

**IBM** **Field Engineering**  
**Programming**  
**Handbook**

**System/360**

#### ACKNOWLEDGEMENT

This Handbook has been prepared by the Plant CE Department, World Trade Corporation, Japan, and is intended to be a Service Aid for the Program Support Customer Engineer.

© International Business Machines Corp. 1967

**IBM Programming Handbook**  
**DOS Version 3**

# **IBM** / **FE Supplement**

**System/Unit Programming Handbook**

**Re: Form No. 229-2129-0**

**This Supplement No. S29-0028**

**Date: September 1, 1968**

**This Supplement is issued to release information on DOS Version 3 differences. File the attached in the Field Engineering Programming Handbook, Form Number 229-2129-0.**

**File this cover letter at the back of the publication. It will then serve as a record of the changes received and incorporated.**



DOS VERSION 3

TABLE OF CONTENTS

ComReg Differences	12
DLBL - DASD Label Information Command	1
EXTENT - DASD EXTENT Information Command	2
Job Control Commands - Operator	5
Library Maintenance - V3 Differences	7
Low Core Usage	14
Partition Save Areas	13
Standard Label Build - Example	3
Standard Label Build - Work Files - Pvt Libraries	4
TPLAB - Tape Table Information Commands	1



Op	Operand
TPLAB	'label fields 3-10' 'label fields 3-13'

## 'label fields 3-10'

This is a 49-byte character string, included within apostrophes (8-5 punch), identical to positions 5-53 of the tape file label. These fields can be included in one line.

## 'label fields 3-13'

This is a 69-byte character string, included within apostrophes (8-5 punch), identical to positions 5-73 of the tape file label. These fields are too to be included on a single line. The character string must extend into column 71, a continuation character (any character) is present in column 72, and the character string is completed on the next line. The continuation line starts in column 16.

DLBL -- DASD Label Information Command

The DLBL command replaces the VOL and DLAB combination used in earlier systems. It contains file label information for DASD label checking and creation. The DLBL command must not be followed by the XTENT command. The current system will, however, continue to accept the VOL, DLAB and XTENT combination. The DLBL command has the following format:

Op	Operand
DLBL	filename,['file-ID'],[date],[codes]

## filename

From one to seven characters and identical to the symbolic name of the program DTF, which identifies the file.

## 'file-ID'

The name associated with the file on the volume. From 1 to 44 bytes of alphanumeric data, contained within apostrophes, including file-ID and, if used, generation number and version number of generation. If fewer than 44 characters are used, the field is left justified and padded with blanks. If this operand is omitted, "filename" is used.

## date

From 1 to 6 characters indicating either the retention period of the file (in the format d through dddd) or the absolute expiration date of the file (in the format yy/ddd). ddd cannot exceed 366. If this operand is omitted for an output file, a 7-day retention period is assumed and the current date is the creation date. If present, this operand is ignored for an input file.

## codes

A 2-3 character field indicating the type of file label as follows:

SD for sequential disk or for DTFPH with MOUNTED=SINGLE.



DA for direct access or for DTFPH with MOUNTED=ALL.

IS C for indexed sequential using Load Create.

IS E for indexed sequential using Load Extension, Add, or retrieve.

If this operand is omitted, SD is assumed.

Additional fields in the standard disk file label are filled with default options for output files and "DOS/360 VER 3" is used as the system code.

Op	Operand
EXTENT	[symbolic-unit], [serial-number], [type], [sequence-number], [relative-track], [number-of-tracks], [split-cylinder-track], [B=bins]

**symbolic unit** A six-character field indicating the symbolic unit (SYSxxx) of the volume for this extent. If this operand is omitted, the symbolic unit of the preceding EXTENT command is used. This operand is not required for a single volume, IJSYSxxx file-name or for a file defined with the DTF DEVADDR=SYSnnn.

**serial number** From 1 to 6 characters indicating the volume serial number for this extent. If fewer than six characters are used, the field is right-justified and padded with zeros. If this operand is omitted the volume serial number of the preceding EXTENT is used. If no serial number was provided in the EXTENT command, the serial number is not checked, and the files may be destroyed if the wrong volume is mounted.

**type** One of the following characters to indicate the extent type:

- 1 - data area (no split cylinder)
- 2 - overflow area (for indexed sequential file)
- 4 - index area (for indexed sequential file)
- 8 - data area (split cylinder)

If this operand is omitted, type 1 is assumed.

**sequence number** One to three characters containing a decimal number from 0 to 255, indicating the sequence number of this extent within a multiextent file. Extent sequence 0 is used for the master index of an indexed sequential file. If a master index is not used, the first extent of an indexed sequential file has the sequence number 1. The extent sequence number for all other types of files begins with 0. If this operand is omitted for the first extent of an ISFMS file, the extent is not accepted. This operand is not required for SD or DA files.

relative track One to five characters indicating the sequential number of the track (relative to zero) where the data extent is to begin. For example, track 0, cylinder 150 on a 2311 equals 1500 in relative track. If this field is omitted on an ISFMS file, the extent is not accepted. This operand is not required for SD or DA input files because the extents from the file labels on a disk are used.

number of tracks One to five characters indicating the number of tracks to be allotted to the file. For SD or DA input files, this operand may be omitted. For split cylinders, the number of tracks must be an even multiple of the number of tracks per cylinder specified for the file.

split cylinder track One or two characters, from 0 to 19, indicating the upper track number for the split cylinder in SD files.

bins One or two characters identifying the 2321 bin for which the extent was created or on which the extent is currently located. If this field is one character, the creating bin is assumed to be zero. There is no need to specify a creating bin number for SD or ISFMS files. If this operand is omitted, bin zero is assumed for both characters. If this operand is included and positional operands are omitted, only one comma is required preceding the key-word operand (bins). (One comma for each omitted positional operand is acceptable, but not necessary.)

---

```
// JOB BUILD STANDARD LABELS          VERSION III
// OPTION STDLABEL
// DLBL IJSYSLN,'SYSTEM WORK FILE N 0.  ',99/365,SD
// EXTENT SYSLNK,111111,8,0,0010,198,0
// DLBL IJSYS01,'SYSTEM WORK FILE N 1  ',99/365,SD
// EXTENT SYS001,111111,8,0,0011,594,3
// DLBL IJSYS02,'SYSTEM WORK FILE N 2.  ',99/365,SD
// EXTENT SYS002,111111,8,0,0014,594,6
// DLBL IJSYS03,'SYSTEM WORK FILE N 3.  ',99/365,SD
// EXTENT SYS003,111111,8,0,0017,594,9
// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE  F',99/365,SD
// EXTENT SYSRES,111111,1,0,0001,1980
// DLBL IJSYSRL,'DOS PVT REL LIB',99/365,SD
// EXTENT SYSRLB,111111,1,1,0380,1000
// DLBL IJSYSSL,'DOS PVT SRS LIB',99/365,SD
// EXTENT SYSSLB,111111,1,1,1390,600
/*
/&
// PAUSE END OF BUILD
```

### Track

0	Background program temporary (USRLABEL) label information
1	Background program standard (PARSTD) label information
2	Foreground-two temporary (USRLABEL) label information
3	Foreground-two standard (PARSTD) label information

- 4       Foreground-one temporary  
          (USRLABEL) label information
- 5       Foreground-one standard  
          (PARSTD) label information
- 6-n     Standard (STDLABEL) label information for any partition. n is 9 for 2311; 19 for 2314.

---

Sample of Control Cards to Build Standard Labels for Work Files and Private Libraries.

```
Definition           // JOB jobname
of a Private        // ASSGN SYSRLD, X'cuu'
Relocatable        // DLBL IJSYSRL, 'user identification
Library            // of private library',date, code
                  // EXTENT extent information
                  // EXEC CORGZ
                  NEWVOL RL=cylin (tracks)
                  /*
                  /&
```

A MAINT function will be required to catalog modules into the private library.

---

```
Definition           // JOB jobname
of a Private        // ASSGN SYSSLB, X 'cuu'
Source             // DLBL IJSYSSL, 'user identification of
Statement         // private library', date, code
Library            // EXTENT extent information
                  // EXEC CORGZ
                  NEWVOL SL=cylin (tracks)
                  /*
                  /&
```

A MAINT function will be required to catalog modules into the private library.

---

```
Definition           // JOB jobname
and Creation        // ASSGN SYSRLB,X'cuu'
of a Private        // DLBL IJSYSRL, 'user identification
Relocatable        // of private library', date, code
Library            // EXTENT extent information
                  // EXEC CORGZ
                  NEWVOL RL=cylin (tracks)
                  COPYR operands
                  /*
                  /&
```

COPY function is effective for private libraries only during their creation.

---

```
Definition           // JOB jobname
and Creation        // ASSGN SYSSLB,X'cuu'
of a Private        // DLBL IJSYSSL, 'user identification
Source             // of private library', date, code
Statement         // EXTENT extent information
Library            // EXEC CORGZ
                  NEWVOL SL=cylin (tracks)
                  COPYS operands
                  /*
                  /&
```

COPY function is effective for private libraries only during their creation

COMMAND	MEANING	IPL	JC	AR	SPI	WHEN ACCEPTED
ADD	Add a device to the PUB table.	X				
DEL	Delete a device from the PUB table	X				During IPL SET date and clock only
SET	Set values in the communication area.	X	X			
CLOSE	Close magnetic tape input or output file or 2311.		X			
DVCDN	Device down (not available to system)		X			
DVCUP	Device up (now available to system)		X			
MTC	Magnetic tape control		X			Between Jobs and Job Steps
RESET	Reset temporary I/O device assignments to system standard					
STOP	Stop execution of background job.		X			
UNBATCH	Terminate batch processing		X			
UCS	Load universal character set buffer		X			
ALLOC	Allocate core storage.		X	X		Between Jobs and Job Steps and after pressing the request key on 1052
MAP	List core storage allocations.		X	X	X	
PAUSE	Suppress processing (enter WAIT state).		X	X	X	
LOG	Log (print) job control statements		X	X	X	Between Jobs and Job Steps after pressing the request key on 1052, and as response to system message, and during single program initiation.

NOLOG	Suppress logging control statements.		X	X	X	
CANCEL	Cancel execution of current job.		X	X	X	
B	End-of-block or communications	X	X	X	X	During IPL between Jobs and Job Steps after pressing the request key on 1052, and as response to system message, and during single program initiation.
C	Cancel terminal response (1052).	X	X	X	X	
ASSGN	Assign Logical name.		X		X	
HOLD	Hold current foreground assignments		X		X	
LISTIO	List current I/O assignments.		X		X	
RELSE	Release current foreground assignments and unassign them at the end of any job initiated for that area.		X		X	Between Jobs and Job Steps and during single program initiation.
UNA	Set all assignments for foreground area to unassigned. The specified area must be inactive.		X		X	
MSG	Give control to a foreground communication routine.			X	X	
TIMER	Transfers timer support to indicated program.			X	X	After pressing the requestkey on the 1052 and during single program initiation
START	Initiates a foreground program or resumes batch processing.			X		After pressing request key on 1052
BATCH	Initiate batch processing.			X		
DLAB	Disk label information.				X	
DLBL	Disk label information.				X	

EXEC	Initiate single program execution				X	
EXTENT	Disk extent information.				X	
LBLTYP	Label information				X	During Single Program Initiation
READ	Specifies a card reader from which further single program initiation commands are read.				X	
TLBL	Tape label information.				X	
TPLAB	Tape label information.				X	
VOL	Disk volume information.				X	
XTENT	Disk extent information.				X	

1. Initial Program Loader (IPL).
2. Job Control (JC).
3. ATTN Routine (AR).
4. Single Program Initiation for F1 or F2.
5. Date and clock only.
6. Valid only if batch job foreground option was specified at system generation.
7. Valid only in a multiprogramming system.

LIBRARY MAINTENANCE FUNCTIONS  
V3 DIFFERENCES

Function	Unit	Element	Control Statements Required
Catalog	Core Image Library	Phase	Same as V2
	Relocatable Library	Module	// JOB jobname // EXEC MAINT CATALR modulename [,V.M[,C]] (module to be cataloged)  /* /&
	Source Statement Library	Book	// JOB jobname // EXEC MAINT CATALS sublib.bookname [,V.M] (book to be cataloged)

			/* /&	
Delete	Core Image Library	Phase	Same as V2	
		Program	Same as V2	
	Relocatable Library	Module	Same as V2	
		Program	Same as V2	
		Library	Same as V2	
Delete	Source Statement Library	Book	Same as V2	
		Sub-Library	Same as V2	
		Library	// JOB jobname // EXEC MAINT DELETS ALL /* /&	
Rename	Core Image Library	Phase	// JOB jobname // EXEC MAINT RENAMC oldname, newname(,oldname, newname,...) /* /&	
		Relocatable Library	Module	// JOB jobname // EXEC MAINT RENAMR oldname, newname(,oldname, newname,...) /* /&
			Book	// JOB jobname // EXEC MAINT RENAMS sublib. oldname,sublib. newname(,sublib. oldname,sublib. newname,...) /* /&
Update	Source Statement Library	Book	// JOB jobname // EXEC MAINT UPDATE sublib. bookname,[s.book1], [v.m],[nn] ) ADD, ) DEL, or ) REP statements as required with source statements to be changed ) END [v.m[,C]] /* /&	

Condense	Core Image Library	Library	// JOB jobname // EXEC MAINT CONDS CL /* /&
	Relocatable Library	Library	// JOB jobname // EXEC MAINT CONDS RL /* /&
	Source Statement Library	Library	// JOB jobname // EXEC MAINT CONDS SL /* /&
	Libraries	All	// JOB jobname // EXEC MAINT CONDS CL,RL,SL /* /&
Set Parameter for Automatic Condense	Libraries	Any or All	// JOB jobname // EXEC MAINT CONDL lib=nnnnn [,lib=nnnnn[, lib=nnnnn]] /* /&  Notes: Values to be substituted for <u>lib</u> : CL--Core Image library RL--Relocatable library SL--Source Statement Library Values to be substituted for <u>nnnnn</u> : One to five decimal digits, with a maximum value of 65536.
Reallocation	System	Library	// JOB jobname // DLBL IJSYSRS, 'DOS SYSTEM RESIDENCE FILE', date,code // EXTENT extent information // EXEC MAINT ALLOC id=cylin (tracks) (,id=cylin(tracks),...) /* /&



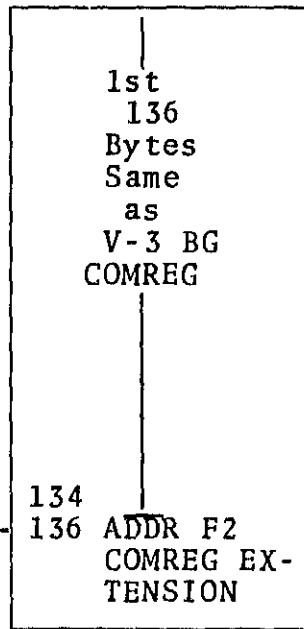
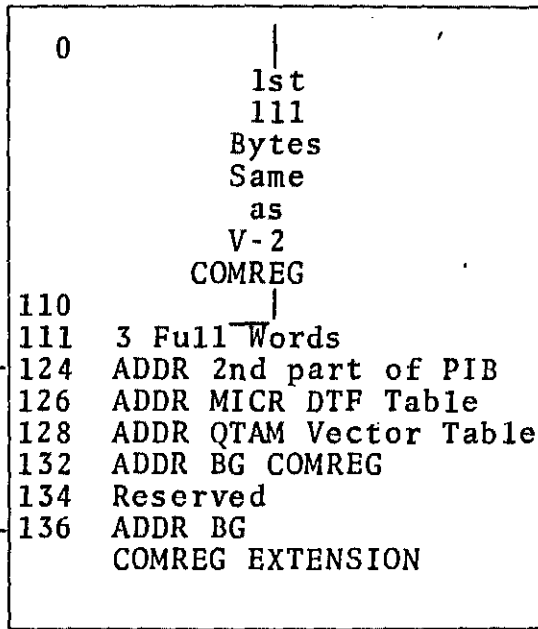
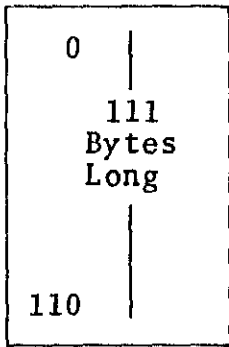
			<p>Notes: Values to be substituted for <u>id</u>:          CL--Core Image Library          RL--Relocatable Library          SL--Source statement library          Values to be substituted for <u>cylin</u> and <u>tracks</u>:          Any integer</p>
Copy	source Statement Library	Book	<pre>// JOB jobname // ASSGN SYS002,X'cuu' // DLBL IJSYSRS,'   DOS SYSTEM   RESIDENCE FILE',   date,code // EXTENT extent in-   formation // EXEC CORGZ   ALLOC id=cylin   (tracks) [,id=   cylin(tracks),...] * PRECEDING ALLOC * STATEMENT REQUIRED * IF NEW LIMITS TO BE * ESTABLISHED   COPYS sublib.book1   [,sublib.book2,...] /* / &amp;</pre>
		Sub- library	<pre>// JOB jobname // ASSGN SYS002,X'cuu' // DLBL IJSYSRS,'DOS   SYSTEM RESIDENCE   FILE',date,code // EXTENT extent in-   formation // EXEC CORGZ   ALLOC id=cylin   (tracks) [,id=   cylin(tracks),...] * PRECEDING ALLOC * STATEMENT REQUIRED * IF NEW LIMITS TO BE * ESTABLISHED   COPYS sublib1.ALL   [,sublib2.ALL,...] /* / &amp;</pre>
		Library	<pre>// JOB jobname // ASSGN SYS002,X'cuu' // DLBL IJSYSRS,'DOS   SYSTEM RESIDENCE   FILE',date,code // EXTENT extent   information // EXEC CORGZ   ALLOC id=cylin   (tracks) [,id=   cylin(tracks),...] * PRECEDING ALLOC * STATEMENT REQUIRED * IF NEW LIMITS TO BE</pre>

			<pre> * ESTABLISHED COPYS ALL /* / &amp; </pre>
	Libraries	All	<pre> // JOB jobname // ASSGN SYS002,X'cuu' // DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',date,code // EXTENT extent in- formation // EXEC CORGZ ALLOC id=cylin (tracks) [,id= cylin(tracks),...] * PRECEDING ALLOC * STATEMENT REQUIRED * IF NEW LIMITS TO BE * ESTABLISHED COPY ALL /* / &amp; </pre>

V-2 COMREG

V-3 BG COMREG

V-3 F2 COMREG



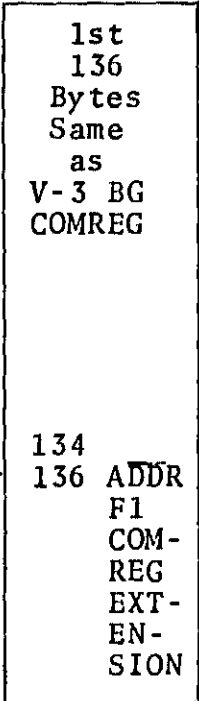
Minimum  
4-Byte  
BG EXTENSION

Minimum  
4-Byte  
F2 EXTENSION

ADDR CE TABLE

ADDR CE TABLE

V-3 F1 COMREG



2nd Part of PIB

Bytes	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
BG	ADDR	BG	SYS-	RESERVED												
		COM-	TEM													
		REG	LUB													
			IN-													
			DEX													
FG2	ADDR	FG2	FG2	RESERVED												
		COM-	SYS-													
		REG	LUB													
			IN-													
			DEX													
FG1	ADDR	FG1	FG1	RESERVED												
		COM-	SYS-													
		REG	LUB													
			IN-													
			DEX													
ATTN	ADDR	BG	00	RESERVED												
		COM-														
		REG														

Minimum  
4-Byte  
F1 EXTEN-  
SION

ADDR CE  
TABLE

Comparison of Communication Regions

PARTITION SAVE AREA

0	JOB NAME*		PSW		REG 9	REG A	23
24	REG B	REG C	REG D	REG E	REG F	REG 0	47
48	REG 1	REG 2	REG 3	REG 4	REG 5	REG 6	71
72	REG 7	REG 8	LENGTH RESER- LABEL VED AREA	PARTITION START TIME	FLTPT R0		95
96	FLTPT R2		FLTPT R4		FLTPT R6		119

\* NOT USED FOR BG

0 ----- Reset to zeros after IPL ----- 15								
14 Current Comm Region Address	18 External Old PSW	20 SVC Old PSW	28 Program Check Old PSW	30 Machine Check Old PSW	38 I/O Old PSW	40 CSW	48 CAW	4C BG Job Dura- tion
50 System Timer	54 System Time of Day	58 External New PSW	60 SVC New PSW	68 Program Check New PSW	70 Machine Check New PSW	78 I/O New PSW		
80 Diagnostic Scan-Out Area --- Model Dependent								
SUPERVISOR NUCLEUS								
General Cancel Routine								
General Exit Routine (Task Selection)								
Background Communications Region								
General Entry Routine								
Channel Scheduler								
Start I/O Routine								
I/O Interrupt								
Unit Check								
Error Recovery Exits								
Fetch Routine								
SVC, Program Check, and External Interrupts								
Resident Device Error Recovery								
Option Routines								
Supervisor constants, Error Recovery Block								
SVC Interrupt Table and PC, OC, and IT Tables								
Logical Transient Save Area								

Low Core

Nucleus Code

2nd Part of BG PIB	2nd Part of F2 PIB	2nd Part of F1 PIB	2nd Part of Attention PIB	All Bound PIB	1st Part of BG PIB	1st Part of F2 PIB	
1st Part of F1 PIB	1st Part of Attention PIB	Quiesce PIB	Supervisor PIB	Channel Queue	LUBID Table	REQID Table	
FOCL	PUB TABLE		FAVP		JIB TABLE		
TEB	FICL	NICL	LUB TABLE				F1 PROG
			BG SYS	BG PROG	F2 SYS	F2 PROG	
Foreground 2 Communications Region							
Foreground 1 Communications Region							
F2 Comm. Reg. Extension		F1 Comm. Reg. Extension		BG Comm. Reg. Extension			
SAB	PATCH AREA						
LOGICAL TRANSIENT AREA (B-TRANSIENTS) \$\$B							
PHYSICAL TRANSIENT AREA (A-TRANSIENTS) \$\$A							
CE Table	CE Area			BG Program Save Area			

I/O Tables and Information Blocks

Additional Comm. Regions

Logical and Physical Transients

PROBLEM PROGRAM AREA

Core Map



# **IBM** / **FE Supplement**

**System/Unit Programming Handbook**

**Re: Form No. 229-2129-0**

**This Supplement No. S29-0029**

**Date: November 1, 1968**

**This Supplement is issued to release information on DOS Version 3 differences. File the attached in the Field Engineering Programming Handbook, Form Number 229-2129-0.**

**File this cover letter at the back of the publication. It will then serve as a record of the changes received and incorporated.**





**IBM** Programming Handbook  
DOS  
CE Serviceability Programs



## CE SERVICEABILITY PROGRAMS

### DOS CE SERVICEABILITY PROGRAMS GENERAL INFORMATION:

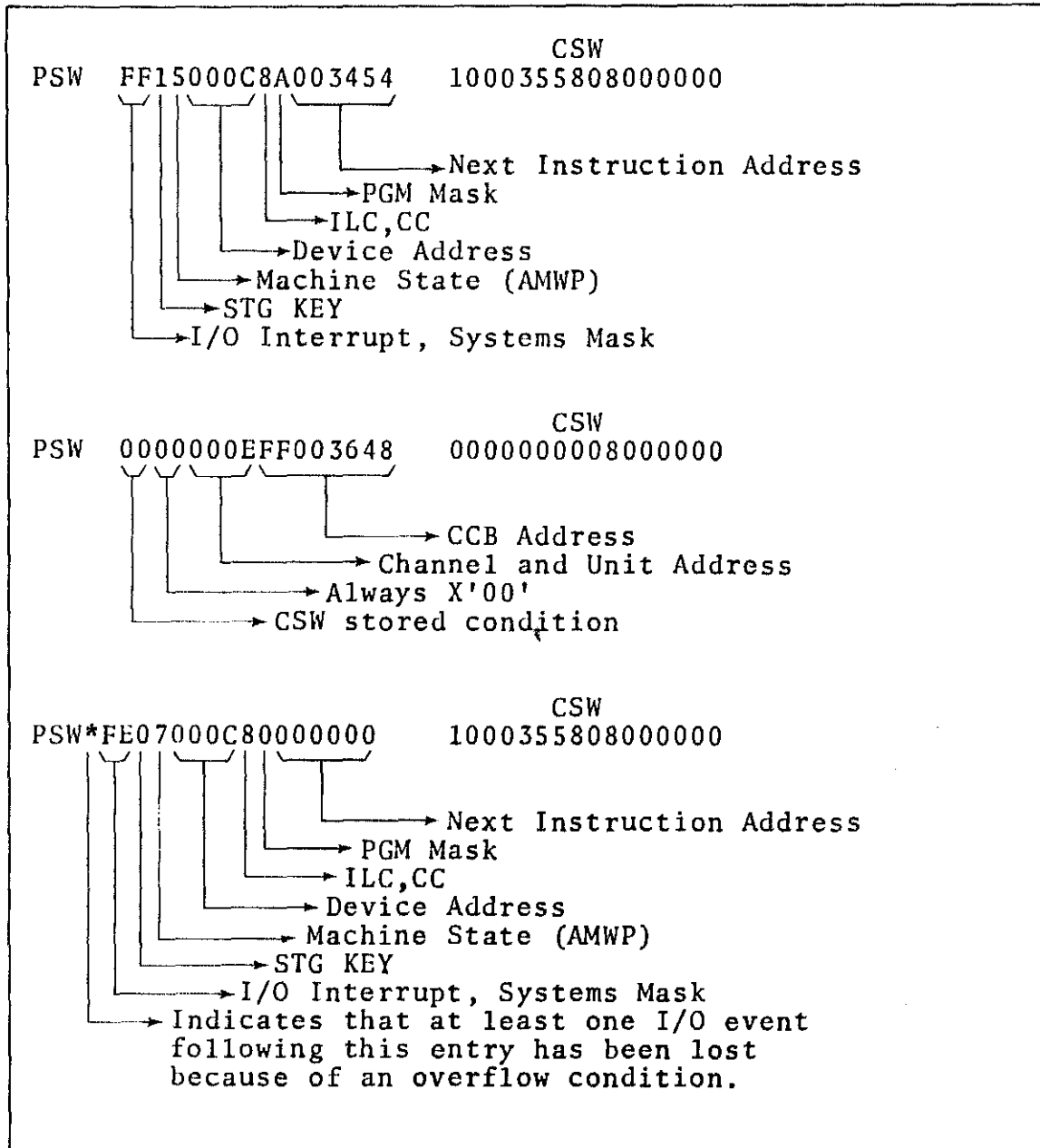
The SERVICEABILITY PROGRAMS can now be included in DOS by using the FOPT parameter CE=YES, in the Supervisor. These programs can be used by Customer Engineers, System Engineers and programmers as a troubleshooting tool simply by requesting them on SYSLOG. A complete outline on how to use these programs can be found in the PLM IBM System/360 DISK OPERATING SYSTEM (Version 3) CE SERVICEABILITY PROGRAMS (Z24-5091).

The Serviceability Program available in Release 16 is the I/O TRACE FUNCTION. In Release 17, FETCH/LOAD TRACE and TRANSIENT DUMP were added.

Release 19 will have two additional Serviceability Programs. They are DYNAMIC DISPLAY FUNCTION and DYNAMIC PATCH FUNCTION.

The following is a brief description of the Serviceability Programs and a summary on how to use them.

I/O TRACE FUNCTION - Provides a trace table for input/output devices. It records the I/O old PSW and CSW when an I/O interruption occurs. It records the device address, the CCB address and the CSW when the CSW is stored, in response to a SIO instruction issued by the DOS supervisor.



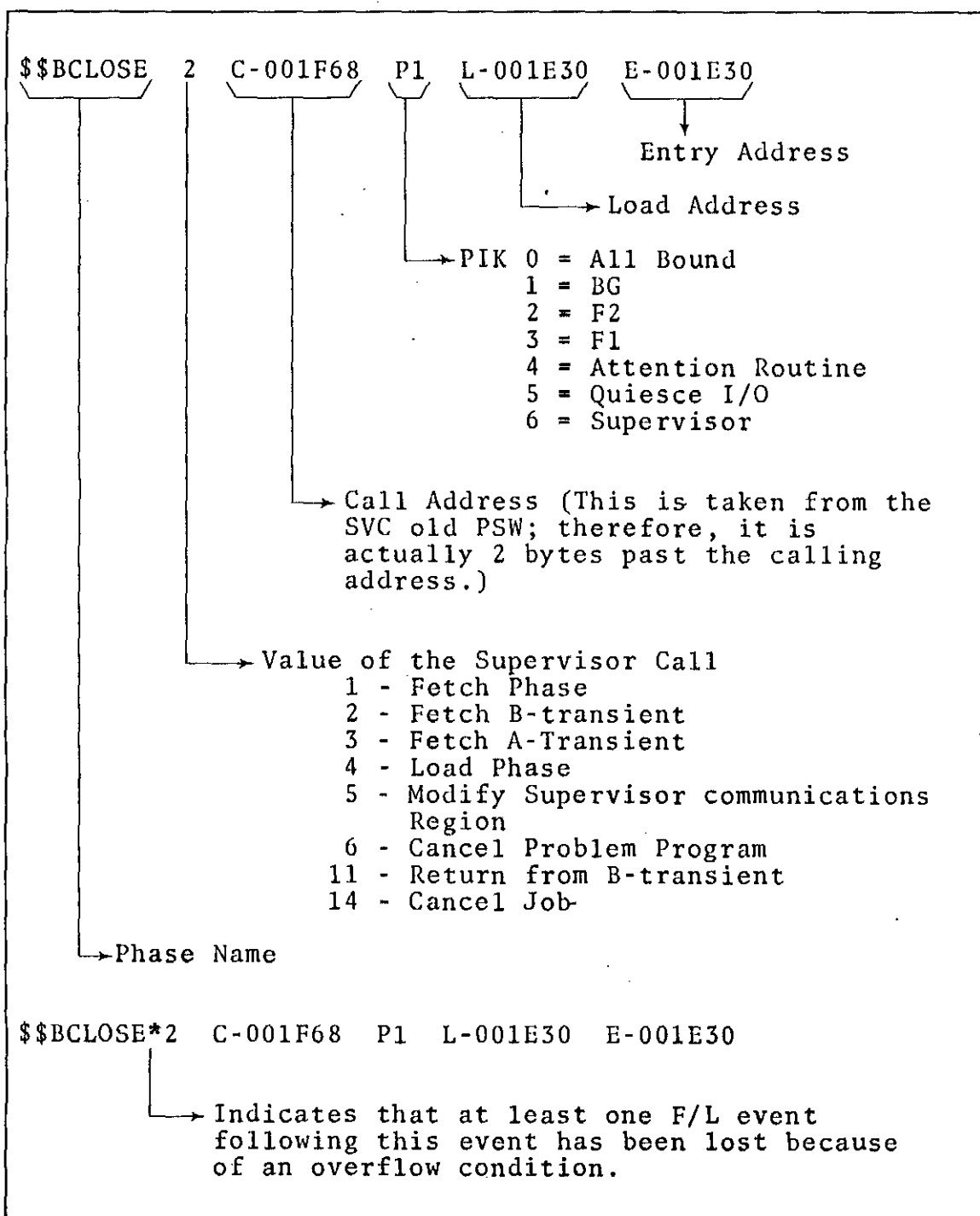
Sample Output for I/O Trace

FETCH/LOAD TRACE FUNCTION - allows tracing the order in which phases and transients are executed under control of DOS. Tracing consists of recording (for SVC1, SVC2, SVC3, SVC4):

- . The location of the supervisor call
- . The program interrupt key
- . The supervisor call number
- . The name of the phase or transient being called.
- . The load address of the phase
- . The entry address of the phase.

NOTE: At times, SVC5, 6, 11 and 14 branch directly into the supervisor fetch or load routine. These are traced whenever they occur, and appear in the output of the trace. (For multiprogramming systems, the calling address and SVC value cannot be guaranteed accurate in such cases.)

Each collection of data is referred to as a F/L event. The events may be preserved in a rotating buffer (first entry overwritten when the area is full), or may be outputted on a printer, punch, tape unit or console printer. When punch or tape is used, the cards or tape must be processed by the CEIDLST utility program to provide readable output data.



Sample Output for F/L Trace

TRANSIENT DUMP FUNTION provides a dump of the supervisor, before it is altered, on a program check interruption. The dump provides:

- . The 16 general registers.
- . The first 144 bytes of low core.
- . The logical transient area (with the label LTA).
- . The physical transient area (with the label PTA).

This information may be outputed on either a printer or tape unit. When tape is used, the tape must be processed by the CEIDLST utility program to provide readable output data.

Both the printer and tape modules are reusable. That is, a dump is given on each program check interruption until the function is reset.

DYNAMIC DISPLAY FUNCTION - will allow the general registers and a specified area of core storage to be displayed each time a designated location (trigger address) is reached in the execution of the program.

The User specifies the parameters to determine the display. Required:

- . The location of the instruction that triggers the display (trigger address).
- . The op-code expected at the trigger address (trigger instruction).
- . An output device address (to be used for the displaying).
- . The name of the phase to trigger the display.

Optional:

- . The partition, in which the monitored phase is to be loaded in a multiprogramming system.
- . The area of core storage to be displayed (in addition to the general registers).

NOTE: AVAILABLE IN RELEASE 19 AND LATER.

DYNAMIC PATCH FUNCTION - will allow the user to dynamically patch desired areas of core. To perform this, the user specifies:

- . The name of the phase that is to trigger the patch.
- . The partition (optional) into which the phase is to be loaded.
- . Start address where the patch is to be inserted.
- . The data (patch) to be inserted.

The patch information cannot exceed 64K bytes.

NOTE: AVAILABLE IN RELEASE 19 AND LATER.

## OPERATING REQUIREMENTS AND PROCEDURES

### EQUIPMENT REQUIREMENTS

The CE Aid option requires at least 24K bytes of main storage. When tape, printer, or punch output is used, it is recommended that the output device be dedicated to the CE Aid function being executed and not share a selector channel or a control unit with other programs being executed. Although it is possible to share devices, this may result in lost events, and should be avoided.

When 7-track tape is used, the data conversion feature is required, and the unit must be in system reset mode while the module (IOTRTAPE, FLTRTAPE, and TDMPTAPE) is operating.

The CE Aid functions support existing DOS hardware and require no new hardware; however, the console feature must be present. The modules using a punch for output do not support the 1442 unless the 1442 is used as a punch only.

### PROGRAM REQUIREMENTS

The CE Aid option require a DOS supervisor (Version 3) in which the FOPT macro option CE=YES, or CE=nnnnn (nnnnn is between 600 and 10,240) has been specified. The SYSGEN option associated with the DOS supervisor provides:

- . A CE area (600 bytes minimum) in the supervisor.
- . Exit points from certain supervisor routines
- . A table of addresses (CE area address, exit point addresses, etc)
- . A pointer to the CE table in the communication region extension
- . A low core pointer to the CETABLE (used by exit points).



The core image and relocatable libraries must contain the following phases.

Core Image	Relocatable
CEAID	ILCEAID1
CEAID001	
CEAID002	
CEAID003	
CEAID004	
CEAID005*	
CEAID006*	
CEAIDLST	ILCEAID2
\$\$BZCE01	

\* Release 19

#### CATALOGING CEAID AND CEAIDLST

All CEAID modules are written as self-relocating code to allow initialization in any partition of a multiprogramming system. Some differences exist in job control programs between batch-only and multiprogramming systems. These differences affect self-relocating code.

For multiprogramming systems, self-relocating programs must be assigned an origin of location zero when they are linkage edited. The zero address allows the supervisor to recognize self-relocating program, and the program can be loaded into the appropriate location for the partition being initialized.

