

# TEXAS INSTRUMENTS

*Improving Man's Effectiveness Through Electronics*

## DX 980 General Purpose Operating System System Documentation

MANUAL NO 943015-9701  
ORIGINAL ISSUE 1 MARCH 1975  
REVISED AND ISSUED 15 MAY 1975

1 August 1975

Change 1

**Digital Systems Division**



The information and/or drawings set forth in this document and all rights in and to inventions disclosed herein and patents which might be granted thereon disclosing or employing the materials, methods, techniques or apparatus described herein are the exclusive property of Texas Instruments Incorporated.

No disclosure of the information or drawings shall be made to any other person or organization without the prior consent of Texas Instruments Incorporated.

INSERT LATEST CHANGED PAGES DESTROY SUPERSEDED PAGES

## LIST OF EFFECTIVE PAGES

Note: The portion of the text affected by the changes is indicated by a vertical bar in the outer margins of the page.

DX980 General Purpose Operating System (System Documentation)  
 (943015-9701)

Original Issue . . . . . 1 March 1975  
 Revised and Reissued . . . . . 15 May 1975 (ECN 388287)  
 Change 1 . . . . . 1 August 1975 (ECN 402623)

Total number of pages in this publication is 68 consisting of the following:

PAGE NO.	CHANGE NO.	PAGE NO.	CHANGE NO.	PAGE NO.	CHANGE NO.
Cover . . . . .	1	14 - 24. . . . .	0		
Eff. Pages . . . . .	1	25 . . . . .	1		
iii . . . . .	0	26 . . . . .	0		
iv . . . . .	1	27 - 38. . . . .	1		
1 - 5 . . . . .	0	39 - 60. . . . .	0		
6 - 7 . . . . .	1	Alphabetical			
8 . . . . .	0	Index Div . . . . .	0		
9 . . . . .	1	Index-1 -			
10 . . . . .	0	Index-2 . . . . .	0		
11 . . . . .	1	User's Resp . . . . .	1		
12 . . . . .	0	Bus. Reply . . . . .	0		
13 . . . . .	1	Cover . . . . .	0		



## TABLE OF CONTENTS

Paragraph	Title	Page
DX980 GENERAL PURPOSE OPERATING SYSTEM SYSTEM DOCUMENTATION		
1.0	General . . . . .	1
2.0	Kit Inspection . . . . .	1
3.0	Source Kit Overview . . . . .	1
3.1	PL/EXUS Language Specification . . . . .	1
3.2	PL/EXUS Support Tape . . . . .	1
3.3	DX980 Operating System, Linkable Parts Tape . . . . .	2
3.4	DX980 Operating System Source Tape . . . . .	2
4.0	Source Module Compilations and Assemblies . . . . .	3
4.1	Executing the PL/EXUS Compiler . . . . .	3
4.2	Executing the SAP Assembler . . . . .	3
5.0	Linking the DX980 Operating System . . . . .	3
6.0	System Documentation . . . . .	4
6.1	System Wide Events . . . . .	4
6.2	Event Attribute Table . . . . .	4
6.3	Control Block Linkage . . . . .	4
6.4	Virtual Buffer Sharing . . . . .	4
6.5	Wait Control Block Table . . . . .	4
6.6	Ready State Job . . . . .	4
6.7	Task Control Block . . . . .	4
6.8	Job Oriented Event Queue . . . . .	5
6.9	Disc Map . . . . .	5
6.10	Reserved Logical Unit Numbers . . . . .	5
6.11	Table of Supervisor Calls (SVC) . . . . .	5
6.12	DX980 Family Trees . . . . .	5
6.13	Macro Structure Definitions . . . . .	5

## LIST OF ILLUSTRATIONS

Figure	Title	Page
1	Control Block Linkage . . . . .	9
2	Virtual Buffer Sharing . . . . .	10
3	System Wide Event WCB Structure . . . . .	11
4	Ready State Job Structure . . . . .	12
5	Task Control Block Structure . . . . .	13



## LIST OF ILLUSTRATIONS (Continued)

Figure	Title	Page
6	DX980 Disc Map (All Disc Types) . . . . .	14
7	Job Management Family Tree . . . . .	27
8	File Management Family Tree . . . . .	31
9	Task Management Family Tree . . . . .	33
10	Operator Communications Family Tree . . . . .	35
11	I/O Management Family Tree . . . . .	37
12	Rollout Family Tree . . . . .	38
13	Rollin Family Tree . . . . .	38
14	JCL Interpreter Family Tree . . . . .	38
15	Memory Management Family Tree . . . . .	39
16	ITS Family Tree . . . . .	39
17	Batch Input Reader Family Tree . . . . .	40
18	Batch Input Spooler Family Tree . . . . .	40
19	Batch Output Spooler Family Tree . . . . .	40
20	CATALOG Family Tree . . . . .	41
21	CATFIL Family Tree . . . . .	41
22	IPRB and DDPRB macro Definition . . . . .	42
23	FIOPRB Macro Definition . . . . .	43
24	PDT and PDTEXT Macro Definitions . . . . .	44
25	FCB Macro Definition . . . . .	46
26	FCBLXT and FCBIXT Macro Definitions . . . . .	48
27	FRCBL, FRCBR and FRCBI Macro Definitions . . . . .	49
28	FMSTAT Macro Definition . . . . .	50
29	SVCDRT Macro Definition . . . . .	51
30	JSB Macro Definition . . . . .	52
31	JLDT Macro Definition . . . . .	54
32	JCB Macro Definition . . . . .	56
33	LDT Macro Definition . . . . .	57
34	JCM, CCB, JPCB and RTRDIR Macro Definitions . . . . .	58
35	TCB Macro Definition . . . . .	59
36	TEVENT and TEDB Macro Definitions . . . . .	60

## LIST OF TABLES

Table	Title	Page
1	System Wide Events . . . . .	6
2	Event Attribute Table (TEVTBL) Bit Description . . . . .	8
3	Reserved Logical Unit Numbers . . . . .	15
4	DX980 Supervisor Calls . . . . .	16
5	Alphabetical List of SVCs . . . . .	25



## DX980 GENERAL PURPOSE OPERATING SYSTEM SYSTEM DOCUMENTATION

### 1.0 GENERAL

The DX980 Source Kit, along with one of the Operating System Kits, provides all the items necessary to build a DX980 system from source modules. The same capability is provided for the Initial Program Load (IPL) procedure and the DX980 subsystems: Batch Input Reader (BIR), Batch Input Spooler (BIS), Batch Output Spooler (BOS), User File Directory Catalog Utility Routine (CATLOG), User File Catalog Utility Routine (CATFIL), and the Interactive Terminal Subsystem (ITS). Source modules are coded in either PL/EXUS or assembly language. PL/EXUS compiler support is included to support the modules of the first group.

### 2.0 KIT INSPECTION

In addition to this manual, the following items are included in the DX980 Source Kit.

- PL/EXUS Language Specification (part no. 944141-9901)
- PL/EXUS Support Tape (part no. 944140-1305)
- DX980 Operating System, Linkable Parts Tape (part no. 944140-1301)
- DX980 Operating System Source Tape (part no. 944140-2301)

If any of the above items are damaged or missing, contact a Texas Instruments field service representative for a replacement part.

### 3.0 SOURCE KIT OVERVIEW

The job control commands listed in the following paragraphs are examples only. Refer to the expanded JCL listing produced during Initial Program Loading (IPL) to obtain the defaults for particular system configurations.

#### 3.1 PL/EXUS LANGUAGE SPECIFICATION

This document describes the block-structured PL/EXUS Language, rules for its use, and a description of the listing produced.

#### 3.2 PL/EXUS SUPPORT TAPE

The PL/EXUS Language Support Tape contains all the items required to provide the DX980 system with PL/EXUS Compiler capability. The first file on this tape is a comment file that contains a description of the other items on



this tape and the procedure to be followed to utilize them. This file is listed with the following job control commands:

```
//JOBPLXLDSYSTEM
//RUNDXCOPY DLST=SC   DIN=MT1   DOUT=DUMMY
```

### 3.3 DX980 OPERATING SYSTEM, LINKABLE PARTS TAPE

This tape contains the linkable parts for all the modules of the DX980 system and all subsystems. Use the same job control as described for the PL/EXUS Support Tape to list the instruction file on this tape.

### 3.4 DX980 OPERATING SYSTEM SOURCE TAPE

This magnetic tape contains PL/EXUS and assembly language source modules for the DX980 system and subsystems. Individual modules are extracted from the tape with the DX980 Source Maintenance Routine (SMR). Refer to the DX980 General Purpose Operating System Programmer's Guide, Manual No. 943005-9701, for detailed instructions for this program. The procedure for extracting individual modules is as follows.

- a. The DX980 Operating system must be installed as outlined in the DX980 General Purpose Operating System - System Operation Guide (Manual No. 943004-9701).
- b. The Source Maintenance Routine is activated with the following job control commands:

```
//JOBDXSRCSYSTEM
//RUNSMR DCOM=DISC1 FCOM=(SYSTEM, SMRWRK) RCOM;
..LCOM=(2, 0, 128, 10) DLST=DUMMY
```
- c. A complete index of the source tape is requested by typing

```
.#INDEX
```
- d. An individual module may be extracted from the tape and placed in the file (SYSTEM, SMRWRK) with the following command:

```
.#COMPILE <nnnnnn>
```

where <nnnnnn> is the name of one of the source modules listed by the INDEX command.
- e. When the required module is extracted, terminate the Source Maintenance Routine with the command

```
.#ENDALL
```
- f. At this point, any of the language processors may be directed to process the source module on the file (SYSTEM, SMRWRK).
- g. Repeat steps b through f for all modules that are to be recompiled.



The linked control deck required to link the operating system from the linkable parts file (SYSTEM, DXLPF) is supplied on the source tape as module DX9LNK.

#### 4.0 SOURCE MODULE COMPILATIONS AND ASSEMBLIES

This section contains procedures for compiling or assembling source modules and for updating the linkable parts file.

##### 4.1 EXECUTING THE PL/EXUS COMPILER

Once a PL/EXUS module has been placed in the file (SYSTEM, SMRWRK) with the Source Maintenance Routine, the following job control is used to execute the PL/EXUS compiler.

```
//JOB PLEX SYSTEM
//RUN PLEXUP FUPD=(SYSTEM, DXLPF);
..FSRC=(SYSTEM, SMRWRK) MEM=(300, >94C0, 2500)
```

This procedure adds to or updates object modules in the linkable parts file (SYSTEM, DXLPF).

##### 4.2 EXECUTING THE SAP ASSEMBLER

The following job control is used to execute the SAP assembler:

```
//JOB ASMBLY SYSTEM
//RUN ASMUP FUPD=(SYSTEM, DXLPF);
..FSRC=(SYSTEM, SMRWRK)
```

#### 5.0 LINKING THE DX980 OPERATING SYSTEM

If system modules are modified, or if new system modules are added, the operating system must be linked. The following procedure performs that operation:

```
//JOB LINK SYSTEM
//RUN SMR DCOM=DISC1 FCOM=(SYSTEM, LINKDX) RCOM;
..LCOM=(2,0,128,10) DLST=DUMMY
```

Then enter

```
.#COMPILE DX9LNK
/*
```

This step makes the link control module (DX9LNK) accessible for the next step. The next step links the operating system. To initiate this step, enter

```
//JOB LINK SYSTEM
//RUN DXLINK FIN=(SYSTEM, LINKDX) RLM
```

This procedure performs the link of the linkable parts file (SYSTEM, DXLPF) and places the load modules in the file (USER01, DXMIP). It also places



the external definition table in file (USER01,DXEXTD). The newly generated system may be loaded using the IPL procedures by setting:

LM=(USER01,DXMIP).

## 6.0 SYSTEM DOCUMENTATION

The remainder of this manual consists of tables, diagrams and listings containing key concepts of the DX980 Operating System. The following paragraphs outline the subject matter of each of these items.

### 6.1 SYSTEM WIDE EVENTS

Table 1 lists the system wide events and their corresponding attributes and ID words. The ID words for event indexes 0 through 19 are the absolute address of some data as indicated in the table.

### 6.2 EVENT ATTRIBUTE TABLE

Table 2 describes the meaning of each bit in the event attribute word associated with each event index listed in table 1. This table exists in the system table area of the operating system under the name, TEVTBL.

### 6.3 CONTROL BLOCK LINKAGE

Figure 1 illustrates the linkage structure among the control blocks in the system for a single job.

### 6.4 VIRTUAL BUFFER SHARING

Figure 2 describes the sharing of virtual buffers in memory between two jobs that access the same file. This figure is valid for key indexed or blocked relative record files only.

### 6.5 WAIT CONTROL BLOCK TABLE

Figure 3 illustrates the structure of the Wait Control Block Table (TWATPT) in the system table area. This table is used for system wide events only.

### 6.6 READY STATE JOB

Figure 4 shows the structure of a job in the Ready state, as well as the relationship between a JSB and its JLDTs.

### 6.7 TASK CONTROL BLOCK

Figure 5 illustrates the structure and contents of a Task Control Block.





## 6.8 JOB ORIENTED EVENT QUEUE

The Job-oriented Event Queue (JEQ) is a linked list containing the event index and an ID for an event that is posted before a suspend occurs. If the event is savable, the posted event is saved until a suspend occurs or until the job terminates. The structure of the JEQ is as follows:

Word 0: A link pointer to the next JEQ.

Word 1: The event index

Word 2: The event ID

## 6.9 DISC MAP

Figure 6 illustrates the disc map for either a system or a non-system disc.

## 6.10 RESERVED LOGICAL UNIT NUMBERS

Table 3 lists the logical unit numbers that are reserved for use by the operating system, together with the function of each of the units.

## 6.11 TABLE OF SUPERVISOR CALLS (SVC)

Most of the modules of DX980 execute in a dynamic system overlay area, called the Procedure Pool. Modules in this group are accessible through either user or system supervisor calls. Table 4 provides a list of all SVCs in order by SVC number, memory image phase numbers, and a short description of each SVC. Table 5 is an alphabetical list of the SVCs.

In table 4, these abbreviations are used:

MIP	Memory Image Phase
SRS	System Root Segment

## 6.12 DX980 FAMILY TREES

Figures 7 through 21 show the interrelationship of DX980 system and subsystem modules. These family trees provide an excellent means to obtain an overview of the various DX980 management systems and subsystems.

## 6.13 MACRO STRUCTURE DEFINITIONS

The various tables and structures of the DX980 system are described in the Macro Structure Definitions shown in figures 22 through 36.



Table 1. System Wide Events

Index Number	User Waitable	User Postable	Single or Multiple Match	Relational Operator	Number of ID Words	ID Word(s)	Event Description
0	No	No	Single	=	1	PDT/FCB Address LDT Address TCB Address Device Name Address (in PDT)	Device/File Total Close LUN Idle "YES" Operator Response "YES" Operator Response
1	No	No	Single	=	1	PDT/FCB Address TCB Address Device Name Address (in PDT)	Device/File Unlock "NO" Operator Response "NO" Operator Response
2	No	No	Single	=	1	PDT/FCB Address JCB Address	Device/File Not Busy Job Step Terminated
3	No	No	Single	=	1	Console PDT Address	CTRL-O on the Console or Escape (ESC) key on the Console
4	No	No	Single	=	1	SVC Directory Entry Address	SVC Procedure Available
5	No	No	Single	=	1	PDT/FCB Address	Device/File Assigned/Deassigned
6	No	No	Single	=	1	Add. of reserve memory	Reserve area available for use
7	-	-	-	-	-	-	-
8	No	No	Single	=	1	JCB Address	Runtime Resource Allocate
9	No	No	Single	=	1	TCB Address	Task Scheduling
10	No	No	Multiple	=	1	SVC Directory Address PDT/FCB Address	Procedure Pool Load Device/File Idle
11	No	No	Multiple	=	1	PDT Address	Device Online
12	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-
20	No	No	Single	=	1	PDT/FCB Address	I/O Complete
21	No	No	Multiple	=	0	-	I/O Overload Removed
22	Yes	No	Multiple	>=	2	Double Precision Word in msec	Time-Of-Day

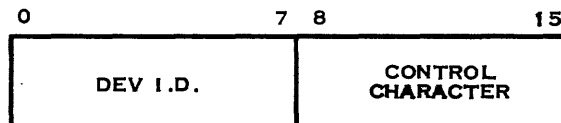


Table 1. System Wide Events (Continued)

Index Number	User Waitable	User Postable	Single or Multiple Match	Relational Operator	Number of ID Words	ID Word(s)	Event Description
23	Yes	No	Multiple	>=	2	Double Precision Time In Msec	Time Delay
24	Yes	No	Multiple	=	0	-	Any Job Step Termination
25	Yes	No	Multiple	=	1	Job Number	Job String Termination
26	No	No	Single	=	1	Job Number	Job String Start
27	No	No	Single	=	1	Job Number	Committed Memory Stealing
28	Yes	No	Single	=	1	LDT Address	Resource Runtime Allocate/ Deallocate
29	Yes	No	Multiple	=note 1	1	JCB Address note 2	Quieting of I/O note 2
30				Reserved			
31				Reserved			
32	-	-	-	-	-	-	-
33	-	-	-	-	-	-	-
34	-	-	-	-	-	-	-
35	Yes	Yes	Single	=	1		Open to User
36	Yes	Yes	Single	=	1		Open to User
37	Yes	Yes	Single	=	1		Open to User
38	Yes	Yes	Single	=	1		Open to User
39	Yes	Yes	Single	=	1		Open to User

Notes:

1. If Bits 0-7 of the event I. D. are not all ones, then the compare relational operator "=" is done on all of the event I. D. ; otherwise, it is only done on bits 8-15 of the event I. D.
2. The event I. D. is a 16-bit composite of an internal device I. D. and a control character.



