

HL

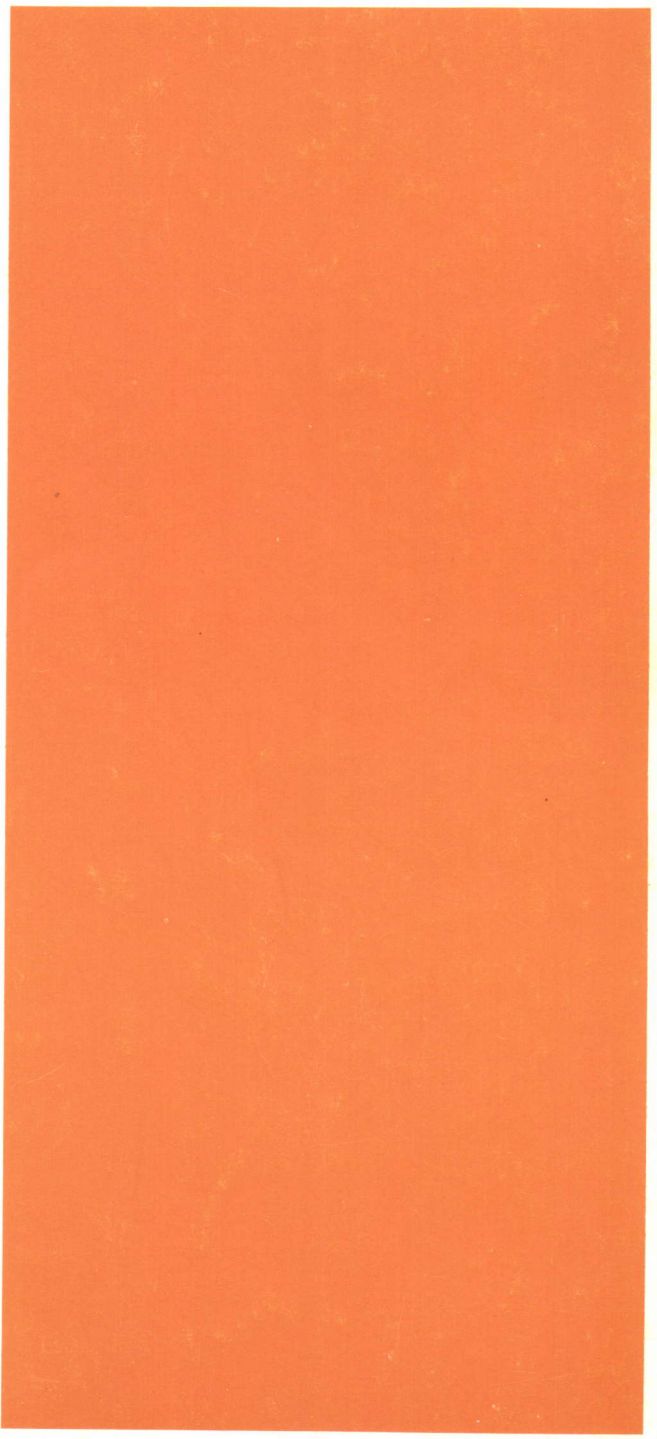
Honeywell Bull

OPERATOR'S GUIDE

SYSTEM 700

OS/700

SOFTWARE



SYSTEM 700

OS/700

SUBJECT:

Operating Procedures and Utility Programs for OS/700.

SPECIAL INSTRUCTIONS:

This manual supersedes the edition dated March, 1974. It has been extensively reorganized and rewritten; therefore, change bars or asterisks are not used to indicate new or changed information.

SOFTWARE SUPPORTED:

This manual supports Release 0310 of OS/700. See the preface of the OS/700 System Generation manual (Order Number AK35) for information as to later releases supported by this publication.

DATE:

July 1975

Printed in France

Ref : 61A.2-AG14,Rev.3

PREFACE

This manual describes operating procedures for loading and running OS/700, for executing user-programs under the operating system, and for executing OS/700 program development tools.

Section I defines terms used within this and other OS/700 manuals.

Section II describes procedures for using the control panel and the console.

Section III describes operating procedures for loading and initializing core and disk operating systems, running programs under core and disk operating systems, and communicating with OS/700.

Section IV describes operating procedures for executing the OS/700 language processors used for program development.

Appendices A through K contain the following information:

Appendix A - Error halts that can occur during system initialization.

Appendix B - Summary of system message formats (for messages issued to the operator) and pointers to documentation of each type of message.

Appendix C - System error messages.

Appendix D - Abort messages that can occur in systems having system integrity.

Appendix E - Executive function call error codes.

Appendix F - Physical I/O device information for all peripheral devices.

Appendix G - Possible data mode assignments for physical I/O requests.

Appendix H - Octal to decimal conversion table.

Appendix J - Table of positive and negative powers of 2.

Appendix K - List of ASCII and Hollerith card codes.

This document is issued for information only. Specifications, data and information may change after the date of printing. Latest specifications, data and information are available upon request and will be the subject of subsequent releases issued from time to time.

PREFACE (cont)

The following symbology is used in this manual:

- Octal numbers are designated by an apostrophe preceding the digits; e.g., '57 is octal 57.
- The symbol (CR) in parentheses represents the carriage RETURN key (to be) pressed by the operator on the console.
- The symbol (LF) in parentheses represents the LINE FEED key (to be) pressed by the operator on the console.
- The symbol (K) underlined and in parentheses represents the control character K on the operator's console; i.e., K typed with the control key held down.
- The symbol (P) underlined and in parentheses represents the control character P on the operator's console; i.e., P typed with the control key held down.
- A mandatory blank space is indicated by a delta (Δ).
- Arrowhead brackets < > are used to enclose and signify parameters.
- A comma is used to separate parameters and to indicate omission of an optional parameter in a command string.
- Square brackets [] (sometimes placed outside arrowhead brackets) indicate that the entry is optional.
- Braces { } enclose a group or set of alternative entries, one of which is to be selected.
- The parameter <name> is represented by and is equal to six ASCII characters.
- All uppercase characters are used in reserved words or symbols that must be written exactly as shown. They are normally used to express utility verbs or mnemonics.
- Lowercase characters are used to represent variable symbolic names or values.

Each section/appendix of this document is structured according to the heading hierarchy shown below. Each heading indicates the relative level of the text that follows it.

<u>Level</u>	<u>Heading Format</u>
1. (highest)	<u>ALL CAPITAL LETTERS, UNDERLINED</u>
2	<u>Initial Capital Letters, Underlined</u>
3	ALL CAPITAL LETTERS, NOT UNDERLINED
4	Initial Capital Letters, Not Underlined
5 (lowest)	ALL CAPITAL LETTERS FOLLOWED BY COLON: Text begins on the same line.

The reader should be familiar with the following documents:
System 700 Programmers' Reference Manual (Order Number AC72)
System 700 Stand-Alone Utility Programs (Order Number AR21)
OS/700 Executive (Order Number AR22)
OS/700 Online Utility Programs (Order Number AR20)

PREFACE (cont)

This document supports Release 0310 of OS/700. See the Preface of the OS/700 System Generation manual (Order Number AK35) for information as to later releases supported by this publication.

CONTENTS

	Page
Section I	Operating Terminology 1-1
Section II	Control Panel and Console Usage 2-1
	Control Panel Usage 2-1
	Control Panel Operating Procedures 2-6
	Turning on the Processor 2-6
	Turning off the Processor 2-7
	Initializing the Processor 2-7
	Displaying Registers 2-7
	Controlling Registers 2-9
	Memory Access 2-9
	Reading Memory Locations 2-10
	Writing in Memory Locations 2-10
	Error Recovery on Writing in Memory 2-11
	Program Execution 2-11
	Run Mode 2-12
	Single Instruction Mode 2-12
	Halt Compare Mode 2-12
	Program Loops 2-13
	64K Access 2-13
	Access to Normal Mode Devices 2-14
	Console Usage 2-14
	Console Components 2-14
	Procedures for Using the Console 2-17
	Turning on the Console 2-17
	Entering Messages 2-17
	Special Characters 2-18
Section III	Operating Procedures 3-1
	Definitions 3-1
	Loading and Initializing OS/700 3-2
	COS Loading and Initialization 3-2
	DOS Loading and Initialization 3-3
	Errors During Loading and Initialization 3-5
	Running Programs Under OS/700 3-6
	Loading and Starting Activities 3-7
	COS Activities 3-7
	Blank COS Activity Areas 3-7
	Permanent COS Activity Areas 3-8
	Temporary COS Activity Area 3-8
	DOS Activities 3-10
	DOS Activity Areas for Disk-Resident Activities 3-11
	Permanent DOS Activity Area 3-11
	Using Activities 3-12
	Command Files (DOS Only) 3-13
	Terminating Activities 3-14
	Honeywell-Supplied Activities 3-14
	Operator Interface With OS/700 3-15
	Formats 3-15
	System Commands 3-18
	Abort Activity Command (\$AB) 3-19
	\$AB Format 3-19
	\$AB Action 3-19
	\$AB Messages 3-19
	Command Input Mode Command (\$CI) 3-20

CONTENTS (cont)

		Page
Section III (cont)	\$CI Format	3-20
	\$CI Action	3-20
	\$CI Messages	3-21
	Comment Command (\$CO)	3-22
	\$CO Format	3-22
	\$CO Action	3-22
	\$CO Messages	3-22
	Load Activity Command (\$LA)	3-23
	\$LA Format	3-23
	\$LA Action	3-23
	\$LA Messages	3-24
	Output Device Command (\$OD)	3-25
	\$OD Format	3-25
	\$OD Action	3-25
	\$OD Messages	3-25
	Start Activity Command (\$SA)	3-26
	\$SA Format	3-26
	\$SA Action	3-26
	\$SA Messages	3-27
	Terminate CI Mode Command (\$TR)	3-28
	\$TR Format	3-28
	\$TR Action	3-28
	\$TR Messages	3-28
	Disk Allocation Messages (DOS Only)	3-29
	Volume Name Requests	3-29
	Mount Messages	3-30
Section IV	Language Processors	4-1
	Program Development	4-1
	EDIT-700 Text Editor Procedures	4-4
	Text Editor Execution	4-5
	Text Editor Commands	4-8
	Text Editor Error Messages	4-10
	FORTRAN-700 Procedures	4-11
	FORTRAN Translator Execution	4-12
	FORTRAN Translator Options	4-13
	FORTRAN Translator Error Messages	4-13
	DAP-700 Macro Assembler Procedures	4-15
	Assembler Execution	4-16
	DAP-700 Macro Assembler Commands	4-17
	DAP-700 Macro Assembler Diagnostic Messages	4-18
	LINKED-700 Linkage Editor Procedures	4-19
	Linkage Editor Execution	4-20
	Linkage Editor Commands	4-21
	Linkage Editor Error Messages	4-24
Appendix A	Initialization Error Halts	A-1
Appendix B	System Message Formats	B-1
Appendix C	System Error Messages	C-1
	I/O Device Errors	C-1
	Executive Errors	C-4
	Communications Supervisor Messages	C-6
	Event Report (CSEVRT)	C-6
	Configuration Errors (CSCNFE)	C-8
	Status Word Formats	C-8
Appendix D	Activity Abort Messages	D-1
Appendix E	Executive Function Call Error Codes	E-1

CONTENTS (cont)

		Page
Appendix F	Physical I/O Device Information	F-1
	Teleprinter (Type 5310 KSR-33)	F-1
	Cartridge Disk Subsystem (Type 476x)	F-3
	High-Speed Paper Tape Reader (Type 5010)	F-6
	High-Speed Paper Tape Punch (Type 5210)	F-7
	Card Reader (Types 5100, 5121-5123, 5151-5153, 5161-5164)	F-8
	Card Punch (Type 5176)	F-8
	Card Reader/Punch (Types 5140 and 5172)	F-8
	Status Information for Card Readers (Types 512x) and Card Reader/Punch (Type 5140) ...	F-8
	Status Information for Card Reader Type 5100	F-10
	Status Information for Card Devices (Types 5151-5153, 5161-5164, 5172, and 5176)	F-11
	Fixed-Head Disk Subsystem (Type 451x)	F-13
	Removable Disk Subsystems (Types 471x, 472x, 473x, 474x, 475x, and 478x)	F-14
	Line Printers (Types 5520, 5524, 5526, 5527, 5529, 5541-2, 5551-2, 5565-9)	F-17
	Status Information for Line Printer Types 552x	F-18
	Status Information for Line Printer Types 554x, 555x, and 556x	F-18
	Status Information for Line Printer Types 556x	F-19
	7-Track Magnetic Tape Subsystems (Types 402x and 404x)	F-20
	9-Track Magnetic Tape Subsystems (Types 405x, 415x, 418x, and 419x)	F-20
	Status Information for Magnetic Tapes (Types 4041 and 4051)	F-22
	Cassette Tape Subsystem (Type 5400)	F-24
	Teleprinters (Keyboard/Printer) (Type 5507 ASR-35 and Type 5307 ASR-33)	F-27
	Teleprinters (Reader/Punch) (Type 5507 ASR-35 and Type 5307 ASR-33)	F-28
Appendix G	Physical I/O Data Mode Assignments	G-1
Appendix H	FORTTRAN Run-Time Error Messages	H-1
Appendix J	Octal/Decimal Conversion Table	J-1
Appendix K	Powers-of-2 Table	K-1
Appendix L	ASCII/Hollerith Card Code Table	L-1

ILLUSTRATIONS

Figure 2-1.	Stand-Alone Operating Mode	2-2
Figure 2-2.	Online Operating Mode	2-3
Figure 2-3.	Type 716 Central Processor Control Panel	2-4
Figure 2-4.	Display Lights Bit Switches	2-7
Figure 2-5.	Example of Display Lights	2-9
Figure 2-6.	ASR/KSR Consoles	2-15
Figure 3-1.	COS Activities	3-9
Figure 3-2.	DOS Activities	3-10
Figure 4-1.	OS/700 Program Development	4-3
Figure 4-2.	Text Editor Flow	4-4
Figure 4-3.	FORTTRAN Translator Flow	4-11
Figure 4-4.	DAP-700 Macro Assembler Flow Diagram	4-16

ILLUSTRATIONS (cont)

		Page
Figure 4-5.	Linkage Editor Flow	4-20
Figure C-1.	Device Status Word	C-8
Figure C-2.	Line Status Word	C-9
Figure C-3.	Terminal Status Word	C-9
Figure C-4.	System Status Word	C-10
Figure C-5.	Device/Line/Terminal Number Word	C-10
Figure C-6.	Parameter 1 of Line Alarm Message	C-11
Figure C-7.	Parameter 1 of Discipline Failure Message	C-11
Figure C-8.	Parameter 1 of Line Poll Failure, Line Select Failure-Output, Line Select Failure-Input Messages	C-11
Figure C-9.	Parameter 1 of VIP Status Message	C-12

TABLES

Table 2-1.	Control Panel Components	2-4
Table 2-2.	Status Display	2-8
Table 2-3.	Octal Bit Representation	2-8
Table 2-4.	Console Functions	2-16
Table 3-1.	Supplied Activities	3-6
Table 3-2.	System Command Restrictions in a DOS With Command Input Mode Configured	3-18
Table 3-3.	\$LA Messages	3-24
Table 3-4.	\$SA Operator Messages (COS Only)	3-27
Table 4-1.	Text Editor Command Summary	4-8
Table 4-2.	Text Editor Error Messages	4-10
Table 4-3.	FORTTRAN-700 Translator Options	4-13
Table 4-4.	FORTTRAN Translator Error Messages	4-14
Table 4-5.	Assembler Command Summary	4-18
Table 4-6.	DAP-700 Macro Assembler Diagnostic Messages	4-19
Table 4-7.	Linkage Editor Commands	4-22
Table 4-8.	Linkage Editor Error Messages	4-25
Table C-1.	I/O Error Codes and Meanings	C-2
Table C-2.	Executive Error Codes and Meanings	C-4
Table C-3.	Communications Supervisor Message Codes	C-7
Table D-1.	Reason for Abort	D-1
Table D-2.	Permissible Functions	D-2
Table D-3.	Nonpermissible Functions	D-3
Table E-1.	Executive Function Call Error Codes	E-1
Table J-1.	Octal/Decimal Conversion Table	J-2
Table L-1.	ASCII/Hollerith Codes	L-1

SECTION I

OPERATING TERMINOLOGY

This manual contains the information necessary for an operator to load and run Operating System/700 (OS/700). It includes descriptions of:

- Loading the system
- Running programs under the system
- Interpreting, issuing, and responding to system messages
- Recovering from and recording errors
- Using language processors and program development

In addition to this manual, the operator should also be familiar with the following documents:

- OS/700 Online Utility Programs manual
- System 700 Stand-Alone Utility Program manual

The operator is assumed to be familiar with the peripheral devices and their use. If he is not, he should refer to hardware programming manuals for the individual devices.

The following definitions are useful in understanding this manual:

Stand-alone - An operating mode in which programs are self-contained; i.e., they do not run under supervisory control of OS/700.

Stand-alone utilities - Honeywell-supplied programs, such as loaders, disk support utilities, and a debugging aid, that run in stand-alone mode. See the System 700 Stand-Alone Utility Programs manual.

Online - An operating mode in which programs run under supervisory control of OS/700.

Online utilities - Honeywell-supplied programs, consisting of system utilities and magnetic tape utilities, which run in online mode. See the OS/700 Online Utility Programs manual.

Control panel - Panel mounted on the 716 mainframe that gives the operator direct access to the central processor's registers and to memory, and permits the operator to set initial conditions, to start program execution, to control program flow with sense switches, and to examine the status of the central processor.

Console - The Automatic Send-Receive (ASR) or Keyboard Send-Receive (KSR) teleprinter device used by OS/700 and online programs to exchange messages with the operator. If the system contains more than one teleprinter device, only one is designated as the console (or operator's console).

SECTION II

CONTROL PANEL AND CONSOLE USAGE

Operation of the system is accomplished by using the control panel and operators console. Use of this equipment differs between online and stand-alone operating modes. Figures 2-1 and 2-2 show control panel and console usage in the stand-alone and online operating modes, respectively. Bracketed items indicate steps that are not always necessary.

CONTROL PANEL USAGE

Primary access to the Type 716 Central Processor is through the control panel. (See Figure 2-3.) Procedures for using the control panel, e.g., to turn the processor on and off, to display and set registers, to read and alter contents of memory locations, and to start program execution are described below. Functions of each component on the panel are listed in Table 2-1.

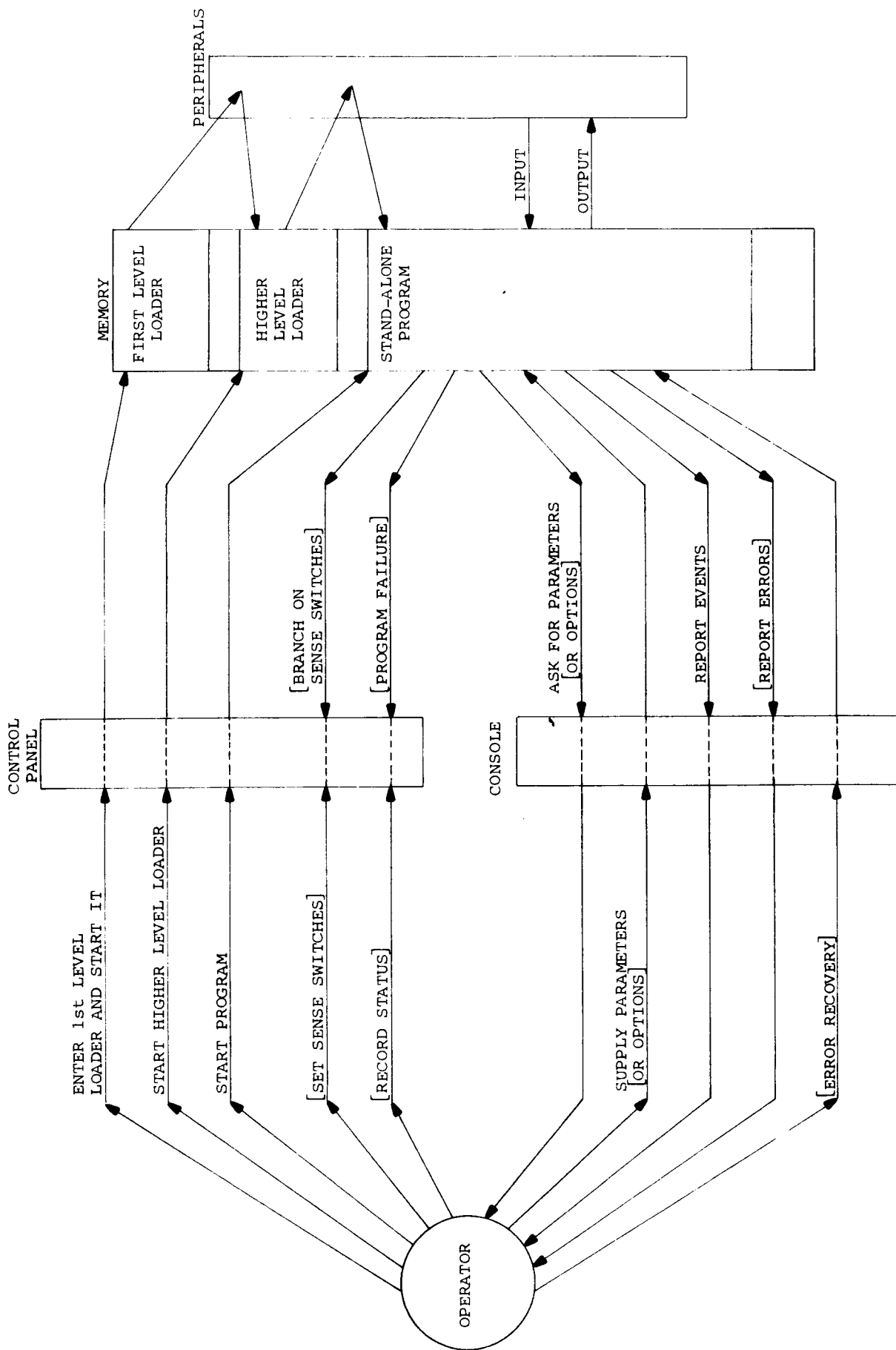
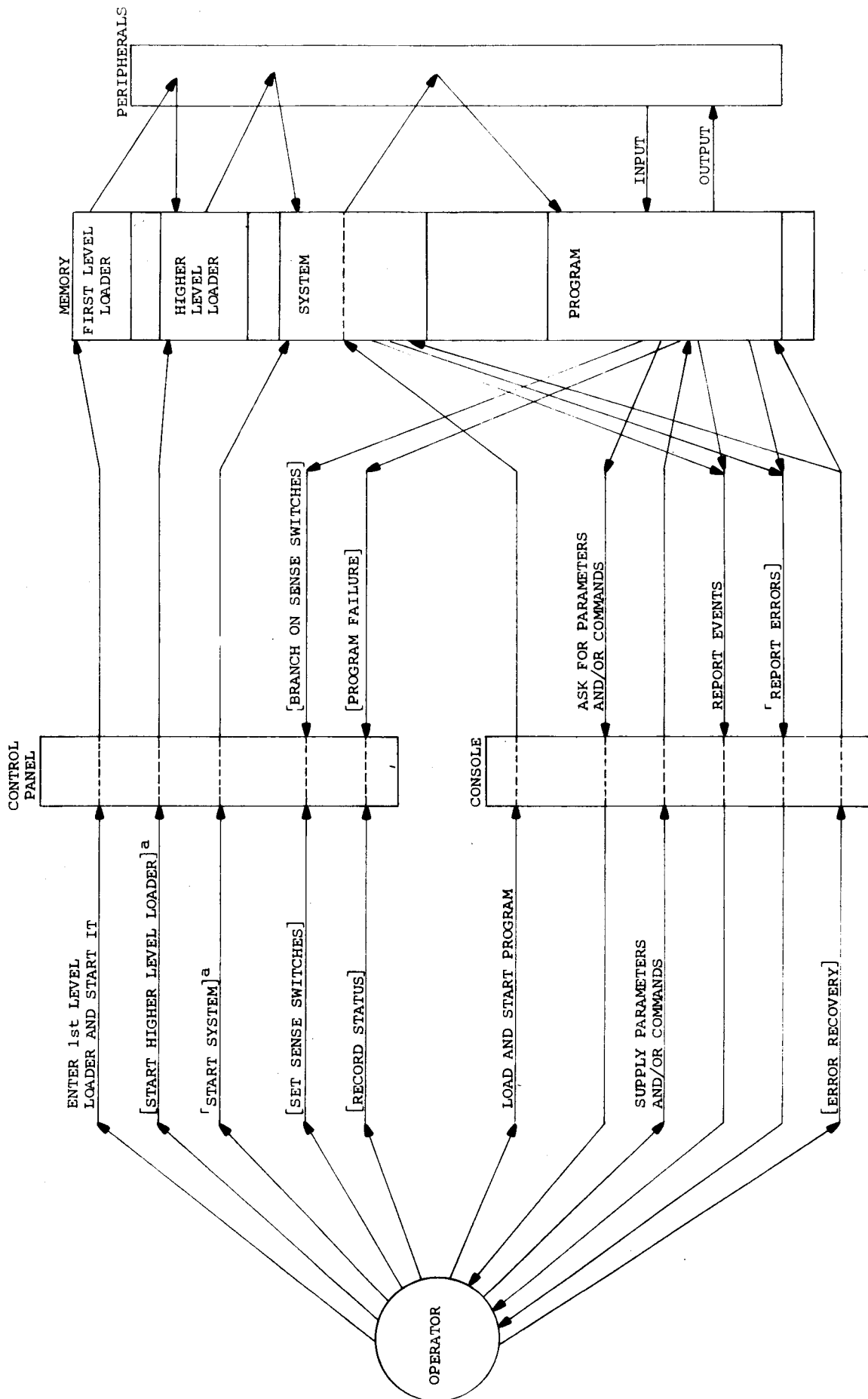


Figure 2-1. Stand-Alone Operating Mode



^aThis step is automatic for some types of systems.

Figure 2-2. Online Operating Mode

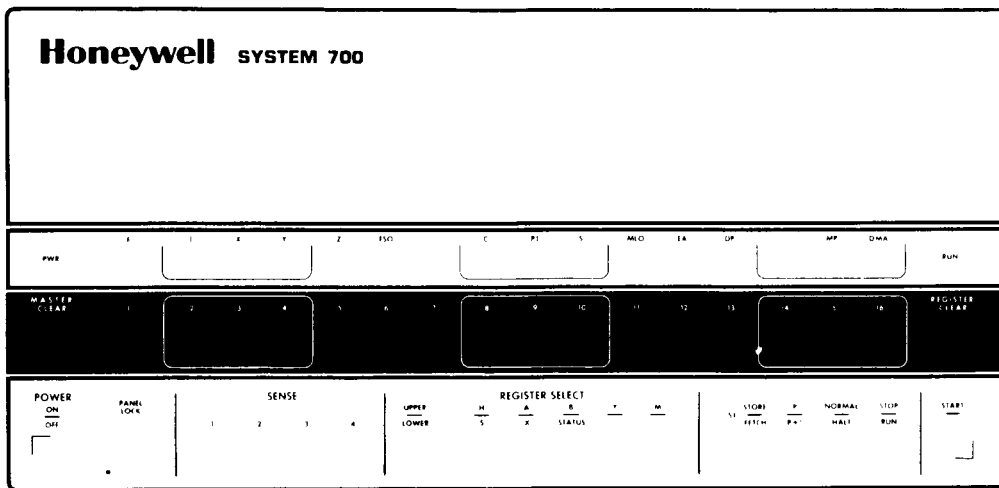


Figure 2-3. Type 716 Central Processor Control Panel

Table 2-1. Control Panel Components

Switch or Indicator	Row/Location	Element Type	Function
PWR	Top	Indicator	Lights when the central processor is on.
F through DMA	Top	Indicators	Displays the contents of selected registers or the status of certain indicator lists. See REGISTER SELECT (in this table) and Table 2-2.
RUN	Top	Indicator	Lights when the central processor is running.
MASTER CLEAR	Middle	Momentary switch	Restores the central processor to the standard cleared state. Registers A, B, H, M, P, and Y are cleared, the clock is stopped, and all timing registers are set to the same conditions that follow the execution of a HLT Instruction. MASTER CLEAR also initializes peripheral devices and mainframe options. (If neither register X nor S is selected, indicators F through DMA should not be lit after a MASTER CLEAR. If they are lit, repeat MASTER CLEAR.)
1 through 16	Middle	Momentary switches	Set the bits of selected register (A, B, H, M, P, or Y).
REGISTER CLEAR	Middle	Momentary switch	Clears the selected register.
POWER ON/OFF	Bottom	Two-position switch	Turns power on and off.

Table 2-1 (cont). Control Panel Components

Switch or Indicator	Row/Location	Element Type	Function
PANEL LOCK	Bottom	Two-position key-activated switch	When this is in the locked position, all switches other than the register select switches and POWER ON/OFF become inoperable, to prevent unwanted alteration of memory or register contents. For example, if the processor is in RUN mode when the panel is locked, it remains in RUN mode even if the STOP/RUN switch is set to STOP.
SENSE 1 SENSE 2 SENSE 3 SENSE 4	Bottom	Two-position switches	In conjunction with skip instructions, these provide a means whereby a program can be selected and/or altered during execution. A sense switch is set when in the down position.
UPPER/ LOWER	Bottom	Two-position switches	Determines whether the five adjacent register select switches are effective in the upper or lower position.
REGISTER SELECT H, A, B, Y, M, S, X, STATUS	Bottom	Mechanically interlocked pushbutton selectors	<p>Select registers for manually changing and/or displaying contents:</p> <ul style="list-style-type: none"> H - 16 bits - Halt address when running in halt compare mode. (See "Program Execution" below.) A - 16 bits - Primary accumulator used for all arithmetic and shift operations and for transferring data to and from memory. B - 16-bits - Secondary accumulator used for double-precision arithmetic and some shift operations. Y - 15 bits - Current memory location. M - 16 bits - Buffer register between memory and central processor. S - 16 bits - Stack pointer used for automatic increment and decrement instructions (see <u>Programmer's Reference Manual, Section II</u>). Also used as a secondary index register. X - 16 bits - Primary index register and a secondary register for transferring data to and from memory. STATUS - 16 bits - Status of certain key indicators in central processor. See Table 2-2.

