

IBM

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE).

COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FASHION, NOR PLACED IN ANY REPOSITORY.

FOR RESEARCH USE ONLY
COPY PROVIDED BY
WICHITA STATE UNIVERSITY
SPECIAL COLLECTIONS

SPACE SHUTTLE
ORBITER AVIONICS SOFTWARE
NAS 9-14444

INTERFACE CONTROL DOCUMENT: HAL/FCOS

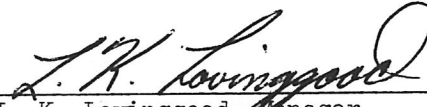
REVISION 5

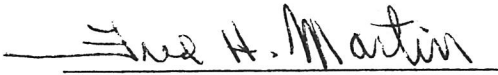
JULY 6, 1976

INTERMETRICS APPROVAL:


D. J. Lickly
Compiler Department Head

IBM APPROVAL:

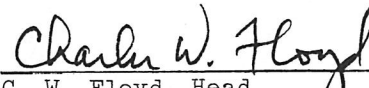

L. K. Lovinggood, Manager
Flight Software



Dr. F. H. Martin
Shuttle Program Manager



D. F. McCabe, Chairman
Software Review Board


D. R. Proctor, Manager
Shuttle Software Development

NASA APPROVAL:


C. W. Floyd, Head
Systems Software Section


J. R. Garman, Deputy Chief
Avionics Software Branch


R. P. Parten, Chief
Spacecraft Software Division

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FASHION, NOR PLACED IN ANY REPOSITORY

IBM FEDERAL SYSTEMS DIVISION, HOUSTON, TEXAS

76-SS-1110
IRD No. 2b8

TYPE I

TABLE OF CONTENTS

- 4. HAL/FCOS
 - 4.1 General Information
 - 4.1.1 Privileged Instructions and the Supervisor State
 - 4.1.2 Register Usage
 - 4.1.3 Memory Organization
 - 4.1.4 Number of CSECT's per Compilation
 - 4.1.5 CSECT Characteristics
 - 4.1.6 Initialization of AUTOMATIC Data
 - 4.1.7 Entry/Exit Conditions for PROGRAMS/TASKS
 - 4.1.8 Linkages
 - 4.2 HAL/S Statements
 - 4.2.1 Real-Time Control Statements
 - 4.2.1.1 SCHEDULE Statement
 - 4.2.1.2 TERMINATE Statement
 - 4.2.1.3 CANCEL Statement
 - 4.2.1.4 WAIT Statement
 - 4.2.1.5 UPDATE Priority Statement
 - 4.2.1.6 SIGNAL/SET/RESET Statement
 - 4.2.1.7 CLOSE Statement
 - 4.2.1.8 Event Expressions
 - 4.2.1.9 Process Directory Entry
 - 4.2.2 EXCLUSIVE PROCEDURES/FUNCTIONS and the UPDATE Block
 - 4.2.2.1 EXCLUSIVE PROCEDURES/FUNCTIONS
 - 4.2.2.2 UPDATE Block
 - 4.2.2.3 Reserve/Release Supervisor Call
 - 4.2.3 Error Control Statements
 - 4.2.3.1 Error Numbering Scheme
 - 4.2.3.2 ON ERROR Statement
 - 4.2.3.3 SEND ERROR Statement
 - 4.2.3.4 HAL/S-FC Defined Errors
 - 4.2.4 HAL/S Functions
 - 4.2.4.1 RUNTIME/CLOCKTIME/DATE Functions
 - 4.2.4.2 PRIO/ERRGRP/ERRNUM Functions

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FASHION, NOR PLACED IN ANY REPOSITORY

4. HAL/FCOS

This section of the document defines the interface between the HAL/S Language and the Flight Computer Operating System (FCOS) for use in the Space Shuttle Onboard Computers. These definitions are in accord with the following documents:

1. General Purpose Computer Specification, MC615-0001
2. HAL/S Language Specification
3. FCOS Detailed Design Specification

Although the interfaces for the following features are defined in this document, they are not supported by FCOS:

1. DEPENDENT option on SCHEDULE statement (p. 4.2.1.1)
2. DEPENDENT option on WAIT statement (p. 4.2.1.4)
3. UPDATE PRIORITY (p. 4.2.1.5)
4. "NOT" operator in event expressions of more than one event variable (p. 4.2.1.8)
5. Mixtures of "OR" and "AND" operators in a single event expression (p. 4.2.1.8)
6. PRIO function (p. 4.2.4.1)
7. NEXTIME function (p. 4.2.4.1)

4.1 GENERAL INFORMATION

4.1.1 Privileged Instructions and the Supervisor State

The compiler will not utilize any privileged instructions nor make assumptions regarding whether the generated programs execute in supervisor mode or not.

4.1.2 Register Use

The flight computer contains two sets of general purpose registers; each set is numbered zero through seven. The compiler shall assume that one set of the registers is available for use by compiler generated programs. However, the compiler will not make any assumptions regarding which set of general registers is available; this decision will be made by the FCOS.

