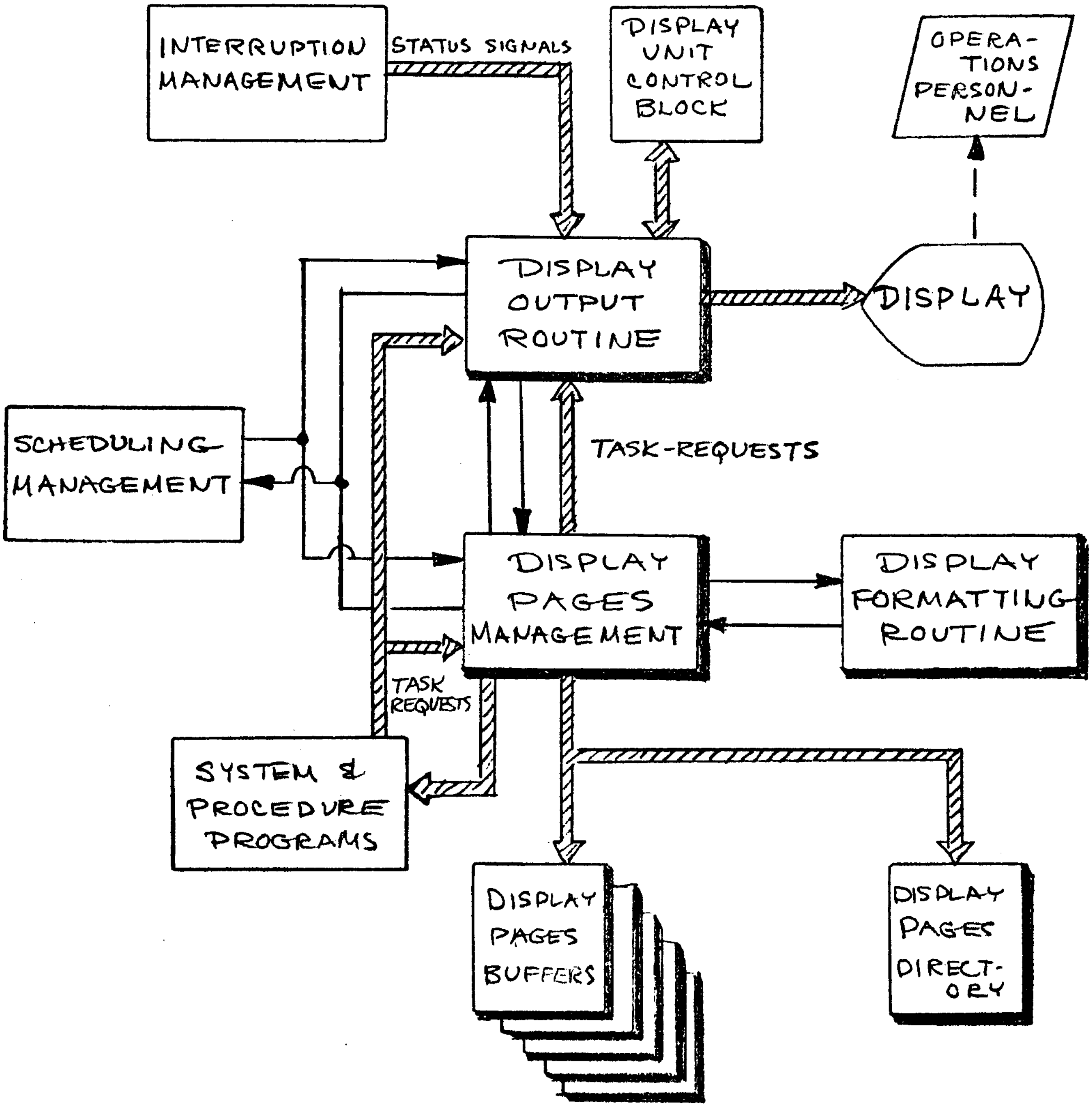


EXHIBIT 9b - CONSOLE COMMUNICATION (DISPLAY)

(5) System Management, Procedure Management, and Language Interpreter functions (3.1.2.1, 3.1.2.4, and 3.1.2.3), and procedure programs

(6) Memory Management functions (3.1.2.2)

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Arrival frequency (to be determined)

3.1.2.7.2 Destination and Types of Outputs for Console Communication

a. Functional outputs shall include, but not necessarily be limited to, the following types:

(1) Displays that compensate for man's limited capability of information storage and retrieval, and that provide status of concurrent processes and high-order integration functions

(2) Data conversions, changing machine symbolic codes to/from information forms meaningful to OCDMS operators

(3) Transmission feedback information

(4) Provisions for error-free console communications and/or failure compensation

(5) Task-requests for allocation/return of storage units from/to the storage "pool"

(6) Request-compliance information indicating either the successful completion of a task or the reason for a failure to comply, as appropriate

b. Destinations of these outputs shall, respectively, include, but not necessarily be limited to, the following:

(1) Operations personnel, via the display or printer hardware

(2) Managed as a function of console communication

(3) Display or printer hardware

(4) Managed as a function of console communication

(5) Memory Management function (3.1.2.2)

(6) Other Supervisory System functions (probably limited to System Management, Language Interpreter, and Procedure Management functions, and procedure programs)

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Output frequency (to be determined)

3.1.2.7.3 Information Processing for Console Communication

The set of programs and routines implementing the console communication function will perform data formatting, internal bookkeeping, and input/output switching operations. These programs will receive control at time intervals determined by scheduling management, and also as a function of interrupt management. The information processing performed by this function is discussed in terms of printer, keyboard, display, and console options subprograms.

a. Printer Subprogram

This subprogram shall be capable of processing a queue of task requests, formatting data records as necessary, and executing a subroutine CALL to the proper input/output routines. It shall access the appropriate Unit Control Block entries for control information concerning the printer and shall have a "tracer" mechanism to determine the status of action items.

b. Keyboard Subprogram

This subprogram shall be capable of enabling and disabling specified sections of the keyboard. An enabled key depressed by the console operations personnel shall result in a corresponding character or function display, on an interrupt basis. When the enabled key is depressed, an interruption will occur which trap-transfers CPU control to a routine causing a single character/function to be displayed. This action shall consist only of transmitting and storing the proper codes; no interpretation shall be made. Completion of an operator message to the machine procedures shall be tagged with an EXECUTE/ TRANSMIT key code, and processed on a scheduled basis.

c. Display Subprogram

The display subprogram shall perform information processing functions involved with the formatting and output switching required to accomplish the requested displays. In order to manage CDU demands associated with concurrent activities, the display subprogram will utilize a system of display pages. Pages will consist of main storage units (buffers) managed by the display subprogram. Each display source will use one or more pages to contain the information pattern to be communicated. The display subprogram shall be capable of generating, modifying, and/or retrieving pages as called for by the various sources. The generation and modification logic will involve formatting and positioning the data, linkage of continuation pages, and getting additional buffers from pool storage. In order to implement the paging technique, the subprogram will maintain an index of the locations of the pages. Operations personnel may call for a specified page or set of pages, and shall be allowed to switch back and forth through the entire set of pages associated with a given source. A set of pages shall typically be provided for at least the following:

(1) Each procedure program requiring display output

(2) The system management subprogram (system status--processing, input/output memory load, etc.)