Table 2-1 (cont). Control Panel Components

Switch or Indicator	Row/Location	Element Type	Function
STORE/SI/FETCH	Bottom	Three-position switch	<p>STORE Position: In conjunction with the STOP/RUN and P/P+1 switches, this changes the content of a single memory location or of successive memory locations. See "Memory Access" below.</p> <p>SI Position: In conjunction with the STOP/RUN switch, this permits the execution of a single instruction or of successive instructions. See "Program Execution" below.</p> <p>FETCH Position: In conjunction with the STOP/RUN and P/P+1 switches, this allows the contents of a single memory location or of successive memory locations to be read. See "Memory Access" below.</p>
P/P+1	Bottom	Two-position switch	<p>P Position: This allows access to a specific memory location when the central processor is in the memory access mode.</p> <p>P+1 Position: This allows access to consecutive memory locations when the central processor is in the memory access mode. See "Memory Access" below.</p>
NORMAL/HALT	Bottom	Two-position switch	Used in halt compare, mode. See "Program Execution" below.
STOP/RUN	Bottom	Momentary switch	<p>STOP Position: terminates processing and permits memory access. See "Memory Access" below.</p> <p>RUN Position: Permits normal program operation; disables MASTER CLEAR, bit switches 1 through 16, and REGISTER CLEAR.</p>
START	Bottom	Two-position switch	Used in various operating procedures explained later in this section.

Control Panel Operating Procedures

TURNING ON THE PROCESSOR

1. Insert the key into PANEL LOCK; turn it clockwise half way around.
2. Set POWER ON/OFF to ON.

When the CPU is cycling up, it automatically performs the MASTER CLEAR function. This prevents unpredictable central processor and peripheral operations and provides a standard operational starting point.

TURNING OFF THE PROCESSOR

1. Set STOP/RUN to STOP.
2. Set POWER ON/OFF to OFF.

INITIALIZING THE PROCESSOR

To initialize the processor, press MASTER CLEAR to restore the system to the standard cleared state. This action clears registers A, B, H, M, P, and Y, stops the clock, and sets all timing registers to the conditions that would exist after execution of a HLT instruction. Pressing MASTER CLEAR also initializes peripheral devices and mainframe options.

MASTER CLEAR has no effect on memory or on the X- or S-registers. It is disabled in the RUN mode.

DISPLAYING REGISTERS

The register select switches, in conjunction with UPPER/LOWER, permit the operator to display the contents of one of eight registers. (See Table 2-1). The procedure for displaying registers is as follows:

1. Set UPPER/LOWER appropriately:
 - to UPPER for H, A, B, Y or M
 - to LOWER for S, X, or STATUS
2. Press the appropriate register select button.

The bits of the selected register are displayed on the data indicator light in the top row of the control panel. (See Figure 2-4). The bits of the STATUS register are explained in Table 2-2. The display lights indicate whether or not the bits in the selected register are set in the following manner:

Light on - Bit is set (or equal to 1)

Light off - Bit is reset (or equal to 0)

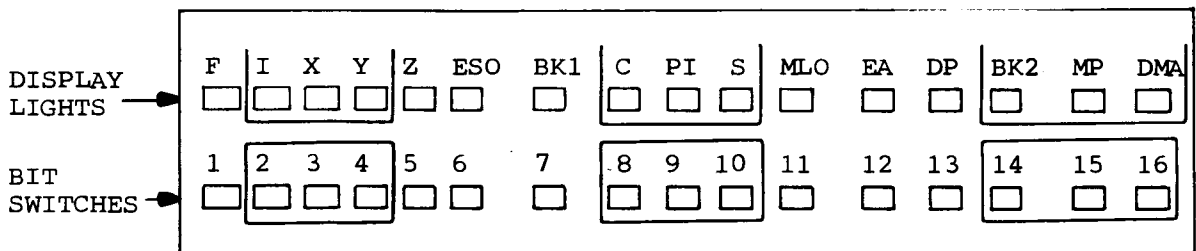


Figure 2-4. Display Lights Bit Switches

Table 2-2. Status Display

Bit Displayed	Meaning
1	Fetch cycle (F)
2	Indirect cycle (I)
3	X cycle (X)
4	Y cycle (Y)
5	Z cycle (Z)
6	Stack overflow enabled (ESO)
7	Bank register (BK1) ^a
8	C-bit (C)
9	Permit interrupt (PI)
10	Index from stack (S)
11	Memory lockout restricted mode enabled (ML0) ^a
12	Previous mode indicator, extended addressing (EA)
13	Double-precision mode (DP) ^a
14	Bank register (BK2) ^a
15	Memory-parity (MP) ^a
16	DMA breaks in process (DMA)
^a Applies only to central processors that have this option.	

The display lights and bit switches are divided in groups of three except for bit 1. Each group represents an octal digit as shown in Table 2-3.

Table 2-3. Octal Bit Representation

If the first bit is:	It represents:	If a group of three bits is:	It represents:
0	'0	000	'0
1	'1	001	'1
		010	'2
		011	'3
		100	'4
		101	'5
		110	'6
		111	'7
NOTE: An octal number is written with an apostrophe.			

Example:

The A-register is selected. A darkened square in Figure 2-5 below represents a display light ON. The A-register contains '143625.

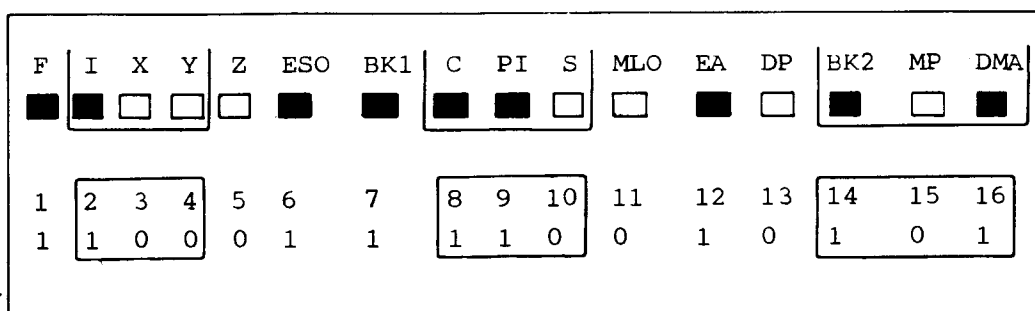


Figure 2-5. Example of Display Lights

CONTROLLING REGISTERS

Registers H, A, B, Y, and M can be cleared or set to any desired octal number:

1. Set RUN/STOP to STOP (not necessary for H-register).
2. Set UPPER/LOWER appropriately.
3. Press the appropriate REGISTER SELECT button.
4. Press REGISTER CLEAR.
5. If the register is to be set (not cleared), press the bit switches necessary to set the bits that represent the desired contents of the register. (For the example above, bit switches 1, 2, 6, 7, 8, 9, 12, 14, and 16 would be pressed.)

The X- and S-registers can be cleared or set to any desired octal number.

1. Set RUN/STOP to STOP.
2. Set UPPER/LOWER to UPPER.
3. Select the M-register.
4. Set M to '010000 to clear or set the X-register;
Set M to '010003 to clear or set the S-register.
5. Select the A-register.
6. Press REGISTER CLEAR.
7. If the register (X or S) is to be set (not cleared), set the A-register to the value X or S is to contain.
8. Set STORE/SI/FETCH to SI.
9. Press START.

Memory Access

The content of any memory location can be read and/or altered by use of the control panel and the following procedures.

READING MEMORY LOCATIONS

1. Unlock the control panel.
2. Set STOP/RUN to STOP.
3. Set STORE/SI/FETCH to FETCH.
4. To access one location:
Set P/P+1 to P.
To access consecutive locations:
Set P/P+1 to P+1.
5. Set UPPER/LOWER to UPPER.
6. Select the Y-register.
7. Press REGISTER CLEAR.
8. To access one location:
Set Y to the location address.
To access locations:
Set Y to the address of the first location.
9. Select the M-register.
10. Press START. The contents of the specified location are displayed in the M-register. To access consecutive locations, continue to press START. Each time START is pressed, the next consecutive memory location is displayed in the M-register.

NOTE: The address of the memory location whose contents are currently in the M-register can be displayed at any time by selecting the Y-register.

WRITING IN MEMORY LOCATIONS

1. Turn on the central processor.
2. Set STOP/RUN to STOP.
3. Set STORE/SI/FETCH to STORE.
4. To write in one location:
Set P/P+1 to P.
To write in consecutive locations:
Set P/P+1 to P+1.
5. Set UPPER/LOWER to UPPER.
6. Select the Y-register.
7. Press REGISTER CLEAR.
8. To write in one location:
Set Y to the location address.
To write in consecutive locations:
Set Y to the address of the first location.
9. Select the M-register.
10. Press REGISTER CLEAR.
11. Set the M-register to the desired contents of the memory location whose address is currently in the Y-register. On successive executions of this step, if the P/P+1 switch is set to P+1, set the M-register to the desired contents of the memory location whose address is Y+1; i.e., set the M-register to the contents of the next memory location.

12. Press START. The contents of the M-register are moved to the memory location specified in the Y-register.

NOTE: If the P/P+1 switch is set to P+1 on consecutive executions of this step, the Y-register is incremented by 1 when START is pressed, and the contents of the M-register are stored at the location of the incremented Y-register.

13. To write in consecutive locations repeat steps 9, 10, 11, and 12.

NOTE: To display the address of the memory location into which the contents of the M-register were just written, select the Y-register.

ERROR RECOVERY ON WRITING IN MEMORY

If an error is made in setting the M-register but START has not been pressed:

1. Press REGISTER CLEAR.
2. Set the M-register to the proper value.
3. Continue at step 12 above.

If an error is made in setting the M-register while writing in only one location (P/P+1 switch set to P) and START has already been pressed:

1. Press REGISTER CLEAR.
2. Set the M-register to the proper value.
3. Continue at step 12 above.

If an error is made in setting the M-register while writing in consecutive locations (P/P+1 switch set to P+1) and START has already been pressed:

1. Set P/P+1 to P.
2. Press REGISTER CLEAR.
3. Set the M-register to the proper value.
4. Press START. The contents of the M-register will be stored in the memory location whose address is in the Y-register.
5. Set P/P+1 to P+1.
6. Continue at step 13 above.

Program Execution

Programs can be executed in one of three modes.

Run mode - Instructions are successively fetched and executed until a HLT instruction occurs.

Single Instruction mode - Instructions are executed one at a time.

Halt Compare mode - Programs are executed up to a predetermined instruction or until a particular memory location is referenced.

RUN MODE

1. Set STOP/RUN to STOP.
2. Press MASTER CLEAR (not necessary but usually desirable).
3. Set desired initial values in A-, B-, and Y-registers by using the procedure outlined in "Controlling Registers" above.
4. Set STOP/RUN to RUN.
5. Press START. The program will now run until a HLT instruction is executed or STOP/RUN is returned to STOP.

If the program tests sense switches, the operator can control its execution by setting or resetting any of the sense switches while in Run mode.

SINGLE INSTRUCTION MODE

1. Set STOP/RUN to STOP.
2. Set STORE/SI/FETCH to SI.
3. Press MASTER CLEAR (not necessary but usually desirable).
4. Set desired initial values in the A-, B-, and Y-registers by using procedure outlined in "Controlling Registers" above.
5. Press START. This fetches the first instruction from memory and places it in the M-register.
6. Read instruction if desired, by displaying M-register.
7. Press START. The instruction in the M-register is executed, the next instruction is fetched, the Y-register is incremented, and the central processor stops. (If the instruction executed is a SKIP or COMPARE AND SKIP, the content of the Y-register may be incremented by more than 1, if the instruction executed is a JUMP, the Y-register is set to the address of the location jumped to.)
8. Read new instruction and/or contents of registers, if desired, by displaying appropriate registers.
9. Repeat steps 7 and 8 for all instructions that are to be executed or read.

The RUN program mode can be started from any step by setting STOP/RUN to RUN and pressing START.

HALT COMPARE MODE

1. Set NORMAL/HLT to NORMAL.
2. Set address where central processor is to be halted in H-register. (See procedure for setting this register in "Controlling Registers" above.)
3. Set NORMAL/HLT to HALT.
4. Execute program in accordance with "Run Mode" above. The program will run until the address in the H-register is accessed by the program or by DMA/DMC.
5. If the program does not access the designated address, specify different address, without stopping the program, by repeating steps 1, 2, and 3.

PROGRAM LOOPS

If the program is not executing as expected and does not halt, it may be caught in a closed loop. The loop can be traced in "Single Instruction Mode" (see above) displaying the instructions (M-register), the memory address (Y-register), or other register contents as desired.

The program may also be in an indirect loop, if a single instruction causes it, and bit 2 of the STATUS register is set.

NOTE: The timing is changed by going into Single Instruction mode and may cause the loop to disappear.

64K Access

The contents of memory locations in the upper 32K may be read or altered and programs residing in the upper 32K may be executed using the following procedures to set the banks register.

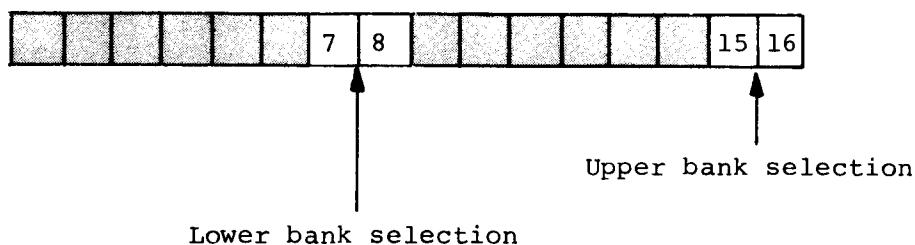
1. Using the procedure for writing in memory locations above, set three locations as follows:

```
SMK '720 output banks
JMP *+1 jump to next location
HLT halt
```

For example, set locations '100 to '102 in sector 0 as follows:

<u>Location</u>	<u>Contents</u>
'100	'170720
'101	'003102
'102	'000000

2. Set the A-register to the desired banks:



Any desired combination of banks in the A-register is acceptable. For example, to select bank 2 as the lower bank and bank 3 as the upper bank, set the A-register as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1

Lower bank Upper bank

or the A-register is set to '001003.

3. Set the Y-register to the address of the SMK instruction ('170720).
4. Set RUN/STOP to RUN.
5. Press START.

The banks register is now set. Memory locations and programs in the selected banks may be accessed/executed using the procedures earlier in this section and setting the Y-register to addresses from:

'0 to '37777 For locations or programs in the lower bank, or
'40000 to '7777 For locations or programs in the upper bank.

Access to Normal Mode Devices

The identifier code of any normal mode (non-DMC) device may be input to the A-register using the following procedure.

1. Clear the A-register.
2. Set the M-register to '1311DD, where DD is the hardware address.
3. Press START. The ID of any normal mode device at address DD is transferred to the A-register.
4. Select the A-register to display the ID.

CONSOLE USAGE

There are three consoles available (see Figure 2-6):

ASR-33 Type 5307 Teleprinters

ASR-35 Type 5507 Teleprinters

KSR-33 Type 5310 Teleprinters

All consoles consist of a keyboard and printer. The two ASR's include a paper tape reader/punch capability; the KSR does not.

Console Components

The console components used by OS/700 software are summarized in Table 2-4.

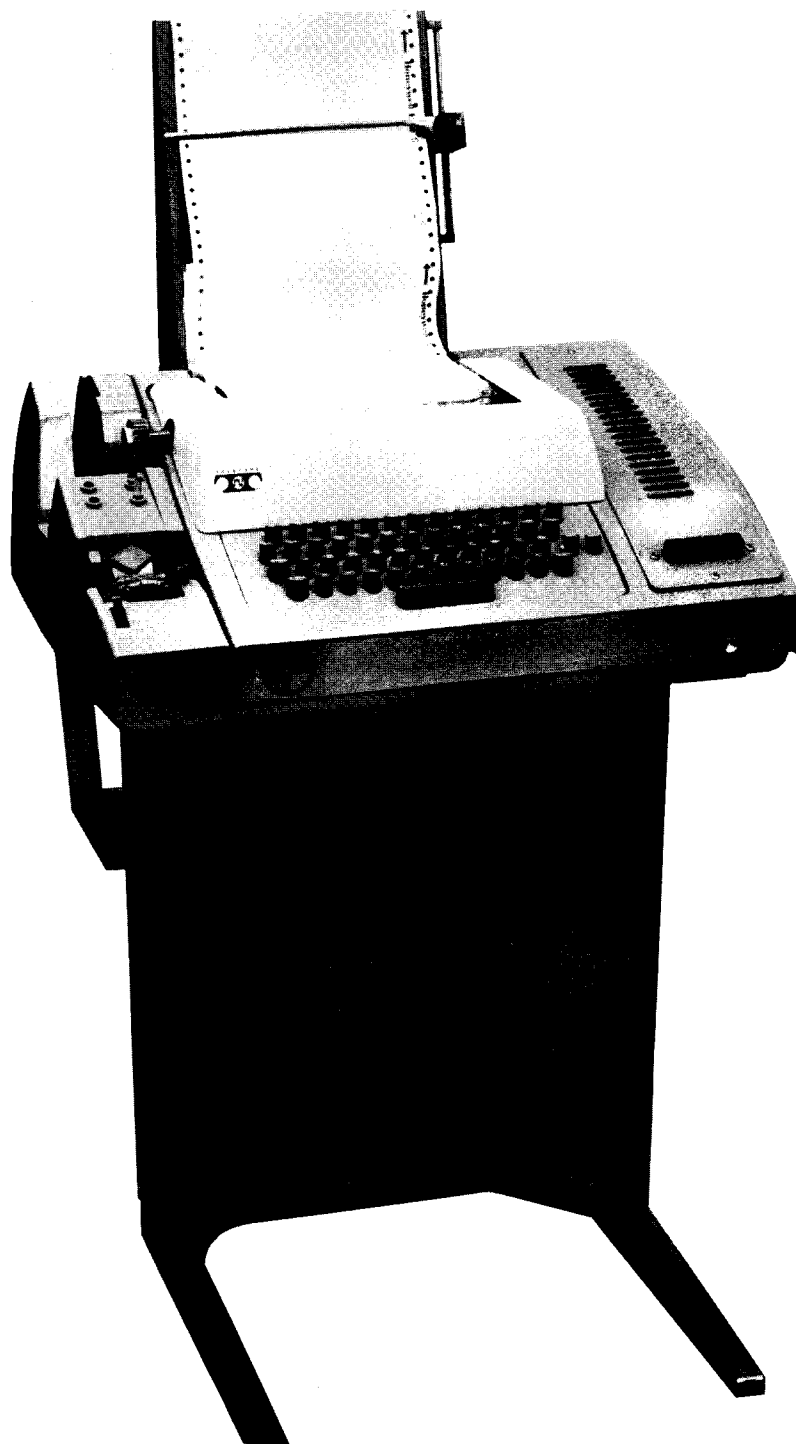


Figure 2-6 ASR-33

Table 2-4. Console Functions

Function	Description	ASR-33	ASR-35	KSR-33
BREAK	Restarts console - Used only for consoles that shut off automatically when unused for a period of time (hardware option).	X	X	X
CTRL	Does not generate any code by itself but allows some of the keys to generate nonprintable function characters.	X	X	X
LINE/OFF/LOCAL	LINE - Device is enabled to receive data from and transmit data to the central processor in full duplex mode. OFF - Device is disabled. LOCAL - Device is offline in half duplex mode and may be used to generate or reproduce paper tape.	X	X	X
LOC CR	Local carriage return - Does not transmit a character to the processor.		X	
LOC LF	Local line feed - Does not transmit a character to the processor.		X	
MODE (K, KT, T, TTS, TTR.)	K - Keyboard and printer are online; card punch and card reader are disabled. KT - Keyboard, printer, card reader and card punch are online. T - Card reader and card punch are online; keyboard and printer are online. Not used by OS/700. TTS - Unused. TTR - Unused.		X	
REPT	When pressed simultaneously with any other key, the other key's action is repeated as long as both keys are pressed.	X	X	X
RETURN	Returns the carriage to the left margin but does not advance the paper vertically.	X	X	X
SHIFT	Does not generate any code by itself, but it causes uppercase characters to be generated.	X	X	X
NOTE: An X indicates the presence of the function on the appropriate teleprinter.				

Procedures for Using the Console

TURNING ON THE CONSOLE

1. Turn LINE/OFF/LOCAL to LINE.
2. On the ASR-35, set MODE to KT.

ENTERING MESSAGES

The operator uses the keyboard to send messages to the system or to stand-alone programs. However, hitting a key sends a character to the central processor only if a character has been requested. If there is no program in memory expecting a character (or whole message), nothing happens when the key is hit. All OS/700 programs send (or echo) the characters received to the printer where they are printed or cause special action (for example, a carriage return).

When a stand-alone utility program needs information from the operator, it issues a message on the printer soliciting a response and then waits for input. When the operator types a response, each character is setn to the central processor, echoed by the system, and then passed to the program when the terminating character of the operator's response is recieved.

NOTE: A carriage return is usually the terminating character; see specific documentation for exceptions.

OS/700 issues messages on the operator's console soliciting input from the operator but does not wait for input. Also, an operator running the system needs to enter unsolicited messages (system commands) on the console which control program execution. Therefore, on the console, a special character, control-P (type CTRL and P simultaneously) is used to notify the system that the operator is ready to enter a message. The system finishes sending to the printer messages waiting to be issued, if any. Then it requests input from the keyboard and waits several minutes¹ for the message from the operator. After that, the request for input from the keyboard will be aborted.

If the system contains teleprinters other than the operators console, they may be used by a program running under OS/700 (not by the system itself) to exchange messages with the operator. The program may issue a message requiring a response and wait several minutes¹ for it. Control-P is not required before entering a message (and will be treated as part of the message if typed). Unsolicited messages (system commands) will be ignored.

¹The time-out may vary from one system to another.

For a description of stand-alone message content, refer to the System 700 Stand-Alone Utility Programs manual; for a description of system messages, refer to "Operator Interface With OS/700" In Section III.

SPECIAL CHARACTERS

The OS/700 operating system uses the following special characters:

Control-P - (Written (P) in this manual.) Used to notify the system that the operator is ready to enter a message (operator's console only).

Control-K - (Written (K) in this manual.) Used to abort output.

At sign (@) - Used to delete an entire line. A new line may be typed without retyping (P). (See Note.)

Back Arrow () - Used to delete the character just typed, except a control-P. (See Note.)

Apostrophe - Used to designate an octal number; for example, '34 is octal 34.

Carriage Return (CR) - Used to terminate a line.

NOTE: These characters are effective only if the line has not yet been terminated.

For special characters used in the Stand-Alone utilities, see the System 700 Stand-Alone Utility Programs manual.

SECTION III

OPERATING PROCEDURES

This section describes procedures for loading into memory and starting OS/700, for loading and running programs under OS/700, and for communicating through a console with OS/700 and with programs running under it.

DEFINITIONS

- Configuration. - The process of designing and creating a system tailored to meet specific hardware capabilities and software requirements.
- Macro deck - The configuration source; contains system macro calls specifying the characteristics of the system being configured.
- DOS (Disk Operating System) - A system configured with a system disk. Portions of the system as well as user programs will reside on the system disk and will be loaded into memory automatically as needed.
- COS (Core Operating System) - A system configured without a system disk. The entire system must be memory resident. Programs to be run may be loaded from peripheral devices by the operator.
- Linking - The part of the program development procedure (see Section IV) that defines where the blocks of code making up a complete program will reside in memory.
- Link map - A listing of symbols produced by the linkage editor. The listing is ordered numerically by the values associated with the symbols. In general, the values are the locations of words of memory referenced symbolically within the linked program.
- Memory image text - A program representation in which sequential words in sequential records correspond on a one-to-one basis with sequential locations in memory.
- Link text - A memory image representation of a program in which repeated words are compressed and in which blocks are not sequential but contain pointers to memory locations.
- External medium - The storage medium associated with a peripheral device, such as magnetic tape, cards, etc.

Disk build - The procedure for building a system disk for a DOS configuration. (See the OS/700 System Generation manual.) A DOS is supplied in several pieces on an external medium. Disk building transfers the operating system and the programs necessary to load and start the system in memory from the medium to the disk.

Keyin loader - A first level loader entered in the first '17 words of memory through the control panel (see Section II) which will load a higher level loader from a peripheral device.

Bootstrap loader - A higher level disk/memory loader brought into memory from a disk by a key-in loader. It is used to load memory image text from disk. The disk build procedure puts a bootstrap loader on the system disk.

Link text loader - A higher level loader brought into memory by a key-in loader from an external medium. It is used to load link text into memory from the medium. The loader must have been written on the medium by a link text loader generator. (Refer to the System 700 Stand-Alone Utility Programs manual.)

Free memory - Blocks of memory available to the user program as needed for buffers, parameter lists, etc.

Generic device type - A 2-digit number assigned to each class of device supported by OS/700. For example, all magnetic tape devices are generic device type 10 ('12).

Logical unit number - A number from 0 to 15 associated with each physical device by configuration. The assigned logical unit numbers are always unique within a generic device type. Therefore, if there is more than one controller of a generic device type, there may be units whose physical unit numbers are not the same as their logical unit numbers.

LOADING AND INITIALIZING OS/700

This section describes system loading and initialization for both core operating systems (COS) and disk operating systems (DOS). It also explains initialization error halts.

COS Loading and Initialization

To load and initialize a COS, the link text of the system must exist on an external medium. This may be the Honeywell-supplied system or one the user has created. (Refer to the OS/700 System Generation manual.)

NOTE: If there is more than one disk unit on a controller which is to be accessed while the system is online, all of the units must be powered up before the system goes online and must stay powered up. Powering a disk drive up or down while other units on the controller are cycled up may destroy the disk pack.

To load and initialize the COS:

1. Load the external medium containing the system link text or the appropriate input device.

2. Enter the key-in loader for the proper medium. (Documentation of the key-in and link text loaders is in the System 700 Stand-Alone Utility Programs manual.)
3. Run the key-in loader to load the link text loader at 'xx400, where 'xx000 is the location of the symbol ZAFTOP in the system link map.
4. Run the link text loader to load the link text of the system in memory. At this time, it is possible to load other link text files that are needed in memory. These files may be code executed as part of the COS as a program under COS control or as a stand-alone program executed independently of the COS.
5. If the system requires patching, it may be patched at this time in one of three ways:
 - a. By loading a patch in link text form
 - b. By running the stand-alone utility I/O debug program, IOAID. (Refer to the System 700 Stand-Alone Utility Programs manual.)
 - c. By entering the patch directly through the control panel
6. Initialize the system:
 - Set STOP/RUN to STOP.
 - Press MASTER CLEAR.
 - Set the Y-register to '201.
 - Set STOP/RUN to RUN.
 - Press START.

The following message, followed by an identification line, should appear on the console:

*****OS/700 INITIALIZATION COMPLETE

If it does not, see the discussion of initialization errors below. If further timeout occurs immediately, refer to the discussion on automatically started activities under "Loading and Starting Activities" later in this section. If it is necessary to restart a COS, it must be reloaded from the external medium and initialized. It cannot be restarted at location '201.

If peripheral devices are to be accessed by the system, the system online utilities must be used to reserve the device (RS) or connect the volume (CV, for disks only).

DOS Loading and Initialization

To load and initialize a disk operating system, a system disk must have been built using the disk building procedure described in the OS/700 System Generation manual. During that procedure, the disk/memory bootstrap and the memory image of the system are written on the disk.

NOTE: If there is more than one disk unit on a controller which is to be accessed while the system is online, all of the units must be powered up before the system goes online and must stay powered up. Powering a disk drive up or down while other units on the controller are cycled up may destroy the disk pack.

To load and initialize the DOS:

1. Mount the system disk.
2. Enter the proper disk key-in loader. (Documentation of the key-in and bootstrap loaders is in the System 700 Stand-Alone Utility Programs manual.)
3. Run the key-in loader, setting B equal to the starting segment number of the bootstrap.¹ The key-in loader loads the disk/memory bootstrap at 'xx400 where 'xx000 is the location of the symbol ZAFTOP in the system link map. The bootstrap loads the system memory image and starts initialization of the system at location '201.
4. If it is necessary to patch the system, set SENSE Switch 2 before running the key-in loader. This causes the bootstrap to halt at 'xx776, after the system memory image is in memory, instead of jumping to '201. The system may be patched through the control panel or by starting the stand-alone utility I/O debug program, IOAID (see the System 700 Stand-Alone Utility Programs manual), if it is in memory. If the patched memory image is to be saved, it is possible to write it on the disk either in place of the previous system memory image or elsewhere on the disk by restarting the bootstrap at location 'xx400 with SENSE Switch 1 set. (See the System 700 Stand-Alone Utility Programs manual for details on transferring a memory image to disk.) After patching:
 - Press MASTER CLEAR.
 - Set the Y-register to '201.
 - Set RUN/STOP to RUN.
 - Press START.

Whether or not the system is patched, the following message, followed by an identification line, should appear on the console:

*****OS/700 INITIALIZATION COMPLETE

If this message is not printed, see the discussion of initialization errors below. If further typeout occurs immediately, see the discussion of automatically scheduled activities under "Loading and Starting Activities" later in this section. The DOS cannot be restarted at '201. If a DOS must be restarted, reload it from the disk using the key-in loader.

¹If the disk build procedure was used to build the disk, the bootstrap segment number will be:

- 1 for any moving-head disk
- 4 for a fixed-head disk with 64-word segments
- '10 for a fixed-head disk with 128-word segments
- '20 for a fixed-head disk with 256-word segments
- '40 for a fixed-head disk with 512-word segments

If peripheral devices are to be accessed by the system, the system online utilities must be used to reserve the device (RS) or connect the volume (CV, for disks only).

Errors During Loading and Initialization

If an error occurs while a link text loader is loading a COS into memory, the loader prints an error message on the console and halts.¹

If an error occurs while a disk/memory bootstrap is loading a DOS into memory, the bootstrap halts.¹

See the System 700 Stand-Alone Utility Programs manual for an explanation of the loader error messages and halt locations.

If an error occurs during initialization, the system halts.¹ The following steps should be performed after such a halt:

1. Display the halt location (Y-register).
2. Find the symbolic name of the halt location on the system link map.
3. See Appendix A for a description of the error halt.

NOTE: Initialization error halts are referred to by symbolic name because the actual halt locations vary with different system configurations.

If the system does not initialize and does not halt, it may be in a loop waiting for free memory to become available. This can be detected as follows:

1. Set the H-register² to ZIOFBK + '14. (ZIOFBK is in the system link map.)
2. Set HALT/NORMAL to HALT.

If the system halts at location ZIOFBK + '14 there is insufficient free memory. (The system may also loop, waiting for free memory, after initialization. In that case, it might be a temporary insufficiency, and the procedure above should only be used if a permanent insufficiency is suspected.)

¹When the RUN light goes out, a halt has occurred.

²The H-register may be set while the system is in RUN mode.

RUNNING PROGRAMS UNDER OS/700

Programs that run under control of OS/700 are called activities. The Honeywell-supplied system includes activities listed in Table 3-1. In a DOS, these activities may exist on the disk as disk resident activities as a result of the system disk building procedure. User-written programs, which are the end result of program development (described in Section IV), can also be run under OS/700. The procedures for running a program as an activity under OS/700 are described below.

The operator can communicate with the system or with activities via the console. Unsolicited typeins to the system are system commands, which initiate and control the execution of activities. The use of the available system commands is defined in the procedures below. The details of their use is described under "Operator Interface With OS/700" later in this section.

Table 3-1. Supplied Activities

Activity Name	Function	Documentation
ZUTY	Online DOS system utilities (file, activity, and disk support; debug and patching)	<u>OS/700 Online Utility Programs manual.</u>
ZUTM	Online COS utility transfer media package	
ZUVM	Online COS utility verify media package	
ZUTV	Online COS utility transfer and verify media package	
ZUDB	Online COS utility debug package	
ZBUT	Magnetic tape blocking utility	
ZEDIT	Text editor	<u>EDIT 700 Source Text Editor manual or Section IV of this manual.</u>
ZFORT	FORTTRAN translator	<u>OS/700 FORTRAN manual or Section IV of this manual.</u>
ZAP	Macro assembler	<u>DAP/700 Macro Assembler manual or Section IV of this manual.</u>
ZLE	Linkage editor	<u>LINK-700 Linkage Editor manual or Section IV of this manual.</u>

The operator can communicate with an activity only when the activity issues a message that requires a response. The standard form of such messages is described under "Operator Interface With OS/700 later in this section. The form of the response must be found in documentation of the individual activity.

Loading and Starting Activities

Activities are loaded into, and run in, areas of memory called activity areas, which are defined when the system is configured. The activity must be linked so that it fits in the activity area in which it is to be run. (Refer to "Program Development" in Section IV. In a DOS, the activity areas may overlap. Activities that must be in memory simultaneously must not be linked for overlapping activity areas. COS activity areas do no overlap.

The system must know an activity's attributes (e.g., name, memory area to be used, starting address) to run it. This information, supplied by the programmer, is specified to the system in different ways, depending on the type of activity area.

Loading and starting activities are described below. It is assumed that a link text file of the program is available, linked for the activity area in which it is to be loaded. For details on the use of system commands, refer to "Operation Interface With OS/700."

COS ACTIVITIES

There are three types of COS activity areas:

- Blank
- Permanent
- Temporary

Procedures for loading and starting activities in each type of activity area are below.

Blank COS Activity Areas

The low memory address and high memory address of the activity area are defined in the system configuration. (Refer to DSKACT in MACxx.) Any program that fits in the area may be loaded and run after system initialization.

The procedure is as follows:

1. Use the Activity Memory Image Text Generator, AMITG (see the Stand-Alone Utility Programs manual), to convert the link text file of the program to an activity memory image file. The activity attributes required by the system are specified to AMITG and the output file contains the attributes and the activity memory image.

2. Load and initialize the COS. (Refer to "COS Loading and Initialization" above.)
3. Load the AMITG output file on the appropriate input device.
4. Enter the \$LA system command to transfer the activity attributes to the system and load the activity memory image into the proper activity area from the input device.
5. Enter the \$SA system command to start execution of the activity. The activity may be restarted with a \$SA command whenever needed, unless it is nonreusable (one of the attributes). If an activity is nonreusable, steps 3 and 4 must be repeated before the activity can be restarted.

Steps 3, 4, and 5 can be repeated as often as desired for any AMITG activity file.

Permanent COS Activity Areas

All attributes of an activity are defined for the activity area in the system configuration. (Refer to DSKACT in MACxx.) The configured activity must be loaded in memory before system initialization. The area can be configured to start the activity automatically when the system initializes, or it can be started by the operator. No other activity can be loaded in the activity area after the system initializes.

To run an activity in a permanent COS activity area, perform the following steps:

1. Load the link text file of the configured activity into memory, using a link text loader, when the system is being loaded.
2. Initialize the COS. If the permanent activity area was configured with the automatic starting option, the activity is started during initialization.
3. If not, enter the \$SA system command to start execution of the activity. The activity can be restarted with a \$SA command whenever needed, unless it is nonreusable. If the activity is nonreusable, repeat steps 1 and 2 above before restarting the activity.

NOTE: If the program was not loaded (step 1) and the \$SA command was entered, the system will attempt to start the activity and will crash.

Temporary COS Activity Area

All of the attributes of an activity are defined for the activity area in the system configuration. (Refer to DSKACT in MACxx.) The activity named in configuration must be loaded in memory before system initialization. The area can be configured to have the activity started automatically when the system initializes, or it can be started by the operator. Other activities can be loaded in the area after the system initializes, but only if the configured activity has been run and has terminated.

To run an activity in a temporary COS activity area, perform the following steps:

1. Use step 1 of the procedure for blank activity areas for any activity to be loaded after system initialization.
2. Use steps 1 and 2 of the procedure for permanent activity areas and the configured activity.
3. If the configured activity is not automatically started, enter the \$SA system command. The activity may be restarted with a \$SA command whenever needed, unless it is nonreusable. If the activity is nonreusable, step 2 above must be repeated or step 1 and the procedure below must be used before the activity can be restarted.

NOTE: If the program was not loaded (step 2) and the \$SA command was entered, the system will attempt to start the activity but will crash.

4. When the configured activity terminates, use steps 3, 4, and 5 of the procedure for blank activity areas to load and start any desired activity.

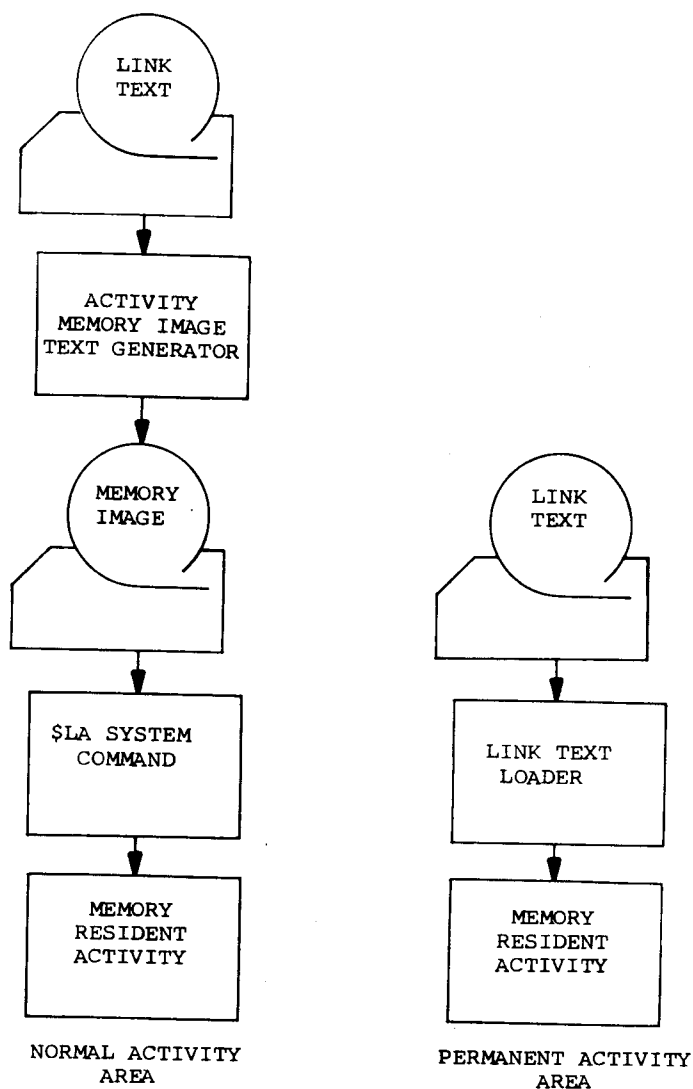


Figure 3-1. COS Activities

DOS ACTIVITIES

There are two types of DOS activities and activity areas:

- Activity areas for disk-resident activities
- Permanent activity areas for memory-resident activities.

Figure 3-2 shows DOS activities as they occur in activity areas. Procedures for loading and starting activities in each type of activity area are below.

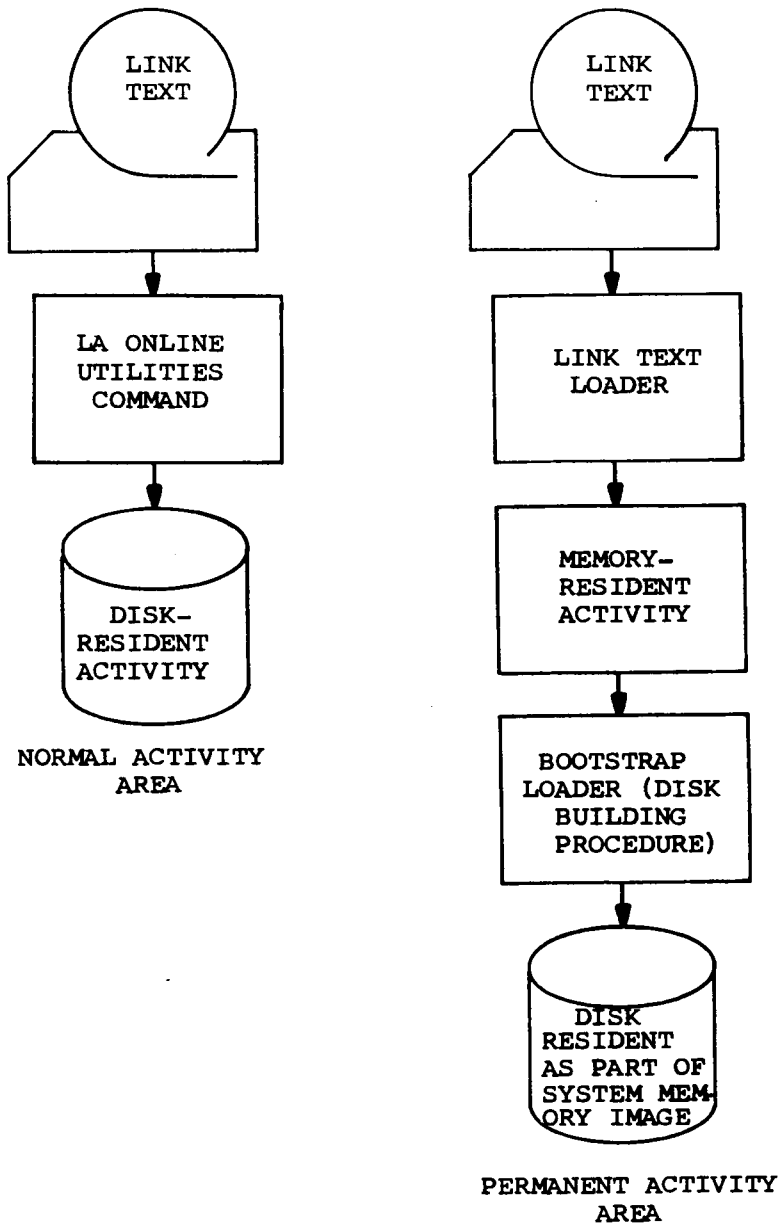


Figure 3-2. DOS Activities

DOS Activity Areas for Disk-Resident Activities

The low memory address and high memory address of the activity area are defined in the system configuration. Any program that fits in the area may be loaded and run after system initialization.

The procedure for loading and starting a disk-resident activity in a DOS activity area is:

1. Load and initialize the DOS.
2. Enter the \$SA system command to start the online system utilities. (Refer to the OS/700 Online Utility Programs manual.)

NOTE: The system utilities exist on the disk as a disk resident activity as a result of the disk building procedure.

3. Load the link text file of the program on an input device, if it does not exist as a file on the disk.
4. Enter the LA online system utilities command to create an activity on the disk, in memory image format, from the program link text file. The activity attributes required by the system are specified in the LA command.
5. Enter the \$SA system command or the SA online system utility command to load the activity from the disk into memory and start execution. If the activity requires parameters or is to be run in debug mode, the system utility command must be used. If the LA command specified a restricted activity, the activity can be loaded and run only in an activity area configured for restricted activities.

- Notes:
1. Once the activity is disk resident, steps 3 and 4 are not needed. The activity may be restarted with a \$SA command whenever needed.
 2. If the activity is to occupy the same activity area as the system utilities, the utilities must be terminated before the activity can be started.

Permanent DOS Activity Area

All attributes of an activity are defined for the activity area in the system configuration. The configured activity must be loaded into memory before system initialization. The area can be configured to start the activity automatically when the system initializes, or the operator can start it. No other activity can be loaded in the activity area after the system initializes.

The procedure for loading and starting an activity in a permanent DOS activity area is:

1. Load the program link text file into memory, using a link text loader, during the system disk building procedure at the point where the system link text is being loaded into memory. (Refer to the OS/700 System Generation manual.) The activity will then be written on the disk as part of the system memory image.

2. Load and initialize the DOS.

If system memory image contains the I/O debug program:

- a. Enter the key-in loader for the disk, but set SENSE Switch 2 before starting the key-in. (Refer to "DOS Loading and Initialization" above.) The bootstrap will halt at 'xx776.
 - Press MASTER CLEAR.
 - Set the Y-register to the start address of IOAID ('4000 in the supplied system).
 - Set the STOP/RUN switch to RUN.
 - Press START.Use IOAID to load the program link text file into memory. (Refer to the System 700 Stand-Alone Utility Programs manual.)
- b. Jump to location '201 (IOAID command JC) to initialize the DOS.
- c. If the permanent activity area was configured with the automatic starting option, the activity is started during initialization. If not, enter \$SA system command to start execution of the activity. Hereafter, the activity can be restarted with a \$SA command whenever needed, unless it is nonreusable. If the activity is nonreusable, steps 1 and 2 above must be repeated before the activity can be started again.

NOTE: If the program was not loaded (step 1) and the \$SA command was entered, the system will attempt to start the activity but will crash.

Using Activities

Activities have a standard message interface with the operator. When an activity needs a question answered or is ready for input, it issues a message requiring a response. The particular form of the response varies with the activity and must be found in the documentation of that activity. The special character control-K, (K) can be used to abort activity output. (Refer to "Operator Interface With OS/700.")

It is possible for one activity to start another. Thus, messages can occur on the console for activities not started by the operator. Also, in a DOS that has more than one disk unit, the system can allocate a disk unit to an activity. This is done through the Volume Manager, which may issue messages requiring operator response. The format of these messages and their responses is described in "Disk Allocation Messages" later in this section.

COMMAND FILES (DOS ONLY)

In a DOS configured with "command input" capability, the operator can put the system in CI mode — a mode in which system commands and activity message responses are not entered by the operator. They are fetched from a command file by the CI logic. In effect control of the system is transferred from the operator's console to the command file.

The \$CI system command puts the system in CI mode. As the command file lines are fetched, they are written to the current "output device." The operator's console is the default output device when CI mode is entered, but the command file can switch to another device with the \$OD system command. The command file can write comments to the output device with the \$CO command, and all messages from an activity running in CI mode are written to the output device, including error messages. However, system error messages will continue to be written on the operator's console.

A command file may start an activity in CI mode with a \$SA command line. No activities may be running when the system enters CI mode and only one activity at a time may be started by the command file. If the activity issues a message to the operator's console requiring a response, the message is sent to the current output device and the response is fetched from the command file. The command file can also transfer control to another command file with a \$CI command. Nothing can be entered on the console when in CI mode except a \$TR (Terminate CI Mode) command. Any other entry causes an error message on the console (SE + 100001), but CI mode continues.

Activity level command files may be used in CI mode (only for activities which have a command input capability). A system level command file may contain a line which transfers control to an activity level command file. When the activity finishes processing the activity level command file, the system returns control to the system level command file at the next line. For example:

System level command file	
:	
:	
\$SA ZLE	
\$CO TRANSFER CONTROL TO AA	ZLE command file, AA
CI = AA	:
QUIT	:
\$CO ZLE DONE	:
:	:
:	:
:	:

In this example, the supplied activity, ZLE (OS/700 Linkage Editor) is started, a ZLE command file is executed and control is returned to the system level command file which terminates ZLE and continues.

CI mode terminates when any of the following conditions arise:

- An end of file is encountered in the command file, (an end-of-file message, CI03, appears on the console.)
- A CI error occurs. (A system error message, SE= 100210 through SE= 100223, appears on the console.)
- A \$TR command is entered on the console. (No message occurs.)

Control returns to the console when CI mode terminates. If an activity is running, when the \$TR command is entered, it will finish its current processing (which may be lengthy). When the system accepts input from the console, CI mode has terminated.

An activity may continue to run when CI mode terminates, requiring responses from the operator to continue or terminate the activity.

Caution must be used with command files. If the required devices and files are improperly set up, the currently running activity will issue an error message to the output device; but the CI logic continues to process commands. The results may be undesirable, and not easily visible to the operator if the current output device is not the operator's console.

Terminating Activities

Activities terminate themselves when they finish processing or when an unrecoverable error occurs. Supplied activities that interact with the operator issue a message when they terminate.

In a COS, the message

<actnam TR>

is typed on the operators console when an activity terminates.

In a DOS with configured system integrity, an activity running in restricted mode can be aborted by the system, by the operator (\$AB command), or by another (nonrestricted) activity. Activities are run in restricted mode:

- To protect the system and other activities
- To prevent the activity from looping or taking too long to execute
- To allow them to be aborted if system resources are scarce.

When a restricted activity is aborted, execution stops, requests associated with the activity (e.g., output requests) are deleted, and its resources are returned to the system.

Honeywell-Supplied Activities

OS/700 supplied activities are listed in Table 3-1.

OPERATOR INTERFACE WITH OS/700

The operator may communicate directly with user activities and the operating system itself (COS or DOS) via a console. An activity or the system may print a message on the console, to which operator typein response may or may not be required. In addition, the operator may type a set of commands (system commands) on the console with no prior prompting by a user activity or by the system.

The message and command formats are given below including the use of the associated characters control-P, and control-K, (K), to abort output. The available system level commands are summarized and their use detailed. Also the Volume Manager interface for disk allocation is described. Error recovery and/or reporting procedures are described.

Formats

The format for messages requiring no response is:

actnam message

actnam - Name of the activity for whom the message was issued.
If the system issues a message on behalf of no particular activity, actnam may be replaced by *****.

message - Message originating from the user activity or from the system.

Exceptions to the above format are abort messages and system error messages.

Abort messages have the format

***** actnam rr [aaaaaa]

See Appendix D, "Abort Activity Messages," for an explanation. Abort messages will occur if the system or the operator aborts an activity. System error messages include I/O device errors, executive errors and communication supervisor errors. The formats are:

I/O device errors:

SE= 0eeeee dddddd

Executive errors:

SE= 1ffffff SSSSSS

SE- 1ffffff SSSSSS actnam

Communications supervisor errors:

SE= xxxxxx yyyyyy CSEVRT
SE= xxxxxx yyyyyy CSCNFE

(See Appendix B for a complete explanation of system error messages.)

The format for messages requiring response is:

xx actnam message

xx - 2-digit message number that allows the operator's response to be associated with a particular message.

actnam - Name of the activity for whom the message was issued.

message - Message originating from the user activity or from the system.

The operator's response has the format:

(P)xx response (CR)

(P) - Nonprinting control-P.

xx - Message number of the message to which the operator is responding.

response - Operator's response to the message.

(CR) - Carriage return. A (CR) terminates the response.

If there are several console typeouts with different message numbers, the operator can respond to them in any order.

If the message requiring response is issued on a teleprinter other than the operator's console, the format is:

message

The operator's response has the format:

<response>

The message number and activity name are unnecessary because such messages will be issued one at a time by the activity which has reserved the device and the activity will wait for the response. The form of the response varies. It must be found in documentation of the program being run. The response for DAP/700 for example, may be found in Section IV or in the DAP/700 Macro Assembler manual.

The format for system level commands is:

(P)\$ aa parameter 1 ,..., parameter n (CR)

(P) - Nonprinting control-P.

aa - 2-letter system command.

parameter - n parameters required by the system command.

(CR) - Carriage return.

All of the available system commands are described under "System Commands" below. The special characters control-P, control-K, back arrow (←), and at sign (@) are used as described below.

Control-P (P), is typed to notify the system that the operator is ready to type input on the console. It causes an input request to be placed on the I/O request queue for the device. If the console is not busy, the input request is processed immediately; i.e., the operator may type a system command or response following the (P). If output is occurring, the input request is placed on the queue; but the operator cannot type the input until the preceding output requests on the queue are processed (i.e., the operator types (P) and waits until the console stops typing output lines. Then, a system command or response - with a message number - is typed without repeating (P).)

Control-K (K), is typed to kill a current output line. If no output is occurring, (K) is ignored. If output is occurring, the line being typed is stopped and a (K) status is reported. The system ignores the (K) status and continues without repeating the line. An activity may do anything from repeating the line to terminating and the reaction of a specific activity must be found in the documentation of that activity.

The back arrow (←) deletes the previous character except control-P. Successive back arrows erase the corresponding number of preceding characters.

The at sign (@) deletes an entire line except control-P. The system automatically issues a carriage return and line feed, ignores the lines just deleted, and waits for a new line to be entered (without retyping control-P).

The back arrow and at sign are effective only if the line has not yet been terminated.

System Commands

In a dedicated COS, there are no system commands. In a nondedicated COS, there are two:

- \$LA - Load activity
- \$SA - Start activity

In a DOS, there are three basic system commands:

- \$SA - Start activity
- \$CI - Command input mode
- \$AB - Abort activity

The system configuration cannot contain both the Command Input (\$CI) and Abort Activity (\$AB) capabilities. Therefore, in any particular DOS, at most two basic system commands are available in the following combinations:

- \$SA and \$CI
- \$SA and \$AB
- \$SA only

When the DOS is in command input mode as a result of a \$CI command, several more system commands are available, but the use of these system commands is restricted as summarized in Table 3-2.

Table 3-2. System Command Restrictions in a DOS
With Command Input Mode Configured

CI Mode Effective	Commands Accepted From Console	Commands Accepted From a Command File
NO	\$SA - Start Activity \$CI - Command Input Mode	None
YES	\$TR - Terminate CI Mode	\$CI - Command Input Mode \$CO - Comment \$OD - Output Device \$SA - Start Activity

\$CO

COMMENT COMMAND (\$CO)

The Comment command (\$CO) causes a comment to be written on the current output device in CI mode. (Refer to "Command Input Mode Command (\$CI)"). The \$CO command can only occur in a CI mode command file; it cannot be entered on the console.

\$CO Format

\$CO comment (CR)

comment - Any desired comment; no more than 64 characters.

(CR) - Carriage return.

\$CO Action

The \$CO command line is written on the current output device as are all the lines in the command file. It takes no action of its own.

\$CO Messages

Errors are reported in system error messages. See Appendix C.

\$CI Messages

The end-of-file message, CI03, is issued when an end-of-file is encountered in a command file. Failure to initiate or to continue in CI mode will be reported in a system error message. See Appendix C.

\$CI

COMMAND INPUT MODE COMMAND (\$CI)

The Command Input Mode command (\$CI) initiates command input mode in which control is transferred from the operator's console to a command file. There may not be any activities running in the system when a \$CI command is typed. It will not be recognized unless the system was configured for command input. \$CI may occur in a command file.

\$CI Format

(P)\$CI	filnam (CR)	(console format)
\$CI	filnam (CR)	(command file format)

(P) - Nonprinting control-P.

<filnam> - 1- to 6-character command file name to which control is transferred.

(CR) - Carriage return.

\$CI Action

The file <filnam> is opened and control is transferred to it; that is, all messages to the system will be input from filnam. If \$CI occurred in a command file, that file is closed before the new file is opened. The operator's console becomes the current output device until a \$OD command changes it. Each line encountered in the command file is written to the current output device (before it is analyzed) as well as all messages issued by the system on its own behalf or on behalf of an activity (except system error messages which will always be output to the operator's console).

No input is accepted from the operator's console in command input mode except a \$TR command. Any other input causes an error message on the console (SE= 100001), but permits command input mode to continue.

CI mode terminates when:

- An end-of-file for <filnam> is encountered. (An end-of-file message, CIO3, occurs.)
- A CI error occurs. (A system error message, SE= 100210 through SE= 100223, occurs on the console.)
- \$TR command is entered on the console. (No message occurs.)

The command file is closed and control returns to the console.

NOTE: If processing appears to have ceased and input from the console is accepted, CI mode has terminated. An activity may continue to run when CI mode terminates, requiring typeins from the console to continue or terminate the activity.

ABORT ACTIVITY COMMAND (\$AB)

The Abort Activity command (\$AB) is used to abort a restricted activity that has been requested or is running. \$AB is recognized as a system command only in a DOS configured with system integrity.

\$AB Format

(P)\$AB actnam (CR)

- (P) - Nonprinting control-P character.
- actnam - 1- to 6-character ASCII name of the restricted activity to be aborted.
- (CR) - Carriage return.

\$AB Action

Execution of the activity stops. All requests for or associated with the activity are terminated. All resources e.g., free memory and devices are returned to the system.

NOTE: Restricted activities are allowed to allocate and deallocate work areas on the disk. If the activity is aborted before work areas have been deallocated, they will remain allocated until the Clean Disk utility (CD) is executed. (See the OS/700 Online Utility Programs manual.)

\$AB Messages

When the activity has been aborted, it is reported by the following message:

***** actnam OP

If free memory is very low, output of this message is suppressed.

A failure to abort the activity will be reported in a system error message. See Appendix C.

LOAD ACTIVITY COMMAND (\$LA)

The Load Activity command (\$LA) is used in nondedicated core operating systems to load activities into memory from an external medium.

Activities to be loaded must be in memory-image text format as output by the Activity Memory-Image Text Generator (AMITG) utility program. Refer to the OS/700 Stand-Alone Utility Programs manual.

\$LA Format

(P)\$LA <actnam>,<ddu>(CR)

(P) - Nonprinting control-P.

actnam - 1- to 6-character ASCII name of the external activity to be loaded into memory. Activity names must be left-justified and followed by enough spaces to fill the 6-position field.

'dd - Generic device type of the external medium; the apostrophe is mandatory. Possible generic device types.

'02 - High-speed paper tape reader

'06 - Card reader

'12 - Magnetic tape

'13 - Cassette tape

'15 - ASR paper tape reader

u - Logical unit number (0 through 7) of the input device.

(CR) - Carriage return.

\$LA Action

The first record of the activity memory image text file is read from the medium. It contains the name of the activity and its attributes. If the name in this record matches actnam, the memory image text is read into the proper activity area.

If the names do not match, an error message is issued. However, if the medium is magnetic tape, the first record of each succeeding file on the tape is checked for the name <actnam>. If a matching name is found, the file in which it is found is read into the activity area. If a matching name is not found before two consecutive end-of-file marks or a file not of AMITG format is encountered, the error message is issued.

NOTE: The double end-of-file, written by AMITG, denotes logical end-of-tape.

A space file command is issued to a cassette or magnetic tape device, after the activity is loaded, to position it at the beginning of the next file.

\$LA Messages

Table 3-3 describes console typeouts that may occur.

Table 3-3. \$LA Messages

Message	Meaning	Operator Action
***** LD	Activity successfully loaded.	Schedule the activity at any time. If an I/O error message is also issued, see Note a at the end of this table.
***** NP	Next activity to be read is not the named activity. ^b	Correct the activity name entered in the command or put the proper activity memory image text file on the input device.
***** AB	Activity area required is occupied or nonexistent. ^b	<p>If the activity area is occupied, wait until the activity using it terminates (types actnam TR on the console).</p> <p>If the activity area is nonexistent the named activity cannot be run in the system. (A configuration source listing shows the activity areas configured - PACBGN macro call).</p>
***** RR	Reserve/release device error. (Device probably nonexistent or reserved). ^b	Correct the generic device type or unit number entered in the command or reenter the command when the activity using the device releases it or terminates. (A configuration source listing shows the devices configured - Dxyz macro call.)
***** IE	Command input error. ^b	Retype command in correct format.
***** IO	Input error. (Usually with a system I/O error message) ^b	Correct the I/O error if possible (e.g., jammed cards) or repeat the operation (e.g., tape parity error). If an "***** LD" message is also issued, see Note a.
<p>^aAn I/O error may occur from the space file command issued after the activity is loaded. If the ***** LD message also occurs, the ***** IO message may be ignored.</p> <p>^bThese errors leave the file at an indeterminate position. For cassette or magnetic tape, errors other than I/O errors leave the tape positioned at the beginning of the next file.</p>		

OUTPUT DEVICE COMMAND (\$OD)

The Output Device command (\$OD) specifies the current output device in CI mode; refer to "Command Input Mode Command (\$CI)" above. The \$OD command can occur only in a CI mode command file; it cannot be entered on the console.

\$OD Format

\$OD <ddu>(CR)

dd - 2-letter device type of the new output device. Possible device types:

OC - Operators console

PP - Paper tape punch

LP - Line printer

CP - Card punch

MT - Magnetic tape

CA - Cassette tape

u - Logical unit number of the new output device; may be 0 through 9 or left blank for the default. Default is 0.

\$OD Action

The following lines in the command file, and all subsequent messages from the system or an activity are output to the device specified. However, system error messages continue to be printed on the operator's console.

\$OD Messages

The \$OD command is output to the device that was the current output device. Any errors are reported in system error messages.

\$\$A

START ACTIVITY COMMAND (\$\$A)

The Start Activity command (\$\$A) is used to schedule an activity for execution in a nondedicated COS. In a DOS, it is used to load and schedule a disk resident activity or to schedule a memory resident activity for execution. It may be entered on the console or occur in a CI command file (in a DOS only).

NOTE: A command file may not run activities simultaneously.
If the \$\$A command occurs in a command file that already has an activity running, an error will occur.

\$\$A Format

(P)\$\$A <actnam>(CR)	(console format)
\$\$A <actnam>(CR)	(command file format)

(P) - Nonprinting control-P.

actnam - 1- to 6-character ASCII name of the activity to be scheduled; in a COS, it must be left-justified and followed by enough spaces to fill the 6-position field.

(CR) - Carriage return.

\$\$A Action

In a COS, \$\$A causes the activity actnam to be scheduled immediately.

- If the activity is reentrant and in memory
- If the activity is reusable or nonreusable, in memory, and not already scheduled or running.

If <actnam> cannot be scheduled immediately, an error message is returned and the operator must retype the command after <actnam> terminates or after <actnam> is loaded from an external medium.

NOTE: A nonreusable activity will not execute properly if it has already been executed and it is started again. The activity must be reloaded from an external medium (\$LA) and then started (\$\$A).

In a DOS, \$\$A causes the activity actnam to be loaded from disk and scheduled if it is:

- Not in memory and the activity area is free
- In memory, not reusable, and not running

\$\$A causes the activity to be scheduled if it is:

- In memory and reentrant
- In memory, reusable, and not running

If actnam cannot be loaded and scheduled or scheduled immediately, the request is saved and actnam will be automatically loaded and scheduled when:

- The activity area is free
- The nonreusable activity actnam terminates or it will only be scheduled when the reusable activity actnam terminates

NOTE: If the \$SA command occurs in a command file, no other activities will be running and the requested activity will always be loaded immediately.

\$SA Messages

In a COS, the messages listed in Table 3-4 report events and error conditions associated with starting activities. Other such errors in a COS and all such errors in a DOS are reported in system error messages. See Appendix C.

Table 3-4. \$SA Operator Messages (COS only)

Message	Meaning	Operator Action
***** SD	Activity successfully scheduled.	None
***** NP	Activity is not in memory.	Load activity into memory.
***** IE	Command input error.	Reenter command in the correct format.
***** AS	Activity already scheduled.	<p>If activity is reusable, wait until it terminates, then retype \$SA command.</p> <p>If activity is nonreusable, wait until it terminates, then reload it from an external medium and retype \$SA command.</p>

\$TR

TERMINATE CI MODE COMMAND (\$TR)

The Terminate CI Mode command (\$TR) terminates CI mode, transferring control from the current CI mode command file to the operator's console. \$TR may be entered only on the console and only when the system is in CI mode.

\$TR Format

(P) \$TR (CR)

(P) - Nonprinting control-P

(CR) - Carriage return

\$TR Action

The command file is closed and control is returned to the console when the \$TR command is detected. If an activity is running, it will finish its current processing before \$TR takes effect. If processing appears to have stopped, and the system accepts input from the console, CI mode has been terminated.

\$TR Messages

Errors are reported in system error messages. See Appendix C.

Disk Allocation Messages (DOS Only)

Disk operating systems having one or more disk units in addition to the system disk may use the Volume Processor to allocate nonsystem disk units to various activities within the system. To do this, the system may request the order to supply a volume name from the console or to mount a moving-head disk pack. Following is a description of the system typeout and operator typein associated with the Volume Processor.

VOLUME NAME REQUESTS

Two forms of volume requests may be typed on the console by the system:

1. xx actnam ENTER dd VOL NAME

2. xx actnam ENTER dd VOL NAME aa,bbbb,cc

xx - 2-digit message identification number;
this number must precede the operator's
response.

actnam - Activity on whose behalf the request is made.

aa - 2-digit decimal specification of how many surfaces
the disk pack must have.

bbbb - 4-digit decimal specification of the segment size,
in words, that was used when the disk pack was
formatted.

cc - 2-digit decimal specification of how many segments
per track the disk pack has.

dd - 2-letter device type mnemonic:

MH - Moving-head disk

FH - Fixed-head disk

CD - Cartridge disk

Requests of form 1 seek the name of a private volume that actnam intends to use. The operator must know this name before starting the activity. Requests of form 2 seek the name of a public unlabeled volume that the system will provide for use by one or more user activities. Any volume name may be supplied, provided the volume fits the four characteristics listed in the message.

The operator may reply to either message with:

1. (P)xx nnnn (CR)

or

2. (P)xx\$K(CR)

- xx - Message number supplied with the volume name request that the operator is answering.
- nnnn - Volume name, one to four ASCII characters; trailing blanks are not required.
- \$K - Fixed reply indicating refusal of request.

If the operator types a reply of form 1, the user's Connect Volume request will continue, and a mount message may be issued. If the operator types a reply of form 2, the user's Connect Volume request is aborted.

MOUNT MESSAGES

A mount message issued by the system has two forms:

1. xx actnam MOUNT nnnn ON dduu
2. xx actnam MOUNT **** ON dduu

xx, actnam, dd - As previously defined
nnnn - Specific volume name
uu - Disk unit number

In response to a form 1 message, power down the unit, remove the volume currently mounted, if there is one, and mount the volume named in the message. If the requested volume is in operation on another unit, power down that unit, remove the volume, and remount it on the specified unit. When the volume has been remounted, power up the unit. Wait until the unit is ready before replying to the mount message.

In response to a form 2 message, select any unmounted public labeled disk, and mount it as detailed above before replying to the mount message.

Two possible replies to either form of mount message are:

1. (P)xx\$G(CR)
2. (P)xx\$K(CR)

xx - Message number associated with the mount message.
\$G - Fixed reply indicating requested mount has been performed.
\$K - Fixed reply indicating refusal to perform mount.

A form 1 reply by the operator signals that the requested volume is mounted, and the unit is ready.

A form 2 reply by the operator signals that the volume has not been mounted. This is normally done when a requested volume is not available, or in the case that the requested volume is already operational on the specified unit. (This condition can arise if the caller of Connect Volume specifies a volume by name and indicates that it is a labeled volume, when in fact it is unlabeled.

The operator will be asked to mount the volume; the Volume Manager will detect that the volume is unlabeled, when labeled was specified, and will re-issue the message, even though the operator mounted the volume successfully. When the second mount message is reissued and the correct volume has been mounted, type a form 2 reply.

SECTION IV

LANGUAGE PROCESSORS

OS/700 program development requires an OS/700 disk operating system (DOS) with the OS/700 language processors and online utilities OR a host-resident system with the OS/700 language processors and the online utilities or their equivalents.

The process of program development and the operator interface to the language processors are described in this section. The following types of files exist at different stages of program development:

1. Source text files (DAP-700 or FORTRAN-700) - 40-word ASCII records.
2. Object text files - 54-word binary records. See the DAP/700 Macro Assembler manual, Appendix D for a detailed description.
3. Link text files - 54-word binary records. See the LINK-700 Linkage Editor manual, Appendix A for a detailed description.
4. Memory image files - 54-word binary records.
5. Listing files - 61-word ASCII records.
6. Link maps (listing file) - 61-word ASCII records.
7. Symbol tables (object text) - 54-word binary records.

The above information is needed to handle the different types of files with the online utilities. For details on use of the utilities, see the OS/700 Online Utility Programs manual.

PROGRAM DEVELOPMENT

To create an OS/700 program, all the necessary DAP-700 and FORTRAN-700 source files or DAP-700 object files must exist on the disk. The source files may be created using the EDIT-700 Text Editor to input a new file or to edit an existing one.¹ Or, they may be transferred onto the disk from an external medium using the online utilities. The FORTRAN-700 FORTRAN translator is then run on all the FORTRAN source files to convert them to DAP-700 source files and to output FORTRAN listings.

¹See the DAP/700 Macro Assembler manual for the format of DAP-700 source statements and the OS/700 FORTRAN manual for the format of FORTRAN-700 source statements.

The DAP-700 source files are assembled with the DAP-700 macro assembler which outputs DAP-700 object text and DAP-700 listings (optional). Or, object files which already exist on some external medium may be transferred to the disk by the online utilities.

Finally, the LINK-700 Linkage Editor links all object modules into a single link text file. The link text file specifies precisely where in memory each word of code will reside and contains the pointers which allow execution to move from one block of code to another. In general, the area of memory to be occupied by the program is specified to the Linkage Editor at link time and the program will always be loaded in that memory area. If any part of the program was originally written in FORTRAN, it will be necessary to link the OS/700 FORTRAN run-time library, which contains the FORTRAN subroutines and math library.

The Linkage Editor outputs a link map and symbol table (both optional). A link map is a listing of locations of defined entry points in memory. The link map, with the DAP-700 listing, can be used to locate code in memory and read memory dumps. The symbol table is the link map in binary form. It may be used to link another object file to a program without relinking the program.

The link text file is the end product of program development. If the program is to run under OS/700 control, it must be loaded into memory and executed as described in Section III. If it is a stand-alone program, it must be transferred to an external medium using the online utilities. Then, a link text loader may be used to load the program into memory. (See the System/700 Stand-Alone Utility Programs manual.) It is executed by setting the Y-register to the program start address and running. (Refer to "Program Execution" in Section II.)

Figure 4-1 shows a flow diagram of OS/700 program development.

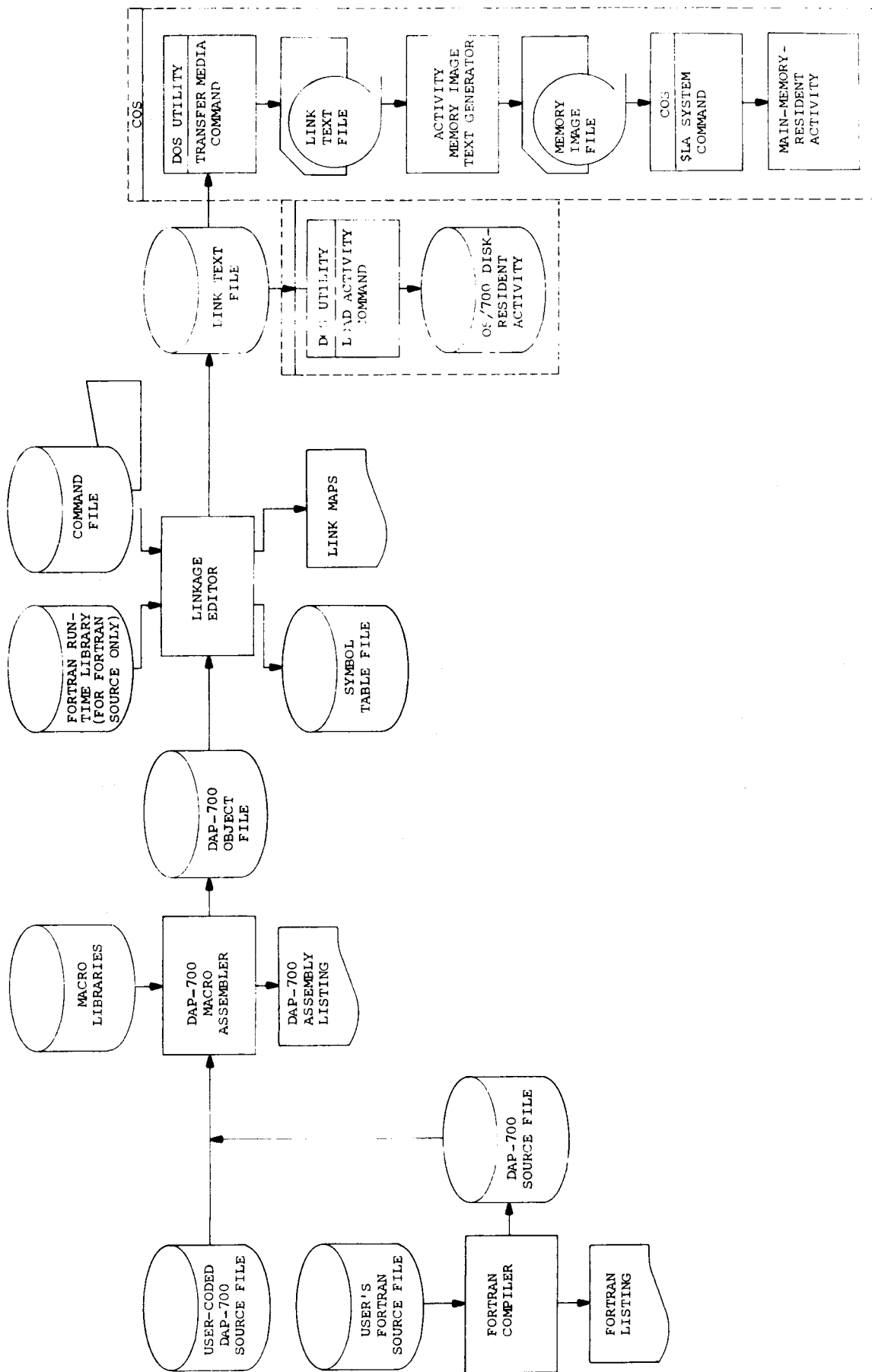


Figure 4-1. OS/700 Program Development

EDIT-700 TEXT EDITOR PROCEDURES

The EDIT-700 Text Editor edits any ASCII file in either batch or conversational mode, and subsequently creates an updated master file. When the text editor is executed in batch mode, command input is from a disk-resident source file; if it is executed in conversational mode, input is from a console. (See Figure 4-2.)

For more details concerning use of the text editor, refer to the OS/700 EDIT-700 Source Text Editor manual.

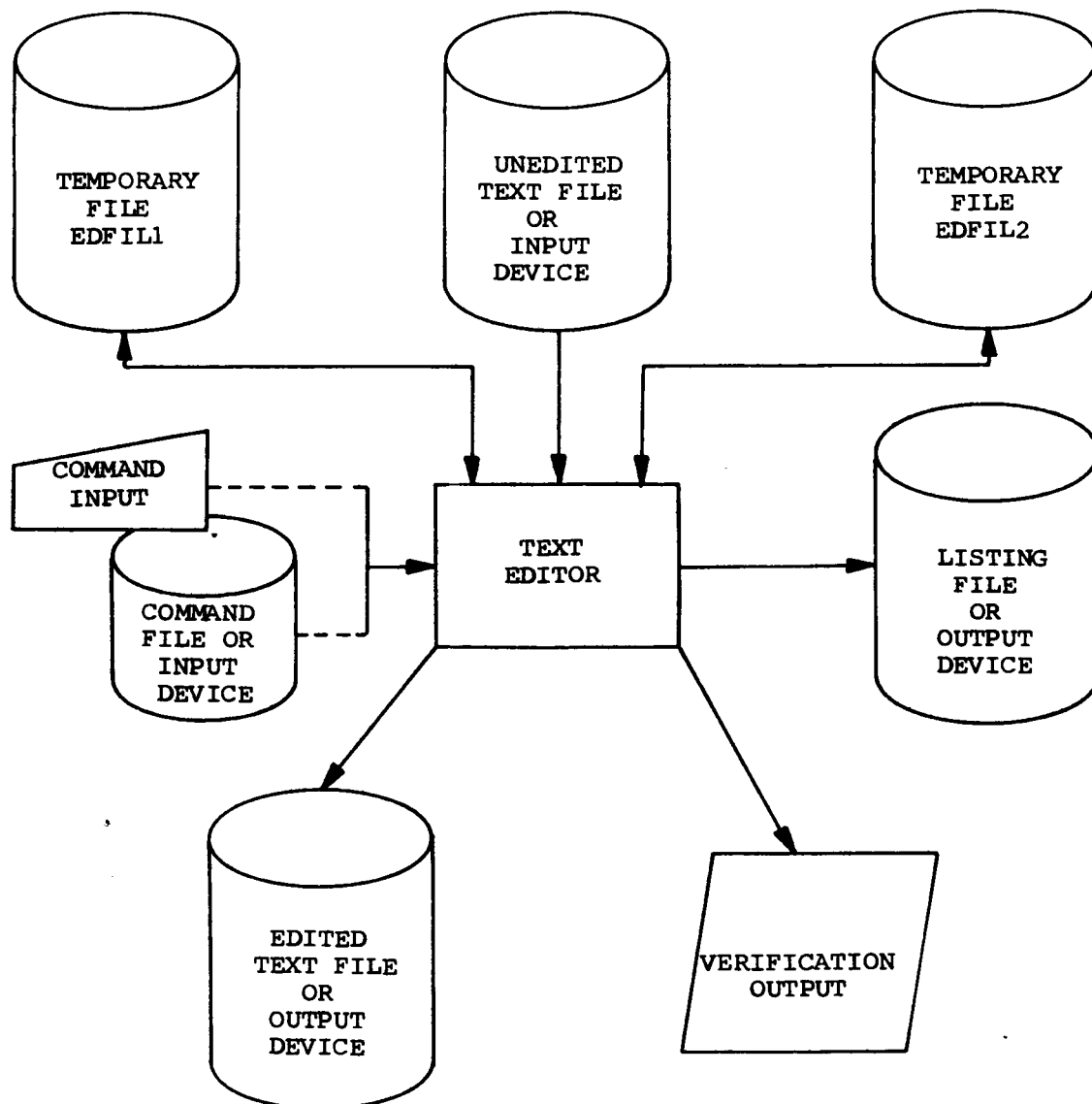


Figure 4-2. Text Editor Flow

Text Editor Execution

Execution of the text editor is initiated by a Start Activity (SA) utility command, which causes the text editor activity to be scheduled. When the activity starts, it types out its identification, followed by an exclamation point to solicit the input of a line of parameters.

The following dialogue represents system typeouts and operator typeins that may appear on the console.

```
(P)$SA ZEDIT (CR)
ZEDIT      EDIT 700 REV n yy/mm/dd
xx ZEDIT    ! (P)xx<fname1>,<fname2>,<fname3>,<fname4>,<fname5> (CR)
xx ZEDIT    ?
```

\$SA - Start activity command

ZEDIT - Name of the text editor activity

EDIT 700 REV - Identifies the version (revision number and date) of
n yy/mm/dd EDIT-700

xx - 2-digit message identification number; this number must precede the operator's response

! - Message soliciting text editor parameter input line

<fname₁> - The source of the input file (old master) - either a filename or an input device mnemonic. If this parameter is omitted or is an asterisk (*), the text editor treats the parameter as an empty old master file; i.e., a new file is created by insertions.

<fname₂> - The destination of the output file (new master) to be generated by the text editor - either a filename or an output device mnemonic. If this parameter is omitted or is an asterisk (*), the text editor performs its normal operations, but no new master file is generated at the end of the editing process.

<fname₃> - The source of the edit commands. If this is a console mnemonic (OC, KS, or AS), conversational mode of operation is assumed; if it is any other type of device or a disk file, batch mode is assumed.

<fname₄> - The destination of the output generated by the PRINT command - either a file name or an output device mnemonic. In conversational mode, PRINT command output is on the command input device named in <fname₃>, if <fname₄> is unspecified. In batch mode, if <fname₄> is unspecified, PRINT command output is suppressed. If <fname₄> is an asterisk (*), PRINT command output is suppressed in both conversational and in batch modes.

<fname₅> - The destination of the verification output of the text editor - either a filename or an output device. In conversational mode, verification output is on the command input device named in <fname₃>, if <fname₅> is unspecified. In batch mode, if <fname₅> is unspecified, verification output is suppressed. If <fname₅> is an asterisk (*), verification output is suppressed in conversational and batch modes.

? - Solicits command input (in conversational mode only).

Files must not be named EDFIL1 or EDFIL2; there must not be files named EDFIL1 or EDFIL2 present on the disk when the editor is started. The editor uses these names for temporary work files during execution.

The device mnemonics, optionally followed by the logical unit number, are:

- OC - Operators console
- KS - KSR keyboard/printer
- PR - High-speed paper tape reader
- PP - High-speed paper tape punch
- CP - Card punch
- CR - Card reader
- LP - Line printer
- MT - Magnetic tape
- CA - Cassette tape
- AS - ASR keyboard/printer
- AP - ASR reader/punch

For example, a conversational edit session can be performed on an alternative KSR (other than the operators console), or the output on the PRINT command can be sent to the line printer. Note that on an alternative KSR the message number and the (P) must be omitted from the response. An appropriate device must be selected; for example, an attempt to write output on the card reader will result in an error, and the text editor will be aborted.

Note that the text editor writes an end-of-file record when it terminates an output operation to a device such as magnetic tape.

Input files to the text editor can contain records longer than 80 characters. However, records on output files are truncated to 80 characters.

In batch mode, typing of the current line and of the output generated by the PRINT command (unless a listing file is specified) is suppressed. Error messages are typed on the operator's console. The text editor terminates when it reaches an end-of-file or QUIT command in the command input file.

In conversational mode, the PRINT command types on the command input device, unless a listing file is specified. Error messages are always typed on the command input device.

If the operator enters (K) while the text editor is typing a message that does not require a response, the text editor ignores the interruption and proceeds with the next line of output. This protects the text editor from being accidentally aborted. If the operator enters (K) while the text editor is typing a message that does require a response, the text editor repeats the message. If the text editor is interrupted five consecutive times while trying to

write the same message that requires a response, the text editor terminates and the message END OF JOB appears on the console.

The following dialogue represents system typeouts and operator typeins that may appear on the console after the parameter input line has been typed in and the text editor types out a "?" to solicit command input:

```
xx ZEDIT      ? (P)xx<command>(CR)
xx ZEDIT      ? (P)xx<input line>(CR)
:
xx ZEDIT      ? (P)xx<command>(CR)
ZEDIT message
xx ZEDIT      ? (P)xxQUIT(CR)
ZEDIT  END OF JOB
```

<command> - Text editor command. See Table 4-1.

<input line> - One or more lines to be put in the file may be typed after an INSERT command.

<message> - Messages output by the text editor may be:

- Verification output - e.g., a line from the edited file containing 'string', typed out in response to a LOCATE command.
- PRINT command output.
- END OF FILE - the text editor reached the end-of-file in processing the last command.
- Error message - See Table 4-2.

QUIT - Text editor termination command.

END OF JOB - Termination confirmed.

The following printout on a console illustrates sample operator typeins and program typeouts that may occur:

```
(P)$SA ZEDIT(CR)
ZEDIT      EDIT 700  REV E 75/06/15
00 ZEDIT    !(P) 00CSUB,CSUB2,OC(CR)
00 ZEDIT    ?(P) 00F /"DATE"/(CR)
ZEDIT      32 DATE\EQU\4023\JAN 23 1975
00 ZEDIT    ?(P) 00S.23.27.(CR)
ZEDIT      32 DATE\EQU\4027\JAN 27 1975
00 ZEDIT    ?(P) 00L.PCOU.,2(CR)
ZEDIT      164 \STA\PCOUNT\PAGE COUNT
ZEDIT      349 PRNTP\LDA\PCOUNT\PAGE COUNT
00 ZEDIT    ?(P) 00P 3(CR)
ZEDIT      349 PRNTP\LDA\PCOUNT\PAGE COUNT
ZEDIT      350 \STA\N\NUMBER FOR CONVERSION
ZEDIT      351 \JST\CONV\CONVERT BINARY TO BCD
00 ZEDIT    ?(P) 00D 349,350(CR)
00 ZEDIT    ?(P) 00I(CR)
00 ZEDIT    ?(P) 00PRNTP\IMA\N\SAVE AND GET OLD VALUE
00 ZEDIT    ?(P) 00!(CR)
00 ZEDIT    ?(P) 00Q(CR)
ZEDIT  END OF JOB
```

In the above example the operator used the text editor to modify a program file CSUB, and to produce a new version in the file CSUB2. Commands were submitted through a console used; i.e., the text editor ran in conversational mode. The operator first used the FIND (F) command to locate the coded string DATE ("is used as delimiter") on line 32 of the file, and the SUBSTITUTE (S) command to change it. The LOCATE (L) command was used to find the second occurrence of the variable PCOUNT in the file, the PRINT (P) command to check the contents of three lines, the DELETE (D) command to delete the first two of these three lines, and the INSERT (I) command to replace them with another line. The QUIT (Q) command terminated the text editor activity. Note the exclamation point typed in by the operator to terminate the insertion, and the line numbers (32, 164, etc.) output by the editor to show the position of the line in the original file.

Text Editor Commands

Text editor commands are listed in Table 4-1.

Table 4-1. Text Editor Command Summary

Format	Abbreviation	Function
COPY COPY i,j,k COPY i,j,* COPY ,j,k	C	Insert current line above current line. Insert a copy of lines i through j before line k. Insert a copy of lines i through j before end of file. Insert a copy of the current line through line j before line k.
DELETE DELETE i DELETE i,j DELETE ,j DELETE i,* DELETE ,*	D	Delete current line Delete line i. Delete lines i through j. Delete current line through line j. Delete lines i through end of file. Delete current line through end of file.
FIND "string" FIND i	F	Find next line commencing with string. Find line i.
INSERT (Note) INSERT i INSERT *	I	Insert before current line. Insert before line i. Insert at end of file
! comments	!	Terminate insertion.
LOCATE "string" LOCATE "string",i LOCATE "string",* LOCATE "string1... string2",i	L	Locate the next line containing string. Locate next i lines containing string. Locate all following lines containing string. Locate next i lines containing string1 followed by string2 (separated by any number of characters).

Table 4-1 (cont). Text Editor Command Summary

Format	Abbreviation	Function
LOCATE "string",i, j,k LOCATE "string",j LOCATE "string",,,k	L (cont)	Locate next i lines containing string between character positions j and k, inclusive. Locate the next line containing string starting at or after character position j. Locate the next line containing string ending at or before character position k.
NEXT NEXT i NEXT * NEXT -i	N	Move to next line. Move forward i lines. Move to end of file. Move backward i lines.
PRINT PRINT ,j PRINT i PRINT i,j PRINT * PRINT *,j	P	Print current line up to 80 characters. Print current line up to j characters. Print i lines starting at current line. Print i lines starting at current line, not more than j characters per line. Print from current line to end of file. Print from current line to end of file, not more than j characters per line.
QUIT	Q	Create new master and terminate EDIT 700.
READ file1,file2 READ n1,file1,n2, file2 READ *,file1,file2	R	Insert files file1, file2, etc. before current line. Insert file1 before line n1. Insert file2 before line n2, etc. Insert file1,file2, etc., at end of file.
SUBSTITUTE "string1 "string2" SUBSTITUTE "string1 "string2",i SUBSTITUTE "string1 "string2", SUBSTITUTE "string1 "string2",i,j,k SUBSTITUTE "string1 "string2",i1,j1,k1, "string3"string4" ,i2,j2,k2,"string5 "string6",i3,j3,k3, "string7"string8" ,i4,j4,k4,"string9 "string10",i5,j5,k5	S	Change the next occurrence of string1 to string2. Change string1 to string2 on the next i lines in which the string occurs. Change all following string1's to string2. Change string1 to string2 on the next i lines containing string1 between character positions j and k, inclusive. Change string1 to string2 on the next i lines containing string1 between character positions j1 and k1, inclusive. Perform the same process simultaneously for each of the other string pairs (up to 5), each pair having its own repetition factor and character limit. The command may not exceed one line in length.
TOP TOP R	T	Position at top of file. Reassign line numbers and position at top of file.
NOTE: The lines from the command input stream immediately following the INSERT command are inserted into the file being edited, until a line beginning with an exclamation point (!) is found in the command input stream.		

Text Editor Error Messages

Text editor error messages are typed out on the console. Table 4-2 lists the error messages, their causes, and results.

Table 4-2. Text Editor Error Messages

Messages	Type of Error	Cause
FILE TOO LARGE	Fatal ^a in batch mode ^a	The input file contains more than 32,000 lines.
SYNTAX ERROR		The most recent command line contains a syntactical error.
END OF FILE	Warning ^b	End of file encountered during command execution.
LINE NUMBER ERROR n		Line n does not exist.
LINE TRUNCATED		A SUBSTITUTE command was used to add a new string and resulted in a line containing more than 80 characters.
ILLEGAL FILE NAME	Fatal ^{c,d}	A file name has been specified (for the source input, source output or listing file) which does not conform to the rules for structuring a file name; i.e., a letter (A-Z) optionally followed by up to five letters (A-Z) and/or digits (0-9). The text editor is aborted.
INPUT FILE NOT FOUND		The specified source input file was not found in the disk directory. The text editor is aborted.
NULL INPUT FILE		The specified source input file is null; i.e., its name is present in the disk directory, but no disk segments are allocated to data. The text editor is aborted.
NO FREE CORE	Fatal ^c	No free core blocks are available for the text editor. The text editor does not use system free memory; it divides memory within its activity area into free memory blocks. This error indicates that the activity area configured for the text editor is not large enough.
DUPLICATE OUTPUT FILE	Fatal ^{c,d}	One of the source output files has been specified as having the same name as another file already present in the disk directory. The message is followed by the file name or device code, if applicable. The text editor is aborted.

Table 4-2 (cont). Text Editor Error Messages

Messages	Type of Error	Cause
SYSTEM ERROR A=aaaaaa X=xxxxxx	Fatal ^d	An error occurred on an executive function call. aaaaaa and xxxxxx are the contents of the A- and X-registers (in octal) when the error return was taken. (See Appendix E for significance of error codes.) The text editor is aborted.
DEVICE ALREADY RESERVED	Fatal ^{c,d}	One of the input or output devices has been reserved by another activity. The text editor is aborted.
^a Classified as fatal in batch mode; the text editor is aborted at that point. ^b Warning message only; the text editor is continued. ^c The text editor has been aborted and no new master will exist for these fatal error conditions. ^d The message is followed by the file name or device code, if applicable.		

FORTRAN-700 PROCEDURES

The FORTRAN translator translates FORTRAN symbolic source statements into DAP-700 assembly language source statements and creates listing files. (See Figure 4-3).

For more complete details concerning operation of the FORTRAN translator refer to the OS/700 FORTRAN manual.

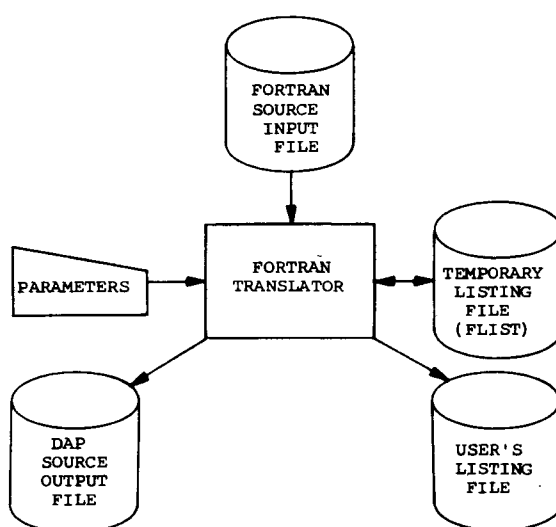


Figure 4-3. FORTRAN Translator Flow

FORTRAN Translator Execution

FORTRAN translator execution is initiated by a Start Activity (SA) command, which causes the FORTRAN translator activity to be scheduled. When the activity starts, it types out its identification, followed by an exclamation point, to solicit the input of a line of parameters.

The following dialogue represents system typeouts and operator typeins that may appear on the console.

```
(P)$SA ZFORT(CR)
ZFORT FORTRAN 700 REV n yy/mm/dd
xx ZFORT  ! (P)xx<fname1>,<fname2>,<fname3>,<opt1>,...,<optn>(CR)
ZFORT      n FORTRAN ERRORS
ZFORT END OF JOB
```

\$SA - Start activity command.

ZFORT - Name of FORTRAN translator activity.

FORTRAN 700 REV n yy/mm/dd - Identifies the version (revision number and date) of FORTRAN-700.

xx - 2-digit message identification number; this number must precede the operator's response.

! - Message soliciting translator parameter line.

<fname₁> - Name of the FORTRAN source input file to be translated.

<fname₂> - Name of the DAP source output file to be generated by the translator (in ASCII format).

<fname₃> - Name of the listing file to be generated by the translator (in listing format).

<opt₁>,...,<opt_n> - FORTRAN translator options - see Table 4-3.

n FORTRAN ERRORS - Number of FORTRAN syntax errors, n; printed for each FORTRAN source in the source input file.

END OF JOB - FORTRAN translation complete.

File names must appear in the order specified; optional parameters can be specified in any number or order after the listing file name (fname₃). Files must not be named FLIST, neither must there be a file named FLIST present in the default library on the disk when the translator is being executed; the translator uses this name for a temporary work file during translation. If either <fname₂> or <fname₃> is omitted, output to that file is suppressed.

If the operator types control-K (K) on the console during the parameter solicitation message (xxZFORT !), the FORTRAN translator types END OF JOB and terminates. If the operator interrupts the FORTRAN translator during any other console typeout, the translator ignores the interruption and continues.

The following dialogue represents a sample printout that may appear on the console during a typical session with the FORTRAN translator.

```
(P)$SA ZFORT(CR)
ZFORT FORTRAN 700 REV 310 75/12/04
00 ZFORT  ! (P) 00PROG3,LPROG3,MO,NS,EL(CR)
ZFORT    11 FORTRAN ERRORS
ZFORT    END OF JOB
```

In this example, the operator used the FORTRAN translator to translate a FORTRAN source program in a file, PROG3, producing the DAP source output file, DPROG3, and a listing of FORTRAN errors (EL option) in file LPROG3. Comments are not written in the DAP source output file, thus keeping it as short as possible (MO option). The translator does not write code to check array subscripts when the program PROG3 is run (NS option). The program in the input file PROG3 had 11 FORTRAN syntax errors.

FORTRAN Translator Options

FORTRAN translator options are listed and explained in Table 4-3.

Table 4-3. FORTRAN-700 Translator Options

Option	Description
BSD=<number>	Decimal number ranging from 0 (default) to 511, indicating the size of the block in each sector the translator will use for BSD's.
LT	Label trace is to be included for run time.
FT	Full trace is to be included for run time.
NS	No subscript checking on array bounds is to be performed.
NO	No optimization is to be performed on expressions containing calls on external functions.
EL	Listing file is to include errors only.
LO	Listing file is to include DAP source language.
MO	The source output is to be minimized by eliminating comments.
CO	Processing is to continue even if errors are detected in the FORTRAN source.
W,X,Y or Z	Translate those source records marked with the same conditional translation characters.

FORTRAN Translator Error Messages

FORTRAN translator error messages issued on the console are listed in Table 4-4.

Table 4-4. FORTRAN Translator Error Messages

Error Message	Description
ERR ABORT	More than 100 errors detected in the program unit. Translation is aborted.
ILLEGAL CHARACTER	An illegal character, e.g., a control character, has been found in the parameter list. Translation is aborted.
ILLEGAL PARAMETER	An error in the parameter list has aborted the translation.
TCMI ERROR A=aaaa B=bbbb X=xxxx TCM ERROR A=aaaa B=bbbb X=xxxx TCMC ERROR A=aaaa B=bbbb X=xxxx	<p>An error in the Transient Code (overlay) manager occurred: in the initialization routine (TCMI), in the overlay fetching routine (TCM), or in the closeout routine (TCMC). The contents of the A-, B- and X-registers are printed in decimal. The contents of the A-register (aaaa) indicate the type of error as follows:</p> <p>aaaa = 0 (TCMI)</p> <p>The FORTRAN translator activity could not identify itself in the disk directory (IDN\$ error).</p> <p>aaaa = 1 (TCMI)</p> <p>The activity could not obtain the system parameters (GSP\$ error).</p> <p>aaaa = 2 (TCMI)</p> <p>The activity could not reserve the system disk (RSV\$ error).</p> <p>aaaa = 3 (TCMI, TCM)</p> <p>The activity could not request input from the system disk (INP\$ error).</p> <p>aaaa = 4 (TCMI, TCM)</p> <p>The activity detected an error status after reading from the system disk.</p> <p>aaaa = 5 (TCMI, TCM)</p> <p>The activity read a segment from the system disk, but the input range was wrong.</p> <p>aaaa = 6 (TCMC)</p> <p>The activity could not release the system disk (REL\$ error).</p> <p>For errors 0, 1, 2, 3 and 6, the B-register contents (bbbb) will be the error code returned in the A-register by the executive function, and the X-register contents (xxxx) will be the secondary error code returned in the X-register by the executive function. See Appendix E for an explanation of bbbb. For errors 4 and 5, the X-register contents (xxxx) will be the hardware status word (word 6 of the status block). For error 4, the B-register contents (bbbb) will be the software status word (word 1 of the status block), and for error 5, bbbb will be the range shortfall (required range minus actual range). Refer to Appendix F for status block words. Translation is aborted.</p>

Table 4-4 (cont). FORTRAN Translator Error Messages

Error Message	Description
ILLEGAL FILE NAME	A specified file name (for the source input, source output, or listing file) does not conform to the normal rules for the structure of a file name; i.e., a letter (A-Z) optionally followed by up to five letters (A-Z) and/or digits (0-9), or the user has specified a file name FLIST. Compilation is aborted.
I/P FILE ABSNT	The specified source input file was not found in the disk directory. Compilation is aborted.
DUP O/P FILE	The source output or the listing file has been specified as having the same name as another file already present in the disk directory, or a file named FLIST exists on the disk. Compilation is aborted.
NULL I/P FILE	Specified source input file is NULL; i.e., its name is present in the disk directory, but it contains no data. Compilation is aborted.
SYS ERR nnnn	nnnn (decimal) specifies the OS/700 executive function call error code. (Refer to Appendix E for significance of error codes.) FORTRAN compilation is terminated.

DAP-700 MACRO ASSEMBLER PROCEDURES

The DAP-700 macro assembler is a 2-pass language processor that translates symbolic source program statements into machine language instructions and generates an object program and an expanded listing. It operates in conversational mode only. (See Figure 4-4.) Refer to OS/700 DAP-700 Macro Assembler manual for further details.

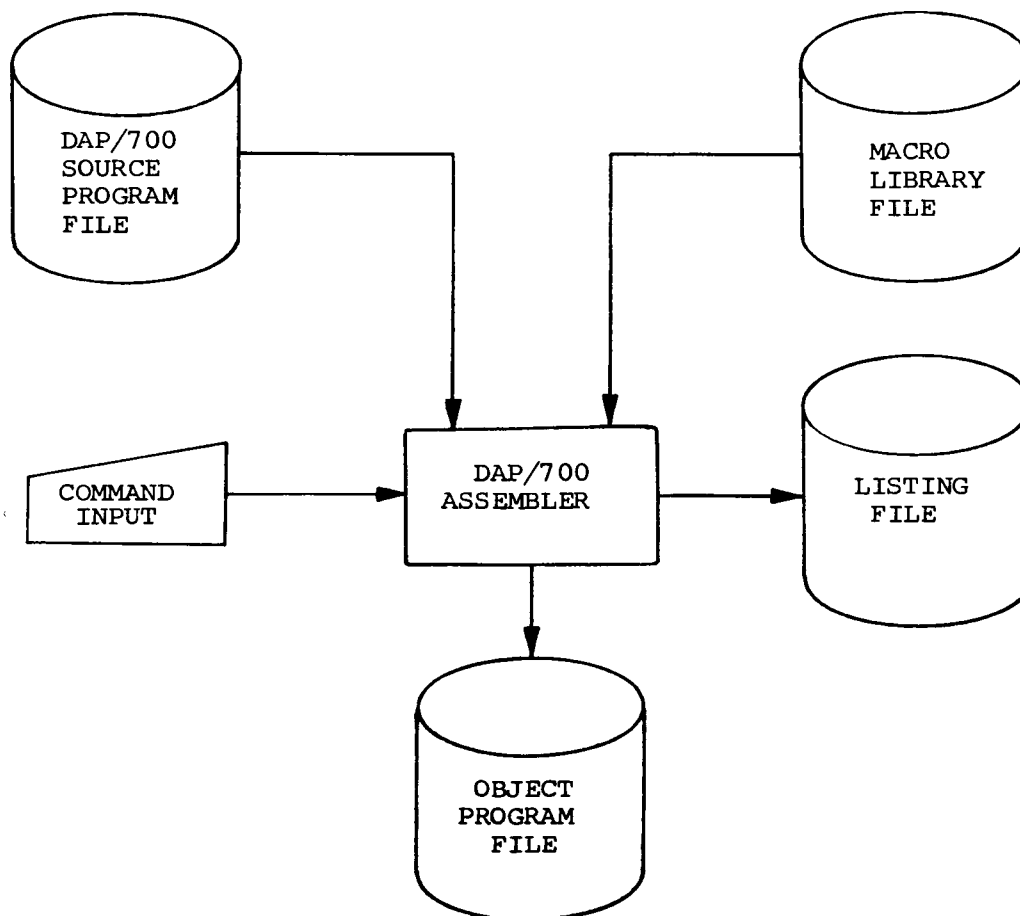


Figure 4-4. DAP-700 Macro Assembler Flow Diagram

Assembler Execution

DAP-700 assembler execution is initiated by a Start Activity (SA) command, which causes the assembler activity to be scheduled. When the activity starts, it types out its identification, followed by an exclamation point to solicit the first assembler command.

The following dialogue represents system typeouts and operator typeins that may appear on the console:

```

(P)$SA ZAP(CR)
ZAP   DAP-700  REV. n yy-mm-dd
xx ZAP   ! (P)xx<comd > <opt1>,<opt2>,...,<optn>(CR)
ZAP   EOP1
ZAP           ASSEMBLY COMPLETED
ZAP           xxxx WARNING OR ERROR FLAGS
xx ZAP   ! (P)xxQUIT(CR)
ZAP   END OF JOB
  
```

\$SA - Start activity command
 ZAP - Assembler activity name
 DAP-700 REV } - Identifies the version (revision number
 n yy-mm-dd } and data) of DAP-700
 xx - 2-digit message identification number;
 this number must precede the operator's
 response
 ! - Message soliciting an assembler command
 <comd> - Assembler command }
 <opt₁>,<opt₂>,...,<opt_n> - Assembler options } See Table 4-5
 EOPl - End of pass 1 message
 ASSEMBLY COMPLETED - Assembler confirmation for each module
 xxxx WARNING OR }
 ERROR FLAGS } - Error summary of preceding assembly
 QUIT - Terminate assembler command
 END OF JOB - Assembler termination

The following sample dialogue illustrates system typeouts and operator typeins necessary to assemble a macro library:

```

xx ZAP      ! (P)xxMCRL MF=MACFYL(CR)
ZAP      EOPl
ZAP      ASSEMBLY COMPLETED
ZAP      0000 WARNING OR ERROR FLAGS
xx ZAP      !
  
```

The assembler then awaits another assembler command. The following sample operator typeins may then be entered to assemble a source program:

```

(P)xxASSM SF=MAC99,LF=LMAC99,NO,RM,EL(CR)
ZAP      EOPl
ZAP      ASSEMBLY COMPLETED
ZAP      0039 WARNING OR ERROR MESSAGES
xx ZAP      !
  
```

This dialogue causes the source module MAC99 to be assembled with relocatable mode as a default. An error-only (EL option) listing file named LMAC99 is generated; no object file is produced.

DAP-700 Macro Assembler Commands

Table 4-5 contains a summary of DAP-700 Macro Assembler commands and optional command parameters.

Table 4-5. Assembler Command Summary

Actions		Description
Commands	ASSM	Start assembly
	MCRL	Enter and transfer macro library
	RMLP	Reset macro library pointers
	QUIT	Terminate assembly process
Options	LF=	<filename> Define listing file
	OF=	<filename> Define object file
	SF=	<filename> Define source file
	MF=	<filename> Define macro source file (for use with MCRL)
	SM	Assemble with system macros (automatic MCRL assembly of MACLIB, followed by the speci- fied ASSM)
	CD	List conditionals
	EL	Error only listing
	RM	Relocatable mode as default
	FL	Full listing (overrides NLST)
	MD	Suppress macro definitions on listing
	NM	Suppress symbol table map on listing
	NE	Suppress macro expansions on listing
	NO	Suppress object file
	NL	Suppress listing
	UF	Use assembler function pre-scan as default
	PF	Produce assembler functions in listing
S1,S2,S3, or S4		Pseudo sense switch settings

DAP-700 Macro Assembler Diagnostic Messages

Table 4-6 lists diagnostic messages that the DAP-700 Macro Assembler may issue on the console.

Table 4-6. DAP-700 Macro Assembler Diagnostic Messages

Message	Assembler Action After Message
DAP-700 REV. n yy-mm-dd	Continues in command processor
!	Awaits user input of command line
NON-ASSM. COMMAND	Resolicits command line from user
WRONG RECORD LENGTH	Source record too large, assembler terminates
SOURCE FILE UNSPECIFIED	Resolicits command line
END OF JOB	On QUIT, terminates
UNRECOGNIZABLE OPTION	Resolicits command line
ASSEMBLY COMPLETED	Pass wrap-up, prints error summary
LIST FILE UNSPECIFIED	Suppresses list file, continues assembly
OBJECT FILE UNSPECIFIED	Suppresses object file, continues assembly
SPURIOUS EOF	Handles as END statement
DISK DRIVER ERROR (status) ^a	TCM, TCMI or disk storage problem, assembly terminates
LOGICAL I/O ERROR (status) ^a	Logical I/O detected, assembly terminates
BAD CONFIGURATION (TCG No.)	TCMI detected, assembly terminates
DISK MANAGER ERROR (status) ^a	TCMI detected, assembly terminates
DEALLOCATE ERROR	Disk storage problem, assembly terminates
ALLOCATE ERROR	Disk storage problem, assembly terminates
NON-EXISTENT SOURCE FILE	Resolicits command
DUPLICATE LIST FILE NAME	Resolicits command
DUPLICATE OBJECT FILE NAME	Resolicits command
NULL INPUT FILE	Resolicits command
DICTIONARY OVERFLOW, LINE NO. xxxx, PASS n	Continue assembly
UNABLE TO TERMINATE ZAP	Repeats attempt to terminate the activity
SYMBOL TABLE OVERFLOW, LINE NO. xxxx, PASS n	Continue assembly
xxxx WARNING OR ERROR FLAGS	Summary of error status for preceding assembly, solicits new command
EOP1	End of pass 1, begins pass 2 for ASSM assemblies; message completion for MCRL assemblies
^a (status) is the contents of word 1 of the status block associated with I/O. Appendix F contains an explanation of the bits.	

LINKED-700 LINKAGE EDITOR PROCEDURES

The linkage editor links DAP-700 object text modules, in either batch or conversational mode, and produces a link text file. When the linkage editor is executed in batch mode, command input is from a disk resident source file; in conversational mode, input is from a console. (See Figure 4-5.)

For further details concerning the use of the linkage editor, refer to the LINKED-700 Linkage Editor manual.

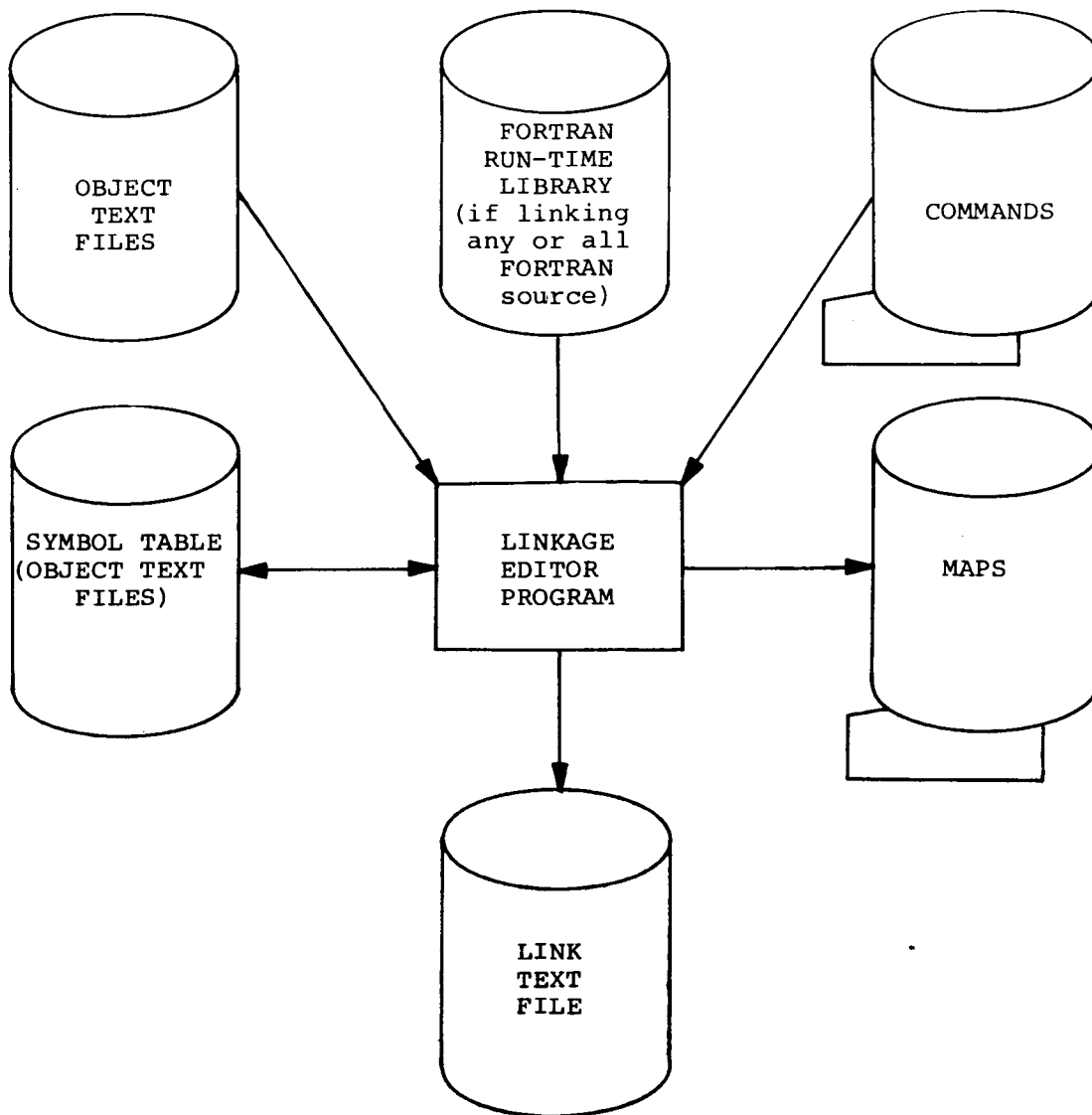


Figure 4-5. Linkage Editor Flow

Linkage Editor Execution

Linkage editor execution is initiated by a Start Activity (\$SA) system command which causes the linkage editor to be scheduled. When the activity starts, it types out its identification, followed by an exclamation point to solicit the first linkage editor command.

The following sample dialogue represents system typeouts and operator typeins that may appear on the console:

```

(P) $SA ZLE(CR)
ZLE LE-700 REV. n yy/mm/dd
xx ZLE ! (P) xx<command1>, ...<commandn>[Δcomment] (CR)
⋮
xx ZLE ! (P) xxQUIT(CR)
  
```

\$SA - Start activity command
 ZLE - Linkage Editor activity name
 ZLE LE-700 REV. } - Identifies the version (revision number and date)
 n yy/mm/dd } of LINKED 700
 xx - 2-digit message identification number; this
 number must precede the operator's response
 ! - Message soliciting a string of linkage editor
 commands
 <command₁>, }
 ...<command_n> } - Linkage editor commands. See Table 4-7.
 [Δcomment] - Any desired comment (optional). The space is a
 line terminator and the linkage editor ignores
 anything between the first space, Δ, and the (CR).
 QUIT - Terminate linkage editor command.¹

The command, CI=<filename>, may be used at any time to put the linkage editor into batch mode. The command file, <filename>, must be an ASCII file (such as those created by the EDIT-700 text editors) whose lines have the format:

<command₁>, ...<command_n> <comment> (CR)

The linkage editor processes the commands in the file until it reaches a:

* QUIT - Linkage editor terminates
 CI } - Linkage editor returns to conversational mode and
 End-of-file } solicits another command line.

Linkage Editor Commands

Table 4-7 contains a summary of linkage editor commands.

¹If command processing is inhibited by an IF (see Table 4-7), nothing will happen when QUIT is typed. Typing ENDC one or more times will enable command processing and QUIT can be retyped.

Table 4-7. Linkage Editor Commands

Command	Syntax	Description
Address	ADDR=num	Sets the current address; i.e., the address at which the linkage editor will start to link the next module.
BASE	BASE=num1<num2> or BASE=num1	Defines the base sector and establishes a primary desectorization area in it.
Assign object text binary input stream	BI=file name ^b	Assigns a binary input stream to an OS/700 object text file.
Assign link text binary output stream	BO= file name ^b	Assigns the link text binary output stream to an OS/700 file.
BSD	BSD=num	Generates a secondary desectorization area (BSD) of <num> words, starting at the current location.
Assign ASCII command input stream	CI or CI=file name	Assigns the command input stream to the console or an OS/700 ASCII file.
Identification with copyright	CIDNT: character string ^c	Places character string and Honeywell copyright in link text identification blocks.
COMMON address	COMM=num	Defines the upper limit of the COMMON storage area; FORTRAN and DAP COMMON blocks will be assigned storage in a downward direction, starting at location <num-1>.
CULL mode	CULL	Invokes CULL mode, in which a symbol defined by a module or by a DEF command is recorded in the symbol table only if it was previously referenced but not yet defined.
Definition of symbol	DEF:symbol=num	Defines the numeric value (num) of symbol
Reverse condition of command execution	ELSE ^a	Reverses effect of preceding IFZ or IFN command; e.g., if command execution was enabled, it becomes inhibited.
End conditional command execution	ENDC ^a	Removes condition imposed upon command execution by preceding IFZ or IFN command.
Enter extended desectorization mode	EXD	Invoke extended desectorization mode for code which will execute in extended addressing mode. (Support 32K).
Finish	FIN	Closes the link text output file and reinitializes the linkage editor, but preserves the values of the defined symbols.
FORCE	FORCE	Forces linking of the next object text module read.

Table 4-7 (cont). Linkage Editor Commands

Command	Syntax	Description
Gap table	GAPT or GAPT=num	Outputs to the link text file a table of unused memory areas.
Gap base	GBASE or GBASE=num	Ignores gaps below current link address or num when linking SECT mode object text modules.
Identification	IDNT:character string ^c	Places character string in link text identification block.
Conditional command execution	(IFN):num ^a (IFZ):num ^a	Inhibits execution of subsequent commands unless num is zero (IFZ) or unless num is nonzero (IFN); condition is in force until corresponding ELSE or ENDC is reached.
Initialize	INIT	Closes the link text output file, deletes all entries from the symbol table, and reinitializes the linkage editor.
Library mode	LIB	Invokes library mode, in which each object text module is linked only if it defines symbols which were previously referenced but not yet defined, or if the module is explicitly force-linked.
LINK	LINK ^b	Initiates the reading and processing of object text.
Leave extended desectorization mode	LXD	Invoke normal desectorization mode, for code which will execute in normal addressing mode (support 16K only).
MAP	MAP MAPS MAPF	Produces a link map.
Assign ASCII map output stream	MO or MO=file name	Assigns ASCII map output stream to console or to OS/700 file.
Modular origin	MORG=num	Advances link address to next location divisible by num.
NCULL mode	NCULL	Enters NCULL mode, in which a symbol defined by a module or by a DEF command is recorded in the symbol table if it was not previously defined in the current transient code group.
Normal mode	NORM	Invokes normal linking mode, in which the linkage editor returns for additional commands following a LINK command only when an end of file is encountered.
Assign binary object text output stream	00=file name	Assigns binary object text output stream to OS/700 file.

Table 4-7 (cont). Linkage Editor Commands

Command	Syntax	Description
QUIT	QUIT	Terminates linkage editor activity.
SKIP	SKIP or SKIP=num	Causes the linkage editor to advance through the object text input file by reading and ignoring the next module or <num> modules.
STEP mode	STEP	Invokes STEP mode, in which the linkage editor stops reading object text and continues command processing when the end of the first linked object text module or end of file is encountered.
Symbol table	SYMT	Outputs the symbol definitions in the symbol table to the object text output stream, as a single object module.
Establish transient code group	TCG=num	Specifies that the object text modules to be linked next form part of the transient code group (overlay) number <num>.
TOTAL mode	TOTAL	Invokes total link mode, in which every module read is force-linked.
<p>^aThe sequence of appearance of these commands must be IFZ (or IFN), ELSE (optional), and ENDC. If such a sequence is nested within another such sequence where command execution is inhibited, the entire sequence will be ignored.</p> <p>^bLINK and SKIP commands must be preceded by a BI stream assignment; the BO assignment must precede all LINK commands.</p> <p>^cThe character string may contain embedded spaces.</p>		

Linkage Editor Error Messages

When the linkage editor detects an error, it stops processing the commands in the command input line, issues a message in the following format, and then requests command input from the console.

```
ERROR xx{FILE=ffffff[CODE=cccccc]}
        {LINE=llllll}
```

xx - 2-letter error mnemonic.

ffffff - File name; printed for file-related errors (if xx is BL, DF, IO, RS, TD, TI, UC, or UF).

cccccc - OS/700 executive macro call error return code (decimal); printed if xx is IO, denoting I/O error. (See Appendix C.)

llllll - Line number (decimal) of the command input file being processed when the error occurred; printed only if command input file is being used, and xx is BO, CE, IA, MO, NT, TO, or US.

Table 4-8 lists and explains error messages, and suggests remedies.

Table 4-8. Linkage Editor Error Messages

Error Mnemonic	Type of Error	Meaning	Operator Action
BL	Block error	Object text has illegal format.	Do not continue linking. Verify that file is an object file (i.e., file was produced by assembler), or regenerate object file.
BO	Base sector overflow	Primary desectorization area has been filled. The linkage editor cannot desectorize an instruction because there is no desectorization area for the indirect address word generated.	Do not continue linking. Provide additional primary or secondary desectorization area, or rearrange order of linking to minimize cross-sector references.
CE	Command error	Syntax error or unrecognized command.	In conversational mode, re-type command on console. In batch mode, QUIT and correct contents of CI file.
DF	Duplicate file	File assigned to the MO, BO, or OO stream already exists.	Quit and delete old file or assign a different file to the stream.
IA	Illegal assignment	File is assigned to the BO stream after a LINK command.	Restart linking process and assign the BO stream before entering a LINK command.
IO	Input/output	Error detected by OS/700 during file handling.	Do not continue linking. See Appendix E.
MO	Memory overflow	Program code overwrites COMMON storage; i.e., HIGH exceeds COMM.	Do not continue linking. Reorganize linking process so any unused areas are used, or decrease memory requirements of one area.
NT	No transient code definition (TCD)	TCD assembly pseudo-operation was not processed before the first transient code group (TCG) command or assembly pseudo-operation (where TCG≠0) was encountered.	Do not continue linking. Be sure module with TCD pseudo-operation is linked before TCG command or pseudo-operation is encountered.
RS	Record sequence	Object text records are improperly sequenced.	Do not continue linking. Verify that file is an object file, or regenerate object file.
TD	TCD error	TCD assembly pseudo-operation in object text after object text which generates memory-image text.	Do not continue linking. Verify that module containing TCD pseudo-operation is linked first.

Table 4-8 (cont). Linkage Editor Error Messages

Error Mnemonic	Type of Error	Meaning	Operator Action
TI	Illegal object text type	Object text contains illegal data.	Do not continue linking. Verify that file is an object file, or regenerate object file.
TO	Symbol table overflow	Linkage editor's symbol table is fully utilized.	Do not continue linking. Use fewer symbols, or Rearrange linking process to minimize the number of unresolved references to external symbols or Generate a linkage editor activity that has a larger symbol table.
UC	Undefined COMMON block	Reference to an undefined COMMON block name. Data read through the BI stream does not conform to the definition of object text (see the <u>OS/700 DAP-700 Macro Assembler</u> manual).	Do not continue linking. Verify that file is an object file, or regenerate object file before referencing it in a linkage editor command.
UF	Undefined file	Undefined input file is assigned to the BI or CI stream. Attempt to read non-existent file.	Be sure to generate a file before using it. Verify file name entered through the console or CI stream.
US	Undefined symbol	Command parameter contains an undefined symbol.	Reorganize linking procedure, defining symbols before referencing them.

APPENDIX A
INITIALIZATION ERROR HALTS

The following error halts may occur during initialization. A halt, rather than an error message, is used in these cases because the system may not be able to process the error message. Error halts are referred to by symbolic location rather than by absolute address, since halt locations vary with system configuration. The symbolic location corresponding to the absolute address of the halts (displayed in the Y-register) may be determined by referring to the system link map.

Error Halt
Symbolic Location

Error Condition

ZISUPE

General initialization error. At the first halt at this location, the A-register contains the number of characters in the error status message.

Leave the STOP/RUN switch in the RUN position.

Press START.

The system advances to the next halt in a series of halts at ZISUPE, which displays the error-status message in the A-register. The error-status message is a maximum of 14 characters, the first of which is a number expressing the severity of the error, as follows:

- 1 - No errors recorded
- 2 - Marginal errors recorded
- 3 - Fatal error recorded

If the message has all 14 characters, then the error is fatal.

Each word following the first word of the message records one error. The left character of each word is reserved. The right character is a unique error code described below:

Error
Code

Error Description

- 1 No error recorded
- 2 Marginal errors recorded
- 3 Fatal error recorded
- 4 Reserved
- 5 Illegal timer requested
- 6 Illegal block size detected in GET block action routine
- 7 Illegal block size detected in return block action routine

Error Halt
Symbolic Location

Error Condition

ZISUPE
 (cont)

<u>Error Code</u>	<u>Error Description</u>
8	No more short blocks
9	No more long blocks
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	I/O device not in system
17	Unexpected I/O device ID
18	I/O device error
ZISUPD	End-of-error status message. All status words have been displayed (A-register = 0).
ZIXTAL	Crystal clock was not operational on the configured address.
ZISUPT	Error detected during attempt to print the initialization complete message via a TYP\$ function call. A-register contains error status. (See Appendix E.)
ZIDSCE	Error detected during disk initialization. The A-register contains the error status as follows:

<u>Error Code</u>	<u>Error Description</u>
1	Volume descriptor does not specify a removable disk or a fixed-head disk.
2	Disk read error in status returned upon I/O completion from an INP\$ function call.
3	Disk is not configured in the system.
4	Bad disk label.
5	Bad disk label, not system disk pack.
6	Not used.
7	Error in reserving disk during a RES\$ function call.
10	Disk read error in status returned upon I/O completion from an INP\$ function call.
11	Error detected during an attempted read via an INP\$ function call.
12	Not used.
13	Not used.
14	Error detected in releasing the disk via a REL\$ function call.

Error Halt
Symbolic Location

Error Condition

ZISURL	Error detected while attempting to return to the remote loader (communications systems only).
ZISUMP	Error detected during attempt to call the system message processor (communications systems only).

Any of these error halts indicate an abnormal situation in OS/700. The user may try to correct the situation, reload the system from disk, and retry the restart procedure.