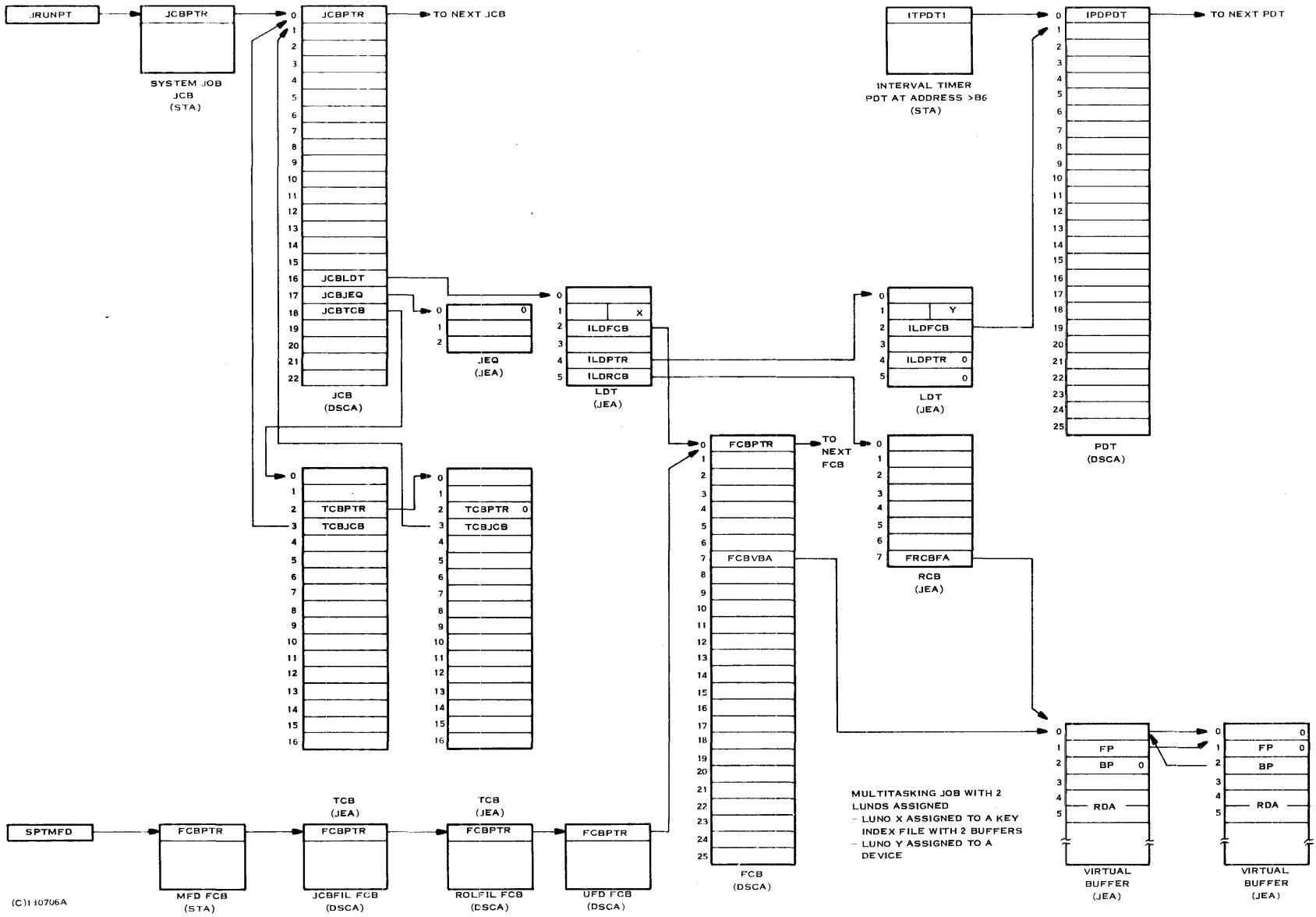
The internal device I. D. is an 8-bit field that identifies the data terminal from which the control character is generated. This field is set to FF<sub>16</sub> if the match is to be made on any data terminal generating the control character. The internal device I. D. may be obtained from the DX980 General Purpose Operating System Programmer's Guide, Manual Number 943005-9701.

The control character is a data terminal generated 7-bit pattern. Refer to the programmer's guide to determine which control characters are postable.



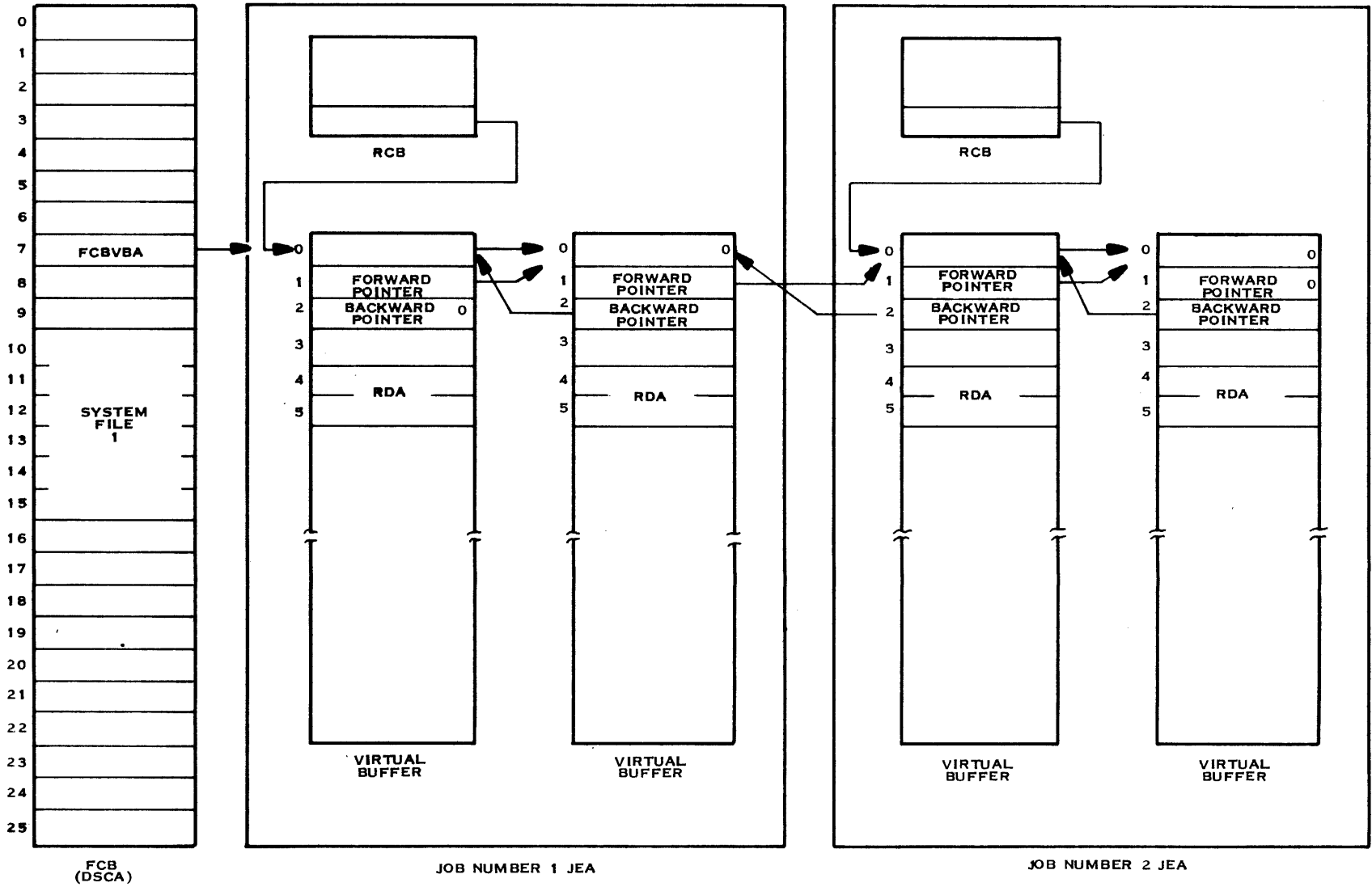
Table 2. Event Attribute Table (TEVTBL) Bit Description

BIT NO.	DESCRIPTION
0	Pure/impure event 0=pure event (no memory alteration) 1=impure event (memory alteration)
1	Job/system oriented event 0=job oriented event 1=system oriented event
2	Save job oriented event 0=do not save event 1=save event (when bit 1=0 only)
3	User waitable event 0=user may not wait for event 1=user may wait for event
4	User postable event 0=user may not post event 1=user may post event
5	Single (multiple match) 0=discontinue match search after one match 1=continue match search throughout link (when bit 2=0 only)
6	Multiple/single failure 0=continue match search throughout link 1=discontinue match search after one failure
7-8	Relational operator 00=equal 10=greater than/equal
9-11	WCB linking option (ignored for job-oriented events) 000=none 001=FIFO 010=LIFO 011=priority/FIFO 100=ascending
12-15	Number of ID words for event