The compiler shall assume that there are eight floating point registers available for use by compiler generated programs. These registers may contain parameters when interfacing with the FCOS. See Section 4.2.1 Real Time Control Statements.

4.1.3 Memory Organization

The compiler shall not make any assumptions concerning in which computer sector the code CSECT's of a program or common subroutine will reside. However, all data except remote data will reside in sector 0 and one other sector.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR

4.1.4 Number of CSECT's Per Compilation

The compiler can generate multiple CSECT's for each compilation. (A CSECT is a relocatable block of AP-101 code). The number and type of CSECT's generated depends on the characteristics of the compilation unit. The generated CSECT's can be grouped into the following categories:

1. Executable code
2. Data areas
3. Program/task directory entries (refer to Section 4.2.1.9). This CSECT must be in sector 0 or the other sector designated for data.
4. 32-bit word for each EXCLUSIVE procedure/function within the compilation. (Refer to Section 4.2.2.1). This CSECT must be sector 0 CSECT.

4.1.5 CSECT Characteristics

The compiler shall make information available such that the linkage editor can generate data concerning type and length of CSECT's. The convention used in naming the various CSECT's will be such that the type of CSECT can be determined by the CSECT name itself.¹ This information is required by the FCOS for storage protection assignment.

4.1.6 Initialization of AUTOMATIC Data

It shall be the responsibility of the compiler to generate the code necessary to initialize data that has been declared with the AUTOMATIC attribute. The FCOS will make no provision for initializing such data on entry to the code block containing its declaration.

4.1.7 Entry/Exit Conditions for PROGRAMS/TASKS

Upon entry to a Program or Task, the FCOS shall provide certain information to the process. This information shall be communicated in the general purpose registers and shall consist of the following:

General Register 0 - the address of a temporary storage area which shall be of sufficient size to satisfy the process needs. This information shall be in bits 0-15.

At the exit point of a program or a task, a CLOSE supervisor call will be executed. (Refer to Section 4.2.1.7.)

¹The CSECT naming conventions used for the currently identified CSECT's are contained in Section 2.6 of the INTERFACE CONTROL DOCUMENT: HAL/SDL.

4.1.8 Linkages

The HAL/S compiler will provide all necessary code for program linkages to internal and external procedures and functions. It will also generate the SVC's as described in Section 4.2 for the HAL/S statements that require FCOS support. The FCOS will contain the code for the initiation and close of a process's execution cycle(s).

4.2 HAL/S STATEMENTS

This section discusses the FCOS/compiler interfaces for the HAL/S statements which require FCOS support. The interface consists of a Supervisor Call (SVC) instruction which has as the address a pointer to a parameter list. (Remote data cannot be passed in an SVC parameter list.) The first halfword of the parameter list contains a number which identifies the type of supervisor call. This number occupies the least significant eight bits of the halfword. The remainder of the parameter list varies with the service being requested; each parameter is one halfword in length unless specifically noted otherwise. Whenever bits are specified as unused or reserved they will be compiled as zero.

The normal return from a supervisor call will be to the location immediately following the SVC instruction. All software or hardware errors detected during execution of a supervisor call will result in processing in accordance with the rules of the ON ERROR statement.

The FCOS will save and restore all general purpose registers and floating point registers across a supervisor call except when parameters are returned to the application program from the SVC routine. (Each of these exceptions is noted under the applicable HAL/S interface.) The FCOS never modifies a supervisor call parameter list nor assumes that the parameter list is still valid after control is returned to the caller.

4.2.1 Real-Time Control Statements

4.2.1.1 SCHEDULE Statement

When a SCHEDULE supervisor call is issued, the following parameters will be passed to the FCOS:

- If required, the following specified floating point registers will contain a double precision time value specified in seconds and fractions of seconds to the nearest millisecond:
 - FPR0, 1 - Floating point registers 0, 1 will contain the AT or IN time value if specified. The AT value will be specified in mission elapsed time; the IN value will be expressed as a delta time value from current mission elapsed time.
 - FPR2, 3 - Floating point registers 2, 3 will contain the REPEAT EVERY or REPEAT AFTER time value if specified. The time values will be expressed as a delta time value from current mission elapsed time.
 - FPR4, 5 - Floating point registers 4, 5 will contain the UNTIL time value if specified. The UNTIL time value will be expressed as mission elapsed time.

- The parameter list will be of the following form:

0	PRIORITY	SVC #
1	FLAGS	
2	PROCESS	
3	EVENT EXPR #1	
4	EVENT EXPR #2	

PRIORITY (8 Bits) PRIORITY value which must be an integer between 1 and 255 where 255 is the highest priority.

SVC # (8 Bits) SVC number which is always 01 for a SCHEDULL request.