APPENDIX B
SYSTEM MESSAGE FORMATS

1. Messages requiring responses:
 - a. xx actnam message
See documentation for the activity, actnam.
 - b. xx actnam {ENTER dd VOL NAME }
 {MOUNT nnnn ON dduu}
See Volume Manager messages, Section III.
 - c. xx actnam F ERR yy
See "FORTRAN Run-Time Error Messages," Appendix H.
2. Messages requiring no response:
 - a. actnam message
See documentation for the activity, actnam.
 - b. SE=message
See System Error Messages, Appendix C.
 - c. ***** actnam rr ['aaaaaa']
See Activity Abort Messages, Appendix D.
 - d. ***** mm
See \$LA or \$SA Messages, Section III.

APPENDIX C
SYSTEM ERROR MESSAGES

Three types of system error messages are possible in OS/700: I/O device errors, executive errors, and communications supervisor errors. These messages are inhibited if there is insufficient free memory.

I/O DEVICE ERRORS

An I/O device error generates a message with the following format on the system operator device:

SE=000000 dddddd

SE - Indicates an error message

000000 - 6-digit octal number that specifies the type of error

ddddd - 6-digit octal number that specifies the device that caused the error

An I/O error is identified by the high-order (leftmost) digit of 000000 equal to zero. To determine the source and cause of an error, proceed as follows:

1. Convert the octal number dddddd to a 16-bit binary number and divide it into two bytes. The left byte contains the generic device type, and is represented in Table C-1 as the number in parentheses in the "Device Type and Device in Error" column. The right byte contains the logical unit number, of the device in error, and is represented in the same column in Table C-1 as the "uu" value in the parentheses.
2. Interpret the value 000000, which specifies the type of error that occurred, by referring to the "Error Type" column in Table C-1.

Table C-1. I/O Error Codes and Meanings

Generic Device Number (Decimal)	Device Type and Device in Error (dddddd) (octal)	Error Type (000000) (octal)	Error Condition
0	KSR-33 teleprinter		None
1	Cartridge disk (4uu)	1	Missed interrupt
		2 ^a	Device not operational
		3	Missed data (transfer rate failure)
		4	Recovery error (miscellaneous)
		5	Protect error on OTP\$
		6	Controller busy
		7	DMA bus parity error
		10	Checksum error
		11	Segment not found
		12	Fixed volume missing
		13	No free memory block available
2	Paper tape reader (10uu)	1	Unit disabled due to hardware error
		2 ^a	Unit not operational - power off
3	Paper tape punch (14uu)	1	Unit disabled due to hardware error
		2 ^a	Not operational - power off
		3	Tape low
5	Card punch (24uu)	1 ^a	Missed interrupt
		2 ^a	Device not operational
		3	Punch check error
		4	Data access error
		7	No free memory block available for data conversion
		10	Controller failed to respond
		11	Retry failed
		12	Operator action timer timed out
6	Card reader (30uu)	1 ^a	Missed interrupt
		2 ^a	Device not operational
		4 ^a	Data access error
		5 ^a	Read cycle error
		6 ^a	Invalid Hollerith code
		7	No free memory block available for data conversion
		10	Unit disabled error (Type 5100 only) Controller failed to respond
		11	Stacker full or hopper empty (Type 5100 only) Retry failed
		12	Registration error (Type 5100 only) Operator action timer timed out
		13	Correct column option settings

Table C-1 (cont). I/O Error Codes and Meanings

Generic Device Number (Decimal)	Device Type and Device in Error (dddddd) (octal)	Error Type (0eeee) (octal)	Error Condition
7	Fixed-head disk (34uu)	1 ^a	Missed interrupt
		2 ^a	Device not operational
		3	Access error
		4	Recovery error
		5	Protect error
		7	Parity error
8	Removable disk (40uu)	1 ^a (DMC only)	Missed interrupt
		2 ^a	Device not operational
		3	Missed data
		4	Recovery error
		5	Protect error
		6 ^a (DMC only)	Controller busy
		7	Bus parity error
		10	Checksum error
9	Line printer (44uu)	1	Unit disabled due to hardware error
		2 ^a	Device not operational
10	Magnetic tape unit (50uu)	1	Missed interrupt
		2 ^a	Device not operational
		3	Operator failed to make device operational
		4	Operator failed to permit writing
		5 ^a	Writing not permitted
		6	Controller busy
		7	Write parity error
		10	Read parity error
11	Cassette tape	1	Missed interrupt
		2 ^a	Device not operational
		3	Operator failed to make device operational
		4	Operator failed to permit writing
		5 ^a	Writing not permitted
		6	Controller busy or not operational
		7	Write parity or access error
		10	Read parity or access error
12	ASR-33 teleprinter		None
13	ASR-35 teleprinter		None
^a The system allows the operator several minutes to recover from this error.			

EXECUTIVE ERRORS

Executive errors have two formats:

SE=1ffffff ssssss

SE=1ffffff ssssss actnam

SE - Indicates system error

1ffffff and ssssss - 6-digit octal numbers

actnam - 6-character ASCII string specifying the name of the activity that was executing or requested when the error occurred.

Executive errors are indicated by the high-order (leftmost) digit of 1ffffff equal to 1. The activity name is printed only for certain errors, which are specified in Table C-2.

In all cases, ssssss, which is called the second error code indicator, is meaningful only for certain errors, as shown in Table C-2. When the second indicator is not meaningful, six zeros are printed.

Table C-2. Executive Error Codes and Meanings

Error Code	Meaning
100001	In a COS, indicates that \$SA or \$LA command was being executed when another such command was entered. Wait for present function to be completed before typing another command. In a DOS, system failed to schedule an activity as requested by \$SA command, either because activity did not exist or because error occurred in scheduling process or while under CI mode, a command other than \$TR was typed on the console. Second error code indicator contains the ASCII characters "OI" ('147711).
100002	Operator typed a line which, subsequent to the initial (P), had neither a dollar sign (indicating system command) nor a valid message number (one associated with an unanswered message). The line is ignored. Second error code indicator contains the ASCII characters "OI" ('147711).
100003	Console I/O error occurred during operator typein; usually means operator waited too long to complete typing in a line once the (P) was typed. The line is ignored. Second error code indicator contains the ASCII characters "OI" ('147711).
100004	Operator's response to system or activity message contained too many characters. The line is ignored. Second error code indicator contains the ASCII characters "OI" ('147711).
100005	The text following a typein of (P)\$ was invalid system command. (The set of valid system commands varies. In a nondedicated COS, \$SA and \$LA are valid. In a DOS, \$SA is valid. If command input mode is configured, \$CI and \$TR are valid. If system integrity is configured, \$AB is valid. Dedicated COS recognize no system commands.) Second error code indicator contains the ASCII characters "OI" ('147711).
100006	The abort activity request which was made by the operator (\$AB) is invalid because either the activity is nonrestricted or the activity was not requested. Second error code indicator contains the ASCII characters "OI" ('147711).

Table C-2 (cont). Executive Error Codes and Meanings

Error Code	Meaning
100010	The file or activity could not be deleted because of deallocation error. The volume name is also specified.
100013	No free memory available to disk initialization.
100021	No work area available for allocation in either system or user area on volume. Operator should write "volume full" or "volume user area full" on appropriate disk volume.
100022	Disk error during disk initialization. Second indicator specifies disk unit number.
100024	Overlay cannot be read into main memory. Second indicator contains starting segment number of desired overlay.
100030	Activity area overrun. Activity name is specified with error indicators.
100031	Activity supervisor disk error while reading activity. Name of activity is specified.
100116	Specified activity name not found in the activity directory. Second indicator contains function number of Connect Clock Activity function. Activity name also specified.
100122	Disk error while referencing the disk activity directory. Second indicator contains function number of Connect Clock Activity function. Activity name also specified.
100126	No activity area for the named activity (the main-memory starting address given in the disk directory is not equal to the beginning of any activity area) or the activity is too large to fit into the allocated activity area (i.e., the activity ending address exceeds the activity area ending address). Second indicator contains function number of Connect Clock Activity function. Activity name also specified.
100163	No work area available for allocation in user area on volume. Operator should write "volume full" or "volume user area full" on appropriate disk volume.
100210	CI failed to get the next line in the CI command file and CI mode terminated. The GET\$ function took the error return and 'ssssss' contains the A-register setting. See Appendix E, Executive Function Call Error Codes.
100211	CI was unable to open the file specified in the \$CI command and CI mode terminated. The OPN\$ function took the error return and 'ssssss' contains the A-register setting. See Appendix E, Executive Function Call Error Codes.
100212	If 'ssssss' contains OI in ASCII ('147711'), a \$SA command attempted to start a second activity under CI control and CI mode terminated. If 'ssssss' does not contain OI, CI was unable to schedule the activity in a \$SA command in the command file. The SAC\$ function took the error return and 'ssssss' contains the A-register setting. See Appendix E, Executive Function Call Error Codes.
100213	CI was unable to close the CI command file and CI mode terminated. The CLS\$ function took the error return and 'ssssss' contains the A-register setting. See Appendix E, Executive Function Call Error Codes.

Table C-2 (cont). Executive Error Codes and Meanings

Error Code	Meaning
100214	A syntax error in a system command: <ul style="list-style-type: none"> ● Invalid command in a CI command file ● Filename following \$CI is too long ● Activity name following \$SA on console is too long ● No initial \$ in a command file line when no activity is running under CI control and a response line is not expected and CI mode terminated. 'ssssss' is OI in ASCII ('147711').
100215	The activity name following \$SA in a CI command file is too long and CI mode terminated. 'ssssss' is OI in ASCII ('147711').
100216	The response line in a CI command file is longer than the activity, running under CI control, expects and CI mode terminated. 'ssssss' is OI in ASCII ('147711').
100217	CI was unable to release the old output device while processing a \$OD command and CI mode terminated. The REL\$ function took the error return and 'ssssss' contains the A-register setting. See Appendix E, Executive Function Call Error Codes.
100220	CI was unable to reserve the new output device while processing a \$OD command and CI mode terminated. The RSV\$ function took the error return and 'ssssss' contains the A-register setting. See Appendix E, Executive Function Call Error Codes.
100221	The device specified in a \$OD command is not configured and CI mode terminated. 'ssssss' contains OI in ASCII ('147711').
100222	CI encountered a bad I/O status on return from the OTP\$ function when writing a record to the current output device and CI mode terminated. 'ssssss' contains word 1 of the physical I/O status block. See Appendix F, Physical I/O Device Information.
100223	CI was unable to output a record to the current output device and CI mode terminated. The OTP\$ function took the error return and 'ssssss' contains the A-register setting. See Appendix E, Executive Function Call Error Codes.

COMMUNICATIONS SUPERVISOR MESSAGES

The OS/700 Communications Supervisor issues system error messages on the console in two formats. They are: communications supervisor event reports (CSEVRT), and communications supervisor configuration errors (CSCNFE). Each is identified by the 6-letter code word in the activity name field of the system error message.

Event Report (CSEVRT)

A communications supervisor event report message reports changes in the status of the communications subsystem in the following format:

SE=xxxxxx yyyyyy CSEVRT

- xxxxxx is treated as a 16-bit binary number.
 Bit 1 indicates:
 0 - User program command response
 1 - Alarm condition response
 Bits 2 and 3 contain internal information.
 Bits 4 through 8 contain an octal message code specified in Table C-3.
 Bits 9 through 16 may contain additional information. See Table C-3.
- Interpretation of yyyyyy depends on the message code. See Table C-3.

Table C-3. Communications Supervisor Message Codes

Code Bits 4-8 (xxxxx)	Message Type	Parameter 2 CSEVRT (yyyyy)	Parameter 2 CSCNFE (yyyyy)	Parameter 1 Bits 9-16 (xxxxx)
01	Device State Change		DLT	
04	System Status	SSW	SSW	
06	Device Failure	DLT	LTA	
07	Free Core Alarm	SSW	SSW	
10	Line Status	LSW	DLT	
11	Terminal Status	TSW	TSW	
12	Format Error	Parameter 2 ^a		
13	Line State Change		DLT	
14	Terminal State Change		DLT	
15	Device Started	DLT	DLT	
16	Line Alarm	DLT	LTA	See Figure C-6.
21	Discipline Failure	DLT	LTA	See Figure C-7.
22	Device Looped	DLT	DTA	
23	Device Unlooped	DLT	DTA	
24	Line Poll Failure	DLT	LTA	See Figure C-8.
25	Line Select Fail	DLT	LTA	See Figure C-8.
26	Output Select Fail	DLT	LTA	See Figure C-8.
27	Device Status		DTA	
30	VIP Status	DLT	LTA	See Figure C-9.
DTA - Device Table Address LTA - Line Table Address DLT - Device/Line/Terminal Number (See Figure C-5.) SSW - System Status Word (See Figure C-4.) DSW - Device Status Word (See Figure C-1.) LSW - Line Status Word (See Figure C-2.) TSW - Terminal Status Word (See Figure C-3.)				
^a The original user parameter is given.				

Configuration Errors (CSCNFE)

A communications supervisor configuration error message has the format:

SE=xxxxxx yyyyyy CSCNFE

This message indicates that a processing request to the communications supervisor cannot be completed because a required communications supervisor function is not configured. The two octal numbers xxxxxx and yyyyyy contain the original communications supervisor parameters for calls that were not completed due to the unconfigured function. Values are interpreted as follows:

1. The xxxxxx is treated as a 16-bit binary number.
 - Bit 1 indicates that the requested function was for:
 - 0 - A communications supervisor command
 - 1 - An alarm condition processor
 - Bits 2 and 3 contain internal information.
 - Bits 4 through 8 contain an octal message code specified in Table C-3.
 - Bits 9 through 16 may contain additional information. See Table C-3.
2. Interpretation of yyyyyy depends on the message code. See Table C-3.

Status Word Formats

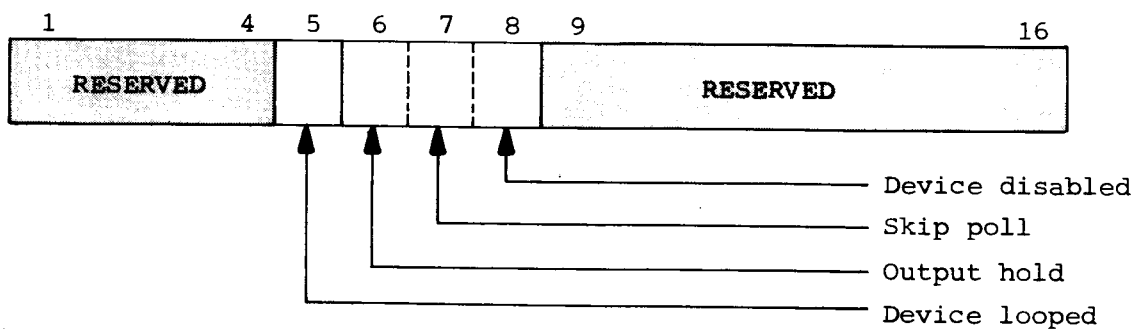


Figure C-1. Device Status Word

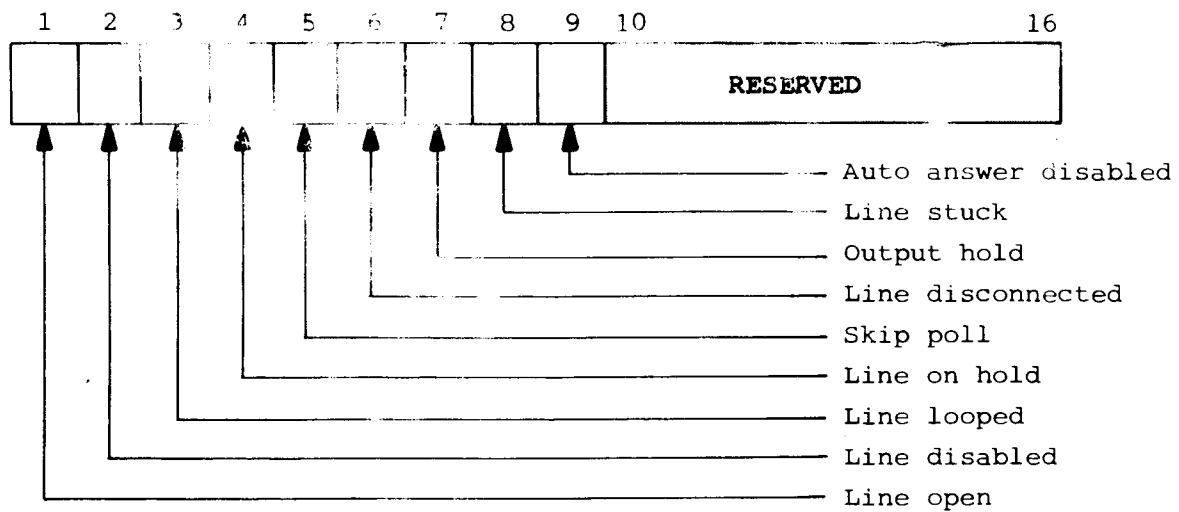


Figure C-2. Line Status Word

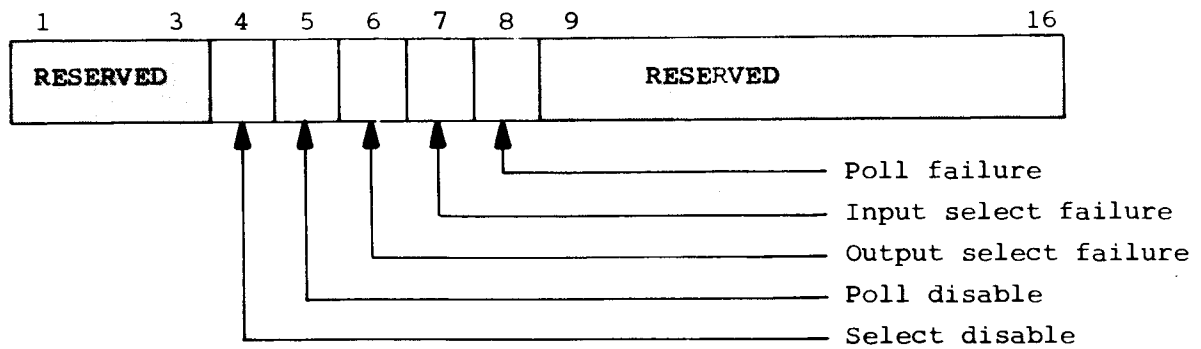


Figure C-3. Terminal Status Word

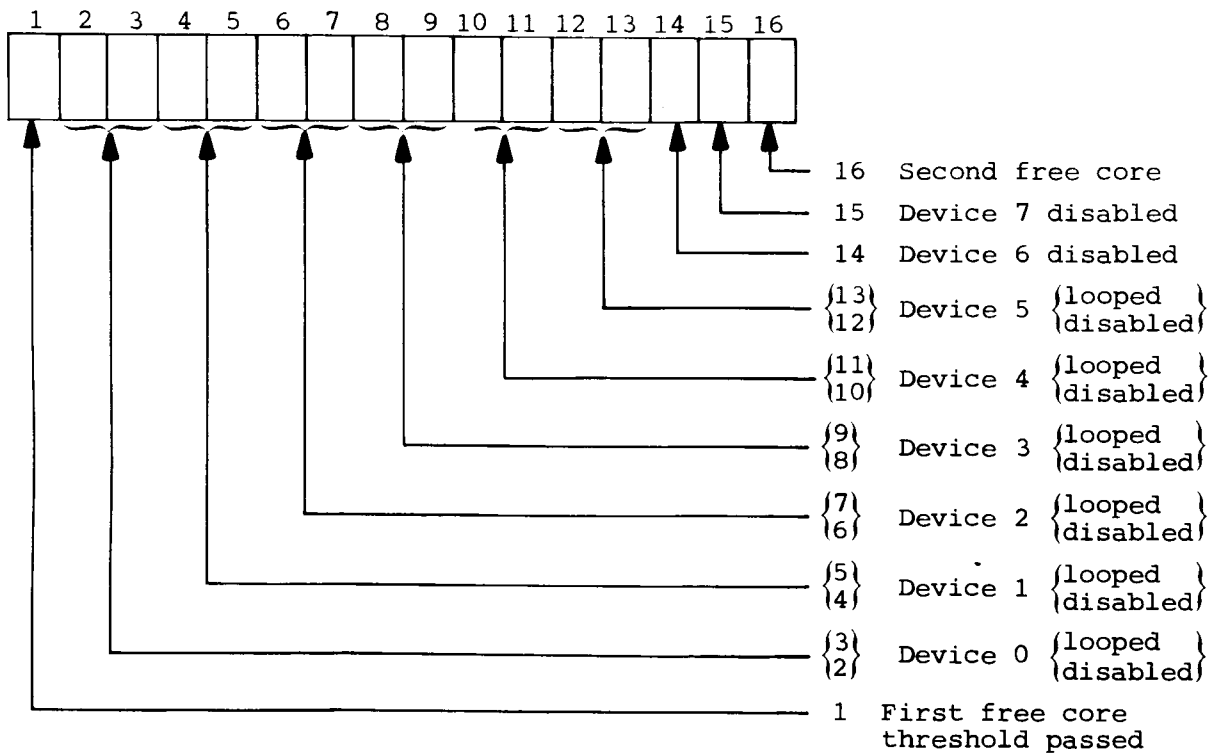


Figure C-4. System Status Word

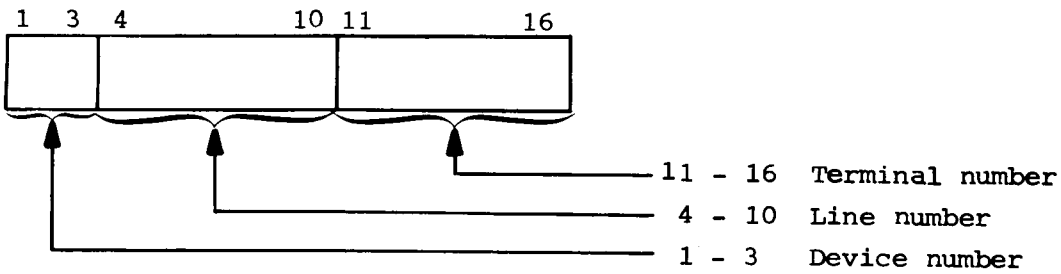


Figure C-5. Device/Line/Terminal Number Word

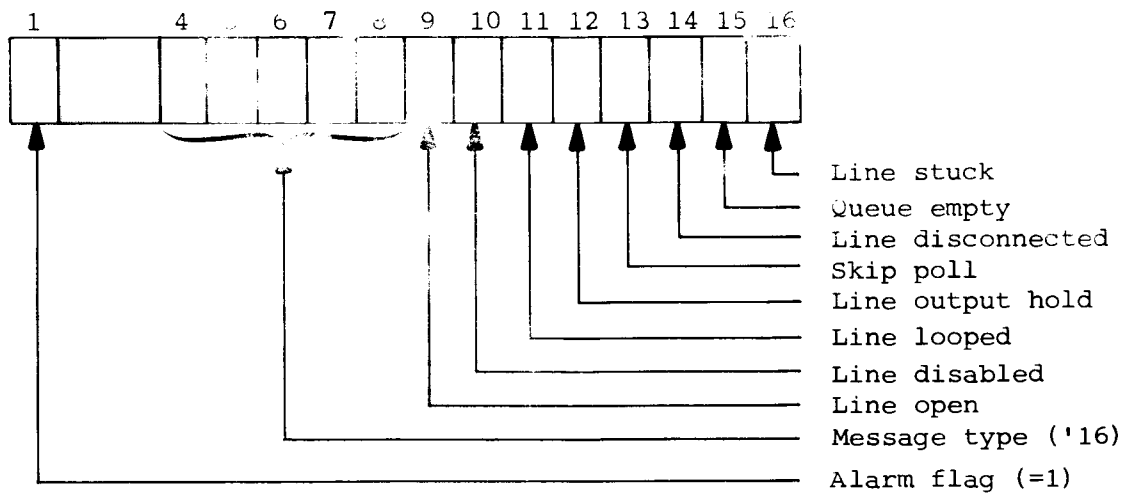


Figure C-6. Parameter 1 of Line Alarm Message

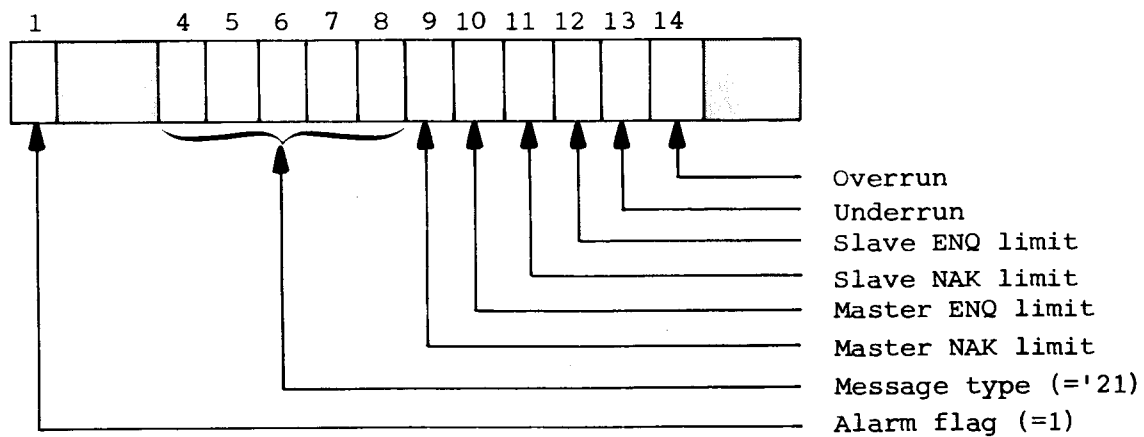


Figure C-7. Parameter 1 of Discipline Failure Message

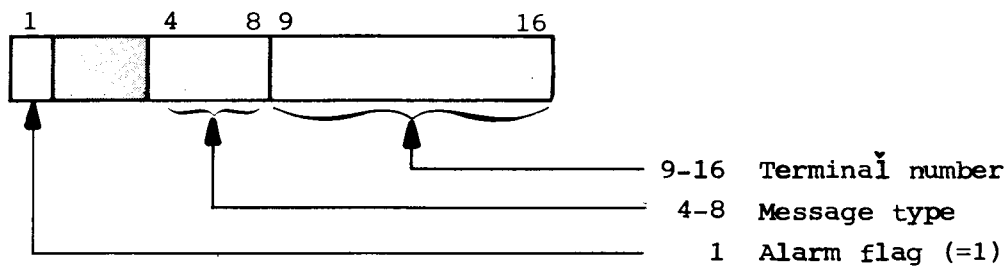


Figure C-8. Parameter 1 of Line Poll Failure, Line Select Failure-Output, Line Select Failure-Input Messages

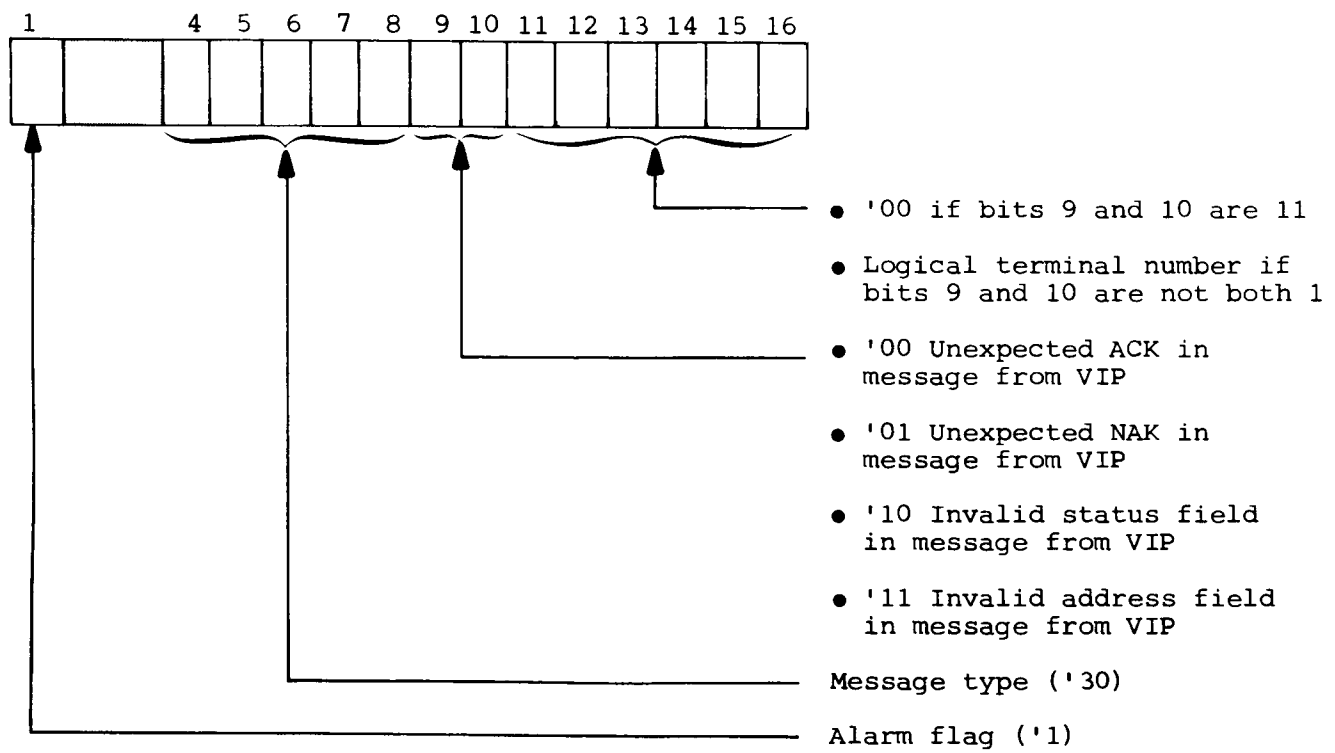


Figure C-9. Parameter 1 of VIP Status Message

APPENDIX D ACTIVITY ABORT MESSAGES

An activity abort message occurs whenever the system or the operator aborts a restricted activity. The format is:

***** <actnam> <rr> [<'aaaaaa>]

<actnam> - 1- to 6-character ASCII name of the restricted activity which was aborted.

<rr> - 2-character abbreviation of the reason for the abort.
See Table D-1.

<'aaaaaa> - Address printed if <rr> is MV, BP or IF.

Output of the message is suppressed if free memory is very low.

Table D-1 contains the reasons for the activity abort.