(C) 110706A

Figure 1. Control Block Linkage



NOTES: 1. THE LINK POINTERS IN WORD 0 OF THE VIRTUAL BUFFER ARE LOCAL TO THE JOB

2. THE FORWARD AND BACK LINKS IN WORDS 1 AND 2 ARE LOCAL TO THE FCB

(B)130707

Figure 2. Virtual Buffer Sharing

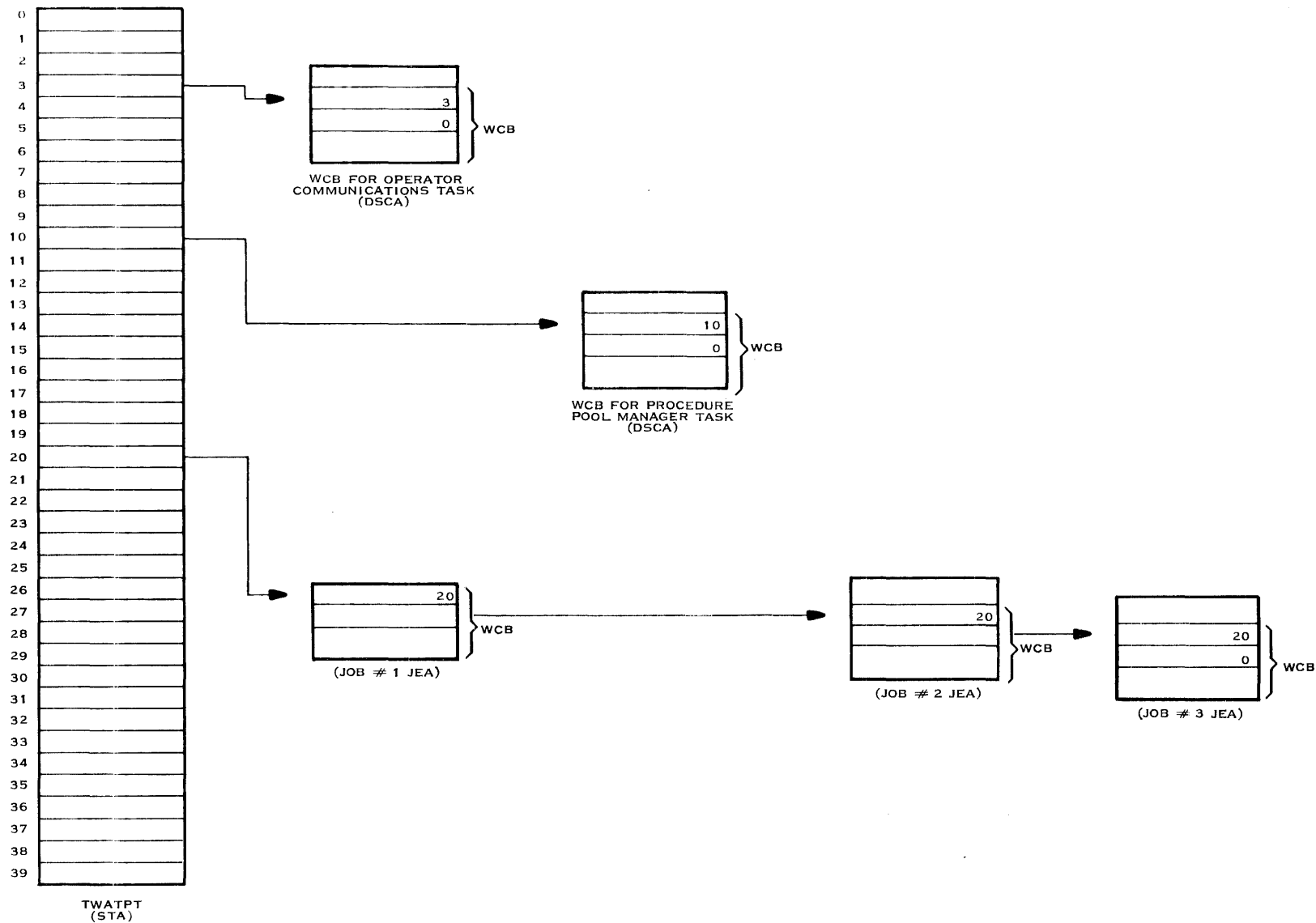


Figure 3. System Wide Event WCB Structure



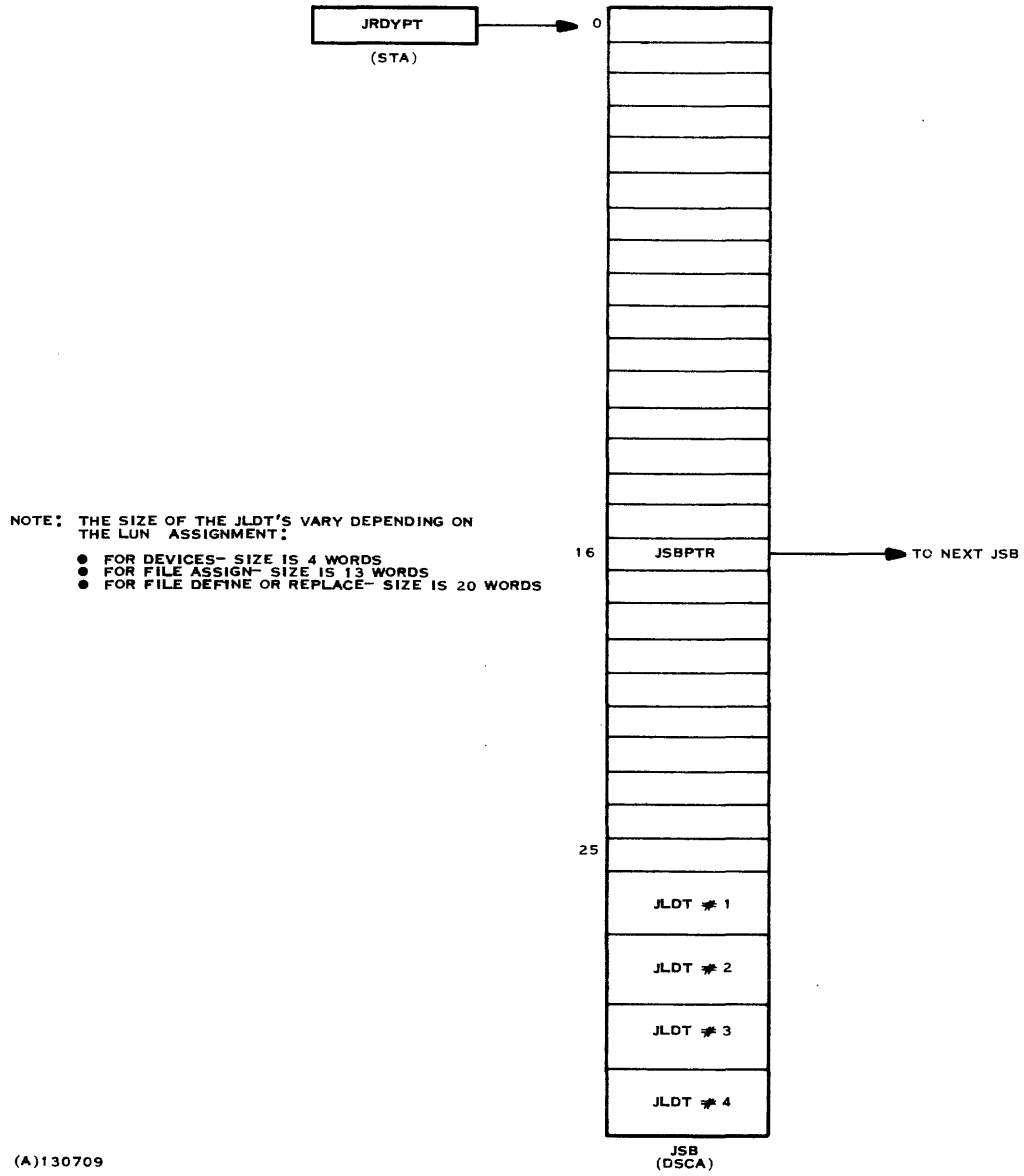


Figure 4. Ready State Job Structure



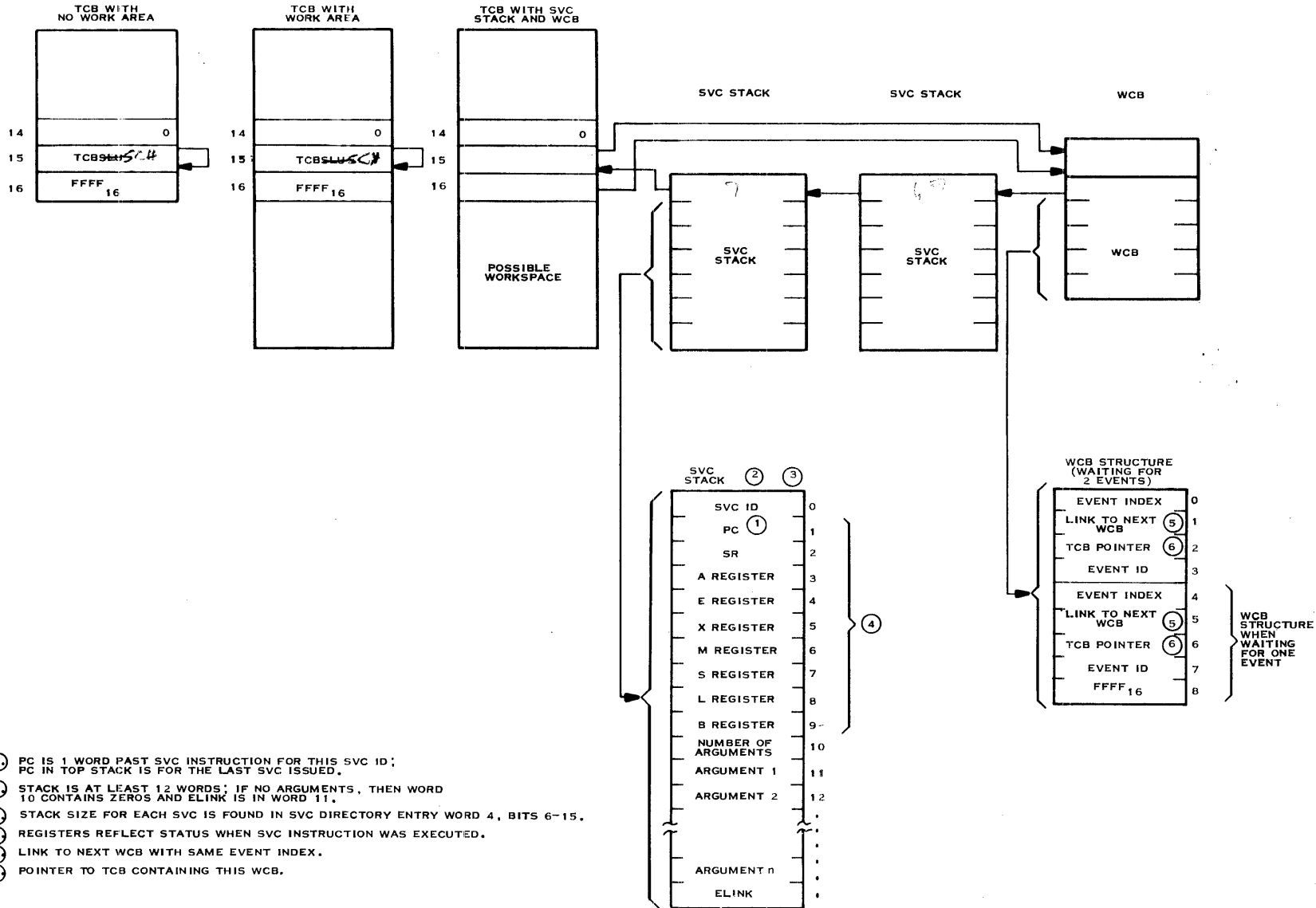
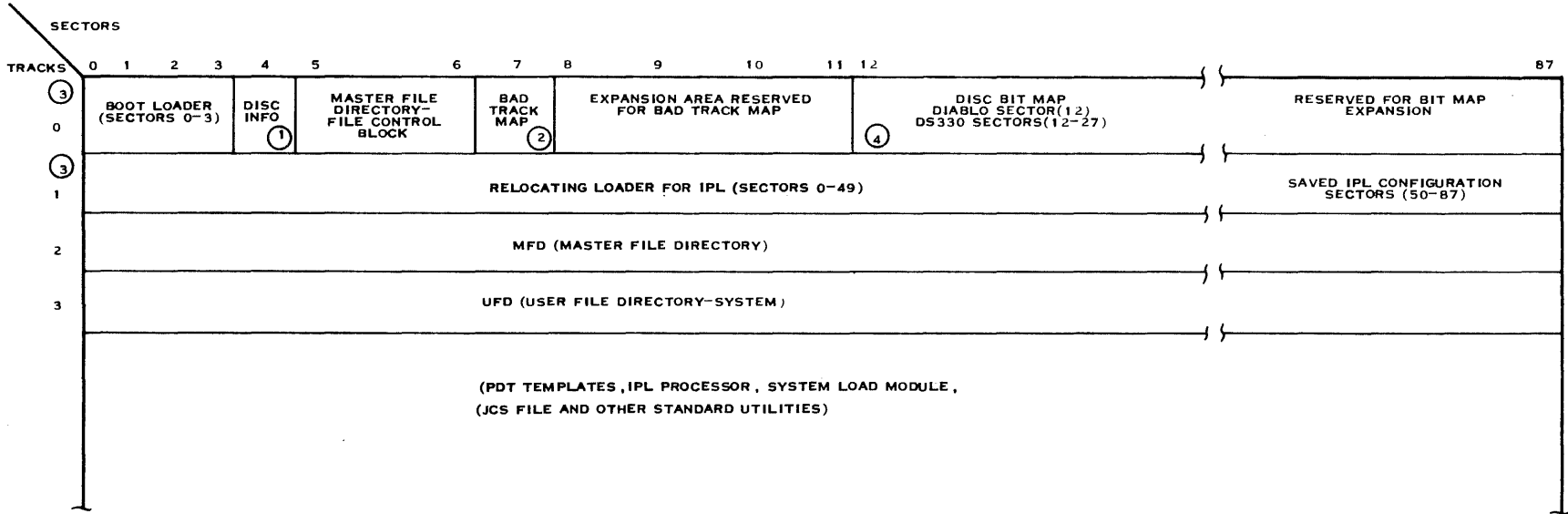


Figure 5. Task Control Block Structure





① SECTOR 4 - DISC INFORMATION

WORD	CONTENTS	NOMINAL VALUE
0	SECTOR CONTAINING START OF MFD-FCB	5
1	NOT USED	0
2	SECTOR CONTAINING START OF BAD TRACK LIST	7
3	SECTOR CONTAINING START OF DISC BIT MAP	12
4-9	DISC VOLUME ID	DX980REV*** (ASCII CODE)
10	NUMBER OF DISC BIT MAP SECTORS	1 FOR DIABLO, 16 FOR DS330
11	NOT USED	0
12	SECTOR CONTAINING START OF RELOCATING LOADER	88
13	LENGTH OF RELOCATING LOADER	32*50
14-19	TRACK 0 SECTOR BIT MAP (0=FREE; 1=USED)	FFFF <sub>16</sub> , FFFF <sub>16</sub> , FFE0 <sub>16</sub> , 0.0.0, ETC

③ FOR THE DS330 DISC: TRACKS 0 AND 1 ARE FORMATTED FOR 32 WORD PHYSICAL RECORDS. TRACK 2 IS FORMATTED FOR 96 WORD PHYSICAL RECORDS

② SECTOR 7 - BAD TRACK MAP

WORD	CONTENTS
0	NUMBER OF BAD TRACKS
1-30	TRACK ADDRESS OF BAD TRACKS
31	POINTER TO NEXT BAD TRACK MAP

④ SECTOR (12-27) - BIT MAP SECTORS

WORD	CONTENTS
0	BIT INDEX TO FIRST AVAILABLE TRACK IN THIS BIT MAP. INDEX IS RELATIVE TO BIT 0 OF WORD 1 OF THIS BIT MAP.
1-31	EACH BIT OF THESE WORDS REPRESENT THE AVAILABILITY OF A SINGLE DISC TRACK. BIT=0 INDICATES A FREE TRACK. BIT=1 INDICATES AN ALLOCATED OR BAD TRACK.

NOTE. THE FIRST BIT MAP SECTOR (SECTOR 12) MAPS TRACKS 0-495. THE SECOND MAP (SECTOR 13) MAPS TRACKS 496-991, ETC.

(B)130711

Figure 6. DX980 Disc Map (All Disc Types)



Table 3. Reserved Logical Unit Numbers

LUN	Reserved for Use By
0	System Console
1	System Logging Device
101 - 120	Direct Disc Input/Output (Disc Units 1 - 20)
121 - 140	Master File Directories (Disc Units 1 - 20)
141	Current User File Directory
160	JSB File
161	Roll File
254	JCL Procedures File
255	System Load Module



Table 4. DX980 Supervisor Calls

SVC No.	MIP No.	Description	Name
0*	SRS	I/O Request	IO
1*	SRS	End of Job	JSEND
2*	87	Set Floating-Point-Package Address	SFLSET
3*	53	Get Memory Limits	SGETML
4*	SRS	Set Control Status Flag	SABEND
5*	SRS	End Task	TSEND
6*	46	User Delete Task	TDLETE
7*	28	User Suspend Task	TPEND
8*	30	User Post Event	TPOST
9	17	Virtual Read	FMPVRD
10	26	Virtual Release	FMPVAL
11	25	Allocate Space for Logical Record and Physical Records (Index File)	FMPGET
12	1	Allocate Tracks	FMPTKG
13	21	Write/Write Replace/Replace (Index File)	FMWRIT
14	8	Key Find Routine (Index File)	FMPFND
15	3	Record Positioning (Index File)	FMPLAC
16	23	Delete/Delete Sequential (Index File)	FMDLT
17	22	Read/Read Delete/Read Hi/Read Lo (Index File)	FMREAD
18	7	Node Split (Index File)	FMNODE
19	24	Insert (Index File)	FMINST
20	32	Post-IPL Procedure	SPIPL
21	157	SOQI: Initialize System Output Queue	SOQI
22	33	OQOUT: System Console Message Output Procedure	OQOUT
23	158	SOQM: Add Unfinished Entry (No. 1)	BADNEW



Table 4. DX980 Supervisor Calls (Continued)

SVC No.	MIP No.	Description	Name
24	159	SOQM: Transfer to Finished Entry (No. 2)	BXFERE
25	160	SOQM: Delete First Finished Entry (No. 3)	BDELET
26	161	SOQM: Delete Specified Unfinished Entry (No. 4)	BDSE
27	144	JLDT Initializer, Part 1	JCBLD1
28	145	JLDT Initializer, Part 2	JCBLD2
29*	59	Time of Day and Date	STIMOD
30*	2	User Create Task	TCREAT
31	10	Relative Record File Read, Write	FMPRIO
32	27	Relative Record File Positioning	FMPOSN
33	15	Close Buffer Write	FMPREL
34	12	Linked Sequential File Read	FMPLSR
35	18	Linked Sequential File Write	FMPLSW
36	13	Linked Sequential File Positioning	FMPLPS
37*	143	LOAD SVC in PP	SULOD
38*	142	LOAD and Relocate SVC in PP	SULODR
39	34	System Console Processor	ODRIVR
40	35	Operator Console Diagnostic Output	ODIAGS
41*	41	Command Scanner in PP	CRSCAN
42	36	Reply to Diagnostic Message	OREPLY
43*	29	Wait for I/O Complete	IOWAIT
44	118	Job Termination	JSTRM
45	89	Message FCB for Load Module (L1, L2)	SETLMF
46	19	Label Table Lookup	CRLOOK
47	20	Command Scanner Descriptor Fetch	JFETCH
48	43	Initialize LDT	JCBLDA



Table 4. DX980 Supervisor Calls (Continued)

SVC No.	MIP No.	Description	Name
49*	113	User Allocate Resource	JALDT
50	5	System Allocate Resource	JSALDT
51*	114	User Deallocate Resource	JDLDT
52	44	System Deallocate Resource	JSDLDT
53	9	Track Release	FMPTKR
54	11	Define File	JFDFIN
55	4	Assign File	JFASGN
56	45	Delete File	JFDLET
57	14	Get UFD FCB	JGTUFD
58	16	Find DSCA Space	JFDSCA
59	115	Place JCB in DSCA or on File	JPDSCA
60	116	Get Pointer to JCB in DSCA	JGDSCA
61	117	Transfer JCB from Disc to Memory	JDTOM
62	122	Transfer JCB from Memory to Disc	JMTOD
63	90	Start Job	JSTART
64	180	Display Number of Procedure Pool Loads and Maximum DSCA size	OPPST
65	92	Schedule Resources	JSHEDR
66	93	Schedule Memory	JSHEDM
67	-	Reserved	-
68	95	Activate Job	JACTIV
69	96	Validate JSB	JVALID
70	97	Check Resource Availability	JAVAIL
71	98	Cancel Job by Job Number	JCANMD
72	-	Reserved	-
73	185	Job Management Lockout Routine	JMLOCK
74	SRS	Cancel Job by Number (PP Lockout)	JCANUM
75	99	Terminate Job String	JSTING
76	31	Lower Tier to User Suspend and Create Task	TWAIT



Table 4. DX980 Supervisor Calls (Continued)

SVC No.	MIP No.	Description	Name
77	137	Cancel Job	OCANCL
78	134	Execute a Cataloged Job Stream	ORUN
79	51	Modify, Display Time	OTIME
80	52	Modify, Display Date	ODATE
81	47	Online Device	OONLIN
82	49	Offline Device	OFFLIN
83	136	Display Job Status	OSTATS
84	135	Job Step Procedure	OJBSET
85	119	JCL Interpreter	JCLINT
86	85	ITS Device Initialization	ITSVC
87	82	Search for Device Index	JCLDEV
88	100	JCL Interpreter-Validator	JCLCKA
89	101	JCL Interpreter-Fixed Part JSB Validator	JCLCKE
90	120	JCL Interpreter-Key's Table Validator	JCLCKI
91	121	Initialize The Fields of the JSB	JBLDEX
92	-	Reserved	-
93	6	File Management Calls Pre- Processor	FPROCS
94	102	Delink Committed Memory	JDCOMM
95	-	Reserved	-
96	104	Create Job Management Message	JSCOMT
97	105	Cancel Job by Number	JSCAND
98*	54	Get Program Limits	SGETPL
99	56	Data Insert Module (Index File)	FMINMD
100	57	Exit for Inserts of Key and Data Records (Index File)	FMINEX
101	162	Generate File Number	BCFN



Table 4. DX980 Supervisor Calls (Continued)

SVC No.	MIP No.	Description	Name
102	163	Mark Entry Submitted	BMES
103	67	File Management Indexed File Interface	FMNDXP
104	68	File Management Relative Record File Interface	FMRLRP
105	69	File Management Linked Sequential File Interface	FMLSQP
106	70	File Management Open Routine	FMOPEN
107	71	File Management Close Routine	FMCLOS
108	50	Offline Disc	OF LNDD
109	48	Online Disc	ONLNDD
110	78	Delink FCB	JFDLNK
111	106	Post Job Resource Event	JWPOST
112	37	Dump Memory	ODPMEM
113	38	Modify Memory	OMDMEM
114	39	Dump Disc	ODDISC
115	40	Modify Disc	OMDISC
116	109	Batch Job Card Processor	BIRJOB
117	110	Reader Run Command	BIRRUN
118	111	Read Console for Assignment	OJCBPR
119	112	Get Device Number	BGTDEV
120	164	Batch Input Spooler File Initializer	BISFIL
121	165	Input Spooling and Conversion Routine	BCVT
122	156	Operator Command to BOS	OBOSCM
123	166	Batch Spooler Error Routine	BERR
124	73	Read Hi (Index File)	FMKYHI
125	74	Read Lo (Index File)	FMKYLO
126	75	Read Lo or Hi (Sequential) (Index File)	FMLOHI





Table 4. DX980 Supervisor Calls (Continued)

SVC No.	MIP No.	Description	Name
127	76	Write Sequential Entered (Index File)	FMIXSQ
128	107	Consolidate WCL's for Job Management Tasks	JCONSL
129*	108	User Start Job	JUSTAR
130	88	Data Terminal DSR (EIA Interface)	IDDT
131	168	Single/Multi-Disc Unit DSR	IDDMHD
132	SRS	Interval Timer DSR	IDITR
133	60	Card Reader DSR	IDCR
134	80	High Speed Paper Tape Punch DSR	IDPTP
135	81	High Speed Paper Tape Reader DSR	IDPTR
136	61	High Speed Line Printer DSR	IDLPH
137	66	Data Terminal Cassettes DSR	IDDTC
138	86	Low Speed Line Printer DSR	IDLPL
139	79	Magnetic Tape DSR	IDMT
140	103	Low Speed Paper Tape Reader DSR (EIA Interface)	IDPTRL
141	42	Low Speed Paper Tape Punch DSR (EIA Interface)	IDPTPL
142	62	Low Speed Paper Tape Reader DSR (TTY Interface)	IDPRLO
143	63	Low Speed Paper Tape Punch DSR (TTY Interface)	IDPPLO
144	155	Data Module DSR	IDDM
145	65	AD/DA Converter DSR	IDADDA
146	64	Data Terminal DSR (TTY Interface)	IDDTO
147	84	Reserved	-
148	169	Reserved	-
149	170	3330 Disc DSR	IDS330
150	SRS	DX980 - TX990 Communications Module DSR	IDCOMM



Table 4. DX980 Supervisor Calls (Continued)

SVC No.	MIP No.	Description	Name
151	-	Reserve	-
152	181	Reserve for DSR	IDSR1
153	182	Reserve for DSR	IDSR2
154	183	Reserve for DSR	IDSR3
155	184	Reserve for DSR	IDSR4
156	72	File Management Track Get - Single Bit Map - New File	FMTGSN
157	77	Single Bit Map - Old File	FMTGSO
158	178	Multiple Bit Maps - New File	FMTGMN
159	179	Multiple Bit Maps - Old File	FMTGMO
160	141	Delete All Processors	FMDELP
161	83	Reuse Deleted Space (Index File)	FMREUS
162	140	File Management Request Check/Validate Routine	FMCKEK
163	139	File Management Error Check/Handle Routine	FMERCK
164	126	Job Initiation - Allocate Resource/Activate	JKIKOF
165	127	Ready Job to Run	JREADY
166	128	Load LM	JLMASN
167	SRS	Terminate Job - Used to Lock Out Job Management	JSTORM
168	130	Allocate File	JFILAS
169	132	Allocate Device/File	JDOFAS
170	131	Allocate Device	JDEVAS
171	123	Define File, Part 1	JDTRAK
172	124	Define File, Part 2	JDUUFD
173	125	Define File, Part 3	JDUMFD
174		Reserved	



Table 4. DX980 Supervisor Calls (Continued)

SVC No.	MIP No.	Description	Name
175	-	Not Used	-
176	-	Not Used	-
177	167	Batch Format Routine (BOS)	BOHEAD
178	91	Initiate Job	JINITN
179	94	Allocate Job Resource	JRESOR
180	133	Initiate Job (PP Lockout)	JINLOK
181	138	Process Job Card for Batch and Operator Communication	OJOBPR
182	146	Invoke Lower Level Rollout Routine	RTROLO
183	-	Reserved	-
184	148	Look for Rollable Jobs	RTCRI2
185	149	Write Job to Roll Out File	RTWRIT
186	150	Quiet I/O for Rollout	RQUIET
187	151	Invoke Lower Level Rollback Routine	RTROBK
188	152	Read Rolled Job from Roll File	RTREAD
189	153	Link Up TCB's, JEQ's, and WCL's After Rollback	RELINK
190	-	Reserved	-
191	55	Runtime Allocate Resource	JUALDT
192	58	Runtime Deallocate Resource	JUDLDT
193	147	Delink Virtual Buffer from Chain	RDVBUF
194-199	-	Reserved	-
200-209	-	Reserved for user-implemented SVC's	-
210-255	-	Reserved	-