Batch-only systems cannot recognize self-relocating programs, and they require all programs to be linkage edited to a valid location beyond the end of the supervisor.

The root-phases CEAID and CEAIDLST (modules ILCEAID1 and ILCEAID2, respectively) are provided in the relocatable library so they can be linkage edited for the system in which they are to be used. All other phases associated with the CEAID functions are loaded by the root-phase CEAID, and because they do not require linkage editing, they are provided only in the core image library.

Before using the functions, perform a linkage editor run and catalog CEAID and CEAIDLST into the core image library. Cataloging may be performed by entering one of the following sets of cards into the DOS job stream:

1. For operation in a batch-job system:

```
// JOB CE1
// OPTION CATAL
// PHASE CEAID,S
// INCLUDE ILCEAID1
/*
// EXEC LNKEDT
/*
/&
// JOB CE2
// OPTION CATAL
// PHASE CEAIDLST,S
// INCLUDE ILCEAID2
/*
// EXEC LNKEDT
/*
/&
```

2. For operation in a multiprogramming system:

```
// JOB CE1
// OPTION CATAL
// PHASE CEAID,+0
// INCLUDE ILCEAID1
/*
// EXEC LNKEDT
/*
/&
// JOB CE2
// OPTION CATAL
// PHASE CEAIDLST,+0
// INCLUDE ILCEAID2
/*
// EXEC LNKEDT
/*
/&
```

## INITIALIZATION

After linkage editing into the core image library, the initializer program is loaded with standard DOS job control statements. The statement:

```
// EXEC CEAID
```

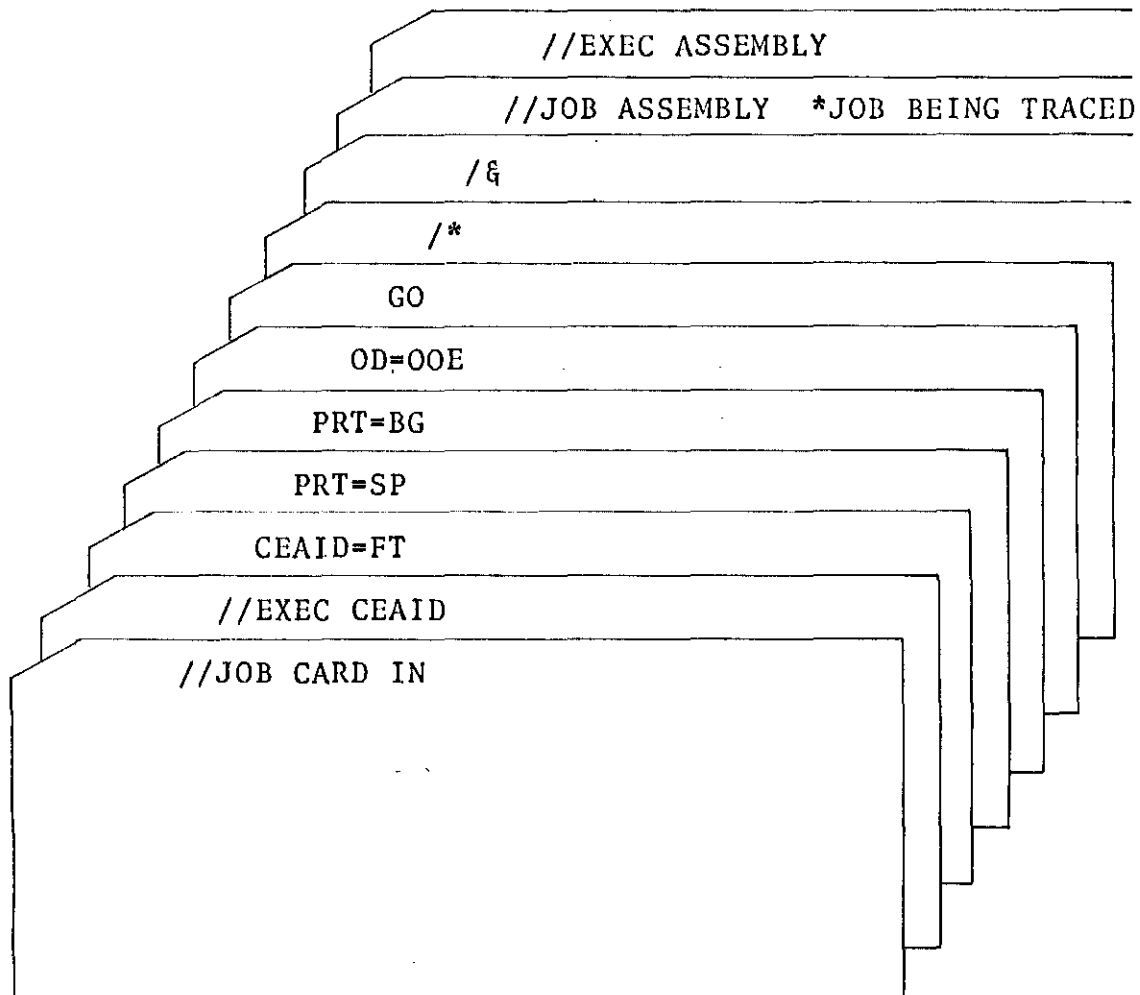
causes the initializer to be loaded. The initializer types out the first keyword, CEAID= and opens the keyboard for a response. A choice can be made at this point to enter the parameters from the console typewriter or from SYSRDR via cards. The following is a description of each of these methods.

### INITIALIZATION VIA CARD

1. Punch desired keywords and parameters into cards. Entries may be punched one-per-card, or multiple entries (separated by commas) can be made in a single card. An entry may not be split between two cards. All 80 columns of a card may be used, but a card is terminated by the first blank following an entry, or a GO entry. The last entry of the last card must be GO and the last card must be followed by a /\* card.
2. Place keyword entry card(s) in SYSRDR.
3. Respond to the console message CEAID= by entering only an end-of-block (B) character (alternate code 5).
4. The initializer reads from cards until the GO entry is encountered.

NOTE: If a bad parameter occurs on a card, corrections are request on SYSLOG.

SAMPLE  
CEAID CARD INPUT STREAM  
FOR FETCH/LOAD TRACE



### INITIALIZATION VIA CONSOLE TYPEWRITER

1. Respond to the console message CEAID=-- by entering IT (B), FT (B), TD (B), DD (B), or DP (B).
2. The initializer responds with the next keyword OD=CUU.

NOTE: This keyword is not issued for dynamic patch.

- a. Respond with the (B) character to select core-wrap mode.

NOTE: This response is invalid for transient dump and dynamic display.

- b. Respond with a channel and unit address (CUU or X'CUU') (B), to select an output device.
- c. Respond with GO (B). This automatically terminates input, and selects defaults for all remaining options.

NOTE: This response is invalid for transient dump, dynamic display and dynamic patch.

3. The initializer continues to "prompt" with appropriate keywords until all options have been satisfied, or a GO (B) response is received, whereupon defaults are selected for any remaining options. (B) character, in response to any keyword "prompt", causes selection of the default option, and causes the initializer to respond with the next option available. Response to IGN= or TRC= with a valid CUU causes the keyword to be repeated until:
  - a. Three entries have been made, or
  - b. (B) response alone is given, or
  - c. GO (B) is entered.
4. The sequence in which keywords are prompted depends on previous responses. Because AAA is valid only in core-wrap mode, it does not occur unless the response to OD is (B) (which selects core-wrap mode by default). Because IGN and TRC are mutually exclusive TRC is prompted only after a default response to IGN.

SAMPLE  
CONSOLE TYPEWRITER INPUT  
FOR FETCH/LOAD TRACE

//JOB TYPINPT3

//EXEC CEAIID (Calls for Initializer)  
CEAIID --- (Console Requests Function)  
FT (B) (CE Response: F/L Trace Function)  
OD = CUU (Console Requests Output Device)  
(B) (CE Response: Core Wrap)  
AAA=X'hhhh',X'hhhh' (Console Request Alternate Area)  
(B) (CE Response: No AAA)  
PRT --- (Console Requests Partition to be Traced)  
BG (B) (CE Response: Background)  
PRT --- (Console Requests Second Partition)  
GO (B) (CE Response: End of Input)

Keywords for IT, FT, and TD.

The CE Aid program uses keywords to initialize the program and provide parameters. When initializing by card, the keywords may be placed in any column, and need not be in any specific order. When initiating by console, requests are made as needed. The following charts explain the keywords and their meanings.

Keyword	Parameter	Meaning	Default
CEAID=	IT FT TD XX	Initiate I/O trace. Initiate F/L trace. Initiate transient dump. Terminate function. (Turn off all hooks.)	-- -- -- Function continues.
OD= (Note 4)	CUU or X'CUU'	Use specified output device for output of trace function.	Core-wrap mode.
AAA=	x'hhhh',x'hhhh' or x'hhhhhh',x'hhhhhh'	Use specified (lower limit address, upper limit address) alternate area addresses for core-wrap trace tables. (The addresses specified cannot be in the supervisor or beyond the machine core limits and must contain either 4 6 numerals.)	Keep trace table in CE area.
PRT= (Note 1)	SP BG F2 F1	Trace supervisor. Trace background partition. Trace foreground 2 partition. Trace foreground 1 partition.	Trace all partitions.

Keyword	Parameter	Meaning	Default
IGN= (Note 2)	CUU or X'CUU'	Ignore I/O activity of specified device. A maximum of three IGN=CUU entries are allowed.	Trace all I/O.
TRC= (Note 2)	CUU or X'CUU'	Trace only device specified. A maximum of three TRC=CUU entries are allowed.	Trace all I/O.
GO (Note 3)		End of initializer keyword entries	--

Note 1 : This keyword is valid only for multiprogramming systems. It applies only to F/L trace.

Note 2 : These keywords apply only to I/O trace.

Note 3 : GO is an invalid response to a request for a console correction to card input.

Note 4 : An output device must be specified for transient dump.

Keywords for IT, FT, and TD.



Keywords for Dynamic Display.

KEYWORD	PARAMETER	MEANING	DEFAULT
CEAID=	DD XX	Initiate dynamic display. Terminate function. (Turn off all hooks. If present, re- place trigger with the trigger instruction.)	-- Function continues
OD= (Note 2)	CUU or X'CUU'	Use specified output device for output of display func- tion.	--
PHASE= (Note 2)	NNNNNNNN (1-8) byte name field	Insert the display trigger when the specified phase is loaded.	--
	*	Instert the display trigger at the trigger address regard- less of which phase is in core.	--
PRT= (Note 3)	SP  BG  F2  F1	Display the trigger when the phase is loaded into the supervisor. Display the trigger when the phase is loaded into the background partition. Display the trigger when the phase is loaded into the foreground 2 partition. Display the trigger when the phase is loaded into the foreground 1 partition.	Display the trigger whenever phase is loaded, regardless of partition.

Keywords for Dynamic Display

KEYWORD	PARAMETER	MEANING	DEFAULT
DISPLIMS=	X'HHHHHH',X'HHHHHH' or - X'HHHHHH',LDDDDDDD or RDD,LDDDDDDD	Display the general registers and the specified area of core (lower limit address, upper limit address; lower limit address, length of display area; register containing lower limit address, length of display area). The addresses cannot be beyond the machine core limits. The display area must be a fullword multiple of bytes. The address parameter must contain 1-6 hex numerals; the length parameter must contain 1-7 decimal numerals; the register parameter must contain one of two decimal numerals, (registers 9,10,11,12 are invalid because they are used by the function at display time.)	Display only the general registers.
OP-CODE= (Note 2)	HH	Op code of the instruction at the trigger address (must be in hex).	--
TRIGADDR= (Note 2)	X'HHHH'	Address of instruction that triggers the display (must be in hex).	--
GO (Note 1)		End of initializer keyword entries.	--

- Note 1 : GO is an invalid response to a request for a console correction to card input.
- Note 2 : This keyword must be specified for dynamic display (printer & tape output are supported).
- Note 3 : This keyword is valid only for multiprogramming systems, and must be used if the monitored phase is to be loaded into two different partitions while the display function is active. The keyword is not specified when an '\*' phase is specified.

Keywords for Dynamic Display.

KEYWORD	PARAMETER	MEANING	DEFAULT
CEAID=	DP XX	Initiate dynamic patch. Terminate function. (Turn off all hooks.)	-- Function Continues
AAA=	X'HHHHHH',X'HHHHHH' or X'HHHHHH',LDDDDDD	Use specified (lower limit address, upper limit address; or lower limit address, length of alternate area) alternate area addresses for dynamic patch table. (The addresses specified cannot be in the supervisor or beyond the machine core limits. The address parameters must contain 1-6 hex numerals; the length parameter must contain 1-7 decimal numerals)	Keep patch table in CE area.
PHASE= (Note 6)	NNNNNNNN (1-8 byte name field)  * (Note 1)	Insert patch when the specified phase is loaded.  Insert the patch at EOJ regardless of which phase is in core.	--
PRT= (Note 7)	SP  BG	Insert patch when the phase is loaded into the supervisor. Insert patch when the phase is loaded into the background partition.	Insert patch whenever the phase is loaded, regardless of partition.

Keywords for Dynamic Patch

KEYWORD	PARAMETER	MEANING	DEFAULT
	F2	Insert patch when the phase is loaded into the foreground 1 partition.	
	F1	Insert patch when the phase is loaded into the foreground 1 partition.	
	SIZE (Note 3)	The size of the remaining patch area is put out on the console and the PRT= keyword is reissued.	Continue processing
	RESET (Note 3)	The system is reset and the PHASE keyword is reissued.	
DPADDR= (Note 1)	X'HHHH'	Address where patch is to be inserted (must be 1-6 hex numerals).	--
	SIZE (Note 3)	The size of the remaining patch area is issued and the DPADDR keyword is reissued.	Continue processing
	RESET (Note 3)	The system is reset and the PHASE= keyword is reissued.	
DATA= (Note 1)	2-74 hex numerals 1-37 bytes of data	Patch information to be inserted, starting at DPADDR.	--

Keywords for Dynamic Patch

KEYWORD	PARAMETER	MEANING	DEFAULT
	SIZE (Note 3)	The size of the remaining patch area is issued and the DATA keyword is re-issued.	Continue processing
	RESET (Note 3)	The system is reset and the PHASE keyword is re-issued.	
DPADDR= (Note 4)	X'HHHH'  *	New patch address requested by initializer when console input is used -- if specified the PRT, DPADDR, and DATA parameters are re-requested.	PHASE keyword is repeated.
GO (Note 5)		End of initializer keyword entries.	--
<p>Note 1 : This keyword must be specified for dynamic patch.</p> <p>Note 2 : This keyword is valid only for multiprogramming systems.</p> <p>Note 3 : This parameter is valid only for dynamic patch.</p> <p>Note 4 : This keyword is repeated until the B response is given.</p> <p>Note 5 : GO is an invalid response for a console correction to card input.</p> <p>Note 6 : If '*' is specified, the Initializer inserts the patch (DPCHWRAP is not called)</p> <p>Note 7 : If '*' is specified, this keyword is not specified.</p>			

Keywords for Dynamic Patch

Sample - trace I/O events from three specified devices, using printer output:

```
// JOB TYPINPT 2
// EXEC CEAID          Calls for initializer
CEAID=- -             Console request function
IT (B)               CE response: I/O trace
                    function
OD=CUU               Console requests output
                    device
00E (B)              CE response: Printer output
IGN=CUU              Console requests IGN para-
                    meters
(B)                  CE response: No devices to
                    be ignored.
TRC=CUU              Console requests and CE
                    specifies devices to be
                    traced
180 (B)
TRC=CUU
090 (B)
TRC=CUU
01F (B)
```

Note: GO does not have to be specified here. The initializer knows this is the end of input because three TRC entries have been made.

#### TERMINATION

Any function can be terminated by recalling the initiator in the same fashion as before (// EXEC CEAID), and responding to the CEAID= keyword with XX(B). It is also possible to reset one function by loading another because the \$\$BZCE01 transient automatically resets all supervisor hooks before loading a function into the CE area.

CARD/TAPE-TO-PRINTER UTILITY PROGRAM (CEAIDLST)

The card/tape-to-printer utility program reads the tape (or card) input created by the tape or punch modules from SYS005, and then prints the information on SYSLST in the same format that is used by the print modules.

Input on tape or cards may contain output from a series of CEAID runs. Only one run of the CEAIDLST program is necessary to handle any combination of one output mode from CE serviceability programs. If an unrecognizable record is encountered in processing input data, a hex dump of that record (with a maximum of 256 bytes) is given.

When card input is used, the data cards must be followed by a /\* card. Failure to do so results in any following cards being formatted as data until a /\* is encountered.

Caution: When tape is used, always reset (terminate) the function before executing CEAIDLST. If this is not done, the CEAIDLST input tape is used by the active function for output.



**IBM Programming Handbook**  
**TOS/DOS**



## STANDARD INSTRUCTION SET

NAME	MNEMONIC	TYPE	OPERAND	CODE
Add	AR	RR	R1, R2	1A
Add	A	RX	R1, D2(X2, B2)	5A
Add Halfword	AH	RX	R1, D2(X2, B2)	4A
Add Logical	ALR	RR	R1, R2	1E
Add Logical	AL	RX	R1, D2(X2, B2)	5E
AND	NR	RR	R1, R2	14
AND	N	RX	R1, D2(X2, B2)	54
AND	NI	SI	D1(B1), I2	94
AND	NC	SS	D1(L, B1), D2(B2)	D4
Branch and Link	BALR	RR	R1, R2	05
Branch and Link	BAL	RX	R1, D2(X2, B2)	45
Branch on Condition	BCR	RR	M1, R2	07
Branch on Condition	BC	RX	M1, D2(X2, B2)	47
Branch on Count	BCTR	RR	R1, R2	06
Branch on Count	BCT	RX	R1, D2(X2, B2)	46
Branch on Index High	BXH	RS	R1, R3D2(B2)	86
Branch on Index Low or Equal	BXLE	RS	R1, R3, D2(B2)	87
Compare	CR	RR	R1, R2	19
Compare	C	RX	R1, D2(X2, B2)	59
Compare Halfword	CH	RX	R1, D2(X2, B2)	49
Compare Logical	CLR	RR	R1, R2	15
Compare Logical	CL	RX	R1, D2(X2, B2)	55
Compare Logical	CLC	SS	D1(L, B1), D2(B2)	D5
Compare Logical	CLI	SI	D1(B1), I2	95
Convert to Binary	CVB	RX	R1, D2(X2, B2)	4F
Convert to Decimal	CVD	RX	R1, D2(X2, B2)	4E
Diagnose		SI		83
Divide	DR	RR	R1, R2	1D
Divide	D	RX	R1, D2(X2, B2)	5D
Exclusive OR	XR	RR	R1, R2	17
Exclusive OR	X	RX	R1, D2(X2, B2)	57
Exclusive OR	XI	SI	D1(B1), I2	97
Exclusive OR	XC	SS	D1(L, B1), D2(B2)	D7
Execute	EX	RX	R1, D2(X2, B2)	44
Halt I/O	HIO	SI	D1(B1)	9E
Insert Character	IC	RX	R1, D2(X2, B2)	43
Load	LR	RR	R1, R2	18
Load	L	RX	R1, D2(X2, B2)	58
Load Address	LA	RX	R1, D2(X2, B2)	41
Load and Test	LTR	RR	R1, R2	12
Load Complement	LCR	RR	R1, R2	13
Load Halfword	LH	RX	R1, D2(X2, B2)	48
Load Multiple	LM	RS	R1, R3, D2(B2)	98
Load Negative	LNR	RR	R1, R2	11
Load Positive	LPR	RR	R1, R2	10
Load PSW	LPSW	SI	D1(B1)	82
Move	MVI	SI	D1(B1), I2	92
Move	MVC	SS	D1(L, B1), D2(B2)	D2
Move Numerics	MVN	SS	D1(L, B1), D2(B2)	D1
Move with Offset	MVO	SS	D1(L1, B1), D2(L2, B2)	F1
Move Zones	MVZ	SS	D1(L, B1), D2(B2)	D3
Multiply	MR	RR	R1, R2	1C
Multiply	M	RX	R1, D2(X2, B2)	5C
Multiply Halfword	MH	RX	R1, D2(X2, B2)	4C
OR	OR	RR	R1, R2	16
OR	O	KX	R1, D2(X2, B2)	56
OR	OI	SI	D1(B1), I2	96
OR	OC	SS	D1(L, B1), D2(B2)	D6
Pack	PACK	SS	D1(L1, B1), D2(L2, B2)	F2
Set Program Mask	SPM	RR	R1	04
Set System Mask	SSM	SI	D1(B1)	80
Shift Left Double	SLDA	RS	R1, D2(B2)	8F
Shift Left Single	SLA	RS	R1, D2(B2)	8B
Shift Left Double Logical	SLDL	RS	R1, D2(B2)	8D
Shift Left Single Logical	SLL	RS	R1, D2(B2)	89
Shift Right Double	SRDA	RS	R1, D2(B2)	8E
Shift Right Single	SRA	RS	R1, D2(B2)	8A
Shift Right Double Logical	SRDL	RS	R1, D2(B2)	8C
Shift Right Single Logical	SRL	RS	R1, D2(B2)	88
Start I/O	SIO	SI	D1(B1)	9C
Store	ST	RX	R1, D2(X2, B2)	50
Store Character	STC	RX	R1, D2(X2, B2)	42
Store Halfword	STH	RX	R1, D2(X2, B2)	40
Store Multiple	STM	RS	R1, R3, D2(B2)	90
Subtract	SR	RR	R1, R2	1B
Subtract	S	RX	R1, D2(X2, B2)	5B
Subtract Halfword	SH	RX	R1, D2(X2, B2)	4B
Subtract Logical	SLR	RR	R1, R2	1F
Subtract Logical	SL	RX	R1, D2(X2, B2)	5F

STANDARD INSTRUCTION SET

(Continued)

Supervisor Call	SVC	RR	I	0A
Test and Set	TS	SI	D1(B1)	93
Test Channel	TCH	SI	D1(B1)	9F
Test I/O	TIO	SI	D1(B1)	9D
Test Under Mask	TM	SI	D1(B1), I2	91
Translate	TR	SS	D1(L, B1), D2(B2)	DC
Translate and Test	TRT	SS	D1(L, B1), D2(B2)	DD
Unpack	UNPK	SS	D1(L1, B1), D2(L2, B2)	F3

DECIMAL FEATURE INSTRUCTIONS

Add Decimal	AP	SS	D1(L1, B1), D2(L2, B2)	FA
Compare Decimal	CP	SS	D1(L1, B1), D2(L2, B2)	F9
Divide Decimal	DP	SS	D1(L1, B1), D2(L2, B2)	FD
Edit	ED	SS	D1(L, B1), D2(B2)	DE
Edit and Mark	EDMK	SS	D1(L, B1), D2(B2)	DF
Multiply Decimal	MP	SS	D1(L1, B1), D2(L2, B2)	FC
Subtract Decimal	SP	SS	D1(L1, B1), D2(L2, B2)	FB
Zero and Add	ZAP	SS	D1(L1, B1), D2(L2, B2)	F8

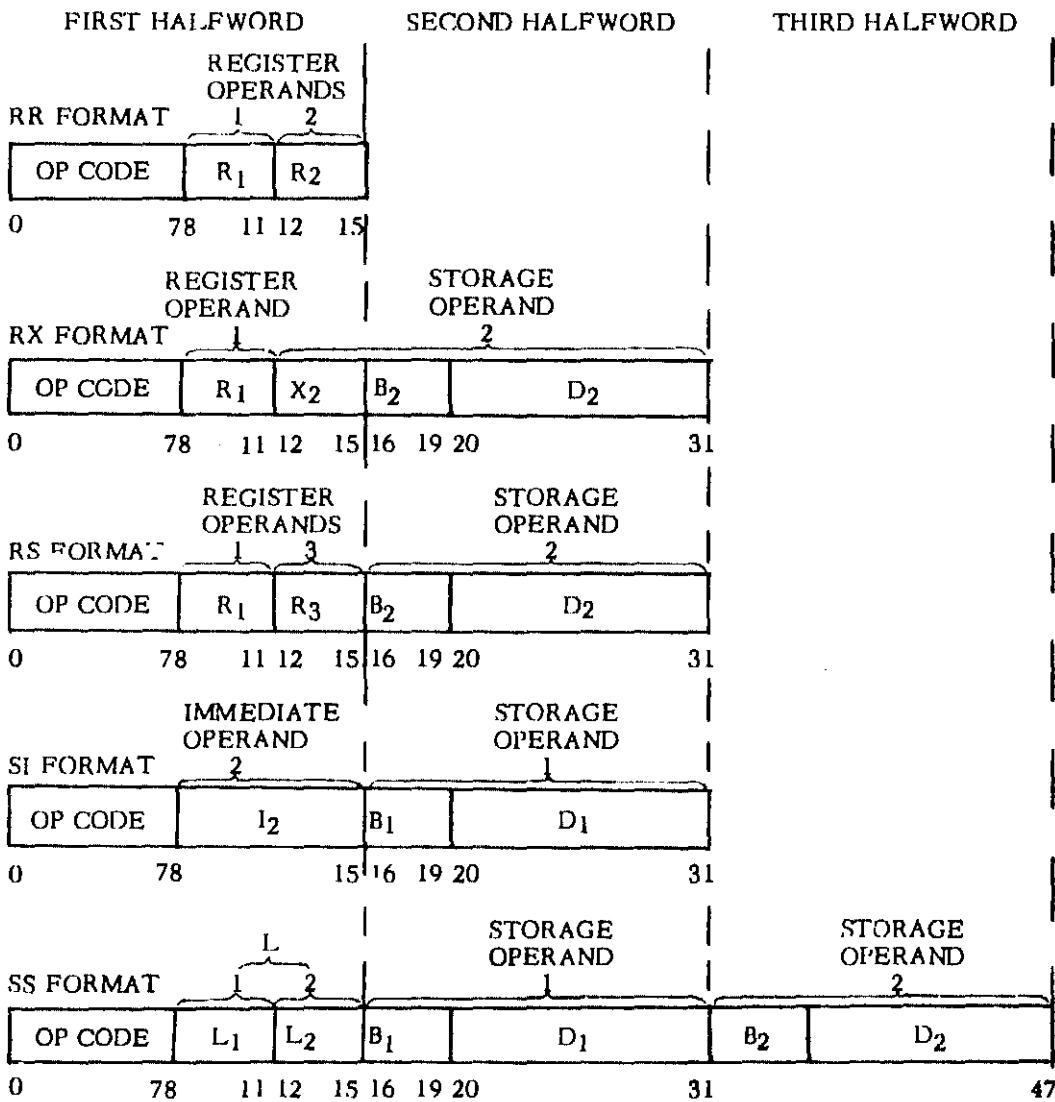
DIRECT CONTROL FEATURE INSTRUCTIONS

Rear Direct	RDD	SI	D1(B1), I2	85
Write Direct	WRD	SI	D1(B1), I2	84

PROTECTION FEATURE INSTRUCTIONS

Insert Storage Key	ISK	RR	R1, R2	09
Set Storage Key	SSK	RR	R1, R2	08

BASIC INSTRUCTION FORMATS



FLOATING-POINT FEATURE INSTRUCTIONS

Add Normalized (Long)	ADR	RR	R1, R2	2A
Add Normalized (Long)	AD	RX	R1, D2(X2, B2)	6A
Add Normalized (Short)	AER	RR	R1, R2	3A
Add Normalized (Short)	AE	RX	R1, D2(X2, B2)	7A
Add Unnormalized (Long)	AWR	RR	R1, R2	2E
Add Unnormalized (Long)	AW	RX	R1, D2(X2, B2)	6E
Add Unnormalized (Short)	AUR	RR	R1, R2	3E
Add Unnormalized (Short)	AU	RX	R2, D2(X2, B2)	7E
Compare (Long)	CDR	RR	R1, R2	29
Compare (Long)	CD	RX	R1, D2(X2, B2)	69
Compare (Short)	CER	RR	R1, R2	39
Compare (Short)	CE	RX	R1, D2(X2, B2)	79
Divide (Long)	DDR	RR	R1, R2	2D
Divide (Long)	DD	RX	R1, D2(X2, B2)	6D
Divide (Short)	DER	RR	R1, R2	3D
Divide (Short)	DE	RX	R1, D2(X2, B2)	7D
Halve (Long)	HDR	RR	R1, R2	24
Halve (Short)	HER	RR	R1, R2	34
Load and Test (Long)	LTDR	RR	R1, R2	22
Load and Test (Short)	LTER	RR	R1, R2	32
Load Complement (Long)	LCDR	RR	R1, R2	23
Load Complement (Short)	LCER	RR	R1, R2	33
Load (Long)	LDR	RR	R1, R2	28
Load (Long)	LD	RX	R1, D2(X2, B2)	68
Load Negative (Long)	LNDR	RR	R1, R2	21
Load Negative (Short)	LNER	RR	R1, R2	31
Load Positive (Long)	LPDR	RR	R1, R2	20
Load Positive (Short)	LPER	RR	R1, R2	30
Load (Short)	LER	RR	R1, R2	38
Load (Short)	LE	RX	R1, D2(X2, B2)	78
Multiply (Long)	MDR	RR	R1, R2	2C
Multiply (Long)	MD	RX	R1, D2(X2, B2)	6C
Multiply (Short)	MER	RR	R1, R2	3C
Multiply (Short)	ME	RX	R1, D2(X2, B2)	7C
Store (Long)	STD	RX	R1, D2(X2, B2)	60
Store (Short)	STE	RX	R1, D2(X2, B2)	70
Subtract Normalized (Long)	SDR	RR	R1, R2	2B
Subtract Normalized (Long)	SD	RX	R1, D2(X2, B2)	6B
Subtract Normalized (Short)	SER	RR	R1, R2	3B
Subtract Normalized (Short)	SE	RX	R1, D2(X2, B2)	7B
Subtract Unnormalized (Long)	SWR	RR	R1, R2	2F
Subtract Unnormalized (Long)	SW	RX	R1, D2(X2, B2)	6F
Subtract Unnormalized (Short)	SUR	RR	R1, R2	3F
Subtract Unnormalized (Short)	SU	RX	R1, D2(X2, B2)	7F

CHARACTERISTICS FOR CONSTANTS

Code	Type	Machine Format
C	Character	8-Bit Code for each Character
X	Hexadecimal	4-Bit Code for each Hexadecimal Digit
B	Binary	Binary Digits (ones and zeros)
F	Fixed-point	Signed, Fixed-point Binary Format; Normally a Full Word
H	Fixed-point	Signed, Fixed-point Binary Format; Normally a Half Word
E	Floating-point	Short Floating-point Format; Normally a Full Word
D	Floating-point	Long Floating-point Format; Normally a Double Word
P	Decimal	Packed Decimal Format
Z	Decimal	Zoned Decimal Format
A	Address	Value of Address; Normally a Full Word
V	Address	Space Reserved for External Symbol Addresses; Each Address Normally a Full Word
S	Address	Address in Base Displacement Form
Y	Address	Value of Address; Normally a Half Word

EXTENDED MNEMONIC INSTRUCTION CODES

GENERAL

	EXTENDED CODE	MACHINE INSTRUCTION	MEANING
B	D2(X2, B2)	BC 15, D2(X2, B2)	Branch Unconditionally
BR	R2	BCR15, R2	Branch Unconditionally
NOP	D2(X2, B2)	BC 0, D2(X2, B2)	No Operation
NOPR	R2	BCR0, R2	No Operation (RR)

AFTER COMPARE INSTRUCTIONS (A:B)

BH	D2(X2, B2)	BC2, D2(X2, B2)	Branch on A High
BL	D2(X2, B2)	BC4, D2(X2, B2)	Branch on A Low
BE	D2(X2, B2)	BC8, D2(X2, B2)	Branch on A Equal B
BNH	D2(X2, B2)	BC13, D2(X2, B2)	Branch on A Not High
BNL	D2(X2, B2)	BC11, D2(X2, B2)	Branch on A Not Low
BNE	D2(X2, B2)	BC7, D2(X2, B2)	Branch on A Not Equal B

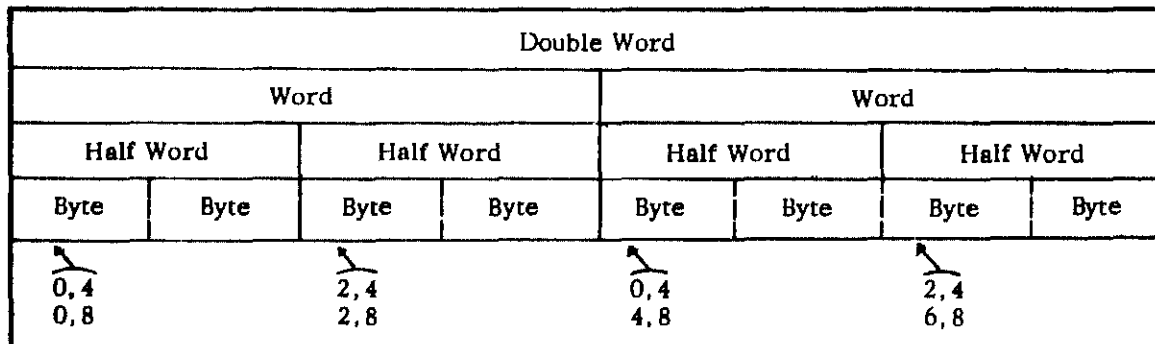
AFTER ARITHMETIC INSTRUCTIONS

BO	D2(X2, B2)	BC1, D2(X2, B2)	Branch on Overflow
BP	D2(X2, B2)	BC2, D2(X2, B2)	Branch on Plus
BM	D2(X2, B2)	BC4, D2(X2, B2)	Branch on Minus
BZ	D2(X2, B2)	BC8, D2(X2, B2)	Branch on Zero

AFTER TEST UNDER MASK INSTRUCTIONS

BO	D2(X2, B2)	BC1, D2(X2, B2)	Branch if Ones
BM	D2(X2, B2)	BC4, D2(X2, B2)	Branch if Mixed
BZ	D2(X2, B2)	BC8, D2(X2, B2)	Branch if Zeros

CNOP ALIGNMENT



EDIT AND EDIT & MARK SYMBOLS

Editorial

Symbol	Mask	Meaning
b	hex 40	blank
	hex 21	significance start character
	hex 22	field separator character
d	hex 20	digit-select character

PERMANENT STORAGE ASSIGNMENT

DEC	ADDRESS HEX	BINARY	LENGTH	PURPOSE
0	0	0000 0000	double-word	Initial program loading PSW
8	8	0000 1000	double-word	Initial program loading CCW1
16	10	0001 0000	double-word	Initial program loading CCW2
24	18	0001 1000	double-word	External old PSW
32	20	0010 0000	double-word	Supervisor call old PSW
40	28	0010 1000	double-word	Program old PSW
48	30	0011 0000	double-word	Machine-check old PSW
56	38	0011 1000	double-word	Input/output old PSW
64	40	0100 0000	double-word	Channel status word
72	48	0100 1000	word	Channel address word
76	4C	0100 1100	word	Unused
80	50	0101 0000	word	Timer (uses bytes 50, 51 & 52)
84	54	0101 0100	word	Unused
88	58	0101 1000	double-word	External new PSW
96	60	0110 0000	double-word	Supervisor call new PSW
104	68	0110 1000	double-word	Program new PSW
112	70	0111 0000	double-word	Machine-check new PSW
120	78	0111 1000	double-word	Input/output new PSW
128	80	1000 0000	(1)	Diagnostic scan-out area

(1) The size of the diagnostic scan-out area depends on the particular model and I/O channels; for models 30 through 75, maximum size is 256 bytes.

