

PRELIMINARY

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

PRC D-1178

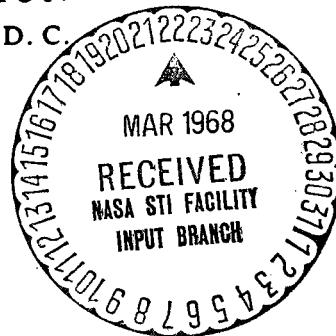
16 June 1967

Prepared for

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

FACILITY FORM 602	<u>N 68-18945</u>	(ACCESSION NUMBER)		(THRU)
	<u>97</u>	(PAGES)		(CODE)
	<u>CI-61614</u>	(NASA CR OR TMX OR AD NUMBER)		<u>OB</u>
				(CATEGORY)

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



PRELIMINARY

This document was prepared by Planning Research Corporation under Contract No. NAS8-20367, "An Airborne Evaluating Equipment Study," for the George C. Marshall Space Flight Center of the National Aeronautics & 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. CG00N8-2036710

Revision No. PRELIMINARY

Release Date 5 May 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 W. H. Stanley
(Planning Research Corp.)

Approved by _____

Date 16 June 1967

Approval Date _____

Contract Number NAS8-20367

PLANNING RESEARCH CORPORATION
SPECIFICATION CHANGE LOG
Specification No. _____

As Of _____
Superseding _____
Page I-iii of 79

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

Specification No. CG00N8-2036710

PLANNING RESEARCH CORPORATION

Rev No. _____

Page I-iv of 79

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

	<u>Page</u>
1.0 SCOPE	I-1 of 79
2.0 APPLICABLE DOCUMENTS	I-2 of 79
3.0 REQUIREMENTS	I-4 of 79
3.1 Performance	I-4 of 79
3.1.1 System Requirements	I-5 of 79
3.1.2 Operational Requirements	I-5 of 79
3.1.2.1 Function 1: System Management	I-7 of 79
3.1.2.1.1 Source and Types of Inputs to System Management Functions	I-8 of 79
3.1.2.1.2 Destination and Type of Outputs for System Management	I-9 of 79
3.1.2.1.3 Information Processing for System Management	I-9 of 79
3.1.2.2 Function 2: Memory Management	I-14 of 79
3.1.2.2.1 Source and Types of Inputs for Memory Management	I-17 of 79
3.1.2.2.2 Destination and Types of Outputs for Memory Management.	I-19 of 79
3.1.2.2.3 Information Processing for Memory Management	I-20 of 79
3.1.2.3 Function 3: Language Interpretation	I-24 of 79
3.1.2.3.1 Sources and Types of Inputs for Language Translation	I-24 of 79
3.1.2.3.2 Destination and Types of Outputs for Language Translation	I-25 of 79
3.1.2.3.3 Information Processing for Language Translation.	I-26 of 79

TABLE OF CONTENTS
(Continued)

	<u>Page</u>
3.1.2.4 Function 4: Procedure Management.	I-27 of 79
3.1.2.4.1 Source and Type of Inputs for Procedure Management	I-27 of 79
3.1.2.4.2 Destination and Type of Outputs for Procedure Management	I-30 of 79
3.1.2.4.3 Information Processing for Procedure Management	I-31 of 79
3.1.2.5 Function 5: Schedule Management	I-33 of 79
3.1.2.5.1 Source and Type of Inputs for Schedule Management.	I-35 of 79
3.1.2.5.2 Destination and Type of Outputs for Schedule Management	I-36 of 79
3.1.2.5.3 Information Processing for Schedule Management	I-37 of 79
3.1.2.6 Function 6: Interruption Management	I-40 of 79
3.1.2.6.1 Source and Type of Inputs for Interruption Management	I-40 of 79
3.1.2.6.2 Destination and Types of Outputs for Interruption Management.	I-42 of 79
3.1.2.6.3 Information Processing for Interrupt Processing.	I-43 of 79
3.1.2.7 Function 7: Console Communication	I-45 of 79
3.1.2.7.1 Sources and Types of Inputs for Console Communication	I-45 of 79
3.1.2.7.2 Destination and Types of Outputs for Console Communication.	I-48 of 79

TABLE OF CONTENTS
(Continued)

	<u>Page</u>
3.1.2.7.3 Information Processing for Console Communication	I-49 of 79
3.1.2.8 Function 8: CIU/Signal Adapter Communication	I-52 of 79
3.1.2.8.1 Source and Types of Inputs for CIU/Signal Adapter Communication	I-54 of 79
3.1.2.8.2 Source and Types of Outputs for CIU/Signal Adapter Communication . .	I-55 of 79
3.1.2.8.3 Information Processing for CIU/Signal Adapter Communication	I-55 of 79
3.1.2.9 Function 9: Uplink Communication	I-56 of 79
3.1.2.9.1 Sources and Types of Inputs for Uplink Communication	I-57 of 79
3.1.2.9.2 Destination and Types of Outputs for Uplink Communication	I-58 of 79
3.1.2.9.3 Information Processing for Uplink Communication	I-59 of 79
3.1.2.10 Function 10: Downlink Communication	I-60 of 79
3.1.2.10.1 Sources and Types of Inputs to the Downlink Communication Function.	I-60 of 79
3.1.2.10.2 Sources and Types of Outputs for Downlink Communication	I-61 of 79
3.1.2.10.3 Information Processing for Downlink Communication	I-62 of 79
3.1.3 Data Base Requirements	I-63 of 79
3.1.4 Human Performance.	I-64 of 79
3.2 CPCEI Definition	I-64 of 79

TABLE OF CONTENTS
(Continued)

	<u>Page</u>
3.2.1 Interface Requirements	I-64 of 79
3.2.1.1 Interface Block Diagram	I-65 of 79
3.2.1.2 Detailed Interface Definition	I-65 of 79
3.2.1.2.1 OCDM Support System CPCEI.	I-65 of 79
3.2.1.2.2 OCDMS Experiment Procedure CPCEI	I-67 of 79
3.2.1.2.3 OCDMS Computer and Associated Peripheral Equipment	I-67 of 79
3.2.1.2.3.1 Computer	I-67 of 79
3.2.1.2.3.2 Manufacturer Supplied Programming Language(s)	I-67 of 79
3.2.1.2.3.3 Control/Display Unit	I-67 of 79
3.2.1.2.3.4 Bulk Storage	I-67 of 79
3.2.1.2.4 OCDMS Experimental Hardware	I-67 of 79
3.2.1.2.5 Operations Personnel	I-67 of 79
3.2.1.2.6 Ground Stations (PIF, ACE, GOSS)	I-68 of 79
3.2.2 Government-Furnished Property List	I-68 of 79
3.3 Design Requirements	I-68 of 79
3.3.1 Programming Standards	I-69 of 79
3.3.2 Program Design	I-69 of 79
3.3.3 Program Modification	I-69 of 79
3.3.4 CPCEI Testing Facilities	I-69 of 79
3.3.5 CPCEI Expandability	I-70 of 79

TABLE OF CONTENTS
(Continued)

	<u>Page</u>
4.0 QUALITY ASSURANCE PROVISIONS	I-71 of 79
4.1 Implementation Test Requirements	I-71 of 79
4.1.1 Design and Development Testing	I-73 of 79
4.1.2 Preliminary Qualification Test	I-77 of 79
4.1.2.1 Qualification Test Requirements	I-77 of 79
4.1.2.2 Resources Required for Testing	I-77 of 79
4.1.2.3 Test Schedules and Locations	I-77 of 79
4.1.3 Special Test Requirements	I-77 of 79
4.2 Integration Test Requirements	I-77 of 79
4.2.1 General	I-77 of 79
4.2.1.1 Sequence of Tests	I-78 of 79
4.2.1.2 Functions to be Tested	I-78 of 79
4.2.1.3 Testing Environment	I-78 of 79
4.2.1.4 Support Computer Programs Required	I-78 of 79
4.2.1.5 Personnel Required	I-78 of 79
4.2.1.6 Equipment Required	I-78 of 79
4.2.2 Acceptance/Qualification Test	I-78 of 79
4.2.2.1 Sequence of Tests	I-78 of 79
4.2.2.2 Functions to be Tested	I-78 of 79
4.2.2.3 Testing Environment	I-79 of 79

Specification No. CG00N008-2036710

Revision No. _____

Page _____

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 *J. H. Stouder*
(Planning Research Corp.)
for C. E. Riley

Date *16 June 1967*

Approved by _____

Approval Date _____

Contract Number NAS8-20367

TABLE OF CONTENTS
(Continued)

	<u>Page</u>
4.2.2.4 Support Computer Programs Required . .	I-79 of 79
4.2.2.5 Personnel Required	I-79 of 79
4.2.2.6 Equipment Required	I-79 of 79
6.0 NOTES	I-79 of 79
10.0 APPENDIX	I-79 of 79

LIST OF EXHIBITS

	<u>Page</u>
1. Functional Operation of OCDMS Supervisory System .	I-6 of 79
2a. System Management (System Initialization)	I-11 of 79
2b. System Management (System Termination)	I-13 of 79
2c. System Management (System Status/Accounting)	I-15 of 79
2d. System Management (Queued Interruption Processing)	I-16 of 79
3. Memory Management	I-18 of 79
4. Language Interpreter (To be added)	I-24a of 79
5a. Procedure Management	I-28 of 79
5b. Procedure Management	I-29 of 79
6a. Scheduling Management	I-34 of 79
6b. Scheduling Management	I-39 of 79
7. Interruption Processing	I-41 of 79
8. Interrupts Allowed	I-44 of 79
9a. Console Communication (Printer & Keyboard)	I-46 of 79
9b. Console Communication (EL Display)	I-47 of 79
10. CIU/Signal Adapter Communication	I-53 of 79
11. OCDMS Supervisory System Interface	I-66 of 79

1.0 SCOPE

This part of this specification establishes requirements for performance, design, test, and qualification of a computer program identified as a supervisory system for On-Board Check-out and Data Management System (OCDMS), CEI O36710A. This CPCEI is used to provide a space-borne vehicle, computer-based system with the functional capability to:

- a) control system input and output,
- b) manage data entities,
- c) assess priorities of system tasks,
- d) schedule system tasks,
- e) respond to interrupt signals,
- f) detect and recover from error or unusual conditions,
- g) allocate computer storage,
- h) queue messages.

The CPCEI requires, as a basis of its operations, a pre-processed 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 detail contents of Sections 3, 4, and 10, 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 Onboard Checkout and Data Mangement, General Specifications for, Specification No.SS2036701A , dated 29 March 1967.

2.2.2 Performance/Design and Product Configuration Requirements, Support System for Onboard 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 Mangement 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, Onboard Checkout System Software Design, 17 November 1966

2.3.4 SR-QUAL-65-48; NASA, (MSFC), Directives for Software Development, 15 December 1965.

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

Specification No. CG00N8-2036710

Revision No. _____

Page I-3 of 79

2.3.6 PRC D-1417, Technical Advisement Memorandum No. 171-4
Onboard Checkout and Data Management System Control
and Display Unit: A Preliminary Study, 7 April 1967

3.0 REQUIREMENTS

The OCDMS system requirements include performance requirements, design and construction requirements, and requirements for functional areas. These requirements define and control a space-borne 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 system specification or the OCDMS Support System. (References 2.2.1 and 2.2.2).