FLAGS Meaning Bit Position (Low Order 10 Bits)

Options Flag:	XX	XXXX	XXXX
PROGRAM0
TASK1
Initial Conditions	XX..
-None-	00..
AT	01..
IN	10..
ON	11..
DEPENDENT1.
REPEAT Options	..	XX..
-None-	..	00..
REPEAT	..	01..
REPEAT EVERY	..	10..
REPEAT AFTER	..	11..
CANCEL Options	XX
-None-	00
UNTIL Time	01
WHILE Event	10
UNTIL Event	11

PROCESS (16-Bits) The address of the directory entry for the process (program or task) which is to be scheduled. (See Section 4.2.1.9 for more information about the directory entry).

EVENT EXP #1 (16-Bits) Address of an event expression if the ON option was specified. (Refer to Section 4.2.1.8 for more information about event expressions).

EVENT EXP #2 (16-Bits) Address of an event expression if the WHILE or UNTIL options specified an event expression. (Refer to Section 4.2.1.8 for more information about event expressions).

NOTE: DEPENDENT option is not supported by FCOS.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY FASHION NOR PLACED IN ANY REPOSITORY

4.2.1.2 TERMINATE Statement

When a TERMINATE supervisor call is issued, the following parameter list will be passed to the FCOS:

0	COUNT	SVC #
1	PROCESS #1	
	⋮	
N	PROCESS #N	

COUNT (8 Bits) Count of processes specified in the parameter list. If no label was specified this field need not be initialized.

SVC # (8 Bits) SVC number which is:

02 if no label was specified.
03 if a label was specified.

PROCESS #1 to PROCESS #N (16-Bits) A list of addresses which point to the directory entries of the processes which are to be terminated. (If no label was specified these fields may be omitted. (See Section 4.2.1.9 for more information about the directory entries.)

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION. THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FASHION, NOR PLACED IN ANY REPOSITORY

4.2.1.3 CANCEL Statement

When a CANCEL supervisor call is issued, the following parameter list will be passed to the FCOS:

0	COUNT	SVC #
1	PROCESS #1	
	⋮	
N	PROCESS #N	

COUNT (8 Bits) Count of processes specified in the parameter list. If no label was specified this field does not have to be initialized.

SVC # (8 Bits) SVC number which is: 04 if no label was specified, or 05 if a label was specified.

PROCESS #1 to PROCESS #N (16-Bits) A list of addresses which point to the directory entries for the processes which are to be canceled. If no label was specified these fields may be omitted. (See Section 4.2.1.9 for more information about the directory entries.)

4.2.1.4 WAIT Statement

When a WAIT supervisor call is issued, the following parameters will be passed to the FCOS:

- FPR 0, 1 Floating point registers 0, 1 will contain a double precision time value specified in seconds and fractions of seconds to the nearest millisecond if desired. If the UNTIL option is specified, the time value will be expressed as mission elapsed time, otherwise the value will be expressed as a delta time from the current mission elapsed time. If a time value is not specified in the WAIT statement, floating point registers 0, 1 do not need to be initialized.
- The parameter list will be of the following form:

0	UNUSED	SVC #
1	EVENT EXP	

SVC # (8 Bits) SVC number which is

- 06 if a delta time value was specified.
- 07 if the UNTIL option was specified.
- 08 if an event expression was specified.
- 09 if DEPENDENT was specified.

EVENT EXP (16-Bits) Address of an event expression if the option FOR (event expression) was specified, otherwise the field may be omitted. (Refer to Section 4.2.1.8 for more information about event expressions.)

NOTE: DEPENDENT option is not supported by FCOS.

4.2.1.5 UPDATE PRIORITY Statement

When an UPDATE PRIORITY supervisor call is issued, the following parameter list will be passed to the FCOS:

0	PRIORITY	SVC #
1	PROCESS	

PRIORITY (8 Bits) PRIORITY value if specified. The value must be numeric between 1 and 255 where 255 is the highest priority.

SVC # (8 Bits) SVC number which is:

- 10 if no label was specified.
- 11 if a label was specified.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FASHION, NOR PLACED IN ANY INFORMATION

PROCESS (16-Bits) The address of the directory entry for the process which is to have its priority modified. If no label was specified these fields may be omitted. (See Section 4.2.1.9 for more information about the directory entry.)

NOTE: UPDATE PRIORITY is not supported by FCOS.

When a supervisor call is issued for a SIGNAL, SET or RESET, the following parameter list will be passed to FCOS:

0	UNUSED	SVC #
1	EVENT VARIABLE	

SVC # (8 Bits) SVC number which is:

- 12 if it is a SIGNAL statement.
- 13 if it is a SET statement.
- 14 if it is a RESET statement.

EVENT VARIABLE (16-Bits) Address of an event variable. (Refer to Section 4.2.1.8 for more information about event variables.)

4.2.1.7 CLOSE Statement

When a CLOSE supervisor call is issued for a program or task, the following parameter list will be passed to the FCOS:

0	UNUSED	SVC #
---	--------	-------

SVC # (8 Bits) SVC number for the CLOSE SVC which is 21.