CONDITION CODE SETTING

	8	4	2	1
<b>FLOATING-POINT ARITHMETIC</b>				
Add Normalized S/L	zero	< zero	> zero	overflow
Add Unnormalized S/L	zero	< zero	> zero	overflow
Compare S/L (A:B)	equal	A low	A high	--
Load and Test S/L	zero	< zero	> zero	--
Load Complements S/L	zero	< zero	> zero	--
Load Negative S/L	zero	< zero	--	--
Load Positive S/L	zero	--	> zero	--
Subtract Normalized S/L	zero	< zero	> zero	overflow
Subtract Unnormalized S/L	zero	< zero	> zero	overflow

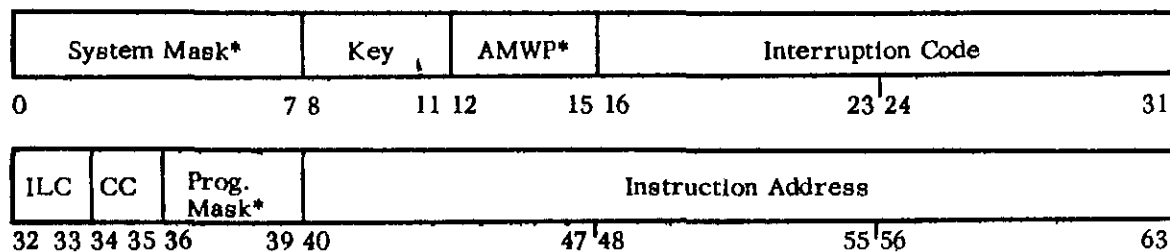
<b>FIXED-POINT ARITHMETIC</b>				
Add H/F	zero	< zero	> zero	overflow
Add Logical	zero, not carry	not zero, not carry	zero, carry	not zero, carry
Compare H/F (A:B)	equal	A low	A high	--
Load and Test	zero	< zero	> zero	--
Load Complement	zero	< zero	> zero	overflow
Load Negative	zero	< zero	--	--
Load Positive	zero	--	> zero	overflow
Shift Left Double	zero	< zero	> zero	overflow
Shift Left Single	zero	< zero	> zero	overflow
Shift Right Double	zero	< zero	> zero	--
Shift Right Single	zero	< zero	> zero	--
Subtract H/F	zero	< zero	> zero	overflow
Subtract Logical	--	not zero, not carry	zero, carry	not zero, carry

<b>DECIMAL ARITHMETIC</b>				
Add Decimal	zero	< zero	> zero	overflow
Compare Decimal (A:B)	equal	A low	A high	--
Subtract Decimal	zero	< zero	> zero	overflow
Zero and Add	zero	< zero	> zero	overflow

<b>LOGICAL OPERATIONS</b>				
AND	zero	not zero	--	--
Compare Logical (A:B)	equal	A low	A high	--
Edit	zero	< zero	> zero	--
Edit and Mark	zero	< zero	> zero	--
Exclusive OR	zero	not zero	--	--
OR	zero	not zero	--	--
Test Under Mask	zero	mixed	--	one
Translate and Test	zero	incomplete	complete	--

<b>INPUT/OUTPUT OPERATIONS</b>				
Halt I/O	not working	halted	stopped	not oper
Star I/O	available	CSW stored	busy	not oper
Test Channel	not working	CSW ready	working	not oper
Test I/O	available	CSW stored	working	not oper

PROGRAM STATUS WORD



- |    |                          |       |                               |
|----|--------------------------|-------|-------------------------------|
| 0  | Multiplexer channel mask | 13    | Machine check mask (M)        |
| 1  | Selector channel 1 mask  | 14    | Wait state (W)                |
| 2  | Selector channel 2 mask  | 15    | Problem state (P)             |
| 3  | Selector channel 3 mask  | 32-33 | Instruction Length code (ILC) |
| 4  | Selector channel 4 mask  | 34-35 | Condition code (CC)           |
| 5  | Selector channel 5 mask  | 36    | Fixed-point overflow mask     |
| 6  | Selector channel 6 mask  | 37    | Decimal overflow mask         |
| 7  | External mask            | 38    | Exponent underflow mask       |
| 12 | ASCII mode (A)           | 39    | Significance mask             |

\*A one-bit equals on, and permits and interrupt.

CODE FOR PROGRAM INTERRUPTION

Interruption Code			Program Interruption Cause
DEC	HEX	BINARY	
1	01	0000 0001	Operation
2	02	0000 0010	Privileged operation
3	03	0000 0011	Execute
4	04	0000 0100	Protection
5	05	0000 0101	Addressing
6	06	0000 0110	Specification
7	07	0000 0111	Data
8	08	0000 1000	Fixed-point overflow
9	09	0000 1001	Fixed-point divide
10	0A	0000 1010	Decimal overflow
11	0B	0000 1011	Decimal divide
12	0C	0000 1100	Exponent overflow
13	0D	0000 1101	Exponent underflow
14	0E	0000 1110	Significance
15	0F	0000 1111	Floating-point divide

HEXADECIMAL AND DECIMAL CONVERSION

To find the decimal number, locate the Hex number and its decimal equivalent for each position. Add these to obtain the decimal number. To find the Hex number, locate the next lower decimal number and its Hex equivalent. Each difference is used to obtain the next Hex number until the entire number is developed.

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

DEC → HEX  
 DEC → HEX

POWERS OF 16				POWERS OF 2							
16 <sup>n</sup>		n		2 <sup>n</sup>		n					
	1		0		512		9				
	16		1	1	024		10				
	256		2	2	048		11				
	4	096	3	4	096		12				
	65	536	4	8	192		13				
	1	048	5	16	384		14				
	16	777	6	32	768		15				
	268	435	7	65	536		16				
*	4	294	967	296	8	131	072	17			
	68	719	476	736	9	262	144	18			
	1	099	511	627	776	10	524	288	19		
	17	592	186	044	416	11	1	048	576	20	
	281	474	976	710	656	12	2	097	152	21	
	4	503	599	627	370	496	13	4	194	304	22
	72	057	594	037	927	936	14	8	388	608	23
1	152	921	504	606	846	976	15	16	777	216	24



OPERATION CODES FOR:  
RR FORMAT INSTRUCTIONS

Deci- mal	Hexa- deci- mal	Mnemonic	Graphic & Con- trol Symbols		(2) 7-Track Tape BCDIC	Punched Card Code	System/360 8-bit Code	(3)
			BCDIC	EBCDIC				
0	00					12-0-9-8-1	0000 0000	CCW
1	01					12-9-1	0000 0001	
2	02					12-9-2	0000 0010	
3	03					12-9-3	0000 0011	
4	04	SPM		PF		12-9-4	0000 0100	
5	05	BALR		HT		12-9-5	0000 0101	
6	06	BCTR		LC		12-9-6	0000 0110	
7	07	BCR		DEL		12-9-7	0000 0111	
8	08	SSK				12-9-8	0000 1000	CCW
9	09	ISK				12-9-8-1	0000 1001	
10	0A	SVC				12-9-8-2	0000 1010	
11	0B					12-9-8-3	0000 1011	
12	0C	(EBCDIC+)				12-9-8-4	0000 1100	
13	0D	(EBCDIC-)				12-9-8-5	0000 1101	
14	0E					12-9-8-6	0000 1110	
15	0F			CUI		12-9-8-7	0000 1111	CCW
16	10	LPR				12-11-9-8-1	0001 0000	
17	11	LNR				11-9-1	0001 0001	
18	12	LTR				11-9-2	0001 0010	
19	13	LCR				11-9-3	0001 0011	
20	14	NR		RES		11-9-4	0001 0100	
21	15	CLR		NL		11-9-5	0001 0101	
22	16	OR		BS		11-9-6	0001 0110	
23	17	XR		IL		11-9-7	0001 0111	
24	18	LR				11-9-8	0001 1000	CCW
25	19	CR				11-9-8-1	0001 1001	
26	1A	AR		CC		11-9-8-2	0001 1010	
27	1B	SR				11-9-8-3	0001 1011	
28	1C	MR				11-9-8-4	0001 1100	
29	1D	DR				11-9-8-5	0001 1101	
30	1E	ALR				11-9-8-6	0001 1110	
31	1F	SLR		CU2		11-9-8-7	0001 1111	
32	20	LPDR				11-0-9-8-1	0010 0000	CCW
33	21	LNDR				0-9-1	0010 0001	
34	22	LTDR				0-9-2	0010 0010	
35	23	LCDR				0-9-3	0010 0011	
36	24	HDR		BYP		0-9-4	0010 0100	
37	25			LF		0-9-5	0010 0101	
38	26			EOB		0-9-6	0010 0110	
39	27			PRE		0-9-7	0010 0111	
40	28	LDR				0-9-8	0010 1000	CCW
41	29	CDR				0-9-8-1	0010 1001	
42	2A	N ADR		SM		0-9-8-2	0010 1010	
43	2B	N SDR				0-9-8-3	0010 1011	
44	2C	N MDR				0-9-8-4	0010 1100	
45	2D	N DDR				0-9-8-5	0010 1101	
46	2E	AWR				0-9-8-6	0010 1110	
47	2F	SWR		CU3		0-9-8-7	0010 1111	
48	30	LPER				12-11-0-9-8-1	0011 0000	CCW
49	31	LNER				9-1	0011 0001	
50	32	LTER				9-2	0011 0010	
51	33	LCER				9-3	0011 0011	
52	34	HER		PN		9-4	0011 0100	
53	35			RS		9-5	0011 0101	
54	36			UC		9-6	0011 0110	
55	37			EOT		9-7	0011 0111	
56	38	LER				9-8	0011 1000	CCW
57	39	CER				9-8-1	0011 1001	
58	3A	N AER				9-8-2	0011 1010	
59	3B	N SER				9-8-3	0011 1011	
60	3C	N MER				9-8-4	0011 1100	
61	3D	N DER				9-8-5	0011 1101	
62	3E	AUR				9-8-6	0011 1110	
63	3F	SUR				9-8-7	0011 1111	

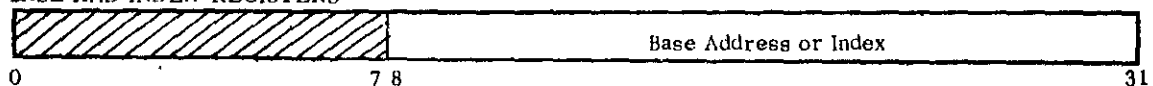
- (2) Note that check bit (C) is not shown; add C bit for odd or even parity as needed except for even parity, decimal 64 is CA, the same as decimal 122
- (3) CCW flag bit assignments
- (4) Decimal feature instructions
- (5) System/360 assembler programs require these codes

RR Format



R<sub>1</sub>, R<sub>2</sub> - Meaningful for all RR instructions except SPM and SVC

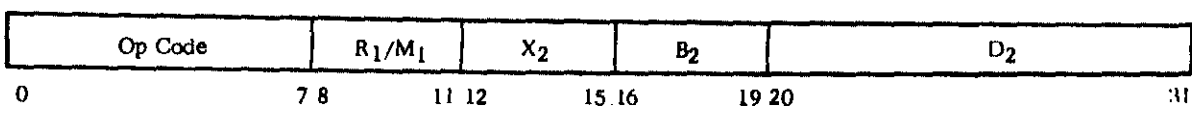
BASE AND INDEX REGISTERS



RX FORMAT INSTRUCTIONS

Deci- mal	Hexa- decim- al	Mnemonic	Graphic & Con- trol Symbols		(2) 7-Track Tape	Punched Card Code	System/360 8-bit Code	(3)	(3)
			BCDIC	EBCDIC	BCDIC				
64	40	STH	SP		(2)	no punches	0100 0000	CCW	
65	41	LA				12-0-9-1	0100 0001		
66	42	STC				12-0-9-2	0100 0010		
67	43	IC				12-0-9-3	0100 0011		
68	44	EX				12-0-9-4	0100 0100		
69	45	BAL				12-0-9-5	0100 0101		
70	46	BCT				12-0-9-6	0100 0110		
71	47	BC				12-0-9-7	0100 0111		
72	48	LH				12-0-9-8	0100 1000	CCW	
73	49	CH				12-8-1	0100 1001		
74	4A	AH	%			12-8-2	0100 1010		
75	4B	SH	.		B A 8 2 1	12-8-3	0100 1011		
76	4C	MH	)		B A 8 4	12-8-4	0100 1100		
77	4D		(		B A 8 4 1	12-8-5	0100 1101		(
78	4E	CVD	< +		B A 8 4 2	12-8-6	0100 1110		+
79	4F	CVB	±		B A 8 4 2 1	12-8-7	0100 1111		
80	50	ST	& + &		B A	12	0101 0000	CCW	
81	51					12-11-9-1	0101 0001		
82	52					12-11-9-2	0101 0010		
83	53					12-11-9-3	0101 0011		
84	54	N				12-11-9-4	0101 0100		
85	55	CL				12-11-9-5	0101 0101		
86	56	O				12-11-9-6	0101 0110		
87	57	X				12-11-9-7	0101 0111		
88	58	L				12-11-9-8	0101 1000	CCW	
89	59	C				11-8-1	0101 1001		
90	5A	A	!			11-8-2	0101 1010		
91	5B	S	\$		B 8 2 1	11-8-3	0101 1011		
92	5C	M	*		B 8 4	11-8-4	0101 1100		
93	5D	D	)		B 8 4 1	11-8-5	0101 1101		)
94	5E	AL	:		B 8 4 2	11-8-6	0101 1110		
95	5F	SL	^		B 8 4 2 1	11-8-7	0101 1111		
96	60	STD	-		B	11	0110 0000	CCW	
97	61		/		A 1	10-1	0110 0001		
98	62					11-0-9-2	0110 0010		
99	63					11-0-9-3	0110 0011		
100	64					11-0-9-4	0110 0100		
101	65					11-0-9-5	0110 0101		
102	66					11-0-9-6	0110 0110		
103	67					11-0-9-7	0110 0111		
104	68	LD				11-0-9-8	0110 1000	CCW	
105	69	CD				0-8-1	0110 1001		
106	6A	N AD				12-11	0110 1010		
107	6B	N SD			A 8 2 1	0-8-3	0110 1011		
108	6C	N MD	%( %'		A 8 4	0-8-4	0110 1100		
109	6D	NDD	√		A 8 4 1	0-8-5	0110 1101		
110	6E	AW	√		A 8 4 2	0-8-6	0110 1110		
111	6F	SW	#		A 8 4 2 1	0-8-7	0110 1111		
112	70	STE				12-11-0	0111 0000	CCW	
113	71					12-11-0-9-1	0111 0001		
114	72					12-11-0-9-2	0111 0010		
115	73					12-11-0-9-3	0111 0011		
116	74					12-11-0-9-4	0111 0100		
117	75					12-11-0-9-5	0111 0101		
118	76					12-11-0-9-6	0111 0110		
119	77					12-11-0-9-7	0111 0111		
120	78	LE				12-11-0-9-8	0111 1000	CCW	
121	79	CE				8-1	0111 1001		
122	7A	N AE			A	8-2	0111 1010		
123	7B	N SE	# = #		8 2 1	8-3	0111 1011		
124	7C	N ME	@' @		8 4	8-4	0111 1100		
125	7D	N DE	:		8 4 1	8-5	0111 1101		
126	7E	AU	> =		8 4 2	8-6	0111 1110		=
127	7F	SU	√		8 4 2 1	8-7	0111 1111		

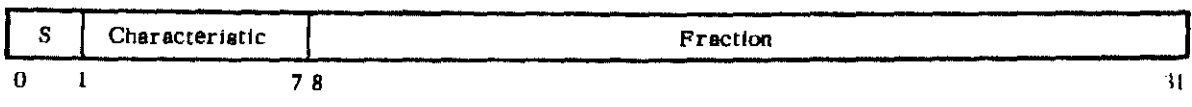
RX Format



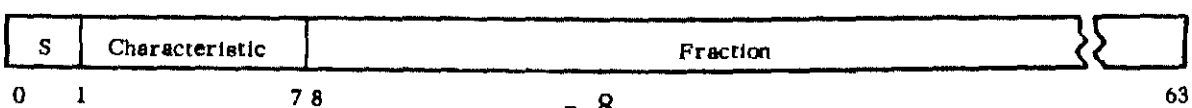
R<sub>1</sub>, D<sub>2</sub>(X<sub>2</sub>, B<sub>2</sub>)  
R<sub>1</sub>, S<sub>2</sub>(X<sub>2</sub>)

R<sub>1</sub>, D<sub>2</sub>(0, B<sub>2</sub>)  
R<sub>1</sub>, S<sub>2</sub>

SHORT FLOATING-POINT NUMBER



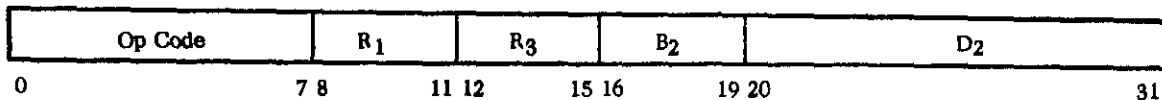
LONG FLOATING-POINT NUMBER



RS, SI FORMAT INSTRUCTIONS

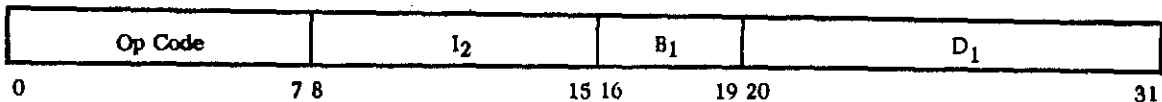
Decimal	Hexadecimal	Mnemonic	Graphic & Control Symbols BCDIC EBCDIC	(2) 7-Track Tape BCDIC	Punched Card Code	System/360 8-bit Code	(3)
128	80	SSM			12-0-8-1	1000 0000	CCW
129	81		a		12-0-1	1000 0001	
130	82	LPSW	b		12-0-2	1000 0010	
131	83	(Diagnose)	c		12-0-3	1000 0011	
132	84	WRD	d		12-0-4	1000 0100	
133	85	RDD	e		12-0-5	1000 0101	CCW
134	86	BXH	f		12-0-6	1000 0110	
135	87	BXLE	g		12-0-7	1000 0111	
136	88	SRL	h		12-0-8	1000 1000	
137	89	SLL	i		12-0-9	1000 1001	
138	8A	SRA			12-0-8-2	1000 1010	
139	8B	SLA			12-0-8-3	1000 1011	
140	8C	SRDL			12-0-8-4	1000 1100	
141	8D	SLDL			12-0-8-5	1000 1101	
142	8E	SRDA			12-0-8-6	1000 1110	
143	8F	SLDA			12-0-8-7	1000 1111	CCW
144	90	STM			12-11-8-1	1001 0000	
145	91	TM	j		12-11-1	1001 0001	
146	92	MVI	k		12-11-2	1001 0010	
147	93	TS	l		12-11-3	1001 0011	
148	94	NI	m		12-11-4	1001 0100	CCW
149	95	CLI	n		12-11-5	1001 0101	
150	96	OI	o		12-11-6	1001 0110	
151	97	XI	p		12-11-7	1001 0111	
152	98	LM	q		12-11-8	1001 1000	
153	99		r		12-11-9	1001 1001	
154	9A				12-11-8-2	1001 1010	
155	9B				12-11-8-3	1001 1011	
156	9C	SIO			12-11-8-4	1001 1100	
157	9D	TIO			12-11-8-5	1001 1101	
158	9E	HIO			12-11-8-6	1001 1110	CCW
159	9F	TCH			12-11-8-7	1001 1111	
160	A0				11-0-8-1	1010 0000	
161	A1				11-0-1	1010 0001	
162	A2		s		11-0-2	1010 0010	
163	A3		t		11-0-3	1010 0011	
164	A4		u		11-0-4	1010 0100	
165	A5		v		11-0-5	1010 0101	
166	A6		w		11-0-6	1010 0110	
167	A7		x		11-0-7	1010 0111	
168	A8		y		11-0-8	1010 1000	CCW
169	A9		z		11-0-9	1010 1001	
170	AA				11-0-8-2	1010 1010	
171	AB				11-0-8-3	1010 1011	
172	AC				11-0-8-4	1010 1100	
173	AD				11-0-8-5	1010 1101	CCW
174	AE				11-0-8-6	1010 1110	
175	AF				11-0-8-7	1010 1111	
176	B0				12-11-0-8-1	1011 0000	
177	B1				12-11-0-1	1011 0001	
178	B2				12-11-0-2	1011 0010	
179	B3				12-11-0-3	1011 0011	
180	B4				12-11-0-4	1011 0100	
181	B5				12-11-0-5	1011 0101	
182	B6				12-11-0-6	1011 0110	
183	B7				12-11-0-7	1011 0111	CCW
184	B8				12-11-0-8	1011 1000	
185	B9				12-11-0-9	1011 1001	
186	BA				12-11-0-8-2	1011 1010	
187	BB				12-11-0-8-3	1011 1011	
188	BC				12-11-0-8-4	1011 1100	
189	BD				12-11-0-8-5	1011 1101	
190	BE				12-11-0-8-6	1011 1110	
191	BF				12-11-0-8-7	1011 1111	

RS Format



R<sub>1</sub>, R<sub>3</sub>, D<sub>2</sub>(B<sub>2</sub>)    BXH, BXLE,    R<sub>1</sub>, D<sub>2</sub>(B<sub>2</sub>)    Shift  
R<sub>1</sub>, R<sub>3</sub>, S<sub>2</sub>    LM, STM    R<sub>1</sub>, S<sub>2</sub>    instructions

SI Format

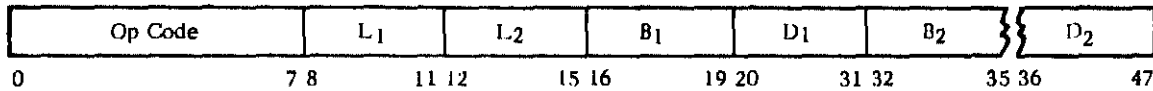


D<sub>1</sub>(B<sub>1</sub>)    LPSW, SSM, HIO, SIO  
S<sub>1</sub>    TIO, TCH, TS  
D<sub>1</sub>(B<sub>1</sub>), I<sub>2</sub> } All other SI Instructions  
S<sub>1</sub>, I<sub>2</sub>

SS FORMAT INSTRUCTIONS

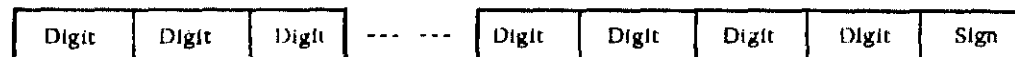
Decimal	Hexadecimal	Mnemonic	Graphic & Control Symbols		(2) 7-Track Tape BCDIC	Punched Card Code	System/360 8-bit Code	(3)
			BCDIC	EBCDIC				
192	C0		?		B A 8 2	12-0	1100 0000	CCW
193	C1		A	A	B A 1	12-1	1100 0001	
194	C2		B	B	B A 2	12-2	1100 0010	
195	C3		C	C	B A 2 1	12-3	1100 0011	
196	C4		D	D	B A 4	12-4	1100 0100	
197	C5		E	E	B A 4 1	12-5	1100 0101	
198	C6		F	F	B A 4 2	12-6	1100 0110	
199	C7		G	G	B A 4 2 1	12-7	1100 0111	
200	C8		H	H	B A 8	12-8	1100 1000	CCW
201	C9		I	I	B A 8 1	12-9	1100 1001	
202	CA					12-0-9-8-2	1100 1010	
203	CB					12-0-9-8-3	1100 1011	
204	CC					12-0-9-8-4	1100 1100	
205	CD					12-0-9-8-5	1100 1101	
206	CE					12-0-9-8-6	1100 1110	
207	CF					12-0-9-8-7	1100 1111	
208	D0				B 8 2	11-0	1101 0000	CCW
209	D1	MVN	j	J	B 1	11-1	1101 0001	
210	D2	MVC	k	K	B 2	11-2	1101 0010	
211	D3	MVZ	L	L	B 2 1	11-3	1101 0011	
212	D4	NC	M	M	B 4	11-4	1101 0100	
213	D5	CLC	N	N	B 4 1	11-5	1101 0101	
214	D6	OC	O	O	B 4 2	11-6	1101 0110	
215	D7	XC	P	P	B 4 2 1	11-7	1101 0111	
216	D8		Q	Q	B 8	11-8	1101 1000	CCW
217	D9		R	R	B 8 1	11-9	1101 1001	
218	DA					12-11-9-8-2	1101 1010	
219	DB					12-11-9-8-3	1101 1011	
220	DC	TR				12-11-9-8-4	1101 1100	
221	DD	TRT				12-11-9-8-5	1101 1101	
222	DE	ED (4)				12-11-9-8-6	1101 1110	
223	DF	EDMK (4)				12-11-9-8-7	1101 1111	
224	E0		*		A 8 2	0-8-2	1110 0000	CCW
225	E1					11-0-9-1	1110 0001	
226	E2		S	S	A 2	0-2	1110 0010	
227	E3		T	T	A 2 1	0-3	1110 0011	
228	E4		U	U	A 4	0-4	1110 0100	
229	E5		V	V	A 4 1	0-5	1110 0101	
230	E6		W	W	A 4 2	0-6	1110 0110	
231	E7		X	X	A 4 2 1	0-7	1110 0111	
232	E8		Y	Y	A 8	0-8	1110 1000	CCW
233	E9		Z	Z	A 8 1	0-9	1110 1001	
234	EA					11-0-9-8-2	1110 1010	
235	EB					11-0-9-8-3	1110 1011	
236	EC					11-0-9-8-4	1110 1100	
237	ED					11-0-9-8-5	1110 1101	
238	EE					11-0-9-8-6	1110 1110	
239	EF					11-0-9-8-7	1110 1111	
240	F0		0	0	8 2	0	1111 0000	CCW
241	F1	MVO	1	1	1	1	1111 0001	
242	F2	PACK	2	2	2	2	1111 0010	
243	F3	UNPK	3	3	2 1	3	1111 0011	
244	F4		4	4	4	4	1111 0100	
245	F5		5	5	4 1	5	1111 0101	
246	F6		6	6	4 2	6	1111 0110	
247	F7		7	7	4 2 1	7	1111 0111	
248	F8	ZAP (4)	8	8	8	8	1111 1000	CCW
249	F9	CP (4)	9	9	8 1	9	1111 1001	
250	FA	AP (4)				12-11-0-9-8-2	1111 1010	
251	FB	SP (4)				12-11-0-9-8-3	1111 1011	
252	FC	MP (4)				12-11-0-9-8-4	1111 1100	
253	FD	DP (4)				12-11-0-9-8-5	1111 1101	
254	FE					12-11-0-9-8-6	1111 1110	
255	FF					12-11-0-9-8-7	1111 1111	

SS Format

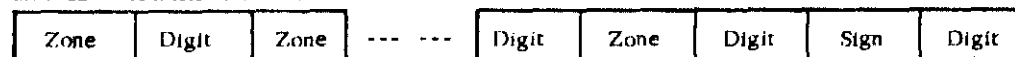


D1(L<sub>1</sub>, B<sub>1</sub>), D2(L<sub>2</sub>, B<sub>2</sub>) | PACK, UNPK, MVO, AP,  
 S1(L<sub>1</sub>), S2(L<sub>2</sub>) | CP, DP, MP, SP, ZAP  
 D1(L<sub>1</sub>, B<sub>1</sub>), D2(B<sub>2</sub>) | NC, OC, XC, CLC, MVC, MVN,  
 S1(L<sub>1</sub>), S2 | MVZ, TR, TRT, ED, EDMK

PACKED DECIMAL NUMBER

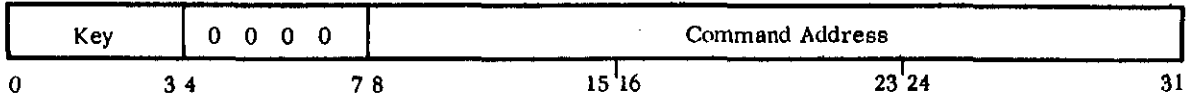


ZONED DECIMAL NUMBER

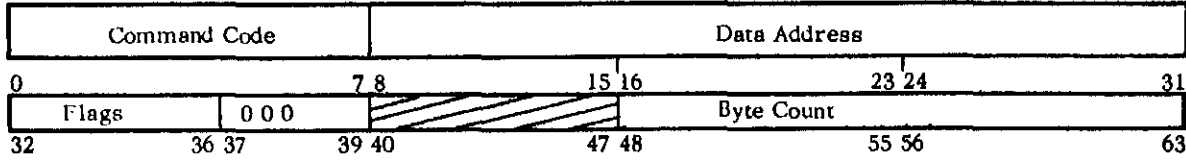


INPUT/OUTPUT

CHANNEL ADDRESS WORD

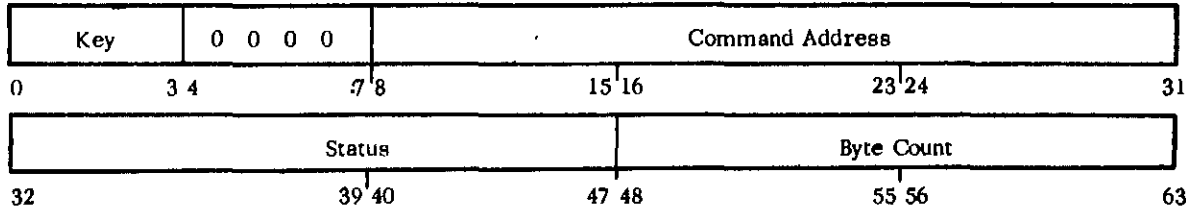


CHANNEL COMMAND WORD



Refer to OPERATION CODE tables for flag bit (Bits 32-36) assignments.  
 Bit 32 causes the address portion of the next CCW to be used.  
 Bit 33 causes the command code and data address in the next CCW to be used.  
 Bit 34 causes a possible incorrect length indication to be suppressed.  
 Bit 35 suppresses the transfer of information to main storage.  
 Bit 36 causes an interruption as Program Control Interrupt

CHANNEL STATUS WORD



- |                     |                                    |                            |
|---------------------|------------------------------------|----------------------------|
| 32 Attention        | 38 Unit check                      | 43 Protection check        |
| 33 Status modifier  | 39 Unit exception                  | 44 Channel data check      |
| 34 Control unit end | 40 Program-controlled interruption | 45 Channel control check   |
| 35 Busy             | 41 Incorrect length                | 46 Interface control check |
| 36 Channel end      | 42 Program check                   | 47 Chaining check          |
| 37 Device end       |                                    |                            |

Count: Bits 48 - 63 form the residual count for the last CCW used.

DASD CHANNEL COMMAND CODES (See A26-5988)

Command for CCW		Count	Multiple Track (M-T) Off		M-T On	
			8-Bit Code 0123 4567	Hex Dec	Hex	Dec
Control	No Op	X	0000 0011	03 03		
	Release*	X	0001 0111	17 23		
	Restore	X	0001 0011	13 19		
	Seek	6	0000 0111	07 07		
	Seek Cylinder	6	0000 1011	0B 11		
	Seek Head	6	0001 1011	1B 27		
	Sense I/O	4	0000 0100	04 04		
	Set File Mask	1	0001 1111	1F 31		
	Space Record	X	0000 1111	0F 15		
	Transfer in Channel	X	xxxx 1000	x8		
Search	Home Address EQ	4 (usually)	0011 1001	39 57	B9	185
	Identifier EQ	5 (usually)	0011 0001	31 49	B1	177
	Identifier HI	5 (usually)	0101 0001	51 81	D1	209
	Identifier EQ or HI	5 (usually)	0111 0001	71 113	F1	241
	Key EQ	1 to 255	0010 1001	29 41	A9	169
	Key HI	1 to 255	0100 1001	49 73	C9	201
	Key EQ or HI	1 to 255	0110 1001	69 105	E9	233
	Key & Data EQ*	Note 1	0010 1101	2D 45	AD	173
	Key & Data HI*		0100 1101	4D 77	CD	205
Key & Data EQ or HI*	0110 1101		6D 109	ED	237	
Read	Home Address	5	0001 1010	1A 26	9A	154
	Count	8	0001 0010	12 18	92	146
	Record R0	Number of bytes transferred	0001 0110	16 22	96	150
	Data		0000 0110	06 06	86	134
	Key & Data Count, Key & Data		0000 1110	0E 14	8E	142
		0001 1110	1E 30	9E	158	
Write	Home Address	5 (usually)	0001 1001	19 25		
	Record R0	8+KL+DL of R0	0001 0101	15 21		
	Count, Key & Data	8+KL+DL	0001 1101	1D 29		
	Special Count, Key & Data*	8+KL+DL	0000 0001	01 01		
	Data	DL	0000 0101	05 05		
	Key & Data	KL & DL	0000 1101	0D 13		

\* Special Feature Note 1. Includes mask bytes in search argument.  
 M-T On = M-T Off except, during Search and Read bit 0 = 1 in M-T On.  
 x = not significant; KL = Key Length DL = Data Length; EQ = Equal; HI = High

CHANNEL COMMAND CODES

Device	Command for CCW	8-Bit Code								Hex	Dec													
		0	1	2	3	4	5	6	7															
1052	Read Inquiry BCD	0	0	0	0	1	0	1	0	0A	10													
	Read Reader 2 BCD	0	0	0	0	0	0	1	0	02	02													
	Write BCD, Auto Carriage Return	0	0	0	0	1	0	0	1	09	09													
	Write BCD, No Auto Carriage Return	0	0	0	0	0	0	0	1	01	01													
	No Op	0	0	0	0	0	0	1	1	03	03													
	Sense	0	0	0	0	0	1	0	0	04	04													
	Alarm	0	0	0	0	1	0	1	1	0B	11													
2540	Read, Feed, Select Stacker SS Type AA	S	S	D	0	0	0	1	0															
	Read Type AB	1	1	D	0	0	0	1	0															
	Read, Feed (1400 compatibility mode only)	1	1	D	1	0	0	1	0															
	Feed, Select Stacker SS Type BA	S	S	1	0	0	0	1	1															
	PFR Punch, Feed, Select Stacker SS Type BA	S	S	D	0	1	0	0	1															
	Punch, Feed, Select Stacker SS Type BB	S	S	D	0	0	0	0	1															
		S	S	D	0	0	0	0	1															
<table border="1" style="display: inline-table; margin-right: 20px;"> <tr><th>SS</th><th>Stacker</th></tr> <tr><td>00</td><td>R1</td></tr> <tr><td>01</td><td>R2</td></tr> <tr><td>10</td><td>RP3</td></tr> </table> <table border="1" style="display: inline-table;"> <tr><th>D</th><th>Data Mode</th></tr> <tr><td>0</td><td>EBCDIC</td></tr> <tr><td>1</td><td>Column Binary</td></tr> </table>		SS	Stacker	00	R1	01	R2	10	RP3	D	Data Mode	0	EBCDIC	1	Column Binary									
SS	Stacker																							
00	R1																							
01	R2																							
10	RP3																							
D	Data Mode																							
0	EBCDIC																							
1	Column Binary																							
1442 N1	Read M M M M	M	M	M	0	0	0	1	0															
	Read	M	M	M	0	0	0	0	1															
	Read	M	M	0	0	0	0	1	1															
	Read	0	0	0	0	0	0	0	1															
	Read	1	1	X	Eject and SS1	0	0	M	0															
	Write	0	0	X	SS1	0	1	0	0															
	Write	1	0	X	Eject and SS1																			
	Write	0	1	X	SS2																			
	Write	1	1	X	Eject and SS2																			
	Control	1	0	Eject and SS1																				
	Control	0	1	SS2																				
	Control	1	1	Eject and SS2																				
	Sense		1	1	Punch diagnostic																			
	Sense		0	1	Read diagnostic																			
<p>X = 0 means EBCDIC mode X = 1 means Column Binary Mode</p>																								
1403 or 1443	Write, No Space	0	0	0	0	0	0	1	0	01	01													
	Write, Space 1 After Print	0	0	0	0	1	0	0	1	09	09													
	Write, Space 2 After Print	0	0	0	1	0	0	0	1	11	17													
	Write, Space 3 After Print	0	0	0	1	1	0	0	1	19	25													
	Write, Skip To Channel N After Print	1	C	H	A	N	0	0	1															
	Diagnostic Read (1403)	0	0	0	0	0	0	1	0	02	02													
	Diagnostic Read (1443)	0	0	0	0	0	1	1	0	06	06													
	Test I/)	0	0	0	0	0	0	0	0	00	00													
	Sense	0	0	0	0	0	1	0	0	04	04													
	Carriage Control	Space 1 Line Immediately	0	0	0	0	1	0	1	1	0B	11												
Space 2 Line Immediately		0	0	0	1	0	0	1	1	13	19													
Space 3 Line Immediately		0	0	0	1	1	0	1	1	1B	27													
Skip To Channel N Immediately		1	C	H	A	N	0	1	1															
No Op		0	0	0	0	0	0	1	1	03	03													
2400 Tape*	Transfer in Channel	0	0	0	0	1	0	0	0	08	08													
	Sense	0	0	0	0	0	1	0	0	04	04													
	Read Backward**	0	0	0	0	1	1	0	0	0C	12													
	Write	0	0	0	0	0	0	0	1	01	01													
	Read	0	0	0	0	0	0	1	0	02	02													
	Control	0	0	C	C	C	1	1	1															
	Mode Set	D	D	M	M	M	0	1	1															

\* 9 track op. forces 800 BPI and odd parity; also, it overrides 7 track but does not reset 7 track. Load/Sys Reset forces 7 track to 800 BPI, odd parity, data converter on, translator off.

