Report No. 75-0044
Contract No. NAS8-26990

HAL/SM SYSTEM FUNCTIONAL

DESIGN SPECIFICATION

(NASA-CR-144098) HAL/SM SYSTEM FUNCTIONAL N76-14843
DESIGN SPECIFICATION (M&S Computing, Inc.)
72 p HC $4.50 CSCL 09B
Unclas
G3/61 06779

December 1, 1975

Prepared for:

George C. Marshall Space Flight Center
NASA
Marshall Space FlightCenter, Alabama 35812

M&S COMPUTING, INC.

PREFACE

This document presents the functional design of the HAL/SM programming system. The HAL/SM programming language is defined in the "HAL/SM Language Specification," and the requirements for the programming system to be used as a guideline for this design are described in the "HAL/SM System Software Requirements Specification."

Prepared by:

C. Ross
G. P. W. Williams, Jr.

Project Manager:

J. Pruitt

Approved by:

T. T. Schansman

TABLE OF CONTENTS

(BLANK)

LIST OF FIGURES

PRECEDING PAGE BLANK NOT FILMED

LIST OF ACRONYMS

CVT	Concept Verification Test
ERP	External Reference Point
I/O	Input/Output
MOSS	Modular Operating System for SUMC
PAF	Program Access File
RTL	Run Time Library
SUMC-S	Space Ultrareliable Modular Computer - Simplex
SVC	Supervisor Call

1. INTRODUCTION

This document describes the functional design of the HAL/SM programming system (a version of the HAL programming language specifically adapted for use in the Concept Verification Test (CVT) environment on the Space Ultrareliable Modular Computer Simplex (SUMC-S) under the Modular Operating System for the SUMC (MOSS)). The HAL/SM programming language is defined in the HAL/SM Language Specification (Reference 1), and the HAL/SM system implementation requirements are defined in the HAL/SM System Software Requirements Specification (Reference 2); familiarity with both documents shall be assumed herein.

As described in Reference 2, the major subsystems of the HAL/SM system and their functions shall be as follows:

- HAL/SM Preprocessor - convert HAL/SM source language modules into HAL/S source language modules, perform minimal syntax verification, and provide various support features which cannot be conveniently or adequately performed by the HAL/S-360 Compiler.

- HAL/S-360 Compiler - convert HAL/S source language modules into IBM S/360 compatible object modules and perform complete syntax verification.

- HALLINK - combine HAL/S object modules into load modules, calculate run time stack size requirements, and add the stack to the load module and add and/or delete certain other CSECTS from the generated load module (depending on options specified).

- Run Time Library (RTL) - when properly linked with each HAL/SM task, provide computational routines and MOSS interface routines to support various features of the HAL/SM language.

The functional design of the preprocessor, HALLINK, and RTL subsystems is described in the following sections through the use of structure charts, data flows, and intermodule interface definitions.

Structure charts and data flows are defined and illustrated in the HAL/SM System Design and Implementation Plan (Reference 3). This reference may be used as a guide for interpretation of the structure charts and data flows which follow. A structure chart and a data flow diagram are included for each subsystem. Also a group of intermodule interface definitions (one definition per module) is included immediately following the structure chart and data flow for a particular subsystem. Each of these intermodule interface definitions consists of the identification of the module, the function the module is to perform, the identification and definition of parameter interfaces to the module, and any design notes associated with the module.

2. PREPROCESSOR

The HAL/SM Preprocessor is represented by the data flow and structure chart shown in Figure 2-1 and 2-2. These figures give an overview of the preprocessor. More detailed descriptions of the input system, output system, and semantic routines are given in Sections 2.1, 2.2, and 2.3, respectively. The remaining modules are described in this section.

Module: XREF Processor

Function: Receives and alphabetizes the HAL/SM identifiers to produce the cross-reference listing included as part of the preprocessor output listings. Each usage of an identifier is referenced in the listing.

Parameters: HAL/SM identifier. Referencing statement number.

Module: PAF Processor

Function: Processes all variables declared with the access attribute against the Program Access File (PAF) to determine the legality of access to these variables.

Parameters: Variable identifier with access attribute.

Note: This module reads the PAF data set.

Module: Directive Processor

Function: Processes both HAL/SM and HAL/S directives. HAL/S directives are passed through to the output system for inclusion in the HAL/S source. The HAL/SM directive selects the processing options for the preprocessor.

Parameters: HAL/SM directives. HAL/S directives.

Module: DDIU Processor

Function: Creates preformatted display messages for the C&D consoles. Coded references to the messages are placed in output HAL/S code where each message is referenced. The display messages are stored in a standard data set accessed by the RTL during program execution.

Parameters: Display statement symbols.

Note: The DDIU module consists of several subroutines; one for each different display statement symbol.

HAL/SM PREPROCESSOR DATA FLOW

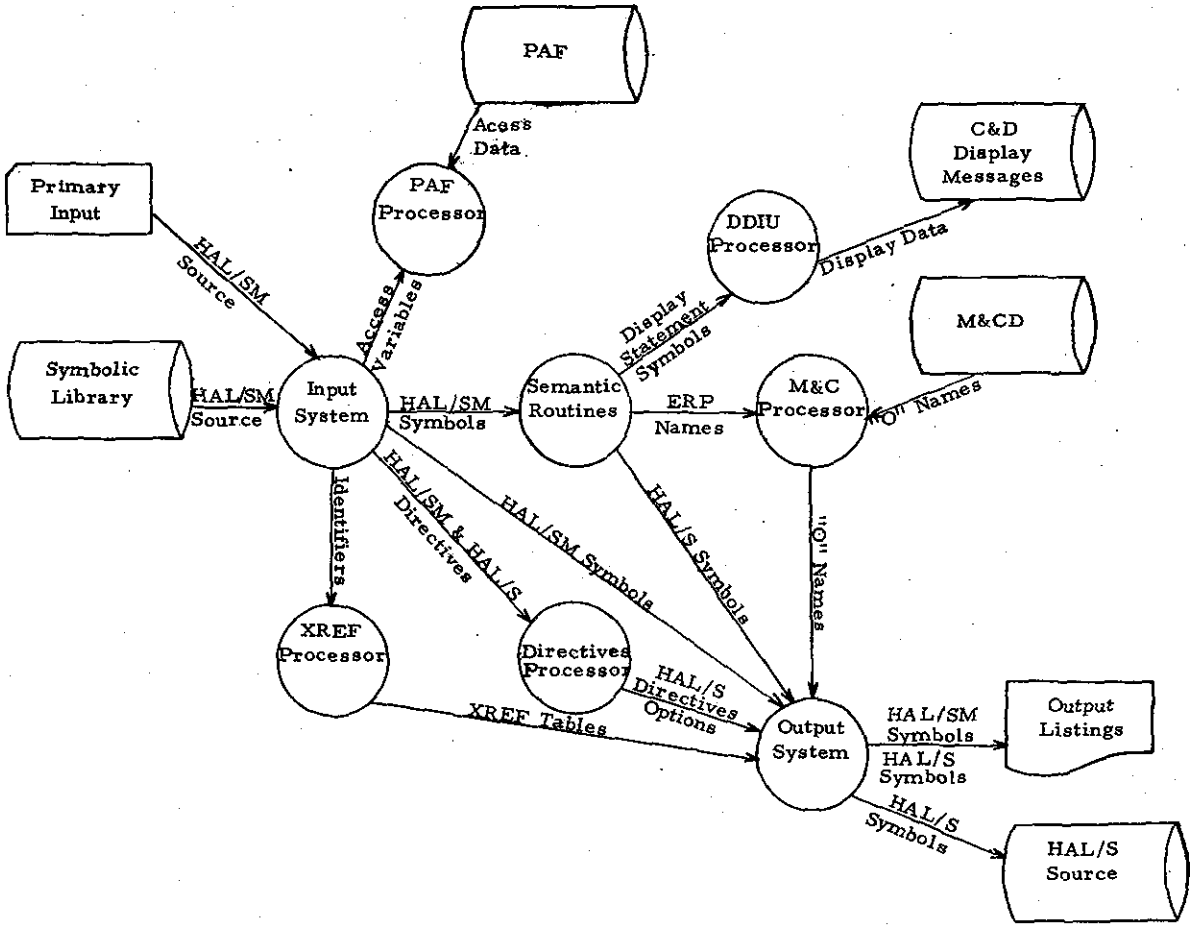
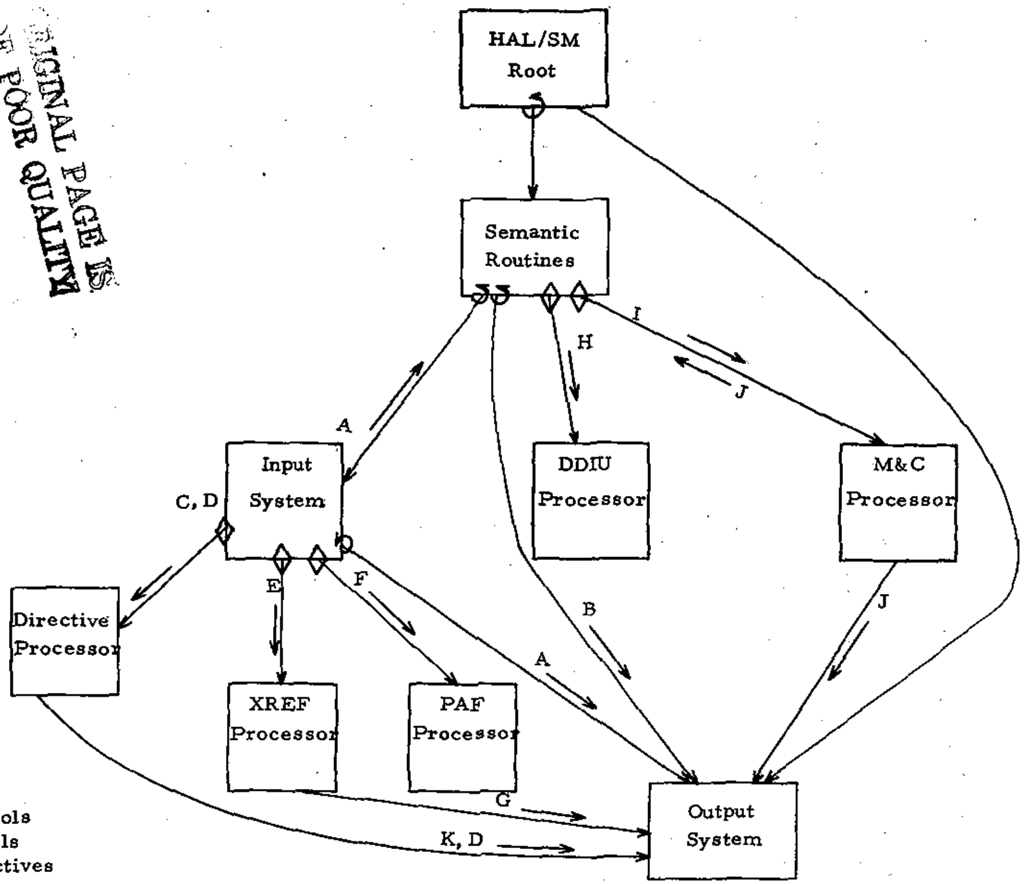


Figure 2-1

HAL/SM PREPROCESSOR STRUCTURE CHART



A.	HAL/SM Symbols
B.	HAL/S Symbols
C.	HAL/SM Directives
D.	HAL/S Directives
E.	Identifiers
F.	Access Variables
G.	XREF Table
H.	Display Statement Symbols
I.	ERP Names
J.	"O" Names
K.	Output Options

Figure 2-2

Module: M&C Processor

Function: Converts each reference to 31-character External Reference Point (ERP) names to 9-character "O-names" in the output HAL/S code. The M&C Definition File is used to verify all ERP references and their correct usage.

Parameters: 31-character ERP name from HAL/SM source statements.

2.1 Input System

The input system data flow and structure chart are shown in Figures 2-3 and 2-4. The input system reads the primary and symbolic source files and prepares the symbols for processing by the semantic routines. The input system interfaces directly with the output system to produce the HAL/SM formatted listings. The modules of the input system are given below.

Module: Get Statement

Function: Provides characters from the current statement. When the current statement is entirely processed, the next statement is read from the primary input file or the symbolic library. Incoming MACRO definitions are passed to the MACRO processor. Incoming directives are passed on to the directives processor.

Parameters: Character buffer to return the next character.

Module: MACRO Processor

Function: Processes MACRO definitions and MACRO expansion requests. MACRO definitions are placed in the MACRO table. MACRO expansions are handled through the MACRO table. The parameters to the MACRO are expanded according to the model definition.

Parameters: MACRO Definition, MACRO Name, or MACRO Parameters for expansion.

Note: The MACRO Processor will be divided into two modules for detailed design - a MACRO definition processor and a MACRO expansion processor.

Module: Symbol Processor

Function: Builds HAL/SM symbols (the smallest logical parts of a source

HAL/SM INPUT SYSTEM DATA FLOW

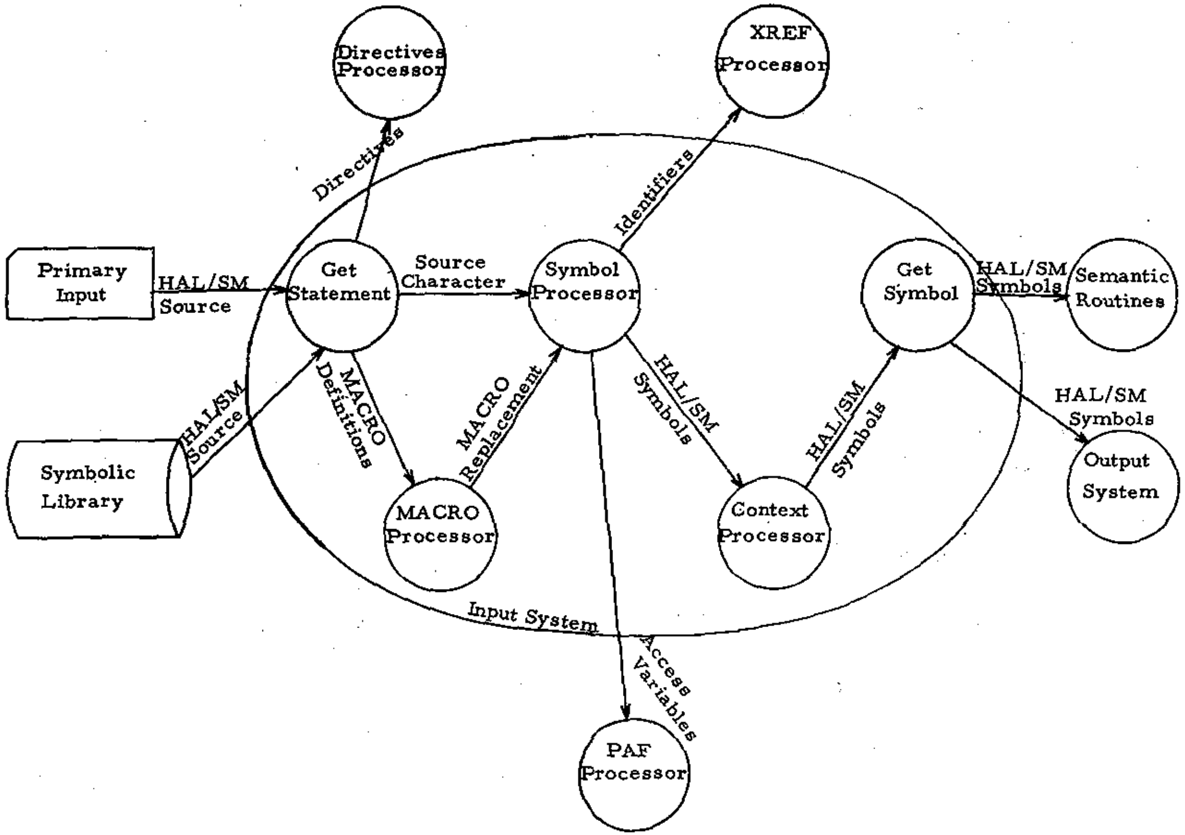
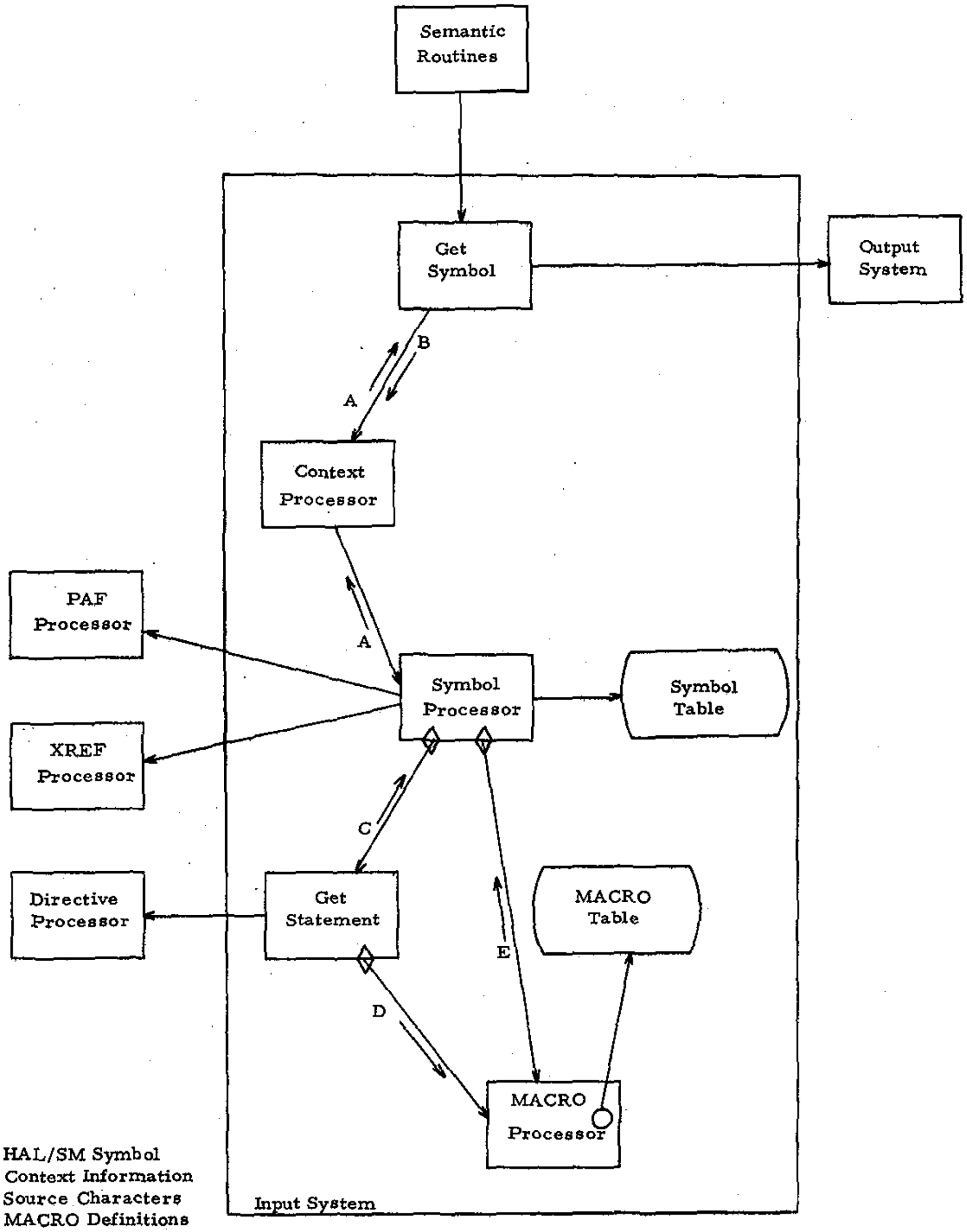


Figure 2-3

HAL/SM INPUT SYSTEM STRUCTURE CHART



A. HAL/SM Symbol
B. Context Information
C. Source Characters
D. MACRO Definitions
E. MACRO Replacements

Figure 2-4

statement) from source statement characters. The symbols and their associated context are maintained in the symbol table.