(3) Operator constructed procedures (on-line)

(4) Operator scratch pad (for use during scientific calculations, construction of procedures, etc.)

(5) Operations status (the system will continuously provide the OCDMS operator with information describing procedure execution status, such as what procedure(s) are currently being executed, what is the current step, etc.)

d. Console Options Subprogram

The console options subprogram will implement information processing to the following extent:

(1) A routine which initiates the transfer of display pages to bulk storage when requested by the operations personnel

(2) A routine which generates requests for display pages to be printed on the line printer when requested by the operations personnel

(3) A routine which enables the OCDMS to be used for scientific calculation and evaluation

(4) A routine which allows the OCDMS operator to enter limited annotations to a displayed page

(5) A routine which allows the OCDMS operator to select a specified display page

3.1.2.8 Function 8: CIU/Signal Adapter Communication

The CIU/Signal Adapter communication function shall be implemented so as to provide real-time and time-sharing operations for OCDMS experiment driving functions and corresponding measurement functions. In keeping with this, communication cells referred to as Unit Control Blocks (UCB's) and Procedure Control Blocks (PCB's) shall be used and updated with current status/configuration information as system tasks are performed. Exhibit 10 illustrates the operations involved.

3.1.2.8.1 Sources and Types of Inputs for CIU/Signal Adapter Communication

a. Functional inputs shall include, but not necessarily be limited to, the following types:

(1) Task-requests to perform stimulus/response activity associated with specified signal adapters

(2) Control information necessary to communicate with signal adapters; this will be contained in the Unit Control Blocks and will include:

o Locations of measurements

o Subsystem abbreviations

o Channel numbers

o Quantity of similar signals

o Usage (factory/prelaunch/mission)

o End item device names

o Confidence factors

o Analog signal descriptions (range, units, accuracy, frequency)

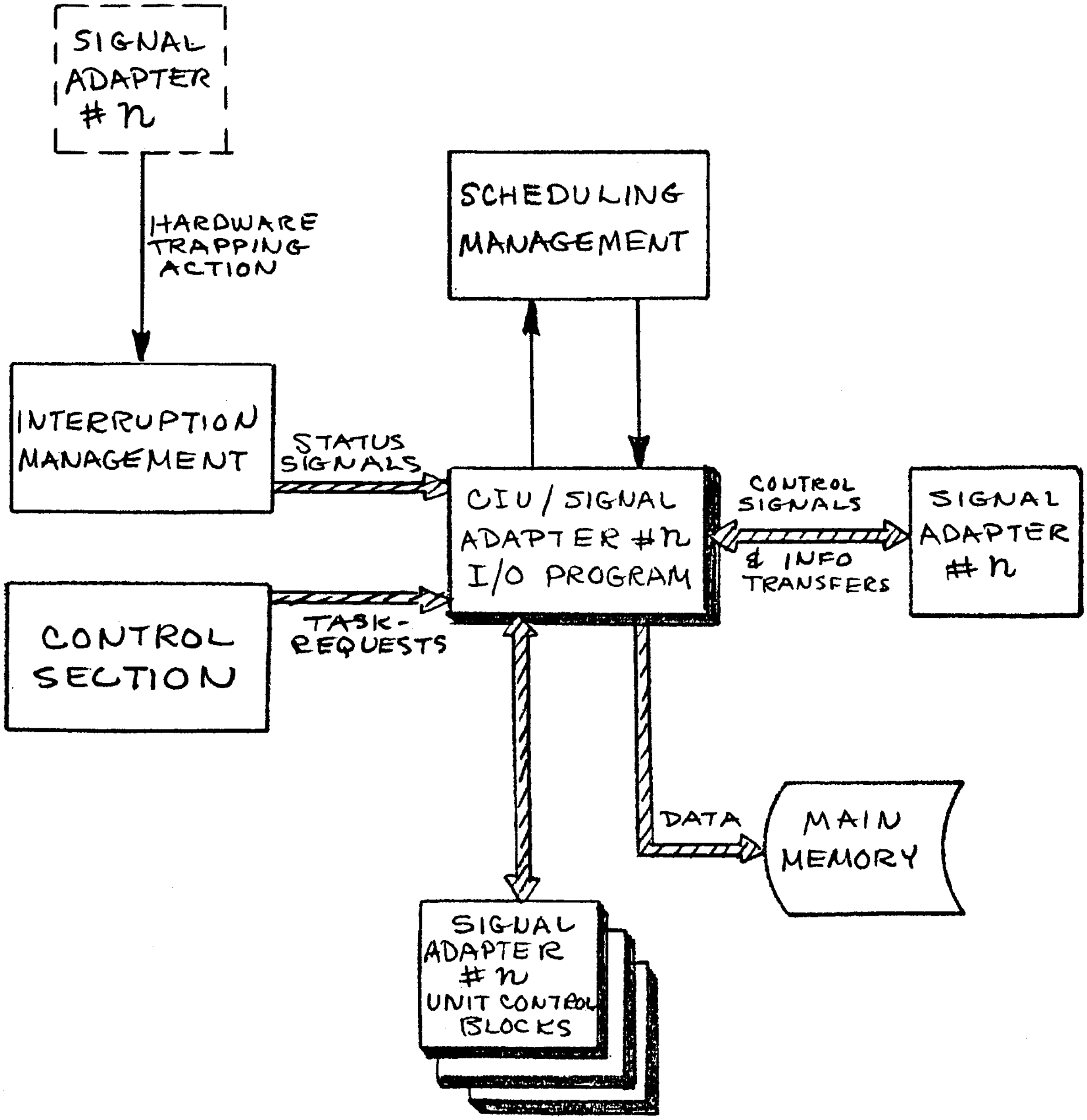


EXHIBIT 10 - CIU/SIGNAL ADAPTER COMMUNICATION

- o Discrete signal descriptions (logic levels, units, time)

- o Coded signal descriptions (bits/bytes/words, conversion codes)

- o Stimulus descriptions (prerequisites for turn-on, associated measurement, ramp/step, harmonic distortions, duty cycle)

- o Measurement descriptions (momentary/continuous, response time, stimuli required, display required)

(3) Control signals and information transfers (response) for discrete, analog, and coded driving functions; in the form of single response and sample pattern measurements

b. Sources of these inputs shall, respectively, include, but not necessarily be limited to the following:

(1) Procedure programs (provided by the OCDMS Support System and created on-line as a function of the Language Interpreter (3.1.2.3)

(2) OCDMS Support System

(3) CIU/Signal Adapter hardware (to be determined)

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Arrival frequency (to be determined)

3.1.2.8.2 Destinations and Types of Outputs for CIU/Signal Adapter Communication

a. Functional outputs shall include, but not necessarily be limited to, the following types:

(1) Request-compliance information indicating the outcome of processing task-requests from procedure programs; this information will indicate if the processing was successful, and if not, why not (i.e., hardware errors occurring)

(2) Device (signal adapter) status information (i.e., busy, inoperative, shutdown, etc.)