\*User Callable



Table 5. Alphabetical List of SVCs

SVC	Number	SVC	Number
BADNEW	23	FMPTKR	53
BCFN	101	FMPVAL	10
BCVT	121	FMPVRD	9
BDELET	25	FMREAD	17
BDSE	26	FMREUS	161
BERR	123	FMRLRP	104
BGTDEV	119	FMTGMN	158
BIRJOB	116	FMTGMO	159
BIRRUN	117	FMTGSN	156
BISFIL	120	FMTGSO	157
BMES	102	FMWRIT	13
BOHEAD	177	FPROCS	93
BXFERE	24	IDADDA	145
CRLOOK	46	IDCOMM	150
CRSCAN	41	IDCR	133
FMCHek	162	IDDM	144
FMCLOS	107	IDDMHD	131
FMDELP	160	IDDT	130
FMDLT	16	IDDTC	137
FMERCK	163	IDDTO	146
FMINEX	100	IDTR	132
FMINMD	99	IDLPH	136
FMINST	19	IDLPL	138
FMIXSQ	127	IDMT	139
FMKYHI	124	IDPPLO	143
FMKYLO	125	IDPRLO	142
FMLOHI	126	IDPTP	134
FMLSQP	105	IDPTPL	141
FMNDXP	103	IDPTR	135
FMNODE	18	IDPTRL	140
FMOPEN	106	IDSR1	152
FMPFND	14	IDSR2	153
FMPGET	11	IDSR3	154
FMPLAG	15	IDSR4	155
FMPLPS	36	IDS330	149
FMPLSR	34	IO	0
FMPLSW	35	IOWAIT	43
FMPOSN	32	ITSVC	86
FMPREL	33		
FMPRIO	31		
FMPTKG	12		



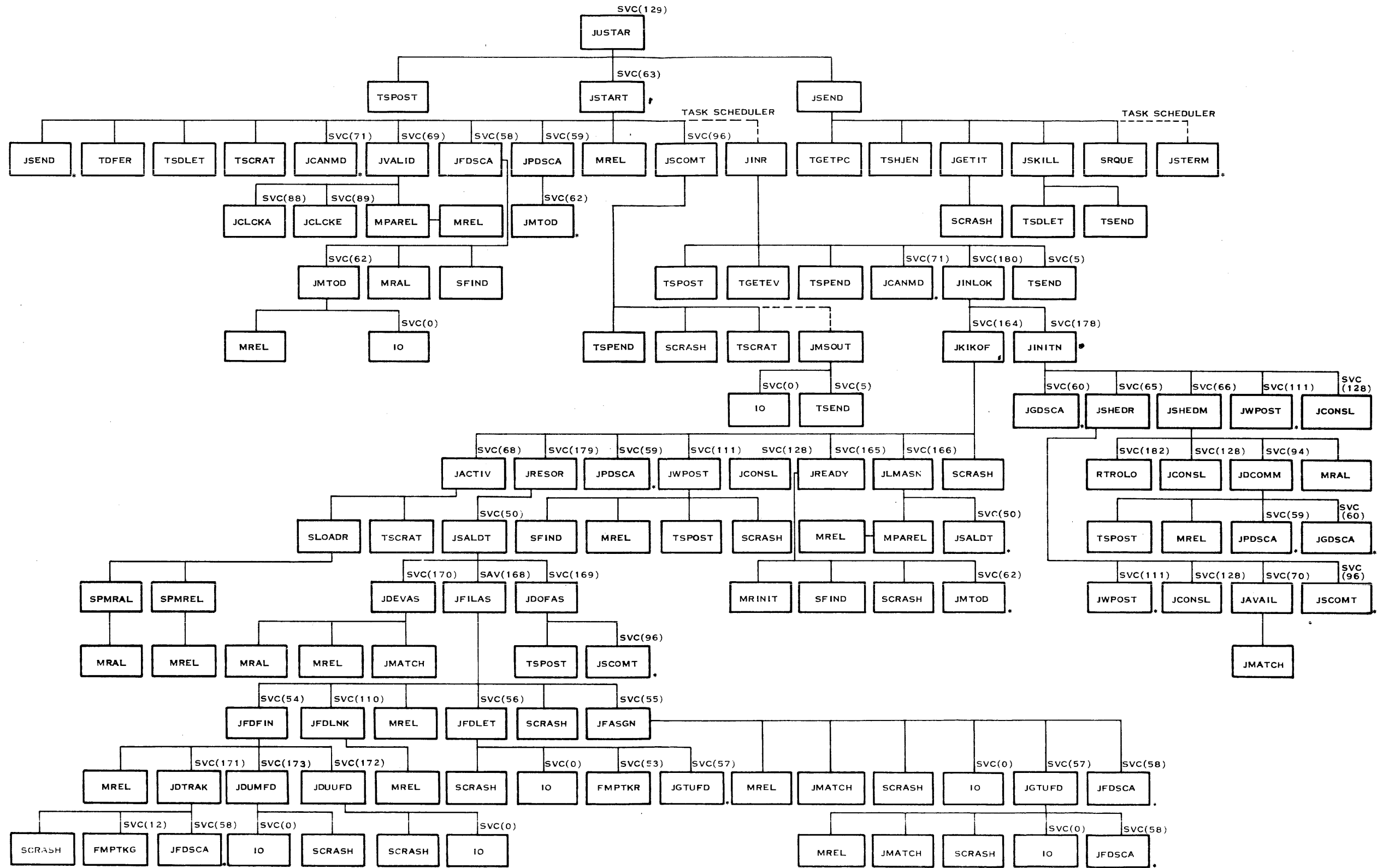
Table 5. Alphabetical List of SVCs (Continued)

SVC	Number	SVC	Number
JACTIV	68	JRESOR	179
JALDT	49	JSALDT	50
JAVAIL	70	JSCAND	97
JBLDEX	91	JSCOMT	96
JCANMD	71	JSDLDT	52
JCANUM	74	JSEND	1
JCBLDA	48	JSHEDM	66
JCBLD1	27	JSHEDR	65
JCBLD2	28	JSTART	63
JCLCKA	88	JSTING	75
JCLCKE	89	JSTORM	167
JCLCKI	90	JSTRM	44
JCLDEV	87	JUALDT	191
JCLINT	85	JUDLDT	192
JCONSL	128	JUSTAR	129
JDCOMM	94	JVALID	69
JDEVAS	170	JWPOST	111
JDLDT	51	OBOSCM	122
JDOFAS	169	OCANCL	77
JDTOM	61	ODATE	80
JDTRAK	171	ODDISC	114
JDUMFD	173	ODIAGS	40
JDUUFD	172	ODPMEM	112
JFASGN	55	ODRIVR	39
JFDFIN	54	OFFLIN	82
JFDLET	56	OFLNDD	108
JFDLNK	110	OJBSET	84
JFDSCA	58	OJCBPR	118
JFETCH	47	OJOBPR	181
JFILAS	168	OMDISC	115
JGDSCA	60	OMDMEM	113
JGTUFD	57	ONLNDD	109
JINITN	178	OONLIN	81
JINLOK	180	OPPST	64
JKIKOF	164	OQOUT	22
JLMASN	166	OREPLY	42
JMLOCK	73	ORUN	78
JMTOD	62	OSTATS	83
JPDSCA	59	OTIME	79
JREADY	165		



Table 5. Alphabetical List of SVCs (Continued)

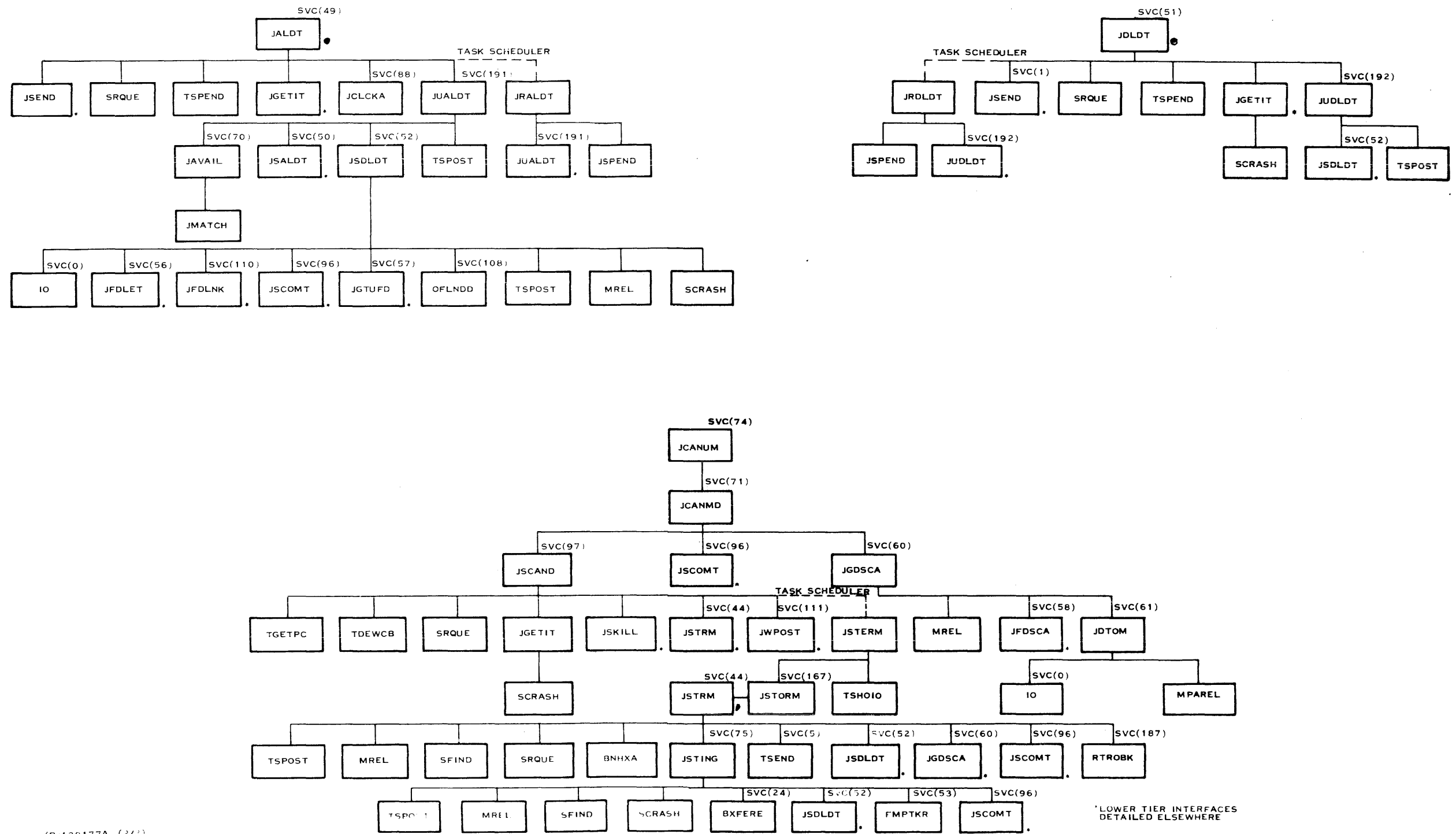
SVC	Number	SVC	Number
RDVBUF	193	SGETPL	98
RELINK	189	SOQI	21
RQUIET	186	SPIPL	20
RTCRI2	184	STIMOD	29
RTREAD	188	SULOD	37
RTROBK	187	SULODR	38
RTROLO	182	TCREAT	30
RTWRIT	185	TDLETE	6
SABEND	4	TPEND	7
SETLMF	45	TPOST	8
SFLSET	2	TSEND	5
SGETML	3	TWAIT	76



(B)130177A (1/2)

\*LOWER TIER INTERFACES DETAILED ELSEWHERE

Figure 7. Job Management Family Tree (Sheet 1 of 2)



(B)130177A (2/2)

Figure 7. Job Management Family Tree (Sheet 2 of 2)





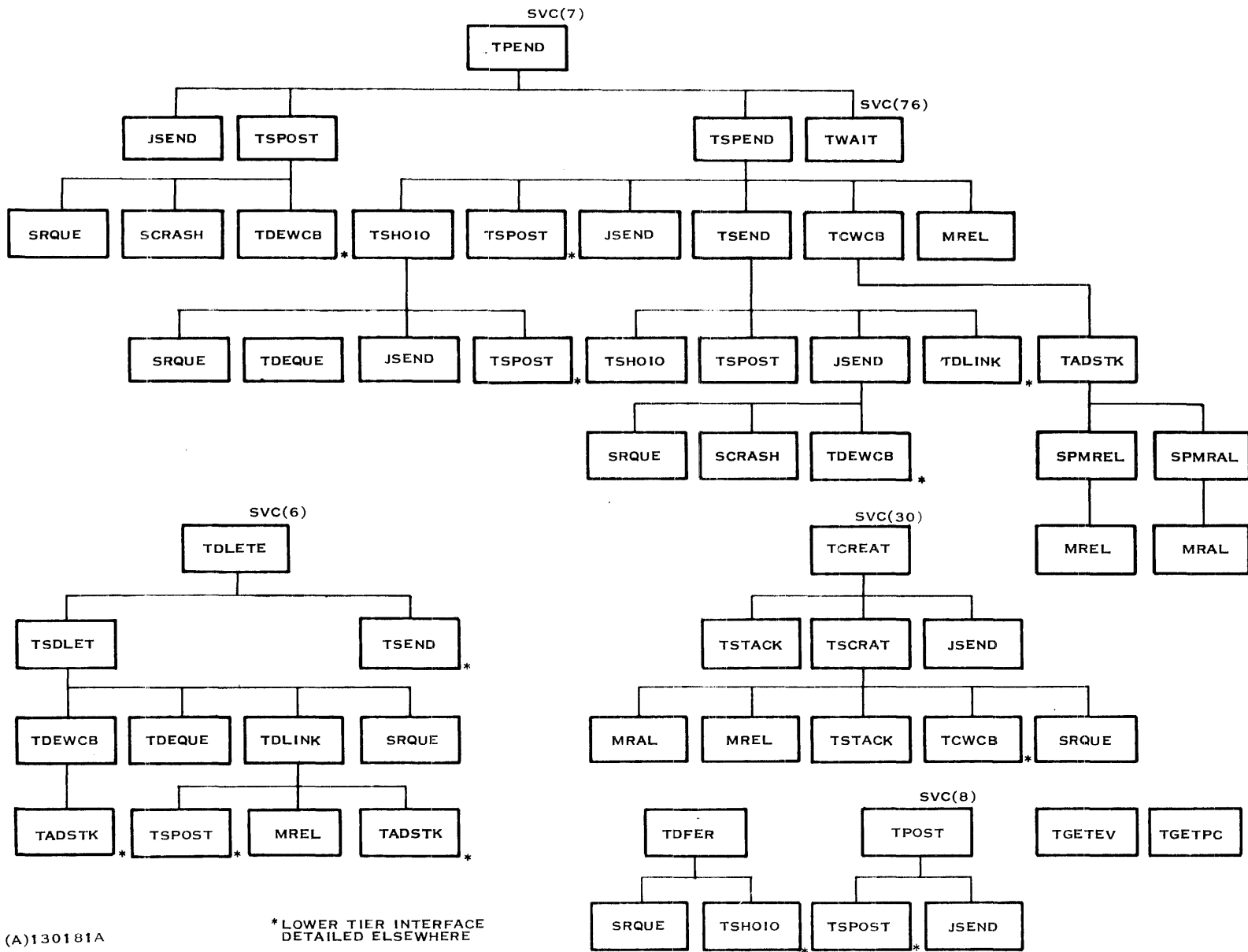
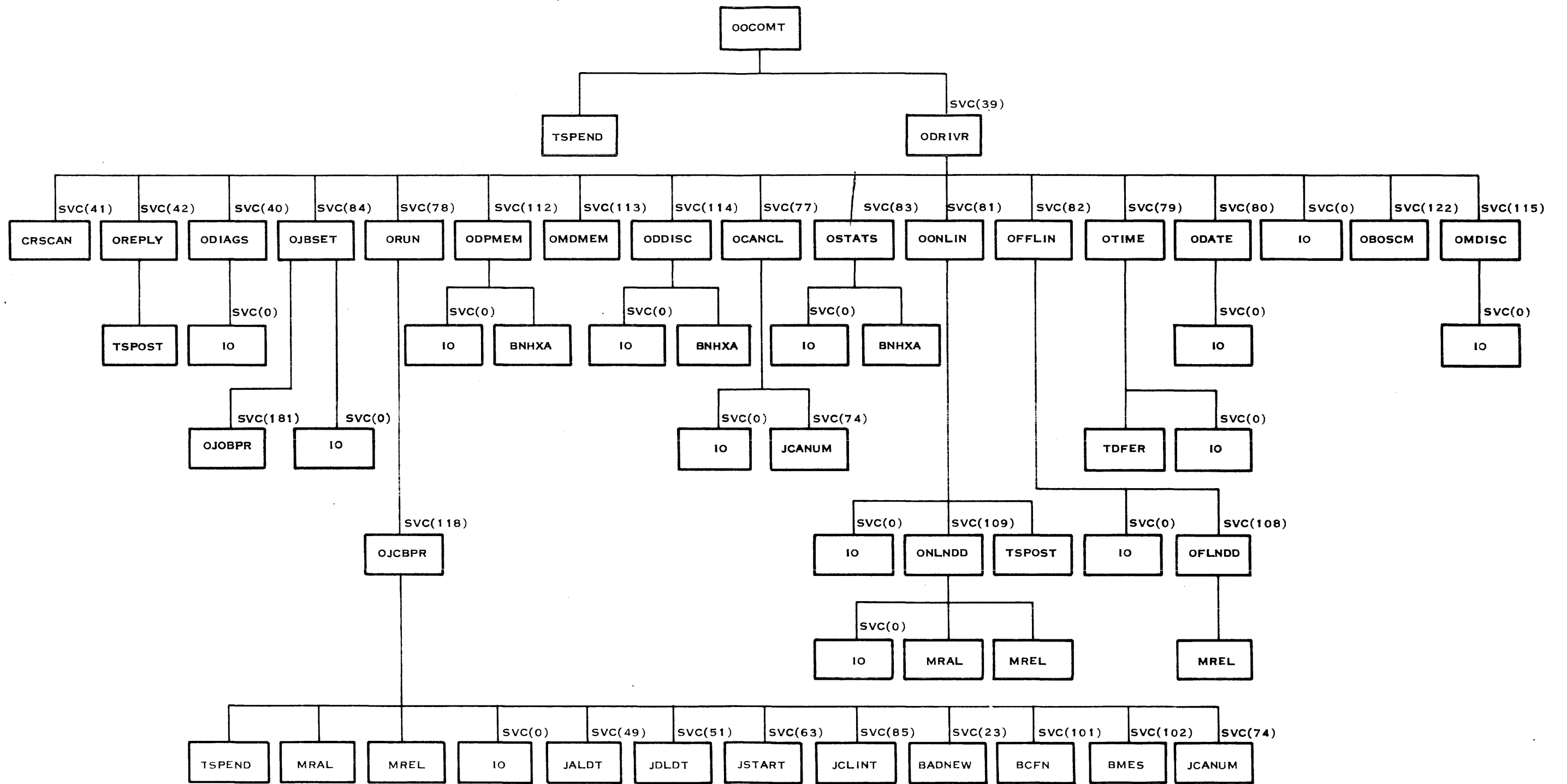