C	C	C	Control Codes	Hex	Dec
0	0	0	REW	7	7
0	0	1	RUN	0F	15
0	1	0	ERG	17	23
0	1	1	WTM	1F	31
1	0	0	BSR	27	39
1	0	1	BSF	2F	47
1	1	0	FSR	37	55
1	1	1	FSF	3F	63

D	D	7 Track Density
0	0	200
0	1	556
1	0	800
1	1	800

M	M	M	(Mode Modifiers)	Set Density	Set Odd Parity	Set Even Parity	Data Converter On	Data Converter Off	Translator On	Translator Off	Request TTE (Track in Error)
0	0	0	No Op								
0	0	1	Not Used								
0	1	0	Reset Condition	x	x		x			x	
0	1	1	Nine-track only								x
1	0	0		x		x		x		x	
1	0	1		x		x		x		x	
1	1	0	Reset Condition	x	x			x		x	
1	1	1		x	x			x		x	

\*\*Overrides Data Converter On

NEW CCB FORMAT

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16 - 23
COUNT	TRANS INFO	CSW STATUS	SYM- BOLIC UNIT AD- RESS	X'00'	CCW AD- DRESS	X'20' X'40' X'80' NOTE	CSW/ CCW ADDR OR CHAN APND	SENSE CCW								

BYTE 2

0	1	2	3	4	5	6	7
TRAFFIC BIT	FLAG /* /&	UNREC I/O ERR	ACCEPT UNREC I/O ERR	RETURN DASD DATA CHECKS OR 2671 ERROR	POST DEV END	RETURN TAPE 2540P DASD ERR	USER ERROR RTN

BYTE 3

0	1	2	3	4	5	6	7
DASD COUNT DATA CHECKS	DASD TRACK OVER- RUN	DASD END OF CYL	2540p EQ CHECK TAPE OR DASD RD ERROR	QUEST'L COND	UNUSED	CH.9OV- FLOW OR DASD VERIFY ERROR	COMMND CHAIN RETRY FROM MXT CCW

ccb - name	CCB	SY5xxx, ccw-name, X'nnnn', <u>sense-address</u>
------------	-----	---

NOTE: X'20' = SENSE INFO DESIRED, X'40' = T/P, X'80' = ERP USING CCB

Byte	Bit	Condition Indicated		On Values for Third Operand in CCB Macro	Mask for Test Under Mask Instruction
		1 (ON)	0 (OFF)		
2	0 Traffic Bit (WAIT)	I/O Completed. Normally set at Channel End. Set at Device End if Bit 5 is ON.	I/O Requested and not completed.		X'80'
	1 End of File on System Input	/* or /& on SYSRDR or SYSIPT. Byte 4, Unit Exception Bit is also ON.			X'40'
	2 Unrecoverable I/O Error	I/O error passed back due to program option or operator option	No program or operator option error was passed back		X'20'
	3* Accept Unrecoverable I/O Error (Bit 2 is ON)	Return to User after Physical IOCS Attempts to correct I/O Error. +	Operator Option: Dependent on the Error	X'1000'	X'10'
	4* 2671 Data Check	Operator Options: Ignore, Retry, or Cancel	Operator Option: Retry or Cancel	X'0800'	X'08'
	----- Return an, DASD Data Checks	Return to User			
	5* Post at Device End	Device End Condition will be posted i. e., byte 2, bit 0 and byte 3, bits 3 and 6 set at Device End. Also byte 4, bit 5 is set.	Device End Conditions will not be posted. Traffic Bit is set at Channel End.	X'0400'	X'04'
6* Return: Uncorrectable tape read data check; 2540 or 2520 punch equipment check; or DASD read or verify data check. (Data checks on count not returned.)	Return to user after physical IOCS at tape or DASD error.	Operator Option: Ignore or Cancel for Tapes and punches. Retry or cancel for DASD.	X'0200'	X'02'	
7* User Error Routine	User will handle error recovery (Test Bit 2). □	A Physical IOCS Error routine will be used.	X'0100'	X'01'	
3	0 Data check in DASD count Field.	Yes-Byte 3, bit 3 is OFF; Byte 2, bit 2 is ON.	No		X'80'
	1 DASD Track overrun.	Yes	No		X'40'
	2 End of DASD Cylinder	Yes	No		X'20'
	3 Tape read data check; 2540 or 2520 punch equipment check; or any DASD data check.	Operation was unsuccessful. Byte 2, Bit 2 is also ON. Byte 3, Bit 0 is OFF.	No		X'10'
	4 Questionable Condition	Card: Unusual Command sequence (2540). DASD: No record found.			X'08'
3	5 Unused: Must be zero				
	6 Verify Error for DASD or Carriage Channel 9 Overflow	Yes. (Set ON when Channel 9 is reached only if Byte 2, Bit 5 is ON.)	No		X'02'
	7* Command Chain Retry	Retry begins at last CCW executed.	Retry begins at first CCW of channel program.	X'0001'	X'01'

\* User Option Bits. Set in CCB macro. Physical IOCS sets the other bits OFF at EXCP time and ON when the condition specified above occurs.

+ I/O program check, command reject, or tape equipment check will always terminate the program.

□ User must handle all error or exceptional conditions except Channel Control Check, Interface Control Check, I/O Program Check, and I/O Protection Check.



## Sense Bytes

	Sense Bit Position	
DISK	Byte 0, Bit 0	Command Reject
	1	Intervention Required
	2	Bus Out Parity
	3	Equipment Check
	4	Data Check
	5	Overflow
	6	Track Condition Check
	7	Seek Check
	Byte 1, Bit 0	Data Check in Count (also causes Byte 0, Bit 4)
	1	Track Overflow
	2	End-of-Cylinder
	3	Invalid Sequence (also causes Byte 0, Bit 0)
	4	No Record Found
	5	File Protected (also causes Byte 0, Bit 0)
	6	Missing Address Marker (also causes Byte 0, Bit 4)
7	Overflow Incomplete	
TAPE	Byte 0, Bit 0	Command Reject
	1	Intervention Required
	2	Bus Out Parity
	3	Equipment Check
	4	Data Check
	5	Overflow
	6	Word Count Zero
	7	Data Converter Check
	Byte 1, Bit 0	Noise Record
	1	Tape Unit Status A (Selected unit is available)
	2	Tape Unit Status B (Selected unit is busy, rewinding, or switched)
	3	7 Track Operation
	4	At Load Point
	5	In Write Status
	6	File Protected
7	Tape Indicator (Status of tape indicate on the selected tape unit)	



## FOREGROUND INITIATION EXAMPLES

Initiating a foreground program in an environment where there is one card reader and it is used for the batch job stream (SYSRDR/SYSIPT).

1. At IPL time:
  - a) Do not assign SYSRDR or SYSIPT or any logical unit to the card reader at System Generation.
  - b) If possible, System Generate all Physical Unit Blocks needed for operating (DVCGEN macro).
  - c) Place JCL statements for foreground program in the card reader followed by batch jobs and ready the reader.
  - d) IPL from SYSRES.
  - e) Press 1052 (SYSLOG) Attention.
  - f) Type in any required ADD and DEL commands on SYSLOG.
  - g) Type in SET command on SYSLOG.
  - h) Type in STOP command.
  - i) Press 1052 (SYSLOG) Attention.
  - j) Request Foreground Initiator. Type in "START Fn."
  - k) Assign card reader to SYSFGL. Type in "READ X'cuu'."
  - l) After Foreground Initiation has completed (EXEC card read), press 1052 (SYSLOG) Attention and enter command "START BG."
  - m) Type in "ASSGN SYSRDR, X'cuu'" (or SYSIN) with the card reader address. Type in @ . Batch processing will commence.
2. Between Jobs or Job Steps (unplanned):
  - a) Press 1052 (SYSLOG) Attention and enter PAUSE and @ commands.
  - b) When PAUSE occurs: Type in "ASSGN SYSRDR, UA" or if both RDR and IPT, "ASSGN SYSIN, UA" and "STOP."
  - c) Place JCL statements for foreground program in card reader followed by batch jobs not yet read and ready the reader.
  - d) Execute steps (i) through (m) under procedure 1 above.
3. Between Jobs or Job Steps (planned):
  - a) Place between Jobs or Job Steps in the card reader the following: "ASSGN SYSRDR, UA" (or SYSIN) followed by the JCL statements for the foreground program.
  - b) When Job Control encounters the unassignment and requests operator commands, execute steps (h) through (m) under procedure 1 above.
4. Asynchronously:
  - a) Press 1052 (SYSLOG) Attention.
  - b) Type in "START Fx" command.
  - c) Type in all JCL statements for foreground program.

Initiating a foreground program in an environment where there is a card reader available exclusively for the foreground program:

- a) Place JCL statements for foreground program in card reader.  
If the program is to use the card reader, its data cards can follow the EXEC card (see Note). Ready the card reader.
- b) Press 1052 (SYSLOG) Attention.
- c) Request Foreground Initiator. Type in "START Fn."
- d) Assign card reader to SYSFGL. Type in "READ X'cuu'."

NOTE: JCL Statements can assign the card reader used to input initiation commands to the foreground program as a SYSnnn unit.

JOB CONTROL COMMANDS

Operation	Operand	Remarks																																																																																																														
ASSGN	<p>SYsxxx, address { ,X'as' } (,TEMP)</p>	<p>SYsxxx: can be SYSRDR SYSIPT SYSIN SYSLST SYSPCH SYSOUT SYSLOG SYSLNK SYS000-SYS244</p> <p>address: can be X'cuu', UA, or IGN</p> <p>X'cuu': c = 0 - 6 uu = 00 - FE (0 - 254) in hex</p> <p>UA: unassign</p> <p>IGN: unassign and ignore</p> <p>X'ss': used for magnetic tape only</p> <table border="1"> <thead> <tr> <th>as</th> <th>Bytes per Inch</th> <th>Parity</th> <th>Trans-late Feature</th> <th>Convert Feature</th> </tr> </thead> <tbody> <tr><td></td><td>10</td><td>200</td><td>odd</td><td>off</td><td>on</td></tr> <tr><td></td><td>20</td><td>200</td><td>even</td><td>off</td><td>off</td></tr> <tr><td></td><td>28</td><td>200</td><td>even</td><td>on</td><td>off</td></tr> <tr><td></td><td>30</td><td>200</td><td>odd</td><td>off</td><td>off</td></tr> <tr><td></td><td>38</td><td>200</td><td>odd</td><td>on</td><td>off</td></tr> <tr><td></td><td>50</td><td>556</td><td>odd</td><td>off</td><td>on</td></tr> <tr><td></td><td>60</td><td>556</td><td>even</td><td>off</td><td>off</td></tr> <tr><td></td><td>68</td><td>556</td><td>even</td><td>on</td><td>off</td></tr> <tr><td></td><td>70</td><td>556</td><td>odd</td><td>off</td><td>off</td></tr> <tr><td></td><td>78</td><td>556</td><td>odd</td><td>on</td><td>off</td></tr> <tr><td></td><td>90</td><td>800</td><td>odd</td><td>off</td><td>on</td></tr> <tr><td></td><td>A0</td><td>800</td><td>even</td><td>off</td><td>off</td></tr> <tr><td></td><td>A8</td><td>800</td><td>even</td><td>on</td><td>off</td></tr> <tr><td></td><td>B0</td><td>800</td><td>odd</td><td>off</td><td>off</td></tr> <tr><td></td><td>B8</td><td>800</td><td>odd</td><td>on</td><td>off</td></tr> <tr><td></td><td>C0</td><td>800</td><td colspan="2">single density 9 track tape</td></tr> <tr><td></td><td>C0</td><td>1600</td><td colspan="2">dual density 9 track tape</td></tr> <tr><td></td><td>C8</td><td>800</td><td colspan="2">dual density 9 track tape</td></tr> </tbody> </table> <p>ALT: specifies alternate unit</p> <p>TEMP: assignment for logical unit will be destroyed by next JOB statement</p>	as	Bytes per Inch	Parity	Trans-late Feature	Convert Feature		10	200	odd	off	on		20	200	even	off	off		28	200	even	on	off		30	200	odd	off	off		38	200	odd	on	off		50	556	odd	off	on		60	556	even	off	off		68	556	even	on	off		70	556	odd	off	off		78	556	odd	on	off		90	800	odd	off	on		A0	800	even	off	off		A8	800	even	on	off		B0	800	odd	off	off		B8	800	odd	on	off		C0	800	single density 9 track tape			C0	1600	dual density 9 track tape			C8	800	dual density 9 track tape	
as	Bytes per Inch	Parity	Trans-late Feature	Convert Feature																																																																																																												
	10	200	odd	off	on																																																																																																											
	20	200	even	off	off																																																																																																											
	28	200	even	on	off																																																																																																											
	30	200	odd	off	off																																																																																																											
	38	200	odd	on	off																																																																																																											
	50	556	odd	off	on																																																																																																											
	60	556	even	off	off																																																																																																											
	68	556	even	on	off																																																																																																											
	70	556	odd	off	off																																																																																																											
	78	556	odd	on	off																																																																																																											
	90	800	odd	off	on																																																																																																											
	A0	800	even	off	off																																																																																																											
	A8	800	even	on	off																																																																																																											
	B0	800	odd	off	off																																																																																																											
	B8	800	odd	on	off																																																																																																											
	C0	800	single density 9 track tape																																																																																																													
	C0	1600	dual density 9 track tape																																																																																																													
	C8	800	dual density 9 track tape																																																																																																													
CLOSE	<p>SYsxxx { ,X'cuu'[,X'ss'] } { ,UA } { ,IGN } { ,ALT }</p>	<p>SYsxxx: for 2311 - SYSIN SYSRDR SYSIPT SYSPCH SYSLST</p> <p>for magnetic tape - SYSPCH SYSLST SYSOUT SYS000-SYS244</p> <p>X'cuu', X'ss', UA, IGN, ALT: values as described in ASSGN command</p>																																																																																																														
DVCDN	X'cuu'	<p>X'cuu': c = 0-6 uu = 00-FE (0-254) in hex</p>																																																																																																														

Operation	Operand	Remarks
DVCUP	X'cuu'	X'cuu': c = 0-6 uu = 00-FE (0-254) in hex
MTC	opcode, { X'cuu'' } [, nn] { SYSxxx }	opcode: BSF, BSR, ERG, FSF, FSR, RUN, REW, or WTM
		X'cuu': c = 0-6 uu = 00-FE (0-254) in hex  SYSxxx: any logical unit  nn: decimal number (01-99)
RESET	{ SYS PROG ALL SYSxxx }	Resets specified I/O device assignments
STOP	blank	Stops background program processing
LISTIO	{ SYS PROG F1 F2 ALL SYSxxx UNITS DOWN UA X'cuu' }	Causes listing of specified I/O assignments
LOG	blank	Causes logging of job control statements and foreground initiation commands on SYSLOG
NOLOG	blank	Suppresses logging of job control statements and foreground initiation commands on SYSLOG
CANCEL	blank	Cancels execution of background job
PAUSE	{ any user comment }	Causes pause at end of current job step
MAP	blank	Causes a map of areas in main storage to be printed on SYSLOG
ALLOC	{ F1 = nk [, F2 = nk ] { F2 = nk [, F1 = nk ] }	Allocates foreground program areas Value of n is an even number
UCS	SYSxxx, phasename { , FOLD } { , BLOCK } { , NULMSG }	Causes the 240-character universal character set contained in the core image library phase specified by phasename to be loaded as buffer storage in the IBM 2821 Control Unit. SYSxxx must be assigned to a 1403 Printer with the UCS feature.
HOLD	{ F1 { , F2 } { F2 { , F1 } }	Causes assignments for foreground logical units to be held across jobs.
RELSE	{ F1 { , F2 } { F2 { , F1 } }	Causes foreground logical units to be unassigned at EOJ.
UNA	{ F1 { , F2 } { F2 { , F1 } }	Causes immediate unassignment of foreground logical units.

Operation	Operand	Remarks
SET	[DATE=value 1] [,CLOCK= value 2] [,UPSI=value 3] [,LINECT=value 4] [,RCLST=value 5] [,RCPCH=value 6]	value 1: in one of the following formats  mm/dd/yy or dd/mm/yy  mm: month (01-12) dd: day (01-31) yy: year (00-99)  value 2: in the following format  hh/mm/ss  hh: hours (00-23) mm: minutes (00-59) ss: seconds (00-59)  value 3: 0, 1, or X  value 4: standard number of lines for output on each page of SYSLST  Value 5: decimal number indicating minimum number of SYSLST disk records remaining to be written before operator warning  value 6: decimal number indicating minimum number of SYSPCH disk records remaining to be written before operator warning
Ⓢ	blank	Ⓢ is alter code 5

#### ATTN COMMANDS

Operation	Operand	Remarks
PAUSE	[any user comments]	Causes pause at end of current job step
CANCEL	{ BG F1 F2 }	Cancel execution of current job in specified area
LOG	blank	Causes logging of job control statements and foreground initiation commands on SYSLOG
NOLOG	blank	Suppresses logging of job control statements and foreground initiation commands on SYSLOG
MAP	blank	Causes a map of areas in main storage to be printed on SYSLOG
ALLOC	{ F1 = nk [,F2 = nk] F2 = nk [,F1 = nk] }	Allocates foreground program areas Value of n is an even number
START	{ BG F1 F2 }	Initiates a background or foreground program
MSG	{ F1 F2 }	Transfers control to foreground program message routine
TIMER	{ BG F1 F2 }	Causes interval timer support to be given to program specified
Ⓢ	blank	Ⓢ is alter code 5

FOREGROUND INITIATION COMMANDS

Operation	Operand	Remarks
READ	X'cuu'	X'cuu': c = 0-6 uu = 00-FE (0-254) in hex  Note: Device must be a card reader
LISTIO	{ BG F1 F2 UA ALL }	Causes listing of specified I/O assignments
ASSGN	SYSnnn, address {,X'ss'} {,ALT}	SYSnnn: can be SYS000, SYS001, ... address: can be X'cuu' or IGN X'cuu': c = 0-6 uu = 00-FE (0-254) in hex IGN: unassign and ignore X'ss': used for magnetic tape only ALT: specifies alternate unit
VOL	SYSnnn, filename	SYSnnn: can be SYS000, SYS001, ... filename: one to seven alphabetic characters
DLAB	'label fields 1-3' xxxx, yyddd, yyddd, 'system code' [,type]	
XTENT	type, sequence, lower, upper, 'serial no.' SYSxxx [,B <sub>2</sub> ]	
TPLAB	'table fields 3-10'	
TPLAB	'table fields 3-10 label fields 11-13'	
CANCEL	blank	Cancels initiation of foreground program
EXEC	programe	programe: one to eight alphameric characters
LOG	blank	Causes logging of job control statements and foreground initiation commands on SYSLOG.
NOLOG	blank	Suppresses logging of job control statements and foreground initiation commands on SYSLOG.
HOLD	{ F1 [,F2] } { F2 [,F1] }	Causes assignments for foreground logical units to be held across jobs.
RELSE	{ F1 [,F2] } { F2 [,F1] }	Causes foreground logical units to be unassigned at EOJ.
UNA	{ F1 [,F2] } { F2 [,F1] }	Causes immediate unassignment of foreground logical units.
Ⓢ	blank	Ⓢ is alter code 5

JOB CONTROL STATEMENTS

Name	Operation	Operand	72	Remarks																																																																																																									
//	JOB	jobname	⊘	jobname: one to eight alphameric characters																																																																																																									
//	EXEC	[programe]	⊘	programe: one to eight alphameric characters. Used only if the program is in the core image library.																																																																																																									
//	ASSGN	SYSxxx, address [X'ss'] [,ALT]	⊘	SYSxxx: can be SYSRDR SYSIPT SYSIN SYSPCH SYSLST SYSLOG SYSLNK SYS000-SYS244  address: can be X'cuu', UA, or IGN X'cuu': c = 0-6 uu = 00-FE (0-254) in hex  UA: unassign IGN: unassign and ignore  X'ss': used for magnetic tape only  <table style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th> <th>Bytes</th> <th></th> <th>Trans-</th> <th>Convert</th> </tr> <tr> <th>ss</th> <th>per Inch</th> <th>Parity</th> <th>late</th> <th>Feature</th> </tr> <tr> <th></th> <th></th> <th></th> <th>Feature</th> <th>Feature</th> </tr> </thead> <tbody> <tr><td>10</td><td>200</td><td>odd</td><td>off</td><td>on</td></tr> <tr><td>20</td><td>200</td><td>even</td><td>off</td><td>off</td></tr> <tr><td>28</td><td>200</td><td>even</td><td>on</td><td>off</td></tr> <tr><td>30</td><td>200</td><td>odd</td><td>off</td><td>off</td></tr> <tr><td>38</td><td>200</td><td>odd</td><td>on</td><td>off</td></tr> <tr><td>50</td><td>556</td><td>odd</td><td>off</td><td>on</td></tr> <tr><td>60</td><td>556</td><td>even</td><td>off</td><td>off</td></tr> <tr><td>68</td><td>556</td><td>even</td><td>on</td><td>off</td></tr> <tr><td>70</td><td>556</td><td>odd</td><td>off</td><td>off</td></tr> <tr><td>78</td><td>556</td><td>odd</td><td>on</td><td>off</td></tr> <tr><td>90</td><td>800</td><td>odd</td><td>off</td><td>on</td></tr> <tr><td>A0</td><td>800</td><td>even</td><td>off</td><td>off</td></tr> <tr><td>A8</td><td>800</td><td>even</td><td>on</td><td>off</td></tr> <tr><td>B0</td><td>800</td><td>odd</td><td>off</td><td>off</td></tr> <tr><td>B8</td><td>800</td><td>odd</td><td>on</td><td>off</td></tr> <tr><td>C0</td><td>800</td><td colspan="3">single density 9-track type</td></tr> <tr><td>C0</td><td>1600</td><td colspan="3">dual density 9-track type</td></tr> <tr><td>C8</td><td>800</td><td colspan="3">dual density 9-track type</td></tr> </tbody> </table> ALT: specifies alternate unit		Bytes		Trans-	Convert	ss	per Inch	Parity	late	Feature				Feature	Feature	10	200	odd	off	on	20	200	even	off	off	28	200	even	on	off	30	200	odd	off	off	38	200	odd	on	off	50	556	odd	off	on	60	556	even	off	off	68	556	even	on	off	70	556	odd	off	off	78	556	odd	on	off	90	800	odd	off	on	A0	800	even	off	off	A8	800	even	on	off	B0	800	odd	off	off	B8	800	odd	on	off	C0	800	single density 9-track type			C0	1600	dual density 9-track type			C8	800	dual density 9-track type		
	Bytes		Trans-	Convert																																																																																																									
ss	per Inch	Parity	late	Feature																																																																																																									
			Feature	Feature																																																																																																									
10	200	odd	off	on																																																																																																									
20	200	even	off	off																																																																																																									
28	200	even	on	off																																																																																																									
30	200	odd	off	off																																																																																																									
38	200	odd	on	off																																																																																																									
50	556	odd	off	on																																																																																																									
60	556	even	off	off																																																																																																									
68	556	even	on	off																																																																																																									
70	556	odd	off	off																																																																																																									
78	556	odd	on	off																																																																																																									
90	800	odd	off	on																																																																																																									
A0	800	even	off	off																																																																																																									
A8	800	even	on	off																																																																																																									
B0	800	odd	off	off																																																																																																									
B8	800	odd	on	off																																																																																																									
C0	800	single density 9-track type																																																																																																											
C0	1600	dual density 9-track type																																																																																																											
C8	800	dual density 9-track type																																																																																																											
//	RESET	{ SYS PROG ALL } [SYSxxx]	⊘	Resets I/O device assignments																																																																																																									
//	DATE	mm/dd/yy or dd/mm/yy	⊘	mm: month (01-12) dd: day (01-31) yy: year (00-99)																																																																																																									
//	UPSI	nnnnnnnn	⊘	n: 0, 1, or X																																																																																																									
//	VOL	SYSxxx, filename	⊘	SYSxxx: can be SYS000-SYS244  filename: one to seven alphabetic characters																																																																																																									



Name	Operation	Operand	72	Remarks
//	DLAB	'label fields 1-3' xxxx, yyddd, yyddd, 'system code' [, type]	c	<p>'label fields 1-3': first three fields of Format 1 DASD file label. Is a 51-byte character string, contained within apostrophes and followed by a comma. Entire 51-byte field must be contained in the first of the two statements. Field 1 is the file name (44-byte alphanumeric); field 2 is the format identifier (1-byte numeric); field 3 is the file serial number (6-byte alphanumeric).</p> <p>c: any non-blank character in column 72</p> <p>xxxx: volume sequence number (4-digit numeric). Must begin in column 16 of the continuation statement. Columns 1-15 are blank.</p> <p>yyddd, yyddd: file creation date followed by file expiration date. Each is 5-digit numeric.</p> <p>'system code': not required. When used, a 13-character string, within apostrophes.</p> <p>type: SD, DA, ISC, or ISE. If omitted, SD is assumed.</p>
//	XTENT	type, sequence, lower, upper, 'serial no.', SYSxxx [, B <sub>2</sub> ]	b	<p>type: 1 for data area (no split cylinder) 2 for overflow area (for indexed sequential file) 4 for index area (for indexed sequential file) 128 for data area (split cylinder)</p> <p>sequence: sequence number of extent within multi-extent file. Can be 0 to 255.</p> <p>lower: lower limit of extent in the form B<sub>1</sub>C<sub>1</sub>C<sub>1</sub>C<sub>2</sub>C<sub>2</sub>C<sub>2</sub>H<sub>1</sub>H<sub>2</sub>H<sub>2</sub> where:</p> <p>B<sub>1</sub> = 0 for 2311; 0-9 for 2321 C<sub>1</sub>C<sub>1</sub> = 00 for 2311; 0-9 for 2321 C<sub>2</sub>C<sub>2</sub>C<sub>2</sub> = 000-199 for 2311; 000-009 for 2321 H<sub>1</sub> = 0 for 2311; 0-4 for 2321 H<sub>2</sub>H<sub>2</sub> = 00-09 for 2311; 00-19 for 2321</p> <p>Note that the last 5 strips of subcell 19 are reserved for alternate tracks for 2321.</p> <p>upper: upper limit of extent in the same form as for lower limit.</p> <p>'serial no.': 6-alphanumeric-character volume serial number contained within apostrophes.</p> <p>SYSxxx: can be SYS000-SYS244</p> <p>B<sub>2</sub>: 0 for 2311; 0-9 for 2321</p>

Name	Operation	Operand	72	Remarks
//	TPLAB	'label fields 3-10'	∅	'label fields 3-10': Indicated fields of the standard tape file label. A 49-byte character string, contained within apostrophes.
//	TPLAB	'label fields 3-10 label fields 11-13'	c	'label fields 3-10: same as above C: any nonblank character in column 72 label fields 11-13': 20-character direct continuation of the same character string begun with fields 3-10 (no blanks, apostrophes, or commas separating)
//	LBLTYP	{TAPE[(nn)] NSD (nn)}	∅	TAPE: used when tape files requiring label information are to be processed and no nonsequential disk files are to be processed.  (nn): optional and is present only for future expansion (it is ignored by Job Control)  NSD: nonsequential disk files are to be processed  (nn): largest number of extents per single file
//	RSTRT	SYSxxx, nnnn	∅	SYSxxx: symbolic unit name of the device on which the checkpoint records are stored. Can be SYS000-SYS244.  nnnn: four character identification of the checkpoint record to be used for restarting
//	LISTIO	{SYS PROG F1 F2 ALL SYSxxx UNITS DOWN UA X'cuu'}	∅	Causes listing of I/O assignments on SYSLST
//	MTC	opcode, SYSxxx (,nn)	∅	opcode: BSF, BSR, ERG, FSF, FSR, REW, RUN, or WTM  SYSxxx: any logical unit  nn: decimal number (01-99)

Name	Operation	Operand	72	Remarks
//	OPTION	optional [,option 2,....]	Ø	option: can be any of the following LOG            Log control statements on SYSLST NOLOG          Suppress LOG option DUMP            Dump registers and main storage on SYSLST in the case of abnormal program end NODUMP         Suppress DUMP option LINK            Write output of language translator on SYSLNK for linkage editing NOLINK         Suppress LINK option DECK            Output object module on SYSPCH NODECK         Suppress DECK option LIST            Output listing of source module on SYSLST NOLIST         Suppress LIST option LISTX           Output listing of object module on SYSLST NOLISTX        Suppress LISTX option SYM            Punch symbol deck on SYSPCH NOSYM          Suppress SYM option XREF            Output symbolic cross- reference list on SYSLST NOXREF         Suppress XREF option ERRS            Output listing of all errors in source program on SYSLST NOERRS         Suppress ERRS option CATAL          Catalog program or phase in core image library after completion of Linkage Editor run STDLABEL       Causes all sequential disk or tape labels to be written on the standard label track USRLABEL       Causes all sequential disk or tape labels to be written on the user label track 48C            48-character set 60C            60-character set
//	PAUSE	[comments]	Ø	PAUSE statement is always printed on 1052 (SYSLOG). If no 1052 is available, the statement is ignored.
/*	ignored	ignored	Ø	Columns 1 and 2 are the only columns checked.
/&	ignored	ignored	Ø	Columns 1 and 2 are the only columns checked.
*		comments	Ø	Column 2 must be blank.

JOB CONTROL (Sample)

COMPILE AND GO

```
// JOB SAMPLE
// OPTION LINK, DUMP
// (PHASE card)
// EXEC ASSEMBLY
// (ASSEMBLER source deck)
/*
// EXEC LNKEDT
// ASSGN for user program)
// EXEC
// (Data for user object program)
/*
// PAUSE
```

JOB LOAD AND GO

```
// JOB
// OPTION NODUMP, LINK
// PHASE card
// INCLUDE ABC
// INCLUDE XYZ
// INCLUDE
// (Object deck to be included)
/*
// ENTRY- // LBLTYP MSD(4)
// EXEC LNKEDT
// EXEC ASSGNS
// (Data for user object program)
/*
// PAUSE
```

SPOOL UTILITY (COMPILE & CATAL)

```
// JOB SOL (job name)
// OPTION CATAL
// PHASE FGPCT, + 0, NOAUTO (FGPCT=program name)
// EXEC ASSEMBLY
START START 0
      INCARD
      OUTAPE BUFSIZ=800
      END
/*
// EXEC LNKEDT
/&
```

PHYSICAL IOCS (SAMPLE)

```
START START 0
      BALR 5,0
      USING *,5
X      EXCP A
      WAIT A
      .
      .
      EXCP B
      WAIT B
      B X
A      CCB SYS004,N
B      CCB SYS005,M
N      CCW 2,INPUT,X'20',80
M      CCW 1,OUTPUT,X'20',80
INPUT DC CL80'0'
OUTPUT DC CL80'0'
      END START
```

PHASE CARD FORMAT

```
PHASE Prog.name, origin, NOANTO
1 ..... SYMBOL [(PHASE)] ± relocation
2 ..... * ± relocation
3 ..... S ± relocation
4 ..... ROOT
5 ..... + displacement
6 ..... F + address
```

*ALLOW SPACE  
FOR LABEL CHECKING*

UTILITY SAMPLE

D - P

```
// JOB DOS
// ASSGN SYSIPI,X'00C'
// ASSGN SYS000,X'191'
// ASSGN SYS005,X'00E'
// VOL SYS000,UI
// DLAB 'BOS UTILITYbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb1000000',bbbbbbbbbC
bbbbbbbbbbbbbbbb0001,66100,66100,'000000000000'
// XTENT 1,000,000111000,000121009,'000000',SYS000
// EXEC DKPR
// UDP TD,FF,A=(80,80),B=(132),OC,S1,Py,R1
// END
```

C - T

```
// JOB COTOTP
// ASSGN SYS004,X'00C'
// ASSGN SYS005,X'182'
// UPSI 1010
// EXEC CDTF
// OCT TC,FF,A=(80,80),B=(80,80),I1,ON
// END
```

C - T (STANDARD LABEL)

```
// JOB COTOTP
// ASSGN SYS004,X'00C'
// ASSGN SYS005,X'181',X'68'
// UPSI 1000
// VOL SYS005,UOUT
// TPLAB 'ASDFGBVCXZBGNHMJK11111100010001000101 66313 66313'
// EXEC CDTF
// OCT TC,FF,A=(80,80),B=(80,80),11,ON
// END
```

T - P (STANDARD LABEL)

```
// JOB CDTOPR
// ASSGN SYS004,X'181',X'68'
// ASSGN SYS005,X'00E'
// UPSI 0010
// VOL SYS004,UI
// TPIAB 'ASDFGBVCXZBGNHMJK11111100010001000101 66313 66313'
// EXEC TPRR
// UTP TD,FF,A=(80,80),B=(120),IN,OC,SI,PY,RI
// END
```

SORT/MERGE SAMPLE CONTROL CARDS

DISK SORT

SAMPLE 1 INPUT=DISK, WORK=DISK, OUTPUT=DISK

```
// JOB DOSORT
// ASSGN SYS004,X'191'
// ASSGN SYS005,X'191'
// ASSGN SYS006,X'191'
// VOL SYS004,FILEA
// DLAB 'BPS UTILITYbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb100000',bbbbbbbbbC
bbbbbbbbbbbbbbbb0001,66100,66100,'000000000000'
// XTENT 1,000,00001000,000010009,'000000',SYS004
// VOL SYS006,FILED
// DLAB 'BOS UTILITYbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb100000',bbbbbbbbbC
bbbbbbbbbbbbbbbb0001,66100,66100,'000000000000'
// XTENT 1,000,00011000,000121009,'000000',SYS006
// VOL SYS005,FILEW
// DLAB 'BWS UTILITYbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb100000',bbbbbbbbbC
bbbbbbbbbbbbbbbb0001,66100,66100,'000000000000',DA
// XTENT 1,000,000050000,000100009,'000000',SYS005
// EXEC DSORT
SORT FIELDS=(3,1,D),FORMAT=B1,FILES=1,SIZE=200
RECORD TYPE=F,LENGTH=(80,,80)
INPFIL INPUT=D,BLKSIZE=(80,X)
OUTFIL BLKSIZE=80,OUTPUT=D
OPTION PRINT,LABEL=(S,S)
END
```

SAMPLE 2 INPUT=TAPE, WORK=DISK, OUTPUT=TAPE

```
// JOB SMPL
// ASSGN SYS001,X'180',X'90'
// ASSGN SYS002,X'270'
// ASSGN SYS003,X'182'
// ASSGN SYS004,X'183'
// ASSGN SYS006,X'190'
// ASSGN SYS007,X'191'
// VOL SYS006,FILEW
// DLAB 'DISK WORK FILEbbbbbbbbbbbbbbbbbbbbbbbbbbbb111111',bbbbbbbbbX
bbbbbbbbbbbbbbbb0001,66311,66311,'DOSbbbbbbbbbbb',DA
// XTENT 1,0,000151000,000198009,'111111',SYS006
// XTENT 1,1,000001000,000198009,'111111',SYS007
// EXEC DSORT
RECORD TYPE=F,LENGTH=(80,,80)
SORT FIELDS=(1,10,A,11,5,A),FORMAT=B1,FILES=3,SIZE=6000
INPFIL INPUT=T,VOLUME=(1,1,1),BLKSIZE=(800,X),BYPASS
OUTFIL BLKSIZE=800,OUTPUT=T,OPEN=RWD,CLOSE=RWD
OPTION PRINT, STORAGE=32000,LABEL=(U,U)
END
```

TAPE SORT

SAMPLE 1

```
// JOB SORT1
// ASSGN SYS001,X'182'
// ASSGN SYS002,X'180'
// ASSGN SYS003,X'182'
// ASSGN SYS004,X'181'
// ASSGN SYS005,X'183'
// ASSGN SYS006,X'180'
// EXEC TSRT
SORT FIELDS=(7,2,A,14,6,A,64,1,A,11,2,A,33,1,A),FORMAT=BI,WORK=4
RECORD TYPE=F,LENGTH=(84,,84)
INPFIL VOLUME=1,BLKSIZE=(1680,X)
OUTFIL BLKSIZE=1680
OPTION PRINT,LABEL=(U,U,U)
END
```

SAMPLE 2 TAPE DRIVE=4 (UNLABEL)

```
// JOB SORTSMPL
// ASSGN SYS002,X'181',X'90'
// ASSGN SYS002,X'180',ALT
// ASSGN SYS001,X'270'
// ASSGN SYS003,X'270'
// ASSGN SYS004,X'182'
// ASSGN SYS005,X'183'
// EXEC TSRT
SORT FIELDS=(75,2,A,40,12,A,1,2,A),FORMAT=BI,WORK=3
RECORD TYPE=F,LENGTH=120
INPFIL VOLUME=3,BLKSIZE=(960,X),OPEN=RWD
OUTFIL BLKSIZE=960,CLOSE=RWD
OPTION LABEL=(U,U,U),PRINT
END
```

SAMPLE 3 TAPE DRIVE=4 (STANDARD LABEL)

```
// JOB DOS TAPE SORT
// ASSGN SYS001,X'180',X'68'
// ASSGN SYS002,X'180',X'68'
// ASSGN SYS003,X'181',X'68'
// ASSGN SYS004,X'182',X'68'
// ASSGN SYS005,X'183',X'68'
// ASSGN SYS006,X'184',X'68'
// VOL SYS002,FILEA
// TPLAB 'LOAN-SUMbbbbbbbb00088800010001000101 66180 66185'
// VOL SYS001,FILED
// TPLAB 'LOAN-SUMbbbbbbbb00088800010001000101 66180 66185'
// EXEC TSRT
SORT FIELDS=(5,9,A,1,4,A,19,24,A),FORMAT=81,WORK=4
RECORD TYPE=F,LENGTH=(200,200,200)
INPFIL VOLUME=1,BLKSIZE=(2000,X),BYPASS,CLOSE=UNLD
OUTFIL BLKSIZE=2000
OPTION LABEL=(S,S,S),PRINT
END
```

FORMAT OF LANGUAGE TRANSLATOR OUTPUT CARDS AND THE USER REPLACE CARD

ESD card

1	Multiple punch (12-2-9). Identifies this as a loader card.
2- 4	ESD -- External Symbol Dictionary card.
11-12	Number of bytes of information contained in this card.
15-16	External symbol identification number (ESID) of the first SD, PC, CM or ER on this card. Relates the SD, PC, CM or ER to a particular control section.
17-72	Variable information. 8 positions - Name 1 position - Type code to indicate SD, PC, LD, CM or ER 3 positions - Assembled origin 1 position - Blank 3 positions - Length, if an SD-type, CM-type, or a PC-type. If an LD-type, this field contains the external symbol identification number (ESID) of the SD containing the label.
73-80	May be used by the programmer for identification.