(3) Control signals and information transfers (stimulus) for discrete, analog, and coded driving functions

b. Destinations of these outputs shall, respectively, include, but not necessarily be limited to, the following:

(1) Procedure programs

(2) Unit Control Blocks

(3) CIU/Signal Adapter hardware (to be determined)

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Output frequency (to be determined)

3.1.2.8.3 <u>Information Processing for CIU/Signal Adapter Communication</u>

a. The CIU/Signal Adapter communication function shall receive control from the schedule management function and other system routines, which process and/or generate "actual" performance profiles. The processes of CIU/ Signal Adapter communication shall include the following operations:

(1) Accepting, queuing, and responding to task requests directed to them

(2) Accessing the necessary fields in a selected Unit Control Block

(3) Transmitting and receiving control data and other status information via input/output channel routines

(4) Processing buffer requests, and explicit communications of buffer pointers to routines which will subsequently operate on the experiment data

b. The logic shall be implemented in conjunction with this system function to compare procedure predicted profiles of discrete and analog stimuli/measurements with actual response measurement profiles. Difference between predicted and actual conditions will cause special handling features to be exercised, depending on a predefined severity code.

3.1.2.9 Function 9: Uplink Communication

The uplink communication function shall be implemented to provide performance compatibility and functional operations for transferring digital data commands from ground site facilities to the OCDMS via the Apollo Unified S-Band Digital Command System (DCS). Operational characteristics require that redundant transmission checks and error protection encoding techniques be employed in order to minimize undetected errors, and to achieve OCDMS design reliability goals.

3.1.2.9.1 Sources and Types of Inputs for Uplink Communication

a. Functional inputs shall include, but not necessarily be limited to, the following types:

(1) Apollo DCS real-time commands

(2) Control information necessary for communication with the DCS uplink hardware; this will be contained in an appropriate Unit Control Block and will include:

- On/off data formats
- Digital/analog data formats
- Timing and event codes
- System addresses
- Redundancy words
- Error protection

b. Sources of these inputs shall, respectively, include, but not necessarily be limited to, the following:

(1) Apollo DCS hardware

(2) OCDMS Support System

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Arrival frequency (to be determined)

3.1.2.9.2 Destinations and Types of Outputs for Uplink Communication

a. Functional outputs shall include, but not necessarily be limited to, the following types:

(1) Symbolic code to be interpreted by the Language Interpreter

(2) Command/control requests (including message verification signals)

(3) OCDMS mode selection requests

b. Destinations of these outputs shall, respectively, include, but not necessarily be limited to, the following:

(1) Language Interpreter function (3.1.2.3)

(2) Apollo DCS hardware

(3) Operation personnel (via Console Communication function (3.1.2.7)

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Output frequency (to be determined)

3.1.2.9.3 Information Processing for Uplink Communication

Information processing by the uplink communication function shall provide for precise actions for each type of message and format received from the multiple ground sites. Each mission will cause changes in the exact context meaning of particular commands and/or messages. Therefore, the routines implementing this function shall perform in such a way that no degradation of OCDMS system performance occurs because of programming or parameter changes.

Message-words shall normally be 48-bit (ACE-SC) or 32-bit (GOSS) bi-phase words. Each word consists of two subwords. One subword will consist of data bits, and the other will contain redundancy bits for use as a transmission check. Information processing by this function of the Supervisory System shall provide for OCDMS handling of DCS transmission check procedures, and for generation of proper command words.

3.1.2.10 Function 10: Downlink Communication

The downlink communication function shall provide for simple and reliable means of transmitting all significant OCDMS data to various operational ground sites via equipment which will be compatible with the Apollo Unified S-Band PCM Telemetry System. Data compression techniques shall be employed which minimize the bandwidth needed to transmit given amounts of information in specified time intervals; or which reduce the time required to transmit a given amount of information in a given bandwidth. Such compression shall be accomplished without degradation of experiment data integrity; at no increased complexity of existing systems; and with an insignificant increase to the programming workloads that shall be assigned to the ground site facilities.

3.1.2.10.1 Source and Types of Inputs to the Downlink Communication Function

a. Functional inputs shall typically be obtained as outputs generated directly from data generation and collection elements of the system. This information shall include, but not necessarily be limited to, the following types:

(1) Task-requests to "dump" data and to perform data compression operations

(2) Control signals pertaining to the status of the CIU/PCM downlink hardware

(3) Data to be dumped via the downlink--this will include:

o Experiment/checkout measurement values

o Event/activity time records

o Equipment items (time or cycle sensitive components) usage information

o OCDMS statistics

(4) Control and identification information pertaining to this data, including:

o Measurement conditions

o Measurement annotations

- o Data segment headers and files
- o Testing tolerances in use

(5) Other control information, including:

- o Ground system site-peculiar parameters
- o Apollo PCM T/M format definitions

(6) Uplink transmission-checks

(7) Change action record information

b. Sources of these inputs shall, respectively, include, but not necessarily be limited to, the following:

(1) Procedure programs and the System Management function (3.1.2.1)

(2) CIU/PCM hardware (to be determined)

(3) Procedure programs and the System Management function

(4) Procedure programs and the System Management function

(5) OCDM Support System

(6) Uplink Communication function (3.1.2.9)

(7) Operations personnel (via Console Communication)

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Arrival frequency (to be determined)

3.1.2.10.2 Destinations and Types of Outputs for Downlink Communication

a. Functional output shall include, but not necessarily be limited to, the following types:

(1) Backlogged PCM prime-frame buffers

(2) Data compression outputs

(3) Identification information pertaining to compressed data, including:

- o Selective monitoring orders
- o Adaptive sampling orders
- o Direct data coding instructions