Parameters: Symbol locator giving the position relative to the current symbol. Symbol return buffer.

Note: 1. Symbols which are identifiers are passed to the XREF Processor.
2. Symbols which are access variable identifiers are passed to the PAF Processor.

Module: Context Processor

Function: Maintains the context of the symbols relative to the HAL/SM grammar. Also allows the context to be selected to obtain a symbol from the Symbol Processor.

Parameters: Context Selector, Symbol Locator, Symbol Return Buffer.

Module: Get Symbol

Function: Returns a selected symbol based on context and symbol position.

Parameters: Context Selector, Symbol Locator, Symbol Return Buffer.

2.2 Output System

The output system data flow and structure chart are shown in Figures 2-5 and 2-6. The output system performs the output functions for the preprocessor including HAL/SM listings and HAL/S listing and source file preparation. The HAL/SM directives indicates the listing options to be performed.

The output HAL/S source code is maintained in two temporary files - one for global statements, the other for local statements. The HAL/SM source statement can cause the generation of global statements in the HAL/S source (which must be placed at the beginning of the program). In order to avoid a second pass to produce the ordered HAL/S source, the output system maintains separate files which are merged for final output. The modules of the output system are described below.

Module: Put Symbol

Function: Receives symbols to merge into source statements for output. The symbol may either be HAL/S symbols or HAL/SM symbols.

HAL/SM OUTPUT SYSTEM DATA FLOW

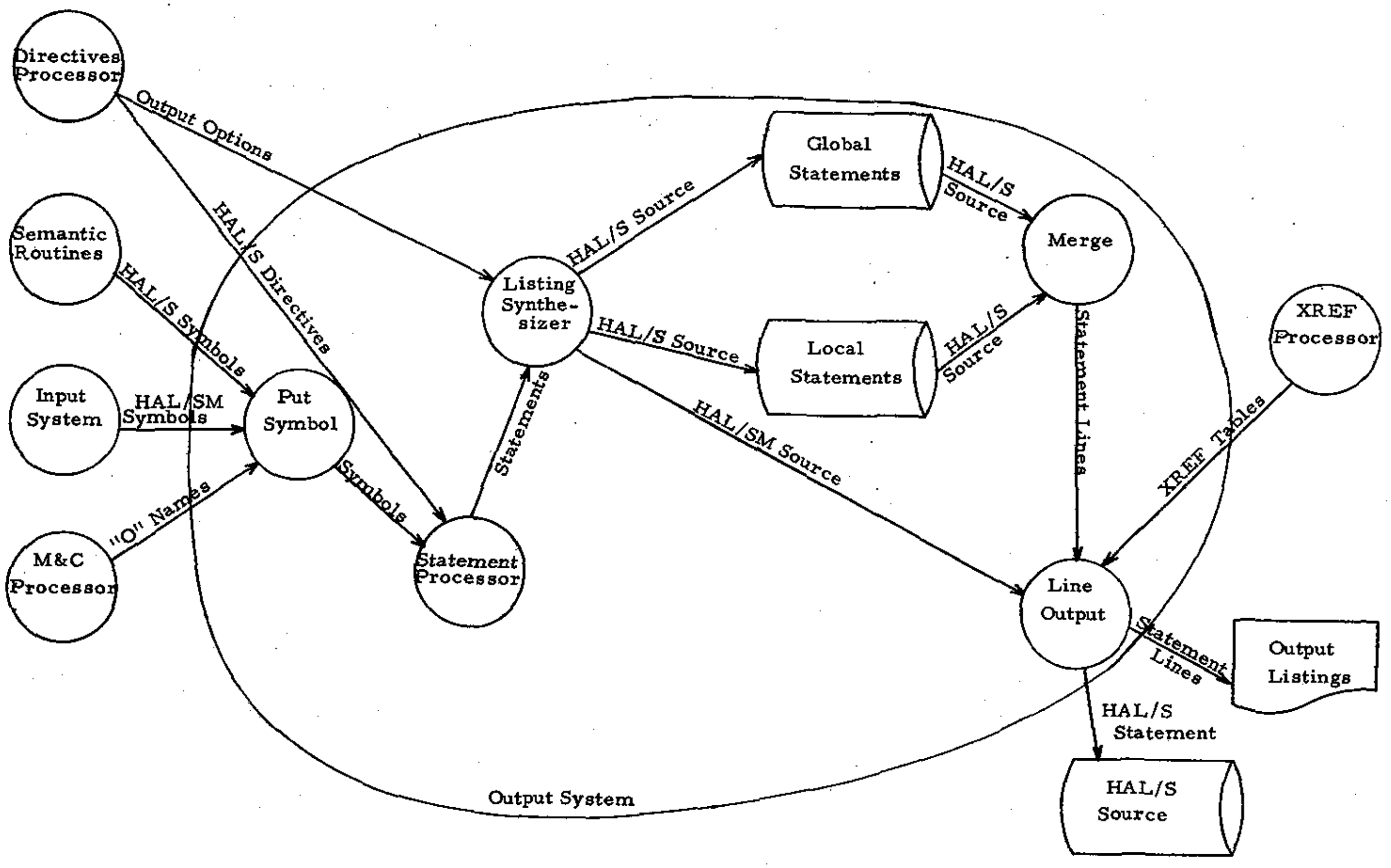
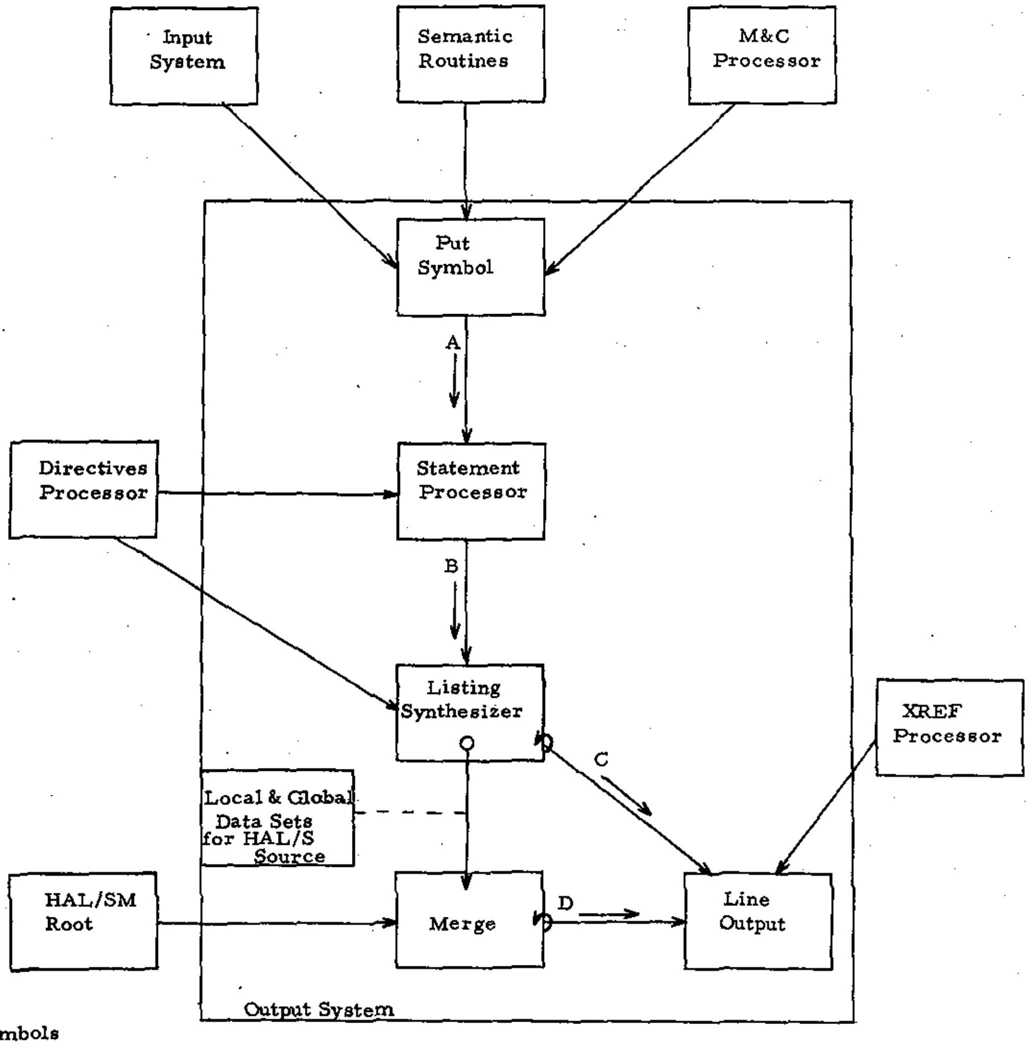


Figure 2-5

HAL/SM OUTPUT SYSTEM STRUCTURE CHART



A. Symbols
B. Statements
C. HAL/SM Source
D. HAL/S Source

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 2-6

Parameters: Symbol to be included in the output.

Module: Statement Processor

Function: Combines the symbols to form source statements for output.

Parameters: Symbol buffer.

Note: Also processes HAL/S directives to be output.

Module: Listing Synthesizer

Function: Builds and formats the listing snd HAL/S source files for output. Maintains the dual output files for HAL/S source which are merged to output at the end of the run. HAL/SM source statements are formatted and sent directly to the line output module to be printed.

Parameters: Statement Buffer.

Note: The output options controlling the processing of this module are provided by the directives processor.

Module: Merge

Function: Combines the local and global HAL/S temporary files to produce the HAL/S source format suitable for the HAL/S compiler.

Parameters: None.

Note: The HAL/SM root module invokes this module at the end of the preprocessor run.

Module: Line Output

Function: Performs the actual output to the printer of HAL/S source file on a line basis.

Parameters: Output statement line to be output.

Note: This module also processes the XREF listing produced for the HAL/SM formatted listing.

2.3 Semantic Routines

The data flow and structure chart for the semantic routines are shown in Figures 2-7 and 2-8. These modules are responsible for the

HAL/SM SEMANTIC ROUTINES DATA FLOW

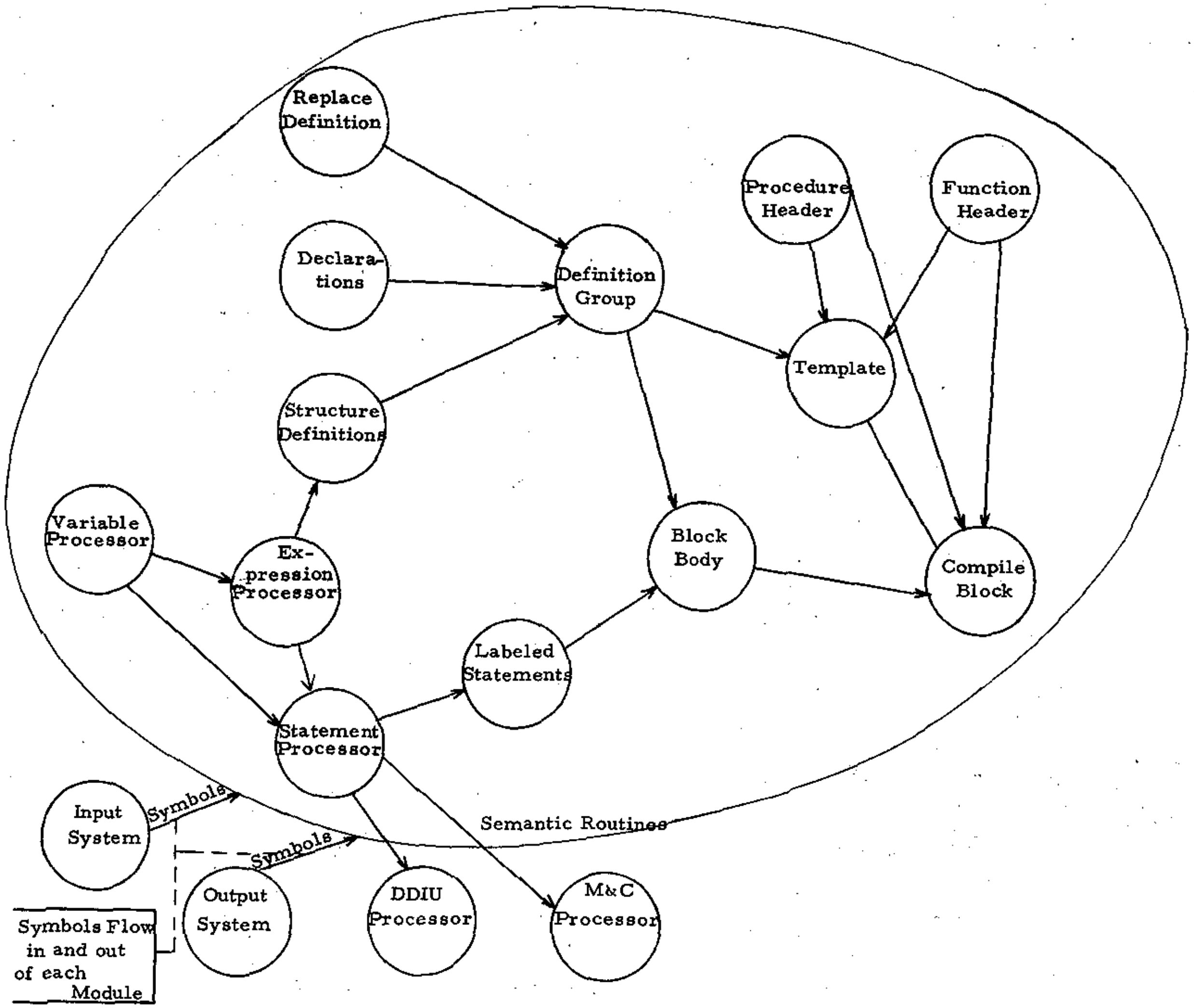


Figure 2-7

HAL/SM SEMANTIC ROUTINES STRUCTURE CHART

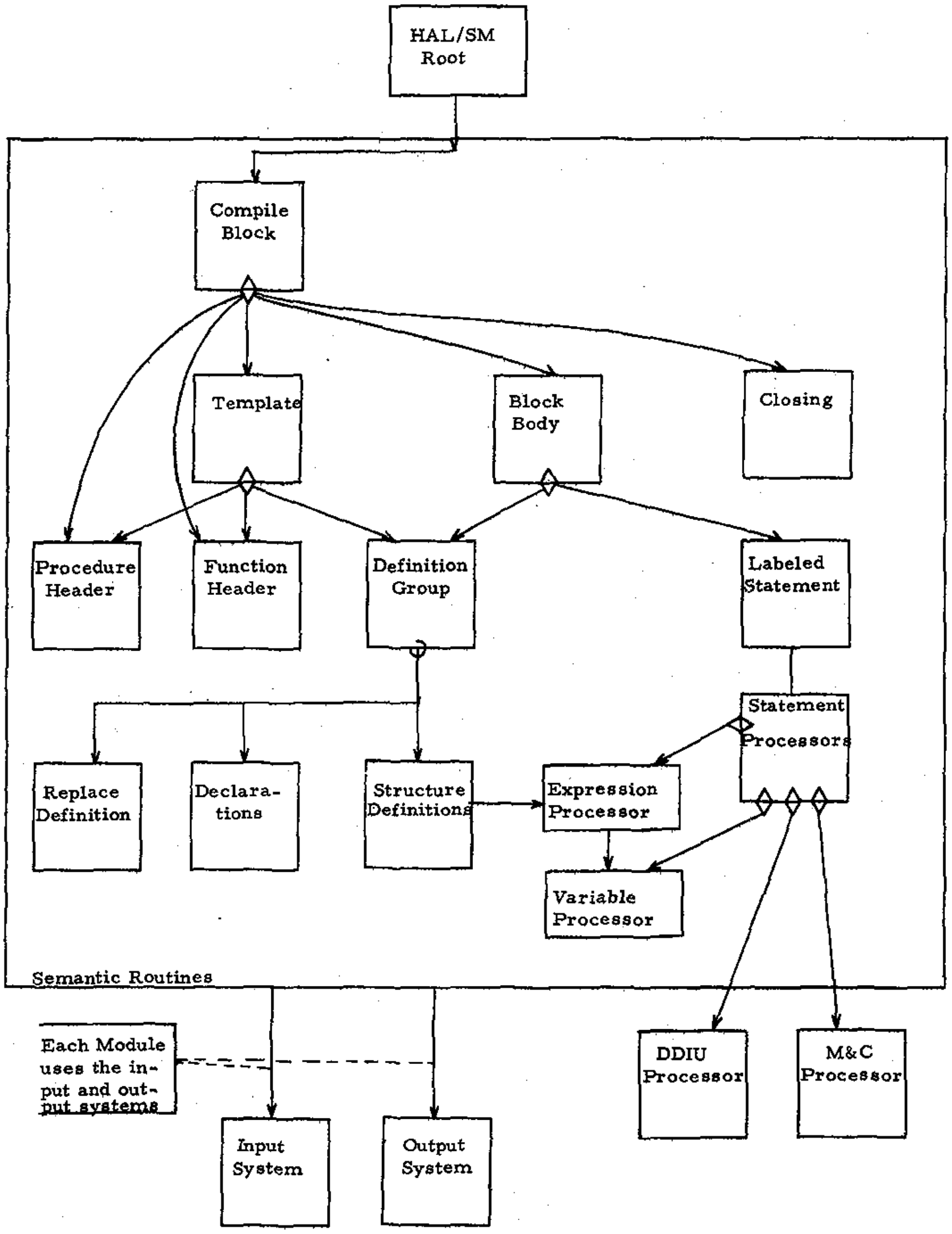


Figure 2-8

semantic processing and output of HAL/S code for the HAL/SM source statements. This processing is carried out according to the HAL/SM grammar in Appendix A. The grammar defines the successive buildup of symbols into the HAL/SM statements and buildup of statements into the compilation as a whole. The figures for the semantic routines show only the top level definitions to illustrate the basic structure.