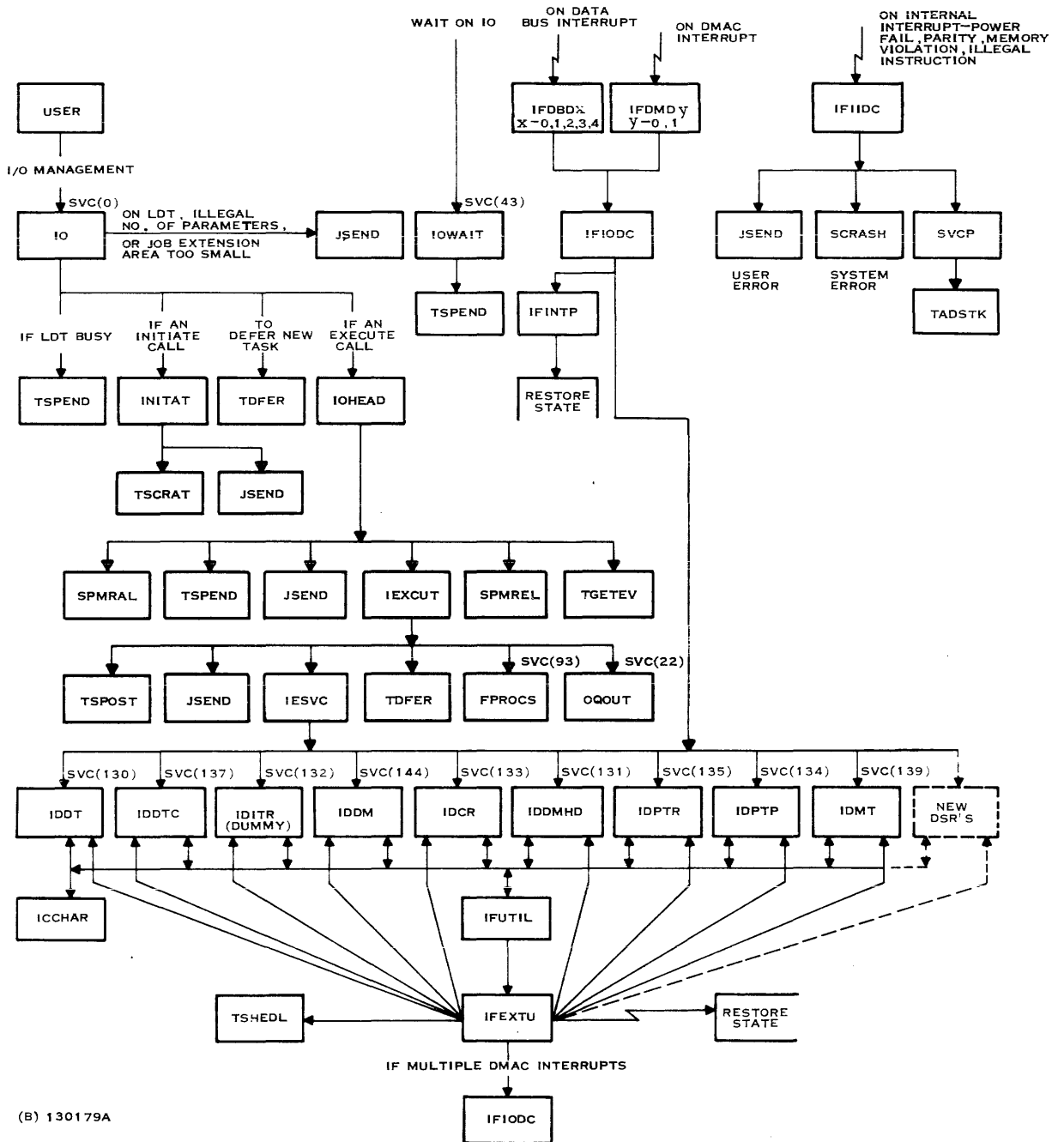
Figure 9. Task Management Family Tree





(B)130272A

Figure 10. Operator Communications Family Tree



(B) 130179A

Figure 11. I/O Management Family Tree

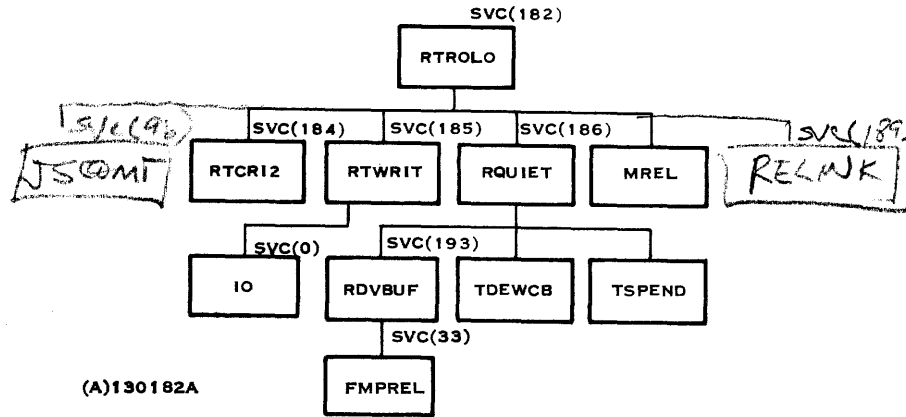


Figure 12. Rollout Family Tree

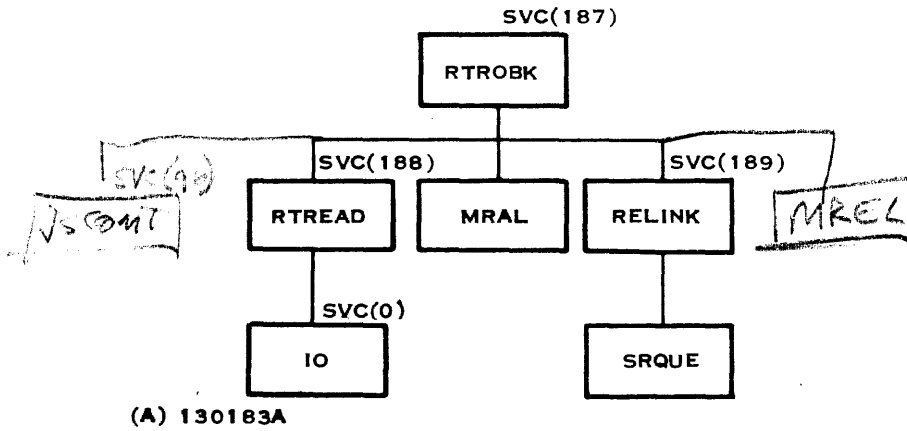


Figure 13. Rollin Family Tree

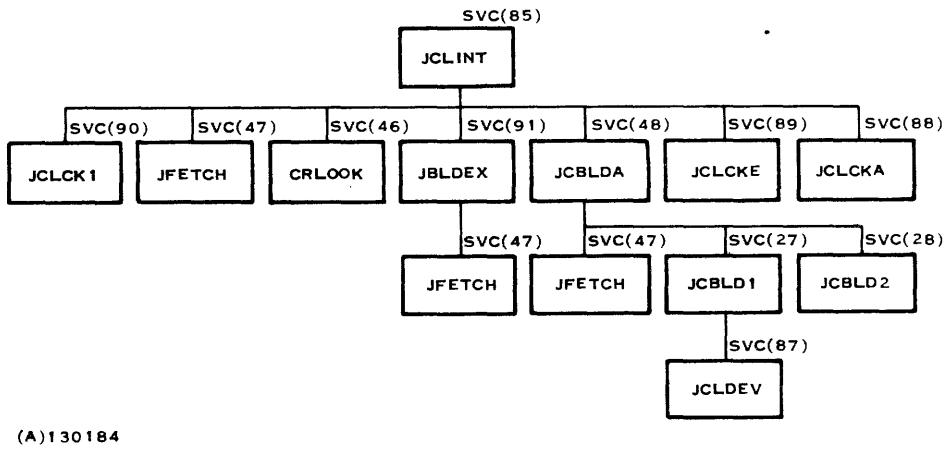
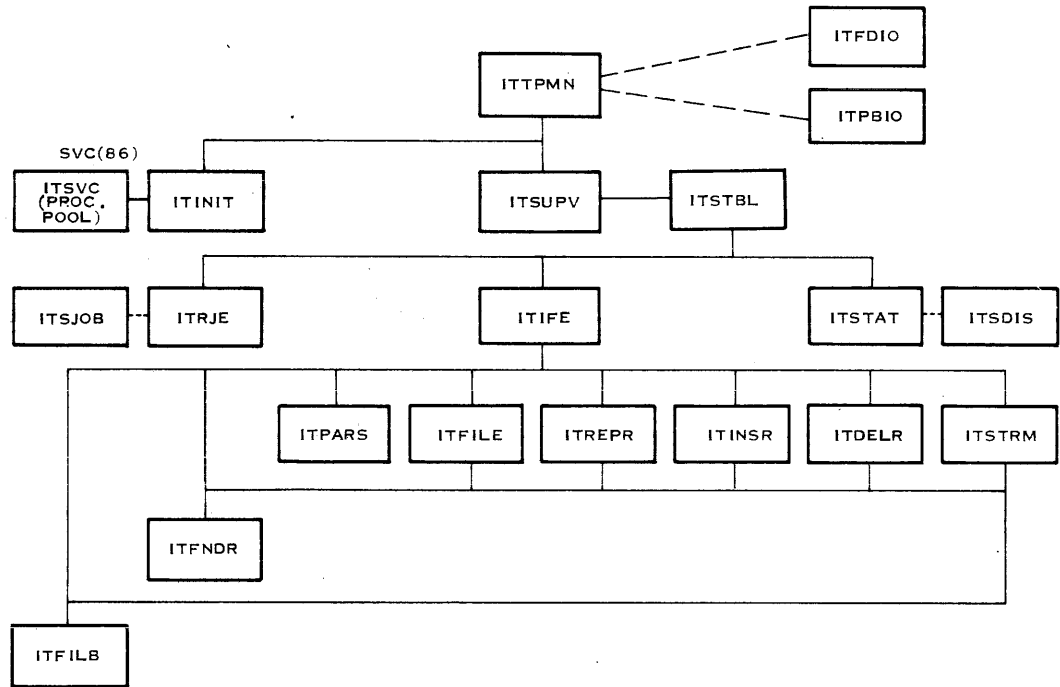


Figure 14. JCL Interpreter Family Tree



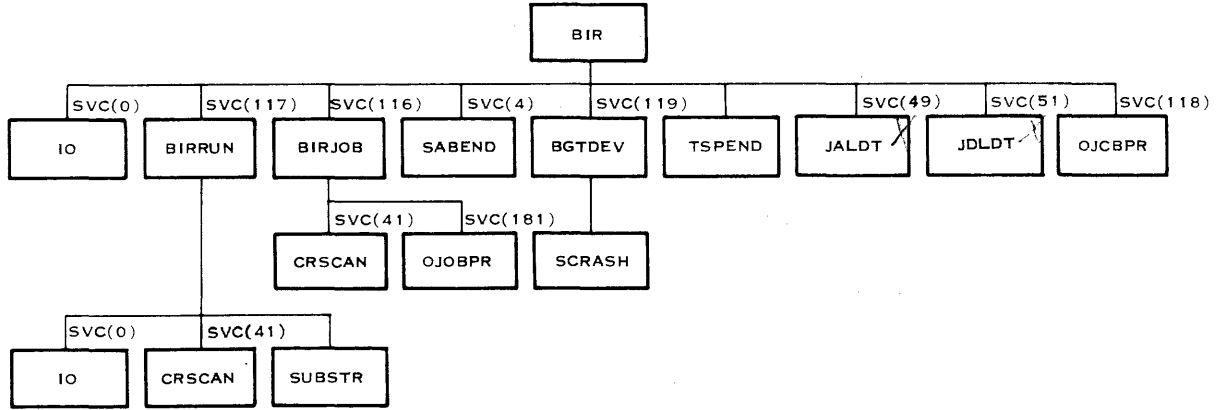
(A)130185

Figure 15. Memory Management Family Tree



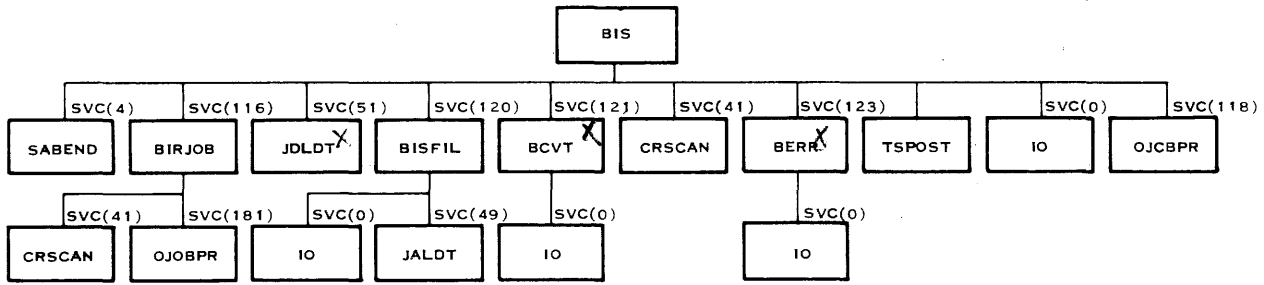
(A)130186

Figure 16. ITS Family Tree



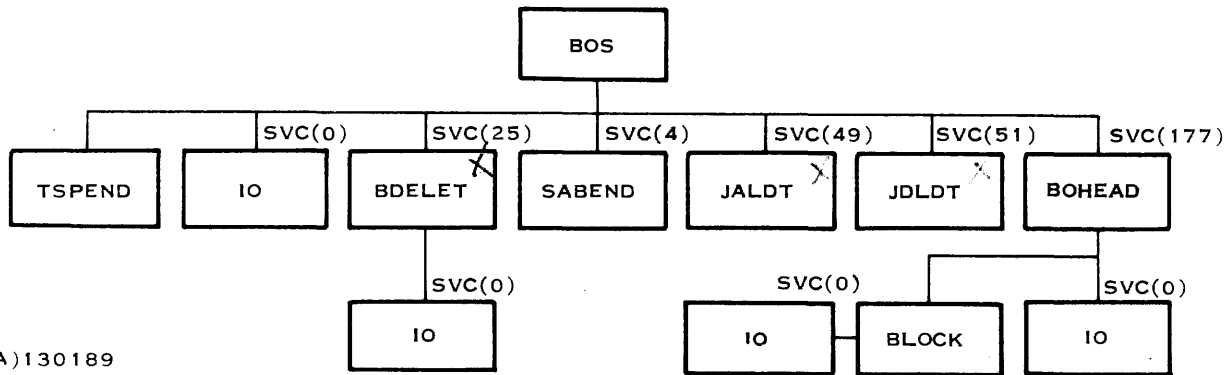
(A)130187

Figure 17. Batch Input Reader Family Tree



(A)130188

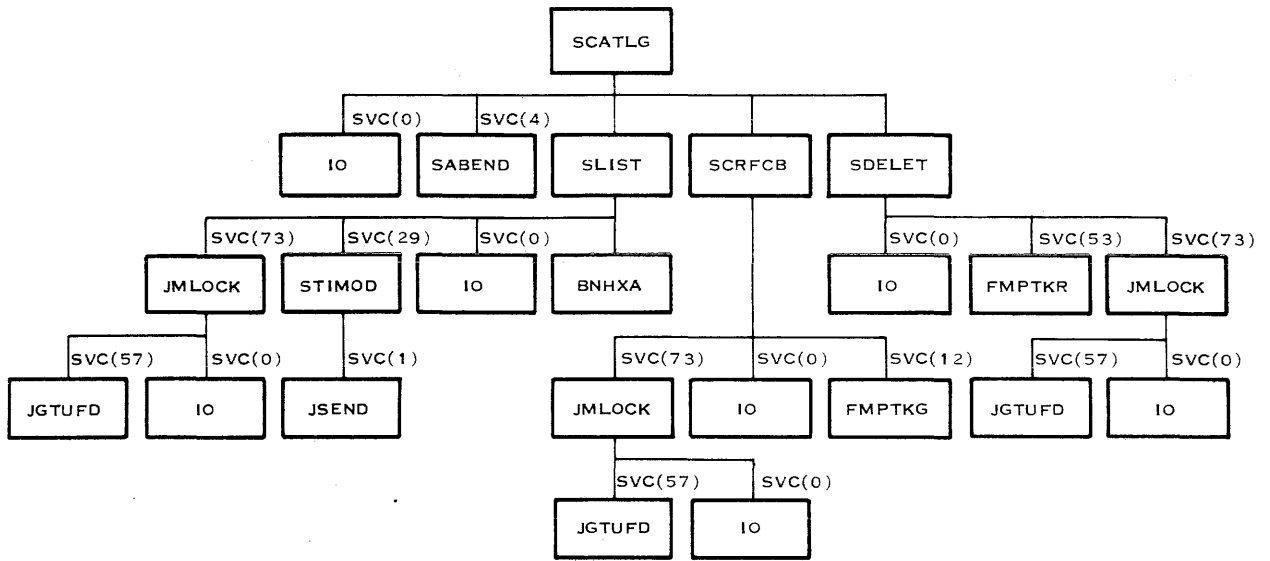
Figure 18. Batch Input Spooler Family Tree



