SYSTEM/32

IBM System/32
System Data Areas
and Diagnostic Aids

IBM System/32
System Data Areas
and Diagnostic Aids
Programming Information

SY21-0532-4
File No. S32-36

Program Number
5725-SC1

Fifth Edition (May 1977)

This is a major revision of, and obsoletes, SY21-0532-3. Changes to text and small changes to illustrations are indicated by a vertical line at the left of the change; new or extensively revised illustrations are denoted by a bullet (●) at the left of the figure caption.

This edition applies to version 06, of IBM System/32, Program Number 5725-SC1 and to all subsequent versions until otherwise indicated in new editions or technical newsletters. Changes are periodically made to the specifications herein; before using this publication, consult the latest *IBM System/32 Bibliography*, GC20-0032, for the editions that are applicable and current.

Use this publication only for the purposes stated in the *Preface*.

Publications are not stocked at the address below. Requests for copies of IBM publications and for technical information about the system should be made to your IBM representative or to the IBM branch office serving your locality.

A Reader's Comment Form is at the back of this publication. If the form has been removed, address your comments to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901. IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatsoever. You may, of course, continue to use the information you supply.

Preface

This handbook is designed to aid the assembler user and IBM personnel responsible for supporting the IBM System/32 by providing:

- An overview of System/32 system control program (SCP) organization
- Descriptions of the data areas within the system
- Descriptions of how to use the diagnostic aids available for diagnosing system malfunctions

This handbook is intended to be a quick reference aid. Although it is revised periodically, it might not reflect the current level of documentation. Consult the source publications if you are in doubt about any material contained in this handbook.

For all data areas shown in this manual, the field displacements (displ) point to the leftmost byte of the field and are expressed in hexadecimal. The field lengths (lng in bytes) are in decimal.

In this manual, *sector address* refers to SS addressing; *relative sector address* refers to displacement from the start of a specified area.

In the description columns, unless otherwise specified, references to displacements, lengths (such as record length), and number of items (such as number of records) are assumed to be in hex notation. Unused bits in listings are generally not shown in the description columns.

Note: Service numbers for the System/32 programs are as follows:

SCP (5725-SC1)–1251040
SCP (Word Processing Feature 6002)–1251060
RPG II (5725-RG1)–2253709
Utilities (5725-UT1)–2253719 (sort)
2253729 (DFU)
2253739 (SEU)
MRJE (5725-SC1)–1251050
BWS (5725-SC1)–1251070
Word Processor/32 (5725-XX1)–2253759
FORTRAN IV (5725-FO1)–2253799
Assembler (5725-AS1)–2253749

Related Publications

IBM Field Engineering Handbook, 6 ring blue binder to bind the *IBM System/32 Data Areas and Diagnostic Aids*. This binder can be ordered by using either the part number 453559 or the form number S-229-4124. See your IBM publications coordinator for information on ordering.

Theory of Operations:

IBM System/32 Theory-Diagrams, SY31-0346
IBM System/32 1255 Attachment Feature Theory-Diagrams, SY31-0468

Maintenance Manuals:

IBM System/32 Introduction and Maintenance, SY31-0373

Parts Catalog:

IBM System/32 Parts Catalog, S131-0595

Operator's Guide:

IBM System/32 Operator's Guide, GC21-7591

Reference Manuals:

- *The IBM Diskette General Information Manual,* GA21-9182
- *IBM System/32 Functions Reference Manual,* GA21-9176
- *IBM System/32 System Control Programming Reference Manual,* GC21-7593
- *IBM System/32 System Control Programming Reference Manual—Word Processing,* GC34-0078
- *IBM System/32 Data Communications Reference Manual,* GC21-7691
- *IBM System/32 RPG II Reference Manual,* SC21-7595
- *IBM System/32 Utilities Program Product Reference Manual—Data File Utility,* SC21-7600
- *IBM System/32 Utilities Program Product Reference Manual—Source Entry Utility,* SC21-7605
- *IBM System/32 Utilities Program Product Reference Manual—Sort,* GC21-7633
- *IBM System/32 SCP Command Statement Reference Summary,* GX21-7687
- *IBM System/32 Basic Assembler Reference Manual,* SC21-7673
- *IBM System/32 FORTRAN IV Reference Manual,* SC21-7682
- *IBM System/32 Magnetic Character Reader Reference and Logic Manual,* GC21-7692
- *IBM System/32 Overlay Linkage Editor Reference Manual,* GC21-5156

Program Logic Manuals:

- *IBM System/32 Control Storage Logic Manual,* SY21-0533
- *IBM System/32 System Logic Manual,* SY21-0567
- *IBM System/32 RPG II Logic Manual,* LY21-0538
- *IBM System/32 Utilities Program Product Logic Manual,* LY21-0539
- *IBM System/32 Data Communications Logic Manual,* SY21-0551
- *IBM System/32 Word Processing Logic Manual,* SY34-0069

General:

- *IBM System/32 Introduction,* GC21-7582
- *IBM System/32 Displayed Messages Guide—System*, GC21-7704
- Titles and abstracts of other related publications are listed in the *IBM System/32 Bibliography,* GC20-0032

List of Acronyms

ABEND	abnormal end
ACK	acknowledgement
AM	address mark
APAR	authorized program analysis report
ARR	address recall register
ASCII	American National Standard Code for Information Interchange

BCC	block check character
BDE	basic data entry
BIU	basic information unit
BPS	bits per second
BSC	binary synchronous communications
BSCA	binary synchronous communications adapter
BWS	batch work station

CAM	compiler access method
CAR	current address register
CCHS	cylinder head sector (disk)
CE	customer engineer
CICS	customer information control system
CMDR	command reject
CRC	cyclic redundancy check
CRT	cathode ray tube (display screen)
CSDE	controlled sequential data entry

DAF	destination address field
DAR	data address register
DB/DC	data base/data communications
DBO	data bus out
DFC	data flow control
DFU	data file utility
DISC	disconnect
DLE	data link escape
DM	data management
DR	definite response
DTF	define the file
DTT	define the table

E	expedite
EB	end bracket
EBCDIC	extended binary coded decimal interchange code
ECS	extended control storage
ECT	error count table
EHT	error history table
ENQ	enquiry
EOJ	end of job
EOT	end of transmission
ERAP	error recording analysis procedure
ERP	error recovery procedure
ERT	error recording table
ETB	end of transmission block
ETX	end of text
EXCP	exception

FCU	file conversion utility
FID	format identifier
FMH	format header

FOC	first of chain
HDR	header
I	information
IAR	instruction address register
ID	identification
IDE	interactive data entry
IMPL	initial microcode program load
IMS	information management systems
INQ	inquiry (key)
IOB	input/output block
IOS	input/output supervisor
IPL	initial program load
ITB	intermediate text block
I1	diskette drive
JJT	job-to-job transition
K	1024 (bytes)
LCB	library control block
LDAM	linked direct access method
LSR	local storage register
LU	logical unit
MCU	mag card unit
MIC	message identification code
MRJE	MULTI-LEAVING remote job entry
MVF	multivolume file
NR	number received
NAK	negative acknowledgement
NRZI	nonreturn-to-zero-inverted
NS	number sent
NSA	nonsequence acknowledgement
OAF	origin address field
OCL	operation control language
OLMV	online multivolume
PLCA	program level communication area
PSR	program status register
PTF	program temporary fix
PU	physical unit
R/I	reader/interpreter
RH	request/response header
RIB	request indicator byte
RJE	remote job entry
RLD	relocation dictionary
RNR	receive not ready
ROL	request online
RR	receive ready
RSP	response
RU	request/response unit
RVI	reverse interrupt

SCA	system communication area
SCB	string control byte
SCP	system control program
SDE	sequential data entry
SDLC	synchronous data link control
SERDES	serialization/deserialization
SEU	source entry/utility
SIAM	shared I/O access method
SIO	start input/output (instruction)
SIS	scientific instruction set
SNA	systems network architecture
SNRM	set normal response mode
SS	sector address
SSCP	system services control point
SSN	starting sector address and number of sectors minus one
STX	start of text characters
SVC	supervisor call (instruction)
SWA	scheduler work area
SYN	synchronization (character for BSC)
TAR	transition address register
TH	transmission header
TRN	transparent
TTD	temporary text delay
UDT	unit definition table
UPSI	user program switches
VTOC	volume table of contents
WACK	wait for acknowledgement
WP	word processing
WPCU	word processing communications utility
WPLCA	word processing library communication area
WPWA	word processing work area
XID	exchange identification
XR1	index register one
XR2	index register two

Figures

General

IOB

DTF

Parameter Lists

Disk Fixed Areas

Disk Library Area

Diskette

Diagnostic Aids

Appendixes

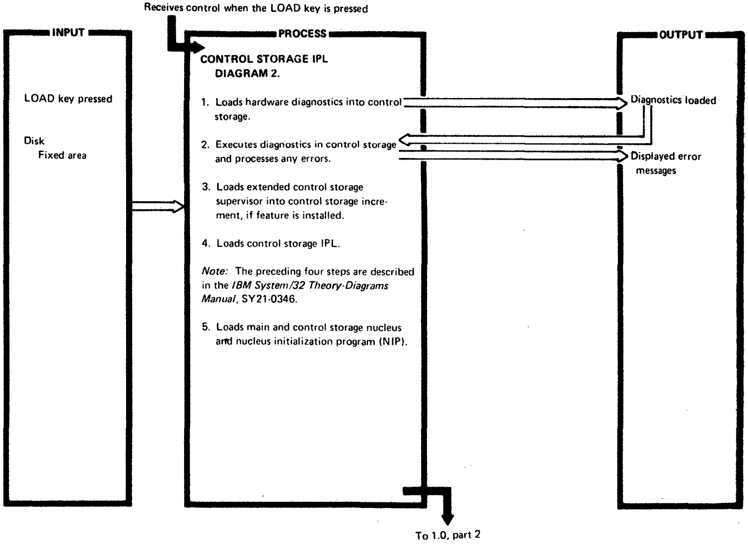
Section 1. System Overview

The IBM System/32 SCP (system control program) consists of several major components. Figure 1-1 shows an example of control flow between these major components from IPL to end of job. Note that within Figure 1-1, references are made to a number of diagrams. These diagrams are assigned diagram numbers.

Diagram numbers are assigned individually on a system-wide basis so that a diagram in a particular SCP logic manual can be referenced from any other SCP logic manual for the system.

Figure 1-2 is the overview table of contents for the IBM System/32 system. It lists the diagrams which describe the SCP functions, and identifies the manual in which each function is discussed.

Figure 1-3 is an overview of control storage operation that illustrates the execution of a program from main storage. Included is the hardware/software interaction and interrelationships of the various control storage routines.



Receives control when the LOAD key is pressed

INPUT

LOAD key pressed

Disk
Fixed area

PROCESS

CONTROL STORAGE IPL DIAGRAM 2.

1. Loads hardware diagnostics into control storage.
2. Executes diagnostics in control storage and processes any errors.
3. Loads extended control storage supervisor into control storage increment, if feature is installed.
4. Loads control storage IPL.

Note: The preceding four steps are described in the *IBM System/32 Theory-Diagrams Manual*, SY21-0346.

5. Loads main and control storage nucleus arnd nucleus initialization program (NIP).

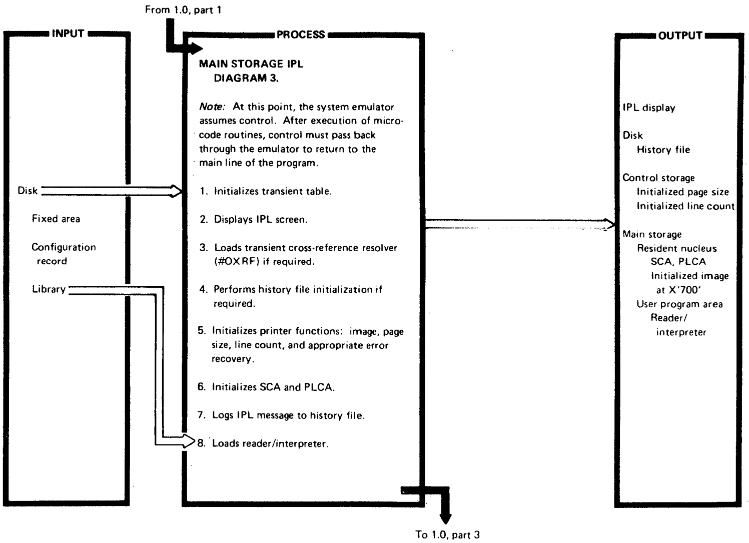
OUTPUT

Diagnostics loaded

Displayed error messages

To 1.0, part 2

Figure 1-1 (Part 1 of 11). System/32 System Control Program Overview



From 1.0, part 1

INPUT

Disk

Fixed area

Configuration record

Library

PROCESS

MAIN STORAGE IPL DIAGRAM 3.

Note: At this point, the system emulator assumes control. After execution of microcode routines, control must pass back through the emulator to return to the main line of the program.

1. Initializes transient table.
2. Displays IPL screen.
3. Loads transient cross-reference resolver (#OXRF) if required.
4. Performs history file initialization if required.
5. Initializes printer functions: image, page size, line count, and appropriate error recovery.
6. Initializes SCA and PLCA.
7. Logs IPL message to history file.
8. Loads reader/interpreter.

OUTPUT

IPL display

Disk
  History file

Control storage
  Initialized page size
  Initialized line count

Main storage
  Resident nucleus
    SCA, PLCA
    Initialized image at X'700'
  User program area
    Reader/interpreter

To 1.0, part 3

Figure 1-1 (Part 2 of 11). System/32 System Control Program Overview

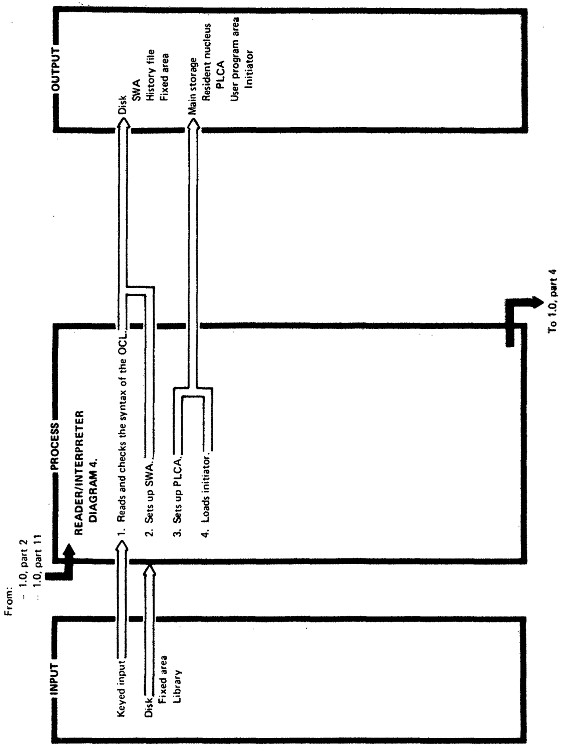
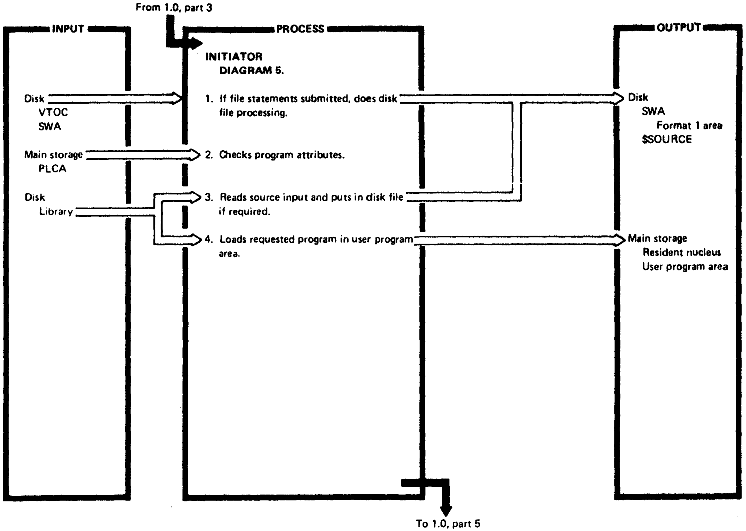


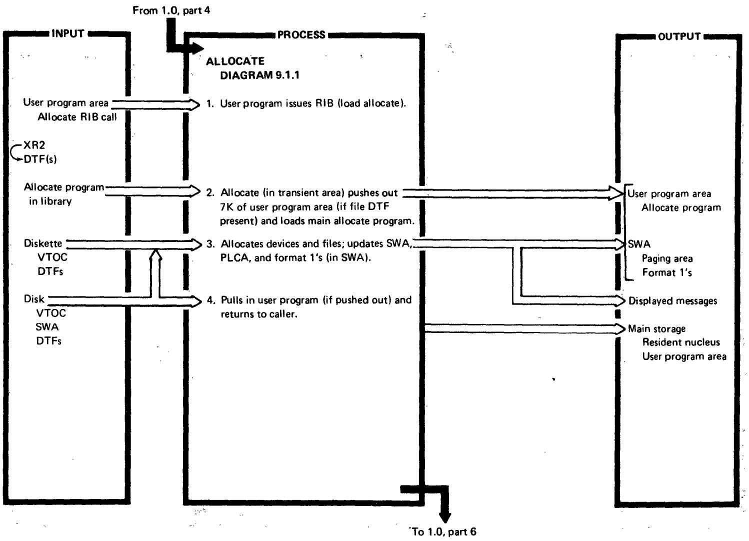
Figure 1-1 (Part 3 of 11). System/32 System Control Program Overview



INITIATOR
DIAGRAM 5.

1. If file statements submitted, does disk file processing.
2. Checks program attributes.
3. Reads source input and puts in disk file if required.
4. Loads requested program in user program area.

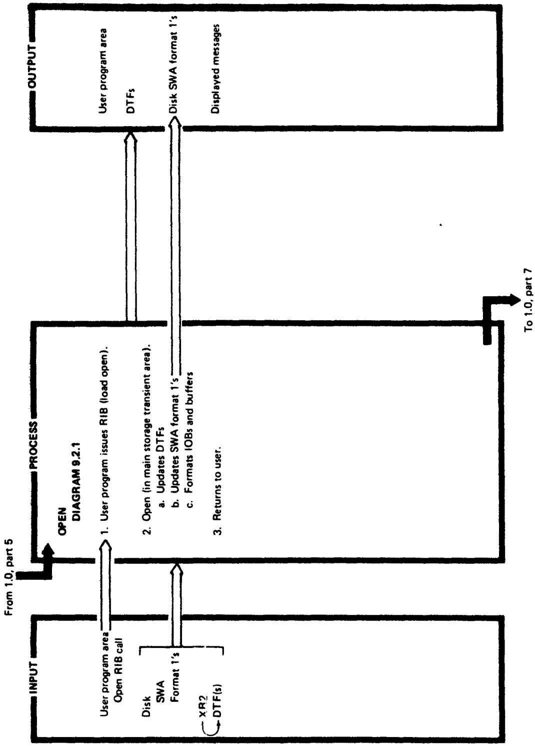
Figure 1-1 (Part 4 of 11). System/32 System Control Program Overview



ALLOCATE DIAGRAM 9.1.1

1. User program issues RIB (load allocate).
2. Allocate (in transient area) pushes out 7K of user program area (if file DTF present) and loads main allocate program.
3. Allocates devices and files; updates SWA, PLCA, and format 1's (in SWA).
4. Pulls in user program (if pushed out) and returns to caller.

Figure 1-1 (Part 5 of 11). System/32 System Control Program Overview



INPUT

User program area

Open RIB call

Disk

SWA

Format 1's

XR2

DTF(s)

From 1.0, part 5

PROCESS

OPEN

DIAGRAM 9.2.1

1. User program issues RIB (load open).

2. Open (in main storage transient area).
   a. Updates DTFs
   b. Updates SWA format 1's
   c. Formats IOBs and buffers

3. Returns to user.

To 1.0, part 7

OUTPUT

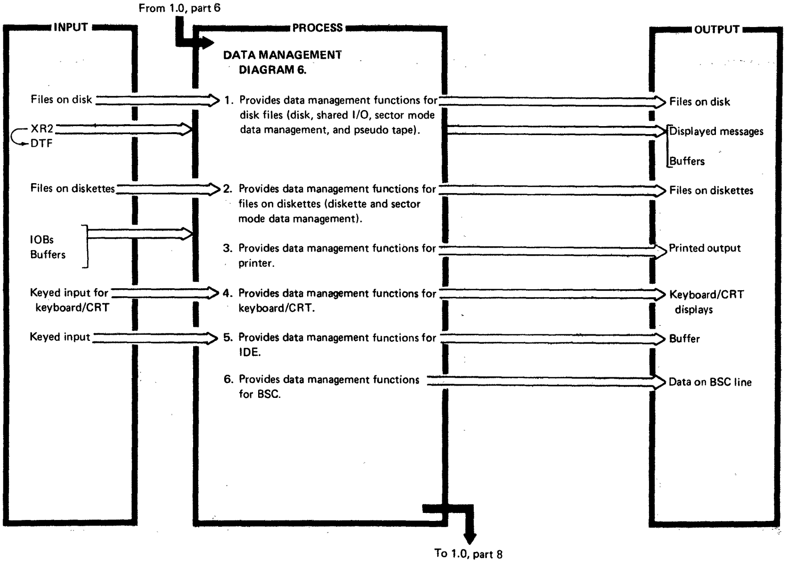
User program area

DTFs

Disk SWA format 1's

Displayed messages

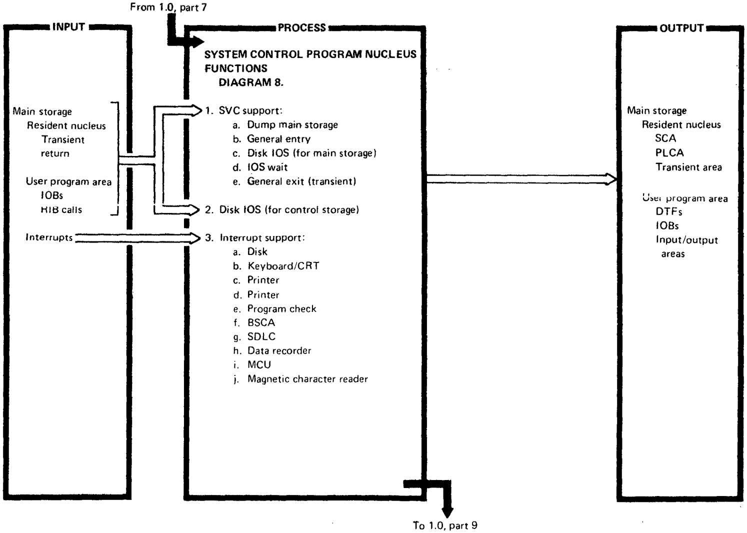
Figure 1-1 (Part 6 of 11). System/32 System Control Program Overview



DATA MANAGEMENT DIAGRAM 6.

1. Provides data management functions for disk files (disk, shared I/O, sector mode data management, and pseudo tape).
2. Provides data management functions for files on diskettes (diskette and sector mode data management).
3. Provides data management functions for printer.
4. Provides data management functions for keyboard/CRT.
5. Provides data management functions for IDE.
6. Provides data management functions for BSC.

Figure 1-1 (Part 7 of 11). System/32 System Control Program Overview



From 1.0, part 7

INPUT

Main storage
Resident nucleus
Transient
return

User program area
IOBs
RIB calls

Interrupts

PROCESS

SYSTEM CONTROL PROGRAM NUCLEUS FUNCTIONS
DIAGRAM 8.

1. SVC support:
   a. Dump main storage
   b. General entry
   c. Disk IOS (for main storage)
   d. IOS wait
   e. General exit (transient)
2. Disk IOS (for control storage)
3. Interrupt support:
   a. Disk
   b. Keyboard/CRT
   c. Printer
   d. Printer
   e. Program check
   f. BSCA
   g. SDLC
   h. Data recorder
   i. MCU
   j. Magnetic character reader

OUTPUT

Main storage
Resident nucleus
SCA
PLCA
Transient area

User program area
DTFs
IOBs
Input/output
areas

To 1.0, part 9

Figure 1-1 (Part 8 of 11). System/32 System Control Program Overview

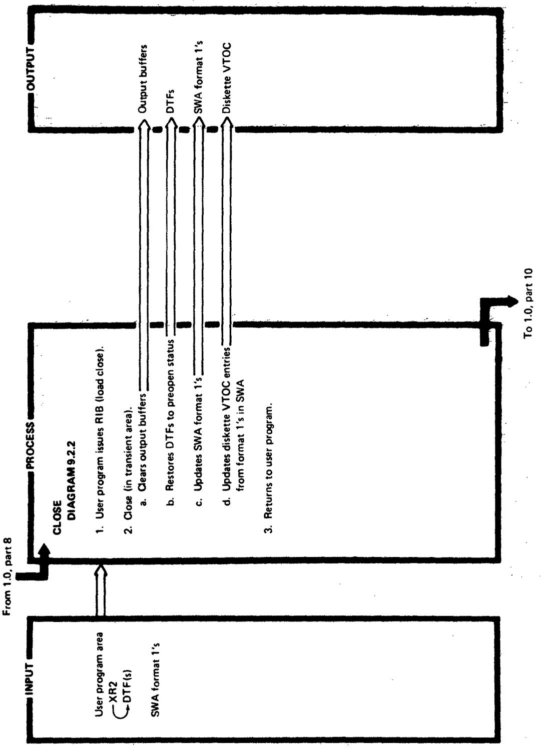
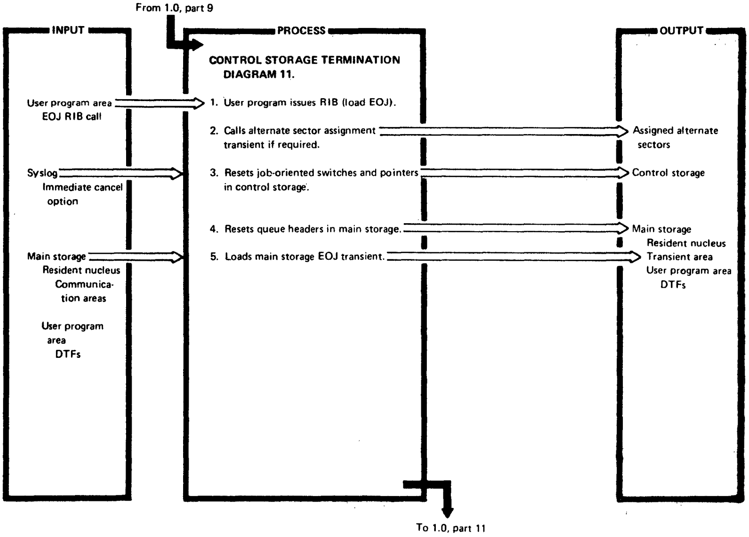

INPUT
User program area
XR2
DTF(s)
SWA format 1's
From 1.0, part 8
PROCESS
CLOSE
DIAGRAM 9.2.2
1. User program issues RIB (load close).
2. Close (in transient area).
a. Clears output buffers
b. Restores DTFs to preopen status
c. Updates SWA format 1's
d. Updates diskette VTOC entries from format 1's in SWA
3. Returns to user program.
To 1.0, part 10
OUTPUT
Output buffers
DTFs
SWA format 1's
Diskette VTOC

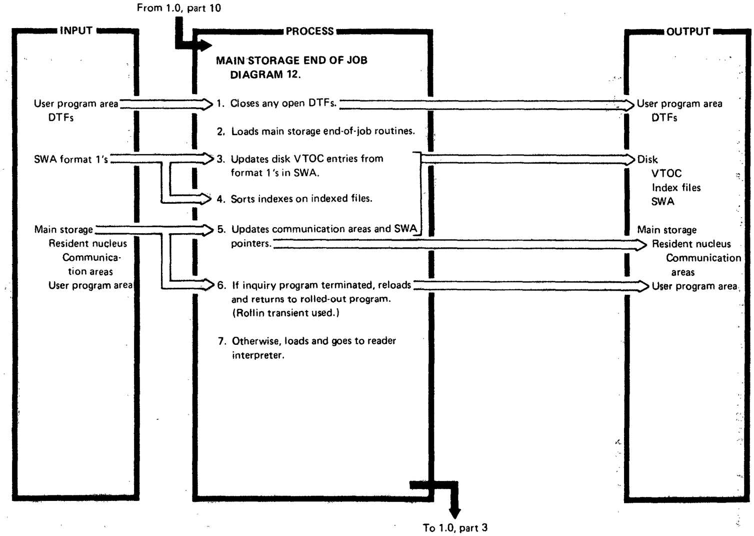
Figure 1-1 (Part 9 of 11). System/32 System Control Program Overview



CONTROL STORAGE TERMINATION DIAGRAM 11.

1. User program issues RIB (load EOJ).
2. Calls alternate sector assignment transient if required.
3. Resets job-oriented switches and pointers in control storage.
4. Resets queue headers in main storage.
5. Loads main storage EOJ transient.

Figure 1-1 (Part 10 of 11). System/32 System Control Program Overview



From 1.0, part 10

INPUT

User program area
DTFs

SWA format 1's

Main storage
Resident nucleus
Communication areas
User program area

PROCESS

MAIN STORAGE END OF JOB
DIAGRAM 12.

1. Closes any open DTFs.
2. Loads main storage end-of-job routines.
3. Updates disk VTOC entries from format 1's in SWA.
4. Sorts indexes on indexed files.
5. Updates communication areas and SWA pointers.
6. If inquiry program terminated, reloads and returns to rolled-out program. (Rollin transient used.)
7. Otherwise, loads and goes to reader interpreter.

OUTPUT

User program area
DTFs

Disk
VTOC
Index files
SWA

Main storage
Resident nucleus
Communication areas
User program area

To 1.0, part 3

Figure 1-1 (Part 11 of 11). System/32 System Control Program Overview

IBM System/32 System Data Areas and Diagnostic Aids, SY21-0532

- System Overview
- Data Areas
- Diagnostic Aids

IBM System/32 System Logic Manual, SY21-0567

Component	Diagram	Chapter
Main Storage IPL	3	1
Reader Interpreter	4	2
Initiator	5	3
Open and Close Functions of Data Management	9.2	4
Data Management	6	5-10.1
Linkage Editor	7.1	11
Overlay Linkage Editor	7.2, 7.3	11.1
SCP Transients and Main Storage Functions	9.1, 9.3	12
Library Maintenance	10	13
Main Storage End of Job	12	14
SCP Utilities	13	15-30
Diagnostic Aids	14	31

IBM System/32 Control Storage Logic Manual, SY21-0533

Component	Diagram
Control Storage IPL	2
Nucleus Functions and Trace	8
Control Storage Termination	11
Emulator	15
Input/Output Control Code	16
Alter/Display	17
Machine Check Interrupt	18
Micro Interrupt Hardware Handler	19
Extended Control Storage Supervisor (ECSS)	20
Scientific Instruction Set (SIS) Interpreter	21

IBM System/32 Data Communications Logic Manual, SY21-0551

Component	Diagram	Chapter
BSC Programming	20.1	1
MRJE	20.2	2
Print Utility	20.3	3
SNA	20.4	4
SDLC	20.5	5
Batch Work Station (RJE)	20.6	6
Batch Work Station (CICS/IMS)	20.7	7

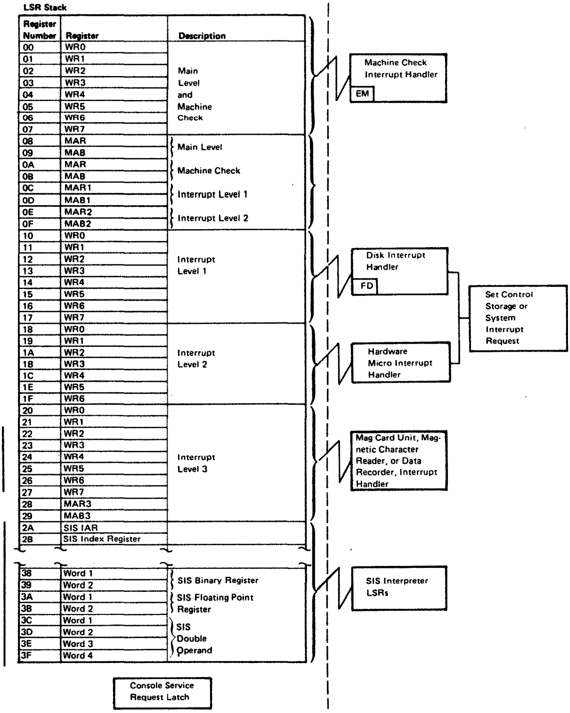
IBM System/32 Word Processing Logic Manual, SY34-0069

Component	Diagram	Chapter
Mag Card Unit	21	1
Job Control	22	2
Library Control	23	3
Word Processing Utilities	24	4-13
Word Processing Communications	26	14

IBM System/32 1255 Magnetic Character Reader Reference and Logic Manual, GC21-7692

Component	Diagram	Chapter
SUBR08 Data Management Overview	25.1	9
Specification Analysis	25.2	9
Specification Error Message Handler	25.3	9
Stacker Select Routine	25.4	9

Figure 1-2. System/32 SCP Overview Table of Contents



LSR Stack

Register Number	Register	Description
00	WR0	Main Level and Machine Check
01	WR1	
02	WR2	
03	WR3	
04	WR4	
05	WR5	
06	WR6	
07	WR7	
08	MAR	Main Level
09	MAB	
0A	MAR	Machine Check
0B	MAB	
0C	MAR1	Interrupt Level 1
0D	MAB1	
0E	MAR2	Interrupt Level 2
0F	MAB2	
10	WR0	Interrupt Level 1
11	WR1	
12	WR2	
13	WR3	
14	WR4	
15	WR5	
16	WR6	
17	WR7	
18	WR0	Interrupt Level 2
19	WR1	
1A	WR2	
1B	WR3	
1C	WR4	
1E	WR5	
1F	WR6	
20	WR0	Interrupt Level 3
21	WR1	
22	WR2	
23	WR3	
24	WR4	
25	WR5	
26	WR6	
27	WR7	
28	MAR3	
29	MAB3	
2A	SIS IAR	
2B	SIS Index Register	
38	Word 1	SIS Binary Register
39	Word 2	
3A	Word 1	SIS Floating Point Register
3B	Word 2	
3C	Word 1	SIS Double Operand
3D	Word 2	
3E	Word 3	
3F	Word 4	

Machine Check Interrupt Handler
EM
Disk Interrupt Handler
FD
Set Control Storage or System Interrupt Request
Hardware Micro Interrupt Handler
Mag Card Unit, Magnetic Character Reader, or Data Recorder, Interrupt Handler
SIS Interpreter LSRs
Console Service Request Latch

Figure 1-3 (Part 1 of 3). Control Storage Data Flow

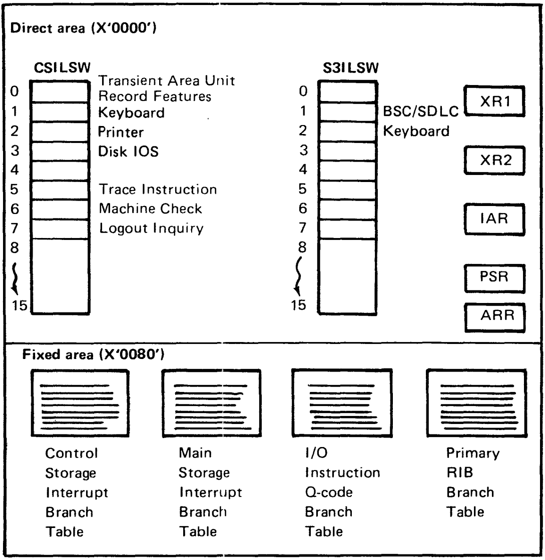


Figure 1-3 (Part 2 of 3). Control Storage Data Flow

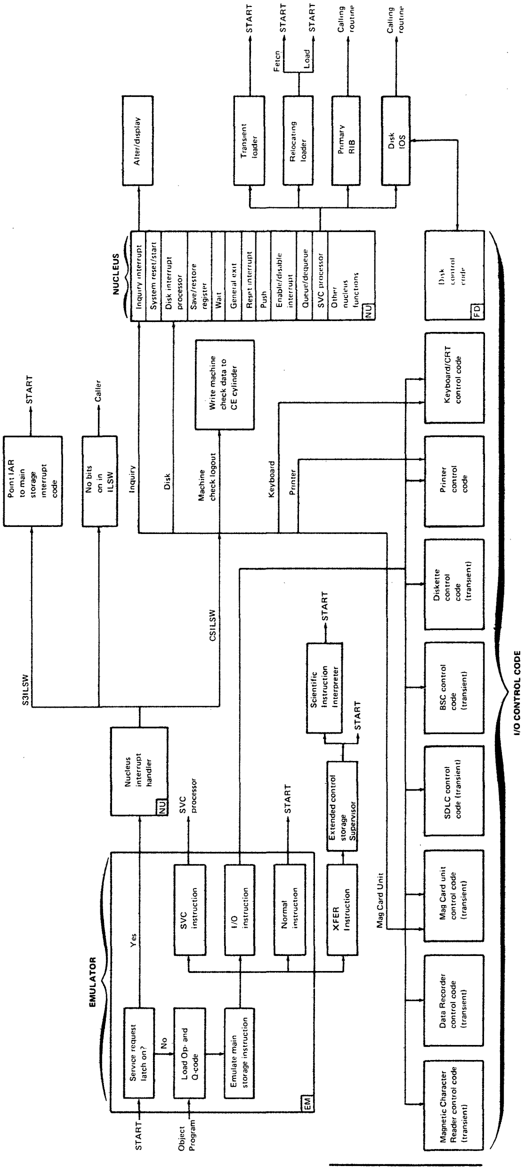


Figure 1-3 (Part 3 of 3). Control Storage Data Flow

Section 2. Data Areas

This section describes the storage areas in the IBM System/32 system (control storage, main storage, disk and diskette) and the applicable data areas for each. Figure 2-1 shows storage organization for the IBM System/32 system.

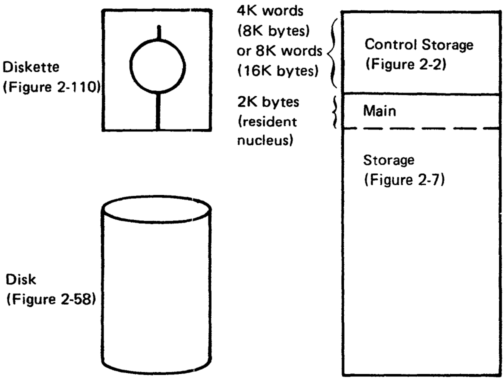


Figure 2-1. General IBM System/32 Storage Map

CONTROL STORAGE ASSIGNMENT

Control storage is an area of 4K words (8K bytes) or 8K words (16K bytes) composed entirely of microcode. The microcode in the first 4K words (8K bytes) of control storage performs the following functions:

- IPL and termination functions of control storage
- SCP nucleus functions
- Emulation
- I/O control microcode functions
- Alter/display function
- Handles interrupts

The microcode in the second 4K word (8K bytes) of control storage (which is optional) performs the following functions:

- Allocates and loads the microcode in the available area of the control storage increment
- Deallocates and resets the microcode in the available area of the control storage increment
- Passes control to the code located in the available area of the control storage increment
- Emulates the scientific instruction set

Figure 2-2 shows the general layout of control storage.

For a detailed description of the functions performed by control storage, see the *IBM System/32 Control Storage Logic Manual*, SY21-0533.

Displ of Leftmost Word in Hex	Lng in Words	Description
0	128	Direct area (Figure 2-4)
80	64	Fixed communications area
C0	64	Keyboard Katakana converter
100	256	Keyboard decode and display screen buffer
200	896	Emulator (see note immediately following *Transient area* in this figure)
580	384	Disk I/O control
700	384	Printer I/O control
880	384	Keyboard/CRT I/O control
A00	512	Transient area (occupied by control storage IPL routine at IPL time)

Transient Number	Name
0	Alter/display (see note)
1	Diskette I/O control
2	Data recorder, mag card, or magnetic character reader
3	Magnetic character reader diagnostics
4	BSCA—ASCII
5	BSCA—EBCDIC and SDLC
6	EOJ/set trace
7	Emulator (see note)
8	Alternate sector assignment
9	DIAG01
A	DIAG02
B	BSCA and SDLC wrap test
C	Single form/ledger cards
D	Reserved
E	Reserved
F	Reserved

Note: Alter/display and emulator share the same area at X'200' (which is not part of the actual transient area).

Figure 2-2 (Part 1 of 4). Control Storage Map

Displ of Leftmost Word in Hex	Lng in Words	Description
C00	65	Micro interrupt handler
C41	959	Nucleus functions:

Name	Entry	Function
NUFRS	NUFROMST	Retrieve a stack entry
NUSTR	NUSTORE	Replace a stack entry
NUADD	NUADDMS	Add a register to main storage
NUKIN	NUKINQRY	Inquiry interrupt process
NUMMM	NUMOVEMM	Move main storage to main storage
NUHTR	NUHTRACE	Log out stack entry
NURTN	NURETURN	Return from main storage transient
NUEXT	NUEXIT	General exit—no unstack
	NUEXITUS	General exit—unstack
NUYEN	NUYDISBL	Disable interrupt
	NUYENABL	Enable interrupt
NUXNT	NUXIENT	Transient loader
	NUXNTEOJ	
NUQUE	NUQUEUE	Queue manipulation
NUDIO	NUDISKIO	General disk I/O interface
	NUDRDMS	Disk read into main storage
	NUDWRMS	Disk write from main storage
	NUDXXCSX	Disk general control storage interface
NUCOM	NUCOMPT	Return communication area address
NUUST	NUUNSTK	Unstack/ignore stack entry
NUWAT	NUWAIT	Wait
NUOUT	NUOPUSH	Push main storage area to disk
NUSVC	NUSVCCS	Initial control storage SVC process
	NUSVCMS	Initial main storage SVC process
NULDR	NULOADER	Relocating loader
NUZIN	NUZPULL	Pull pushed area into storage
NUGET	NUGETXNT	Load control storage transient
NURSV	NUREST	Unstack/restore registers 1-6
	NURSAVCS	Stack registers 1-6
	NURUNSTK	Unstack/ignore stack entry

Figure 2-2 (Part 2 of 4). Control Storage Map

Displ of Leftmost Word in Hex	Lng in Words	Description		
C41	959	Nucleus functions (continued):		
		Name	**Entry**	**Function**
		NUTER	*Note:* Following are separate entry points for each nucleus error condition to terminate job through syslog.	
			NUTERLD1	Load from transient
			NUTERLD2	Load below user area
			NUTERLD3	Relocate attempt without RLDs
			NUTERSV1	SVC not from first 256 bytes
			NUTERPS1	Disk push area overflow
			NUTERPS2	Push stack overflow
			NUTERPS3	Push stack underflow
			NUTERST1	Register stack overflow
			NUTERST2	Register stack underflow
			NUTERSS1	Invalid stack store displacement
			NUTERIO1	Invalid disk IOB parameters
			NUTERIO2	Unrecoverable disk I/O error
			NUTERIO3	Disk interrupt timeout check
			INVALIDQ	Invalid main storage instruction Q-code
			NUTERS3A	Invalid main storage instruction address
			INVALOP	Invalid main storage instruction op code
			NUTERDUM	Dump/terminate job
			NUTERGE1	Invalid control storage transient number
			NUTERPCK	Control storage processor check
			NU$ABORT	Error while processing another error
		NURES	NURESET	System reset and start
		NUNRD	NUNREAD	Disk read/write interface
		NU$RE	NU$RESET	Reset main storage interrupt
			NU$RSETC	Reset control storage level 1 interrupt
			NU$RUSET	Reset control storage level 0 interrupt
		NU$IN	NU$INTER	Nucleus interrupt handler
		NU$DK	NU$DKIOS	Disk IOS
		NU$XI ($NU1) or	NU$XIOB1	Initiate next disk IOB (NU$DK)
		NU$XT ($NU2)	NU$XIOB2	Initiate next disk IOB (NU$EI)

Figure 2-2 (Part 3 of 4). Control Storage Map

Displ of Leftmost Word in Hex	Lng in Words	Description
C41	959	Nucleus functions (continued):

Name	Entry	Function
NU$NI	NU$NEXT1 NU$NEXT2	Start next disk operation
NU$EI	NU$ENDIO	Terminate disk IOB
NU$GI ($NU1) or NU$TI ($NU2)	NU$GIOB	Initialize disk IOB processing
NU$MI	NU$MINTR	Disk interrupt processing
NU$LI	NU$LDREG	Get disk IOB values into registers
NUINL	NUINLINE	Process inline parameters
NUVCM	NUVMOVCM	Move control storage to main storage
NUPST	–	Register save stack
NUBRT	–	RIB status table
NUPCH ($NU1) or NUTCH ($NU2)	–	Patch area

Figure 2-2 (Part 4 of 4). Control Storage Map

Displ of Leftmost Word in Hex	Lng in Words	Description
1000	50	Control Storage increment communication area (See Figure 2-5)
1032	206	Extended control storage supervisor
1100	3839	Available area (scientific instruction set interpreter)
1FFF	1	Reserved for ECS supervisor

● **Figure 2-3. Control Storage Increment Map (Additional 4K Words)**

Direct Area

The direct area is a 128-word area of control storage beginning at control storage location X'0000'. This area is used primarily by the emulator and I/O control code.

Format

Fields of interest in the direct area are described in Figure 2-4. The location is represented in hex words. For a complete description of this area, see the ***IBM System/32 Control Storage Logic Manual,*** SY21-0533.

Label	Control Storage Location	Description
XR1	76	Main storage index register 1
XR2	77	Main storage index register 2
OPQ	78	Main storage operation code and Q-byte
IAR Program registers	79	Main storage instruction address register (points to main storage user area X'0800' for IPL)
PSR	7A	Main storage program status register (set to equal (EQ) for IPL; see Figure 2-6 for a description of other possible settings)
ARR	7B	Main storage address recall register
DECRECAL	7C	Address recall register for decimal operations
TRACE	7D	Address of operation code being executed (used for retry of instruction after machine check)
CSILSW	7E	Control storage interrupt level status word: *First byte:*
	Interrupt level 0	X'80' = Transient I/O features X'40' = Keyboard interrupt X'20' = Printer interrupt X'10' = Disk IOS
	Interrupt level 1	X'04' = Trace instruction X'02' = Machine check logout interrupt X'01' = Inquiry interrupt
		Second byte: Not used; set to zeros.
S3ILSW	7F	Main storage interrupt level status word: *First byte:* X'40' = BSCA/SDLC X'20' = Keyboard interrupt *Second byte:* Not used; set to zeros.

Figure 2-4. Control Storage Direct Area Fields

Displ of Leftmost Word in Hex	Lng in Words	Description
1000	1	XR = Scientific index register
1001	1	AR = Scientific address register (holds address for certain scientific operands)
1002	6	BR = Scientific binary register
1008	12	FSIGN FR = Scientific floating point register (holds long or short precision floating point hex values)
1014	1	XM = Scientific index multiplier register
1015	1	IAREG = Scientific address register (next scientific instruction)
1016	1	Bits 0-7 = unused Bits 8-15: CCREG = Scientific condition code register
1017	1	Constants X'03' and X'04'
1018	1	Constant X'0001'
1019	1	Constant X'FFFF'
101A	1	Constant X'FFFF'
101B	1	Constant X'0002'
101C	1	Constant X'8000'
101D	1	Constant X'0000'
101E	1	Constant X'0000'
101F	1	Constant X'0000'
1020	11	FINTM1 = Temporary work area
102B	1	SCADDR used to temporarily hold addresses
102C	6	Unused

● **Figure 2-5. Control Storage Increment Communication Area**

Figure 2-6. Program Status Register Settings

Binary Value	**8**	**4**	**2**	**1**	**8**	**4**	**2**	**1**
Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Meaning			Binary overflow (BO)	Test false	Decimal overflow (DO)	HI	LO	EQ
Decimal								
Add decimal			—	—	overflow	> zero	< zero	zero
Sub decimal			—	—	overflow	> zero	< zero	zero
Zero and add			—	—	—	> zero	< zero	zero
Logical								
Add to register			overflow	—	—	1 > 2	1 < 2	zero
Add logical			overflow	—	—	1 > 2	1 < 2	zero
Sub logical			—	—	—	1 > 2	1 < 2	zero
Compare			—	—	—	1 > 2	1 < 2	EQ
CLI			—	—	—	1 > I	1 < I	1 = I
Edit (second operand)			—	—	—	> zero	> zero	zero
Test bits ON			—	(note 1)	—	—	—	—
Test bits OFF			—	(note 2)	—	—	—	—
Branch or jump on condition (note 3)	(note 4)		—	(note 5)	(note 5)	—	—	—

Notes:

1. Selected bits are not all one.
2. Selected bits are not all zero.
3. Instructions are main storage instructions.
4. When 1, branch if any of the tested bits are ON; when 0, branch if all of the tested bits are OFF.
5. Turn off if tested.

MAIN STORAGE ASSIGNMENT

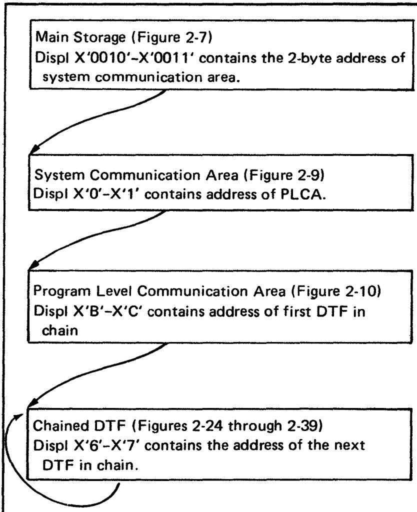
Figure 2-7 shows the assigned areas of main storage. The first 2K (2048) bytes of main storage are used by the IBM System/32 System Control Program. This area, which cannot be overlaid, is called the resident nucleus.

Address of Leftmost Byte (in hex)	Contents
00	Entry point for dump routine
03	CE dump program check byte: X'A5' = CE cylinder dump data is valid X'5A' = CE cylinder data already dumped
04	General entry point (resident function linkage)
07	Unused
08	Entry point for disk IOS
0B	Unused
0C	Entry point for wait
0F	Unused
10	Address of the system communication area (Figure 2-8)
12	Return from transient entry point
15	Unused
16	Address of keyboard interrupt handler
18	Pointer to current printer IOB
1A	Printer IOS switch: X'80' = Error occurred on last operation X'40' = Physical buffer free X'20' = Print operation has occurred X'03' = Belt speed check counter X'02' = Belt speed check retry X'01' = Belt speed check retry
1B	Interrupt IOB completion code byte (see PODDCMP field in printer IOB, Figure 2-16)
1C	Reserved (2 bytes)
–	System communication area (Figure 2-9)
–	Program level communication area (Figure 2-10)

Figute 2-7 (Part 1 of 2). Main Storage Organization

Address of Leftmost Byte (in hex)	Contents
—	Reserved area available for patch use
E9	Keyboard interrupt routine (23 bytes)
100	Transient area (1024 bytes)
500	Queue headers (Figure 2-13)
510	System disk IOB
528	Keyboard reset IOB (See Figures 2-14 through 2-23 for IOB descriptions.)
537	Printer IOBs (3)
549	Transient load table (3-byte entries—SSN) (RIB X'81'-X'98')
594	Disk error IOB
5C4	Resident nucleus error message
5EC	Keyboard reset message
600	Trace area (12-byte entries) Refer to TRACE function in Section 3.
6FC	ABEND error indicator (dump only) (see Appendix F for abnormal termination error MICs)
6FE	Pointer to next trace entry (dump only)
700–7FF	Printer translation table
800 to	User area
3FFF	16K main storage
5FFF	24K main storage
7FFF	32K main storage

Figure 2-7 (Part 2 of 2). Main Storage Organization



Main Storage (Figure 2-7)
Displ X'0010'-X'0011' contains the 2-byte address of system communication area.

System Communication Area (Figure 2-9)
Displ X'0'-X'1' contains address of PLCA.

Program Level Communication Area (Figure 2-10)
Displ X'B'-X'C' contains address of first DTF in chain

Chained DTF (Figures 2-24 through 2-39)
Displ X'6'-X'7' contains the address of the next DTF in chain.

For information on pointers to IOBs, see Figures 2-14 through 2-23.

Figure 2-8. System Linkage

System Communication Area (SCA)

The system communication area resides in the System/32 main storage nucleus and passes information between system programs.

How to Find

The 2-byte address of the SCA is one of the fixed main storage nucleus entry points and is located at X'0010' and X'0011' in main storage.

Format

Figure 2-9 shows the format of the system communication area.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	NCPL1	2	Address of program level communication area
2	NCSGEN	1	System usage byte: X'80' = Inquiry supported X'40' = Syslog transient called X'20' = IPL successful X'10' = Disk I/O error during DLOG X'08' = 13.7 megabyte disk X'04' = 9.1 megabyte disk X'02' = 5.0 megabyte disk X'01' = 3.2 megabyte disk
3	NCCMSG	2	Sector address of command message member (##MSG3)
5	NCHMSG	2	Sector address of headings message member (##MSG2)
7	NCSMSG1	2	Sector address of first level system message member (##MSG1)
9	NCSMSG2	2	Sector address of second level system message member (##MSG4)
B	NCSWRK	2	Sector address of scheduler work area (SWA)
D	NCSLOG	1	System log device: X'E0' = Printer and display screen X'10' = Display screen only
E	NCODIR	2	Sector address of library directory
10	NCOLIB	2	Sector address of start of library members

Figure 2-9 (Part 1 of 3). Format of System Communication Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
12	NCSCH	1	Data management/scheduler switches: X'80' = Printer interlock for data management X'40' = Rollin necessary X'20' = System using printer X'10' = Printer error recovery; call syslog X'08' = Rollout was requested X'04' = Use CSEOJ trace function X'02' = Rollout was performed X'01' = Inquiry request pending
13	NCSCH1	1	Scheduler switches: X'80' = Do not allow 1 option for syslist print error X'40' = System date received X'20' = Printer data management call X'10' = 8 LPI operating X'08' = Eject at end of job X'04' = World Trade date format—DDMMYY X'02' = Domestic date format—MMDDYY X'01' = International date format—YYMMDD
14	NCINQCTR	1	Inquiry enabled counter
15	NCLPSZ	1	Printer page size (in hex)
16	NCFDQUE	2	Disk queue header value when dump occurred
18	NCSVCINS	2	SVC instruction (op code and Q-byte)
1A	NCMBSV	1	Model indicator: X'80' = Attachment controller is loaded (NCAMLOAD) X'40' = CHAIN procedure (NCAMCHAN) X'20' = List RPG (NCAMRPG1) X'10' = Call cycle OCL suppress (NCAMCALL) X'08' = Offline dummy open request (NCAMOLDO) X'04' = Diskette basic exchange (NCAMI1BI) X'02' = System/32 model indicator bit X'01' = Attachment controller in nonoverlap mode (NCAMNOVL)
1B	NCROLLSS	2	Sector address of rollout area

Figure 2-9 (Part 2 of 3). Format of System Communication Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
ID	NCRSVD3	4	Reserved
21	NCCDSIO	2	Data recorder SIO counter: *First byte:* X'80' on = Output file X'80' off = Input file Bits 1-15 contain the number of SIOs issued for the current program. Maximum count is 32,767. If this is exceeded, the counter will wrap to zero without warning.
23	NCRDSIO1	2	Diskette read counter
25	NCRDSIO2	2	Diskette write counter
27	NCRDSIO3	2	Diskette seek counter
29	NCPRTSIO	2	Printer SIO counter
2B	NCKBSNS	2	Keyboard sense counter
2D	NCKBDTF@	2	Address of active keyboard DTF
2F	NCKBINLK	1	Keyboard interlock: X'80' = Request control from interactive keyboard X'40' = Interactive mode between records X'20' = Keyboard/display screen in use by data management X'10' = Interactive mode X'02' = Intermittent error retry counter X'01' = Horizontal sync check counter
30	NCSTORBG	20	Start of transient work area
44	NCSTOR	1	Last byte of transient work area
45	NCLOGSS	2	Sector address of syslog
47	NCLOGN	1	Number of sectors in syslog
48	NCEOLB	2	Sector address of library end

Figure 2-9 (Part 3 of 3). Format of System Communication Area

Program Level Communication Area (PLCA)

This area contains information relative to the program that is currently loaded.

How to Find

Bytes 0 and 1 of the system communication area contain the address of the program level communication area.

Format

Figure 2-10 shows the format of the program level communication area.

The first 11 bytes of the program level communication area beginning at NPCS are used for the parameter list save area (NPARM) by the control storage relocating loader; they cannot be used for other purposes. The initial format (prior to executing a find) of the parameter list save area is shown in Figure 2-44).

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	NPCS	2	Sector address of module
2	NP#S	1	Number of text sectors to be read
3	NPLNK	2	Link-edited address
5 [1]	NPRLD	1	Displacement of RLD in first sector containing RLDs
6	NPENT	2	Address of entry point
8	NPLOD	2	Main storage load address
A	NPEXTN	1	Extension for 8-byte name
B	NPDTF@	2	Address of first DTF in the last DTF chain opened
D	NPNAME	8	Program name
15	NPBEG	2	Program level beginning address
17	NPEND	2	Program level end address
19	NPQ	1	Program pack Q-byte

[1] Parameter list save area (NPARM).

Figure 2-10 (Part 1 of 6). Format of Program Level Communication Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
1A	NPUPSI	1	UPSI switches: X'80' = Switch 1 X'40' = Switch 2 X'20' = Switch 3 X'10' = Switch 4 X'08' = Switch 5 X'04' = Switch 6 X'02' = Switch 7 X'01' = Switch 8
1B	NPBCFG	1	First BSC configuration byte: X'80' = Full rate/half rate X'40' = Internal clock indicator X'20' = IBM modem indicator X'10' = World Trade answer tone indicator X'08' = Standby line indicator X'04' = Multipoint tributary line X'03' = World Trade switched line X'02' = Domestic switched line X'01' = Point-to-point indicator
1C	NPBCFG2	1	Second BSC configuration byte: X'80' = Debug facility X'40' = BSCA active/mini-log X'20' = MRJE active
1D	NPBLCD	1	Switched line connection type X'C0' = Manual call X'80' = Manual answer X'40' = Auto answer
1E	NPBTR@	1	Tributary station address
1F	NPBTR#	1	Reserved
20	NPBRSP	1	Reserved
21	NPBERC	1	Error retry count
22	NPBEHI	1	Error logging indicator
23	NPBWRK	2	Pointer for trace output
25	NPRSVD1	2	Reserved
27	NPRDPRE	1	Diskette indicator: X'00' = 128 byte sectors X'02' = 512 byte sectors

Figure 2-10 (Part 2 of 6). Format of Program Level Communication Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
28	NPSYSL	1	Syslist device: X'E0' = Printer X'10' = Display screen X'00' = List off (null)
29	NPINTRLK	1	Program level interlock byte: X'80' = Sysin transient called X'40' = Diskette build transient called X'20' = Module has utility control statements X'10' = Read diskette volume label X'08' = Syslist overflow occurred X'04' = VTOC read/write called X'02' = Utility control statements flushed X'01' = UPSI switch saved
2A	NPSWSAVE	1	UPSI save switch
2B	NPSINSS	2	Sysin sector address
2D	NPSINN	1	Number of sectors—sysin
2E	NPSYSI	1	Sysin device indicator: X'A0' = Source sysin X'10' = Keyboard sysin
2F	NPSCH1	1	Reader/interpreter switches: X'80' = // DATE received (intramode) X'40' = // COMPILE received X'20' = // SWITCH received X'10' = Merge procedure X'08' = Override to procedure statement received X'04' = Intrastep mode X'02' = Interstep mode X'01' = IPL mode
30	NPSCH2	1	Scheduler switches: X'80' = Continuation X'40' = Include procedure X'20' = Cancel inquiry X'10' = File statement read X'08' = Display screen saved X'04' = Flush remainder of step or job X'02' = Immediate cancel X'01' = Controlled cancel

Figure 2-10 (Part 3 of 6). Format of Program Level Communication Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
31	NPSCH3	1	Scheduler switches: X'80' = Tag sort required X'40' = Allocate transient X'20' = Source required X'10' = Allocate rollout (7K) X'08' = Multivolume file has been allocated X'04' = Additional procedure statement read; also used to indicate disk DTF allocated Scheduler switches (continued): X'02' = First LOAD/RUN job read X'01' = Prepare new diskette
32	NPSCH4	1	Scheduler switches: X'80' = Operational bit (program level is active) X'40' = // IMAGE received X'20' = // FORMS received X'10' = Disk needed for allocation X'08' = Diskette needed for allocation X'04' = Deferred allocate registered X'02' = Resource allocation X'01' = Nested procedure
33	NPSCH8	1	Scheduler switches: X'80' = EOJ print indicator X'40' = DTF chain error X'20' = Valid FILE statements received X'10' = Shared I/O program X'08' = I- or B-type program X'04' = End of procedure reached X'02' = Enable INQ key (normal inquiry is disabled) X'01' = INQ key pressed
34	NPPMSG1	2	Sector address of level 1 program product message member
36	NPEOJ	1	End-of-job ID: X'80' = EOJ call #OXRF X'40' = EOJ scheduler bit X'20' = EOJ halt indicator X'10' = EOJ sort control X'0E' = Reserved for RPG X'01' = Do not close DTFs at EOJ
37	NPPMSG2	2	Sector address of level 2 program product message member

Figure 2-10 (Part 4 of 6). Format of Program Level Communication Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
39	NPUMSG1	2	Sector address of level 1 user message member
3B	NPUMSG2	2	Sector address of level 2 user message member
3D	NPDATE	6	Program date
43	NPRDVOL	6	Diskette volume ID
49	NPDCFG1	1	SDLC line type definition: X'80' on = Full rate X'80' off = Half rate X'40' on = Internal clocking X'40' off = Modem clocking X'20' = IBM modem X'10' = World Trade answer tone X'08' = Switched standby line X'04' = Multipoint tributary line X'03' = World Trade switched line X'02' = Domestic switched line X'01' = Point-to-point leased line
4A	NPDCFG2	1	Timeout value in seconds
4B	NPDCFG3	1	SDLC station address
4C	NPDCFG4	1	UDT reserved byte
4D	NPDIND	1	SDLC indicator byte: X'80' = SDLC trace in progress (debug) X'40' = SDLC adapter active X'20' = #SDJ5 called by #SDJ0 X'0C' = Manual call X'08' = Manual answer X'04' = Auto answer
4E	NPDUID	3	XID information field: Bits 0-3 = Last hex character of System/32 identifier Bits 4-23 = Unique ID field
51	NPDRES	2	Reserved
53	NPWPKBDF	1	Keyboard definition ID
54	NPWPGSTA	1	Mag card microcode status indicator: X'80' = Microcode loaded X'10' = WPWRKFILE VTOC search done X'08' = MCU active

Figure 2-10 (Part 5 of 6). Format of Program Level Communication Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
55	NPWPGWAD	2	Address of word processing work area
57	NPWPFMT1	3	Format 1 address of word processing work file
5A	NPWPBTID	1	Belt ID

Figure 2-10 (Part 6 of 6). Format of Program Level Communication Area

Word Processing Work Area (WPWA)

This work area contains the data needed to define and run word processing.

How to Find

The two bytes starting at X'55' of the PLCA contain the address of the WPWA.

Format

Figure 2-11 shows the format of the word processing work area.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	WPWST1	1	System status byte 1: X'80' = Mag card unit is sysin device X'40' = Keyboard/display is sysin device X'20' = Disk file is sysin device X'10' = Document library is the sysin device X'08' = Mag card unit attached X'02' = 32K storage X'01' = 24K storage X'00' = 16K storage
1	WPWST2	1	System status byte 2: X'80' = Sysin data in WPRDBFAD X'40' = Nonstop mode X'20' = Stop at end of job X'10' = Stop at end of task X'08' = Immediate stop requested X'04' = Inquiry in process X'02' = Keyboard/display in use at inquiry time X'01' = Issue second screen for inquiry
2	WPWST3	1	System status byte 3: X'80' = $WPJAT first time through X'40' = Sysin device opened X'20' = Document library is opened X'10' = Job statement expected X'08' = If on only ,, statements valid X'04' = Terminate this job X'02' = Checkpoint error has occurred X'01' = Terminate this task
3	WPAPPCSC	2	Sector address of procedure parameters
5	WPAPPDCS	1	Displacement of procedure parameters
6	WPSSGED@	2	Source get end address
8	WPSSGCRT	2	Source get current address
A	WPSSGDSP	1	Source get current displacement
B	WPSSGLNR	1	Source get length of record
C	WPDTFLOC	72	DTF area for word processing
54	WPIOBMCR	36	IOB area for MCU
78	WPKBDFBM	1	Keyboard rollout definition
79	WPWLDM	2	LDAM main module address

Figure 2-11 (Part 1 of 2). Format of Word Processing Work Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
7B	WPWMCR	11	MCUAM module linkage
or			
7B	WPSVXR1	3	Store XR1 in MCU DTF instruction
7E	WPSVARR	3	Store ARR in MCU DTF instruction
81	WPBRNCH	4	Branch to general entry
85	WPRIB80	1	Using RIB 80
86	WPSSN	3	S/S/N for MCUAM
89	WPWJOBID	4	Job identification
8D	WPJACKEY	4	Job access key
91	WPIOBSIN	22	Sysin IOB area
A7	WPIOASIN	256	The read buffer for sysin = MCU and the keyboard deblock area for sysin = fixed disk
1A7	WPMENLO	163	Application work area
24A	WPRBDFAD	2	Address of sysin read buffer
24C	WPFDRDBF	128	Fixed disk sysin read buffer
2CC	WPMCUSEL	1	MCU tilt/rotate table select byte
2CD	WPROLLCD	17	Word processing rollout code
2DE	WPSVACCM	4	Save area for base access map
2E2	WPKBDFLT	1	Default keyboard ID
2E3	WPRESRV1	1	Reserved
2E4	WPRESRVD	30	Reserved

Figure 2-11 (Part 2 of 2). Format of Word Processing Work Area

Word Processing Library Communication Area

This area contains all the data required for communication with the linked direct access method (LDAM). It contains the member name of the document to be worked on and the function to be performed on that document. LDAM returns a completion code and other data to the user.

How to Find

The word processing library communication area is defined in the application program for word processing.

Format

Figure 2-12 shows the format of the word processing library communication area.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	CMFUNC	1	Function to be performed on document.
1	CMNAME	16	Member name (from document library)
11	CMCOMP	1	Completion code
12	CMNEWN	16	New member name
or 12	CMRELA	2	Relative entry address to WPLAMT
14	CSDADD	2	Address of the document storage record area
16	CLEVEL	1	Level of document
17	CMACCN	1	Access number
18	CMACCM	4	Access map
1C	CSRELA	2	Document storage relative record number
1E	CRESVD	4	Reserved

Figure 2-12. Format of Word Processing Library Communication Area

Queue Header Table

The queue header table is a 16-byte area of main storage containing the addresses of the queue header (the first IOB in the queue) for each supported device.

How to Find

The queue header table begins at location X'0500' in main storage.

Format

Figure 2-13 shows the format of the queue header table.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	MSQUEUES	2	Reserved; set to X'FFFF'
2	QHDBSCA	2	Address of queue header for BSC or SDLC (devices are mutually exclusive)
4	QHDKBCRT	2	Address of queue header for keyboard
6	QHDPRT	2	Address of queue header for printer
8	QHDDATRC or QHDMICR	2	Address of queue header for 129/5496 data recorder or mag card unit or 1255 magnetic character reader (devices are mutually exclusive)
A	Reserved	4	Set to X'FFFFFFFF'
E	QHDFD	2	Address of queue header for disk.

Figure 2-13. Queue Header Table Format

Input/Output Block (IOB)

An IOB is the interface between the user (usually data management) and IOS. The area for an IOB must be assigned by the calling routine.

How to Find

If data management is the caller, the postopen DTF normally will point to the IOB. The exception to this is the printer; printer data management IOBs can be found starting at main storage address X'0537'. Active IOBs (which can be found by referring to the system queue headers starting at main storage location X'0500') can be chained; bytes 0 and 1 of the IOB will point to the next IOB on the chain. If a permanent disk error occurs, the error IOB is saved starting at main storage location X'0594'. When IOS or wait is called, XR1 must contain the IOB address.

Format

IOBs for different units have different formats. The IOB formats are shown as follows:

Unit	Figure
Disk	2-14
Diskette	2-15
Printer	2-16
Keyboard/CRT	2-17
BSC	2-19
Data recorder	2-20
Magnetic character reader	2-21
Mag card unit	2-22
SDLC	2-23

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	IOBCHN	2	Storage address of the next active IOB in the chain. This area is always present, even when chaining is not used, and contains X'FFFF' if it is the last IOB in the chain. When the operation specified by this IOB is complete, this area contains the disk address last used (SS).
2	IOBCMP	1	Completion code: X'40' = Normal completion/scan hit X'41' = Permanent I/O error X'42' = Scan not hit X'44' = Scan equal hit Completion code (IOS use only): X'80' = Call from control storage X'20' = Main operation issued X'10' = Verify has been issued X'08' = Seek has been issued X'04' = Seek before verify issued
3	IOBQB	1	Q-byte: X'A0' = Control X'A1' = Read X'A2' = Write X'A3' = Scan
4	IOBRB	1	R-byte: *Control only:* X'00' = Seek X'01' = Recalibrate *Read and write:* X'80' = Repeat same data n times X'40' = Control store low only X'08' = Fast sync extended X'04' = Control storage select X'01' = ID read/write X'00' = Data

Figure 2-14 (Part 1 of 3). Format of Disk IOB

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
			Read only: X'03' = Read verify X'02' = Read diagnostic *Scan only:* X'02' = Scan high or equal X'01' = Scan low or equal X'00' = Scan equal
5	IOBSS	2	Sector address
7	IOBNB	1	Number of sectors minus 1, in hex, involved in data transfer
8	IOBDAT	2	Data field address
A	IOBSNS1	2	Sense bytes 0 and 1: *Byte 0:* X'80' = Not ready X'40' = Alternate sector processing X'20' = Sector sync check X'10' = Off-track check X'08' = Cyclic redundancy check X'04' = Parallel parity check X'02' = Writer echo check X'01' = Channel overrun *Byte 1:* X'80' = No-op X'40' = Device check (unsafe) X'20' = Invalid seek X'10' = Attachment equipment check X'08' = No record found X'04' = Scan equal hit X'02' = Scan not hit X'01' = Seek check
C	IOBSNS2	2	Sense bytes 2 and 3: *Byte 2:* X'80' = SERDES check X'20' = Channel transfer error X'10' = Reserved X'08' = Interrupt timeout check *Byte 3:* X'40' = Select unsafe X'20' = Write unsafe X'10' = Brake failure X'08' = Servo unsafe X'04' = 13.7 megabyte disk X'02' = 9.1 megabyte disk X'01' = 3.2 megabyte disk

Figure 2-14 (Part 2 of 3). Format of Disk IOB

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
E	IOBERR	1	Retry count in hex (maximum = 16)
F	IOBFLG	1	Flag byte: X'80' = Do not attempt error recovery X'40' = Do not verify after write data operation X'20' = Transient area cannot be overlaid (do not log errors) X'10' = User supplied NFCCHS field (see IOB extension) *Note:* IOBSS and IOBNB not valid if this bit is on. X'08' = Reserved for use by RPG compiler X'04' = Do not return on permanent error X'02' = Reserved for nucleus functions X'01' = Do not seek before operation *Note:* Reserved bits must be set to 0 when used for other than internal control storage or disk ERP purposes.
10	IOBPRTY	1	Priority byte (used by IOS queue)
11	IOBPAD	1	Reserved
		End of Basic IOB	
Beginning of IOB extension for data management			
12	IOBDCH	2	Data management chain address
14	IOBDTF	2	DTF address
IOB extension for user-supplied NFCCHS control field (IOBFLG X'10' on)			
12	IOBN	1	N byte (number of sectors minus 1)
13	IOBF	1	Flag byte: X'04' = Bad data may be written in data field X'03' = Defective alternate sector X'02' = Defective primary sector (alternate assigned) X'01' = Good alternate sector X'00' = Good primary sector
14	IOBCC	2	Cylinder number (hex)
16	IOBH	1	Head number
17	IOBS	1	Sector number (X'00'-X'3B')

Figure 2-14 (Part 3 of 3). Format of Disk IOB

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	IIOBCHN	2	Address of the last sector processed
2	IIOBCMP	1	Diskette completion code: X‘40’ = Successful completion X‘41’ = Permanent I/O error X‘42’ = End of volume X‘43’ = Not ready X‘49’ = Unsupported control record
3	IIOBQB	1	Q-byte: X‘D0’ = Seek X‘D1’ = Read data X‘D2’ = Read ID X‘D4’ = Read data/CAM; D and F control records are not squeezed out as in read data X‘D5’ = Write data/verify X‘D6’ = Write control address marks/ verify X‘D7’ = Write ID/verify
4	IIOBRB	1	R-byte: X‘80’ = Seek after X‘00’ = Null
5	IIOBSS	2	Sector address
7	IIOBNB	1	Number of sectors, minus 1, involved in data transfer
8	IIOBDAT	2	Data buffer address
A	IIOBSNS1	2	Sense bytes 0 and 1: *Byte 0:* X‘80’ = Missing data address mark (2 consecutive AMs found) X‘40’ = CRC found in ID field X‘20’ = CRC found in data field X‘10’ = Cylinder byte in ID miscompare X‘08’ = Head byte in ID miscompare X‘04’ = Record byte in ID miscompare X‘02’ = Length byte in ID miscompare X‘01’ = First ID has been found (no error)

Figure 2-15 (Part 1 of 2). Format of Diskette IOB

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
			Byte 1: X'80' = Due to prior condition, no action attempted X'40' = Invalid control record found (not D or F) X'20' = Lines to diskette were not set/read correctly X'10' = Control record(s) squeezed out X'08' = Physical cylinder number greater than 76
C	IIOBSNS2	2	Sense bytes 2 and 3: *Byte 2:* X'80' = Index pulses too close together X'40' = Index pulses too far apart X'20' = End-of-cylinder found (not tested) X'10' = No IDs on track X'08' = Read overrun X'02' = Write overrun X'01' = Write parity check *Byte 3 (not referred to in SCP):* X'80' = Head is loaded X'40' = Low write current is set X'20' = Write gate is on X'10' = Erase gate is on X'08' = Seek to track 3 or 0 is on X'04' = Seek to track 0 or 1 is on X'02' = Seek to track 1 or 2 is on X'01' = Seek to track 2 or 3 is on
E	IIOBERR	1	Error retry count
F	IIOBFLG	1	Flag byte: X'80' = No ERPs attempted X'40' = Automatic error display and correction allowed X'20' = No error logging X'10' = Allow seek past logical cylinder 74 (to 75 or 76) X'08' = Do not return to user program if completion code is X'41', or X'49' X'01' = User supplied control field (CHRNX)
10	Reserved	1	Reserved
11	IIOBEXP	1	Sector size
12	IIOBXR2	2	Save area for XR2
14	IIOBARR	2	Save area for ARR
16	IIOBDTF	2	DTF address

Figure 2-15 (Part 2 of 2). Format of Diskette IOB

Printer IOBs

There are three system printer IOBs in the main storage nucleus (Figure 2-7) for use by printer data management. Routines such as syslog (RIB X'85') use printer IOS and would furnish their own IOB.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	PODCHAIN	2	Address of the next IOB in the chain
2	PODDCMP	1	Printer completion code: X'40' = Operation complete X'20' = IOB active
	(overlays)		
2	PODDQ	1	Printer Q-byte: X'80' = Check for unprintable characters and set status bit if detected X'40' = IOB operation complete X'20' = IOB operation (indicated by bits 6 and 7) is in progress; used by microcode to maintain active status of IOB X'03' = Print X'02' = Skip desired X'01' = Space operation X'00' = Check reset
3	PODDR	1	Printer R-byte—specifies (in binary) the amount of carriage movement for any carriage operation X'80' = Halfline space print option (line printer only)
4	PODDNEXT	2	Address of next available system printer IOB

Figure 2-16. Format of Printer IOB

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	KBCHN	2	Address of next IOB in chain
2	KBCMP	1	Completion code: X'80' = Reserved X'40' = Operation complete if set to 1 Bits 2-7 = Scan code for the key; presented to attachment by the keyboard before microcode conversion of the scan code to EBCDIC
3	KBSNS1	1	First sense byte—area where status byte of current keystroke is placed, when in SDE, CSDE, or BDE mode with a function key: X'80' = Keyboard overrun X'40' = Data key (A-Z, 0-9, special character) X'20' = Function key X'08' = Shift key (shift occurred on current keystroke; denotes upper case) X'01' = Katakana mode (See Figure 2-18 for an explanation of possible entries.)
4	KBSNS2	1	Sense byte 2—area where data byte of current keystroke is placed, when in SDE, CSDE, or BDE with function key mode. (See Figure 2-18 for an explanation of possible contents.)
5	KBNCP	1	Cursor position within the display screen buffer—set to 0 for no cursor or 1 to 240 for possible display screen positions
6	KBRSP	1	Record start position—displacement into the display screen buffer of leftmost byte of record to be moved to or from display screen buffer
7	KBLEN	1	Record length—maximum length of current record, 1–240 characters

Figure 2-17 (Part 1 of 2). Format of Keyboard/CRT IOB

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
8	KBFLAG	1	Flag byte: X'C0' = CSDE mode—console display processes keystroke, but also passes characters to SCP and provides sense bytes X'80' = SDE mode—console display does no processing, but passes characters to SCP and provides sense bytes X'40'= BDE mode—console display processes data a record at a time X'20' = Numeric mode X'10' = Read data from display screen buffer to main storage X'08' = Roll display screen buffer up one line X'04' = Display current contents of display screen buffer X'02' = Reset display screen (erase display) X'01' = Write data from main storage to display screen buffer
9	KBMS@	2	Address in main storage from which data can be moved to or from the display screen buffer
B	KBOPCD	1	Additional flag byte: X'40' = *Do not wait* after issuing request X'20' = Loop control X'10' = Last key hit was the CMD key X'08' = Last request was *do not wait* X'04' = Disable 2 request
C	KBFKM1	1	First byte of function key mask
D	KBFKM2	1	Second byte of function key mask
E	KBFKM3	1	Third byte of function key mask *Note:* If a bit in the function key mask is on, a corresponding function key is returned to user when pressed; default is not provided.