3.1 Performance

Pertinent performance requirements established by the references include:

- Stimuli generation and application;
- Response measurement;
- On-line operation communications;
- Up-link/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 System Requirements:

- Reliability
- Maintainability
- Useful Life
- Human Performance
- Safety
- Environmental Constraints

3.1.1 System Requirements

The limits and/or capacities of the OCDMS Supervisory System performance shall be constrained to operational envelopes including system self-management, 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 Software (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 Supervisory CPCEI is represented by the functional block diagram of Exhibit 1. 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
CFU control

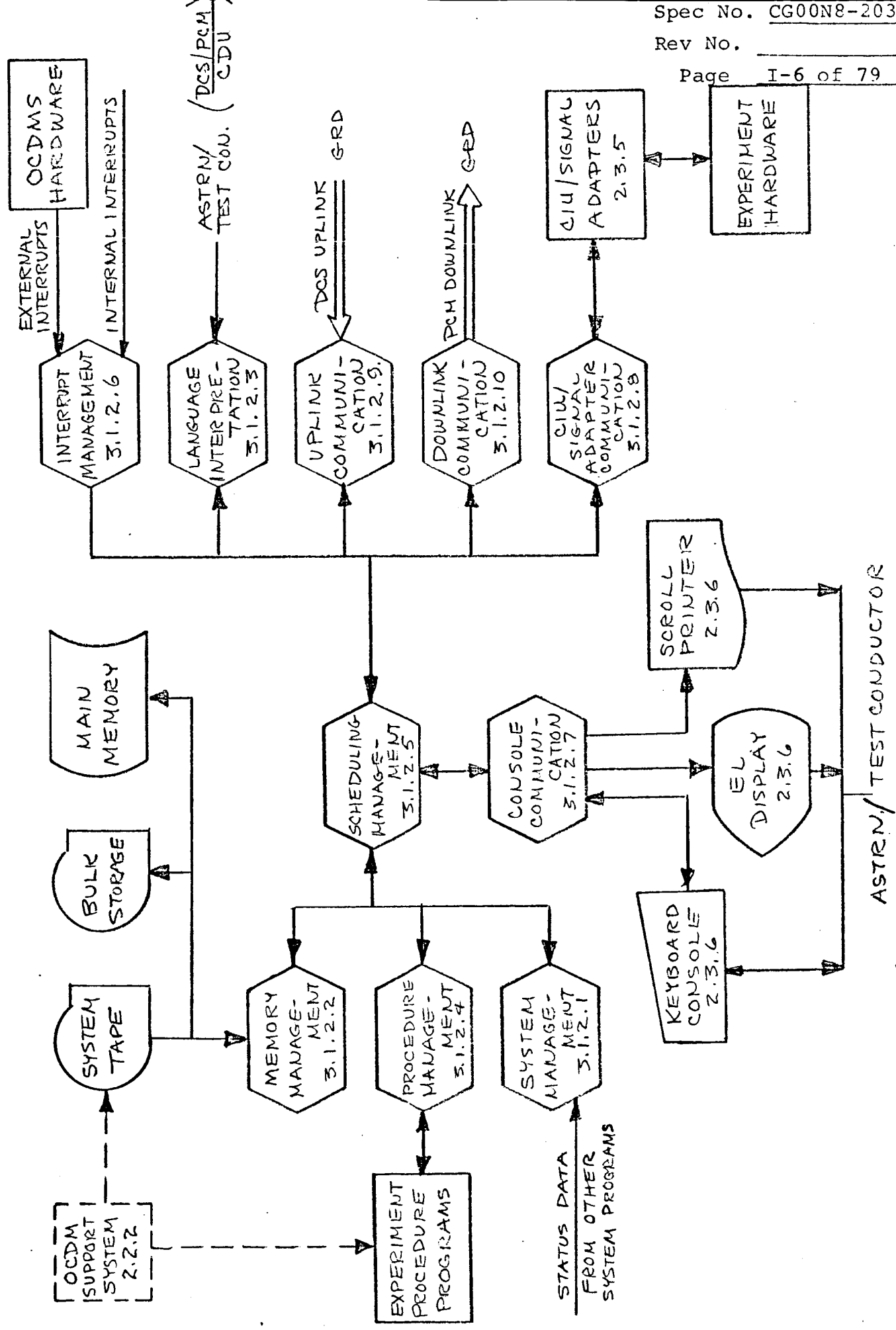
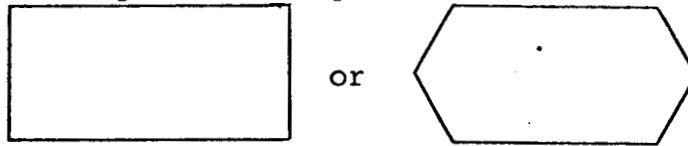


EXHIBIT I FUNCTIONAL OPERATION OF OCDS SUPERVISORY SYSTEM

- A double line (\rightleftarrows) 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 not as a restrictive design definition of computer program component (CPC) organization or as functional descriptions of particular main or subprograms.

3.1.2.1 Function 1: System Mangement

The system management function shall provide the overall executive control for the CPCEI. The particular performance criterion for computer program components (CPCs) covered by this function include:

- Initialization of the CPCEI
- Termination of the CPCEI
- Input/Output Resource Allocation
- 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 to System Management Functions

a) Functional inputs shall typically be obtained as preprocessed outputs generated by the OCDMS Support System software, and the information shall generally originate at sources which will include:

- Keyboard and manual control switch signals
 - Storage index lists of all programs and data sets (main and bulk storage)
 - Command/control routine macro-call instructions for I/O requests
 - Load element task control blocks
 - Status-Word for active programs
 - Status-Word for non-active programs
 - Status-Word for busy programs
 - Status-Word for program detected errors
 - Miscellaneous flags and other entries from command/control routines
- b) Units of Measure - TO be determined
- c) Limits/Ranges - TO be determined
- d) Accuracy/Precision - TO be determined
- e) Arrival Frequency - TO be determined

3.1.2.1.2 Destination and Types of Outputs for System Management

a) Functional output shall consist of, but not necessarily be limited to, the following actions:

- Collect and save descriptions of systems status--System Status and Configuration Dictionary (SCDY)
 - Collect and save system statistics
 - System status displays, reports, messages, and accounting and events records to the operations personnel via the display and/or printer, or transfer of this data to bulk storage
 - Control signals to the hardware to implement equipment shutdown.
- 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

3.1.2.1.3 Information Processing for Systems Management

a) System Initialization Subprogram Function

The system shall be capable of starting or re-starting (after power shutdown) system activity in response to an interruption initiated by the operations personnel via

the Control Display Unit (CDU), by a Digital Command System (DCS) uplink message, or by the wake-up timer (previously set by the system). When entered, this subprogram will proceed to fetch into core, from bulk storage, those CPC's containing that portion of the program necessary for start-up. A definition of the required CPCs shall be provided by header-information on the bulk storage media, by the operations personnel, and/or by the System Status and Configuration Dictionary. In addition to loading the necessary modules into main storage, system initialization will involve the execution of self-check and error detection/recovery routines to a level necessary to establish confidence in the computer, CIU, auxiliary memory devices, and CDU. 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). The SCDY will contain a profile of the system indicating processing load, memory utilization, peripheral equipment utilization (I/O load), and hardware status. This together with the Unit

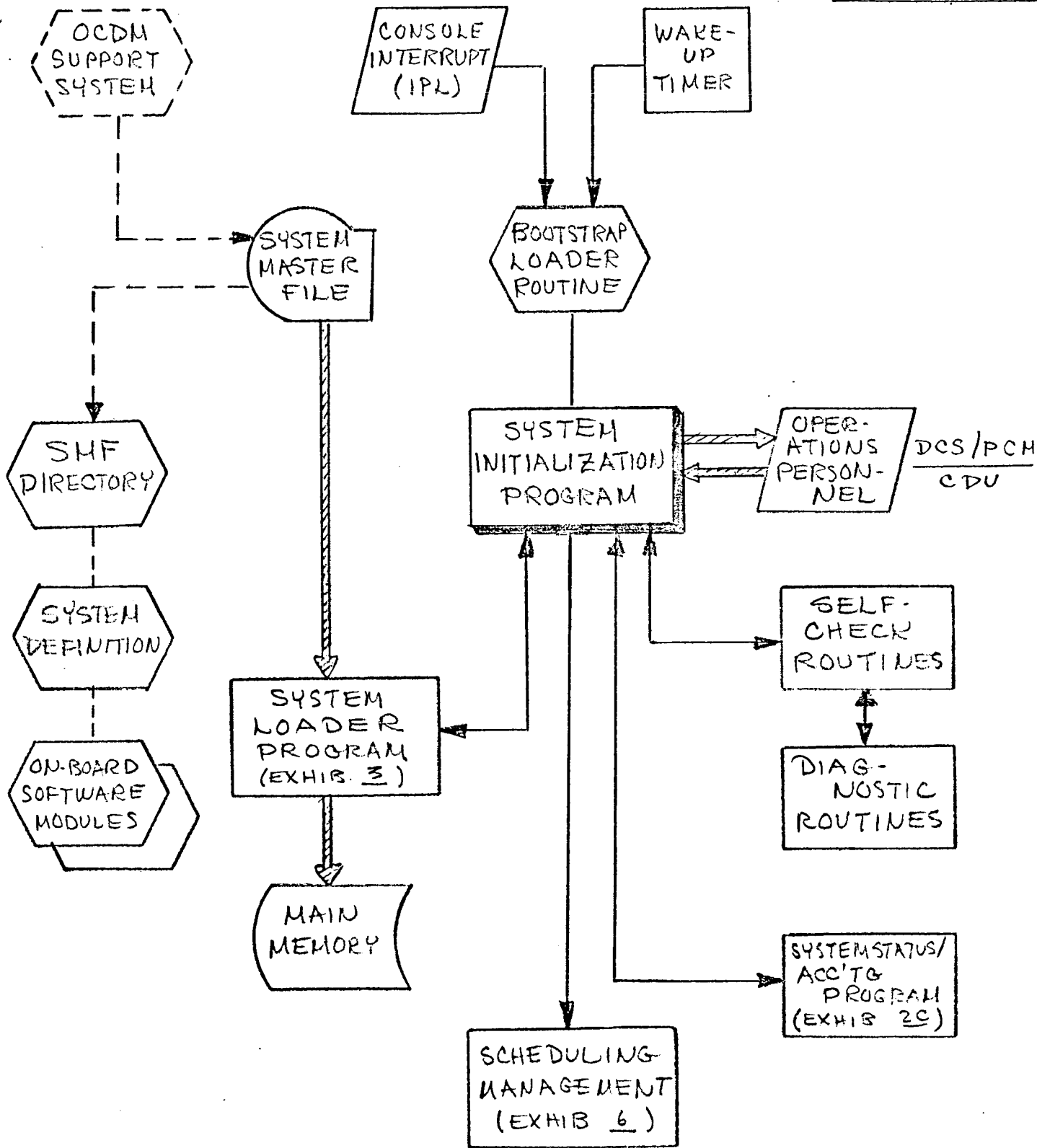


EXHIBIT 2a

SYSTEM MANAGEMENT (SYSTEM INITIALIZATION)

Control Blocks and Procedure Status Files shall provide a complete description of the status and configuration of the on-board system. This information will be continuously summarized into a form suitable for output to the operations personnel and saved on bulk storage as an accounting events record.