(4) Downlink transmission check information

(5) Burst-mode PCM transmission commands

(6) Request-compliance information

b. Destinations of these outputs shall, respectively, include, but not necessarily be limited to, the following:

(1) CIU/PCM downlink hardware (to be determined)

(2) Managed as function of Downlink Communication (formatted into prime-frame buffers)

(3) Managed as function of Downlink Communication (included as header information for the data when constructed into prime-frames)

(4) CIU/PCM downlink hardware

(5) CIU/PCM downlink hardware

(6) Procedure programs and System Management function

c. Units of measure (to be determined)

d. Limits/ranges (to be determined)

e. Accuracy/precision (to be determined)

f. Output frequency (to be determined)

3.1.2.10.3 Information Processing for Downlink Communication

a. The information processing procedure shall format data in accordance with prescribed Apollo Unified S-Band PCM T/M prime-frames and/or subframes (see discussion in Section 5.3.2, Reference 2.3.3). Logical functions shall provide for auxiliary storage of data whenever collection exceeds the data dispersal capabilities. Provisions shall be made for expedited transfer rates of PCM data when mission conditions are proper (i. e., spacecraft in transmitting range of ground station and other communication traffic and operations at an acceptable activity level).

b. Data compression algorithms shall be called by the system for all conditions in which data redundancy reduction is appropriate. Selection of a particular compression technique shall be predefined for particular classes of data. Data compression methods that shall be available include:

(1) Parameter extractions

- o Probability distribution
- o Power spectrums
- o State parameters
- o Fourier coefficients
- o Other irreversible information-describing transformations

(2) Direct redundancy reduction

- o Peak error predictors
- o RMS error predictors
- o Peak error interpolators
- o RMS error interpolators

(3) Encoding

- o Adaptive coefficients
- o Nonadaptive increments
- o Bit plane

(4) Adaptive Sampling

- o Variable rate
- o Fixed rate
- o Command controlled

3.1.3 Data Base Requirements

All parameters which affect the design of this CPCEI shall be prepared and organized on bulk storage media by the OCDMS Support System Software. No additional conversion of system parameters shall be required prior to usage by the system routines. Site adaptation parameters for factory, launch, and remote mission sites shall constitute data sets reserved in bulk storage, which on request shall be transferred to the appropriate ground station.

3.1.4 Human Performance

In accordance with requirement identified by Reference 2.1.2, General Specification for the OCDMS (see Paragraphs 3.1.3.6 and 3.3.3.1.3.f), the CPCEI shall have performance characteristics that reflect established human engineering design standards. In order to enhance the reliability of human performance and to reduce operating inefficiencies and training requirements, MSFC-STD-267A, Human Engineering Design Criteria, shall be used as a guideline for OCDMS design, as applicable.

3.2 CPCEI Definition

The functional relationship of the CPCEI to other equipment and computer programs, and the identification of Government-furnished computer programs incorporated in the CPCEI are specified by the following subparagraphs.

3.2.1 Interface Requirements

The OCDMS Supervisory System will be utilized by programming, engineering, test, and astronaut/scientist personnel during all operational phases of S/AA Program missions. The overall mission support activities relating to various installations, sites, and operating locations are identified by the system specification, Reference 2.2.1. These activities are the basis of the interface requirements delineated in the following subparagraphs.

3.2.1.1 Interface Block Diagram

Exhibit 11 defines the interface relationships of this CPCEI to other equipment/computer programs for which interface requirements shall be specified.

3.2.1.2 Detailed Interface Definition

3.2.1.2.1 OCDMS Support System CPCEI

The primary functional interface between the supervisory and support system software shall be the experiment procedure programs

	SUPPORT SYSTEM (3.2.1.2.1)	EXPERIMENT PROCEDURES (3.2.1.2.2)	COMPUTER & PERIPHERAL EQUIPMENT (3.2.1.2.3)	EXPERIMENT HARDWARE (3.2.1.2.4)	GROUND SITES (3.2.1.2.5)	OPERATIONS PERSONNEL (3.2.1.2.6)
SYSTEM MANAGEMENT (3.1.2.1)	X	X	X	X	X	X
MEMORY MANAGEMENT (3.1.2.2)	X	X	X			
LANGUAGE INTERPRETER (3.1.2.3)		X				X
PROCEDURE MANAGEMENT (3.1.2.4)		X				X
SCHEDULE MANAGEMENT (3.1.2.5)	X	X		X		X
INTERRUPT MANAGEMENT (3.1.2.6)		X	X	X	X	X
CONSOLE COMMUNICATION (3.1.2.7)		X	X		X	X
CIU/SIGNAL ADAPTER COMMUNICATION (3.1.2.8)	X	X	X	X		
UPLINK COMMUNICATION (3.1.2.9)	X		X		X	
DOWNLINK COMMUNICATION (3.1.2.10)	X		X		X	

EXHIBIT 11 - OCDMS SUPERVISORY SYSTEM INTERFACES

and on-board system data base (experiment/OCDMS parameters) prepared by the Support System. This implies that the Support System CPCEI functions establish functional interface requirements (i.e., the Language Translation, Data Management, Program Production, and Program Test and Verification functions of the OCDMS Support System). These functions compile and generate the procedure programs and data sets that are incorporated together with the OCDMS Supervisory System on the auxiliary storage medium in a format and order suitable for OCDMS mission operations.

Below is a list of the elements that the OCDMS Support System will generate and/or assemble and place on the auxiliary storage medium (System Master File). The contents of these elements and their relationships to the various functions of the Supervisory System have been specified previously, in Section 3.1.2, Operational Requirements; hence, they will be mentioned here only by name. They will include, but not necessarily be limited to, the following:

- Auxiliary Storage Directory
- Procedure Programs (executable segments)
- Procedure Control Blocks
- System definition (description of resident supervisory system)
- Resident Supervisory System (programs and data sets)
- Nonresident Supervisory System (programs and data sets)
- Storage "pool" definition
- Unit Control Blocks
- On-Line Interface Dictionary
- Scheduling Algorithm Parameter Table
- Procedure Schedule

Prior to CPCEI qualification, the Supervisory System design and development testing shall utilize the test services provided by the Support System Program Test and Verification functions.

Exhibit 12 indicates the particular Supervisory System function with which these elements interface with reference to the comparable exhibit in the Support System CPCEI (Reference 2.2.2) will further illustrate the precise interface between the Support and Supervisory Systems.