Figure 2-17 (Part 2 of 2). Format of Keyboard/CRT IOB

Key	Status Byte—Sense Byte 0 (in hex)	Data Byte—Sense Byte 1 (in hex)
INQ (l/c)	20	01
ERROR RESET (u/c)	28	06
ERROR RESET (l/c)	20	06
↑	20	0A
↓	20	0B
← (l/c)	20	0C
→ (l/c)	20	0D
ENTER (u/c)	28	10
ENTER (l/c)	20	10
ENTER+ (u/c)	28	11
ENTER+ (l/c)	20	11
ENTER– (u/c)	28	12
ENTER– (l/c)	20	12
FIELD ADV (u/c)	28	13
FIELD ADV (l/c)	20	13
REC ADV (u/c)	28	14
REC ADV (l/c)	20	14
FIELD BKSP (u/c)	28	15
FIELD BKSP (l/c)	20	15
REC BKSP (u/c)	28	16
REC BKSP (l/c)	20	16
DUP (u/c)	28	17
DUP (l/c)	20	18
ROLL↑	20	19
ROLL↓	20	1A
← (u/c)	28	1B
→ (u/c)	28	1C
CODE (u/c)	48	1E
CODE (l/c)	40	1E
CMD		30
DATA keys		(EBCDIC characters)

Notes:

1. l/c means lower case and denotes no SHIFT key operation.
2. u/c means upper case and denotes a SHIFT key operation.

Figure 2-18. Possible Sense Byte Entries for Keyboard/CRT IOB

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	IOBNXT	2	Address of next IOB
2	IOBCMP	1	IOB completion code: X'88' = IOB being transmitted X'84' = IOB ready for transmit X'80' = IOB in process X'56' = Forward abort sequence check X'55' = Adapter check X'54' = Invalid response X'53' = Lost connection X'52' = Lost data X'51' = Data check X'50' = No response from remote terminal X'4F' = Permanent error X'4E' = Delay count exceeded X'4D' = Invalid request X'4C' = No connection X'4B' = Invalid ASCII character X'43' = Invalid ID X'42' = End of file X'40' = Normal completion
3	IOBDBL	2	Data buffer length
5	IOBFLA	1	Flag byte: X'80' = ENQ has been sent X'40' = Invalid ACK received X'20' = 2-second timeout started X'08' = IOB set up for transfer X'04' = First-time logic in current program X'02' = Delay IOB bit on X'01' = Data has been sent from this IOB
6	IOBQ	1	SIO Q-byte of last operation: X'83' = Receive initial X'82' = Transmit and receive X'81' = Receive only

Note: This BSC IOB is not used by MRJE.

Figure 2-19 (Part 1 of 2). Format of BSC IOB

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
7	IOBFLG	1	Flag byte: X'80' = Input file X'40' = Output file X'20' = ITB mode X'10' = Transparent mode X'08' = GET file X'04' = On—ASCII Off—EBCDIC
8	IOBDAT	2	Address of BSC data buffer
A	IOBSNS	2	Sense area: *Byte 0:* X'80' = Timeout error X'40' = Data check X'10' = Micro interrupt overrun X'08' = Data contains invalid ASCII character X'04' = Abortive disconnect X'02' = BSCA busy *Byte 1:* X'02' = Data set ready
C	IOBERR	1	Retry count
D	IOBCAR	2	IOB current address register save area
F	IOBTAR	2	IOB transition address register save area
11	IOBSAR	2	IOB stop address register save area
13	IOBDTF	2	Address of associated DTF
15	IOBENC	1	ETB character hold area

Note: This BSC IOB is not used by MRJE.

Figure 2-19 (Part 2 of 2). Format of BSC IOB

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	SIOBCHAN	2	Chain address of the next IOB. Not used, set to X'FFFF' by IOS
2	SIOBCCDE	1	Completion code: X'40' = Set on by microcode when processing is complete. Must be set off by IOS. If on when SIO is issued, sense is returned with no command issued X'01' = Error detected. Set on by microcode when error is detected. Reset by IOS
3	SIOBQ	1	Q-byte: X'51' = Read command X'52' = Punch command
4	SIOBR	1	R-byte: Reserved; initialized to X'00'.
5	–	1	Reserved
6	SIOBCURA	2	Current address. This field must point to the leftmost byte of the data buffer. Current address is updated at completion time to point one byte beyond last data byte transferred
8	SIOBSNS0	2	Sense bytes 0 and 1: *Byte 0:* X'80' = Offline X'40' = Transport jam X'20' = Hopper jam, stacker full, or hopper empty X'10' = Hardware timeout X'08' = Incorrect card code X'04' = Compare error on read or punch X'02' = 0 = 5496 attached 1 = 129 attached X'01' = Reserved for KATAKANA *Byte 1:* Reserved

Figure 2-20. Format of Data Recorder IOB

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
00	IOBMCH@	2	Two-byte chain address of next IOB on queue
02	IOBMCC	1	Completion code posted by System/32 microcode: X'80' = IOB active X'41' = Error in processing IOB request X'40' = IOB processing complete X'10' = Count flag
03	IOBMQC	1	Q-byte X'56' = Load compressed specifications or work area X'54' = Load device run/stacker select/ modify check code X'53' = Single document request X'52' = Load diagnostic code X'51' = Get document request
04	IOBMRC	1	Q-byte X'56' = X'00' = Compression group X'02' = Work area X'54' = Number of sectors of microcode to load X'53' = Same as X'51' X'52' = Number of sectors of microcode to load X'51' = Number of documents to be read
05	IOBMSB	1	Sense byte: Bit 0 = Bring up diagnostic error 1 = Document count limit reached 2 = Reserved 3 = Magnetic character reader controller DBO/DBI parity check 4 = External I/O light 5 = Magnetic character reader controller memory parity check 6 = Magnetic character reader controller long timeout 7 = Sorter is stopped

● **Figure 2-21 (Part 1 of 2). Format of Magnetic Character Reader IOB**

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
06	IOBM@1	2	Q-byte: X'56' = Start of compression or work area in main storage X'54' = Same as X'52' X'53' = Same as X'51' X'52' = Main storage address to use as I/O buffer in loading microcode into magnetic character reader controller area >= 512 bytes X'51' = Main storage address of area to read documents into
08	IOBM@2	2	Q-byte: X'56' = End of compression or work area in main storage X'54' = Same as X'52' X'53' = End address of buffer X'52' = Disk address of microcode to be loaded into magnetic character reader control storage X'51' = End address of buffer
0A	IOBMSB2	1	Sense byte 2: Bit 0 = Document auto reject 1 = Reserved 2 = MIS read with reject 3 = MIS read without reject 4 = Reserved 5 = Reserved 6 = Reserved 7 = Reserved
0B	IOBMRSVD	1	Reserved

● **Figure 2-21 (Part 2 of 2). Format of Magnetic Character Reader IOB**

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	MCBCPT	2	Address of next IOB in chain
2	MCBCMP	1	Completion code: X'80' = IOB active X'41' = Error completion X'40' = Operation complete
3	MCBCMMD	1	Command code: X'53' = Sense X'52' = Write current track X'51' = Read current track X'50' = Control
4	MCBMOD	1	Command modifier code: X'0C' = Set/reset indicators and alarm X'0B' = Exit X'0A' = Translate table redefined X'08' = Home X'07' = Feed X'06' = Stack X'05' = Track step-up X'04' = Eject
5	MCBBUF	2	Address of data buffer
7	MCBSENS0	1	Sense byte 0: X'80' = Device parity error X'40' = Head at track 1 X'20' = Invalid command X'10' = Card is present in a valid position X'08' = Interrupts enabled X'04' = Card handling error X'02' = Read or write error X'01' = MCU is started
8	MCBSENS1	1	Sense byte 1: X'80' = Ready X'40' = Card jam X'20' = A card is present in the throat X'10' = Head is at track 50 X'08' = Overrun X'04' = Read or write error X'02' = Data not found X'01' = Timeout interrupt occurred

Figure 2-22 (Part 1 of 2). Format of Mag Card Unit IOB

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
9	MCBCNT	1	Character count
A	MCBFLAG	1	Flag byte: X'80' = Invalid character in buffer X'20' = No TTC found in I/O X'10' = No dummy character found on read X'08' = Carriage return character found X'04' = Track link character found X'02' = Page end character found X'01' = Card repeat character found
B	MCBIAB	1	Indicator alarm byte: X'08' = Message lamp X'04' = Reserved X'02' = Reset start latch X'01' = Audio alarm
C	MCBRTC	1	Retry count
D	MCBRSV	2	Reserved
F	MCBARR	2	MCU ARR storage area
11	MCB	1	Reserved for MCUAM

Figure 2-22 (Part 2 of 2). Format of Mag Card Unit IOB

SDLC IOB

The SDLC IOBs are located in main storage. They are used as an interface between SNA and SDLC. The number of IOBs can vary, depending on the size of main storage.

How to Find

The SNA DTF bytes X'10' and X'11' will point to the SNA/SDLC common area. The SNA/SDLC common area bytes X'08' and X'09' will point to the SDLC receive IOB chain and bytes X'0C' and X'0D' will point to the SDLC transmit IOB chain.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	SIOBCHN	2	Address of next IOB in chain
2	SIOBCMP	1	Completion code: X'40' = Operation completed normal by SDLC X'41' = Permanent SDLC error X'42' = Disconnect received X'43' = Data set dropped ready X'44' = Ignore IOB, go on to next X'80' = Buffer in use by SNA X'84' = Buffer ready for SDLC I/O X'88' = Buffer in process by SDLC
3	SIOBLEN	2	Length of data received (unused)
5	SIOBFG1	1	Reserved
6	SIOBQ	1	Q-byte: X'80' = Control X'81' = Reçeive only X'82' = Transmit/receive X'83' = Receive initial X'84' = Transmit final X'85' = Transmit only
7	SIOBREA	2	Received data end address
9	SIOBSNS	2	Sense bytes: *Byte 0:* X'80' = Timeout X'40' = Block check X'20' = Transmit adapter check X'10' = Receive adapter check X'08' = Invalid frame X'04' = Abortive disconnect X'01' = Stack overflow *Byte 1:* X'02' = Data set ready
B	SIOBNSC	1	Transmit wrap count
C	SIOBSBA	2	Buffer start address
E	SIOBEBA	2	Buffer end address

Figure 2-23. Format of SDLC IOB

DTF

A DTF control block is an area of main storage used as the primary interface between the system and data management routines.

How to Find

Postopen DTFs are chained. Field NPDTF@ (X'0B' and X'0C') in the program level communication area contains the address of the first DTF on the chain. The next DTF on the chain can be found by referring to the chain field in the appropriate DTF. End of chain is indicated by X'FFFF'. XR2 points to specified DTF when a data management function is evoked.

Format

The DTF format varies by unit type:

Unit Type	Figure
Disk, preopen	2-24
Disk, postopen	2-25
PTAM, preopen	2-26
PTAM, postopen	2-27
Diskette, preopen	2-28
Diskette, postopen	2-29
Printer, preopen	2-30
Printer, postopen	2-31
IDE data management	2-32
Keyboard/CRT	2-33
BSC	2-34
Mag card unit	2-35
SNA, preopen	2-36
SNA, postopen	2-37
Data recorder	2-39
Magnetic character reader	2-40

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	DTFDEV	1	Device code (Q-byte of SIO; set to X'A0')
1	DTFUPS or	1	External indicator (UPSI)
1	DTFSPR	1	Special allocate return code: X'08' = File statement given X'04' = Special allocate unsuccessful X'02' = Maximum space allocated X'01' = Minimum space allocated
2	DTFATR	2	Attribute: *Byte 0:* X'80' = Indexed X'40' = Consecutive X'20' = Direct X'10' = Offline multivolume X'08' = Input X'04' = Output X'02' = Update X'01' = Add *Byte 1:* X'80' = Binary (ADDROUT) X'40' = Ordered load X'20' = Random X'10' = Limits X'08' = Dual I/O or move mode on input (SIAM only)
4	DTFCHA	2	Record length
6	DTFCHB	2	DTF chain pointer B—forward (address of next DTF in chain: X'FFFF', if end of chain)
8	DTFARR	2	ARR save area (return address)
A	DTFXRS	2	XR1 save area (contents of object program XR1)
C	DTFWKB	2	Logical record address (move mode)
E	–	2	Reserved

Figure 2-24 (Part 1 of 3). Format of Disk DTF (Preopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
10	DTFIOB	2	Address of I/O area Address of IOB area (SIAM only)
12	DTFPRB	2	Reserved Address of left byte of the user record buffer (SIAM only)
14	DTFBKL	2	Block length (used to determine size of data I/O buffers)
16	—	9	Reserved (not special allocate)
	Special allocate only (9 bytes: 16-1E)		
16	DTFMIN	2	Minimum blocks requested
18	DTFPRT or	DTFSPL BIT 4 2 OFF	Address of left byte of the I/O buffer (SIAM only)
18	DTFMAX	2	Maximum blocks requested
16	—	DTFSPL[1]	Unused
17	DTFSBL	BIT 4 ON 3	Blocks or records to be allocated (space request by blocks or records)
1A	DTFLOC	2	Block number to begin file
1C	DTFRET	1	Special allocate retain code
1D	DTFSPL	1	Special allocate request indicator: X'80' = Special allocate DTF X'40' = Location given in special DTF X'10' = Date given in special DTF X'08' = Space request by blocks or records X'01' = System utilities internal indicator
1E	—	1	Reserved
1F	DTFNAM	8	Filename

Figure 2-24 (Part 2 of 3). Format of Disk DTF (Preopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
27	DTFDAT	6	File date (special allocate only)
2D	–	4	Reserved
31	DTFKAD	2	Address of requested key (indexed random) or address of record numbers (direct)
	or DTFCUR	2	Address of current or last key hold area (indexed sequential add)
	or DTFHI	2	Address of high or low key area (processing within limits)
End of DTF for Direct Input (binary and decimal) and Direct Update (binary and decimal)			
33	–	4	Reserved
37	DTFKL	2	Key length (indexed)
39	–	2	Reserved
3B	DTFKD	2	Displacement of key in record (indexed)
End of DTF for Indexed Output and Indexed Sequential (input and input/update)			
3D	DTFMIX	2	Address of master track index (indexed random)
End of DTF for Indexed Sequential Input (limits) and Indexed Sequential Input/Update (limits)			
3F	DTFBYT	2	Number of bytes in master track index (indexed random)
41	–	1	Reserved
End of DTF for Indexed Add, Indexed Random (input/add), Indexed Sequential Random Input, Indexed Random Input, and Indexed Random (update/input)			
42	–	6	Reserved

Figure 2-24 (Part 3 of 3). Format of Disk DTF (Preopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	DTFDEV	1	Device code (Q-byte of SIO; set to X'A0')
1	DTFERP	1	Work byte
2	DTFATR	2	Attribute: *Byte 1:* X'80' = Indexed X'40' = Consecutive X'20' = Direct X'10' = Offline multivolume X'08' = Input X'04' = Output X'02' = Update X'01' = Add *Byte 2:* X'80' = Binary (ADDROUT) X'40' = Ordered load X'20' = Random X'10' = Limits X'08' = Dual I/O or move mode on input (SIAM only) X'04' = Reserved X'02' = End of limits X'01' = Opened
4	DTFCHA	2	DTF chain pointer A—backward (address of next DTF in chain: X'FFFF', if end of chain)
6	DTFCHB	2	DTF chain pointer B—forward (address of next DTF in chain: X'FFFF', if end of chain)
8	DTFARR	2	ARR save area (return address)
A	DTFXRS	2	XR1 save area (contents of object program XR1)
C	DTFWKB	2	Logical record address
E	DTFCMP	1	Completion code: X'40' = Normal completion X'41' = Controlled cancel taken on permanent I/O error X'42' = End of file (input) X'44' = Record not found (out of extent for direct files, normal return on open) X'50' = Key field does not match key in update record

Figure 2-25 (Part 1 of 4). Format of Disk DTF (Postopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
			Completion code (continued): X'60' = Duplicate load or add attempted X'62' = Out of sequence (load or add attempted) X'70' = End of extent (output) Completion codes other than X'40' are returned before the data management function is actually completed.
F	DTFOPC	1	Operation code: X'84' = Indexed sequential forward get X'82' = Indexed sequential backward get X'80' = Get X'40' = Put X'20' = Update
10	DTFIOB	2	Address of current I/O IOB Address of IOB (SIAM only)
12	DTFPRB	2	Address of current process IOB Address of left byte of the user's record buffer (SIAM only)
14	DTFBKL	2	Block length
16	DTFRCL	2	Logical record length
18	DTFPTR	2	Displacement of current record in data buffer (rightmost byte) or for SIAM, points to the left byte of the I/O buffer
1A	DTFXTA	2	Disk address of start of data (SS)
1C	DTFXTB	2	Disk address of end of data (SS)
1E	DTFSWA	1	SWA format 1 label sequence number
1F	DTFWAA	1	Work area A
20	DTFWAB	1	Work area B
21	DTFWAC	1	Work area C
22	DTFWAD	1	Work area D
23	DTFRMA	2	Work area (buffer spanning record, first part)

Figure 2-25 (Part 2 of 4). Format of Disk DTF (Postopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
25	DTFRMB	2	Work area (buffer spanning record, second part)
27	DTFIND	1	Indicator bits: X'80' = End of file X'40' = End of extent X'20' = Add operation X'10' = Pseudo get X'08' = Record added to file X'04' = Index buffer needs writing X'02' = Data buffer needs writing (SIAM, last operation was a *put*) X'01' = First time called
28	DTFNXR	4	Disk address (S/S/D/D) of current record
			End of DTF for Consecutive (output/add)
2C	DTFEOF or	3	Disk address (S/S/D) of logical end of file
2C	DTFNXK	3	Disk address (S/S/D) of logical end of index
			End of DTF for Consecutive (input) and Consecutive (update)
2F	DTFKPR	2	Index buffer pointer
31	DTFKAD	2	Address of requested key (indexed random) or address of rightmost byte of the relative record number (direct). (This field may not be changed after the first SIAM call)
	or		
31	DTFCUR	2	Address of rightmost byte of the current key hold area (indexed sequential add)
	or		
31	DTFHI	2	Address of rightmost byte of the high key hold area (processing within limits)
			End of DTF for Direct Input (binary and decimal) and Direct Update (binary and decimal)
33	DTFKXA	2	Disk address of start of index (SS)
35	DTFKBF	2	Address of index IOB
37	DTFKL	2	Key length (indexed)
39	DTFKXB	2	Disk address of end of index (SS)
3B	DTFKD	2	Displacement of rightmost byte of key in record (indexed)
			End of DTF for Indexed Output and Indexed Sequential (input and input/update)

Figure 2-25 (Part 3 of 4). Format of Disk DTF (Postopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
3D	DTFMIX	2	Address of master track index (indexed random)
	or		
	DTFLST	2	Address of rightmost byte of previous key hold area (indexed sequential add)
	or		
	DTFLOW	2	Address of rightmost byte of low key hold area (processing within limits)
			End of DTF for Indexed Sequential (input—limits) and Indexed Sequential (input/update—limits)
3F	DTFKXP	3	Disk address (S/S/D) of logical end of original index (indexed add)
			End of DTF for Indexed (add), Indexed Random (input/add), Indexed Sequential Random (input), Indexed Random (input), and Indexed Random (update/input)
42	DTFHKB	2	Save area for address of high key on file
	or		
	DTFSNP	2	Save area for address of next index buffer entry
			End of DTF for Indexed Sequential (input/add) and Indexed Random (input/update/add)
44	DTFSLA	2	Save area for current index address (SS)
46	DTFSLP	2	Save area for current buffer entry address
			End of DTF for Indexed Sequential (input/update/add)

Figure 2-25 (Part 4 of 4). Format of Disk DTF (Postopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	DTFDEV	1	Device code (set to X'A0')
1	DTFUPS	1	External indicator (UPSI)
2	DTFPA0	1	Attribute 0 X'60' = PTAM X'08' = Input X'04' = Output X'02' = Update
3	DTFPA1	1	Attribute 1
4	DTFCHA	2	Reserved
6	DTFCHB	2	Forward chain pointer

● **Figure 2-26 (Part 1 of 2). Format of PTAM DTF (Preopen)**

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
8	DTFARR	2	ARR save area
A	DTFXRS	2	XR1 save area
C	DTFWKB	2	Logical record address
E	—	1	Reserved
F	—	1	Reserved
10	DTFIOB	2	Address of I/O area
12	—	1	Reserved
13	—	1	Reserved
14	DTFBKL	2	Block length
16	DTFPHT	7	Reserved
1D	DTFPA2	1	Reserved (Must be set preopen but postallocate)
1E	DTFSWA	1	Reserved
1F	DTFPNM	8	PTAM filename
27	DTFIND	1	Reserved
28	—	1	Reserved
29	DTFPST	3	Reserved
2C	—	1	Reserved
2D	DTFPND	3	Reserved
30	—	1	Reserved
31	DTFPNB	3	Reserved
34	—	1	Reserved
35	DTFPEF	3	Reserved
38	—	1	Reserved
39	DTFPSC	3	Reserved
3C	DTFPBL	2	Reserved
3E	DTFPDB	2	Reserved

● **Figure 2-26 (Part 2 of 2). Format of PTAM DTF (Preopen)**

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	DTFDEV	1	Device code (set to X'A0')
1	DTFUPS	1	External indicator (UPSI)
2	DTFPA0	1	Attribute 0 X'60' = PTAM X'08' = Input X'04' = Output X'02' = Update
3	DTFPA1	1	Attribute 1 X'01' = DTF opened
4	DTFCHA	2	Backward chain pointer
6	DTFCHB	2	Forward chain pointer
8	DTFARR	2	ARR save area
A	DTFXRS	2	XR1 save area
C	DTFWKB	2	Logical record address
E	DTFCMP	1	Completion code X'70' = End of extent (output) X'42' = End of file (input) X'41' = Permanent I/O error X'40' = Normal completion
F	DTFOPC	1	Operation code X'1E' = Forward space block X'1C' = Backward space block X'0E' = Forward space file X'0C' = Backward space file X'0A' = Write tape mark X'04' = Read
10	DTFIOB	2	Address of IOB
12	DTFPOS	1	Previous op-code save areas
13	—	1	Reserved
14	DTFBKL	2	Block length
16	DTFPHT	7	Halt parameter list
1D	DTFPA2	1	Attribute 2
1E	DTFSWA	1	SWA F1 number
1F	DTFPNM	8	PTAM filename

● **Figure 2-27 (Part 1 of 2). Format of PTAM DTF (Postopen)**

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
27	DTFIND	1	Indicator byte
28	–	1	Reserved
29	DTFPST	3	PTAM start of data
2C	–	1	Reserved
2D	DTFPND	3	PTAM end of data
30	–	1	Reserved
31	DTFPNB	3	PTAM next block
34	–	1	Reserved
35	DTFPEF	3	PTAM end of file
38	–	1	Reserved
39	DTFPSC	3	PTAM sector in core
3C	DTFPBL	2	PTAM block length = 1
3E	DTFPDB	2	PTAM displacement into block

● **Figure 2-27 (Part 2 of 2). Format of PTAM DTF (Postopen)**

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	DTFDEV	1	Device code (Q-byte of SIO; set to X'D0')
1	–	1	Reserved
2	DTFATR	2	File attributes: *Byte 1:* X'80' = System file X'40' = Basic exchange X'20' = Include disk format 1 X'08' = Input X'04' = Output X'01' = Add

Figure 2-28 (Part 1 of 3). Format of Diskette DTF (Preopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
			Byte 2: X'80' = Clean pack X'40' = Skip MVF sequence X'20' = End-of-volume return X'10' = Full track mode X'08' = Move mode (dual mode if X'04' is on) X'04' = Sector data management
4	DTFCHA	2	Record length
6	DTFCHB	2	DTF chain pointer B—forward (address of next DTF in chain: X'FFFF', if end of chain)
8	DTFARR	2	ARR save area (return address)
A	DTFXRS	2	XR1 save area (contents of object program XR1) (first byte not used for preopen DTF)
B	DTFRDF	1	Physical format request byte: X'04' = Ensure 512-byte format X'01' = Ensure 128-byte format X'00' = Do not check physical format
C	DTFWKB	2	Logical record address (move mode)
E	—	1	Reserved
F	DTFOPC	1	Operation code: X'80' = Get X'40' = Put
10	DTFIOB	2	Area for IOBs and I/O buffers
12	—	2	Reserved
14	DTFBKL	2	Block length (used to determine size of data I/O buffers)
16	—	9	Reserved
1F	DFTNAM	8	Filename
27	—	10	Reserved

Figure 2-28 (Part 2 of 3). Format of Diskette DTF (Preopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
31	DTFIOS	2	Address of diskette IOS
33	–	7	Reserved
3A	DTFPRE	1	Switch for diskette prepare: X'FF' = Prepare requested
3B	–	64	Reserved

Figure 2-28 (Part 3 of 3). Format of Diskette DTF (Preopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	DTFDEV	1	Device code (Q-byte of SIO; set to X'D0')
1	–	1	Reserved
2	DTFATR	2	File attributes: *Byte 1:* X'80' = System file X'40' = Basic exchange X'20' = Include disk format 1 X'10' = Multivolume X'08' = Input X'04' = Output X'01' = Add *Byte 2:* X'80' = Clean pack X'40' = Skip MVF sequence X'20' = End-of-volume return X'10' = Track X'08' = Move mode (dual I/O if X'04' is on) X'04' = Sector data management X'02' = End-of-volume close X'01' = Opened
4	DTFCHA	2	DTF chain pointer A—backward (address of next DTF in chain: X'FFFF', if end of chain)
6	DTFCHB	2	DTF chain pointer B—forward (address of next DTF in chain: X'FFFF', if end of chain)

Figure 2-29 (Part 1 of 3). Format of Diskette DTF (Postopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
8	DTFARR	2	ARR save area (return address)
A	DTFXRS	2	XR1 save area (contents of object program XR1)
C	DTFWKB	2	Logical record address
E	DTFCMP	1	Completion code: X'40' = Normal completion X'41' = Permanent I/O error X'42' = End of file (input) Completion codes other than X'40' are returned before the data management function is actually completed
F	DTFOPC	1	Operation code: X'80' = Get X'40' = Put
10	DTFIOB	2	IOB address
12	DTFBRL	2	Basic exchange record length
14	DTFBKL	2	Block length
16	DTFRCL	2	Logical record length
18	DTFPTR	2	Address of rightmost byte of current record in data buffer
1A	DTFXTA	2	Data extent start (sector address)
1C	DTFXTB	2	Data extent end (sector address)
1E	DTFSWA	1	SWA format 1 label sequence number
1F	DTFWAA	1	Work area A
20	DTFWAB	1	Work area B
21	DTFWAC	1	Work area C
22	DTFWAD	1	Work area D
23	DTFRMA	2	Work area (buffer spanning record, first part)
25	DTFRMB	2	Work area (buffer spanning record, second part)

Figure 2-29 (Part 2 of 3). Format of Diskette DTF (Postopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
27	DTFIND	1	Indicator bits: X'80' = Diskette multivolume indicator X'40' = Offline disk multivolume file X'20' = Embedded disk format 1 present X'10' = Reserved X'08' = Aligned on track boundary X'04' = Buffer end reached X'02' = First diskette data management call X'01' = Diskette output end of volume
28	DTFNXR	4	Disk address (S/S/D/D) of current record
2C	DTFEOF	4	Disk address (S/S/D/D) of logical end of file
30	DTFWKA	1	Work area
31	DTFIOS	2	Address of diskette IOS
33	DTFPBF	2	Physical I/O buffer address
35	DTFBFE	2	Pointer to end of I/O buffer
37	DTFAR1	2	End-of-volume save area (ARR)
39	DTFXR1	2	End-of-volume save area (XR1)
3B	DTFIF1	64	Start of saved 64-byte format 1 (Figure 2-106)
7A	DTFIFL	1	Last byte of saved format 1

Figure 2-29 (Part 3 of 3). Format of Diskette DTF (Postopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	PRTDEV	1	Device code (Q-byte of SIO; set to X'E0')
1	PRTUPS	1	External indicator (UPSI)
2	PRTAT1	1	Attribute byte 1: X'02' = Halt on unprintable character
3	PRTAT2	1	Attribute byte 2: X'40' = Device allocated bit X'01' = File opened bit
4	PRTRLN	2	Logical record length
6	PRTCHB	2	DTF chain pointer B—forward (address of next DTF in chain: X'FFFF', if end of chain)
8	PRTARR	2	ARR save area (return address)
A	PRTXR1	2	XR1 save area (contents of calling program register 1)
C	PRTLRA	2	Logical record address
E	—	16	Unused
1E	PRTPR1	2	Address of physical buffer
20	—	1	Reserved
21	PRTPOV	1	Overflow line number
22	PRTPNL	1	Form length
23	—	4	Unused

Figure 2-30. Format of Printer DTF (Preopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	PRTDEV	1	Device code (Q-byte of SIO set to X'E0')
1	PRTUPS	1	External indicator (UPSI)
2	PRTAT1	1	Attribute byte 1: X'04' = Noncontinuous mode indicator X'02' = Halt on unprintable characters
3	PRTAT2	1	Attribute byte 2: X'40' = Device allocated X'01' = File opened
4	PRTCHA	2	DTF chain pointer A—backward (address of next DTF in chain: X'FFFF', if end of chain)
6	PRTCHB	2	DTF chain pointer B—forward (address of next DTF in chain: X'FFFF', if end of chain)
8	PRTARR	2	ARR save area (return address)
A	PRTXR1	2	XR1 save area (contents of calling program register 1)
C	PRTLRA	2	Logical record address
E	PRTCMP	1	Completion code: X'40' = Normal completion X'41' = Abnormal completion; operator selected option 2 in response to error condition X'48' = Overflow
F	PRTOPR	1	Operation code: X'40' = Print
10	PRTSKB	1	Skip before value (line number; maximum = 84)
11	PRTSPB	1	Space before value (number of lines; maximum = 3)[1]
12	PRTSKA	1	Skip after value (line number; maximum = 84)
13	PRTSPA	1	Space after value (number of lines; maximum = 3)[1]

[1] The high order bit controls the half line spacing option (it is ignored if the option is not installed); for example, X'02' provides two line spaces; X'82' provides 2-1/2 line spaces.

Figure 2-31 (Part 1 of 2). Format of Printer DTF (Postopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
14	PRTQ	1	Printer Q-byte for IOB
15	PRTR	1	Printer R-byte for IOB
16	PRTUSE	1	Usable area of form (noncontinuous document mode)
17	PRTSVA	2	Save address for physical buffer (leftmost byte)
19	PRTXLC	1	System value for lines per page (before open)
1A	PRTWKA	2	Work area for printer sense bytes
1C	PRTBUF	2	Address of the physical buffer (rightmost byte)
1E	PRTIOB	2	Unused
20	PRTPRA	2	Must be zero
22	PRTLRL	1	Logical record length minus 1
23	PRTOFL	1	Overflow line
24	PRTPCT	1	Printer position counter
25	PRTDMA	2	Data management disk address

Figure 2-31 (Part 2 of 2). Format of Printer DTF (Postopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	CNDEV@	1	Device address (X'10')
1	CNUPSI	1	External indicator (UPSI)
2	CNATR1	1	Attributes: *Byte 1:* X'80' = End of record X'20' = IDE called by open/DM X'10' = IDE called by DM X'08' = Error has occurred X'04' = Sequence checking on X'02' = Normal completions X'01' = Last operation was error
3	CNATR2	1	Attributes: *Byte 2:* X'40' = File allocated X'20' = Record address file (preopen only) X'01' = File opened (postopen only)
4	CNDTFA	2	DTF chain pointer A—backward (address of next DTF in chain: X'FFFF', if end of chain)
6	CNDTFB	2	DTF chain pointer B—forward (address of next DTF in chain: X'FFFF', if end of chain)
8	CNXRIS	2	XR1 save area
A	CNARRS	2	ARR save area
C	CNREC@	2	User buffer address—where record is to be passed back to user
E	CNCMCD	1	Completion code: X'40' = Normal completion X'42' = End of file

Figure 2-32 (Part 1 of 2). Format of IDE Data Management DTF

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
F	CNOPCD	1	Operation code: X'10' = Flush IDE buffer
10	CNBUF@	2	IDE buffer address (for IDE interrupt handler)
12	CNBUFL	2	Length of IDE buffer (bytes)
14	CNRP@	2	Record process address (where to put next record from operator, in IDE buffer)
16	CNCFLD	2	Address of field in format description currently being processed
18	CNCFMT	2	Address of current format description being processed
1A	CNCTYP	2	Address of current entry in sequence table being processed
1C	CNIH@	2	Address of interrupt handler (#$BIIH)
1E	CNSS	2	Sector address of console data management (postopen only)
20	CNRA@	2	Address of next record available to IDE data management (in IDE buffer)
22	CNSEQ	2	Address of sequence table
24	CNWKA	4	Work area

Note: The fields are identical for both preopen and postopen DTFs unless otherwise specified in the description column.

Figure 2-32 (Part 2 of 2). Format of IDE Data Management DTF

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	KCDEV@	1	Device address X'12' = Keyboard X'90' = Display screen
1	KCUPSI	1	External indicator (UPSI)
2	KCATTR	2	File attribute bytes: *Byte 0:* X'04' = Enable/disable request *Byte 1:* X'01' = File open
4	KCDTFA	2	DTF chain pointer A—backward (address of next DTF in chain: X'FFFF', if end of chain)
6	KCDTFB	2	DTF chain pointer B—forward (address of next DTF in chain: X'FFFF', if end of chain)
8	KCARRS	2	ARR save area
A	KCXR1S	2	XR1 save area
C	KCREC@	2	Address in user area at which input is to be put upon return from keyboard data management
E	KCCMCD	1	Completion code: X'40' = Normal completion X'42' = File not opened X'43' = Illegal cursor position
F	KCOPCD	1	Operation code: X'C0' off = Output only X'80' off and X'40' on = Basic data entry (BDE) only X'80' on and X'40' off = SDE only X'C0' = CSDE X'20' off = Alphameric mode X'20' = Numeric only X'10' = Move record from KCMSG@ to display screen buffer X'08' = Pass control back to user after any CMD key X'04' = Controlled output of one line at a time (operator must hit a key after every line) X'02' = Move record from display screen buffer to KCREC@ X'01' = Overlapped key entry wait will not be issued until second call to data management

Figure 2-33 (Part 1 of 2). Format of Keyboard/CRT DTF

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
10	KCSPAC	1	Space before count—the number of lines buffer should be rolled before accepting keyboard input (0-256)
11	KCNCP	1	Cursor position within display screen buffer—specifies how much to change position (0-240)
12	KCRSP	1	Record start position—location in display screen buffer of first logical character in user record; cursor cannot be located before the record start position
13	KCLEN	1	Record length—maximum number of characters of keyboard input allowed; if exceeded, keyboard is disabled and ERROR RESET must be pressed
14	KCIOB@	2	IOB address—15-byte area used by keyboard data management
16	KCSTB	2	Status byte—in SDE mode, last character entered is placed here; in BDE mode, this is user function key or CMD key code
18	KCFKM	3	Function key mask—mask of keys which user can redefine for use during this input cycle
1B	KCCKM	3	Command key mask—command keys acceptable as input; numbered consecutive from 1 (bit 0, byte 0) to 24 (bit 7, byte 2)
1E	KCCKSM	3	Command key set mask—command keys hit during input cycle
21	KCFLI	1	Message indicator to be set on during current input cycle
22	KCSSN	2	Sector address of keyboard/CRT data management (#KID10)
24	KCMSGP	1	Position at which message (prompt) is to be placed within display screen buffer
25	KCMSGL	1	Length of message
26	KCMSG@	2	Address of message (prompt) to be displayed before allowing keyboard input

Note: The fields are essentially the same for both preopen and postopen DTFs.

Figure 2-33 (Part 2 of 2). Format of Keyboard/CRT DTF

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	DTFDEV	1	Device ID: X'80'
1	DTFUPS	1	UPSI: U1-U8, user-controlled program switches
2	DTFATT	1	File attributes: X'80' = Input file X'40' = Output file X'20' = ITB mode X'10' = Transparent mode X'08' = GET file X'04' = ASCII code X'02' = Unused X'01' = Debug function requested
3	DTFATR	1	File attributes: X'80' = Multipoint line X'40' = DTF allocated X'20' = Manual line X'10' = Answer line X'08' = Switched line X'04' = File used X'02' = File active X'01' = File opened
4	DTFCHN	2	DTF chain pointer A—backward (X'FFFF' if end of chain)
6	DTFNXT	2	DTF chain pointer B—forward (X'FFFF' if end of chain)
8	DTFWK1	2	Work area 1 (XR1 save area)
A	DTFWK2	2	Work area 2 (XR2 save area)
C	DTFWKB	2	Address of user logical buffer
E	DTFCMP	1	Completion code: X'4F' = Permanent error X'4D' = Invalid request X'4B' = Invalid ASCII character X'42' = End of file X'41' = User error X'40' = Normal completion
F	DTFOP	1	Operation code: X'80' = GET operation X'40' = PUT operation
10	DTFLGR	2	Reserved

Note: The fields are essentially the same for both preopen and postopen DTFs.

Figure 2-34 (Part 1 of 2). Format of BSC DTF

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
12	DTFAT1	1	File attributes: X'20'—two IOBs required
13	DTFPSC	2	Poll/address character for multipoint tributary line
15		1	Reserved
16	DTFRID	2	Address of receive ID character for switched ID parameter list for switched line (must be a valid address)
18	DTFRC	1	Number of receive ID characters (must be zero if no ID)
19	DTFSID	2	Address of send ID character for switched line (must be a valid address)
1B	DTFSC	1	Number of send ID characters (must be zero if no ID)
1C	DTFDLY	2	Wait time allowed BSC between block transmissions
1E	DTFREL	2	Record length
20	DTFBKL	2	Block length
22	DTFIOB	2	Address of IOB in process
24	DTFBKX	2	Pointer to data in BSC buffer
26	DTFITB	2	ITB character count
28	DTFPRM	3	Permanent error indicator mask (first byte) and displacement (next two bytes)—(address must be valid and mask must be zero if not used)
2B	DTFRVI	3	Record available indicator mask (first byte) and displacement (next two bytes)—(address must be valid and mask must be zero if not used)
2E	DTFNDX	1	Index for line initialization
2F	DTFWKA	2	Address of BSC work area
31	DTFCS	2	Disk address of first line initialization load
33	DTFXTA	2	Reserved
35	DTFERC	1	Retry count

Note: The fields are essentially the same for both preopen and postopen DTFs.

Figure 2-34 (Part 2 of 2). Format of BSC DTF

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	MCRDEV	1	Device ID (X'50')
1	MCRCMP	1	Completion code: X'44' = Operator initiated cancel X'43' = Invalid request X'42' = Character error X'41' = Hardware error X'40' = Normal completion
2	MCROPC	1	Request code: X'24' = Select MCU translate table X'20' = Wait X'1C' = Get special X'18' = Home X'14' = Close X'10' = Track step-up X'0C' = Stack X'08' = Write X'04' = Read X'00' = Open
3	MCRMOD	1	Operation modifier code: *For write request:* X'01' = Check for blank, then record X'00' = Record, do not check card for blank *For read request:* X'01' = Flush mode requested. If card is blank, returns X'42' to requestor, indicating no data found on track 1 X'00' = Flush mode not allowed, operator resolves error *For track step-up:* X'nn' = Number of tracks to step; nn can be no greater than the number of tracks left in the card
4	MCRWKB	2	Address of the data buffer or 36-byte area at open time if not running under job to job transition.
6	MCRRCL	1	Number of bytes read or written
7	MCRTKC	1	Track number (0 when no card present)
8	MCRARR	2	ARR save area

Figure 2-35 (Part 1 of 2). Format of Mag Card Unit DTF

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
A	MCRXR1	2	XR1 save area
C	MCRWAA	1	Work area A
D	MCRWAB	1	Work area B
E	MCRIOC	16	SIO counters, 2 bytes/operation
1E	MCRWAC	8	Work area C
26	MCRIOB1	2	Address of IOB1
28	MCRIOB2	2	Address of IOB2
2A	MCRWCC	2	Write card count
2C	MCRTRDEF	1	Binary default tilt/rotate table ID
2D	MCRTRID	1	Binary present tilt/rotate table ID
2E	MCRTRSEL	1	Tilt/rotate status/request byte
2F	MCRWAD	25	Work area D

Figure 2-35 (Part 2 of 2). Format of Mag Card Unit DTF

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	SDTFDEV	1	Device code (X'81')
1		3	Reserved
4	SDTFCHA	2	Backward chain pointer
6	SDTFCHB	2	Forward chain pointer
8		4	Reserved
C	SDTFINB	2	Address of input buffer
E		2	Reserved
10	SDTFCOM	2	Address of SNA/SDLC common area
12	SDTFOTB	2	Address of output buffer
14		6	Reserved
1A	SDTFVTT@	2	Address of vertical tab table
1C	SDTFVTTL	1	Length of vertical tab table
1D	SDTFHTT@	2	Address of horizontal tab table
1F	SDTFHTTL	1	Length of horizontal tab table
20	SDTFCT@	2	Address of compaction table
22		16	Reserved
32	SDTFPIOB	2	Address of first IOB
34	SDTFPBUF	2	Address of first buffer
36	SDTFPNRB	1	Number of receive buffers
37	SDTFPNTB	1	Number of transmit buffers
38	SDTFPNAM	8	Primary LU name
40	SDTFSNAM	8	Secondary LU name
48	SDTFFSZ	2	Frame size for each buffer
4A	SDTFPMRP	1	Secondary maximum receive pacing count
4B		2	Reserved

Figure 2-36. Format of SNA DTF (Preopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	SDTFDEV	1	Device code (X'81')
1	SDTFUPS	1	Reserved
2	SDTFCTL	1	Control byte: These indicators are set on in an inbound request header or they were received in an outbound request header: X'80' = Unformatted X'40' = Change direction X'20' = FM header included X'10' on = DR1/DR2 required X'10' off = Exception response only X'08' = End chain X'04' = Start chain X'02' = End bracket X'01' = Begin bracket
3	SDTFSTAT	1	State indicator: X'80' = Inbound chain X'40' = DTF allocated X'20' = Data request/response unit is transparent data X'10' = Purging chain state X'08' = Outbound chain X'04' on = In brackets X'04' off = Between brackets X'02' on = Receive X'02' off = Send X'01' on = DTF is opened X'01' off = DTF is closed
4	SDTFCHA	2	Backward chain pointer
6	SDTFCHB	2	Forward chain pointer
8	SDTFARR	2	Open ARR save area
A	SDTFXRS	2	Open XR1 save area
C	SDTFINB	2	Address of input buffer

Figure 2-37 (Part 1 of 4). Format of SNA DTF (Postopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
E	SDTFCMP	1	Completion code: X'40' = Normal completion X'41' = Permanent error X'42' = Cancel chain received X'43' = Shutdown received X'44' = Invalid format identifier received X'45' = Receive data management error X'46' = Begin bracket received in transmit state X'47' = Soft break received X'48' = Hard break received X'49' = SNA session completed,. deinitialization complete X'4A' = Invalid request to SNA X'4B' = Basic information unit too short X'4C' = DFC basic information unit too short X'4D' = Segmenting error X'4E' = Negative response received X'4F' = DTF not open, invalid call X'50' = Initialization error or data traffic state reset X'51' = ERP in progress X'52' = Extraneous request/response received X'53' = Data received in transmit state X'54' = Bid received X'55' = Vertical tab table received
F	SDTFOPC	1	Operation code: X'40' = Put data X'41' = Put positive response X'42' = Put negative response X'43' = Put soft break X'44' = Put hard break X'45' = Put LU status X'46' = Put request shutdown X'47' = Put shutdown complete X'80' = Get data X'81' = Get response X'82' = Read data
10	SDTFCOM	2	Address of SNA/SDLC common area
12	SDTFOTB	2	Address of output buffer
14	SDTFIOB	2	Address of current receive IOB
16	SDTFROB	2	Address of current transmit IOB
18	SDTFLEN	2	Input/output data length

Figure 2-37 (Part 2 of 4). Format of SNA DTF (Postopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
1A	SDTFVTT@	2	Address of vertical tab table
1C	SDTFVTTL	1	Length of vertical tab table
1D	SDTFHTT@	2	Address of horizontal tab table
1F	SDTFHTTL	1	Length of horizontal tab table
20	SDTFCT@	2	Address of compaction table
22	SDTFRSV1	16	Reserved
32	SDTFSYSS	2	System sense bytes (see Figure 2-38 for a description of possible entries)
34	SDTFUSNS	2	User sense data save area
36	SDTFRSV2	4	Reserved
3A	SDTFCCFB	1	First before carriage control byte: Bit 0 off = Space operation on = Channel operation Bits 1-7 = For space operations: Number of lines to be spaced (up to 127) For channel operations: If 0, a vertical tab If nonzero, the channel number to skip to.
3B	SDTFCCSB	1	Second before carriage control byte. See bit descriptions for label SDTFCCFB above
3C	SDTFCCFA	1	First after carriage control byte. See bit descriptions for label SDTFCCFB above
3D	SDTFCCSA	1	Second after carriage control byte. See bit descriptions for label SDTFCCFB above
3E	SDTFFMHL	1	Function manager header length byte (X'06')
3F	SDTFFMHT	1	Function manager header type byte (X'01')

Figure 2-37 (Part 3 of 4). Format of SNA DTF (Postopen)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
40	SDTFFMHM	1	Device select byte: X'80' off = Inbound X'80' on = Outbound X'30' = Printer X'20' = Card X'10' = Disk X'00' = Console Bits 4-7 are the device subaddress field
41	SDTFFMHR	1	Reserved
42	SDTFFMHP	1	Function manager header data set control byte: X'80' = Data set interrupt indicator X'60' = All of data set X'40' = Beginning of data set X'20' = End-of-data set X'04' = Compress X'02' = Compact
43	SDTFFMH	1	Reserved
44	SDTFGUBS	2	Data management user buffer save area
46	SDTFPPBS	2	Data management buffer save pointer
48	SDTFSCTR	2	Data management SCB/TRN count save area
4A	SDTFSCBS	1	Data management SCB save
4B	SDTFCHAR	1	Data management SCB character save
4C	SDTFLAG	1	DTF flag byte: X'80' = Return without system wait, no completion code set

- **Figure 2-37 (Part 4 of 4). Format of SNA DTF (Postopen)**

Sense Bytes	Description
X'0001'	Component now available
X'0802'	Intervention required
X'0805'	Session limit exceeded
X'0809'	Mode inconsistency
X'080A'	Permission rejected
X'0811'	Hard break
X'0813'	Bracket bid reject
X'0815'	Function active
X'081B'	Receiver in transmit mode
X'081C'	Function not executable
X'0821'	Invalid session parameter
X'0824'	Component aborted
X'0825'	Component not available
X'0827'	Intermittent error
X'1002'	RU length error
X'1003'	Function not supported
X'1007'	Category not supported
X'1008'	Invalid FM header
X'2001'	Invalid sequence number
X'2002'	Invalid chain
X'2005'	Data traffic state reset
X'2007'	Data traffic not reset
X'4010'	Alternate code not supported
X'8004'	Invalid destination address field
X'8005'	No session
X'8007'	Segmenting error
X'8008'	Physical unit not active
X'8009'	Logical unit not active

Figure 2-38. Possible Sense Bytes for System/32 SNA

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	SPCDEVC	1	Device code, set to X'00' by RPG II. Internally allocated as X'51'
1	SPCUPSI	1	External switches (UPSI). Checked by open routine
2	SPCATTR1	2	Attribute bytes (set by RPG II) *Byte 1:* X'80' = Input file X'40' = Output file
	SPCATTR2		*Byte 2:* X'08' = Dual I/O buffer X'01' = File is open
4	SPCRSV1	2	DTF chain pointer A: Backward address of previous DTF in chain: (X'FFFF', if first in chain)
6	SPCCHAIN	2	DTF chain pointer B: Forward address of next DTF in chain: (X'FFFF', if end of chain)
8		2	Reserved
A		2	Reserved
C	SPCLRADD	2	Address of logical record.
E	SPCCODE	1	Completion code: Set by data management before returning to RPG II X'40' = Normal completion X'41' = Controlled cancel X'42' = End of job
F	SPCCOMMD	1	Command issued by RPG II X'80' = Read card X'40' = Punch card X'10' = Close file
10	SPCINADD	2	Input I/O address: Address of buffer area where record is read into for a read card command

Figure 2-39 (Part 1 of 2). Format of Data Recorder DTF

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
12	SPCOUTAD	2	Output I/O address: Address of buffer area where record is punched. The record is first moved from the buffer pointed to by the logical record address to this area
14	SPCBLGTH	2	Block length (not checked by data management)
16	SPCRLGTH	2	Record length: Length of record to be read or punched X'50' = 80 bytes (129) X'60' = 96 bytes (129 or 5496) *Note:* If record length is X'60' for the 129, the last 16 bytes of each record will be blanks on input or will not be punched on output.
18	SPCARRAY	2	Reserved

Figure 2-39 (Part 2 of 2). Format of Data Recorder DTF

	Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
	0	SPCDEVC	1	Device code Initially set to X'51' by data management
	1	SPCUPSI	1	External switches (UPSI)
	2	SPCATTR1	1	First attribute byte. Bits set by user program: X'80' = Input file
	3	SPCATTR2	1	Second attribute byte
4	5	SPCRSV1	2	Backward chain pointer
6	7	SPCCHAIN	2	Forward chain pointer
8	9	SPCRSV2	2	ARR save area
A	B	SPCRSV3	2	XR1 save area
C	D	SPCLRADD	2	Beginning address of current record to be returned to the user program

- **Figure 2-40 (Part 1 of 2). Format of Magnetic Character Reader DTF**

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
E	SPCCCODE	1	Completion code: X'40' = Normal completion code X'41' = Controlled cancel X'42' = End of job
F	SPCCOMMD	1	Command: X'80' = Read command X'10' = Close request
10 ~~11~~	SPCINADD	2	Beginning address of I/O buffer
12 ~~13~~	SPCOUTAD	2	

14 ~~15~~	SPCBLGTH	2	Block length of I/O area:
			Minimum = 550
			Maximum = 4070

16 ~~17~~	SPCRLGTH	2	Record length (must be = 55)
18 ~~19~~	SPCARRAY	2	Address of DTT

- **Figure 2-40 (Part 2 of 2). Format of Magnetic Character Reader DTF**

Parameter Lists

When one routine calls another routine, the calling routine can pass data to the other routine. The calling routine tells the other routine precisely how to perform a function. The main storage area in which this data is placed is called parameter lists.

How to Find

Parameter lists do not have fixed addresses. When a routine is requested, XR2 contains the address of the start of the parameter list associated with that routine (with the exception of the inline parameter lists). Inline parameter lists are used with an RIB and follow the RIB.

Format

Figures 2-41 through 2-56 show the format of the IBM System/32 system parameter lists. Note that an RIB value can follow the instruction requesting a routine; the related RIB value, if any, is identified for each parameter list.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	HFPDISPY	1	X'80' = Record not previously displayed X'00' = Record previously displayed
1	HFPLEN	1	Record length
2	HFPINBUF	2	Address of leftmost byte of record to be placed in history file

Note: This 4-byte area is required by the history file put routine (no RIB).

Figure 2-41. Format of History File Put Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	–	1	RIB byte (X'0F')
1	–	1	Load/fetch option: X'00' = Load X'10' = Fetch
2	–	1	Q-code: X'00' = Alter/display control storage X'01' = Diskette I/O X'02' = Initialize data recorder, MCU, or magnetic character reader X'03' = 1255 magnetic character reader diagnostic X'04' = BSCA ASCII X'05' = BSCA EBCDIC or SDLC X'06' = EOJ/dump control storage/set trace table X'07' = Emulator X'08' = Alternate sector assignment X'09' = Diagnostic program 1 X'0A' = Diagnostic program 2 X'0B' = BSCA or SDLC wrap test X'0C' = Single form/ledger cards
3	–	1	Reserved

Note: This 4-byte area is required as input to the load control storage routine (RIB = X'0F').

Figure 2-42. Format of Load Control Storage Inline Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	MCLDMSGM	1	Message member type—identifies which message member will be used. The disk address of the specified member is found in the system communication area or the program level communication area: X'80' = Headings and miscellaneous text X'40' = Command key message member X'20' = System/32 SCP message member X'10' = Program message member X'08' = User message member X'01' = Second level message X'00' = First level message
1	MCLDMIC	2	Message identification code (MIC)
3	MCLDADDR	2	Address of message return buffer (40-byte buffer for first level message; 200-byte buffer for second level message)
5	MCLDLGTH	1	Length of message being returned; if 0, indicates that specified message was not retrieved

Note: This 6-byte area is required as input to the message retrieve routine (RIB = X'94').

Figure 2-43. Format of Message Retrieve Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	LDRDTYPE	1	Module type, X'D6'
1	LDRDNAM6	6	Name of module to be loaded
7	LDRDRESV	1	Reserved
8	LDRDLOD6	2	Main storage load address for 6-character name

Note: This 10-byte area for 6-character module names is required for the relocating loader find routine. For the final format of the relocating loader parameter list, see Figure 2-46 (RIBs = X'69', X'71', X'79', and X'7D').

Figure 2-44. Initial Format of Relocating Loader Parameter List—Find (6-character Module Name)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	LDRDTYPE	1	Module type, X'08'
1	LDRDNAM8	8	Name of module to be loaded
9	LDRDLOD8	2	Main storage load address for 8-character name

Note: This 11-byte area for 8-character module names is required for the relocating loader find routine. For the final format of the relocating loader parameter list, see Figure 2-46 (RIBs = X'69', X'71', X'79', and X'7D').

Figure 2-45. Initial Format of Relocating Loader Parameter List—Find (8-character Module Name)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	LDRDSS	2	Sector address of member
2	LDRDTEXT	1	Number of text sectors
3	LDRDLINK	2	Link-edit address
5	LDRDRLD	1	Displacement of the first byte of the relocation dictionary (RLD) in the first sector containing RLDs
6	LDRDSTRT	2	Start control address to which control is to be passed after module is loaded

Figure 2-46 (Part 1 of 2). Format of Nonfind Relocating Loader Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
8	LDRDLOAD	2	Main storage load address (set by caller or left blank)

Note: This 10-byte area is required for nonfind relocating loader routines (RIBs = X'68', X'70', X'78', and X'7C').

Figure 2-46 (Part 2 of 2). Format of Nonfind Relocating Loader Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
Initial List (Input):			
0	GETFNCT	1	Function/request byte: X'80' = Find X'40' = Get first X'20' = Get next
1	GETQCODE	1	Q-byte (must be X'A1')

2	GETDTYPE	1	Member type:
			S = Source
			P = Procedure

3	GETDNAME	8	Member name
B	GETDADDR	2	Address of record buffer
D	GETDSIZE	1	Maximum size of buffer (bytes)
Updated List (Output):			
0	–	1	Function/reply byte: X'08' = Terminal error, bad data X'04' = Truncated record X'02' = No find X'01' = End of file
1	GETQCODE	1	Reserved; do not change
2	GETDSTRT	2	Sector address of start of module
4	GETDEND	2	Sector address of end of module
6	GETDNEXT	3	Sector address and displacement of next record

Figure 2-47 (Part 1 of 2). Format of Source/Procedure Get Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
9	GETDLGTH	1	Record length of member
A	GETDRSV	4	Reserved

Note: This 14-byte area is required for the source/procedure get routine (RIB = X'93'). The contents of this list change after a find or find-and-first-read is performed.

Figure 2-47 (Part 2 of 2). Format of Source/Procedure Get Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	SGTDFUNC	1	Function byte 1: X'80' = Read record from initiator work area X'40' = Read record from format 1 area X'10' on = Diskette X'10' off = Disk—unit for get-by-name-and-unit-request (valid only if X'40' and X'04' are on) X'08' = First operation of type—indicates first record in specified area should be read; otherwise, next logical record read X'04' = Get-by-name-and-unit request; valid only if X'40' is on X'02' = Logical record pointed to by user's I/O area is written back into the last record read in the format 1 area; valid only if X'40' is on X'01' = Read record from inquiry rollout SWA Test made for X'01', X'40', and X'80', in that order; if none is on, return code set to X'41'
1	SGTDLGTH	1	Function byte 2: indicates number of bytes (X'01'-X'FF') in the logical record being processed
2	SGTDRTRN	1	Return code: X'80' = End of file X'41' = Invalid request X'40' = Successful completion

Figure 2-48 (Part 1 of 2). Format of SWA Get Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
3	SGTDIOAR	2	Storage address of the leftmost byte of the user's I/O area; the requested record is placed at this address
5	SGTDNAME	8	Filename for get-by-name-and-unit request (left-justified)

Note: This 13-byte area is required as input to the SWA get routine (RIB = X'87').

Figure 2-48 (Part 2 of 2). Format of SWA Get Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	SPTDFUNC	1	Function byte 1: X'80' on = Place record in initiator work area X'40' on = Place record in SWA format 1 area X'20' on = Place record in procedure parameter save area X'08' on = First operation of type—indicates data is to be written as first logical record of the specified area X'08' off = Data written as next logical record in the specified area X'04' on and X'40' on = Used internally by allocate function; zeros rest of sector X'02' on and X'40' on = Data is written into the last logical record read in format 1 area Test made for X'20', X'40', and X'80', in that order; if none is set on, return code X'41' set
1	SPTDLGTH	1	Function byte 2: indicates number of bytes (X'01'-X'FF') in the logical record being processed
2	SPTDRTRN	1	Return code: X'80' = End of file X'41' = Invalid request X'40' = Successful completion

Figure 2-49 (Part 1 of 2). Format of SWA Put Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
3	SPTDIOAR	2	Storage address of the leftmost byte of the user's I/O area; the record to be written is placed in this field

Note: This 5-byte area is required as input by the SWA put routine (RIB = X'88').

Figure 2-49 (Part 2 of 2). Format of SWA Put Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	SRWDFNC1	1	Function byte 1: X'80' off = Read a sector X'80' on = Write a sector X'08' on = First operation of type—indicates first sector in area is to be accessed X'08' off = Next logical sector in area is to be accessed X'02' on = Last sector read is to be updated; X'08' in function byte 2 must be on X'01' on = Access inquiry rollout SWA
1	SRWDFNC2	1	Function byte 2: X'40' = Access initiator work area X'20' = Access procedure parameter save area X'08' = Access format 1 area X'04' = Access SWA format 5 area X'02' = Access SWA index area X'01' = Access SWA common area Bits in function bytes are tested in the following order: X'01' of SRWDFNC2 X'01' of SRWDFNC1 X'02' of SRWDFNC2 X'04' of SRWDFNC2 X'20' of SRWDFNC2 X'08' of FRWDFNC2 X'40' of SRWDFNC2 If none of these bits are on, return code X'41' is set

Figure 2-50 (Part 1 of 2). Format of SWA Read/Write Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
2	SRWDRTRN	1	Return code: X'80' = End of file X'41' = Invalid request X'40' = Successful completion
3	SRWDIOAR	2	Address of the leftmost byte of the user I/O area

Note: This 5-byte area is required as input by the SWA read/write routine (RIB = X'89').

Figure 2-50 (Part 2 of 2). Format of SWA Read/Write Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	SINDOPTN	1	Option/reply byte provides a means of communicating with sysin. Bits 0–3 are not used as input to sysin, but are used by sysin to indicate option codes on output (see note 1): X'08' = Read next record from current buffer X'04' = Log to history file and syslog device X'02' = Handle records with asterisk in column 1 (source sysin)
1	SINDNEXT	2	Reserved
3	SINDCRNT	2	Current buffer: address of the leftmost byte of the 120-byte record in the buffer

Notes:

1. Byte 0 (the operation byte) is returned by sysin with one of the following codes:
   X'40' = Successful operation; the current buffer contains the address of a record ready for processing
   X'50' = End of file found in the input record addressed by the current buffer—caused when /* is found by keyboard/sysin
2. This 5-byte area is required by the source and keyboard sysin routines (RIB = X'86').

Figure 2-51. Format of Sysin Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
Type 1 – Output from Message Member			
0	LSTDOPT1	1	Function: X'80' = Output from message member (type 1) X'40' = Output from message member Bits 4–7 = 4-bit identifier X'0' = SCP X'1' = Data management X'2' = IOS X'3' = RPG X'4' = SCP nucleus X'5' = SCP linkage editor and word processing X'6' = SCP utilities X'7' = Sort X'8' = DFU (data file utilities) X'9' = SCP librarian X'A' = User-defined message access X'B' = SEU X'C' = SCP BSC and SNA/SDLC X'D' = SCP system service X'E' = SCP MRJE/BWS X'F' = HDR (headings)
1	LSTDCHAR	1	Minor ID (not used by syslist)
2	LSTDMIC	2	Message identification code (MIC)
Type 2 – Output from Program			
0	LSTDOPT2	1	Output from program: X'80' off = Output from program (type 2) X'20' off = Truncate message length to 40 if syslist is the display screen X'20' = Print entire message if syslist is the display screen X'0F' = Print option bits, not used by System/32 SCP
1	LSTDPAGE	1	Print control byte: X'80', X'40', X'20', or X'10' = Any of these bits on will cause skip to line 1 before printing X'03' = Space 3 lines after printing X'02' = Space 2 lines after printing X'01' = Space 1 line after printing X'00' = No space after printing

Figure 2-52 (Part 1 of 2). Format of Syslist Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
2	LSTDLEN	1	Length of output message (maximum 132)
3	LSTDADDR	2	Address of leftmost byte of buffer in which message is contained

Note: This parameter list is required as input to the syslist routine (RIB = X'92'), and can be in either of two formats (4 or 5 bytes long), depending on where the output comes from.

Figure 2-52 (Part 2 of 2). Format of Syslist Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
Type 1 - Output from Message Member			
0	LOGDFNC1	1	Function byte: X'80' = Output from message member (4-byte parameter list) X'40' = Output from message member with operator action X'20' off = Omit format line for output from message member X'10' off = 5-byte parameter list X'10' = 7-byte parameter list Bits 4–7—message member identifier: X'0' = C (SCP) X'1' = D (data management magnetic character reader) X'2' = I (IOS) X'3' = R (RPG, assembler, FORTRAN IV, file conversion utility) X'4' = V (SCP nucleus) X'5' = E (SCP linkage editor, overlay linkage editor, and word processing) X'6' = U (SCP utilities) X'7' = S (sort utility) X'8' = F (data file utility) X'9' = L (SCP librarian) X'A' = P (user-defined message access) X'B' = K (source entry utility) X'C' = B (SCP BSC/SDLC) X'D' = T (SCP system services) X'E' = X (SCP MRJE) X'F' = H (heading and miscellaneous text)

Figure 2-53 (Part 1 of 2). Format of Syslog Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
1	LOGDMIN	1	Subcomponent (minor) ID–a printable EBCDIC character to be displayed on the format line
2	LOGDMIC	2	Message identification code (MIC)
4	LOGDACT	1	Operation and action byte:
		Output	X'80' = Proceed X'40' = Retry X'20' = Controlled cancel X'10' = Call EOJ
		Input	X'08' = Ignore X'04' = Retry X'02' = Controlled cancel X'01' = Terminate the job
5	LOGDSUB	2	Subcomponent-ID—two printable EBCDIC characters to be displayed on the format line (blanks displayed, if no characters specified)
Type 2 - Output from Program			
0	LOGDFNC2	1	Function byte: X'80' off and X'40' off = Output from program
1	LOGDPAGE	1	Page control byte: X'80', X'40', X'20', or X'10' = Skip is made to line 1 of next page before printing X'03' = Space three lines after print X'02' = Space two lines after print X'01' = Space one line after print X'00' = No space after print
2	LOGDLEN	1	Message length (maximum 132)
3	LOGDADDR	2	Address of leftmost byte of buffer in which message is contained

Note: This parameter list is required as input to the syslog routine (RIB = X'85') and is in one of two formats (7 or 5 bytes long) depending on where the output comes from.

Figure 2-53 (Part 2 of 2). Format of Syslog Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
Initial List (input) - 6-character Name			
0	FNDDTYPE	1	Library type: C'O' = Load module C'R' = Subroutine
1	FNDDNAM6	6	Module name
7	FNDDPCK	1	Reserved
8	–	6	Six additional bytes will be filled in by the find routine
Initial List (input) - 8-character Name			
0	FNDDTYPE	1	Library type: X'08' = Load member X'04' = Subroutine X'02' = Source member X'01' = Procedure member
1	FNDDNAM8	8	Module name
9	FNDDPCK8	1	Reserved
A	–	4	Four additional bytes will be filled in by the find routine
Updated List (output) - 6- and 8-character Names			
0	FNDDADDR	2	Disk address of requested library module
2	FNDDNUMS	1	For load module, number of text sectors; for subroutines, category; for procedure and source members, record size
(The next six bytes are filled in for load modules only:)			
3	FNDDLINK	2	Link-edit address
5	FNDDRLDD	1	Displacement of RLDs in last text sector
6	FNDDSCTL	2	Entry point of module
8	FNDDCRSZ	1	Size of programs (in sectors)
9	FNDDATTR	2	Attributes of module (Figure 2-89—DIRATTR)

Figure 2-54 (Part 1 of 2). Format of System Find Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
B	FNDDRELL	1	Release level of module
C	FNDDTOTL	2	Total size of module (in sectors)

Note: This 14-byte area is required for the system find routine (RIB = X'81').

Figure 2-54 (Part 2 of 2). Format of System Find Parameter List

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	VTIDFNCT	1	Function byte: X'80' off = Read X'80' on = Write X'40' = Existence test request X'20' = Volume label request X'10' = Perform operation on specific header 1 label (HDR1) X'08' = Perform operation on first HDR1 in VTOC X'04' = Perform operation on next HDR1 in VTOC X'02' = Verify date X'01' = Delete HDR1 from VTOC
1	VTIDRTRN	1	Return code: X'40' = Normal completion X'41' = Invalid request or invalid return code from diskette IOS X'43' = Diskette door opened before this request was handled; diskette now mounted has same VOL-ID as diskette mounted before door was opened X'44' = Request not met
2	VTIDIOAR	2	Address of leftmost byte of user I/O area
4	VTIDSSFL	2	Sector address of the last sector read on the diskette, initially set to zero; used internally for next HDR1 request and when deleting an HDR1 by sector address
6	VTIDPREP	1	X'FF' = Call diskette prepare routine
7	VTIDIOS@	1	Reserved, set to X'00'
8	VTIDLBL@	8	Data set identifier; used for specific header 1 requests

Figure 2-55 (Part 1 of 2). Format of VTOC Read/Write Parameter List (Diskette)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
10	VTIDDATF	2	Address of leftmost byte of 6-byte data field

Note: This 18-byte area is required for the diskette VTOC read/write function (RIB = X'91').

Figure 2-55 (Part 2 of 2). Format of VTOC Read/Write Parameter List (Diskette)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	VTFDFNCT	1	Function byte: X'80' off = Read X'80' on = Write X'40' = Index request X'20' = Volume label request X'10' = Format 1 request X'08' = Format 5 request; if X'10' on, verify date field X'02' = Search for format 1 in locate mode X'01' off = Sector/displacement is given X'01' on = File label is given *Note:* A value of X'00' is used for the format 1 existence test. The bits are tested in the following order: 3, 6, 7, 0, 1, 2, 4. The first bit on indicates the function to be performed.
1	VTFDIXNM	1	Index number used for index request: X'00' = First index X'01' = Second index X'02' = Third index X'03' = Fourth index X'FF' = Entire index *Note:* For any request other than an index request, must be X'00'
2	VTFDRTRN	1	Return code: X'44' = Request not met X'41' = Invalid request X'40' = Good completion
3	VTFDIOAR	2	Address of leftmost byte of user I/O area (256 bytes for volume label request; 768 bytes for format 5 request; 512 bytes for individual VTOC request; 2048 bytes for entire index request; or 64 bytes for format 1 request)

Figure 2-56 (Part 1 of 2). Format of VTOC Read/Write Parameter List (Disk)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
5	VTFDWORK	2	Address of leftmost byte of user work area (256 bytes for locate mode format 1 request; 6 bytes for date field for normal format 1, if date verify is requested)
7	VTFDFMT1	2	Sector/displacement of rightmost byte of format 1 (displacement from beginning of sector)
9	VTFDLBL@	8	Label of format 1

Note: This 17-byte area is required for the disk VTOC read/write routine (RIB = X'8A').

Figure 2-56 (Part 2 of 2). Format of VTOC Read/Write Parameter List (Disk)

Library Control Blocks

Library control blocks are main storage areas used by $MAINT library routines.

How to Find

The addresses of library control blocks are not fixed. When library routines are called, XR2 points to the start of the associated library control block.

Format

The format of a library control block is shown in Figure 2-57.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	LCBADDRS	1	Switches to indicate modules loaded: X'80' = $MAILD loaded X'40' = $MACOM loaded X'10' = $MAFND loaded X'08' = Library control sector is in main storage X'04' = $MAIST loaded
1	LCBILD@	2	Main storage address of $MAILD module or $MAIST module
3	LCBCOM@	2	Main storage address of $MACOM module
5	LCBFND@	2	Main storage address of $MAFND module
	or		
5	LCBFND@		Main storage address of 3 sectors for save area (256 bytes) and directory entry stack (512 bytes) during sector mode file-to-library copy

Figure 2-57 (Part 1 of 3). Library Control Block Format

This page intentionally left blank.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
7	LCBLCS@	2	Main storage address of the library control sector
9	LCBOPER	2	Operation (request) bytes: *Byte 0:* X'80' = Open—first call per module X'20' = Close—operation for module is complete X'10' off = Put—putting a module (records or sector) into the library X'10' on = Get—retrieving a module (records or sectors) from the library X'08' off = Record mode X'08' on = Sector mode X'04' = Replace module if duplicate exists in library X'02' = A find is unnecessary; the directory entry in the library control block (LCB) is complete; this entry is valid only for a sector get/put in get mode *Byte 1:* X'80' = Update PTF log X'40' = Get PTF information X'20' = Delete PTF information X'10' = Replace PTF information X'08' = Linkage editor call
B	LCBOMP	2	Completion code (reply) bytes: *Byte 0:* X'80' = Operation successful X'40' = Invalid record length X'20' = Not enough space in library for module X'10' = Not enough space for another directory entry X'08' = Duplicate name in directory; replace was not specified X'04' = Module not found X'02' = Invalid data (character); record has been ignored X'01' = Module closed (bit used by $MAPGS and $MAPUR) *Byte 1:* Reserved (do not modify)

Figure 2-57 (Part 2 of 3). Library Control Block Format

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
D	LCBLIBSW	1	Librarian switches: X'80' = LCB has been opened X'40' = Replacing existing module X'20' = Replacing module in original slot X'10' = Open/close switch for room is available X'08' = Do not close X'04' = Insert directory entries X'03' = Reserved
E[1]	LCBREC@	2	Address of record
10[1]	LCBRECL	1	Length of record
11	LCBUFF@	2	Address of the leftmost byte of the buffer supplied
13	LCBUFFS	1	Buffer size (in sectors; minimum is 2 sectors)
14	LCBDIR	23	Library directory entry (for a detailed description, see Figure 2-99)
2B	LCB#AVMB	2	Number of available sectors in the library
2D	LCBCOARR	2	ARR save area for open/close
2F	LCBPTFLG	8	PTF information for the module
37	LCBOPNT1	2	Save area for open
39	LCBFNDPM	14	Storage area for find parameter list—the first seven bytes are used to store halt parameter list; the first byte of this area identifies the number of sectors in the user buffer (for a detailed description, see Figure 2-54)
47	LCBTOGP	2	Number of sectors left to get or put
49	LCBDAGP	2	Disk address of next get or put
4B	LCBSAVTN	9	Type and name used by PTF log module

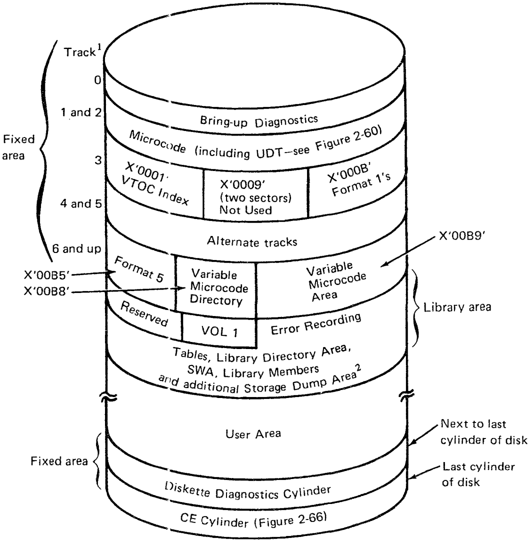
[1] Valid only for source and procedure members, in record mode.

Figure 2-57 (Part 3 of 3). Library Control Block Format

DISK ORGANIZATION

The disk on IBM System/32 can be divided into three major functional areas (Figure 2-58):

- Fixed areas
- Library area
- User data file area


Track[1]
0
1 and 2
3
4 and 5
6 and up
Fixed area
Bring-up Diagnostics
Microcode (including UDT—see Figure 2-60)
X'0001' VTOC Index
X'0009' (two sectors) Not Used
X'000B' Format 1's
Alternate tracks
X'00B9'
X'00B5'
X'00B8'
Format 5
Variable Microcode Directory
Variable Microcode Area
Library area
Reserved
VOL 1
Error Recording
Tables, Library Directory Area, SWA, Library Members and additional Storage Dump Area[2]
User Area
Next to last cylinder of disk
Last cylinder of disk
Fixed area
Diskette Diagnostics Cylinder
CE Cylinder (Figure 2-66)

[1] The 13.7 megabyte disk has the following differences:

a. Tracks 2 and 4 are unused.
b. The microcode resides on tracks 1 and 3.
c. The VTOC is on track 5.
d. The alternate tracks are tracks 6 and 7.
e. The user area begins with track 8 and ends with track 902.
f. Tracks 903, 904, and 905 contain the diskette diagnostics.
g. The last cylinder (tracks 906, 907, and 908) is the CE cylinder.

[2] See Figures 2-67, 2-68.

Figure 2-58. IBM System/32 Disk Format

For addressing purposes, the disk and diskette are divided into cylinders, tracks, and sectors (Figure 2-59). Note that tracks 0, 1, and 2 (0 through 4 for 13.7 megabyte disk) on disk cannot be accessed from main storage by any means; this prevents accidental destruction of information on those tracks. (IBM System/32 SCP distribution on diskettes does not modify tracks 0, 1, and 2.) The capacities shown in Figure 2-59 are total capacities, not user capacities. For user capacities, see *IBM System/32 Functions Reference Manual,* GA21-9176.