RLD card

1	Multiple punch (12-2-9). Identifies this as a loader card.
2- 4	RLD -- Relocation List Dictionary card.
11-12	Number of bytes of information contained in this card.
17-72	Variable information (multiple items). a. Two positions - pointer (relocation identifier) to the relocation factor of the contents of the load constant. b. Two positions - pointer (position identifier) to the relocation factor of the control section in which the load constant occurs. c. One position - flag indicating type of constant, as follows:  <u>Bits</u> 0-2 ignored 3 0 - a non-branch type load constant 1 - a branch type load constant 4-5 00 - load constant length = 1 byte 01 - load constant length = 2 bytes 10 - load constant length = 3 bytes 11 - load constant length = 4 bytes 6 0 - relocation factor is to be added 1 - relocation factor is to be subtracted

TXT card

1	Multiple punch (12-2-9). Identifies this as a loader card.
2- 4	TXT -- Text card.
6- 8	Assembled origin (address of first byte to be loaded from this card).
11-12	Number of bytes of text to be loaded.
15-16	External symbol identification number (ESID) of the control section (SD or PC) containing the text.
17-72	Up to 56 bytes of text -- data or instructions to be loaded.
73-80	May be used for program identification.



RLD (continued)

	<p>7 0 - Next load constant has different R and P identifiers: therefore, both R and P must be present.</p> <p>1 - Next load constant has the same R and P identifiers: therefore they are both omitted.</p> <p>Five significant bits of this byte are expanded in the RSERV printout.</p> <p>d. Three positions - assembled origin of load constant.</p>
73-80	May be used for program identification.

END card

1	Multiple punch (12-2-9). Identifies this as a loader card.
2- 4	END
6- 8	Assembled origin of the label supplied to the Assembler in the END card (optional).
15-16	ESID number of the control section to which this END card refers (only if 6-8 present).
17-22	Symbolic label supplied to the Assembler if this label was not defined within the assembly.
29-32	Control section length (if not specified in last SD or PC).
73-80	Not used.

REP card

1	Multiple punch (12-2-9). Identifies this as a loader card.
2- 4	REP -- Replace text card.
5- 6	Blank.
7-12	Assembled address of the first byte to be replaced (hexadecimal). Must be right justified with leading zeros if needed to fill the field.
13	Blank.
14-16	External symbol identification number (ESID) of the control section (SD) containing the text (hexadecimal). Must be right justified with leading zeros if needed to fill the field.
17-70	From 1 to 11 4-digit hexadecimal fields separated by commas, each replacing two bytes. A blank indicates the end of information in this card.
71-72	Blank.
73-80	May be used for program identification.

## LIBRAIAN

Function	Unit	Element	Control Statements Required
Catalog	Core Image Library	Phase	// JOB jobname // OPTION CATAL (Linkage Editor control statements) // EXEC LNKEDT / &
	Relocatable Library	Module	// JOB jobname // EXEC MAINT CATALR modulename /* / &
	Source Statement Library	Book	// JOB jobname // EXEC MAINT CATALS sublib. bookname /* / &
Delete	Core Image Library	Phase	// JOB jobname // EXEC MAINT DELETC phase 1 [, phase 2, ...] /* / &
		Program	// JOB jobname // EXEC MAINT DELETC prog 1.ALL [, prog 2.ALL, ...] /* / &
	Relocatable Library	Module	// JOB jobname // EXEC MAINT DELETR module 1 [, module 2, ...] /* / &
		Program	// JOB jobname // EXEC MAINT DELETR prog 1.ALL [, prog 2.ALL, ...] /* / &
		Library	// JOB jobname // EXEC MAINT DELETR ALL /* / &
	Source Statement Library	Book	// JOB jobname // EXEC MAINT DELETS sublib. book 1 [, sublib. book 2, ...] /* / &
		Sub-library	// JOB jobname // EXEC MAINT DELETS sublib. ALL /* / &
Rename	Core Image Library	Phase	// JOB jobname // EXEC MAINT RENAMC oldname, newname [, oldname, newname, ...] /* / &

Function	Unit	Element	Control Statements Required
	Relocatable Library	Module	// JOB jobname // EXEC MAINT RENAMR oldname, newname [,oldname, newname,...] /* /&
	Source Statement Library	Book	// JOB jobname // EXEC MAINT RENAMS sublib. oldname, sublib. newname [,sublib. oldname, sublib. newname,...] /* /&
Condense	Core Image Library	Library	// JOB jobname // EXEC MAINT CONDS CL /* /&
	Relocatable Library	Library	// JOB jobname // EXEC MAINT CONDS RL /* /&
	Source Statement Library	Library	// JOB jobname // EXEC MAINT CONDS SL /* /&
	Libraries	All	// JOB jobname // EXEC MAINT CONDS CL, RL, SL /* /&
Set Parameter for Automatic Condense	Libraries	Any or All	// JOB jobname // EXEC MAINT CONDL lib=nnnnn [,lib=nnnnn [,lib=nnnnn]] /* /& Notes: Values to be substituted for <u>lib</u> : CL -- Core image library RL -- Relocatable library SL -- Source statement library Values to be substituted for <u>nnnnn</u> : One to five decimal digits, with a maximum value of 65536.
Reallo- cation	System	Library	// JOB jobname // VOL SYSRES, IJSYSRES // DLAB 'DOS SYSTEM RESIDENCE FILE ...' // XTENT extent information // EXEC MAINT ALLOC id=cylin (tracks) [, id=cylin (tracks),...] /* /& Notes: Values to be substituted for <u>id</u> : CL -- Core image library RL -- Relocatable library SL -- Source statement library Values to be substituted for <u>cylin</u> and <u>tracks</u> : Any integer

Function	Unit	Element	Control Statements Required
Display	Core Image Library	Directory	// JOB jobname // EXEC DSERV DSPLY CD /* /&
	Relocatable Library	Module	// JOB jobname // EXEC RSERV DSPLY module 1, [, module 2, ...] /* /&
		Program	// JOB jobname // EXEC RSERV DSPLY prog 1.ALL [, prog 2. ALL, ...] /* /&
		Library	// JOB jobname // EXEC RSERV DSPLY ALL /* /&
		Directory	// JOB jobname // EXEC DSERV DSPLY RD /* /&
	Source Statement Library	Book	// JOB jobname // EXEC SSERV DSPLY sublib.book 1 [, sublib. book 2, ...] /* /&
		Sub-library	// JOB jobname // EXEC SSERV DSPLY sublib. ALL /* /&
		Directory	// JOB jobname // EXEC DSERV DSPLY SD /* /&
	Transient Directory	Directory	// JOB jobname // EXEC DSERV DSPLY TD /* /&
	System Directory	Directory	// JOB jobname // EXEC DSERV /* /&
	Directories	All	// JOB jobname // EXEC DSERV DSPLY ALL /* /&
	Punch	Relocatable Library	Module

Function	Unit	Element	Control Statements Required
Punch	Relocatable Library	Program	// JOB jobname // EXEC RSERV PUNCH prog 1. ALL [, prog 2. ALL,...] /* /&
		Library	// JOB jobname // EXEC RSERV PUNCH ALL /* /&
	Source Statement Library	Book	// JOB jobname // EXEC SSERV PUNCH sublib. book 1 [, sublib. book 2,...] [, CMPRSD] /* /&
		Sub-library	// JOB jobname // EXEC SSERV PUNCH sublib.ALL [, CMPRSD] /* /&
Display and Punch	Relocatable Library	Module	// JOB jobname // EXEC RSERV DSPCH module 1 [, module 2,...] /* /&
		Program	// JOB jobname // EXEC RSERV DSPCH prog 1. ALL [, prog 2. ALL,...] /* /&
		Library	// JOB jobname // EXEC RSERV DSPCH ALL /* /&
	Source Statement Library	Book	// JOB jobname // EXEC SSERV DSPCH sublib. book 1 [, sublib. book 2,...] [, CMPRSD] /* /&
		Sub-library	// JOB jobname // EXEC SSERV DSPCH sublib.ALL [, CMPRSD] /* /&
		Phase	// JOB jobname // ASSGN SYS002, X'cuu' // VOL SYS002, IJSYSRES // DLAB 'DOS SYSTEM RESIDENCE FILE... // XTENT extent information // ELEC CORGZ // ALLOC id=cylin (tracks) [, id=cylin (tracks),...] * PRECEDING ALLOC STATEMENT REQUIRED IF NEW LIMITS TO BE ESTABLISHED COPYC phase 1 [, phase 2,...] /* /&
Copy	Core Image Library	Phase	// JOB jobname // ASSGN SYS002, X'cuu' // VOL SYS002, IJSYSRES // DLAB 'DOS SYSTEM RESIDENCE FILE... // XTENT extent information // ELEC CORGZ // ALLOC id=cylin (tracks) [, id=cylin (tracks),...] * PRECEDING ALLOC STATEMENT REQUIRED IF NEW LIMITS TO BE ESTABLISHED COPYC phase 1 [, phase 2,...] /* /&

Function	Unit	Element	Control Statements Required
Copy	Core Image Library	Program	// JOB jobname // ASSGN SYS002, X'cuu' // VOL SYS002, IJSYSRES // DLAB 'DOS SYSTEM RESIDENCE FILE ...' // XTENT extent information // EXEC CORGZ // ALLOC id=cylin (tracks) [, id=cylin (tracks), ...] * PRECEDING ALLOC STATEMENT REQUIRED IF NEW LIMITS TO BE ESTABLISHED COPYC prog 1. ALL [, prog 2. ALL, ...] /* /&
		Library	// JOB jobname // ASSGN SYS002, X'cuu' // VOL SYS002, IJSYSRES // DLAB 'DOS SYSTEM RESIDENCE FILE ...' // XTENT extent information // EXEC CORGZ // ALLOC id=cylin (tracks) [, id=cylin (tracks), ...] * PRECEDING ALLOC STATEMENT REQUIRED IF NEW LIMITS TO BE ESTABLISHED COPYC ALL /* /&
	Relocatable Library	Module	// JOB jobname // ASSGN SYS002, X'cuu' // VOL SYS002, IJSYSRES // DLAB 'DOS SYSTEM RESIDENCE FILE ...' // XTENT extent information // EXEC CORGZ // ALLOC id=cylin (tracks) [, id=cylin (tracks), ...] * PRECEDING ALLOC STATEMENT REQUIRED IF NEW LIMITS TO BE ESTABLISHED COPYR module 1 [, module 2, ...] /* /&
		Program	// JOB jobname // ASSGN SYS002, X'cuu' // VOL SYS002, IJSYSRES // DLAB 'DOS SYSTEM RESIDENCE FILE ...' // XTENT extent information // EXEC CORGZ // ALLOC id=cylin (tracks) [, id=cylin (tracks), ...] * PRECEDING ALLOC STATEMENT REQUIRED IF NEW LIMITS TO BE ESTABLISHED COPYR prog 1. ALL [, prog 2. ALL, ...] /* /&
		Library	// JOB jobname // ASSGN SYS002, X'cuu' // VOL SYS002, IJSYSRES // DLAB 'DOS SYSTEM RESIDENCE FILE ...' // XTENT extent information // EXEC CORGZ // ALLOC id=cylin (tracks) [, id=cylin (tracks), ...]

Function	Unit	Element	Control Statements Required
			<pre> * PRECEDING ALLOC STATEMENT   REQUIRED IF NEW LIMITS TO BE   ESTABLISHED COPYR ALL /* / &amp; </pre>
Copy	Source Statement Library	Book	<pre> // JOB jobname // ASSGN SYS002, X'cuu' // VOL SYS002, IJSYSRES // DLAB 'DOS SYSTEM RESIDENCE   FILE ... // XTENT extent information // EXEC CORGZ // ALLOC id=cylin (tracks) [, id=cylin   (tracks),...] * PRECEDING ALLOC STATEMENT   REQUIRED IF NEW LIMITS TO BE   ESTABLISHED COPYS sublib. book 1   [, sublib. book 2, ...] /* / &amp; </pre>
		Sub-library	<pre> // JOB jobname // ASSGN SYS002, X'cuu' // VOL SYS002, IJSYSRES // DLAB 'DOS SYSTEM RESIDENCE   FILE ...' // XTENT extent information // EXEC CORGZ // ALLOC id=cylin (tracks) [, id=cylin   (tracks),...] * PRECEDING ALLOC STATEMENT   REQUIRED IF NEW LIMITS TO BE   ESTABLISHED COPYS sublib. ALL /* / &amp; </pre>
		Library	<pre> // JOB jobname // ASSGN SYS002, X'cuu' // VOL SYS002, IJSYSRES // DLAB 'DOS SYSTEM RESIDENCE   FILE ...' // XTENT extent information // EXEC CORGZ // ALLOC id=cylin (tracks) [, id=cylin   (tracks),...] * PRECEDING ALLOC STATEMENT   REQUIRED IF NEW LIMITS TO BE   ESTABLISHED COPYS ALL /* / &amp; </pre>
	Libraries	All	<pre> // JOB jobname // ASSGN SYS002, X'cuu' // VOL SYS002, IJSYSRES // DLAB 'DOS SYSTEM RESIDENCE   FILE ... // XTENT extent information // EXEC CORGZ // ALLOC id=cylin (tracks) [, id=cylin   (tracks),...] * PRECEDING ALOC STATEMENT   REQUIRED IF NEW LIMITS TO BE   ESTABLISHED COPY ALL /* / &amp; </pre>

I/O MODULE TABLE

	1ST CHARACTER	2ND	3RD	4TH	5TH	6TH	7TH	8TH
CDMOD	I	J	C	F V U RECFORM=FIXUNB =VARUNB =UNDEF	A Y C Z CTLCHR=ASA =YES CONTROL=YES (NEITHER) CTLCHR (NOR) CONT'OL (SPECIFIED)	I O C TYPEFLE=INPUT =OUTPUT =CMBND	Z W I B (NEITHER) WORKA (NOR) IOAREA2 WORKA=YES IOAREA2=YES (BOTH) WORKA (AND) IOAREA2	0 1 2 3 4 DEVICE=2540 =1442 =2520 =2501 =2540 (& CRDERR SPECIFIED)
PRMOD	I	J	D	F V U (AS IN) CDMOD	A Y C Z (AS IN) CDMOD	P Z PRINTOV=YES (PRINTOV NOT SPECIFIED)	I Z IOAREA2=YES IOAREA2 (NOT SPECIFIED)	W Z WORKA=YES WORKA (NOT SPECIFIED)
MTMOD (GET/ PUT)	I	J	F	F V U FIXUNB (OR) FIXBLK VARUNB (OR) VARBLK UNDEFINED	B Z READ=BACKWARD =FORWARD	C Z CKPTREC=YES (CKPTREC NOT SPECIFIED)	W Z WORKA=YES (WORKA NOT SPECIFIED)	Z
MTMOD (WORK FILES)	I	J	F	W IGNORE ERROPT=SKIP NAME (ERROPT NOT SPECIFIED)	E Z ERROPT=SKIP NAME (ERROPT NOT SPECIFIED)	N S Z NOTEPT=YES =POINTS (NOTEPT NOT SPECIFIED)	Z	Z
SDMOD (GET/ PUT)	I	J	G	F V U (AS IN) MTMOD	I O U TYPEFLE=INPUT =OUTPUT INPUT & UPDATE	O Z ERROPT=YES (ERROPT NOT SPECIFIED)	T Z TRUNCS=YES (TRUNCS NOT SPECIFIED)	C Z CONTROL=YES (CONTROL NOT SPECIFIED)
SDMOD (WORK- FILES)	I	J	G	W (AS IN) MTMOD	E Z (AS IN) MTMOD	R N Z NOTEPT= POINTRW =YES (NOTEPOINT NOT SPECIFIED)	M Z XTENT=MULTPLE XTENT (NOT SPECIFIED SINGLE XTENT IMPLIED)	Z
DAMOD	I	J	I	F B RECFORM=FIXUNB =UNDEF	A I Z AFTER=YES AFTERID=YES (IF NEITHER OF THESE IS SPECIFIED)	I Z IDLOC=YES IDLOC (NOT SPECIFIED)	Z	Z
ISMOD	I	J	H	U F B RECFORM=FIXUNB =FIXBLK =BOTH	L I R A IOROUT=LOAD =ADD =RETRVE =ADDRTR	R S B Z TYPEFLE= RANDOM =SEQNTL =RANSEQ TYPEFLE (NOT SPECIFIED)	R S B Z UPDATE=RANDOM =SEQNTL =RANSEQ UPDATE (NOT SPECIFIED)	W Z WORK=YES WORK (NOT SPECIFIED)



## DTFCD

OPERATION	OPERAND	MUST BE INCLUDED	REMARKS
DTFCD		Each file	Header card. Specify symbolic file name.
	BLKSIZE=n	Each file	Length of I/O areas
	CONTROL= YES	If a CNTRL macro is issued to the file	
	CRDERR= RETRY	To RETRY card output error	For 2520 and 2540 only
	CTLCHR= YES ASA	For first-character control.	YES for S/360 character set. ASA for American Standard Set.
	DEVADDR= SY Snnn	Each file	Specifies symbolic unit
	DEVICE= 2540 1442 2520 2501	For device other than 2540.	To indicate I/O device
	EOFADDR= name	For input or combined file	Specifies end-of-file routine
	IOAREA1= name	Each file	Name=address expression
	IOAREA2= name	Combined file output area	Second output area Name=address expression
	IOREG=(r)	If two I/O areas are used.	Specify r (register 2-12)
	MODNAME= name	If name other than standard or a more inclusive module is referenced	Specifies a user-named I/O module
	OUBLKSZ=n	For a combined file if IOAREA2 is specified.	n=maximum number of characters to be transferred at one time
	RECFORM= FIXUNB UNDEF VARUNB	If other than FIXUNB.	Specifies record format Only FIXUNB valid for input.
	RECSIZE=(r)	For undefined records	For undefined records, r=register containing * length of output record
	SEPASMB= YES	If DTF is assembled separately	
	SSELECT=n	For stacker selection to pocket other than NR or NP	n=stacker select character
	TYPEFLE= INPUT OUTPUT CMBND	If other than input	Specifies an input, output, or combined file
	WORKA=YES		If I/O records are processed in work areas, answer YES

## DTFPR

OPERATION	OPERAND	MUST BE INCLUDED	REMARKS
DTFPR		Each file	Specifies header card. Specifies symbolic file name
	BLKSIZE=n	If other than 121	n=length of I/O area. If the record is not fixed, enter the length of the longest record.
	CONTROL= YES	If a CNTRL macro will be used	
	CTLCHR= YES ASA	For first character control	YES=S/360 character set ASA=American Standard Association set
	DEVADDR= SYSnnn	Each file	Symbolic unit
	DEVICE= 1403 1404 1443 1445	For file other than 1403.	Actual device
	IOAREA1= name	Each file	Specifies name as output area
	IOAREA2= name	If two I/O areas are specified	Second name output area
	IOREG=(r)	For two output areas	r=register 2-12
	MODNAME= name	If name other than standard or a more inclusive module is referenced	Specifies a user-named I/O module
	PRINTOV= YES	If PRTOV macro is used	
	RECFORM= FIXUNB UNDEF VARUNB	If record is not FIXUNB	
	RECSIZE=(r)	For undefined records	r=register 2-12 containing length of output record
	SEPASMB= YES	If DTF is assembled separately	
	WORKA= YES	If records are processed in work areas	

## DTFMT

OPERATION	OPERAND	MUST BE INCLUDED	REMARKS
DTFMT		Each file	Specifies header card and symbolic file name.
	*BLKSIZE=n	Each file	n=length of the I/O area. If the record is not fixed enter the length of the longest record.
	CKPTREC= YES	If input file has checkpoint records among data records	IOCS will bypass checkpoint records.
	*DEVADDR= SY5nnn	Each file	Specifies symbolic unit
	*EOFADDR= name	Input or work files	Specifies name of user's end-of-file routine.
	*ERROPT= IGNORE SKIP name	For error correction	IGNORE allows IOCS to handle the record as if no errors were detected. SKIP allows IOCS to bypass the record. Name is the name of a routine to which the program will branch.
	FILABL= STD NSTD NQ	Each file	Specifies the type of labels.
	IOAREA1= name	Each file	Specifies I/O area. Name=address expression.
	IOAREA2= name	If two I/O areas are specified	Specifies a second I/O area. Name=address expression
	IOREG=(r)		r=register (2-12) if records are blocked or processed in the I/O area.
	LABADDR= name	To process user's labels	name=symbolic name of user's routine
	MODNAME =name	If name other than standard or a more inclusive module is referenced	Specifies a user named I/O module
	*NOTEPT= YES POINTS	YES is required if NOTE, POINTR, POINTW, or POINTS is used	
	*READ= FORWARD BACK	If READ=BACK is specified	
	*RECFORM= FIXUNB FIXBLK VARUNB VARBLK * UNDEF	For other than FIXUNB	
	RECSIZE=n or (r)	For FIXBLK or UNDEF	n=number of characters in each record for fixed-length blocked records. r=register (2-12) containing record length for undefined records.
	*REWIND= UNLOAD NORWD		Specifies procedure for rewinding tape.
	SEPASMB= YES	If DTF is assembled separately	
	*TYPEFLE= INPUT OUTPUT *WORK	For output or work files	Specifies whether file is input or output or if a work file is specified.
	TPMARK= NO	If a tapemark is not to be the first record of a tape output file when no labels are specified, or if no tapemark is to be written following non-standard labels.	
	VARBLD= (r)	For variable-length blocked records.	r=general register (2-12)
	WLRERR= name	To branch to user's routine to handle wrong-length error correction.	name=symbolic name of user's routine. If omitted, error handled as in ERROPT, or if both entries omitted, WLR indication will be ignored.
	WORKA= YES	If records are processed in work areas	

\* Entries for work files

## DTFPT

OPERATION	OPERAND	MUST BE INCLUDED	REMARKS
DTFPT		Each file	Header card. Specify symbolic file name
	BLKSIZE=n	Each file	UNDEF: at least one greater than the longest record FIXUNB: size of every record
	DEVADDR=SYSn	Each file	Symbolic unit
	EOFADDR=Name	Each file	Name of user's EOF routine
	ERROPT=IGNORE SKIP Name		Error option. If absent, IOCS will terminate the job
	FTRANS=Name	For shifted codes	Name of user's figure shift translate table
	IOAREA1=Name	Each file	Name of input area
	IOAREA2=Name	For two input areas	Name of second input area
	IOREG=(r)	For two input areas	Register (2-12) containing current input area address
	LTRANS=Name	For shifted codes	Name of user's letter shift translate table
	MODNAME=Name	For name other than standard	
	OVBLKSZ=n	Only when FIXUNB format is used with SCAN	Number of characters to be read in to produce number specified in BLKSIZE. If omitted, will be equal to BLKSIZE
	RECFORM= <u>FIXUNB</u> <u>UNDEF</u>	If UNDEF format	
	RECSIZE=(r)		Register (2-12) containing final length of input records
	SCAN=Name	When shift and/or delete characters are in code	Name of user's scan table for shift and delete characters
	SEPASMB=YES	If the DTF is assembled separately	
	TRANS=Name	When unshifted codes are to be translated	Name of user's translate table
	WLRERR=Name		Name of user's wrong length record routine. If omitted, error handled as in ERROPT; if both entries omitted, WLR indication will be ignored.

## DTFSD

OPERATION	OPERAND	MUST BE INCLUDED	REMARKS
DTFSD		Each file	Header card, specify symbolic file name.
	*BLKSIZE=n	Each file	Length of I/O area. n=maximum number of characters.
	*CONTROL= YES	CNTRL macro used	
	*DELETFL= NO	If CLOSE macro is not to delete the Format 1 and Format 3 label for a work file	Applies To work files.
	*DEVICE= 2311 2321	For 2321	
	*EOFADDR= name	For input or work files.	Symbolic name of end-of-file routine.
	*ERROPT= IGNORE SKIP name	For error correction on input or output files	Provides facility to handle errors. SKIP not allowed for output.
	IOAREA1= name	Each file	Symbolic name of I/O area.
	IOAREA2= name	If two I/O areas are used by GET and PUT	Symbolic name of I/O area.
	IOREG=(r)	For blocked records processed in the I/O area, or if two I/O areas are used and the records are processed in the I/O areas.	Register 2-12 contains address of the record. r=register 2-12.
	LABADDR= name	If user processes his own labels.	name=symbolic name of user's routine.
	MODNAME= name	If name other than standard or a more inclusive module is referenced.	Specifies a user-named I/O module.
	*NOTEPNT= YES POINTRW	If NOTE, POINTR or POINTW macros will be issued, POINTRW is specified. Specify YES for NOTE, POINTR, POINTW, and POINTS.	
	RECSIZE=n or (r)	Number of characters in record (fixed-length and blocked) or register 2-12 for undefined records.	Required for each file.
	*RECFORM= FIXUNB FIXBLK VARUNB VARBLK * UNDEF	For other than FIXUNB	Work files permit fixed blocked or undefined records only.
	SEPASMB= YES	If DTF is assembled separately	
	TRUNC= YES	With TRUNC macro for Output FIXBLK file, or if FIXBLK DASD files contain shaft blocks within an input file.	
	*TYPEFLE= INPUT OUTPUT * WORK	For OUTPUT or WORK	
	UPDATE= YES	If DASD input file is updated	
	*VERIFY= YES	To check a record written on 2311	Assumed for 2321
	VARBLD=(r)	Variable-length blocked records are built in output area	r=register 2-12
	WLRERR= name	To process wrong length records on input file	name=symbolic name of user's routine
	WORKA= YES		Omit this entry if IOREG=n

## DTFDA

OPERATION	OPERAND#	MUST BE INCLUDED	REMARKS*
DTFDA <sup>+</sup>		Each file	Header Card. Specify symbolic file name (up to 7 ch.)
	AFTER= YES	Record reference AFTER used for an output record	CARREC is assumed.
	<sup>+</sup> BLKSIZE=n	Each file	Length of I/O area. n=maximum number of characters.
	CONTROL= YES	CNTRL macro used	
	<sup>+</sup> DEVICE= <del>2311</del> 2321	Each file	Identifies DASD unit
	<sup>+</sup> ERRBYTE= Name	Each file	Symbolic name of 2-byte field for error/status codes supplied by IOCS.
	IDLOC=Name	ID of same or next record to be supplied by IOCS	Symbolic name of 5-byte field for ID.
	<sup>+</sup> IOAREA1= Name	Each file	Symbolic name of input/output area. Same as used in DS.
	KEYARG= Name	Record reference by Key	Symbolic name of <u>key</u> field.
	KEYLEN=n	Records contain key areas	All keys must be the same length. n=length of keys.
	LABADDR= Name	Check/write additional labels	Symbolic name of user's label routine.
	MODNAME= Name	If a name other than the standard or a more inclusive module is referenced.	
	READID=YES	Record reference by ID used for an input record	
	READKEY= YES	Record reference by key used for an input record	
	RECFORM= FIXUNB UNSEF	Fixed-length records Records not fixed-length, or Records added to a file and EOF record written	If this entry is omitted, IOCS assumes FIXUNB.
	RECSIZE=(r)	Undefined records	r=number of a register 2-12
	<sup>+</sup> SEEKADR= Name	Each file	Symbolic name of track-reference field. Field is 8 bytes long.
	SEPASMB=YES	If the DTF is assembled separately	
	SRCHM=YES	Search multiple tracks	Applies to record reference by key.
	<sup>+</sup> TYPEFLE= INPUT OUTPUT	Each file	Read and check standard labels. Write standard labels.
	VERIFY=YES	Check record written on 2311	Assumed for 2321
	WRITEID= YES	Record reference by ID used for an output record	
	WRITEKY= YES	Record reference by key used for an output record	
	XTNEXIT= Name	To process extent card information	Symbolic name of user's extent routine.

<sup>+</sup>Must be included. Other entries are included when applicable.

#When two choices are shown, select only the appropriate one and enter it after the = sign.

\*The header card and each detail card except the last one used in a file set must contain a continuation punch in column 72. Each detail card except the last one used must also contain a comma after the last operand.

In all entries: Solid caps must be entered as shown (For example, AFTER=YES)

Lowercase letters are to be replaced by programmer's symbolic name or a number (For example, Filename in header card, or BLKSIZE=n where n is replaced).

n is a decimal self-defining value.

ERRBYTE (DTFDA)

Byte	Bit	Error/Status code
0	0	---
	1	Wrong-length record
	2	---
	3	---
	4	No room found
	5	---
	6	---
	7	---
1	0	Data check in count area
	1	Track overrun
	2	End of cylinder
	3	Data check when reading key or data
	4	No record found
	5	End of file
	6	End of volume
	7	---

MACRO INSTRUCTION	ID SUPPLIED	
	With SRCHM	Without SRCHM
READ Filename, KEY	Same record	Next record
READ Filename, ID	Next record	Next record
WRITE Filename, KEY	Same record	Next record
WRITE Filename, ID	Next record	Next record
WRITE Filename, RZERO	None	None
WRITE Filename, AFTER [, EOF]	None	None

ID Supplied After a READ or WRITE Instruction

## DTFIS

OPERATION	OPERAND#	MUST BE INCLUDED	REMARKS*
DTFIS <sup>+</sup>		Each file	Header card. Specify symbolic file name.
	CYLOFL=n	Cylinder overflow areas	May be specified alone or with an independent overflow area. n=number of tracks for each area.
	<sup>+</sup> DSKXTNT=n	Each file	Maximum number of extents specified for the file.
	DEVICE= $\frac{2311}{2321}$	Each file	Specifies Unit
	HINDEX= $\frac{2311}{2321}$		Specifies Unit containing high level indices.
	IOAREAL= Name	IOROUT specifies LOAD, ADD, or ADDRTR	Symbolic name of input/output area. Same as used in DS.
	IOAREAR= Name	TYPEFLE specifies RANDOM or RANSEQ	At least one I/O area must be specified for a file.
	IOAREAS= Name	TYPEFLE specifies SEQNTL or RANSEQ	
	IOREG=(r)	Process blocked records in I/O area	r=number of register 2-12
	<sup>+</sup> IOROUT= LOAD ----- ADD ----- RETRVE ----- ADDRTR	Each file	Build or extend a file on DASD. ----- Insert new records in an organized file. ----- Retrieve records for processing/updating. ----- Insert and retrieve records.
	KEYARG= Name	Required if random retrieval or sequential retrieval is initiated by key.	Symbolic name of <u>key</u> field in main storage.
	<sup>+</sup> KEYLEN=n	Each file	All keys must be the same length. n=length of key.
	KEYLOC=n	With blocked records.	n=high-order position of key field <u>within</u> each record.
	MODNAME= Name	If a name other than the standard or a more inclusive module is referenced.	
	MSTIND= YES	Master index	
	<sup>+</sup> NRECDS=n	Each file	n=number of records in a block.
	<sup>+</sup> RECFORM= FIXUNB ----- FIXBLK	Unblocked records ----- Blocked records	Applies to records in prime data area only.
	<sup>+</sup> RECSIZE=n	Each file	n=number of characters in each logical record.
	SEPASMB= YES	If the DTF is assembled separately.	



OPERATION	OPERAND#	MUST BE INCLUDED	REMARKS*
DTFIS <sup>+</sup>	TYPEFLE= RANDOM ----- SEQNTL ----- RANSEQ	IOROUT specified RETRVE or ADDRTR	Random processing. ----- Sequential processing. ----- Random and sequential processing.
	VERIFY=YES	Check records written on 2311	Assumed for 2321
	WORKL=Name	IOROUT specifies LOAD, ADD, or ADDRTR	
	WORKR=Name	TYPEFLE specifies RANDOM or RANSEQ and records are pro- cessed in a work area	Takes precedence over IOREG.
	WORKS=YES	TYPEFLE specifies SEQNTL or RANSEQ and records are pro- cessed in work areas	Takes precedence over IOREG.

<sup>+</sup>Must be included. Other entries are included when applicable.