(See Section 4.2.2.1 for a discussion of the CLOSE Statement for EXCLUSIVE PROCEDURES and FUNCTIONS. See Section 4.2.2.2 for a discussion of the CLOSE statement for the UPDATE block.)

4.2.1.8 Event Expression

An event expression consists of up to seven halfwords. The format of the event expression follows:

0	COUNT	OPERATORS
1	OPERATORS	
2	EVENT VARIABLE #1	
6	EVENT VARIABLE #5	

COUNT (4-Bits) Number of operators specified in the OPERATORS field.
Maximum count is 14.

OPERATORS (28-Bits) From the left most position, the halfword contains a set of operators each two bits in length. The operators are defined as follows:

01	logical OR
10	logical NOT
11	logical AND
00	push

Unused operators will be compiled as zero.

The first operator is always an implied "push" and is never included in the operator string or in the count.

EVENT VAR (16-Bits) Address of an event variable.
#1 to EVENT. An event variable is a halfword; bits 0-14 are reserved for FCOS
VAR #5 use and bit 15 designates the value of the event variable as follows:

0	false or off
1	true or on

Bits 0-14 of the event variable will be compiled as zero.

The event expression represented here is encoded in reverse Polish form.

All operators and operands in the source event expression must be present in the encoded expression even if this means repetition of operators and/or operands.

The event variables are placed in the encoded expression in an order which will produce the intended results when operators are sequentially interpreted from left to right.

See paragraph 4, page 1 for FCOS restrictions on event expressions.

4.2.1.9 Process Directory Entry

Based on a user supplied input, the compiler will generate the directory entries for all programs and tasks within a compilation unit. The format of the directory entry follows:

0	PROCESS EVENT
1	(FCOS USE)
2	ENTRY POINT
4	STACK
5	FLAGS

PROCESS
EVENT

(16-Bits) The process event variable for the task or program.

If a process event variable is true then the associated process has been scheduled for execution but is not yet terminated or canceled. If a process event variable is false then the associated process is not in the execution queue.

(FCOS USE)

This halfword is for FCOS use and will be compiled as zero.

ENTRY POINT

(32-Bits) Address of the entry point for this program or task. The address will conform to the format of the proposed 32-bit address constant for the AP-101.

STACK

This halfword is for FCOS use and will be compiled as zero. It will contain either the address of the stack area if pre-allocated or the size of the stack area needed if FCOS is to assign it.

FLAGS

This halfword is for FCOS use and will be compiled as zero. If bit 0 of this field is set, the address of the area is provided in the previous field. Otherwise, the size is provided.

4.2.2 Exclusive PROCEDURES/FUNCTIONS and the UPDATE Block

4.2.2.1 Exclusive PROCEDURES/FUNCTIONS

Procedures and functions which have been defined with the EXCLUSIVE attribute will execute a reserve supervisor call on entry. The execution of the CLOSE statement which closes an EXCLUSIVE procedure or function will issue a release supervisor call.

The compiler will generate one word in the CSECT generated for this purpose for each EXCLUSIVE procedure or function encountered. This word is compiled as zero and reserved for FCOS use. The FCOS uses this word to maintain the status of the EXCLUSIVE procedure or function.

4.2.2.2 UPDATE Block

The execution of an UPDATE block will result in the reserve supervisor call. The execution of the CLOSE statement in an UPDATE block will result in the release supervisor call.

4.2.2.3 Reserve/Release Supervisor Call

The reserve supervisor call reserves a code block (an EXCLUSIVE procedure or function) or data area for the exclusive use of the calling process. The address

of the reserve parameter list points to halfword 0 and is in the format defined on the following page.

The release supervisor call releases a code block or data area which had previously been reserved for the calling process. The address of the release parameter list points to halfword 1 and is defined on the following page.

The format of the reserve/release supervisor call parameter list follows:

0	T Y P	UNUSED	RESERVE SVC #
1	UNUSED		RELEASE SVC #
2	LOCK ID		

TYP (1-Bit) The high order bit will be set to one if the data variables are to be read only. It will be set to zero if the data variables are to be written. If this is a reserve/release supervisor call for an EXCLUSIVE procedure or function, the TYP field will be set to zero.

RESERVE SVC # (8-Bits) SVC number for the reserve supervisor call which is:

15 for a code block.
16 for a data block.

RELEASE SVC # (8-Bits) SVC number for the release supervisor call which is:

17 for a code block.
18 for a data area.

LOCK ID (16-Bits) An identifier indicating which code block or data areas are being used. The identifier for a code block is the address of the word in the Sector 0 CSECT generated for the requested procedure/function.