IOS begins sector addressing on disk with track 3 (track 5 for 13.7 megabyte disk), addressed as X'0001', and numbers sequentially in hex. IOS converts the sector address to the CCHS format for internal use (Appendix G). For a detailed description of the diskette, see Figure 2-110.

	Disk				Diskette	
	3.2 Megabyte	**5.0 Megabyte**	**9.1 Megabyte**	**13.7 Megabyte**	**128 Byte**	**512 Byte**
Bytes per sector	256	256	256	256	128	512
Sectors per track	60	60	60	60	26	8[1]
Bytes per track	15,360	15,360	15,360	15,360	3,328	4,096[1]
Tracks per cylinder	2	2	2	3	1	1
Sectors per cylinder	120	120	120	180	26	8[1]
Bytes per cylinder	30,720	30,720	30,720	46,080	3,328	4,096[1]
Total cylinders	109	169	303	303	77	77
Total tracks	218	338	606	909	77	77
Total sectors	13,080	20,280	36,360	54,540	2,002	634[1]
Total bytes	3,348,480	5,191,680	9,308,160	13,962,240	256,256	314,624[1]

[1] The first track (track 0) of a 512-byte per sector diskette contains the same number of sectors and bytes per sector as the 128-byte per sector diskette.

Figure 2-59. Storage Capacities for IBM System/32 Disks and Diskettes

Fixed Areas

The fixed areas consist of 10 tracks, which contain the following:

- Track 0—Bring-up diagnostics (part of the micro IPL function)
- Tracks 1 and 2 (1 and 3 for 13.7 megabyte disk)—Microcode for emulator and I/O device control, control storage nucleus, main storage nucleus, miscellaneous control storage subroutines, and save areas and the extended control storage supervisor; microcode transients (end-of-job, alternate sector assignment, I/O control, control storage diagnostics transients)
- Track 3 (5 for 13.7 megabyte disk) VTOC index, and format 1's (Figures 2-62, 2-63, and 2-64)
- Tracks 4 and 5 (6 and 7 for 13.7 megabyte disk)—Sectors available for assignment as alternates for defective sectors
- Track 6 (8 for 13.7 megabyte disk)—Format 5
- Variable microcode directory, and variable microcode area
- Next-to-last cylinder (tracks 903, 904, and 905 for 13.7 megabyte disk)—Diskette diagnostic area
- Last cylinder (tracks 906, 907, and 908)—First two sectors of each track are used as a read/write area for CE diagnostics; remaining sectors used as save area for an abnormal termination dump of main storage, control storage, and history file (see Figure 2-66 for a layout of the CE cylinder)

Unit Definition Table (UDT)

The unit definition table is a 256-byte area consisting of sixteen 16-byte entries which describe the devices supported by IBM System/32 SCP.

How to Find

The UDT table is found on track 2 (track 3 for a 13.7 megabyte disk) sector X'3B' of disk storage, which is microcoded and unaddressable (last sector of DIAG01 control storage transient).

Format

The formats of the possible entries in this table are shown in Figure 2-60.

Figure 2-60 (Part 1 of 5). Unit Definition Table Format

Byte	0	1-8	9	10	11	12	13	14	15
Description	Device Code	Device Symbolic Name (Left-Justified)	Program Support Flag[1]	Device Dependent Information			System Information[2]		Module Number (Hex)
Possible Entries	01	CPU	44	Control storage size (4K byte increments)	Main storage size (4K byte increments)	Not used	01	Reserved	80 (16K system) 90 (24K or 32K)
	10	KEYBOARD	D4	Language code[3]	Not used	Not used	01	Reserved	[3]
	40	CRT (display screen)	04	Language code[3]	Not used	Not used	01	Reserved	00
	50	MCU (Mag Card Unit)	D5	Not used	Not used	Not used	03	Reserved	A0
	51	Data Recorder	D5	Not used	Not used	Not used	03	Reserved	A2
	52	Magnetic Character Reader	F1	Not used	Not used	Not used	04	Reserved	86
	80	BSC	D5	Microcode configuration[5]	Not used	Diagnostic configuration[6]	03	Reserved	91 (Domestic) 92 (World Trade)
	81	SDLC	D5	Microcode configuration[5]	Timer count	Station address	03	Diagnostic configuration[7]	96 (Domestic) 97 (World Trade)
	A0	62GV (disk)	F4	Cylinder number of CE cylinder		Not used	01	Reserved	10 (3.2 megabyte) 11 (5.0 megabyte) 12 (9.1 megabyte) 13 (13.7 megabyte)
	D0	33FD	D4	Not used	Not used	Not used	01	Reserved	81 (EBCDIC) 82 (ASCII)

Byte	0	1-8	9	10	11	12	13	14	15
Description	Device Code	Device Symbolic Name (Left-Justified)	Program Support Flag[1]	Device Dependent Information			System Information[2]		Module Number (Hex)
Possible Entries	E0	PRINTER (line)	D4	Print belt size[8]	Image size[9]	Printer configuration[10]	01	Reserved	15 Half line, 285 lines per minute, and keyboard module number = B1 16 Half line, 285 lines per minute, and keyboard module number not = B1 17 Half line, standard, and keyboard module number = B1 18 Half line, standard, and keyboard module number not = B1 21 Standard and Katakana 22 Standard and 8 lines per inch 25 285 lines per minute and Katakana 26 285 lines per minute and 8 lines per inch
	E1	PRINTER (serial)	D4	Language code[4]	Not used	Printer configuration[11]	01	Reserved	Module number[4]
	00	Empty UDT entries							

Figure 2-60 (Part 2 of 5). Unit Definition Table Format

[1] Program support flag byte:

X'80' = System test module
X'40' = ERAP module
X'20' = FRIEND module
X'10' = MDI module
X'04' = Special diagnostic module
X'01' = Feature device

[2] Byte 13 contains an ID corresponding to the diagnostic diskette where the basic support resides:

X'01' = DIAG01
X'02' = DIAG02
X'03' = DIAG03
X'04' = DIAG04

[3] When byte 10 of the keyboard UDT entry contains one of the following language codes, byte 15 must contain the corresponding module number.

Language Code	Module Number	Language Code	Module Number
X'01' = USA	X'71'	X'0A' = Norway	X'79'
X'02' = ASCII	X'42'	X'0B' = Spain	X'7A'
X'03' = Germany/Austria	X'72'	X'0C' = Spanish speaking	X'7B'
X'04' = Finland	X'73'	X'0D' = Sweden	X'7C'
X'05' = Brazil	X'74'	X'0E' = France I	X'7D'
X'06' = Denmark	X'75'	X'0F' = United Kingdom	X'7E'
X'07' = Belgium/France II	X'76'	X'10' = Katakana	X'7F'
X'08' = Italy	X'77'	X'11' = Portugal	X'65'
X'09' = Japan	X'78'	X'12' = Upper lowercase USA	X'A1'

Figure 2-60 (Part 3 of 5). Unit Definition Table Format

[4] The following language codes and module numbers are used in the 40 and 80 characters per second printer; or the 40, 80, and 120 characters per second printer. If the part number is 1607033 or lower, the card at location A2R2 is for the 40/80 characters per second printer. If the part number is higher than 1607033, the card is for the 40/80/120 characters per second printer.

40 and 80 characters per second printer:

Language Code	Module Number
X'01' = USA	X'07' = 40 characters per second unidirectional
X'01' = USA	X'0A' = 40 characters per second
X'01' = USA	X'0C' = 80 characters per second
X'10' = Katakana	X'08' = 40 characters per second unidirectional
X'10' = Katakana	X'0B' = 40 characters per second
X'10' = Katakana	X'0D' = 80 characters per second
X'11' = Portugal	X'09' = 40 characters per second unidirectional
X'11' = Portugal	X'0E' = 40 characters per second
X'11' = Portugal	X'0F' = 80 characters per second

40, 80, and 120 characters per second printer:

Language Code	Module Number
X'01' = USA	X'50' = 40 characters per second unidirectional
X'01' = USA	X'53' = 40 characters per second
X'01' = USA	X'56' = 80 characters per second
X'01' = USA	X'04' = 120 characters per second
X'10' = Katakana	X'51' = 40 characters per second unidirectional
X'10' = Katakana	X'54' = 40 characters per second
X'10' = Katakana	X'57' = 80 characters per second
X'10' = Katakana	X'05' = 120 characters per second
X'11' = Portugal	X'52' = 40 characters per second unidirectional
X'11' = Portugal	X'55' = 40 characters per second
X'11' = Portugal	X'58' = 80 characters per second
X'11' = Portugal	X'06' = 120 characters per second

● **Figure 2-60 (Part 4 of 5). Unit Definition Table Format**

[5] Microcode configuration byte:
X'80' On = Half rate
X'80' Off = Full rate
X'40' = Internal clock feature
X'20' = IBM modem
X'10' = World Trade answer tone
X'08' = Standby line
X'04' = Multipoint tributary station
X'02' = DTR switched line
X'01' = Point-to-point leased
X'02' and X'01' = CDSTL switched line

[6] Diagnostic configuration byte (BSCA):
X'80' Off = EBCDIC system
X'80' On = ASCII system
X 02' = IBM 1200 bps modem
X'01' = IBM 2400 bps modem

[7] Diagnostic configuration byte (SDLC):
X'80' = NRZI selected
X'10' = World Trade
X'04' = Other IBM modem
X'02' = IBM 1200 bps modem
X'01' = IBM 2400 bps modem

[8] Print belt size:
X'60' = 192 characters
X'40' = 128 characters (96 characters)
X'20' = 64 characters
X'00' = 48 characters

[9] Image size:
X'C0' = 192 characters
X'60' = 96 characters
X'40' = 64 characters
X'20' = 48 characters

[10] Printer configuration byte (line):
X'80' = 155 lines per minute
X'40' = 100 lines per minute
X'20' = 50 lines per minute
X'10' = 285 lines per minute
X'08' = Half line spacing feature
X'04' = 8 lines per inch feature
X'02' = Katakana
X'01' = Special image

[11] Printer configuration byte (serial):
X'80' = 40 characters per second (bidirectional)
X'40' = 80 characters per second
X'20' = 120 characters per second
X'10' = 40 characters per second (unidirectional)

● **Figure 2-60 (Part 5 of 5). Unit Definition Table Format**

Disk Volume Table of Contents (VTOC)

The disk VTOC is a 60-sector area on the disk which describes the location, size, and other characteristics of each data file on the disk. The disk VTOC consists of three logical areas: the disk VTOC index, the disk format 5, and the disk format 1's.

- The first eight sectors of the VTOC contain the disk VTOC index. This is a table consisting of four 2-sector indexes each containing fifty 10-byte tag entries; each tag entry points to a corresponding disk format 1.
- Sectors X'000B' through X'003C' contain the 200 disk format 1's, each 64 bytes long.
- Sectors X'00B5' through X'00B7' contain the disk format 5, which contains block usage information.

Each active file on the disk has a tag entry in the disk VTOC index, containing the file label and a pointer to the corresponding disk format 1.

How to Find

The first eight addressable sectors of the disk (X'0001' through X'0008') contain the first VTOC index. Sectors X'000B' through X'003C' contain the format 1's. Sectors X'00B5' through X'00B7' contain the format 5.

Format

Figure 2-61 shows the organization of the disk VTOC. Figure 2-62 shows the format of the VTOC index entries. Figure 2-63 shows the format of the disk format 5. Figure 2-64 shows the format of the disk format 1's.

Track 3

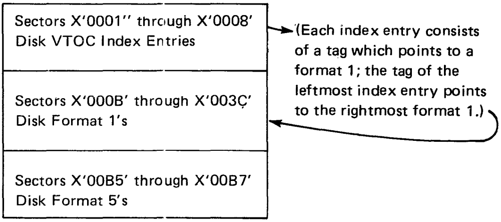

Sectors X'0001'' through X'0008'
Disk VTOC Index Entries
Sectors X'000B' through X'003C'
Disk Format 1's
Sectors X'00B5' through X'00B7'
Disk Format 5's
(Each index entry consists of a tag which points to a format 1; the tag of the leftmost index entry points to the rightmost format 1.)

Figure 2-61. Disk VTOC Organization

Disk VTOC Index

This area consists of four 2-sector (512 bytes) subindexes. Each subindex contains a label and sector/displacement entry for each format 1 entry in the disk VTOC (fifty 10-byte entries in each subindex).

How to Find

This 8-sector area is located at disk addresses X'0001' through X'0008'.

Format

The format of this area is shown in Figure 2-62.

The disk VTOC index consists of four subindexes; each subindex is 512 bytes long and consists of:

A	B	C	D	E
4 Bytes	2 Bytes	500 Bytes	5 Bytes	1 Byte

Where:

- A = 4 bytes (reserved, each set to X'00')
- B = 2 bytes (sector address of next subindex, X'0000' for last subindex)
- C = 500 bytes (fifty 10-byte tags, each corresponding to a format 1.) Each tag is in the following format:

Label of Corresponding Format 1 (see note)	Sector/Displacement of Rightmost Byte of Corresponding Format 1
8 Bytes	2 Bytes (one byte sector address and one byte displacement)

- D = 5 bytes (each set to X'00')
- E = 1 byte (the number of free tags in this subindex in hex)

Note: If a tag is not being used (no file corresponds to this tag and its format 1), the label area for the tag contains eight bytes of X'00'.

The sector/displacement values (see C above) are fixed for each tag and point to a specific format 1 entry regardless of whether an existing file corresponds to the tag.

For date chained files, the leftmost byte of the label field in each tag entry is set to X'20', except the file with the latest date. That tag entry contains the label of the file. Refer to F1CHAN in the disk format 1 (Figure 2-64) for the description of the chain address.

Figure 2-62. Format of Disk VTOC Index

Disk Format 5

This 3-sector (768 byte) area contains data file and system area block usage information used by the IBM System/32 system control program to allocate files on the disk.

How to Find

This area is located at sector address X'00B5' through X'00B7' on the disk.

Format

The format of the disk format 5 is shown in Figure 2-63.

Each bit in bytes X'00' through X'FB' in sector X'00B7' and all bits in sectors X'00B6' and X'00B5' indicate the status of a specific block (10-sector area) of the disk. For example, bit 0 of byte X'04' in sector X'00B7' indicates the status of block 1983, while bit 7 of byte X'FB' in sector X'00B7' indicates the status of block 0. If a bit is on, its related block is being used and is, therefore, protected. If a bit is off, its related block is not being used and is available for file allocation.

Format 5 is used to protect the fixed areas of the disk and the blocks which cannot be allocated for a given disk configuration.

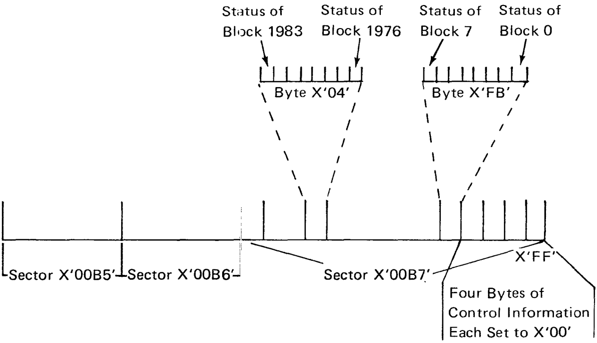


Figure 2-63. Disk Format 5 Organization Example

Disk Format 1

A 64-byte format 1 (format 1 record) describes each data file maintained on disk. A maximum of 200 data files can be maintained on disk at one time; therefore, 200 format 1's can be defined in the disk VTOC (at X'000B' through X'003C'). When a file is processed, its format 1 is converted and stored in the SWA (Figure 2-105).

How to Find

All format 1's are contained in the area at X'000B' through X'003C'. Each format 1 is addressed by a corresponding entry in the VTOC index (this entry points to the rightmost byte of the format 1).

Format

The format of each disk format 1 is shown in Figure 2-64. Note that if a VTOC format 1 is not being used (its corresponding index tag contains all zeros in the label area), the format 1 contains the tag ID in its leftmost byte and the remaining 63 bytes are set to hex zeros.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	F1TAG	1	Tag identifier of VTOC index entry (each format 1 in the VTOC has a unique tag ID, ranging from X'01' to X'C8')
1	F1CHAN	2	Chain address (sector/displacement of next VTOC format 1 in date chain) for date-differentiated files with the same file label; for nondate-chained files or the last format 1 in date chain, set to hex zeros
3	F1LABL	8	File label
B	F1DATE	6	Date the file was created (YYMMDD)
11	F1RTIN	1	Retain code: C'P' = Permanent C'S' = Scratch C'T' = Temporary

Figure 2-64 (Part 1 of 3). Format of Disk Format 1

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
12	F1TYPE	2	File type:
	Current status (status when file was opened)		*Byte 0:* X'80' = Indexed X'60' = Pseudo tape X'40' = Consecutive X'20' = Direct X'10' = Offline multivolume file X'08' = Input X'04' = Output X'02' = Update X'01' = Add
	Status when file created		*Byte 1:* X'80' = Indexed X'60' = Pseudo tape X'40' = Consecutive X'20' = Direct X'10' = Reserved
	Status when file closed		X'08' = Reserved X'04' = Sequential add X'02' = Random add X'01' = Unordered load
14	F1RECL	2	Logical record length
16	F1KEYL	1	Key length
17	F1KEYO	2	Displacement of key within record (S/D)
	or		
	F1LIBDIR	2	Sector address of library directory (configuration record) valid only for a #LIBRARY format 1
19	F1LSTR	3	Next available record (SS/D)
1C	F1LSTK	3	Next available key (SS/D)
1F	F1STDA	2	Sector address of the start of data; for nonindexed files, this is also the physical begin extent (sector address of the volume label for #LIBRARY format 1)
21	F1ENDA	2	Sector address of end of data; this is the physical end extent (next available sector)
23	F1STIX	2	Sector address of start of the file; this is the physical begin extent (first sector of file); for indexed files, it is also the beginning of the index

Figure 2-64 (Part 2 of 3). Format of Disk Format 1

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
25	F1ENIX	2	Sector address of end of index (first sector past index for indexed files)
27	F1RECN (if leftmost bit of field = 0) or	3	Number of records allocated when file created
27	F1BLKN (if leftmost bit of field = 1)	3	Number of blocks allocated when file created
2A	F1RSV1	1	Reserved
2B	F1BACK	2	Reserved (must be set to hex zeros)
2D	–	19	Reserved for use by the scheduler work area disk format 1's (set to hex zeros in VTOC format 1)
31	F1FLAG	1	Used by X'80' = $PACK or X'81' = $FREE to indicate that a data file was in process of being moved when either program abnormally terminated
32	F1SSTRT	2	Used by $PACK or $FREE to recover data file through restarting after an abnormal termination
34	F1SSEND	2	

Figure 2-64 (Part 3 of 3). Format of Disk Format 1

Variable Microcode Directory	Scientific Instruction Set Microcode	Magnetic Character Reader Run Microcode	Reserved	Magnetic Character Reader Diagnostic Microcode

Figure 2-65. Format of Variable Microcode Area

CE Cylinder

The CE cylinder is composed of the last two tracks on disk and contains the information shown in Figure 2-66.

How to Find

The CE cylinder is located at sector address X'4E0D' for 5.0 megabyte disks, X'8CDD' for 9.1 megabyte disks, and X'D32D' for 13.7 megabyte disks.

Sector Displ (hex)	Lng in Sectors (dec)	Description
0	2	CE diagnostic area
2	32	Control storage dump area (X'0000-X'0FFF' words)[1]
22	26	Main storage dump area (X'0000'-X'19FF')
3C	2	CE diagnostic area
3E	38	Main storage dump area (X'1A00'-X'3FFF')[2]
64	20	SWA history file dump area (most current 20 sectors)

[1] For 8K-word control storage systems and the magnetic character reader, the amount of control storage that cannot be dumped in this area is dumped in the additional control storage dump area following the SWA.

[2] For 24K and 32K systems, the amount of main storage that cannot be dumped in this area is dumped in the additional main storage dump area following the SWA (Figure 2-67).

Figure 2-66. Map of CE Cylinder

Disk Library Area

The library area immediately follows the variable microcode area and contains the areas shown in Figure 2-67. These areas are discussed in the pages following. (Note that the library is itself considered to be a system file called #LIBRARY and that it has an associated VTOC index tag entry and corresponding VTOC format 1.)

Reserved Area	Disk Volume Label (VOL1) (Figure 2-69)	Error Recording Tables (Variable Number) (Figure 2-72–2-90)	Diskette ID Save Area (1 sector)

Library Directory Area

Error Recording Table Directory (Figure 2-70)	System Configuration Record (Figure 2-97)	Library Control Sector (Figure 2-98)	Library Directory Entries (Figure 2-99)	Rollout/rollin (Optional) (Figure 2-100)

16K Main Storage Size

Scheduler Work Area (Figure 2-101)	Library Members { O-Load P-Procedure R-Subroutine S-Source

or

24K Main Storage Size

Scheduler Work Area (Figure 2-101)	Magnetic Character Reader Storage Dump Area (32 sectors	Additional Control Storage Dump Area 4K words–8K words (32 sectors)	Additional Main Storage Dump Area 16K–24K (32 sectors)	Library Members { O-Load P-Procedure R-Subroutine S-Source

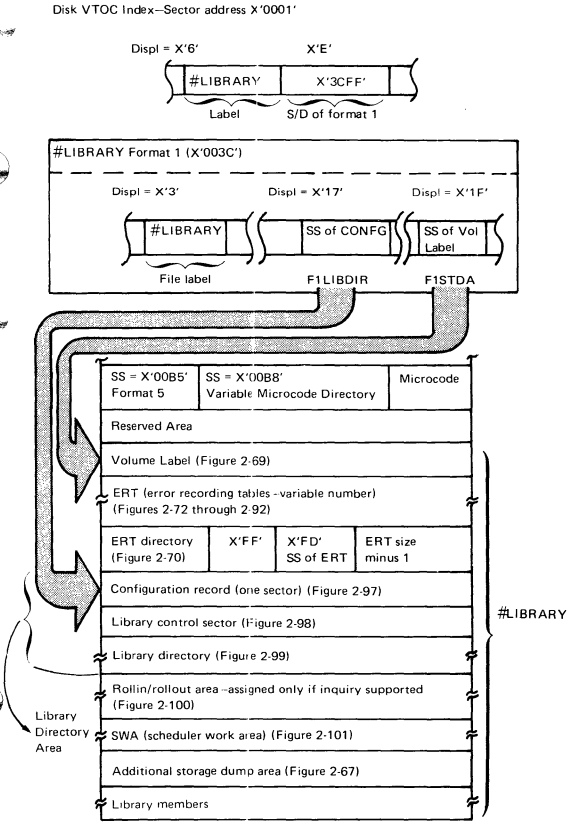
or

32K Main Storage Size

Scheduler Work Area (Figure 2-101)	Magnetic Character Reader Storage Dump Area (32 sectors)	Additional Control Storage Dump Area 4K words–8K words (32 sectors)	Additional Main Storage Dump Area 16K–32K (64 sectors)	Library Members { O-Load P-Procedure R-Subroutine S-Source

● **Figure 2-67. Format of Disk Library Area (#LIBRARY)**

Figure 2-68 shows some of the pointers available for use in finding library areas.


Disk VTOC Index—Sector address X'0001'
Displ = X'6'
X'E'
#LIBRARY
X'3CFF'
Label
S/D of format 1
#LIBRARY Format 1 (X'003C')
Displ = X'3'
Displ = X'17'
Displ = X'1F'
#LIBRARY
SS of CONFG
SS of Vol Label
File label
F1LIBDIR
F1STDA
SS = X'00B5' Format 5
SS = X'00B8' Variable Microcode Directory
Microcode
Reserved Area
Volume Label (Figure 2-69)
ERT (error recording tables - variable number) (Figures 2-72 through 2-92)
ERT directory (Figure 2-70)
X'FF'
X'FD' SS of ERT
ERT size minus 1
Configuration record (one sector) (Figure 2-97)
Library control sector (Figure 2-98)
Library directory (Figure 2-99)
Rollin/rollout area - assigned only if inquiry supported (Figure 2-100)
SWA (scheduler work area) (Figure 2-101)
Additional storage dump area (Figure 2-67)
Library members
#LIBRARY
Library Directory Area

Notes:

1. Start of configuration record plus one sector = Sector address of library control sector
2. Start of configuration minus 1 sector = Sector address of ERT directory
3. Start of ERT minus one sector = Start of volume label
4. Library control sector contains values that point to:
   Start of library (volume label)
   Start of directory
   Start of SWA
   Start of rollin/rollout area
   Start of library members

Figure 2-68. Disk Library Pointers

Reserved Area

The reserved area is an area on disk between the variable microcode area and the volume label. Its purpose is to allow for future SCP library expansion without changing the amount of disk space available to the user. For example, if the SCP library grows from one release to the next, the reserved area will decrease by the same amount.

How to Find

The size of the reserved area can be found in the library control sector, Figure 2-98.

Volume Label

This is a 1-sector area containing owner identification information and System/32 SCP information about the disk.

How to Find

The volume label in the library area can be found by adding X'00B8' to the value of LCSRSVSZ (reserved area size), which is in the library control sector (Figure 2-98) or referring to displacement of X'1F' of the #LIBRARY format 1.

Format

The format of this area is shown in Figure 2-69.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	VLFDID	4	Label identifier (C'VOL1')
4	VLFDVLID	6	Volume identifier (set by $LOAD utility)
A	VLFDACES	1	Accessibility (set to blanks)
B	VLFDVTOC	5	VTOC pointer—binary hardware address (CCHSS) of the first sector of the VTOC index on disk
10	VLFDRSV1	21	Reserved (set to blanks)
25	VLFDOWNR	14	Owner identification (set to blanks)
33	VLFDRSV2	28	Reserved (set to blanks)
4F	VLFDLSTV	1	Label standard version (set to blanks)
50	VLFDPADD	176	Padding (set to binary 0's)

Figure 2-69. Format of Volume Label for Disk

Error Recording Table Directory

This is a 256-byte area composed of a maximum of eighteen 14-byte entries which contain the disk address (SSD) of error logging information.

How to Find

To find the address of this area, subtract 1 from the SS value at displacement X'17' in the #LIBRARY format 1; the ERT directory is the sector before the configuration record. Also, you can subtract 1 from the value contained in the system communication area at NCODIR.

Format

Figure 2-70 shows the format of the error recording table directory, the format of an entry in this directory is shown in Figure 2-71.

256 bytes

Eighteen 14-byte entries (ERTDNT01-ERTDNT18) (Figure 2-71)	1-byte dummy ending entry (X'00')	Relative sector displacement of table entries (ERTSSN—3 bytes)

Figure 2-70. Error Recording Table Directory (ERTDIR) Format

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	ERTDDVQ	1	Q-byte
1	ERTDDVCD	1	UDT device code
2	ERTDSIOT	4	Start I/O count table: Bytes 1, 2—Sector address of SIO count table Byte 3—Length of SIO count table (sectors minus 1) Byte 4—Displacement within starting sector (will always be zero)
6	ERTDECT	4	Error count table (ECT): Bytes 1, 2—Sector address of ECT Byte 3—Length of the ECT (sectors minus 1) Byte 4—Displacement within starting sector

Figure 2-71 (Part 1 of 2). Format of Error Recording Table Directory Entry

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
A	ERTDEHT	4	Error history table (EHT): Bytes 1, 2—Sector address of EHT Byte 3—Length of the EHT (sectors minus 1) Byte 4—Displacement within starting sector

Figure 2-71 (Part 2 of 2). Format of Error Recording Table Directory Entry

Error Recording Tables

The IOS programs for each unit will call the error recording transient when an error is encountered. The error recording transient determines on which unit the error occurred, and calls in the appropriate error recording load module with its accompanying table.

How to Find

These tables are found immediately following the volume label. The following tables may be in this area:

I/O Counter
Disk (Figure 2-72)
Diskette (Figure 2-73)
Line printer (Figure 2-74)
Serial printer (Figure 2-75)
Keyboard (Figure 2-76)
Data recorder (Figure 2-77)
Magnetic character reader (Figure 2-78)

Error Counter
Disk (Figure 2-79)
Diskette (Figure 2-80)
Line printer (Figure 2-81)
Serial printer (Figure 2-82)
BSC (Figure 2-83)
Data recorder (Figure 2-84)
Magnetic character reader (Figure 2-85)
Mag card unit (Figure 2-86)
SDLC (Figure 2-87)

Error History
Disk (Figure 2-88)
Diskette (Figure 2-89)
Line printer (Figure 2-90)
Serial printer (Figure 2-91)
BSC (Figure 2-92)
Data recorder (Figure 2-93)
Mag card unit (Figure 2-94)
SDLC (Figure 2-95)
Magnetic character reader (Figure 2-96)

(SIO tables will always start on a sector boundary.)

Note: The last two bytes (X'FE'-X'FF') of the first sector (SIO table sector) for each device will contain a level identifier for ERAP (error recording analysis procedure).

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	FDSIO#1	4	Number of disk verifies
4	FDSIO#2	4	Number of disk writes
8	FDSIO#3	4	Number of disk reads or scans
C	FDSIO#4	4	Number of disk seeks (nonzero)

Figure 2-72. Disk I/O Counter Table (FDIOCTAB)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	I1SIO#1	4	Number of read data, read ID, read data/CAM, write ID, and write CAM SIOs
4	I1SIO#2	4	Number of write SIOs
8	I1SIO#3	4	Number of seek SIOs

Figure 2-73. Diskette I/O Counter Table (I1IOCTAB)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	PRSIO#1	4	Number of SIOs which have occurred

Figure 2-74. Line Printer I/O Counter Table (PRIOCTAB)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	BASIO#1	4	Number of SIOs that have occurred

Figure 2-75. Serial Printer I/O Counter Table (BAHSIOC)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	KBSNS#1	4	Total number of sense instructions to keyboard

Figure 2-76. Keyboard I/O Counter Table (KBIOCTAB)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	CDPUNSIO	4	Number of punch SIOs
4	CDRDSIO	4	Number of read SIOs

Figure 2-77. Data Recorder I/O Counter Table (NCCDSIO)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	MICRSIO	4	Count number of documents

● Figure 2-78. Magnetic Character Reader I/O Counter Table (MIIOCTAB)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	FDECTOTC	4	Temporary off-track check error counter (two bytes) and permanent off-track check error counter (two bytes)
4	FDECTCRC	4	Temporary cyclic redundancy check error counter (two bytes) and permanent cyclic redundancy check error counter (two bytes)
8	FDECTPPC	4	Temporary parallel parity check error counter (two bytes) and permanent parallel parity check error counter (two bytes)
C	FDECTCHO	4	Temporary channel overrun error counter (two bytes) and permanent channel overrun error counter (two bytes)
10	FDECTDUN	4	Temporary data unsafe error counter (two bytes) and permanent data unsafe error counter (two bytes)
14	FDCTNRF	4	Temporary no record found counter (two bytes) and permanent no record found counter (two bytes)
18	FDECTSK	4	Temporary invalid seek address error counter (two bytes) and permanent invalid seek address error counter (two bytes)
1C	FDECTSDC	4	Temporary SERDES check counter (two bytes) and permanent SERDES check counter (two bytes)
20	FDECTWEC	4	Temporary write echo check error counter (two bytes) and permanent write echo check error counter (two bytes)
24	FDECTSSC	4	Temporary sector sync check error counter (two bytes) and permanent sector sync check error counter (two bytes)
28	FDECTNR	4	Temporary disk not ready error counter (two bytes) and permanent disk not ready error counter (two bytes)
2C	FDECTSCK	4	Temporary seek error counter (two bytes) and permanent seek error counter (two bytes)

Figure 2-79 (Part 1 of 2). Disk Error Counter Table (FDECTAB)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
30	FDECTSUN	4	Temporary select unsafe error counter (two bytes) and permanent select unsafe error counter (two bytes)
34	FDECTWUN	4	Temporary write unsafe error counter (two bytes) and permanent write unsafe error counter (two bytes)
38	FDECTSVU	4	Temporary servo unsafe error counter (two bytes) and permanent servo unsafe error counter (two bytes)
3C	FDECTBRK	4	Temporary brake failure error counter (two bytes) and permanent brake failure error counter (two bytes)

Figure 2-79 (Part 2 of 2). Disk Error Counter Table (FDECTAB)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	I1EIDCRC	4	Temporary ID CRC error counter (two bytes) and permanent ID CRC error counter (two bytes)
4	I1EDCRC	4	Temporary data CRC error counter (two bytes) and permanent data CRC error counter (two bytes)
8	I1ECNTRL	4	Temporary control error counter (two bytes) and permanent control error counter (two bytes)
C	I1EFAST	4	Temporary diskette fast counter (two bytes) and permanent diskette fast counter (two bytes)
10	I1ENOORN	4	Temporary no orient counter (two bytes) and permanent no orient counter (two bytes)
14	I1ERDOVR	4	Temporary read overrun counter (two bytes) and permanent read overrun counter (two bytes)
18	I1EWTOVR	4	Temporary write overrun counter (two bytes) and permanent write overrun counter (two bytes)
1C	I1EWTPAR	4	Temporary write parity check counter (two bytes) and permanent write parity check counter (two bytes)

Figure 2-80. Diskette Error Counter Table (I1ECTAB)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	PRBELTSP	2	Belt speed check counter
2	PRCARIAB	2	Carriage sync check counter
4	PRCCURNT	2	Coil current check counter
6	PRBELTSY	2	Belt sync check counter
8	PRFRMJAM	2	Forms jam check counter
A	PRCIOLC1	2	Coil 1 parity odd counter
C	PRCIOLC2	2	Coil 2 parity odd counter
E	PRCIOLC3	2	Coil 3 parity odd counter
10	PRDATACH	2	Buffer data check counter
12	PRHAMMER	2	Hammer parity check counter
14	PREMITER	2	Emitter check counter

Figure 2-81. Line Printer Error Counter Table (PRECTAB)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	BAHORSYN	2	Horizontal sync check
2	BAWIRE	2	Wire check
4	BASTORPA	2	Storage parity check
6	BAEMMCK	2	Emitter check
8	Unassigned	4	–
C	BAUNPRIN	2	Unprintable character check
E	BAFORMHG	2	Forms hung
10	BAFORMRA	2	Forms runaway

Figure 2-82. Serial Printer Error Counter Table (BAHECT)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	BSLTBTFJ	2	Text blocks transmitted for this job
2	BSLTTSLE	4	Text blocks transmitted since last error
6	BSLTBRFJ	2	Text blocks received for this job
8	BSLTRSLE	4	Text blocks received since last error
C	BSLNAKFJ	2	NAKs received for this job
E	BSLNAKRC	4	NAKs received since last error
12	BSLDCKFJ	2	Data checks received for this job
14	BSLDATCC	4	Data checks received since last error
18	BSLFARFJ	2	Forward aborts received for this job
1A	BSLFAREC	4	Forward aborts received since last error
1E	BSLABTFJ	2	Aborts received for this job
20	BSLABORC	4	Aborts received since last error
24	BSLACTFJ	2	Adapter checks during transmission for this job
26	BSLACTKC	4	Adapter checks during transmission since last error
2A	BSLACRFT	2	Adapter checks during receive for this job
2C	BSLADCRC	4	Adapter checks during receive since last error
30	BSLIRREJ	2	Invalid responses for this job
32	BSLINRRC	4	Invalid responses since last error
36	BSLEAAFJ	2	ENQs received as ACK responses for this job
38	BSLEAACC	4	ENQs received as ACK responses since last error
3C	BSLLDEFJ	2	Lost data errors for this job
3E	BSLLDERC	4	Lost data errors since last error

Figure 2-83 (Part 1 of 2). BSC Error Counter and SIO Counter Table

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
42	BSLDTOFJ	2	Lost connections for this job
44	BSLDTOUC	4	Lost connections since last error
48	BSLRTOFJ	2	Receive timeouts for this job
4A	BSLRTOUC	4	Receive timeouts since last error
4E	BSLTWTFJ	2	Transmit timeouts while transmitting text for this job
50	BSLTWTRC	4	Transmit timeouts while transmitting text since last error

Figure 2-83 (Part 2 of 2). BSC Error Counter and SIO Counter Table

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	CIECOFF	2	5496/129 offline
2	CIECTJAM	2	Transport jam
4	CIECSTHP	2	Stacker full, hopper empty, hopper jam
6	CIECTOCK	2	Timeout check
8	CIECINCC	2	Incorrect card code
A	CIECCERR	2	Compare error

Figure 2-84. Data Recorder Error Counter Table

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	MIECMWOR	2	Misreads without rejects
2	MIECMWR	2	Misreads with rejects
4	MIECAURJ	2	Auto reject
6	MIECDDPC	2	DBI/DBO parity check
8	MIECAMPC	2	Attachment controller memory parity check
A	MIECLTOC	2	Long timeout check

● **Figure 2-85. Magnetic Character Reader Error Counter Table**

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	MCWRITES	4	Number of writes
4	MCREADS	4	Number of reads
8	MCFEEDS	4	Number of feeds
C	MCTKSTEP	4	Number of track steps
10	MCHOMES	4	Number of homes
14	MCEJECTS	4	Number of card ejects
18	MCEXITS	4	Number of exits
1C	MCSTACKS	4	Number of stacks
20	MCDNFT	2	Number of temporary data not found errors
22	MCDNFP	2	Number of permanent data not found errors
24	MCFERRT	2	Number of temporary format errors
26	MCFERRP	2	Number of permanent format errors
28	MCOVRNT	2	Number of temporary data overruns
2A	MCOVRNP	2	Number of permanent data overruns
2C	MCJAMT	2	Number of temporary jams
2E	MCJAMP	2	Number of permanent jams
30	MCDPET	2	Number of temporary device parity errors
32	MCDPEP	2	Number of permanent device parity errors
34	MCITIMOT	2	Number of temporary invalid timeout errors
36	MCITOMOP	2	Number of permanent invalid timeout errors

Figure 2-86. MCU Error Counter and SIO Counter Table

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	SDIFTJ	2	I-frames transmitted for this job
2	SDIFTLE	4	I-frames transmitted (cumulative)
6	SDIFRJ	2	I-frames received for this job
8	SDIFRLE	4	I-frames received (cumulative)
C	SDTFTJ	2	Total frames transmitted for this job
E	SDTFTLE	4	Total frames transmitted (cumulative)
12	SDTFRJ	2	Total frames received for this job
14	SDTFRLE	4	Total frames received (cumulative)
18	SDBCEJ	2	Block check errors this job
1A	SDBCEC	4	Block check errors (cumulative)
1E	SDIFEJ	2	Invalid frame errors for this job
20	SDIFEC	4	Invalid frame errors (cumulative)
24	SDADTJ	2	Abortive disconnect timeouts for this job
26	SDADTC	4	Abortive disconnect timeouts (cumulative)
2A	SDRTJ	2	Receive timeouts for this job
2C	SDRTC	4	Receive timeouts (cumulative)
30	SDACJT	2	Adapter checks for this job
32	SDACCT	4	Adapter checks (cumulative)

Figure 2-87. SDLC Error Counter and SIO Counter Table

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	FDHTDATE	2	Date that error occurred
2	FDHTQ	1	Q-byte of actual operation
3	FDHTR	1	R-byte of actual operation
4	FDHTSNS1	2	Sense bytes 0 and 1
6	FDHTSNS2	2	Sense bytes 2 and 3
8	FDHTCNT	1	Retry count
9	FDHTPREV	2	Cylinder location before seek
B	FDHTTARG	2	Target cylinder to seek
D	FDHTIDCY	2	ID of cylinder actually arrived at
F	FDHTCF	6	Control field (NFCCHS)
15	FDHTSIO1	4	Number of times read verify occurred
19	FDHTSIO2	4	Number of times write data occurred
1D	FDHTSIO3	4	Number of times read data/scan occurred
21	FDHTSIO4	4	Number of times seek (nonzero) occurred

Figure 2-88. Disk Error History Table

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	I1HTDATE	2	Date that error occurred
2	I1HTQ	1	Q-byte of actual operation
3	I1HTR	1	R-byte of actual operation
4	I1HTSNS1	2	Sense bytes 0 and 1
6	I1HTSNS2	2	Sense bytes 2 and 3
8	I1HTCNT	1	Retry count
9	I1HTPREV	1	Cylinder location before seek (if X'F0', invalid for use by ERAP)
A	I1HTIDCY	1	Starting cylinder
B	I1HTCHRN	4	ID of cylinder arrived at (CHRN: C = cylinder, H = head, R = record, and N = sector size)
F	Reserved	2	Reserved
11	I1HTSIO1	3	Number of times read data, read ID, read data/CAM, write ID, and write CAM SIO occurred
14	I1HTSIO2	3	Number of times write SIO occurred
17	I1HTSIO3	3	Number of times seek SIO occurred
1A	I1HTVOL	6	Volume ID

Figure 2-89. Diskette Error History Table

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	PRHTDATE	2	Date that error occurred
2	PRHTQ	1	Q-byte of SIO
3	PRHTR	1	R-byte of SIO
4	PRHTSNS	2	Sense bytes 0 and 1
6	PRHTSIO	3	Number of SIOs which have occurred since installation

Figure 2-90. Line Printer Error History Table

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	BAHTDATE	2	Date that error occurred
2	BAHTQ	1	Q-byte of SIO
3	BAHTR	1	R-byte of SIO
4	BAHTSNS	4	Sense bytes 0 to 3
8	BAHTSIO	3	Number of SIOs that have occurred since installation

Figure 2-91. Serial Printer Error History Table

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	BSLEDATE	2	Date error occurred
2	BSLQBYTE	1	Q-byte of SIO instruction
3	BSLRBYTE	1	R-byte of SIO instruction
4	BSLSENSE	2	Sense byte of SIO instruction
6	BSLERCNT	1	Error retry count
7	BSLBSCCC	1	BSC completion code
8	BSLTRANS	4	Number of text blocks transmitted since last error
C	BSLBLRCV	4	Number of text blocks received since last error
10	BSLTERAD	2	Terminal address
12	BSLRESVD	2	Reserved

Figure 2-92. BSC Error History Table

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	CIHTDATE	2	Date error occurred
2	CIHTQ	1	Q-byte of SIO
3	CIHTR	1	R-byte of SIO
4	CIHTSNS	2	Sense bytes 0 and 1
6	CPHTSIO	3	Number of punch SIOs
9	CRHTSIO	3	Number of read SIOs

Figure 2-93. Data Recorder Error History Table

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	MCHTDATE	2	Date error occurred
2	MCHTCOMP	1	Completion code
3	MCHTCMD	1	Command
4	MCHTCMOD	1	Command modifier
5	MCHTBUF@	2	Read/write buffer address
7	MCHTSNS0	1	Sense byte 0
8	MCHTSNS1	1	Sense byte 1
9	MCHTCCNT	1	Character count
A	MCHTFLAG	1	Flag byte
B	MCHTINDA	1	Indicator/alarm byte
C	MCHTRETC	1	Retry count
D	MCRSVD	2	Reserved

Figure 2-94. MCU Error History Table

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	SDHTDATE	2	Date error occurred
2	SDHTQ	1	Q-byte of SIO
3	SDHTSNS	2	Sense information
5	SDHTCONT	1	SDLC control field
6	SDHTADDR	1	SDLC station address field
7	SDHTIFX	4	I-frames transmitted
B	SDHTIFR	4	I-frames received
F	SDHTTA	1	Terminal address

Figure 2-95. SDLC Error History Table

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	MIHTDATE	2	Date error occurred
2	MIHTQ	1	Q-byte
3	MIHTR	1	R-byte
4	MIHTSNS0	1	Sense byte 0
5	MIHTSNS	1	Sense byte 1
6	MIHTSIO	4	Number of documents

● **Figure 2-96. Magnetic Character Reader Error History Table**

LIBRARY DIRECTORY AREA

This area contains:

- System configuration record (Figure 2-97)
- Library control sector (Figure 2-98)
- Library directory (Figure 2-99)

System Configuration Record

The configuration record (one sector) contains information about the system configuration of a particular IBM System/32 system.

How to Find

The configuration record is pointed to by the entry at displacement X'17' in the disk format 1 for #LIBRARY (Figure 2-64). The configuration record is always located in the first sector of the directory and is also pointed to by NCODIR in the system communication area.

Format

Figure 2-97 shows the contents of the configuration record.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	CONDFRMT	1	System date format: X'04' = World Trade (DDMMYY) X'02' = Domestic (MMDDYY) X'01' = International (YYMMDD)
1	CONDDATE	6	System date (format defined by byte 0)
7	CONDLINE	1	Number of lines on forms
8	CONDCHAR	1	Character set size: X'30' = 48 characters X'40' = 64 characters X'60' = 96 characters X'C0' = 192 characters
9	CONDINQ	2	Inquiry bytes: *Byte 0:* Reserved *Byte 1:* X'80' = Inquiry supported

Figure 2-97 (Part 1 of 2). System Configuration Record

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
B	CONDBSCA	10	BSCA bytes (for a description of these bytes, see PLCA, displacements X'1B' through X'24', Figure 2-10)
15	CONDRELL	2	Release level
17	CONDMODL	1	Modification level
18	CONDDEVC	1	Disk capacity: X'01' = 3.2 megabytes X'02' = 5.0 megabytes X'04' = 9.1 megabytes X'08' = 13.7 megabytes
19	CONDMSIZ	1	Main storage size: X'40' = 16K bytes X'60' = 24K bytes X'80' = 32K bytes
1A	CONDCSIZ	1	Control storage size: X'10' = 4K words X'20' = 8K words
1B	CONDSDLC	10	SNA/SDLC bytes (for a description of these bytes, see PLCA, displacements X'49' through X'52', Figure 2-10)
25	CONDWPID	1	Word processing indicator: X'80' = Magnetic card unit attached X'40' = Half space feature X'20' = Upper/lower case feature X'10' = Data recorder X'08' = Magnetic character reader X'04' = No task command delimiter
26	CONDKBDF	1	Console keyboard identification
27	CONDMCDF	1	Default mag card identification
28	CONDBTID	1	Printer belt identification
29		23	Reserved
40	CONDIMAG	192	Print image

Figure 2-97 (Part 2 of 2). System Configuration Record

Library Control Sector

This 1-sector area contains information concerning the library and directory.

How to Find

The library control sector is always in the second sector of the library directory area, and its address can be found at entry NCODIR (in the system communication area) plus 1; the address of this area can also be found by adding 1 to the sector address contained at displacement X'17' in the disk format 1 for #LIBRARY (Figure 2-64).

Format

The format of this area is shown in Figure 2-98.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	LCSWCHS	1	Library control sector switches: X'80' = #OXRF should be run X'40' = Compactor ($MACMP) should be run X'08' = Reload was last job run
1	LCSSOLB	2	Sector address of start of library
3	LCSEOLB	2	Sector address of end of library
5	LCSSODR	2	Sector address of start of directory
7	LCSEODR	2	Sector address of end of directory
9	LCS#ACDR	2	Number of active directory entries
B	LCS#AVDR	2	Number of available directory entries
D	LCSSSDLE	3	Sector address (two bytes)/displacement (one byte) of last active directory entry
10	LCSSOSW	2	Sector address of start of scheduler work area (SWA)

Figure 2-98 (Part 1 of 2). Format of Library Control Sector

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
12	LCSEOSW	2	Sector address of end of scheduler work area
14	LCSSORR	2	Sector address of start of rollout/rollin area
16	LCSEORR	2	Sector address of end of rollout/rollin area
18	LCSSOMB	2	Sector address of start of library members area
1A	LCSEOMB	2	Sector address of end of library members area
1C	LCS#ACMB	2	Number of active library sectors
1E	LCSNXMB	2	Sector address of next sector available for library members
20	LCS#AVMB	2	Number of sectors available for library members
22	LCSRSVSZ	2	Reserved area size in sectors
24	LCSSCPSZ	2	Base SCP size in sectors
26	LCSBASE	2	Reserved area plus SCP base (sum of LCSRSVSZ and LCSSCPSZ)
28	LCSASTK@	2	Address on next library directory stack entry
2A	LCSNUMDE	1	Number of directory entries in the stack
2B	—	20	Reserved
3F	LCSCHLNG	1	Length of current print image
40	LCSIMAGE	192	Current print image

Figure 2-98 (Part 2 of 2). Format of Library Control Sector

Library Directory

The directory is a variable length area which contains one 23-byte entry for each member in the library. Members are grouped in ascending alphabetical order by name within each member type (load, procedure, subroutine, and source members). There can be 11 entries per sector (110 entries per block), starting at the beginning of each sector. The directory area is initially filled with X'FF's.

How to Find

The address of the library directory can be found by examining the *System Information* evoked by the LISTLIBR command. The address can also be found by adding 2 to the value at INCODIR in the system communication area or by adding 2 to the sector address entry found at displacement X'17' in the disk format 1 for #LIBRARY (Figure 2-64).

Format

The format of a library directory entry is shown in Figure 2-99.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	DIRTYPE	1	Library type: C'O' = Load module C'R' = Subroutine module C'P' = Procedure module C'S' = Source module
1	DIRNAME	8	Module name
9	DIRADDR	2	Relative sector address—number that when added to start address of library member sectors equals actual sector address of module
B	DIR#TXT	1	For load modules—number of text sectors (length of module)
	or		
B	DIRRECL	1	For source and procedure modules—record size (40 to 120 bytes)
The next 6 bytes are used by load modules only:			
C	DIRLINK	2	Storage address to which entry has been link-edited (start address)
E	DIRRLD	1	Displacement of first relocation directory (RLD) in first sector containing RLDs in the module

Figure 2-99 (Part 1 of 2). Format of Library Directory Entry

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
F	DIRSCA	2	Entry point of module
11	DIRCORE	1	Amount of storage (sectors) needed to run the program
12	DIRATTR	2	Attributes of member: *Byte 0:* X'80' = System control program module; this bit is used to prevent an SCP module from being deleted from the library X'08' = Program requires that $WORK and $SOURCE be allocated; $SOURCE must be filled from the keyboard, a library source member, or an inline source from a procedure (queued job stream) X'04' = Optional SCP module X'02' = A program temporary fix (PTF) has been applied to this program X'01' = Load module containing overlays *Byte 1:* X'20' = Program can have a // COMPILE statement with byte 0 above set to X'08'; the program will access the source itself X'10' = Program requires that $WORK2 be allocated X'08' = This module's keyword/verb tables, if any, have been translated by SCP translation utility X'04' = Program requires that a new load address be calculated at load time to ensure that the new load address is placed in main storage at a point beyond its own common region X'02' = Program reads utility control statements X'01' = Program contains a where-to-go table for #OXRF
14	DIRREL	1	Release level of this entry
15	DIRTOTL	2	Total number of sectors in this module

Figure 2-99 (Part 2 of 2). Format of Library Directory Entry

Rollout/Rollin Area

This area contains information needed to run inquiry, and is assigned at installation or reload time, if inquiry/offline is requested.

How to Find

The sector address of this area can be found by examining the *System Information* evoked by the LISTLIBR command. This area is pointed to by the sector address in field NCROLLSS in the system communication area, or by the value in field LCSSORR in the library control sector.

Format

The format of this area is shown in Figure 2-100.

Sectors (in hex)	Description
0–2C	Save area for the first 45 sectors of the scheduler work area (see Figure 2-101 for a description of these sectors)
2D	SWA sysin duplicate sector
2E	SWA display screen buffer
2F–31	Rollout/rollin work area:

Byte(s)	Contents
0-1	ARR
2-3	XR1
4-5	XR2
6	Line position
7	Form size
8-9	Size of image character set
A–B	Reserved
C	User program N byte (size)
D–E	User program sector address
F-10	User program relocation
11–12	Printer data address register
13–3F	Reserved
40–FF	Print image save area
0100–01FF	Figure 256 bytes of main storage (Figure 2-7)
0200–02DF	Display screen image

Sectors (in hex)	Description
32–69	User program (2K–16K)
6A–89	User program (16K–24K)
8A–A9	User program (24K–32K)

Figure 2-100. Format of Rollout/Rollin Area

Scheduler Work Area (SWA)

The SWA is a 170-sector area on the disk, reserved for use by the system. Functions performed are explained in this section, as shown in Figure 2-101.

How to Find

The sector address of the SWA can be found by examining field LCSSOSW in the library control sector (Figure 2-98). The sector address is in the system communication area (SCA) at label NCSWRK.

Format

The SWA is set up as shown in Figure 2-101.

Sector(s) (in hex)	Description
0	SWA index-common sector (Figure 2-102)
1-3	SWA format 5 (Figure 2-103)
4-B	Procedure parameter save area (Figure 2-104)
C-18	SWA format 1's (Figures 2-105 and 2-106)
19-2C	Initiator work area (Figure 2-107)
2D-53	History file (Figure 2-108)
54-A6	Push/pull area
A7	Scheduler display screen save area
A8	Inquiry work area
A9	Sysin work area

Figure 2-101. SWA Organization

Scheduler Work Area (SWA) Index-Common Sector

This 256-byte area contains begin-, end-, and last-sector-accessed information for each area within the SWA, device allocate information, and scheduler module loader information. This information is used by the SWA access main storage transients.

How to Find

The SWA index-common sector is the first sector of the scheduler work area, and is pointed to by NCSWRK in the system communication area; the sector address of this area can also be found by examining the *Start Sector of Scheduler Work Area* that is found in the *System Information* evoked by the LISTLIBR command.

Format

The format of this area is shown in Figure 2-102.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	SWAPP1SC	2	Sector address (SS) of the first sector that contains procedure parameters
2	SWAPPLSC	2	Sector address of the last sector that contains procedure parameters
4	SWAPPCSC	2	Sector address of current sector that contains procedure parameters
6	SWAPPDCS	1	Displacement of procedure parameter in current sector
7	SWAPPNSC	2	Sector address of next sector that will contain procedure parameters
9	SWAPPDNS	1	Displacement of next procedure parameter in next sector
A	SWAF11SC	2	Sector address of the first sector that contains format 1 records
C	SWAF1LSC	2	Sector address of the last sector that contains format 1 records
E	SWAF1LRD	2	Sector address of the last sector of format 1's that was read from
10	SWAF1DRD	1	Rightmost displacement into the last sector of format 1's from which information was read
11	SWAF1LWT	2	Sector address of the last sector of format 1's that was written into
13	SWAF1DWT	1	Rightmost displacement into the last sector of format 1's that was written into
14	SWAIN1SC	2	Sector address of the first sector containing the initiator work area

Figure 2-102 (Part 1 of 4). Format of Scheduler Work Area Index-Common Sector

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
16	SWAINLSC	2	Sector address of the last sector containing the initiator work area
18	SWAINLRD	2	Sector address of the last sector in the initiator work area that was read from
1A	SWAINDRD	1	Displacement into the last sector in the initiator work area from which information was read
1B	SWAINLWT	2	Sector address of the last sector in the initiator work area that was written into
1D	SWAINDWT	1	Displacement into the last sector in the initiator work area that was written into
1E	SWAHFSTR	2	Sector address of the first sector containing the history file
20	SWAHFOLD	2	Sector/displacement of leftmost byte of the oldest entry relative to the start of the history file
22	SWAHFNEW	2	Sector/displacement of rightmost byte of the newest entry relative to the start of the history file
24	SWAHFRSV	4	Reserved
28	SWAPU1SC	2	Sector address of first sector containing page area
2A	SWAPULSC	2	Sector address of last sector containing page area
2C	SWASINSC	2	Sector address of sysin duplication save area
2E	SWAUNUSD	1	Unused
2F	SWADIRET	12	Directory entry of program to be loaded
3B	SWACOMNM	8	Name of source given on compile statement
43	SWAF1NSS	2	Saved sector address of the last format 1 sector that was written into
45	SWAF1NSD	1	Saved displacement into the last format 1 sector that was written into
46	Reserved	1	Reserved

Figure 2-102 (Part 2 of 4). Format of Scheduler Work Area Index-Common Sector

Displ of Leftmost Byte in Hex	Lng in Bytes in Dec	Description
47	113	Scheduler module loader information (filled in by #OXRF routine at IPL after reload)—twenty-eight 4-byte entries each in the following format:

A	B	C

A = Sector address of module (2 bytes)
B = Number of text sectors in module (1 byte)
C = RLD displacement (1 byte)

The last byte of the table is the table delimiter byte (X'FF')

Label	Description
TME3	EOJ reinitialization phase
INPO	Program setup phase 1
INP1	Program setup phase 2
INDO	DFI (disk file initialization) phase 1
IND1	DFI phase 2
IND3	DFI phase 3
IND2	DFI phase 3
INF0	Allocation phase 0
INF1	Allocation phase 1
INF2	Allocation phase 2
INF3	Allocation phase 3
INF4	Reserved for system compatibility
TME1	EOJ deallocation phase 1
STFT	Allocation terminate transient
TME2	EOJ deallocation phase 2
SSRW	SWA read/write
SSVF	Disk VTOC read/write
SSVI	Diskette VTOC read/write
RDML	R/I mainline
RDFL	R/I file statement
RDRT	R/I root phase
RDMK	R/I keyword merge
INMS	Disk file allocate support
TMSK	Key sort phase 1
TMDS	Key sort phase 2
STF1	File allocate initiate
STF7	Reserved for system compatibility
TMST	Reserved for system compatibility

B8	8	Reserved

Figure 2-102 (Part 3 of 4). Format of Scheduler Work Area Index-Common Sector

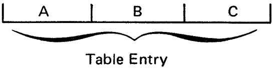
Displ of Leftmost Byte in Hex	Lng in Bytes in Dec	Description
C0	31	Device allocate table—ten 3-byte table entries, each in the following format, plus one table end indicator byte (X'FF'):  A = One byte device Q-byte B = Device status: X'80' = Device available X'40' = Device allocated X'3C' = Reserved X'02' = Interaction with other devices for microprocessor transient X'01' = Multiple allocation allowed C = Error code (value added to 1320 to get error MIC for this table entry)
DF	33	Reserved

Figure 2-102 (Part 4 of 4). Format of Scheduler Work Area Index-Common Sector

Scheduler Work Area (SWA) Format 5

This 768-byte area indicates which blocks of disk space have been allocated to new files for the current job.

How to Find

To find the sector address of the SWA format 5, add 1 to the value at NCSWRK in the system communication area, the sector address of this area can also be found by adding 1 to the *Start of Sector of Scheduler Area* entry found in the *System Information* evoked by the LISTLIBR command.

Format

The format of the SWA format 5 is shown in Figure 2-103.

Each bit in bytes X'00' through X'FB' of the rightmost sector and all bits of the middle and leftmost sectors indicate whether a specific block (10-sector area) of disk has been allocated for the current job. In the rightmost sector, for example, bit 0 of byte X'04' indicates the status of block 1983, while bit 7 of byte X'FB' in the rightmost sector indicates the status of block 0. If a bit is on, the related block has been allocated.

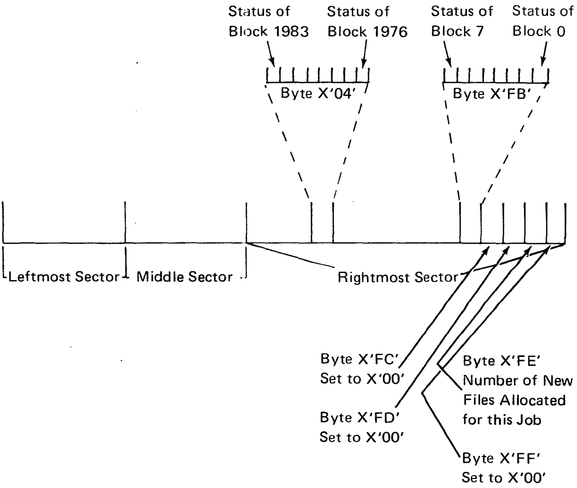


Figure 2-103. Format of SWA Format 5

Procedure Parameter Save Area

This 8-sector area contains parameters specified by the // INCLUDE statement. Each parameter is described by one entry in this area. Entries (Figure 2-107) are 115 bytes long; two entries are stored in each sector, and the last 13 bytes after each entry are unused.

How to Find

The entry in the SWA index-common sector at SWAPP1SC contains the address of the first entry in the procedure parameter save area.

Format

Figure 2-104 shows the format of an entry in this area.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	PPSPRG1M	2	Sector address of program 1 message member save area
2	PPSPRG2M	2	Sector address of program 2 message member save area
4	PPSUSR1M	2	Sector address of user 1 message member save area
6	PPSUSR2M	2	Sector address of user 2 message member save area
8	PPSSGED@	2	End address of library member being accessed by source get
A	PPSSGCRT	2	Address of current record being read by source get
C	PPSSGDSP	1	Displacement within address of current sector being read by source get
D	PPSSGLNR	1	Length of record-source get
E	PPSUPSI	1	UPSI switches
F	PPSVR01L	1	Length of variable 1
10	PPSVAR01	8	Variable 1
18	PPSVR02L	1	Length of variable 2
19	PPSVAR02	8	Variable 2
21	PPSVR03L	1	Length of variable 3
22	PPSVAR03	8	Variable 3
2A	PPSVR04L	1	Length of variable 4
2B	PPSVAR04	8	Variable 4
33	PPSVR05L	1	Length of variable 5
34	PPSVAR05	8	Variable 5
3C	PPSVR06L	1	Length of variable 6
3D	PPSVAR06	8	Variable 6
45	PPASVR07L	1	Length of variable 7

Figure 2-104 (Part 1 of 2). Format of Procedure Parameter Save Area Entry

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
46	PPSVAR07	8	Variable 7
4E	PPSVAR08L	1	Length of variable 8
4F	PPSVAR08	8	Variable 8
57	PPSVAR09L	1	Length of variable 9
58	PPSVAR09	8	Variable 9
60	PPSVAR10L	1	Length of variable 10
61	PPSVAR10	8	Variable 10
69	PPSVAR11L	1	Length of variable 11
6A	PPSVAR11	8	Variable 11
72	PPSCPROC	1	SCP control byte: X'80' = SCP procedure

Figure 2-104 (Part 2 of 2). Format of Procedure Parameter Save Area Entry

Scheduler Work Area Format 1's

The SWA format 1 area is comprised of a maximum of fifty-two 64-byte entries. These entries can be either converted 64-byte disk format 1's (Figure 2-64) or converted 128-byte diskette data set (HDR1) labels (Figure 2-112). The SWA format 1's are used by the scheduler to process files being accessed by the program currently running on the system.

How to Find

The index sector of the SWA is pointed to by NCSWRK in the system communication area; SWAF11SC in the SWA index-common sector points to the first sector of format 1's.

Format

The format of each SWA format 1 for disks is shown in Figure 2-105. The format of each SWA format 1 for diskettes is shown in Figure 2-106.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	F1TAG	1	Tag identifier of VTOC index entry (set to 0 for new files)
1	F1CHAN	2	Chain address (sector/displacement of next VTOC format 1 in date chain) for date-differentiated files with the same file label; for nondate-chained files, set to hex zeros
3	F1LABL	8	File label
B	F1DATE	6	Date the file was created (YYMMDD)
11	F1RTIN	1	Retain code: C'P' = Permanent C'S' = Scratch C'T' = Temporary
12	F1TYPE	2	File type: *Byte 0:* Current status (status when file was opened): X'80' = Indexed X'60' = Pseudo tape X'40' = Consecutive X'20' = Direct X'10' = Offline multivolume file X'08' = Input X'04' = Output X'02' = Update X'01' = Add *Byte 1:* Status when file created: X'80' = Indexed X'60' = Pseudo tape X'40' = Consecutive X'20' = Direct X'10' = Reserved Status when file closed: X'08' = Reserved X'04' = Sequential add X'02' = Random add X'01' = Unordered load
14	F1RECL	2	Logical record length
16	F1KEYL	1	Key length
17	F1KEYO	2	Key location (S/D)
19	F1LSTR	3	Next available record (SS/D)
1C	F1LSTK	3	Next available key (SS/D)

Figure 2-105 (Part 1 of 3). Format of Scheduler Work Area Format 1 (Converted Disk Format 1)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
1F	F1STDA	2	Sector address of start of data for an index file; for non-indexed files this is also the physical begin extent
21	F1ENDA	2	Sector address of end of data; this is the physical end extent (next available sector)
23	F1STIX	2	Sector address of start of the file; this is the physical begin extent (first sector of file); for indexed files this is also the beginning of the index
25	F1ENIX	2	Sector address of end of index (first sector past index for indexed files)
27	F1RECN	3	Number of records allocated when file created (if leftmost bit of field is 0)
	or		
27	F1BLKN	3	Number of blocks allocated (if leftmost bit of field is 1)
2A	F1RSV1	1	Reserved
2B	F1BACK	2	Reserved (set to hex zeros)
2D	F1RSV2	3	Reserved
30	F1OLSN	1	Offline sequence number (disk data management only)
31	F1NAME	8	Filename
39	F1UNIT	1	Device address for disk
3A	F1ATT1	1	Attribute byte 1: X'80' = Set on for system compatibility X'40' = File processed X'20' = Reserved X'10' = Offline multivolume file (restricted usage on disk) X'08' = New file X'04' = Location given X'02' = Space and location equal X'01' = Two labels are the same

Figure 2-105 (Part 2 of 3). Format of Scheduler Work Area Format 1 (Converted Disk Format 1)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
3B	F1ATT2	1	Attribute byte 2: X'40' = RETAIN parameter was not specified; default to T for new disk file X'20' = Remove format 1 from the VTOC at end of job X'10' = File open X'08' = File closed X'04' = Processed by special EOJ X'02' = DATE parameter specified on file statement X'01' = File allocated (output)
3C	F1ATT3	1	Attribute byte 3: X'80' = Disk used X'40' = Pending 1350 MIC error; old file not on disk VTOC X'10' = File statement was given for special allocate DTF X'08' = This is a load to an existing file X'02' = DTF supplied for this file X'01' = File allocated by special allocate DTF
3D	F1INDX	1	DTF SWA index (relative format 1 number in SWA)
3E	F1UPSI	1	Save area for open—UPSI
3F	F1BYTE	1	Save area for open—non-UPSI

Figure 2-105 (Part 3 of 3). Format of Scheduler Work Area Format 1 (Converted Disk Format 1)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	IF1DTAG	1	Sector address of VTOC entry
1	IF1DBPND	1	Bypass indicator
2	IF1DSORG	1	Data set organization indicator
3	IF1DLABL	8	Data set identifier (file label)
B	IF1DDATE	6	Creation date

Figure 2-106 (Part 1 of 3). Format of Scheduler Work Area Format 1 (Converted Diskette Data Set Label—HDR1)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
11	IF1DVFMK	1	Verify mark
12	IF1DTYPE	2	File type
14	IF1DRECL	2	Logical record length (equal to block length in HDR1)
16	IF1DRTPT	3	Retention period (number of days file is to be retained)
19	IF1DSSLR	2	Sector address of end of data
1B	IF1DLSTR	1	Negative displacement of end of data within sector (bytes available in sector)
1C	IF1DVFMK	1	Verify mark
1D	IF1DWTPT	1	Write protect indicator: C'P' = Protected Blank = Not protected
1E	IF1DINTL	1	Interchange level indicator: C'E' = System file Blank = Basic exchange file
1F	IF1DSTDA	2	Sector address of beginning of extent
21	IF1DENDA	2	Sector address of end of extent
23	IF1DEXDT	6	Expiration date
29	IF1DMVID	1	Multivolume data set indicator: Blank = Completely contained on this volume C'C' = Continued on another volume C'L' = Last volume of a multivolume data set
2A	IF1DMVNM	1	Multivolume sequence number
2B	IF1DPACK	6	Volume ID (PACK parameter on file statement)
31	IF1DNAME	8	Filename
39	IF1DUNIT	1	Diskette device address

Figure 2-106 (Part 2 of 3). Format of Scheduler Work Area Format 1 (Converted Diskette Data Set Label—HDR1)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
3A	IF1DATT1	1	Attribute byte 1: X'80' = Must be set on for system compatibility X'20' = File may contain spanning records X'08' = New file X'04' = Extended format diskette
3B	IF1DATT2	1	Attribute byte 2: X'80' = Rewrite updated diskette header X'10' = File open X'08' = File closed X'02' = DATE parameter specified on file statement X'01' = File allocated
3C	IF1DATT3	1	Attribute byte 3: X'20' = Diskette drive used X'04' = End of extent on diskette reached X'02' = DTF was supplied for this file
3D	IF1DINDX	1	DTF SWA index (relative format 1 number in SWA)
3E	IF1DBYT1	1	Save area
3F	IF1DBYT2	1	Save area

Figure 2-106 (Part 3 of 3). Format of Scheduler Work Area Format 1 (Converted Diskette Data Set Label—HDR1)

Initiator Work Area

The initiator work area is a 20-sector area within the SWA, built by the reader/interpreter phases. This area is composed of one coded statement for each // FILE statement read; a maximum of 52 such statements are allowed for this area.

How to Find

The index sector of the SWA is pointed to by NCSWRK in the system communication area; SNAIN1SC in the SWA index-common sector points to the first sector of the initiator work area.

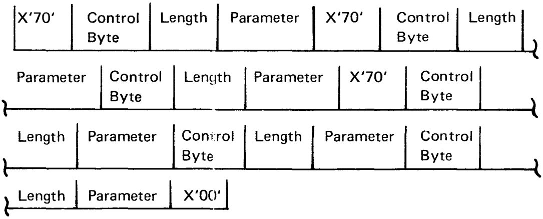
Format

The format of each coded // FILE statement is shown in Figure 2-107.

Displ of Leftmost Byte in Hex	Lng in Bytes in Dec	Description
0	1	Delimiter byte (X'70')
1	1	Control byte: *Bits 0-4--Keyword identifier:* X'90' = PACK parameter X'80' = BLOCKS parameter X'70' = RETAIN parameter X'60' = LOCATION parameter X'50' = RECORDS parameter X'40' = DATE parameter X'30' = UNIT parameter X'20' = LABEL parameter X'10' = NAME parameter *Bits 5-7--Parameter type:* X'04' = Null parameter X'02' = Sub parameter X'01' = Regular parameter
2	1	Length of parameter in next field
3	1-8	The parameter specified; length of field is equal to value in previous field

Note: End of table is denoted by X'00'.

The type of information specified in bytes 1, 2, and 3 is included for each parameter in the // FILE statement. Then, when all parameters in the statement have been described, a delimiter byte is coded (X'70', if another // FILE statement was read but not described; X'00', if all // FILE statements read have been described). If three file statements were read having 1, 2, and 3 parameters respectively, the initiator work area would be formatted as follows:



X'70' | Control Byte | Length | Parameter | X'70' | Control Byte | Length

Parameter | Control Byte | Length | Parameter | X'70' | Control Byte

Length | Parameter | Control Byte | Length | Parameter | Control Byte

Length | Parameter | X'00'

Figure 2-107. Initiator Work Area Encoded File Statement Format

History File

This 39-sector area (within the SWA) contains variable length entries which document both what was displayed on the display screen and what was generated by the source sysin, keyboard sysin, and syslog routines. This area has a wraparound structure, with each new entry being added to the right of the last entry.

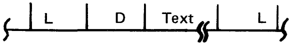
The text length of each entry may vary, but may not exceed 40 bytes. If the text portion of an entry is longer than 40 bytes, it will be broken into segments. For example, if a 50-byte entry is to be put in the file, it is segmented into a 40-byte piece and a 10-byte piece, each of which becomes an entry.

How to Find

The sector address of the first sector of the SWA containing the history file area is in the SWA index-common sector at SWAHFSTR; the relative address of the oldest entry in the history file is found in the SWA index-common sector at SWAHFOLD; the relative address of the newest entry in the history file is found in the SWA index-common sector at SWAHFNEW.

Format

Figure 2-108 shows the format for a history file record entry.



where:

L = One-byte record length entry (length of text + 3 bytes of control information); an identical length entry is found at both ends of the history file entry.

D = Display indicator (one byte) tells whether this record has been previously displayed to the user.

Text = Actual text of the entry (1–40 bytes); trailing blanks are truncated.

Figure 2-108. Format of a History File Entry

Library Members

The remainder of the library area contains System/32 library members, which are of the following forms:

- *Procedures*—Procedures are groups of OCL and utility control statements used to control the system. These library members are identified by a P in the type column of the library directory listing. Tables showing the OCL statements and parameters and IBM-supplied procedures for System/32 are presented in Appendix H. These topics are also discussed in more detail in the *IBM System/32 System Control Programming Reference Manual,* GC21-7593.

- *Source statements*—Source statements are sets of data (such as RPG II source programs and sort sequence specifications). These sets are identified by an S in the type column of the library directory listing.

- *Load modules*—Load modules are programs that can be loaded for execution (for example, compiled user programs or system utility programs). These programs are identified by an O in the type column of the library directory listing.

- *Subroutines*—Subroutines are nonexecutable object programs that need to be link-edited before being executed. They are identified by an R in the type column of the library directory listing. Also, the subroutines include data management modules for RPG II.

In addition to the procedure, source, load, and subroutine members, the following are included as members in the library:

- *PTF log area ($PTFLOG)*—This source member identifies each PTF applied to the system and will be dynamically increased in size to accommodate the number of PTFs applied (Figure 3-1).

- *Error recording analysis program (ERAP)*—This program will reside permanently on the library or will be loaded from a diskette, when required.

How to Find

The sector number of the first sector containing library members can be found by examining the *Start Sector of Library Members* entry in the *System Information* evoked by the LISTLIBR command. It can also be found at LCSSOMB in the library control sector.

User Data File Area

This area comprises the major portion of a disk. It consists of up to 199 data files allocated in blocks of 10 sectors each. These files are classified as permanent (P) or temporary (T) files. Disk data management operates only on information contained in these files. Note that IBM System/32 SCP supports up to 200 P and/or T files in the disk VTOC (199 user data files and the system library file #LIBRARY).

Figure 2-109 shows the types of disk file organization and the processing types applicable for each.

	Type of File Organization			
	Sequential	**Indexed**	**Direct**	**Pseudo Tape**
Type of Processing	1. Consecutive 2. Random by decimal relative record number 3. Random by binary relative record number (ADDROUT) 4. Pseudo tape (PTAM)	1. Consecutive (ignore keys) 2. Sequential by key 3. Sequential within limits 4. Random by key 5. Random by binary relative record number (ADDROUT) 6. Pseudo tape (ignores keys)	1. Consecutive 2. Random by decimal relative record number 3. Random by binary relative record number (ADDROUT) 4. Pseudo tape (PTAM)	1. Consecutive 2. Random by decimal relative record number 3. Random by binary relative record number (ADDROUT) 4. Pseudo tape (PTAM)

Note: To process any file organization using pseudo tape, the record = 2^n, where, n is greater than or equal to 2 and less than or equal to 8.

● **Figure 2-109. File Organization and Processing**

DISKETTE ORGANIZATION

The usable portion of each diskette is composed of 75 tracks (numbered 0–74) organized in the manner shown in Figure 2-110. Track 0 contains 26 records (sectors), each 128 bytes long. Using the 128-byte format, tracks 1-74 each contain 26 sectors, each of which is 128 bytes long. Using the 512-byte format, tracks 1-74 each contain 8 sectors, each of which is 512 bytes long (see Figure 2-59 for storage capacities). The system uses sector (SS) addressing, starting with sector address X'0001' (track 0, record 1), and numbers sequentially in hex. For more information concerning diskette organization or field contents, see *IBM Diskette General Information Manual,* GA21-9182.

Track	Sector	Contents
0	1, 2	Reserved for IPL and IMPL functions
	3	Reserved for system scratch use
	4	Reserved
	5	Error map
	6	Reserved
	7	VOL1 label (Figure 2-111)
	8–26	Data set header labels (HDR1) (This 19-sector area is also known as the diskette VTOC—Figure 2-112
1–74	—	Data or unallocated available space *Note:* Track 74 is not used when creating basic exchange files.

Figure 2-110. Diskette Organization

Diskette Volume Label (VOL1)

This 128-byte area identifies the volume, the owner, the security, and the sequence of the physical records on the tracks of the specified volume.

How to Find

The VOL1 is located at sector address X'0007' for each diskette.

Format

Figure 2-111 shows the format of the diskette volume label.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	VLIDID	3	Volume label identifier (C'VOL')
3	VLIDIDNO	1	Volume label number (C'1')
4	VLIDVLID	6	Volume identification field (volume ID)
A	VLIDACES	1	Accessibility indicator (blank indicates accessible)
B	VLIDRSV1	26	Reserved
25	VLIDOWNR	14	Owner identification field
33	VLIDRSV2	24	Reserved
4B	VLIDPRLN	1	Physical record length indicator: C'b̸' = 128-byte sector C'2' = 512-byte sector
4C	VLIDPRSC	2	Physical record (sector) sequence code C'b̸b̸' or C'01' through C'13'
4E	VLIDRSV3	1	Reserved
4F	VLIDSTRD	1	Label standard version (should always be C'W')
50	VLIDRSV4	48	Padding (binary zeros)

Figure 2-111. Format of Diskette Volume Label (VOL1)

Diskette Data Set Header (HDR1) Label

Each data set allocated on a diskette must be described by a data set label (HDR1) on that diskette. For a more detailed discussion of the fields in this area, see *IBM Diskette General Information Manual,* GA21-9182.

When the data set is processed, its HDR1 is converted to a 64-byte format 1 which is stored in the SWA (Figure 2-106).

How to Find

The HDR1s are located at sector address X'0008'-X'001A'.

Format

The format of an HDR1 is shown in Figure 2-112.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	IH1DLBID	3	Label identifier ('HDR')—(for deleted data set 'DDR')
3	IH1DLBNM	1	Label number (C'1')
4	IH1DRSV1	1	Reserved
5	IH1DATID	8	Data set identifier
D	IH1DRSV2	4	Reserved
11	IH1DOFNR	5	Unused
16	IH1DBKLN	5	Record length
1B	IH1DRCAT	1	Record attribute (blocked/spanned record = C'R')
1C	IH1DBGEX	5	Beginning of extent (CCHSS)
21	IH1DSCTS	1	Physical sector length indicator: C'b̸' = 128-byte sector C'2' = 512-byte sector
22	IH1DENEX	5	End of extent (CCHSS)
27	IH1DRSV5	1	Reserved
28	IH1DBPND	1	Bypass indicator (blank indicates process)
29	IH1DATSC	1	Data set security (nonblank indicates secured)

Figure 2-112 (Part 1 of 2). Preversion 5 Format Diskette Header Label (HDR1)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
2A	IH1DWTPT	1	Write protected file indicator: C'P' = Read only
2B	IH1DINTL	1	Exchange type indicator: Blank = Basic exchange file C'E' = System file
2C	IH1DMVID	1	Multivolume data set indicator: Blank = File completely contained on this volume C'C' = Continued on another volume C'L' = Last volume of a multivolume data file
2D	IH1DMVNM	2	Multivolume sequence number
2F	IH1DCRDT	6	Creation date (YYMMDD, DDMMYY, or MMDDYY)[1]
35	IH1DRSV6	7	Reserved
3C	IH1DONRN	5	Offset to next record space
41	IH1DRSV7	1	Reserved
42	IH1DEXDT	6	Expiration date (YYMMDD, DDMMYY, or MMDDYY)[1]
48	IH1DVFMK	1	Verify mark: C'V' = Data verified Blank = Not verified
49	IH1DSORG	1	Data set organization
4A	IH1DEDAT	5	End of data (CCHSS)
4F	IH1DRSV8	1	Reserved
50	IH1DPADD	48	Padding (X'00')

[1] YYMMDD is recommended for basic data exchange.

Figure 2-112 (Part 2 of 2). Preversion 5 Format Diskette Header Label (HDR1)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	IH1DLBID	3	Label identifier ('HDR')–(for deleted data set ('DDR')
3	IH1DLBNM	1	Label number (C'1')
4	IH1DRSV1	1	Reserved
5	IH1DATID	8	Data set identifier
D	IH1DRSV2	4	Reserved
11	IH1DOFNR	5	Unused
16	IH1DBKLN	5	Block length[2]
1B	IH1DRCAT	1	Record attribute (blocked/spanned record = C'R')
1C	IH1DBGEX	5	Beginning of extent (CCHSS)
21	IH1DRSV4	1	Physical sector length indicator: C'b̸' = 128-byte sector C'2' = 512-byte sector
22	IH1DSCTS	5	End of extent (CCHSS)
27	IH1DRSV5	1	Reserved
28	IH1DRCBC	1	Bypass indicator (blank indicates process)
29	IH1DATSC	1	Data set security (nonblank indicates secured)
2A	IH1DWTPT	1	Write protected file indicator: C'P' = Read only
2B	IH1DINTL	1	Exchange type indicator: Blank = Basic exchange file C'E' = System file
2C	IH1DMVID	1	Multivolume data set indicator: Blank = File completely contained on this volume C'C' = Continued on another volume C'L' = Last volume of a multivolume data file
2D	IH1DMVNM	2	Multivolume sequence number
2F	IH1DCRDT	6	Creation date (YYMMDD, DDMMYY, or MMDDYY)[1]

[1] YYMMDD is recommended for basic data exchange.

[2] Fields changed from preversion 5 to version 5 form.

Figure 2-113 (Part 1 of 2). Version 5 Format Diskette Header Label (HDR1)

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
35	IH1DRECL	4	Record length[2]
39	IH1DOFST	5	Offset to next record space[2]
3E	IH1DRSV3	4	Reserved[2]
42	IH1DEXDT	6	Expiration date (YYMMDD, DDMMYY, or MMDDYY)[1]
48	IH1DVFMK	1	Verify mark: C'V' = Data verified Blank = Not verified
49	IH1DSORG	1	Data set organization
4A	IH1DEDAT	5	End of data (CCHSS)
4F	IH1DRSV4	1	Reserved
50	IH1DPADD	48	Padding (X'00')

[1] YYMMDD is recommended for basic data exchange.

[2] Fields changed from preversion 5 to version 5 form.

Figure 2-113 (Part 2 of 2). Version 5 Format Diskette Header Label (HDR1)

Embedded Format 1

Each file copied from disk to diskette by the disk copy/display utility has 128 bytes of control information in the first diskette sector immediately preceding the initial sector containing file data on diskette. This control information is called the embedded format 1. It is an extension and modification of the 64-byte disk format 1 for the file, and it is used to ensure that the file reappears on disk with its original attributes.

Figure 2-114 shows the format and contents of the embedded format 1.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0		4	Four-character header FMT1
4		3	Not used
7	F1LABL	8	Filename
F	F1DATE	6	Date the file was created on disk
15	F1RTIN	1	Retain code: C'P' = Permanent C'S' = Scratch C'T' = Temporary
16	F1TYPE	2	File type: *Byte 0*
	Current status (status when file was opened)		X'80' = Indexed X'60' = Pseudo tape X'40' = Consecutive X'20' = Direct X'10' = Offline multivolume X'08' = Input X'04' = Output X'02' = Update X'01' = Add
			Byte 1
	Status when file created		X'80' = Indexed X'60' = Pseudo tape X'40' = Consecutive X'20' = Direct X'10' = Reserved
	Status when file closed		X'08' = Reserved X'04' = Sequential add X'02' = Random add X'01' = Unordered load

Figure 2-114 (Part 1 of 2). Diskette Embedded Format 1

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
18	F1RECL	2	Record length (from 1 through 4096)
1A	F1KEYL	1	Key length (from 1 through 29)
1B	F1KEYO	2	Displacement of the last byte of the key within the record (maximum value of 999)
1D		14	Not used
2B	F1RECN or F1BLKN	3	Number of records or number of blocks
2E		22	Not used
44	F1XDNDXL	2	Number of 256-byte index buffers from which the file was copied
46	F1XDNDXF	1	Index flag: X'00' = Keys do not need sorting X'01' = Keys need sorting
47	F1XDCTYP	1	Copy control: X'00' = Single file copy X'03' = Multiple file (COPYALL) copy
48	F1XDALNM	8	Name associated with the set of files copied via SAVE ALL
50	F1XDALSQ	1	File sequence in set of files copied via SAVE ALL
51	F1XDOFSG	1	File sequence indicator: C'C's More files in set of files copied via SAVE ALL C'L's Last file in set of files copied via SAVE ALL
52		46	Reserved

Figure 2-114 (Part 2 of 2). Diskette Embedded Format 1

BSC Work Area

The BSC work area is 128 bytes long; it contains a series of constants used to load DTF and IOB bytes and supplies working program constants. The work area resides within the BSC object code (#$BSST). The following figure lists the bytes and describes the contents of each. This work area is not used by MRJE.

How to Find

To find the address of the BSC work area, look at field DTFWKA, displacement X'2F', in the BSC DTF.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	DLYIOB	21	BSC IOB—useo to send delay messages and termination sequences (this area includes WKDTFD)
13	WKDTFD	2	Address of BSC DTF
15	BSRJ2D	1	Flag byte: X'80' = Last operation was a GET X'40' = Permanent error during close X'02' = First time BSCA enabled
16	WKERRD	1	Error retry count
17	WKIOBD	2	Address of the last IOB
19	BSFL3D	1	Flag byte: X'80' = Error message logged by line initialization X'10' = Post error to DTF from wait routine
1A	ADIOBD	2	Address of the delay IOB used by interrupt and data management to locate BSC work area

Figure 2-115 (Part 1 of 5). BSC Work Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
1C	BSFLGD	1	Flag byte: X'80' = Signals WAIT to return via ARR after permanent errors (set by line initialization and CLOSE) X'40' = Signals error recovery procedures that a TTD message was the last message received from a remote station X'20' = Signals error recovery procedures to transmit and receive rather than receive only X'10' = BSC line is enabled X'08' = Signals interrupt that a disconnect sequence is being effected by error recovery procedures (when on, interrupt reenters error recovery procedures at BSDISC) X'04' = Signals interrupt that a forward abort sequence is being effected by error recovery procedures (when on, interrupt reenters error recovery procedures at WRTEOT) X'02' = Signals CLOSE that a new file is being opened and CLOSE should not disable BSCA X'01' = Used by error recovery procedures to force reading of an error message after an abort sequence
1D	ACKSD	1	Flag byte: X'80' = Null message sent X'40' = Reserved X'20' = Give invalid ID halt X'10' = EOT received X'08' = Switched line X'04' = Error posted, line not active X'02' on = Receive ACK1 X'02' off = Receive ACK0 X'01' on = Send ACK1 X'01' off = Send ACK0

Figure 2-115 (Part 2 of 5). BSC Work Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
1E	BSFL2D	1	Flag byte: X'80' = CLOSE in process X'40' = Invalid call by user X'10' = Error already logged X'04' = Issue permanent error message from CLOSE X'02' = NAK was the last message sent by this station X'01' = Signals WAIT to return to line initialization after an error occurred during line initialization
1F	WKNEWD	2	Save area for address of new DTF on multiple file BSC jobs
21	INTADD	2	Address of entry to interrupt
23	WKIOS	2	Address of entry to IOS
25	WKWAIT	2	Address to wait
27	WKCLOZ	2	Disk address of close
29	AKEVND	2	Even acknowledgement (constant)
2B	ACKID	2	Odd acknowledgement (constant)
2D	DLESTD	2	DLE STX sequence (constant)
2F	SYND	1	SYN sequence (constant)
30	SNEOTD	1	SYN EOT sequence (constant)
31	WAKD	2	WACK sequence (constant)
33	TTDD ENQD	2	TTD (STX ENQ sequence) (constant)
35	RVID	2	RVI sequence (constant)
37	DISCD	2	Disconnect sequence (constant)
39	ETBCON	1	ETB sequence (constant)
3A	NAKD	1	NAK sequence (constant)
3B	PCTD	1	Not used
3C	LOCS	2	Disk address of line initialization

Figure 2-115 (Part 3 of 5). BSC Work Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
3E	SVRS1D	1	First character received
3F	SVRS2D	1	Second character received
40	SVRS3D	1	Next to last character received
41	SVRS4D	1	Last character received
42	ZEROD	2	Constant of zero (0)
44	ONED	2	Constant of one (1)
46	SNTTXD	2	Number of text blocks sent
48	RCVTXD	2	Number of text blocks received
4A	NAKTBD	2	Number of NAKs received
4C	DATTBD	2	Number of data checks
4E	FABTBD	2	Number of forward aborts received
50	ABRTBD	2	Number of aborts received
52	TADTBD	2	Number of adapter checks on transmit
54	ADTTBD	2	Number of adapter checks on receive
56	INVTBD	2	Number of invalid responses received
58	ENQTBD	2	Number of ENQs received to number of ACKs sent
5A	LSTTBD	2	Number of lost data errors
5C	LATBUD	2	Number of lost connections
5E	TOTTBD	2	Number of receive timeouts
60	TOWTTX	2	Number of transmit timeouts
62	DCOUND	2	Delay time count
64	TOENBD	2	Number of timeouts enabled
66	DSENTD	2	Number of delay messages sent
68	EHTIND	1	Indicator value for error history table: X'04' = Close time X'02' = Reset temporary error X'01' = EHT entry to be logged

Figure 2-115 (Part 4 of 5). BSC Work Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
69	ENTQ	1	Q-byte for error history table (moved from IOB)
6A	EHTR	1	R-byte for error history table (moved from IOB)
6B	EHTSNS	2	Sense bytes for error history table (moved from IOB)
6D	EHTERC	1	Error retry count for error history table
6E	PSTCD	1	Halt characters
6F	DLYIO	12	Delay IOB area
7B	BSQIOB	1	First byte of wait element
7C	BSQEND	2	Last two bytes of wait element
7E	L5CS	2	Disk address of error logging routine (#BSL5)
80	TIOBQ	1	Save area for trace Q-byte

Figure 2-115 (Part 5 of 5). BSC Work Area

SNA/SDLC Common Area

The SNA/SDLC common area is 216 bytes long; it contains a series of constants, pointers, flag bytes, and work areas. Figure 2-116 lists the bytes and describes the contents of each.

How to Find

To find the address of the SNA/SDLC common area, look at field SDTFCOM, displacement X'10' and X'11', in the SNA DTF.

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
0	SCSDLC	3	Common identifier field (DLC)
3	SCMFL1	1	Flag byte 1: X'80' = Stack error X'40' = First chain transmit indicator X'20' = NSA sent to disconnect X'10' = Error logging indicator X'08' = Temporary SDLC error X'04' = Line initialization required
4	SCMFL2	1	Flag byte 2: X'80' = Set normal response mode received X'40' = Send receive ready X'20' = Send receive not ready X'10' = IOB ready for I/O X'08' = Data set ready X'04' on = Send X'04' off = Receive X'02' = Transmit verify done X'01' = Last frame indicator
5	SCMFL3	1	Flag byte 3 (used by command reject only): X'08' = NR count out of range X'04' = Data too long X'03' = Control field with information X'01' = Invalid control field
6	SCMFL4	1	Flag byte 4: X'80' = Purge receive in process X'40' = Disable issued X'20' = Transmission indicator X'10' = ROL required indicator X'08' = Error recovery indicator X'04' = Control byte previously saved X'02' = Command reject state indicator X'01' = Line initialization retry indicator

Figure 2-116 (Part 1 of 8). SNA/SDLC Common Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
7	SCMFL5	1	Flag byte 5: X'80' = NSA to disconnect sent X'40' = Test without poll contained information X'20' = Skip indicator X'10' = SDLC adapter enabled indicator X'08' = SDLC WRAP test to be run X'04' = SNRM received while in normal response mode
8	SCMRIOB	2	Address of receive IOB chain
A	SCMRAIOB	2	Address of active receive IOB
C	SCMTIOB	2	Address of transmit IOB chain
E	SCMTAIOB	2	Address of active transmit IOB
10	SCMNRC	1	SDLC NR count (bits 5-7)
11	SCMNSC	1	SDLC NS count (bits 5-7)
12	SCMCTL	1	SDLC control save area
13	SCMTCT	2	Number of I-frames transmitted
15	SCMRCT	2	Number of I-frames received
17	SCMTTCT	2	Total frames transmitted
19	SCMTRCT	2	Total frames received
1B	SCMBCC	2	Block check error count
1D	SCMIFR	2	Invalid frame count
1F	SCMADS	2	Abortive disconnect count
21	SCMTMO	2	Timeout count
23	SCMADCT	2	Adapter check count
25	SCMRES	2	Reserved
27	SCMSWK1	2	SDLC work area 1
29	SCMSWK2	2	SDLC work area 2
2B	SCMTSMCT	2	Transmit count
2D	SCMSIOB	2	Address of SDLC internal IOB
2F	SCMSTIOB	2	Pointer to start of transmit chain

Figure 2-116 (Part 2 of 8). SNA/SDLC Common Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
31	SCM@IOB1	2	Address of last good processed IOB
33	SCM@IOB2	2	Address of chain pointer
35	SCMTWORK	2	Work area for NR translate
37	SCMCMDT	2	System date
39	SCMTRQ	1	Q-code for I/O operation
3A	SCMTSNS	2	Sense information
3C	SCMTRC	1	SDLC control byte
3D	SCMST@	1	SDLC station address
3E	SCMIFMT	4	I-frames transmitted
42	SCMIFMR	4	I-frames received
46	SCMTERM@	1	Terminal address
47	SCMONE	2	Two-byte constant of one
49	SCMIH@	2	Interrupt handler address
4B	SCMNXIOB	2	Pointer to next receive IOB
	The following 24 bytes contain the SDLC internal IOB and buffer:		
50	SCMIOBWK	2	SDLC IOB work area (SDLC IOB only)
65	SCMCID1	1	Fixed transmit ID field 1
66	SCMID2	1	Fixed transmit ID field 2
67	SCMXID	4	Unique ID information
6B	SCMCON2	2	Constant of two (2)
6D	SCMCON4	2	Constant of four (4)
6F	SCMCON7	2	Constant of seven (7)
71	SCMCRCTL	1	Save area for CMDR control byte
72	SCMDUMMY	2	Dummy IOB for System/32 queue

Figure 2-116 (Part 3 of 8). SNA/SDLC Common Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
74	SCMIOBLN	2	Constant of hex 10
76	SCMLTYP	1	Communication line type: X'02' = Switched line indicator X'01' = Leased line indicator
77	SCMREQ	1	Save area for requests without poll
78	SCMEPAN1	5	Reserved
7D	SCMDTF	2	Address of the SNA DTF
7F	SCMISC	2	Inbound sequence count
81	SCMOSC	2	Outbound sequence count
83	SCMTHO	1	THO save area
84	SCMTH1	1	TH1 save area
85	SCMDAF	1	Destination address field save area
86	SCMOAF	1	Origin address field save area
87	SCMSNF	2	Sequence number field save
89	SCMRH0	1	RH0 save area
8A	SCMRH1	1	RH1 save area
8B	SCMRH2	1	RH2 save area
8C	SCMRU	1	RU0 save area
8D	SCMRU1	1	RU1 save area
8E	SCMRU2	1	RU2 save area
8F	SCSFL1	1	SNA flag byte 1: X'80' = Start data traffic received X'40' = Session bound indicator X'20' = Logical unit activated X'10' = Physical unit activated X'08' = Clear received X'01' = SNA initialization completed

Figure 2-116 (Part 4 of 8). SNA/SDLC Common Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
90	SCSFL2	1	SNA flag byte 2: X'40' = Return to data management required X'20' = New buffer required by data management X'02' = Expedited flow response required from primary X'01' = Normal flow response required from primary
91	SCSFL3	1	SNA flag byte 3: X'40' = Command response required from secondary X'20' = Normal response required from secondary X'10' = Pacing response received X'08' = Pacing response required X'04' = Send LUSTAT X'02' = Contention ERP state
92	SCSFL4	1	SNA flag byte 4: X'10' = Return to send command X'02' = Data received in transmit state X'01' = Deinitialization in progress
93	SCSFL5	1	SNA flag byte 5: X'80' = LOGON required X'40' = LOGON sent X'20' = #SDJ2 has been called X'04' = Request discontact required X'02' = Request discontact sent
94	SCSFL6	1	SNA flag byte 6: X'40' = Change direction indication X'20' = Function manager header indication X'10' = Definite response required X'08' = End chain indication X'04' = Begin chain indication X'02' = End bracket indication X'01' = Begin bracket indication

Figure 2-116 (Part 5 of 8). SNA/SDLC Common Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
95	SCMEPAN2	5	Reserved
9A	SCMEFCMD	1	Expedited flow command byte sent
9B	SCPLNAME	8	Primary station LU-name (PLUNAME)
A3	SCSLNAME	8	Secondary station LU-name (SLUNAME)
AB	SCMNFCMD	1	Normal flow command byte sent
AC	SCMSOAF	1	Host LU session OAF
AD	SCMSDAF	1	Host LU session DAF
AE	SCMARR	2	ARR save area
B0	SCMXR1	2	XR1 save area
B2	SCMBPTR	2	Data management buffer work address
B4	SCMRPAC	1	Secondary receive pacing count
B5	SCMSPAC	1	Secondary send pacing count
B6	SCMSPCNT	1	Secondary send pacing counter
B7	SCMBPNAU	1	Bind primary unit attributes: X'80' on = Multiple element chains X'80' off = Single element chains X'40' on = Multiple chains outstanding X'40' off = Single chain outstanding X'20' and X'10': 00 = No response 01 = EXCP response 10 = DR1/DR2 11 = DR1/DR2 or EXCP response X'02' on = Compression X'02' off = No compression X'01' on = Primary send EB X'01' off = Primary not send EB

Figure 2-116 (Part 6 of 8). SNA/SDLC Common Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
B8	SCMBSNAU	1	Bind secondary unit attributes: X'80' on = Multiple element chain X'80' off = Single element chain X'40' on = Multiple chains outstanding X'40' off = Single chain outstanding X'20' and X'10': 00 = No response 01 = EXCP response 10 = DR1/DR2 11 = DR1/DR2 of EXCP response X'02' on = Compression X'02' off = No compression X'01' on = Secondary send EB X'01' off = Secondary not send EB
B9	SCMBCP1	1	Common protocols byte 1: X'40' on = FMH allowed X'40' off = No FMH X'20' on = Brackets X'20' off = No brackets X'10' on = Bracket rule number 1 X'10' off = Bracket rule number 2 X'08' on = Alternate code okay X'08' off = No alternate code
BA	SCMBCP2	1	Common protocols byte 2: X'80' and X'40': 00 = Full duplex 01 = Half-duplex contention 10 = Half-duplex flip/flop 11 = Master/slave X'20' on = Sender responsible for recovery X'20' off = Primary responsible for recovery X'10' on = Primary first speaker X'10' off = Secondary first speaker X'01' on = Primary wins contention X'01' off = Secondary wins contention

Figure 2-116 (Part 7 of 8). SNA/SDLC Common Area

Displ of Leftmost Byte in Hex	Label	Lng in Bytes in Dec	Description
BB	SCMINIT@	2	SS of #SDJ1
BD	SCMLOG@	2	SS of #SDJ2
BF	SCMINL@	2	SS of #SDJ4
C1	SCMSLOG	2	SS of #SDJ5
C3	SCMJ6@	2	SS of #SDJ6
C5	SCMLTAB@	2	SS of error log table

● **Figure 2-116 (Part 8 of 8). SNA/SDLC Common Area**

Section 3. Diagnostic Aids

The IBM System/32 diagnostic aids include the following:

- PTF (program temporary fix) ($FEFIX)
- Dump ($FEDMP)
- Trace
- APAR ($FEAPR)
- Alter/display storage
- Patch ($FEPCH)

The command formats for the diagnostic aids are listed in this section for quick reference. For a detailed description of the associated parameters, refer to Appendix H.

For a discussion of the history file display utility program, refer to the *IBM System/32 System Control Programming Reference Manual,* GC21-7593.

CUSTOMER ENGINEERING SUPPORT PROGRAMS

The IBM System/32 program, $FEFIX, is used to fix other system programs, and the dump program, $FEDMP, is used to display a log of what PTFs have been applied.

Program	**Use**
$FEFIX	
Field developed patch	To apply a user-provided patch to library modules (R, S, P, or O)
Program temporary fix installation program	To apply an IBM-supplied PTF to library modules (R, S, P, or O)
$FEDMP	
Dump PTF programs	To list the names of library modules that have PTFs applied

Normally when errors are found in either the system control program (SCP) or in a program product (PP), they are fixed by module replacements on a release update disk. However, due to the critical nature of some errors, response time becomes an important factor. This requires that PTFs be applied. A PTF is a temporary patch to a library module that can be applied by an IBM customer engineer. The APPLYPTF command is available to apply these module replacements.

For example, 'APPLYPTF filename,ALL' will apply all PTFs from the file specified by filename. Each PTFMASTR diskette will contain a file (PTFXREF) which cross-references the PTF name to the PTF number and module. To obtain the cross-reference list, copy the file to disk using the 'TOLIBR PTFXREF' command, then call 'LISTLIBR PTFXREF' to produce the listing. The following is contained in PTFXREF:

PTF NUMBER
LOG#
MODULE(S)
REL APPL
COMP
PREREQ/CONCURRENT
DATE APPL

where:

REL APPL = Release applicable
COMP = Component
PREREQ/CONCURRENT = Prerequisite/concurrent PTFs
DATE APPL = Date applied

Refer to Appendix H for the command format.

Field-Developed Patch

When an error in a library module (R, S, P, or O) is found and an IBM-supplied PTF that covers the problem does not exist, a temporary repair can be made by using the field-developed patch program prior to the distribution of an updated module or formal PTF.

Program Temporary Fix Installation Program

PTFs are applied to library modules (R, S, P, or O) by executing $FEFIX and are made available in a format of utility control statements that $FEFIX can recognize.

The same program ($FEFIX) applies field-developed patches or formal IBM-supplied PTFs. To load and run the program, the following OCL statements must be entered:

```
// LOAD $FEFIX
// RUN
```

The OCL statements are followed by these utility control statements:

- For field-developed patch:

```
HDR
     { R }
PTF  { S }   name, ov
     { P }
     { O }
DATA ck,disp,dd,dddd,ddddR comment
END
```

- For formal IBM-supplied PTF:

```
HDR cksm,ptfidxxxxx
           { R }
PTF cksm,  { S }   name, lev, ov
           { P }
           { O }
DATA cksm,disp,dd,dddd,ddddR comment
END cksum
```

Utility Control Statement Descriptions

The statement and field descriptions having the same name or identifier are applicable for both field-developed patches and IBM-supplied PTFs.

Statement Descriptions

HDR — The HDR statement must start in column 1 and must be followed by a blank. HDR followed by all blanks identifies a field-developed patch (using a check byte). HDR is the first statement of a PTF and only one is allowed per PTF.

PTF — The PTF statement must start in column 1 and must be followed by a blank. Ten PTF statements are allowed per PTF. The first PTF statement must follow the HDR statement. The PTF statement identifies the module being patched.

DATA — The DATA statement must start in column 1 and must be followed by a blank. DATA identifies the area to be patched and the patch data. The DATA statement must follow a PTF statement or another DATA statement. It may be followed by another DATA statement, another PTF statement, or the END statement.

END — The END statement must start in column 1 and must be followed by a blank. The END statement must follow a DATA statement. END terminates the PTF. If $FEFIX is executed from a procedure, a // END statement is required after the END statement to terminate $FEFIX.

* — An asterisk signifies a comment statement; this statement is ignored by $FEFIX.

Field Descriptions

Fields are not column dependent; they are sequence dependent and can be separated by commas if desired. The cksm, ck, disp, and data fields must be hex characters (0-9, A-F).

cksm — This field is a 4-character accumulative hash total of the PTF. The check sum from the statement is compared against one developed by $FEFIX and, if not equal, an error message is displayed. The cksm on the END statement is a checksum of the entire PTF, HDR, to END. The cksm on the HDR statement identifies this as an IBM-supplied PTF and not a field-developed patch.

ptfidxxxxx — This field is a 10-character field identifying the PTF. This field is not valid for a field-developed patch. The first five characters of the PTF ID must match the first five characters of the first module being patched. The last five characters are a unique 5-digit number associated with the PTF. This number is placed on the PTF log module after the END statement is received. The system month and day is placed in the PTF log for a field-developed patch.

$\left\{\begin{matrix}\mathrm{R}\\ \mathrm{S}\\ \mathrm{P}\\ \mathrm{O}\end{matrix}\right\}$ name — This field is the module type and name. Allowable types are S (source), R (subroutine), P (procedure), and O (load). Name is the name of the module being patched. The first five characters of the name on the first PTF statement must match the first five characters of the PTF ID. Subsequent module names are not checked against the PTF ID.

lev — This field is a 3-digit decimal release level of the module being patched. If the level of the module does not match the level on the PTF statement, a formatted message is displayed. Option 0 (ignore) continues with the PTF. This field is invalid on a field-developed patch.

ov — This field is a 2-position decimal overlay number of the module being patched. If the module has no overlays or the root is being patched, this field should be omitted.

ck — This field is a 1-byte 2 character (verify byte) field identifying the first byte of the patch area. This field is only valid on a field-developed patch.

disp — This field is a 4-position field marking the start of the patch. It is the absolute displacement of the patch into the module.

dd — This field is one byte of patch data.

dddd — This field is two bytes of patch data.

ddddR — This field is a relocatable patch.

Patch data can be separated by commas if desired, but a blank signifies the end of the patch data. An even number of patch characters must be entered. A maximum number of 105 characters, which includes Rs and commas, can be entered on a DATA statement.

comment — Comments can be placed on data statements providing they are separated from the patch data by a blank.

Dump ($FEDMP)

Command Format

$$\text{DUMP}\left[\begin{array}{l}\underline{\text{MAIN}}\\\text{CONTROL}\\\text{HISTORY}\\\text{PTF}\\\text{CONFIG}\\\text{DISK}\\\text{MICR}\end{array}\right]\;,\;\left[\begin{array}{l}\underline{\text{PRINTER}}\\\text{CRT}\end{array}\right]\quad\left[\begin{array}{l}\underline{\text{F1}}\\\text{'I1}\end{array}\right]$$

Description

Dump displays data on the display screen or printer from various locations in the IBM System/32 system. Dump also displays the APARFILE from the diskette.

When errors occur, system information can be saved on the CE cylinder (2-track area reserved on the disk) and then dumped by $FEDMP at a later time. When the system diagnoses an abnormal termination error (that is, invalid address, invalid op, or invalid Q), the contents of the first 16K of main storage, all of control storage (4K words), and the last 20 sectors of the SWA history file are automatically written to the CE cylinder. Additional main and control storage is saved in the additional main storage dump area following the SWA. The 1255 attachment controller (9 sectors) is also dumped to the additional main storage area following the SWA.

A dump to the CE cylinder can also be invoked by each of the following:

1. Branching to main storage address X'0000'.

2. Pressing RESET and then CE START on the CE console.

3. Taking D option to a formatted message display.

Dump can be loaded by the following OCL statements:

```
// LOAD $FEDMP
// RUN
```

Dump options are then requested by the // DUMP control statement:

$$\text{// DUMP LIST -}\left\{\begin{array}{l}\underline{\text{MAIN}}\\\text{CONTROL}\\\text{DISK}\\\text{PTF}\\\text{CONFIG}\\\text{HISTORY}\\\text{MICR}\end{array}\right\}\text{, OUTPUT -}\left\{\begin{array}{l}\underline{\text{PRINTER}}\\\text{CRT}\end{array}\right\}\text{,}$$

$$\text{INPUT -}\left\{\begin{array}{l}\underline{\text{F1}}\\\text{I1}\end{array}\right\}$$

If INPUT-I1, dump will test for the existence of an APARFILE for all options except LIST-DISK. If no APARFILE exists, error message 1869 CE CYLINDER DATA MIGHT BE INCORRECT will be issued.

Notes:

1. The dump utility allows multiple // DUMP control statements.
2. A dump following a dump results in a PROC CHK.

Dump can be terminated by the control statement // END.

Dump Format (Prologue)

ERRMIC—MIC number that caused the error.

OP/Q—System instruction operation code and Q-byte when the error occurred.

TRACE—Next available entry in the main storage RIB TRACE area, except when communication trace is active.

Dump Options

Six dump options are available on the IBM System/32. Figure 3-2 and the following list gives a description of each. Since main and control storage dumps are basically the same, they are referred to only as storage dumps. A storage dump refers to dumps from the CE cylinder or the APARFILE on an APAR diskette only and not actual real time storage.

Storage Dump to Printer

Command format:

$$\text{DUMP}\begin{bmatrix}\underline{\text{MAIN}}\\ \text{CONTROL}\\ \text{MICR}\end{bmatrix},\ \text{PRINTER}\begin{bmatrix},\underline{\text{F1}}\\ \text{I1}\end{bmatrix}$$

If INPUT is from I1, DUMP tests for the existence of APARFILE; if none exists, an error message is issued.

This dump prompts for a starting storage address and an ending storage address. The address limits for main storage are 0000-3FFF, (16K), 0000-5FFF (24K), and 0000-7FFF (32K); for control storage, 0000-0FFF (4K) and 0000-1FFF (8K). Addresses are rounded down to a 256-byte (main storage) or 256-word (control storage) boundary. The minimum amount of storage printed is 256 bytes (main storage) or 256 words (control storage). Storage data is listed 32 bytes to a line. Storage addresses for control storage are listed as words.

Note: Dump tests the CE cylinder or APARFILE for valid data before passing control to the storage dump. If the data is not valid, a formatted message is issued with options 0, 1, or 3:

0 = Dump anyway (prolog prompt will be bypassed)
1 = Read a new control statement
3 = Cancel

If the data has previously been dumped from the CE cylinder, DUMP issues a formatted message with options 0, 1, or 3 (0 = continue, 1 and 3 = same as above).

Storage Dump to Display Screen

Command format:

```
       [MAIN   ]
DUMP   [CONTROL] ,   CRT [, F1]
       [MICR   ]         [  I1]
```

If INPUT is from I1, DUMP tests for the existence of APARFILE; if no APARFILE exists, an error message is issued. This dump does not prompt for a starting address; it assumes a starting address of zero.

The display consists of one line of registers (XR1, XR2, PSR, IAR, ARR), and five lines of data (8 bytes per line). The display is continuous until the upper limit of storage is reached. The display is rolled up or down by use of the ROLL ↑/↓ keys. The combination of the ROLL keys and the repeat (REP) key causes the display to be rolled at a maximum of 8 lines per second. Line 1 remains unchanged. The type of dump (main or control) is indicated by an M or C in the fifth position of the first data line, display line 2.

The display screen cursor is positioned at the start of display line 2. This allows a new area of storage to be selected. A new starting address can be entered (must be hex) at the time of display and this new area of storage will be displayed. An additional feature of this dump is that the user can view either main or control storage without returning to read another control statement. If, when a new starting address is entered, the fifth character is an M or C, main or control storage is dumped respectively. If the fifth character is left blank, no change occurs. All entries are checked for validity (that is, hex digits), and they are checked to determine if they are within limits. If an error exists, the display is not changed, except an error indicator of ???? appears at the start of data line.

Disk Dump to Printer

Command format:

DUMP DISK, PRINTER $\left[, \frac{\text{F1}}{\text{I1}}\right]$

This dump prompts for a starting address (SS format) and for the number of sectors (in hex) to be dumped (see note). Prompt replies are checked for validity and any error causes the prompt to be reissued. Replies should be in hex and right-justified. Disk address limits are:

Disk (3.2 megabyte):	0001-3264
Disk (5.0 megabyte):	0001-4E84
Disk (9.1 megabyte):	0001 to 8D54
Disk (13.7 megabyte):	0001-D3E0
Diskette (128-byte format):	0001-079E
Diskette (512-byte format):	0001 to 026A

The alternate sector areas cannot be displayed.

Note: If the fifth character of the reply is a B, the reply is treated as a decimal block number, and that block is dumped.

Disk Dump to Display Screen

Command format:

DUMP DISK, CRT $\left[, \frac{\text{F1}}{\text{I1}}\right]$

This dump prompts for a starting sector address (SS format). It calls $FEKEY (CRT WINDO DISPLAY) to display the sector requested. The option passed to $FEKEY is HEX-YES. If HEX-NO is desired, enter an N as the fifth character of the sector address prompt.

Data keys are not active. The following function keys are pertinent to disk dump to CRT (see PATCH utility for other key functions during the $FEKEY display):

REC ADV ROLL↑	Dump reads the next sector (SS + 1) and displays it.
ROLL↓	Dump reads the previous sector (SS – 1) and displays it.
ENTER ENTER– ENTER+	Dump issues a new prompt.

Magnetic Character Reader Controller Storage Dump to Printer

Command format:

$$\text{DUMP MICR, PRINTER} \quad \left[,\frac{\text{F1}}{\text{I1}}\right]$$

If INPUT is from I1, DUMP tests for the existence of APARFILE; if no APARFILE exists, DUMP issues an error message. Then DUMP prompts for a starting storage address and an ending storage address. The address limits for magnetic character reader controller storage are 0000-0FFF. Addresses are rounded down to a 256-byte boundary. A minimum of 256 bytes of storage is printed at 32 bytes per line.

Note: DUMP tests the CE cylinder or APARFILE for valid data before passing control to the storage dump. If the data is not valid, DUMP issues a formatted message with options 0, 1, or 3:

0 = Dump anyway
1 = Read a new control statement
3 = Cancel

If the data has previously been dumped from the CE cylinder, DUMP issues a formatted message with options 0, 1, or 3:

0 = Continue
1 = Read a new control statement
2 = Cancel

Magnetic Character Reader Controller Storage Dump to Display Screen

Command format:

$$\text{DUMP MICR, CRT} \quad \left[,\frac{\text{F1}}{\text{I1}}\right]$$

If INPUT is from I1, DUMP tests for the existence of APARFILE; if no APARFILE exists, DUMP issues an error message. This dump does not prompt for a starting address; it assumes a starting address of zero.

One line of registers (XR1, XR2, PSR, IAR, ARR) and five lines of data are on the display screen (8 bytes per line). The display is continuous until the upper limit of storage is reached. Use the ROLL↑/↓ keys to scroll the display forward or backward one line at a time, or a combination of the ROLL keys and the repeat (REP) key to scroll a maximum of 8 lines per second. Line 1 does not change. The A in position 5 of line 2 (the first data line) indicates this is a magnetic character reader controller dump.

The cursor is at position 1 of line 2 on the display screen. To display a new area of storage, enter a new starting address (must be hexadecimal) in line 2. If you attempt to dump main or control storage by changing the dump type from A to M or C, an error will occur. All addresses are checked for validity (hexadecimal digits) and they are checked to determine if they are within storage limits. If an error exists the display does not change, however, an error indicator of ???? will appear at the start of the data line.

History Dump

Command format:

```
                     ┌PRINTER┐   ┌  F1┐
DUMP HISTORY,        │-------│   │  --│
                     └CRT    ┘   └, I1┘
```

History records are shown in the same format as they are in the SWA history file. The history file referred to here is the 20 sectors of the SWA saved on the CE cylinder or APAR diskette. If OUTPUT-PRINTER, all entries are printed. If OUTPUT-CRT, successive entries can be viewed by using the ROLL keys. The screen display can be terminated by the ENTER key. Exit is to the DUMP control statement read. If INPUT-I1, DUMP tests for the existence of APARFILE on the mounted diskette; if no APARFILE exists, an error message is issued.

PTFLOG Dump

Command format:

```
                 ┌PRINTER┐   ┌  F1┐
DUMP PTF,        │-------│   │  --│
                 └CRT    ┘   └, I1┘
```

The PTF log is dumped from the disk library ($PTFLOG) or the APARFILE on an APAR diskette. PTFLOG entries are 40 bytes long and are shown just as they exist in the PTF log (see Figure 3-1 for the printed format of the PTF log).

If OUTPUT-PRINTER, all PTFLOG entries are printed. If OUTPUT-CRT, entries can be viewed by using the ROLL keys. The screen display can be terminated by the ENTER key. Exit is to the DUMP control statements read. If INPUT-I1, DUMP tests for the existence of APARFILE; if no APARFILE exists, an error message is issued.

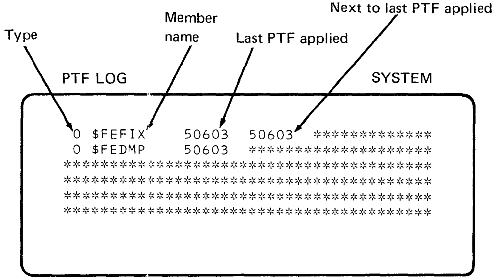
Attachment Controller Trace Buffer

The 16-byte attachment controller trace buffer maintains a history log of all commands issued to the attachment controller by the system. The number of commands saved depends on the type of command issued and the sense byte status that is logged. Each logged command is preceded by a 1-byte command identifier (hex FF). When searching the trace buffer reference location hex 01B3 in the attachment controller dump for the pointer to the latest entry in the trace buffer. Starting from the last entry logged, continue backward to the first command identifier (hex FF). This is the start of the latest entry logged. To find the next latest entry, continue backward through the trace buffer to the next command identifier. Continue this process until a complete wraparound to the byte pointed to from the last entry pointer is reached. The trace function is always active.

The following is an example ot the entries in the attachment controller trace buffer.

FF	Command identifier
00	Wait command
FF	Command identifier
12	Read command (normal)
04	No data sense (acknowledge)
00	Sense byte 1
00	Sense byte 2
FF	Command identifier
00	Wait command
FF	Command identifier
12	Read command (normal)
04	No data sense (acknowledge)
00	Sense byte 1
00	Sense byte 2
FF	Command identifier
00	Wait command

Contact your source of technical support for assistance in determining the malfunctioning area. The command modifiers are shown as a reference aid.



```
  PTF LOG                                        SYSTEM

 0 $FEFIX    50603  50603  **************
 0 $FEDMP    50603  *********************
*****************************************
*****************************************
*****************************************
*****************************************
```

Note: There can be only one PTF record (40 bytes) for any library member. The last five characters of the PTFID are placed in the log. (For a field-developed patch, the character 5, followed by the first four characters of the date, is logged.) Only the four most current PTFs applied can be shown and any previous PTF IDs would be lost.

Figure 3-1. Format of Printed PTF Log

Configuration Record Dump

Command format:

```
DUMP CONFIG, [PRINTER]  [, F1]
             [CRT    ]  [  I1]
```

The configuration record is dumped from either #LIBRARY on the fixed disk (F1) or the APARFILE data set on an APAR diskette (I1). The format of the display and the printer output is the same with the exception that the print image is listed only on the printer.

The following items of the configuration record are displayed:

System date
Date format
Release level
Form length
Inquiry information
Disk size
Image character set size
Print image

Exit is to the DUMP control statement read. To exit from the screen display, press the ENTER key.

Dump	Function	How to Use
Storage—main or control (to printer)	Lists storage from CE cylinder	Reply to prolog prompt. Reply to address prompts with low and high limits.
Storage—main or control (to display screen)	Displays storage from CE cylinder	Fifth type digit on first data line identifies type of storage displayed. To display another section of storage, enter the address followed by the type digit. Additional storage can be scrolled by using ROLL↑/↓ keys. Enter E in the type digit to terminate option.
Disk (to printer)	Lists disk storage from input device	Reply to prompts. (Second prompt is number of sectors to dump in hex.) If the fifth character of the reply is B, one block will be listed.
Disk (to display screen)	Displays disk storage from input device	Reply to prompts. See PATCH utility for a description of keyboard functions during display. Enter N or Y as fifth character of prompt to change display hex option.
History (to printer)	Lists history file from CE cylinder	—
History (to display screen)	Displays history file from CE cylinder	Display additional records by using ROLL↑/↓ keys. Use ENTER key to terminate option.
PTF (to printer)	Lists $PTFLOG	—
PTF (to display screen)	Displays $PTFLOG	Display additional record by using ROLL↑/↓ keys. Use ENTER key to terminate option.
CONFIG (to printer)	Lists configuration record	—
CONFIG (to display screen)	Displays configuration record	Use ENTER key to terminate option. Print image is not displayed.
MICR (to printer)	Lists magnetic character reader controller storage from CE cylinder or APARFILE	Reply to address prompts with low and high limits.
MICR (to display screen)	Displays magnetic character reader controller storage from CE cylinder or APARFILE	Fifth type digit on first data line identifies type of storage displayed. To display another section of storage, enter the address. Additional storage can be scrolled by using ROLL↑/↓ keys. Enter E in the type digit to terminate option.

Figure 3-2. System/32 Dump Options

TRACE Function

Command Format

```
TRACE  ⎡ ALL ⎤  [,WAIT]  [,FDIOS]  [,CSFDIOS]  [,PUSH]
       ⎣ OFF ⎦

[,PULL]  [,DISABLE]  [,ENABLE]  [,QUEUE]  [,LDCS]  [,LOADER]

[,XIENT]  [,XFER]
```

Description

The TRACE function provides the ability to keep a history of the last 21 important events occurring in the system. Selected system functions are logged into a sector of the main storage area as they occur during normal operation. This area is a wraparound area containing 21 entries of 12 bytes each. The entries in the main storage area are at locations X'600' to X'6FB'. (X'6FE' and X'6FF' contain the address of the next trace entry after a dump is taken.) The information contained includes registers, parameter lists, and selected disk IOB information.

Notes:

1. For a description of the parameters, see Appendix H.
2. After an IPL or an alter/display, only LDCS, LOADER, and XIENT are traced.

Function

Whenever an XFER instruction is executed or a RIB (request indicator byte) or other branch to the main storage general entry point is issued, and if the function is one for which a trace is specified, a 12-byte entry is built and placed into main storage.

Whenever a dump to the CE cylinder is taken, this area, being part of main storage, is also dumped. When the DUMP command or $FEDMP utility is evoked, this area is formatted during the dump prolog. See Figure 3-3.

Figure 3-3. Format of TRACE Output

```
TYPE RIB  RET@   UN   P2   P0   P1   XR1    XR2      TYPE RIB  LSR1   LSR2   LSR3   LSR4   LSR5   LSR6
 M   68   2508   00   08   06   16   0804   2514
 M   94   2575   00   03   05   A7   0899   2883
 M   89   2AF2   00   01   06   ID   28AD   2DE4
 M   89   2D55   00   01   06   ID   301C   2DE4
 M   89   2D6C   00   01   06   ID   2DFA   2DE4
 M   69   1117   61   37   00   00   0804   0A29
                                                    C    81   8081   0D3D   8002   0616   0069   0068
 M   89   145A   00   01   06   ID   0804   1C8F
 M   80   09A4   09   02   06   28   0804   0804
 M   85   01F1   00   02   06   41   02AF   0832
 M   78   01A6   00   F0   25   30   001E   01EA
 M   94   26FB   00   03   05   A7   2C4C   2E13
 M   94   27B7   00   03   05   A7   2C9C   2E25
 M   89   3039   00   01   06   ID   2DC7   332B
 M   89   329C   00   01   06   ID   3563   332B
 M   89   32B3   00   01   06   ID   3341   332B
 M   81   3039   00   01   06   ID   2DC7   332B
 M   89   329C   00   01   06   ID   3563   332B
 M   89   32B3   00   01   06   ID   3341   332B
 M   85   2A70   00   02   06   41   001E   2500
 XFER     INST   Q-BYTE =  R-BYTE =  PARAM =        C    0F   800F   0C77   241F   1006   0000   A204
```

where (in this example):

M (type)	= Main storage
81 (RIB)	= Library find
3039	= The return address
00	= Unused
01	= The number of sectors to load less one (N byte)
06 ID	= The disk address of transient
2DC7	= Contents of XR1
332B	= Contents of XR2

Note: See Figure 3-4 for all possible entries.

TRACE Entry Data

Information traced includes RIBs, ARR, a 3-byte parameter list, XR1 and XR2. RIBs 00, 01, 05–08, 0C and 0D cannot be traced.

Figure 3-4 describes the contents of the 12-byte trace entry with their associated functions.

		12-Byte Trace Entry									
Control Registers		1		2	3		4		5	6	
12 Bytes (0-B)		0	1	2 3	4	5	6	7	8 9	A	B
		TYPE	RIB	RETURN ADDR	UN	P2	P0	P1	XR1	XR2	
Primary RIBs	WAIT FDIOS PULL DISABLE ENABLE PUSH Q/DE-Q LDCS	M ↓	02 03 04 09 0A 0B 0E 0F	RETURN @ ↓		Param List[1] (Byte 2) ↓	Param List (Byte 0) ↓	Param List (Byte 1) ↓	XR1 ↓	XR2 ↓	
Transient Loader RIBs	(Refer to Figure E-3, transient loader)	M ↓	80 To DF	RETURN @ ↓		N Byte (number of sectors to load minus 1)	Disk Address (SS) ↓		XR1 ↓	XR2 ↓	
Relocating Loader RIBs	(Refer to Figure E-2, relocatable loader)	M ↓	40 To 7F	RETURN @ ↓	N/A		N/A		XR1 ↓	XR2 ↓	
XFER Instruction	XFER	FS Op Code F5	Q Code	N/A	N/A				Address of Parameter List	N/A	R Code

[1] Byte 2 contains the inline parameter list.

Notes:

1. The 12 bytes (0-B) of the trace entry are control storage registers 1–6, that is, bytes 0 and 1 are register 1, bytes 2 and 3 are register 2, etc.
2. Normally, for control storage calls, bytes 0-3 will be the same as for main storage calls, that is, 0 (type), 1 (RIB), and 2-3 (return address). Values for bytes 4–B will depend on the operation and the location from which they have been issued.

Figure 3-4. 12-Byte Trace Entry (Main Storage)

BSC Trace

Input/output information for BSC can also be stored as entries in the trace table when BSC DEBUG is requested (see PLCA, Figure 2-10). ITB interrupts, enables, and disables are not recorded. Data is recorded in the trace table when an I/O interrupt occurs. The format of this data is shown in Figure 3-5.

Note: DEBUG remains on until IPL or until it is reset by the ALTERBSC command.

See the ALTERBSC command (Figure H-3) for further information on how to start the BSC trace.

Byte	Contents
0	Q-byte of SIO
1	Reserved
2, 3	Sense information
4, 5	CAR and following byte (before I/O) (These are the first two bytes transmitted)
6, 7	Two bytes preceding TAR (These are the last two bytes transmitted)
8, 9	TAR and following byte (These are the first two bytes received)
10, 11	Two bytes preceding CAR (after I/O) (These are the last two bytes received)

Notes:

1. When a 2-second timeout is enabled, all the I/O data bytes to be stored are stored as X'FF'.
2. For a receive time-out, the byte following the TAR and the two bytes preceding the CAR after I/O are stored as X'FF'.
3. For receive initial and receive only, the two bytes preceding and the byte following the TAR are stored as X'FF'.

Figure 3-5. Format of BSC Trace Table Entry

MRJE/WS BSC Trace

Input/output information for BSC when the MRJE/WS utility is being run is stored as entries in the trace table when BSC DEBUG is requested (see PLCA, Figure 2-10). ITB interrupts, enables, disables, and two-second timeouts are not recorded. Data is recorded in the trace table when an I/O interrupt occurs. The format of this data is shown in Figure 3-6.

Note: DEBUG remains on until IPL or until it is reset by the ALTERBSC command.

See the ALTERBSC command (Figure H-3) for further information on how to start the BSC trace.

Byte	Contents
0	Q-byte of SIO
1	R-byte of SIO
2, 3	Sense information
4 through 11	First 8 bytes of the MRJE/WS BSC buffer

Note: For a detailed description of these bytes see *IBM System/32 Data Communications Logic Manual,* SY21-0551.

Figure 3-6. Format of MRJE/WS BSC Trace Table Entry

SDLC Trace

Input/output information for SDLC can also be stored as entries in the trace table when SDLC DEBUG is requested (see PLCA, Figure 2-8). Data is recorded in the trace table when an I/O interrupt occurs. The format of this data is shown in Figure 3-7.

Note: DEBUG remains on until IPL or until it is reset by the ALTERSDL command.

See the ALTERSDL command (Figure H-3) for further information on how to start the SDLC trace.

Byte	Contents
0	SIO Q-byte
1-2	IOB sense bytes
3	Transmit control byte
4	Receive control byte
5-6	IOB address
7	One byte of transmission header
8-10	Three bytes of request header
11	One byte request unit

Notes:

1. Receive ready swapping is not traced. During a transmit and receive operation, if both the transmit and receive control bytes are RR (receive ready), no entry is made in the trace table.
2. If the transmit control byte or receive control byte field is not applicable, X'FF' is inserted into the field.
3. For a detailed description of these bytes, see the *IBM System/32 Data Communications Logic Manual,* SY21-0551.

Figure 3-7. Format of SDLC Trace Entry

APAR ($FEAPR)

Command Format

```
APAR vol-id, [object program name] [,source program name]
```

Description and Operation

The APAR procedure collects information that can help IBM service personnel isolate and correct programming problems that might occur in the system. The procedure creates files on a diskette called APARFILE and FIXDFILE. These files can later be displayed using the DUMP utility or they can be submitted with an APAR (authorized program analysis report).

The APAR procedure is executed by the APAR command or by the following OCL statements:

```
// LOAD $FEAPR
// FILE NAME-APARFILE,UNIT-I1,PACK-vol-id,RETAIN-999
// FILE NAME-FIXDFILE,UNIT-I1,PACK-vol-id,RETAIN-999
// RUN
```

Note: The APAR command should be executed as soon as possible after an abnormal termination or suspected point of failure for the information to be valid.

Information Copied by APAR

In creating files on a diskette, certain information is written into APARFILE and into FIXDFILE. The information written into each includes:

- APARFILE
  1. CE cylinder
     a. First 16K bytes of main storage
     b. Control storage
     c. Last 20 sectors of the history file
  2. PTF log
  3. Configuration record
  4. Additional main storage (over 16K)
  5. Additional control storage (over 4K words)
  6. Attachment controller storage (Magnetic Character Reader Controller)
- FIXDFILE
  1. VTOC
  2. SWA
  3. Rollin/rollout area
  4. Error history table
  5. Library control sector

When using the APAR command, the diskette data files created for the object program are called APARLOAD; those created for the source program are called APARSRCE.

After creating all necessary files on the diskette, an informational message is issued. This message suggests that other information, which might aid in the problem resolution, be written to the diskette. This information includes user data files and programs that may be associated with a failure.

CAUTION
Beware of customer security requirements

ALTER/DISPLAY

Description

This function allows examination and modification of the data present in registers and control or main storage on a real time basis. It can be used to:

- Display any 40 bytes of main storage.
- Display any 20 words of control storage.
- Alter any main or control storage location.
- Display and/or alter the IAR, ARR, XR1, XR2 and PSR registers.

Note: Alter/display is not available when the BSCA or SDLC is active.

Function

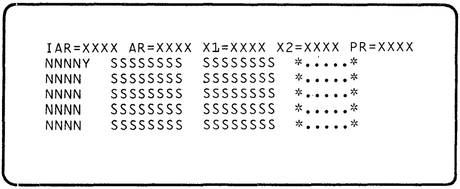
At any point in the operation of the machine, the user can stop the machine and examine or modify the contents of main or control storage. If the IPL switch on the CE panel is in the DISKETTE position, this function can be evoked by first pressing the STOP key on the operator panel and then the INQ key on the keyboard.

Note: If the IPL switch is left in the DISKETTE position, the INQ key will not perform other functions properly.

Alter/display is evoked automatically each time the START key is pressed when the CE console mode selector switch is in the system instruction step position.

Instead of the normal inquiry function, a microcode routine is evoked that allows the customer engineer or program systems representative to display the contents of main or control storage. To exit from the display, press the START key.

The format of the display is as follows:

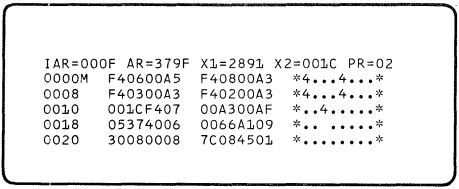


```
IAR=XXXX AR=XXXX X1=XXXX X2=XXXX PR=XXXX
NNNNY  SSSSSSSS  SSSSSSSS  *.....*
NNNN   SSSSSSSS  SSSSSSSS  *.....*
NNNN   SSSSSSSS  SSSSSSSS  *.....*
NNNN   SSSSSSSS  SSSSSSSS  *.....*
NNNN   SSSSSSSS  SSSSSSSS  *.....*
```

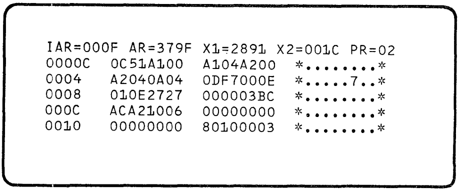
where:

XXXX = Register values
NNNN = Main or control storage address
Y = C—Control storage.
M—Main storage.
T—An instruction trace where registers are listed on the printer for each main storage instruction executed. (See Figure 3-9 for an example of the instruction trace printout.) To terminate the instruction trace, evoke ALTER/DISPLAY with the STOP and INQ keys and enter an E in the Y field.
E—End. Terminate and go back to the machine state at the time the function was evoked.
SSSSSSSS = Actual storage values at given address.

Figure 3-8 gives two examples of actual displays, one from main storage and one from control storage.



```
IAR=000F AR=379F X1=2891 X2=001C PR=02
0000M  F40600A5  F40800A3  *4...4...*
0008   F40300A3  F40200A3  *4...4...*
0010   001CF407  00A300AF  *..4.....*
0018   05374006  0066A109  *.. .....*
0020   30080008  7C084501  *........*
```



```
IAR=000F AR=379F X1=2891 X2=001C PR=02
0000C  0C51A100  A104A200  *........*
0004   A2040A04  0DF7000E  *.....7..*
0008   010E2727  000003BC  *........*
000C   ACA21006  00000000  *........*
0010   00000000  80100003  *........*
```

Figure 3-8. ALTER/DISPLAYs from Main and Control Storage

The address value NNNN is derived initially from the address switches on the CE panel. (If the address switches are set above the highest storage address, the last 40 bytes of storage are selected.) The Y value is initially set to display main storage (M).

To display another location:

1. Key in new address NNNN.

2. Set Y to M or C.

3. Press the ENTER key.

To alter any location:

1. Display location desired.

2. → (l/c) over to position to be changed.

3. Key in new values.

4. Press ENTER.

To alter the registers:

1. Press ↑ (l/c) to move cursor to line 1.

2. → (l/c) over to register to be changed.

3. Key in new value.

4. Press ENTER.

The valid function keys used by ALTER/DISPLAY are:

→ (l/c)	Move cursor right one position.
← (l/c)	Move cursor left one position.
↑ (l/c)	Move cursor up one line.
↓ (l/c)	Move cursor down one line.
ROLL↑	Roll through storage 8 bytes at a time. The address NNNN is incremented by 8 bytes (MS) or 4 words (CS).
ROLL↓	The storage address NNNN is decremented by 8 bytes (MS) or 4 words (CS).
ENTER	Values on the display screen replace the current values at those locations. If Y = E, exit from alter/display and return to the original machine state.
START	Returns to user program.

```
IAR 000F AR 37A9 X1 2872 X2 001E PR 02
IAR 000F AR 37A9 X1 2872 X2 001E PR 02
IAR 37AC AR 37A9 X1 2872 X2 001E PR 02
IAR 37B6 AR 37A9 X1 2872 X2 001E PR 02
IAR 37B9 AR 37A9 X1 2872 X2 001E PR 02
IAR 37BC AR 37A9 X1 2872 X2 001E PR 02
IAR 37BF AR 37A9 X1 2872 X2 001E PR 02
IAR 37C2 AR 37A9 X1 2872 X2 001E PR 02
IAR 37C6 AR 37A9 X1 2872 X2 001E PR 02
IAR 37C9 AR 37A9 X1 2872 X2 001E PR 02
IAR 37CC AR 37A9 X1 2872 X2 001E PR 02
IAR 37CF AR 37A9 X1 2872 X2 001E PR 12
IAR 37E0 AR 37A9 X1 2872 X2 001E PR 02
IAR 37E3 AR 37A9 X1 2872 X2 001E PR 04
IAR 37E6 AR 37A9 X1 2872 X2 001E PR 04
IAR 37EB AR 37A9 X1 2872 X2 001E PR 04
IAR 37F1 AR 37A9 X1 2872 X2 001E PR 02
IAR 37F7 AR 37A9 X1 2872 X2 001E PR 02
IAR 37FB AR 37A9 X1 2872 X2 001E PR 02
IAR 3801 AR 37A9 X1 2872 X2 001E PR 02
```

Note: The last instruction executed will be listed twice when the T option is first selected.

Figure 3-9. Instruction Trace

PATCH Utility ($FEPCH)

Command Format

```
PATCH [F1]   [,NOHEX]
      [I1]
```

Description

The PATCH utility can be used to patch sector data on either the diskette or the disk. The sector to be patched is displayed, 40 bytes at a time, on the display screen. Then, the keyboard is utilized to enter patch data.

CAUTION
PATCH can alter any sector of disk storage with the exception of tracks 0, 1, 2, 4, and 5, but it does not test whether the disk data is a VTOC, library directory, etc. Therefore, an *alteration error could cause unpredictable results.*

Function

PATCH can be loaded by using the PATCH command or by the following control statements:

```
// LOAD $FEPCH
// RUN
                ┌           ┐     ┌        ┐
                │        F1 │     │    YES │
// PATCH        │INPUT-  -- │  ,  │HEX --- │     (see note)
                │        I1 │     │    NO  │
                └           ┘     └        ┘
// END
```

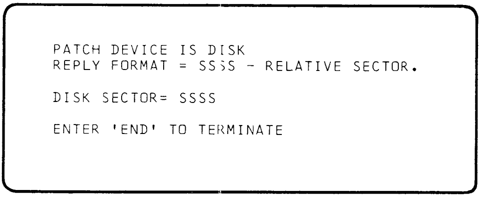
Note: The hex representation of only unprintable characters is to be displayed if HEX-NO. If HEX·YES, all characters will be displayed.

After the control statements or the command is entered, PATCH will prompt for the disk address of the sector to be patched. The sector address is of the (SS) type format and must be entered in hex characters. Four characters is the maximum, but less than four can be entered, since PATCH will right-justify and zero fill. To change the hex optionduring prompt, enter N or Y as the fifth character of the prompt. The address limits are as follows:

Disk (3.2 megabyte)—0001-3264
Disk (5.0 megabyte)—0001-4E84
Disk (9.1 megabyte)—0001-8D54
Disk (13.7 megabyte)—0001-D3E0
Diskette (128-byte format)—0001-079E
Diskette (512-byte format)—0001-026A

Figure 3-10 shows the prompt display.

Note: Alternate cylinder area on disk cannot be displayed.



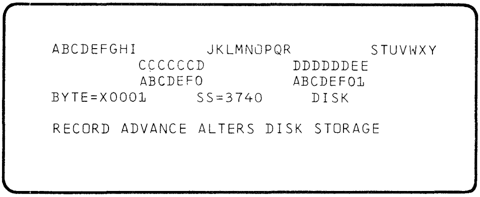
```
PATCH DEVICE IS DISK
REPLY FORMAT = SSSS - RELATIVE SECTOR.

DISK SECTOR= SSSS

ENTER 'END' TO TERMINATE
```

Figure 3-10. Patch Utility Prompt

The sector is then displayed, 40 bytes at a time, as shown in Figure 3-11, and patch data is entered using the keyboard data keys. After all changes have been made, and the sector is written back to the disk, the next sequential sector is displayed.

Each line in Figure 3-11 is as follows:

Line 1—Displayable EBCDIC characters
Line 2, 3—Hex representation (zone and digit), of the characters in line 1, displayed under the appropriate byte
Line 4—Used to display the cursor position and the current sector format and address
Line 6—User warning



```
ABCDEFGHI       JKLMNOPQR         STUVWXY
         CCCCCCD          DDDDDDEE
         ABCDEF0          ABCDEF01
BYTE=X0001     SS=3740      DISK

RECORD ADVANCE ALTERS DISK STORAGE
```

Note: Display shown with options INPUT = F1, HEX = NO.

Figure 3-11. Display Screen Format During Patch

Key functions during PATCH display are:

→ (l/c)—Increment the cursor position by one.
← (l/c)—Decrement the cursor position by one.
↑ (u/c)—Move cursor up a line.
↓ (u/c)—Move cursor down a line. Maximum is line 3.
← (u/c) SHIFT LEFT—Move the display one position to the left.
→ (u/c) SHIFT RIGHT—Move the display one position to the right.
↓ (l/c) ROLL↓—Display the previous sector (SS - 1).
↑ (l/c) ROLL↑—Display the next sector (SS + 1).
REC ADV—Write the sector back to disk with any alterations that might have been made. Display the next sector (SS + 1).
FLD ADV—Tab left (shift the display 40 bytes to the left until you reach the end of the sector).
REC BKSP—Reset the display to the record start.
FLD BKSP—Tab right (shift the display 40 bytes to the right).
ENTER—(ENTER, ENTER+, ENTER-) Prompt for new sector address. Data is not changed.
Data keys—All data keys will be accepted when the cursor is on line 1. Only hex characters (0-9, A-F) will be accepted when the cursor is on lines 2 or 3. When data is being entered on lines 2 or 3, the cursor action is different. Because the zone portion of the character is on line 2 and the digit portion on line 3, the cursor will perform a cursor down (↓) function when data is entered on line 2, and a cursor up (↑) and cursor right (→) function when data is entered on line 3. The exception is that when data is entered at the end of line 3, cursor advance stops.

Appendix A. IBM System/32 EBCDIC

Below are definitions of the column headings used in the following table.

Hex Value

Internal EBCDIC used by the system, expressed in hex notation.

Binary Value

Internal EBCDIC expressed in binary notation.

Printer Graphic

The graphic printed by this system for the EBCDIC shown. For example, graphics printed for the EBCDIC stored in the print data field (the field specified by the printer IOB as data to be printed) correspond to the binary values shown in the chart. Hence, a main storage value of hex 6C is printed as %.

Related Keyboard Key

This column specifies the key that must be pressed to send the associated character in EBCDIC to the keyboard/display screen IOB as a data byte. For example, pressing the ENTER key on the keyboard stores hex 10 in the keyboard/display screen IOB data byte.

Display Screen Graphic

This column shows the graphic that is displayed on the display screen for the associated main storage EBCDIC shown in the binary value column. For example, if the program issues a start keyboard/display screen IOB instruction and if the IOB specifies that a field in main storage be transferred to the keyboard/display screen buffer and subsequently displayed, the characters displayed for the EBCDIC binary values in the bytes moved from main storage correspond to the associated graphics shown in the display screen graphic column. A main storage value of hex 50 is displayed as &.

Communications Character

The system may display data entered from a diskette prepared on a 3741 with a communications feature. In such cases, the communications characters shown in this column correspond to the EBCDIC binary value on the table. For example, SYN is displayed as graphic 2, but is not printed (hex 32 has no printable graphic).

Special Keyboard Key

This column specifies the key that must be pressed to send the associated character in EBCDIC to the keyboard/display screen IOB as a data byte. Some of the keys, when used with the SHIFT key, produce a different character when the dual case feature is included. In some cases, no assignment of a related keyboard key is made for the special graphic characters as this key assignment changes as the keyboard is redefined.

Special Printer Graphic

This column shows the graphic printed by the system when equipped with dual case printing and the 96-character print belt. Certain characters are altered from the standard 48- and 64-character belts. For example, a greater-than symbol (>) (hex 2E) on the standard print belt prints as a degree symbol (°) on the special print belt.

Special Display Graphic

This column shows the graphic displayed by the system when equipped with dual case feature. Certain characters displayed as upper case characters are displayed as lower case with the feature. For example, the alphabet (A-Z), represented by hex 81-89, 91-99, and A2-A9, is displayed as lower case letters.

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character	Special Printer Graphic	Special Keyboard Key	Special Display Graphic
00	00000000							
01	00000001		INQ	A	SOH		INQ	a
02	00000010			B	STX			b
03	00000011			C	ETX			c
04	00000100			D				d
05	00000101			E				e
06	00000110		ERROR RESET	F			ERROR RESET	f
07	00000111			G				G
08	00001000			H				h
09	00001001			I				i
0A	00001010		↑	¢			↑	¢
0B	00001011		↓	.			↓	.
0C	00001100		←	≤			←	±
0D	00001101		→	(			→	(
0E	00001110			±				+
0F	00001111			|				[
10	00010000		ENTER	&	DLE		ENTER	&
11	00010001		ENTER +	J			ENTER +	j
12	00010010		ENTER –	K			ENTER –	k
13	00010011		FIELD ADV	L			FIELD ADV	l
14	00010100		REC ADV	M			REC ADV	m
15	00010101		FIELD BKSP	N			FIELD BKSP	n
16	00010110		REC BKSP	O			REC BKSP	o
17	00010111		DUP (SHIFT)	P			DUP (SHIFT)	p

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character	Special Printer Graphic	Special Keyboard Key	Special Display Graphic
18	00011000		DUP	Q			DUP	Q
19	00011001		ROLL↑	R			ROLL↑	r
1A	00011010		ROLL↓	!			ROLL↓	!
1B	00011011		← (SHIFT)	$			← (SHIFT)	$
1C	00011100		→ (SHIFT)	*			→ (SHIFT)	*
1D	00011101			)				)
1E	00011110			;			CODE	;
1F	00011111			¬	ITP			]
20	00100000			=				-
21	00100001			/				/
22	00100010			S				s
23	00100011			T				t
24	00100100			U				u
25	00100101			V				v
26	00100110			W	ETB			w
27	00100111			X				x
28	00101000			Y				y
29	00101001			Z				z
2A	00101010			\				§
2B	00101011			,				.
2C	00101100			%				%
2D	00101101			_	ENQ			_

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character	Special Printer Graphic	Special Keyboard Key	Special Display Graphic
2E	00101110			>				o
2F	00101111			?				?
30	00110000		CMD	0			CMD	2
31	00110001			1				3
32	00110010			2	SYN			½
33	00110011			3				ʘ
34	00110100			4				¼
35	00110101			5				®
36	00110110			6				£
37	00110111			7	EOT			↑
38	00111000			8				↓
39	00111001			9				¶
3A	00111010			:				:
3B	00111011			#				#
3C	00111100			@				@
3D	00111101			'	NAK			'
3E	00111110			=				=
3F	00111111			"				"
40	01000000	Blank	Space	Blank	Blank	Blank	Space	Blank
41	01000001			A				a
42	01000010			B				b
43	01000011			C				c

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character	Special Printer Graphic	Special Keyboard Key	Special Display Graphic
44	01000100			D				d
45	01000101			E				e
46	01000110			F				f
47	01000111			G				G
48	01001000			H				h
49	01001001			I				i
4A	01001010	¢	¢	¢	¢	¢	¢	¢
4B	01001011	.	.	.	.	.	.	.
4C	01001100	<	<	<	<	±	<	±
4D	01001101	(	(	(	(	(	'	(
4E	01001110	+	+	+	+	+	+	+
4F	01001111	|	|	|	|	[	!	[
50	01010000	&	&	&	&	&	&	&
51	01010001			J				j
52	01010010			K				k
53	01010011			L				l
54	01010100			M				m
55	01010101			N				n
56	01010110			O				o
57	01010111			P				p
58	01011000			Q				Q
59	01011001			R				r
5A	01011010	!	!	!	!	!	!	!
5B	01011011	$	$	$	$	$	$	$

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character	Special Printer Graphic	Special Keyboard Key	Special Display Graphic
5C	01011100	*	*	*	*	*	*	*
5D	01011101	)	)	)	)	)	)	)
5E	01011110	;	;	;	;	;	;	;
5F	01011111	¬	¬	¬	¬	]	¬	]
60	01100000	—	—	—	—	—	—	—
61	01100001	/	/	/	/	/	/	/
62	01100010			S				s
63	01100011			T				t
64	01100100			U				u
65	01100101			V				v
66	01100110			W				w
67	01100111			X				x
68	01101000			Y				y
69	01101001			Z				z
6A	01101010			\		§		§
6B	01101010	,	,	,	,	,	,	,
6C	01101100	%	%	%	%	%	%	%
6D	01101101	—	—	—	—	—	—	—
6E	01101110	>	>	>	>	o	>	o
6F	01101111	?	?	?	?	?	?	?
70	01110000			0	(70)			2
71	01110001			1				3
72	01110010			2				½
73	01110011			3				θ

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character	Special Printer Graphic	Special Keyboard Key	Special Display Graphic
74	01110100			4				¼
75	01110101			5		®		®
76	01110110			6		£		£
77	01110111			7				↑
78	01111000			8				↓
79	01111001			9		¶		¶
7A	01111010	:	:	:	:	:	:	:
7B	01111011	#	#	#	#	#	#	#
7C	01111100	@	@	@	@	@	@	@
7D	01111101	'	'	'	'	'	'	'
7E	01111110	=	=	=	=	=	=	=
7F	01111111	"	"	"	"	"	"	"
80	10000000			_		Blank		Blank
81	10000001			A		a	Nonshift A	a
82	10000010			B		b	Nonshift B	b
83	10000011			C		c	Nonshift C	c
84	10000100			D		d	Nonshift D	d
85	10000101			E		e	Nonshift E	e
86	10000110			F		f	Nonshift F	f
87	10000111			G		g	Nonshift G	G
88	10001000			H		h	Nonshift H	h
89	10001001			I		i	Nonshift I	i
8A	10001010			¢				¢
8B	10001011			.				.

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character	Special Printer Graphic	Special Keyboard Key	Special Display Graphic
8C	10001100			≤				±
8D	10001101			(				(
8E	10001110			+				+
8F	10001111			[				[
90	10010000			&				&
91	10010001			J		j	Nonshift J	j
92	10010010			K		k	Nonshift K	k
93	10010011			L		!	Nonshift L	i
94	10010100			M		m	Nonshift M	m
95	10010101			N		n	Nonshift N	n
96	10010110			O		o	Nonshift O	o
97	10010111			P		p	Nonshift P	p
98	10011000			Q		q	Nonshift q	Q
99	10011001			R		r	Nonshift r	r
9A	10011010			!				!
9B	10011011			$				$
9C	10011100			*				*
9D	10011101			)				)
9E	10011110			;				;
9F	10011111			_				]

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character	Special Printer Graphic	Special Keyboard Key	Special Display Graphic
A0	10100000			=̲				—
A1	10100001			/̲		¼		¼
A2	10100010			S̲		s	Nonshift S	s
A3	10100011			T̲		t	Nonshift T	t
A4	10100100			U̲		u	Nonshift U	u
A5	10100101			V̲		v	Nonshift V	v
A6	10100110			W̲		w	Nonshift W	w
A7	10100111			X̲		x	Nonshift X	x
A8	10101000			Y̲		y	Nonshift Y	y
A9	10101001			Z̲		z	Nonshift Z	z
AA	10101010			\̲				§
AB	10101011			,̲				,
AC	10101100			%̲				%
AD	10101101			=̲				—
AE	10101110			≥				o
AF	10101111			?̲				?
B0	10110000			0̲				^{2}
B1	10110001			1̲				^{3}
B2	10110010			2̲				½
B3	10110011			3̲				Θ

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character	Special Printer Graphic	Special Keyboard Key	Special Display Graphic
B4	10110100			4				¼
B5	10110101			5				®
B6	10110110			6				£
B7	10110111			7				↑
B8	10111000			8				↓
B9	10111001			9				¶
BA	10111010			:				:
BB	10111011			#				#
BC	10111100			@				@
BD	10111101			'				'
BE	10111110			=				=
BF	10111111			"				"
C0	11000000			Blank		2		2
C1	11000001	A	A	A	A	A	Shift A	A
C2	11000010	B	B	B	B	B	Shift B	B
C3	11000011	C	C	C	C	C	Shift C	C

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character	Special Printer Graphic	Special Keyboard Key	Special Display Graphic
C4	11000100	D	D	D	D	D	Shift D	D
C5	11000101	E	E	E	E	E	Shift E	E
C6	11000110	F	F	F	F	F	Shift F	F
C7	11000111	G	G	G	G	G	Shift G	G
C8	11001000	H	H	H	H	H	Shift H	H
C9	11001001	I	I	I	I	I	Shift I	I
CA	11001010			¢	¢			¢
CB	11001011			.	.			.
CC	11001100			<	<			<
CD	11001101			(	(			(
CE	11001110			+	+			+
CF	11001111			|	|			|
D0	11010000			&		3		3
D1	11010001	J	J	J	J	J	Shift J	J
D2	11010010	K	K	K	K	K	Shift K	K
D3	11010011	L	L	L	L	L	Shift L	L
D4	11010100	M	M	M	M	M	Shift M	M
D5	11010101	N	N	N	N	N	Shift N	N
D6	11010110	O	O	O	O	O	Shift O	O
D7	11010111	P	P	P	P	P	Shift P	P
D8	11011000	Q	Q	Q	Q	Q	Shift Q	Q
D9	11011001	R	R	R	R	R	Shift R	R
DA	11011010			!				!
DB	11011011			$				$

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character	Special Printer Graphic	Special Keyboard Key	Special Display Graphic
DC	11011100			*				*
DD	11011101			)				)
DE	11011110			;				;
DF	11011111			¬				¬
E0	11100000		\	—		½		½
E1	11100001			/				/
E2	11100010	S	S	S	S	S	Shift S	S
E3	11100011	T	T	T	T	T	Shift T	T
E4	11100100	U	U	U	U	U	Shift U	U
E5	11100101	V	V	V	V	V	Shift V	V
E6	11100110	W	W	W	W	W	Shift W	W
E7	11100111	X	X	X	X	X	Shift X	X
E8	11101000	Y	Y	Y	Y	Y	Shift Y	Y
E9	11101001	Z	Z	Z	Z	Z	Shift Z	Z
EA	11101010			\				\
EB	11101011			,				,
EC	11101100			%				%
ED	11101101			_				_
EE	11101110			>				>
EF	11101111			?				?
F0	11110000	0	0	0	0	0	Nonshift 0	0
F1	11110001	1	1	1	1	1	Nonshift 1	1
F2	11110010	2	2	2	2	2	Nonshift 2	2
F3	11110011	3	3	3	3	3	Nonshift 3	3

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character	Special Printer Graphic	Special Keyboard Key	Special Display Graphic
F4	11110100	4	4	4	4	4	Nonshift 4	4
F5	11110101	5	5	5	5	5	Nonshift 5	5
F6	11110110	6	6	6	6	6	Nonshift 6	6
F7	11110111	7	7	7	7	7	Nonshift 7	7
F8	11111000	8	8	8	8	8	Nonshift 8	8
F9	11111001	9	9	9	9	9	Nonshift 9	9
FA	11111010			:				:
FB	11111011			#				#
FC	11111100			@				@
FD	11111101			,				,
FE	11111110			=				=
FF	11111111			"				"

MAIN STORAGE INSTRUCTION FORMATS

Op Code (one byte)																	Q-Code	Operands		Total Instr Length	Summary			
Bits 0-3	Bits 4-7																One Byte	First	Second		Op	Q	Operand	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F								
0	MVN				ZAZ		AZ	SZ	MVX		ED	ITC	MVC	CLC	ALC	SLC		2 Bytes Direct	2 Bytes Direct	6				
1	MVN				ZAZ		AZ	SZ	MVX		ED	ITC	MVC	CLC	ALC	SLC			1 Byte Disp Index-By XR1	5				X1
2	MVN				ZAZ		AZ	SZ	MVX		ED	ITC	MVC	CLC	ALC	SLC			1 Byte Disp Index-By XR2	5				X2
3	SNS	LIO			ST	L	A		TBN	TBF	SBN	SBF	MVI	CLI						4				
4	MVN				ZAZ		AZ	SZ	MVX		ED	ITC	MVC	CLC	ALC	SLC		1 Byte Displacement Indexed By XR1	2 Bytes Direct	5			X1	
5	MVN				ZAZ		AZ	SZ	MVX		ED	ITC	MVC	CLC	ALC	SLC			1 Byte Disp Index-By XR1	4			X1	X1
6	MVN				ZAZ		AZ	SZ	MVX		ED	ITC	MVC	CLC	ALC	SLC			1 Byte Disp Index-By XR2	4			X1	X2
7	SNS	LIO			ST	L	A		TBN	TBF	SBN	SBF	MVI	CLI						3			X1	
8	MVN				ZAZ		AZ	SZ	MVX		ED	ITC	MVC	CLC	ALC	SLC		1 Byte Displacement Indexed By XR2	2 Bytes Direct	5			X2	
9	MVN				ZAZ		AZ	SZ	MVX		ED	ITC	MVC	CLC	ALC	SLC			1 Byte Disp Index-By XR1	4			X2	X1
A	MVN				ZAZ		AZ	SZ	MVX		ED	ITC	MVC	CLC	ALC	SLC			1 Byte Disp Index-By XR2	4			X2	X2
B	SNS	LIO			ST	L	A		TBN	TBF	SBN	SBF	MVI	CLI						3			X2	
C	BC	TIO	LA																2 Bytes Direct	4				
D	BC	TIO	LA																1 Byte Disp Index-By XR1	3			X1	
E	BC	TIO	LA																1 Byte Disp Index-By XR2	3			X2	
F	HPL	APL	JC	SIO	SVC	XFER														3				

[FE 120855]

START I/O (SIO) INSTRUCTION FORMATS

Op Code	Q-Byte			Control Code	Description
	DA	M	N[1]		
0 7	8 11	12	13 15	16 23	
F3	1110				Device address—printer (E)
		0	000	0000 0000	(9 for serial printer single for forms/ledger cards mode only) Do IOB as specified in flag byte
Printer					
	0001				Device address—display screen (1)
Display Screen		0	000	Unused	Do IOB as specified by flag byte. The keyboard will be enabled if flag bits 0 and 1 are not both zero.
			100		Disables the keyboard. The keyboard is enabled by a do IOB SIO with flag bits 0 and 1 not both zero. ERROR RESET, INQ, and the printer control keys remain enabled.
			110		Sets enable/disable bits and resets the keyboard; executes current IOB. Need error reset key to reenable the keyboard if flag bits 0 and 1 are both off.
	1101				Device address—diskette (D)
		0	000	0000 0000	Control-seek
			001		Read-data
Diskette	(see note 1)		010		Read-ID
			100		Read-data and control record
			101		Write-data and verify
			110		Write-CAM and verify
			111		Write-ID and verify

[1] N codes not shown are invalid.

Notes:

1. Control storage transient area must be loaded prior to execution.
2. All other Q-codes are invalid.
3. Bits 2, 3, 4, 6, 7 are not used, but should be 0. If bit 0 is 0, bit 1 is not examined.

START I/O (SIO) INSTRUCTION FORMATS (continued)

Op Code	Q-Byte			Control Code	Description
	DA	M	N[1]		
0 7	8 11	12	13 15	16 23	
F3	1000	0	000		Control BSCA
				0000 0000	Cancel 2-second time-out
				0000 0100	Start 2-second time-out
BSCA				1000 0000	Disable BSCA and cancel 2-second time-out
				1000 0100	Disable BSCA and start 2-second time-out
				1100 0000	Enable BSCA and cancel 2-second time-out
				1100 0100	Enable BSCA and start 2-second time-out
			011	1100 0000	Receive initial
			010	1100 0000	Transmit and receive
			001	1100 0000	Receive only
MCU	0101	0	000	0000 0000	Start mag card unit
F3	1000	0	000		Control SDLC
				1000 0000	Disable SDLC
SDLC				1100 0000	Enable SDLC
			011	0000 0000	Receive initial
			010	0000 0000	Transmit and receive
			001	0000 0000	Receive only
			100	0000 0000	Transmit final
			101	0000 0000	Transmit only
F3 Data Recorder	0101	0	000		Start data recorder IOB
F3 Magnetic Character Reader	0101	0	000	0000 0000	Start controller
				0000 0001	Reset controller
				1111 1111	Stop controller

[1] N codes not shown are invalid.

Notes:

1. Control storage transient area must be loaded prior to execution.
2. All other Q-codes are invalid.
3. Bits 2, 3, 4, 6, 7 are not used, but should be 0. If bit 0 is 0, bit 1 is not examined.

LOAD I/O (LIO) INSTRUCTION FORMATS

Op Code	Q-Byte			Control Code	Description
	DA	M	N[1]		
0 7	8 11	12	13 15	16 23	
31					Direct addressing
71					Indexed by XR1
B1					Indexed by XR2
	1110				Device address—printer (E)
		0	000		Byte 2—forms length; byte 1—current line
Printer			010		Byte 1—character set size; byte 2 not used (no-op for serial printer)
			100		Line printer image address register (no-op for serial printer)
			110		Line printer data address register
					Storage address can be one or two bytes long, addressed by rightmost byte
	0001				Device address—display screen (1)
		0	000		Address is indirect starting address of display screen IOB
Display Screen			011		Address is the indirect address of the system interrupt address register for display screen
			101		Address of keyboard redefine table
			110		Set lower case
			111		Set upper case
31					Direct addressing
71					Indexed by XR1
B1					Indexed by XR2
	1101				Device address—diskette (D)
		0	000		Load control address register (CAR)
Diskette			001		Load data field address register (DAR)
					Storage address can be one or two bytes long, addressed by leftmost byte

[1] N codes not shown are invalid.

LOAD I/O (LIO) INSTRUCTION FORMATS (continued)

Op Code	Q-Byte			Control Code	Description
	DA	M	N[1]		
0 7	8 11	12	13 15	16 23	
31					Direct addressing
71					Indexed by XR1
B1					Indexed by XR2
	1000	0	101		Load unit definition table register
			100		Load BSCA current address register
			011		Load BSCA interrupt address register
BSCA			001		Load BSCA stop address register
			010		Load BSCA transition address register
31					Direct addressing
71					Indexed by XR1
B1					Indexed by XR2
	1000	0	000		Receive buffer addresses
			010		UDT
SDLC			011		Interrupt address
			100		Transmit buffer addresses

[1] N codes not shown are invalid.

SENSE I/O (SNS) INSTRUCTION FORMATS

Op Code	Q-Byte			Description	
	DA	M	N[1]		
0 7	8 11	12	13 15		
30				Direct addressing	
70				Indexed by XR1	
B0				Indexed by XR2	
	1110			Device address—printer (E)	
		0		Must be 0	
				Low Address	**High Address**
			000	Forms length	Current line
			011	*Status Byte 0:*	*Status Byte 1:*
				0 – Coil 1 parity odd	0 – Forms jam check
Printer				1 – Coil 2 parity odd	1 – Belt speed check
				2 – Coil 3 parity odd	2 – Carriage sync check
				3 – Emitter check	3 – End of forms
				4 – Buffer data check	4 – Throat open
				5 – Hammer parity check	5 – Coil current check
				6 – No op	6 – Belt sync check
				7 – Unprintable character	7 – Cover open
			100	*Status Byte 2:*	*Status Byte 3:*
				0 – Printer version (off = line printer on = serial printer)	Not used
				1-4 – Not used	
				5 – Forms mode (off = continuous forms on = single form/ledger cards)	
				6-7 – Not used	
Data and Address Switches	0000	0	000	Switches 1 and 2	Switches 3 and 4
				The storage address can be one or two bytes long, addressed by rightmost byte	

[1] N codes not shown are invalid.

SENSE I/O (SNS) INSTRUCTION FORMATS (continued)

Op Code	Q-Byte			Description	
	DA	M	N[1]		
0 7	8 11	12	13 15		
30				Direct addressing	
70				Indexed by XR1	
B0				Indexed by XR2	
	1101			Device address—diskette (D)	
		0	000	Sense control address register (CAR)	
			001	Sense data field address register (DAR)	
			010	**Low Address**	**High Address**
				Sense Byte 0:	*Sense Byte 1:*
				0 – Missing data AM	0 – No-op
				1 – ID CRC	1 – Invalid control record
				2 – Data CRC	2 – Control error
				3 – Cylinder mismatch	3 – Control AM record found
				4 – Head mismatch	4 – Cylinder address invalid
				5 – Record mismatch	5 – Write error
				6 – Length mismatch	6 – Reserved
				7 – Reserved	7 – Reserved
Diskette					
			011	*Sense Byte 2:*	*Sense Byte 3:*
				0 – Diskette fast	0 – Head loaded
				1 – Not ready	1 – Low write current to diskette is set
				2 – Not applicable	2 – Write gate to diskette is on
				3 – No orient	3 – Erase gate to diskette is on
				4 – Read overrun	4 – Seek to track 3 or 0 is on
				5 – Not applicable	5 – Seek to track 0 or 1 is on
				6 – Write overrun	6 – Seek to track 1 or 2 is on
				7 – Write parity check	7 – Seek to track 2 or 3 is on
				The storage address can be one or two bytes long, addressed by rightmost byte.	

[1] N codes not shown are invalid.

SENSE I/O (SNS) INSTRUCTION FORMATS (continued)

Op Code	Q-Byte			Description
	DA	M	N[1]	
0 7	8 11	12	13 15	
30				Direct addressing
70				Indexed by XR1
B0				Indexed by XR2
	1000	0		
BSCA			100	Sense BSCA current address
			011	Sense BSCA status
30				Direct addressing
70				Indexed by XR1
B0				Indexed by XR2
	1000	0	000	Sense SDLC completion table
SDLC				

[1] N codes not shown are invalid.

Note: This instruction causes 4 bytes of information to be placed into the main storage data area. The first 2 bytes are the address + 1 of the last byte transmitted or received and 2 bytes of status information.

Additional BSCA Instructions

X'C0870004'—Branch to main storage location X'0004'

RIB X'0F'

Inline parameter list:

X'000400'—Initialize BSCA for ASCII mode (see notes)
X'000500'—Initialize BSCA for EBCDIC mode (see notes)
X'000B00'—BSCA wrap test

RIB X'0E'

Inline parameter list:

X'xx0200'—Queue/dequeue BSCA IOB

↓↓

00	—Load BSCA IOB in last position on system IOB queue
10	—Load BSCA IOB in first position on system IOB queue
01	—Remove BSCA IOB from system IOB queue

Notes:

1. BSCA must be initialized to ASCII or EBCDIC mode for each job using BSCA.
2. See Figure 2-36 for other inline parameter lists for load control storage.

Additional MCU Instructions

X'C0870004'—Branch to main storage location X'0004'

RIB X'0F'

Inline parameter list:

X'000200'—Initialize mag card unit

RIB X'0E'

Inline parameter list:

X'xx0800'—Queue/dequeue mag card unit IOB

↓↓

00	—Load MCU IOB in last position on system IOB queue
10	—Load MCU IOB in first position on system IOB queue
01	—Remove MCU IOB from system IOB queue

Note: See Figure 2-36 for other inline parameter lists for load control storage.

TEST I/O AND BRANCH (TIO) INSTRUCTION FORMATS

Op Code	Q-Byte			Control Code	Description
	DA	M	N[1]		
0 7	8 11	12	13 15	16 23	
C1					Direct addressing
D1					Indexed by XR1
E1					Indexed by XR2
	1101				Device address—diskette (D)
		0	000		Not ready/unit check
Diskette	(see note)			xxxx xxxx	Branch to address if condition met; op codes D1 and E1 are indexed

[1] N codes not shown are invalid.

Note: Control store transient area must be loaded prior to execution.

SUPERVISOR CALL (SVC) INSTRUCTION FORMAT

Op Code	Q-Byte[1]			Control Code	Description
	DA	M	N		
0 7	8 11	12	13 15	16 23	
F4					
	0000	0	010		Wait service X‘000C’
			011		Disk IOS X‘0008’
			110		Dump X‘0000’
SVC			111	0000 0000	General exit transient X‘0012’
		1	000		General entry (must be followed by an RIB) X‘0004’
					Note: This instruction can only be used in the first 256 bytes of main storage

[1] All Q-bytes are valid, but only those shown are currently used. (See Figure E-1 for a definition of these Q-bytes.)

TRANSFER (XFER) INSTRUCTION FORMAT

Op Code	**Q-Byte**			**Control Code**	**Description**
	DA	**M**	**N**		
0 7	**8 11**	**12**	**13 15**	**16 23**	
F5	0000	0	000		Requests ECS function
				0000 0000	Allocates and loads the microcode in the available area of the control storage increment
				0000 0001	Deallocates and resets the microcode in the available area of the control storage increment
	0000	0	001		Requests SIS interpreter
				0000 0001	Initiate main program execution
				0000 0010	Initiate subprogram execution
				0000 0011	Reenter user program after CALL operation
				0000 0100	Subroutine return to calling module
				0000 0101	Execute next scientific instruction
				0000 0111	Execute next scientific instruction after INVOKE scientific instruction
				0000 1000	Copy scientific registers to main storage
				0000 1001	Copy scientific registers from main storage
				0000 1010	Copy scientific instruction interpreter to double mode
				0000 1011	Switch scientific instruction interpreter to real mode
				0000 1100	Perform fixed to floating point conversion
				0000 1101	Perform Real*8 floating point to fixed conversion
				0000 1110	Perform Real*4 floating point to fixed conversion

Note: Other values for Q-byte and control code are invalid.

BRANCH ON CONDITION (BC) INSTRUCTION FORMAT

Op Code	**Q-Byte (binary)**			**Description**		
0 7	**DA** **8 11**	**M** **12**	**N** **13 15**			
C0				Direct addressing		
D0				Indexed by XR1		
E0				Indexed by XR2		
				Condition Tested	**Bit to be Tested in Program Status Register**	**Effect on Bit Tested**
	1xxx			Condition true	–	–
	0xxx			Condition false	–	–
	x0xx			Not used; must be zero	–	–
	xx1x			Binary overflow	2	None
	xxx1			False	3	Bit turned off, if condition tested
		1		Decimal overflow	4	Bit turned off, if condition tested
			1xx	High	5	None
			x1x	Low	6	None
			xx1	Equal	7	None
				Note: Q-bytes 80, x7, or xF (where x = 0–7) cause a branch to no-op; Q-bytes 00, x7, or xF (where x = 8–F) cause an unconditional branch. Displacement is added to the index register (selected by op code) to get the branch-to address.		

Note: ARR is not changed if branch is not taken.

JUMP ON CONDITION (JC) INSTRUCTION FORMAT

Op Code	Q-Byte (binary)				Description		
	DA	M	N				
0 7	8 11	12	13 15	16 23			
F2					Condition Tested	Bit to be Tested in Program Status Register	Effect on Bit Tested
	1xxx				Condition true	–	–
	0xxx				Condition false	–	–
	x0xx				Not used; must be zero	–	–
	xx1x				Binary overflow	2	None
	xxx1				False	3	Bit turned off, if condition tested
		1			Decimal overflow	4	Bit turned off, if condition tested
			1xx		High	5	None
			x1x		Low	6	None
			xx1		Equal	7	None
					Note: Q-bytes 80, x7, or xF (where x = 0–7) cause a branch to no-op; Q-bytes 00, x7, or xF (where x = 8–F) cause an unconditional branch.		
				xxxx xxxx	Displacement added to address in IAR to get jump-to address		

Appendix C. Main Storage Instruction Descriptions

Mnemonic	Instruction	Description
ALC	Add logical characters Op codes: 0E, 1E, 2E, 4E, 5E, 6E, 8E, 9E, AE	Positive binary number in second operand is added byte by byte to positive binary number in first operand; result stored in first operand. Operands addressed by rightmost byte. Q-byte specifies one less than length of operands; both operands must be same length. Second operand is not changed unless it overlaps first operand. Result sets condition register.
A	Add to register Op codes: 36, 76, B6	Positive binary number contained in first operand address added to contents of 2-byte register selected by the Q-byte; result replaces contents of register. Operand is addressed by rightmost byte. Q-byte indicates the following: X'01' = XR1 X'02' = XR2 X'04' = PSR X'08' = ARR X'10' = IAR X'20' = IAR X'40' = No-op X'80' = No-op *Note:* Must not be used to add to more than one register at a time. Result sets condition register. Operands remains unchanged.
AZ	Add zoned decimal Op codes: 06, 16, 26, 46, 56, 66, 86, 96, A6	Second operand added algebraically to first operand; result placed in first operand. Operands addressed by rightmost bytes. Zone bits of result, except rightmost byte set to 1's; zone bits of rightmost byte set to 1's if the result is positive or zero or to 1101 if the result is negative. Q-byte designates length of both operands; the high 4 bits is the length of the first operand minus the length of the second operand, and the low 4 bits is the length minus one of the second operand. Second operand remains unchanged unless overlapped.

Mnemonic	Instruction	Description
AZ (cont.)		No check is made for valid digits in operands. Result sets condition register.
	Advance program level	Executes as a no-op.
APL	Op code: F1	—
	Branch on condition	Condition register is tested for conditions specified by the Q-byte.
BC	Op codes: C0, D0, E0	If condition register satisfies condition(s) tested for, the next instruction is taken from the branch address. The test false and decimal overflow bits are turned off (if tested). Otherwise, the condition register remains unchanged.
	Compare logical characters	First operand is compared byte by byte to second operand; result sets condition register.
CLC	Op codes: 0D, 1D, 2D, 4D, 5D, 6D, 8D, 9D, AD	Operands are addressed by rightmost byte. Q-byte specifies one less than length of operands; both operands must be same length. Neither operand is changed.
	Compare logical immediate	The Q-byte is compared with the operand in storage location of operand address; result sets condition register.
CLI	Op codes: 3D, 7D, BD	Neither operand is changed.
	Edit	Decimal numeric characters in second operand replace bytes containing X'20' in first operand.
ED	Op codes: 0A, 1A, 2A, 4A, 5A, 6A, 8A, 9A, AA	Second operand remains unchanged. Operands are addressed by rightmost bytes. Q-byte specifies one less than length of first operand; second operand contains same number of bytes as X'20's in first operand. Condition register is not affected.
	Halt program level	Executes as a no-op.
HPL	Op code: F0	—
	Insert and test characters	Single character at second operand address replaces all characters to the left of the first significant digit in first operand.

Mnemonic	Instruction	Description
ITC	Op codes: 0B, 1B, 2B, 4B, 5B, 6B, 8B, 9B, AB	First operand is addressed by leftmost byte that can contain a character that should be replaced. Q-byte contains length minus one of first operand. After execution, the ARR contains the address of the first significant digit in first operand, or the address of the byte to the right of the first operand if there are no significant digits. Second operand remains unchanged. Condition register is not affected.
JC	Jump on condition Op code: F2	Condition register is tested for conditions specified by Q-byte. If condition register satisfies the condition(s) tested for, the control code is added to the IAR and the sum becomes the address of the next instruction. The test false and decimal overflow bits are turned off (if tested). Otherwise, the condition register remains unchanged.
LA	Load address Op codes: C2, D2, E2	If op code is D2 or E2, 1-byte operand is added to contents of index register specified by operand; result is loaded into register specified by Q-byte. If op code is C2, operand is loaded into register specified by Q-byte. Q-byte indicates the following: X'01' = XR1 X'02' = XR2 *Note:* Not to be used for loading both registers at the same time. Operand remains unchanged. Condition register remains unchanged.
LIO	Load I/O Op codes: 31, 71, B1	The contents of the 2 bytes addressed by the operand are transferred to the destination specified by the N-code of the Q-byte. A Q-byte of X'00' results in a no-op. Condition register is not affected.

Mnemonic	Instruction	Description
L	Load register Op codes: 35, 75, B5	Contents of 2-byte field addressed by operand are placed in the 2-byte register specified by the Q-byte. Operand is addressed by rightmost byte. The condition register is not affected unless the PSR is the register being loaded. Q-byte indicates the following: X'01' = XR1 X'02' = XR2 X'04' = PSR X'08' = ARR X'10' = IAR X'20' = IAR X'40' = No-op X'80' = No-op *Note:* Must not be used to load more than one register at a time. Operand remains unchanged.
MVC	Move characters Op codes: 0C, 1C, 2C, 4C, 5C, 6C, 8C, 9C, AC	Second operand is placed byte by byte in first operand location. Operands are addressed by rightmost byte. Q-byte specifies length of operands; both operands must be the same length. Condition register is not affected.
MVX	Move hexadecimal character Op codes: 08, 18, 28, 48, 58, 68, 88, 98, A8	Numeric portion or zone portion of single byte second operand is placed in numeric portion or zone portion of first operand. Q-byte specifies portion of each operand: X'00' = Zone to zone X'01' = Numeric to zone X'02' = Zone to numeric X'03' = Numeric to numeric Condition register is not affected. Second operand remains unchanged.

Mnemonic	Instruction	Description
MVN	Move inverse[1] Op codes: 10, 20, 40, 50, 60, 80, 90, A0	Second operand placed byte by byte in reverse order in first operand. Second operand addressed by rightmost byte; first operand addressed by leftmost byte. Second operand remains unchanged. Q-byte specifies length minus one of operands; both operands must be the same length. Condition register is not affected.
MVI	Move logical immediate Op codes: 3C, 7C, BC	Data contained in Q-byte moved to byte located in operand address. Condition register is not affected.
SNS	Sense I/O Op codes: 30, 70, B0	Contents of data source specified by N portion of Q-byte placed in a 2-byte field specified by the operand address. Condition register is not affected.
SBF	Set bits off masked Op codes: 3B, 7B, BB	Byte of data in Q-byte is used to set off corresponding bits of byte located at operand address. Bits on in Q-byte set bits off in operand; other operand bits remain unchanged. Condition register is not affected.
SBN	Set bits on masked Op codes: 3A, 7A, BA	Byte of data in Q-byte is used to set on corresponding bits in byte located at operand address. Bits on in Q-byte set bits on in operand; other operand bits remain unchanged. Condition register is not affected.
SIO	Start I/O Op code: F3	No-op, if unit check condition exists in addressed device that prevents execution of this instruction If a unit check condition exists that does not prevent execution of this instruction, it is executed and unit check status is reset. If instruction addresses a device that is busy, a test for busy loop results. Condition register is not affected.

[1] Move inverse supported for World Trade only.

Mnemonic	Instruction	Description
ST	Store register Op codes: 34, 74, B4	Contents of 2-byte register specified by Q-byte placed in 2-byte field addressed by operand. Operand is addressed by rightmost byte. Register remains unchanged. Q-byte indicates the following: X'01' = XR1 X'02' = XR2 X'04' = PSR X'08' = ARR X'10' = IAR X'20' = IAR X'40' = No-op X'80' = No-op *Note:* Must not be used to store more than one register at a time. Condition register is not affected.
SLC	Subtract logical characters Op codes: 0F, 1F, 2F, 4F, 5F, 6F, 8F, 9F, AF	Positive binary number in second operand subtracted from positive binary number in first operand; result stored in first operand. Operands addressed by rightmost byte. Result can never be negative. Q-byte specifies one less than length of operands; both operands must be the same length. Second operand is not changed unless it overlaps the first operand. Result sets condition register.
SZ	Subtract zoned decimal Op codes: 07, 17, 27, 47, 57, 67, 87, 97, A7	Second operand subtracted algebraically from first operand byte by byte; result placed in first operand. Operands addressed by rightmost byte. Zone bits of result, except rightmost byte set to 1's; zone bits of rightmost byte set to 1's if the result is positive or zero, or to 1101 if the result is negative. Q-byte designates length of both operands. The high 4 bits is the length of the first operand minus the length of the second operand, and the low 4 bits is the length minus one of the second operand. Second operand remains unchanged unless overlapped.

Mnemonic	Instruction	Description
SZ (cont.)		No check is made for valid decimal digits. Result sets condition register.
SVC	Supervisor call Op code: F4	Control is passed to the control storage nucleus. Function performed is defined by Q-byte in conjunction with request indicator byte (RIB). Can only be issued from first 256 bytes of main storage. Return is made to address in: ARR, ARR+1, or ARR+4. Condition register is not affected.
TBF	Test bits off masked Op codes: 39, 79, B9	Byte of data contained in Q-byte is used to test for bits off in corresponding bits in the byte located at operand address. Bits on in the Q-byte test bits in operand; other operand bits remain untested. Operand remains unchanged. Result sets condition register.
TBN	Test bits on masked Op codes: 38, 78, B8	Byte of data contained in Q-byte is used to test for bits on in corresponding bits in the byte located at operand address. Bits on in the Q-byte test.bits in operand; other operand bits remain untested. Operand remains unchanged. Result sets condition register.
TIO	Test I/O and branch Op codes: C1, D1, E1	Condition specified by the Q-byte is tested in the addressed device. If condition is present, branch-to address is transferred to IAR. If condition is not present, branch to address is transferred to ARR (no branch taken). Condition register is not affected.
XFER	Transfer control Op code: F5	1. Control passed to extended control storage supervisor 2. Function performed defined by Q-byte in conjunction with instruction R-byte. 3. Control returns to the byte following the XFER plus the number of bytes in the in-line parameter list.

Mnemonic	Instruction	Description
ZAZ	Zero and add zoned Op codes: 04, 14, 24, 44, 54, 64, 84, 94, A4	Second operand placed byte by byte in first operand. High order zeros inserted in first operand if necessary. Zone bits of result, except rightmost byte set to 1's; zone bits of rightmost byte set to 1's if the result is positive or zero, or to 1101 if the result is negative. Operands addressed by rightmost byte. Q-byte designates length of both operands. The high 4 bits is the length of the first operand minus the length of the second operand, and the low 4 bits is the length minus one of the second operand. The second operand remains unchanged unless overlapped. No check is made for valid digits in operands. Result sets condition register.

Appendix D. Micro Instruction Formats

MICRO INSTRUCTION MNEMONICS

Op Code	Micro Instruction	Mnemonic
0	Branch	B
1	Branch and Link	BAL
2	Branch on Condition	JC
	Branch on Carry	JCY
	Branch on High	JH
	Branch on Low	JL
	Branch on Equal	JE
	Branch on Positive	JP
	Branch on Negative	JN
	Branch on Mixed	JM
	Branch on Zero	JZ
	Branch all Ones	JO
	Branch on Flag	JFLG
	Branch Service Request	JSR
	Branch not High	JNH
	Branch not Low	JNL
	Branch not Equal	JNE
	Branch not Positive	JNP
	Branch not Negative	JNN
	Branch not Zero	JNZ
	Return	RETRN
3	Branch on I/O Condition	JIO
	I/O Load from Control Storage High/Low	WTCH/L
	I/O Store to Control Storage High/Low	RDCH/L
	I/O Load from Main Storage	WTM
4	I/O Store to Main Storage	RDM
	Load from Control Storage	LC
	Load from Main Storage	LM
	Store to Control Storage	STC
	Store to Main Storage	STM
5	Test Mask	TM
6	Logical/Arithmetic 1	LA1
7	Logical/Arithmetic 2	LA2
Used with	Exclusive OR	XR
L/A 1 and	OR	OR
L/A 2	AND Complement	NCR
	AND	NR
	OR Complement	OCR
	Decrement Register by 1	DEC
	Add Registers with Carry	ACYR
	Subtract Registers	SR
	Add Registers	AR
	Subtract with Borrow	SCYR
	Increment Register by 1	INC
	Shift left logical	SLL
	Shift left logical double	SLLD
8	Set Bits Off	SBF
9	Set Bits On	SBN
A	Load Immediate	LI

Op Code	Micro Instruction	Mnemonic
B	Sense Interrupt Level Status Byte	SILSB
	I/O Load	IOL
	I/O Sense	IOS
	I/O Control Load	IOCL
	I/O Control Sense	IOCS
	Microprocessor Load	MPL
	Microprocessor Sense	MPS
	Microprocessor Load for Special Functions	MPLF
C	Compare Immediate	CI
D	Subtract Immediate	SI
	Add Immediate	AI
E	Move LSR	MVR
	Load Direct from Control Storage	L
	Store Direct from Control Storage	ST
F	Hex Branch Numeric	HBN
	Hex Branch Zone	HBZ
	Shift Right Logical	SRL
	Shift Right Logical Double	SRLD
	Move Zone to Numeric	MZN
	Move Zone to Zone	MZZ

Branch

Mnemonic: B

0 0 0 0	Branch Address
0 3	4 15

The branch instruction is used for an unconditional branch. It allows branching to one of 4096 addresses in control storage.

Branch and Link

Mnemonic: BAL

0 0 0 1	Branch Address
0 3	4 15

This instruction is used for an unconditional branch and link operation. It allows branching to one of 4096 addresses in control storage.

Branch on Condition

Mnemonic: JC

0 0 1 0	Condition	Page Address
0 3	4 7	8 15

This instruction allows branching within 256 locations if the condition specified by bits 4-7 is met:

Bits 4-7	Mnemonic	Test Condition
0000	JCY	Carry
0001	JH	High (condition code bit 5)
0010	JL	Low (condition code bit 6)
0011	JE	Equal (condition code bit 7)
0100	JP	Positive (condition code bit 1)
0100	JO	All 1's (condition code bit 1)
0101	JN	Negative (condition code bit 2)
0101	JM	Mixed (condition code bit 2)
0110	JZ	Zero (condition code bit 3)
0111	JFLG	Flag (condition code bit 0)
1000	JSR	Service request
1001	JNH	Not high
1010	JNL	Not low
1011	JNE	Not equal
1100	JNP	Not positive
1101	JNN	Not negative
1110	JNZ	Not zero
1111	RETRN	Return

Note: See Appendix J for a description of the condition register (PCR).

Branch on I/O Condition

Mnemonic: JIO

0 0 1 1	Mod	Page Address
0 3	4 7	8 15

This instruction tests I/O conditions and does a branch depending on the results.

The device address is contained in work register 0, low and is used in conjunction with the modifier (bits 4 through 7) to test the following conditions:

10 = Keyboard
Bits 4–7 = 0000 — Jump overrun
0100 — Jump micro interrupt enable
0101 — Diagnostic jump true
0110 — Diagnostic jump false

40 = Display screen
Bits 4–7 = 0101 — Jump diagnostic false (used only in diagnostic mode)
0110 — Jump diagnostic true (used only in diagnostic mode)
1000 — Jump invalid buffer

50 = Magnetic character reader
Bits 4-7 = 0000 — Controller check
0001 — Not controller check
0100 — Diagnostic true
0110 — Diagnostic false

A0 = Disk
Bits 4–7 = 0000 — Adapter check/not ready file[1]
0001 — (Not) file ready interface[2]
0010 — Seek busy
0011 — File busy
0100 — Any interrupt enabled
0101 — Diagnostic true
0110 — Diagnostic false
0111 — File home
1000 — Seek complete interface
1001 — Data unsafe interface
1010 — Scan hit LA
1011 — Select last head
1101 — Sector hit LA
1111 — Index pulse latch off

[1] Attachment equipment check
Parallel parity check
Channel overrun
Serdes check
Interrupt time-out
PLO out of sync
Channel transfer error
Sector sync check
Off track
CRC
Write echo check
Invalid seek
Not ready

[2] File not ready
Unsafe

D0 = Diskette
Bits 4–7 = 0000 – Diskette drive not ready, or diskette error
0001 – AM not found
0010 – CRC not zero
0011 – Index pulse off
0101 – Jump I/O true (test 1)
0110 – Jump I/O true (test 2)
0111 – Jump I/O true (test 3)
1000 – Set load head latch
1001 – Set low write current
1010 – Set erase gate
1011 – Set ID orientation
1100 – Reset error status
1110 – Reset erase gate
1111 – Reset sector operation

E0 = Printer
Bits 4–7 = 0000 – Adapter check[1]
0001 – Not ready[2]
0010 – Carriage busy
0011 – Print busy
0100 – Interrupt enabled
0101 – Diagnostic true (diagnostic use only)
0110 – Diagnostic false (diagnostic use only)
1000 – Elapsed time counter busy (line)
1001 – Motor not up to speed (line)
1010 – Even scans selected (line)/not Katakana character set (serial)
1011 – Subscan reset on (line)
1100 – New adapter control card
1111 – Serial printer attached

[1]

Line printer	**Serial printer**
Belt speed check	Forms hung
Belt sync check	Horizontal check
Hammer power down	Forms runaway
Carriage sync check	Memory parity check
Emitter check	Unprintable character check
Data check	Wire check
Echo check	
Forms check	Emitter check

[2]

Line printer	**Serial printer**
Cover open	End of forms
Throat open	Printer not ready
End of forms	Wire check

Logical/Arithmetic 1

Mnemonic: LA1

0 1 1 0	H1	Reg 1	Function	H2	Reg 2
0 3	4	5 7	8 11	12	13 15

This instruction allows arithmetic and logical functions to be processed through the System/32 ALU. The Logical/Arithmetic 1 instruction is for 1-byte operations only.

The logical/arithmetic functions that can be performed are specified by bits 8–11 as follows:

Bits 8	9	10	11	Mnemonic	Function	Description
0	0	0	1	XR	R1 (XOR) R2 → R1	Exclusive OR
0	0	1	1	OR	R1 (OR) R2 → R1	OR
0	1	0	1	NCR	R1 (AND) $\overline{R2}$ → R1	AND complement
0	1	1	0	NR	R1 (AND) R2 → R1	AND
0	1	1	1	OCR	R1 (OR) $\overline{R2}$ → R1	OR complement
1	0	0	0	DEC	R1 - 1 → R1	Decrement R1
1	0	0	1	ACYR	R1 + R2 + C → R1	Add registers and carry
1[1]	0	1	1	AR	R1 + R2 → R1	Add registers
1	1	0	0	SR	R1 - R2 + 1 → R1	Subtract registers
1	1	1	0	SCYR	R1 - R2 - C → R1	Subtract with borrow
1	1	1	1	INC	R1 + 1 → R1	Increment R1

[1] By adding a register to itself (R1 + R1), the shift left logical function can be executed. This function causes the eight bits to be shifted one position to the left and the low order bit (7) to be replaced with a zero (mnemonic = SLL).

Logical/Arithmetic 2

Mnemonic: LA2

0 1 1 1	▨	Reg 1	Function	H2	Reg 2
0 3	4	5 7	8 11	12	13 15

This instruction allows arithmetic and logical functions to be processed through the ALU. The Logical/Arithmetic 2 instruction always involves both bytes of register 1 (operand 1) and one or both bytes of register 2 (operand 2) depending on the function involved.

The logical and arithmetic functions that can be performed are specified by bits 8-11 as follows:

Bits						
8	**9**	**10**	**11**	**Mnemonic**	**Function**	**Description**
0	0	0	1	XR	R1 (XOR) R2 (2) → R1	Exclusive OR
0	0	1	1	OR	R1 (OR) R2 (2) → R1	OR
0	1	0	1	NCR	R1 (AND) $\overline{R2}$ (2) → R1	AND complement
0	1	1	0	NR	R1 (AND) R2 (2) → R1	AND
0	1	1	1	OCR	R1 (OR) $\overline{R2}$ (2) → R1	OR complement
1	0	0	0	DEC	R1 - 1 → R1	Decrement R1
1	0	0	1	ACYR	R1 + R2 (2) + C → R1	Add registers and carry
1	0	1	0	SR	R1 - R2 (1) + 1 → R1	Subtract registers
1[1]	0	1	1	AR	R1 + R2 (2) → R1	Add registers
1	1	0	0	SR	R1 - R2 (2) + 1 → R1	Subtract registers
1	1	0	1	AR	R1 + R2 (1) → R1	Add registers
1	1	1	0	SCYR	R1 - R2 (2) - C → R1	Subtract with borrow
1	1	1	1	INC	R1 + 1 → R1	Increment R1

[1] By adding a register to itself (R1 + R1), the shift left logical double (SLLD) function can be executed. This function causes the 16 bits to be shifted one position to the left and the low-order bit (15) to be replaced with a zero.

Test Mask

Mnemonic: TM

0 1 0 1	H1	Reg 1	Mask
0 3	4	5 7	8 15

The bits in one byte of a working register may be tested by this micro instruction. A mask in the immediate data field of the micro instruction identifies the bits to be tested. As a result of this test, one of three conditions will be detected and this condition will be set in the PCR:

1. Tested bits are all equal to 1 (positive bit set on).

2. Tested bits are a mixture of 1's and 0's (negative bit set on).

3. Tested bits are equal to 0 (zero bit set on).

Set Bits Off

Mnemonic: SBF

1 0 0 0	H1	Reg 1	Data
0 3	4	5 7	8 15

This instruction is used to set bits off in the specified byte of the selected register in the LSR stack.

Set Bits On

Mnemonic: SBN

1 0 0 1	H1	Reg 1	Data
0 3	4	5 7	8 15

This instruction is used to set bits on in the specified byte of the selected register in the LSR stack.

Load Immediate

Mnemonic: LI

1 0 1 0	H1	Reg 1	Data
0 3	4	5 7	8 15

This instruction takes the data in the data field and loads it directly into a selected register of the appropriate LSR stack. Data can be placed into the high or low byte of the selected register.

Compare Immediate

Mnemonic: CI

1 1 0 0	H1	Reg 1	Data
0 3	4	5 7	8 15

This instruction is used to compare the eight bits of data in the selected LSR with the corresponding eight bits of data in the data field. The results of the compare are reflected in the condition code setting. The selected LSR is not altered by the compare immediate instruction.

Subtract Immediate

Mnemonic: SI (subtract immediate)
AI (add immediate)

1 1 0 1	H1	Reg 1	Data
0 3	4	5 7	8 15

This instruction is used to decrement data in the specified LSR. It is also used to perform a compare function between the specified LSR and the data. The results of the function are loaded back into the specified LSR (register 1).

Move LSR

Mnemonic: MVR

1 1 1 0	Reg 1	1	S	Reg 2
0 3	4 7	8	9	10 15

This instruction moves the contents of one LSR to another LSR. Two bytes of data are always moved. Any of the 32 LSRs in the data flow can be accessed. Data can be moved either from register 1 to register 2 (if bit 9 = 0) or from register 2 to register 1 (if bit 9 = 1).

Storage Direct

Mnemonic: L (load register)
ST (store register)

1 1 1 0	W	Reg 1	0	SAR'
0 3	4	5 7	8	9 15

This instruction directly accesses the address of control storage (the fixed storage area; addresses 0000-007F) during read or write operations. Main storage cannot be accessed with this instruction. Two bytes of data are transferred.

Hex Branch

Mnemonic: HBN (numeric)
HBZ (zone)

1 1 1 1	H1	Reg 1	MAR'	0	Z
0 3	4	5 7	8 11	12	13 14 15

This instruction operates as a 16-way branch. Either the zone or digit portion of either the high or low byte of the selected register is used to replace bits 12-15 of MAR. Bits 8-11 of MAR are replaced by the bit settings of MAR .

Hex Move

Mnemonic: SRL (shift right logical)
SRLD (shift right logical double)
MZN (move zone to numeric)
MZZ (move zone to zone)

1 1 1 1	H1	Reg 1		Function	H2	1	Reg 2
0 3	4	5 7	8	9 10	11	12	13 15

This instruction performs the following functions:

Bits 9 and 10 = 00: Register 1 shift right logical (SRL). The eight bits of the selected byte are shifted one position to the right. The high-order (leftmost) bit is replaced with a 0. The register 2 and H2 fields of the hex move instruction are not used for the shift right logical function.

Bits 9 and 10 = 01: Register 1 shift right logical double (SRLD). The 16 bits of the selected register are shifted one position to the right. The high-order bit (bit 0) is replaced with a 0. The H1, H2, and register 2 fields of the hex move instruction are not used for the shift right logical double.

Bits 9 and 10 = 10: Link zone portion of register 2 to the zone portion of register 1 (MZN). The zone digit of the register specified in register 2 is moved to the zone position of the register specified in register 1 and the zone digit of register 1 is moved to the numeric position of register 1. The results are put in the register specified by register 1 and have the following format:

Reg 2	Reg 1
Zone	Zone

Bits 9 and 10 = 11: Link the zone portion of register 2 to the numeric portion of register 1 (MZZ). The zone digit of the register specified in register 2 is moved to the zone position of the register specified by register 1 and the numeric digit of the register specified by register 1 remains unchanged. The results are put in the register specified by register 1 and have the following format:

Reg 2	Reg 1
Zone	numeric

I/O Immediate

1 0 1 1	Mod	Func	H2	Reg
0 3	4 7	8 11	12	13 15

The I/O immediate instruction has two main functions:

1. Transfer of a single byte of data between the LSRs and I/O

2. Direct control channel and I/O function that may or may not include the data transfer

Instruction Fields

The device address is contained in WR0, bits 8-15 and is assigned as follows:

00 = Channel
10 = Keyboard
40 = Display screen
50 = Data recorder
50 = Mag card unit
50 = Magnetic character reader
80 = BSCA
80 = SDLC
A0 = Disk
D0 = Diskette
E0 = Printer

Mod: The modifier bits are device-dependent and are sent to the attachment. Along with the command, they define what is to be done with the data byte that will be sent.

Func: The function bits are sent on the CBO to the attachment, where they are decoded as one of the following commands:

0000 = I/O Load

0100 = I/O Sense

0101 = Sense Interrupt Level Status Byte

Interrupt level status byte is placed in the WR designated by bits 12–15 of the instruction.

10 = Disk	Device address from
20 = Keyboard, Printer	WR0, bits 8–15

0110 = Microprocessor Sense (note 1)

1000 = I/O Control Load

1010 = Microprocessor Load (note 2)

1100 = I/O Control Sense

H2: This bit is used to select the high or low byte of the selected LSR:

H2 = 0 – Select low byte
H2 = 1 – Select high byte

Reg: This field selects one of eight registers in an LSR stack. This register will contain the byte of data or control information that is to be sent to the attachment.

For more detail on these instructions see *IBM Maintenance Library, System/32 Theory Diagrams,* SY31-0346.

Notes:

1. CPU Microprocessor Sense:

Modifier Bits 4–7	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
0100 Console status byte	Stop key	Main storage address compare	Overlap off	IPL device select switch	I/O request	Sys step mode	Go flag	Micro interrupt check
0101 Address data switches 3 and 4	Switch 3 8	Switch 3 4	Switch 3 2	Switch 3 1	Switch 4 8	Switch 4 4	Switch 4 2	Switch 4 1
0110 Timer low byte[1]	8.19 ms	16.38 ms	32.77 ms	65.54 ms	131.1 ms	262.1 ms	524.3 ms	1 ms
0111 Timer high byte[1]	32 µs	64 µs	128 µs	256 µs	512 µs	1.02 ms	2.05 ms	4.10 ms
1001 Address data switches 1 and 2	Switch 1 8	Switch 1 4	Switch 1 2	Switch 1 1	Switch 2 8	Switch 2 4	Switch 2 2	Switch 2 1
1010 CPU error byte	SDR P check	MOR P check	Storage gate P check	ALU gate P check	Control storage invalid addr/SAR check	Main storage invalid addr/SAR check	Not used	Microloop check
1011 PCR	Flag	Plus	Minus	Zero	Carry log	Hi log	Low log	Equal log

[1] Contents of these bytes are in 1's complement form.

Notes (continued):

2. Microprocessor Load:

Modifier Bits 4-7	Function	Modifier 2 Bits 12-15
0000	Load PCR	1111
0001	Reset carry-set equal	1111
0010	Reset event indicator 2	1111
0011	Reset event indicator 3	1111
0100	Reset event indicator 4	1111
0101	Reset event indicator 5	1111
0110	Reset event indicator 6	1111
0111	Reset event indicator 7	1111
1000	Set flag	1111
1001	No-op	1111
1010	No-op	1111
1011	Reset flag	1111
1100	No-op	1111
1101	No-op	1111
1110	No-op	0000
1110	Set I/O service request	0001
1110	Reset I/O service request	0010
1110	Processor check halt	0011
1110	Disable checks	0100
1110	Enable interrupts	0101
1110	Disable interrupts	0110
1110	Enable checks	0111
1110	No op	1000
1110	No op	1110
1111	Set CPU working	0000
1111	Reset stop latch	0001
1111	Reset MCI latch	0010
1111	Reset go latch	0011
1111	Enable microloop timeout	0100
1111	Set stop latch	0101
1111	Reset retry/microloop timeout and set go	0110
1111	Set retry	0111
1111	Enable I/O clocks	1000
1111	No-op	1001
1111	Reset I/O clocks	1010
1111	Disable I/O clocks	0011
1111	No-op	1100
1111	Reset CPU working	1101
1111	Processor wait	1110

Storage (LC, LM, STC, STM)

0 1 0 0	H1	Reg 1	1	W	C	D	V	Reg 2
0 3	4	5 7	8	9	10	11	12	13 15

Storage is used for accessing either control storage or main storage. Data can be transferred to or from the LSRs.

The functions that can be performed are specified by bits 4 and 8–12 as follows:

Bit 4 = High byte
Bit 8 = Always 1
Bit 9 = 0—Read
1—Write
Bit 10 = 0—Main
1—Control
Bit 11 = 0—Increment
1—Decrement
Bit 12 = 0—No increment or decrement
1—Increment or decrement by 1

I/O Storage

	Modifier	O	W	C	D	V	Reg 2
0 1 0 0	xxxx						xxx
0 3	4 7	8	9	10	11	12	13 15

The function of I/O storage is to transfer one byte of data between main/control storage and the I/O device.

The device address is contained in work register 0, low, as follows:

00 = Channel
10 = Keyboard
40 = Display screen
50 = Data recorder
50 = Mag card unit or magnetic character reader
80 = BSCA
80 = SDLC
A0 = Disk
D0 = Diskette
E0 = Printer

The functions that can be performed are specified by bits 4 and 8-12 as follows:

Bits							
4	8	9	10	11	12	Mnemonic	Description
0	0	1	1	0	1	WTCL	I/O load to control storage
1	0	1	1	0	1	WTCH	increase R2 by 1
0	0	1	1	1	1	WTCL	I/O load to control storage
1	0	1	1	1	1	WTCH	decrease R2 by 1
0	0	1	1	0	0	WTCL	I/O load to control storage
1	0	1	1	0	0	WTCH	R2 (no change)
0	0	0	1	0	1	RDCL	I/O storage from control storage
1	0	0	1	0	1	RDCH	increase R2 by 1
0	0	0	1	1	1	RDCL	I/O storage from control storage
1	0	0	1	1	1	RDCH	decrease R2 by 1
0	0	0	1	0	0	RDCL	I/O storage from control storage
1	0	0	1	0	0	RDCH	R2 (no change)
X	0	1	0	0	1	WTM	I/O load to main storage, increase R2 by 1
X	0	1	0	1	1	WTM	I/O load to main storage, decrease R2 by 1
X	0	1	0	1	0	WTM	I/O load to main storage, no change
X	0	0	0	0	1	RDM	I/O storage from main storage, increase R2 by 1
X	0	0	0	1	1	RDM	I/O store from main storage; decrease R2 by 1
X	0	0	0	0	0	RDM	I/O store from main storage (no change)

Appendix E. Request Indicator Byte (RIB) Values

The RIB is used as a means of communication between the calling routine and the supervisor. Three types of RIB values are described in this section:

- Primary RIB equates (X'00'-X'0F') (Figure E-1)
- Relocatable loader RIB equates (X'40'-X'7D') (Figure E-2)
- Transient loader RIB equates (X'80'-X'97') (Figure E-3)

Requested Function	RIB (hex)
Return program level Communications area address	00
Reset interrupt	01
Wait SVC (X'000C'—see note)	02
Disk IOS (X'0008'—see note)	03
Pull main storage from disk	04
Perform stack functions (unstack)	05
Dump (X'0000'—see note)	06
Return from transient (X'0012'—see note)	07
Reserved for use by control storage function	08
Disable interrupts	09
Enable interrupts	0A
Push main storage to disk	0B
Perform stack functions (load from stack)	0C
Perform stack functions (store to stack)	0D
Queue/dequeue	0E
Load control storage transient area	0F

Note: Functions noted have unique SVCs in the first 256 bytes of main storage, and can be evoked by branching to the indicated address. The RIB value shown is the Q-byte for the SVC instruction at that address. These functions can also be evoked by branching to the general RIB request SVC at X'0004' and using the indicated RIB following the branch.

Functions not noted can be evoked by branching to the general RIB request SVC at X'0004' and using the indicated RIB following the branch.

RIB 08 is the Q-byte for the general RIB request SVC at location X'0004'. It must not be used as an RIB by a main storage function branching to the general RIB request SVC, since its RIB value is reserved for internal storage purposes and unpredictable results can follow.

Figure E-1. Primary RIB Equates

Requested Function	RIB (hex)
Load by relative sector	40
Load to address	68
Load to address with find	69
Fetch	70
Fetch with find	71
Fetch to address	78
Fetch to address with find	79
System fetch to address	7C
System fetch to address with find	7D

Figure E-2. Relocatable Loader RIB Equates

Requested Function	RIB (hex)
Explicit SS/#transient request	80
Library find	81
Open	82
Close	83
End of job	84
Syslog	85
Sysin	86
Scheduler work area—Get	87
Scheduler work area—Put	88
Scheduler work area—Read/write	89
VTOC read/write disk	8A
Allocate	8B
Fast find	8C
Rollin	8D
Rollout	8E
RPG halt	8F
Error logging	90
VTOC read/write diskette	91
Syslist	92
Source get	93
System message	94
Library open/close	95
Prepare diskette	96
Printer error recovery	97
Offline linkage	98
Special request RIB (See the secondary RIB table for a list of word processing transients associated with each secondary RIB)	99

Note: X'80' is the bit mask for transient RIB calls; X'40' is the refresh bit mask for transient RIBs.

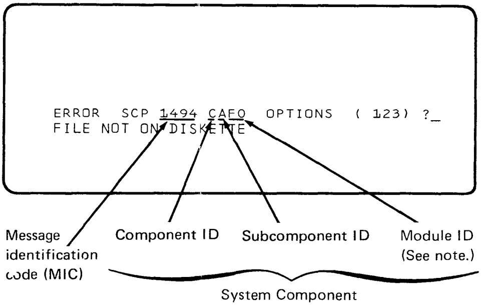
Figure E-3. Transient Loader RIB Equates

Requested Function	RIB (hex)
Load and save WPWA (word processing work area)	9902
Keyboard redefine	9903
Security key conversion transient	9904
Job-to-job inquiry	9905
Variable microcode location finder	9906
Scratch file deallocator	9908

Figure E-4. Secondary RIB Table

Appendix F. Message-Issuing Module Identification

Figure F-1 shows a typical formatted error display, certain areas of which are further defined and listed to assist in isolating a message and in determining which system control program module issued the message.



```
ERROR   SCP 1494 CAF0  OPTIONS   ( 123) ?_
FILE NOT ON DISKETTE
```

Note: Two identification characters are contained in the syslog parameter list. If the syslog parameter list does not contain these optional characters, blanks will be displayed. (In this example, F0 represents the module ID of the module issuing the error message.)

		System Component		
Function	**Message Identification Code (MIC)**	**Component ID**	**Subcomponent ID**	**Module ID**
Level 1 – Headings	0000–0999	H	H	
Level 2 – Abnormal termination errors	0000–0018	H	H	
(ABEND)				
Load request not from user area	0000			
Attempt to load module below user area	0001			
Attempt to relocate module with no RLDs	0002			
SVC instruction not in first 256 bytes of main storage	0003			

Figure F-1 (Part 1 of 10). Message-Issuing Module Identification Aid

		System Component		
Function	Message Identification Code (MIC)	Component ID	Subcomponent ID	Module ID
(ABEND) continued				
Disk save area overflow	0004			
Push stack overflow	0005			
Push stack underflow	0006			
Register stack overflow	0007			
Register stack underflow	0008			
Invalid stack displacement	0009			
Bad IOB parameter	0010			
Unrecoverable user IOB disk error	0011			
Disk interrupt time-out check	0012			
Invalid Q-byte or Invalid R-byte	0013			
Invalid address	0014			
Invalid op code or invalid operation	0015			
Dump storage request or terminate job	0016			
Invalid transient number parameter	0017			
Control storage process check	0018			

Note: Information on what caused this error can be found at MCLGAREA in control storage. (See *IBM System/32 Control Storage Logic Manual,* SY21-0533.)

Figure F-1 (Part 2 of 10). Message-Issuing Module Identification Aid

Function	Message Identification Code (MIC)	System Component		
		Component ID	Subcomponent ID	Module ID
Level 2 - Abnormal termination errors (continued)				
When this error occurs, the field $ERRMIC at location X'0070' in control storage contains a number identifying the error.	0033			

Number at $ERRMIC		Description of Error
0	Loader SVC errors	Load request not from user area
1		Attempt to load module below user area
2		Attempt to relocate module with no RLDs
3	SVC processor error	Main storage SVC instruction not in first 256 bytes of main storage
4	Push SVC errors	Disk save area overflow
5		Push stack overflow
6		Push stack underflow
7	Reg stack errors	Stack overflow
8		Stack underflow
9	Store-to-stack SVC error	Invalid stack displacements
10	Disk IOS errors	Bad IOB parameters
11		Unrecoverable user IOB disk error
12		Disk interrupt timeout check
13	Emulator/ECS supervisor detected errors	Invalid Q-code or R-code
14		Invalid address
15		Invalid op code or operation
16	Entry for dump-storage-and-terminate-job SVC	Dump storage/terminate job
17	CS xient loader SVC errors	Invalid transient number parameter
18		Control storage process check

Figure F-1 (Part 3 of 10). Message-Issuing Module Identification Aid

		System Component		
Function	Message Identification Code (MIC)	Component ID	Subcomponent ID	Module ID
BSCA/SDLC-SNA	3200-3305 4400-4499 4800-4899	B		
BSCA		B	C	
#BSCH		B	C	CH
#BSCL		B	C	CL
#BSLO		B	C	LO
#BSOB		B	C	OB
#BSST		B	C	ST
SDLC-SNA		B	S	
#SDJ1		B	D	LC
#SDJ3		B	S	NA
SCHEDULER		C		
INITIATOR	1030-1309 1351-1355 1364, 1367	C	I	
#IND1		C	I	D1
#IND2		C	I	D2
#INPO		C	I	PO
#INP1		C	I	P1
ALLOCATE	1320-1350 1356-1363 1365-1366 1368-1473 1476-1479 1481-1483 1487-1488 1490-1492 1496-1499	C	A	
#STDA		C	A	DA
#STDD		C	A	DD
#STFI		C	A	FI
#STFT		C	A	FT
#INF0		C	A	F0
#INF1		C	A	F1
#INF3		C	A	F3
READER/INTERPRETER	3700-3825	C	R	
#RDCM		C	R	CM
#RDDT		C	R	DT
#RDFL		C	R	FL
#RDFM		C	R	FM
#RDIC		C	R	IC
#RDIM		C	R	IM
#RDLD		C	R	LD
#RDLG		C	R	LG
#RDMK		C	R	MK
#RDML		C	R	ML
#RDMM		C	R	MM
#RDMS		C	R	MS
#RDPS		C	R	PS
#RDRN		C	R	RN
#RDRT		C	R	RT
#RDSL		C	R	SL
#RDSW		C	R	SW
TERMINATOR	1310-1319	C	T	
#TME1		C	T	E1

Figure F-1 (Part 4 of 10). Message-Issuing Module Identification Aid

Function	Message Identification Code (MIC)	System Component		
		Component ID	Subcomponent ID	Module ID
COMMANDS				
DATA MANAGEMENT	2200-2219	D		
DISK CLOSE	2230-2234	D	C	
#TMDS	2251-2252	D	C	DS
#CDIF		D	C	IF
#CLOZ		D	C	OZ
#CDUF		D	C	UF
#FSCP		D	C	CP
DISK OPEN		D	D	
#FSPO		D	D	PO
#ODCB		D	D	CB
#ODDI		D	D	DI
#ODD2		D	D	D2
#OPEN		D	D	EN
#ODFM		D	D	FM
#ODF1		D	D	F1
#ODPM		D	D	PM
#ODSB		D	D	SB
#ODSO	1474, 1485-1486	D	D	SO
DISKETTE CLOSE	1489, 1493-1495,	D	L	
#CDR1	2214-2227	D	L	R1
#CDZL		D	L	ZL
DISKETTE OPEN		D	P	
#ODI1		D	P	I1
#ODZP		D	P	ZP
DISKETTE OPEN NEXT		D	E	
#ODNV		D	E	NV
MAGNETIC CHARACTER READER	4950-4999			
#MI08		D	M	MI
OLMV		D	O	
#OMOAD		D	O	AD
#$OMDO		D	O	DO
#$OMLK		D	O	LK
LINKAGE EDITOR	3000-3099	E	L	
$LINKB		E	L	KB
$LINKD		E	L	KD
$LINKE		E	L	KE
$LINKF		E	L	KF
$LINKG		E	L	KG
$LINKH		E	L	KH
$LINKJ		E	L	KJ
$LINKK		E	L	KK
$LINKM		E	L	KM

Figure F-1 (Part 5 of 10). Message-Issuing Module Identification Aid

		System Component		
Function	Message Identification Code (MIC)	Component ID	Subcomponent ID	Module ID
OVERLAY LINKAGE EDITOR	3100-3199			
#OLER		E	O	ER
#OLINK		E	O	NK
WORD PROCESSING	4700-4799	E	W	
#WP#AX		E	W	AX
#WP#BR		E	W	BR
#WP# DI		E	W	DI
#WP#JI		E	W	JI
$WPJAT		E	W	JA
$WPJCT		E	W	JC
$WPJIT		E	W	JI
$WPLUF		E	W	LU
$WPLUP		E	W	UP
$WPURG		E	W	PU
#WP#1M		E	W	MC
#WP#2M		E	W	MC
#WP#3M		E	W	MC
#WP#4M		E	W	MC
#WP#5M		E	W	MC
#WP#6M		E	W	MC
#WP#7M		E	W	MC
#WP#8M		E	W	MC
#WP#9M		E	W	MC
#WP#NL		E	W	NL
$WPPBI		E	W	BI
$WPSET		E	W	SE
WORD PROCESSING COMMUNICATIONS UTILITY	2300-2399			
$WCBS		E	W	CB
$WCIN		E	W	CI
$WCLI		E	W	CL
$WCSI		E	W	CS
IOS		I		
DISK		I	F	
CARD I/O	2900-2915	I	H	
#SBR12		I	H	12
KEBOARD/CRT		I	K	
#KIKEY		I	K	CH
#KIOPN		I	K	OD
PRINTER		I	P	
#POE30		I	P	LE
#POE50		I	P	ME

Figure F-1 (Part 6 of 10). Message-Issuing Module Identification Aid

		System Component		
Function	**Message Identification Code (MIC)**	**Component ID**	**Subcomponent ID**	**Module ID**
I1		I	R	
#IGBLD		I	R	DB
#I1ERP		I	R	DE
#I1PRE		I	R	DP
#I1IOS	2069-2073 2099-2103	I	R	DS
LIBRARIAN	2139-2143	L		
$MADLT	2176-2181	L	M	DL
$MAHLT	2400-2599	L	M	HL
$MAINT		L	M	IN
$MALOC		L	M	LO
$MALTL		L	M	LT
$MACND		L	M	ND
#ODNP		L	M	NP
#OXRF		L	M	OX
$MAPUR		L	M	PU
$MARDR		L	M	RD
$MARFF		L	M	RF
$MARTF		L	M	RT
$MAIST		L	M	ST
$MATFS		L	M	TF
$MATLS		L	M	TL
SYSTEM SUPPORT		T		
SYSTEM FIND		T	F	
#SPFN		T	F	SP
SYSLOG		T	G	
#SYTGL		T	G	GL
#SYSLG		T	G	LG
#SYXTG		T	G	TG
SYSLIST		T	L	
#SYLST		T	L	LY
MESSAGE RETRIEVE		T	M	
#MGRET		T	M	RT
SYSIN		T	S	
#SYSIN		T	S	IN
#STSS		T	S	SS
#STS2		T	S	S2
ROLLIN/ROLLOUT		T	R	
#STRI		T	R	RI
#STRO		T	R	RO
VTOC HANDLING		T	V	
#SSVL		T	V	UL

Figure F-1 (Part 7 of 10). Message-Issuing Module Identification Aid

		System Component		
Function	Message Identification Code (MIC)	Component ID	Subcomponent ID	Module ID
UTILITIES		U		
$BACK	1500-1519	U	A	BA
$BICR	1520-1549, 2182-2194	U	B	IC
$COPY	1560-1690 2052-2061 2109-2121 2144-2162	U	C	
$COADD		U	C	AD
$COALL		U	C	AL
$COGET		U	C	CG
$COINT		U	C	CI
$COPY		U	C	PY
$COMSK		U	C	SK
$COZIP		U	C	ZP
$BUILD		U	D	BU
$SETCF	1770-1779, 2163-2170	U	E	CF
$DELET	1484, 1620-1639, 2042-2051	U	F	
$DELET		U	F	DL
$DELF1		U	F	F1
$DELI1		U	F	I1
$FEAPR	1800-1899	U	G	FE
$HIST	1660-1669, 2074-2079	U	H	$H
$INIT	1670-1689, 2080-2088	U	I	NT
$CNVRT		U	J	CN
$PACK	2845	U	K	PA
$LABEL	1690-1719, 2015-2021	U	L	AB
$MGBLD	1720-1749, 2030-2036	U	M	GB
$RENAM	2089-2099	U	N	AM
$LOAD	3900-4099, 2133-2138	U	O	
$LOAD		U	O	LO
$LOADI		U	O	LI
$FEDMP	1800-1899	U	P	FE
$QJOB	2261-2299	U	Q	JB
$REBLD		U	R	BL
$STATS		U	S	TS
$FEPCH	1800-1899	U	T	
$FEPCH		U	T	FE
$FEKEY		U	T	KE
$DUPRD		U	U	RD

Figure F-1 (Part 8 of 10). Message-Issuing Module Identification Aid

		System Component		
Function	Message Identification Code (MIC)	Component ID	Subcomponent ID	Module ID
$FREE	2843-2849	U	V	
$FREE		U	V	EE
$FREE1		U	V	E1
$FREE2		U	V	E2
$FREE3		U	V	E3
FEFIX	1800-1899	U	X	
$FEFIX		U	X	FE
$FESUM		U	X	SU
SYNTAX CHECKER	4100-4299	U	Y	
NUCLEUS	0000-0255, 6420-6439, 2800-2839, 2850-2880, 6300-6399, 6500-6699	V	C	
BWS/MRJE	4500-4599, 4614-4699	X		
BWS		X	B	
$BWDBI		X	W	
$BWFAL		X	B	AL
$BWCCO		X	B	CC
$BWDCI		X	B	CI
$BWFCL		X	B	CL
$BWDCO		X	B	CO
$BWCCR		X	B	CR
$BWFDA		X	B	DA
$BWDDI		X	B	DI
$BWCD2		X	B	D2
$BWSEH		X	B	EH
$BWCER		X	B	ER
$BWDFT		X	B	FT
$BWSIH		X	B	IH
$BWDIO		X	B	IO
$BWDKI		X	B	KI
$BWDLI		X	B	LI
$BWCMD		X	B	MD
$BWDMH		X	B	MH
$BWCM1		X	B	M1
$BWCM2		X	B	M2
$BWINR		X	B	NR
$BWINT		X	B	NT
$BWFOI		X	B	OI
$BWFOO		X	B	OO

Figure F-1 (Part 9 of 10). Message-Issuing Module Identification Aid

Function	Message Identification Code (MIC)	System Component Component ID	Subcomponent ID	Module ID
$BWDPI		X	B	PI
$BWDPO		X	B	PO
$BWDRI		X	B	RI
$BWDR1		X	B	R1
$BWDR2		X	B	R2
$BWCR3		X	B	R3
$BWCSC		X	B	SC
$BWFSE		X	B	SE
$BWCSG		X	B	SG
$BWCSI		X	B	SI
$BWCSL		X	B	SL
$BWCSR		X	B	SR
$BWSUR		X	B	UR
MRJE		X	M	
#MRBM		X	M	BM
#MRBO		X	M	BO
#MRCD		X	M	CD
#MRI1		X	M	I1
#MRI6		X	M	I6
#MRKP		X	M	KP
#MRRP		X	M	RP
#DCSUP		X	M	RT
#MRTM		X	M	TM

Figure F-1 (Part 10 of 10). Message-Issuing Module Identification Aid

Appendix G. Conversion Aids

This appendix includes the following conversion aids:

- Hexadecimal and decimal conversion/addition
- Records to blocks conversion for disk
- Sector number to block number conversion for disk
- Block number to first sector in block conversion for disk
- Disk sector address (SS) to FDIOS (4-byte CCHS) format conversion
- Blocks to sector address conversion
- Diskette sector address to track conversion table

HEXADECIMAL AND DECIMAL CONVERSION/ADDITION

Conversion (hex to decimal)

To find the decimal value of a hexadecimal number, locate the hex number and its decimal equivalent for each position from the conversion table (Figure G-1). Add these values to obtain the decimal number.

Example: Convert X'1FA' to decimal

Hex	Dec	Hex	Dec	Hex	Dec
1	256	F	240	A	10
3		2		1	
Positions					

X'1FA' = (256 + 240 + 10) = 506

Conversion (decimal to hex)

To find the hex value of a decimal number, locate the next lower decimal number and its hex equivalent from the conversion table (Figure G-1). Use the decimal difference to obtain the remaining hex numbers until the entire number is developed.

Example: Convert 534 to hex

Hex	Dec	Hex	Dec	Hex	Dec
2	512	1	16	6	6
3		2		1	
Positions					

Decimal 534 = X'216'

Figure G-1. Conversion Table

Byte				Byte				Byte			
0123		4567		0123		4567		0123		4567	
Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
0	0	0	0	0	0	0	0	0	0	0	0
1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15
6		5		4		3		2		1	

Figure G-2. Hexadecimal Addition Table

	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
2	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11
3	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12
4	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
5	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14
6	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15
7	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16
8	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17
9	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18
A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19
B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A
C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

RECORDS TO BLOCKS CONVERSION FOR DISK

Determining the Number of Sequential or Direct File Blocks

Do the following to determine the number of blocks in a sequential or direct file:

1. *Multiply:* number of records x record length = number of characters

2. *Divide:* $\frac{\text{number of characters (from step 1)}}{\text{number of characters per block (2560)}} = \text{number of blocks}$

   (if there is a remainder, round to the next higher whole number)

Determining the Number of Indexed File Blocks

Do the following to determine the number of data blocks in an indexed file:

1. *Multiply:* number of records x record length = number of characters

2. *Divide:* $\frac{\text{number of characters (from step 1)}}{\text{number of characters per block (2560)}} = \text{number of data}$

   blocks (if there is a remainder, round to the next higher whole number)

Do the following to determine the number of index blocks in an indexed file:

1. *Add:* key field length + 3 = index entry length

2. *Divide:* $\frac{\text{number of characters in a sector (256)}}{\text{index entry length (from step 1)}} = \text{number of entries}$

   per sector (drop fraction)

3. *Divide:* $\frac{\text{number of records}}{\text{number of entries per sector (from step 2)}} = \text{number of}$

   sectors (if there is a remainder, round to the next higher whole number)

4. *Divide:* $\frac{\text{number of sectors (from step 3) + 3}}{\text{number of sectors per block (10)}} = \text{number of index}$

   blocks (if there is a remainder, round to the next higher whole number)

To determine the total number of blocks required for an indexed file, add the number of data blocks required to the number of index blocks required.

SECTOR NUMBER TO BLOCK NUMBER CONVERSION FOR DISK

To convert sector number to block number, subtract 1 from the sector number, divide the result by 10, and drop the remainder.

Examples:

10511 = sector number
(10511 - 1) ÷ 10 = 1051.0
1051 = block number

10520 = sector number
(10520 - 1) ÷ 10 = 1051.9
1051 = block number

BLOCK NUMBER TO FIRST SECTOR IN BLOCK CONVERSION FOR DISK

To find the first sector in a block, multiply the block number by 10 and add 1.

Example:

1051 = block number
(1051 x 10) + 1 = 10511
10511 = first sector in block 1051

DISK SECTOR ADDRESS (SS) TO FDIOS (4-BYTE CCHS) FORMAT CONVERSION (FOR 3.2, 5.0, OR 9.1 MEGABYTE DISKS)

Directions		**Example (assume SS to be X'00BF')**
1.	Convert SS to decimal and subtract 1 from result (SS - 1).	X'00BF' = 191 191 - 1 = 190
2.	Add 180 to (SS - 1) (for reserved tracks 0, 1, and 2).	190 + 180 = 370
3.	Divide sum by 120 (number of sectors in a cylinder).	370 ÷ 120 = 3, remainder of 10
	a. The decimal quotient is the CC value.	CC = 3
	b. If the remainder is greater than or equal to 60, the H byte = X'01'	
	c. If the remainder is less than 60, the H byte = X'00'.	H = X'00'
	d. If the remainder is greater than or equal to 60, the remainder minus 60 = S; if the remainder is less than 60, the remainder = S.	S = 10
4.	Convert the CC value to hex (two bytes).	CC = X'0003'

5.	Convert the S value to hex (one byte).	S = X'0A'
		CCHS = X'0003000A'

DISK SECTOR ADDRESS (SS) TO FDIOS (4 BYTE-CCHS) FORMAT CONVERSION (FOR 13.7 MEGABYTE DISK)

	Directions	Example (SS = X'00BF)
1.	Convert SS to decimal and subtract 1 from the result (SS – 1)	X'00BF' = 191 191 – 1 = 190
2.	Add 300 to (SS – 1) (for reserved tracks 0 through 5)	190 + 300 = 490
3.	Divide the sum by 180 (number of sectors in a cylinder)	490 ÷ 180 = 2, remainder of 130
	a. The decimal quotient is the CC value	CC = 2

	b. If the remainder is greater than or equal to 120, the H byte is X'02' and the S value is the remainder minus 120	H = X'02'
		S = 10

	c. If the remainder is less than 120 and greater than or equal to 60, the H byte is X'01' and the S value is the remainder minus 60	
	d. If the remainder is less than 60, the H byte = X'00' and the S value is the remainder	
4.	Convert the CC value to hex (2 bytes)	CC = X'0002'

5.	Convert the S value to hex (1 byte)	S = X'0A'
		CCHS = X'0002020A'

BLOCKS (VTOC DISPLAY) TO SECTOR ADDRESS CONVERSION

1. Multiply the number of blocks by 10.

2. Add 1 to the product.

3. Convert the sum to hex.

Sector Address (SS) (in decimal)		Sector Address (SS) (in hex)		Track Number	
From	Through	From	Through	Decimal	Hex
1	26	0001	001A	0	00
27	52	001B	0034	1	01
53	78	0035	004E	2	02
79	104	004F	0068	3	03
105	130	0069	0082	4	04
131	156	0083	009C	5	05
157	182	009D	00B6	6	06
183	208	00B7	00D0	7	07
209	234	00D1	00EA	8	08
235	260	00EB	0104	9	09
261	286	0105	011E	10	0A
287	312	011F	0138	11	0B
313	338	0139	0152	12	0C
339	364	0153	016C	13	0D
365	390	016D	0186	14	0E
391	416	0187	01A0	15	0F
417	442	01A1	01BA	16	10
443	468	01BB	01D4	17	11
469	494	01D5	01EE	18	12
495	520	01EF	0208	19	13
521	546	0209	0222	20	14
547	572	0223	023C	21	15
573	598	023D	0256	22	16
599	624	0257	0270	23	17
625	650	0271	028A	24	18
651	676	028B	02A4	25	19
677	702	02A5	02BE	26	1A
703	728	02BF	02D8	27	1B
729	754	02D9	02F2	28	1C
755	780	02F3	030C	29	1D
781	806	030D	0326	30	1E
807	832	0327	0340	31	1F
833	858	0341	035A	32	20
859	884	035B	0374	33	21
885	910	0375	038E	34	22
911	936	038F	03A8	35	23
937	962	03A9	03C2	36	24
963	988	03C3	03DC	37	25
989	1014	03DD	03F6	38	26
1015	1040	03F7	0410	39	27
1041	1066	0411	042A	40	28
1067	1092	042B	0444	41	29
1093	1118	0445	045E	42	2A
1119	1144	045F	0478	43	2B
1145	1170	0479	0492	44	2C
1171	1196	0493	04AC	45	2D
1197	1222	04AD	04C6	46	2E
1223	1248	04C7	04E0	47	2F
1249	1274	04E1	04FA	48	30

Figure G-3 (Part 1 of 2). 128 Byte Format Diskette Sector Address to Track Conversion Table

Sector Address (SS) (in decimal)		Sector Address (SS) (in hex)		Track Number	
From	Through	From	Through	Decimal	Hex
1275	1300	04FB	0514	49	31
1301	1326	0515	052E	50	32
1327	1352	052F	0548	51	33
1353	1378	0549	0562	52	34
1379	1404	0563	057C	53	35
1405	1430	057D	0596	54	36
1431	1456	0597	05B0	55	37
1457	1482	05B1	05CA	56	38
1483	1508	05CB	05E4	57	39
1509	1534	05E5	05FE	58	3A
1535	1560	05FF	0618	59	3B
1561	1586	0619	0632	60	3C
1587	1612	0633	064C	61	3D
1613	1638	064D	0666	62	3E
1639	1664	0667	0680	63	3F
1665	1690	0681	069A	64	40
1691	1716	069B	06B4	65	41
1717	1742	06B5	06CE	66	42
1743	1768	06CF	06E8	67	43
1769	1794	06E9	0702	68	44
1795	1820	0703	071C	69	45
1821	1846	071D	0736	70	46
1847	1872	0737	0750	71	47
1873	1898	0751	076A	72	48
1899	1924	076B	0784	73	49
1925	1950	0785	079E	74	4A
1951	1976	079F	07B8	75	4B
1977	2002	07B9	07D2	76	4C

Figure G-3 (Part 2 of 2). 128 Byte Format Diskette Sector Address to Track Conversion Table

Sector Address (SS) (in decimal)		Sector Address (SS) (in hex)		Track Number	
From	Through	From	Through	Decimal	Hex
1	26	1	1A	0	00
27	34	1B	22	1	01
35	42	23	2A	2	02
43	50	2B	32	3	03
51	58	33	3A	4	04
59	66	3B	43	5	05
67	74	43	4A	6	06
75	82	4B	52	7	07
83	90	53	5A	8	08
91	98	5B	62	9	09
99	106	63	6A	10	0A
107	114	6B	72	11	0B
115	122	73	7A	12	0C
123	130	7B	82	13	0D
131	138	83	8A	14	0E
139	146	8B	92	15	0F
147	154	93	9A	16	10
155	162	9B	A2	17	11
163	170	A3	AA	18	12
171	178	AB	B2	19	13
179	186	B3	BA	20	14
187	194	BB	C2	21	15
195	202	C3	CA	22	16
203	210	CB	D2	23	17
211	218	D3	DA	24	18
219	226	DB	E2	25	19
227	234	E3	EA	26	1A
235	242	EB	F2	27	1B
243	250	F3	FA	28	1C
251	258	FB	102	29	1D
259	266	103	10A	30	1E
267	274	10B	112	31	1F
275	282	113	11A	32	20
283	290	11B	122	33	21
291	298	123	12A	34	22
299	306	12B	132	35	23
307	314	133	13A	36	24
315	322	13B	142	37	25
323	330	143	14A	38	26
331	338	14B	152	39	27
339	346	153	15A	40	28
347	354	15B	162	41	29
355	362	163	16A	42	2A
363	370	16B	172	43	2B
371	378	173	17A	44	2C
379	386	17B	182	45	2D
387	394	183	18A	46	2E
395	402	18B	192	47	2F
403	410	193	19A	48	30
411	418	19B	1A2	49	31
419	426	1A3	1AA	50	32
427	434	1AB	1B2	51	33
435	442	1B3	1BA	52	34

Figure G-4 (Part 1 of 2). 512 Byte Format Diskette Sector Address to Track Conversion Table

.Sector Address (SS) (in decimal)		Sector Address (SS) (in hex)		Track Number	
From	Through	From	Through	Decimal	Hex
443	450	1BB	1C2	53	35
451	458	1C3	1CA	54	36
459	466	1CB	1D2	55	37
467	474	1D3	1DA	56	38
475	482	1DB	1E2	57	39
483	490	1E3	1EA	58	3A
491	498	1EB	1F2	59	3B
499	506	1F3	1FA	60	3C
507	514	1FB	202	61	3D
515	522	203	20A	62	3E
523	530	20B	212	63	3F
531	538	213	21A	64	40
539	546	21B	222	65	41
547	554	223	22A	66	42
555	562	22B	232	67	43
563	570	233	23A	68	44
571	578	23B	242	69	45
579	586	243	24A	70	46
587	594	24B	252	71	47
595	602	253	25A	72	48
603	610	25B	262	73	49
611	618	263	26A	74	4A
619	626	26B	272	75	4B
627	634	273	27A	76	4C

Figure G-4 (Part 2 of 2). 512 Byte Format Diskette Sector Address to Track Conversion Table

Appendix H. Formats and Parameters for OCL Statements, IBM Command Statement Formats, and Command Statement Formats for IBM Service Procedures

For an extended description of the parameters listed in the figures in this appendix (Figures H-1, H-2, H-3, and H-4), refer to the *IBM System/32 System Control Programming Reference Manual,* GC21-7593.

Statement	Function	Where Statement Appears in Job Stream	Restrictions on Use
// COMPILE	Tells the system which source program to be compiled	Must follow LOAD statement and precede the RUN statement	—
// DATE	Supplies the system with a date; this date is given to disk files being created	Must follow LOAD statement and precede the RUN statement except at IPL time, when it must precede the first LOAD statement	Only one DATE statement is allowed between a LOAD and a RUN statement
// FILE	Supplies information to the system about the disk file	Must follow LOAD statement and precede the RUN statement	Required for every new disk file created and existing disk files being allocated
// FORMS	Instructs the system to change the number of lines printed per page	May appear anywhere among the OCL statements	—
// IMAGE	Tells the system to replace the print belt image area with characters keyed in or read from a member in the source library	May appear anywhere among the OCL statements	Required if the printer belt has been changed to one with a different character set
// INCLUDE	Identifies the procedure member to be merged into job stream	May appear anywhere among the OCL statements	Can be no more than 16 levels of nested procedures

Figure H-1 (Part 1 of 3). Table of OCL Statements

Statement	Function	Where Statement Appears in Job Stream	Restrictions on Use
// LOAD	Identifies the LOAD program to be executed	Must be the first statement in a set of statements for a LOAD program	Required in the job stream for the program to be run
// LOG	Instructs system to start or stop printing OCL statements and messages on the printer—also allows control of page skipping at end of job	May appear anywhere among the OCL statements	Only one logging device (CRT or PRINTER) may be specified
// MEMBER	Changes the message member from which messages come	May appear anywhere among the OCL statements	—
// PAUSE	Tells the program to stop in order to give the operator time to perform a function	May appear anywhere among the OCL statements	Operator must restart the program
// RUN	Indicates the end of the OCL statements for a program and tells system to run the program	Must be the last OCL statement	Required in the job stream for the program which is to be run
// SWITCH	Used to set one or more external indicators on or off or leave the indicator as it is	May appear anywhere among the OCL statements	Only one switch statement is allowed between a LOAD and a RUN statement
// SYSLIST	Changes the device used as syslist (printer or display screen)	May appear anywhere among the OCL statements	—

Figure H-1 (Part 2 of 3). Table of OCL Statements

Statement	Function	Where Statement Appears in Job Stream	Restrictions on Use
* (comment)	Explains the job and does not affect the program in operation	May appear anywhere among the OCL statements	The * must be in the first position
/* (end of data)	Indicates the end of a data file entered from the keyboard	Last record of an input data file	Not allowed in a procedure
// * (message)	Displays message to operator	May appear anywhere among the OCL statements	

Figure H-1 (Part 3 of 3). Table of OCL Statements

Statement	Parameter	Meaning of Parameter
// COMPILE	SOURCE-name	Name of source program.
// DATE	mmddyy or yymmdd or ddmmyy	System date, or date within a set of statements (job date).
// FILE (disk)	NAME-filename or NAME-COPYIN or NAME-COPYO	Name the program uses to refer to the file. For certain utilities programs, refer to the input file. For certain utilities programs, refer to the output file.
	UNIT-F1	Location of the file is or will be the disk. If the parameter is not specified, default is F1.
	LABEL-filename	Name by which your file is identified on disk.
	RECORDS-number or BLOCKS-number	Amount of space needed on a disk for a file.
	LOCATION-blocknumber	Number of the block where the file begins or is to begin.
	RETAIN-T or RETAIN-S or RETAIN-P	Temporary file. Scratch file. Permanent file.
	DATE-mmddyy or DATE-ddmmyy or DATE-yymmdd	Tells the system the date the file was created.

Figure H-2 (Part 1 of 3). Table of Parameters

Statement	Parameter	Meaning of Parameter
// FILE (diskette)	NAME-filename	Name the program uses to refer to the file.
	UNIT-I1	Location of the file is or will be the diskette.
	LABEL-filename	Name by which your file is identified on the diskette.
	RETAIN-retention-days	The number of days a file is retained before it expires; maximum is 998. If you specify 999, the expiration date is set to 99/99/99, creating a permanent file. Default is 1 day.
	DATE-mmddyy or DATE-ddmmyy or DATE-yymmdd	Tells the system the date the file was created.
	PACK-vol-id	Indicates the volume identification of the diskette.
// FORMS	LINES-value	Indicates number of lines to be printed per page.
// IMAGE	HEX or	Indicates characters are in hex form.
	CHAR or	Indicates characters from the input device are in EBCDIC form.
	MEM or MEMBER	Indicates characters are in a source member in the library.
	number	Number of new characters.
	name	Identifies the source member name in the library containing the chain image data.
// INCLUDE	procedure-name	Name that identifies the procedure in the library.
// LOAD	program-name	Name of program that is to be loaded from disk.
// LOG	CRT or	Use display screen only as logging device.
	PRINTER	Use printer and display screen as logging device.
	EJECT or	Skip to next page at end of job.
	NOEJECT	Do not skip to next page at end of job.

Figure H-2 (Part 2 of 3). Table of Parameters

Statement	Parameter	Meaning of Parameter
// MEMBER	PROGRAM1-name	Name of the load member used for problem program level 1 messages; if name is 0, the member name is cleared.
	PROGRAM2-name	Name of the load member used for problem program level 2 messages; if name is 0, the member name is cleared.
	USER1-name	Name of the load member used for user program or level 1 OCL message statements; if name is 0, the member name is cleared.
	USER2-name	Name of the load member used for user program or level 2 OCL message statements; if name is 0, the member name is cleared.
// PAUSE	None	
// RUN	None	
// SWITCH	Eight external indicators (UPSI)	1 = Set on X = Unaffected 0 = Set off
// SYSLIST	CRT	Use the CRT as the syslist device.
	PRINTER	Use the printer as the syslist device. After an IPL, the printer is the syslist device.
	OFF	Ignore subsequent request for syslist output.
* (comment)	None	
/* (end of data)	None	
// * (message)	msg-id	The identification of a message in the assigned USER1 message member.
	'message'	A character string which is the actual message.

Figure H-2 (Part 3 of 3). Table of Parameters

```
ALTERBSC  [BRATE- {F}] [,CLOCK- {Y}] [,DEBUG- {Y}]
                  {H}            {N}            {N}
   [,ERC- {number}] [,SLINE- {Y}] [,TEST- {Y}] [,TONE- {Y}]
          {7     }           {N}          {N}          {N}
```

Note: At least one parameter must be given in each ALTERBSC statement.

```
ALTERSDL  [BRATE- {F}] [,CLOCK- {Y}] [,DEBUG- {Y}]
                  {H}            {N}            {N}
                    [,SLINE- {Y}] [,TEST- {Y}] [,TONE- {Y}]
                             {N}          {N}          {N}
```

Note: At least one parameter must be given in each ALTERSDL statement.

```
APCHANGE   [#blocks] [,filename [,CLEAR] ]
```

Note: At least one parameter must be given in each APCHANGE statement.

```
BACKUP vol-id,  [retention-days]  [,filename ]
                [1             ]  [,#LIBRARY ]

BWSUD   sluname, host

BWSUR   sluname

CATALOG  [ALL     ]  [,I1]
         [filename]  [,F1]

COMPRESS

CONDENSE

CONVERT

COPYI1  [ALL] ,,vol-id,  [DELETE] ,  [PRESERVE]  [,number of copies]
                                                 [,1               ]

                   or
                  [mmddyy]
COPYI1 filename,  [ddmmyy]  ,vol-id,,  [PRESERVE]  [,number of copies]
                  [yymmdd]                         [,1               ]

CREATE sourcename  [,REPLACE]

      {mmddyy}
DATE  {ddmmyy}
      {yymmdd}

DCPRINT filename

                                [SCRATCH]  [,mmddyy]
DELETE filename,  [F1]  ,       [REMOVE ]  [,ddmmyy]
                  [I1]          [ERASE  ]  [,yymmdd]
```

Figure H-3 (Part 1 of 6). IBM Command Statement Formats

```
                     ┌,mmddyy┐
DISPLAY filename     │,ddmmyy│
                     └,yymmdd┘
        or
                     ┌mmddyy┐
DISPLAY filename,    │ddmmyy│   ,RECORD,value-1 [,value-2]
                     └yymmdd┘

                            ┌SOURCE ┐
                            │PROC   │
FROMLIBR library-name,      │LOAD   │ ,  ┌filename-1  ┐ ,
                            │SUBR   │    └library-name┘
                            └LIBRARY┘

    ┌I1┐       ┌ADD           ┐
    └F1┘--,---│retention-days│       ,vol-id
     \ \      └1             ┘
      \ \
       \ \   ┌P┐  ┌,blocks┐
        \ \ ,│T│  │,8     │
         \   └S┘  └,ADD   ┘     or
          \
           [,ADD]

                       ┌SOURCE ┐
                       │PROC   │
FROMLIBR  {name,ALL} , │LOAD   │   ,  ┌filename-2┐ ,
          {ALL     }   │SUBR   │      └name      ┘
                       └LIBRARY┘

    ┌I1┐       ┌ADD           ┐
    └F1┘--,---│retention-days│       ,vol-id
     \ \      └1             ┘
      \ \
       \ \   ┌P┐  ┌,blocks┐
        \ \ ,│T│  └,8     ┘
         \   └S┘
          \
           [,ADD]

           ┌ALL    ┐  ┌,RESET  ┐
HISTORY    │VIEWED │  └,NORESET┘
           └NOLIST ┘

                                                ┌,RENAME ┐
INIT    ┌vol-id     ┐ ,  ┌owner-id┐             │,DELETE │
        └system-date┘    └OWNERID ┘             │,FORMAT │
                                                └,FORMAT2┘

JOBSTR {filename} , [procname,  ┌SAVE  ┐]] [,  number of records]
       {   *    }               └NOSAVE┘          500
```

Figure H-3 (Part 2 of 6). IBM Command Statement Formats

```
KEYBOARD  {nnn}

LINES  [number]
       [66    ]

               [,SOURCE ]
               [,PROC   ]
LISTLIBR DIR   [,LOAD   ]
               [,SUBR   ]
               [,LIBRARY]
      or
LISTLIBR DIR,SYSTEM
      or
                                    [,SOURCE ]
           { library-name }         [,PROC   ]
LISTLIBR   { name, ALL    }         [,LOAD   ]
           { ALL          }         [,SUBR   ]
                                    [,LIBRARY]

LOG    [CRT    ]  [,EJECT  ]
       [PRINTER]  [,NOEJECT]

MRJE   [filename for TDISKPR1] ,  [number of blocks for TDISKPR1] ,
                                  [number of blocks for PDISKPR1]
                                  [,number of blocks for PDISKPU1]

                          [mmddyy]
ORGANIZE filename-1,      [ddmmyy]     ,F1,filename-2,
                          [yymmdd]

   [T]
   [S]   [,position,character]
   [P]
      or
                          [mmddyy]
ORGANIZE filename-1,      [ddmmyy]  ,  [I1]   ,vol-id,
                          [yymmdd]

   [retention-days]  [,position,character]
   [1             ]

                                 { C }
                                 { P }           { AA }
OVERRIDE [ADDR-nn]    [ ,LINE-   { R } ]  [,SWTYP-  { MA } ]
                                 { S }           { MC }
                                 { T }
```

Note: At least one parameter must be given in each OVERRIDE statement.

Figure H-3 (Part 3 of 6). IBM Command Statement Formats

```
REBUILD

                   [mmddyy]   [,filename ]
RELOAD [vol-id],   [ddmmyy]   [,#LIBRARY ]
                   [yymmdd]

                                [,SOURCE ]
           { library-name }     [,PROC   ]
REMOVE     { name,ALL     }     [,LOAD   ]
           { ALL          }     [,SUBR   ]
                                [,LIBRARY]

                                    [,mmddyy]
RENAME filename-1, filename-2       [,ddmmyy]
                                    [,yymmdd]

                   [filename-1]   [,mmddyy]
RESTORE [ALL],     [#SAVE     ]   [,ddmmyy]
                                  [,yymmdd]

        or

                         [mmddyy]   [,RECORDS,value-1]
RESTORE filename-2,      [ddmmyy]   [,BLOCKS,value-2 ]
                         [yymmdd]

SAVE [ALL],   [retention-days] ,   [filename-1]    ,vol-id
              [1             ]     [#SAVE     ]

        or

                    [retention-days]     [mmddyy]
SAVE filename-2,    [1             ] ,   [ddmmyy]    ,vol-id
                    [ADD           ]     [yymmdd]

                                     [MDY]    [,mmddyy]
SET    [value]  ,  [source-name] ,   [DMY]    [,ddmmyy]
                                     [YMD]    [,yymmdd]
```

Note: At least one parameter must be specified in each SET statement.

```
SETMICR      CYCLE- [Y]
                    [N]

                              { C }               {AA}
SPECIFY   [ADDR-nn]  [,Line-  { P }] [,SWTYP-     {MA}] [,ID-nnnnn]
                              { S }               {MC}
                              { T }
```

Note: At least one parameter must be specified in each SPECIFY statement.

```
STATUS

           [PRINTER]
SYSLIST    [CRT    ]
           [OFF    ]

                                [mmddyy]
TOLIBR filename,   [F1]    ,    [ddmmyy]     [,REPLACE]
                   [I1]         [yymmdd]
```

Figure H-3 (Part 4 of 6). IBM Command Statement Formats

```
                                    ┌mmddyy┐
TRANSFER filename-1,  [I1] ,        │ddmmyy│  ,ADD,
                                    └yymmdd┘

  ┌filename-2┐  [,date]
  └filename-1┘

          or
                                    ┌mmddyy┐
TRANSFER filename-1,  [I1] ,        │ddmmyy│ , [NOADD] ,
                                    └yymmdd┘

  { value-1,value-2 }  ┌,RECORDS,value-3┐
  {        ,        }  └,BLOCKS,value-4 ┘

          or
                            ┌mmddyy┐
TRANSFER filename-1,F1,     │ddmmyy│  ,vol-id   ┌,retention-days┐
                            └yymmdd┘            └,1             ┘

WCU  ┌libname┐ , ┌specname┐ ┌,Y┐
     └WPL    ┘   └KEYBOARD┘ └,N┘

WCUS filename,  ┌specname┐ , ┌Y┐ ┌,RECORDS,value-1┐
                └KEYBOARD┘   └N┘ └,BLOCKS,value-2 ┘

WPBELT  { nn }

| WCPKPT

WPDELETE libname

WPFREE  ┌libname┐, ┌key ┐ , [group] , [docname] , ┌yymmdd-1┐ ┌ yymmdd-2┐
        └WPL    ┘  └NULL┘                        └000000  ┘ └'999999  ┘

WPIMID  { nnn }

WPINIT  ┌libname┐, ┌records-1┐ ┌ records-2┐
        └WPL    ┘  └80       ┘ └'80       ┘

WPKEY

| WPLMAINT

WPMCCARD [ERASE]
```

Figure H-3 (Part 5 of 6). IBM Command Statement Formats

```
WPPRINT  [libname] , [key ]  ,ACCEPT , [group] , [docname] ,
         [WPL    ]   [NULL]

         [yymmdd-1] [  yymmdd-2]
         [000000  ] [' 999999  ]

      or

WPPRINT  [libname] [,key ]
         [WPL    ] [,NULL]

      or

WPPRINT   ,$$$$

WPPURGE  [libname]  [  records-1]
         [WPL    ]  [' 80       ]

WPRESTOR  [libname]    ,ACCEPT , [group] , [docname] ,
          [WPL    ]

          [yymmdd-1] [,yymmdd-2]
          [000000  ] [,999999  ]

WPSAVE  [libname]  ,  [retention days]  , [vol-id]  ,ACCEPT
        [WPL    ]     [999           ]

          [group] , [docname] , [yymmdd-1] [,yymmdd-2]
                                [000000  ] [,999999  ]

WPSET   { nn }

        [KB]
WPSTART [MC] [  libname]
        [FD] [' WPL    ]
        [DL]

{ ,, }  WPSTOP
{ // }
```

Figure H-3 (Part 6 of 6). IBM Command Statement Formats

```
APAR vol-id, [object program name] [,source program name]
```

where:

vol-id = Volume identification of the diskette to contain the two files APARFILE and FIXDFILE.

object program name = The name of the object program causing the problem.

source program name = The name of the source program from which the object program causing the problem was created.

```
          ⎧ SC1nn ⎫ ⎡,OLD            ⎤
          ⎪ RG1nn ⎪ ⎢,ALL            ⎥
APPLYPTF  ⎨ UT1nn ⎬ ⎣,ptf log number ⎦
          ⎪ UT2nn ⎪
          ⎪ FO1nn ⎪
          ⎩ AS1nn ⎭
```

where:

SC1nn = PTFs that change the SCP are applied; nn is the version number (release number) of the system.

RG1nn = PTFs that change the RPG II program product are applied; nn is the version number of the product.

UT1nn = PTFs that change the IBM System/32 utilities program products (DFU, SEU, sort) are applied; nn is the version number of the utility.

UT2nn = PTFs to change the IBM System/32 File Conversion Utility (FCU) program product are applied; nn is the version number of the program product.

FO1nn = PTFs to change the FORTRAN IV program product are applied; nn is the version number of the program product.

AS1nn = PTFs to change the basic assembler program product are applied; nn is the version number of the program product.

OLD = Apply PTFs to existing modules only.

ALL = Apply all PTFs from the selected PTF file.

ptf log number = Apply only the PTF corresponding to the number given. This number is the PTF log number and is indicated on the cover letter for each PTF. The PTF log number is also indicated in the PTFXREF source member on each PTF diskette.

Figure H-4 (Part 1 of 3). Service Command Formats

BUILD

$$\text{DUMP}\begin{bmatrix}\underline{\text{MAIN}}\\ \text{CONTROL}\\ \text{HISTORY}\\ \text{PTF}\\ \text{CONFIG}\\ \text{DISK}\\ \text{MICR}\end{bmatrix}\;,\begin{bmatrix}\underline{\text{PRINTER}}\\ \text{CRT}\end{bmatrix}\begin{bmatrix}\underline{\text{F1}}\\ ,\text{I1}\end{bmatrix}$$

where:

MAIN = Main storage area of the CE cylinder or the APARFILE on an APAR diskette is dumped.

CONTROL = Control storage area of the CE cylinder or the APARFILE on an APAR diskette is dumped.

Figure H-4 (Part 1 of 3). Service Command Formats (Cont.)

HISTORY = The history file from the CE cylinder or the APARFILE on an APAR diskette is dumped.

PTF = The PTF log module from the disk or the APARFILE on an APAR diskette is dumped.

CONFIG = The system configuration record from the disk, or the APARFILE on an APAR diskette, is dumped.

DISK = Area of F1 or I1 can be dumped.

MICR = The magnetic character reader controller storage from disk or the APARFILE on an APAR diskette is dumped.

PRINTER = Output is on the printer. Printer is the default.

CRT = Output is on the screen display, 240 characters at a time. The keyboard function keys can be used to display different portions of the dump.

F1 = The disk contains the information requested by the MAIN, CONTROL, HISTORY, PTF, CONFIG, or DISK parameter. F1 is the default.

I1 = The APARFILE on an APAR diskette contains the information requested by the MAIN, CONTROL, HISTORY, PTF, CONFIG, or DISK parameter.

PATCH $\begin{bmatrix}\underline{\text{F1}}\\ \text{I1}\end{bmatrix}$, [NOHEX]

where:

F1 = A disk sector is to be patched (F1 is the default).

I1 = A diskette sector is to be patched.

NOHEX = The hex representation of only unprintable characters is to be displayed. If NOHEX is not specified, the hex representation of all characters in the sector is displayed on lines 2 and 3.

TRACE $\begin{bmatrix}\underline{\text{ALL}}\\ \text{OFF}\end{bmatrix}$ [,WAIT] [,FDIOS] [,CSFDIOS] [,PUSH] [,PULL]

[,DISABLE] [,ENABLE] [,QUEUE] [,LDCS] [,LOADER] [,XIENT]
[,XFER]

Note: If either ALL or OFF is specified, ALL or OFF must be the first parameter. The remaining parameters can be specified in any order. A maximum of ten parameters can be specified.

where:

ALL = All traceable system functions are to be traced. ALL is the default.

OFF = None of the system functions are to be traced.

Figure H-4 (Part 2 of 3). Service Command Formats

WAIT = Each evocation of the wait function is to be traced.

FDIOS = Each evocation of fixed disk IOS (input/output supervisor) is to be traced.

CSFDIOS = Each evocation of control storage fixed disk IOS is to be traced.

PUSH = Each evocation of the push (rollout) function is to be traced.

PULL = Each evocation of the pull (rollin) function is to be traced.

DISABLE = Each evocation of the disable interrupt function is to be traced.

ENABLE = Each evocation of the enable interrupt function is to be traced.

QUEUE = Each evocation of the queue function is to be traced.

LDCS = Each evocation of the control storage loader is to be traced.

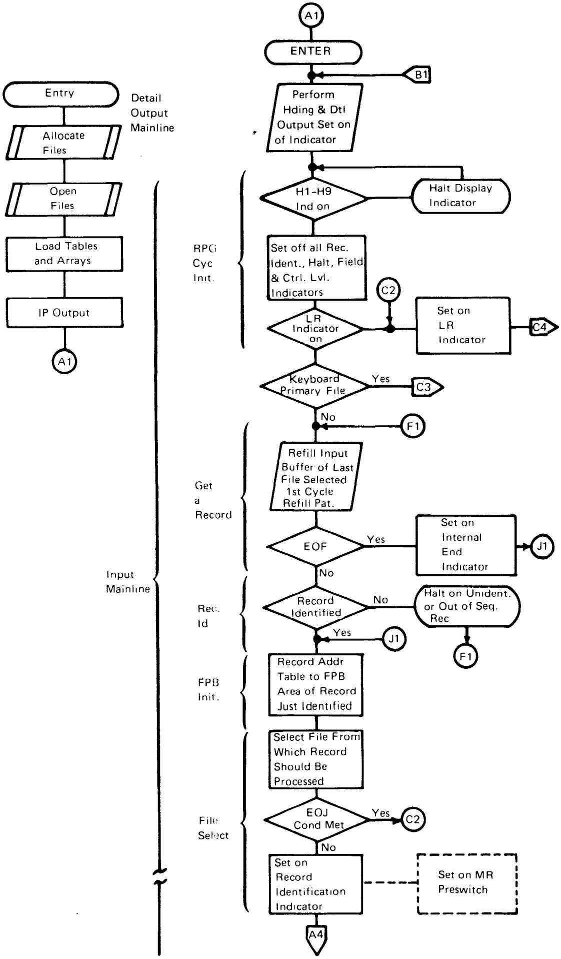
LOADER = Each evocation of the main storage loader is to be traced.

XIENT = Each evocation of the transient loader is to be traced.

XFER = Each execution of the transfer instruction is to be traced.

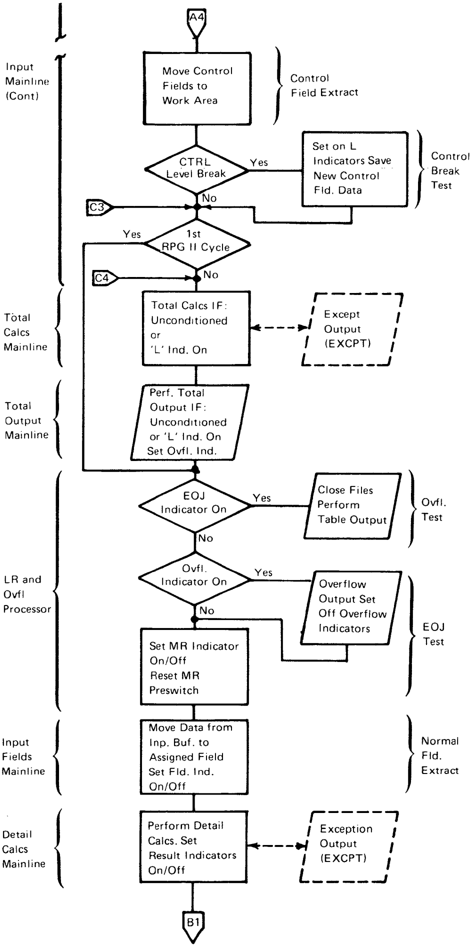
Figure H-4 (Part 3 of 3). Service Command Formats

RPG II PROGRAM OBJECT CYCLE—GENERAL (Part 1 of 2)



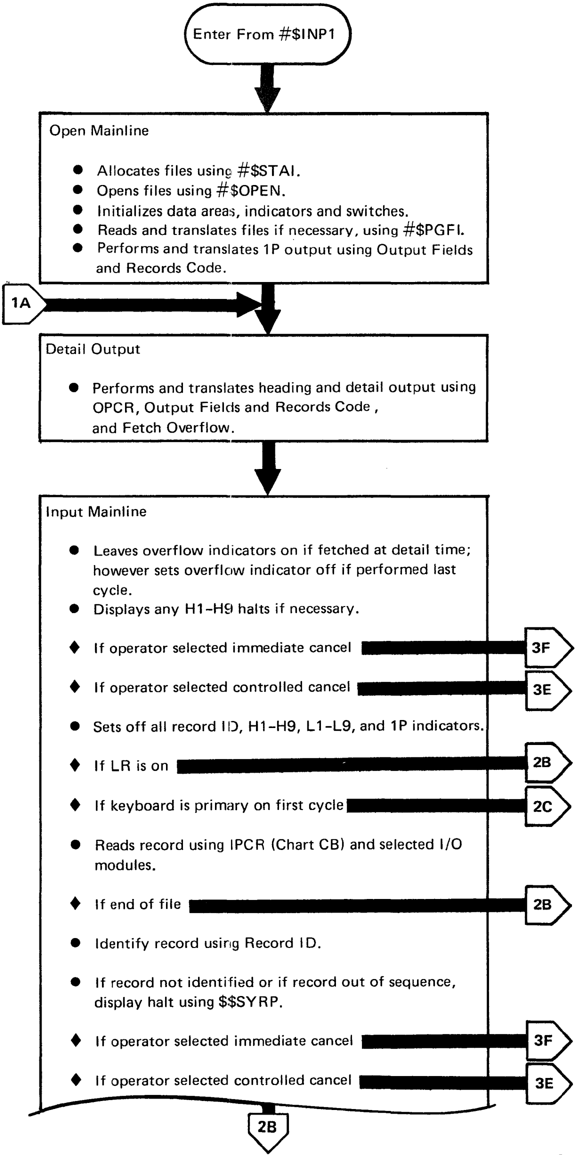
Entry
Allocate Files
Open Files
Load Tables and Arrays
IP Output
A1
Detail Output Mainline
Input Mainline
A1
ENTER
B1
Perform Hding & Dtl Output Set on of Indicator
RPG Cyc Init.
H1-H9 Ind on
Halt Display Indicator
Set off all Rec. Ident., Halt, Field & Ctrl. Lvl. Indicators
C2
LR Indicator on
Set on LR Indicator
C4
Keyboard Primary File
Yes
C3
No
F1
Get a Record
Refill Input Buffer of Last File Selected 1st Cycle Refill Pat.
EOF
Yes
Set on Internal End Indicator
J1
No
Rec. Id
Record Identified
No
Halt on Unident. or Out of Seq. Rec
F1
Yes
J1
FPB Init.
Record Addr Table to FPB Area of Record Just Identified
File Select
Select File From Which Record Should Be Processed
EOJ Cond Met
Yes
C2
No
Set on Record Identification Indicator
Set on MR Preswitch
A4

RPG II PROGRAM OBJECT CYCLE—GENERAL (Part 2 of 2)



Input Mainline (Cont)
A4
Move Control Fields to Work Area
Control Field Extract
CTRL Level Break
Yes
Set on L Indicators Save New Control Fld. Data
Control Break Test
No
C3
Yes
1st RPG II Cycle
C4
No
Total Calcs Mainline
Total Calcs IF: Unconditioned or 'L' Ind. On
Except Output (EXCPT)
Total Output Mainline
Perf. Total Output IF: Unconditioned or 'L' Ind. On Set Ovfl. Ind.
EOJ Indicator On
Yes
Close Files Perform Table Output
Ovfl. Test
No
LR and Ovfl Processor
Ovfl. Indicator On
Yes
Overflow Output Set Off Overflow Indicators
EOJ Test
No
Set MR Indicator On/Off Reset MR Preswitch
Input Fields Mainline
Move Data from Inp. Buf. to Assigned Field Set Fld. Ind. On/Off
Normal Fld. Extract
Detail Calcs Mainline
Perform Detail Calcs. Set Result Indicators On/Off
Exception Output (EXCPT)
B1

RPG II PROGRAM OBJECT PROGRAM CYCLE–DETAILED (Part 1 of 3)



Enter From #$INP1

Open Mainline

- ● Allocates files using #$STAI.
- ● Opens files using #$OPEN.
- ● Initializes data areas, indicators and switches.
- ● Reads and translates files if necessary, using #$PGFI.
- ● Performs and translates 1P output using Output Fields and Records Code.

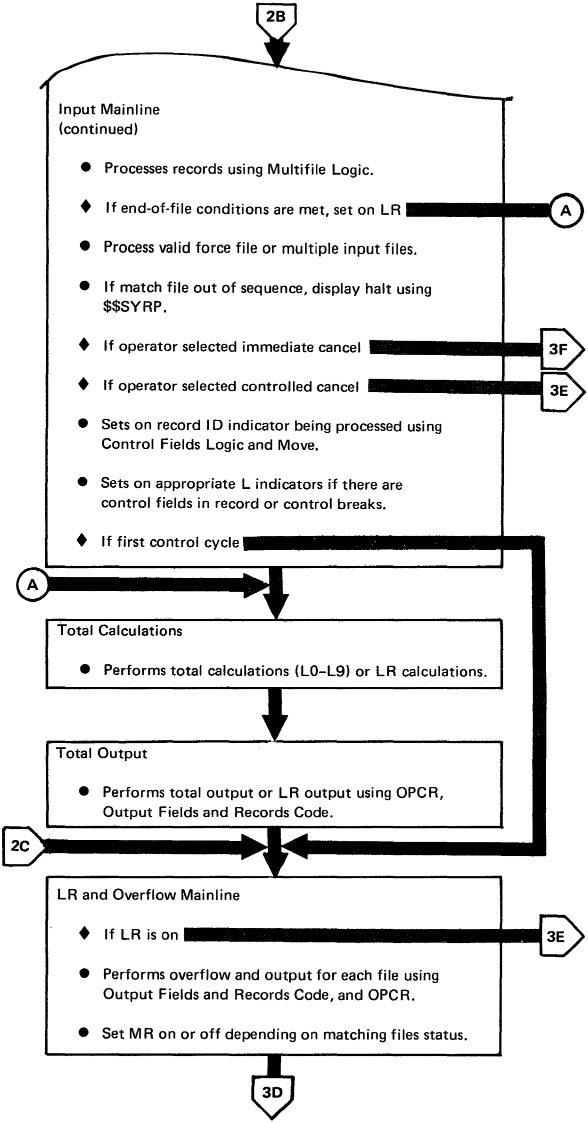
Detail Output

- ● Performs and translates heading and detail output using OPCR, Output Fields and Records Code , and Fetch Overflow.

Input Mainline

- ● Leaves overflow indicators on if fetched at detail time; however sets overflow indicator off if performed last cycle.
- ● Displays any H1-H9 halts if necessary.
- ♦ If operator selected immediate cancel
- ♦ If operator selected controlled cancel
- ● Sets off all record ID, H1-H9, L1-L9, and 1P indicators.
- ♦ If LR is on
- ♦ If keyboard is primary on first cycle
- ● Reads record using IPCR (Chart CB) and selected I/O modules.
- ♦ If end of file
- ● Identify record using Record ID.
- ● If record not identified or if record out of sequence, display halt using $$SYRP.
- ♦ If operator selected immediate cancel
- ♦ If operator selected controlled cancel

RPG II PROGRAM OBJECT PROGRAM CYCLE—DETAILED (Part 2 of 3)



Input Mainline (continued)

- ● Processes records using Multifile Logic.
- ♦ If end-of-file conditions are met, set on LR
- ● Process valid force file or multiple input files.
- ● If match file out of sequence, display halt using $$SYRP.
- ♦ If operator selected immediate cancel
- ♦ If operator selected controlled cancel
- ● Sets on record ID indicator being processed using Control Fields Logic and Move.
- ● Sets on appropriate L indicators if there are control fields in record or control breaks.
- ♦ If first control cycle

Total Calculations

- ● Performs total calculations (L0–L9) or LR calculations.

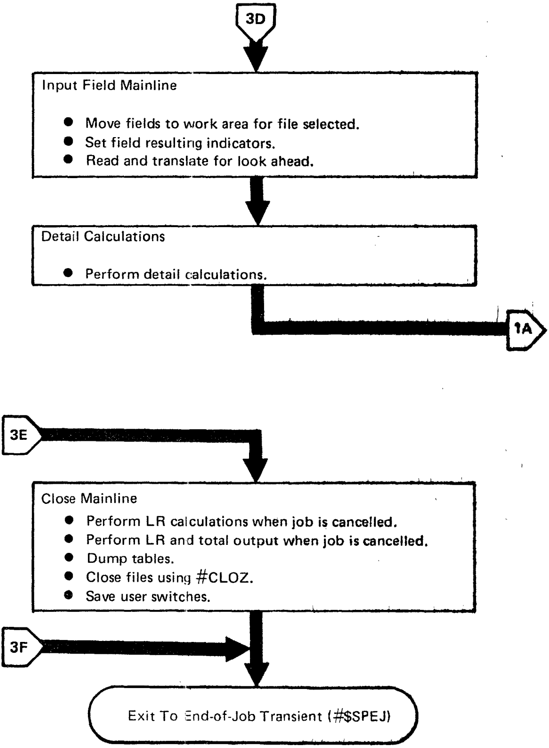
Total Output

- ● Performs total output or LR output using OPCR, Output Fields and Records Code.

LR and Overflow Mainline

- ♦ If LR is on
- ● Performs overflow and output for each file using Output Fields and Records Code, and OPCR.
- ● Set MR on or off depending on matching files status.

RPG II PROGRAM OBJECT PROGRAM CYCLE—DETAILED (Part 3 of 3)



Input Field Mainline

- Move fields to work area for file selected.
- Set field resulting indicators.
- Read and translate for look ahead.

Detail Calculations

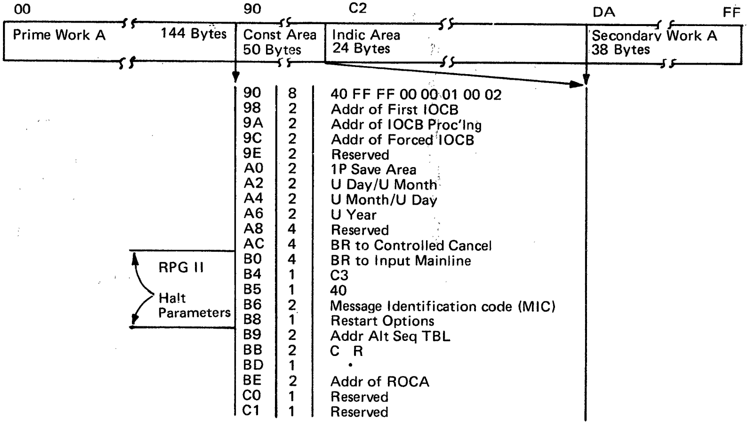
- Perform detail calculations.

Close Mainline

- Perform LR calculations when job is cancelled.
- Perform LR and total output when job is cancelled.
- Dump tables.
- Close files using #CLOZ.
- Save user switches.

Exit To End-of-Job Transient (#$SPEJ)

RPG II RESERVED OBJECT COMMUNICATION AREA



90	8	40 FF FF 00 00 01 00 02
98	2	Addr of First IOCB
9A	2	Addr of IOCB Proc'Ing
9C	2	Addr of Forced IOCB
9E	2	Reserved
A0	2	1P Save Area
A2	2	U Day/U Month
A4	2	U Month/U Day
A6	2	U Year
A8	4	Reserved
AC	4	BR to Controlled Cancel
B0	4	BR to Input Mainline
B4	1	C3
B5	1	40
B6	2	Message Identification code (MIC)
B8	1	Restart Options
B9	2	Addr Alt Seq TBL
BB	2	C R
BD	1	.
BE	2	Addr of ROCA
C0	1	Reserved
C1	1	Reserved

Indicators	File Description Specifications		Input Specifications			Calculation Specifications			Output Specifications
	Overflow (33-34)	File Conditioning (71-72)	Record Identifying (19-20)[1]	Field Record Relation (63-64)[1]	Field (65-70)	Control Level (7-8)	Conditioning (9-17)	Resulting (54-59)	Conditioning (23-31)
01-99			X	X	X		X	X	X
H1-H9			X	X	X		X	X	X
1P									X[3]
MR				X[2]			X		X
OA-OG, OV	X						X	X	X[4]
L0						X			X
L1-L9			X	X[2]		X	X	X	X
LR			X			X	X	X	X
U1-U8		X[5]		X			X	X	X
KA-KN, KP, KQ							X	X	X

Note: X denotes the indicators that can be used.

[1] Not valid on look-ahead fields.
[2] When field named is not a match field or a control field.
[3] Only for detail or heading lines.
[4] Cannot condition an exception line, but can condition fields within the exception record.
[5] Not valid for table input files.

	Hex Byte Mask							
Displacement from XR1	80	40	20	10	08	04	02	01
C2	H4	H3	H2	H1	—	MR (Int.)	MR (Ex.)	1P
C3	L1	L0	LR	H9	H8	H7	H6	H5
C4	L9	L8	L7	L6	L5	L4	L3	L2
C5	U1	U2	U3	U4	U5	U6	U7	U8
C6	KH	KG	KF	KE	KD	KC	KB	KA
C7	KQ	KP	KN	KM	KL	KK	KJ	KI
C8	Reserved							
C9	07	06	05	04	03	02	01	
CA	15	14	13	12	11	10	09	08
CB	23	22	21	20	19	18	17	16
CC	31	30	29	28	27	26	25	24
CD	39	38	37	36	35	34	33	32
CE	47	46	45	44	43	42	41	40
CF	55	54	53	52	51	50	49	48
D0	63	62	61	60	59	58	57	56
D1	71	70	69	68	67	66	65	64
D2	79	78	77	76	75	74	73	72
D3	87	86	85	84	83	82	81	80
D4	95	94	93	92	91	90	89	88
D5	—	—	—	—	99	98	97	96
D6	OV Ex.	OG Ex.	OF Ex.	OE Ex.	OD Ex.	OC Ex.	OB Ex.	OA Ex.
D7	OV 1st Int.	OG 1st Int.	OF 1st Int.	OE 1st Int.	OD 1st Int.	OC 1st Int.	OB 1st Int.	OA 1st Int.
D8	OV 2nd Int.	OG 2nd Int.	OF 2nd Int.	OE 2nd Int.	OD 2nd Int.	OC 2nd Int.	OB 2nd Int.	OA 2nd Int.
D9	Total cycle switch	Control fields processed	Overflow being processed	EOF on look-ahead	Close has been entered	* * RESERVED		

Note: For each overflow indicator there are two internal indicators. The first internal indicator indicates that overflow has occurred; the second indicator indicates that the overflow output code has been fetched.

Ex. = External   Int. = Internal

INPUT/OUTPUT CONTROL BLOCK (IOCB)

The input/output control block (IOCB) contains information about files. During the compile time phase, =RPGN builds a 17-byte IOCB for each output file and a 38-byte IOCB for each input file. The address of the first IOCB can be found at X'98' of ROCA. IOCBs are chained together with the address of the next IOCB location at bytes 2-3 of each IOCB. The chain and read record parameters are moved into bytes 24-30 by the chain and read routine. Bytes 21-37 are entered into the IOCB at object time by the record ID routine. Each IOB contains:

Byte	Bit	Contents
0	0	1 = End of file has occurred
	1	1 = File not open
	2	1 = Identify look ahead file
	3	1 = Noninput control file (not primary or secondary)
	4	1 = Translation file
	6	1 = End of file specified on file description specifications
	7	1 = Buffer full (does not need to be read from this cycle)
1	0	BSCA last file
	1	1 = File processed by limits
	2	1 = Combined file
	3	1 = Update file
	4-6	Record address type
		000 = Index file processed consecutively
		010 = Key (alphameric)
		011 = Key (packed)
		100 = ADDROUT file
		110 = Relative record number
	7	1 = Limits file (file containing limits)
2-3		IOCB chain address
4-5		DTF address
6-7		Translation table address
8-9		File relation address (from or to IOCB addresses for record address files or tables)
A		Overflow indicator mask
B-C		Record length
D-E		Address of output work area
F		Sequence number (in binary)
10		External indicator

If the file is an output file only, the following entries will not be present:

11-12	Input buffer address
13-14	Alphabetic sequence input record
15-17	Address of last numeric input record processed and sequence information (byte X'17' bit definitions are identical to those of byte X'25')

18	Communication byte: X'02' = Data fields present in records X'04' = Control fields present in records X'08' = Matching fields present in records X'10' = Numeric sequence in record X'20' = Console file X'40' = Numeric sequence in this file X'80' = Recycle check bit (if all numeric sequence checking is optional, this bit is used to determine if a record does not fit any of the numeric sequences, indicating an error)
19-1A	Resulting indicator mask and displacement
1B	Operation code for IOCS
1C	Not used
1D-1E	Address of move input fields code for this record type
1F-20	Address of control fields move code for this record type
21-22	Address of matching records moves code for this record type
23-24	Address of next numeric sequence checking code for this file
25	Numeric sequence information X'01' 1 = Numerous 0 = One X'02' 1 = Mandatory record found X'04' 1 = Mandatory record found 0 = Optional

Appendix J. Operator Processor Controls

LOAD Key

This key is pressed to start the initial program loading; IMPL followed by IPL. The LOAD light turns on when the key is pressed and remains on until the first 4096 bytes of the IMPL have been successfully loaded.

Note: After power is turned on, the LOAD key is not immediately functional due to a short power-on delay.

START Key

When the STOP light is on, the microprogram is looping and interrogating the START key. When the START key is pressed, the STOP light turns off and processing of main storage instructions continues.

Keyboard Ready Light

This light is on whenever the keyboard is enabled and ready to operate.

Processor Check Light

This light is turned on whenever an unrecoverable error is detected by the processing unit. Whenever this occurs, the only way to restart is by initiating an IMPL via the LOAD key.

POWER ON/OFF Switch

This switch initiates a power-on or power-off sequence. As part of the power-on sequence, a system reset is performed to initialize the system. At the completion of the power-on sequence (approximately 35 seconds), the STOP light will be turned on. The contents of the registers and storage are destroyed during power off.

STOP Key

When this micro instruction controlled key is pressed, the system is stopped at the end of the current main storage instruction and the STOP light is turned on. At the end of each main storage instruction, the STOP key is interrogated by the microprogram and if the key is active, the microprogram loops in a stopped state.

On power up, the STOP light turns on when the power-up sequence has been completed. It turns off when the LOAD key is pressed. The light also turns on if the microprogram is loaded and a main storage address compare stop occurs or the mode selector switch is placed in the SYS INSN STEP position.

Power Check Light

This light indicates that a check in the power system has occurred and that power has been removed. However, voltage necessary to display the check condition is still on.

Thermal Check Light

This indicator is turned on whenever an over-temperature condition is detected in the power supplies, printer, or logic gate. Power is removed from the system when the thermal check occurs. After the thermal condition has gone away, system power may again be brought up.

Mode Selector Switch

When this switch is moved from the PROC RUN position, the processing unit clock stops after execution of the current micro instruction. To restart the processing unit, return the switch to PROC RUN and press CE START. The processing unit starts at the micro instruction addressed by MAR. PROC RUN is the normal position of the mode selector switch when the system is running.

INSN STEP/DPLY LSR

With the mode selector switch in this position, each time the CE START switch is pressed the next sequential micro instruction or branch is executed. In addition, the contents of a selected LSR may be displayed. The LSR to be displayed is specified by display switches 3 and 4. The hex values '00' through '1F' will select LSRs 0 through 31 respectively.

Figure J-1 gives a description of the LSR stack. There are 32 LSRs in the LSR stack (0-31), each containing 16 data bits. (There are up to 64 LSRs with the MCU, Card I/O, or Magnetic Character Reader features.) Each LSR is divided into two parts; bits 0-7 are the high LSR and bits 8-15 are the low LSR.

The LSRs are used as data buffers and address registers for both main and control storage. In addition, they are used as operand registers for calculations and I/O control data registers that can be loaded from or sent to the I/O attachments.

The 32 LSRs are subdivided into four groups as shown. Interrupt levels 0, 1, and 2 use the MAR/MAB stack (microprogram address register/microprogram address backup). Each of the other groups of work registers can only be used by the named level.

MAR contains the address of the next micro instruction to be executed. MAB contains the return address when a branch and link instruction is executed.

	LSR		Hex
Micro interrupt 0 (main level or machine check)	WR0	0	00
	WR1	1	01
	WR2	2	02
	WR3	3	03
	WR4	4	04
	WR5	5	05
	WR6	6	06
	WR7	7	07
MAR/MAB MAR/MAB stack	MAR	8	08
	MAB	9	09
	MAR (0)	10	0A
	MAB (0)	11	0B
	MAR (1)	12	0C
	MAB (1)	13	0D
	MAR (2)	14	0E
	MAB (2)	15	0F
Micro interrupt 1 (disk)	WR0	16	10
	WR1	17	11
	WR2	18	12
	WR3	19	13
	WR4	20	14
	WR5	21	15
	WR6	22	16
	WR7	23	17
Micro interrupt 2 (keyboard, BSCA, SDLC and printer)	WR0	24	18
	WR1	25	19
	WR2	26	1A
	WR3	27	1B
	WR4	28	1C
	WR5	29	1D
	WR6	30	1E
	WR7	31	1F
Micro interrupt 3 (data recorder MCU, and magnetic character reader) *Note:* Data recorder MCU, and magnetic character reader are mutually exclusive.	WR0	32	20
	WR1	33	21
	WR2	34	22
	WR3	35	23
	WR4	36	24
	WR5	37	25
	WR6	38	26
	WR7	39	27
Micro interrupt 3 (data recorder MCU, and magnetic character reader)	MAR(3)	40	28
	MAB(3)	41	29

Figure J-1 (Part 1 of 2). LSR Stack

	LSR		Hex
SIS interpreter LSRs			
SIS IAR	Word 1	42	2A
SIS index register	Word 2	43	2B
SIS binary register	Word 1	56	38
	Word 2	57	39
SIS floating register	Word 1	58	3A
	Word 2	59	3B
SIS double operand	Word 1	60	3C
	Word 2	61	3D
	Word 3	62	3E
	Word 4	63	3F

● **Figure J-1 (Part 2 of 2). LSR Stack**

ALTER STOR

This position is used to alter the contents of main storage or control storage. It is used in conjunction with the STOR SEL switch, MAR, and the data switches. The STOR SEL switch controls whether main storage or control storage will be accessed. MAR contains the address of the location to be altered.

If main storage is being addressed, the contents of data switches 3 and 4 are stored in the addressed location. If control storage is being addressed, the contents of switches 1, 2, 3, and 4 are stored. Data switch settings are displayed in the display lights.

Pressing the CE START switch initiates the alter storage operation and causes a storage cycle to occur. During this cycle, the address in MAR is incremented by 1. Thus, it is possible to alter several sequential positions of storage without entering a new address in MAR each time.

ALTER MAR IRPT

This position of the mode selector switch allows the MAR for the current interrupt level to be altered. With the switch in this position, the 16 binary bits from the data switches 1, 2, 3, and 4 are entered into the current MAR when the CE START switch is pressed. Data switch settings are displayed in the display lights. When altering this register for the display or alter function, the initial contents of this register must be noted. This register must be reinitialized to its original value before restarting in the microprogram.

DPLY STOR

This position is used to display the contents of SDR or the contents of main storage or control storage. When displaying main storage or control storage, the mode selector switch is used in conjunction with the STOR SEL switch and MAR. Turning the mode selector switch to this position will display the current contents of SDR.

To display the contents of a position in main or control storage:

1. Set the address of the position to be displayed in MAR (alter MAR IRPT).
2. Select main or control storage with the STOR SEL switch.
3. Turn the mode selector switch to DPLY STOR position; then press CE START switch to initiate the operation.

During the storage cycle which is initiated by the CE START switch, the contents of storage are set into the SDR and displayed in the lights. MAR is incremented by 1. Thus, sequential bytes can be displayed without setting a new address into MAR each time.

When control storage is displayed, all 18 bits will appear in the lights. When main storage is displayed, the 9 bits are displayed in the rightmost byte of the display lights. The leftmost byte is not significant (contains all bits on).

INSN STEP/DPLY CHKS

Various processing unit and channel errors that occur can be displayed via the CE panel lights. These errors are recorded in the processing unit error byte and the channel check byte as seen in Figure J-2. The mode selector switch must be set to DPLY CHKS to display these two bytes. Each time CE START is pressed, the next sequential micro instruction or branch is executed.

Processor Error Byte (left byte)

Bit	Error	Cause
0	SDR parity check	Parity is incorrect in the storage data register.
1	MOR parity check	Parity is incorrect in the micro operation register.
2	Storage gate parity check	Parity is incorrect at the output of the storage gate in the data flow.
3	ALU gate parity check	Parity predicted does not agree with the generated parity at the ALU gate.
4	Invalid control storage address	Indicates that control storage was being addressed outside its boundaries.
5	Invalid main storage address	Indicates that the address being used to address main storage exceeds the system main storage size.
4 & 5	SAR parity check	Parity is incorrect in the storage address register.
6	Not used	–
7	Microcode check	Indicates that the microprocessor has been lost in a loop for six seconds.

Port Check Byte (right byte)

Bit	Error	Cause
0	DBO parity check	Incorrect parity has been detected by an I/O attachment on the DBO (data bus out).
1	Invalid device assignment	Indicates that an address has been put on the DBO but no response has been received from an attachment within the required time. (Service in must respond to control out within 5.4 µs.)
2	DBI parity check	Incorrect parity has been detected by the channel during the transfer of data from an I/O attachment.
3	Timeout check	The channel has detected an error in the normal channel sequence.
4	CBI/DBI not zero	The I/O interface lines were not cleared at the specified time.

Figure J-2 (Part 1 of 2). Port and Processor Error Bytes

Port Check Byte (right byte) - continued

Bit	Error	Cause
5	System bus parity check	Incorrect parity has been detected on the data being sent from the CPU to the channel or when data is being sent to the disk during a burst mode operation.
6	Cycle steal or burst mode operation check	If any of the CPU or channel errors listed under DPLY CHKS occur during a cycle steal operation, this bit will be turned on. In addition, if any CPU or channel parity error is detected during a burst mode operation, this bit will be turned on.
7	Invalid port address	Indicates that an invalid port address has been used.

Figure J-2 (Part 2 of 2). Port and Processor Error Bytes

INSN STEP/DPLY PCR

With the mode selector switch in this position, each time CE START is pressed the next sequential instruction or branch is executed. Also, the eight bits of the processor condition register will be displayed in the leftmost byte of the display lights (the rightmost byte is not significant). Figure J-3 gives a description of the processor condition register (PCR).

The PCR is changed by system reset, program loading, or instructions that modify register bits.

Figure J-3 (Part 1 of 2). Processor Condition Register

PCR		Flag (Bit 0)	Positive (Bit 1)	Negative (Bit 2)	Zero (Bit 3)
LA1 or LA2[1] Logical	Set	—	R1 or $\overline{R2}$ equals all ones, and the result does not equal zero.	R1 or $\overline{R2}$ does not equal all ones, or (not) R2 does not equal all ones.	Result equals all zeros.
	Reset	—	Result equals all zeros, or R1 or $\overline{R2}$ does not equal all ones.	Result equals all zeros, or R1 or $\overline{R2}$ equals all ones.	Result does not equal all zeros.
LA1 or LA2 Arithmetic	Set	—	Result has a carry and does not equal zero.	Result has no carry and does not equal zero.	Result equals zero
	Reset	—	Result has no carry or equals zero.	Result has a carry or equals zero.	Result does not equal zero.
Test Mask	Set	—	Tested bits equal all ones.	Tested bits do not equal all ones and do not equal all zeros.	All tested bits equal zero (or no bits tested).
	Reset	—	Tested bits do not equal all ones.	Tested bits equal all ones or all zeros.	Tested bits do not equal zero.
Compare or Subtract Immediate	Set	—	Register data is greater than Immediate data.	Register data is less than Immediate data.	Reg data is equal to Immediate data.
	Reset	—	Register data is not greater than Immediate data.	Register data is not less than Immediate data.	Reg data is not equal to Immediate data.
I/O Immediate Reset Carry—Set Equal (Bits 0-3 not applicable)					
I/O Immediate Load PCR	Set	Loaded bit 0 is on.	Loaded bit 1 is on.	Loaded bit 2 is on.	Loaded bit 3 is on.
	Reset	Loaded bit 0 is off.	Loaded bit 1 is off.	Loaded bit 2 is off.	Loaded bit 3 is off.

[1] For logical operations, two things are done: (1) The logical operation is performed (OR, AND, EXCLUSIVE OR, etc). (2) The contents of R1 are ORed with the 1's complement of the contents of R2, expressed as (R1 or $\overline{R2}$). Positive (bit 1) and negative (bit 2) are set to reflect the outcome of *both* operations.

PCR		Flag (Bit 0)	Positive (Bit 1)	Negative (Bit 2)	Zero (Bit 3)
System Reset	Set	–	–	–	–
	Reset	Set off.	Set off.	Set off.	Set off.
I/O Immediate Flag Latch	Set	Set on.	–	–	–
	Reset	Set off.	–	–	–

PCR		Carry (Bit 4)	High (Bit 5)	Low (Bit 6)	Equal (Bit 7)
LA1 or LA2 Logical (Bits 4-7 not applicable)					
LA1 or LA2 Arithmetic	Set	Result had a carry (add), or no borrow (subtract).	Result has a carry and does not equal zero.	Result has no carry and does not equal zero.	–
	Reset	Result has no carry (add) or a borrow (subtract).	Result has no carry or equals zero.	Result has a carry or equals zero.	Result does not equal zero.
Test Mask (Bits 4-7 not applicable)					
Compare or Subtract Immediate (Bits 4-7 not applicable)					
I/O Immediate Reset Carry–Set Equal	Set	–	–	–	Equal set on.
	Reset	Carry set off.	Decoded from carry and equal, and set off.	Decoded from carry and equal and set off.	–
I/O Immediate Load PCR	Set	Loaded bit 4 is on.	Loaded bit 4 on and bit 7 off.	Loaded bit 4 off and bit 7 off.	Loaded bit 7 is on.
	Reset	Loaded bit 4 is off.	Loaded bit 4 off or bit 7 on.	Loaded bit 4 on or bit 7 on.	Loaded bit 7 is off.
System Reset	Set	–	–	–	Equal set on.
	Reset	Carry set off.	Decoded from 4 and 7, and set off.	Decoded from 4 and 7 and set off.	–
I/O Immediate Flag Latch (Bits 4-7 not applicable)					

Figure J-3 (Part 2 of 2). Processor Condition Register

SYS INSN STEP

When the mode selector switch is in this position, the STOP light comes on. Each time the START switch is pressed and released, one main storage instruction is executed. One exception to this is the supervisor call instruction (SVC), an instruction which is not executed. The alter/display function is invoked each time the START key is pressed. Display is from the main storage address set in the data switches.

If the processing unit was running when the mode selector switch was moved to the SYS INSN STEP position, the CE START switch must be pressed and released to complete the main storage instruction that was in progress.

Address—Data—Display Switches

These switches are used in conjunction with several positions of the mode selector switch. They are used to enter addresses and data into main storage or control storage. In addition, they are used to address the LSRs. Their specific use is covered under the various positions of the mode selector switch.

The switches have the following meanings when on during IMPL:

1	2	3	4	Description
F	F	0	0	Bypass all wrap tests.
F	F	x	x	Run only wrap test(s) xx, where: xx = 80—Diskette 08—Display screen 04—Keyboard 02—Printer 01—Disk Bit on means run wrap test. Any combination of bits may be used.
F	E	x	x	Loop on routine xx and bypass errors. xx is any routine from 13 through 64. (Routine 64, control storage test, cannot bypass errors.)
F	D	x	x	Loop on routine xx. xx is any routine from 13 through 64. Only applicable when IMPL from diskette.
F	C	0	1	Loop on first 2K words and bypass errors.
F	C	0	2	Loop on first 4K words and bypass errors.
F	B	0	1	Loop on first 2K words.
F	B	0	2	Loop on first 4K words.
F	A	0	1	Stop after executing first 2K words.
F	A	0	2	Stop after executing first 4K words.
F	9	0	x	Load code from track x, where x is from 4 through 7[1].
F	8	0	0	Load DCP and diskette diagnostics from the disk.
F	7	x	x	Same function as FFxx, but no display on display screen.
F	1	0	0	Run keyboard diagnostics.

[1] When the F9xx option is used, data is loaded into control storage beginning at address X'0080' and executes that code. You can enter any microcode on these tracks on the DIAG 01 diskette by using the diagnostic program DSPA33FD. Thus, you can load and execute any microcode.

Proc Interrupt Lights

These lights indicate which interrupt level is currently in progress. The interrupt is indicated in the lights as follows:

Lights			Interrupt
4	2	1	
		X	Level 1
	X		Level 2
	X	X	Level 3
X	X	X	Level 0

All lights off indicate main level.

Micro Interrupt Priority by Device

Level 0	Level 1	Level 2	Level 3
Machine check	62GV	Keyboard, Printer, BSCA, SDLC	Data Recorder, MCU, Magnetic Character Reader (data recorder, MCU, and magnetic character reader are mutually exclusive)

RESET Switch

Pressing this switch causes the following to occur:

1. The microcode address register (MAR) is initialized to X'0000'.

2. Present power fault conditions are transferred to previous power fault conditions and the present power fault condition latches are reset.

3. The processing unit timing circuitry is set to an initialized state.

4. Error and status indicators are reset.

5. The PCR is initialized to the equal condition.

To restart the system, either the CE START switch or the LOAD key must be pressed.

CE START Switch

Pressing this switch causes execution of instructions to begin at the address specified by the current micro address register (MAR).

LAMP TEST Switch

When this switch is pressed, all system lights are turned on unless they or their circuits are defective.

Display Intensity Control

This control adjusts the intensity of the display screen.

STOR SEL Switch

This switch controls whether main storage or control storage is addressed on manual operations or address compare operations. When addressing main storage on a manual operation or address compare operation, the switch must be in the MAIN position. To address control storage on a manual operation or address compare operation the switch must be in the CTL position.

IMPL-IPL Switches

The IMPL and IPL switches select the IMPL (initial microprogram load) and IPL (initial program load) devices (disk and diskette). During normal operations both IMPL (control storage load) and IPL (main storage load) are from the disk. When the LOAD key is pressed, control storage is first loaded with system diagnostic tests from the IMPL device. After successful completion of these tests, the emulator and SCP (system control program) are loaded. The microprogram then loads main storage from the selected IPL device.

FORCE CLOCK Switch

This switch initiates continuous storage cycles when in the alter storage mode. With the mode selector switch in the ALTER STOR position, the contents of the data switches are transferred to consecutive main or control storage locations (depends on STOR SEL switch setting) when the force clock switch is turned on. The starting address is contained in the MAR. Turning the switch to OFF terminates the operation.

CHECK RUN/STOP Switch

This switch controls whether the system runs or stops when a parity error occurs. If the switch is in the STOP position, the system stops at the end of the current micro instruction when a parity error occurs. If in the RUN position, the error is retained but the system continues to run.

Clock Light

This indicator is turned on by the run latch or the block processor clock signal from an I/O device.

PWR FAULT DISP Switch and DPLY PWR CHK Switch

When a system power failure occurs, the power supply at fault and the type of failure are stored in latches. These latches are on the power sequence card and retain their information as long as the main line switch is kept on. These latches are known as the present power fault latches.

When the RESET switch is pressed and console power brought up, the information recorded in the present power fault latches is transferred to another set of latches known as the previous power fault latches. Thus it is possible to have stored in latches the reasons for a current power failure and a preceding power failure. The contents of the power fault latches may be displayed even though console power is down. To display the present power fault latches, the PWR FAULT DISP switch must be in the PRES position when the DPLY PWR CHK switch is pressed. If the PWR FAULT DISP switch is in the PREV position, the previous power fault latches are displayed.

The power fault conditions are displayed in the leftmost byte on the CE panel. The meaning of the specific bits is as follows:

Bits 0 and 1:

01 = under voltage
10 = over voltage
11 = over current

Bits 2 and 3:

01 = multi-level supply
11 = dual-level supply

Bits 4 through 7:

0001 = –4V at fault	Multi-level supply
0010 = +5V at fault	
0011 = -5V at fault	
0100 = +6V at fault	
0101 = +8.5V at fault	
0110 = +12V at fault	
0111 = –12V at fault	
1000 = +24V at fault	Dual-level supply
1001 = –24V at fault	

1111 = Both levels failing in dual-level supply or +5V, +8.5V and –12V failing in the multi-level supply

ADD COMP STOP/RUN Switch

ADD COMP RUN

The address compare run switch is used in conjunction with the STOR SEL switch and the address switches. An address compare sync signal (A-A1J2D12) is provided whenever the address switches match an address in SAR. The STOR SEL switch determines whether the sync occurs on a main storage or control storage address.

ADD COMP STOP

The address compare stop switch is used in conjunction with the STOR SEL switch and the addresses switches. The system stops and an address compare sync signal (A-A1J2D12) is provided whenever the address switches match an address in the SAR. The STOR SEL switch determines whether the stop and sync will occur on a main storage or control storage address. The exact time at which the system will stop is determined by the following considerations:

1. If the address compare is on a main storage address, the emulator completes the main storage instruction being executed and then stops the system with the stop key light on. The system may be restarted by pressing the operator panel START key.

2. With the exception of I/O operations, an address compare on a control storage address stops the processing unit clock after executing the micro instruction at that address. To restart, press the CE START switch. If an address compare stop is made on a control storage address during the execution of a system I/O instruction, the results of the system instruction are unpredictable.

Console Display Lights

These are lights used in conjunction with the MODE SELECTOR switch or the COMM DPLY switch.

During IMPL, the leftmost nine lights indicate the following:

- When the LOAD key is pressed, all nine lights plus the LOAD light come on.
- The lights are turned off as the load sequence is performed.

Hardware Reset

P—Goes off when diskette or disk attachment has received IMPL signal and responded with block processor clock.
0—Goes off when first cycle steal request has been received by the processor (write trigger).
1—Goes off when data transfer is completed (4096 bytes). ALU bit 4 will come on as a result of SAR going to 07FF. 2K words have been written into control storage.
Load light—Goes off if the data has been read correctly (no proc check and good CRC).

Load 1 Reset

2—Goes off on load 1, routine 02.
3—Goes off at the beginning of the loader routine at the end of load 1.

Load 2 Reset

4—Goes off at the beginning of load 2, routine 36. This means load 1 has successfully executed and load 2 has been read in.
5—Goes off at the beginning of load 2, routine 64. Main storage is now checked out.
6—Goes off at the end of routine 64 in load 2. Control storage is now checked out.

Load 3 Reset

7—Goes off during wrap test supervisor in load 3.

COMM DPLY (Communications Display) Switch

This switch is on the CE panel if BSCA or SDLC is installed on the system. When set at the ON position, the switch activates the leftmost six lights on the bottom of the CE panel. The lights (numbers 0 through 5 from left to right) indicate the following about data communications, if on:

0 — Data terminal (remote) ready
1 — Data set (local) ready
2 — Request to send
3 — Clear to send (communications link is established)
4 — Send data
5 — Receive data

When set at the OFF position, the console display lights are under control of the MODE SELECTOR switch.

Appendix K. Higher-Level Languages

A large portion of the system programs are written in two higher level languages, Structured Program Language (SPL) and Program Language for Systems (PLS). A brief description of PLS and SPL statements and expressions follows:

1. Comment—Comments are enclosed by the characters /* on the left and */ on the right.

   Example: /* THIS IS A COMMENT */

2. Label—Labels reference addresses in main storage.

   Example: LABEL:

3. Variable—Used to reference data areas.

   Example: FIELD

4. Expression—Combination of variables separated by relational operators.

   Example:
   ```
   A + B
   A > B (comparison expression)
   ```

5. Assignment—Assigns the value of an expression to a variable.

   Example:
   ```
   A = B
   A = A + B
   ```

6. GOTO and CALL—Transfers control to the location (label) specified.

   Example:
   ```
   GOTO    AB1
   CALL    AB1
   ```

7. DO and END—A means of grouping statements (called a DO group) with optional conditional execution control. END terminates the group.

   a. DO;—Specifies a nonlooping DO group executed once.

   Example:
   ```
   DO;
      A = B + C;
      B = B + D;
   END;
   ```

   b. DO WHILE (comparison expression);—A looping DO group which specifies a test by means of a comparison expression at the beginning of the DO group. The DO group is executed until the test is false.

   Example:
   ```
   DO WHILE (A < B);
    ↓
   END;
   ```

   c. DO UNTIL (comparison expression);—A looping DO group which specifies a test by means of a comparison expression at the end of the DO group. If the test is false, the DO group is executed again.

   Example:
   ```
   DO UNTIL (A > B);
    ↓
   END;
   ```

8. IF (comparison expression) THEN statement; ELSE statement;—Where statement is a single statement or a DO group.

Example:

```
IF A>= B THEN
   DO;
      A = A + 1;
   END;
ELSE
   A = B;
```

9. PROC and END—A means of grouping statements. Defines the main program as well as subprograms.

Example:

```
PROC;
   A = B;
PROC;
   B = C;
END;
END;
```

10. RETURN—Transfers control to the end of the current procedure.

11. GENERATE (GEN) and @ENDGEN;—A means of grouping assembler statements in the high-level statements. @ENDGEN terminates the group. GENERATE DATA generates assembler code at the end of the main program.

Example:

```
   GEN DATA;
ABC DC XL2'00'
   @ENDGEN;
GEN(LA A,1);
```

12. DCL—Defines data areas for PLS.

SPL and PLS operators are defined below:

:	terminates a label
;	terminates a statement
>	greater than
<	less than
=	equal
¬	not
+	addition
-	subtraction
REG	register
→	expression is indexed
(a:b)	length and offset
addr	address of
based	based on the value of

Appendix L. Library Load Modules

This appendix contains a list of the library load modules. The list is intended to be used as an aid for identifying the main storage ID with the associated directory name. The list also contains a reference to the program logic manual which describes the modules.

Use the following key to find the program logic manual in which the modules are described:

ASM = *IBM System/32 Basic Assembler and Macro Processor Logic Manual,* LY21-0566

DC = *IBM System/32 Data Communications Logic Manual,* SY21-0551

DFU = DFU chapter in *IBM System/32 Utilities Program Product Logic Manual,* LY21-0539

FCU = *IBM System/32 File Conversion Utility Program Product Logic Manual,* LY34-0074

MCR = *IBM System/32 1255 Magnetic Character Reader Reference and Logic Manual*, SC21-7692

RPG = *IBM System/32 RPG II Logic Manual,* LY21-0538

SEU = SEU chapter in *IBM System/32 Utilities Program Product Logic Manual,* LY21-0539

SORT = Sort chapter in *IBM System/32 Utilities Program Product Logic Manual,* LY21-0539

SYS = *IBM System/32 System Logic Manual,* SY21-0567

WP = *IBM System/32 Word Processing Logic Manual,* SY34-0069

Main Storage ID	Directory Name	Descriptive Name	Manual
$COADD	$COADD	Add to diskette file	SYS
$COALL	$COALL	Copy all files	SYS
$COGET	$COGET	Copy read/write	SYS
$COINT	$COINT	Copy initialization	SYS
$COMSK	$COMSK	Unordered key sort	SYS
$COPRT	$COPRT	Copy print/display	SYS
$COPY	$COPY	Disk copy/display mainline	SYS
$COSEL	$COSEL	Record selection/deletion	SYS
$COZIP	$COZIP	Sectorized data management interface	SYS

Main Storage ID	Directory Name	Descriptive Name	Manual
$LAB	$LABEL	VTOC display utility	SYS
CDIF	#CDIF	Flush index and update format 1	SYS
CDUF	#CDUF	Update format 1 and restore DTF	SYS
CLOZ	#CLOZ	Common close	SYS
#GSC	#GSORT	Phase 0A	SORT
ROFX	#ROFX	Rolled out SWA read	SYS
#SEU	#SEU	Common	SEU
#WP@		Word Processing Load Module	See note.

Note: If you encounter a library load module with the prefix #WP@, see the *IBM Word Processor/32 Program Logic Manual*, LH30-0115.

Main Storage ID	Directory Name	Descriptive Name	Manual
#99G	#99#G	Scratch file deallocator	SYS
ACMP	$MACMP	Library directory compactor	SYS
ACOM	$MACOM	Library open/close	SYS
ACND	$MACND	Library members area condense	SYS
ADLT	$MADLT	Library directory entry delete	SYS
ADSP	$MADSP	Library copy to print	SYS
AFND	$MAFND	Library find	SYS
AHLT	$MAHLT	Library open/close halt	SYS
AILD	$MAILD	Library directory insert	SYS
AIST	$MAIST	Library directory fast insert	SYS
AINT	$MAINT	Library mainline	SYS
ALOC	$MALOC	Library expand/contract	SYS
ALTL	$MALTL	Library to library copy	SYS
APAR	$FEAPR	APAR	SYS
APGS	$MAPGS	Library sector get/put	SYS
APTF	$MAPTF	PTF log handler	SYS
APUR	$MAPUR	Library record put	SYS
AP2F	$MAP2F	PTF log update/replace	SYS

Main Storage ID	Directory Name	Descriptive Name	Manual
AP3F	$MAP3F	PTF log expansion	SYS
ARDR	$MARDR	Library copy from reader	SYS
ARFF	$MARFF	Library copy from file—record mode	SYS
ARTF	$MARTF	Library copy to file—record mode	SYS
ATFS	$MATFS	Library copy to file—sector mode	SYS
ATLS	$MATLS	Library copy from file—sector mode	SYS
AUTO	#AUTO	/COPY and merge	RPG
AU00A	#AU00A	Catalog generated code	RPG
AU00B	#AU00B	Print message and call compiler	RPG
AU00C	#AU00C	Generate output specifications (H-*AUTO)	RPG
AU00D	#AU00D	Generate heading, detail specifications (D/T-*AUTO)	RPG
AU00E	#AU00E	Generate total specifications (D/T-*AUTO)	RPG
AU002	#AU002	Sort sequence and build name table	RPG
AU003	#AU003	Define field table names and build table	RPG
AU004	#AU004	H-*AUTO line specifications and diagnostics	RPG
AU005	#AU005	D/T-*AUTO line specifications and diagnostics	RPG
AU006	#AU006	Unadjusted D/T-*AUTO line lengths	RPG
AU007	#AU007	Generate calculation specifications for totaling	RPG
AU008	#AU008	Control generation of output specifications	RPG
AU009	#AU009	Create source program input to compiler	RPG
AXNT	#WP#AX	JJT load/save high core	WP
BACK	$BACK	Backup library utility	SYS
BDEF	#WP#BR	Keyboard redefine transient	WP

Main Storage ID	Directory Name	Descriptive Name	Manual
BICR	$BICR	Basic data exchange utility	SYS
BIKE	#IKBR1	IDE interrupt	SYS
BR12	#SBR12	Data recorder data management	SYS
BSCL	#BSCL	Close	DC
BSL0	#BSL0	Line initialization	DC
BSL1	#BSL1	Line initialization	DC
BSL2	#BSL2	Line initialization	DC
BSL3	#BSL3	Line initialization	DC
BSL4	#BSL4	Line initialization	DC
BSOB	#BSOB	Open	DC
BUIL	$BUILD	Alternate sector rebuild	SYS
CBS	$WCBS	WPCU communications BSC data management	WP
CCAR	#FCCAR	FCUCONV format line processor	FCU
CCCS	#FCCCS	FCUCONV case conversion	FCU
CCDR	#FCCDR	FCUCONV direct random I/O interface	FCU
CCFP	#FCCFP	FCUCONV sequential/direct/ indexed I/O interface	FCU
CCIR	#FCCIR	FCUCONV indexed random I/O interface	FCU
CCMC	#FCCMC	FCUCONV mag card interface	FCU
CCNM	#FCCNM	FCUCONV numeric conversion and summary routine	FCU
CCRQ	#FCCRQ	FCUCONV record qualification	FCU
CCSM	#FCCSM	Summary line processor	FCU
CCVR	#FCCVR	FCU conversion processor	FCU
CC1I	#FCC1I	Storage allocation and data prompter	FCU
CC2I	#FCC2I	I/O data management load and initialization	FCU

Main Storage ID	Directory Name	Descriptive Name	Manual
CDR1	#CDR1	Diskette close	SYS
CDZL	#CDZL	Sector mode close	SYS
CEDT	#FCEDT	FCUEDIT mainline	FCU
CEDU	#FCEDU	FCUEDIT direct file update	FCU
CEIN	#FCEIN	FCUEDIT input	FCU
CEIU	#FCEIU	FCUEDIT indexed file update	FCU
CID4	#CID40	IDE data management	SYS
CID6	#CID60	Record address limits processor	SYS
CIN	$WCIN	WPCU communications initialization	WP
CLI	$WCLI	WPCU communications LDAM I/O interface	WP
CR01	$WCR01	WPCU receive translate table—keyboards 001-008	WP
CR09	$WCR09	WPCU receive translate table—keyboard 009	WP
CR17	$WCR17	WPCU receive translate table—keyboard 017	WP
CR18	$WCR18	WPCU receive translate table—keyboard 018	WP
CSA2	#FCSA2	FCUSPEC F processor	FCU
CSB2	#FCSB2	FCUSPEC Q processor	FCU
CSC2	#FCSC2	FCUSPEC C processor	FCU
CSD2	#FCSD2	FCUSPEC A/. processor	FCU
CSI	$WCSI	WPCU communications sequential I/O interface	WP
CSPC	#FCSPC	FCUSPEC mainline	FCU
CS10	#FCS10	FCUSPEC source Specs 1. file and list	FCU
CS20	#FCS20	FCUSPEC diagnose spec statements	FCU
CS30	#FCS30	FCUSPEC complete and build format description	FCU

Main Storage ID	Directory Name	Descriptive Name	Manual
CUM	$WCUM	WPCU communications utility mainline	WP
CX01	$WCX01	WPCU transmit translate table—keyboards 001-008	WP
CX09	$WCX09	WPCU transmit translate table—keyboard 009	WP
CX17	$WCX17	WPCU transmit translate table—keyboard 017	WP
CX18	$WCX18	WPCU transmit translate table—keyboard 018	WP
CVRT	$CNVRT	Convert diskette header labels utility	SYS
DCSU	$DCSUP	Print utility	DC
DELE	$DELET	File delete mainline	SYS
DELF	$DELF1	Disk file delete	SYS
DELI	$DELI1	Diskette file delete	SYS
DEOJ	#DLEOJ	End-of-job update SIO counters mainline	SYS
DLA0	#DLA0C	Disk SIO counter table	SYS
DLD0	#DLD0C	Diskette SIO counter table	SYS
DLE0	#DLE0C	Line printer counters and error history	SYS
DLE9	#DLE9C	Serial printer counters and error history	SYS
DLOG	#DLOG	Error logging mainline	SYS
DL50	#DL50C	Error logging routing	SYS
DL51	#DL51C	Data recorder error logging	SYS
DL52	#DL52C	Magnetic character reader error logging	SYS

Main Storage ID	Directory Name	Descriptive Name	Manual
DL59	#DL59C	MCU error logging	WP
DL80	#DL80C	BSCA error logging	SYS
DPRD	$DUPRD	Diskette copy utility	SYS
DS$A	#GS$A	Print COMMON—dynamic request	SORT
DS$D	#GS$D	Phase 0D parameter list	SORT
DS$E	#GS$E	Dynamic dump loader	SORT
DS$X	#GS$X	Execution phases parameter list	SORT
DSBA	#GSBA	Phase 0B	SORT
DSBC	#GSBC	Alternate sequence collating sequence table	SORT
DSCA	#GSCA	Compile the select/build routine	SORT
DSCB	#GSCB	Error logging	SORT
DSCC	#GSCC	Move generation code	SORT
DSCE	#GSCE	Include/omit generator	SORT
DSCF	#GSCF	Field generator	SORT
DSCK	#GSCK	Keyword generator	SORT
DSCL	#GSCL	Calculate length module	SORT
DSCZ	#GSCZ	End of file for compiler	SORT
DSDA	#GSDA	Phase 0D, Part A	SORT
DSDB	#GSDB	Phase 0D, Part B	SORT
DSEA	#GSEA	Phase 0E	SORT
DSGA	#GSGA	Phase 0G	SORT
DSZA	#GSZA	Variable length move	SORT
DS1D	#GS1D	NEXTDB for phase 1	SORT
DS1L	#GS1L	Phase 1L	SORT
DS1X	#GS1X	Phase 1X	SORT
DS1Z	#GS1Z	End-of-pass reporter	SORT
DS2A	#GS2A	Phase 2A	SORT

Main Storage ID	Directory Name	Descriptive Name	Manual
DS2L	#GS2L	Phase 2L	SORT
DS3A	#GS3A	Phase 3A	SORT
DS3L	#GS3L	Phase 3L	SORT
DS3S	#GS3S	Phase 3S	SORT
DS4A	#GS4A	Phase 4A	SORT
DS8A	#GS8A	Data management consecutive get	SORT
DS8G	#GS8G	Data management consecutive put	SORT
DS9G	#GS9G	Work file get/locate	SORT
DS9I	#GS9I	Read statement from sysin	SORT
DS9P	#GS9P	Work file put/locate	SORT
DS9S	#GS9S	Read statement from source member	SORT
DUMP	$FEDMP	Dump	SYS
EBLD	$REBLD	Rebuild data file	SYS
ECSF	#99#ECSF	Variable microcode location finder	SYS
EPMP	#RPMP	Build calculation segment list	RPG
EPMQ	#RPMQ	Complete calculation segment list	RPG
ETCF	$SETCF	Set utility	SYS
ESTM	$SETSM	Set utility specification module	SYS
EUDL	#SEUDL	Delete	SEU
EUDX	#SEUDX	SEU data management	SEU
EUEJ	#SEUEJ	End of job	SEU
EUEO	#SEUEO	Change format	SEU
EUET	#SEUET	Enter/update	SEU
EUIC	#SEUIC	Include	SEU
EUIN	#SEUIN	Initialization	SEU
EUMV	#SEUMV	Move	SEU

Main Storage ID	Directory Name	Descriptive Name	Manual
EURB	#SEURB	Syntax checker for calculation specifications	SEU
EURC	#SEURC	Syntax checker for calculation specifications	SEU
EURD	#SEURD	Syntax checker for calculation specifications	SEU
EURE	#SEURE	Syntax checker for extension, header, /COPY	SEU
EURF	#SEURF	Syntax checker for file description	SEU
EURG	#SEURG	Syntax checker for file description	SEU
EURI	#SEURI	Syntax checker for input specifications	SEU
EURJ	#SEURJ	Syntax checker for input specifications	SEU
EURL	#SEURL	Syntax checker for extension specifications	SEU
EURO	#SEURO	Syntax checker for output specifications	SEU
EURP	#SEURP	Syntax checker for output specifications	SEU
EURT	#SEURT	Syntax checker for telecommunications specifications	SEU
EURU	#SEURU	Syntax checker for telecommunications specifications	SEU
FIX	$FEFIX	Program temporary fix installation	SYS
FLDM	#FCCLD	FCUCONV LDAM I/O interface	FCU
FREE	$FREE	Disk reorg utility phase 0	SYS
FRE1	$FREE1	Disk reorg utility phase 1	SYS
FRE2	$FREE2	Disk reorg utility phase 2	SYS
FRE3	$FREE3	Disk reorg utility phase 3	SYS
FSCP	#FSCP	PTAM close routine	SYS
FSPO	#FSPO	PTAM open routine	SYS

Main Storage ID	Directory Name	Descriptive Name	Manual
FUDS	#DFUDS	Convert data file to source or procedure	DFU
FUEU	#DFUEU	Enter/update initialization	DFU
FUFB	#DFUFB	Build format description	DFU
FUIN	#DFUIN	Inquiry processing	DFU
FULS	#DFULS	List processing	DFU
FUML	#DFUML	Enter/update mainline processing	DFU
FUMP	#DFUMP	Build attribute specifications	DFU
FUQR	#DFUQR	Format builder query and response	DFU
FUSB	#DFUSB	Convert sort sequence specifications	DFU
FUSD	#DFUSD	Convert source or procedure to data file	DFU
FUUD	#DFUUD	Update and diagnose DFU specifications	DFU
GSBC	#GSBC	Alternating collating sequence table	SORT
HFIT	$HINT	History file initialization	SYS
HIST	$HIST	History file display	SYS
IBLD	#IGBLD	Diskette read error display and correction	SYS
IND0	#IND0	Disk file initialization, phase 0	SYS
IND1	#IND1	Disk file initialization, phase 1	SYS
IND2	#IND2	Disk file initialization, phase 2	SYS
IND3	#IND3	Disk file initialization, phase 3	SYS
INF0	#INF0	File allocation, phase 0	SYS
INF1	#INF1	File allocation, phase 1	SYS
INF2	#INF2	File allocation, phase 2	SYS
INF3	#INF3	File allocation, phase 3	SYS

Main Storage ID	Directory Name	Descriptive Name	Manual
INIT	$INIT	Diskette labeling and initialization	SYS
INKB	#LINKB	Pass 1 root and initialization	SYS
INKD	#LINKD	Pass 1 TEXT-RLD and END record processor	SYS
INKE	#LINKE	Pass 1 ESL and OPTNS record processor	SYS
INKF	#LINKF	Pass 1 phase control record processor	SYS
INKG	#LINKG	Pass 2 root phase	SYS
INKH	#LINKH	TEXT-RLD output build phase	SYS

This page intentionally left blank.

Main Storage ID	Directory Name	Descriptive Name	Manual
INKJ	#LINKJ	Overlay fetch table build	SYS
INKK	#LINKK	Error and overlay fetch table-print	SYS
INMS	#INMS	File allocation, miscellaneous	SYS
INP0	#INP0	Program initialization, phase 0	SYS
INP1	#INP1	Program initialization, phase 1	SYS
I1BT	#I1BLT	Diskette read error display and correction	SYS
I1ER	#I1ERP	Error recovery	SYS
I1IO	#$I1IO	Diskette IOS	SYS
I1IO	#$UI1D	Diskette interface	SYS
I1PR	#I1PRE	Diskette prepare	SYS
KALL	#KIKLZ	Device close	SYS
KID1	#KID10	Keyboard/CRT data management	SYS
KID5	#KID50	Field light handler	SYS
KIKE	#KIKEY	Command key handler	SYS
LINK	#OLINK	User entry phase 1	SYS
LINKM	#LINKM	ESL table process phase	SYS
LIN1	#OLIN1	User entry phase 2	SYS
LIN2	#OLIN2	User entry phase 3	SYS
LIN3	#OLIN3	User entry phase 4	SYS
LMSG	#OLMSG	Error message print phase	SYS
LOAD	$LOAD	Pseudo IPL	SYS
LYNX	#OLYNX	Compiler entry phase	SYS
MGBL	$MGBLD	Create message member	SYS
MGRE	#MGRET	Message retrieve	SYS
MICR	#MICR	Stacker specification analysis compression error message handler	MCR
MIDMC	#MIDMC	Attachment controller diagnostic	MCR

Main Storage ID	Directory Name	Descriptive Name	Manual
MIO8	#MIO8	Magnetic character reader data management	MCR
MIRMC	#MIRMC	Attachment controller stacker select and modulus check	MCR
MRBE	#MRBE	BSC error recording	DC
MRBM	#MRBM	Buffer manager	DC
MRBO	#MRBO	BSC open	DC
MRBP	#MRBP	BSC I/O processor	DC
MRCD	#MRCD	Console output processor	DC
MRCL	#MRCL	Close routine	DC
MRCO	#MRCO	Utility control statement manager	DC
MRCP	#MRCP	Console input processor	DC
MRCR	#MRCR	CARRIAGE utility control statement processor	DC
MRCS	#MRCS	Utility control statement scanner	DC
MRDP	#MRDP	Disk output processor	DC
MRFC	#MRFC	Full compression routine	DC
MRI1	#MRI1	CONFIG utility control statement processor	DC
MRI5	#MRI5	DTF allocation routine	DC
MRI6	#MRI6	Storage allocation routine	DC
MRJE	$MRJE	Dispatching supervisor	DC
MRKP	#MRKP	Reader keyboard input processor	DC
MRMO	#MRMO	MODIFY utility control statement processor	DC
MROP	#MROP	Open processing routine	DC
MRRC	#MRRC	Reader close routine	DC
MRRF	#MRRF	READFILE utility control statement processor	DC
MRRO	#MRRO	Reader open routine	DC

Main Storage ID	Directory Name	Descriptive Name	Manual
MRRP	#MRRP	Reader disk input processor	DC
MRTM	#MRTM	Terminator	DC
MRUP	#MRUP	Printer output processor	DC
MRVM	#MRVM	VTOC manager	DC
MR1	$MR1	Magnetic character reader IOCH CS microcode X	MCR
MR2	$MR2	Magnetic character reader diagnostic CS microcode transient	MCR
MSNP	#MSNIP	Main storage initialization	SYS
OADI	$LOADI	Reload library routine	SYS
OALL	#KIOPN	Device open	SYS
ODCB	#ODCB	Carve buffers	SYS
ODDI	#ODDI	Diagnostics, Part 1	SYS
ODD2	#ODD2	Diagnostics, Part 2	SYS
ODFM	#ODFM	Format	SYS
ODF1	#ODF1	Get extents from format 1 image	SYS
ODI1	#ODI1	Diskette open	SYS
ODLM	#ODLM	Obtain limits	SYS
ODNP	#ODNP	Transient resolver no-op	SYS
ODNV	#ODNV	Open next diskette volume	SYS
ODPM	#ODPM	Prime buffers	SYS
ODSB	#ODSB	Carve buffers, shared I/O	SYS
ODSO	#ODSO	Format, shared I/O	SYS
ODZP	#ODZP	Sector mode open	SYS
OLAF	#OLAF	Auto-link segment list build	SYS
OLAH	#OLAH	Cross-reference segment list build	SYS
OLAJ	#OLAJ	Sort auto-link segment list	SYS
OLAP	#OLAP	Overlay design	SYS
OLAR	#OLAR	Overlay segment list build	SYS

Main Storage ID	Directory Name	Descriptive Name	Manual
OLAT	#OLAT	Storage map phase	SYS
OLBE	#OLBE	Relocate, resolve EXTRNs, and build load module phase	SYS
OLBO	#OLBO	Library control phase	SYS
OLER	#OLER	Error routine	SYS
OMDO	#$OMDO	Offline multivolume dummy open	SYS
OMLK	#$OMLK	Offline multivolume prime buffer	SYS
OMOA	#OMOAD	Offline multivolume data transfer	SYS
OPEN	#OPEN	Common open	SYS
OXRF	#OXRF	Transient cross-reference resolver	SYS
P#CK	#WP#CK	Security key conversion	WP
P#FS	#WP#FS	Security file	WP
P#ML	#WP#ML	Start chain/go	WP
P#NL	#WP#NL	LDAM error logging	WP
P#QL	#WP#QL	LDAM erase chain	WP
P#RL	#WP#RL	LDAM terminate, close files, clear levels	WP
P#SL	#WP#SL	LDAM rekey/rename	WP
P#TL	#WP#TL	LDAM free chain	WP
P#UL	#WP#UL	LDAM get other	WP
P#VL	#WP#VL	LDAM get initial	WP
P#WL	#WP#WL	LDAM put final	WP
P#XL	#WP#XL	LDAM put record	WP
P#YL	#WP#YL	LDAM start chain/check	WP
P#ZL	#WP#ZL	LDAM initiate, open, set level	WP
P#1M	#WP#1M	MCU normal operation	WP
P#2M	#WP#2M	MCU error initialization	WP

Main Storage ID	Directory Name	Descriptive Name	Manual
P#3M	#WP#3M	MCU error processing	WP
P#4M	#WP#4M	MCU message processing	WP
P#5M	#WP#5M	MCU tilt/rotate select	WP
P#6M	#WP#6M	MCU Select non-WP table EBCDIC	WP
P#7M	#WP#7M	MCU Select WP table EBCDIC	WP
P#8M	#WP#8M	MCU Build tilt/rotate table	WP
P#9M	#WP#9M	MCU tilt/rotate automatic switching	WP
PACK	$PACK	Disk reorganization utility	SYS
PJAT	$WPJAT	Job-to-job processing	WP
PJCT	$WPJCT	Job-to-job create	WP
PJIT	$WPJIT	Job-to-job initialization	WP
PJST	$WPJST	Job-to-job stop	WP
PKBD	$WPKBD	Keyboard redefine	WP
PKEY	$WPKEY	Security key file update	WP
PLCU	$WPMCU	Magnetic card eject/erase	WP
PLDM	#WPLDM	LDAM main module	WP
PLUF	$WPLUF	LDAM free utility	WP
PLUI	$WPLUI	LDAM initiate utility	WP
PLUL	$WPLUL	LDAM load utility	WP
PLUM	$WPLUM	LDAM maintenance utility	WP
PLUP	$WPLUP	LDAM print utility	WP
PLUU	$WPLUU	LDAM unlead utility	WP
PPBI	$WPPBI	Printer belt identification utility	WP
PPBM	$WPPBM	Printer belt map utility	WP
POD3	#POD30	Continuous forms processing	SYS
POD7	#POD70	Single form/ledger cards processing	SYS
POE3	#POE30	Line printer error recovery	SYS

Main Storage ID	Directory Name	Descriptive Name	Manual
POE5	#POE50	Serial printer error recovery	SYS
PSET	$WPSET	Setup utility	WP
PTCH	$FEPCH	Patch	SYS
PURG	$WPURG	WP document directory purge	WP
PXDV	#MPXDV	Macro processor driver	SYS
PXP1	#MPXP1	Macro processor phase I	SYS
PXP2	#MPXP2	Macro processor phase II	SYS
PXP3	#MPXP3	Macro processor phase III	SYS
PXP4	#MPXP4	Macro processor phase IV	SYS
QJOB	$QJOB	Queued job stream card to library utility	SYS
RDCM	#RDCM	Compile OCL statement processor	SYS
RDDT	#RDDT	Date OCL statement processor	SYS
RDFL	#RDFL	File OCL statement processor	SYS
RDFM	#RDFM	Forms OCL statement processor	SYS
RDIC	#RDIC	Include OCL statement processor	SYS
RDIM	#RDIM	Image OCL statement processor	SYS
RDLD	#RDLD	Load OCL statement processor	SYS
RDLG	#RDLG	Log OCL statement processor	SYS
RDMK	#RDMK	File statement keyword diagnostics	SYS
RDML	#RDML	Reader/interpreter mainline	SYS
RDMM	#RDMM	Member OCL statement processor	SYS
RDMS	#RDMS	* (message) OCL statement processor	SYS
RDPS	#RDPS	Pause OCL statement processor	SYS
RDRN	#RDRN	Run OCL statement processor	SYS
RDRT	#RDRT	Reader/interpreter root phase	SYS
RDSL	#RDSL	Syslist OCL statement processor	SYS
RDSW	#RDSW	Switch OCL statement processor	SYS
RDS3	#RDS3	Positional parameter encode and scan	SYS

Main Storage ID	Directory Name	Descriptive Name	Manual
RENA	$RENAM	Disk file rename utility	SYS
RIN=	#STRI	Rollin	SYS
ROFX	#ROFX	Rolled-out SWA read	SYS
ROUT	#STRO	Rollout	SYS
RPEA	#RPEA	File description compression phase	RPG
RPEB	#RPEB	Extension compression	RPG
WCBS	$WCBS	WPCU BSC data management	WP
WCIN	$WCIN	WPCU initialization	WP
WCLI	$WCLI	WPCU LDAM file processing	WP
WCSI	$WCSI	WPCU S/32 file processing	WP
WCUM	$WCUM	WPCU mainline	WP
WP#J	#WP#JI	Job-job inquiry handler for termination	WP

This page intentionally left blank.

Main Storage ID	Directory Name	Descriptive Name	Manual
RPEC	#RPEC	Line counter compression	RPG
RPEE	#RPEE	Telecommunication compression	RPG
RPEI	#RPEI	Input compression	RPG
RPEK	#RPEK	Calculation compression	RPG
RPEO	#RPEO	Output compression	RPG
RPEW	#RPEW	Post compression and initialization	RPG
RPFA	#RPFA	Copy tables into compression	RPG
RPG	#RPG	Compiler initialization	RPG
RPGF	#RPGF	Assign and check indicators	RPG
RPGG	#RPGG	Check calculation specifications	RPG
RPGH	#RPGH	Assign filename and IOCB	RPG
RPGI	#RPGI	Assign filename and DTF	RPG
RPGK	#RPGK	Check telecommunications specifications	RPG
RPGL	#RPGL	Check telecommunications specifications (phase 2)	RPG
RPGN	#RPGN	Build DTFs	RPG
RPGS	#RPGS	Assign shared I/O and other disk device	RPG
RPGT	#RPGT	Assign I/O areas, nondisk devices	RPG
RPGU	#RPGU	Build name table	RPG
RPGV	#RPGV	Check table/array	RPG
RPGW	#RPGW	Check name table	RPG
RPGX	#RPGX	Print name table	RPG
RPGY	#RPGY	Build compile time tables	RPG
RPGZ	#RPGZ	Compile time table/array code	RPG
RPHA	#RPHA	Build symbol table	RPG
RPHC	#RPHC	Check symbol table	RPG
RPHD	#RPHD	Print symbol table	RPG

Main Storage ID	Directory Name	Descriptive Name	Manual
RPHQ	#RPHQ	Assign collating and translation tables	RPG
RPHS	#RPHS	Check input specifications	RPG
RPHT	#RPHT	Assign control, match field hold areas	RPG
RPHU	#RPHU	Assign limits file hold area	RPG
RPHW	#RPHW	Prompt table builder for IDE	RPG
RPIC	#RPIC	Post compression initialization	RPG
RPIG	#RPIG	Check file description continuation	RPG
RPJA	#RPJA	Check input specifications	RPG
RPJE	#RPJE	Check calculation specifications	RPG
RPJG	#RPJG	Check calculation specifications	RPG
RPJJ	#RPJJ	Control field and match field move	RPG
RPJK	#RPJK	Check file and table/array	RPG
RPJL	#RPJL	Check calculation specifications	RPG
RPJM	#RPJM	Check output specifications	RPG
RPJN	#RPJN	Check output specifications	RPG
RPJO	#RPJO	Check output specifications	RPG
RPJP	#RPJP	Check input specifications	RPG
RPJU	#RPJU	Check compile time tables/arrays	RPG
RPJW	#RPJW	Check file description specifications	RPG
RPJX	#RPJX	Check file description specifications	RPG
RPJY	#RPJY	Check extension and line counter specifications	RPG
RPJZ	#RPJZ	Check file description specifications	RPG
RPKA	#RPKA	Error sort, print, error message list	RPG
RPLB	#RPLB	Output indicator optimization	RPG
RPLG	#RPLG	DTF parameter assign	RPG
RPLJ	#RPLJ	Preassemble calculations	RPG

Main Storage ID	Directory Name	Descriptive Name	Manual
RPLN	#RPLN	Constant, literal, edit word assign	RPG
RPLR	#RPLR	Constant, literal, edit word, DTF assign	RPG
RPLV	#RPLV	Output fields move optimization	RPG
RPLZ	#RPLZ	Output indicator testing	RPG
RPMA	#RPMA	Output object code block length	RPG
RPMB	#RPMB	Preassemble calculations	RPG
RPMH	#RPMH	Preassemble calculations	RPG
RPMI	#RPMI	Preassemble indicator optimization	RPG
RPMK	#RPMK	Build segment list	RPG
RPMM	#RPMM	Output segment list entries build	RPG
RPPA	#RPPA	Generate IPCR	RPG
RPPB	#RPPB	Generate OPCR	RPG
RPPC	#RPPC	Input mainline code	RPG
RPPE	#RPPE	Build input record recognition	RPG
RPPF	#RPPF	Build input record recognition	RPG
RPPG	#RPPG	File selection and match field extraction	RPG
RPPJ	#RPPJ	Control field extraction code	RPG
RPPL	#RPPL	Look-ahead field extraction	RPG
RPPM	#RPPM	Input field extraction code	RPG
RPPN	#RPPN	Chain and read files code	RPG
RPPO	#RPPO	Last record and overflow control mainline	RPG
RPPS	#RPPS	Output record code	RPG
RPPU	#RPPU	Output fields 1 code	RPG
RPPV	#RPPV	Output fields 2 code	RPG
RPPW	#RPPW	Output fields 3 code	RPG
RPQA	#RPQA	Operation independent calculations	RPG

Main Storage ID	Directory Name	Descriptive Name	Manual
RPQB	#RPQB	Calculations 2 code	RPG
RPQD	#RPQD	Calculations 3 code	RPG
RPQE	#RPQE	Calculations 4 code	RPG
RPQF	#RPQF	Calculations 5 code	RPG
RPQG	#RPQG	Calculations 6 code	RPG
RPQH	#RPQH	Calculations 1 code	RPG
RPQK	#RPQK	Calculations 7 code	RPG
RPQL	#RPQL	Calculations 8 code	RPG
RPQT	#RPQT	Calculations 9 code	RPG
RPQV	#RPQV	Calculations 11 code	RPG
RPRA	#RPRA	Initialization and close code	RPG
RPRC	#RPRC	Table load/table dump code	RPG
RPRW	#RPRW	Chain and read files move list	RPG
RPRX	#RPRX	Initialize segment list	RPG
RPRY	#RPRY	Library of subroutines	RPG
RPSA	#RPSA	Resolve EXTRN	RPG
RPSB	#RPSB	Overlay editor	RPG
RPSC	#RPSC	Calculate start address	RPG
RPSD	#RPSD	Print segment list	RPG
RPSE	#RPSE	Sort object code	RPG
RPSF	#RPSF	Overlay editor	RPG
RPSG	#RPSG	Overlay fetch and transfer vector	RPG
RPSI	#RPSI	Final output generation	RPG
RPSK	#RPSK	Final output generation	RPG
RPSN	#RPSN	Sort segment list	RPG
RPSP	#RPSP	Eliminate duplicate segment list entries	RPG
RPZY	#RPZY	Dump facility	RPG

Main Storage ID	Directory Name	Descriptive Name	Manual
SBX0	#ASBX0	Build XREF file	ASM
SCM0	#ASCM0	Compression	ASM
SDJ0	#SDJ0	SNA Open routine	DC
SDJ1	#SDJ1	SDLC line initialization	DC
SDJ2	#SDJ2	SDLC temporary error logging	DC
SDJ3	#SDJ3	SNA close routine	DC
SDJ4	#SDJ4	SNA session initialization and deinitialization routine	DC
SDJ5	#SDJ5	SDLC statistical error logging	DC
SDJ6	#SDJ6	SNA session initialization and deinitialization subroutine	DC
SDLC	#SDLC	SDLC	DC
SNA	#SNA	SNA mainline	DC
SPD0	#ASPD0	Print diagnostics	ASM
SPEJ	#SPEJ	End-of-job transient	SYS
SPE0	#ASPE0	ESL output	ASM
SPFN	#SPFN	System find	SYS
SPS0	#ASPS0	Source/object output	ASM
SSB0	#ASSB0	Symbol processing	ASM
SSEM	#ASSEM	Assembler initialization	ASM
SSF0	#ASSF0	Symbol table processing	ASM
SSGR	#SSGR	SWA record get	SYS
SSRP	#SSRP	SWA record put	SYS
SSRW	#SSRW	SWA read/write	SYS
SSS0	#ASSS0	Symbol substitution	ASM
SSVF	#SSVF	VTOC read/write disk	SYS
SSVI	#SSVI	VTOC read/write diskette transient	SYS
SSVL	#SSVL	VTOC read/write diskette load module	SYS

Main Storage ID	Directory Name	Descriptive Name	Manual
SSX0	#ASSX0	Merge and list XREF	ASM
STDA	#STDA	Device allocate transient	SYS
STDD	#STDD	Device deallocation transient	SYS
STFA	#STFA	Allocate initiator transient	SYS
STFI	#STFI	File allocate initiator transient	RPG
STFT	#STFT	Allocate terminator transient	SYS
STHB	#STHB	Reader/interpreter error message build	SYS
STH0	#STH0	Syslog parameter table	SYS
STH1	#STH1	Syslog parameter table	SYS
STH2	#STH2	Syslog parameter table	SYS
STH3	#STH3	Syslog parameter table	SYS
STH4	#STH4	Syslog parameter table	SYS
STSS	#STSS	Source sysin	SYS
STS2	#STS2	Source sysin existence test	SYS
SUBR08	SUBR08	Magnetic character reader dummy data management	MCR
SYLC	#SYLCT	Syslist transient for display screen	SYS
SYLS	#SYLST	Syslist transient for printer	SYS
SYSG	#SYSG	Source get	SYS
SYIN	#SYSIN	Keyboard sysin	SYS
SYLG	#SYSLG	Syslog mainline	SYS
SYRP	$$SYRP	RPG II halt processor	RPG
SYTI	#SYSTI	Sysin transient	SYS
SYTG	#SYTGL	Syslog transient	SYS
TATS	$STATS	Status display utility	SYS
TMAL	#TMAL	End-of-job find work file	SYS
TMDS	#TMDS	End-of-job key search	SYS

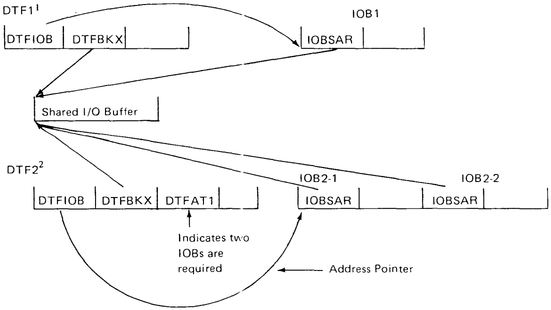
Main Storage ID	Directory Name	Descriptive Name	Manual
TME1	#TME1	Main storage end of job	SYS
TME2	#TME2	Main storage end of job	SYS
TME3	#TME3	Main storage end of job	SYS
TMI1	#TMI1	Scheduler IPL	SYS
TMMI	#TMMI	End-of-job index merge	SYS
TMSE	#TMSE	Special end of job	SYS
TMSK	#TMSK	End-of-job key sort	SYS
UC01	#$UC01	Disk consecutive input and output/add	SYS
UC02	#$UC02	Disk direct input	SYS
UC03	#$UC03	Disk indexed output	SYS
UC04	#$UC04	Disk limits input	SYS
UC05	#$UC05	Disk indexed update with add	SYS
UC06	#$UC06	Disk direct output/update/add	SYS
UC07	#$UC07	Disk indexed sequential input/update	SYS
UF1Z	#$UF1Z	Sector mode interface	SYS
UI1Z	#$UI1Z	Sector mode interface	SYS
US00	#US00	Syntax checker	SYS
WCCO	$BWCCO	CONFIG statement processor	DC
WCCR	$BWCCR	CARRIAGE statement processor	DC
WCD2	$BWCD2	Statement interpreter #2	DC
WCER	$BWCER	Error handler transient	DC
WCMD	$BWCMD	Statement interpreter #1	DC
WCM1	$BWCM1	MODIFY statement processor #1	DC
WCM2	$BWCM2	MODIFY statement processor #2	DC
WCR1	$BWCR1	READFILE statement processor #1	DC
WCR2	$BWCR2	READFILE statement processor #2	DC
WCR3	$BWCR3	READFILE statement processor #3	DC

Main Storage ID	Directory Name	Descriptive Name	Manual
WCSC	$BWCSC	Console specification module	DC
WCSI	$BWCSI	Initialization specification module	DC
WCSL	$BWCSL	Specification module loader	DC
WCSR	$BWCSR	Header specification module	DC
WDB1	$BWSUD	Workstation supervisor for CICS and IMS	DC
WDB2	$BWDB2	Specification module	DC
WDB4	$BWDB4	SNA interface	DC
WDCO	$BWDCO	Console output driver	DC
WDDI	$BWDDI	Disk interface	DC
WDFT	$BWDFT	Function terminator	DC
WDIO	$BWDIO	Input/Output SNA interface	DC
WDKI	$BWDKI	Keyboard interface	DC
WDLI	$BWDLI	Library interface	DC
WDMH	$BWDMH	Message handler	DC
WDPI	$BWDPI	Printer interface	DC
WDPO	$BWDPO	Print/Punch driver	DC
WDRI	$BWDRI	Reader input driver	DC
WFAL	$BWFAL	File allocation	DC
WFCL	$BWFCL	File close	DC
WFDA	$BWFDA	File deallocation	DC
WFOI	$BWFOI	File open (input files)	DC
WFOO	$BWFOO	File open (output files)	DC
WFSE	$BWFSE	Special end of job	DC
WINT	$BWINT	Initialization module transient	DC
WP#D	#WP#DI	Job to job inquiry handler first load	WP

Main Storage ID	Directory Name	Descriptive Name	Manual
WP#H	#WP#HI	Job to job inquiry handler second load	WP
WPC1	#WPCP1	Checkpoint first load	WP
WPC2	#WPCP2	Checkpoint second load	WP
WPLJ	#WPLJA	Syntax checker for job control utility specification	WP
WPLT	#WPLTA	Syntax checker for utility specification	WP
WPLM	#WPLMP	LDAM maintenance utility, part 2	WP
WPLS	#WPLSP	Syntax checker for LDAM interface	WP
WSEH	$BWSEH	SNA error handler	DC
WSIH	$BWSIH	SNA interrupt handler	DC
WSUR	$BWSUR	BWS supervisor/scheduler	DC
WIND	$FEKEY	Display function	SYS
XLOG	#SYXTG	Mini-syslog for type 1 messages	SYS
99RB	#99RIB	Special request RIB	SYS

Appendix M. Data Communications Miscellaneous Information

BSCA



[1] DTF1 is generated for a single buffer, transmit-only file.
[2] DTF2 is generated for a double buffer, receive-only file.

Figure M-1. BSC IOBs and Shared I/O Buffer

Notes:

1. Buffer length = (record length x blocking factor) + 21 + number of characters needed for ITB
2. Blocking factor = Number of data records per block
3. Number of characters needed for ITB = (Blocking factor - 1) x ITB count
   ITB count:
   1 for ITB nontransparent
   3 for ITB transparent receive

Receive Buffer

Acknowledgement to Preceding Block Received	S T X	--------Data--------	E E T or T X B	1 Byte Reserved

Transmitted Before Receiving Data

Transmit Buffer

S T X	--------Data--------	E E T or T X B	Acknowledgement Received to Block Transmitted	1 Byte Reserved

Received Subsequent to Transmitting Data

Note: Each buffer contains one block of data.

Figure M-2. Sample BSC Buffers (Non-ITB, Nontransparent)

Non-ITB, nontransparent buffer

STX	. . . . Data	ETX or ETB

Non-ITB, transparent buffer

DLE	STX	. . . . Data	ETX or ETB

ITB, nontransparent buffer

STX	Data Record 1	ITB	Data Record 2	ITB	Data Record n-1	ITB	Data Record n	ETX or ETB

ITB, transparent buffer (receive only)

DLE	STX	Data Record 1	ITB	DLE	STX	Data Record n-1	ITB	DLE	STX	Data Record n	ETX or ETB

Figure M-3. Examples of BSC Buffer Usage

Byte	Bit	Meaning When On	Reset Off By
2	0-5	Not assigned.	–
2	6	Data set ready. This indicates that the modem is ready to operate and that the BSCA has been enabled.	Modem losing its ready state or BSCA being disabled
2	7	Not assigned.	–
1	0	Timeout status. A receive timeout (3.25 seconds) occurred during a receive operation.	Any noncontrol BSCA SIO
1	1	Data check during receive operation. a. A CRC occurred (EBCDIC). b. A VRC occurred (ASCII). *Note:* Characters having VRCs are distinguished by a high-order bit in main storage. These characters are never recognized as control characters by the BSCA.	Any noncontrol BSCA SIO
1	2	Not assigned.	–

Figure M-4 (Part 1 of 2). BSCA Status Indications

Byte	Bit	Meaning When On	Reset Off By
1	3	Overrun. BSCA did not move a character to or from main storage before the next character had to be moved to accommodate the line. An overrun does not terminate the operation.	Any noncontrol BSCA SIO
1	4	Invalid ASCII. BSCA found leftmost bit in ASCII byte on during transmit operation.	Any noncontrol BSCA SIO
1	5	Abortive disconnect. Indicates BSCA on switched network was enabled, then the modem became ready, then not ready. This indicates the connection has been released and causes data terminal ready to turn off. The program must allow enough time for a forced disconnect (BSCA-controlled) to occur. The program can use the 2-second timeout to ensure this.	Disable BSCA
1	6	Adapter busy.	Op-end interrupt
1	7	Not assigned.	—

Figure M-4 (Part 2 of 2). BSCA Status Indications

Figure M-5 (Part 1 of 2). BSCA Line Control Codes and Their Functions

Name	Mnemonic	EBCDIC	ASCII
Start of Heading	SOH	SOH X'01'	SOH X'01'
Start of Text	STX	STX X'02'	STX X'02'
End of Transmission Block[1]	ETB	ETB X'26'	ETB X'17'
End of Text[1]	ETX	ETX X'03'	ETX X'03'
End of Transmission[1]	EOT	EOT X'37'	EOT X'04'
Enquiry[1]	ENQ	ENQ X'2D'	ENQ X'05'
Negative Acknowledge[1]	NAK	NAK X'3D'	NAK X'15'
Synchronous Idle	SYN	SYN X'32'	SYN X'16'
Data Link Escape	DLE	DLE X'10'	DLE X'10'
Intermediate Block Character	ITB	IUS X'1F'	US X'1F'
Even Acknowledge[1]	ACK0	DLE 70 X'1070'	DLE 0 X'1030'
Odd Acknowledge[1]	ACK1	DLE / X'1061'	DLE 1 X'1031'
Wait Before Transmit—Positive Acknowledge[1]	WACK	DLE , X'106B'	DLE ; X'103B'
Mandatory Disconnect	DISC	DLE EOT X'1037'	DLE EOT X'1004'

[1] Change of direction.

Figure M-5 (Part 2 of 2). BSCA Line Control Codes and Their Functions

Name	Mnemonic	EBCDIC	ASCII
Reverse Interrupt[1]	RVI	DLE@ X'107C'	DLE< X'103C'
Temporary Text Delay[1]	TTD	STX ENQ X'022D'	STX ENQ X'0205'
Transparent Start of Text	XSTX	DLE STX X'1002'	
Transparent Intermediate Block	XITB	DLE IUS X'101F'	
Transparent End of Text[1]	XETX	DLE ETX X'1003'	
Transparent End of Transmission Block[1]	XETB	DLE ETB X'1026'	
Transparent Synchronous Idle	XSYN	DLE SYN X'1032'	
Transparent Block Cancel[1]	XENQ	DLE ENQ X'102D'	
Transparent TTD[1]	XTTD	DLE STX DLE ENQ X'10021020'	
Date DLE in Transparent Mode	XDLE	DLE DLE X'1010'	
Leading Pad	—	X'55'	X'55'
Trailing Pad	—	X'7F' or X'FF'	X'FF'

[1] Change of direction.

SDLC

frame: Basic unit of commands, responses, and all information transmitted on SDLC. A valid frame must contain at least 32 bits, in multiples of 8 bits.

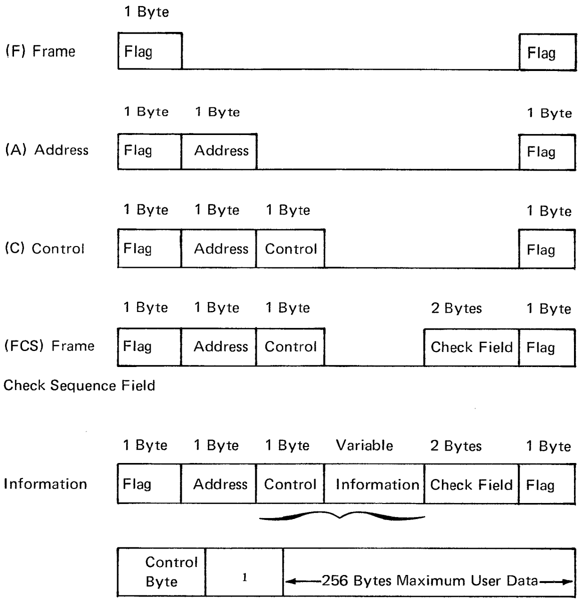
frame sequence: Group of frames, up to 7 (0 through 7). Uses an NS count at transmit and NR count at receive. The count is to identify missing or duplicate frames.

flag (F) 8 bits—01111110: This combination is never found in the transmission, except as the flag (beginning and end of frame).

SDLC zero bit insertion/deletion: If a transmit data stream has five 1 bits in sequence, a 0 bit is inserted into the data stream before transmission.

receive: If a character has five 1 bits in sequence when received and the next bit is a 0, it is deleted. If six 1 bits are received in a row, it indicates a flag or error condition.

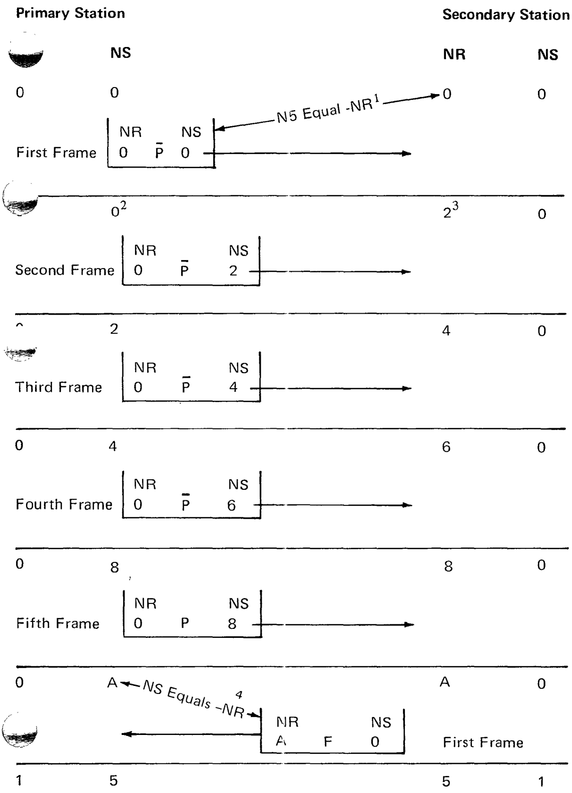
NRZI (nonreturn-to-zero-inverted): A method of transmitting data which reverses the state of the line for a 0 bit. NRZI prevents extended periods of no line reversals normally incurred when sending a long string of 0 bits. This aids in line synchronization. Zero bit insertion prevents a long string of 1 bits from being transmitted.



[1] SNA transmission header and request/response header information.

Figure M-6. SDLC Frame Format

Assume all counts at primary and secondary station start at zero.



[1] NS count in frame must equal NR count at secondary station.
[2] NS count incremented at primary station after frame is sent.
[3] NR count at secondary station incremented after accepting frame.
[4] NR count in frame must equal NS count at primary station.

● **Figure M-7. SDLC Sequenced Transmission Frames**

	0 1 2	3	4 5	6	7
Information Transfer Format	NR	P/F	NS		0
Supervisory Format	NR	P/F	[1]	0	1
Nonsequenced Format	[2]	P/F	[2]	1	1

Poll/final Bit

[1] Codes for supervisory commands/responses
[2] Codes for nonsequenced commands/responses

Figure M-8. SDLC C-Field Layout

Primary Command (Poll)	**Expected Secondary Response**
Nonsequence Format:	
-SNRM (set normal response mode)	NSA (nonsequence ACK)
DISC (disconnect)	NSA
TEST	TEST
XID (exchange ID)	XID
Supervisory Format:	
RR (receive ready)	I,RR,RNR,ROL
RNR (receive not ready)	RR,RNR,ROL
Information Format:	
I (information)	I,RR,RNR

Secondary Response	**Expected Primary Command**
Nonsequence Format:	
ROL (request on line)	SNRM
NSA	I,RR,RNR
CMDR (command reject)	SNRM, DISC
TEST	SNRM
XID	SNRM, DISC
Supervisory Format:	
RR	I,RR,RNR
RNR	RR,RNR
Information Format:	
I	I,RR,RNR

Figure M-9. SDLC Batch Work Station Command/Response

1st Byte	2nd Byte		3rd Byte	
0 thru 7	0 thru 3	4 thru 7	0 thru 3	4 thru 7
C Field	Send NS	Receive NR	0 0 0 0	z y x w

C field of the rejected command, received

System/32 send NS and receive NR sequence count

3rd byte reason for command reject:

X'01' Invalid control field
X'03' Control field with information
X'04' Data too long
X'08' NR out of range

Figure M-10. SDLC Command Reject Response

	Nonsequence Commands Byte								Hexadecimal Value		Name
	0	1	2	3	4	5	6	7	P	$\overline{\text{P}}$	
				P/$\overline{\text{P}}$ [1]			1	1			
SNRM	1	0	0	P/$\overline{\text{P}}$	0	0	1	1	93	83	Set normal response mode
DISC	0	1	0	P/$\overline{\text{P}}$	0	0	1	1	53	43	Disconnect
TEST	1	1	1	P/$\overline{\text{P}}$	0	0	1	1	F3	E3	Test
XID	1	0	1	P/$\overline{\text{P}}$	1	1	1	1	BF	AF	Exchange ID

	Nonsequence Responses Byte								Hexadecimal Value		Name
	0	1	2	3 [1]	4	5	6	7	F	$\overline{\text{F}}$	
NSA	0	1	1	F/$\overline{\text{F}}$	0	0	1	1	73	63	Nonsequence acknowledgement
CMDR	1	0	0	F/$\overline{\text{F}}$	0	1	1	1	97	87	Command reject
ROL	0	0	0	F/$\overline{\text{F}}$	1	1	1	1	1F	0F	Request on line
TEST	1	1	1	F/$\overline{\text{F}}$	0	0	1	1	F3	E3	Test
XID	1	0	1	F/$\overline{\text{F}}$	1	1	1	1	BF	AF	Exchange ID

	Supervisory Command/Response Byte								Hexadecimal Value		Name
	0	1	2	3	4	5 [2]	6	7	P/F	$\overline{\text{P}}$/$\overline{\text{F}}$	
RR	NR			P/F	0	0	0	1	+1	+1	Receive ready
RNR	NR			P/F	0	1	0	1	+5	+5	Receive not ready

	Information Frames Byte								Hexadecimal Value		Name
	0	1	2	3	4	5	6	7	P/F	$\overline{\text{P}}$/$\overline{\text{F}}$	
I	NR			P/F	NS			0	[3]	[3]	Information
Example	0	0	0		0	0	0	0	10	00	Example NR = 0, NS = 0
	0	0	1		0	0	0	0	30	20	Example NR = 1, NS = 0
	0	1	0		0	1	0	0	54	44	Example NR = 2, NS = 2
	1	0	0		1	1	0	0	9C	8C	Example NR = 4, NS = 6

[1] Code for Nonsequenced Commands/Responses

[2] Code for Supervisory Commands/Responses

[3] Information frames with hexadecimal value of NR and poll/final bits (left bits 0 thru 3), NS value for right bits 4 thru 7 (always even).

+	-Hexadecimal value of NR and poll/final bits.
NR	-Receive information frame count.
NS	-Send information frame count.
F	-Final bit = 1
P	-Poll bit = 1
$\overline{\text{F}}$	-Final bit = 0
$\overline{\text{P}}$	-Poll bit = 0

Figure M-11. SDLC Command and Responses In Hexadecimal Notation

SDLC Completion Table

The results of a transmit and/or receive operation can be determined by examining the status bytes of the SDLC completion table (Figure M-12):

How to Find: The SDLC completion table is contained in control storage in the SDLC transient microcode at addresses 0A10 (data end address) and 0A11 (status bytes). SDLC IOS senses this data into the SDLC IOB (see Figure 2-20).

Bytes 1 and 2 = Data end address

Byte 3

Bit	Meaning	Bit set on when:
0	Timeout	The inactivity timer completes.
1	BC check	A valid addressed frame is detected with an invalid block check.
2	Adapter check (transmit)	An error is detected in the SDLC adapter during a transmit operation. This bit is also set if another SIO is not issued before expiration of the flag fill timer.
3	Adapter check (receive)	An error is detected in the SDLC adapter during a receive operation.
4	Invalid frame	Any of the following occurs: – A flag is detected off a byte boundary – An ending flag is detected within 32 bits of the starting flag – An abort sequence is detected – An idle condition is detected between a starting and an ending flag
5	Abortive disconnect	The line 'data set ready' comes on and then goes off on a switched line.
6	Not used	
7	Stacked operation complete	A stacked operation is completed before interrupt processing of the previous operation begins.

Byte 4:

0-5	Not used	
6	Data set ready	The adapter or modem (data set) is ready.
7	Not used	

Figure M-12. SDLC Completion Table

DATA LINK CONTROL CHARACTERS AND CODES

EBCDIC

		Main Storage Bit Positions 0, 1, 2, 3							
Main Storage Bit Positions		0000	0001	0010	0011	0100	0101	0110	0111
4, 5, 6, 7	Hex	0	1	2	3	4	5	6	7
0000	0	NUL	DLE	DS		SP	&	-	
0001	1	SOH	DC1	SOS					
0010	2	STX	DC2	FS	SYN				
0011	3	ETX	DC3						
0100	4	PF	RES	BYP	PN				
0101	5	HT	NL	LF	RS				
0110	6	LC	BS	EOB/ETB	UC				
0111	7	DEL	IL	PRE/ESC	EOT				
1000	8		CAN						
1001	9	RLF	EM						\
1010	A	SMM	CC	SM		¢	!	¦	:
1011	B	VT				.	$	,	#
1100	C	FF	IFS		DC4	<	*	%	@
1101	D	CR	IGS	ENQ	NAK	(	)	_	'
1110	E	SO	IRS	ACK		+	;	>	=
1111	F	SI	IUS	BEL	SUB	|	¬	?	"

⧄ Duplicate Assignment

Figure M-13 (Part 1 of 2). EBCDIC Data Link Control Characters and Codes

		Main Storage Bit Positions 0, 1, 2, 3							
Main Storage Bit Positions 4, 5, 6, 7		1000	1001	1010	1011	1100	1101	1110	1111
	Hex	8	9	A	B	C	D	E	F
0000	0					{	}	\	0
0001	1	a	j	~		A	J		1
0010	2	b	k	s		B	K	S	2
0011	3	c	l	t		C	L	T	3
0100	4	d	m	u		D	M	U	4
0101	5	e	n	v		E	N	V	5
0110	6	f	o	w		F	O	W	6
0111	7	g	p	x		G	P	X	7
1000	8	h	q	y		H	Q	Y	8
1001	9	i	r	z		I	R	Z	9
1010	A								
1011	B								
1100	C								
1101	D								
1110	E								
1111	F								

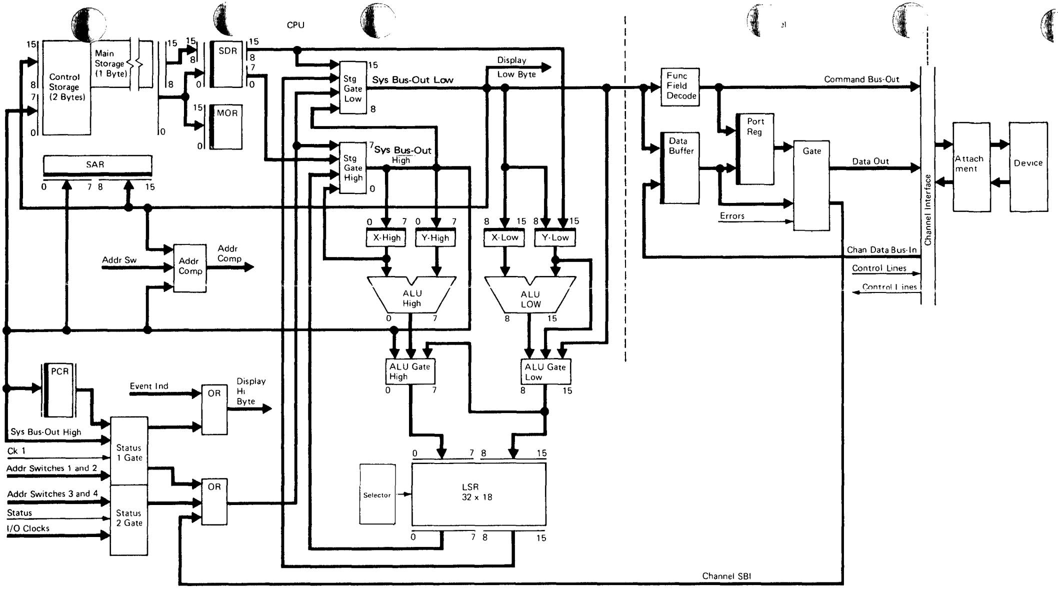
Figure M-13 (Part 2 of 2). EBCDIC Data Link Control Characters and Codes

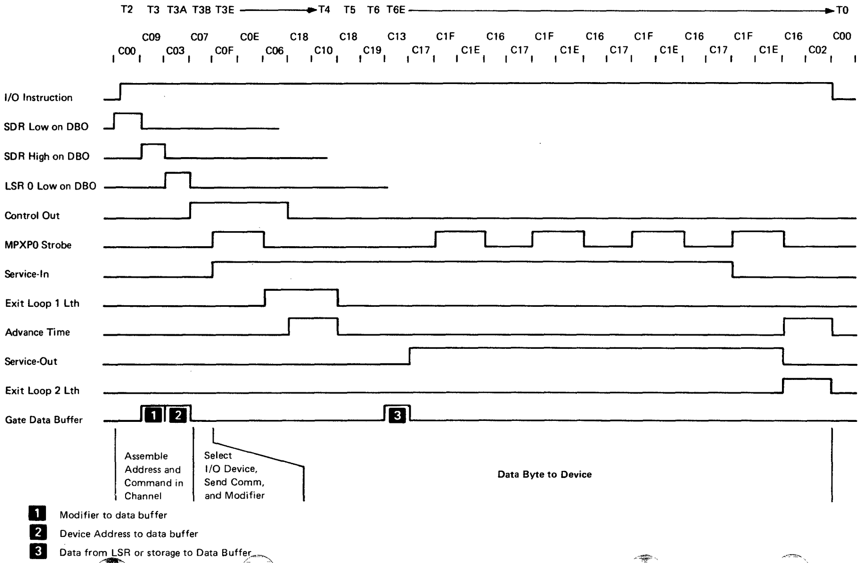
		Main Storage Bit Positions 0, 1, 2, 3[1]							
Main Storage Bit Positions 4, 5, 6, 7		0000	0001	0010	0011	0100	0101	0110	0111
	Hex	0	1	2	3	4	5	6	7
0000	0	NUL	DLE	SP	0	@	P	`	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z
1011	B	VT	ESC	+	;	K	[	k	{
1100	C	FF	FS	,	<	L	\	l	¦
1101	D	CR	GS	-	=	M	]	m	}
1110	E	SO	RS	.	>	N	¬	n	~
1111	F	SI	US	/	?	O	_	o	DEL

[1] Characters with position 0 on (equal to 1) have no meaning.

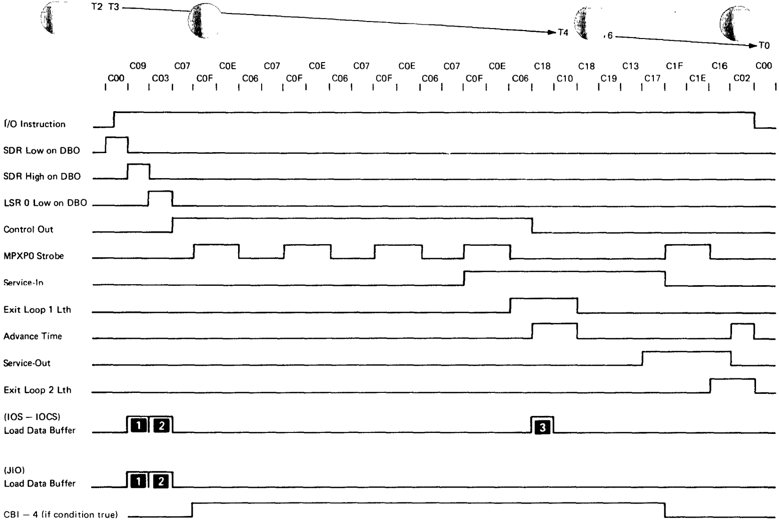
Figure M-14. ASCII Data Link Control Characters and Codes

SYSTEM DATA FLOW




T2 T3 T3A T3B T3E → T4 T5 T6 T6E → T0
C09 C07 C0E C18 C18 C13 C1F C16 C1F C16 C1F C16 C1F C16 C00
C00 C03 C0F C06 C10 C19 C17 C1E C17 C1E C17 C1E C17 C1E C02
I/O Instruction
SDR Low on DBO
SDR High on DBO
LSR 0 Low on DBO
Control Out
MPXP0 Strobe
Service-In
Exit Loop 1 Lth
Advance Time
Service-Out
Exit Loop 2 Lth
Gate Data Buffer
1
2
3
Assemble Address and Command in Channel
Select I/O Device, Send Comm, and Modifier
Data Byte to Device
1 Modifier to data buffer
2 Device Address to data buffer
3 Data from LSR or storage to Data Buffer

IOS—IOCS—JIO CHANNEL TIMING



1 Modifier to Data Buffer

2 Device Address to Data Buffer

3 Data From I/O Device to Data Buffer

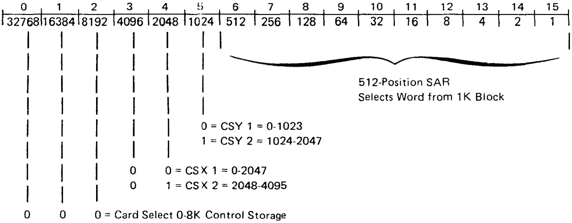
COMMAND BUS IN

Bits	0	1	2	3	4	5		
	0	0					Cycle Steal Sense, no Increment	
	0	1					Cycle Steal Load, no Increment	
	1	0					Cycle Steal Sense, Increment	
	1	1					Cycle Steal Load, Increment	
			0	0			Cycle Steal LSR Select 0 WR4	
			0	1			Cycle Steal LSR Select 1 WR5	
			1	0			Cycle Steal LSR Select 2	Not Used
			1	1			Cycle Steal LSR Select 3	
					1		Cycle Steal Select Control Storage (plus)	
						1	DBO Parity Check	

Note:
Sense—Disk to processor
Load—Processor to disk

SAR DECODING (PN 020)

Control Storage



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

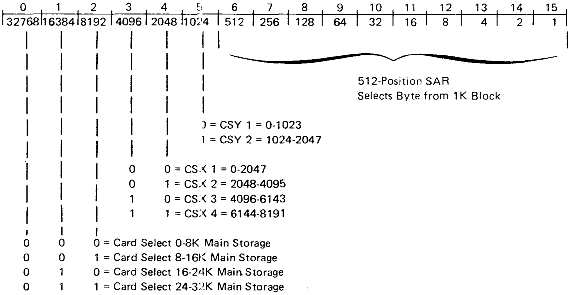
512-Position SAR
Selects Word from 1K Block

0 = CSY 1 = 0-1023
1 = CSY 2 = 1024-2047

0 0 = CSX 1 = 0-2047
0 1 = CSX 2 = 2048-4095

0 0 0 = Card Select 0-8K Control Storage

Main Storage



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

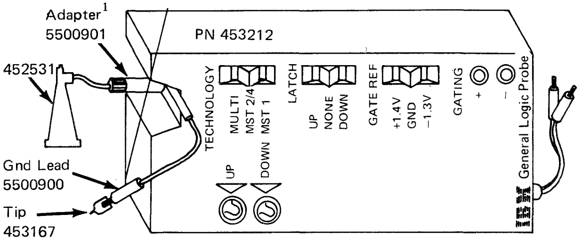
512-Position SAR
Selects Byte from 1K Block

0 = CSY 1 = 0-1023
1 = CSY 2 = 1024-2047

0 0 = CSX 1 = 0-2047
0 1 = CSX 2 = 2048-4095
1 0 = CSX 3 = 4096-6143
1 1 = CSX 4 = 6144-8191

0 0 0 = Card Select 0-8K Main Storage
0 0 1 = Card Select 8-16K Main Storage
0 1 0 = Card Select 16-24K Main Storage
0 1 1 = Card Select 24-32K Main Storage

GENERAL LOGIC PROBE



[1] A 24-inch extender is available, PN 453605.

Specifications:

1. Size: 6 inches long, 2½ inches wide, and 1½ inches deep
2. Technology: SLT, SLD, TTL (VTL), FET, MST-1, 2, and 4
3. Built-in latch
4. Up and down indicators
5. Two gating pins
6. Will detect a 5 nsec pulse for MST and a 6 nsec pulse for VTL, SLD, SLT, FET

Test Points:

A1-J2 S02 1 second
A1-J2 S13 16 ms pulse

Switches:

1. Three-position: Select the technology you are using.
   Multi — Used with SLT, SLD, VTL, and FET
   MST-2/4 — Self-explanatory
   MST-1 — Self-explanatory

2. Latch:
   Up — Up level set
   None — Latch not used
   Down — Down level set

3. Gating: Plus and minus gating pins are provided. The gate reference switch is used along with these two pins for gating the probe. When gating is to be used with the probe, the indicators are inhibited until the gate signal is present with the probe input signal.

4. Gate Ref: Select Correct gate level for the technology you are using:
   +1.4V — For VTL, SLT, SLD
   Gnd — For MST-2/4
   -1.3V — For MST-1

5. Up and down indicator lights:

Up	Down	Range
On	Off	+2.0V to +60.0V
Off	On	+0.8V to -60.0V
Off	Off	+0.8V to +2.0V

Pulsing Signals: Depending on the frequency of the signal, either the up or down indicators will be on alternately, or both indicators will be lit at the same time.

6. Ground lead (PN 5500900) must be used as the input signal is independent of the power supply.

7. Probe Power: Can be connected to any dc voltage source in the range of 4V to 12V. The black lead must be connected to the negative potential and the read lead to the positive potential. Black lead any D08, red lead any D03.

8. Probe Support Hook: Should be hooked on the gate when probing. (For further information, refer to Probe handbook.)

CPU LOGIC CARD SUMMARY

Port Card A1-H2

1. Channel register
2. Channel controls
3. Channel clocks
4. Channel checks
5. Channel interrupt
6. Channel status
7. Cycle steal controls
8. I/O immediate decodes (channel)

Status Card 2 A1-J2

1. I/O clocks
2. Address compare (low byte and compare logic)
3. Run latch
4. Stop latch
5. I/O immediate decodes (processing unit)
6. Display bits 12-15, P

7. Assembly bus:
   a. Console switches 3 and 4
   b. Console status
   c. I/O clock low byte
   d. I/O clock high byte

8. Machine check latch and processor check trigger

Status Card 1 A1-K2

1. Address compare (high byte and compare logic)

2. Branch on condition:
   a. SDR register (bits 4-7)
   b. Decode logic
   c. PCR and controls

3. Event indicators

4. Processor check register

5. Display bits 0-7, P (high byte)

6. I/O immediate decodes (processing unit)

7. Assembly bus
   a. Console switches 1 and 2
   b. Event indicators
   c. Processing unit checker
   d. Processor condition register
   e. System bus-out high byte

Data Flow Card A1-L2

1. SDR register

2. LSR register (Kayak 64 x 9)

3. Storage gates

4. X and Y-registers and reset for Y-register

5. ALU and parity predict and control bits

6. ALU gates

7. Checks: ALU, SDR, and storage gates

System Control Card A1-M2

1. MOR register

System clock

3. Data flow control per timing charts and data flow:
   a. LSR selection
   b. LSR write
   c. Clock X, Y, SAR
   d. Storage gate selection
   e. ALU gate selection
   f. ALU function bits, carry-in, and 16-bit ALU operation
   g. Y-gate selection

4. Storage selection and controls (MS/CS)

5. Decode of instructions

6. Decode of mode selector switch

7. Cycle control for console and instruction

8. Timing control for ALU gate and storage gate check

9. Cycle steal interrupt controls

10 MOR parity check

Storage Control Card A1-N2

1. Storage address register

2. Invalid address checking

3. IMPL sequence control

4. System reset generation

5. Storage addressing and timing

6. Display bits 8-11

7. Meter controls

PLUG CHART 01A-A1 CARD SIDE

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	S	T	U	V	
	Y1			Y2				Y3			Y4				Y5			Y6			
2	A2 Pwr Seq Cable	B2 Op Panel Cable		D2 Magnetic Character Feat Driver/Receiver/Regulator	E2 Magnetic Character Feat Controller	F2 Magnetic Character Feat Adapt		H2 Chnl Card	J2 Status 2 Card	K2 Status 1 Card	L2 Data Flow Card	M2 Sys Ctrl Card	N2 Stor Ctrl Card	P2 Ctrl Stor Lo Note 1 0-4K or 0-8K		R2 Main Store 0-8K	S2 Main Store 8-16K	T2 Main Store Notes 2 & 3 16-24K	U2 Main Store Note 3 24-32K		2
3	A3 CE Panel LEDs	Magnetic Character Reader Feat Cable																			3
4	A4 Magnetic Character Reader Feat Cable	B4 CE Panel Sw Cable	C2 Driver/Receiver/Regulator												Q4 Ctrl Stor Hi Note 1 0-4K or 0-8K						4
5	A5 Magnetic Character Reader Feat Cable	B5 CE Panel Sw Cable																			5
	BSCA Crossover Z1			Basic Crossover Z2				Basic Crossover Z3			Basic Crossover Z4				Z5			Z6			
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	S	T	U	V	

Notes:

1. 8K is a feature.
2. For 24K feature.
3. For 32K feature.

PLUG CHART 01A-A2 CARD SIDE

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	S	T	U	V
	BSCA/SDLC Crossover Y1			Basic Crossover Y2			Basic Crossover Y3				Basic Crossover Y4				Y5			Y6		
2	A2 Disk Cable	B2 Diskette Cable	C2 Detect & PIO Disk	D2 Data SEP Disk	E2 Ctrl & Safety Disk	F2 Attachment Disk	G2 Attachment Disk		J2 Data SEP Diskette	K2 Attachment Diskette	L2 BSCA Attachment or SDLC Attachment	M2 Kybd CRT Attachment	N2 Mag Card Unit Attachment MCA1 or Card I/O Attachment	P2 Mag Card Unit Attachment MCA2	Q2 Belt PRT Attachment Ctrl or Serial Ptr Attachment	R2 Belt PRT Attachment Hmr Drvr or Serial Ptr Attachment	S2 285 lpm Printer Hmr Drvr	T2 Note PTR	U2 CRT Cable	V2 PTR Cable
3	A3 Disk Cable																	T3 285 Printer Cable	U3 Kybd Cable	V3 PTR Cable
4	A4 Disk Cable	B4 Int Clock	C4 Drvr Disk	D4 Velcty Ctrl Disk					J4 Half Line Spacing									T4 285 Printer Cable	U4 MCU or Card I/O	V4 PTR Cable
5	A5 Disk Cable	B5																T5 EIA Cable	U5 MCU or Card I/O	V5 PTR Cable
	Z1			Z2			Z3				Z4				Z5			285 Printer Cable Z6		

Note: Print speed part number varies with feature installed.

PLUG CHART 01B-A1 CARD SIDE (2400 INT MODEM)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	S	T	U	V
			Y1			Y2				Y3			Y4			Y5				Y6
2				D2 Auto Equalizer SW	E2 Auto Equalizer SW		G2 FET Digital B			K2 Jumper B	L2 Com-prom Equal SBU	M2 Sw Net Backup SBU		P2 Caducee CAD	Q2 Preamp B	R2 Auto-answer Basic AA	S2	T2 Inter-face B		V2 Cable B
3								H3 Timing SW		K3 Timing SBU							S3 Auto-answer Inter-face AA		U3 Cable B	V3 Cable CAD/AA
4									J4 Transmit B	K4 Limiter B			N4 Post B		Q4 Test Operate B		S4 Transmit Equalizer MP	T4 Receive Equalizer LSD		V4 LSD
5																			U5 Cable B	V5 Cable B
			Z1			Z2				Z3						Z5				Z6

Legend:

B	= Basic	MP	= Multipoint net	AA	= Auto answer
LSD	= Leased network	SBU	= Sw net backup		
SW	= Switched network	CAD	= Caducee (French)		

PLUG CHART 01C-A1 CARD SIDE (MINI-12 MODEM)

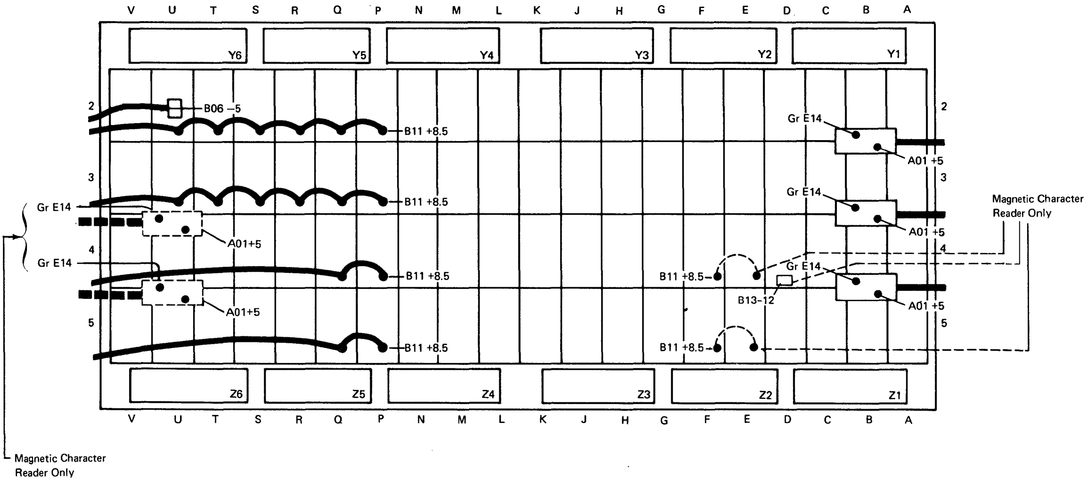
Switched Network

A1	B1	C1	D1	E1
Auto Answer	Wrap	Transmit	Receive	Cable
				Cable

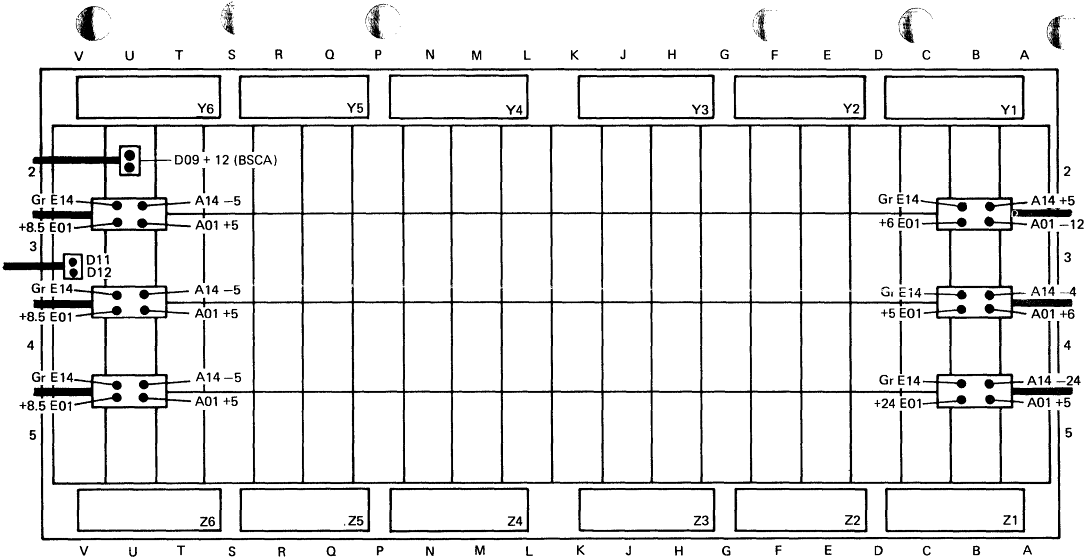
Leased Line

B1	C1	D1	E1
Wrap	Transmit	Receive	Cable
			Cable

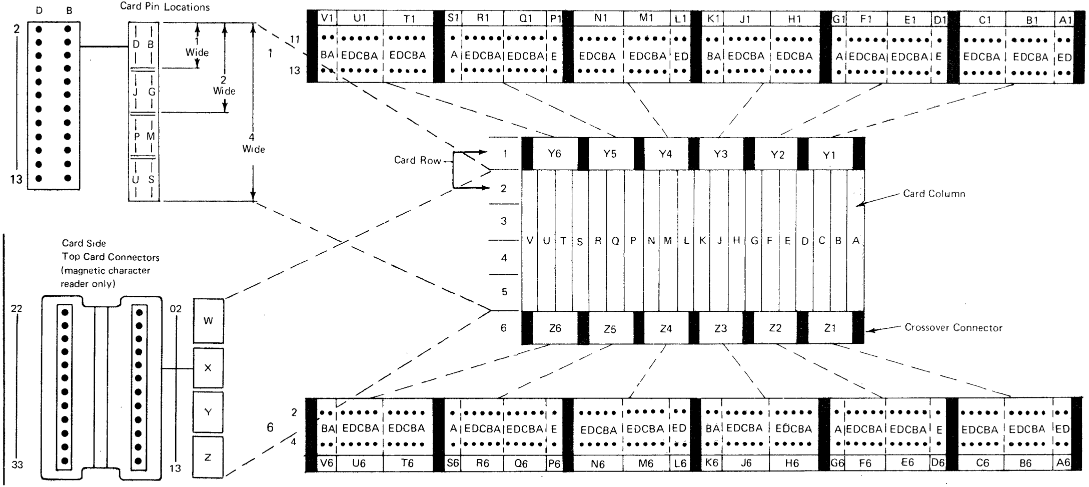
VOLTAGE DISTRIBUTION 01A-A1



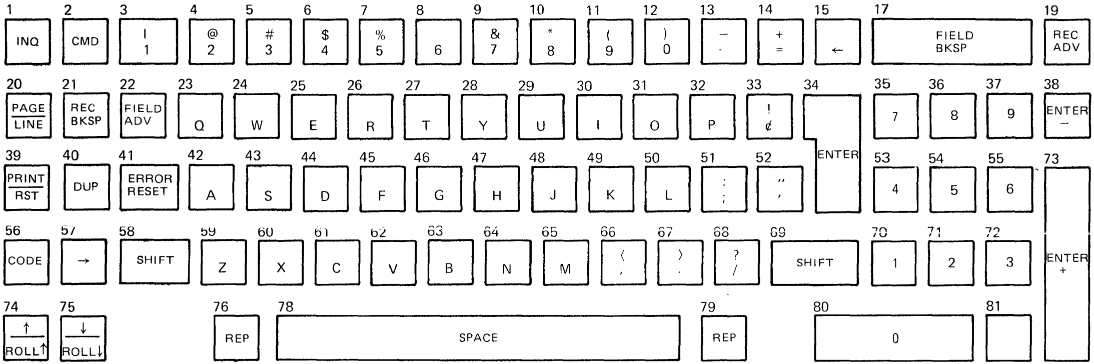
VOLTAGE DISTRIBUTION 01A-A2



CROSSOVER CABLE-BOARD PIN LOCATIONS—PIN SIDE VIEW


D
B
2
13
Card Pin Locations
D B
J G
P M
U S
1 Wide
2 Wide
4 Wide
1
11
13
V1 U1 T1 S1 R1 Q1 P1 N1 M1 L1 K1 J1 H1 G1 F1 E1 D1 C1 B1 A1
BA EDCBA EDCBA A EDCBA EDCBA E EDCBA EDCBA ED BA EDCBA EDCBA A EDCBA EDCBA E EDCBA EDCBA ED
Card Row
1
2
3
4
5
6
Y6 Y5 Y4 Y3 Y2 Y1
V U T S R Q P N M L K J H G F E D C B A
Card Column
Z6 Z5 Z4 Z3 Z2 Z1
Crossover Connector
Card Side
Top Card Connectors
(magnetic character
reader only)
22
33
02
13
W
X
Y
Z
6
2
4
BA EDCBA EDCBA A EDCBA EDCBA E EDCBA EDCBA ED BA EDCBA EDCBA A EDCBA EDCBA E EDCBA EDCBA ED
V6 U6 T6 S6 R6 Q6 P6 N6 M6 L6 K6 J6 H6 G6 F6 E6 D6 C6 B6 A6

KEYBOARD LAYOUT



Note: Numbers 16, 18, and 77 are not used.