Table D-1. Reason for Abort

<rr>	Meaning
FC	Free memory is low.
OP	Abort was requested by the operator (\$AB command) or by a non-restricted activity (ABT\$ executive function call).
MV	<p>A memory lockout violation occurred:</p> <ul style="list-style-type: none"> ● An attempt to write in a protected area of memory (STA, DST, STX, LDX, IMA, IRS, and JST). ● An illegal instruction (HLT, INH, INA, IMK, OTA, OTK, OCP, SKS, SMK, and CAI). ● More than eight levels of indirect addressing. <p><'aaaaaa> contains the address where the memory lockout violation occurred.</p>
BP	<p>A bad parameter was passed to an action routine.</p> <ul style="list-style-type: none"> ● A word or block specified directly or indirectly by the parameter list does not reside entirely in the activity. ● The FCBB or LCBB pointer specified by the FCB or LCB is not the one given to the activity by the system when the activity opened the file or library. <p><'aaaaaa> contains the address of the function number of the executive function called.</p>

Table D-1 (cont). Reason for Abort

<rr>	Meaning
IF	<p>An illegal function was requested</p> <ul style="list-style-type: none"> • A TMT\$ request with no other task of the activity scheduled. • A TMA\$ request with a bad TCB. • A WIO\$ request with no I/O request pending or no queued reserve request waiting. • One or more I/O requests pending and the function requested is not: <ul style="list-style-type: none"> EOF\$, INP\$, OTP\$ RWD\$, SPF\$, SPR\$ ULD\$, WIO\$. • Not a permissible function. See Tables D-2, D-3.

The subset of executive functions which may be requested by Restricted Activities is listed in Table D-2.

Table D-2. Permissible Functions

Function	Meaning
ALC\$	Allocate a work area
ATQ\$	Attach entry to queue
CFP\$	Change file password
CLL\$	Close library
CLP\$	Change library password
CLS\$	Close file
CRL\$	Create library
CRQ\$	Create queue
CVL\$	Connect volume
DLC\$	Deallocate a work area
DVL\$	Disconnect volume
EOF\$	End of file
GDT\$	Get date and time
GET\$	Get a record
GSP\$	Get system parameters
GTQ\$	Get top entry from queue
INP\$	Input
OPL\$	Open library
OPN\$	Open file
OTP\$	Output
PUT\$	Put a record
REL\$	Release a device
RSV\$	Reserve a device
RWD\$	Rewind
SAC\$	Schedule an activity

Table D-2 (cont). Permissible Functions

Function	Meaning
SPF\$	Space file
SPR\$	Space record
STSS\$	Schedule task
SUSS\$	Suspend task
TMA\$	Terminate an activity
TMT\$	Terminate task
TPR\$	Type a message and input a response
TYP\$	Type a message
ULD\$	Unload (Rewind with automatic release)
WIO\$	Wait for I/O completion

Executive functions that may not be requested by Restricted Activities are listed in Table D-3.

Table D-3. Nonpermissible Functions

Function	Meaning
ABT\$	Abort activity
CCA\$	Connect clock to activity
CCL\$	Connect clock to task
CCSS\$	Change station status
CCST\$	Connect station
CDST\$	Disconnect station
CGCB\$	Get communications block
CGSS\$	Get station status
CRAR\$	Receive and reformat
CRAS\$	Reformat and send
CRCB\$	Return communications block
CREC\$	Receive
CSDC\$	Send control
CSND\$	Send
CTC\$	Create a task control block
CTMC\$	Terminate communication task
DCA\$	Disconnect activity from clock
DCL\$	Disconnect task from clock
GBL\$	Get storage block
RBL\$	Return storage block
SDT\$	Set date and time
STC\$	Schedule a task control block

APPENDIX E EXECUTIVE FUNCTION CALL ERROR CODES

When error returns are taken by executive function action routines, the A-register and possibly the X-register contain information describing the error. If the A-register contains a value less than '777, the X-register value should be ignored. However, if the A-register value exceeds '777, the X-register contains physical I/O status information; it is decoded in the following manner:

If A-register contents are:	Then the X-register contains:
'1nnn	Setup error from the I/O input or output request; see the error codes below for INP\$ and OTP\$.
'2nnn	Software status from word 1 of the I/O status block; refer to Appendix F for status information.
'3nnn	Hardware status from word 4 of the status block; refer to Appendix F for status information.
NOTE: nnn - Octal digits representing the real error code.	

Table E-1 indicates executive function call error codes, excluding those for communications.

Table E-1. Executive Function Call Error Codes

Error Codes (A-register)	Indication(s)
0	<ul style="list-style-type: none"> • A device is reserved under another user ID when an RSV\$ request is made. • A nonsharable device has already been reserved under the same user ID when another RSV\$ request is made. • No free memory blocks were available when a GBL\$ request was made. • Block size parameter specified in the RBL\$ parameter list is illegal.

Table E-1 (cont). Executive Function Call Error Codes

Error Codes (A-register)	Indication(s)
1	<ul style="list-style-type: none"> • The generic device type (GDT) specified in the device control block (DCB) is not configured. • Block size parameter specified in GBL\$ parameter list is illegal. • CCA\$ or CCL\$ request to connect an unconfigured absolute timer has been made.
2	Attempt to enable a device already enabled.
3	<ul style="list-style-type: none"> • The logical unit number specified in the device control block is not configured. • An issued TPR\$ request cannot be acknowledged because the message table is full.
4	<ul style="list-style-type: none"> • The mode indicator specified in the device control block is incorrect, i.e., <0 or >4, or is an invalid mode for the specified device when an INP\$ or OTP\$ request is issued. (Binary mode is specified for an ASCII only device, ASR-35 or KSR teleprinters, or line printer; verbatim mode only was configured for the card reader, and the mode requested is ASCII or binary.) • TYP\$ or TPR\$ request issued, but error was encountered before or during message output.
5	The range value specified in the INP\$ or OTP\$ parameter list is incorrect; i.e., <1 or >4095, or the I/O buffer crosses the 32K boundary between banks 1 and 2 (64K system only).
6	The number of records or files specified in the SPR\$ or SPF\$ parameter list for magnetic tape is zero. For cassette tape, the record or file number is illegal when negative or zero.
7	The function requested (SPF\$, SPR\$, EOF\$, RWD\$, or ULD\$) for the generic device type specified in the device control block is illegal.
10	The user ID specified in the device control block is incorrect for the REL\$, INP\$, OTP\$, SPF\$, SPR\$, EOF\$, RWD\$, or ULD\$ request.
11	An INP\$ request was issued for an output only device; i.e., paper tape punch, card punch, or line printer.
12	An OTP\$ request was issued for an input only device; i.e., paper tape reader or card reader.
13	No free memory block available for an entry in usage request queue (OPN\$).
14	RSV\$, INP\$, OTP\$, SPF\$, SPR\$, EOF\$, RWD\$, or ULD\$ request issued for a disabled device.
15	A restricted activity issued a REL\$, INP\$, OTP\$, SPF\$, SPR\$, EOF\$, RWD\$, or ULD\$ request for a device which was not reserved by the activity.
16	The file or activity name was not found in the disk directory (OPN\$ or SAC\$).

Table E-1 (cont). Executive Function Call Error Codes

Error Codes (A-register)	Indication(s)
17	The file or activity name already exists in the disk directory (OPN\$ or LA utility program).
20	No additional space is available, nor is there room for expansion to add a file or activity name to the disk directory (OPN\$ or LA utility program).
21	No work areas are available (ALC\$, OPN\$ to create a file, or PUT\$).
22	The request is illegal because the pointers to blocks (LCBB and LNB) used by the library and file managers are invalid for the specified library (CLL\$, CLP\$, CLS\$, CFP\$, GET\$, or PUT\$).
23	GET\$, PUT\$, CLS\$, or CFP\$ request issued for an unopened file, or the request is illegal for the mode specified when the file was opened.
24	<ul style="list-style-type: none"> • OPN\$ request issued for a null file. • The maximum record length address was not specified in the OPN\$ parameter list or in the file control block (FCB). • The maximum record length is 0 or negative for an OPN\$ request for a file that is to be created.
25	<ul style="list-style-type: none"> • The record buffer address was not specified in the FCB for a PUT\$ request. • The record buffer address was not specified in the FCB for a GET\$ request when the record length is greater than or equal to the segment length for fixed-length records, or the maximum record length +1 equals or exceeds the segment length for variable-length records.
26	There is no activity area configured for the requested activity, or the activity area(s) configured is too small for the requested activity (SAC\$).
27	GET\$ or PUT\$ request issued for a file opened in update mode, but the maximum record length address was not specified when the file was opened.
30	An RSV\$ request was issued for a device, the driver for which is disk resident; however, the driver could not be brought into memory, because its size exceeded the size of the activity area.
31	An RSV\$ request was issued for a device, the driver for which is disk resident; however, the driver could not be brought into memory, because there was a disk read error.
32	An INP\$, OTP\$, SPF\$, SPR\$, EOF\$, RWD\$, or ULD\$ request was issued for a device, the driver for which is disk resident, but the device was not reserved.
33	The requested activity could not be scheduled immediately (SAC\$).
34	The requested executive function is not configured.

Table E-1 (cont). Executive Function Call Error Codes

Error Codes (A-register)	Indication(s)
35	The requested executive function cannot be loaded from disk.
36	<ul style="list-style-type: none"> • A GET\$ or PUT\$ request was made for a file that has variable length records, but direct access was specified in the FCB. • A GET\$ or PUT\$ request was issued for a file that has variable length records, but the record length address in the FCB is zero or the FCB is short (4 words) and does not contain the record length address. • A PUT\$ request was issued for a file that has variable length records, but the record length specified by the record length address in the FCB is zero or minus one (the legal record length is 1 to 65,534).
37	The file type parameter in the user's file control block (FCB) indicates that the direct access method is desired; however, the record number address in the FCB is zero, or the record number itself is zero or negative. The legal range for record numbers is from 1 to 32,767 (GET\$ and PUT\$).
x040	Physical I/O error occurred when PUT\$ request was being processed.
x041	Physical I/O error occurred when OPN\$ request in update mode, CLS\$ request in delete mode, GET\$ request or PUT\$ request was being processed.
42	The record is invalid (GET\$).
43	The maximum number of records that may exist in a file (32,767) was exceeded (PUT\$).
44	Maximum number of work areas that may be allocated for a file was exceeded (PUT\$).
45	The CFP\$ request is illegal, because the I/O mode parameter specified when the file was opened was not update.
46	The nondefault library is not open (CLP\$, CLL\$, OPN\$, CLS\$, CFP\$, GET\$, or PUT\$).
47	A restricted activity issued a ULD\$ request with one or more I/O requests queued but not processed.
50	The OPN\$ request is illegal, because the password check failed. A file password exists in the file descriptor entry, but not in the user's FCB; or the file password in the user's FCB does not match the password in the file descriptor entry.
51	The number of work areas in the volume descriptor is less than two (OPN\$).
52	Maximum number of work areas that can be allocated for a file is specified as 0 in the volume descriptor (OPN\$).
53	Segment size specified in the volume descriptor is illegal (OPN\$).

Table E-1 (cont). Executive Function Call Error Codes

Error Codes (A-register)	Indication(s)
54	Either file cannot be opened in update mode, because it is being read or updated by another user (OPN\$), or requested file cannot be opened in input mode, because it is being updated by another user (OPN\$).
55	Delete file request is illegal (CLS\$).
x056	The last data segment could not be written correctly; the file was saved, but the record(s) in the last data segment contain(s) extraneous information (CLS\$).
x057	The control segment could not be updated properly; the file was saved, but records were lost. Only those records that were in the file when the control segment was last updated properly were saved (CLS\$).
x060	The request to save the file could not be accomplished; the file was deleted (CLS\$).
x061	The file could not be updated; it remains as it was prior to the open request (CLS\$).
x062	The data segment containing the record for the last PUT\$ request could not be written properly (CLS\$).
x063	Physical I/O error on system library directory segment input (CRL\$, OPL\$, CLL\$, CLP\$).
x064	Physical I/O error on system library directory segment output (CRL\$, OPL\$, CLL\$, CLP\$).
x065	Physical I/O error on volume library directory segment input (CRL\$, CLL\$, or CLS\$).
x066	Physical I/O error on volume library directory segment output (CRL\$, CLL\$, or CLS\$).
x067	Physical I/O error on file directory segment input (CRL\$, CLL\$, OPN\$, CLS\$, or SAC\$).
x070	Physical I/O error on file directory segment output (CRL\$, CLL\$, OPN\$, or CLS\$).
x071	Physical I/O error on file descriptor directory segment input (OPN\$, CLS\$, or SAC\$).
x072	Physical I/O error on file descriptor directory segment output (OPN\$, or CLS\$).
73	No more library space in the system library directory (CRL\$).
74	No available chunk in the volume library directory or in the file directory (CRL\$ or OPN\$).
75	Library already exists in the system library directory (CRL\$).
76	Library already exists in the volume library directory (CRL\$).

Table E-1 (cont). Executive Function Call Error Codes

Error Codes (A-register)	Indication(s)
77	Library not found in the system library directory (CLL\$ or CLP\$).
100	Add library failed because information in library control block buffer and volume descriptor are inconsistent (CRL\$).
101	Delete library failed because library name does not agree with directories (CLL\$).
102	Delete library failed because another user has library open or is opening it (CLL\$).
103	Delete library failed because library is not empty (first chunk is not empty or it is not the only chunk) (CLL\$).
104	Library already deleted from system library directory. It was removed from the volume library directory at this time (CLL\$).
105	Improper library master password (CLP\$).
106	Change library password function out of range (CLP\$).
107	Restricted activity specified a secondary TCB when scheduling another activity.
110	The delete or update file directory entry failed because the file name specified does not agree with the file name in the disk directory entry when the relative position of the entry is specified (CLS\$).
111	The delete or update file directory entry failed because the file name specified is not in the disk directory (CLS\$).
112	Attempt to open library with invalid password (OPL\$).
113	Illegal mode on open or close library request (OPL\$ and CLL\$).
114	Library to be opened not in identified libraries' queue (OPL\$).
115	The abort request failed because the activity to be aborted is not a restricted activity or was not requested (ABT\$).
116	Cannot close and delete a library that is not opened in master mode. Normal close is attempted (CLL\$).
117	Allocation management is prohibited on unlabeled volumes (ALC\$ or DLC\$).
120	User I/O is prohibited on this volume (ALC\$ or DLC\$).
121	Bit map segment contains invalid ID (ALC\$, DLC\$, OPN\$, CLS\$, or PUT\$).
122	Starting segment number for deallocate is not a work area boundary (DLC\$ or CLS\$).

Table E-1 (cont). Executive Function Call Error Codes

Error Codes (A-register)	Indication(s)
123	Starting segment number for deallocate is not accessible to the user (DLC\$).
124	Starting segment number for deallocate is beyond the range of the bit map (DLC\$ or CLS\$).
125	Work area to be deallocated had not been allocated (DLC\$ or CLS\$).
x126	Physical I/O error during input of allocation bit map segment (ALC\$, DLC\$, CLS\$, or PUT\$).
x127	Physical I/O error during output of allocation bit map segment (ALC\$, DLC\$, CLS\$, or PUT\$).
130	Special action error return on close; data in the block buffer was not written successfully (CLS\$).
131	Special action error return on close; the control segment was not updated successfully (CLS\$).
132	Special action error return on close; the file descriptor was not successfully updated (CLS\$).
133	Request to open library that is already open (OPL\$).
134	Request to close library with open files in it (CLL\$).
135	Generic device type is not configured or has no associated volume descriptor (CVL\$, DVL\$, ALC\$, or DLC\$).
136	Logical unit number is not configured or has no associated volume descriptor (CVL\$, DVL\$, ALC\$, or DLC\$).
137	Incorrect specification of labeled/unlabeled volume in the volume control block (CVL\$).
140	Incorrect specification of public/private volume in the volume control block (CVL\$).
141	Segment size specified in parameter 1 of the volume control block is incorrect (CVL\$).
142	Number of segments per track specified in parameter 2 of the volume control block is incorrect (CVL\$).
143	Incorrect surface code in the volume control block (CVL\$).
144	Segment size specified in the volume control block is not a power of two between 64 and 512 (CVL\$).
145	Operator did not supply volume name or mount volume (CVL\$).
146	Requested volume is in use on another unit (CVL\$).
147	No unit is available for mounting requested volume (CVL\$).
x150	Physical I/O error on input by volume manager (CVL\$).

Table E-1 (cont). Executive Function Call Error Codes

Error Codes (A-register)	Indication(s)
151	Disconnect volume failed because the volume specified is not mounted on the unit specified (DVL\$).
152	Could not obtain control segment block for first direct access GET\$ request.
153	Segment number specified by the OTP\$ request is illegal.
154	Device being released is still processing I/O request (REL\$).
155	Illegal reserve request (RSV\$).
156	Mount is in process on requested unit (ALC\$, DLC\$, INP\$, or OTP\$).
157	TPR\$ request on behalf of a restricted activity was aborted because the activity is being aborted. This error code is passed to an action routine, such as CVL\$, which is making the TPR\$ request on behalf of the activity.
160	Cannot change library passwords on library not opened in master mode (CLP\$).
161	Configuration not correct for direct access (GET\$).
162	Number of buffers to be used for multiple buffering is not 2 or 3 (OPN\$).
163	No work area available in user area (ALC\$).
164	GSP\$ parameter set number invalid.
165	GSP\$ called with invalid password in an attempt to get disk volume parameters.
166	The library name or file name to be added to the directory is in an illegal format. The name must be an alphanumeric string, starting with an alphabetic (CRL\$ or OPN\$).
167	This error code is passed from the abort cleanup routine to the terminate activity action routine when there are no 16-word blocks available to terminate a restricted activity.
170	A restricted activity attempted to schedule a non-restricted activity (SAC\$).
171	Attempt to schedule a restricted activity failed because free memory is low (SAC\$).
172	Attempt to schedule a restricted activity which is being aborted (SAC\$).
173	Attempt to delete an activity which is running or requested (Delete Activity utility (DA)).
174	Attempt to terminate an activity which is not running (TMA\$).

Table E-1 (cont). Executive Function Call Error Codes

Error Codes (A-register)	Indication(s)
175	No free memory blocks configured of sufficient size to handle segments on a non-system disk (CRL\$ or CLL\$).
176	There is no suitable volume available to satisfy the request to connect a nonremovable volume (CVL\$).

APPENDIX F
PHYSICAL I/O DEVICE INFORMATION

TELEPRINTER (TYPE 5310 KSR-33)

Legal physical I/O requests for the KSR-33 teleprinter are input (INP\$) and output (OTP\$); the only legal data mode for it is 0 (ASCII).

When input is requested from the KSR, the characters are read from the keyboard, packed two characters per word, and stored in the user's buffer. Characters are read until the user's buffer is full, or until a carriage return character read from the keyboard is stored in the user's buffer. The following control characters are checked on input and the action described is taken:

- @ - Ignore last input line and start reading a new line.
- - Ignore last character typed in.
- Control-K - Terminate output. If control-K is struck when an OTP\$ (output) request is in progress, output is terminated and a bit is set in the status word. If an INP\$ (input) request is in progress, the control-K is treated as an ordinary character. If no input or output is in progress, the control-K is ignored.
- Control-P - Schedule attention task. If control-P is struck when no input or output is in progress, or when an OTP\$ (output) request is in progress, the control-P TCB supplied when the device was reserved is scheduled. If the control-P TCB is already scheduled, or if no control-P TCB was specified, the character is ignored. If an INP\$ (input) request is in progress, the control-P is treated as an ordinary character.
- Control-Shift-M - Terminate input request and return end-of-file status word.

When output to the teleprinter is requested, all data specified in the buffer must be packed two characters per word, and is transmitted to the printer until the range count is exhausted. Before physical output is started, the first word of the buffer is examined for line spacing control. If required, the necessary carriage control characters are set up. The second byte of the first word of the caller's buffer is treated as follows:

- Blank - Advance one line (insert CR, LF)
- 0 - Advance two lines (insert CR, LF, LF)
- + - Do not advance (no insert)
- Other - Ignore character and advance one line (insert CR, LF)

All other output carriage control is the user's responsibility and must be included in his buffer.

The following is a description of the status information returned to the user when control is transferred to the user's I/O completion return:

Word 1 (second word) of I/O Status Block

<u>Bit</u>	<u>Interpretation</u>
1-3	Reserved
4	Data not ready
5	Missed interrupt
6-8	Reserved
9	Wrong mode
10	Control-K received during output
11-13	Reserved
14	Range error
15-16	Reserved

The above described states are indicated if the appropriate bit is set.

Word 3 (fourth word) of I/O Status Block

Word 3 contains the actual number of words transferred or received.

CARTRIDGE DISK SUBSYSTEM (TYPE 476x)

The only legal physical I/O requests for the cartridge disk subsystem are input (INP\$) and output (OTP\$).

For each input and output request, the range, which must be specified in the I/O parameter list, must be a positive nonzero number which is less than or equal to the physical disk record length. If the range is zero, negative, or greater than the physical disk record length, the error return is taken to the calling program with the error code of 5 in the A-register.

The physical disk record length is used for all I/O except for segment 0. Therefore, if an input request is issued with a range which is less than the physical disk record length, a free memory block is used as the input buffer. The range is then used to determine how many words are to be transferred from the free memory block to the user-supplied buffer. The use of the free memory block means that the system free memory requirements are increased by one segment size block for sequential I/O.

The range is always used for I/O to segment 0; however, since the label resides on segment 0 of labeled volumes, the user should not write on segment 0 unless the volume is unlabeled.

The following is a description of the status information returned to the user when control is transferred to the user's I/O completion return. The states described below are indicated if the appropriate bit is set.

Word 1 (second word) of I/O Status Block

The following error status is returned in word 1 of the I/O status block. If the data transfer was successful, this word will be 0.

<u>Bit</u>	<u>Interpretation</u>
1	Word 4 contains the hardware status word which indicates the error. Word 6 also contains hardware status information.
2	Reserved
3	Missed interrupt
4	Not operational
5	Reserved
6	Transfer rate failure
7	Checksum error
8	Parity error
9	Reserved
10	Recovery error (Miscellaneous)
11	Volume protected on OTP\$ request
12-14	Reserved
15	No free memory available
16	Controller busy

Word 3 (fourth word) of I/O Status Block

Word 3 contains the number of words transferred or received. This word will always contain the range.

Word 4 (fifth word) of I/O Status Block

In addition to the status returned in word 1 of the I/O status block, the following hardware status is returned in word 4 of the I/O status block whenever bit 1 of word 1 is set.

<u>Bit</u>	<u>Interpretation</u>
1	Reserved
2	Write timing error
3	Failure of CPU to maintain transfer rate
4	Format error
5	Sector pulse time-out
6	Record address comparison failure
7	DMA has parity error
8	Write inhibit error
9	Data check word comparison failure
10	Time-out error
11-12	Reserved
13	Read timing
14	Seek error
15	Reserved
16	Missed data synchronization pulse

word 6 (seventh word) of I/O Status Block

The following hardware status is always returned in word 6 of the I/O status block.

<u>Bit</u>	<u>Interpretation</u>
1	Operational
2	Busy
3	Ready
4	Busy seek initiation
5	Unit 0 ready
6	Unit 1 ready
7	Unit 2 ready
8	Unit 3 ready
9	Unit 0 interrupt
10	Unit 1 interrupt
11	Unit 2 interrupt
12	Unit 3 interrupt
13	Reserved
14-15	Physical unit number of selected device
16	Busy being reset interrupt

HIGH-SPEED PAPER TAPE READER (TYPE 5010)

The only legal physical I/O request for the paper tape reader is input (INP\$). Legal data modes for the paper tape reader are those listed in Appendix G. When input from the paper tape reader is requested, data on the paper tape is read until an X-OFF character is reached, and then stored in the user's input buffer. The word count in word 3 (fourth word) of the I/O status block indicates the number of words stored in the user's input buffer.

The following is a description of the status information returned to the user when control is transferred to the user's I/O completion return:

Word 1 (second word) of I/O Status Block

<u>Bit</u>	<u>Interpretation</u>
1-2	Reserved
3	Missed interrupt
4	Data not ready
5	Device disabled
6	Reserved
7	Checksum error
8	Parity error
9	Format error
10	Recovery error
11-12	Reserved
13	End of file detected; not an error condition
14	Range error
15	No free memory available
16	Reserved

The above described states are indicated if the appropriate bit is set.

Word 3 (fourth word) of I/O Status Block

Word 3 contains the actual number of words received.

HIGH-SPEED PAPER TAPE PUNCH (TYPE 5210)

Legal physical I/O requests for the paper tape punch are output (OTP\$) and end of file (EOF\$); the legal data modes are listed in Appendix G. When output to the paper tape punch is requested, data in the user's output buffer is punched on the paper tape until the range count is exhausted. Punching of the end-of-file characters is initiated only by an EOF\$ request.

The following is a description of the status information returned to the user when control is transferred to the user's I/O completion return:

Word 1 (second word) of I/O Status Block

<u>Bit</u>	<u>Interpretation</u>
1-2	Reserved
3	Missed interrupt
4	Data not ready
5	Device disabled
6-9	Reserved
10	Recovery error
11-14	Reserved
15	No free memory available
16	Reserved

The above described states are indicated if the appropriate bit is set.

Word 3 (fourth word) of I/O Status Block

For OTP\$ requests, word 3 contains the actual number of words transferred.

CARD READER (TYPES 5100, 5121-5123, 5151-5153, 5161-5164)

CARD PUNCH (TYPE 5176)

CARD READER/PUNCH (TYPES 5140 AND 5172)

The legal physical I/O request for the card reader is input (INP\$); legal physical I/O requests for the card punch are output (OTP\$) and end of file (EOF\$). Legal data modes for the card reader and card punch are 0 (ASCII), 1 (binary), and 2 (verbatim).

In the verbatim mode, bits 5 through 16 of the buffer word are replaced by a card-image character.

Card-Image Character
(Row Number)

Card-Image Character (Row Number)					12	11	0	1	2	3	4	5	6	7	8	9
Buffer Word	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

When input from the card reader is requested, the data is read from one card and converted from 026 or 029 Hollerith card code to ASCII code, if the configurable conversion routines have been loaded in the system. The conversion routine 026 or 029 Hollerith is specified at system configuration time. The number of words stored in the user's input buffer will equal the range value, unless the range value exceeds the number of words read on a single card. In this case, only the number of words on the card are stored in the user's input buffer.

When output to the card punch is requested, the data in the user's output buffer is converted, if the configurable conversion routines have been loaded in the system. If the range of the words to be written is greater than the number of words that can be punched on a single card, only one card is punched, and the remaining words in the user's output buffer are ignored.

Status Information for Card Readers (Types 512x) and Card Reader/Punch (Type 5140)

The following is a description of the status information returned to the user when control is transferred to the user's I/O completion return.

Word 1 (second word) of I/O Status Block

<u>Bit</u>	<u>Interpretation</u>
1	An error has occurred; the hardware status is in word 4
2-4	Reserved
6-9	Reserved
10	Recovery error; an attempt to retry has resulted in an error
11-12	Reserved
13	End of file detected; not an error condition
14-16	Reserved

The above described states are indicated if the appropriate bit is set.

Word 3 (fourth word) of I/O Status Block

For INP\$ and OTP\$ requests, word 3 contains the actual number of words transferred or received.

Word 4 (fifth word) of I/O Status Block for Input Requests for the Type 5121 Card Reader and Type 5140 Card Reader/Punch Devices

In addition to the status information returned in word 1 of the I/O status block, the following hardware status is always returned in word 4 of the I/O status block.

<u>Bit</u>	<u>Interpretation</u>
1	Busy indicator
2	Ready indicator
3	End-of-card indicator
4-12	Always zero
13	Cycle indicator
14	Validity indicator
15	Data access error indicator
16	End-of-file indicator

Word 4 (fifth word) of I/O Status Block for Output Requests for the Type 5140 Card Reader/Punch Device

In addition to the status information returned in word 1 of the I/O status block, the following hardware status is always returned in word 4 of the I/O status block.

<u>Bit</u>	<u>Condition Tested</u>
1	Controller busy
2	Ready
3	End of card
4-11	Reserved
12	Punch check error
13	Read check error
14	Validity error
15	Data access error
16	Read end of file

Status Information for Card Reader Type 5100

The following is a description of the status information returned to the user when control is returned to the user's I/O completion return. The states described below are indicated if the appropriate bit is set.

Word 1 (second word) of I/O Status Block

<u>Bit</u>	<u>Interpretation</u>
1	An error has occurred; the hardware status is in word 4.
2	Reserved
3	Missed interrupt
4	Data not ready (not operational)
5	Device disabled (not operational)
6-9	Reserved
10	Recovery error
11-12	Reserved
13	End of file; not an error condition
14	Reserved
15	No free memory block available
16	Reserved

Word 3 (fourth word) of I/O Status Block

Word 3 contains the actual number of words received.

Word 4 (fifth word) of I/O Status Block

In addition to the status information returned in word 1 of the I/O status block, the following hardware status is always returned in word 4 of the I/O status block.

<u>Bit</u>	<u>Interpretation</u>
1	Option operational
2	Option busy
3	Device active
4	Access error present
5	Registration error present
6	Invalid character present
7	Trap flop
8	Reserved
9	Cycle error bit 1
10	Cycle error bit 2
11	Cycle error bit 3
12	Cycle error bit 4
13	Cycle timing error
14	Busy reset interrupt
15	Device going active interrupt
16	Ready interrupt

The above described states are true if the appropriate bit is set.

Status Information for Card Devices (Types 5151-5153, 5161-5164, 5172, and 5176)

The following is a description of the status information returned to the user when control is returned to the user's I/O completion return. The states described below are indicated if the appropriate bit is set.

Word 1 (second word) of I/O Status Block

<u>Bit</u>	<u>Interpretation</u>
1	An error has occurred; the hardware status is in word 4.
2	Reserved
3	Missed interrupt
4	Device not operational
5	Device disabled
6	Reserved
7	Read or punch error
8	Reserved
9	Mode error (conversion not configured)
10-12	Reserved
13	End of file; not an error condition
14	Reserved
15	No free memory block available for conversion
16	Reserved

Word 3 (fourth word) of I/O Status Block

For input and output requests, word 3 contains the actual number of words transferred or received.

Word 4 (fifth word) of I/O Status Block

In addition to the status information returned in word 1 of the I/O status block, the following hardware status is always returned in word 4 of the I/O status block.

<u>Bit</u>	<u>Interpretation</u>
1	Operational
2	Busy
3	Active
4	Stop code
5	Read check error or range error
6	Validity error
7	Punch echo error
8	Punch cycle error
9	Mark sense (option)
10	40 column (option)
11	51 column (option)
12	External clock track (option)
13-14	Reserved
15	Device going active interrupt
16	Busy reset interrupt

FIXED-HEAD DISK SUBSYSTEM (TYPE 451x)

The only legal physical I/O requests for the fixed-head disk subsystem are input (INP\$) and output (OTP\$).

For each input and output request, the range, which must be specified in the I/O parameter list, is used to determine the number of words to transfer to or from the user-supplied I/O buffer. The range must be a positive nonzero number less than or equal to the physical disk record length. If the range is zero, negative, or greater than the physical disk record length, the error return is taken to the calling program with the error code of 5 in the A-register.

The following is a description of the status information returned to the user when control is transferred to the user's I/O completion return. The states described below are true if the appropriate bit is set.

Word 1 (second word) of I/O Status Block

<u>Bit</u>	<u>Interpretation</u>
1	An error has occurred; the hardware status is in word 4
2-16	Reserved

Word 3 (fourth word) of I/O Status Block

Word 3 contains the actual number of words transferred or received.

Word 4 (fifth word) of the I/O Status Block

The hardware status is always returned in word 4 of the I/O status block.

<u>Bit</u>	<u>Condition Tested</u>
1	Operational
2	Busy
3	Active
4	Check byte error
5	Time-out error
6	Device not active error
7	Write protect error
8	Access error
9	Bus parity error
10	Device 0 active
11	Device 1 active
12	Device 2 active
13	Device 3 active
14	Reserved
15	Device going active interrupt
16	Busy reset interrupt

REMOVABLE DISK SUBSYSTEMS (TYPES 471x, 472x, 473x, 474x, 475x, AND 478x)

The only legal physical I/O requests for the removable disk subsystem are input (INP\$) and output (OTP\$).

For each input request, the range, which must be specified in the I/O parameter list, is used to determine the number of words to transfer to the user-supplied I/O buffer. For each output request to a DMC disk, the number of words transferred from the user-supplied I/O buffer is determined by the physical disk record length. The range specified for input and output requests to a DMA or DMC disk must be a positive nonzero number less than or equal to the physical disk record length. If the range is zero, negative, or exceeds the physical disk record length, the error return to the calling program is taken; an error code of 5 is returned in the A-register.

The following is a description of the status information returned to the user when control is transferred to the user's I/O completion return. The states described below are indicated if the appropriate bit is set.

Word 1 (second word) of I/O Status Block

<u>Bit</u>	<u>Interpretation</u>
1	Word 4 contains the hardware status word which indicates the error
2	Reserved
3	Missed interrupt
4-15	Reserved
16	Device busy

Word 3 (fourth word) of I/O Status Block

Word 3 contains the actual number of words transferred or received.

Word 4 (fifth word) of I/O Status Block

In addition to the software status returned in word 1 of the I/O status block, the following hardware status is returned in word 4 of the I/O status block whenever bit 1 of word 1 is set and also whenever word 1 is 0.

<u>Bit</u>	<u>Interpretation</u>
1	Controller busy
2	Data ready for transfer
3	Requested address not found
4	Attempt to format over index mark
5	Heads not loaded on selected unit
6	Requested unit not available
7	Seek error (attempt to seek track outside - 0 to 202 limits)
8-10	Reserved
11	Write operation requested while in protect mode
12	Data unsafe (inconsistency in internal logic such as read and write at the same time or erase with detent; refer to the hardware manual for complete list of data unsafe conditions)
13	Checksum error
14	Data transfer rate failure
15	Logical OR of bits 3 through 7 and 11 through 14
16	End-of-record mark found

Status information returned for requests to a DMA disk.

Word 1 (second word) of I/O Status Block

<u>Bit</u>	<u>Interpretation</u>
1	Word 4 contains the hardware status word which indicates the error; word 6 also contains hardware status information.
2	Reserved
3	Missed interrupt
4	Unit not operational
5	Reserved
6	Missed data; transfer rate failure
7	Checksum error
8	Bus parity error
9	Reserved
10	Recovery error (miscellaneous)
11	Write protect error
12-15	Reserved
16	Controller busy

Word 3 (fourth word) of I/O Status Block

Word 3 contains the actual number of words transferred or received.

Word 4 (fifth word) of I/O Status Block

In addition to the status returned in word 1 of the I/O status block, the following hardware status is returned in word 4 of the I/O status block whenever bit 1 of word 1 is set.

<u>Bit</u>	<u>Interpretation</u>
1	Seek error
2	Data unsafe condition
3	Failure of CPU to maintain transfer rate
4	Format error
5	Head selection error
6	Record address comparison failure
7	Bus parity error
8	Write operation requested while write protect is in force
9	Data check word comparison failure
10	Time-out error
11	Wrong cylinder comparison failure
12-15	Reserved
16	Missed data synchronization pulse

Word 6 (seventh word) of I/O Status Block

The following hardware status is always returned in word 6 of the I/O status block.

<u>Bit</u>	<u>Interpretation</u>
1	Operational
2	Busy
3	Active
4	Reserved
5	Unit 0 ready
6	Unit 1 ready
7	Unit 2 ready
8	Unit 3 ready
9-12	Reserved
13	Write track format end-of-range interrupt
14	Seek complete interrupt
15	Device going active interrupt
16	Busy being reset interrupt

LINE PRINTERS (TYPES 5520, 5524, 5526, 5527, 5529, 5541-2, 5551-2, 5565-9)

The only legal physical I/O request for the line printer is output (OTP\$); the only legal data mode is 0 (ASCII). When output to the line printer is requested, the characters that are to be printed must be packed two characters per word; the first word must contain a right-justified control character and must be included in the range parameter of the OTP\$ function call. There are no restrictions on the contents of the left byte of this word. If the number of words to be printed exceeds the limit for one line on the line printer, only one line is printed and the remaining words in the user's output buffer are ignored (Printer Types 554x will overprint). The control character is acted upon before the line is printed.

The first word of the user's output buffer must contain an ASCII forms control character, right-justified, as follows:

Character	ASCII (Octal)	Description
Δ (space)	240	Advance one line
+ (plus)	253	No line advance
0 (zero)	260	Advance two lines
1	261	Advance to top of form
2 ^a ,b,c	262	Advance according to channel 2
3 ^b	263	Advance according to channel 3
4 ^b	264	Advance according to channel 4
5 ^b	265	Advance according to channel 5
6 ^b	266	Advance according to channel 6
7 ^b	267	Advance according to channel 7
8 ^b ,c	270	Advance according to channel 8
9 ^a ,b,c	271	Advance according to channel 9
A ^a ,b,c	301	Advance according to channel 10
B ^a ,b,c	302	Advance according to channel 11
C ^a ,b,c	303	Advance according to channel 12
H	310	Advance to top of form (channel 1)
^a For Line Printer Types 5565-9, these characters will result in a single line space.		
^b For Line Printer Types 5551-2, if the vertical format unit (VFU) option is not present, a single line space will occur. For Line Printer Types 5541-2, a single line space will occur.		
^c For Line Printer Types 552x, these characters will result in a single line space.		

If none of the above is specified, an advance of one line is the default condition.

The following is a description of the status information returned to the user when control is transferred to the user's I/O completion return. The states described below are indicated if the appropriate bit is set.

Word 1 (second word) of I/O Status Block

<u>Bit</u>	<u>Interpretation</u>
1	Word 4 contains the hardware status word which indicates the error. If no error, this bit = 0.
2-3	Reserved
4	Not operational
5	Disabled
6-9	Reserved
10	Recovery error
11-16	Reserved

Word 3 (fourth word) of I/O Status Block

Word 3 contains the actual number of words transferred.

Status Information for Line Printer Types 552x

Word 4 (fifth word) of I/O Status Block

In addition to the software status returned in word 1 of the I/O status block, the following hardware status is always returned in word 4 of the I/O status block.

<u>Bit</u>	<u>Interpretation</u>
1	Busy indicator
2	Ready indicator
3	Paper advancing indicator
4	Vertical format tape channel 2 (end of form)
5-13	Always zero
14	Cycle indicator
15-16	Always zero

Status Information for Line Printer Types 554x, 555x, and 556x

Word 4 (fifth word) of I/O Status Block

In addition to the software status returned in word 1 of the I/O status block, the following hardware status is always returned in word 4 of the I/O status block.

<u>Bit</u>	<u>Interpretation</u>
1	Operational indicator
2	Busy indicator
3	Active indicator
4	End of form detected (See Note.)
5	Cycle check error (Line Printer Types 556x only) or 0 (See Note.)
6-8	Always zero
9	DMA parity error
10-14	Always zero
15	Active interrupt
16	Not busy interrupt

NOTE: Bits 4 and 5 are interchanged from the status word received from the hardware for compatibility with Printer Types 552x. For Printer Types 554x, 555x, and 556x, the end-of-form bit will be set in the I/O status block only if it is detected by the hardware and the forms control specified to the hardware was not a channel advance, including top of form.

Status Information for Line Printer Types 556x

Word 6 (seventh word) of I/O Status Block

In addition to the status information returned in words 1 and 4 of the I/O status block, the following hardware status is always returned in word 6 of the I/O status block.

<u>Bit</u>	<u>Interpretation</u>
1	Printer pattern parity error
2	Line buffer parity error
3	Sentinel bit error
4	Index check error
5	Load cycle in progress
6	Print cycle in progress
7	Format cycle in progress
8	Test mode
9	Interrupt request
10	Trap circuit
11	True comparison
12-16	Always zero

7-TRACK MAGNETIC TAPE SUBSYSTEMS (TYPES 402x AND 404x)
9-TRACK MAGNETIC TAPE SUBSYSTEMS (TYPES 405x, 415x, 418x, AND 419x)

The following applies to 7-track magnetic tape (Types 4021 and 4041) and to 9-track magnetic tape (Types 4051, 4150, 4180, and 4190).

Physical I/O requests for magnetic tape are input (INP\$), output (OTP\$), end of file (EOF\$), space file (SPF\$), space record (SPR\$), rewind (RWD\$), and unload (ULD\$). Legal data modes for magnetic tape are 0 (ASCII), 1 (binary), and 2 (verbatim).

For 9-track tapes, ASCII data is not translated to BCD, but remains in ASCII format. For all devices, the entire 16 bits of each data word are transferred to/from the magnetic tape.

For 7-track tapes (Type 4021), binary mode specifies that all 16 bits of each word (three characters per word) are to be transferred to/from the magnetic tape:

1	6	7	12	13	16
first frame		second frame		third frame	

Verbatim mode specifies that the high-order 12 bits of each word (two characters per word) are to be transferred to/from the magnetic tape:

1	6	7	12	13	16
first frame		second frame		not used	

For 7-track tapes (Type 4041), binary mode specifies that the 16 bits of each data word are to be transferred to/from the magnetic tape in binary/word mode:

1	6	7	12	13	16
first frame		second frame		third frame	

Verbatim mode specifies that 12 bits of data are transferred to/from the magnetic tape in binary/byte mode:

1	2	3	8	9	10	11	16
first frame			second frame				

(This BCD tape format is also used for 9-track magnetic tape Type 4051.)

When input from magnetic tape is requested, one record is read, and the data stored in the input buffer. The range value must be greater than or equal to the length of the record that is to be read from the tape. If the range value is less than the physical tape record, the parity error indicator is set in the I/O status block. If the data mode specified was ASCII, the data read from the magnetic tape is converted from BCD to ASCII for 7-track tape and stored in the user's buffer.

When output to the magnetic tape is requested, all data in the user's output buffer is written to the record on the tape. If the data mode specified is ASCII, the data in the user's output buffer is converted to BCD for 7-track tape before being written to the magnetic tape. Note that for 7-track tapes when the ASCII data mode is specified, the conversion from ASCII to BCD on output to tape and the reverse conversion from BCD to ASCII on input will result in the @ (at sign) character (ASCII '300) being converted to a ' (single quote) character (ASCII '247). This occurs as follows:

Character in Memory → Character on 7-track Tape → Character in Memory

<u>Graphic</u>	<u>ASCII</u>		<u>BCD</u>		<u>Graphic</u>	<u>ASCII</u>
'	'247	→	'14	→	'	'247
@	'300	→	'14	→	'	'247

If the end of tape or the beginning of tape is detected before the specified number of files to be spaced has been reached when an SPF\$ request is being processed, spacing of the files does not continue. The number of files actually spaced is returned in word 3 of the I/O status block.

If the end of file is detected before the specified number of records to be spaced has been reached when an SPR\$ request is being processed, spacing of the records is not continued. The number of records actually spaced is returned in word 3 of the I/O status block.

The following is a description of the status information returned to the user when control is transferred to the user's I/O completion return.

Word 1 (second word) of I/O Status Block

<u>Bit</u>	<u>Interpretation</u>
1	Types 4041 and 4051 only: word 4 contains one hardware status word indicating the error; word 6 contains the second hardware status word.
2	Reserved
3	Missed interrupt
4	Reserved
5	Device disabled
6-7	Reserved
8	Parity error
9-10	Reserved
11	Protect error
12	Reserved
13	End of file
14	End of tape
15	Beginning of tape
16	Device busy

The above described states are indicated if the appropriate bit is set.

Word 3 (fourth word) of I/O Status Block

For input and output requests, this word contains the actual number of words transferred or received. For Space File and Space Record requests, this word contains the number of files or records spaced.

Status Information for Magnetic Tapes (Types 4041 and 4051)

Word 1 (second word) of I/O Status Block

These tapes return the same status information to the user when control is returned to the user's I/O completion return as defined for Types 4021 and 4150 Magnetic Tapes. (See word 4.)

Word 3 (fourth word) of I/O Status Block

For input and output requests, this word contains the actual number of words transferred or received. For Space File and Space Record requests, this word contains the number of files or records spaced.

Word 4 (fifth word) of I/O Status Block

In addition to the status information returned in word 1 of the I/O status block, these tapes return the following hardware status word in word 4 of the I/O status block:

<u>Bit</u>	<u>Interpretation</u>
1	Operational
2	Busy
3	Active
4	Device 0 selected
5	Device 1 selected
6	Device 2 selected
7	Device 3 selected
8	Rewind in process
9	Write protect
10-14	Reserved
15	Device going active interrupt
16	TCU busy reset interrupt

Word 6 (seventh word) of I/O Status Block

In addition to the status information returned in words 1 and 4 of the I/O status block, these tapes also return the following status word in word 6 of the I/O status block.

<u>Bit</u>	<u>Interpretation</u>
1	Beginning of tape status
2	End of tape status
3	File mark detected
4	Premature termination
5	Range too short
6	Range equals zero
7	LRC error
8	Low not high error
9	Skew error
10	CRC error
11	False gap/error detectable
12	Invalid setup
13	Data rate error
14	Write current failure
15	Write runaway
16	CRC parity/VRC error

CASSETTE TAPE SUBSYSTEM (TYPE 5400)

The physical I/O requests for the cassette tape are input (INP\$), output (OTP\$), end of file (EOF\$), space file (SPF\$), space record (SPR\$), rewind (RWD\$), and unload (ULD\$). Spacing records and/or files, however, can only be forward. The legal data modes for cassette tape are 0 (ASCII), 1 (binary), and 2 (verbatim). Each of these data modes results in all 16 bits of each word being transferred.

When input from cassette tape is requested, one record is read from the tape and stored in the user-specified buffer. The range value must be greater than or equal to the length of the record that is to be read. If the range value is less than the length of the record, the parity error indicator is set in word 1 of the I/O status block. If a file mark is encountered when a record is read, the EOF status is set in word 1 of the I/O status block, and tape motion ceases immediately after the file mark.

When output to the cassette tape is requested, all data in the user's output buffer is written as a single record on the cassette tape. If a tape trailer label is encountered while a record is being written, no word count is returned to the user's status array. The end-of-tape indicator in word 1 of the I/O status block is set when the end-of-tape marker (18" before the physical EOT) is encountered. Writing after this is permitted but not recommended, as it is impossible to tell when the tape trailer will be encountered. Writing on this section of tape is mostly for a file mark, to denote EOT.

If the physical end of tape is encountered before the specified number of records or files have been spaced, tape motion ceases and word 3 of the I/O status block reflects the number of records or files successfully spaced. Further, if a file mark is encountered when records are spaced, tape motion ceases, and the number of records successfully spaced is returned in word 3 of the I/O status block.

The following describes status information returned to the user when control is transferred to the user's I/O completion return. The states described below are indicated if the appropriate bit is set.

Word 1 (second word) of I/O Status Block

<u>Bit</u>	<u>Interpretation</u>
1	An error has occurred; the hardware status is in word 4
2	Reserved
3	Missed interrupt
4	Reserved
5	Device disabled
6-7	Reserved
8	Parity error
9-10	Reserved
11	Write protect error
12	Reserved
13	End of file
14	End of tape
15	Beginning of tape
16	Device busy

Word 3 (fourth word) of I/O Status Block

For input and output requests, this word contains the actual number of words transferred or received. For Space File and Space Record requests, this word contains the actual number of files/records spaced.

Word 4 (fifth word) of I/O Status Block

In addition to the status information returned in word 1 of the I/O status block, the following hardware status word is returned in word 4 of the I/O status block, if an irrecoverable error has occurred.

<u>Bit</u>	<u>Interpretation</u>
1	Operational
2	Busy
3	Active, first handler
4	Active, second handler
5	EOT marker
6	BOT marker
7	Rewind
8	Handler select
9	Access error
10	Not used
11	Write phase-encoded data
12	Read phase-encoded data
13	Write protect
14	Not busy interrupt
15	Active interrupt
16	Data ready interrupt

TELEPRINTERS (KEYBOARD/PRINTER) (TYPE 5507 ASR-35 AND TYPE 5307 ASR-33)

The physical I/O requests, the mode, and the control characters for the ASR-35 (keyboard/prINTER) are the same as described earlier for the KSR-33 teleprinter. The only difference is that bit 15 in word 1 (second word) of the I/O status block is set if there are no free memory blocks available for the ASR device driver to use.

TELEPRINTERS (READER/PUNCH) (TYPE 5507 ASR-35 AND TYPF 5307 ASR-33)

The legal physical I/O requests for the ASR-33 (reader/punch) are input (INP\$), output (OTP\$), and end of file (EOF\$). Legal data modes for the ASR device are listed in Appendix G.

When input is requested, data on paper tape is read until an X-OFF character is reached, and then stored in the user's input buffer. The word count in word 3 (fourth word) of the I/O status block indicates the number of words stored in the user's input buffer.

When output is requested, data in the user's output buffer is punched on paper tape until the range count is exhausted. Punching the end-of-file character is initiated only by an EOF\$ request.

The following is a description of the status information returned to the user when control is transferred to the user's I/O completion return.

Word 1 (second word) of I/O Status Block

<u>Bit</u>	<u>Interpretation</u>
1-2	Reserved
3	Missed interrupt
4	Device not ready
5-6	Reserved
7	Checksum error
8	Parity error
9	Mode error or format error
10	Control-K received during printer output
11-12	Reserved
13	End of file
14	Range error
15	Free memory not available
16	Reserved

Word 3 (fourth word) of I/O Status Block

For INP\$ and OTP\$ requests, word 3 contains the actual number of words transferred or received.

NOTE: If power failure occurs, the state of the I/O devices is not saved. When the user restarts the system, he must again reserve the I/O devices before using them.

APPENDIX G
PHYSICAL I/O DATA MODE ASSIGNMENTS

Mode Number	Data Mode
0	ASCII without checksum
1	Binary without checksum
2	Verbatim
3	ASCII with checksum for Type 5307 ASR-33, Type 5507 ASR-35, Type 5010 Paper Tape Reader, and Type 5210 Paper Tape Punch only.
4	Binary with checksum for Type 5307 ASR-33, 5507 ASR-35, Type 5010 Paper Tape Reader, and Type 5210 Paper Tape Punch only.

APPENDIX H

FORTRAN RUN-TIME ERROR MESSAGES

FORTRAN run-time error messages are displayed in the following format:

xx actnam F ERR yy

xx - Is the message number.
actnam - Is the activity name.
yy - Is an error mnemonic. The mnemonics are listed below.

Following an error report, response is limited from the operator. The operator responds in the normal way by typing:

(P)xx{A/C}

(P) - Nonprinting character.
xx - Message number of the message to which the operator is responding.
A - Character to abort the activity.
c - Any other legal character to continue the activity.

In the following summary table of error mnemonics, the error mnemonics are defined, the OS/700 FORTRAN Library component which produces the error is indicated in parentheses after the mnemonic, and recovery procedures, where possible, are described. Errors classified as fatal have no recovery and the same abort action is undertaken irrespective of the continuation procedure indicated by the operator. For nonfatal errors, the operator can either abort or continue if the recovery condition is acceptable. Note that no action takes place following the error detection until the operator responds, whether the error is fatal or not. The OS/700 FORTRAN Library component codes are as follows:

M - Math library
R - Run-time library
S - System function library

<u>Mnemonic</u>	<u>Meaning</u>	<u>Recovery</u>
AD (M)	Overflow in double-precision operation.	Continuation uses a value with the maximum possible double-precision magnitude and the sign of the true result.
AO (M)	Array element referenced is outside the array boundaries.	Continuation uses the illegal array element referenced.
AR (R)	Device already reserved.	Continuation attempts to reserve the device again which will either be successful or result in the error being repeated.
CE (M)	The absolute value of the complex number to be raised to a complex power is zero.	Continuation uses an undefined value. Abort is recommended.
DL (M)	A negative or zero argument is being used for double-precision logarithmic routines.	Continuation uses an undefined value. Abort is recommended.
DT (M)	Both arguments in a double-precision quotient arctangent calculation are zero.	Continuation uses an undefined value. Abort is recommended.
DZ (M)	Zero dividend specified or underflow for real divide.	Continuation uses zero result.
EF (R)	End of file reached and no end of file or error return specified in READ statement.	Continuation uses input record buffer with undefined contents. A further device input will be attempted if later specified by the same or another READ statement.
EQ (M)	During double-precision exponentiation, overflow occurred when multiplying the double-precision accumulator by a power of 2.	Continuation uses a value with the maximum possible double-precision magnitude and the sign of the true result.
EX (M)	During real exponentiation, exponent overflow occurred.	Continuation uses the maximum positive real value.
FE (R)	During a formatted read or write, an illegal format specification or statement has been encountered.	Continuation ignores the rest of the current list and goes on to the next statement.
FU (R)	FORTTRAN Unit number specified was not present in the FORTRAN unit table.	Continuation uses by default the initial device in the FORTRAN logical unit table; this device should be a sharable device, typically the operator's console.
GO (R)	The control variable contents for an assigned GO TO did not match a list entry.	No continuation attempted - the activity is terminated.
II (M)	Error in raising an integer to an integer power.	Continuation uses the maximum possible integer value with the sign of the true result.
IM (M)	Overflow or underflow in integer multiplication.	Continuation uses either the maximum positive or the maximum negative integer value, according to the sign of the true result.

<u>Mnemonic</u>	<u>Meaning</u>	<u>Recovery</u>
IO (R)	A device error condition has been encountered and no error return was specified.	Continuation uses input record buffer with undefined contents. A further device input will be attempted if later specified by the same or another READ statement.
IZ (M)	Integer division by zero attempted, or Integer division of -32768/-1 attempted	Continuation uses the maximum positive or negative integer value, the sign reflecting the sign of the divisor. Continuation uses +32767.
LG (M)	The logarithm of a negative or zero argument has been attempted.	Continuation uses an undefined value. Abort is recommended.
MD (M)	Overflow or underflow occurred during double-precision multiply or divide.	Continuation uses the maximum positive or negative double-precision value, according to the sign of the true result
PZ (M)	Attempted to divide a double-precision argument by zero.	Continuation uses the maximum positive or negative double-precision value - the sign reflecting the sign of the divisor.
RD (R)	Characters input during a formatted READ inconsistent with format specification or produced a number that overflowed or underflowed.	Continuation defaults a zero on false value to the offending list item and continues with the remainder of the READ statement.
RI (M)	Conversion from real to integer caused overflow.	Continuation uses an undefined value. Abort is recommended.
RL (R)	A requested or forced "release device" was impossible to perform.	Continuation assumes device has been released.
RS (R)	A requested or forced "reserve device" produced an error condition, excluding the case of device already in use.	Continuation assumes device is reserved. This may produce further errors.
SA (M)	Overflow occurred during real add or subtract.	Continuation uses a value with the maximum possible real magnitude and the sign of the true result.
SD (M)	The real divisor in a real division operation was not normalized.	Continuation uses an undefined value. Abort is recommended.
SF (S)	A mandatory device control block is not defined. Array in a Get Block request was not DYNAMIC.	Continuation is at the subroutine error return, if specified, or at the normal return. Get Block or I/O request is ignored.
SM (M)	Arithmetic overflow during real multiply or divide.	Continuation uses a value with the maximum possible real magnitude and the sign of the true result.
SQ (M)	The argument for a real square root was negative.	Continuation uses an undefined value. Abort is recommended.

APPENDIX J
OCTAL/DECIMAL CONVERSION TABLE

Octal/Decimal Integers

Octal	Decimal
1	1
2	2
3	3
4	4
5	5
6	6
7	7
10	8
20	16
30	24
40	32
50	40
60	48
70	56
100	64
200	128
300	192
400	256
500	320
600	384
700	448
1000	512
2000	1024
3000	1536
4000	2048
5000	2560
6000	3072
7000	3584
10000	4096
20000	8192
30000	12288
40000	16384
50000	20480
60000	24576
70000	28672
100000	32768

Table J-1. Octal/Decimal Conversion Table

Octal	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

Octal	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

Table J-1 (cont). Octal/Decimal Conversion Table

Octal	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

Octal	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

Table J-1 (cont). Octal/Decimal Conversion Table

Octal	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

Octal	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

Table J-1 (cont). Octal/Decimal Conversion Table

Octal	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

Octal	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	4022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

APPENDIX K
POWERS-OF-2 TABLE

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25

APPENDIX L ASCII/HOLLERITH CARD CODE TABLE

Table L-1 presents the various card codes that can be employed by OS/700 users. The Hollerith card code set used in a particular system is determined at system configuration time.

Table L-1. ASCII/Hollerith Codes

ASCII	Character	ASCII	Character	026		029		GBCD (Series 6000)		ASCII	Character
				Hollerith	Octal ^a	Hollerith	Octal	Hollerith	Octal		
200	NUL	240	Space	Blank	20	Blank	20	Blank	20	340	
201	SOH	241	!	8-6	16	12-8-7	77	0-7-8	77	341	a
202	STX	242	"	0-8-7	37	8-7	17	0-6-8	76	342	b
203	ETX	243	#	0-8-2	32	8-3	13	3-8	13	343	c
204	EOT	244	\$	11-8-3	53	11-8-1	53	11-3-8	53	344	d
205	ENQ	245	%	12-8-5	75	0-8-4	34	0-4-8	74	345	e
206	ACK	246	&	11-8-6	56	12	60	12	32	346	f
207	BEL	247	'	8-4	14	8-5	15	11-7-8	57	347	g
210	BS	250	(0-8-4	34	12-8-5	75	12-5-8	35	350	h
211	HT	251)	12-8-4	74	11-8-5	55	11-5-8	55	351	i
212	LF	252	*	11-8-4	54	11-8-4	54	11-4-8	54	352	j
213	VT	253	+	12	60	12-8-6	76	12-0	60	353	k
214	FF	254	,	0-8-3	33	0-8-3	33	0-1-8	73	354	l
215	CR	255	-	11	40	11	40	11	52	355	m
216	SO	256	.	12-8-3	73	12-8-3	73	12-3-8	33	356	n
217	SI	257	/	0-1	21	0-1	21	0-1	61	357	o
220	DLE	260	0	0	00	0	00	0	00	360	p
221	DC1	261	1	1	01	1	01	1	01	361	q
222	DC2	262	2	2	02	2	02	2	02	362	r
223	DC3	263	3	3	03	3	03	3	03	363	s
224	DC4	264	4	4	04	4	04	4	04	364	t
225	NAK	265	5	5	05	5	05	5	05	365	u
226	SYN	266	6	6	06	6	06	6	06	366	v
227	ETH	267	7	7	07	7	07	7	07	367	w
230	CAN	270	8	8	10	8	10	8	10	370	x
231	EM	271	9	9	11	9	11	9	11	371	y
232	SS	272	:	8-5	15	8-2	12	5-8	15	372	z
233	ESC	273	;	11-8-2	52	11-8-6	56	11-6-8	56	373	[
234	FS	274	<	11-8-7	57	12-8-4	74	12-6-8	36	374	\
235	GS	275	=	8-3	13	8-6	16	0-5-8	75	375]
236	RS	276	>	8-7	17	0-8-6	36	6-8	16	376	^
237	US	277	?	0-8-5	35	0-8-7	37	7-8	17	377	DEL
300	␣			8-2	12 ^a	8-4	14	4-8	14		
301	A			12-1	61	12-1	61	12-1	21		
302	B			12-2	62	12-2	62	12-2	22		
303	C			12-3	63	12-3	63	12-3	23		
304	D			12-4	64	12-4	64	12-4	24		
305	E			12-5	65	12-5	65	12-5	25		
306	F			12-6	66	12-6	66	12-6	26		
307	G			12-7	67	12-7	67	12-7	27		
310	H			12-8	70	12-8	70	12-8	30		
311	I			12-9	71	12-9	71	12-9	31		
312	J			11-1	41	11-1	41	11-1	41		
313	K			11-2	42	11-2	42	11-2	42		
314	L			11-3	43	11-3	43	11-3	43		
315	M			11-4	44	11-4	44	11-4	44		
316	N			11-5	45	11-5	45	11-5	45		
317	O			11-6	46	11-6	46	11-6	46		
320	P			11-7	47	11-7	47	11-7	47		
321	Q			11-8	50	11-8	50	11-8	50		
322	R			11-9	51	11-9	51	11-9	51		
323	S			0-2	22	0-2	22	0-2	62		
324	T			0-3	23	0-3	23	0-3	63		
325	U			0-4	24	0-4	24	0-4	64		
326	V			0-5	25	0-5	25	0-5	65		
327	W			0-6	26	0-6	26	0-6	66		
330	X			0-7	27	0-7	27	0-7	67		
331	Y			0-8	30	0-8	30	0-8	70		
332	Z			0-9	31	0-9	31	0-9	71		
333	[11-8-5	55	12-8-2	72	2-8	12		
334	\			12-8-6	76	0-8-2	32	12-7-8	37		
335]			0-8-6	36	11-8-2	52	12-4-8	34		
336	^			12-8-2	72	11-8-7	57	11-0	40		
337	_			12-8-7	77	0-8-5	35	0-2-8	72		

^aThe 026 Octal Values are used on 7-track magnetic tape except for ␣ which cannot be represented.

Honeywell Bull

Technical Publications Remarks Form* (*please print*)

Title: _____

SYSTEM 700 OPERATOR'S GUIDE
OS/700

Order: _____

61A.2-AG14, Rev.3

Dated: _____

JULY 1975

Errors in publication: _____

Suggestions for improvement to publication: _____

from : Name _____

Company _____

Title _____

Address _____

* Your comments will be promptly investigated by appropriate technical personnel, action will be taken as required, and you will receive a written reply. If you do not require a written reply, please check here : ☐

CUT ALONG LINE

**Please hand this technical publication remark form
to your Honeywell Bull representative,**

or mail to :

Honeywell Bull

**Marketing Communications
Documentation/Publications**

**94, avenue Gambetta
75960 PARIS CEDEX 20 - FRANCE**

Honeywell Bull

HONEYWELL INFORMATION SYSTEMS