#When two choices are shown, select only the appropriate one and enter it after the = sign.

\*The header card and each detail card except the last one used in a file set must contain a continuation punch in column 72. Each detail card except the last one used must also contain a comma after the last operand.

In all entries: Solid caps must be entered as shown (For example, IOROUT=LOAD).

Lowercase letters are to be replaced by programmer's symbolic name or a number (For example, Filename in header card, or CYLOFL=n where n is replaced).

n is a decimal self-defining value.

ADD, RETRVE, and ADDRTR

Bit	Cause
0	DASD error
1	Wrong length record
2	End of file
3	No record found
4	Illegal ID specified
5	Duplicate record
6	Overflow area full
7	Overflow

LOAD

Bit	Cause
0	DASD error
1	Wrong length record
2	Prime data area full
3	Cylinder Index area full
4	Master Index full
5	Duplicate record
6	Sequence check

FUNCTION	OUTPUT AREA REQUIREMENTS (IN BYTES)			
	Count	Key	Sequence Link	Data
Load Unblocked Records	8	Key Length	—	Record Length
Load Blocked Records	8	Key Length	—	Record Length x Blocking Factor
Add Unblocked Records	8	Key Length	10	Record Length
Add Blocked Records	8	Key Length	—	Record Length x Blocking Factor
	8	Key Length	10	Record Length

\* Whichever Is Larger

Output Area Requirements for Loading or Adding Records to a File by ISFMS

FUNCTION	I/O AREA REQUIREMENTS (IN BYTES)			
	Count	Key	Sequence Link	Data
Retrieve Unblocked Records	—	Key Length for sequential unblocked records	10	Record Length
Retrieve Blocked Records	—		—	Record Length x Blocking Factor
	—		10	Record Length

\*Whichever Is Larger

I/O Area Requirements for Random or Sequential Retrieval by ISFMS

DTFSR

OPERATION	OPERAND#	APPLIES TO								MUST BE INCLUDED	REMARKS*
		2311 DISK DRIVE	2400 MAGNETIC TAPE UNIT	1442/2501/2520 2540 READER	1442/2520/2540 PUNCH	1403/1404/1443/1445 PRINTER	1052 PRINTER - KEYBOARD	2671 PAPER TAPE READER			
DTFSR <sup>+</sup>		x	x	x	x	x	x	x		Each File	Header Card. Specify Symbolic File Name.
	ALTTAPE= SYSmnn		x							Multivolume File Using Two Tape Drives	Symbolic Unit for Alternate Tape Drive. (For compatibility purposes)
	BLKSIZE= n	x	x	x	x	x	x	x		Each File Except Combined File with Separate I/O Areas	Length of I/O Area. n=Maximum Number of Characters.
	CHECKPT= n		x							CHKPT Macro Used	(For compatibility purposes)
	CKPTREC= YES		x							Bypass Checkpoint Records on Input	Applies to Input File Only.
	CONTROL= YES	x	x	x	x	x				CNTRL Macro Used	CTLCHR Must Be Omitted. Does Not Apply to 2501.
	CRDERR= RETRY				x					Punch Again on Error Condition	Applies Only to 2540 and 2520
	CTLCHR= YES				x	x				Logical Records Have Control Character in First Position	Each Record Must Contain a Control Character. CONTROL Must Be Omitted and the CNTRL Macro Must Not Be Used for This File.
	DEVADDR= SYSmnn	x	x	x	x	x	x	x		Each File Except Disk	Symbolic Unit for the I/O Device Used for the File.
	+DEVICE= DISK11	x								Disk Input/Output File	Include for Each File, and Specify Proper Name after DEVICE=.
	TAPE		x							Tape Input/Output File	
	READ01			x						2501 Input File	
	READ20			x	x					2520 Input/Output File	
	READ40			x	x					2540 Input/Output File	
	READ42			x	x					1442 Input/Output File	
	PRINTER						x			1403, 1404, 1443, 1445 Output	
	CONSOLE							x		1052 Input/Output	
	PTAPERD								x	2671 Input File	
	EOFADDR= Name	x	x	x					x	Input File	
	ERROPT= IGNORE	x	x						x	Process Error Records	Applies Only to Disk or Tape Input.
	SKIP									Skip over Error Records	Prevents Job Termination on Error
	Name									User Routine for Error Records	Condition. Enter Desired Specification after ERROPT=.

OPERATION	OPERAND#	APPLIES TO								MUST BE INCLUDED	REMARKS*		
		2311 DISK DRIVE	2400 MAGNETIC TAPE UNIT	1442/2501/2520/ 2540 READER	1442/2520/2540 PUNCH	1403/1404/1443/ 1445 PRINTER	1052 PRINTER - KEYBOARD	2671 PAPER TAPE READER					
DTFSR <sup>+</sup>	FILABL=STD		x								Check or Write Standard Labels	Include for Tape Input/Output and Specify Desired Operation after FILABL=.	
	NSTD										File Contains Nonstandard Labels		
	NO										Unlabelled File		
	INAREA=Name			x								Separate Areas for Input and Output for a Combined File	Applies Only to 1442.
	INBLKSZ=n			x								Separate Areas for Input and Output for a Combined File	Applies Only to 1442. Length of INAREA.
	IOAREA1=Name	x	x	x	x	x	x	x				Each File Except Combined File with Separate I/O Areas	Symbolic Name of Input/Output Area.
	IOAREA2=Name	x	x	x	x	x	x	x				Two I/O Areas Used	Symbolic Name of Second Input/Output Area.
	IOREG=(r)	x	x	x	x	x	x	x				Process Blocked Records in I/O Area, or Process in Two I/O Areas	r=number of General Purpose register 2-12. Omit WORKA=YES.
	LABADDR=Name	x	x									Check/Build Additional User-Standard Labels, or Process Nonstandard Labels	Symbolic Name of User's Label Routine.
	OUAREA=Name			x								Separate Areas for Input and Output for a Combined File	Applies Only to 1442.
	OUBLKSZ=n			x								Separate Areas for Input and Output for a Combined File	Applies Only to 1442. Length of OUAREA.
	PRINTOV=YES								x			PRTOV Macro Used	
	READ=FORWARD			x									If This Entry Omitted, IOCS Assumes FORWARD.
	BACK											Read Tape Backwards	
RECFORM=FIXUNB	x	x	x	x	x	x	x					Specify as Needed for Fixed-Length Unblocked Records. If This Entry Omitted, IOCS Assumes FIXUNB.	
FIXBLK	x	x									Fixed-Length Blocked Records		
VARUNB	x	x			x		x				Variable-Length Unblocked Records	Disk or Tape Records Require Record-Length Field.	
VARBLK	x	x									Variable-Length Blocked Records	Blocks Require Block-Length Field. Records Require Record-Length Field.	

OPERATION	OPERAND#	APPLIES TO								MUST BE INCLUDED	REMARKS*
		2311 DISK DRIVE	2400 MAGNETIC TAPE UNIT	1442/2503/2520/2540 READER	1442/2520/2540 PUNCH	1403/1404/1443/1445 PRINTER	1052 PRINTER-KEYBOARD	2671 PAPER TAPE READER			
DTFSR <sup>+</sup>	UNDEF	x	x		x	x	x	x		Undefined Records	
	RECSIZE= n or (r)	x	x		x	x	x	x		Fixed-Length Blocked Records Undefined Records	n=Number of Characters in Record. r=Number of register 2-12.
	REWIND= UNLOAD ----- NORWD		x							Unload on CLOSE or End-of-Volume Prevent Rewinding	Omit to Rewind Only at OPEN, CLOSE, or End-of-Volume.
	TPMARK= NO		x							If FIL ABL=NO or NSTD and No Tape Mark is Desired	If Omitted, YES is Assumed.
	TRANS= Name								x	For Tape Punched With Code Other than EBCDIC and IOCS is to Perform Translation	Symbolic Name of Code Translation Table.
	TRUNCS= YES	x								Fixed-Length Blocked Records with Short Blocks	Include for Output if TRUNC Macro Used. Include for Input if TRUNC Macro Was Used When File Was Created.
	TYPEFLE= INPUT ----- OUTPUT ----- CMBND	x	x	x				x	x	Each File INPUT is assumed	Specify Proper Type after TYPEFLE=. CMBND, Applies to 1442, 2520, or to 2540 if Punch-Feed-Read Special Feature is Installed.
	UPDATE= YES	x								PUT Used For a Disk Input File	
	VARBLD= (r)	x	x							Variable-Length Blocked Records Built in Output Area	r=Number of a register 2-12.
	VERIFY= YES	x								Check Record Written on Disk	
	WLRERR= Name	x	x						x	User Routine for Wrong-Length Records	Symbolic Name of User's Routine. If Omitted, Error Handled As in ERROPT; or if Both Entries Omitted; WLR indication will Be Ignored.
**WORKA= YES	x	x	x	x	x	x			GET or PUT Specifies a Work Area	Omit IOREG=(r)	

+Must be included. Other entries are included when applicable.

#When two or more choices are shown, select only the appropriate one and enter if after the = sign.

\*The header and each detail card except the last one used in a file set must contain a continuation punch in column 72. Each detail card except the last one used must also contain a comma after the last operand.

\*\*The 16K DTFSR does not support a work area when the 2671 Paper Tape Reader is specified as the device.

In all entries: Solid caps must be entered as shown (For example, CONTROL=YES)

Lowercase letters are to be replaced by programmer's symbolic name or a number (For example, Filename in header card, or BLKSIZE=n where n is replaced).

n is a decimal self-defining value.

DTFPH

NAME	OPERATION	OPERAND	APPLIES TO							MUST BE INCLUDED	REMARKS	
			2311 DISK DRIVE	2400 MAGNETIC TAPE UNIT	1442/2501/2520 2540 READER	1442/2520/2540 PUNCH	1403/1404/1443 1445 PRINTER	1052 PRINTER- KEYBOARD	2321 Data Cell Drive			
Filename	DTFPH		X	X						X	Labeled DASD/Tape File	Symbolic File Name.
		DEVICE= { Tape 2311 2321 }	X	X						X	Each File	
		DEVADDR= SY Snn			X						Labeled Tape File	Symbolic Unit for the Device Used for the File.
		LABADDR= Name	X	X						X	Check/Build Additional User-Standard Labels	Symbolic Name of User's Label Routine. For Tape Input and DASD Files, Applies to Header Labels Only.
		MOUNTED= ALL  SINGLE	X							X	Each DASD File	All Extents Are to Be Available at the Initial OPEN. Only the First Extent Is to Be Available at the Initial OPEN.
		TYPEFLE= INPUT OUTPUT	X	X						X	Labeled DASD/Tape Input File Labeled DASD/Tape Output File	
		XTNTXIT= Name	X							X		Symbolic Name of User's Extent Routine.

DTFCN

OPERATION	OPERAND	MUST BE INCLUDED	REMARKS.
DTFCN		Yes	Specify the header entry. Specify symbolic file name.
	BLKSIZE=n		Length of I/O area. n=actual length.
	DEVADDR= (SY SLOG) (SY Snnn)	Yes	Symbolic unit.
	IOAREA1=name	Each file	Specifies the symbolic name of the I/O area.
	RECFORM= (FIXUNB) (UNDEF)		Specifies the record format.
	RECSIZE=(r)	For each undefined output file	Indicates register containing length of each record.
	TYPEFLE= (INPUT) (OUTPUT)	Yes	Specifies the type of file.
	WORKA=YES		Specifies that a work area will be used.

DIRECT ACCESS FILE

Name	Operation	Operand
(name)	READ	Filename, KEY
(name)	READ	Filename, ID
(name)	WRITE	Filename, KEY
(name)	WRITE	Filename, ID
(name)	WRITE	Filename, AFTER
(name)	WRITE	Filename, AFTER, EOF
(name)	WRITE	Filename, RZERO
(name)	WAITF	Filename
(name)	CNTRL	Filename, code

INDEXED SEQUENTIAL FILE

LOAD

Name	Operation	Operand
(name)	SETFL	Filename
(name)	WRITE	Filename, NEWKEY
(name)	ENDFL	Filename

ADD

Name	Operation	Operand
(name)	WRITE	Filename, NEWKEY
(name)	WAITF	Filename

RANDOM RETRIEVAL

Name	Operation	Operand
(name)	READ	Filename, KEY
(name)	WRITE	Filename, KEY
(name)	WAITF	Filename

SEQUENTIAL RETRIEVAL

Name	Operation	Operand
(name)	SETL	Filename, { ID Name KEY BOF GKEY }
(name)	GET	Filename
(name)	GET	Filename, Workname
(name)	PUT	Filename
(name)	PUT	Filename, Workname
(name)	ESETL	Filename

UTILITIES

JOB CONTROL

	File to File Programs	Tape Compare Program	Clear Disk	Clear Data Cell
JOB	Required	Required	Required	Required
LBLTYP(DOS) NMTLB(TOS)	If tape label checking	Not Used	Not Used	Not used
VOL	For label	Not Used	Required	Required
TPLAB	For tape label	Not Used	Not Used	Not Used
DLAB	For DASD label	Not Used	For DASD label	For DASD label
XTENT	For DASD	Not Used	Required for DASD	Required for DASD
ASSGN	Required if devices are different from those assigned at IPL time	Required if devices are different from those assigned at IPL time	Required if devices are different from those assigned at IPL time	Required if devices are different from those assigned at IPL time
UPSI	Optional	Optional	Optional	Optional
EXEC	Required	Required	Required	Required
/*	For card input program*	Not Used	Not Used	Not Used
/&	Required	Required	Required	Required

\* This card must immediately follow the data cards for card input programs. In addition, card columns 3-80 of the card must be entirely blank, otherwise the card will be ignored and treated as data.

	File to File Programs	Tape Compare Program	Clear Disk	Clear Data Cell
VOL (filename)	UIN, UOUT	Not Used	UOUT	UOUT
ASSGN device for logging operator messages	SYSLOG	SYSLOG	SYSLOG	SYSLOG
ASSGN utility control statement input device	SYSIPT	SYSIPT	SYSIPT	SYSIPT
ASSGN device for logging programmer messages	SYSLST	SYSLST	SYSLST	SYSLST
ASSGN Primary tape and card input and alternate tape input	SYS004	SYS004	Not Used	Not Used
ASSGN Primary tape and printer output and alternate tape output	SYS005	SYS005	Not Used	Not Used
ASSGN linkage editor*	SYS000 SYS001 SYS002	SYS000 SYS001 SYS002	SYS000 SYS001 SYS002	SYS000 SYS001 SYS002
ASSGN card output	SYS006	Not Used	Not Used	Not Used
ASSGN DASD input and or output device**	SYS000- SYSnnn	Not Used	SYS000- SYSnnn	SYS000- SYSnnn

\* These units are available when not in use by linkage editor.  
 \*\* SYSnnn can be no greater than the greatest physical unit block assigned and must not conflict with the assignment of any other device.



Program	NAME	Assumed Modifier Card
Tape to Tape	TFTP	//bU <sub>b</sub> TC, FU, A=(1000), B=(1000), IU, OU, RI
Tape to Disk	TPDK	//bU <sub>b</sub> TC, FU, A=(1000), B=(1000), IU, OY, RI
Tape to Data Cell	TPDC	//bU <sub>b</sub> TC, FU, A=(1000), B=(1000), IU, OY, RI
Tape to Card	TPCD	//bU <sub>b</sub> TC, FF, A=(80, 80), B=(80, 80), IU, O1, R1, S2
Tape to Printer	TPPT	//bU <sub>b</sub> TD, FU, A=(1000), B=(120), IU, OX, PY, R1, S1
Disk to Tape	DKTP	//bU <sub>b</sub> TC, FU, A=(1000), B=(1000), OU, R1
Disk to Disk	DKDK	//bU <sub>b</sub> TC, FU, A=(1000), B=(1000), OY, R1
Disk to Data Cell	DKDC	//bU <sub>b</sub> TC, FU, A=(1000), B=(1000), OY, R1
Disk to Card	DKCD	//bU <sub>b</sub> TC, FF, A=(80, 80), B=(80, 80), O1, R1, S2
Disk to Printer	DKPT	//bU <sub>b</sub> TD, FU, A=(1000), B=(120), OX, S1, PY, R1
Data Cell to Tape	DCTP	//bU <sub>b</sub> TC, FU, A=(1000), B=(1000), OU, R1
Data Cell to Disk	DCDK	//bU <sub>b</sub> TC, FU, A=(1000), B=(1000), OY, R1
Data Cell to Data Cell	DCDC	//bU <sub>b</sub> TC, FU, A=(1000), B=(1000), OY, R1
Data Cell to Printer	DCPT	//bU <sub>b</sub> TD, FU, A=(1000), B=(120), OX, S1, PY, R1
Card to Tape	CDTP	//bU <sub>b</sub> TC, FF, A=(80, 80), B=(80, 80), I1, OU, R1
Card to Disk	CDDK	//bU <sub>b</sub> TC, FF, A=(80, 80), B=(80, 80), I1, OY, R1
Card to Printer and/or punch	CDPP (PUNCH)	//bU <sub>b</sub> TB, FF, A=(80, 80), B=(80, 80), I1, O1, S2, R1
	(PRINT)	//bU <sub>b</sub> TB, FF, A=(80, 80), B=(120), I1, OC, S2, R1
Clear Disk	CLRDSK	//bUCLbB=(K=0, D=100), X'00', OY

#### LABEL CHECKING

- Bit 0 Off for standard input-label checking; on for nonstandard or no input-label checking.
- Bit 1 Off if not doing user input-label checking; on if user input-label checking.
- Bits 2 and 3 are switches for output-label checking.
- Bit 2 Off for standard output-label checking; on for nonstandard or no output-label checking.
- Bit 3 Off if not user output-label checking; on if user output-label checking.
- Bit 4 is for nonstandard or no output-label handling.
- Bit 4 Off = Write tape mark separating the label from data.  
On = Do not write a tape mark to separate the label from the data.

A user label routine must be supplied only if bits 1 or 3 of the UPSI byte are ON.

PARAMETER	POSSIBLE FORMS	ENTRIES	EXPLANATION
Function Tt	TC TF TR TRF TD TL TLF TB TBF	T C F R RF D L LF B BF	The initial T identifies this as the type of function parameter. Copy Field Select Reblock Reblock and Field Select Display List List and Field Select Both print and punch Both print and punch with field select.
Format Ff	FF FV FU	F V U	Fixed-length records. Variable-length records. Undefined records.
Input Description	A=(n, m)  A=(K=1, D=1)  A=(g)	(n, m)  A= (K=1, D=1)  A= (g)	For fixed-length input records, the input record length (the letter n) and the input block length (the letter m). For field select with variable-length input records, the fixed portion of each input record (the letter n) and the maximum block length (the letter m) must be enclosed in parentheses and separated by a comma.  This letter and symbol indicate this is the input-description parameter.  For fixed-length DASD input records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field.  This letter and symbol indicate this is the input-description parameter.  For undefined input records or variable input records without field select.
Output Description	B=(n, m)  B=(K=1, D=1)  B=(g)	(n, m)  (K=1, D=1)  (g)	For fixed-length output records, the output record length (the letter n) and the output block length (the letter m). For field select with variable-length output records, the fixed portion of each output record (the letter n) and the maximum output block length (the letter m) must be enclosed in parentheses and separated by a comma.  For fixed-length DASD output records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses.
	B=(p)  B=(n, p)	(g)  (p)  (n, p)	For undefined output records or variable input records without field select, the maximum block length must be enclosed in parentheses.  For printer output the size of the print line (120, 132, 144) must be entered.  For field select of variable length records with printer output records, the fixed portion of each output record (the letter n) and the size of the print line (the letter p) must be enclosed in parentheses and separated by a comma.
Optional	Ix Ox Sx Px Rx Q=(x, y)		These parameters are unique to each program and are explained under the discussions of the individual programs.

Utility-Modifier Statement Parameters

CARD TO PRINTER AND/OR PUNCH (CDPP)

Card to Punch:

//bUCPbTt, FF, A=(n, m), B=(a, b), Ix, Ox, Rx, Sx, Q=(x, y)

Card to Printer:

//bUCPbTt, FF, A=(n, m), B=(p), Ix, Ox, Rx, Sx, Q=(x, y)

Card to Printer and Punch:

//bUCPbTt, FF, A=(n, m), B=(a, b), Ix, Px, Rx, Sx, Q=(x, y)

Card Input Ix	I1 I2	1 2	EBCDIC input. Binary input.
Printer or Punch Output Ox	O1 O2 Ox OC	1 2 X C	EBCDIC output (punch only). Binary output (punch only). Hexadecimal output (printed only). Character output (printer only).
Page Numbering Px	PY PN	Y N	Number pages. Do not number pages.
Sequence numbering Q=(x, y)	Q=(x, y)	x ' y	This represents the position of a field in a card (relative to one) for sequence-numbering (1 or 2 digits). Separator. This represents the length of the field (maximum 10).
First Record Rx	Rx	x	(x-1 records will be bypassed)
Spacing and Stacker Control Sx	S1 S2 S3	1 2 3	Printer output; Single spacing. Punch output; Select stacker 1. Printer and Punch; Printer control only. Printer output; Double spacing. Punch output; Select stacker 2. Printer and Punch; Printer control only. Printer output; Triple spacing. Punch output; Invalid. Printer and Punch; Printer control only.

CARD TO DISK (CDDK)

//bUCDbTt, FF, A=(input), B=(output), Ix, Ox, Q=(x, y), Rx

Card Input Ix	I1 I2	1 2	EBCDIC input. Binary input.
Disk Check Ox	OY ON	Y N	Write-disk check. Do not write-disk check.
Sequence- numbering Q=(x, y)	Q=(x, y)	x ' y	This represents the first position of a field in a card (relative to one) for sequence-numbering (1 or 2 digits). Separator. This represents the length of the field (maximum 10).
First Record Rx	Rx	x	(x-1 records will be by passed).

CARD TO TAPE (CDTP)

//bUCTbTt, Ff, A=(input), B=(output), Ix, Rx, Ox, Q=(x, y)

Card Input Ix	I1 I2	1 2	EBCDIC input Binary input.
Rewind Output Ox	OR ON OU	R N U	Rewind both before and after data transfer. Do not rewind either before or after data transfer. Rewind before and rewind and unload after data transfer.
Sequence- numbering Q=(x, y)	Q=(x, y)	x , y	This represents the first position of a field in a card (relative to one) for sequence-numbering (1 or 2 digits). Separator. This represents the length of the field (maximum 10).
First Record Rx	Rx	x	(x-1 records will be bypassed).

TAPE TO PRINTER (TPPT)

//bUTPbTt, Ff, A=(input), B=(output), Ix, Ox, Px, Rx, Sx

Rewind Input Ix	IR IN IU	R N U	Rewind both before and after data transfer. Do not rewind either before or after data transfer. Rewind before and rewind and unload after data transfer.
Print Output Ox	OX OC	X C	Hexadecimal printout. (For data display only) Character printout. (Forced for data list)
Spring Option Sx	S1 S2 S3 SA SB SC SD	1 2 3 A B C D	Single spacing (Forced for data display) Double spacing. Triple spacing. Type A first character forms control. Type B first character forms control. Type C first character forms control. Type D first character forms control.
Page Numbering Px	PY PN	Y N	Number pages. (Forced for data display) Do not number pages. (Forced for first character forms control)

TAPE TO DISK (TPDK)

//bUTDbTt, Ff, A=(input), B=(output), Ix, Ox, Rx

Rewind Input Ix	IR IN IU	R N U	Rewind both before and after data transfer. Do not rewind either before or after data transfer. Rewind and unload both before and after data transfer.
Disk Check Ox	OY ON	Y N	Write-disk check. Do not write-disk check.
First Record Rx	Rx	x	(x-1 records will be bypassed).

TAPE TO CARD (TPCD)

//bUTCbTt, FF, A=(input), B=(output), Ix, Ox, Rx, Sx, Q=(x,y)

Rewind Input Ix	IR IN IU	I	The first letter in these forms identifies this parameter.
		R	Rewind both before and after data transfer.
		N	Do not rewind either before or after data transfer.
		U	Rewind before and rewind and unload after data transfer.
Sequence Numbering Q=(x,y)	Q=(x,y)	Q=	The letter and symbol identify this parameter.
		x	This represents the first position of a field in a card (relative to one) for sequence-numbering (1 or 2 digits).
		y	This represents the length of the field (maximum 10). The (x,y) parts of this parameter must be enclosed in parentheses. Absence of this parameter indicates no sequence numbers.
First Record Rx	Rx	R	The first letter in this form identifies this parameter.
		x	This represents the position of the first logical input record to be output (x-1 records will be bypassed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values.
Stacker Control Sx	S1 S2 S3	S	The first letter in these forms identifies this parameter.
		1	Select pocket 1
		2	Select pocket 2
		3	First character stacker control.

TAPE TO TAPE (TPTP)

//bUTTbTt, Ff, A=(input), B=(output), Ix, Ox, Rx

Rewind Option for input Ix	IR	R	Rewind both before and after data transfer.
		N	Do not rewind either before or after data transfer.
		U	Rewind before and rewind and unload after data transfer.
First Record Rx	Rx	x	(x-1 records will be bypassed.)
Rewind Output Ox	OR ON OU	R	Rewind both before and after data transfer.
		N	Do not rewind either before or after data transfer.
		U	Rewind before and rewind and unload after data transfer.

DISK TO CARD (DKCD)

//bUDCbTt, Ff, A=(input), B=(output), Ox, Rx, Sx, Q=(x,y)

Output Mode Ox	O1 O2	1	EBCDIC punching
		2	Binary punching
Sequence- Numbering Q=(x,y)	Q=(x,y)	Q=	The first letter and symbol identify this parameter.
		x	This represents the first position of a field in a card (relative to one) for sequence-numbering (1 or 2 digits).
		y	This represents the length of the field (maximum 10).
First Record Rx	Rx	x	(x-1 records will be bypassed.)
Stacker Select Sx	S1 S2 S3	1	Select pocket 1
		2	Select pocket 2
		3	First character stacker select

DISK TO TAPE (DKTP)

//bUDTbTt, Ff, A=(input), B=(output), Ox, Rx

Rewind Output Ox	OR ON OU	R N U	Rewind both before and after data transfer. Do not rewind either before or after data transfer. Rewind before and rewind and unload after data transfer.
First Record Rx	Rx	x	(x-1 records will be bypassed.)

DISK TO DISK (DKDK)

//bUDDbTt, Ff, A=(input), B=(output), Ox, Rx

Disk Check Ox	OY ON	Y N	Write-disk check. Do not write-disk check.
First Record Rx	Rx	x	(x-1 records will be bypassed.)

DISK TO PRINTER (DKPT)

//bUDPbTt, Ff, A=(input), B=(output), Ox, Sx, Px, Rx

Printer Output Ox	OX OC	X C	Hexadecimal printout. (For data display only). Alphameric printout. (Forced for data list mode).
Page-numbering Px	PY PN	Y N	Number pages (Forced for data display). Do not number pages. (Forced for first character forms control).
First Record Printed Rx	Rx	x	(x-1 records will be bypassed.)
Spacing Sx	S1 S2 S3 SA SB SC SD	1 2 3 A B C D	Single spacing. (Forced for data display) Double spacing. Triple spacing. Type A first character forms control. Type B first character forms control. Type C first character forms control. Type D first character forms control.

CLEAR DISK (CLRDSK)

//bUCLbB=(K=1, D=1), [C'c' / X'xx' ], O [Y / N]

Parameter	Entry	Explanation
B=(K=1, D=1)	B=	Identifies this parameter.
	(K=1, D=1)	Indicates the length of the key and data block in bytes. If a key length is not desired, the key length must be zero.
C'c' or	C'c'	C is entered and followed by the fill character (EBCDIC) enclosed in apostrophes.
X'xx'	X'xx'	The letter X is entered and followed by the hexadecimal fill character enclosed in apostrophes.
OY or	O	Identifies this as the output parameter.
ON	Y N	Indicates write-disk check. Indicates do not write-disk check.

## SPOOL UTILITIES

OPERATION	OPERAND	MUST BE INCLUDED	REMARKS
INCARD	[,UNIT= {SYS001 SYSnnn}]	If other than SYS001 is wanted.	nnn - the programmer logical unit containing input. A 160 byte buffer is generated.
INTAPE	BUFSIZ=n	For each file.	n=maximum size of input block. A buffer of 2n bytes is generated.
	[,RECSIZ= {m name1}]	If logical record size within blocks is to be specified.	m=logical record size, name1=symbolic location containing logical record size. If omitted or value=0, records are treated as unblocked.
	[,UNIT= {SYS001 SYSnnn}]	If other than SYS001 is wanted.	nnn - the programmer logical unit containing input.
	[,FILE= {filename (r)}]	For each file bearing standard labels.	Identifies the filename for label processing. r=register pointing to the 7 character filename.
	[,LBL=name2]	If special label processing is desired.	name2=full-word area to be interpreted. If value=0, standard labels are assumed and user label are bypassed. If value is positive, value=address of user HDR record processing routine. If value=negative, or the parameter is omitted no label processing is attempted.
	[,ERROR= {SKIP IGNORE name3}]	If read errors are not to be entirely ignored.	SKIP=bypass any unreadable block. IGNORE=ignore error condition. name3=symbolic address of a user's routine for unreadable block processing.
INDISK	[,CHKPT= {name4 NO}]	If checkpoint records are to be treated as data.	name4=full-word area to be interpreted. If value=non-zero, checkpoint will be treated as data. If value=zero or the parameter is omitted, checkpoint records are bypassed. If CHKPT=NO, checkpoint records are treated as data.
	BUFSIZ=n	For each file.	n=maximum size of input block. A buffer of 2n bytes is generated.
	[,RECSIZ= {m name1}]	If logical record size within blocks is to be specified.	m=logical record size name1=symbolic address containing logical record size. If omitted or value=0, records are treated as unblocked.
	[,FILE= {filename (r)}]	For each file.	Identifies the filename for label processing. r=register containing 8 character filename.
	[,LBL=name2]	If additional user label processing is desired.	name2=full-word area (designated name2) to be interpreted. If value=0, additional user labels are treated like data. Otherwise, value is address of user's HDR label processing routine.
[,ERROR= {SKIP IGNORE name3}]	If read errors are not to be entirely ignored.	SKIP=bypass any unreadable block. IGNORE=ignore error condition. name3=symbolic address of a user's routine for unreadable block processing.	
INLOG	BUFFER= {name1 (r <sub>b</sub> )}	For each input via the 1052.	name1=symbolic location of the user's buffer area. r <sub>b</sub> =register containing the buffer address.

OPERATION	OPERAND	MUST BE INCLUDED	REMARKS
INLOG	,COUNT= $\left\{ \begin{matrix} n \\ (r_c) \end{matrix} \right\}$	If BUFFER=(r <sub>b</sub> ).	b=number of characters to be read. r <sub>c</sub> =register containing the number of characters to be read.
OUTCARD	[UNIT= $\left\{ \begin{matrix} \text{SYS002} \\ \text{SYSnnn} \end{matrix} \right\}$ ]	If other than SYS002 is wanted.	nnn=programmer logical unit for output. A buffer of 160 bytes is generated for the 1442 and 2520 (240 bytes for the 2540).
	[STCTL= $\left\{ \begin{matrix} \text{NO} \\ \text{YES} \\ \text{name1} \end{matrix} \right\}$ ]	If stacker selection is desired.	Omission, or NO=pocket2; YES=stacker selection is controlled by first character of each record; name1=symbolic location of area to be interpreted: 0=pocket 2; nonzero=YES.
	[,DEVICE= $\left\{ \begin{matrix} 1442 \\ 2520 \\ 2540 \end{matrix} \right\}$ ]	If other than a 1442 card punch is used.	Specifies error-recovery procedures.
	[,RETURN= $\left\{ \begin{matrix} \text{YES} \\ \text{NO} \end{matrix} \right\}$ ]	If user routine or additional output utility macros follow.	NO=continue to next sequential statement. YES=return to input routine or to optional user routine.
OUTAPE	BUFSIZ=n	For each file.	n=maximum size of output block. A buffer of 2n bytes is generated.
	[,BLK= $\left\{ \begin{matrix} m \\ \text{name1} \end{matrix} \right\}$ ]	If record blocking is desired.	m=blocking factor; value of 0 indicates unblocked records. name1=symbolic location to be interrogated. A value of 0 indicates no blocking is performed. Otherwise value=blocking factor.
	[,UNIT= $\left\{ \begin{matrix} \text{SYS002} \\ \text{SYSnnn} \end{matrix} \right\}$ ]	If other than SYS002 is wanted.	nnn=programmer logical unit for output.
	[,FILE= $\left\{ \begin{matrix} \text{filename} \\ (r) \end{matrix} \right\}$ ]	For each file bearing standard labels.	Identifies label information. r=register pointing to an 8 character filename.
	,LBL=[name2]	If label processing is desired.	name2=fullword area symbolically designated name2 to be interpreted. If value=0, standard labels are assumed and user labels are bypassed; if value is negative, no label processing is attempted; if value is positive, value=address of HDR processing routine.
	[,RETURN= $\left\{ \begin{matrix} \text{YES} \\ \text{NO} \end{matrix} \right\}$ ]	If user routine or additional output utility macros follow.	NO=continue to next sequential statement. YES=return to input routine or to optional user routine.
OUTDISK	BUFSIZ=n	For each file.	n=maximum size of output block. A buffer of 2n bytes is generated.
	[,BLK= $\left\{ \begin{matrix} m \\ \text{name1} \end{matrix} \right\}$ ]	If record blocking is desired.	m=blocking factor; value of 0 indicates unblocked records. name1=symbolic location to be interrogated. A value of 0 indicates no blocking is performed. Otherwise value=blocking factor.
	FILE= $\left\{ \begin{matrix} \text{filename} \\ (r) \end{matrix} \right\}$	For each file.	Identifies label information. r=register pointing to an 8 character filename.



OPERATION	OPERAND	MUST BE INCLUDED	REMARKS
OUTDISK (Cont'd)	[,LBL=name2]	If user label processing is desired.	name2=fullword area symbolically designated name2 to be interpreted. If value=0, no user HDR label processing is attempted; if value is nonzero, value=address of HDR record processing routine.
	[,ERROR= $\left\{ \begin{array}{l} \text{IGNORE} \\ \text{name3} \end{array} \right\}$ ]	If a write check is desired.	IGNORE=ignore error condition. name3=symbolic address of user routine for error-handling.
	[,FORMAT= $\left\{ \begin{array}{l} \text{FULL} \\ n \\ \text{name4} \end{array} \right\}$ ]	When a specified number of records are to appear on each track.	FULL=fill track as much as possible. n=number of records per track. name4=symbolic address containing number of records per track value. If value is 0, FULL is assumed.
	[,RETURN= $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$ ]	If user routine or additional output utility macros follows.	NO=continue to next sequential statement. YES=return to input routine or to optional user routine.
OUTPRT	BUFSIZ=n	If n should be less than 144.	n=maximum size of output block.
	[,UNIT= $\left\{ \begin{array}{l} \text{SYS002} \\ \text{SYSnnn} \end{array} \right\}$ ]	If other than SYS002 is wanted.	nnn=programmer logical unit for output.
	[,FORMS= $\left\{ \begin{array}{l} \text{A} \\ \text{B} \\ \text{C} \\ \text{D} \end{array} \right\}$ ]	If other than single-spaced lines are desired.	Refer to the description of carriage control codes provided with macro description.
	[,RETURN= $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$ ]	If user routine or additional output utility macros follow.	NO=continue to next sequential statement. YES=return to input routine or to optional user routine.
OUTLOG	BUFFER= $\left\{ \begin{array}{l} \text{name1} \\ (\text{r}_b) \end{array} \right\}$	For each output on the 1052.	name1=symbolic location of the user's buffer area. r <sub>b</sub> =register containing the buffer address (absolute or symbolic).
	[,COUNT= $\left\{ \begin{array}{l} n \\ (\text{r}_c) \end{array} \right\}$ ]	If Buffer= [r <sub>b</sub> ]	n=number of characters to be written. r <sub>c</sub> =register containing the number of characters to be written.
	[,RETURN= $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$ ]	If user routine or additional output utility macros follow.	NO=continue to next sequential statement. YES=return to input routine or to optional user routine.