(A)130189

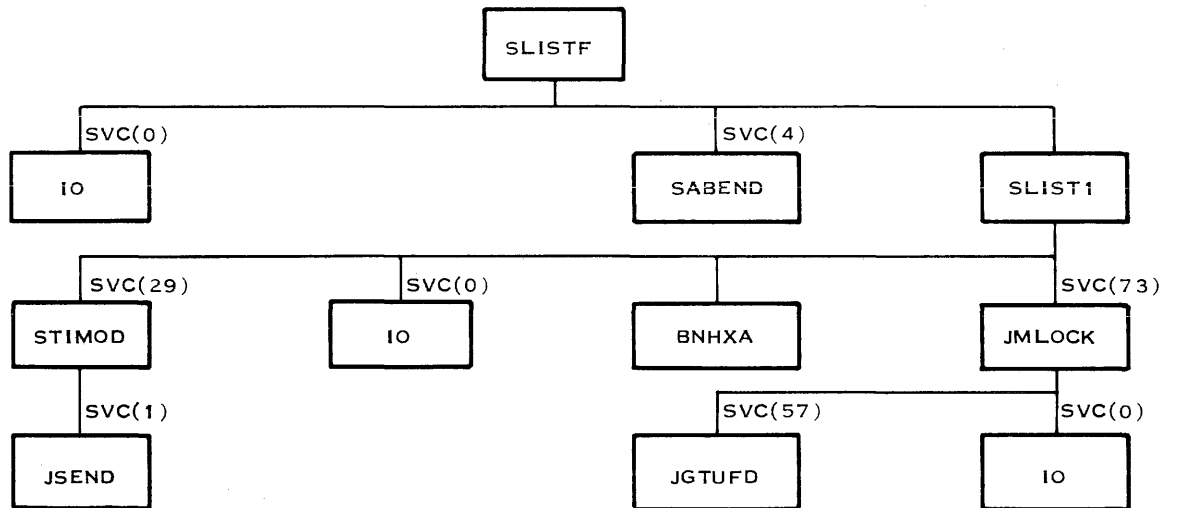
Figure 19. Batch Output Spooler Family Tree





(A)130270A

Figure 20. CATLOG Family Tree



(A)130271A

Figure 21. CATFIL Family Tree



```

/*****
/*
/*      STANDARD PHYSICAL RECORD BLOCK (IPRB)
/*
/*****
3 IPRWD0,          /* WORD 0
5 IPRSSF,          /* SYSTEM SET FLAGS
7 IPRBSY BIT(1),  /* IDLE/BUSY
7 IPRERR BIT(1),  /* NO ERROR/ERROR
7 IPREOF BIT(1),  /* NOT EOF/EOF
7 IPRIGN BIT(1),  /* OP NOT IGNORED/IGNORED
7 IPREOM BIT(1),  /* NOT EOM/EOM
7 IPRBOM BIT(1),  /* NOT BOM/BOM
7 IPRLER BIT(1),  /* LOGICAL ERROR
7 IPRSNU BIT(1),  /* NOT USED
5 IPRLUN BIT(8),  /* LOGICAL UNIT NUMBER
3 IPRWD1,          /* WORD 1
5 IPRUSF,          /* USER SET FLAGS
7 IPRITA BIT(1),  /* EXECUTE/INITIATE
7 IPRREP BIT(1),  /* NO REPLY/REPLY
7 IPRART BIT(1),  /* NO AUTO REC TERM/AUTO
7 IPRFAO BIT(1),  /* NO FORM ASCII OUT/FORM
7 IPRSCR BIT(1),  /* ECHO CR LF ON IN/NO ECH*
7 IPRBLR BIT(1),  /* RING BELL ON IN/NO BELL*
7 IPRULK BIT(1),  /* NO UNLOCK/UNLOCK DEVICE*
7 IPREAR BIT(1),  /* SHARED/EXCLUSIVE OPEN
5 IPROPC BIT(8),  /* I/O OP CODE
3 IPRLEN BIT(16), /* RECORD LENGTH
3 IPRBUF POINTER, /* BUFFER ADDRESS
3 IPRKEY POINTER, /* KEY ADDRESS
IPRLCK LITERALLY 'IPREAR' /* NO LOCK/LOCK DEVICE*
/*****
/*
/*      DIRECT DISC I/O PHYSICAL RECORD BLOCK (DDPRB)
/*
/*****
3 DDPSSF,          /* SYSTEM SET FLAGS
5 DDPBN1 BIT(1),  /* IDLE/BUSY
5 DDPERR BIT(1),  /* NO ERROR/ERROR
5 DDPBN2 BIT(6),  /* NOT USED 10/31/73
3 DDPLUN BIT(8),  /* LOGICAL UNIT NUMBER
3 DDPUSF,          /* USER SET FLAGS
5 DDPBN3 BIT(4),  /* NOT USED 10/31/73
5 DDPMXF BIT(1),  /* 1/>1 BUFFERS
5 DDWTVF BIT(1),  /* NO WRITE VERIFY/VERIFY
5 DDPBN4 BIT(2),  /* NOT USED 01/05/74
3 DDPOP BIT(8),   /* I/O OPCODE
3 DDPRL1 FIXED(16), /* 1ST BUFFER REC LENGTH
3 DDPBF1 POINTER, /* 1ST BUFFER ADDRESS
3 DDPDSK FIXED(31), /* TRANSFR START DA (SCTR)*
3 DDPRL2 FIXED(16), /* 2ND BUFFER REC LENGTH
3 DDPBF2 POINTER) /* 2ND BUFFER ADDRESS

```

Figure 22. IPRB and DDPRB Macro Definition



```

/*****
/*
/*      FILE I/O PHYSICAL RECORD BLOCK (FIOPRB)
/*
/*
/*****
    3 FLOWD0,          /* WORD 0
    5 FIOSSF,         /* SYSTEM SET FLAGS
      7 FIOBSY BIT(1), /* NOT BUSY/BUSY
      7 FIOERR BIT(1), /* NO ERROR/ERROR
      7 FIOEOF BIT(1), /* NOT EOF/EOF
      7 FIOIGN BIT(1), /* DP NOT IGNORED/IGNORED
      7 FIOEOM BIT(1), /* NOT EOM/EOM
      7 FIOBOM BIT(1), /* NOT BOM/BOM
      7 FIOLER BIT(1), /* LOGICAL ERROR
      7 FIOKRT BIT(1), /* KEY NOT RET/RET(INDEX)
    5 FIO LUN BIT(8), /* LOGICAL UNIT NUMBER
    3 FLOWD1,          /* WORD 1
    5 FIOUSF,         /* USER SET FLAGS
      7 FIOIIO BIT(1), /* EXECUTE/INITIATE
      7 FIOKSP BIT(1), /* KEY NOT SPEC/SPECIFIED
      7 FIOKRD BIT(1), /* NO KEY RECOV/RECOVERY
      7 FIORSO BIT(1), /* RETURN DATA REC SIZE
      7 FIO MRX BIT(1), /* SINGLE/MULT REC TRANSFR
                        /* SYSTEM SET ONLY
      7 FIOFWV BIT(1), /* NO WRITE VERIFY/VERIFY
      7 FUNLCK BIT(1), /* UNLOCK FILE/NO UNLOCK
      7 FIOEXA BIT(1), /* SHARED/EXCLUSIVE OPEN
    5 FIOCOD BIT(8), /* I/O OPCODE
    3 FIORL1 BIT(16), /* 1ST REC LENG (FIOKSP=1)
    3 FIORA1 POINTER, /* 1ST BUFR ADDR(FIOKSP=1)
    3 FIOKEY POINTER, /* KEY ADDRESS
    3 FIORL2 BIT(16), /* 2ND REC LENG (FIOMRX=1)
    3 FIORA2 POINTER, /* 2ND BUFR ADDR(FIOMRX=1)
    FLOCK LITERALLY 'FIOEXA' /* NO LOCK/LOCK FILE

```

Figure 23. FIOPRB Macro Definition



```

/*****
/*
/*      PHYSICAL DEVICE TABLE (PDT)
/*
/*****
3 IPDPDT POINTER,      /* NEXT PDT ADDRESS      */
3 IPDFLG,              /* PDT FLAGS              */
  5 IPDBSY BIT(1),     /* IDLE/BUSY              */
  5 IPDCMT BIT(1),     /* NOT COMMITTED/COMMITTED*/
  5 IPDCSH BIT(1),     /* EXCLUSIVE/SHARED COMMIT*/
  5 IPDASH BIT(1),     /* EXCLUSIVE/SHARED ASSIGN*/
  5 IPDOSH BIT(1),     /* SHARED/EXCLUSIVE OPEN  */
  5 IPDLOK BIT(1),     /* NOT LOCKED/LOCKED      */
  5 IPDOFL BIT(1),     /* ON LINE/OFF LINE       */
  5 IPDNSH BIT(1),     /* SHARABLE/NOT SHARABLE  */
  5 IPDEXT BIT(1),     /* NO EXT/EXTENDED PDT    */
  5 IPDITR BIT(1),     /* ENTER ON INT TIMER INTR*/
  5 IPDSTM BIT(1),     /* USER/SYSTEM MODE       */
  5 IPDCTU BIT(1),     /* CONTROL U FLAG         */
  5 IPDDFG BIT(4),     /* DSR FLAGS              */
3 IPDWD3,              /*
  5 IPDDEV BIT(8),     /* DEVICE INDEX           */
  5 IPDCPR BIT(8),     /* COMMIT PRIORITY        */
3 IPDWD4,              /*
  5 IPDLNA BIT(8),     /* ASSIGN COUNT           */
  5 IPDLNO BIT(8),     /* OPEN COUNT             */
3 IPDLLD POINTER,     /* LOCK TCB (IPDLOK=1)    */
3 IPDWD6,              /* ERROR DESC/DEVICE TYPE */
  5 IPDERR,            /* ERROR DESCRIPTOR       */
    7 IPDERT BIT(2),   /* ERROR TYPE             */
                          /* 00=LOGICAL ERROR      */
                          /* 01=PROGRAM ERROR      */
                          /* 10=I/O ERROR          */
                          /* 11=OP INTRVENT REQ    */
    7 IPDECD BIT(6),   /* ERROR CODE NUMBER      */
  5 IPDSVC BIT(8),     /* SVC INDEX              */
3 IPDATR,              /* DEVICE ATTRIBUTES      */
  5 IPDSYC BIT(1),     /* SYSTEM CONSOLE         */
  5 IPDDMY BIT(1),     /* DUMMY DEVICE           */
  5 IPDREW BIT(1),     /* REWINDABLE             */
  5 IPDFSP BIT(1),     /* FORWARD SPACABLE       */
  5 IPDBSB BIT(1),     /* BACK SPACABLE          */
  5 IPDPRT BIT(1),     /* PRINTING DEVICE        */
  5 IPDCAS BIT(1),     /* ASR733 CASSETTE        */
  5 IPDDTM BIT(1),     /* DATA TERMINAL         */
  5 IPDDSC BIT(1),     /* DISC                   */
  5 IPDINP BIT(1),     /* INPUT                   */
  5 IPDOUT BIT(1),     /* OUTPUT                  */

```

Figure 24. PDT and PDTEXT Macro Definitions (Sheet 1 of 2)



```
5 IPDASC BIT(1),      /* ASCII          */
5 IPDBIN BIT(1),     /* BINARY         */
5 IPDANU BIT(1),     /* NOT USED       */
5 IPDDFW BIT(2),     /* DEVICE/FILE DESCRIPTOR */
                    /* 00=DEVICE      */
                    /* 01=LINKED SEQ FILE */
                    /* 10=REL RECORD FILE */
                    /* 11=INDEXED FILE */
3 IPDNAM CHARACTER(6), /* DEVICE NAME    */
3 IPDEXP POINTER,    /* EXT PDT ADDR (IPDEXT=1)*/
3 IPDLFT,            /*                */
5 IPDLDF BIT(8),     /* I/O LOAD FACTOR */
5 IPDTMO BIT(8),     /* DEVICE TIMEOUT (SEC) */
3 IPDDSR POINTER,    /* DSR ADDRESS     */
3 IPDPRB POINTER,    /* PRB ADDRESS     */
3 IPDLDT POINTER,    /* LDT ADDRESS     */
3 IPDEL POINTER,     /* I/O DONE EVENT LINK */
3 IPDBUF POINTER,    /* BUFFER ADDRESS  */
3 IPDINT POINTER,    /* UNSOLIC INTRPT PROCESS */
3 IPDOCC FIXED(16),  /* OUTPUT CHARACTER COUNT */
3 IPDSCT FIXED(16),  /* NO. SECTORS/TRACK */
3 IPDWRD FIXED(16),  /* NO. WORDS/TRACK */
IPDTRK LITERALLY 'IPDOCC' /* NO. TRACKS/DISC */
/*****/
/*
/* EXTENDED PHYSICAL DEVICE TABLE (PDTEXT)
/*
/*****/
3 IPENUS FIXED(16),  /* NOT USED       */
3 IPEPDT POINTER,    /* PDT BACK POINTER */
3 IPENU1 FIXED(16),  /* NOT USED       */
3 IPEFLG,            /* EXTENSION FLAGS */
5 IPEBSY BIT(1),     /* CONTROLLER BUSY */
5 IPES70 BIT(1),     /* S700 SERIES D.T. */
5 IPEBNB BIT(1),     /* BUSY BUT NOT FOR SYSTEM */
5 IPEFNU BIT(13))    /* NOT USED       */
```

Figure 24. PDT and PDTEXT Macro Definitions (Sheet 2 of 2)



```

/*****
/*
/*      FILE CONTROL BLOCK (FCB)
/*
/*****
0 3 FCBPTR POINTER,      /* ADDRESS OF NEXT FCB      */
1 3 FCBFLG,              /* RUNTIME FLAGS            */
    5 FCBSY BIT(1),      /* NOT BUSY/BUSY            */
    5 FCBCMT BIT(1),     /* NOT COMMITTED/COMMITTED */
    5 FCBCSH BIT(1),     /* EXCLUSIVE/SHARED COMMIT */
    5 FCBASH BIT(1),     /* EXCLUSIVE/SHARED ASSIGN */
    5 FCBOSH BIT(1),     /* SHARED/EXCLUSIVE OPEN   */
    5 FCBLOK BIT(1),     /* NOT LOCKED/LOCKED      */
    5 FCBOFL BIT(1),     /* ON LINE/OFF LINE        */
    5 FCBNSH BIT(1),     /* SHARABLE/NOT SHARABLE   */
    5 FCBEXT BIT(1),     /* NO EXT/EXTENDED PDT     */
    5 FCBCHG BIT(1),     /* FCB NOT CHANGED/CHANGED */
    5 FCBNU1 BIT(2),     /* NOT USED 11/28/73       */
    5 FCBDFG BIT(4),     /* DSR FLAGS                */
2 3 FCBWD3,              /*                          */
    5 FCBVOL BIT(8),     /* DISC VOLUME ID          */
    5 FCBCPR BIT(8),     /* COMMIT PRIORITY         */
3 3 FCBWD4,              /*                          */
    5 FCBLNA BIT(8),     /* LUN'S ASSIGNED COUNT    */
    5 FCBLNO BIT(8),     /* LUN'S OPENED COUNT      */
4 3 FCBLLD POINTER,     /* LOCK TCB PTR(IPDLOK=1)  */
5 3 FCBERR,              /* ERROR DESCRIPTOR        */
    5 FCBERT BIT(2),     /* ERROR TYPE:             */
                          /* 00=LOGICAL ERROR        */
                          /* 01=ABORT - PROGRAM ERR  */
                          /* 10=ABORT UNLESS RETURN  */
                          /*   REQ--I/O TYPE ERROR  */
                          /* 11=ASK FOR RETRY        */
    5 FCBECD BIT(6),     /* ERROR CODE NUMBER       */
    5 FCBDTP BIT(8),     /* NOT USED 09/25/74       */
6 3 FCBATR,              /* DEVICE ATTRIBUTES       */
    5 FCBSYC BIT(1),     /* SYSTEM CONSOLE          */
    5 FCBDMY BIT(1),     /* DUMMY DEVICE            */
    5 FCBREW BIT(1),     /* REWINDABLE              */
    5 FCBFSP BIT(1),     /* FORWARD SPACABLE        */
    5 FCBBSB BIT(1),     /* BACK SPACABLE           */
    5 FCBPRT BIT(1),     /* PRINTING DEVICE         */
    5 FCBBKR BIT(1),     /* BLOCK REVERSIBLE        */

```

Figure 25. FCB Macro Definition (Sheet 1 of 2)



```
5 FCBANU BIT(7), /* NOT USED 11/28/73 */
5 FCBDFW BIT(2), /* DEVICE/FILE DESCRIPTOR */
/* 00=DEVICE */
/* 01=LINKED SEQ FILE */
/* 10=REL RECORD FILE */
/* 11=INDEXED FILE */
7 3 FCBVBA POINTER, /* BUFR LINK START ADDR */
8 3 FCBFFG, /* FILE FLAGS */
5 FCBPRS BIT(8), /* PHYS REC LENGTH(SECTRS)*/
5 FCBOV BIT(1), /* FILE MAX TRK OVERFLOW */
5 FCBNWF BIT(1), /* NEW FILE */
5 FCBFN2 BIT(4), /* NOT USED 10/15/73 */
5 FCBTYP BIT(2), /* FILE TYPE-SEE FCBDFW */
9 3 FCBPDT POINTER, /* POINTER TO DISC PDT */
10 3 FCBUID CHARACTER(6), /* USER ID */
13 3 FCBFIL CHARACTER(6), /* FILE NAME */
14 3 FCBPWD CHARACTER(4), /* PASSWORD */
18 3 FCBICD, /* INTEG CODE */
5 FCBRED, /* READ CODE */
7 FCBRC BIT(1), /* READ-CREATOR ONLY */
7 FCBRWP BIT(1), /* READ-USER PASSWORD */
7 FCBRNP BIT(1), /* READ-USER NO PASSWORD */
5 FCBWRT, /* WRITE CODE */
7 FCBWC BIT(1), /* WRITE-CREATOR ONLY */
7 FCBWWP BIT(1), /* WRITE-USER PASSWORD */
7 FCBWNP BIT(1), /* WRITE-USER NO PASSWORD */
5 FCBDLT, /* DELETE CODE */
7 FCBDC BIT(1), /* DELETE-CREATOR ONLY */
7 FCBDWP BIT(1), /* DELETE-USER PASSWORD */
7 FCBDNP BIT(1), /* DELETE-USER NO PASSWORD*/
5 FCREXC, /* EXECUTE CODE */
7 FCREC BIT(1), /* EXECUTE-CREATOR ONLY */
7 FCBEWP BIT(1), /* EXECUTE-USER PASSWORD */
7 FCRENP BIT(1), /* EXECUTE-USER NO PSWD */
5 FCBBN1 BIT(4), /* NOT USED 4/01/74 */
19 3 FCBIGN FIXED(16), /* INIT FILE SIZE (TRACKS)*/
20 3 FCBFDA BIT(16), /* 1ST DISC ADDR */
21 3 FCBPRW FIXED(16), /* PHYS REC LENGTH (WORDS)*/
22 3 FCBMXF FIXED(16), /* MAX FILE SIZE (TRACKS)*/
23 3 FCBLRL FIXED(16), /* LOG REC/KFY LENG (CHRS)*/
24 3 FCBLR1 BIT(16), /* REL REC LM,#1 REC LENG */
25 3 FCBLR2 BIT(16) /* REL REC LM,#2 REC LENG */
```