Additionally, the system shall recognize potential problems such as an overload on a system program or main and auxiliary storage, and adjust the system execution preference to compensate for these conditions. Subprograms shall also respond to hardware interruptions indicating system error conditions. Actions, in these cases, may cause a reconfiguration of the system to attempt a "work around" to the source of error (e.g., failure of an I/O 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 onto bulk storage. They

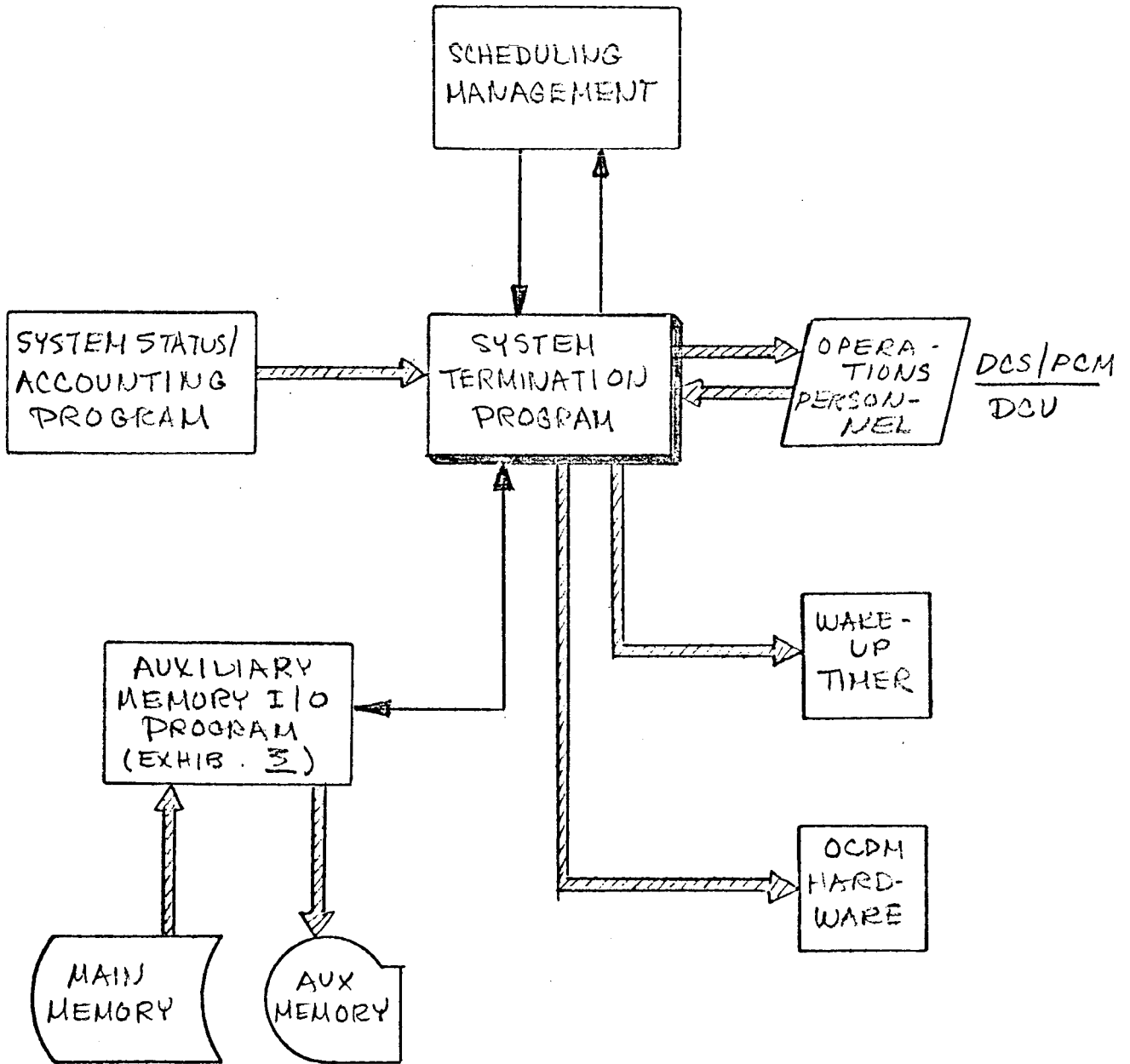


EXHIBIT 2b

SYSTEM MANAGEMENT (SYSTEM TERMINATION)

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 the normal program sequence in the event of emergencies.

The routines implementing this system function shall 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 non-immediate interruption signals. The function of this subprogram shall be to examine this queue and call and/or release control to the routines(s) required to process the interruption signal. This subprogram will receive control on a regular basis as a function of Scheduling Management (see section 3.1.2.5). Exhibit 2d illustrates the functional operation of this subprogram.

3.1.2.2 Function 2: Memory Mangement

The memory management function shall provide for dynamic storage allocation and recovery of storage after task completions. Program and data set transfers between

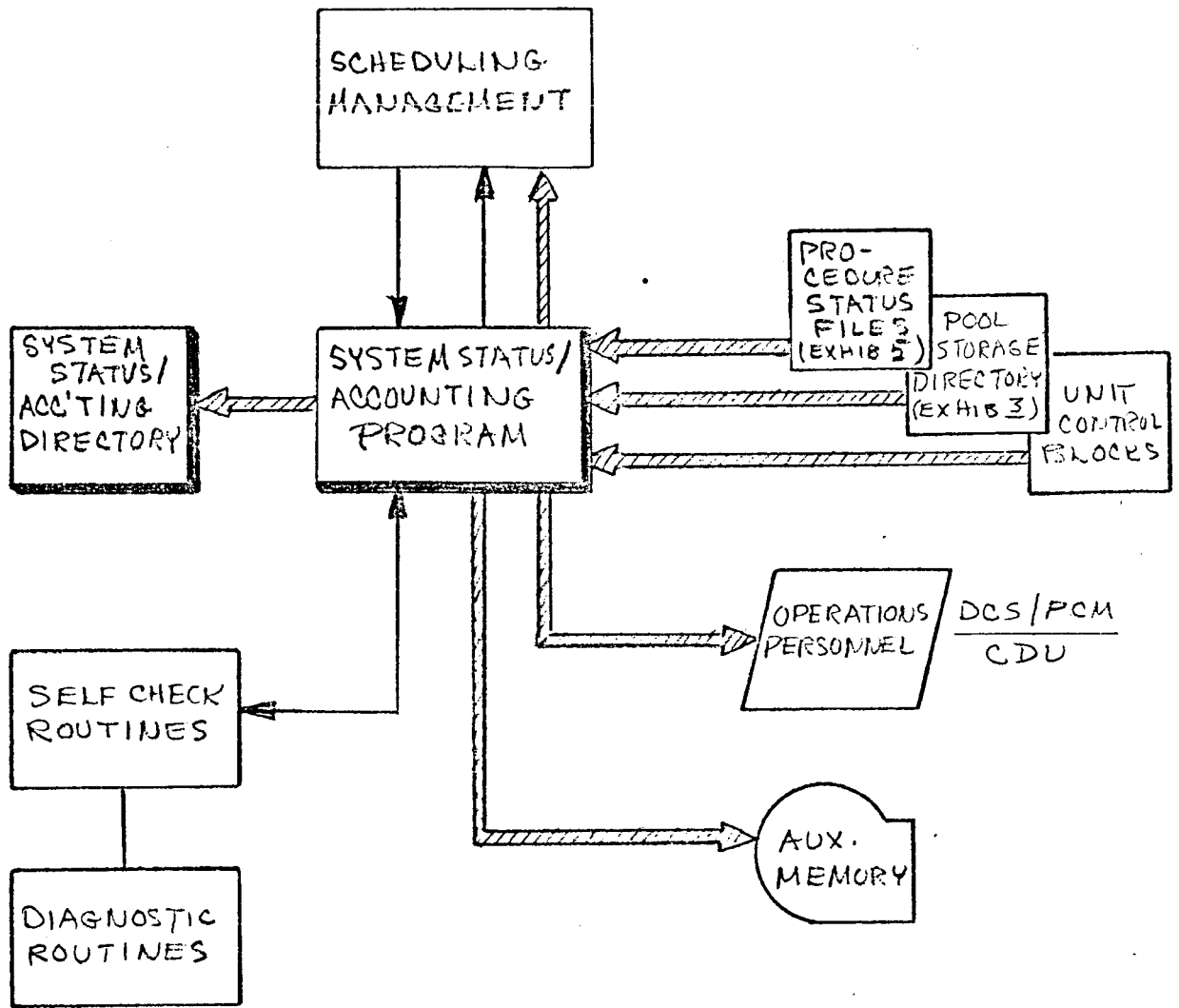


EXHIBIT 2C

SYSTEM MANAGEMENT (SYSTEM STATUS/ACCOUNTING)

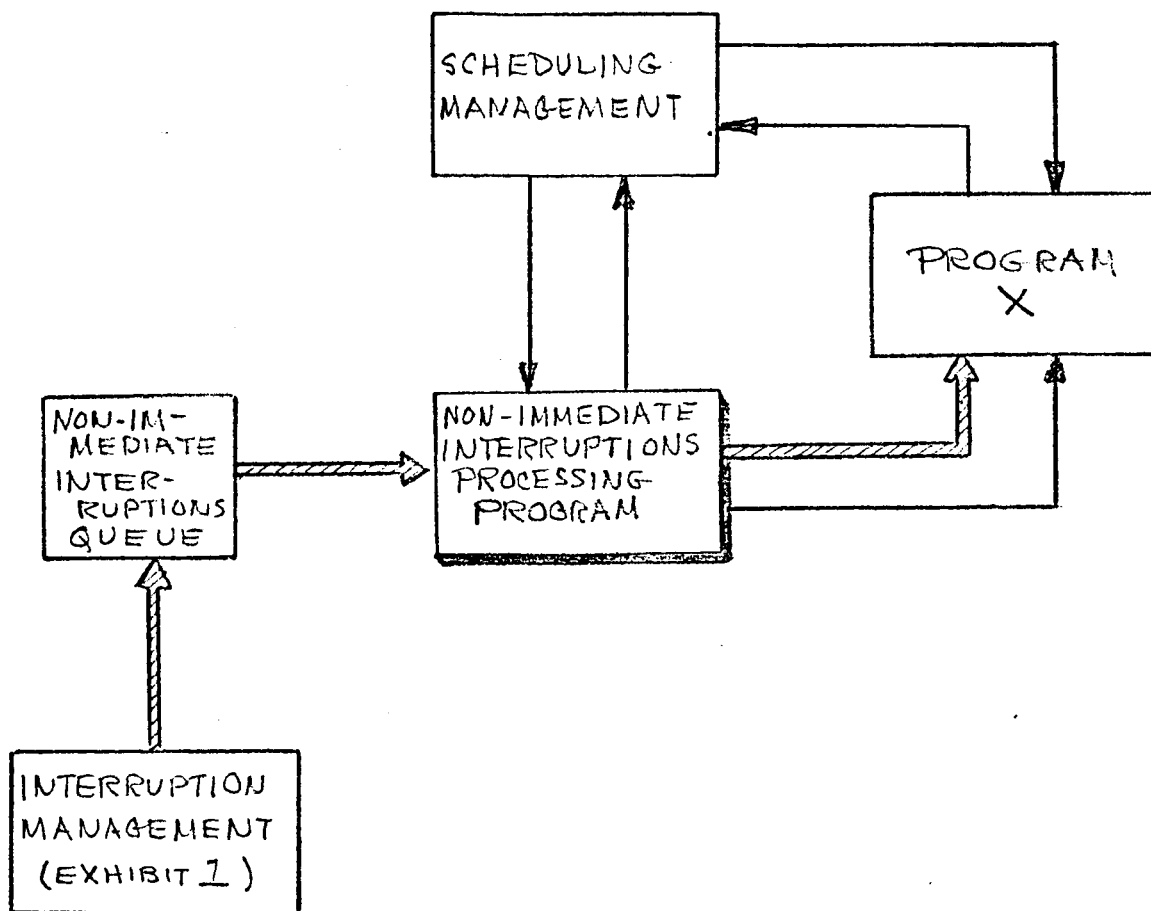


EXHIBIT 2d

SYSTEM MANAGEMENT (QUEUED INTERRUPTION PROCESSING)

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 typically be obtained from information sources that shall include the following:

- Macro-instructions with GET/RETURN storage units from/to the storage pool,
 - Compliance information from "called" system programs concerning task-requests generated by Memory Management routines;
 - Description of the contents of main storage (main Storage Directory);
 - Description of the contents of bulk storage (Auxiliary Storage Directory);
 - Description of system status (System Status and Configuration Directory);
 - Control signals from auxiliary storage devices;
 - Information transfers (load modules) from auxiliary storage devices.
- b) Units of Measure- TO Be Determined
- c) Limits/Range - TO Be Determined