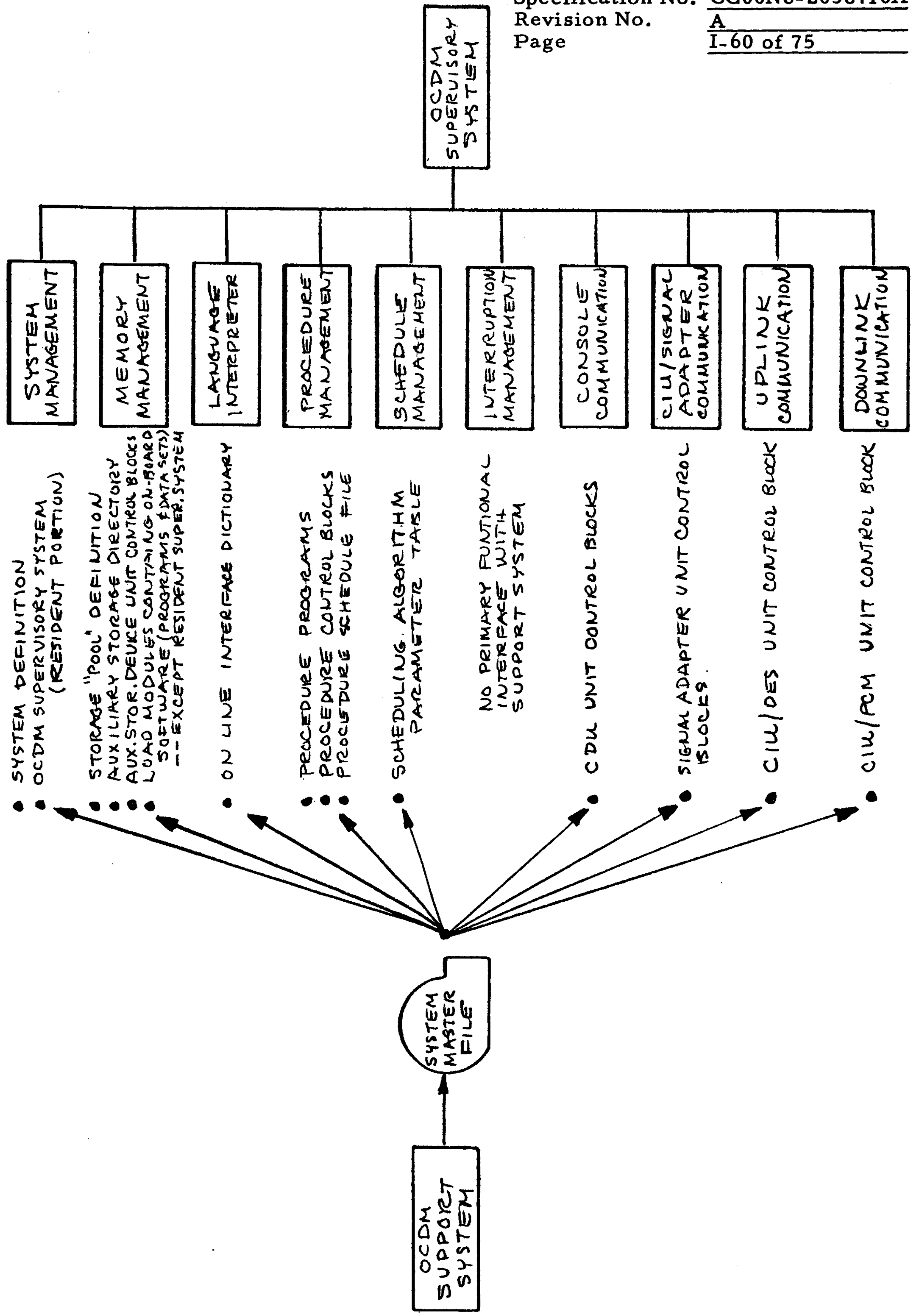

OCDM SUPERVISORY SYSTEM
SYSTEM MANAGEMENT
MEMORY MANAGEMENT
LANGUAGE INTERPRETER
PROCEDURE MANAGEMENT
SCHEDULE MANAGEMENT
INTERRUPTION MANAGEMENT
CONSOLE COMMUNICATION
CIU/SIGNAL ADAPTER COMMUNICATION
UPLINK COMMUNICATION
DOWNLINK COMMUNICATION
• SYSTEM DEFINITION
• OCDM SUPERVISORY SYSTEM (RESIDENT PORTION)
• STORAGE "POOL" DEFINITION
• AUXILIARY STORAGE DIRECTORY
• AUX. STOR. DEVICE UNIT CONTROL BLOCKS
• LOAD MODULES CONTAINING ON-BOARD SOFTWARE (PROGRAMS & DATA SETS) -- EXCEPT RESIDENT SUPER. SYSTEM
• ON LINE INTERFACE DICTIONARY
• PROCEDURE PROGRAMS
• PROCEDURE CONTROL BLOCKS
• PROCEDURE SCHEDULE FILE
• SCHEDULING ALGORITHM PARAMETER TABLE
NO PRIMARY FUNTIONAL INTERFACE WITH SUPPORT SYSTEM
• CDU UNIT CONTROL BLOCKS
• SIGNAL ADAPTER UNIT CONTROL BLOCKS
• CIU/DES UNIT CONTROL BLOCK
• CIU/PCM UNIT CONTROL BLOCK
SYSTEM MASTER FILE
OCDM SUPPORT SYSTEM

EXHIBIT 12 - OCDMS SUPPORT SYSTEM--SUPERVISORY SYSTEM INTERFACE ELEMENTS

3.2.1.2.2 OCDMS Experiment Procedure CPCEI

To be determined.

3.2.1.2.3 OCDMS Computer and Associated Peripheral Equipment

3.2.1.2.3.1 Computer

To be determined.

3.2.1.2.3.2 Manufacturer-Supplied Programming Language(s)

To be determined.

3.2.1.2.3.3 Control/Display Unit

To be determined.

3.2.1.2.3.4 Bulk Storage

To be determined.

3.2.1.2.4 OCDMS Experimental Hardware

To be determined.

3.2.1.2.5 Operations Personnel

The OCDMS Supervisory System shall provide the necessary functional interface characteristics which provide the capability of operations personnel to control and/or monitor all facets of OCDMS operations. This includes direction and/or participation in activities such as:

a. Overall system startup/shutdown

b. Procedure execution control and modification

c. On-line procedure generation

d. Scientific calculation and evaluation

e. On-line data retrieval and analysis

f. General man-machine adaptive operations

The communications interface characteristics necessary for implementation shall be a function of Console Communication (3.1.2.7) and Uplink Communication (3.1.2.9). Simple and reliable communication of information to operational ground sites shall be accomplished in accordance with Paragraph 3.1.2.10.

3.2.1.2.6 Ground Stations (PIF, ACE, GOSS)

Functional interface requirements for this CPCEI with overall mission support activities relating to various installations, sites, and operating location are delineated by Section 3.1.1.1.2 of Reference 2.2.1, and Section 3.2.1.1 of Reference 2.2.2. The operational concepts of these interfaces are discussed in Section 5.1.4 of Reference 2.3.3.

3.2.2 Government-Furnished Property List

Not applicable.

3.3 Design Requirements

This section of the specification contains requirements and standards that affect the design of the CPCEI and are distinguishable from the performance requirements of Section 3.1. The requirements identified are, in general, a direct result of overall OCDMS design criteria, and logically follow the design requirements of the OCDMS Support System CPCEI. Exhibit 13 illustrates the generalized overall system operation flow.

3.3.1 Programming Standards

Programming standards that are applicable to the CPCEI shall be those identified by the OCDMS Support System Specification, Reference 2.2.2 (Paragraph 3.3.1).

3.3.2 Program Design

Program organization, construction, communication, control and naming conventions to be adopted for this CPCEI shall be those described by the OCDMS Support System Specification, Reference 2.2.2 (Section 3.3.2 and subparagraphs thereof).

3.3.3 Program Modification

CPCEI modification shall adhere to configuration management accounting practices delineated by Reference 2.3.1. The development phase of the CPCEI shall in general be conducted in accordance with guidelines given by Reference 2.3.4. Each S/AA mission shall have a specific OCDMS Supervisory System configuration subject to change control actions, accounting, and reporting procedures.

3.3.4 CPCEI Testing Facilities

The CPCEI shall be designed, coded, and implemented on the computers using the testing facilities provided by the OCDMS Support

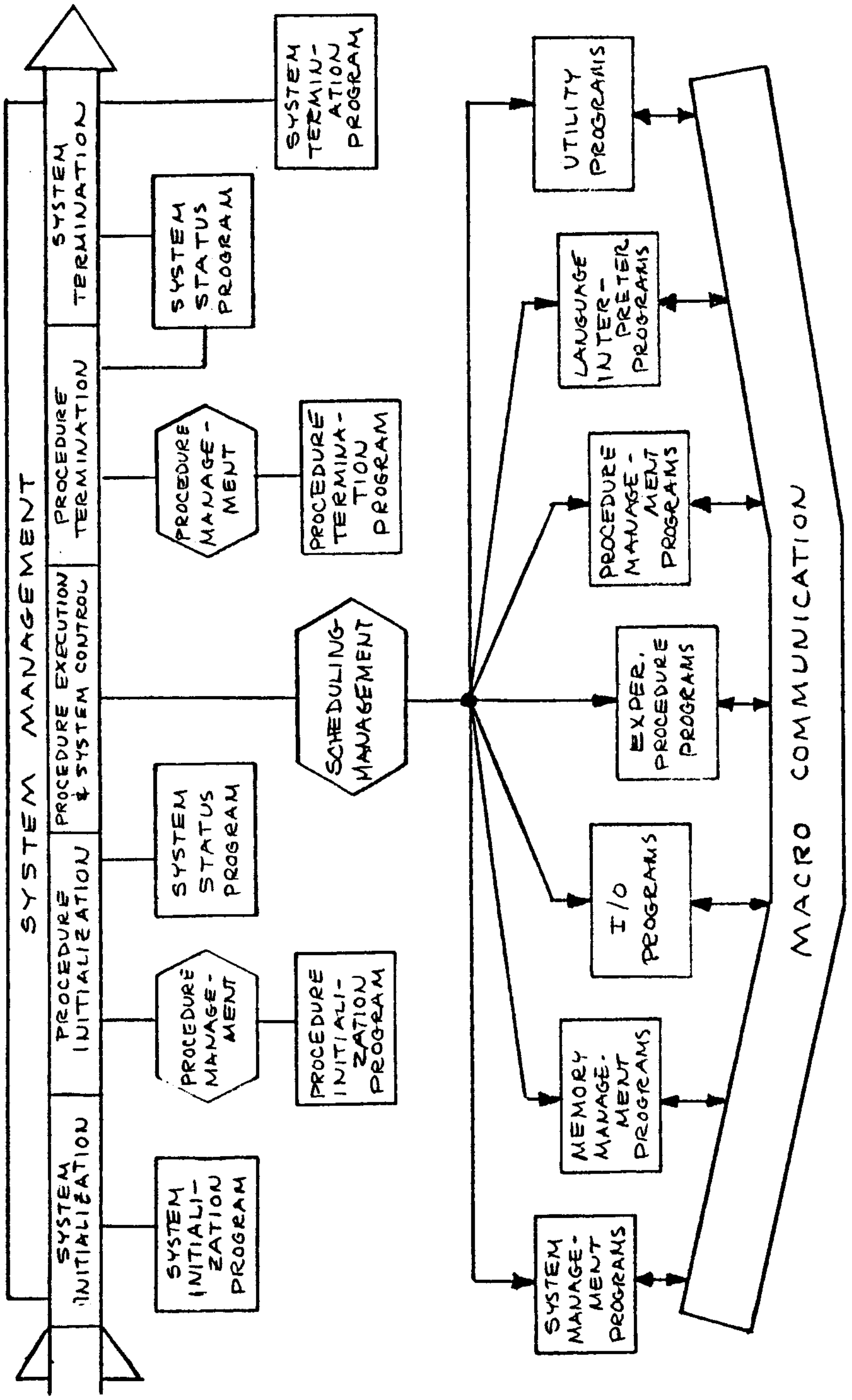

SYSTEM MANAGEMENT
SYSTEM INITIALIZATION
PROCEDURE INITIALIZATION
PROCEDURE EXECUTION & SYSTEM CONTROL
PROCEDURE TERMINATION
SYSTEM TERMINATION
SYSTEM INITIALI-ZATION PROGRAM
PROCEDURE MANAGE-MENT
PROCEDURE INITIALI-ZATION PROGRAM
SYSTEM STATUS PROGRAM
SCHEDULING-MANAGEMENT
PROCEDURE MANAGE-MENT
PROCEDURE TERMINA-TION PROGRAM
SYSTEM STATUS PROGRAM
SYSTEM TERMIN-ATION PROGRAM
SYSTEM MANAGE-MENT PROGRAMS
MEMORY MANAGE-MENT PROGRAMS
I/O PROGRAMS
EXPER. PROCEDURE PROGRAMS
PROCEDURE MANAGE-MENT PROGRAMS
LANGUAGE INTER-PRETER PROGRAMS
UTILITY PROGRAMS
MACRO COMMUNICATION

EXHIBIT 13 - GENERALIZED SYSTEM OPERATION FLOW

System, Reference 2.2.2 (Section 3.3.4). These test services will be employed to demonstrate acceptance and verification of the CPCEI in accordance with Section 4.0.

3.3.5 CPCEI Expandability

The requirements specified in Sections 3.3.1 and 3.3.2 constitute a modular design concept relative to this CPCEI. Expandability of the CPCEI shall be provided by means of CPC modular construction techniques. Additions to the CPCEI shall conform to the requirements of Paragraph 3.3.3, with particular emphasis given to ensure that specified file protection, program protection, and program control conventions are not disregarded.

4.0 QUALITY ASSURANCE PROVISIONS

Requirements for formal verification of the performance of the CPCEI in accordance with the requirements for Section 3 of this specification are specified in order to:

a. Determine if the components of the CPCEI are implemented correctly

b. Determine if the CPCEI satifies the requirements of its Part I Specification

c. Obtain test results that are used to determine if scheduled milestones have been achieved

d. Formally qualify the completed computer programs for operations use

The methods of verification that are specified herein include inspection of the CPCEI, review of analytical data, demonstration tests, and review of test data.

4.1 Implementation Test Requirements

Implementation tests include all tests of the CPCEI other than those accomplished during integration tests (see paragraph 4.2). Several stages of tests shall be required to validate the design of the CPCEI and to verify that the implementation of the design is correct. These shall include, but not be limited to, the following categories:

a. Pre-ImplementationDesign Tests, which are tests run on trial designs prior to establishing an initial design approach. These tests indicate real-time performance characteristics, computational accuracies, storage limitations, etc. When appropriate, tests of this type shall be continued throughout the design process.

b. Subprogram Checkout, which includes visual inspections and hand manipulations with selected data of coded CPCEI subprograms, followed by assembling the subprogram on the computer. Each assembled subprogram shall be tested by use of controlled data inputs. The goal is to identify and reduce idigenous and exogenous failure mechanisms prior to combining the subprograms into main programs or other functional program aggregates.

c. Main Program Checkout, which includes tests performed on functionally related subprograms. These shall be executed initially with no input, to verify the ability to cycle. Controlled inputs shall then be introduced to establish correct performance. Purpose of the testing is to eliminate logic and coding errors from the interfaces between subprograms. Testing levels shall be accomplished at this stage to the corresponding levels of subprograms within the CPCEI.

d. CPCEI Simulated Environment Tests

(1) The CPCEI shall be tested in a simulated environment prior to its integration into the total computer-based system. Such tests are contingent on the availability of a system environment simulator and associated test support tools; therefore, these tests may be deferred until overall OCDMS qualification testing.

(2) Simulated environment testing objectives are as follows:

o To obtain a more controlled test of the CPCEI than could be accomplished in the total OCDMS

o To determine the safety of the CPCEI in the OCDMS without exposing the system to unnecessary hazard

o To serve as the basis for preliminary qualification to the CPCEI prior to transferring it from a development facility to a using facility

o To provide preliminary training in the use of the CPCEI and to evaluate proposed operating procedures for the CPCEI

4.1.1 Design and Development Testing

Computer program components and program tests shall be conducted in the acquisition phase prior to the preliminary qualification tests. These shall be validation and verification tests that prove the design and demonstrate specified performance requirements for each of the major functions.

a. System Management

(1) Program initialization

(2) Program termination

(3) Input/output resource allocation

(4) Power cycle restart provisions

(5) Error detection and recovery

(6) Status data collection and reporting

(7) Accounting and event trial facilities

b. Memory Management

(1) Main storage allocation

(2) Main storage retrieval

(3) Bulk storage READ/WRITE

(4) Program and data set relocation

c. Language Interpreter

(1) Functional expression decomposition

(2) Calling sequence and linkage generation

(3) Command/control action routines

(4) List processing

(5) Parameter changing

d. Procedure Management

(1) Procedure Block File processing

(2) Task Control Block processing

(3) Mode selection (automatic, semiautomatic, manual)

(4) "Predicted" and "actual" comparison actions

(5) Manual operation support

e. Schedule Management

(1) Sequence assignment tasks

(2) Priority sequencing

(3) Normal sequencing

(4) Periodic sequencing

(5) Dispatching

(6) Input/output channel scheduling

f. Interruption Management

(1) Priority interruption

(2) Program and supervisor-call interruption

(3) Input/output, external, and machine-check interruptions

(4) Completed interruption services

(5) Partial interruption services

g. Console Communication

(1) Procedure setup commands

(2) Procedure control commands

(3) Function monitor commands

(4) Procedure annotations

(5) Display commands

h. CIU/Signal Adapter Communication

(1) Discrete driving functions

(2) Analog driving functions

(3) Coded driving functions

(4) Single-response measurements

(5) Sampled measurements

i. Uplink Communication

(1) Real-time commands

(2) Error protection encoding

(3) Symbolic code for the interpreter

(4) Input command/control actions

j. Downlink Communication

(1) Prime-frame generator

(2) Buffering and data transfer

(3) Data compression

4.1.2 Preliminary Qualification Test

Preliminary qualification tests shall verify each Section 3 requirement which can be tested in a simulated environment.

4.1.2.1 Qualification Test Requirements

The procuring agency shall review procedures and audit results of critical demonstration tests which as a minimum shall include:

a. System compatability: all computer program components (CPC's) shall be tested for proper program linkage and operation of the functional hardware units.

b. Man-machine relationships: particular emphasis shall be given to qualifying the man-machine compatibility, and to the adequacy of the man-machine to fulfill the mission requirements.

c. Simulated environment: emphasis shall be given to simulating the most adverse conditions possible for the computer and other OCDMS hardware elements during demonstration of the software system.

4.1.2.2 Resources Required for Testing (to be determined)

4.1.2.3 Test Schedules and Locations (to be determined)

4.1.3 Special Test Requirements (to be determined)

4.2 Integration Test Requirements

This section specifies the verification test requirements applicable to performance/design requirements, identified in Section 3.0, which cannot be accomplished until the CPCEI is assembled into or used with the OCDMS computer-based system environment and other CPCEI's.

4.2.1 General

The OCDMS Support System CPCEI tests that are required in direct support of system integration are as follows.

4.2.1.1 Sequence of Tests (to be determined)

4.2.1.2 Functions To Be Tested (to be determined)

4.2.1.3 Testing Environment (to be determined)

4.2.1.4 Support Computer Programs Required (to be determined)

4.2.1.5 Personnel Required (to be determined)

4.2.1.6 Equipment Required (to be determined)

4.2.2 Acceptance/Qualification Test

The requirements imposed against the CPCEI for formal qualification of the integrated computer program components (CPC's) with OCDMS are identified in the following subparagraphs. Verification of the requirements shall be accomplished by inspection, or review of analytical data, or by demonstration, or test and review of test data, or a combination of these as required by the procuring agency.

4.2.2.1 Sequence of Tests (to be determined)

4.2.2.2 Functions To Be Tested (to be determined)

4.2.2.3 Testing Environment (to be determined)

4.2.2.4 Support Computer Programs Required (to be determined)

4.2.2.5 Personnel Required (to be determined)

4.2.2.6 Equipment Required (to be determined)

6.0 NOTES

None.

10.0 APPENDIX

10.1 Glossary

Access method: Any of the data management techniques available to the user for transferring data between main storage and an input/output device.

Address: A value, or an expression representing a value, used in the calculation of storage addresses.

Allocate: To grant a resource to, or reserve it for, a job or task.

Asynchronous: Without regular time relationship; hence, as applied to program execution, unexpected or unpredictable with respect to instruction sequence.

Attach (task): To create a task control block and present it to the supervisor control program.

Attribute: A characteristic; for example, attributes of data include record length, record format, data set name, associated device type and volume identification, use, creation date, etc.

Auxiliary storage: Data storage other than main storage.

Block (records): (1) To group records for the purpose of conserving storage space or increasing the efficiency of access or processing. (2) A physical record so constituted, or a portion of a telecommunications message defined to be a unit of data transmission.

Block loading: The form of fetch that brings control sections of a load module into contiguous positions of main storage.

Buffer (program input/output): A portion of main storage into which data is read, or from which it is written.

Common: An area of storage used by more than one program concurrently (i. e., shared).

Control block: A storage area through which a particular type of information required for control of the supervisory system is communicated among its parts.

Control program: A collective or general term for all routines in the supervisory system that contribute to the management of resources, implement the data organization or communications conventions of the operating system, or contain privileged operations.

Control section: The smallest separately relocatable unit of a program; that portion of text specified by the programmer to be an entity, all elements of which are to be loaded into contiguous main storage locations. That element of the system that receives CPU control as a function of Schedule Management.

Data management: A general term that collectively describes those functions of the control program that provide access to data sets, enforce data storage conventions, and regulate the use of input/output devices.

Data organization: A term that refers to any one of the data management conventions for the arrangement of a data set.

Data set: The major unit of data storage and retrieval in the operating system, consisting of a collection of data in one of several prescribed arrangements and described by control information that the system has access to.

Data set label: A collection of information that describes the attributes of a data set, and that is normally stored with the data set.

Deferred entry: An entry into a subroutine that occurs as a result of a deferred exit from the program that passed control to it.

Deferred exit: The passing of control to a subroutine at a time determined by an asynchronous event rather that at a predictable time.

Dump (main storage): (1) To copy the contents of all or part of main storage onto an output device, so that it can be examined. (2) The data resulting from (1). (3) A routine that will accomplish (1).

Entry point: Any location in a program to which control can be passed by another program.

Event: An occurrence of significance to a task; typically, the completion of an asynchronous operation, such as input/output.

Fetch (program): (1) To obtain requested load modules and load them into main storage, relocating them as necessary. (2) A control routine that accomplishes (1).

Installation: A general term for a particular computing system, in the context of the overall function it serves and the individuals who manage it, operate it, apply it to problems, service it, and use the results it produces.

Linkage: The means by which communication is effected between two routines or modules.

Load: To fetch, that is, to read a load module into main storage preparatory to executing it.

Load module: The output of the program production programs of the support system in a format suitable for loading into main storage for execution.

Logical record: A record from the standpoint of its content, function, and use rather than its physical attributes; that is, one that is defined in terms of the information it contains.

Macro-instruction: A general term used to collectively describe a macro-instruction statement, the corresponding macro-instruction definition, the resulting assembler language statements, and the machine language instructions and other data produced from the assembler language statements; loosely, any one of these representations of a machine language instruction sequence.

Main storage: All addressable storage from which instructions can be executed or from which data can be loaded directly into registers.

Module (programming): The input to, or output from, a single execution of an assembler, compiler, or linkage editor; a source, object, or load module; hence, a program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading.

Multiprogramming: A general term that expresses use of the computing system to fulfill two or more different requirements concurrently.

Name: A set of one or more characters that identifies a statement, data set, module, etc., and that is usually associated with the location of that which it identifies.

Operator command: A statement to the control program, issued via a console device, which causes the control program to provide requested information, alter normal operations, initiate new operations, or terminate existing operations.

Path: A series of segments which, as represented in an overlay tree, form the shortest distance in a region between a given segment and the root segment.

Pointer (to): Address (absolute, indirect, referenced, etc.).

Procedure program: Any of the class of routines that perform processing of experiment procedures for checkout, data management, equipment, equipment self-check, etc.; logically equivalent to application program.

Queue (request queue): A table of task-requests arranged in a manner reflecting the order in which they were received by the routine managing the queue.

Request-compliance information: A set of one or more computer words (bytes) containing codes, flags, and/or alphanumeric data, indicating the outcome of the action of one control section in processing a task-request received by it from another control section. This information is transmitted from the former control section to the latter.

Resource: Any facility of the computing system or operating system required by a job or task and including main storage, input/output devices, the central processing unit, data sets, and control and processing programs.

Reusable: The attribute of a routine that the same copy of the routine can be used by two or more tasks. (See renterable, serially reusable).

Serially reusable: The attribute of a routine that when in main storage the same copy of the routine can be used by another task after the current use has been concluded.

Service program: Any of the class of standard routines that assist in the use of a computing system and in the successful execution of problem programs, without contributing directly to control of the system or production of results, and including utilities, simulators, test and debugging routines, etc.

Severity code: A numerical rank indicating the extent of corrective action to be taken by the control processes for error or fault conditions.

Storage "pool": That area of main memory used as working storage by the various elements of the Supervisory Systems and procedure programs, and allocated for such use as a function of Memory Management.

Storage unit: The smallest contiguous area of storage "pool" is allocated in predefined blocks (storage units), not completely randomly.

Supervisor: A routine or routines executed in response to a requirement for altering or interrupting the flow of operations through the central processing unity, or for performance of input/output operations, and therefore, the medium through which the use of resources is coordinated and the flow of operations through the central processing unit is maintained; hence, a control routine that is executed in supervisor state.

Supervisory System, resident portion: That portion of the system required and kept in main memory at all times during the execution of programs under control of the system itself.

Synchronous: Occurring concurrently, and with a regular or predictable time relationship.

Task: A unit of work for the central processing unit from the standpoint of the control program; therefore, the basic multiprogramming unit under the control program.

Task-request: A set of one or more computer words (bytes), containing identifiers, pointers, flags and/or parameters; this set of information is transmitted from one control section to another when the former has a requirement for the latter to execute a particular task for it. The information specifies the particular nature of the task and supplies any necessary parameters required for execution of the task.