The semantic routines consist of a module for each HAL/SM symbol in the grammar. The code generation is carried out in the statement processors (one for each HAL/SM statement) according to the templates defined in Reference 2.

Each of the semantic routines is driven by the symbols obtained from the input system, and those generated and passed to the output system. Individual module descriptions will not be provided for the semantic routines.

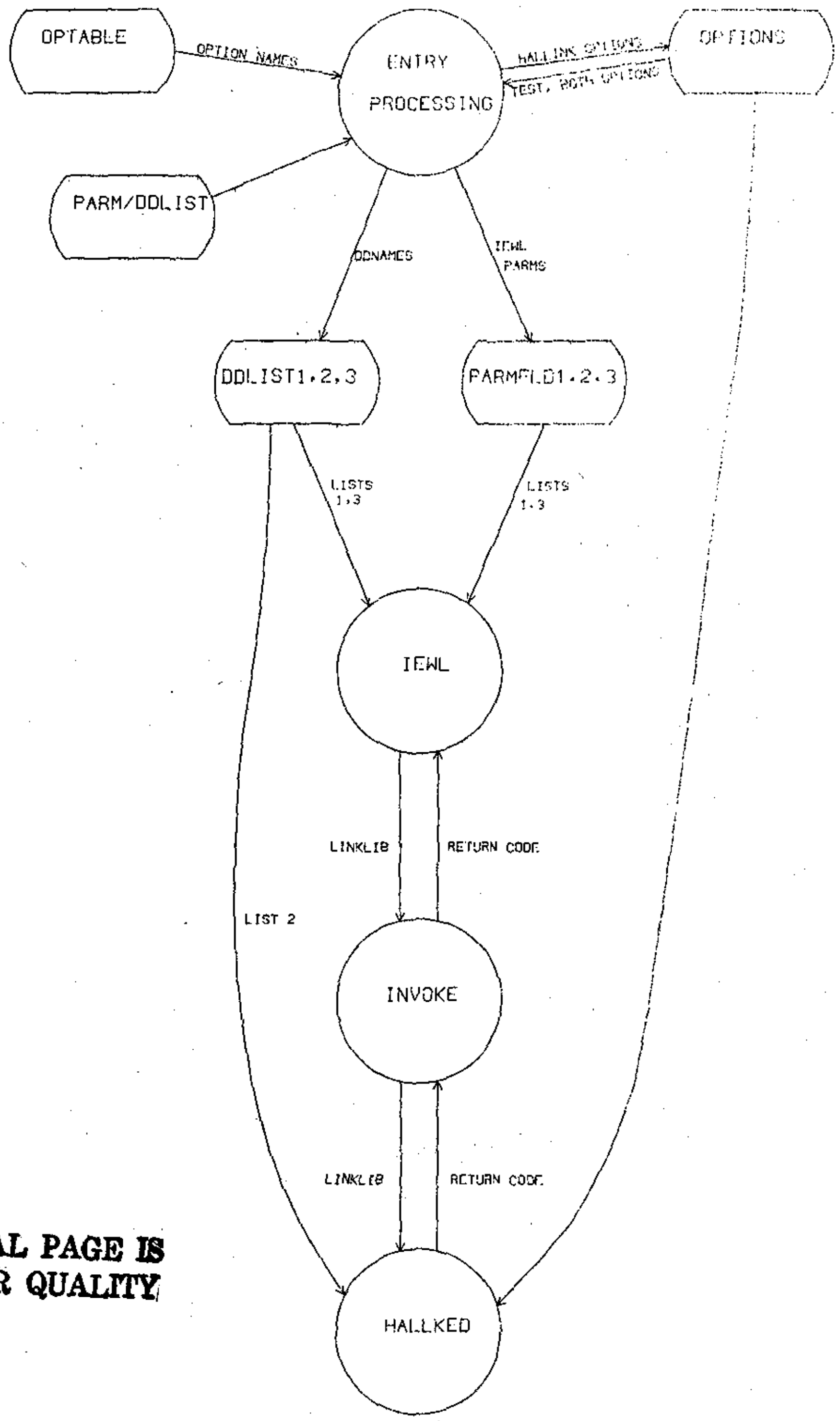
(BLANK)

3. HALLINK

The HALLINK data flow and structure chart are shown in Figures 3-1 and 3-2. The necessary modifications to adapt HALLINK for use with the HAL/SM Preprocessor system are in the HALLKED module. The modified data flow and structure chart for HALLKED are shown in Figures 3-3 and 3-4. The following paragraph describes the changes made.

The HALMAP control section has been suppressed in the Entry Processing and in Phase 4A. Cost-use arrays have been suppressed in ESDLOOP. RC has been modified to limit the number of process control sections per load module to one.

HALLINK DATA FLOW



ORIGINAL PAGE IS OF POOR QUALITY

Figure 3-1

HALLINK STRUCTURE CHART

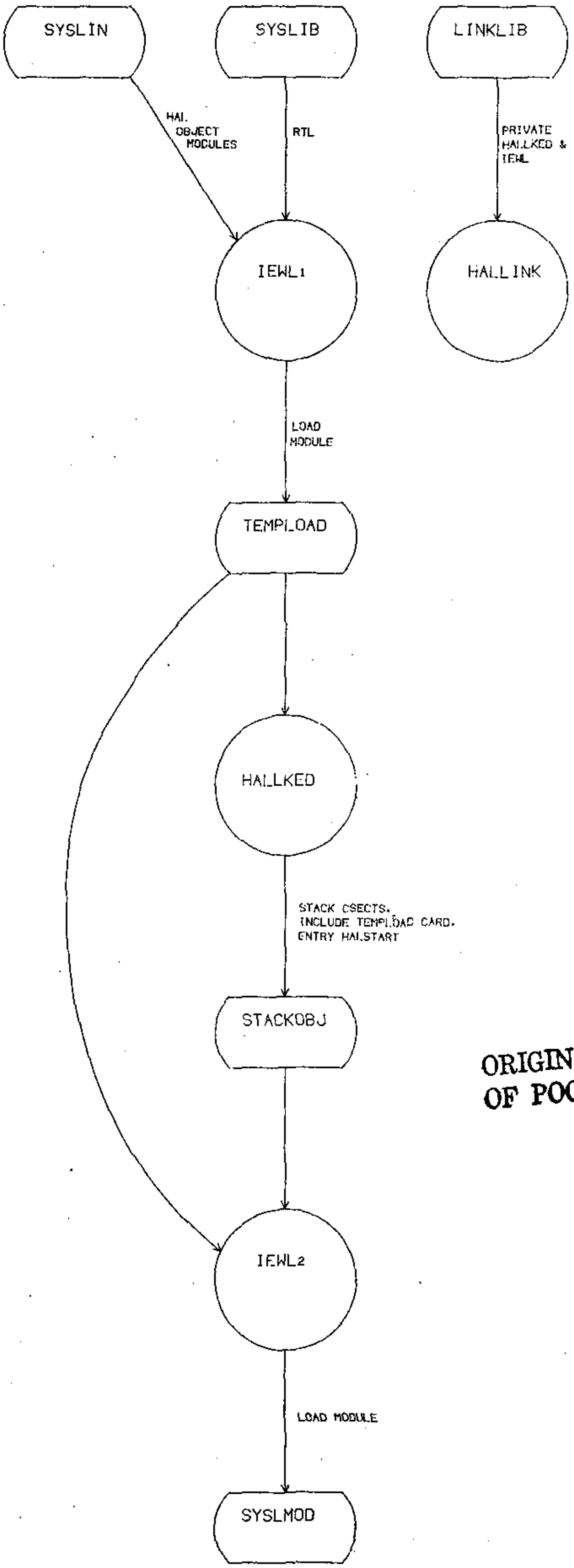


Figure 3-2

HALLKED DATA FLOW

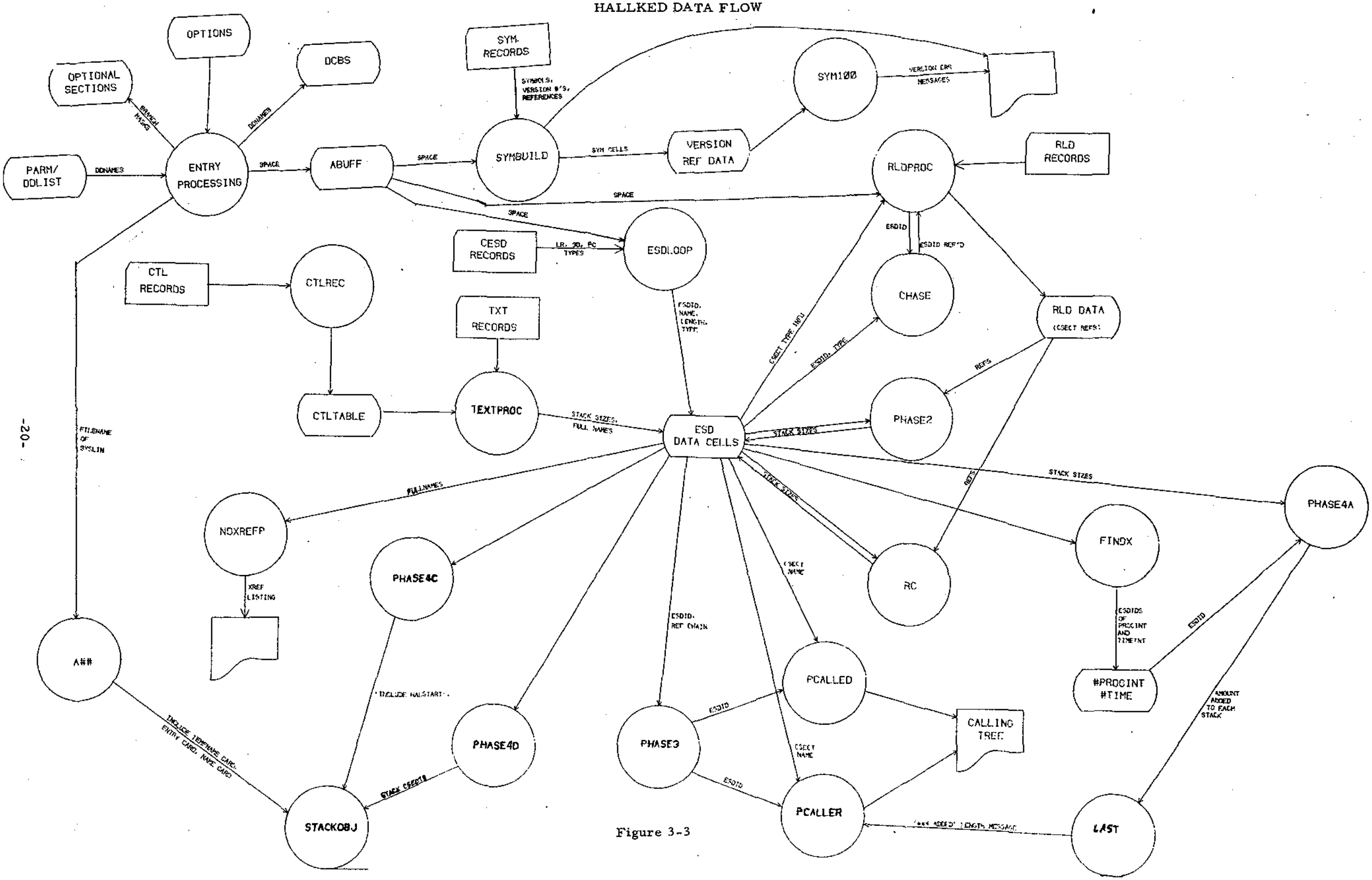


Figure 3-3

HALLKED STRUCTURE CHART

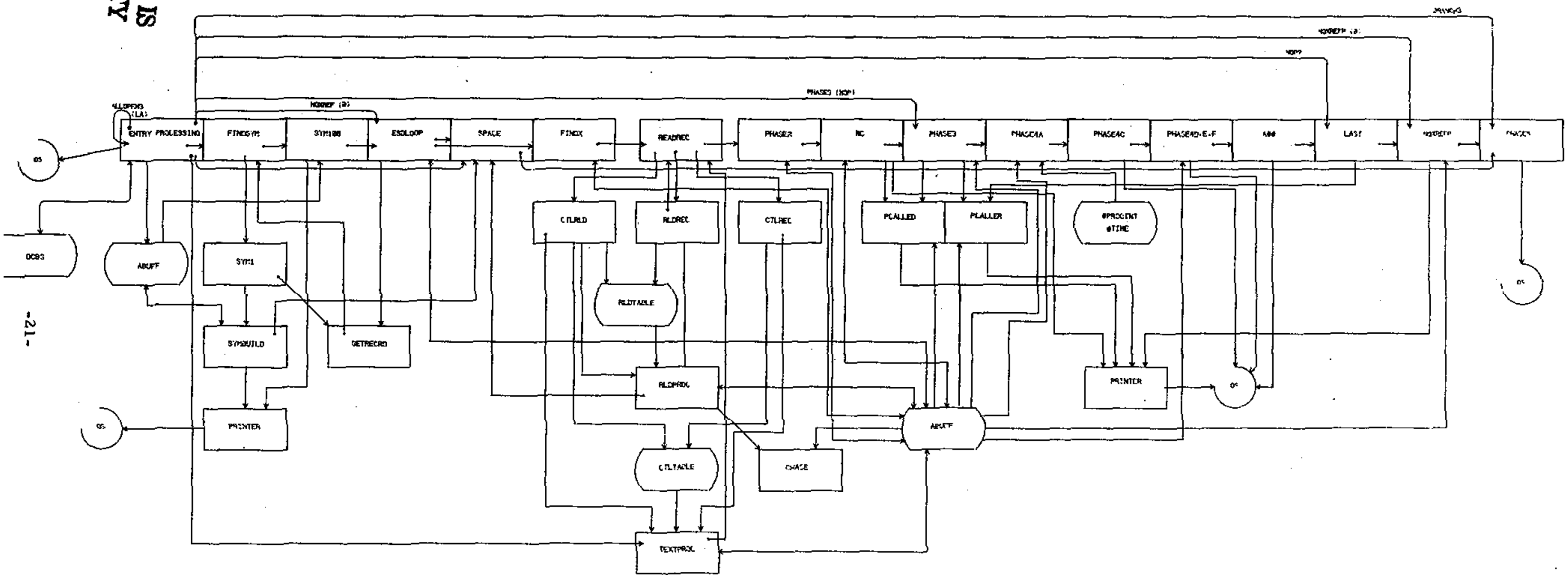


Figure 3-4

(BLANK)

4. RUN TIME LIBRARY

The only portions of the HAL/SM RTL considered in this section are those modified portions of HAL/S RTL which are included in the HAL/SM RTL and modules which are to be created for HAL/SM RTL.

Each module that appears in the structure chart and data flow in Figures 4-1 through 4-7 is described in more detail by an intermodule interface definition. More detailed definitions of module interface parameters for those modules being created for the HAL/SM RTL may be found in Reference 2.

Module: ONERR

Function: To transfer the error information associated with the ON ERROR statement to an error stack. (The module shall have two entry points, ONERR1 and ONERR2, to perform this function.)

Parameters: (For ONERR1) R0 - error group number and error number within the group.
R1 - error action indicator.
R2 - program flag descriptor.

(For ONERR2) R0 - error group number and error number within the group.
R1 - address of statement to which execution is transferred when the error occurs.

Notes:
1. The first execution of this routine links a portion of the error stack to the executing program by placing a pointer to the specified portion of the error stack in the ERRORLINK parameter of the program stack. Subsequent calls to this routine reference that link.
2. Each call to this routine fills a double word in the error stack.
3. This routine is invoked by using the HAL linkage convention.

Module: OFFERR

Function: To delete a specific error specification from a program error stack.

Parameters: R0 - error group number and error number within the group.