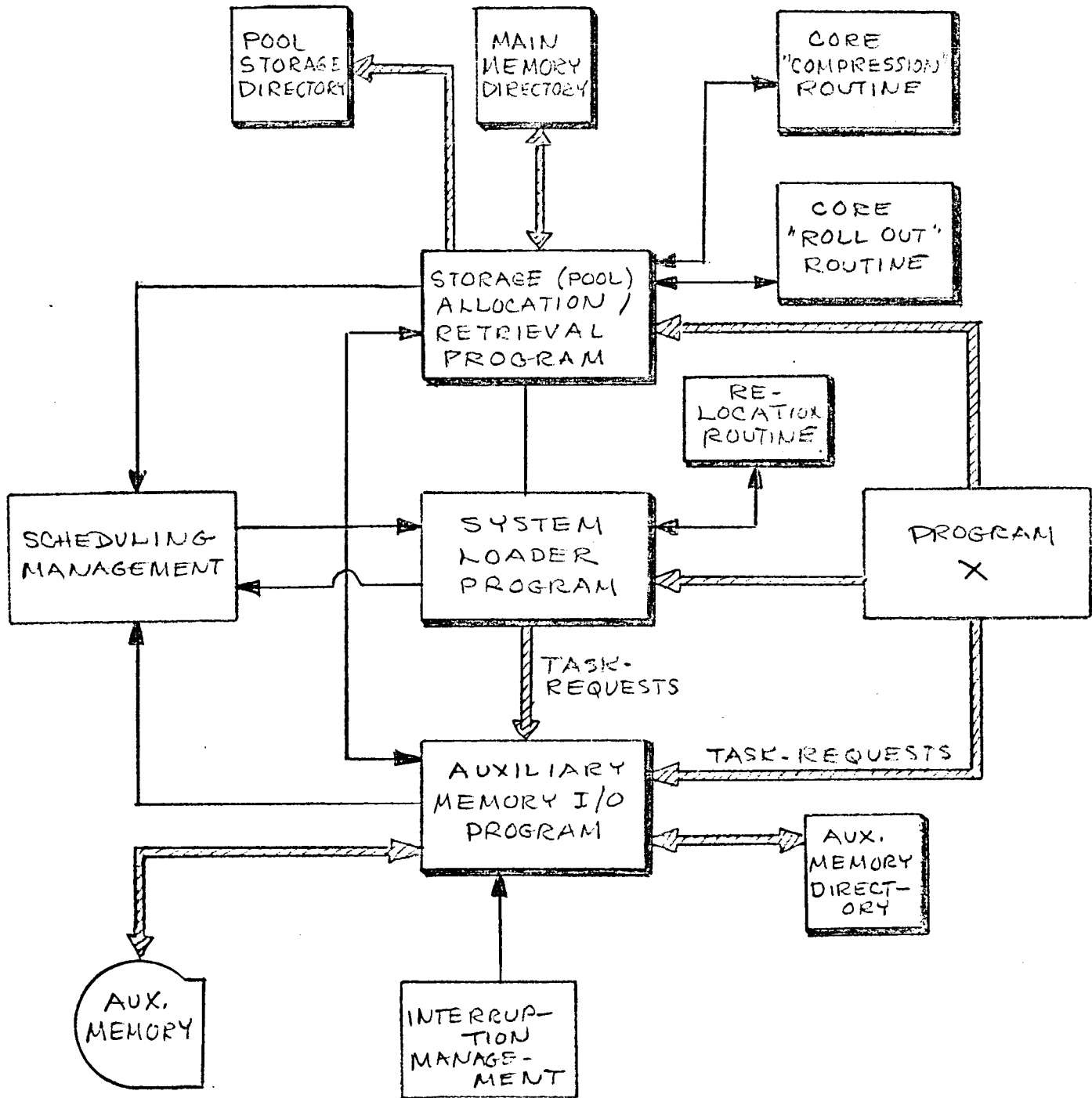


EXHIBIT 3
MEMORY MANAGEMENT

d) Accuracy/Precision - TO Be Determined

e) Arrival Frequency - TO Be Determined

3.1.2.2.2 Destination and Type of Outputs for Memory Management

a) Functional outputs shall consist of,

but not necessarily be limited to, the following:

- GET/RETURN storage units from/to the storage pool;
- loading of specified load modules;
- outputting of data sets to bulk storage;
- compliance information to various system programs, concerning task-request directed by them to the Memory Management routines;
- modifications to the Main and Auxiliary Storage Directories;
- control signals to Auxiliary storage devices;
- information transfers to auxiliary storage devices.

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

3.1.2.2.3 Information Processing For Memory Management

The memory management function shall be implemented by subprograms and routines which perform as follows:

a) Loading - During system initialization, the loader subprogram will be used implicitly by the System Initialization subprogram, otherwise it will be used explicitly by various system programs via scheduled task-requests. When the Loader receives control, it will first respond to all compliance-indications received from the bulk storage management subprogram; thereby, eliminating redundant load requests.

Actions which are involved include the following:

- Adjusting any location dependent quantities within a module, using the relocation factor specified in the module's header file;
- Updating the Main Storage Directory with the label and location of the module;
- Issuing a compliance-indication to the program requesting the module. This shall include as a minimum, the label and location of the module. If a module can not be located, the indication will specify the cause.

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

- Obtaining storage space for the module;
- Sending a task-request to the bulk 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 compliance-indicators are to be posted.
- Transfer of control to the schedule management program.

b) Main Storage Management Subprogram Functions

The main storage management CPCs shall handle any necessary interface between main memory and other executable sections of the system. These shall be entered from the schedule management routines, and will be capable of the following operations:

- Accepting, queuing, and responding to task requests from other system programs concerning the allocation or return of storage area in main-memory;

- Allocating storage areas from a pre-defined storage pool (defined at system generation time) 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 interface with the relocation routine of the Loader.
- Returning allocated storage area to the pool either on request from an applicable program or on the basis of a supervisory clean-up activity;
- Editing and updating of the status records of the storage pool to be used by the system management program;
- Providing compliance indications to programs, specifying either the location of an allocated buffer or positive conditions under which the request must be reinitiated.

c) Bulk Storage Management Subprograms
Functions

The characteristics of bulk storage management are essentially identical with the Main Storage management performance. The routines shall be capable of the following operations:

- Accepting, queuing and responding to task-request from other functional routines;
- Transmitting/receiving control signals and data transfers to and from bulk storage devices;
- Maintaining an auxiliary storage dictionary indicating relative location of load modules, amount of storage space available, and the hardware status of the device;
- Providing compliance indicators to programs specifying the status of requests;
- Reformatting data on bulk storage devices to utilize device more efficiently.

3.1.2.3 Function 3: Language Interpretation

The language interpretation function shall be implemented to provide translation and execution of online 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 calls from application programs).

3.1.2.3.1 Sources and Types of Inputs for Language Translation

a) Functional inputs shall typically be obtained from information sources that shall include the following:

- Labeled procedure steps, substeps and other procedure hold-points (See 3.1.2.4)
- Computer Interface Unit function key signals (See 3.1.2.7)
- Real-time commands via the Apollo Digital Command System (See 3.1.2.9)
- Basic elements to be interpreted shall include; letters, digits, identifiers, numbers, and strings

Specification No. CG00N8-2036710

Revision No. _____

Page I-24a of 79

EXHIBIT 4 - LANGUAGE INTERPRETER

To be Added

- Expressions to be interpreted shall include; variables, arithmetic, Boolean, and the CIU function keys.
- Statements to be interpreted (to the extent of parameter modification) shall include procedural, conditional, assignment and GO-TO-types
- b) Units of Measure - TO Be Determined
- c) Limits/Ranges - TO Be Determined
- d) Accuracy/Precision - TO Be Determined
- e) Arrival Frequency - TO Be Determined

3.1.2.3.2 Destination and Types of Outputs for Language Translation

- a) Functional outputs shall consist of, but not necessarily be limited to, the following:
- Create and attach tasks to existing application procedures;
 - Linkage and data transfer macro-instructions;
 - Generated command/control routines calling sequences;
 - Supervisory system executive table entries;

- Definition of data control blocks, task control blocks, and for access to given fields within these blocks.
- 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

3.1.2.3.3 Information Processing for Language Translation

- a) The language processing criteria shall be in accordance with paragraph 3.1.2.1.3, Reference 2.2.2.
- b) The interpreter expression decomposition logic shall be in accordance with paragraph 3.1.2.1.3.3, Reference 2.2.2.
- c) Supervisor parameters and table update generation logic shall be in accordance with paragraph 3.1.2.1.2.4, Reference 2.2.2.
- d) 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 paragraph 3.1.2.1.2.5 and 3.1.2.1.3.6, Reference 2.2.2.

3.1.2.4 Function 4: Procedure Management

The processes implementing Procedure Management shall provide all supervisory services and controls for the Experiment Procedure Programs and other application programs pre-processed by the OCDMS Support System. Program schedules, inter-communication between procedures, and the coordination of all mechanics pre-requisite 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 Type of Inputs for Procedure Management

a) Functional inputs shall typically be obtained from information sources that shall include the following:

- Experiment Procedure Programs- which shall consist of calling sequences and sets of executable code as described by section 3.1.2.3 and paragraph 3.1.2.1.3.5 of reference 2.2.2.

- Procedure Status Files- which shall consist of sets of data relating the status and identification of the procedure parameters. This information will consist of at least the following:

- (1) procedure ID,
- (2) associated signal adapter(s) ID,

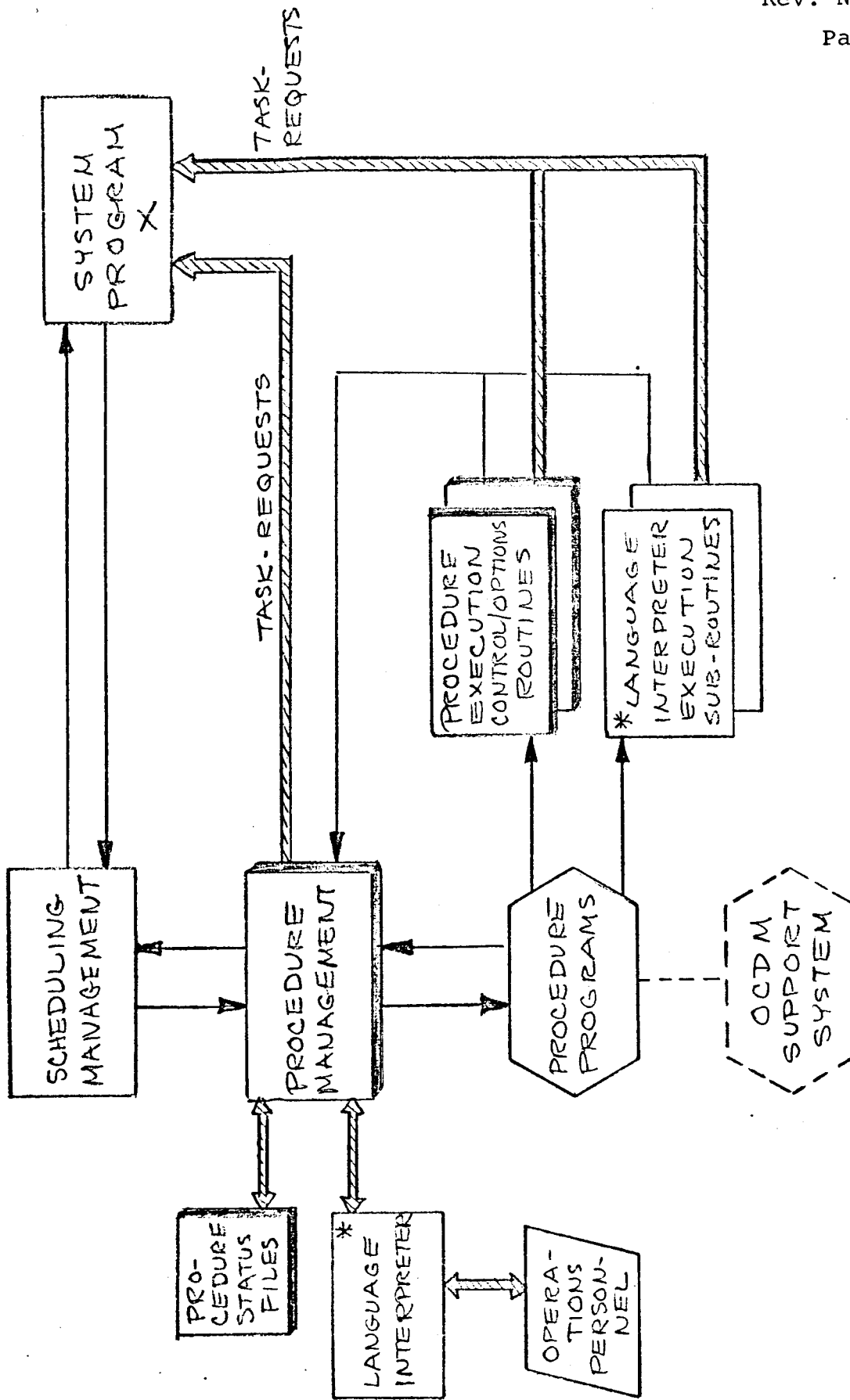


EXHIBIT 50
PROCEDURE MANAGEMENT

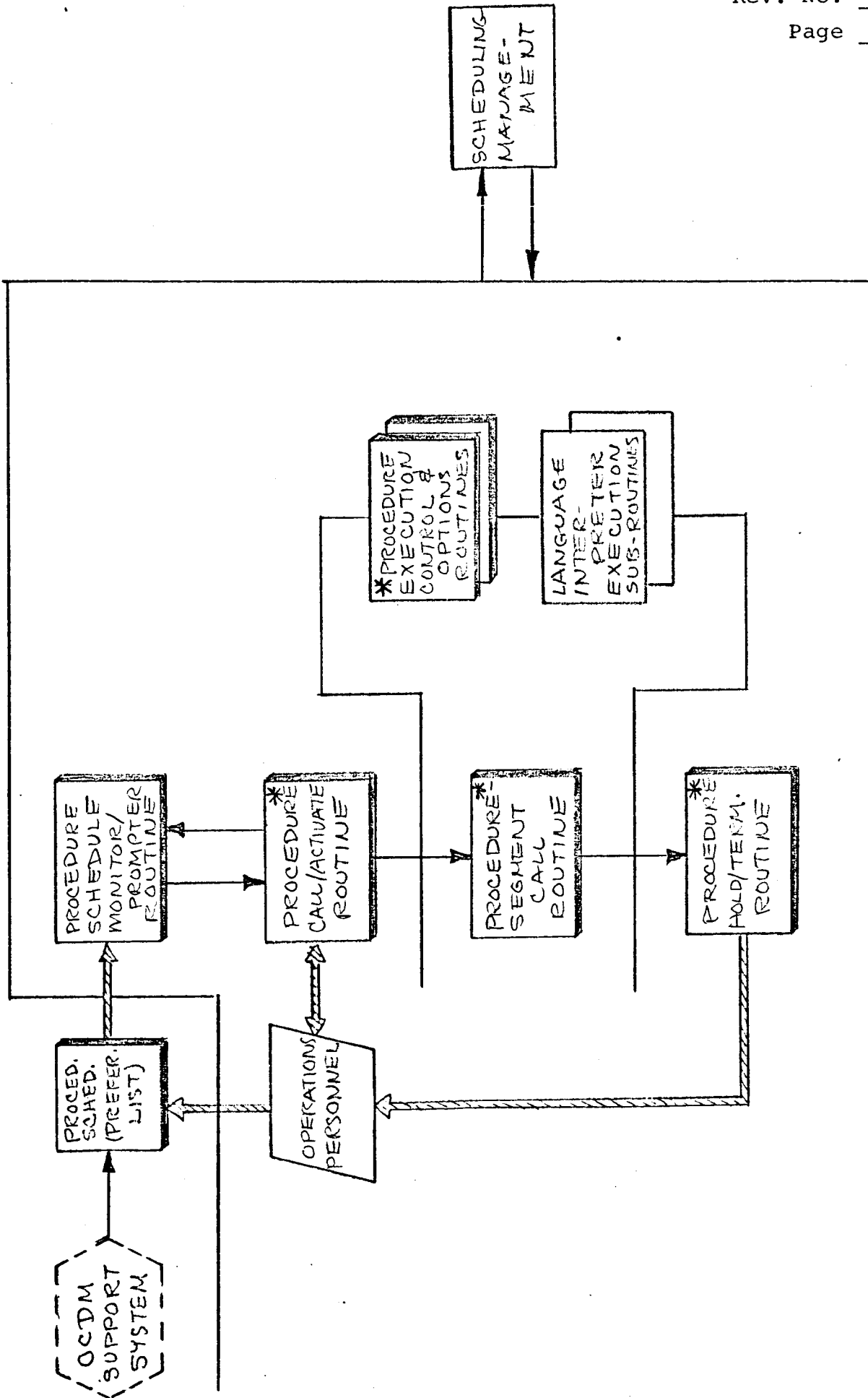


EXHIBIT 5b

PROCEDURE MANAGEMENT

- (3) fixed-storage locations,
- (4) allocated-storage locations,
- (5) relocation factors,
- (6) current PSW (or equivalent),
- (7) prerequisite information,
- (8) compatibility codes (indicates other, compatible procedures which may be executed concurrently).

- Procedure Schedule File (precedence List) - which shall consist of an array indicating the conditions (range time, vehicle position, hardware status, preceding programs, etc.) pre-requisite for turn-on and execution of a given procedure. This information shall be compiled by the OCDMS Support System and up-dated on-line as necessary (see 3.1.2.7).

- b) Units of measure- TO Be Determined
- c) Limits/ranges- TO Be Determined
- d) Accuracy/precision- TO Be Determined
- e) Arrival frequency- TO Be Determined

3.1.2.4.2 Destination and Type of Outputs

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

- Procedure status data to the Procedure Status Files and Scheduling Management function;

- 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 - 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

3.1.2.4.3 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 the conditions specified by the Procedure Schedule File as prerequisites for procedure turn on/off. When conditions are satisfied, a prompting message shall be issued to the console operator 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 whether the present scheduling load will allow execution of that procedure. Communications with the operator in conjunction within 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 Status File, and generating an appropriate Scheduling Parameter and transferring it to the Scheduling Management routine.

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

(1) manage the Procedure Status File; ie. update that information pertaining to allocated-storage locations, current PSW (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 perform all overlay functions;

(4) transfer data sets and/or allocated storage from one procedure to another;

(5) receive control as a function of scheduling management and activate the proper procedure.

d) execution options-during initialization and/or at pre-defined hold-points in a procedure, the operations personnel shall be given an 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 task to perform. The programmed processes shall provide the mechanics to commutate control sections in a fixed order to perform an increment of processing (during which I/o interrupts are generally not recognized). Control shall be returned to system and schedule management routines from points determined by the experiment procedure, subject to the constraint that the elapsed time within an operational program shall not exceed a specified amount. This constraint is implemented by programming convention, not by hardware timer interrupt. Exhibit 6a illustrates the operation of this function.

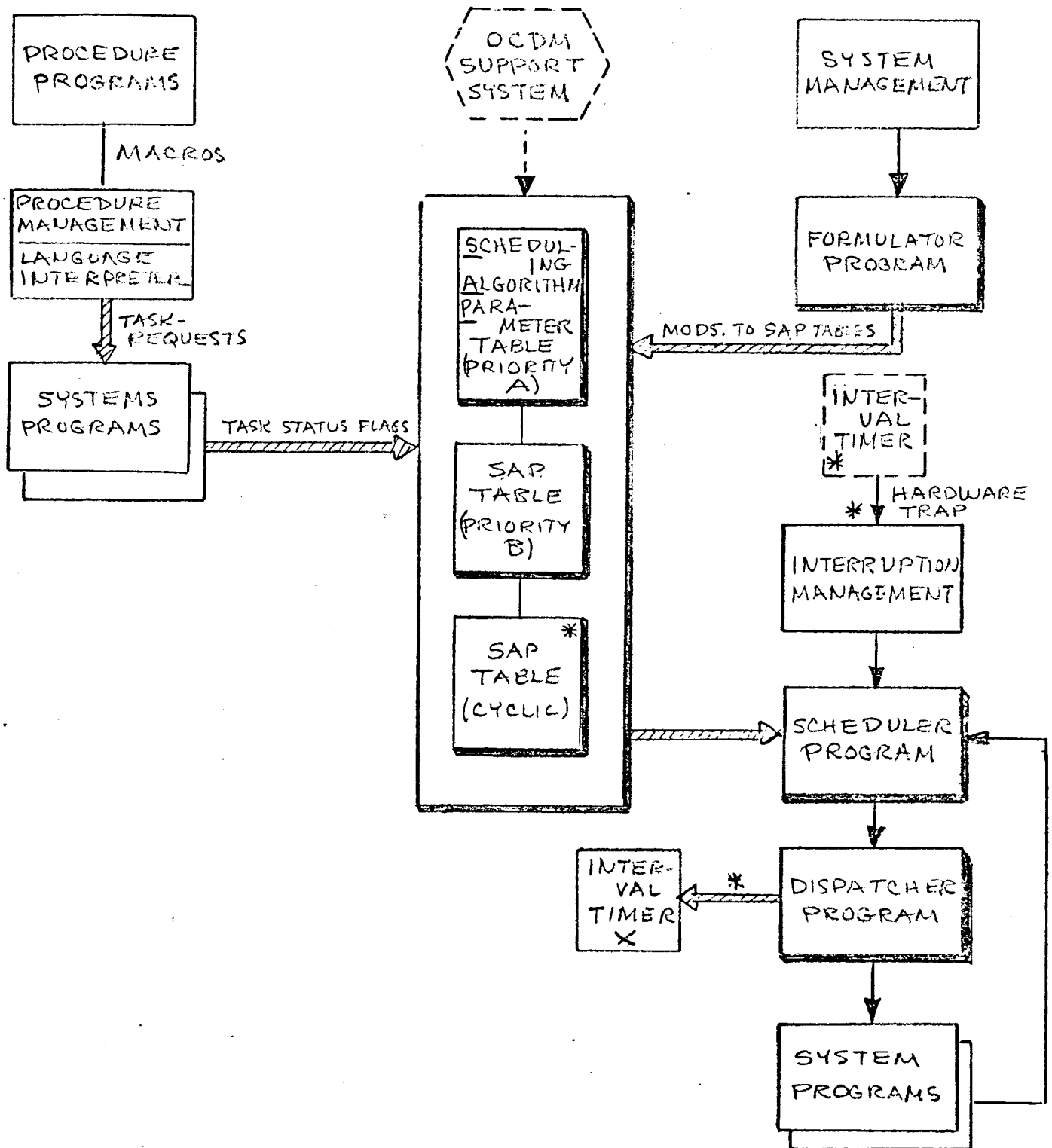


EXHIBIT 6a

SCHEDULING MANAGEMENT

3.1.2.5.1 Source and Type Of Inputs For Schedule Management

a) Functional inputs shall typically be obtained from information sources that shall include the following:

- Scheduling Algorithm Parameter Table- a list by which the Scheduler communicates between operational programs. Presence on this list is a necessary condition for any control segment to receive processing increments from the Scheduler. The table will contain subsets of parameters depending on the priority class of the control segment.