The identifier for a data area is a bit pattern indicating which lock groups are to be reserved or released. (A lock group is one of the fifteen user-defined sets of locked variables. A locked variable is defined to belong to a specific lock group.) Bit 15 corresponds to lock group one; bit 1 corresponds to lock group fifteen. If the universal lock (all fifteen lock groups) was specified, bits 0-15 will be all ones.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITH WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY FASHION, NOR PLACED IN ANY REPOSITORY

4.2.3 Error Control Statements

4.2.3.1 Error Numbering Scheme

The error numbering scheme categorizes all errors under six error groups:

1. Group 2 for FCOS software defined errors.
2. Group 3 for Program Interrupt errors as defined per AP-101 interrupt code.
3. Group 4 for HAL/S-FC defined errors.
4. Group 5 for other FCOS defined system errors.
5. Group 6 for user errors.

The error number format consists of a halfword where bits 0-7 define the error group and bits 8-15 define the error code corresponding to the error condition.

Under error group 3, three error conditions are maskable:

1. HEX '0304' - fixed point arithmetic overflow.
2. HEX '0309' - floating point exponent underflow.
3. HEX '0305' - significance.

4.2.3.2 ON ERROR Statement

When an ON ERROR or OFF ERROR statement is compiled one or both of the following sequences of code will be generated by the compiler:

1. If the specified error is maskable in the PSW, the compiler will generate a set program mask (SPM) instruction to modify the current PSW accordingly:
 - a) If IGNORE is specified, the bit in the program mask associated with the error condition is set to zero.
 - b) If any other option is specified, the bit in the program mask associated with the error condition is set to one.
 - c) If OFF ERROR affects maskable interrupts, the program mask will be changed to the inherited program mask contained in the return address field in the stack.
 - d) ON ERROR for group designation does not apply to maskable interrupts (program mask will not be updated).

In issuing the SPM instruction, the compiler will not alter the current settings of the other bits in the program mask. To insure this, the inherited program mask setting will be maintained in the stack area of the process. The initial setting will be placed in the stack area by FCOS in the PSW field of the first stack frame (first word of stack). The DSR and BSR bits in this field will also be valid.

Refer to Section 4.2.3.1 for identification of the maskable interrupts.

2. When the error recovery action requested can not be satisfied by a set program mask instruction, the compiler will generate code to process the request and to store appropriate information in the stack

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER, NOR PLACED IN ANY REPOSITORY.

concerning the error environment. OFF ERROR processing in this case will consist of zeroing the group number of the appropriate error table entry.

The stack area of each process (program task) will be divided into blocks called stack frames, each of which corresponds to a block of code (e.g., any UPDATE block or EXCLUSIVE procedure executed within the main procedure has a stack frame associated with it). Each stack frame contains pointers to the stack frames of its calling block. (This field is zero in the initial block.) The layout of a stack frame appears in Figure 4.2.3.2a.

Each stack frame also contains a pointer to a local data area (which contains additional information concerning the stack frame) for the current code block. This pointer is also zeroed when the area does not exist. The layout of a local data area appears in Figure 4.2.3.2b.

The error environment of a process is maintained in these stack frames in the form of error table entries which are two half-words in length. The error table entries correspond to the ON ERROR statements issued and are placed in the stack frame of the code block issuing the ON ERROR statement. The layout of an error table entry appears in Figure 4.2.3.2c.

Refer to Figure 4.2.3.2d for an overview of the stack organization.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER, NOR PLACED IN ANY REPOSITORY

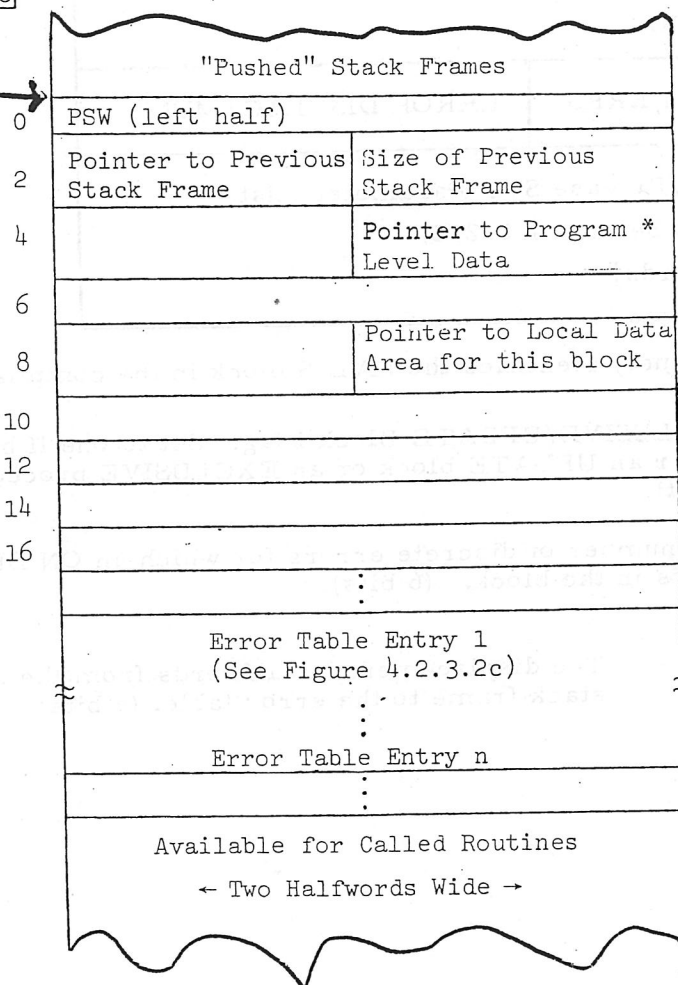
STACK LAYOUT

Register 0:

Address of Current Stack Frame	Size of Current Stack Frame
--------------------------------------	-----------------------------------

Stack Size (minimum = 18 halfwords)

Error Displacement
contained in local
data area (See
Figure 4.2.3.2b)



*For HAL
blocks only

Figure 4.2.3.2a

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)

COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER IN ANY REPOSITORY

LOCAL DATA AREA

0	BLOCK ID		
1	XU	ONERRS	ERROR DISPLACEMENT
2	Reserve/Release SVC Parameter List (Refer to Section 4. 2. 2. 3) [3 halfwords]		

BLOCK ID - Uniquely identifies the HAL/S block in the compilation unit.

XU - EXCLUSIVE/UPDATE Block Flag. Set to one if block is either an UPDATE block or an EXCLUSIVE procedure/function. (1 bit).

ONERRS - The number of discrete errors for which an ON ERROR statement exists in the block. (6 bits).

ERROR DISPLACEMENT - The displacement in halfwords from the beginning of the stack frame to the error table. (9 bits).

Figure 4. 2. 3. 2 b

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FASHION, NOR PLACED IN ANY REPOSITORY

ERROR TABLE ENTRY

0	ACTION CODE	ERROR CODE	ERROR GROUP
1	ADDRESS		

- ACTION CODE - (4 Bits):
- 0000 GO TO address
 - XX01 - System
 - XX11 - IGNORE
 - 00XX - No Event Action
 - 01XX - SET
 - 10XX - RESET
 - 11XX - SIGNAL
- ERROR CODE - (6 Bits) Error Code (within an error group).
If error code = 63 then all numbers within the group are implied. (Will be last entry for this group).
- ERROR GROUP - (6 bits) Error Group.
If Error Group = 63 then all groups are implied. (Will be last entry in error table).
If Error Group = 0 then null entry.
- ADDRESS - (16 bits) Address of GO TO location or event variable depending on the action code.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE).
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY FASHION, NOR PLACED IN ANY REPOSITORY