RTL I/O AND ERROR SYSTEMS STRUCTURE CHART

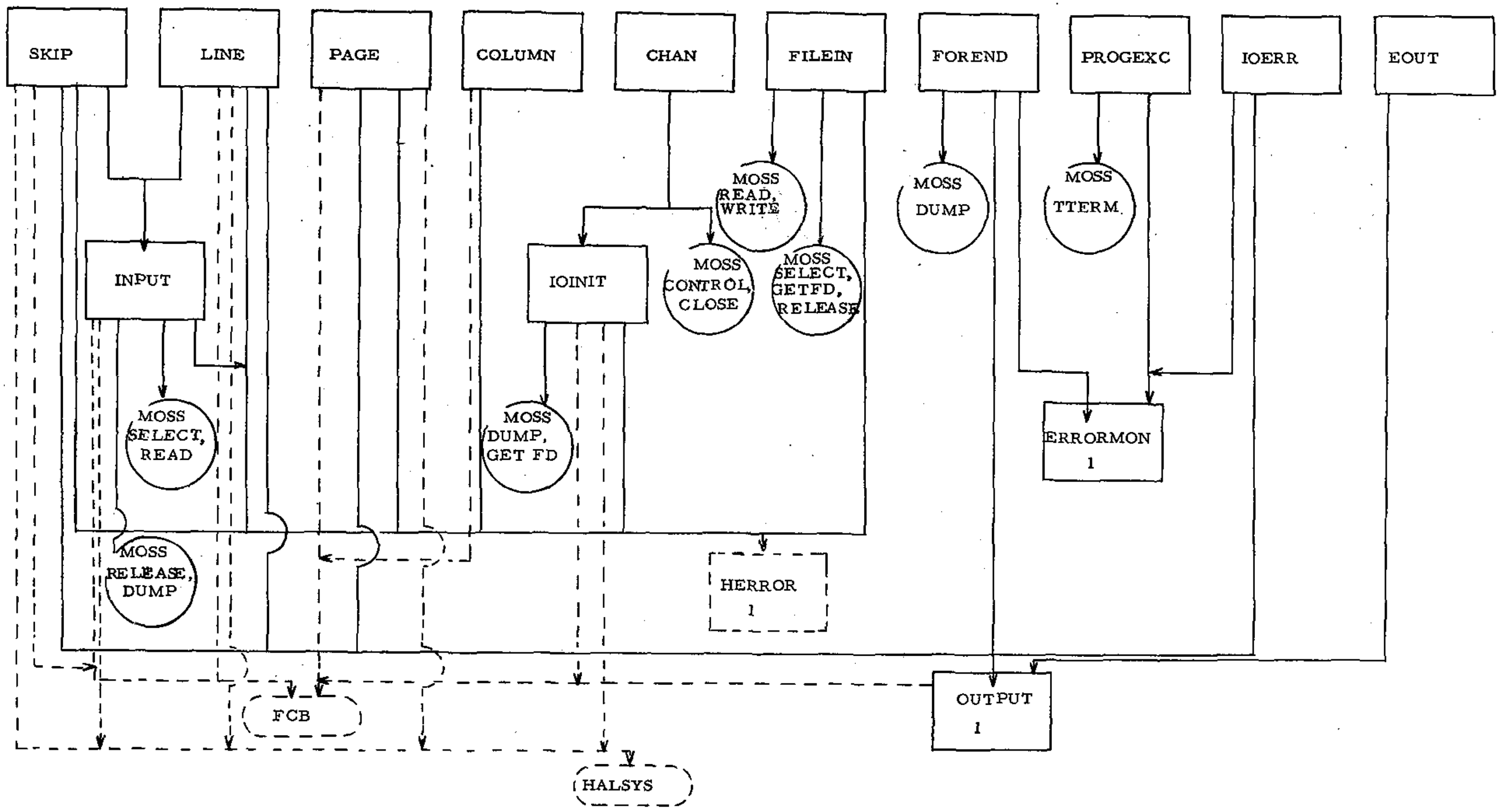


Figure 4-1

RTL INITIALIZATION, ERROR, AND REAL TIME SYSTEMS STRUCTURE CHART

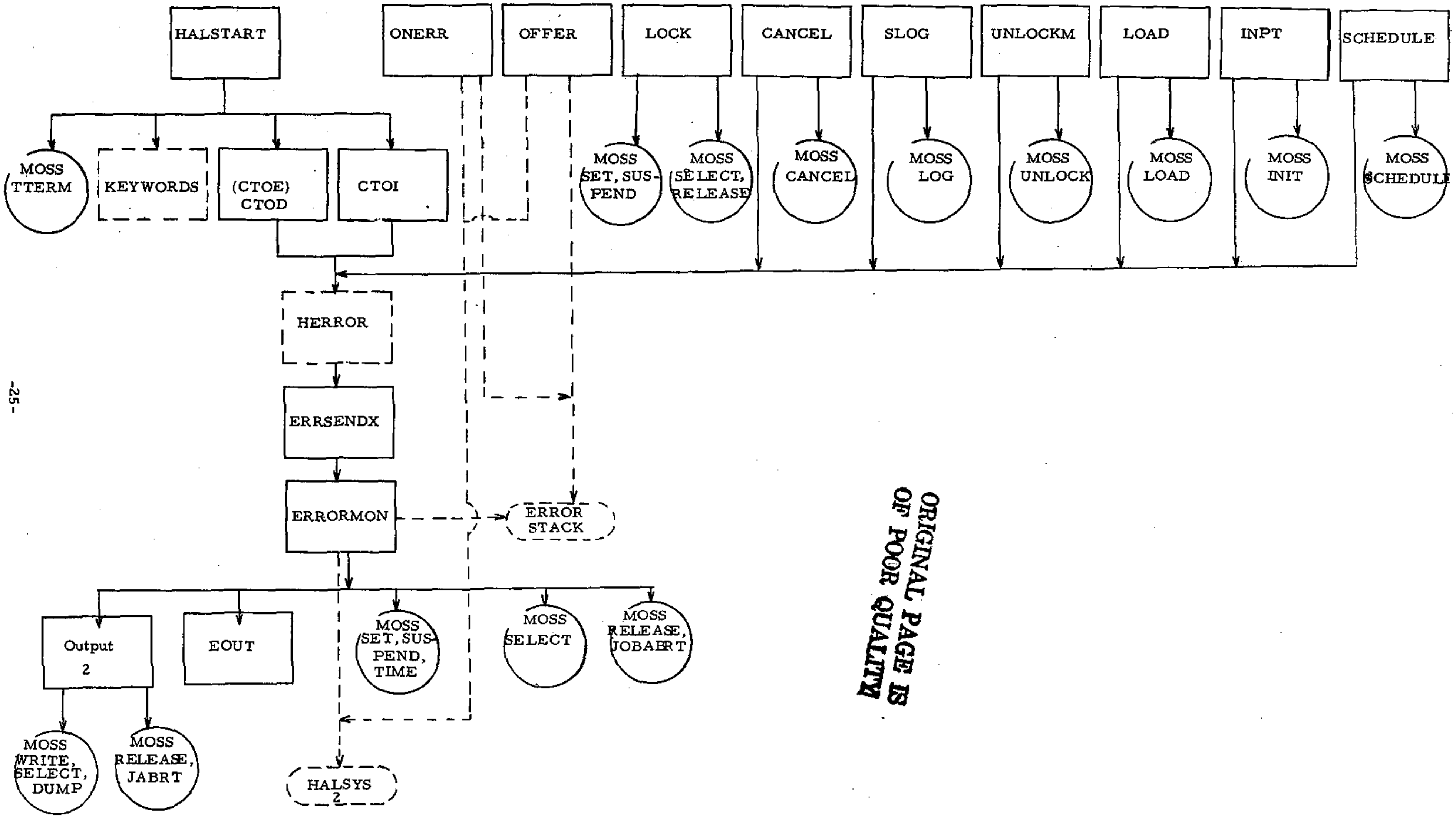


Figure 4-2

RTL REAL TIME SYSTEM STRUCTURE CHART

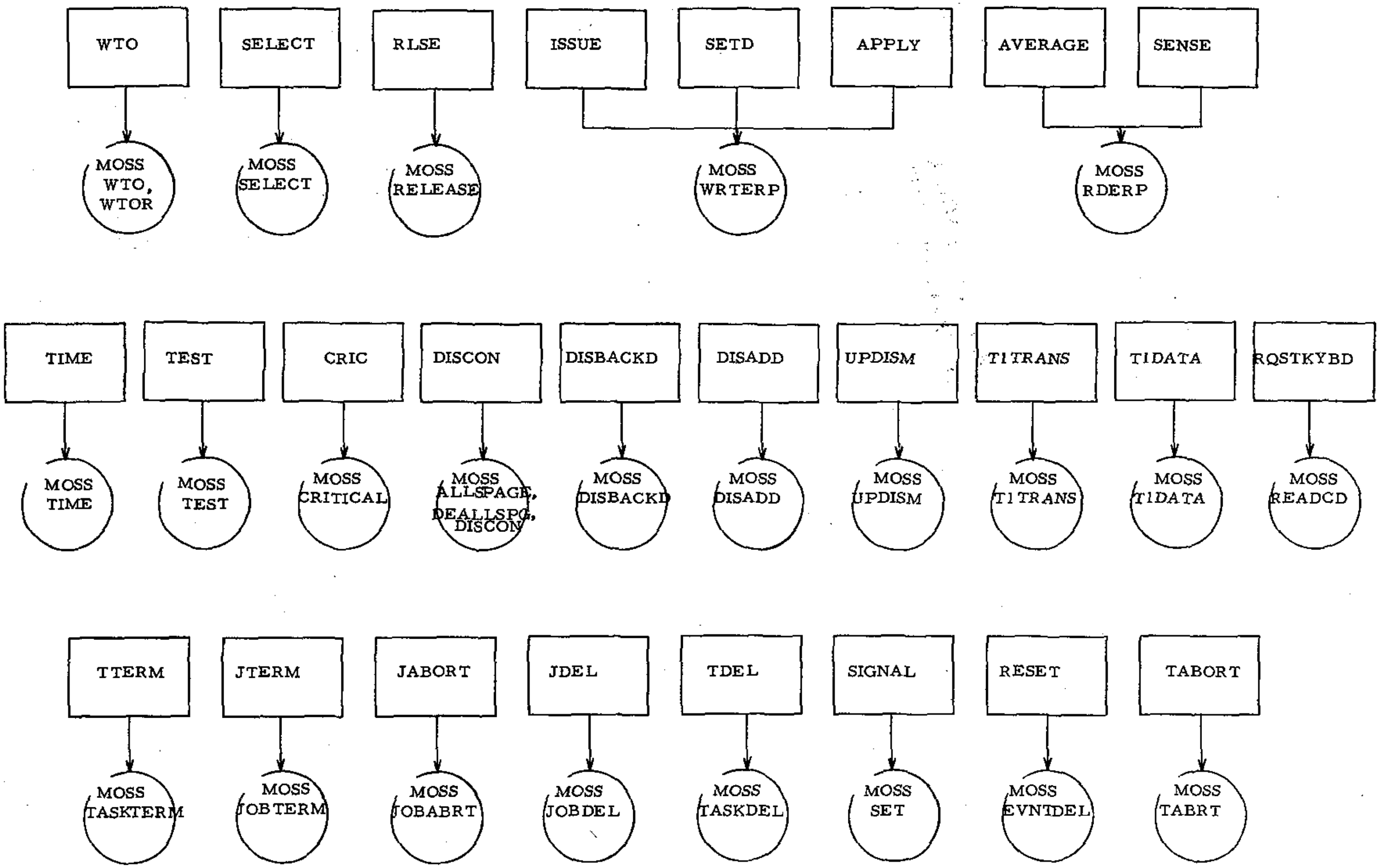


Figure 4-3

RTL REAL TIME SYSTEM DATA FLOW

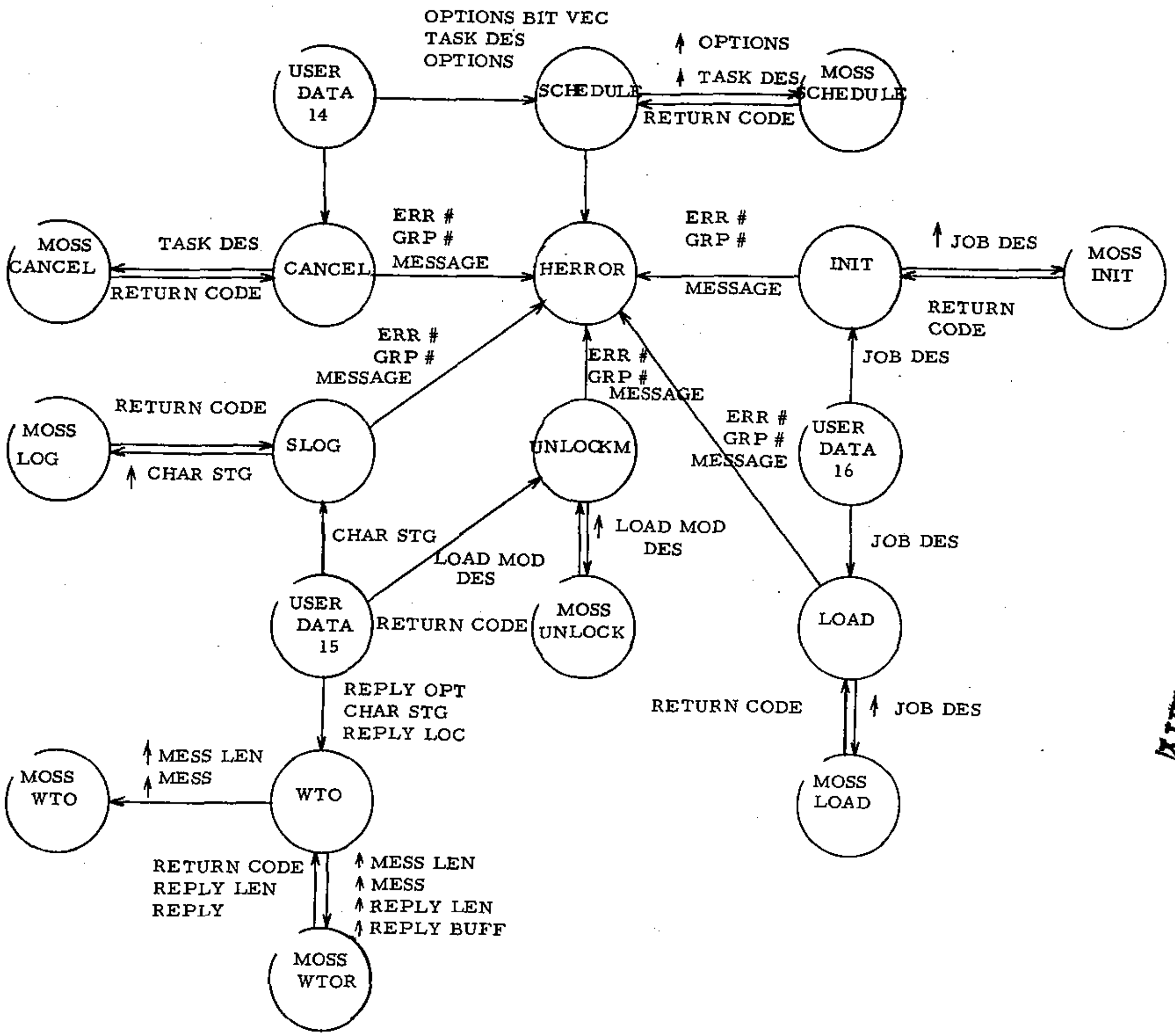


Figure 4-4

ORIGINAL PAGE IS OF POOR QUALITY

RTL REAL TIME SYSTEM DATA FLOW

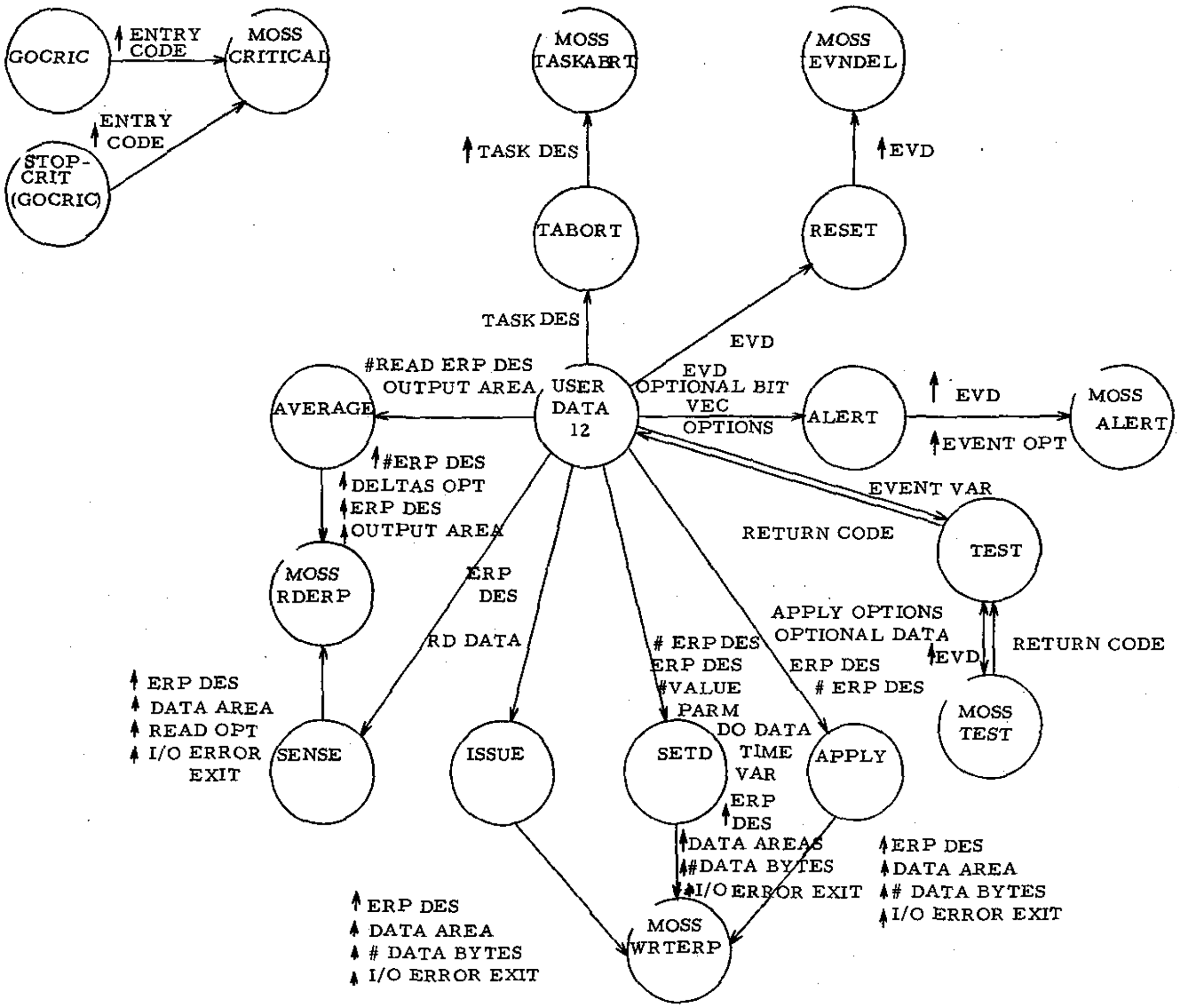


Figure 4-4
(Continued)

RTL CONTROL AND DISPLAY SYSTEM DATA FLOW

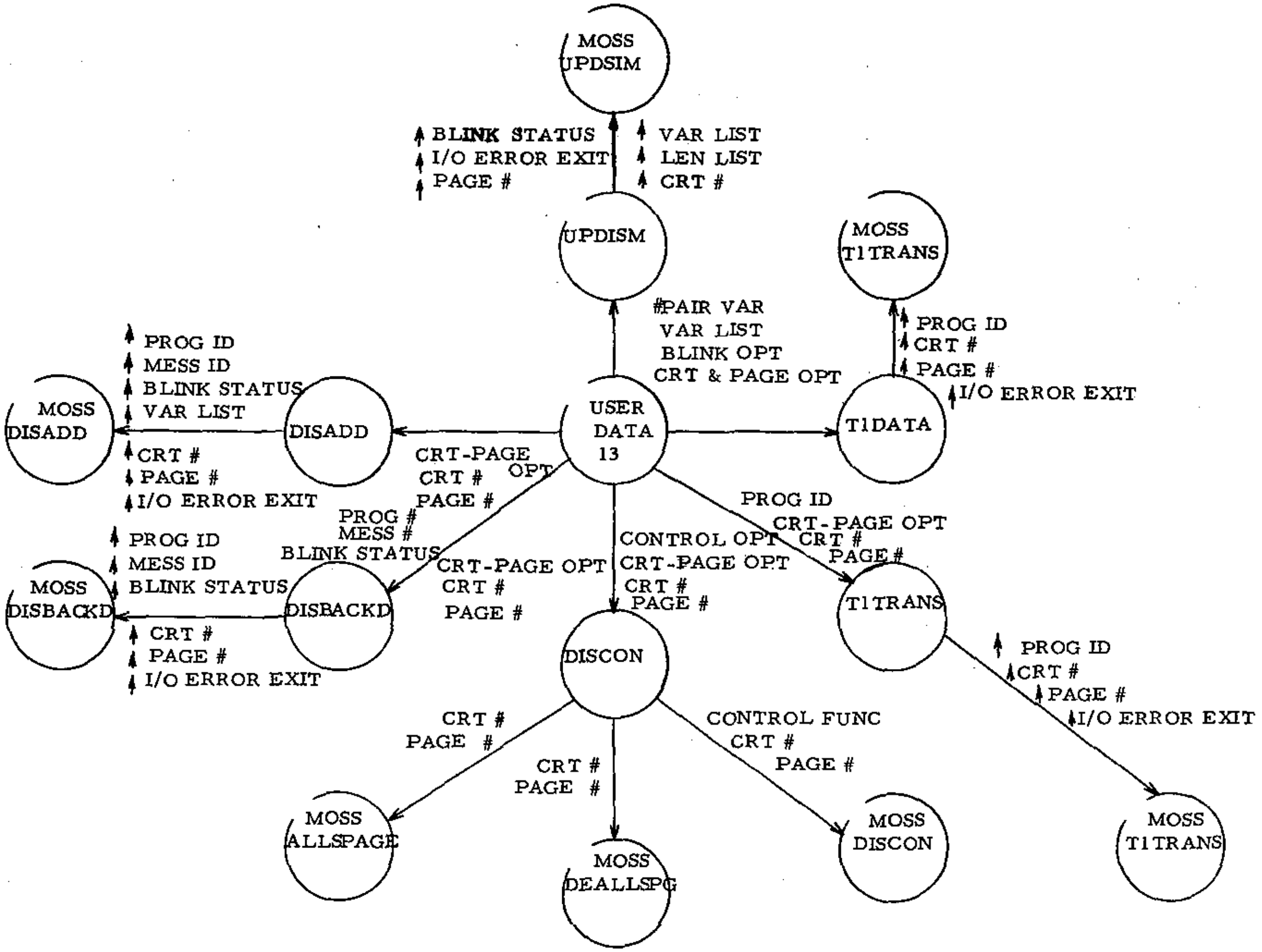


Figure 4-5

ORIGINAL PAGE IS OF POOR QUALITY

RTL I/O SYSTEM DATA FLOW

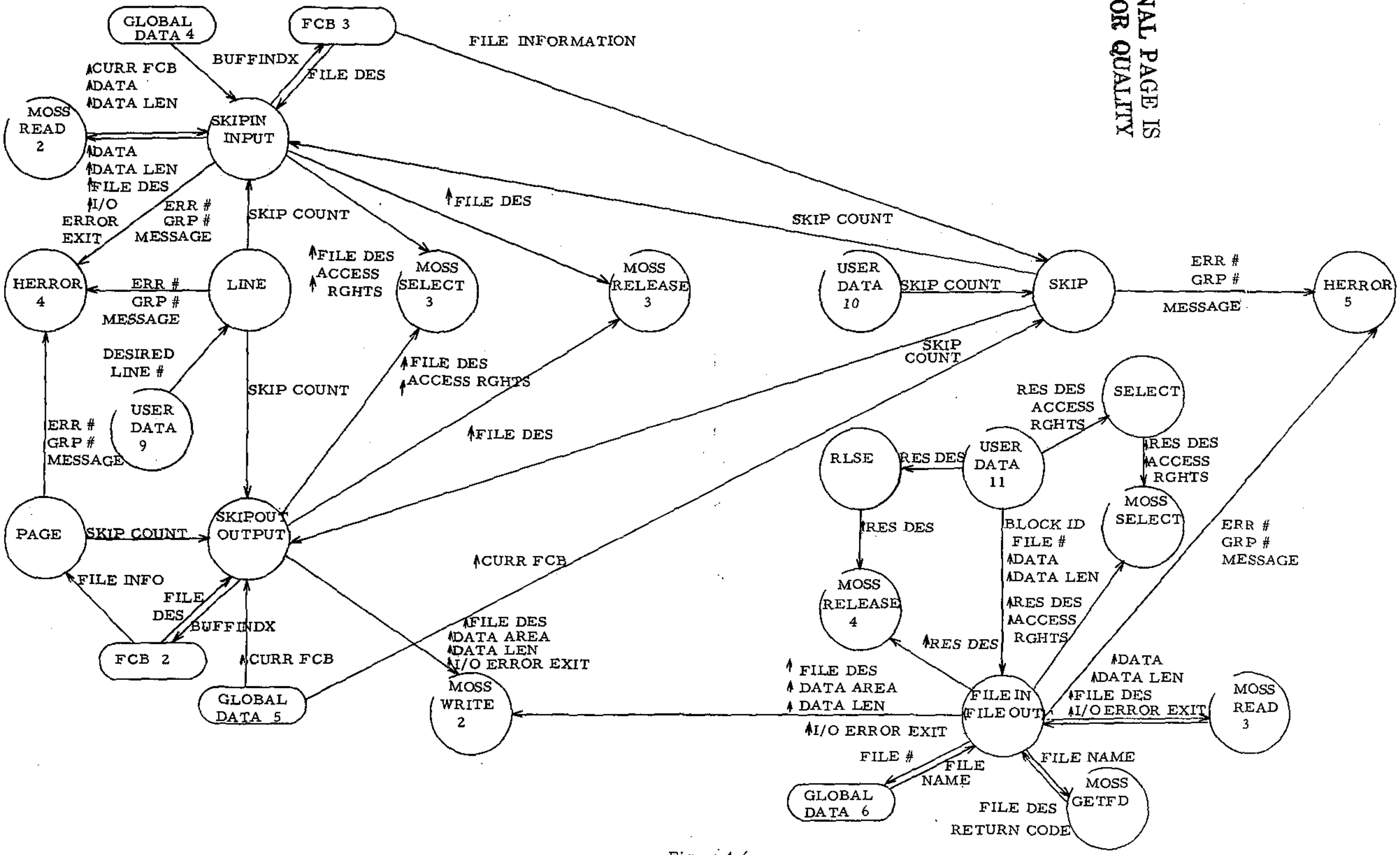


Figure 4-6

RTL I/O SYSTEM DATA FLOW

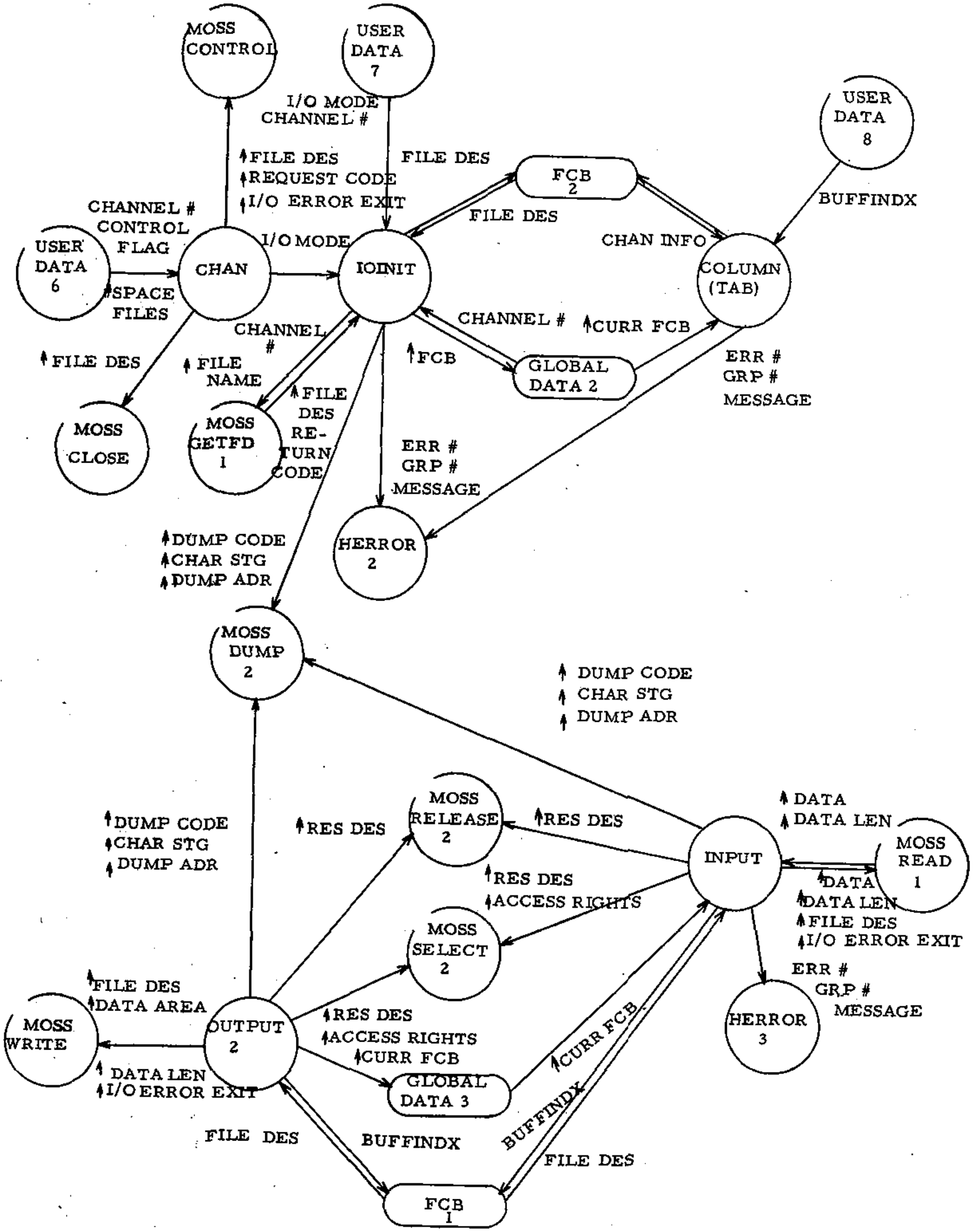


Figure 4-6
(Continued)

RTL ERROR AND REAL TIME SYSTEM DATA FLOW

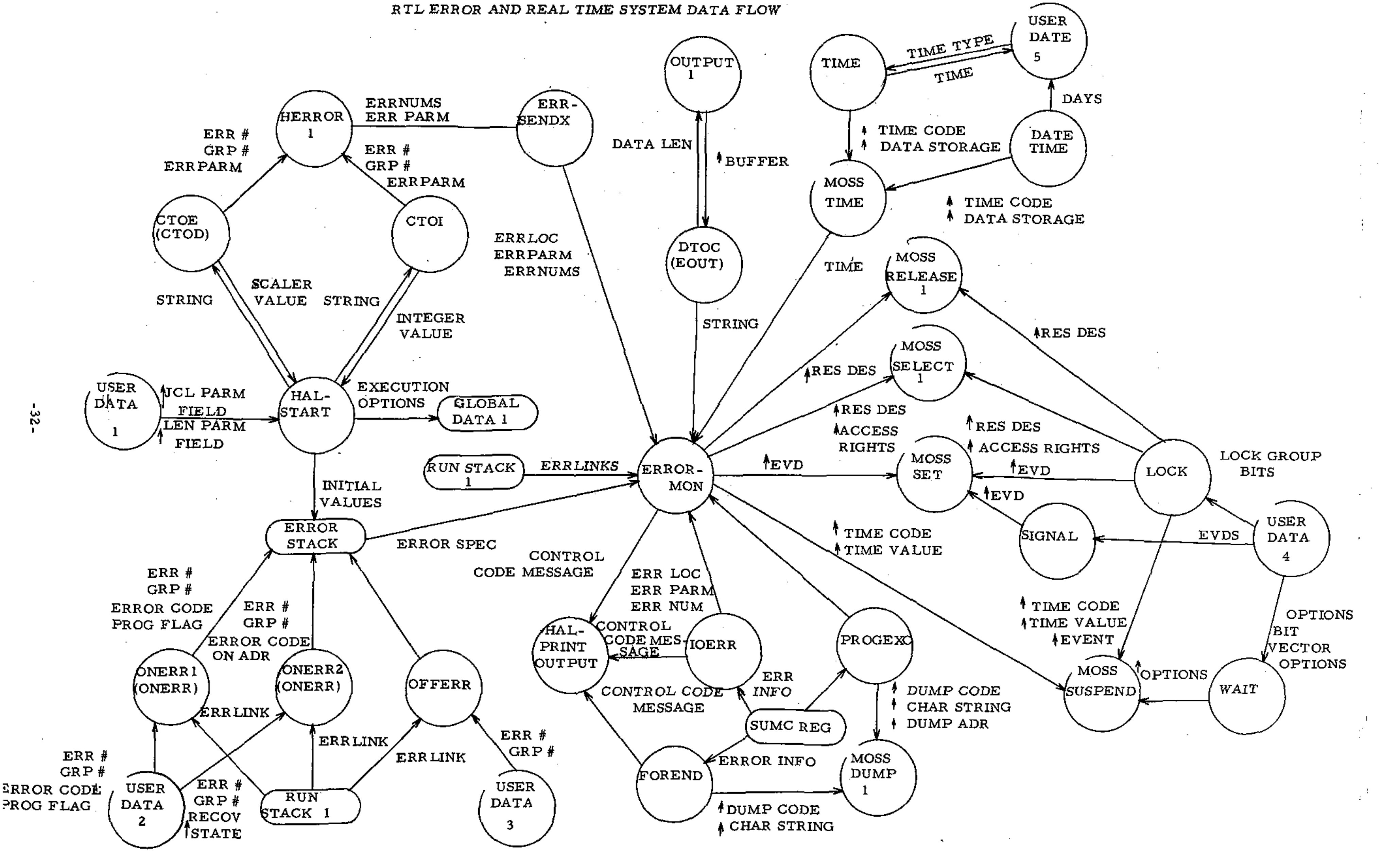


Figure 4-7

Notes: 1. This routine uses the ERRORLINK in the program stack to determine the address of the error stack then zeros the specified entry in the stack.
2. This routine is invoked by using the HAL linkage convention.

Module: SIGNAL

Function: To establish the parameter list for the MOSS SET SVC and invoke that SVC as required.

Parameters: FLAG_LIST - a list of the program flags to be set.

Note: This routine is invoked by using the linkage convention for NON-HAL programs.

Module: RESET

Function: To establish the parameter list for the MOSS EVNTDEL SVC and invoke that SVC as required.

Parameters: EVENT_VARIABLE_LIST - a list of event variable descriptors of event variables to be reset.

Notes: 1. The event variable must be communicated to MOSS as an event variable descriptor.
2. This routine is invoked by using the linkage convention for NON-HAL programs.

Module: CANCEL

Function: To establish the parameter list for the MOSS CANCEL SVC and invoke that SVC as required.

Parameters: TASK_ DESCRIPTOR - descriptor of the task to be canceled. If no descriptor is specified, the requesting task is canceled.

Notes: 1. If an error code is returned from MOSS, this routine must alert the HAL/SM error monitor.
2. This routine is invoked using the linkage convention for NON-HAL programs.

Module: TTERM

Function: To invoke the MOSS TASKTERM SVC as required.

Parameters: None.

Note: This routine is invoked using the HAL linkage convention.

Module: JTERM

Function: To invoke the MOSS JOBTERM SVC as required.

Parameters: None.

Note: This routine is invoked using the HAL linkage convention.

Module: JABORT

Function: To invoke the MOSS JOBABORT SVC as required.

Parameters: None.

Note: This routine is invoked by using the HAL linkage convention.

Module: TABORT

Function: To establish the parameter list for the MOSS TASKABRT SVC and invoke that SVC as required.

Parameters: TASK_DESCRIPTOR_LIST - a list of task descriptors of tasks to be aborted. If this parameter is omitted, the requesting task is aborted.

Notes: 1. If no parameter is specified, the calling task is aborted.
2. This routine requires NON-HAL linkage.

Module: SLOG

Function: To establish the parameter list for the MOSS LOG SVC and invoke the SVC as required.

Parameters: R0 - the address of the character expression to be written to the system log.

Notes: 1. If an error code is returned from MOSS, this routine must alert the HAL/SM error monitor.
2. This routine is invoked by using the normal HAL linkage convention.

Module: UNLOCKM

Function: To establish the parameter list for the MOSS UNLOCK SVC and invoke that SVC as required.

Parameters: LOAD_MODULE_DESCRIPTOR_LIST - list of descriptors of load modules to be unlocked.

Notes: 1. If an error code is returned from MOSS, this routine must alert the HAL/SM error monitor.
2. This routine is invoked by using the NON-HAL linkage convention.

Module: LOAD

Function: To establish the parameter list for the MOSS LOAD SVC and invoke that SVC as required.

Parameters: JOB_DESCRIPTOR - descriptor for the job to be loaded.

Notes: 1. If an error code is returned from MOSS, this routine must alert the HAL/SM error mointor.
2. This routine is invoked by using the NON-HAL linkage convention.

Module: INIT

Function: To establish the parameter list for the MOSS INIT SVC and invoke that SVC as required.

Parameters: JOB_DESCRIPTOR - descriptor of the job to be initiated.

Notes: 1. If an error code is returned from MOSS, this routine must alert the HAL/SM error monitor.
2. This routine is invoked by using the NON-HAL linkage convention.

Module: JDEL

Function: To invoke the MOSS JOBDEL SVC as required.

Parameters: None.

Note: This routine is invoked by using the HAL linkage convention.

Module: TDEL

Function: To invoke the MOSS TASKDEL SVC as required.

Parameters: None.

Note: This routine is invoked by using the HAL linkage convention.

Module: SCHEDULE

Function: To establish the parameter list for the MOSS SCHEDULE SVC and invoke that SVC as required.

Parameters: OPTIONS_BIT_VECTOR - bit vector indicating the presence and format of the scheduling options.
TASK_DESCRIPTOR - description of the task to be scheduled.
OPTIONS_LIST - list of the options indicated by the bit vector.

Notes:
1. If an error code is returned from MOSS, this routine must alert the HAL/SM error monitor.
2. This routine is invoked by using the NON-HAL linkage conventions.

Module: WAIT

Function: To establish the parameter list for the MOSS SUSPEND SVC and invoke that SVC as required.

Parameters: OPTIONS_BIT_VECTOR - a bit variable indicating the presence and format of the WAIT options.
OPTIONS_LIST - list of the options indicated by the bit vector.

Note: This routine is invoked by the NON-HAL linkage convention.

Module: ALERT

Function: To establish the parameter list for the MOSS ALERT SVC and invoke that SVC as required.

Parameters: EVENT_VARIABLE - bit vector corresponding to the event variable to be alerted.
OPTIONS_BIT_VECTOR - bit vector indicating presence and format of optional parameters.
OPTIONS_LIST - list of the options indicated by the bit vector.

Note: This routine is invoked by using NON-HAL linkage convention.

Module: AVERAGE

Function: To establish the parameter list for the MOSS RDERP SVC and invoke that SVC as required.

Parameters: R0 - the number of ERP readings to be averaged.
R1 - address of the descriptor of the ERP to be read.
R2 - address of variable into which the average is to be stored.

Notes:
1. The address of an I/O error exit routine shall be provided via the SVC for resolving I/O errors associated with the read operation.
2. This routine is invoked by using the normal HAL linkage conventions.

Module: SENSE

Function: To establish the parameter list for the MOSS RDERP SVC and invoke that SVC as required.

Parameters: DELTAS_OPTIONS - a number indicating the presence of the read AI delta option.
NUMBER_OF_ERPS - number of ERP's to be read.
ERP_LIST - list of ERP descriptors of ERP's to be read.
OUTPUT_AREA - list of variables into which the results of the read are to be placed.

Notes:
1. The address of an I/O error exit routine shall be provided via the SVC for resolving I/O errors associated with the read operation.
2. This routine is invoked by using the NON-HAL linkage.

Module: ISSUE

Function: To establish the parameter list for the MOSS WRTERP SVC and invoke that SVC as required.

Parameters: ERP_DESCRIPTOR - the R0 ERP designator to which the issue applies.
R0_DATA - the list of parameters which form the data to be issued.

Notes:
1. The address of an I/O error exit routine shall be provided via the SVC for resolving I/O errors associated with the write operation.

2. This routine is invoked by using a NON-HAL linkage convention.

Module: SETD

Function: To establish the parameter list for the MOSS WRTERP SVC and invoke that SVC as required.

Parameters: NUMBER_ERP_DESIGNATORS - the number of D0 ERP designators to be set.

ERP_DESIGNATOR_LIST - the list of discretes to be set.

NUMBER_OF_D0_VALUES - the number of output value to be set.

D0_VALUE_LIST - the list of output values to be applied to the D0.

TIME_VALUE - time in milliseconds that D0 is to be pulsed.

Notes:
1. The time value may only be used when one D0 and output value are specified.
2. The address of an I/O error exit routine shall be provided via the SVC for resolving I/O errors associated with the write operation.
3. The routine is invoked by using a NON-HAL linkage convention.

Module: APPLY

Function: To establish the parameter list for the MOSS WRTERP SVC and invoke that SVC as required.

Parameters: APPLY_OPTIONS - indicator of special options for WRTERP.

NUMBER_OF_ERP_DESIGNATORS- number of A0 ERP designators to which a value is to be applied.

VALUE LIST - list of values to be applied.

ERP_LIST - list of designators to which values are to be applied.

Notes:
1. The address of an I/O error exit routine shall be provided via the SVC for resolving I/O errors associated with the WRITE operation.
2. This routine is invoked by using a NON-HAL linkage convention.

Module: WTO

Function: To establish the parameter lists for either the MOSS WTO or WTOR SVC and invoke either of these SVC's as required.

Parameters: REPLY_OPTION - a bit variable indicating whether or not a reply is to be expected.

MESSAGE - a character string to be transmitted to the operator.

REPLY_BUFFER - the character variable into which the reply is to be stored.

Notes:
1. WTOR is to be invoked if a reply is required.
2. WTO is to be invoked if no reply is required.
3. This routine is invoked by using a NON-HAL linkage convention.

Module: DISCON

Function: To establish the parameter lists to the MOSS ALLSPAGE, DEALLSPG, or DISCON SVC's and invoke one of these SVC's as required.

Parameters: CONTROL_OPTIONS - the option indicator as to which control function is to be performed.

CRT_PAGE_OPTION - bit vector indicating the specification of a CRT and PAGE number.

CRT_# - the number of the CRT to which the control is directed.

PAGE_# - the number of the PAGE to which the control is directed.

Notes:
1. The ALLSPAGE SVC is invoked if the control option indicates allocate page.
2. The DEALLSPG SVC is invoked if the control option indicates deallocate page.
3. The DISCON SVC is invoked for all other control options.
4. This routine is invoked by using a NON-HAL linkage convention.

Module: DISBACKD

Function: To establish the parameter list to the MOSS DISBACKD SVC and invoke that SVC as required.

Parameters: PROGRAM_IDENTIFIER - the identifier by which the data set containing the message is identified.

MESSAGE_ID_# - the record number within the data set containing the message.

BLINK_OPTION - a bit variable indicating the blink status option.

CRT_PAGE_OPTION - indicator as to specification of a CRT and PAGE number.

CRT_# - the CRT number.

PAGE_# - the PAGE number.

Notes: 1. The address of an I/O error exit routine shall be provided via the SVC for resolving I/O errors associated with the display operation.

2. This routine is invoked by using a NON-HAL linkage convention.

Module: DISADD

Function: To establish the parameter list for the MOSS DISADD SVC and invoke that SVC as required.

Parameters: PROGRAM_IDENTIFIER - the program identifier by which the data set containing the addon message is identified.

MESSAGE_ID_# - the number specifying the record number within the data set which contains the message.

BLINK_OPTION - the bit variable which indicates the blink status option.

#_PAIR_VARIABLES - the number of variable control word/variable pairs which follow.

VARIABLE_LIST - the list of variable control words and associated variables.

CRT_PAGE_OPTION - indicator as to specification of a CRT and PAGE number.

CRT_# - the CRT number.

PAGE_# - the PAGE number.

Notes: 1. The address of an I/O error exit routine shall be provided via the SVC for resolving I/O errors associated with the addon operation.

2. This routine is invoked by using a NON-HAL linkage convention.

Module: UPDISM

Function: To establish the parameter list to the MOSS UPDISM and invoke that SVC as required.

Parameters: #_VARIABLE_PAIRS - the number of pairs of variable control words and associated variables to be transmitted.

VARIABLE_LIST - the list of variable control words and associated variables.

BLINK_OPTION - the bit variable which specifies the blink option.

CRT_PAGE_OPTION - the bit variable that indicates the specification of a CRT and PAGE number.

CRT_# - the CRT number.

PAGE_# - the PAGE number.

Notes: 1. The address of an I/O error exit routine shall be provided via the SVC for resolving I/O errors associated with the update operation.
2. This routine is invoked by using a NON-HAL linkage convention.

Module: T1 TRANS

Function: To establish the parameter list for the MOSS T1 TRANS SVC and invoking that SVC as required.

Parameters: TYPE1_PROGRAM_IDENTIFIER - the program identifier of the TYPE1 program in the program library.

CRT_PAGE_OPTION - the bit variable which specifies the presence of a CRT and PAGE number.

CRT_# - the CRT number.

PAGE _# - the PAGE number.

Notes: 1. The address of an I/O error exit routine shall be provided via the SVC for resolving I/O errors associated with the transmit operation.
2. This routine is invoked by using a NON-HAL linkage convention.

Module: T1DATA

Function: To establish the parameter list for the MOSS T1DATA SVC and invoke that SVC as required.

Parameters: TYPE1_PROGRAM_IDENTIFIER - identifier of TYPE1program for which the data is specified.

#_OF_VARIABLES - number of TYPE1 update variables.

VARIABLE_LIST - list of update variables.

CRT_PAGE_INDICATOR - bit indicator of CRT - PAGE number specification.

CRT_# - the CRT number (if specified).

PAGE_# - the PAGE number (if specified).

Notes: 1. The entry point to an I/O error exit routine shall be provided via the SVC.
2. This routine is invoked by using a NON-HAL linkage convention.

Module: RQSTKYBD

Function: To establish the parameter list for the MOSS READCD and invoke that SVC as required.

Parameters: ADR_CHARACTER_EXP - address of character data area.

WAIT_INDICATOR - the automatic wait indicator; also the parameter in which a return code is specified.

Notes: 1. The entry point of an I/O error exit routine is provided via the SVC list.
2. This routine is invoked by using a NON-HAL linkage convention.

Module: SELECT

Function: To establish the parameter list for the MOSS SELECT SVC and invoke that SVC as required.

Parameters: DESCRIPTOR_LIST - a list of paired variables; the first variable specifies the resource descriptor and the second is a bit vector which specifies access rights.

Note: This routine is invoked by using a NON-HAL link convention.

Module: RLSE

Function: To establish the parameter list for the MOSS RELEASE SVC and invoke that SVC as required.

Parameters: DESCRIPTOR_LIST - list of resource descriptors.

Note: This routine is invoked by using a NON-HAL linkage convention.

Module: CHAN

Function: To determine which of the MOSS SVC's CONTROL or CLOSE is to be invoked; establish the parameter list to the appropriate SVC and invoke it as required.

Parameters: CHANNEL_# - channel to be controlled.
FUNCTION_INDICATOR - control function to be performed.
FILE_COUNT - if files are to be spaced, this specifies the count.

Notes: 1. This module must invoke the RTL routine IOINIT to determine the required file descriptors for the SVC invocation.
2. Also the entry point to an I/O error exit routine shall be provided with the CONTROL SVC.
3. This routine is invoked by using a NON-HAL linkage convention.

Module: GOCRIC (STOPCRIC)

Function: To invoke the MOSS CRITICAL SVC with the enter critical processing mode parameter.

Parameters: None.

Notes: 1. This routine is invoked by using HAL linkage conventions.
2. The mode parameter sent to the SVC is determined by which of the two entries, GOCRIC or STOPCRIC, is invoked.

Module: TEST

Function: To test an event variable; if the variable is zero, a zero is returned; however, a non-zero value is assumed to be a MOSS event variable descriptor in which case the MOSS TEST SVC is invoked.

Parameters: R0 - the event variable.

Note: This routine is invoked by using the normal HAL linkage convention.

Module: TIME (DATE)

Function: To establish the parameter list for the MOSS TIME SVC and invoke that SVC as required.

Parameters: R0 - an indicator of the type of time value requested.
F0 - return time value (TIME Entry).
R1 - return number of days (DATE Entry).

Notes: 1. This routine is invoked (as a FUNCTION by) using the normal HAL linkage convention.
2. The value returned from the TIME Entry is a double precision floating point number equal to the value of time in milliseconds.
3. The value returned from the DATE Entry contains the number of days portion of the MOSS GMT value.

Module: COLUMN

Function: To increment the column index count of the associated I/O device.

Parameters: R0 - the buffer index count (I/O Column Counter - 1).

Notes: 1. The I/O device to which the COLUMN is directed is determined from the global data area (HALSYS).
2. All data associated with the I/O device is found in the File Control Block data section.
3. A check is made on the index to determine if it is within the allowable range. If it is not, an error message is sent to the error monitor.
4. This routine is invoked by using normal HAL linkage.

Module: FILEIN (FILEOUT)

Function: To initiate and perform file I/O operations.

Parameters: R0 - block identification.
R1 - file number.
R2 - address of data area.
R3 - length of data area.

Notes:
1. This routine uses the file number to index into the FILES table to find the associated file control block. If no file descriptor is found in the file control block, the MOSS GETFD SVC is invoked with the file name as a parameter to yield the file descriptor. If no file descriptor is found, an error message is sent to the error monitor. If a file descriptor is found, it is placed in the file control block.
2. The SELECT, READ, WRITE, and RELEASE SVC's are invoked as required.
3. FILEIN performs input; whereas, FILEOUT performs output.
4. This routine is invoked by using the normal HAL linkage convention.

Module: IOINIT

Function: To initiate sequential I/O operations.

Parameters: R0 - I/O mode.
R1 - channel number.

Notes:
1. The I/O mode is checked for accuracy. If inaccurate, the DUMP SVC is invoked.
2. The channel number is used to find the associated file control block, then the file control block is interrogated to find the file descriptor. If a file descriptor has not been entered, the MOSS GETFD SVC is invoked to determine an appropriate one. If none exists an error is sent to the error monitor.
3. If a file descriptor exists, it is entered in the file control block.
4. This routine is invoked by using the normal HAL linkage convention.

Module: INPUT

Function: To perform input from a specified sequential I/O device.

Parameters: None.

Notes:
1. If data is read, R0 contains the length of data and R1 contains the address of the data. R0 = zero indicates data is not changed.
2. Error checks are made on parameters in the FCB associated with the device. Any errors are sent to the error monitor. Severe errors require use of the MOSS DUMP SVC.
3. The MOSS SELECT, READ, and RELEASE SVC's are used to perform the input operation.
4. The routine is invoked by using the normal HAL linkage convention.
5. CIN and SKIPIN are entries in INPUT. Their parameters are:
   CIN - R3 = address of character string to be filled.
   SKIPIN - R0 = skip count (must be >0).

Module: LINE

Function: To perform the LINE function associated with sequential I/O devices.

Parameters: R0 - desired line number.

Notes:
1. The FCB for the associated I/O device is determined from information in the global data area.
2. The calculation of the number of lines to be skipped is determined using FCB information and SKIPOUT or SKIPIN is invoked (depending on the I/O mode) to perform the required skip operation.
3. Any line errors are reported to the error monitor.
4. This routine is invoked by using the normal HAL linkage convention.

Module: OUTPUT

Function: To perform output to a specified sequential I/O device.

Parameters: R0 - length of buffer segment.
R1 - address of buffer.

Notes:
1. This routine uses the buffer index, etc., in the FCB as well as the file descriptor for preparing the output data for transmission to MOSS.
2. The MOSS SELECT, WRITE, and RELEASE SVC's are used for data transmission.

3. COUT, HALPRINT, FLUSH, and SKIPOUT are also entries to this module. For COUT, R3 contains the address of a character string to be output. For HALPRINT, R1 contains the address +1 of a HAL type character string, and R0 contains an output spacing control code. For FLUSH, R0 is used to return buffer position on first entry and to set buffer position on secondary entries. For SKIPOUT, R0 contains skip count.
4. This routine is invoked by using the normal HAL linkage convention.
5. Error checks are made on mode, etc., and severe error conditions detected require use of the MOSS DUMP SVC.

Module: PAGE

Function: To perform the paging function associated with sequential output devices.

Parameters: R0 - page count (number of pages).

Notes:
1. The FCB for the associated output device is determined from the global data area.
2. The number of pages is resolved into number of lines to be skipped by using maximum lines/page. SKIPOUT is then invoked to skip the appropriate number of lines.
3. Any errors such as negative page count are reported to the error monitor.
4. This routine is invoked by using the normal HAL linkage convention.

Module: SKIP

Function: To position a particular I/O device ahead by a specified number of lines.

Parameters: R0 - skip count (number of lines).

Notes:
1. The associated I/O device is specifed in the global data area.
2. The line information is found in the FCB. The skip count is determined from the line information and either SKIPOUT or SKIPIN is invoked.
3. Any errors detected are reported to the error monitor.
4. This routine is invoked using the normal HAL linkage convention.

Module: ERRORMON

Function: To process certain program errors as to any action that should be taken on their occurrence.

Parameters: None.

Notes:
1. The error location, option, message address, group, and number can be found in the global data area.
2. The routine finds the program error vector from the stack ERRORLINK parameter. This vector is interrogated for a match with the error that has occurred.
3. The MOSS SELECT, RELEASE, SUSPEND, SET, TIME, and JOBART SVC's are used to perform the specified error recovery action.
4. This routine is invoked by using the normal HAL linkage convention.

Module: LOCK (UNLOCK)

Function: To perform coordination of accesses to individual data items by update blocks.

Parameters: R0 - lock group bits.

Notes:
1. The lock group variable shall be placed in a job common area which must be SELECTed and RELEASEd using MOSS SVC's.
2. The MOSS SUSPEND SVC is used to wait the task on the completion of use of locked group data.
3. The MOSS SET SVC is used to indicate completion of use of locked group data.
4. This routine is invoked by using the normal HAL linkage convention.

Module: HALSTART

Function: To perform task initialization prior to initiation and cleanup prior to termination.

Parameters: R1 - address of JCL parameter list.

Notes:
1. This routine processes execution time options specified via JCL by doing error checking on the specifications and then initializing the associated global data parameters.

2. This routine initializes such area as the error vector stack, etc.
3. Any errors detected are reported to the error monitor.
4. The MOSS TTERM SVC is used to terminate the task after the required cleanup is performed.
5. This routine is invoked by using standard OS/360 linkage convention.

Module: PROGEXC

Function: To process MOSS program exception interrupts.

Parameters: None.

Notes:
1. The SUMC registers are interrogated to determine the location of the error and other error information.
2. Errors are handled by reporting to error monitor; continuing by use of MOSS CONTINUE SVC; taking a dump by using the MOSS DUMP SVC; and/or task termination by use of MOSS TASKTERM SVC.
3. This routine is invoked by using standard MOSS linkage convention.

Module: FORCEND

Function: To process MOSS fault or error conditions; i.e., errors not processed by the program exception or I/O error exit routines.

Parameters: None.

Notes:
1. The SUMC registers are interrogated to determine the location of the error and other error information.
2. An error summary is performed by the routine and the MOSS DUMP SVC is used to perform a system dump.
3. The routine is invoked by using the standard MOSS linkage convention.

Module: IOERR

Function: To process exceptional conditions encountered in processing I/O requestes.

Parameters: None.

Notes:

1. The SUMC registers are interrogated to determine the location of the error and other error information.
2. The error is reported to the error monitor.
3. There shall be entries into the module for each of the types of I/O errors.
4. This routine is invoked by using the standard MOSS linkage convention.

(BLANK)

APPENDIX

HAL/SM WORKING GRAMMAR

Following is the HAL/SM working grammar, written in standard BNF notation.

*** BNF XREF ***

```
 1      <COMPILATION> ::= <COMPILE LIST> _|_

 2      <COMPILE LIST> ::= <COMPILE BLOCK>
~~ 3          | <COMPILE BLOCK> <COMPILE LIST>~~

~~ 4      <COMPILE BLOCK> ::= <LABEL> :~~ <LABELED COMP BLOCK> ; <CLOSING>

~~ 5      <LABELED COMP BLOCK> ::= EXTERNAL <TEMPLATE>~~
 6          | <COMP BLOCK DEFINITION>

 7      <TEMPLATE> ::= <COMP BLOCK STMT> ; <DEFINITION GROUP>

 8      <COMP BLOCK DEFINITION> ::= <COMP BLOCK STMT> ; <BLOCK BODY>

 9      <COMP BLOCK STMT> ::= TASK <ACCESS OPTION>
~~10          | COMPOOL <RIGID & ACCESS OPTIONS>~~
11          | <FUNCTION HEADER> <REENTRANT & ACCESS OPTIONS>
~~12          | <PROCEDURE HEADER> <REENTRANT & ACCESS OPTIONS>~~

~~13      <RIGID OPTION> ::= RIGID~~
14          | <NIL>

15      <ACCESS OPTION> ::= ACCESS
~~16          | <NIL>~~

~~17      <RIGID & ACCESS OPTIONS> ::= <RIGID OPTION> <ACCESS OPTION>~~
18          | <ACCESS OPTION> <RIGID OPTION>

19      <FUNCTION HEADER> ::= FUNCTION <TYPE SPEC> <PARM LIST>

20      <PROCEDURE HEADER> ::= PROCEDURE <PARM LIST>

21      <REENTRANT & ACCESS OPTIONS> ::= <REENTRANT OPTION> <ACCESS OPTION>

22      <REENTRANT OPTION> ::= REENTRANT
~~23          | <NIL>~~

~~24      <PARM LIST> ::= ( <ID LIST> )~~
25          | ASSIGN ( <ID LIST> )
~~26          | ( <ID LIST> ) ASSIGN ( <ID LIST> )~~
27          | <NIL>

28      <ID LIST> ::= <IDENTIFIER>
~~29          | <IDENTIFIER> , <ID LIST>~~

~~30      <BLOCK BODY> ::= <DEFINITION GROUP> ; <STATEMENT LIST>~~

~~31      <DEFINITION GROUP> ::= <DEFINITION ELEMENT>~~
32          | <DEFINITION ELEMENT> ; <DEFINITION GROUP>

33      <STATEMENT LIST> ::= <LAB STMT>
~~34          | <LAB STMT> ; <STATEMENT LIST>~~

~~35      <LAB STMT> ::= <STATEMENT>~~
36          | <LABEL> : <LAB STMT>
```

```
~~79          | FLAG~~
80          | DCW
~~81          | <ARITH SPEC> <OPTIONAL PREC SPEC>~~

~~82    <OPTIONAL ARRAYNESS> ::= ( <DIMENSIONS> )~~
83          | <NIL>

84    <ARITH SPEC> ::= <OPTIONAL ARITH TYPE SPEC>
~~85          | <OPTIONAL PREC SPEC>~~

~~86    <PREC SPEC> ::= SINGLE~~
87          | DOUBLE
~~88          | <RADIX>~~

~~89    <OPTIONAL PREC SPEC> ::= <PREC SPEC>~~
90          | <NIL>

91    <ARITH TYPE SPEC> ::= INTEGER
~~92          | SCALAR~~
93          | MATRIX <OPTIONAL ARRAYNESS>
~~94          | VECTOR <OPTIONAL ARRAYNESS>~~

~~95    <OPTIONAL ARITH TYPE SPEC> ::=~~ <ARITH TYPE SPEC>
96          | <NIL>

97    <LITERAL EXP OR *> ::= <LITERAL EXP>
~~98          | *~~

~~99    <LITERAL EXP> ::= <EXPRESSION>~~

~~100   <MINOR ATTR LIST> ::= <MINOR ATTRIBUTE>~~
101         | <MINOR ATTRIBUTE> <MINOR ATTR LIST>

102   <MINOR ATTRIBUTE> ::= STATIC
~~103         | AUTOMATIC~~
104         | DENSE
~~105         | ALIGNED~~
106         | ACCESS
~~107         | LOCK ( <LITERAL EXP OR *> )~~
108         | RIGID
~~109         | NONHAL ( <NUMBER> )~~
110         | <INITIALIZATION>

111   <INITIALIZATION> ::= INITIAL ( <VALUE LIST> )
~~112         | CONSTANT ( <VALUE LIST> )~~

~~113   <VALUE LIST> ::= <VALUE>~~
114         | <VALUE> , <VALUE LIST>

115   <VALUE> ::= <LITERAL EXP OR *>
~~116         | <LITERAL EXP> # <REPEATABLE VALUE>~~
117         | [ <DCW VALUE LIST> ]

118   <REPEATABLE VALUE> ::= ( <VALUE LIST> )
~~119         | [ <DCW VALUE LIST> ]~~
120         | <LITERAL>
~~121         | <IDENTIFIER>~~
122         | <NIL>

123   <DCW VALUE LIST> ::= <DCW COLOR OPTION> , <DCW CHAR SIZE OPTION> ,
~~123                        <DCW BLINK OPTION> , <DCW INTENSITY OPTION> ,~~
```

```
 123                            <DCW OUTPUT FORMAT OPTION>

 124       <DCW COLOR OPTION> ::= RED
 125            | ORANGE
 126            | YELLOW
 127            | GREEN
 128            | <NIL>

 129       <DCW CHAR SIZE OPTION> ::= 15 MM
 130            | 10 MM
 131            | 7.5 MM
 132            | 5 MM
 133            | <NIL>

 134       <DCW BLINK OPTION> ::= BLINK ON
 135            | BLINK OFF
 136            | <NIL>

 137       <DCW INTENSITY OPTION> ::= 0
 138            | 1
 139            | 2
 140            | 3
 141            | 4
 142            | 5
 143            | 6
 144            | <NIL>

 145       <DCW OUTPUT FORMAT OPTION> ::= TIME
 146            | EBCDIC <DCW FORM>
 147            | <NIL>

 148       <DCW FORM> ::= CHARACTER
 149            | INTEGER
 150            | DECIMAL
 151            | BINARY
 152            | OCTAL
 153            | HEXIDECIMAL

 154       <CLOSING> ::= CLOSE
 155            | CLOSE <LABEL>
 156            | <LABEL> : <CLOSING>

 157       <STATEMENT> ::= EXIT <OPTIONAL LABEL>
 158            | REPEAT <OPTIONAL LABEL>
 159            | GO TO <LABEL>
 160            | RETURN
 161            | RETURN <EXPRESSION>
 162            | SEND <ERROR SPEC>
 163            | OFF <ERROR SPEC>
 164            | ON <ERROR SPEC> <ON ERROR OPTIONS>
 165            | SIGNAL <FLAG LIST>
 166            | RESET <EVENT VAR LIST>
 167            | CANCEL <OPTIONAL LABEL LIST>
 168            | TERMINATE <JOB OR TASK>
 169            | ABORT <JOB OR TASK OR LABEL LIST>
 170            | LOG <EXPRESSION>
 171            | UNLOCK <LABEL LIST>
 172            | LOAD <LABEL>
 173            | INITIATE <LABEL>
 174            | DELETE <JOB OR TASK>
 175            | <CONDITIONAL STMT>
```

~~176 | <ASSIGNMENT STMT>~~
177 | <CALL STMT>
~~178 | <SCHEDULE STMT>~~
179 | <WAIT STMT>
~~180 | <ALERT STMT>~~
181 | <READ STMT>
~~182 | <WRITE STMT>~~
183 | <AVERAGE AI STMT>
~~184 | <READ ERP STMT>~~
185 | <ISSUE STMT>
~~186 | <SET DISCRETE STMT>~~
187 | <APPLY ANALOG STMT>
~~188 | <WTO STMT>~~
189 | <DISPLAY CONTROL STMT>
~~190 | <DISPLAY DATA STMT>~~
191 | <MODIFY VCW STMT>
~~192 | <REQUEST KEYBD STMT>~~
193 | <SELECT STMT>
~~194 | <RELEASE STMT>~~
195 | <CHANNEL CONTROL STMT>
~~196 | <NESTED SCOPE>~~
197 | <NIL>

198 <OPTIONAL LABEL> ::= <LABEL>
~~199 | <NIL>~~

~~200 <ERROR SPEC> ::= ERROR <SUBSCRIPT>~~

~~201 <ON ERROR OPTIONS> ::= SYSTEM <ON~~ ERROR TAIL>
202 | IGNORE <ON ERROR TAIL>
~~203 | <STATEMENT>~~

~~204 <ON ERROR TAIL> ::= AND SIGNAL <FLAG~~ ID>
205 | <NIL>

206 <FLAG LIST> ::= <FLAG ID>
~~207 | <FLAG ID> , <FLAG LIST>~~

~~208 <EVENT VAR LIST> ::= <EVENT VAR>~~
209 | <EVENT VAR> , <EVENT VAR LIST>

210 <OPTIONAL LABEL LIST> ::= <LABEL LIST>
~~211 | <NIL>~~

~~212 <LABEL LIST> ::= <LABEL>~~
213 | <LABEL> , <LABEL LIST>

214 <JOB OR TASK> ::= JOB
~~215 | TASK~~

~~216 <JOB OR TASK OR LABEL LIST> ::=~~ <JOB OR TASK>
217 | <LABEL LIST>

218 <CONDITIONAL STMT> ::= <IF STMT>
~~219 | <IF STMT> ; ELSE <STATEMENT>~~

~~220 <IF STMT> ::= IF <EXPRESSION> THEN~~ <STATEMENT>

~~221 <ASSIGNMENT STMT> ::= <VARIABLE LIST~~> = <EXPRESSION>

~~222 <VARIABLE LIST> ::= <VARIABLE>~~

223 | <VARIABLE> , <VARIABLE LIST>

~~224 <CALL STMT> ::= CALL <LABEL VAR> <ARGUMENT SET>~~

~~225 <ARGUMENT SET> ::= <ARG LIST>~~

226 | <ARG LIST> ASSIGN (<VARIABLE LIST>)

227 <ARG LIST> ::= (<EXPRESSION LIST>)

~~228 | <NIL>~~

~~229 <EXPRESSION LIST> ::= <EXPRESSION>~~

230 | <EXPRESSION> , <EXPRESSION LIST>

231 <SCHEDULE STMT> ::= SCHEDULE <LABEL> <SCHED PARM OPTION>

~~231 <SCHED TIME OPTION> <SCHED EVENT OPTION>~~

231 <SCHED CYCLIC OPTION>

232 <SCHED PARM OPTION> ::= (<EXPRESSION>)

~~233 | <NIL>~~

~~234 <SCHED TIME OPTION> ::= IN <TIME VALUE>~~

235 | AT <CLOCK> <TIME VALUE>

~~236 | <NIL>~~

~~237 <SCHED EVENT OPTION> ::= WHEN <EVENT EXP> <SCHED POST-EVENT TIME OPTION>~~

238 | <NIL>

239 <SCHED POST-EVENT TIME OPTION> ::= THEN WAIT <TIME VALUE>

~~240 | <NIL>~~

~~241 <SCHED CYCLIC OPTION> ::= , REPEAT <SCHED INTERVAL OPTION>~~

242 | <NIL>

243 <SCHED INTERVAL OPTION> ::= AFTER <TIME VALUE>

~~244 | <NIL>~~

~~245 <CLOCK> ::= GMT~~

246 | MET

247 <TIME VALUE> ::= <ARITH VAR>

~~248 | <TIME LITERAL>~~

~~249 <ARITH VAR> ::= <VARIABLE>~~

~~250 <WAIT STMT> ::= WAIT <WAIT FOR OPTION> <WAIT UNTIL OPTION>~~

~~251 <WAIT FOR OPTION> ::= FOR <OPTIONAL CLOCK> <TIME VALUE>~~

252 | <NIL>

253 <OPTIONAL CLOCK> ::= <CLOCK>

~~254 | <NIL>~~

~~255 <WAIT UNTIL OPTION> ::= UNTIL <EVENT EXP> <THEN WAIT OPTION>~~

256 | <NIL>

257 <THEN WAIT OPTION> ::= , THEN WAIT <TIME VALUE>

~~258 | <NIL>~~

~~259 <EVENT EXP> ::= <EVENT TERM>~~

260 | <EVENT TERM> <OR> <EVENT EXP>

```
261       <EVENT TERM> ::= <EVENT FACTOR>
262            | <EVENT FACTOR> <AND> <EVENT TERM>

263       <EVENT FACTOR> ::= <EVENT PRIMARY>
264            | <NOT> <EVENT FACTOR>

265       <EVENT PRIMARY> ::= ( <EVENT EXP> )
266            | <EVENT VAR>

267       <EVENT VAR> ::= <EVENT ID> <SUBSCRIPT'>

268       <AND> ::= AND
269            | &

270       <OR> ::= OR
271            | |

272       <NOT> ::= NOT
273            | ¬

274       <ALERT STMT> ::= ALERT <EVENT VAR> TO <EVENT>

275       <EVENT> ::= TERMINATION OF <LABEL>
276            | <FLAG ID>
277            | <CLOCK> <REL OP> <TIME VALUE>
278            | <DATA BUS EVENT>

279       <DATA BUS EVENT> ::= <ERP DESIGNATOR> <RANGE OR LIMIT>

280       <RANGE OR LIMIT> ::= <REL OP> <QTY OR AEXP>
281            | IS <OPTIONAL NOT> BETWEEN <QTY OR AEXP> AND <QTY OR AEXP>

282       <REL OP> ::= <
283            | =
284            | >
285            | < =
286            | > =
287            | <NOT> =
288            | <NOT> >
289            | <NOT> <

290       <OPTIONAL NOT> ::= NOT
291            | <NIL>

292       <QTY OR AEXP> ::= ( <ARITH EXP> )
293            | <DIM LITERAL>

294       <ARITH EXP> :: <EXPRESSION>

295       <DIM LITERAL> ::= <NUMBER> <DIM>

296       <DIM> ::= VDC
297            | V
298            | KV
299            | MV
300            | UV
301            | A
302            | MA
303            | UA
304            | HZ
305            | KHZ
```

```
306            | MHZ
307            | PPS
~~308            | KPPS~~
309            | DAY
~~310            | HR~~
311            | MIN
~~312            | SEC~~
313            | MSEC
~~314            | USEC~~
315            | OHM
~~316            | KOHM~~
317            | MOHM
~~318            | H~~
319            | MH
~~320            | UH~~
321            | FD
~~322            | UFD~~
323            | PFD
~~324            | W~~
325            | KW
~~326            | MW~~
327            | UW
~~328            | VAR~~
329            | DBW
~~330            | DB~~
331            | KVA
~~332            | KVAR~~
333            | DBM
~~334            | PCT~~
335            | PSIG
~~336            | PSIGA~~
337            | PSI
~~338            | MMHG~~
339            | INHG
~~340            | MB~~
341            | IN
~~342            | FT~~
343            | M
~~344            | KM~~
345            | MM
~~346            | NM~~
347            | FT / SEC
~~348            | M / SEC~~
349            | KT
~~350            | MACH~~
351            | DEG
~~352            | ARCMIN~~
353            | ARCSEC
~~354            | RAD~~
355            | MRAD
~~356            | REV~~
357            | DEGC
~~358            | DEGF~~
359            | M / S / S
~~360            | F / S / S~~
361            | G
~~362            | GPM~~
363            | CFM
~~364            | LB / HR~~

~~365     <READ STMT> ::= <READ STMT TYPE> ( <NUMBER> ) <SEQ IO LIST OPTION>~~
```