Figure 25. FCB Macro Definition (Sheet 2 of 2)



```

/*****
/*
/* FILE CONTROL BLOCK EXTENSION-LINKED SEQ FILES (FCBLXT) */
/*
/*****
      3 FCBEMP FIXED(31),      /* EOM POINTER (ABS SCTR)*/
      3 FCBEMI FIXED(16),     /* INDEX INTO FCBEMP(WRDS)*/
      3 FCBFFP FIXED(31),     /* 1ST EOF PTR (ABS SCTR)*/
      3 FCBFFI FIXED(16),     /* INDEX INTO FCBFFA(WRDS)*/
      3 FCBFST FIXED(16),     /* CUR FILE SIZE (TRACKS)*/
      3 FCBCPP FIXED(31),     /* CUR POS PTR (ABS SCTR)*/
      3 FCB CPI FIXED(16),    /* INDEX INTO FCB CPP(WRDS)*/
      3 FCBPFP FIXED(31),     /* PREV EOF PTR (ABS SCTR)*/
      3 FCBPFI FIXED(16),     /* INDEX INTO FCBPFP(WRDS)*/
      3 FCBNFP FIXED(31),     /* NEXT EOF PTR (ABS SCTR)*/
      3 FCBNFI FIXED(16)     /* INDEX INTO FCBNFP(WRDS)*/
/*****
/*
/* FILE CONTROL BLOCK EXTENSION-INDEXED FILES (FCBIXT) */
/*
/*****
      3 FCBIFD FIXED(31),     /* 1ST LOG DISC ADDR(SCTR)*/
      3 FCBIFI FIXED(16),     /* INDEX INTO FCBIFD(WRDS)*/
      3 FCBIEL FIXED(31),     /* END LOG (ABS SCTR)*/
      3 FCBIEI FIXED(16),     /* INDEX INTO FCBIEL(WRDS)*/
      3 FCBIBH FIXED(31),     /* B-TREEHEAD DA(ABS SCTR)*/
      3 FCBIEC FIXED(31),     /* END CHRON DA (ABS SCTR)*/
      3 FCBICS FIXED(16),     /* CUR FILE SIZE (TRACKS)*/
      3 FCBICA FIXED(31),     /* CUR ALLOC DA (ABS SCTR)*/
      3 FCBICI FIXED(16),     /* INDEX INTO FCBICA(WRDS)*/
      3 FCBINA FIXED(31)     /* NEXT AVAIL DA(ABS SCTR)*/

```

Figure 26. FCBLXT and FCBIXT Macro Definitions





```

/*****
/*
/*      RECORD CONTROL BLOCK-LINKED SEQ FILES (FRCBL)      */
/*
/*****
      3 FRCFLG,                /* RUNTIME FLAGS          */
      5 FRCFLRO BIT(1),        /* READ ONLY              */
      5 FRCFLMU BIT(1),        /* MULT USERS ON RECORD   */
      5 FRCFLFN BIT(14),       /* NOT USED 11/20/73     */
      3 FRCFLD POINTER,        /* LDT ADDRESS            */
      3 FRCFLDA FIXED(31),     /* PHYS REC D A (ABS SCTR)*/
      3 FRCFLDI FIXED(16),     /* INDEX INTO BRCLDA(WRDS)*/
      3 FRCFLRL FIXED(16),     /* INPUT REC LENG (BYTES)*/
      3 FRCFLFC FIXED(16),     /* NO. OF BUFRS (THIS LUN)*/
      3 FRCFLFA POINTER)       /* 1ST BUFR ADDR         */
/*****
/*
/*      RECORD CONTROL BLOCK-RELATIVE RECORD FILES (FRCBR)  */
/*
/*****
      3 FRCRFG,                /* RUNTIME FLAGS          */
      5 FRCRBLK BIT(1),        /* UNBLOCKED/BLOCKED REC */
      5 FRCRFNU BIT(15),       /* NOT USED 11/20/73     */
      3 FRCRLD POINTER,        /* LDT ADDRESS            */
      3 FRCRCA FIXED(31),      /* CUR DISC ADDR(ABS SCTR)*/
      3 FRCRCI FIXED(16),      /* INDEX INTO FRCRCA(WRDS)*/
      3 FRCRBK FIXED(16),      /* BUFFER COUNT           */
      3 FRCRBA POINTER,        /* 1ST BUFFER ADDR       */
      3 FRCNXT FIXED(16))      /* NEXT KEY               */
/*****
/*
/*      RECORD CONTROL BLOCK-INDEXED FILES (FRCBI)          */
/*
/*****
      3 FRCIFG,                /* RUNTIME FLAGS          */
      5 FRCIRO BIT(1),         /* READ ONLY              */
      5 FRCIMU BIT(1),         /* MULT USERS ON RECORD   */
      5 FRCIFN BIT(14),        /* NOT USED 10/15/73     */
      3 FRCILD POINTER,        /* LDT ADDRESS            */
      3 FRCIFDA FIXED(31),     /* PHYS REC D A (ABS SCTR)*/
      3 FRCIFDI FIXED(16),     /* INDEX INTO FRCIFDA(WRDS)*/
      3 FRCIURL FIXED(16),     /* INPUT REC LENG (BYTES)*/
      3 FRCIFFC FIXED(16),     /* NO. OF BUFRS (THIS LUN)*/
      3 FRCIFFA POINTER)       /* 1ST BUFFER ADDR       */

```

Figure 27. FRCBL, FRCBR and FRCBI Macro Definitions





```

/*****
/*
/*      SUPERVISOR CALL DIRECTORY (SVC DRT)
/*
/*****
3  SVC MNO FIXED(16),      /* MAX SVC NUMBER
3  SVC DEF(1),            /* SVC DEFINITION ENTRY
5  SVC WD1,               /* WORD 1
7  SVCLM BIT(8),         /* LOAD MODULE NUMBER
7  SVCEBS BIT(8),        /* EXCLUSIVE ENTRY INDEX
5  SVC WD2,               /* WORD 2
7  SVCBSY BIT(1),        /* SVC BUSY FLAG
7  SVCRES BIT(1),        /* RESIDENT FLAG
7  SVCPRV BIT(1),        /* PRIVILEGED ONLY FLAG
7  SVCLIP BIT(1),        /* LOAD IN PROGRESS FLAG
7  SVCMM BIT(1),         /* MIN/MAX REQUEST FLAG
7  SVCSCO BIT(1),        /* SCHEDULER OFF FLAG
7  SVCREN BIT(1),        /* REENRANT FLAG
7  SVCVAG BIT(1),        /* VAR NO OF ARGS FLAG
7  SVCUCT BIT(8),        /* USER COUNT
5  SVCEPT POINTER,      /* ENTRY POINT ADDRESS
5  SVC WD4,               /* WORD 4
7  SVCNAG BIT(5),        /* NUMBER OF ARGUMENTS
7  SVCEXT BIT(11))      /* AMOUNT OF STACK USED

```

Figure 29. SVC DRT Macro Definition



```

/*****
/*
/*      READY JOB CONTROL BLOCK (JSB)
/*
/*****
2 JSBPRT,          /* JCB FIXED PART
3 JSBSIZ BIT(16), /* TOTAL JCB SIZE
3 JSBFLG,          /* FLAG WORD
5 JSBEND BIT(1),  /* END OF STRING FLAG
5 JSBMOD BIT(1), /* UNPRIV/PRIV FLAG
5 JSBROL BIT(1), /* ROLL/NO ROLL FLAG
5 JSBWAT BIT(1), /* NOT WAITING/WAITING FLG
5 JSBRUN BIT(1), /* READY/RUNNING FLAG
5 JSBNU1 BIT(2), /* NOT USED
5 JSBRAL BIT(1), /* READY ALLOC ERROR FLAG
5 JSBBIR BIT(8), /* BIR DEVICE INDEX
3 JSBUID CHARACTER(6), /* JOB USER I.D.
3 JSBNAM CHARACTER(6), /* JOB NAME
3 JSBNOS,          /*
5 JSBNU2 BIT(12), /* NOT USED
5 JSBSTP BIT(4), /* JOB STEP NO
3 JSRSNP,          /*
5 JSBNU3 BIT(8), /* NOT USED
5 JSBPRI BIT(8), /* JOB PRIORITY
3 JSBTPL BIT(16), /* NO OF TASK PRI LEVELS
3 JSRUPS BIT(16), /* USER PARTITION SIZE
3 JSBJES BIT(16), /* JOB EXTENSION SIZE
3 JSBSTC BIT(16), /* DEFAULT TCB STACK SIZE
3 JSBTIM BIT(16), /* TIME LIMIT(SEC)
3 JSBSOQ POINTER, /* SYS OUTPUT Q ENTRY PTR
3 JSBPTR POINTER, /* NEXT READY JCB
3 JSBLMV BIT(16), /* LOAD MODULE VOLUME I.D.
3 JSBLMI CHARACTER(6), /* LOAD MODULE USER I.D.
3 JSBLMF CHARACTER(6), /* LOAD MODULE FILE NAME
3 JSBLMP CHARACTER(4), /* LOAD MODULE PASSWORD
2 JLDPRT(30),     /* JCB LOGICAL ASSIGN BLKS
3 JLDFLG,          /* LDT FLAG WORD
5 JLDNU1 BIT(1), /* NOT USED
5 JLDDOF BIT(1), /* DEVICE/FILE FLAG
/* NEXT BIT NEEDED FOR BLOCKABLE DEVICES ONLY
5 JLDLCK BIT(1), /* UNBLOCKED/BLOCKED FLAG
5 JLDEOS BIT(1), /* EXCLUSIVE/SHARED FLAG
5 JLOPOS BIT(1), /* NOT PASSED/PASSED FLAG
5 JLOFND BIT(1), /* DEVICE INDEX/ADDR FLAG
5 JLOPAS BIT(1), /* NO PASS/PASS FLAG
/* REMAINDER OF WORD NEEDED FOR FILES ONLY

```

Figure 30. JSB Macro Definition (Sheet 1 of 2)



```
5 JLDDEL BIT(1), /* NO DELETE/DELETE */
5 JLDADR BIT(2), /* ASN/DEF/REP FLAG */
5 JLDFTP BIT(2), /* FILE TYPE(NUL,LS,RR,KI)*/
5 JLDTMP BIT(1), /* NO TEMP/TEMP FILE FLAG */
5 JLDCRE BIT(1), /* USER/CREATOR (SYS SET) */
5 JLDNU4 BIT(2), /* NOT USED */
3 JLDWD1, /*
5 JLDTYP BIT(8), /* LDT TYPE FLAG */
5 JLDLUN BIT(8), /* LOGICAL UNIT NUMBER */
3 JLDDEV POINTER, /* DEVICE INDEX */
/* NEXT WORD NEEDED FOR BLOCKED DEVICES ONLY */
3 JLDBSZ FIXED(16), /* BLOCK SIZE */
/* REMAINDER NEEDED FOR FILE ASN/DEF/REP ONLY */
3 JLDLUN BIT(8), /* LOGICAL UNIT NUMBER */
3 JLDUID CHARACTER(6), /* FILE OWNER USER ID */
3 JLDFIL CHARACTER(6), /* FILE NAME */
3 JLOPWD CHARACTER(4), /* FILE PASSWORD */
/* REMAINDER NEEDED FOR FILE DEFINE/REPLACE ONLY*/
3 JLDICD, /* INTEGRITY CODE */
5 JLDRED, /* READ CODE */
7 JLDRC BIT(1), /* READ-CREATOR ONLY */
7 JLDRWP BIT(1), /* READ-USER PASSWORD */
7 JLDRNP BIT(1), /* READ-USER NO PASSWORD */
5 JLDWRT, /* WRITE CODE */
7 JLDWC BIT(1), /* WRITE-CREATOR ONLY */
7 JLDWWP BIT(1), /* WRITE-USER PASSWORD */
7 JLDWNP BIT(1), /* WRITE-USER NO PASSWORD */
5 JLDDLT, /* DELETE CODE */
7 JLDDC BIT(1), /* DELETE-CREATOR ONLY */
7 JLDDWP BIT(1), /* DELETE-USER PASSWORD */
7 JLDDNP BIT(1), /* DELETE-USER NO PASSWORD*/
5 JLDEXC, /* EXECUTE CODE */
7 JLDEC BIT(1), /* EXECUTE-CREATOR ONLY */
7 JLDEW BIT(1), /* EXECUTE-USER PASSWORD */
7 JLDENP BIT(1), /* EXECUTE-USER NO PSWD */
5 JLDNU5 BIT(4), /* NOT USED */
3 JLDIQN FIXED(16), /* INIT DISC QUAN(TRACKS) */
3 JLDFDA BIT(16), /* FIRST DISC ADDRESS */
3 JLDPRW FIXED(16), /* PHYS REC LENGTH(WORDS) */
3 JLDMXF FIXED(16), /* MAX FILE SIZE(TRACKS) */
/* NEXT WORD NEEDED FOR REL REC/INDEX FILES ONLY*/
3 JLDLRL FIXED(16), /* LOGICAL RECORD LENGTH */
/* NEXT WORD NEEDED FOR KEY INDEXED FILE ONLY */
JLDPDT LITERALLY 'JLDBSZ', /* PDT PTR (SYS SET) */
JLDKEY LITERALLY 'JLDLRL' /* KEY LENG(CHRS)(INDX)*/
```

Figure 30. JSB Macro Definition (Sheet 2 of 2)



```

/*****
/*
/*      ASSIGN-TIME LOGICAL DEVICE TABLE (JLDT)
/*
/*
/*****
3  JLDLFG,          /* LDT FLAG WORD          */
5  JLDNU1 BIT(1),  /* NOT USED                */
5  JLDDOF BIT(1),  /* DEVICE/FILE FLAG        */
    /* NEXT BIT NEEDED FOR BLOCKABLE DEVICES ONLY */
5  JLDLTK BIT(1),  /* UNBLOCKED/BLOCKED FLAG */
5  JLDEOS BIT(1),  /* EXCLUSIVE/SHARED FLAG  */
5  JLDPOS BIT(1),  /* NOT PASSED/PASSED FLAG */
5  JLDFND BIT(1),  /* DEVICE INDEX/ADDR FLAG */
5  JLDPAS BIT(1),  /* NO PASS/PASS FLAG       */
    /* REMAINDER OF WORD NEEDED FOR FILES ONLY */
5  JLDDEL BIT(1),  /* NO DELETE/DELETE        */
5  JLDADR BIT(2),  /* ASN/DEF/REP FLAG        */
5  JLDFTP BIT(2),  /* FILE TYPE(NUL,LS,RR,KI)*/
5  JLDTMP BIT(1),  /* NO TEMP/TEMP FILE FLAG */
5  JLDCRE BIT(1),  /* USER/CREATOR (SYS SET) */
5  JLDNU4 BIT(2),  /* NOT USED                 */
3  JLDWD1,          /*                          */
5  JLDTYP BIT(8),  /* LDT TYPE FLAG           */
5  JLDLUN BIT(8),  /* LOGICAL UNIT NUMBER     */
3  JLDDEV POINTER, /* DEVICE INDEX            */
    /* NEXT WORD NEEDED FOR BLOCKED DEVICES ONLY */
3  JLDBSZ FIXED(16), /* BLOCK SIZE              */
    /* REMAINDER NEEDED FOR FILE ASN/DEF/REP ONLY */
3  JLDBUF FIXED(16), /* NUMBER OF BUFFERS       */
3  JLDUID CHARACTER(6), /* FILE OWNER USER ID     */
3  JLDFIL CHARACTER(6), /* FILE NAME                */
3  JLDPWD,          /* FILE PASSWORD           */
5  JLDD2 CHARACTER(4), /*                          */
    /* REMAINDER NEEDED FOR FILE DEFINE/REPLACE ONLY */
3  JLDICD,          /* INTEGRITY CODE          */
5  JLDRED,          /* READ CODE                */
    7  JLDRC BIT(1), /* READ-CREATOR ONLY       */
    7  JLDRWP BIT(1), /* READ-USER PASSWORD      */
    7  JLDRNP BIT(1), /* READ-USER NO PASSWORD   */
5  JLDWRT,          /* WRITE CODE               */
    7  JLDWC BIT(1), /* WRITE-CREATOR ONLY      */
    7  JLDWWP BIT(1), /* WRITE-USER PASSWORD     */
    7  JLDWNP BIT(1), /* WRITE-USER NO PASSWORD  */

```

Figure 31. JLDT Macro Definition (Sheet 1 of 2)



```
5 JLDDLT, /* DELETE CODE */
7 JLDDC BIT(1), /* DELETE-CREATOR ONLY */
7 JLDDWP BIT(1), /* DELETE-USER PASSWORD */
7 JLDDNP BIT(1), /* DELETE-USER NO PASSWORD*/
5 JLDEXC, /* EXECUTE CODE */
7 JLDEC BIT(1), /* EXECUTE-CREATOR ONLY */
7 JLDEW BIT(1), /* EXECUTE-USER PASSWORD */
7 JLDENP BIT(1), /* EXECUTE-USER NO PSWD */
5 JLDNUS BIT(4), /* NOT USED */
3 JLDIGN FIXED(16), /* INIT DISC QUAN (TRACKS)*/
3 JLDFDA BIT(16), /* FIRST DISC ADDRESS */
3 JLDPRW FIXED(16), /* PHYS REC LENGTH (WORDS)*/
3 JLDMXF FIXED(16), /* MAX FILE SIZE (TRACKS)*/
/* NEXT WORD NEEDED FOR REL REC/INDEX FILES ONLY*/
3 JLDLRL FIXED(16), /* LOG REC LENG (CHRS)(RR)*/
JLDPDT LITERALLY 'JLDBSZ', /* PDT PTR (SYS SET) */
JLDKEY LITERALLY 'JLDLRL' /* KEY LENG(CHRS)(INDX)*/
```