Code

blank  
0  
-  
+ (EBCDIC or BCDIC)  
1  
2  
3  
4  
5  
6  
7  
8  
9  
A  
B  
C

Space or Skip Action

Space one line before printing  
Space two lines before printing  
Space three lines before printing  
Suppress space before printing  
Skip to Channel 1 before printing  
Skip to Channel 2 before printing  
Skip to Channel 3 before printing  
Skip to Channel 4 before printing  
Skip to Channel 5 before printing  
Skip to Channel 6 before printing  
Skip to Channel 7 before printing  
Skip to Channel 8 before printing  
Skip to Channel 9 before printing  
Skip to Channel 10 before printing  
Skip to Channel 11 before printing  
Skip to Channel 12 before printing

Type A

<u>8-Bit Code</u>	<u>Punch Combination</u>	<u>Function</u>
00000001	12, 9, 1	Write (no automatic space)
00001001	12, 9, 8, 1	Write and space 1 line after printing
00010001	11, 9, 1	Write and space 2 lines after printing
00011001	11, 9, 8, 1	Write and space 3 lines after printing
10001001	12, 0, 9	Write and skip to channel 1 after printing
10010001	12, 11, 1	Write and skip to channel 2 after printing
10011001	12, 11, 9	Write and skip to channel 3 after printing
10100001	11, 0, 1	Write and skip to channel 4 after printing
10101001	11, 0, 9	Write and skip to channel 5 after printing
10110001	12, 11, 0, 1	Write and skip to channel 6 after printing
10111001	12, 11, 0, 9	Write and skip to channel 7 after printing
11000001	12, 1	Write and skip to channel 8 after printing
11001001	12, 9	Write and skip to channel 9 after printing
11010001	11, 1	Write and skip to channel 10 after printing
11011001	11, 9	Write and skip to channel 11 after printing
11100001	11, 0, 9, 1	Write and skip to channel 12 after printing
00001011	12, 9, 8, 3	Space 1 line immediately
00010011	11, 9, 3	Space 2 lines immediately
00011011	11, 9, 8, 3	Space 3 lines immediately
10001011	12, 0, 8, 3	Skip to channel 1 immediately
10010011	12, 11, 3	Skip to channel 2 immediately
10011011	12, 11, 8, 3	Skip to channel 3 immediately
10100011	11, 0, 3	Skip to channel 4 immediately
10101011	11, 0, 8, 3	Skip to channel 5 immediately
10110011	12, 11, 0, 3	Skip to channel 6 immediately
10111011	12, 11, 0, 8, 3	Skip to channel 7 immediately
11000011	12, 3	Skip to channel 8 immediately
11001011	12, 0, 9, 8, 3	Skip to channel 9 immediately
11010011	11, 3	Skip to channel 10 immediately
11011011	12, 11, 9, 8, 3	Skip to channel 11 immediately
11100011	0, 3	Skip to channel 12 immediately
00000011	12, 9, 3	No op

Type B

d immediate skip to

- 1 channel 1
- 2 channel 2
- 3 channel 3
- 4 channel 4
- 5 channel 5
- 6 channel 6
- 7 channel 7
- 8 channel 8
- 9 channel 9
- 0 channel 10
- # channel 11
- @ channel 12

d immediate space

- J 1 space
- K 2 spaces
- L 3 spaces

d Skip after print to

- A channel 1
- B channel 2
- C channel 3
- D channel 4
- E channel 5
- F channel 6
- G channel 7
- H channel 8
- I channel 9
- ? channel 10 (EBCDIC or BCDIC)
- . channel 11
- channel 12 (EBCDIC or BCDIC)

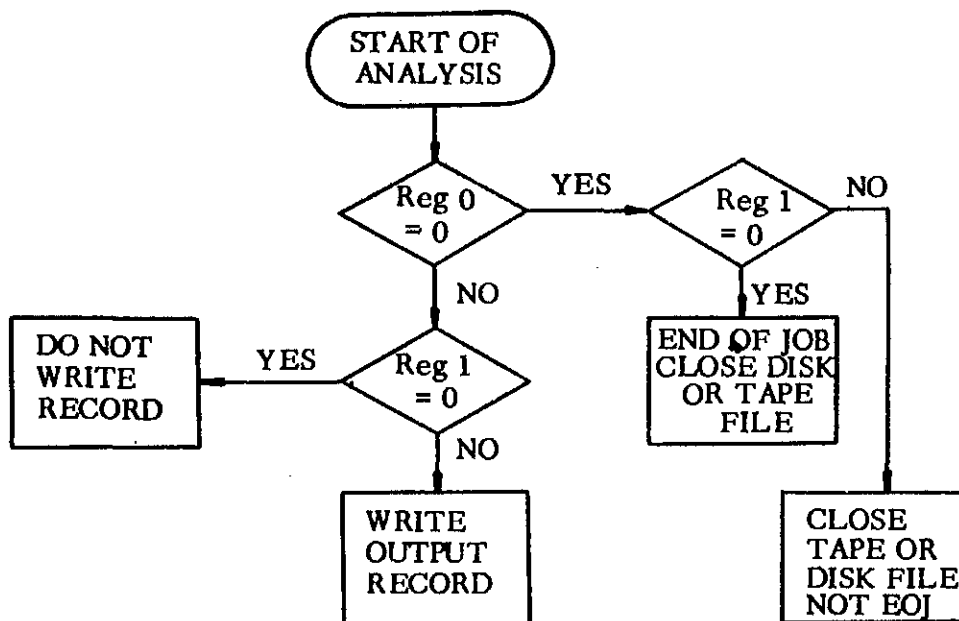
d after print-space

- / 1 space
- S 2 spaces
- T 3 spaces

Type C

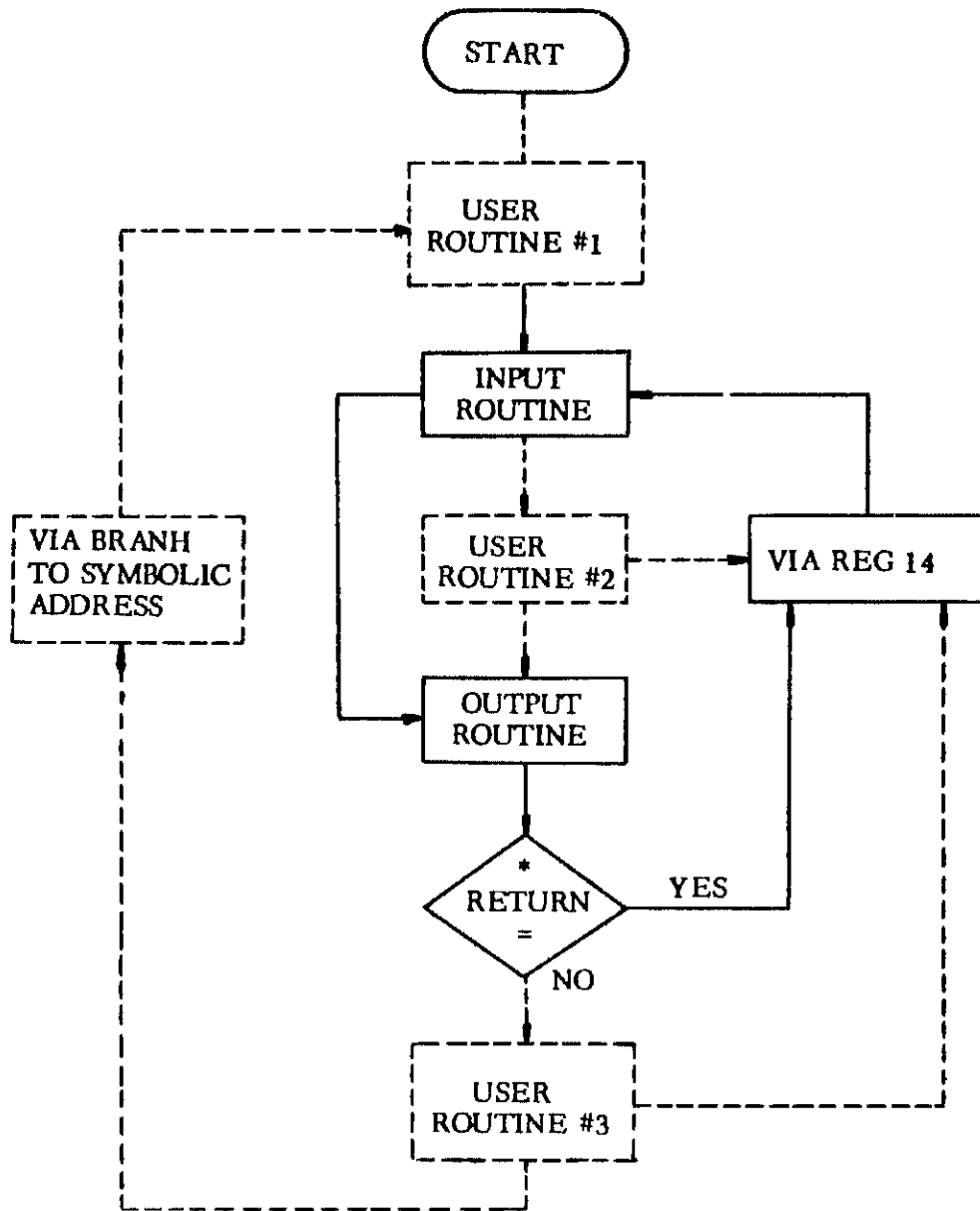
<u>Code</u>	<u>Space or skip Action</u>
Plus (EBCDIC or BCDIC)	Suppress space and print
blank	Print and single space
zero	Double space, print, and space
-	Triple space, print, and space
1-9 or J-R	Immediate skip to channel 1-9 (that is, 1 or J=skip to channel 1; 2 or K=skip to channel 2; etc), print, and then space.

OUTPUT ROUTINE DECISION USING REG 0, 1



REGISTER CONVENTIONS		CONTROL OF OUTPUT
USER ---- 2 ——— 12		$R_0 \neq 0, R_1 \neq 0, \dots$ WRITE RECORD
REG 0	RECORD ADDRESS	$R_0 = 0, R_1 = 0, \dots$ CLOSE FILE, EOJ
1	RECORD LENGTH	$R_0 \neq 0, R_1 = 0, \dots$ DON'T WRITE RECORD
14	RETURN ADDRESS	
15	BASE REG.	$R_0 = 0, R_1 \neq 0, \dots$ CLOSE FILE, NOT EOJ

# TECHNIQUES FOR PASSING CONTROL BETWEEN ROUTINES



SOLID LINES REPRESENT NORMAL OPTIONS AND PROGRAM FLOW.

\* ASSEMBLY TIME DECISION

**SORT/MERGE**

**DISK SORT/MERGE, TAPE SORT/MERGE**

STATE- MENT DEFINER	STATUS	OPERAND	STATUS	OPERAND REMARKS
SORT	Required for SORT runs	FIELD=(P <sub>1</sub> ,M <sub>1</sub> ,S <sub>1</sub> ,P <sub>2</sub> , M <sub>2</sub> ,.....) FORMAT=XX XX=ZD ... Zoned decimal PD ... Packed decimal FI ... Fixed point BI ... Binary FL ... Floating point FILES=m (DS only) SIZE=n (DS only)	Required Required Optional Required	P=Starting Position M=length S=A,... ascending order =D,... dissending order  m=The number of files to be sorted. WORK (TS only) Required CHKPT(TS only) Optional n=logical record length
MERGE	Required for Merge runs	FIELD=(P,M,S,...) FORMAT= XX FILES= X		
RECORD	Required	TYPE=X LENGTH=(L <sub>1</sub> ,L <sub>2</sub> ,L <sub>3</sub> ) =(L <sub>1</sub> ,L <sub>2</sub> ,L <sub>3</sub> ,L <sub>4</sub> , L <sub>5</sub> )	for F for V	x=F ... Fixed length V ... Variable length L <sub>1</sub> =input L <sub>2</sub> =not used L <sub>3</sub> =output L <sub>4</sub> =minimum of input L <sub>5</sub> =average of output
INPFIL	Required	INPUT=X (DS only) VOLUME=(Na,Nb ...)  BLKSIZE=(n, x) OPEN=m CLOSE=m BYPASS	Required Optional Required Optional Optional Optional	X=T ... Tape D ... Disk Na= file A Nb= file B : n=length  incorrect record will be bypassed
OUTFIL	Required	BLKSIZE=n OUTPUT=x (DS only) OPEN=m CLOSE=m NOIPMK (DS only)	Required Required Optional Optional Optional	n=length x=T ... Tape D ... Disk
MODS	for user's Routine	PH <sub>1</sub> =(n, m, E <sub>11</sub> , E <sub>12</sub> , E <sub>13</sub> ) PH <sub>2</sub> =(n, m, E <sub>31</sub> , E <sub>32</sub> ) PH <sub>3</sub> =(n, m, E <sub>n</sub> )	Optional Optional Optional	n=name m=address E <sub>n</sub> =(E <sub>41</sub> , E <sub>42</sub> , E <sub>43</sub> , E <sub>44</sub> , E <sub>45</sub> )

STATE- MENT DEFINER	STATUS	OPERAND	STATUS	OPERAND REMARKS
OPTION	Optional	PRINT	Optional	Control cards Printout
		STORAGE=X	Optional	* DS only
		NOTPMK	Optional	TS only
		VERIFY	Optional	DS only
		KEYLEN=X	Optional	DS only
		ADDROUT=X	Optional	DS only
		CALCAREA	Optional	DS only
		LABEL*(O,W,I) *(O, I)	Optional	(I,W,O)=TS    I=input (I,O) =DS     W=work O=output
		N=nonstandard label		
		S=standard label		
		U=unlabel		
		ALTWK	Optional	TS only
		FILES=X	Optional	TS only
		RESTART	Optional	DS only
END	Required			

\* DS = DISK SORT  
TS = TAPE SORT

NO ALTERNATE DRIVE  Remarks	USER'S REWIND SPECIFICATIONS		ALTERNATE DRIVE/S  Remarks
	OPEN	CLOSE	
The volume is rewind and unloaded. The next volume is rewind.	RWD	RWD	The volume is rewind. The next volume is rewind.
The volume is rewind and unloaded. The next volume is rewind.	RWD	UNLD	The volume is rewind and unloaded. The next volume is rewind.
The volume is rewind and unloaded. The next volume is not rewind.	RWD	NORWD	The volume is not rewind, and the next volume is not rewind.
The volume is rewind and unloaded. The next volume is rewind.	NORWD	RWD	The volume is rewind, and the next volume is rewind.
The volume is rewind and unloaded. The next volume is not rewind.	NORWD	UNLD	The volume is rewind and unloaded. The next volume is rewind.
The volume is rewind and unloaded. The next volume is not rewind.	NORWD	NORWD	The volume is not rewind, and the next volume is not rewind.

Rewind Action Taken at EOJ Time  
for Multi-Volume Files

OPTION Statement



**LABEL STORAGE AREA (1 CYLINDER)**

**Sequential DASD File Label Blocks**

1. Always 104 bytes per extent.
2. Always read into the transient area.

**Non-sequential DASD File Label Blocks**

1. Always at least 104 bytes per file.
2. Additional extents will require an additional 20 bytes similar to bytes 85-104.
3. Always read into lower portion of user core.

BYTE	FUNCTION	CONTENT/FORMAT
1	No. of Extents	X 'bb' A. SEQ DISK 'bb' = '01' for all except last extent. B. SEQ DISK 'bb' = 'FF' for last extent. C. Non-SEQ 'bb' = number of extents (maximum of 125)
2-9	File Name	CL8 'filename' (DTF name)
10-53	File ID	CL44 'Qualified name (Gen# Ver#)'
54	Format ID	CL1 '1'
55-60	File Serial	CL6 'serial'
61-62	Volume Sequence	H 'n'
63	Creation Date	X 'yy'
64-65	" "	H 'ddd'
66	Expiration Date	X 'yy'
67-68	" "	H 'ddd'
69-71	Reserved	3X '00'
72-84	System Code	CL13 'alphanumeric'
85-90	Extent Serial	CL6 'serial'
91	Extent type	X 'bb'
92	Extent Seq. No.	X 'bb' A. ISFMS without a master index 'bb' = '01' B. All others 'bb' = '00'
93-96	Lower limit	4X 'C1', 'C2', 'H1', 'H2'
97-100	Upper limit	4X 'C1', 'C2', 'H1', 'H2'
101-102	Symbolic unit from CCB	2X '0b', 'bb'
103	Old cell	X 'B1'
104	New cell	X 'B2'

**Labeled Tape Label Blocks**

1. Always 80 bytes per file.
2. Always read into lower portion of user core.

BYTE	FUNCTION	CONTENT/FORMAT
1	Reserved	X '00'
2-9	File name	CL8 'filename'
10-58	Mandatory File ID	CL49 'field 3 . . . . field 10'
59-78	Optional File ID	CL20 'field 11 . . . field 13'
79-80	Symbolic unit from CCB	2X '0b', 'bb'



LABEL BLOCK KEY FOR EVERY LABEL BLOCK

BYTE	FUNCTION	CONTENT/FORMAT
1-8	File Name Reserved Extent Sequence #	CL8 'file name'
9		X '00'
10		X 'bb'
		A. Mag Tape 'bb' = '00'
		B. DASD 'bb' = # of first extent

LABEL BLOCKS SUMMARIZED

FILE TYPE	KEY	LABEL BLOCK
TAPE	10	80 for each file
SEQ DASD	10	104 for each extent
NON-SEQ DASD	10	84 + 20 (number of extents)

DISK STORAGE SPACE (Library and Directory)

ELEMENT	Logical record length	Physical record length	Logical records per physical record	Total physical records	Total logical records
CIL Directory	20	360	18	8	144
CIL	1728	1728	1	2	2
RL Directory	14	322	23	9	207
RL	322	322	1	9	9
SSL Directory	15	150	10	16	160
SSL	160	160	1	16	16

Functions which require transients are as follows:

<u>Mnemonic</u>	<u>Transient</u>	<u>Function</u>
OPEN	\$\$BOPEN	open a file - other transient(s) will be called
CLOSE	\$\$CLOSE	close a file - other transient(s) will be called
DUMP	\$\$DUMP	dump core and cancel job
- - -	\$\$BCNCL	cancel a problem program (all job steps)
- - -	\$\$BPCHK	program check routine
- - -	\$\$A----	Type A transients (primarily device error routines)
CHKPT	\$\$BCHKPT	checkpoint
- - -	\$\$BRSTRT	restart
- - -	\$\$BILSVC	illegal (undefined) SVC
- - -	\$\$BMSCIN	process messages from operator
- - -	\$\$BATST	autotest transients
PDUMP	\$\$BPDUMP	dump core and return to caller
- - -	\$\$BOPNLP	OPEN SYSRES (disk system only)
SETFL	\$\$BSETFL	load routine (initialize) ISFMS
- - -	\$\$BSETFF	load routine (initialize) ISFMS
ENDFL	\$\$BENDFL	load routine (terminate) ISFMS
- - -	\$\$BENDFF	load routine (terminate) ISFMS
SETL	\$\$BSETL	sequential return routine (initialize) ISFMS
ESETL	&&BESETL	sequential return routine (terminate) ISFMS

Note: The IPL and Supervisor Nucleus will bear the names \$\$A\$IPL and \$\$A\$SUP. An eighth character may be used to distinguish phases of these programs.

DOS SUPERVISOR CALLS

Macro Supported	SVC	Function
EXCP	0	Execute channel programs.
FETCH	1	Fetch any phase.
	2	Fetch a logical transient (B-transient).
	3	Fetch or return from a physical transient (A-transient).
LOAD	4	Load any phase.
MVCOM	5	Modify supervisor communications region.
CANCEL	6	Cancel a problem program.
WAIT	7	Wait on a CCB or TECB.
	8	Transfer control to the problem program from a logical transient (B-transient).
LBRET	9	Return to a logical transient (B-transient) from the problem program after a SVC 8.
SETIME	*10	Set timer interval.
	11	Return from a logical transient (B-transient).
	12	Logical AND (Reset) to second Job Control byte (displacement 57 in communications region).
	13	Logical OR (Set) to second Job Control byte (displacement 57 in communications region).
FOJ	14	Cancel job and go to Job Control for end of job step.
	15	Same as SVC 0 except ignored if CHANQ table is full.
STXIT (PC)	*16	Provides supervisor with linkage to user's PC routine for program check interrupts.
EXIT (PC)	*17	Return from user's PC routine.
STXIT (IT)	*18	Provides Supervisor with linkage to user's IT routine for interval timer interrupts.
EXIT (IT)	*19	Return from user's IT routine.
STXIT (OC)	*20	Provides Supervisor with linkage to user's OC routine for external or attention interrupts (operator communications).
EXIT (OC)	*21	Return from user's OC routine.
	*22	The first SVC 22 seizes the system for the issuing program by disabling multiprogram operation. The second SVC 22 releases the system (enables multiprogram operation).
	*23	Load phase header. Phase load address is stored at user's address.
	*24	Provide Supervisor with linkage to user's TECB and set timer interval.
	*25	Issues HALT I/O on a teleprocessing device.
	*26	Validate address limits.

\* = optional

0 ← Reset to zeros after IPL → 15								
16 Comm Region Address	18 External Old PSW	20 SVC Old PSW	28 Program Check Old PSW	30 Machine Check Old PSW	38 I/O Old PSW	40 CSW	48 CAW	4C BG Job Dura- tion
50 System Timer	54 System Time of Day	58 External New PSW	60 SVC New PSW	68 Program Check New PSW	70 Machine Check New PSW	78 I/O New PSW		
80 Diagnostic Scan-Out Area --- Model Dependent								
SUPERVISOR NUCLEUS								
General Cancel Routine								
General Exit Routine								
Communications Region								
Entry Register Area								
General Entry Routine								
Execute Channel Program (EXCP)								
I/O Interrupt								
Unit Check								
Fetch								
Other Interrupts								
Option Routines								
SVC Routines								
Resident Device Error Recovery								
Supervisor Constants								
SVC Interrupt Table								
Logical Transient Save Area								
All Bound PIB	BG PIB	F2 PIB	F1 PIB	Attention PIB	Quiesce PIB	Supervisor PIB	Channel Queue	
FOCL	PUB TABLE			FAVP	JIB TABLE			
TEB	FICL	NICL	LUB TABLE					
PATCH AREA								
LOGICAL TRANSIENT AREA (B-TRANSIENTS) \$\$B								
PHYSICAL TRANSIENT AREA (A-TRANSIENTS) \$\$A								
BG Program Save Area	PROBLEM PROGRAM AREA							

Low  
Core

Nucleus  
Code

I/O Tables  
and  
Information  
Blocks

Logical  
and  
Physical  
Transients



- 0** MM/DD/YY or DD/MM/YY obtained from the job control date statement. Format controlled by COMREG +53 (date convention byte) bit 0.
- 8** Address of the problem program label area. (End of transient area +1).
- 10** Address of the beginning of the problem program area. Y (EOSSP) = Y (PPBEG)-if the storage protection option has not been selected. Y (EOSSP) equals the first main storage location with a storage protection key of 1, if storage protection is supported.
- 12** User area
- 23** User program switch indicator.
- 24** Job name set by the job control program from information found in the job statement.
- 32** Address of the uppermost byte of the problem program area as determined by the IPL program. (Clear storage routine determines the address, ENDRT routine of \$\$A\$I PL2 stores it.)
- 36** Address of the uppermost byte of the last phase of the problem program fetched or loaded. The initial value (as shown) is overlaid by the first fetch or load to the problem program area.
- 40** Address of the uppermost byte of the longest phase of the problem program fetched or loaded. The initial value is overlaid by the first fetch or load to the problem program area.
- 44** Length of the problem program label area.
- 46** Program interrupt Key: Value is equal to the displacement from the start of the PIB table to the PIB For the Task.  
 First Byte - always zero  
 Second Byte - Contains the key of the program that was last enabled for interrupts. (When an interrupt occurs, the PIK indicates to the supervisor which program was interrupted.)

<u>Task</u>	<u>PIK Value</u>
All Bound	X'00'
BG	X'10'
*F2	X'20'
*F1	X'30'
Attn Rtn	X'40'
Quiesce I/O	X'50'
Supervisor	X'60'

\*These tasks do not exist in a batch-job-only system.

- 48** Logical end of main storage address
- 52** Configuration Byte (Values set at supervisor generation time.)
  - Bit 0: 1 = Storage protect  
0 = No storage protect
  - 1: 1 = Decimal feature  
0 = No decimal feature
  - 2: 1 = Floating point feature  
0 = No floating point feature
  - 3: Reserved
  - 4: 1 = Timer feature  
0 = No timer feature
  - 5: 1 = Channel switching device  
0 = No channel switching device
  - 6: 1 = Burst mode on multiplex channel support  
0 = No burst mode on multiplex support
  - 7: 1 = 7-track SYSRES  
0 = No 7-track SYSRES

**53** Date Convention Byte

- Bit 0: 1 = DDMMYYJJ  
0 MMDDYYJJ (Set at generation time by STDJC)
- 1: 1 = Multiprogramming environment  
0 = Batch job environment
- 2: 1 = DASD file protect supported  
0 = No file protect support for DASD
- 3: 1 = DASD SYSIN-SYSOUT  
0 = No DASD SYSIN-SYSOUT
- 4: 1 = RTAM = YES  
0 = RTAM = NO
- 5-7: Reserved

**54** This byte contains the standard language translator I/O options (set by the STDJC macro).

- Bit 0: DECK option 1 = yes, output object modules on SYSPCH
- 1: LIST option 1 = yes, output source module listings and diagnostics on SYSLST.
- 2: LISTX option 1 = yes, output hexadecimal object module listings on SYSLST (compilers only).
- 3: SYM option 1 = yes, output symbol tables on SYSLST/SYSPCH
- 4: XREF option 1 = yes, output symbolic cross reference list on SYSLST
- 5: ERRS option 1 = yes, output diagnostics on SYSLST (compilers only).
- 6: CHARSET option 1 = 48, input on SYSIPT is 48 or 60 character set
- 7: Reserved

**55** This byte contains the standard supervisor options for abnormal EOJ and control statement display.

- Bit 0: Not used
- 1: DUMP option 1 = yes, dump registers and storage on SYSLST
- 2: Not used
- 3: LOG option 1 = yes, list all control statements on SYSLST
- 4-6: Not used
- 7: Reserved

**56** Job control byte (JBCSW0)

- Bit 0: Reserved
- 1: 1 = Return to caller on LIOCS disk open failure  
0 = Do not return to caller on LIOCS disk open failure
- 2: 1 = Job control input from SYSRDR  
0 = Job control input from SYSLOG
- 3: 1 = Job control output on SYSLOG  
0 = Job control output not on SYSLOG
- 4: 1 = Cancel job  
0 = Do not cancel job
- 5: 1 = Pause at end-of-job step  
0 = No pause at end-of-job step
- 6: 1 = SYSLOG is not a 1052  
0 = SYSLOG is a 1052
- 7: 1 = SYSLOG is assigned to the same device as SYSLST  
0 = SYSLOG is not assigned to the same device as SYSLST

**57** Job control byte (JBCSW1)

- Bit 0: 1 = SYSLNK open for output  
0 = SYSLNK not open for output
- 1: Reserved
- 2: 1 = Allow EXEC  
0 = Suppress EXEC
- 3: 1 = Catalog linkage editor output  
0 = Do not catalog linkage editor output
- 4: 1 = Supervisor has been updated  
0 = Supervisor has not been updated
- 5: 1 = Executing in AUTOTEST mode  
0 = Not executing in AUTOTEST mode
- 6: 1 = Ignore attention interrupt on 1052  
0 = Do not ignore 1052 attention interrupt
- 7: 1 = Fetch MAINEOJ at end of job to update system directory  
0 = Do not fetch MAINEOJ at end of job for update

**58** Job control byte (JBCSW2) called the language processor control byte. This is a set of switches used to specify nonstandard language translator options. The switches within the byte are controlled by job control OPTION statements and when set to 1, override standard options. The format of this byte is identical to the standard option byte (displacement 54).

**59** Job control byte (JBCSW3)

- Bit 0: 1 = Within a job condition  
0 = Outside a job condition
- 1: 1 = Dump on an abnormal end-of-job condition  
0 = No dump on abnormal (O)
- 2: Reserved
- 3: 1 = Job control output on SYSLST  
0 = Output not on SYSLST
- 4: 1 = Job is being run out of sequence with a temporary assignment for SYSRDR  
0 = Conditions for 1 setting not met
- 5: 1 = No OPEN  
0 = Initial entry to OPEN
- 6: 1 = OPEN monitor entry is from the DTFCP OPEN phase  
0 = Conditions for a 1 setting not met
- 7: Reserved

**60** Binary disk address of the volume label area

**62** → **76** As illustrated

**78** Set to the value nn specified in the LINES = nn parameter of the STDJC macro.

**79** The format of the system date contained within this field is determined by the IPL program from information supplied in the date convention byte (displacement 53). Bytes 85 - 87 contain the day count.

**88** Reserved

**89** Byte reserved for use by LIOCS. Transient dump programs insert a key to indicate to the LIOCS end-of-volume routine, \$BCMT07, that it was called by a B-transient.

**90** Address of the program information block (PIB) table.

**92** ID number of the last checkpoint

**94** Length of the LUB ID queue (in bytes). This equals the number of channel queue entries. It can also be used to access the REQUID queue.

**96** Address of disk I/O position data. This is the starting address of the disk information block (DIB) table.

**98** Address of the beginning of the error recovery block. The error recovery block contains addresses of error recovery exits, error recovery queue information that can be used by physical transient routines, and defines storage for the error queues entries.

**100** → **104** As illustrated

**106** Key of the program (BG, F2, or F1) that has timer support.

**108** As illustrated

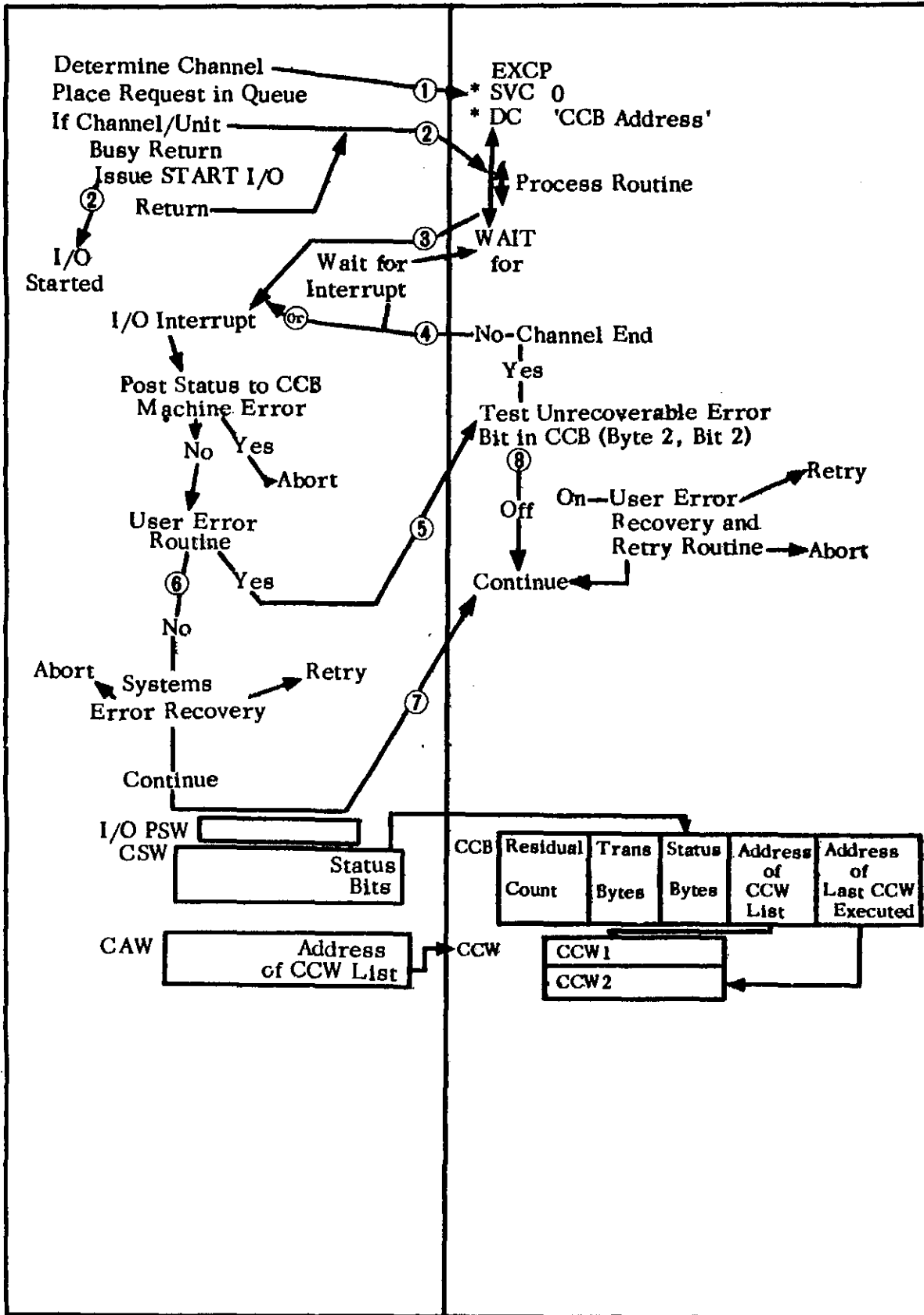
**110** Logical transient Key (LTK) contains the same value as the PIK (displacement 46) when the logical transient is requested. When the transient area is not in use, LTK is equal to zero. The SVC 2 routine sets the LTK. The SVC 11 routine resets the LTK.