KEYBOARD SCAN CODE TABLE

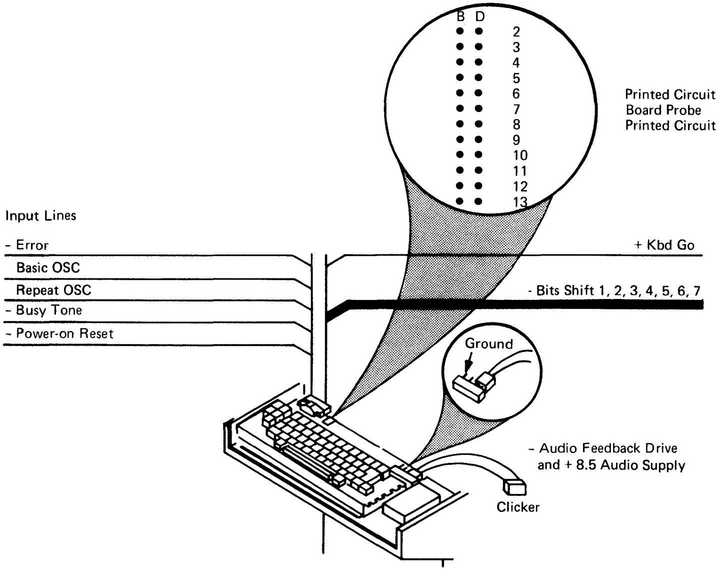
Key #	Domestic Symbol or Function Shift Bit 0	 Shift Bit 1	Shift	Keyboard Data Bits 1234567
1	Inq			0110010
2	Cmd			0111111
3	1	|		0110100
4	2	@		0111101
5	3	#		0111000
6	4	$		0111110
7	5	%		0111001
8	6	¬		0110101
9	7	&		0111010
10	8	*		0110001
11	9	(		0111100
12	0	)		0110111
13	-	—		0110110
14	=	+		0110000
15	←	←		1100000
16	(not used)			
17	Field bksp			1000010
18	(not used)			
19	Rec Adv			1000111
20	Line	Page		0100010
21	Rec Bksp			0100011
22	Field Adv			0100100
23	Q	Q		0101101
24	W	W		0101000
25	E	E		0101110
26	R	R		0101001
27	T	T		0100101
28	Y	Y		0101010
29	U	U		0100001
30	I	I		0101100
31	O	O		0100111
32	P	P		0100110
33	¢	!		0100000
34	Enter			0101111
35	7			1010000
36	8			1010010
37	9			1010100
38	Enter–			1010111
39	Rst	Print		0010010
40	Dup	Dup		0010011
41	Error Reset			0010100
42	A	A		0011101
43	S	S		0011000
44	D	D		0011110
45	F	F		0011001
46	G	G		0010101
47	H	H		0011010
48	J	J		0010001
49	K	K		0011100
50	L	L		0010111
51	;	:		0010110
52	'	"		0010000
53	4			1010001
54	5			1010011
55	6			1010101
56	CODE			0000010
57				0000011
58		Shift	I	xxxxxxx
59	Z	Z		0001101
60	X	X		0001000
61	C	C		0001110
62	V	V		0001001
63	B	B		0000101
64	N	N		0001010
65	M	M		0000001
66	,	‹		0001100
67	.	›		0000111
68	/	?		0000110
69		Shift	I	xxxxxxx
70	1			1000001
71	2			1000011
72	3			1000101
73	Enter +			1010110
74	Roll ↑	↑		0111011
75	Roll ↓	↓		0011011
76	Rep			
77	(not used)			
78	Space			0011111
79	Rep			
80	0			1100001
81	.			1000110

KEYBOARD ERROR CONDITIONS

Keyboard Overrun: Byte of data in data register and second keystroke occurs before the system handled the first keystroke.

Operator Keying Error: Keying out of limits of the defined field, including cursor movements or keying alpha in a numeric field.

KEYBOARD INPUT AND OUTPUT LINES



BELT PRINTER ERROR CHECKING

Forms Jam Check

The forms jam check indicates that the carriage tractor was told to move, but no paper motion occurred. A forms motion device (light emitting diode) is used to detect the time between holes in the paper. If no hole is detected in eight lines, the forms jam check is set.

Belt Speed Check

This check indicates that either the belt has failed to get into motion within two seconds after the start time or the printer belt motion is lost after having reached an up-to-speed condition. Motion is considered lost if there is a ten percent loss in operating velocity. The speed is determined by measuring the time between timing marks on the print belt.

Carriage Sync Clock

Two conditions may set this check:

1. If a carriage feedback pulse (carriage advance pulse) occurs when no carriage.

2. If a carriage feedback pulse fails to occur within eight milliseconds, during carriage space time.

Coil Current Check

An eight millisecond timer is started when the hammer fire latch is set. The three hammer check lines are monitored for the possibility of a hammer being outside this eight millisecond time. If this condition is detected, power is dropped to the printer, and coil current check is set. The status of the coil current lines will be saved until the check is cleared.

Note: If an even number of hammers on one hammer drive card are on (outside the 8 msec time) the coil current check will not be set, since the hammer check lines are only active for an odd condition.

Belt Sync Check

This check is set by three possible conditions:

1. If a home pulse, on the belt, occurs when not expected.

2. If a home pulse fails to occur when expected.

3. The bit ring generating the five subscan pulses is continuously monitored for an abnormal condition, that is, one and only one bit should be on at all times.

The timing for the home pulse is determined by counting the number of print scans. This count is compared with the character set size (only one home pulse per character set).

Emitter Check

Once the print belt motor has reached an up-to-speed condition, the print subscan line is monitored to verify that it is oscillating. If no change occurs during any two millisecond period, the emitter check will be set. This check supplements the belt sync check which cannot detect print subscans and home both missing.

Data Check

Parity is maintained on the data in the print buffer. If invalid parity is detected during a print cycle, this check is set.

Hammer Parity Check

This check compares the parity of hammers selected to fire with the parity actually fired. If a mismatch occurs, the hammer parity check latch is set.

End of Forms

End of forms is checked on the first line printed of each new form. If active, the printer will go not ready.

Throat Interlock

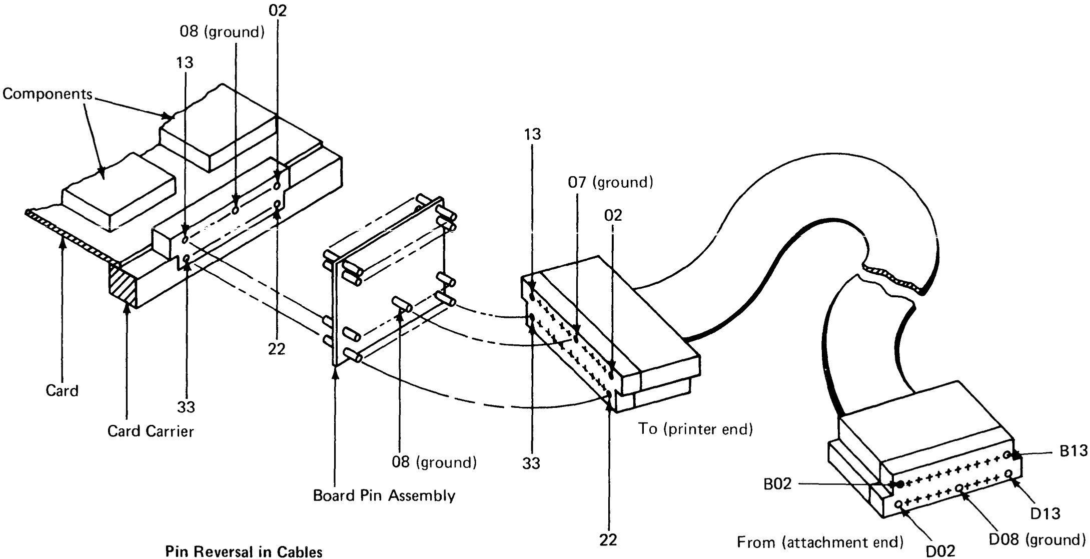
The printer is not ready if the throat is not closed on the paper path.

Cover Interlock

The printer is not ready if the cover is open.

Unprintable Character

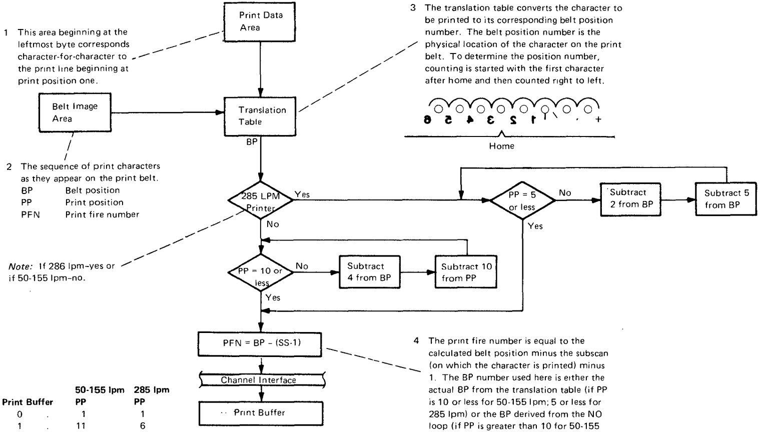
One or more of the characters requested to be printed were not in the print image. Unprintable character is checked by the microprogram. There is no hardware checking involved. Setting of this check is a programmer option.



Pin Reversal in Cables

1	2	3	4	5	6	7	8	9	0	#	@	/	S	T	U	V	W	X	Y	Z	&	.	%
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

J	K	L	M	N	O	P	Q	R	–	$	*	A	B	C	D	E	F	G	H	I	+		'
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47



1 This area beginning at the leftmost byte corresponds character-for-character to the print line beginning at print position one.

2 The sequence of print characters as they appear on the print belt.
BP Belt position
PP Print position
PFN Print fire number

Note: If 286 lpm-yes or if 50-155 lpm-no.

3 The translation table converts the character to be printed to its corresponding belt position number. The belt position number is the physical location of the character on the print belt. To determine the position number, counting is started with the first character after home and then counted right to left.

4 The print fire number is equal to the calculated belt position minus the subscan (on which the character is printed) minus 1. The BP number used here is either the actual BP from the translation table (if PP is 10 or less for 50-155 lpm; 5 or less for 285 lpm) or the BP derived from the NO loop (if PP is greater than 10 for 50-155 lpm; greater than 5 for 285 lpm).

		50-155 lpm	285 lpm
Print Buffer		**PP**	**PP**
0	.	1	1
1	.	11	6
2	.	21	11
3	.	31	16
	.	etc.	etc.

5 The print buffer contains the sorted print fire numbers (PFN). They are sorted into the sequence in which they will be addressed.

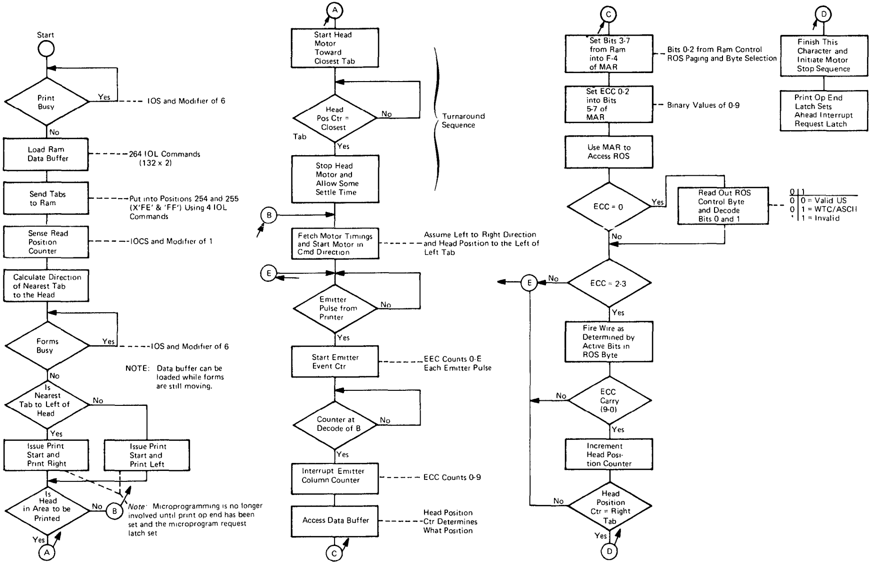
50-155 lpm

If the last digit of PP =		subscan =
	1 or 2	1
	3 or 4	2
	5 or 6	3
	7 or 8	4
	9 or 0	5

285 lpm

If the last digit of PP =		subscan =
	1	1
	3	2
	5	3
	2	4
	4	5

SERIAL PRINTER OPERATION FLOWCHART



Start
Print Busy
Yes
IOS and Modifier of 6
No
Load Ram Data Buffer
264 IOL Commands (132 x 2)
Send Tabs to Ram
Put into Positions 254 and 255 (X'FE' & 'FF') Using 4 IOL Commands
Sense Read Position Counter
IOCS and Modifier of 1
Calculate Direction of Nearest Tab to the Head
Forms Busy
Yes
IOS and Modifier of 6
No
Is Nearest Tab to Left of Head
No
Yes
Issue Print Start and Print Right
Issue Print Start and Print Left
Is Head in Area to be Printed
No
B
Yes
A
A
Start Head Motor Toward Closest Tab
Head Pos Ctr = Closest Tab
No
Yes
Turnaround Sequence
Stop Head Motor and Allow Some Settle Time
B
Fetch Motor Timings and Start Motor in Cmd Direction
Assume Left to Right Direction and Head Position to the Left of Left Tab
E
Emitter Pulse from Printer
No
Yes
Start Emitter Event Ctr
EEC Counts 0-E Each Emitter Pulse
Counter at Decode of B
No
Yes
Interrupt Emitter Column Counter
ECC Counts 0-9
Access Data Buffer
Head Position Ctr Determines What Position
C
C
Set Bits 3-7 from Ram into F-4 of MAR
Bits 0-2 from Ram Control ROS Paging and Byte Selection
Set ECC 0-2 into Bits 5-7 of MAR
Binary Values of 0-9
Use MAR to Access ROS
ECC = 0
Yes
Read Out ROS Control Byte and Decode Bits 0 and 1
0 1
0 0 = Valid US
0 1 = WTC/ASCII
1 1 = Invalid
No
ECC = 2-3
No
E
Yes
Fire Wire as Determined by Active Bits in ROS Byte
ECC Carry (9-0)
No
Yes
Increment Head Position Counter
Head Position Ctr = Right Tab
No
Yes
D
D
Finish This Character and Initiate Motor Stop Sequence
Print Op End Latch Sets Ahead Interrupt Request Latch

NOTE: Data buffer can be loaded while forms are still moving.

Note: Microprogramming is no longer involved until print op end has been set and the microprogram request latch set

SERIAL PRINTER ERROR CHECKING

Forms hung check: If forms emitter pulses do not occur within a specified length of time.

Horizontal Check:

1. Emitters out of sequence – When head is moving to the right, the sequence should be 1, 2, 3, 1. When the head is moving to the left, the sequence should be 3, 2, 1, 3.

2. Print head hung – Print head stepper motor being told to go and no emitter pulses.

3. Emitters too fast – The print head is moving too fast for proper synchronization.

4. Memory parity check – A byte with even parity was read out of ROS or RAM.

5. Unprintable character check – A character requested to print was not in the defined character set. The condition occurs when the position of ROS that is addressed is coded as invalid by its control byte.

6. Printer not ready – +10.8 volt undervoltage, +24 volt undervoltage or overvoltage, or a wire check has occurred.

Forms Runaway Check: Monitors the time from when 'forms go' becomes active until the forms line/print time counter gets to zero. If it exceeds the time needed to move 127 lines, forms runaway is set.

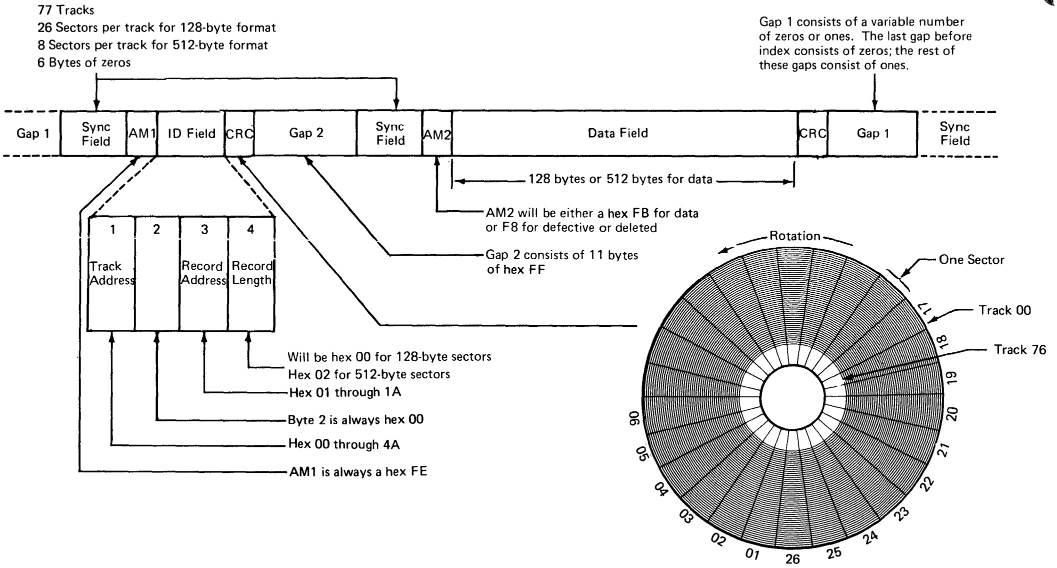
End of Forms: Set when end-of-forms switch senses the absence of forms.

Printer Not Ready: See horizontal check number 6.

Wire Check: A print wire magnet was energized for more than 1.6 msec. Forces printer not ready.

Memory Parity Check: See horizontal check number 4.

Unprintable Character Check: See horizontal check number 5.


77 Tracks
26 Sectors per track for 128-byte format
8 Sectors per track for 512-byte format
6 Bytes of zeros
Gap 1 consists of a variable number of zeros or ones. The last gap before index consists of zeros; the rest of these gaps consist of ones.
Gap 1
Sync Field
AM1
ID Field
CRC
Gap 2
Sync Field
AM2
Data Field
CRC
Gap 1
Sync Field
128 bytes or 512 bytes for data
AM2 will be either a hex FB for data or F8 for defective or deleted
Gap 2 consists of 11 bytes of hex FF
1
2
3
4
Track Address
Record Address
Record Length
Will be hex 00 for 128-byte sectors
Hex 02 for 512-byte sectors
Hex 01 through 1A
Byte 2 is always hex 00
Hex 00 through 4A
AM1 is always a hex FE
Rotation
One Sector
Track 00
Track 76
17
18
19
20
21
22
23
24
25
26
01
02
03
04
05
06

DISKETTE ERROR CONDITIONS

Missing Data: A data record was not found after an ID field.

ID CRC: Cyclic code noncompare in an ID field.

Data CRC: Cyclic code noncompare in a data field.

Cylinder Mismatch: Mismatch between cylinder address of ID field and the control field during an ID search.

Head Mismatch: Mismatch between head address of ID field and the control field during an ID search.

Record Mismatch: Mismatch between record address of ID field and the control field during an ID search.

Length Mismatch: Mismatch between record length of ID field and control field during an ID search.

Invalid Control Record: Leftmost byte of a control record contained other than an F or D control graphic.

Control Error: Low write current or write or erase selected during a command other than write.

Cylinder Address Invalid: End of diskette has been reached with a seek still pending.

Write Error: Write overrun or write parity during a write operation.

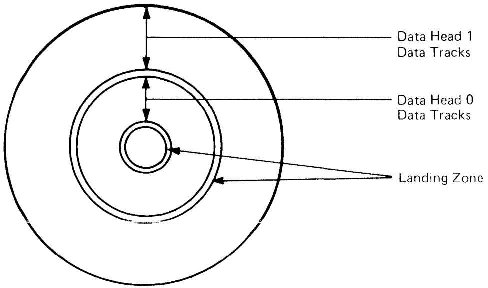
33FD Fast: Disk speed faster than maximum of 369 rpm.

No Orient: ID field could not be found in the selected track.

Read Overrun: Minimum data transfer rate not maintained during transfer from diskette to main storage.

Write Overrun: Minimum data transfer rate not maintained during transfer from main storage to diskette.

Write Parity Check: Mismatch between DBO parity and generated serial write data parity during a write operation.



Megabytes	Cylinders
3.2	109
5.0	169
9.1	303

Each cylinder has two data tracks and one servo track:

Track = 60 sectors

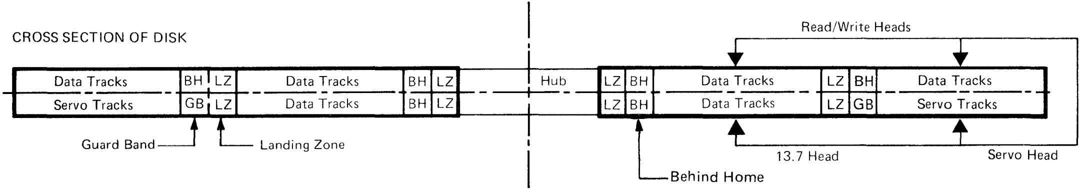
Sector = 256 bytes

Alternate sector = Cyl 2

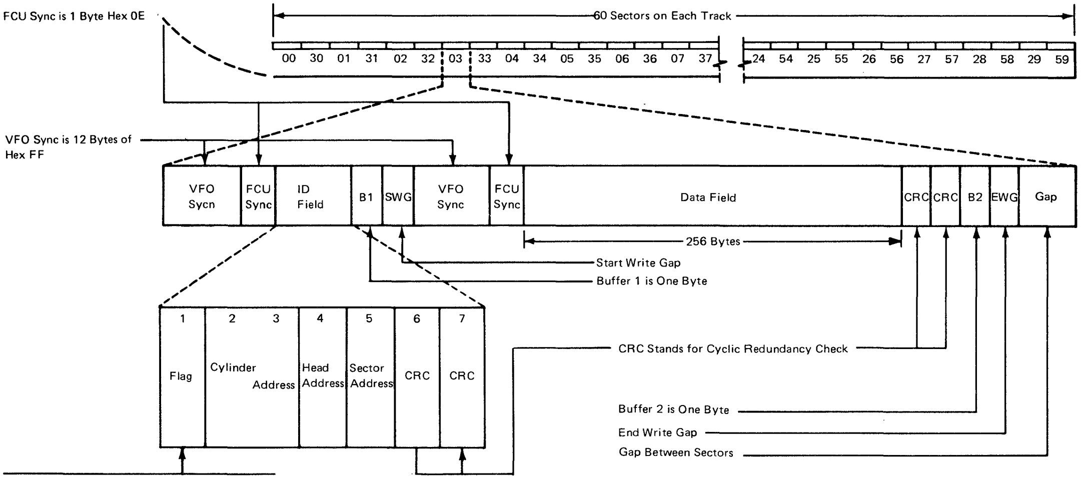
CE diagnostics = Cyl 167 or 301

CE track = Cyl 168 or 302

CROSS SECTION OF DISK



DISK SECTOR FORMAT



Flag Byte:

Bits 0-3—Do not care

Bits 4 and 5—Unassigned

Bits 6 and 7—00 = Good primary sector
10 = Defective primary sector
01 = Good alternate sector
11 = Defective alternate sector

DISK ERROR CONDITIONS

Sector Synch Check: Sync byte compare failed on either ID or data field.

Offtrack Check: Off servo track condition detected.

CRC Check: A cyclic code noncompare was detected on a data field during a read or scan or an ID field — NRF will be set concurrently if CRC causes sector not hit after two index pulses.

Parallel Parity Check: An I/O channel check was detected on DBO or CCB during execution of an SIO — equipment check is set concurrently.

Write Echo Check: Noncompare of serial write data and write data echo — equipment check set concurrently.

Channel Overrun: *Cycle Steal Response* was not received to maintain channel throughput — equipment check set concurrently.

Data Unsafe: A select unsafe, write unsafe, or servo unsafe was detected.

Invalid Seek Address: An attempt was made to seek beyond cylinder capacity.

Attachment Equipment Check: Attachment hardware check was detected.

No Record Found: Sector specified in DCF could not be found in one full revolution — not a hardware check.

Seek Check: An actuator hangup, actuator behind home in restricted area or PLO out of sync during a data operation.

Serdes Check: A mismatch detected between parallel and serial hardware checks — equipment check set concurrently.

Write Check: Write current without write selected — equipment check set concurrently.

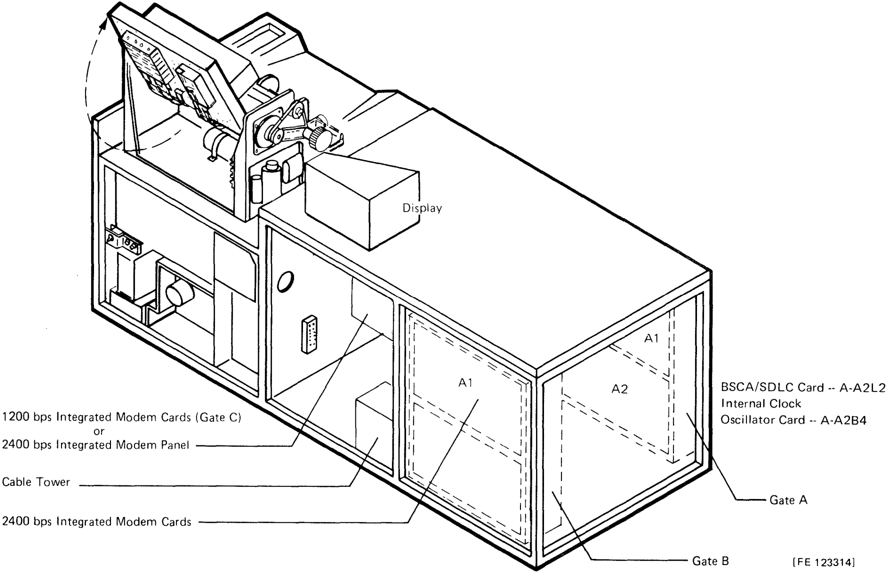
Interface Transfer Error: A hardware error occurred during channel transfer of data.

Interrupt Timeout Check: A control program timeout occurred within one to two seconds after the issuance of a microcontrol command — equipment check set concurrently.

Select Unsafe: Incorrect head selection during a write operation — data unsafe and not ready set concurrently.

Write Unsafe: Write selected no write transitions or write not selected and write current on. Data unsafe and not ready set concurrently.

Brake Failure: Write selected and head offtrack or write selected and PLO out of sync — data unsafe and not ready set concurrently.

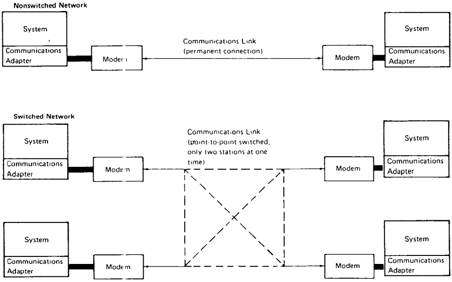

Display
A1
A1
A2
BSCA/SDLC Card -- A-A2L2
Internal Clock
Oscillator Card -- A-A2B4
1200 bps Integrated Modem Cards (Gate C)
or
2400 bps Integrated Modem Panel
Cable Tower
2400 bps Integrated Modem Cards
Gate A
Gate B

[FE 123314]

COMMUNICATION NETWORK CONFIGURATIONS

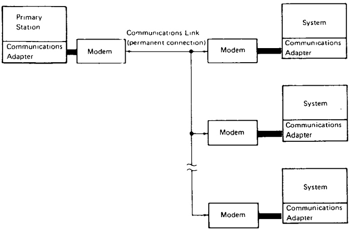
Point-to-Point Networks

Secondary Stations

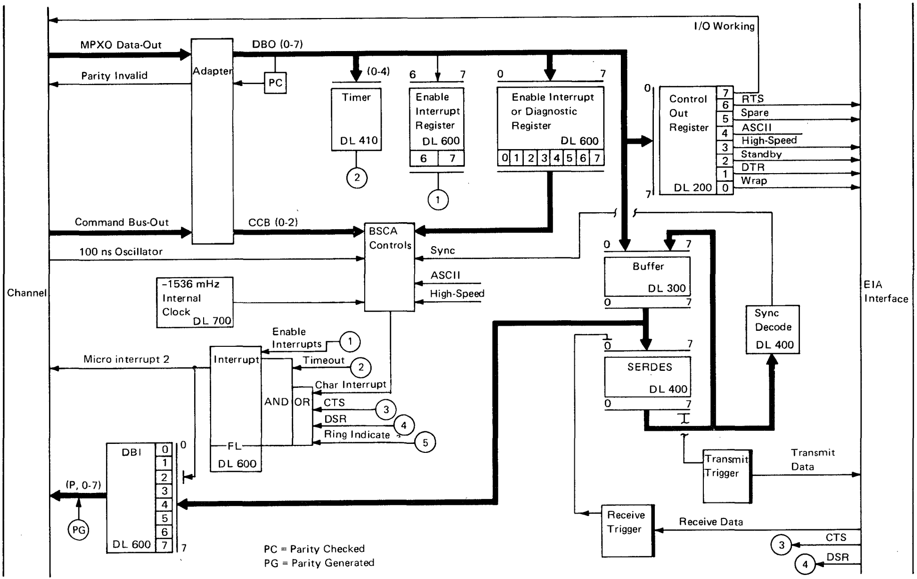


Multipoint Network

Secondary Stations



BSCA DATA FLOW



PC = Parity Checked
PG = Parity Generated

BSCA ONLINE TEST

This program provides a method of communicating with another system having an equivalent online test. This program resides on diagnostic diskette 02, and can be executed by entering the program ID BSCA in the SCP main menu.

This online test has two routines, requester and responder. The requester routine is used when your System/32 is used to request a test. This test, dependent on option selection, will either transmit a message to the responder, receive a specific message from the responder, or transmit a message and receive the same message from the responder.

Thus, you as the requester, can select a test type (xx) and the number of times you wish to repeat the test (yy).

The responder routine is used to support other systems that are requesting a test. Thus, to run the online test, one system must be a requester and the other a responder.

Operating Procedure — Requestor

1. BSCA configuration must have been done. See Section 2 of the BSCA User's Guide for configuration details.

2. IMPL SCP from the fixed disk.

3. Enter BSCA when main menu appears.

4. After the online test is loaded, select suboption 3 (program options) to:
   Scan or alter the BSCA configuration.
   Set terminal ID (if you are a multipoint tributary) or the terminal address for the system you wish to communicate with (only if multipoint).
   Enter the message to be sent or received (only if test type 00 or 01). If you use this option, you must enter the framing control characters at the beginning and end of the text message. (That is, STX—message—ETX, or for transparency, DLE STX—message—DLE ETX).
   Loop and/or bypass print options (when all suboptions are done, the system returns to the main option menu).

5. Select option 1 (requestor routine) from the main option menu (screen 0400).

6. Key in test type xx and number of times to repeat the test yy. yy can be any value 00 thru 99.

XX	Test Type
00	Requestor sends the message you entered in step 4, option 3.
01	Requestor sends the message you entered in step 4, the responder then sends it back.
02	Receive 256 EBCDIC characters from responder (transparent operation).
04	Receive 245 EBCDIC characters from responder (nontransparent).
05	Receive 117 ASCII character from responder.
06	Receive 36 ASCII character (A-Z, 0-9).
14	Receive 36 EBCDIC character (A-Z, 0-9).
15	Receive 84 EBCDIC character (74 '00's and 10 SYNs).
16	Receive 80 EBCDIC character (40 'AA's and 40 '55's).
19	Receive 290 EBCDIC character (280 '00's and 10 SYNs—transparent).

Note: The responder should be ready with its online test before you enter the xx and yy values. For point-to-point leased line and multipoint networks, the test will begin when you press the enter key.

7. For point-to-point switched networks, dial the phone number of the responder. When dialing, you must be in talk mode until the call is answered. The responder then goes to data mode. Now go to data mode and the communication link will be established. Then press the enter key to begin the test.

For explanation of error messages, see the BSCA Diagnostic Service Guide.

Operating Procedure — Responder

1. Follow requestor procedure up to (not including) step 4.

2. Select option 2 (responder routine) from the main option menu.

3. Press the enter key and wait for phone call from requestor to establish the communication link.

For explanation of error messages, see the BSCA Diagnostic Service Guide.

For further details of operation and additional information, see the BSCA Diagnostic Service Guide.

BSCA ERROR CONDITIONS

Micro Interrupt Overrun

- A micro interrupt overrun is generated when a character interrupt is generated and there is still a character interrupt pending.

- During transmit operations, a character micro interrupt is generated during character time. At this time, data is also transferred from the BSCA buffer to the SERDES register. However, if the buffer was not loaded before character time, a character micro interrupt is pending. Because the previous micro interrupt has not been processed, the buffer is empty when the second character micro interrupt is generated. Therefore, a character overrun condition exists. The BSCA continues to transmit wrong information and error recovery becomes necessary. To recover from a character overrun condition, the message must be retransmitted.

- During receive operations, a character micro interrupt is generated when the buffer is to be transferred to the CPU. At the same time, the BSCA continues to receive data and fills the SERDES register. A micro interrupt overrun occurs when the microprogram did not transfer the buffer data to the CPU before the second character micro interrupt occurred. Both the SERDES register and the buffer are full of data.) The BSCA logic then gates data from the SERDES register to the buffer at character time. As a result, the data that was in the buffer is lost.

DBO Parity

- Data on DBO is checked at 'strobe' time. DBO data is valid during this time.

- If a parity check (DBO even) occurs, CBI 5 is activated to indicate a DBO parity check.

- A DBO parity check results in a machine check.

Microprogram Detected Errors

The following error conditions are sensed by the microprogram:

- Invalid ASCII character.

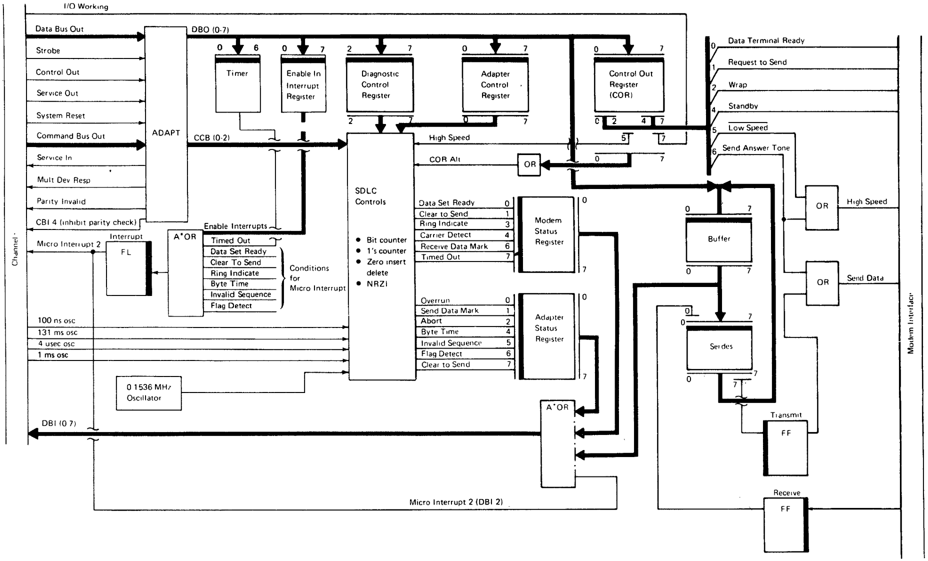
- Abortive disconnect.

- Receive timeout (3.25 seconds).

- BCC error.

- Lost data error. (The current address equals the stop address during receive operations and a valid ending character has not been received.)

SDLC DATA FLOW



SDLC LINK TEST

This program provides a method of communicating with another system having an equivalent link test. This program resides on diagnostic diskette 02, and can be executed by entering the program ID SDLC in the main menu.

This online test has two routines, primary and secondary. The primary routine is used when your System/32 is used to send a test message. This test will transmit a message to the secondary and receive the same message from the secondary.

You as the primary, can select a test type (xx) and the number of times you wish to repeat the test (yy).

The secondary routine is used to support other systems that are sending a test. To run the link test, one system must be a primary and the other a secondary.

Operating Procedure — Primary

1. SDLC configuration must have been done. See Configuration of SDLC Diagnostics in the *Diagnostic Service Guide (SDLC)* (MAP Section 3200).

2. IMPL SCP from the fixed disk.

3. Enter SDLC when main menu appears.

4. After the online test is loaded, select suboption 3 (program options) to:
   - Scan or alter the SDLC configuration.
   - Specify message block size (used only for test type 08 or 09).
   - Data field entry to enter up to 15 hex bytes (used only for test type 00, 09, and 10).
   - Print status errors as they occur.
   - Loop on first message selected.
   - Bypass errors.
   - Print data received.

   After selecting suboption, main option menu will be returned.

5. Select option 1 (primary routine) from the main option menu (screen 0420).

6. Key in test type xx and number of times to repeat the test yy. yy can be any value 00 through 99.

XX Value	Content of Message
00	Primary station sends message from storage. (You must put the message into storage)
01	256 EBCDIC characters
02	36 EBCDIC characters A-Z, 0-9
03	80 EBCDIC characters 40 'AA' and 40 '55'
04	256 X'00'
05	256 X'FF'
06	X'F3' (just TEST command)
07	Transmit only all 256 characters
08	Random data of specified block size
09	Repeated character of specified block size (use option 3 display 0422)
10	Transmit only data specified by you. (You enter message into storage)

Note: The secondary should be ready with its link test before you enter the xx and yy values. For point-to-point leased line and multipoint networks, the test will begin when you press the ENTER key.

7. For point-to-point switched networks, dial the phone number of the secondary. When dialing, you must be in talk mode until the call is answered. The secondary then goes to data mode. Now go to data mode and the communication link will be established. Then press the ENTER key to begin the test.

For explanation of error messages, see the *Diagnostic Service Guide (SDLC)* (MAP Section 3200).

Operating Procedure – Secondary

1. Follow primary procedure up to (not including) step 5.
2. Select option 2 (secondary routine) from the main option menu.
3. Press the ENTER key and wait for a phone call from the requestor to establish the communication link.

For explanation of error messages, see the *Diagnostic Service Guide (SDLC)* (MAP Section 3200).

For further details of operation and additional information, see the *Diagnostic Service Guide (SDLC)* (MAP Section 3200).

SDLC ERROR CONDITIONS

Micro Interrupt Overrun

A micro interrupt overrun is generated when a byte time interrupt is generated and there is still a byte time interrupt pending.

During transmit operations, a byte time micro interrupt is generated when data is transferred from the SDLC buffer to the SERDES register. However, if the buffer was not loaded before byte time, a byte time micro interrupt is pending. Because the previous micro interrupt has not been processed, the buffer is empty when the second byte time micro interrupt is generated. Therefore, an overrun condition exists. The SDLC stops transmitting and error recovery becomes necessary. To recover from an overrun condition, the message must be retransmitted.

During the receive operations, a byte time micro interrupt is generated when the buffer is to be transferred to the CPU. At the same time, the SDLC continues to receive data and fills the SERDES register. A micro interrupt overrun occurs when the microprogram did not transfer the buffer data to the CPU before the second byte time micro interrupt occurred. (Both the SERDES register and the buffer are full of data.) The SDLC logic then gates data from the SERDES register to the buffer at character time. As a result, the data that was in the buffer is lost.

DBO Parity

Data on DBO is checked at 'strobe' time. DBO is valid during this time. If a parity check (DBO even) occurs, DBI 5 is activated to indicate a DBO parity check. A DBO parity check results in a machine check.

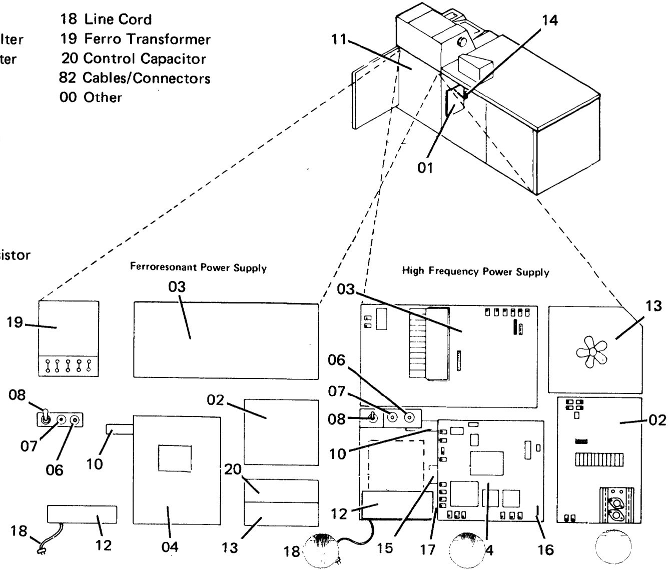
Microprogram Detected Errors

The following error conditions are sensed by the microprogram:

- Invalid frame
- Abortive disconnect
- Inactivity timeout (3.25 seconds)
- BCC error

POWER LOCATIONS

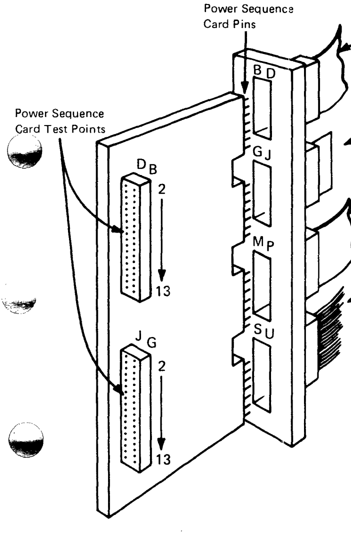
01 Sequence Card
02 Dual Level Supply/Filter
03 Multilevel Supply/Filter
04 AC Board
06 Fuse F102
07 Fuse F101
08 Power On/Off Switch
10 Control Capacitor
11 Thermal
12 Line Filter
13 Fan
14 Jumper Card
15 Lightening Strike Resistor
16 Surge Resistor
17 RC Network
18 Line Cord
19 Ferro Transformer
20 Control Capacitor
82 Cables/Connectors
00 Other



POWER SEQUENCE CARD TEST POINTS (PSC TP)

D		B
- [illegible] Vdc	2	- 6 Vdc Control
- [illegible]k K1	3	Pick K2
- 24 Vdc Control	4	+ 24 Vdc Control
+ 5 Vdc Control	5	+ 6 Vdc Control
+ Pwr-On Rst SS	6	+ Pwr Reset
+ Start Dual SS	7	- Delay SS
+ Start Multi SS	8	Ground
Thermal Sns	9	+ Off
[illegible]vate Line Fault	10	- Start Dual
Sys POR	11	Disk Brake Fault
- Start Multi	12	
- Stop Multi	13	Stop Dual

J		G
[illegible] V UV	2	+ 12 V UV
[illegible] V UV	3	- 24 V No Volt Test
Svcm Test	4	+ 8.5 V UV
OV Inhibit	5	+ 24 V UV
+ 8.5 V OV	6	- 5 V UV
- 4 V OV	7	-24 V OV
+ 24 V No Volt Test	8	+ 6 V OV
- 5 V OV	9	- 12 V UV
+ 24 V OV	10	- 24 V UV
+ 12 V OV	11	- 4 V UV
- 12 V OV	12	
+ 5 V OV	13	+ No Voltage Dual



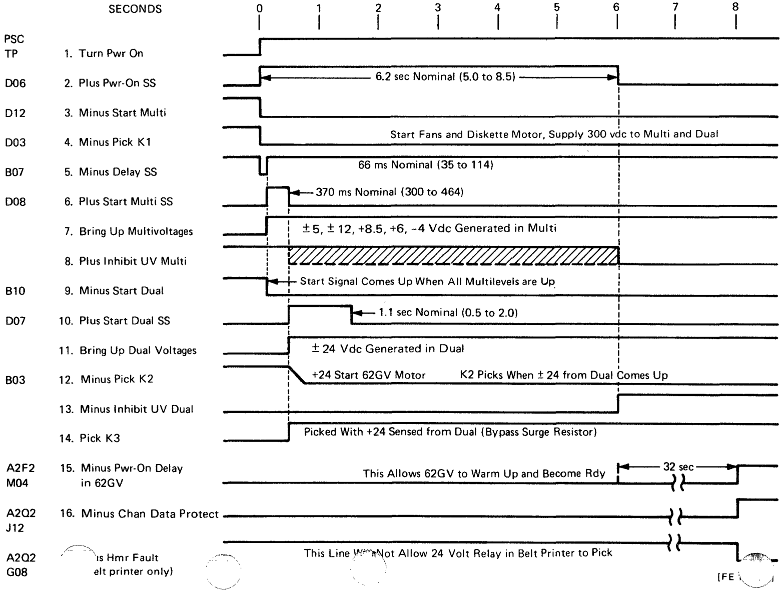
Flat cable going to 01A - A1A2 for CE Panel Display power-on switch, etc.

Jumper card for testing ties all the SS outputs back to the sequence card.

Flat cable going to the AC board. Control voltages, K1 and K2 drivers, thermal sense, channel data protect, etc.

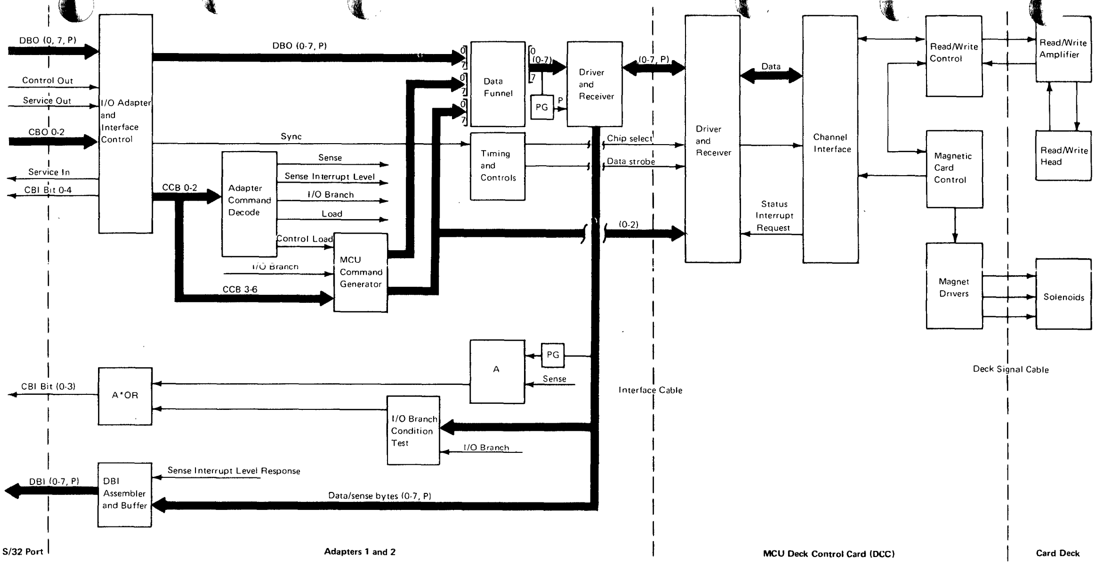
Discrete wire cable from multi, dual controls start and stop of multi and dual, senses all voltages for OV, UV, OC.

POWER-ON SEQUENCE (SYNC ON AND OFF PSCTP B9)


SECONDS
0
1
2
3
4
5
6
7
8
PSC TP
1. Turn Pwr On
D06
2. Plus Pwr-On SS
6.2 sec Nominal (5.0 to 8.5)
D12
3. Minus Start Multi
D03
4. Minus Pick K1
Start Fans and Diskette Motor, Supply 300 vdc to Multi and Dual
B07
5. Minus Delay SS
66 ms Nominal (35 to 114)
D08
6. Plus Start Multi SS
370 ms Nominal (300 to 464)
7. Bring Up Multivoltages
± 5, ± 12, +8.5, +6, -4 Vdc Generated in Multi
8. Plus Inhibit UV Multi
B10
9. Minus Start Dual
Start Signal Comes Up When All Multilevels are Up
D07
10. Plus Start Dual SS
1.1 sec Nominal (0.5 to 2.0)
11. Bring Up Dual Voltages
± 24 Vdc Generated in Dual
B03
12. Minus Pick K2
+24 Start 62GV Motor
K2 Picks When ± 24 from Dual Comes Up
13. Minus Inhibit UV Dual
14. Pick K3
Picked With +24 Sensed from Dual (Bypass Surge Resistor)
A2F2 M04
15. Minus Pwr-On Delay in 62GV
This Allows 62GV to Warm Up and Become Rdy
32 sec
A2Q2 J12
16. Minus Chan Data Protect
A2Q2 G08
ıs Hmr Fault
elt printer only)
This Line W Not Allow 24 Volt Relay in Belt Printer to Pick
[FE

Note: Approximately 40 seconds after power switch is turned on, 62GV becomes ready and 24 volt relay in belt printer picks. If IMPL diagnostics are run before 40 seconds have elapsed, a Wrap Error on belt printer and 62GV will occur.

MCU ADAPTER DATA FLOW



MCU ERROR CONDITION AND SENSE BYTE INFORMATION

Sense Byte 0

Bit **Description**

0 *DPE* indicates a device parity error. A bit or bits were lost in transferring data to or from the MCU. The IOB is posted with error completion.

1 *TK1* indicates the read/write head is at track 1. This is an error condition if a home command is issued and the read/write head is at track 1. In this case, the IOB is posted with error completion and bit 2 of sense byte 0 is set.

2 *INV* indicates the command or command modifier is invalid or the command cannot be accepted by the MCU because of error conditions. Interrogation of the sense bytes and flag byte describes the error conditions. The IOB is posted with error completion.

3 *CARD* indicates a card is present in the MCU in a valid position. This is an error condition in the following cases:

1. An exit, eject, stack, read, or write command is attempted and *CARD* is not on. In this case, the IOB is posted with error completion and bit 2 of sense byte 0 is set.
2. *CARD* is not on after a feed command. The IOB is posted with error completion.

4 *ENAB* indicates the MCU interrupts have been enabled by microcode. Power on, IPL, or microcode disables interrupts. This is an error condition in the following cases:

1. An exit, eject, track step-up, stack, feed, home, read, or write command is attempted and interrupts are not enabled. The IOB is posted with error completion and bit 2 of sense byte 0 is set.
2. An interrupt is received from a completed track step-up, stack, feed, or home command and interrupts are not enabled. The IOB is posted with error completion.

5 *LUP* indicates a card handling error. See descriptions of *JAM* or *TCRD* in Sense Byte 1.

6 *ERR* indicates a read or write error. See descriptions of *FERR, DNF,* or *OVRN* in sense byte 1.

7 *START* indicates the start latch is on. This condition is indicated by the READY light on the MCU operator console. The latch is set on by the START key on the MCU operator console and set off by the STOP key on the MCU console or by the set/reset indicators and alarm command. If an exit, eject, track step-up, stack, feed, home, read, or write is attempted and the start latch is not IOB error completion is posted and bit 2 of sense byte 0 is set.

Sense Byte 1

Bit **Description**

0 *RDY* indicates the MCU is ready to accept a command. Once a command is accepted by the MCU, *RDY* turns off until the command is completed. If *JAM, TCRD,* or *DNF* occurs during execution of the command, *RDY* remains off until the condition is removed. This is an error condition if:

1. An exit, eject, track step-up, stack, feed, home, read or write is attempted and *RDY* is not on. The IOB is posted with error completion and bit 2 of sense byte 0 is set.

2. An interrupt is received from a completed track step-up, stack, feed, or home command and *RDY* is not on. The IOB is posted with error completion.

1 *JAM* indicates a card jam has occurred in the MCU transport. *RDY* is held off until the condition is corrected. The IOB is posted with error completion.

2 *TCRD* indicates a card is present in the single feed slot. RDY is held off until the condition is corrected. This is an error condition if the *TCRD* is not on after an exit or eject. The IOB is posted with error completion.

3 *TK50* indicates the MCU read/write head is at track 50. This is an error condition if a track step-up command is attempted and the read/write head is at track 50. The IOB is posted with error completion and bit 2 of sense byte 0 is set.

4 *OVRN* indicates that an interrupt occurred from the MCU and a previous interrupt had not yet been serviced. The IOB is posted with error completion.

5 *FERR* indicates a read or write intracharacter error occured on a read, write, or write readback checking. The read or write IOB is posted with error completion.

6 *DNF* indicates data not found. An error has occurred on a read, write, or write readback checking. The read or write IOB is posted with error completion.

7 *TIMEOUT* indicates a timeout interrupt has occurred. This is an error condition in the following cases:

1. A track step-up, stack, feed, home, read, or write command is started and the interrupt received at the completion of the operation is a timeout interrupt. The IOB is posted with error completion.

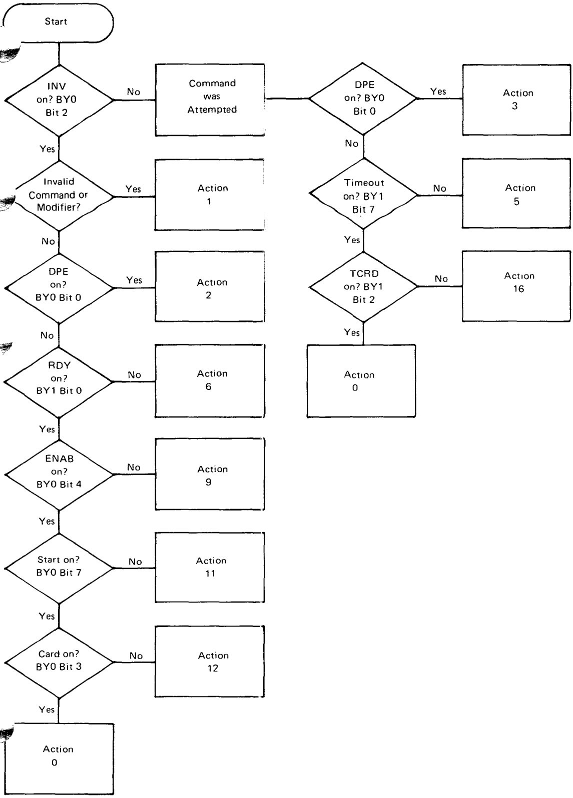
2. An eject or exit command is started and the interrupt received at completion of the operation is not a timeout interrupt. The IOB is posted with error completion.

MCU IOB ERROR COMPLETION FLOWCHARTS BY COMMAND TYPE

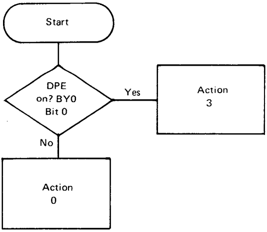
These flowcharts can be used to determine the cause of an IOB being posted with error completion. Any indicator that is not referenced in a command flowchart is not a determining factor in the IOB being posted with error completion. BY0 BIT2 means the indicator can be found in sense byte 0 bit 2. FLG BIT2 means the indicator can be found in the flag byte bit 2.

An action description follows the flowcharts.

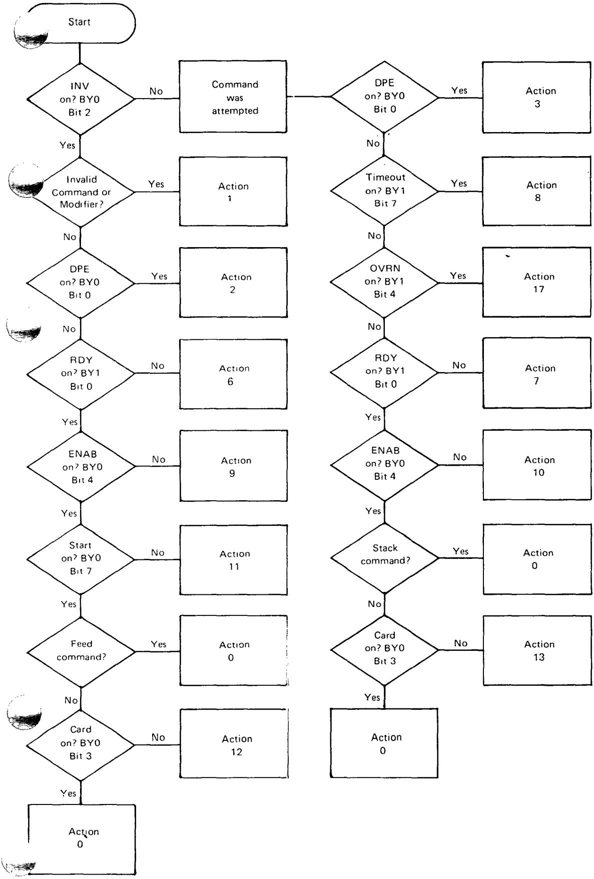
Eject or Exit IOB Error Completion



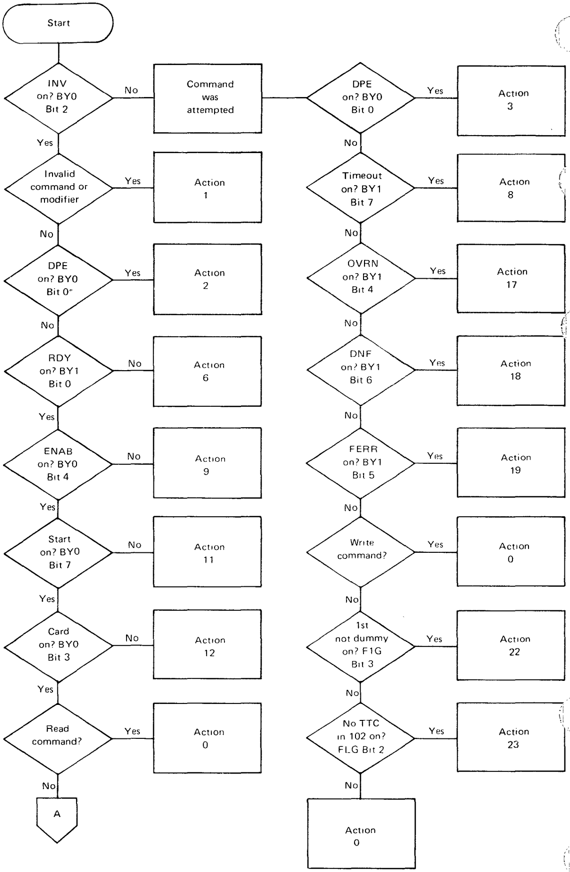
Sense or Set/Reset Indicators and Alarm Job Error Completion



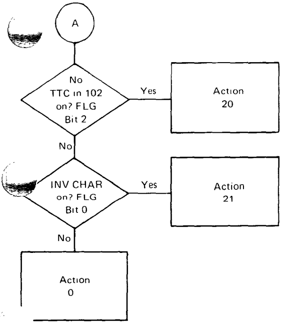
Feed or Stack IOB Error Completion



Read or Write IOB Error Completion 1 of 2



Read or Write IOB Error Completion 2 of 2



Action **Description**

0 An unidentified error has occurred. An improper flowchart branch was taken, the flowchart is in error, or the microcode is in error. Retry the flowchart.

1 Correct the IOB command or command modifier in error.

2 A device parity error occurred while interrogating device status before the command was executed. This is a hardware error.

3 A device parity error occurred during the execution of a command. The results of the command are unpredictable. This is a hardware error.

5 The interrupt received from the eject or exit command was not a timeout interrupt. This is a hardware error. If TCRD (byte 1 bit 2) is on, the command executed correctly.

6 The MCU is not ready to accept the command. JAM (byte 1 Bit 1) or TCRD (byte 1 Bit 2) being on causes this condition.

7 The MCU did not ready after execution of a command JAM or TCRD on causes this condition.

8 A command was attempted and the MCU did not respond within the expected amount of time. JAM, TCRD, or DNF will cause this condition.

9 The MCU will not accept the command because interrupts are not enabled. This is a hardware error.

10 Interrupts are not enabled after execution of the command. This is a hardware error.

11 The MCU will not accept the command because START is not on. Press the START key.

12 The MCU will not accept the command because a card is not present in a valid position in the MCU. Move a card to a valid position in the MCU.

13 A feed command was issued and did not result in a card being present in a valid position in the MCU. The hopper is empty or it failed to feed a card.

14 The MCU will not accept the track step-up command as the head is already at track 50.

15 The MCU will not accept the home command as the head is already at track 1.

16 An eject or exit command was issued and did not result in a card being present in the single feed slot. JAM being on will cause this condition.

17 An interrupt was received before a previous interrupt could be serviced. This is a hardware error.

Action **Description**

18 A read or write error occurred or a blank card is being read. This error can occur while reading, writing, or write readback checking. Retry the operation. If the error persists, clean the failing card.

19 A read or write readback intracharacter error occurred. Retry the operation. If the error persists, clean the failing card.

20 The write command was not accepted by the MCU as no track terminator character was found in the 102 character buffer in main storage. Correct the main storage buffer.

21 The write command was not accepted by the MCU as an invalid character was found in the 102 character buffer in main storage before a track terminator was found. Correct the main storage buffer.

22 A read command was terminated because the first character found on this track of the card was not a dummy character, or an error occurred reading the first character. Retry the operation. If the error persists, clean the failing card.

23 The read command was terminated because no track terminator was found on this track of the card in 102 characters, or an error occurred before the track terminator was found. Retry the operation. If the error persists, clean the card.

Appendix O. Mag Card Unit Character Translate Table

When an IOB with a command of X'50' and a modifier of X'0A' is issued, 256 bytes are moved from the main storage buffer pointed to by the buffer address in the IOB to the control storage character conversion table.

Character conversion from EBCDIC to tilt/rotate on a write or from tilt/rotate to EBCDIC on a read requires a control storage character conversion table. (Figure O-1 shows the format of the conversion table and Figure O-2 shows an example of conversion table loaded.)

The table is 128 words (256 bytes) in length. The high-order byte of each of the 128 words is the character conversion table. Conversion from tilt/rotate on a read is accomplished by using the tilt/rotate character's displacement into the table to find its EBCDIC equivalent. Conversion from EBCDIC to tilt/rotate on a write is accomplished by comparing the characters in the table against the EBCDIC value to be converted until a match is found.

When the match is found, the displacement into the table where the match is found represents the tilt/rotate value for that character. The character comparison is against the high-order byte of each of the 128 words starting with the beginning of the table. It is not always necessary to test every EBCDIC value in the table for a character match on a write. By testing only the required sections, EBCDIC to tilt/rotate conversion time is optimized.

Byte	EBCDIC Character for Tilt/Rotate Value (hex)	Byte	Contents[1] (hex)
1	00	2	00
3	01	4	00
5	02	6	00
7	03	8	00
9	04	10	00
11	05	12	00
13	06	14	00
15	07	16	00
17	08	18	00
19	09	20	00
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
225	70	226	Indicator byte for EBCDIC 00 to 0F
227	71	228	Indicator byte for EBCDIC 10 to 1F
229	72	230	Indicator byte for EBCDIC 20 to 2F
.	.	.	
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
253	7E	254	Indicator byte for EBCDIC E0 to EF
255	7F	256	Indicator byte for EBCDIC F0 to FF

[1] Even bytes from 2 through 224 must contain 00.

Figure O-1. Format of Conversion Table

Word	Data	Word	Data
1	CA00	41	3F00
2	A800	42	9300
3	6000	43	0500
4	0900	44	1500
5	9800	45	8300
6	9700	46	8400
7	7E00	47	A400
8	9100	48	A700
9	3F00	49	F900
10	6100	50	F000
11	3F00	51	2F00
12	2500	52	0700
13	6B00	53	F600
14	5E00	54	F500
15	8600	55	F200
16	8700	56	A900
17	A600	57	4000
18	A200	58	F400
19	2300	59	4100
20	3F00	60	2900
21	8900	61	F800
22	7D00	62	F700
23	4B00	63	F300
24	E000	64	F100
25	3F00	65	6D00
26	9600	66	E800
27	2700	67	1A00
28	3F00	68	3F00
29	8100	69	D800
30	9900	70	D700
31	A500	71	4E00
32	9400	72	D100
33	8200	73	3F00
34	8800	74	6F00
35	2A00	75	1600
36	3800	76	2800
37	9200	77	6B00
38	8500	78	7A00
39	9500	79	C600
40	A300	80	C700

Figure O-2 (Part 1 of 2). Example of Conversion Table Loaded Each Time Mag Card is Initialized

Word	Data	Word	Data
81	E600	97	C200
82	E200	98	C800
83	3600	99	3F00
84	3F00	100	2C00
85	C900	101	D200
86	7F00	102	C500
87	4B00	103	D500
88	A100	104	E300
89	3F00	105	3F00
90	D600	106	D300
91	3F00	107	3900
92	0300	108	0600
93	C100	109	C300
94	D900	110	C400
95	E500	111	E400
96	D400	112	E700
Byte 2 indicator byte for:			
113	4D48	X‘00’ to X‘0F’	
114	5DD7	X‘10’ to X‘1F’	
115	3F05	X‘20’ to X‘2F’	
116	3FD9	X‘30’ to X‘3F’	
117	4AA6	X‘40’ to X‘4F’	
118	6C7E	X‘50’ to X‘5F’	
119	7C76	X‘60’ to X‘6F’	
120	E932	X‘70’ to X‘7F’	
121	E11F	X‘80’ to X‘8F’	
122	5B1F	X‘90’ to X‘9F’	
123	0C0B	X‘A0’ to X‘AF’	
124	0AFF	X‘B0’ to X‘BF’	
125	5C71	X‘C0’ to X‘CF’	
126	50F1	X‘D0’ to X‘DF’	
127	7BB0	X‘E0’ to X‘EF’	
128	5AEF	X‘F0’ to X‘FF’	

Figure O-2 (Part 2 of 2). Example of Conversion Table Loaded Each Time Mag Card is Initialized

If the same EBCDIC value is used for more than one tilt/rotate character and that EBCDIC value is to be converted, the tilt/rotate character for the first time that EBCDIC value is found in the conversion table is the character written. The exception is the EBCDIC X'3F' which is the value used for an invalid tilt/rotate.

Any EBCDIC character that is not found in the conversion table is an invalid write character.

The low bytes of the last 16 words of the table are used as indicator bytes for the different EBCDIC groups. Each of the EBCDIC groups from X'0x' to X'Fx' has its own indicator byte. The indicator bytes are used to determine if a section of 16 positions in the character conversion table are to be tested for a character match. The table is divided into eight 16-position sections:

Section 1	Tilt/rotate X'00' - X'0F'
Section 2	Tilt/rotate X'10' - X'1F'
Section 3	Tilt/rotate X'20' - X'2F'
.	.
.	.
.	.
.	.
Section 8	Tilt/rotate X'70' - X'7F'

A bit off in the indicator byte means 'test that section'.

Example 1: An EBCDIC X'F4' is in the write buffer and is to be converted into its tilt/rotate equivalent. It has been predetermined that any character from X'F0' to X'FF' has a tilt/rotate equivalent that is only found in Section 4 of the table (X'30' to X'3F'). Therefore, the indicator byte for EBCDIC X'F0' to X'FF' (the low byte of the 128th word in the table) contains X'EF' (1110 1111). When a character match for X'F4' is attempted, only the 16 positions in Section 4 of the table are compared.

Example 2: An EBCDIC X'D6' is in the write buffer and is to be converted to its tilt/rotate equivalent. It has been predetermined that any character from X'D0' to X'DF' has a tilt/rotate equivalent that resides in Section 5 (X'40' to X'4F'), Section 6 (X'50' to X'5F'), or Section 7 (X'60' to X'6F'). Therefore, the indicator byte for EBCDIC X'D0' to X'DF' (the low byte of the 126th word of the table) contains X'F1' (1111 0001). When a character match for X'D6' is attempted, only the 16 positions in Sections 5, 6, and 7 are compared.

Appendix P. System/32 Scientific Instruction Set Summary

System/32 scientific programs are executed under the control of an interpreter resident in the control storage increment. The object program language, processed by the interpreter, is called the scientific instruction set. The major component of the scientific instruction set is the scientific instruction. A 3-byte scientific instruction is generated for each executable statement in the processed source string. Byte 0 contains the operation code (bits 0 through 6) and the index bit (bit 7). Bytes 1 and 2 contain a 16-bit System/32 address. The effective address for a scientific instruction is the address part of the instruction plus the scientific instruction set XR (index register) if the index bit is 1. Scientific instruction addresses consistently refer to the leftmost byte of entries in the symbol table.

The principal scientific instruction set registers are:

- XR. Index register: A 2-byte value used in indexing for effective address.
- XMR. Index multiplier register: 2 bytes used for temporary storage in computing index values.
- BR. Binary register: 4-byte two's complement register used for integer arithmetic.
- FR. Floating-point register: Holds short or long precision floating-point hexadecimal value in System/360 format.
- Scientific IAR. Instruction address register: Contains 2 bytes that hold the address for the next scientific instruction to be executed.
- AR. Address register: Holds addresses for certain scientific operands.
- CR. Condition code register: 1 byte containing the result of a compare operation.

When control is passed to the load module for execution, the first instruction in the program entry record is a branch to the interpreter code. The interpreter locates the first scientific instruction following the branch and before decoding and executing it, sets the scientific IAR to point to the next instruction. This continues until all the scientific instructions are executed. In executing the various instructions, other interpreter modules or sections of code may be used.

The following table describes the scientific instructions and operations:

Hex Value	Scientific Instruction Mnemonic	Scientific Macro Instruction	Functional Description
X'00'	CGO		Sequence control for computed GOTO
X'02'	GO	$GOTO	Sequence control for GO branch
X'04'	IFGO	$BIF or $RIF	Sequence control for arithmetic IF
X'06'	XL	$XLD	Index register load
X'08'	XA	$XADD	Index add
X'0A'	XLI	$XLI	Index register load immediate
X'0C'	XST	$XST	Index register store
X'0E'	XM	$XMLT	Index multiply

Hex Value	Scientific Instruction Mnemonic	Scientific Macro Instruction	Functional Description
X'10'	XMA	$XMTA	Index multiply and add
X'12'	XMLI	$MLI	Index multiplier register load immediate
X'14'	XMST	$MST	Index multiplier register store
X'16'	BST	$BST	Binary register store
X'18'	BD	$BDIV	Binary register divide
X'1A'	BA	$BADD	Binary register add
X'1C'	BS	$BSUB	Binary register subtract
X'1E'	BM	$BMLT	Binary register multiply
X'20'	BL	$BLD	Binary register load
X'22'	HST	$HST	Binary register half store
X'24'	HD	$HDIV	Binary register half divide
X'26'	HA	$HADD	Binary register half add
X'28'	HS	$HSUB	Binary register half subtract
X'2A'	HM	$HMLT	Binary register half multiply
X'2C'	HL	$HLD	Binary register half load
X'2E'	RST	$RST	Floating-point register store
X'30'	RD	$RDIV	Floating-point register divide
X'32'	RA	$RADD	Floating-point register add
X'34'	RS	$RSUB	Floating-point register subtract
X'36'	RM	$RMLT	Floating-point register multiply
X'38'	RL	$RLD	Floating-point register load
X'3A'	DST	$DST	Floating-point register double-precision store
X'3C'	DD	$DDIV	Floating-point register double-precision divide
X'3E'	DA	$DADD	Floating-point register double add
X'40'	DS	$DSUB	Floating-point register double-precision subtract
X'42'	DM	$DMLT	Floating-point register double-precision multiply

Hex Value	Scientific Instruction Mnemonic	Scientific Macro Instruction	Functional Description
X'44'	DL	$DLD	Floating-point register double-precision load
X'46'	ADR	$ALI	Addressing operations
X'48'	INV	$INVK	Invoke branch
X'4A'	DOBGN	[1]	DO loop initialization
X'4C'	DOEND	[1]	DO loop variable control
X'4E'	CALL	$CALL	Subprogram call
X'50'	IO	[1]	Input/output control
X'52'	DED	[1]	Data element descriptor
X'54'	DODED	[1]	DO control variable DED
X'56'	HC	$HCMP	Binary register compare (Integer*2)
X'58'	BC	$BCMP	Binary register compare (Integer*4)
X'5A'	RC	$RCMP	Floating-point register compare (Real*4)
X'5C'	DC	$DCMP	Floating-point register compare (Real*8)
X'5E'	LSET	$LSET	Test condition code register
X'60'	AND	$BAND	Logical AND
X'62'	OR	$BOR	Logical OR
X'64'	NOT	$BNOT	Logical NOT

[1] These scientific instructions do not have macro instruction equivalents and cannot be used by the assembler programmer.

NOTES

NOTES

IBM Technical Newsletter

This Newsletter No.	SN21-8093
Date.	30 May 1980
Base Publication No.	SY21-0532-4
File No.	S32-36
Previous Newsletters	SN21-7957 SN21-8000

IBM System/32
System Data Areas
and Diagnostic Aids

This technical newsletter applies to version 09, modification 00 of the system control programming (Program Number 5725-SC1), and provides replacement pages for the subject publication. These pages remain in effect for subsequent versions and modifications unless specifically altered. Pages to be inserted and/or removed are:

2-15, 2-16
2-31, 2-32
2-45, 2-46
2-95 through 2-98
3-1, 3-2
3-11, 3-12
3-19, 3-20
B-1, B-2
B-11 through B-14
C-1 through C-8
I-5, I-6
L-1, L-2
L-11, L-12

Changes to text and illustrations are indicated by a vertical line at the left of the change.

Summary of Amendments

- Miscellaneous technical changes

Note: Please file this cover letter at the back of the manual to provide a record or changes.

IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901

IBM Technical Newsletter

This TNL No.	SN21-8000
Date	22 November 1978
Base Publ. No.	SY21-0532-4
File No.	S32-36
Previous TNLs	SN21-7957

IBM System/32
System Data Areas
and Diagnostic Aids

This technical newsletter, a part of version 08, modification 00 of the system control programming (Program Number 5725-SC1), provides replacement pages for the subject publication. These replacement pages remain in effect for subsequent versions or modifications unless specifically altered. Pages to be inserted and/or removed are:

Title Page, Edition Notice
2-15, 2-16
2-21, 2-22
2-25, 2-26
2-69 through 2-72
2-91, 2-92
2-92.1, 2-92.2 (added)
2-93, 2-94
2-107, 2-108
2-167 through 2-170
C-3, C-4
D-3, D-4
F-1 through F-10
H-5 through H-12
I-9, I-10
L-1 through L-4
L-9, L-10
L-10.1, L-10.2 (text rearranged)
L-15, L-16
L-16.1, L-16.2
N-5, N-6

Changes to text and illustrations are indicated by a vertical line at the left of the change.

Summary of Amendments

- Add new parameters to COPYI1 command
- Add new RENAME command
- Miscellaneous corrections

Note: Please file this cover letter at the back of the manual to provide a record of changes.

IBM Corporation, Publications, Department 245,
Rochester, MN 55901

Printed in U.S.A.

IBM Technical Newsletter

This TNL No.	SN21-7957
Date	25 Nov 1977
Base Publ. No.	SY21-0532-4
File No.	S32-36
Previous TNLs	None

IBM System/32
System Data Areas
and Diagnostic Aids

This technical newsletter, a part of version 07, modification 00 of the IBM System/32 (Program Number 5725-SC1), provides replacement pages for the subject publication. These replacement pages remain in effect for subsequent versions and modifications unless specifically altered. Pages to be inserted and/or removed are:

xi, xii	3-9.1, 3-9.2 (added)
2-13 through 2-16	A-7 through A-14
2-67, 2-68	B-5, B-6
2-87, 2-88	F-5 through F-8
2-91 through 2-96	H-7 through H-12
2-103, 2-104	H-12.1, H-12.2 (added)
2-129 through 2-132	L-1, L-2
2-135, 2-136	L-7, L-8
2-145, 2-146	L-15, L-16
3-5, 3-6	L-16.1, L-16.2 (added)

Changes to text and illustrations are indicated by a vertical line at the left of the change; new or extensively revised illustrations are denoted by the symbol ● at the left of the caption.

Summary of Amendments

- Add $MAIST library directory fast insert
- Miscellaneous changes

Note: Please file this cover letter at the back of the manual to provide a record of changes.

IBM Corporation, Publications, Department 245,
Rochester, MN 55901

Printed in U.S.A.

IBM®

International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

IBM System/32 System Data Areas and Diagnostic Aids (File No. S32-36) Printed in U.S.A. SY21-0532-4

IBM System/32
System Data Areas
and Diagnostic Aids

READER'S COMMENT FORM

Please use this form only to identify publication errors or request changes to publications. Technical questions about IBM systems, changes in IBM programming support, requests for additional publications, etc, should be directed to your IBM representative or to the IBM branch office nearest your location.

Error in publication (typographical, illustration, and so on).
No reply.

Page Number *Error*

Inaccurate or misleading information in this publication. Please tell us about it by using this postage-paid form. We will correct or clarify the publication, or tell you why a change is not being made, provided you include your name and address.

Page Number *Comment*

□ Check if reply is requested.

Note: All comments and suggestions become the property of IBM.

Name ______________________________________

Address ______________________________________

____

- No postage necessary if mailed in the U.S.A.

IBM Corporation
GSD Development Lab
Rochester, MN 55901

Attn: Publications, Dept 245