Figure 31. JLDT Macro Definition (Sheet 2 of 2)



```

/*****
/*
/*      RUNNING JOB CONTROL BLOCK (JCB)
/*
/*****
  3 JCBPTR POINTER,      /* NEXT RUNNING JCB PTR  */
  1 3 JCBFLG,           /* FLAG WORD              */
    5 JCBEND BIT(1),    /* END OF STR/NOT END FLAG*/
    5 JCBMOD BIT(1),    /* PROT/PRIV FLAG        */
    5 JCBROL BIT(1),    /* ROLLABLE/NOT ROLL FLAG */
    5 JCBWAT BIT(1),    /* NOT WAITING/WAIT FLAG  */
    5 JCBRUN BIT(1),    /* READY/RUNNING FLAG    */
    5 JCBACT BIT(1),    /* MARKED FOR DELETION    */
    5 JCBNUL BIT(1),    /* NOT USED                */
    5 JCBRAL BIT(1),    /* READY ALLOC ERROR FLAG */
    5 JCBBIR BIT(8),    /* BIR DEVICE INDEX       */
  2 3 JCBUID CHARACTER(6), /* JOB USERID             */
  3 3 JCBNAM CHARACTER(6), /* JOB NAME                */
  3 3 JCBNOS,           /*                          */
    5 JCBNUM BIT(12),   /* SYS ASSIGNED JOB STR NO*/
    5 JCBSTP BIT(4),    /* JOB STEP NO            */
  3 JCB SNP,           /*                          */
    5 JCBSCD BIT(8),    /* JOB SCHEDULING PRIORITY*/
    5 JCBPRI BIT(8),    /* JOB PRIORITY           */
  3 JCBLOL POINTER,    /* USER AREA 1ST WORD PTR */
  3 JCBRUPL POINTER,   /* JEA 1ST WORD PTR       */
  3 JCBJEL POINTER,    /* JEA LAST WORD + 1 PTR  */
  3 JCBSTC BIT(16),    /* DEFAULT TCB STACK SIZE */
  3 JCBTIM BIT(16), FFF /* TIME LIMIT(SEC)        */
  3 JCBERC BIT(16),    /* TERMINATION ERROR CODE */
  3 JCBLDY POINTER,    /* FIRST LDT POINTER      */
  3 JCBJEQ POINTER,    /* JOB EVENT QUEUE POINTER*/
  3 JCBTCB POINTER,    /* FIRST TCB POINTER      */
  3 JCBTMU FIXED(31),  /* TIME USED (MSEC)       */
  13 JCBERP POINTER,   /* PTR TO ERROR IDENTIFIER*/
  23 JCBERA POINTER)   /* TERMINATION ADDRESS    */

```

Figure 32. JCB Macro Definition





```

/*****
/*
/*      RUN-TIME LOGICAL DEVICE TABLE (LDT)
/*
/*
/*****
3  ILDFLG,          /* FLAG WORD          */
5  ILDOPN BIT(1),  /* CLOSED/OPEN FLAG  */
5  ILDDOF BIT(1),  /* DEVICE/FILE FLAG  */
5  ILDBLK BIT(1),  /* UNBLOCKED/BLOCKED FLAG */
5  ILDICS BIT(1),  /* RETURN ON ALL ERRORS */
5  ILDRET BIT(1),  /* NO RETURN/RETURN FLAG */
5  ILDBSY BIT(1),  /* IDLE/BUSY FLAG     */
5  ILDPAS BIT(1),  /* NO PASS/PASS FLAG   */
5  ILDEL BIT(1),   /* NO DELETE/DELETE FLAG */
5  ILDRRT BIT(1),  /* RETURN ALL PTRYs    */
5  ILDPWO BIT(3),  /* PW-OWNER/PW/NO-PW FLAG */
5  ILDTMP BIT(1),  /* NO TEMP/TEMPORARY FLAG */
5  ILDCRE BIT(1),  /* USER/CREATOR (SYS SET) */
5  ILDNU3 BIT(2),  /* NOT USED           */
3  ILDWD1,         /* WORD 1             */
5  ILDKLN BIT(8),  /* KEY LENGTH         */
5  ILDLUN BIT(8),  /* LOGICAL UNIT NUMBER */
3  ILDPDT POINTER, /* PDT/FCB POINTER    */
3  ILDRLN BIT(16), /* RECORD LENGTH      */
3  ILDPTR POINTER, /* NEXT LDT POINTER   */
3  ILDMLT POINTER, /* MULTI-PURPOSE POINTER */
                    /* RCB POINTER (FILES), */
                    /* BUFR PTR (MAG TAPE)  */
                    /* ATTN FLAG ADDR (DT)  */
ILDFCB LITERALLY 'ILDPDT', /* REPLACE FOR FILE USE*/
ILDRCB LITERALLY 'ILDMLT', /* REPLACE FOR RCB USE*/
ILDBUF LITERALLY 'ILDMLT', /* REPLACE FOR BUF USE*/
ILDATN LITERALLY 'ILDMLT' /* REPLACE FOR ATN USE*/

```

Figure 33. LDT Macro Definition



```

/*****
/*
/*      JCB CONTROL BLOCK (JCM)
/*
/*****
      3 JCMWD0,           /* WORD 0
      5 JCMASG FIXED(8), /* ACT NO OF JOB STRINGS
      5 JCMMSG FIXED(8), /* MAX NO OF JOB STRINGS
      3 JCMWD1,           /* WORD 1
      5 JCMAST FIXED(8), /* ACTUAL NO OF JOB STEPS
      5 JCMHST FIXED(8), /* MAXIMUM NO OF JOB STEPS
      3 JCMADA FIXED(16), /* ACT DSCA SPACE FOR JCBS
      3 JCMMDA FIXED(16), /* MAX DSCA SPACE FOR JCBS
      3 JCMMLU FIXED(16), /* MAX NO OF JOB LUN'S
      3 JCMDIR(96),       /* RDY JCB FILE DIRECTORY
      5 JCMNUM BIT(12),   /* JOB NUMBER
      5 JCMSTP BIT(4)     /* JOB STEP NUMBER
/*****
/*
/*      COMMITTED MEMORY CONTROL BLOCK (CCB)
/*
/*****
      3 CCBSIZ FIXED(16), /* MEMORY BLOCK SIZE
      3 CCBPTR POINTER,   /* POINTER TO NEXT CCB
      3 CCBPRI FIXED(16), /* JOB STEP PRIORITY
      3 CCBJCB FIXED(16)  /* JOB NUMBER
/*****
/*
/*      PASS CONTROL BLOCK (JPCB)
/*
/*****
      3 JPCCNT FIXED(16), /* MAX NO PASSED RESOURCES
      3 JPCADR(15) POINTER /* PDT/FCB ADDRESS
/*****
/*
/*      ROLLOUT FILE DIRECTORY (RTRDIR)
/*
/*****
      3 RTRENT(11),       /* 12 POSSIBLE ROLLEES
      5 RTJNSN,           /* JOB NO/STEP NO
      7 RTJOBN BIT(12),   /* JOB NO
      7 RTSTPN BIT(4),    /* STEP NO
      5 RTSTRT FIXED,     /* 1ST RR REC FOR ROLLEE
      5 RTLENG FIXED)     /* NO OF RR RECORDS

```

Figure 34. JCM, CCB, JPCB and RTRDIR Macro Definitions



```

/*****
/*
/*      TASK CONTROL BLOCK (TCB)
/*
/*****
    3 TCBROD POINTER,      /* READY LINK/DORMANT CT */
    2 3 TCBFLG,           /* TCB FLAG WORD */
    5 TCBRDY BIT(1),     /* READY/DORMANT FLAG */
    5 TCBACT BIT(2),     /* ACTION CODE */
                          /* 00=NORMAL */
                          /* 01=MARKED FOR DELETE */
                          /* 10,11 NOT DEFINED */
    5 TCBXXX BIT(2),     /* DELETION CONTROL FLAGS */
                          /* VALID ONLY WHEN DORMANT */
                          /* 00=DELETE OK */
                          /* 01=FORCE ACTIVE */
                          /* 10=LEAVE ALONE */
                          /* 11= NOT DEFINED */
    5 TCBQUI BIT(1),     /* QUIETING FOR ROLLOUT */
    5 TCBNUL BIT(2),     /* NOT USED */
    5 TCBPRI BIT(8),     /* PRIORITY */
    3 TCBPTR POINTER,    /* NEXT TCB POINTER */
    3 TCBJCB POINTER,    /* PARENT JCB POINTER */
    3 TCBID BIT(16),     /* TASK ID */
    3 TCBSTB,           /* STATUS BLOCK */
    5 TCBPCT POINTER,    /* PROGRAM COUNTER (PC) */
    5 TCBSTR BIT(16),    /* STATUS REGISTER (SR) */
    3 TCBTRF,           /* REGISTER FILE */
    5 TCBARG FIXED(16), /* ARITHMETIC REGISTER (A) */
    5 TCBERG FIXED(16), /* EXTENSION REGISTER (E) */
    5 TCBXRG FIXED(16), /* INDEX REGISTER (X) */
    5 TCBMRG FIXED(16), /* MAINTENANCE REGISTER (M) */
    5 TCBSRG FIXED(16), /* STORAGE REGISTER (S) */
    5 TCBLRG FIXED(16), /* LINK REGISTER (L) */
    5 TCBBRG FIXED(16), /* BASE REGISTER (B) */
    3 TCBSLA POINTER,    /* LAST STACK WORD PTR */
    3 TCBSLU POINTER)   /* LAST USED STACK WD PTR */

```

Figure 35. TCB Macro Definition



```

/*****
/*
/*      EVENT ATTRIBUTE TABLE (TEVENT)
/*
/*
/*****
    2 TEVTRL(40),          /*
    3 TEVPUR BIT(1),      /* PURE/IMPURE FLAG
    3 TEVJOB BIT(1),      /* JOB/SYS ORIENTED FLAG
    3 TEVSAV BIT(1),      /* SAVE FLAG(JOB ORIENTED)
    3 TEVUWT BIT(1),      /* USER WAITABLE FLAG
    3 TEVUPT BIT(1),      /* USER POSTABLE FLAG
    3 TEVSOM BIT(1),      /* SINGLE/MULT MATCH FLAG
    3 TEVMOS BIT(1),      /* MULT/SINGLE FAILURE
    3 TEVROP BIT(2),      /* RELATIONAL OPERATOR
    3 TEVLNK BIT(3),      /* EVENT LINKAGE OPTION
    3 TEVLNG BIT(4)       /* EVENT ID LENGTH (WORDS)
/*****
/*****
/*
/*      EVENT DESCRIPTOR BLOCK (TEDB)
/*
/*
/*****
    3 TEDUM,              /* WORD 0
    5 TEDNUL BIT(1),      /* NOT USED
    5 TEDNDX FIXED(15),   /* EVENT INDEX
    3 TEDIDW(30) FIXED(16) /* EVENT ID WORD
/*****

```

Figure 36. TEVENT and TEDB Macro Definitions



943015-9701

---

ALPHABETICAL INDEX



## ALPHABETICAL INDEX

<u>Subject</u>	<u>Reference</u>	<u>Subject</u>	<u>Reference</u>
Batch Input Reader family tree	F17	File management family tree	F8
Batch Input Spooler family tree	F18	FIOPRB Macro Definition	F23
Batch Output Spooler family tree	F19	FMSTAT Macro Definition	F28
CATFIL family tree	F21	FRCBI Macro Definition	F27
CATLOG family tree	F20	FRCBL Macro Definition	F26
CCB Macro Definition	F34	FRCBR Macro Definition	F27
Compiler, PL/EXUS, execution of	4.1	I/O management family tree	F11
Compiler support, PL/EXUS	1.0	IPRB Macro Definition	F22
Control Block Linkage	6.3, F1	ITS family tree	F16
DDPRB Macro Definition	F22	JCB Macro Definition	F32
Disc Map	6.9, F6	JCL interpreter family tree	F14
DX980 Operating System, Linkable Parts Tape	2.0, 3.3	JCM Macro Definition	F34
DX980 Operating System Source Tape	2.0, 3.4	JEQ	6.8
DX980 Source Kit	1.0, 2.0, 3.0	JLDT	F4
DX980 Source Main- tenance Routine	3.4, 4.1	JLDT Macro Definition	F31
Dynamic system overlay area	6.11	Job control commands	3.0, 3.2, 3.3, 3.4, 4.1
Event Attribute Table	6.2, T2	Job management family tree	F7
Family trees, DX980	6.12	JPCB Macro Definition	F34
FCB Macro Definition	F25	JSB	F4
FCBIXT Macro Definition	F26	JSB Macro Definition	F30
FCBLXT Macro Definition	F26	LDT Macro Definition	F33



## ALPHABETICAL INDEX (Continued)

<u>Subject</u>	<u>Reference</u>	<u>Subject</u>	<u>Reference</u>
Linkable parts file	4.1	Source modules	1.0, 3.4
Linkable Parts Tape, DX980 Operating System	2.0, 3.3	Source Tape, DX980 Operating System	2.0, 3.4
Linking DX980	5.0	Source tape, index of	3.4
Logical Unit Numbers, Reserved	6.10, T3	Supervisor calls	6.11, T4, T5
Memory image phase	6.11, T4	SVCDRT Macro Definition	F29
Memory management family tree	F15	SVC's	6.11, T4, T5
Macro Structure Definitions	6.13	System Wide Events	6.1, T1
Operating System Kit	1.0	Task Control Block	see TCB
Operator Communi- cations family tree	F10	Task management family trees	F9
PDT Macro Definition	F24	TCB	F5
PDTEXT Macro Definition	F24	TCB Macro Definition	F35
PL/EXUS compiler, execution	4.1	TEDB Macro Definition	F36
PL/EXUS compiler support	1.0	TEVENT Macro Definition	F36
PL/EXUS Language Specification	2.0, 3.1	TEVTBL	6.2, T2
PL/EXUS Support Tape	2.0, 3.2	TWATPT	6.5, F3
Procedure Pool	6.11	Virtual Buffer Sharing	6.4, F2
Rollin family tree	F13	Wait Control Block Table	6.5, F3
Rollout family tree	F12		
RTRDIR Macro Definition	F34		
SAP Assembler, Execution	4.2		
Source Kit, DX980	1.0, 2.0, 3.0		
Source Maintenance Routine, DX980	3.4, 4.1		





---

First Class  
PERMIT NO. 3135  
Austin, Texas

**BUSINESS REPLY MAIL**  
No Postage Necessary if Mailed in the United States

Postage Will Be Paid by

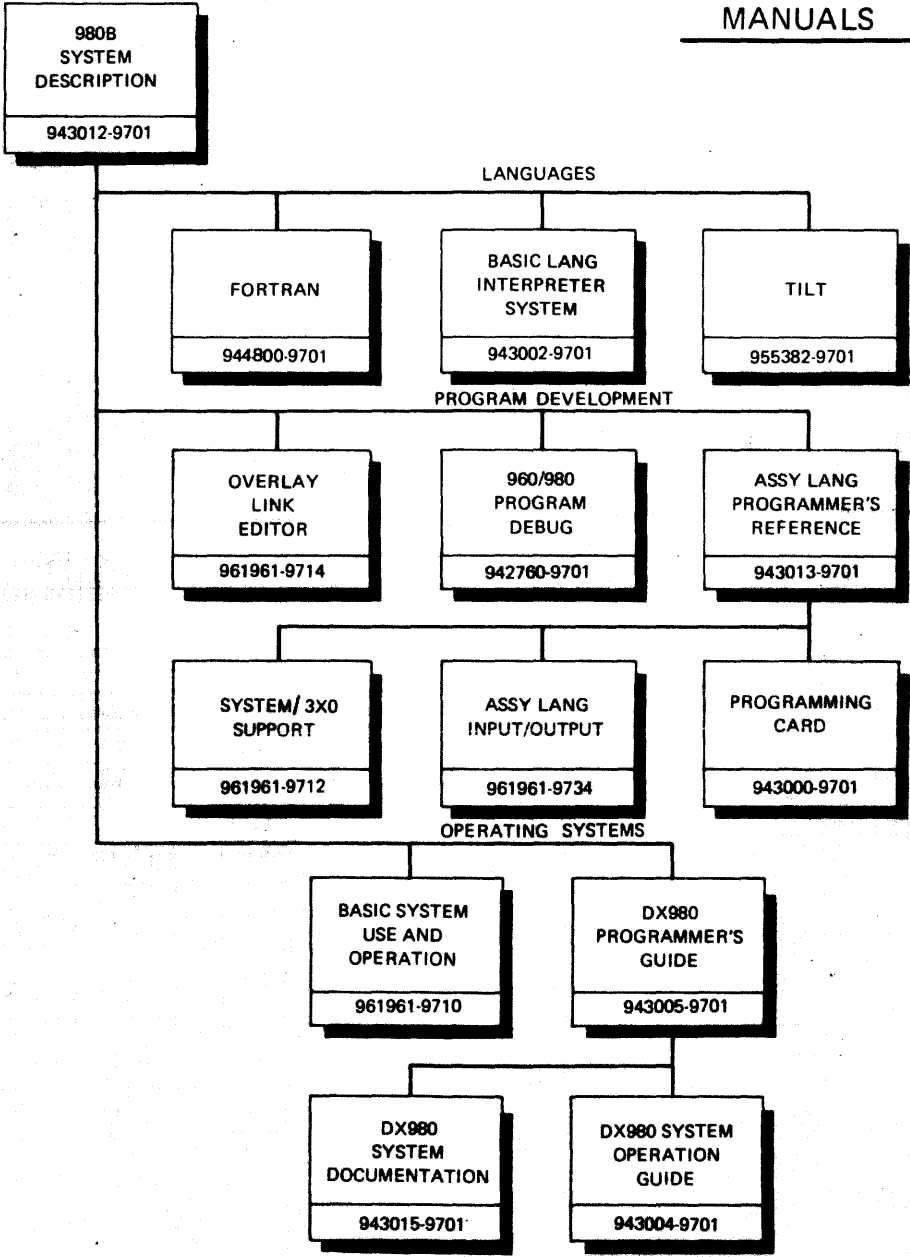
**TEXAS INSTRUMENTS INCORPORATED**  
DIGITAL SYSTEMS DIVISION

**P.O. BOX 2909 · AUSTIN, TEXAS 78767**

**Attn: TECHNICAL PUBLICATIONS, MS 2146**

---

**980 COMPUTER  
SYSTEM  
SOFTWARE  
MANUALS**



**TEXAS INSTRUMENTS**

INCORPORATED

DIGITAL SYSTEMS DIVISION

POST OFFICE BOX 2909 • AUSTIN, TEXAS 78767