Figure 4.2.3.2 c

STACK ORGANIZATION

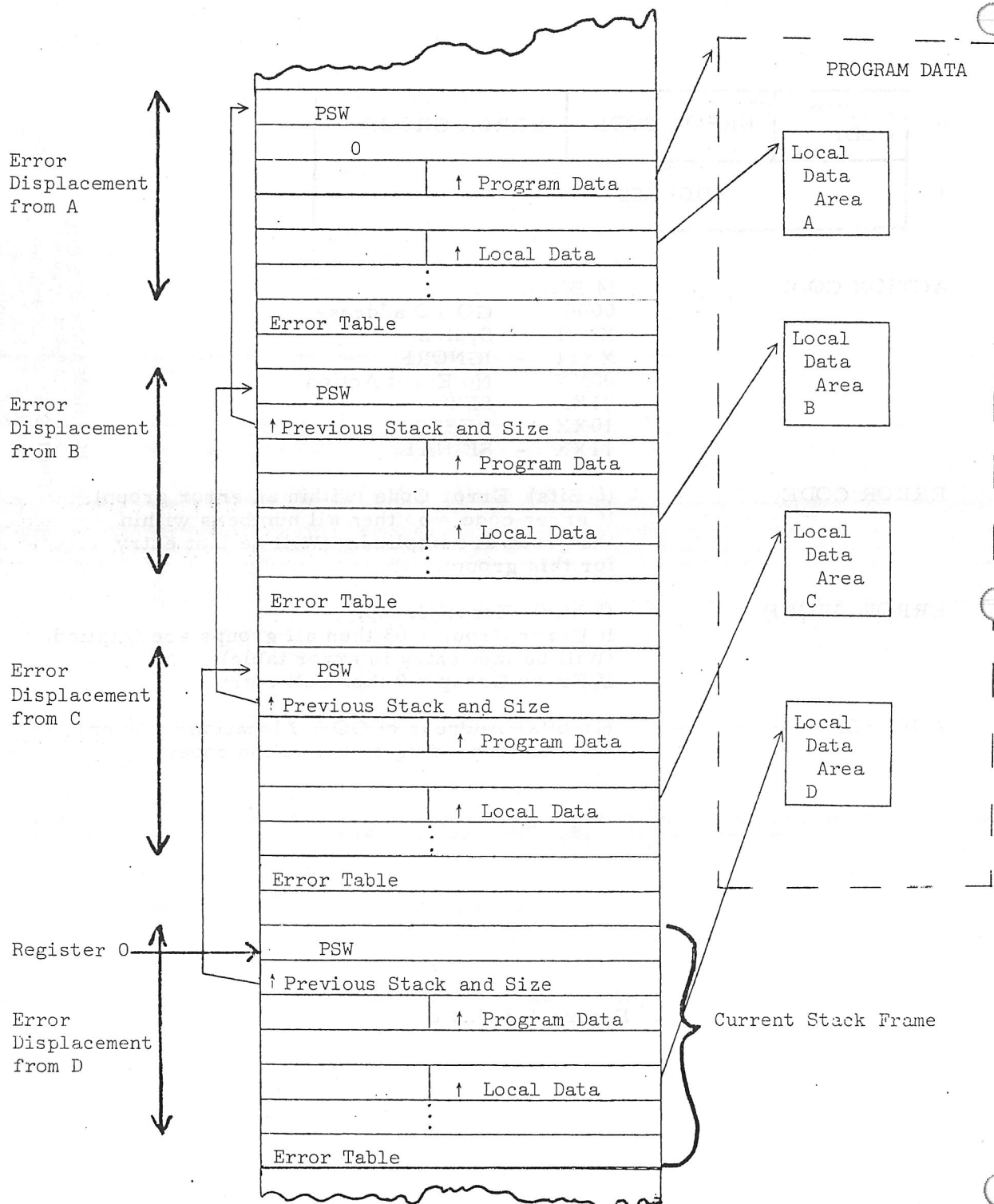


Figure 4.2.3.2d

4.2.3.3 SEND ERROR Statement

When a supervisor call is issued for a SEND ERROR statement the following parameter list will be passed to FCOS:

0	UNUSED	SVC #
1	ERROR GROUP	ERROR NUMBER

SVC # (8-Bits) SVC number which is 20 for SEND ERROR statement.

ERROR GROUP (8-Bits) The group number corresponding to the error condition which occurred.

ERROR NUMBER (8-Bits) The number corresponding to the error condition which occurred.

When processing a SEND ERROR statement or processing any error which occurred during application execution, FCOS will search the error environment of the process beginning with the stack frame of the current code block (pointed to by application general purpose register 0 at all times) and proceeding backward. Within a stack frame, FCOS will start at the beginning of the error table and process according to the first applicable error table entry.

When the action to be taken is to go to a specified location and the location is in a different code block than that in which the error condition occurred, the following processing will be done by FCOS:

- Application general purpose register 0 will be updated to contain the address of the stack frame containing the error table entry for the error condition.
- Application register 1 will be updated to reflect the address of the program data area.
- Application register 3 will be updated to reflect the address of the local data area of the code block containing the specified location.
- If going to the specified location results in leaving an UPDATE or EXCLUSIVE block, the reserved resources will be released.
- If a sector other than the current code sector is being addressed by the specified location, the Base sector register of the PSW will be updated.
- The program mask field of the PSW is updated with the program mask field that was in force for the code block containing the ON ERROR.

4.2.3.4 HAL/S-FC Defined Errors

Table 4.2.3.4 lists HAL/S-FC defined errors.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FASHION, NOR PLACED IN ANY REPOSITORY

Error Number	Message	Explanation	Standard Fixup
4	EXPONENTIATION OF ZERO TO POWER ≤ 0	a negative or zero power was specified	the result is set to zero
5	SQUARE ROOT HAS ARG < 0		the result is the square root of the absolute value of the argument
6	EXP FUNCTION HAS ARG > 174.673		the result is set to maximum value representative on the machine
7	LOG FUNCTION HAS ARG ≤ 0		if the argument was zero then the result is set to maximum representable nega- tive value, else it is set to the log of the absolute value of the arg.
8	SIN OR COS FUNCTION HAS $\left \text{ARG} \right \begin{cases} 2.621\text{E}5 \\ 1.126\text{E}15 \end{cases} \text{ PI}$	the two figures are for single and double precision arguments respectively	the result is set to $\frac{\sqrt{2}}{2}$
9	SINH OR COSH FUNCTION HAS ARG $> 175,366$		the result is set to the maximum value representative
10	ARCSIN OR ARCCOS FUN- CTION HAS $\left \text{ARG} \right > 1$		the result is set to zero

Table 4.2.3.4 HAL/S-FC Defined Errors

-18-

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE).
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FASHION, NOR PLACED IN ANY REPOSITORY