- Scheduling Algorithm Parameters (SAP)- In general, a SAP will contain a flag (task-status flag) indicating whether the related program has an active task to perform, a pointer to the program, the label associated with the program, and periodic time value for reptitious tasks.

- SAP Table Changes- Parameter changes shall be generated implicitly by the System Mangement and Procedure Management functions and explicitly communicated by the Procedure Management function as directed by the system operator personnel.

- Control signals (task-status flag settings)- which are set/reset by the associated control segment (indicating an active task or an inactive task).

b) Units of measure - TO Be Determined

- c) Limits/ranges - TO be determined
- d) Accuracy/precision - TO be determined
- e) Arrival frequency- TO be determined

3.1.2.5.2 Destination and Type of Outputs for Schedule Management

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

- Modification to the scheduling algorithm-internal to the Schedule management routines.
- SAP address-This will be the absolute address of a given SAP and will be transmitted to its related program when processing of a task request has resulted in the creation of the SAP by the Schedule Management function.
- 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.
- 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 - 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

3.1.2.5.3 Information Processing for Schedule Management

The set of programmed processes making up the schedule management function shall be grouped into two major subprograms; the Formulator and the Sequencer subprograms.

a) The Formulator subprogram shall contain the entry points within the Schedule Management Routine to which other sections of the system have transferred. One group of entry points shall exist for periodic control sections (those requiring control at regular intervals) and another group shall be available for non-periodic programs. For mission requirements dictating both priority and normal scheduling, additional entry point groups may be defined.

- "Periodic" Entry - When the Formulator is entered from a periodic control segment, it will identify the program by searching the SAP table for an Entry Flag (set by the program). It will then reset the proper interval timer with the value contained in the time field value of the SAP. Execution will then proceed as described for non-periodic entries.

- Non-Periodic Entry- Control segments which shall pass control to the the Formulator may also enter a task-request in its execution queue. This will cause the creation of a SAP (e.g., initialization of a Procedure Program) or the duplication of a SAP (to increase the repetition rate of

execution of that control segment). If the conditions exists whereby no additions to the SAP Table can be made, the requesting program will be so informed. Upon completion of these operations control shall be passed to the Sequencer.

b) The Sequencer subprogram shall process inputs as follows:

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

- poll the table unit until it finds a SAP with its Task Status Flag set;

- save the cycle status (point of resumption);

- set the proper interval timer with the maximum time allowed for task execution;

- Dispatch control to the program referenced by the selected SAP. The dispatching interface with other 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.

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

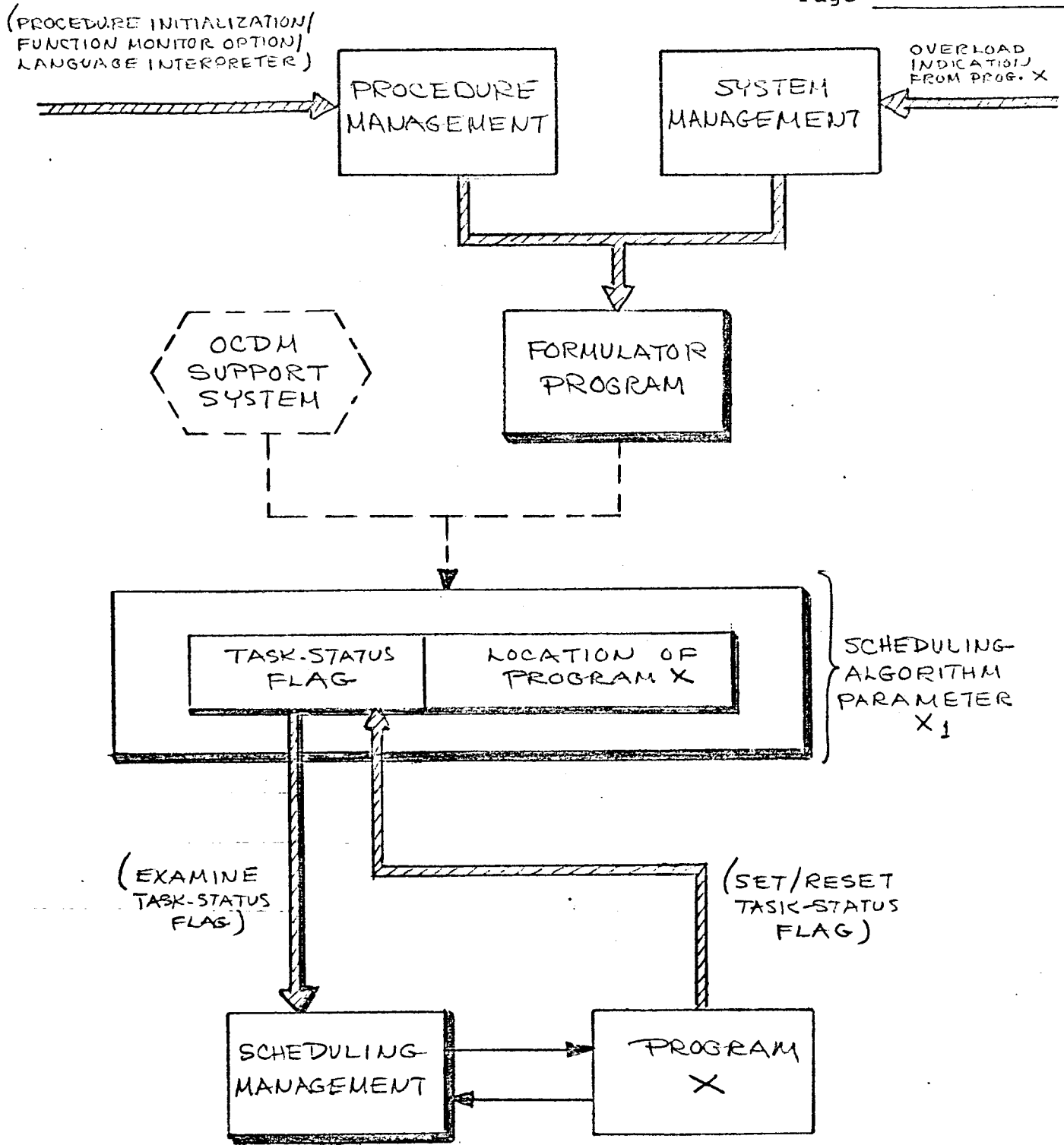


EXHIBIT 6b

SCHEDULING MANAGEMENT

3.1.2.6 Function 6: Interruption Management

The interruption management function shall provide the CPCEI capability to recognize the occurrence at, and identify the action to be taken in regard to hardware/software interruption signals. Interrupts 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 interrupts will also indicate that data from a low-rate input source is available, that a buffered I/O operation has been completed, that a pre-selected 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 typically be obtained from information sources that shall include the following:

- 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

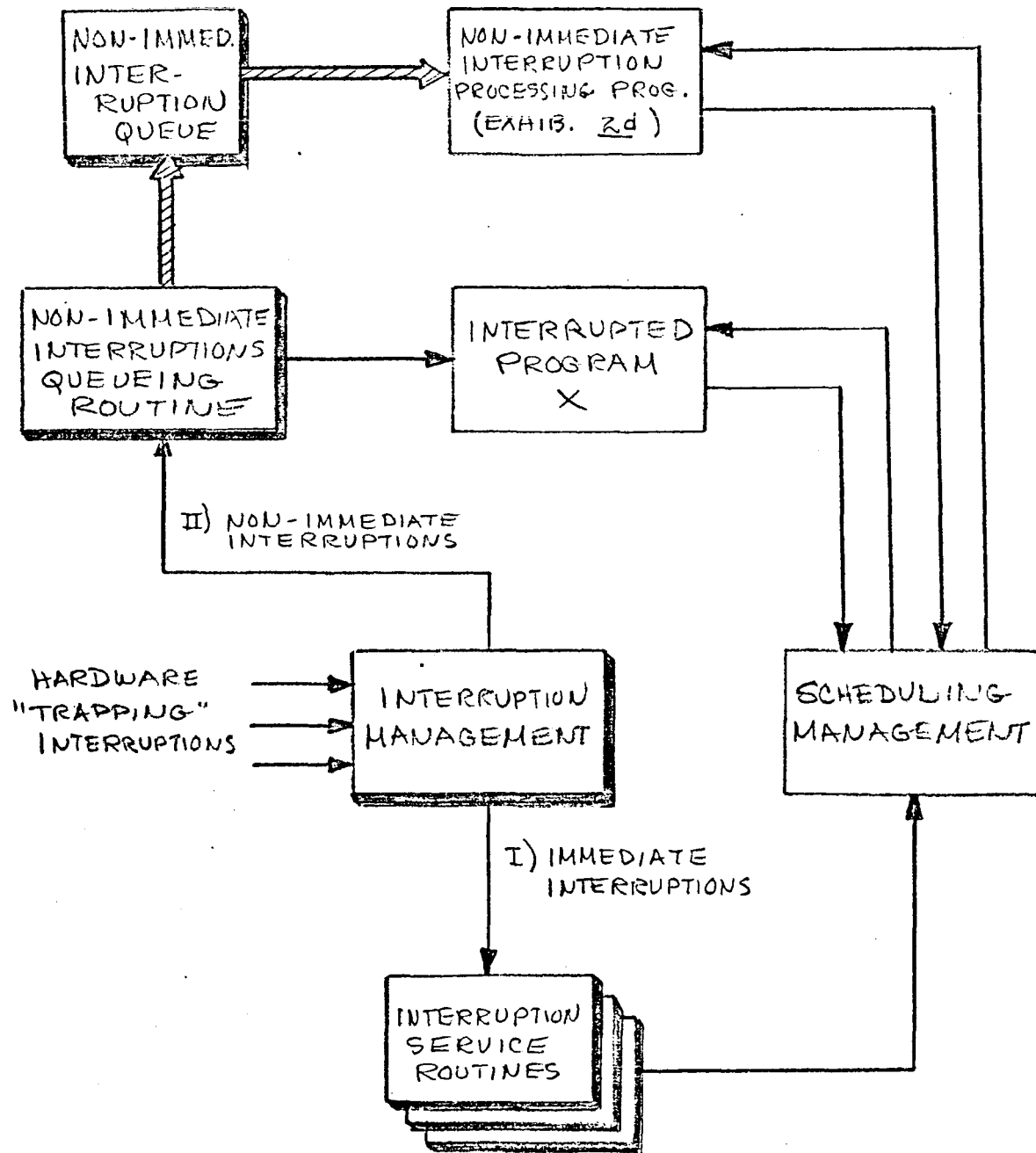


EXHIBIT 7

INTERRUPTION PROCESSING

or more of the following types of interruptions.

(1) Input/output - fault or status conditions of I/O devices on channels.

(2) Program-overflow, underflow, etc.

(3) Supervisor Call- instruction(s) used to transfer control to 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) Units of Measure - TO be determined

c) Limits/Ranges - TO be determined

d) Accuracy/Precision - TO be determined

e) Arrival Frequency - TO be determined

3.1.2.6.2 Destination and Types of Outputs for Interruption Management

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

- Interrupts enabled
- Interrupts allowed
- Interrupts disabled
- Interrupts not allowed
- Interruptions completely serviced
- Interruptions partially serviced

- b) Destinations - TO be determined
- c) Units of Measure- TO be determined
- e) Accuracy/Precision - TO be determined
- f) Output Frequency - TO be determined

3.1.2.6.3 Information Processing for Interrupt Processing

a) Exhibit 8 diagrammatically 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

- SAVE program counter and registers, as required.
- ISOLATE Interrupt type.
- TRANSFER to Interrupt Routine for Action Preference or priority.
- SERVICE Interrupt request.
- RETURN Interrupt State to READY.
- RESTORE program counter, etc.

(2) Partial Service

- SAVE program counter and registers, as required.
- ISOLATE interrupt type.

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 (BAL)	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 interrupt 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

- TRANSFER to interrupt routine for action preference or priority.
- Enter interrupt type in a queue list.
- RETURN interrupt state to WAITING.
- RESTORE program counter, etc.

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

3.1.2.7 Function 7: Console Communication

The console communication function shall be implemented to provide proper operation of OCDMS controls and displays. System operators must be able to effectively change equipment performance characteristics and 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 E1 display, respectively.

3.1.2.7.1 Sources and Types of Inputs for Console Communication

a) Functional inputs shall typically be obtained from information sources that shall include the following:

- Keyboard entry commands for procedure control, and test-point monitor requests;
- Keyboard annotations of manual experiment results, system status, and message-to-tag particular data, for later analysis;

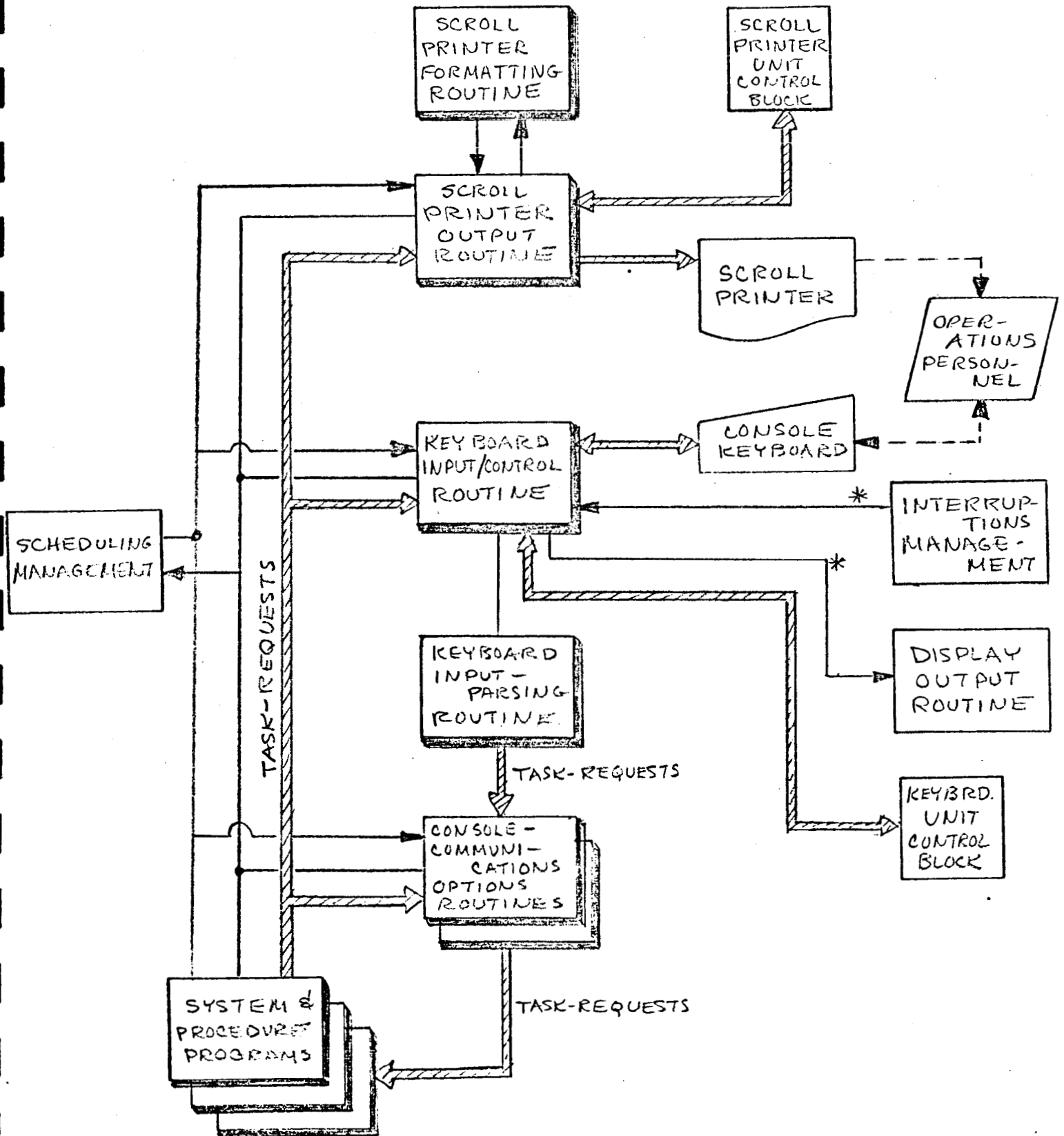


EXHIBIT 9a

CONSOLE COMMUNICATION (PRINTER & KEYBOARD)

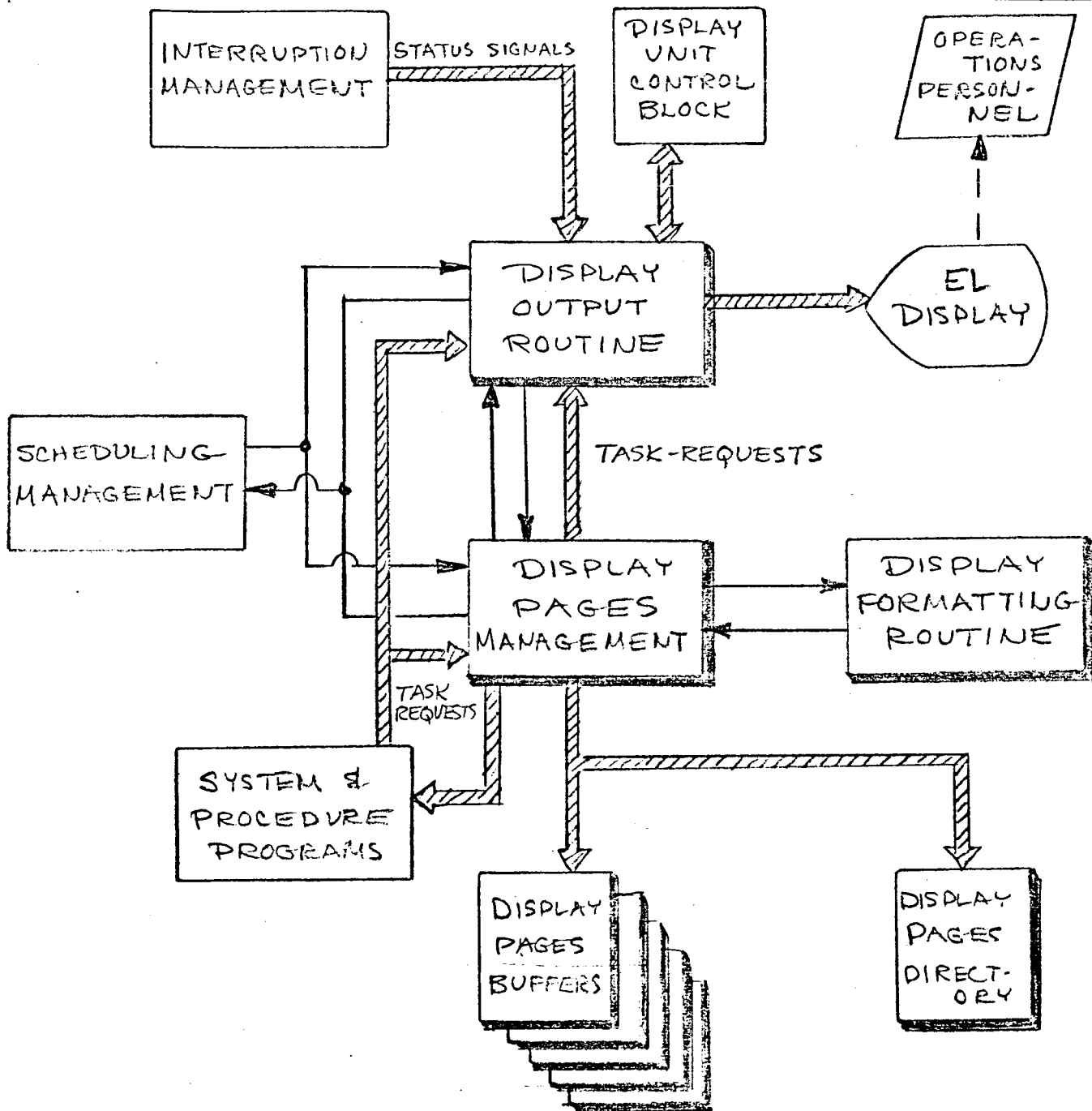


EXHIBIT 9b

CONSOLE COMMUNICATION (EL DISPLAY)

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

- Query and response operations between the OCDMS console operator and Supervisory System.

- b) Units of Measure - TO be determined

- c) Limits/Ranges - TO be determined

- d) Accuracy/Precision - TO be determined

- e) Arrival Frequency - TO be determined

3.1.2.7.2 Destination and Types of Outputs for Console Communication

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

- Displays that compensate for man's limited capability of information storage and retrieval;

- Displays that provide status of concurrent processes and high order integration functions;

- Data conversions, changing machine symbolic codes to/from information forms meaningful to OCDMS operators;

- Transmission feedback information;

- Provisions for error-free console communications and/or failure compensation.

- 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

3.1.2.7.3 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 line-printer, keyboard, display, and console options subprograms.

a) Line-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 I/O routines. It shall access the appropriate Unit Control Block entries for control information concerning the line 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 non-interrupt basis.

c) Display subprogram - The display subprogram shall perform information processing functions involved with the set of 144 alphanumeric display characters of the electro-luminescent (EL) type. These EL elements will be arranged in a display matrix and utilized for Experiment Procedure programs, system programs, operator specified procedures, and scratch-pad activities. In order to manage CDU demands associated with concurrent activities, the display sub-program will utilize a system of display pages. Pages will consist of main storage segments (buffers) managed by the display sub-program. 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, getting additional buffers from pool storage, and initiating I/O requests. In order to implement the paging technique, the subprogram will maintain an index of the locations and contents 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:

- Each procedure program requiring display output;
- The system management subprogram (system status --processing, I/O, memory load; etc.);
- operator constructed procedures (on-line);
- Operator scratch pad (for use during scientific calculations, construction of procedures, etc.);
- Operations status (The system will continuously provide the OCDMS operator with information describing procedure execution status).

d) Console Options Subprogram - The console options subprogram will implement information processing to the following extent:

- A routine which initiates the transfer of display pages to bulk storage;
- A routine which generates requests for display pages to be printed on the line printer;
- A routine which enables the OCDMS to be used for scientific calculations and for other data management functions which require operator participation;
- A routine which employs the EL display matrix as a scratch pad;
- A routine which allows the OCDMS operator to enter limited annotations to a displayed page;
- 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 (UCBs) and Procedure Block Files (PBFs) shall be used and updated with current status/configuration information as system tasks are performed.

Exhibit 10 illustrates the operations involved.

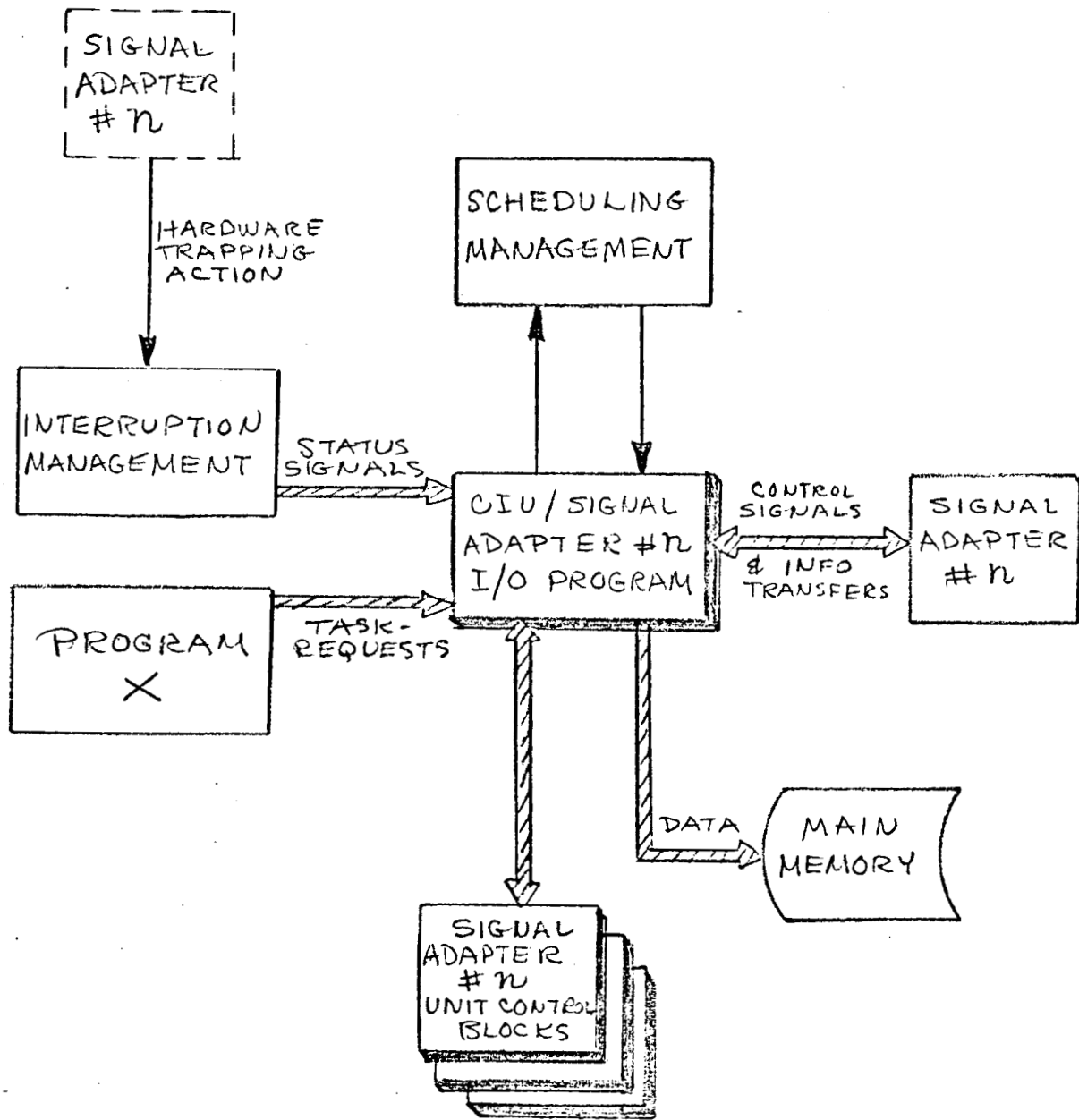


EXHIBIT 10

CIU/SIGNAL ADAPTER COMMUNICATION

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

a) Functional inputs shall typically include the following:

- Locations of measurement;
- Subsystem abbreviations;
- Channel numbers;
- Quantity of similar signals;
- Usage (factory/prelaunch/mission);
- And device names;
- Confidence factors;
- Analog signal descriptions
(range, units, accuracy, frequency);
- Discrete signal descriptions
(logic levels, units, time);
- Coded signal descriptions
(bits/bytes/words, conversion codes);
- Stimulus descriptions (prerequisites for
turn-on, associated measurement, ramp/step,
harmonic distortions, duty cycle);
- Measurement descriptions (momentary/
continuous, response time, stimuli required,
display required).

- b) Unit of Measure - TO be determined
- c) Limits/Range - To be determined
- d) Accuracy/Precision - TO be determined
- e) Arrival Frequency - TO be determined

3.1.2.8.2 Source and Type of Outputs for CIU/Signal Adapter Communication

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

- Discrete driving function;
- Analog driving function;
- Coded driving function;
- Single response measurement;
- Sample pattern measurement;
- UCB status entries
- Requesting subroutine communication.

- b) Destination - 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

3.1.2.8.3 Information Processing for CIU/Signal Adapter Communication

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:

- Accepting, queuing, and responding to task requests;
- Accessing the necessary fields in a selected Unit Control Block;
- Transmitting and receiving control data and other status information via I/O channel routines;
- 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 application 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 pre-defined 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 typically be obtained from data sources that shall include the following:

- Apollo DCS real-time commands;
- On/Off data formats
- Digital/analog data formats;
- Timing and event codes;
- System addresses;
- Redundancy words;
- Error protection codes;

- b) Units of Measure - TO be determined
- c) Limits/Ranges - TO be determined
- d) Accuracy/Precision - TO be determined
- e) Arrival Frequency- TO be determined

3.1.2.9.2 Destination and Types of Outputs for Uplink Communication

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

- Symbolic code to Interpreter;
- Command/control requests;
- Verification of messages;
- Message posting;
- Interrupt actions;
- Error detection and recovery;
- OCMDS mode selection requests.

- 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

3.1.2.9.3 Information Processing for Uplink Communication

a) 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.

b) 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 functions of the Supervisory System shall provide for OCDMS handling of DCS transmission check procedures, and 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 work loads 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. The information sources shall include, but not necessarily be limited to, the following:

- Experiment/checkout measurement values;
- Measurement conditions;
- Experiment annotations;
- Event/activity time records;

- Equipment items usage (time or cycle sensitive components);
 - Data segment headers and labels;
 - OCDMS status and system statistics;
 - Uplink transmission-checks;
 - Testing tolerances in use;
 - Change action records;
 - Ground system site-peculiar parameters.
- b) Units of Measure -TO be determined
- c) Limits/Ranges - TO be determined
- d) Accuracy/Precision - TO be determined
- e) Output Frequency - TO be determined

3.1.2.10.2 Sources and Types of Outputs for Downlink Communication

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

- Apollo PCM T/M formats;
- Backlogged PCM prime-frame buffers;
- Data compression outputs;
- Selective monitoring orders;
- Adaptive sampling orders;
- Direct data coding instructions;
- Downlink transmission check;
- Burst -mode PCM transmission.

- b) Destination - 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

3.1.2.10.3 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 main OCDMS data generation and collection exceed 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 types of data. Data compression methods that shall be available include:

(1) Parameter Extractions

- probability distribution;
- power spectrums;

- state parameters;
 - fourier coefficients;
 - other irreversible information-
describing transformations.
- (2) Direct Redundancy Reduction
- Peak error predictors;
 - RMS error predictors;
 - peak error interpolators;
 - RMS error interpolators.
- (3) Encoding
- Adaptive coefficients;
 - non-adaptive increments;
 - bit plane.
- (4) Adaptive Sampling
- Variable rate;
 - fixed rate;
 - 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 adaption parameters for factory, launch, and remote mission sites shall constitute data sets, reserved

in system bulk storage, which upon 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, 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, and software development shall be in accord, as applicable.

3.2 CPCEI Definition

The functional relationship of the CPCEI to other equipment, 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 OCDM Support System CPCEI

The primary functional interface between the supervisory and support system software shall be identified in terms of the experiment procedure programs and on-board system data base (experiment/OCDMS parameters). 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 which are incorporated together with the OCDM Supervisory System on the bulk storage medium in a format and order suitable for OCDMS mission operations. Prior to CPCEI qualification, the Supervisory System on the bulk storage medium in a format and order suitable for OCDMS mission operations. Prior to CPCEI qualification, the Supervisory System design and development testing shall

	COMPUTER & OPERATIONS					
	SUPPORT SYSTEM (3.2.1.2.1)	EXPERIMENT PROCEDURES (3.2.1.2.2)	PERIPHERAL EQUIPMENT (3.2.1.2.3)	EXPERIMENT HARDWARE (3.2.1.2.4)	GROUND SITES (3.2.1.2.5)	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
CONSOLE COMMUNICATION (3.1.2.7)		X		X	X	X
CIU/SIGNAL ADAPTER COMMUNICATION (3.1.2.8)	X	X		X		
UPLINK COMMUNICATION (3.1.2.9)	X				X	
DOWNLINK COMMUNICATION (3.1.2.10)	X				X	

EXHIBIT 11 - OCDMS SUPERVISORY SYSTEM INTERFACE

utilize the test services provided by the Support System Program Test and Verification functions.

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 provides the capability of operations personnel control and/or monitor all facets of OCDMS operations. This includes direction and/or participation in activities such as:

- overall system start-up/shutdown;

- procedure execution control and modification;
- on-line procedure generation;
- scientific calculation and evaluation;
- on-line data retrieval and analysis;
- 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.

3.1.2.6 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.

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, 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 satisfies 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 test 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-Implementation Design 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. Tests of this type shall, when appropriate, be continued throughout the design process.

b) Subprogram Checkout- which are 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 indigenous and exogenous failure mechanisms prior to combining the subprograms into main programs or other functional program aggregates.

c) Main Program Checkout- which are 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) CPECI 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 upon the availability of a system environment simulator and associated test support tools. Therefore, these test may be deferred until overall OCDMS qualification testing.

(2) Simulated environment testing objectives are the following:

- To obtain a more controlled test of the CPCEI than could be accomplished in the total OCDMS;
- To determine the safety of the CPCEI in the OCDMS without exposing the system to unnecessary hazard;
- To serve as the basis for preliminary qualification to the CPCEI prior to transferring it from a development facility to a using facility;
- 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

- Program Initialization
- Program Termination
- I/O Resource Allocation
- Power cycle restart provisions
- Error detection and recovery
- Status data collection and reporting
- Accounting and event trial facilities

b) Memory Mangement

- Main storage allocation
- Main storage retrieval
- Bulk storage READ/WRITE
- Program and data set relocation

c) Language Interpreter

- Functional expression decomposition
- Calling sequence and linkage generation
- Command/control action routines
- List processing
- Parameter changing

d) Procedure Mangement

- Procedure Block File processing
- Task Control Block processing
- Mode selection (Automatic, Semi-Automatic, Manual)
- "Predicted" and "Actual" comparison actions
- Manual operation support

e) Schedule Management

- Sequence assignment tasks
- Priority sequencing
- Normal sequencing
- Periodic sequencing
- Dispatching
- I/O channel scheduling

f) Interruption Management

- Priority Interruption
- Program and Supervisor-CALL Interruption
- I/O, External, and Machine-Check Interruptions
- Completed Interruption Services
- Partial Interruption Services

g) Console Communication

- Procedure Set-Up Commands
- Procedure Control Commands
- Function Monitor Commands
- Procedure Annotations
- Display Commands

h) CIU/Signal Adapter Communication

- Discrete Driving Functions
- Analog Driving Functions
- Coded Driving Functions
- Single-Response Measurements
- Sampled Measurements

i) Uplink Communication

- Real-Time Commands
- Error Protection Encoding
- Symbolic code for the Interpreter
- Input Command/Control Actions

j) Downlink Communication

- Prime Frame Generator
- Buffering and Data Transfer
- Data Compression

4.1.2 Preliminary Qualification Test

Preliminary qualification tests shall verify each requirement of Section 3 which can be tested in a simulated environment. If appropriate, they are to be held at the contractor's development facility and will serve as the basis for transfer of the CPCEI to the NASA Facility.

4.1.2.1 Qualification Test Requirements

To be determined

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 CPCEIs.

4.2.1 General

The OCDMS Support System CPCEI tests which are required in direct support of system integration are:

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

4.1.2.6 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 (CPCs) 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

None