Physical IOCS

Supervisor Core

Problem Program Core



Record Retrieval Using Physical IOCS

PUB & LUB TABLE

LUB TABLE

FICLST

00	0B	15	1A
----	----	----	----

NICLST

0B	0A	05	05
SYS	BG	F2	F1

PUB TABLE

POS	CHAN CLASS	UNIT NL.M	QUE PTR	TEB PTR	DEV TYPE	MODE	CHAN SCHED FLAG	JOB CTL FLAG
00	00	0C	05	00	11	00	80	F9
01	00	0D	FF	00	21	00	00	FA
02	00	0E	03	00	40	00	84	FA
03	00	1F	FF	00	00	00	00	F8
04	01	80	FF	00	50	C3	00	C4
05	01	81	02	01	50	C3	00	C4
06	01	82	FF	02	50	C3	00	C2
07	01	83	FF	03	50	C3	00	C0
08	01	84	FF	04	50	C3	00	C0
09	01	90	04	00	60	00	80	FD
10	01	91	FF	00	60	00	00	F8
11	FF	00	00	00	00	00	00	00
	FF							

SYS	POS	PUB PTR	JIB PTR	FLG
RDR	00	FF	FF	
IPT	01	FF	FF	
PCH	02	FF	FF	
LST	03	FF	FF	
LOG	04	03	FF	
LNK	05	09	00	
RES	06	09	FF	
SLB	07	FF	FF	
RLB	08	FF	FF	
USE	09	FF	FF	
FGI	0A	FF	FF	
000	0B	09	FF	
001	0C	04	FF	
002	0D	05	FF	
003	0E	09	FF	
004	0F	FF	FF	
005	10	FF	FF	
006	11	FF	FF	
007	12	FF	FF	
008	13	FF	FF	
009	14	FF	FF	
000	15	FF	FF	
001	16	00	FF	
002	17	FF	FF	
003	18	FF	FF	
004	19	09	01	
000	1A	FF	FF	
001	1B	06	FF	
002	1C	01	FF	
003	1D	02	FF	
004	1E	FF	FF	

FF=UA  
FE=IGN

CHAN. SCHED. FLAG

0	1	2	3	4	5	6	7
DEV BUSY	SWITCHABLE	EOF SYSRDR OR IPT	I/O ERR QUEUED FOR RETRY	OPER INTV REQ	DEV END POST	BURST DEV ON MAX	7 TRK TAPE

JOB CTL FLAG

0	1	2	3	4	5	6	7
	MADE FOR TAPE				ASSGN TO BG	ASSGN TO F1	ASSGN TO F2

ALL ONE'S FOR NON-TAPE  
ALL ZERO'S FOR DEV DOWN

ZERO IF DEV UNASSIGNED  
OR SHARED

FLPTR

00

CHANQ

POS	CHAIN PTR	CCB ADDR		
00	FF	00	4B	58
01	FF	00	64	90
02	FF	00	21	1E
03	01	00	64	70
04	FF	00	4B	58
05	FF	00	49	98

LUBID REQID

CHQUE LUB PTR	CHQUE ID LIST	
19	F	F
1D	3	0
0D	1	0
1D	3	0
19	2	0
16	2	0

FAVP

02

JIB TABLE

POS	STORED STD ASSGN'S XTENT INFO		CON-TENT	OWNER	CHAIN PTR
00	BE	C6	2	8	FF
01	BE	C6	2	8	FF
02	00	00	0	0	03
03	00	00	0	0	04
04	00	00	0	0	05
05	00	00	0	0	06
06	00	00	0	0	FF

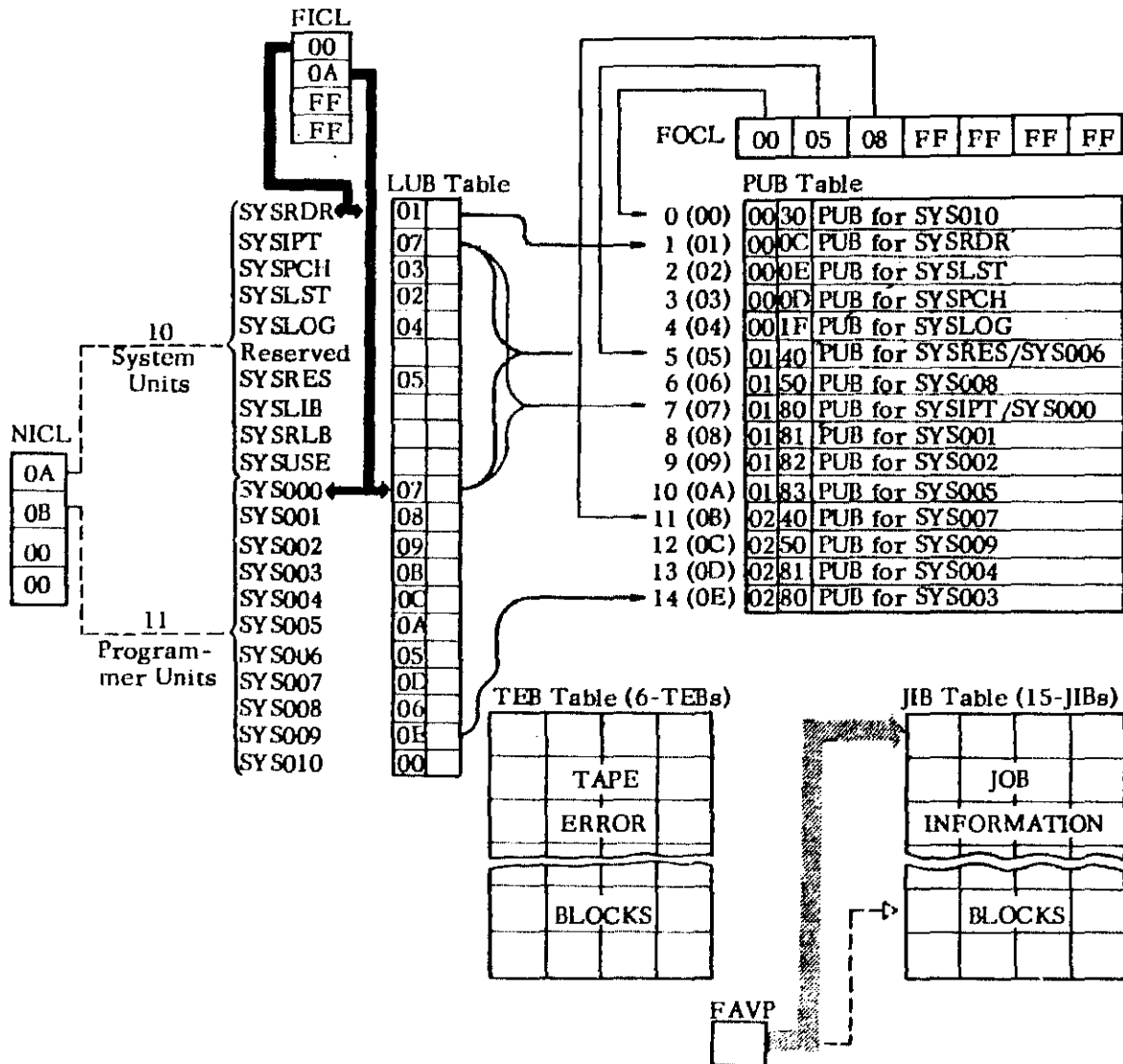
STOR KEY

USE:  
FF=UNUSED  
60=SUPVR  
61=SUP CCB  
FOR SYSLOG  
62=FCH CCB

STORED STD	ALT ASSGN	2311 XTENT	2321 XTENT	SHARED ASSGN	BG	F1	F2
0	1	2	3	4	5	6	7

NOTE: TWO ENTRIES REQUIRED FOR 2321 JIB.

I/O TABLES GENERATED



**FICL (First in Class):** First byte always points to position 0 of the LUB table (first system unit).  
Second byte always points to the first programmer unit.

**NICL (Number in Class):** First byte contains the number of system units.  
Second byte contains the number of programmer units.

**LUB Table:** The minimum LUB table has 15 entries; ten for system units plus five for programmer units SYS000-SYS004. Each assigned LUB points to a PUB in the PUB table with byte 1. Byte 2 is a LUB flag or JIB pointer.  
See next page for LUB format.

**PUB Table:** There must be a PUB for each I/O device on the system. A PUB contains the physical characteristics of the I/O device it represents.  
See Page 81 for PUB format.

**FOCL (First on Channel List):** The FOCL is seven bytes long. Byte 0 points to the first PUB (highest priority) on channel zero. Byte 1 points to the first PUB (highest priority) on channel one, etc. Hexadecimal FF indicates the associated channel is not supported.

**TEB (Tape Error Block):** One TEB is generated for each tape unit at supervisor generation time if tape error statistics are required.  
See Page 82 for TEB format.

**JIB (Job Information Block):** Each 4-byte JIB contains the temporary or alternate assignments for a LUB. The LUB points to the associated JIB.  
See Page 82 for JIB format.

**FAVP (First Available Pointer):** The FAVP is a 1-byte pointer that points to the next available JIB (next JIB not in use).

LUB TABLE

LUB TABLE

SYSROR		
SYSIPT		
SYSPCH		
SYSLST		
SYSLOG		
RE-SERVED		
SYSRES		
SYSSLB		
SYSRLB		
SYSUSE		
SYS000		
SYS001		
SYS002		
SYS003		
SYS004		
...		
SYS244		

Format of Any Lub

Byte 0	Byte 1
PUB Pointer	Flag Byte

HEX  
 00 = Points to first PUB  
 01 = Points to second PUB  
 02 = Points to third PUB  
 .  
 .  
 .  
 FF = Null pointer, the LUB is unassigned

If bit 7 = 1, then:  
 Bits 0-3 are standard LUB flags.  
 Bit 0 : 1 = ignore (IGN)  
           0 = unassigned (UA)  
           1 : 1 = standard assignement  
           2 : 1 = standard unassigned (UA)  
           3 : 1 = standard ignore (IGN)  
           4 - 6 = unused

If bit 7 = 0, then:  
 The standard LUB has been stored in a JIB. Bits 0-6 contain the displacement (0, 4, 8, ...) within the JIB table where the LUB is stored.  
 Note that Bit 7 is always 0 for a value of 0, 4, 8, 12, ...

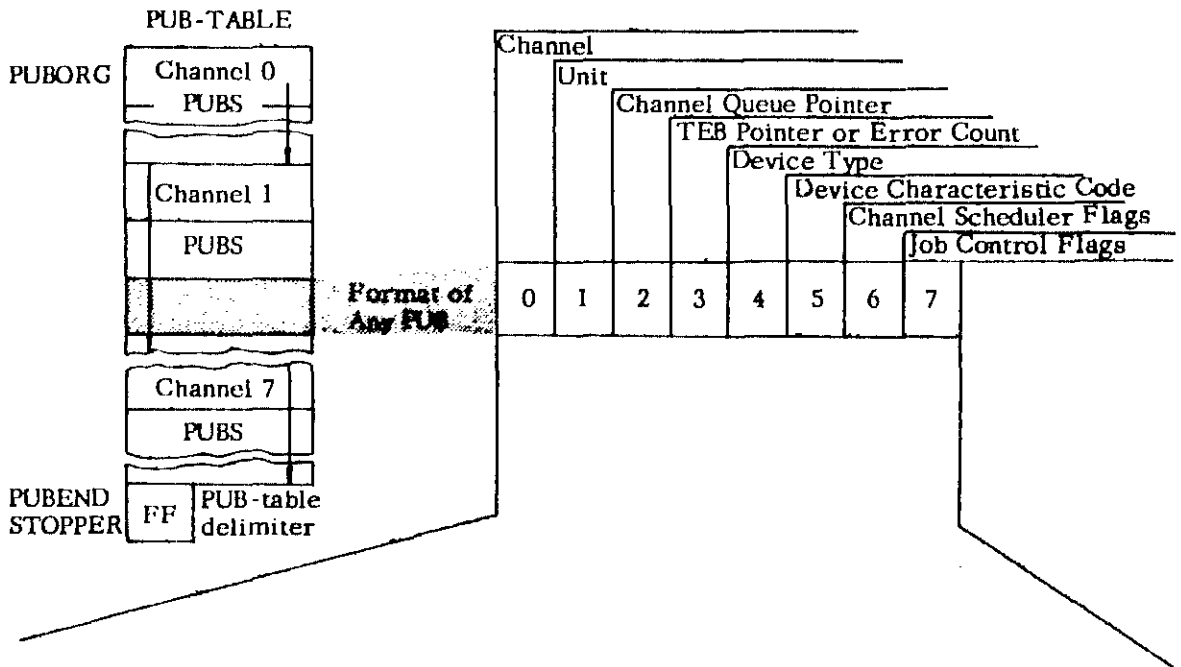
NOTE:

The number of entries in the LUB table is determined at supervisor generation time. The number of LUBS is equal to the number of system units plus the number of programmer units. The number of system units is always 10. The number of programmer units is specified in the IOTAB macro (PGR = n).

IF: LUBS = number of LUBS in the LUB table.  
       10 = number of system units (SYSRDR - SYSUSE).  
       n = number of programmer units (SYS000 - SYSXXX).

THEN: LUBS = 10 + n

PUB TABLE



BYTE 0 - Channel number. (Hex 0-7, FF = NULL)

BYTE 1 - I/O device unit number. (HEX IF = 1052, HEX 80 = magnetic, tape unit 0 ...).

BYTE 2 - HEX 0, 1, 2, ... points to the channel queue currently using this device.

BYTE 3 - If device is a magnetic tape unit and TEBS are specified, this byte is a TEB pointer (HEX 1, 2, 3 ...).  
If device is a magnetic tape unit but TEBS are not specified, this byte is an error counter.  
If device is not a magnetic tape unit, this byte is an error counter.

BYTE 4 - See Figure 17 for device type codes.

BYTE 5 - SS of the MODE = parameter in the DVCGEN macro for a tape unit.

BYTE 6 -

- Bit 0 : 1 = Device busy
- 1 : 1 = Switchable device
- 2 : 1 = EOF for SYSRDR or SYSIPT
- 3 : 1 = Error recovery (I/O being reexecuted)
- 4 : 1 = DASD file protection required
- 5 : 1 = Device end received
- 6 : 1 = Burst device on MPX
- 7 : 1 = 7-track tape unit

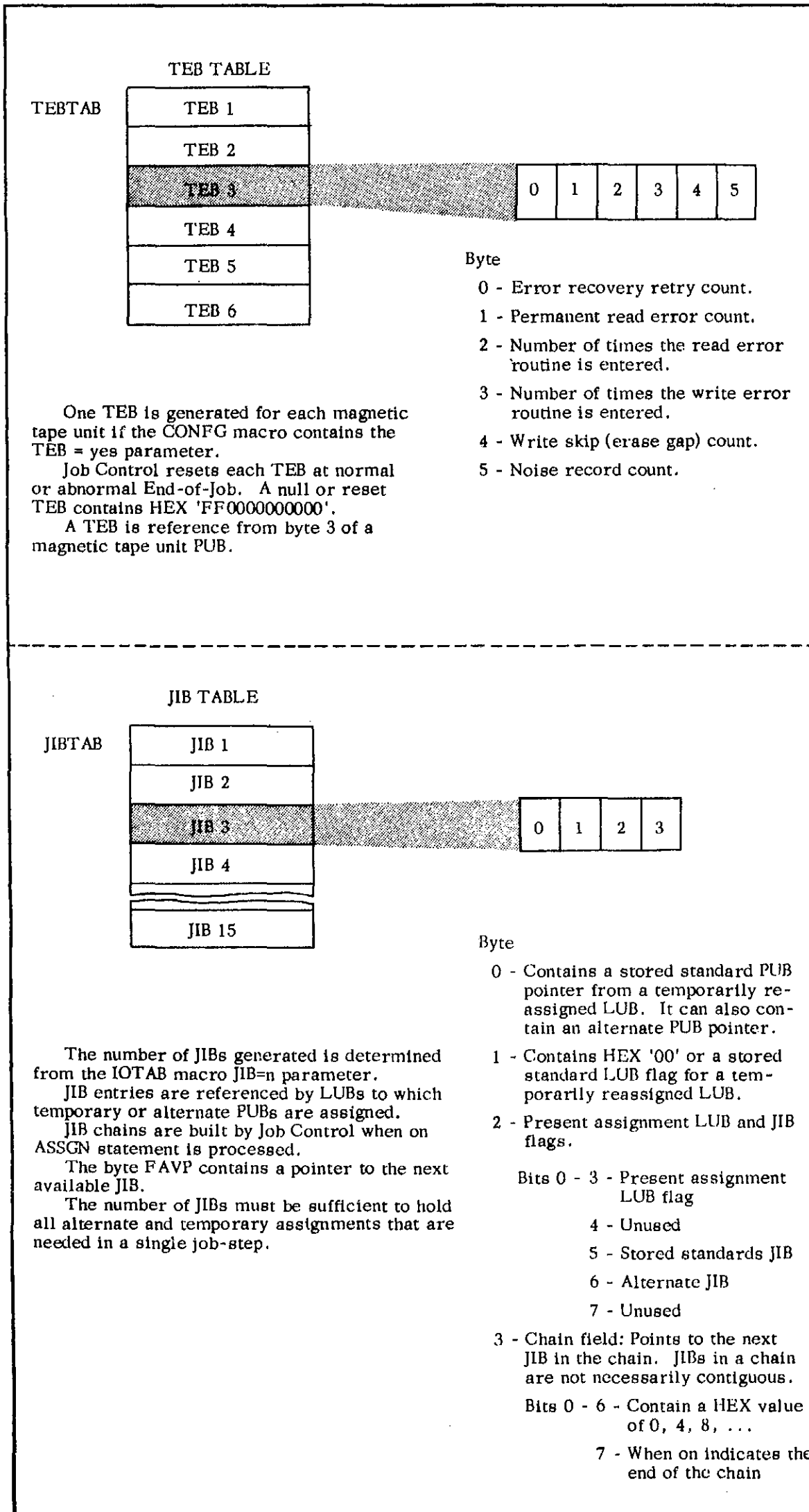
BYTE 7 -

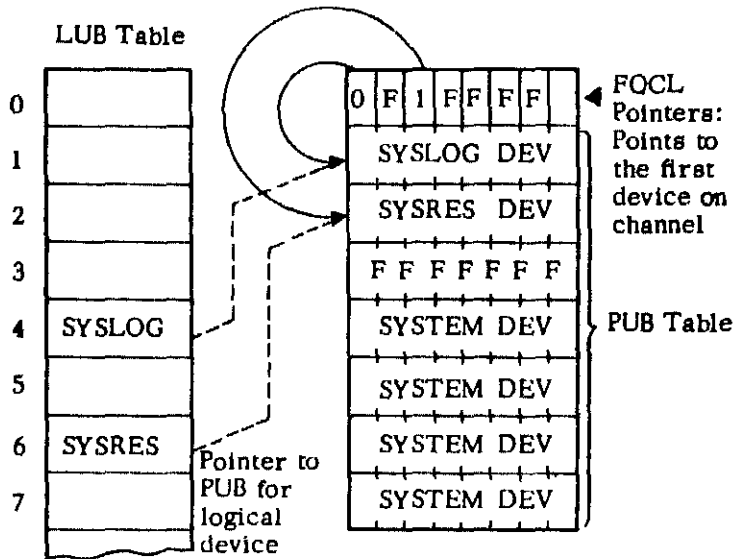
- Bit 0-4 : standard MODE assignment for 7-track tape (all ones if not tape, all zeros if device is down).
- 5 : device is assigned to a system unit (0 is UA).
- 6 : device is assigned to a programmer unit (0 if UA).
- 7 : unused (zeros if UA).

NOTE:

A null PUB is generated for each device to be supported by the supervisor. (See IOTAB macro in this section.)

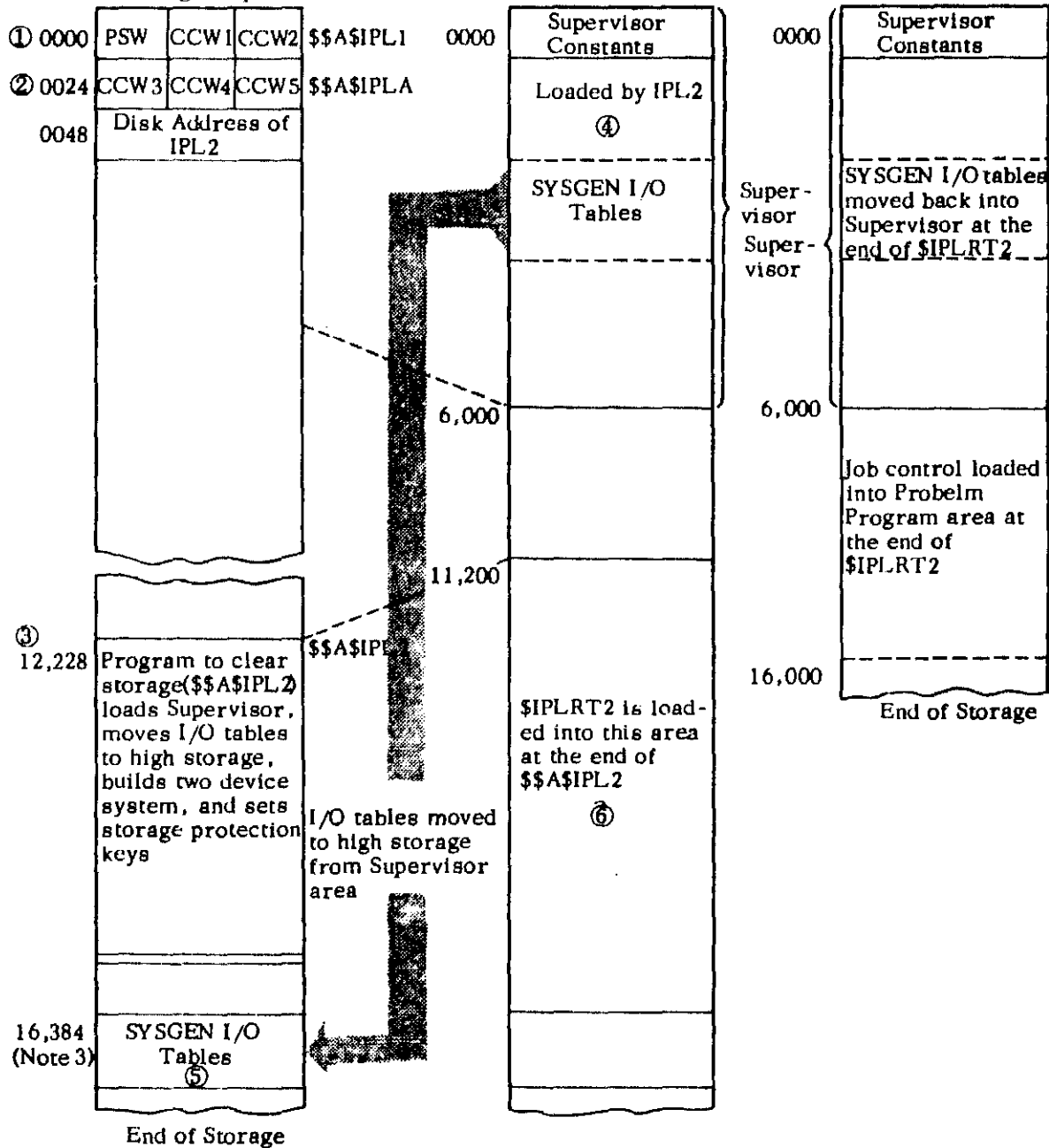
Standard physical unit assignments are made to the PUB table at supervisor generation time. PUBS are ordered by channel and priority within a channel. (See DVCGEN macro in this section.)





NOTE: It is assumed that SYSRES is on channel 2 and that the communication device SYSLOG is on channel 0.

IPL Main Storage Map



Note 1: Circled numbers represent sequence.

Note 2: Storage addresses are in decimal.

Note 3: The I/O tables will be at the end of the Supervisor in 16K machine. In larger machines, they will be at decimal 16,384.



### Clear Storage - Manually (Model 40)

The following procedure for manually cleaning memory does not clear any of the general purpose or floating point registers:

1. Set "rate switch" to "process".
2. Press "system reset" key.
3. Set "rate switch" to "single cycle".
4. Set "diagnostic control" to "MS address".
5. Set bit 3 of byte 0 of the "address" bit switches in the down position.
6. Flip the "store status" switch. This calls in a micro-program that sets all of memory to zeros.
7. Insure that "Y3 Stat" is turned on.
8. Reset the "rate switch" back to "process".
9. Hit "start". Nothing should appear in bytes 0 and 1 of the "address" and "data" registers.
10. Restore "diagnostic CTL switch" to "off".

### Address Compare (Address Stop) (Model 40)

To stop the CPU at a predetermined location:

1. Turn the "address compare" switch to "stop".
2. Set the bits in the "address" bit switches for the desired address.
3. Press the "start" key.

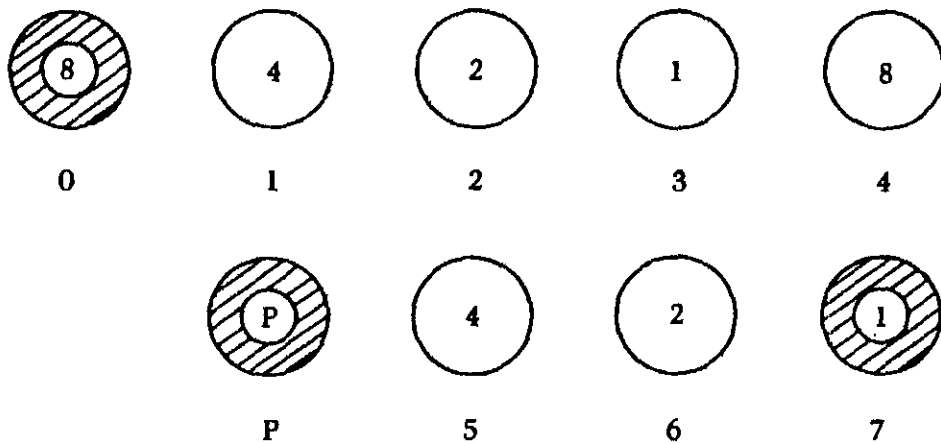
The system will resume processing until an equal address comparison is made. The CPU then switches itself to the "stopped" state.

**HARDWARE**

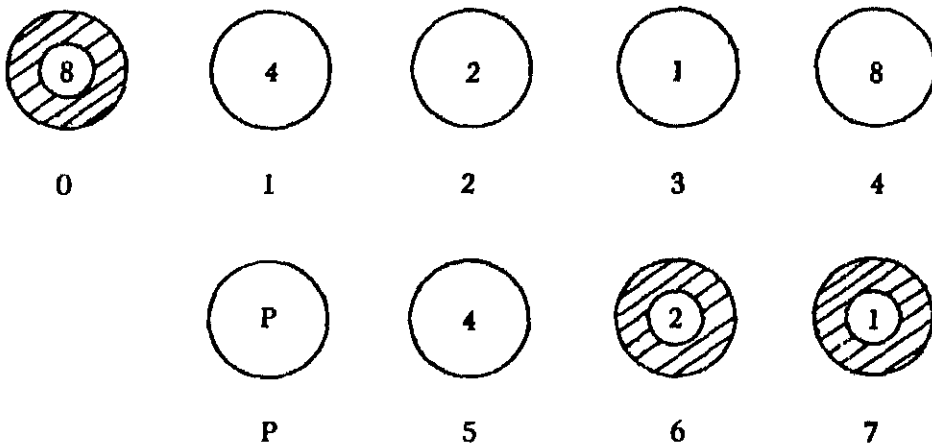
**1. To Write Tape Mark From TAU C. E. Console**

- a. Turn lower left corner switch to OFF-LINE.
- b. Address the tape drive by inserting pegs in the two rows under ADDRESS labeled 01234 and P567, with odd parity. The positions have the values 84218 and P421 respectively. The top set (0123), addresses the ten's position and the lower set (4567) the units' position of the drive address. The P is used for parity when required.

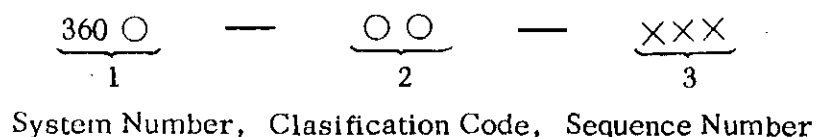
**Example 1 Address for Drive '81'**



**Example 2 Address for Drive '83'**



- c. Coding for TAPE MARK requires pegs in 34567 STOP of the first COMMAND row.
- d. Peg INTE either A or B in lower right corner.
- e. Set toggle switch on SINGLE and INTERFACE on AUTO.
- f. Press STOP on console to remove from SYSTEM state.
- g. Press MACHINE RESET, CHK RESET and START to write the tape mark.
- h. Restore by returning left corner switch to ON-LINE.



1. System Number

360T	.....	Model 20
360U	.....	Model 20 Tape
360P	.....	BPS
360B	.....	BOS
360M	.....	TOS
360N	.....	DOS
360S	.....	OS
360F	.....	Model 44
360C	.....	Emulator
360A	.....	Type II (Last character of sequence Number is x)
360D	.....	Type III (Format 2, 3 ... xx. x. xxx)

2. Programming System Type I

AS	.....	Assembler
CB,CO	.....	COBOL
CL	.....	Control program
CQ	.....	Communications Input/Output
CV	.....	Conversion
DC	.....	Documentor
DM	.....	Data management
DN	.....	Diagnostic
ED	.....	Editor
EU	.....	Emulator program
FO	.....	Fortran
IO	.....	Input/Output
LM	.....	Library Material
LD	.....	Loader
MI	.....	Miscellaneous
OS	.....	Operating system
PL	.....	PL/I
PT	.....	Autotest
RC	.....	Remote computing
RG	.....	RPG
SI	.....	Simulator
SM	.....	Sort/Merge
SV	.....	Supervisor
UT	.....	Utility

Programming Application Type II

Industry-Oriented Programs

DP	.....	Publishing
DR	.....	Retail
DW	.....	Wholesale
DX	.....	Other

Finance

FB	.....	Banking
FF	.....	Finances Companies
FI	.....	Brokerage and Investment
FX	.....	Other

Federal Government

GF	.....	Government, Federal
----	-------	---------------------

**Insurance**

IB ..... Blue Cross and Shield  
IF ..... Fire and Casualty  
IL ..... Life  
IX ..... Other

**Manufacturing**

MA ..... Aerospace  
MD ..... Drug, Food, Chemical Products  
ME ..... Electrical and Machinery  
MF ..... Fabrication and Primary Metals  
MP ..... Petroleum and Industrial Chemicals  
MR ..... Transportation Equipment  
MT ..... Textiles and Paper  
MX ..... Other

**Services**

SC ..... Communication  
ST ..... Transportation  
SU ..... Utilities  
SX ..... Other

**Universities and Government**

UC ..... Colleges and Universities  
UG ..... Government, State and Local  
UH ..... Hospital and Medical  
US ..... Secondary Schools  
UX ..... Other

**Industry-Independent Programs**

**Cross Industry Group**

CA ..... Statistical Applications  
CC ..... Process Control  
CM ..... Mathematical Applications  
CN ..... Numerical Control Applications  
CO ..... Operations Research  
CP ..... Critical Path Scheduling  
CR ..... Information Retrieval  
CS ..... Simulators  
CX ..... Other

**Engineering**

EC ..... Civil Engineering  
EE ..... Electrical Engineering  
EH ..... Chemical Engineering  
EM ..... Mechanical Engineering  
EN ..... Nuclear Codes  
EO ..... Optics  
EX ..... Other

**Exploratory**

XP ..... Mathematics and Applications

TOS/360	Assembler	360M-AS-406
TOS/360	COBOL	360M-CB-402
TOS/360	System control	360M-CL-405
TOS/360	FORTRAN IV	360M-FO-409
TOS/360	IOCS	360M-IO-404
TOS/360	Autotest	360M-PT-407
TOS/360	RPG	360M-RG-408
TOS/360	SORT/Merge	360M-SM-400
TOS/360	Utility Programs	360M-UT-403
DOS/360	Assembler-Basic modules	360N-AS-465
DOS/360	COBOL	360N-CB-452
DOS/360	COBOL DASD macros	360N-CB-468
DOS/360	Sys. cont. & Basic IOCS	360N-CL-453
DOS/360	BTAM	360N-CQ-469
DOS/360	FORTRAN IV	360N-FO-451
DOS/360	Direct Access method	360N-IO-454
DOS/360	Consecutive Disk IOCS	360N-IO-455
DOS/360	Consecutive Tape IOCS	360N-IO-456
DOS/360	ISFMS	360N-IO-457
DOS/360	Consecutive P. T. IOCS	360N-IO-458
DOS/360	Compiler I/O modules	360N-IO-476
DOS/360	RPG	360N-RG-460
DOS/360	SORT/MERGE-Tape	360N-SM-400
DOS/360	SORT/MERGE-Disk/Tape	360N-SM-450
DOS/360	Supervisor 6K	360N-SU-474
DOS/360	Supervisor 8K	360N-SU-475
DOS/360	GROUP 1 utilities	360N-UT-461
DOS/360	GROUP 2 utilities	360N-UT-462
DOS/360	GROUP 3 utilities	360N-UT-463
DOS/360	MPS utility macros	360N-UT-471



I N D E X

	<u>PAGE</u>		<u>PAGE</u>
ADDRESS STOP (M40)	87	LOG	19, 20, 21
ALLOC COMMAND	19, 20	LUB ADDRESS	75
ASSGN CARD	18, 22	LUB TABLE	80, 82, 83, 86
ATTN COMMAND	20	MAP	19, 20
CANCEL COMMAND	19, 21	MSG	20
CAW FORMAT	11	MTC	19, 24
CARD TO DISK	57	NOLOG	19
CARD TO PRINTER/PUNCH	57	OPTION	25, 68
CARD TO TAPE	58	OUTAPE	62
CATALOG	32	OUTCARD	62
CCB FORMAT	13, 14	OUTDISK	62, 63
CCW FORMAT	11	OUTLOG	63
CHANNEL COMMAND CODE	12	OUTPRT	63
CHANNEL COMMAND (DASD)	11	PERMANENT STORAGE	4
CHANQ	81 - 82	ASSIGNMENT	
CLEAR DISK	60	PAUSE	19, 25
CLEAR STORAGE (M40)	87	PHASE CARD FORMAT	26
CLOSE COMMAND	18	PHYSICAL IOCS (SAMPLE)	26
COMMAND TABLE	11, 12	PHYSICAL IOCS	79
COMMUNICATION REGION	75 - 78	PSW FORMAT	5
COMRG	75	PROGRAM INTERRUPTION	6
CONDENSE (SYSREG)	33	PROGRAM NAME	89 - 91
CONDITION CODE SETTING	5	PUB TABLE	80, 82, 84
COPY (LIBRARIAN)	35 - 37	PUNCH (LIBRARIAN)	34 - 35
DATE COMMAND	22	PUB ADDRESS	75
DELETE	32	READ COMMAND	21
DISK LABEL	69	REALLOCATION	33
DISK SORT SAMPLE	28	RELSE COMMAND	19, 21
DISPLAY	34	RENAME	32
DISK TO CARD	59	REP FORMAT	31
DISK TO DISK	60	RESERT COMMAND	19, 22
DISK TO PRINTER	60	RLD FORMAT	30 - 31
DISK SORT	67, 68	RSTRT	24
DISK TO TAPE	60	SAMPLE	26
DLAB	21, 23	SENSE BYTE	15
DTFCD	39	SET COMMAND	20
DTFCN	52	SORT/MERGE	67, 68
DTFDA	44	SORT SAMPLE (DISK)	28
DTFDA ERRBYTE	45	SORT SAMPLE (TAPE)	29
DTFIS	46, 47	STANDARD INSTR. SET	1 - 2
DTFIS ERREBYTE	48	STANDARD LABEL	69
DTFMT	41	START COMMAND	20
DTFPH	52	STORAGE MAP	74, 86
DTFPR	40	STOP COMMAND	19
DTFPT	42	SVC CODE	73
DTFSD	43	SYSREG SPACE	71
DTFSR	49 - 51	TAPE LABEL	69
DUCUP	19	TAPE MARK (M40)	88
END CARD FORMAT	31	TAPE SORT	67, 68
ESD FORMAT	30	TAPE SORT SAMPLE	29
EXCP	26	TAPE TO CARD	59
EXEC	21, 22	TAPE TO DISK	58
FLOATING POINT INSTR.	3	TAPE TO PRINTER	58
FOREGROUND INITIATION	17	TAPE TO TAPE	59
(EXAMPL)		TEP ADDRESS	75
HOLD COMMAND	19, 21	TEB TABLE	85
INCARD	61	TIMER	20
INDISK	61	TPLAB	21, 24
INLOG	62	TRANSIENT ROUTION	72
INSTRUCTION FORMAT	7 - 10	TXT FORMAT	30
INTAPE	61	UAN COMMAND	19
I/O MACRO	53	UCS COMMAND	19
I/O MIDULE NAME	38	UNA COMMAND	21
I/O TABLE	81, 82, 86	UPSI	22
JIB ADDRESS	75	UTILITY	54
JIB TABLE	81, 82	UTILITY JOB CARD	54
JOB	22	UTILITY LABEL CHECKING	55
JOB CONTROL COMMAND	18	UTILITY MACRO	61
JOB CONTROL SAMPLE	26	UTILITY MACRO (Rg. USAGE)	65, 66
JOB FLOW	16	UTILITY MODIFIER CARD	56
LABEL CYLINDER	70 - 71	UTILITY SAMPLE	27
LBLTYP	24	VOL	21, 22
LIBRARIAN	32	WRITE TAPE MARK	88
LISTIO	19, 21, 24	XTENT	21, 23





**IBM** Programming Handbook

**BOS/BPS**

**IBM**

*International Business Machines Corporation  
Field Engineering Division  
112 East Post Road, White Plains, New York*

Printed in U.S.A. 229 - 2129 - 0