366 <READ STMT TYPE> ::= READ
~~367 | READALL~~

~~368 <WRITE STMT> ::= WRITE (<NUMBER>)~~ <SEQ IO LIST OPTION>

~~369 <SEQ IO LIST OPTION> ::= <SEQ IO~~ LIST>
370 | <NIL>

371 <SEQ IO LIST> ::= <SEQ IO ELEMENT>
~~372 | <SEQ IO ELEMENT> , <SEQ IO LIST>~~

~~373 <SEQ IO ELEMENT> ::= <SEQ IO CONTROL>~~
374 | <EXPRESSION>

375 <SEQ IO CONTROL> ::= <SEQ IO CONTROL TYPE> (<EXPRESSION>)

376 <SEQ IO CONTROL TYPE> ::= TAB
~~377 | COLUMN~~
378 | SKIP
~~379 | LINE~~
380 | PAGE

381 <AVERAGE AI STMT> ::= AVERAGE <ARITH EXP> READINGS OF <ERP DESIGNA
~~381 <ASSIGN> <VARIABLE>~~

~~382 <ASSIGN> ::= ASSIGN~~
383 | AND SAVE AS

384 <READ ERP STMT> ::= <READ ERP KEYWORD> <ERP DESIGNATOR LIST>
~~384 <DELTAS OPTION>~~ <ASSIGN> <VARIABLE LIST>

~~385 <READ ERP KEYWORD> ::= MEASURE~~
386 | SENSE

387 <ERP DESIGNATOR LIST> ::= <ERP DESIGNATOR>
~~388 | <ERP DESIGNATOR> , <ERP DESIGNATOR~~ LIST>

~~389 <DELTAS OPTION> ::= DELTAS~~
390 | <NIL>

391 <ISSUE STMT> ::= ISSUE <RTIO VALUE LIST> TO <ERP DESIGNATOR>

392 <RTIO VALUE LIST> ::= <RTIO VALUE>
~~393 | <RTIO VALUE> , <RTIO VALUE LIST>~~

~~394 <RTIO VALUE> ::= <VARIABLE>~~
395 | <BIT LITERAL>
~~396 | <ARITH LITERAL>~~

ORIGINAL PAGE IS
OF POOR QUALITY

~~397 <ARITH LITERAL> ::= <CONSTANT>~~

~~398 <BIT LITERAL> ::= <CONSTANT>~~

~~399 <SET DISCRETE STMT> ::= SET <ERP DESIGNATOR LIST> <TO VALUE OPTION~~
399 <PULSED OPTION>

400 <TO VALUE OPTION> ::= TO <RTIO VALUE LIST>
~~401 | <NIL>~~

~~402 <PULSED OPTION> ::= FOR <TIME VALUE>~~

```
403           | <NIL>

~~404       <APPLY ANALOG STMT> ::= <APPLY KEYWORD> <APPLY VALUE LIST> TO~~
404                                 <ERP DESIGNATOR LIST> <APPLY TAIL>

405       <APPLY KEYWORD> ::= APPLY
~~406           | SEND~~

~~407       <APPLY VALUE LIST> ::= <QTY OR AEXP>~~
408           | <QTY OR AEXP> , <APPLY VALUE LIST>

409       <APPLY TAIL> ::= FOR <TIME VALUE> <RAMPING>
~~410           | <DELTAS OPTION>~~

~~411       <RAMPING> ::= <RAMPING KEYWORD> <QTY OR AEXP>~~
412           | <NIL>

413       <RAMPING KEYWORD> ::= UNTIL
~~414           | RAMPED TO~~

~~415       <WTO STMT> ::= DISPLAY TO OPERATOR <EXPRESSION> <WTO TAIL>~~

~~416       <WTO TAIL> ::= , ACCEPT REPLY IN <CHAR VAR>~~
417           | <NIL>

418       <CHAR VAR> ::= <VARIABLE>

419       <DISPLAY CONTROL STMT> ::= PAGE SELECT <CRT OPTION>
~~420           | CLEAR <CRT OPTION>~~
421           | ALLOCATE <CRT SPEC>
~~422           | DEALLOCATE <CRT SPEC>~~
423           | <VIDEO> <STROKE OPTION> <CRT OPTION>

424       <VIDEO> ::= VIDEO1
~~425           | VIDEO2~~

~~426       <STROKE OPTION> ::= STROKE~~
427           | <NIL>

428       <CRT OPTION> ::= <CRT SPEC>
~~429           | <NIL>~~

~~430       <CRT SPEC> ::= TO < CRT <NUMBER> - <NUMBER> >~~

~~431       <DISPLAY DATA STMT> ::= DISPLAY <IDENTIFIER> <CRT OPTION>~~
432           | DISPLAY <BACKGROUND OR ADDON> <CRT OPTION> <BLINK OPTION>
~~432                                 <DISPLAY FORMAT OPTION> <VECTOR OPTION>~~
433           | UPDATE <IDENTIFIER> <VARIABLE LIST> <CRT OPTION>
~~434           | UPDATE <LABEL>~~

~~435       <BACKGROUND OR ADDON> ::= BGROUND~~
436           | ADDON

437       <BLINK OPTION> ::= BLINK STATUS
~~438           | <NIL>~~

~~439       <DISPLAY FORMAT OPTION> ::= <DISPLAY FORMAT LIST>~~
440           | <NIL>

441       <DISPLAY FORMAT LIST> ::= <DISPLAY FORMAT ENTRY>
~~442           | <DISPLAY FORMAT ENTRY> , <DISPLAY FORMAT LIST>~~
```

```
443      <DISPLAY FORMAT ENTRY> ::= /
444           | <DF LINE OPTION> <DF COL OPTION> <DF DATA OPTION>

445      <DF LINE OPTION> ::= LINE <NUMBER>
446           | <NIL>

447      <DF COL OPTION> ::= COL <NUMBER>
448           | <NIL>

449      <DF DATA OPTION> ::= WIDTH <NUMBER> <DCW VAR> FOR <VARIABLE>
450           | <OPTIONAL DCW VAR> TEXT <CHAR LITERAL>

451      <CHAR LITERAL> ::= <CHAR STRING>
452           | CHAR ( <NUMBER> ) <CHAR STRING>

453      <OPTIONAL DCW VAR> ::= <DCW VAR>
454           | <NIL>

455      <DCW VAR> ::= <VARIABLE>

456      <VECTOR OPTION> ::= <VECTOR LIST>
457           | <NIL>

458      <VECTOR LIST> ::= <VECTOR SPEC>
459           | <VECTOR SPEC> <VECTOR LIST>

460      <VECTOR SPEC> ::= VECTOR FROM ( <NUMBER> , <NUMBER> ) TO ( <NUMBER>
460                             <NUMBER> ) <OPTIONAL DCW VAR>

461      <MODIFY VCW STMT> ::= MODIFY <DCW VAR> = <DCW VALUE LIST>

462      <REQUEST KEYBD STMT> ::= REQUEST ENTRY FROM < KEYBOARD > AND SAVE AS
462                                 <CHAR VAR> <POST OPTION>

463      <POST OPTION> ::= AND POST <BIT VAR>
464           | <NIL>

465      <BIT VAR> ::= <VARIABLE>

466      <SELECT STMT> ::= SELECT <RESOURCE TYPE> <SELECT LIST>

467      <RESOURCE TYPE> ::= DSD
468           | DATA
469           | PROGRAM

470      <SELECT LIST> ::= <SELECT ELEMENT & MODE>
471           | <SELECT ELEMENT & MODE> , <SELECT LIST>

472      <SELECT ELEMENT & MODE> ::= <SELECT ELEMENT> <SELECT MODE>

473      <SELECT ELEMENT> ::= JOB COMMON <LABEL>
474           | SYSTEM COMMON <LABEL>
475           | CHANNEL ( <NUMBER> ) <DS OPTIONS>
476           | FILE ( <NUMBER> ) <DS OPTIONS>

477      <SELECT MODE> ::= EXCLUSIVE
478           | SHARED
479           | <NIL>

480      <DS OPTIONS> ::= <DS ACCESS MODE> , DISP = <DS DISP>
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
481     <DS ACCESS MODE> ::= READ
482          | READ AND WRITE
483          | WRITE
484          | UPDATE

485     <DS DISP> ::= LEAVE
486          | REWIND

487     <RELEASE STMT> ::= RELEASE <RELEASE LIST>

488     <RELEASE LIST> ::= <RELEASE ELEMENT>
489          | <RELEASE ELEMENT> , <RELEASE LIST>

490     <RELEASE ELEMENT> ::= <LABEL>
491          | CHANNEL ( <NUMBER> )
492          | FILE ( <NUMBER> )

493     <CHANNEL CONTROL STMT> ::= CHANNEL ( <NUMBER> ) <CHANNEL CTRL TAIL>

494     <CHANNEL CTRL TAIL> ::= <SPACER> <OPTIONAL AEXP> <FILES>
495          | REWIND
496          | UNLOAD
497          | END FILE
498          | CLOSE FILE

499     <SPACER> ::= SPACE
500          | BACKSPACE

501     <OPTIONAL AEXP> ::= <ARITH EXP>
502          | <NIL>

503     <FILES> ::= FILE
504          | FILES

505     <NESTED SCOPE> ::= <DO BLOCK>
506          | <PROCEDURE BLOCK>
507          | <FUNCTION BLOCK>
508          | <UPDATE BLOCK>
509          | <CRITICAL SECTION>

510     <COMPOUND TAIL> ::= <BLOCK BODY> ; <CLOSING>

511     <DO BLOCK> ::= DO <DO OPTIONS> ; <STATEMENT LIST> ; <ENDING>

512     <PROCEDURE BLOCK> ::= <PROCEDURE HEADER> ; <COMPOUND TAIL>

513     <FUNCTION BLOCK> ::= <FUNCTION HEADER> ; <COMPOUND TAIL>

514     <UPDATE BLOCK> ::= UPDATE ; <COMPOUND TAIL>

515     <CRITICAL SECTION> ::= CRITICAL SECTION ; <COMPOUND TAIL>

516     <ENDING> ::= END <OPTIONAL LABEL>

517     <DO OPTIONS> ::= <DO OPTIONS> <FOR LIST>
518          | <DO OPTIONS> <WHILE CLAUSE>
519          | CASE <EXPRESSION> <OPTIONAL ELSE>
520          | <DO OPTIONS> <OPTIONAL TEMP LIST>
521          | <NIL>
```

```
522     ~~<OPTIONAL ELSE> ::= ; ELSE <STATEMENT>~~
523          | <NIL>

524     <OPTIONAL TEMP LIST> ::= <TEMP DCL LIST>
525          ~~| <NIL>~~

526     ~~<TEMP DCL LIST> ::= <TEMP DCL>~~
527          | <TEMP DCL> ; <TEMP DCL LIST>

528     <FOR LIST> ::= <FOR KEY> <ITERATION>

529     <FOR KEY> ::= FOR TEMPORARY <IDENTIFIER> =
530          ~~| FOR <VARIABLE> =~~

531     ~~<ITERATION> ::= <EXPRESSION>~~
532          | <EXPRESSION> <ITERATION'>

533     <ITERATION'> ::= , <EXPRESSION>
534          ~~| , <EXPRESSION> <ITERATION'>~~
535          | TO <EXPRESSION>
536          ~~| TO <EXPRESSION> <ITERATION'>~~
537          | BY <EXPRESSION>
538          ~~| BY <EXPRESSION> <ITERATION'>~~

539     ~~<WHILE CLAUSE> ::= WHILE <EXPRESSION>~~
540          | UNTIL <EXPRESSION>

541     <EXPRESSION> ::= <OPERAND>
542          ~~| <OPERAND> <OPERATOR>~~ <OPERAND>

543     ~~<OPERATOR> ::= +~~
544          | -
545          ~~| *~~
546          | /
547          ~~| .~~
548          | **
549          ~~| <CAT>~~
550          | <AND>
551          ~~| <OR>~~
552          | <REL OP>
553          ~~| <NIL>~~

554     ~~<CAT> ::= CAT~~
555          | ||

556     <OPERAND> ::= <PRIMARY>
557          ~~| <PRIMARY> <SUBSCRIPT'>~~

558     ~~<SUBSCRIPT'> ::= <SUBSCRIPT>~~
559          | <SUBSCRIPT> <SUBSCRIPT'>

560     <PRIMARY> ::= ( <EXPRESSION> )
561          ~~| + <OPERAND>~~
562          | - <OPERAND>
563          ~~| <NOT> <OPERAND>~~
564          | <LITERAL>
565          ~~| <VARIABLE>~~
566          | <FUNCTION HEAD>
567          ~~| <FUNCTION HEAD> ( <EXPRESSION~~ LIST> )
568          | <INLINE DEF> <COMPOUND TAIL> ;
569          ~~| <ARITH CONVERSION>~~
```

ORIGINAL PAGE IS
OF POOR QUALITY

570 | <BIT CONVERSION>
571 | <CHARACTER CONVERSION>
~~572 | <BIT PSEUDO VAR>~~

573 <ARITH CONVERSION> ::= <ARITH CONV> <SUBSCRIPT'> (<CONV PARM LIST>)

574 <CONV PARM LIST> ::= <CONV PARM ELEMENT> , <CONV PARM LIST>
575 | <CONV PARM ELEMENT>

576 <CONV PARM ELEMENT> ::= <ARITH EXP> # <EXPRESSION>
577 | <EXPRESSION>

578 <ARITH CONV> ::= VECTOR
579 | MATRIX
~~580 | SCALAR~~
581 | INTEGER

582 <BIT CONVERSION> ::= BIT <CONV TAIL>

583 <CHARACTER CONVERSION> ::= CHARACTER <CONV TAIL>

584 <BIT PSEUDO VAR> ::= SUBBIT <CONV TAIL>

585 <CONV TAIL> ::= <SUBSCRIPT'> (<EXPRESSION>)

586 <FUNCTION HEAD> ::= <PREFIX> <FUNCTION ID> <SUBSCRIPT'>

587 <INLINE DEF> ::= FUNCTION <TYPE SPEC> ;

588 <VARIABLE> ::= SUBBIT <SUBSCRIPT'> (<VARIABLE>)
~~589 | FILE (<NUMBER> , <EXPRESSION>)~~
590 | NAME (<NAME VAR>)
~~591 | <EVENT VAR>~~
592 | <PREFIX> <VAR ID> <SUBSCRIPT'>
~~593 | CHARACTER <SUBSCRIPT'>~~ <ARG LIST>
594 | BIT <SUBSCRIPT'> <ARG LIST>

595 <NAME VAR> ::= <VARIABLE>
~~596 | <LABEL VAR>~~

~~597 <LABEL VAR> ::= <VARIABLE>~~

~~598 <PREFIX> ::= <STRUCT ID>~~
599 | <STRUCT ID> . <PREFIX>
~~600 | <NIL>~~

~~601 <SUBSCRIPT> ::= $ <SUBSCRIPT LINE>~~
602 | <NIL>

603 <SUBSCRIPT LINE> ::= <NUMBER>
~~604 | <VARIABLE>~~
605 | (<SUB LIST>)

606 <SUB LIST> ::= <SUB>
~~607 | <SUB> <SUB LIST'>~~

~~608 <SUB LIST'> ::= , <SUB>~~
609 | , <SUB> <SUB LIST'>
~~610 | : <SUB>~~
611 | : <SUB> <SUB LIST'>
~~612 | ; <SUB>~~

```
~~613              | ; <SUB> <SUB LIST'>~~

~~614      <SUB> ::= *~~
615              | @ <PREC SPEC>
~~616              | <EXPRESSION> <SUB TAIL>~~
617              | <SUB EXP> <SUB TAIL>

618      <SUB EXP> ::= #
~~619              | # + <EXPRESSION>~~
620              | # - <EXPRESSION>

621      <SUB TAIL> ::= <AT OR TO> <SUB EXP>
~~622              | <AT OR TO> <EXPRESSION>~~
623              | <NIL>

624      <AT OR TO> ::= AT
~~625              | TO~~

~~626      <RADIX> ::= HEX~~
627              | OCT
~~628              | BIN~~
629              | DEC

630      <LITERAL> ::= NULL
~~631              | ON~~
632              | OFF
~~633              | TRUE~~
634              | FALSE
~~635              | OPEN~~
636              | CLOSED
~~637              | NAME ( NULL )~~
638              | <RADIX> ( <NUMBER> ) <CHAR STRING>
~~639              | <RADIX> <CHAR STRING>~~
640              | <CHAR LITERAL>
~~641              | <NUMBER>~~

~~642      _|_~~
```

(BLANK)

REFERENCES

1. HAL/SM Language Specification, M&S Computing, Inc. Report No. 75-0043, November 21, 1975.

2. HAL/SM System Software Requirement Specification, M&S Computing, Inc., Report No. 75-0045, December 1, 1975.

3. HAL/SM System Design and Implementation Plan, M&S Computing, Inc., Report No. 75-0030, August 7, 1975.