Error Number	Message	Explanation	Standard Fixup
11	TAN FUNCTION HAS ARG > $\begin{cases} 2.621E5 \\ 1.126E15 \end{cases}$ PI	the two figures are for single and double precision respectively	the result is set to one
12	TAN FUNCTION TOO CLOSE TO SINGULARITY	the argument is too close to an odd multiple of $\pi/2$	The result is set to the maximum representable value
13	CASE VARIABLE OUT OF RANGE	The value of the case variable is either < 1 or greater than the number of cases and there was no ELSE clause	the do case statement is ignored
14	CLOSE REACHED ON FUNCTION	no return statement was encountered prior to reaching the close of the function	none: IGNORE not allowed
15	SCALAR TOO LARGE FOR INTEGER CONVERSION		the result is set to the maximum representable value
17	ILLEGAL CHARACTER SUBSCRIPT	Character component subscribing out-of-bounds	The out-of-bounds subscript(s) set to first or last character
18	BAD LENGTH IN LJUST OR RJUST	The length is less than the string length	Truncation to the specified length occurs on the left (RJUST) or right (LJUST)

Table 4.2.3.4 HAL/S-FC Defined Errors (cont.)

Error Number	Message	Explanation	Standard Fixup
19	MOD DOMAIN ERROR	In $A \bmod B$ $B=0$ and $A < 0$	Returns A (negative)
20	CHARACTER TO SCALAR CONVERSION	The string was not in standard internal for- mat for integers or scalars.	The result is zero.
22	CHARACTER TO IN- TEGER CONVERSION	The string was not in standard internal for- mat for integers.	The result is zero.
24	NEGATIVE BASE IN EXPONENTIATION	$A^{**}B$ where $ A < 0$	The result is $ A ^{**}B$
25	VECTOR/MATRIX DI- VISION BY ZERO		The result is the original vector/mat- rix.
27	ARG OF INVERSE IS SINGULAR		The result is the identity matrix.
28	ARG OF UNIT FUNCTION IS NULL VECTOR	Every component of the vector was zero in vaule.	The result is a vector all of whose compo- nents is zero.
29	ILLEGAL BIT STRING		Returns zero.
30	Illegal SUBBIT sub- script	value of routine SUBBIT subscript exceeded bit length	Subscript takes value of exceeded limit.

Table 4.2.3.4 HAL/S-FC Defined Errors (cont.)

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE).
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FASHION, NOR PLACED IN ANY REPOSITORY

Error Number	Message	Explanation	Standard Fixup
31	BIT@OCT - INVALID character		Return 0
32	BIT@HEX - INVALID character		Return 0
59	ARCCOSH function has argument < 1		Return 0
60	ARCTANH function has $ ARG \geq 1$		Return 0
61	ARCSINH arg < -4080.6704		Return -1

Table 4.2.3.4 HAL/S-FC Defined Errors (cont.)

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE).
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FASHION, NOR PLACED IN ANY REPOSITORY

4.2.4 HAL/S Functions

4.2.4.1 RUNTIME/CLOCKTIME/DATE Functions

When a supervisor call is issued for a RUNTIME, CLOCKTIME, or DATE function, the following parameter list will be passed to the FCOS:

0	TYPE	SVC #
1	PROCESS	

TYPE (8-Bits) The type of request. The three possible types are as follows:

- 0 RUNTIME request
- 1 CLOCKTIME request
- 2 DATE request
- 3 NEXTIME request

SVC # (8-Bits) SVC number which is 22.

PROCESS (16-Bits) The address of the directory entry for the process whose associated next time information is being requested. If type 3 request is not indicated, this field may be omitted.

The requested information will be returned as follows:

- o For the RUNTIME function, the FCOS internal Greenwich Mean Time (in seconds) will be returned in floating point registers 0, 1. RUNTIME is not identical for all participating GPC's in a redundant set.
- o For the CLOCKTIME function, the Greenwich Mean Time (in seconds) of the last expired FCOS timer queue element will be returned in floating point registers 0, 1. CLOCKTIME is identical for all participating GPC's in a redundant set.
- o For the DATE function, the current date will be returned in general purpose register 5. Bits 0-15 will contain the year and bits 16-31 will contain the day of the year.
- o FCOS will not support the NEXTIME function.

4.2.4.2 PRIO/ERRGRP/ERRNUM Functions

When a supervisor call is issued for a PRIO, ERRGRP, or ERRNUM function, the following parameter list will be passed to the FCOS:

0	TYPE	SVC #
---	------	-------

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FASHION NOR PLACED IN ANY REPOSITORY

TYPE

(8-Bits) The type of the request. The three possible types are as follows:

- 1 ERRGRP request
- 2 ERRNUM request
- 3 PRIO request

SVC #

(8-Bits) SVC number which is 23.

The requested information will be returned as follows:

- For the PRIO function, the priority of the current process will be returned in bits 0-15 of general purpose register 5.
- For the ERRGRP function, the error group number for the latest error condition which occurred during execution of the current process will be returned in bits 0-15 of general purpose register 5.
- For the ERRNUM function, the error code for the last condition which occurred will be returned in bits 0-15 of general purpose register 5.

NOTE: The PRIO function will not be supported by FCOS.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM.