

Chapter
2

**PROGRAM
COMMUNICATION
WITH DL/1**

*Get on the
Fast Track!*



TM

**SYS-ED/
COMPUTER
EDUCATION
TECHNIQUES, INC.**

Objectives

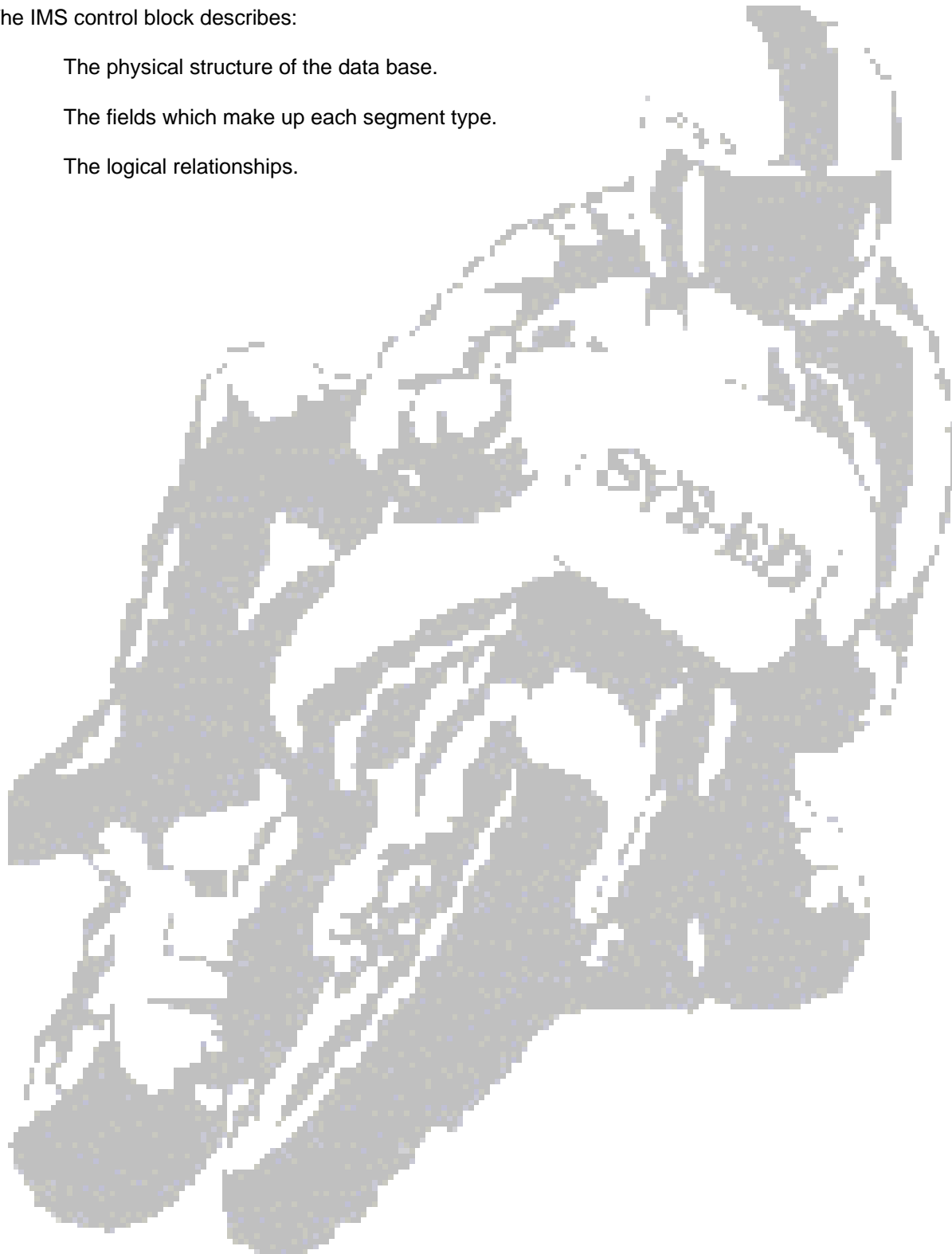
You will learn:

- C DBD: Data Base Description.
- C DBD control statements.
- C PSB: Program Specification Block.
- C Logic flow using I/O PCB - COBOL, PL/1, and Assembler.
- C ALTPCB.
- C Alternate destination message flow - fixed and modifiable.
- C Telecommunication PCBs.
- C PCB masks.

1 DBD: Data Base Description

The IMS control block describes:

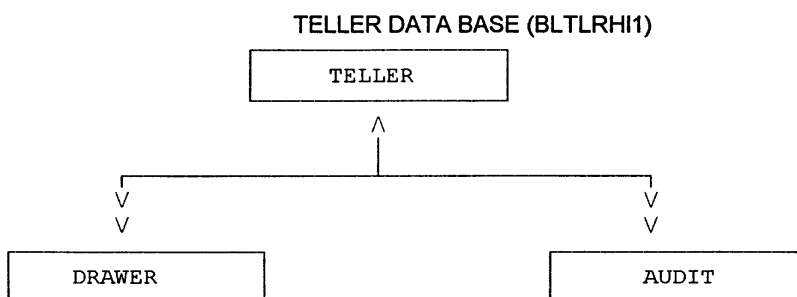
- C The physical structure of the data base.
- C The fields which make up each segment type.
- C The logical relationships.



1.1 DBD Control Statements

DBD	Names the data base and provides dl/l information as to organization.
DATASET	Defines a data set group within the data base.
SEGM	Defines the segment type, the position in the hierarchy, the physical characteristics and the segments relation to other segments.
FIELD	Defines the length, location and type of field within a segment type. Fields are used by PSBs' when defining sensitivity and by application programs in a SSA.
LCHILD	Used to define a logical relationship between tow segment types.
XDFLD	Used for defining secondary index relationships.
DBBGEN	Indicates the end of a DBD generation control statements.
FINISH	Sets a nonzero condition code for linkedit if any records were generated.
END	Indicates the end of all input statements to the os/vs Assembler.

1.2 DBD Sample

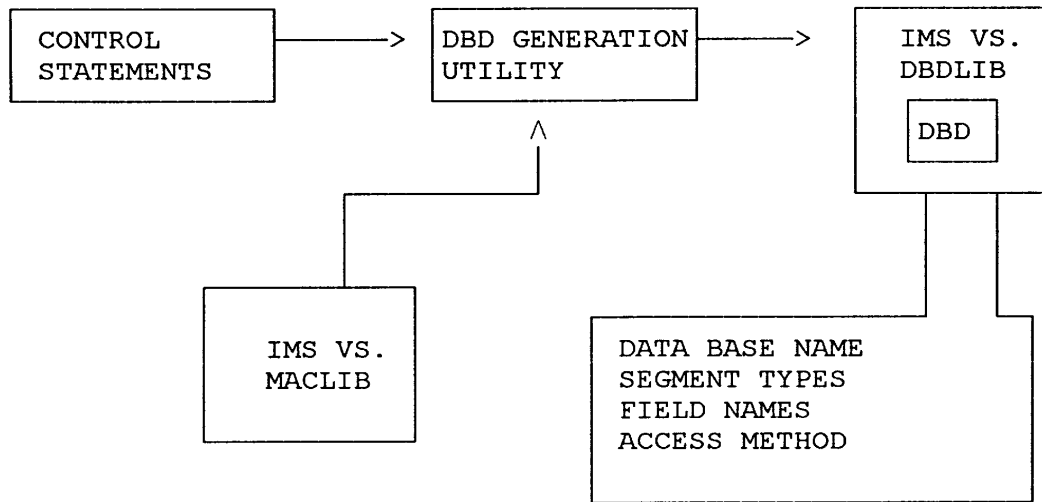


DBD	NAME = BKTLRH1 ACCESS=HIDAM
DATASET	DD1=BLTLHIES, DEVICE=3350
SEGM	NAME=TELLER, PARENT=0, BYTES=25
LCHILD	POINTS TO INDEX
FIELD	NAME=(TELLRNO, SEQ,U), BYTES=4, START=1, TYPE=C
FIELD	NAME=(TELLADDRD), BYTES=25, START=26, TYPE=C
FIELD	NAME=(TELLNME), BYTES=21, START=5, TYPE=C
SEGM	NAME=DRAWER, PARENT=TELLER, BYTES=20
FIELD	NAME=(DRAWKY, SEQ, U), BYTES=9, START=1
FIELD	NAME=(CASH), BYTES=11, START=10
SEGM	NAME=AUDIT, PARENT=TELLER, BYTES=10
FIELD	NAME=(AUDITNO, SEQ,U), BYTES=3, START=1
FIELD	NAME=(AUDITDTE), BYTES=7, START=4
DBDGEN	
FINISH	
end	

2 PSB: Program Specification Block

The PSB are the IMS control blocks which describe the:

- C application program's view of the data base (s).
- C type of access allowed.
- C type of processing allowed.



PSB generation processes input control cards and creates a member on PSBLIB.

2.1 PSB Control Statements

PCB

Describes the type of PCB, which may be either data base or teleprocessings.

Type=DB (MPP, BMP, BATCH)

Defines the name of the DBD, the type of processing allowed and the length of the longest key

Type=TP (MPP, BMP)

Defines the output message destination.

SENSEG

Defines which segments the application program is sensitive to and what type of process is allowed. It must be spelled exactly the same as it's related 'SEGM' in the DBD.

SENFLD

Defines which fields the application is sensitive to. It must be spelled exactly the same as it's related 'FIELD' in the DBD.

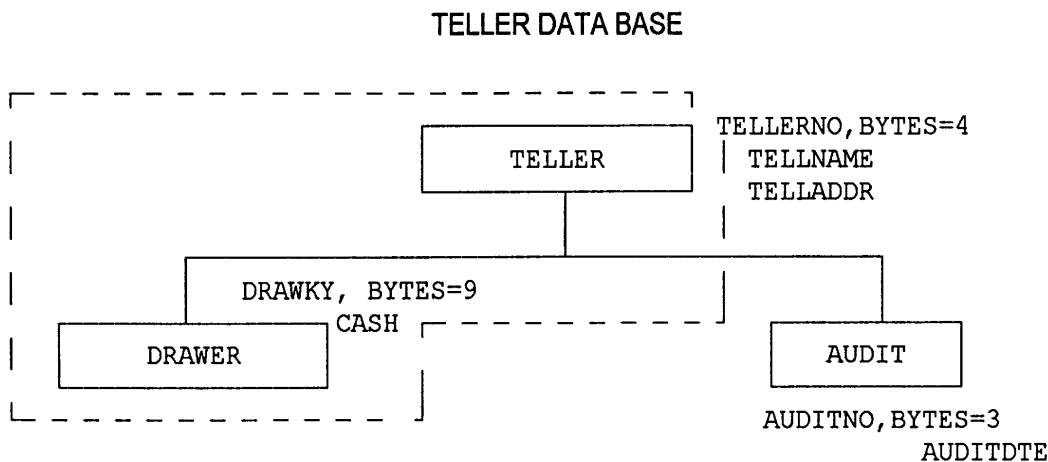
PSBGEN

Specifies characteristics of the application program and the PSB name.

END

Indicates the end of the assembly data to OS/VS assembler.

2.2 PSB Sample



Entire Database PSB

```

PCB          TYPE=DB, KEYLEN=13, DBDNAME=BKTLRHI1, PROCOPT=A
SENSEG      NAME=TELLER, PARENT=0
SENSEG      NAME=DRAWER, PARENT=TELLER, PROCOPT=G
SENSEG      NAME=AUDIT, PARENT=TELLER
PSBGEN      LANG=COBOL, PSBNAME=BKTL0001
END
  
```

PSB for Only A

```

PCB          TYPE=DB, KEYLEN=13, DBDNAME=BKTLRHI1, PROCOPT=A
SENSEG      NAME=TELLER, PARENT=0
SENFLD      NAME=TELLERNO, START=1, REPL=NO
SENFLD      NAME=TELLNME, START=5, REPL=YES
SENSEG      NAME=DRAWER, PARENT=TELLER
PSBGEN      LANG=COBOL, PSBNAME=BKTL0002
END
  
```

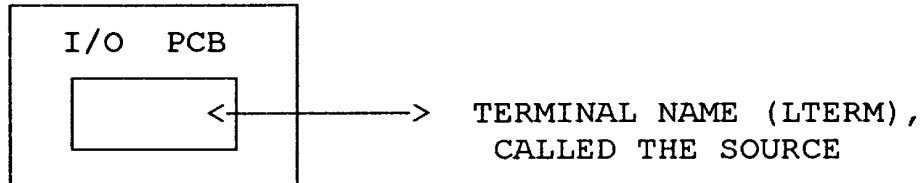
2.3 On-line PSB

- C Describes a program's use of logical terminals.
- C The PSB must have same name as application program in on-line system (not required for BMP programs).

3 Telecommunication PCBs

I/O PCB is the vehicle used for communication between the DC system and application program.

It contains the terminal that originated the message:



It is always positionally the first PCB in the entry statement.

(COBOL) ENTRY 'DLITCBL' USING IO-PCB...

(PL/I) DLITPLI: PROC (IOPCB_PTR [, ...]

3.1 COBOL Logic Flow Using I/O PCB

```
LINKAGE SECTION                                WORKING-STORAGE
01  IO-PCB                                     01 INPUT MESSAGE  PIC X ( )
    |
    |
    |
01  DB-PCB                                     01 OUTPUT MESSAGE PIC X ( )
    |
    |
    |
PROCEDURE
ENTRY  `DLITCBL'  USING IO-PCB, DB-PCB.
    |
    |
    |
CALL  `CBLTDLI'  USING GU, IO-PCB, INPUT MESSAGE
CHECK STATUS CODE IN IO-PCB
"QC"      NO MORE MESSAGES.
CALL  `CBLTDLI'  ACCESS DATA BASES
    |
    |
    |
CALL  `CBLTDLI'  USING ISRT, IO-PCB,
OUTPUT MESSAGE
CHECK STATUS CODE
GOBACK.
```

3.2 PL/1 Logic Flow Using I/O PCB

```
DLITPLI: PROC      (IOPCB_ PTR, DBPCB_ PTR)
    DCL 1          IOPCB BASED (IOPCB_ PTR)
        |
        |
        |
        1          DBPCB BASED (DBPCB_ PTR)
        |
        |
        |
    DCL 1          INPUT_ MESSAGE
        |
        |
        |
        1          OUTPUT_ MESSAGE
        |
        |
        |
    READ_ MESSAGE:
        CALL PLITDLI (3, GU, IOPCB_ PTR, INPUT_ MESSAGE)
        CHECK STATUS CODE IN IOPCB
        'QC' NO MORE MESSAGES
        |
        |
        |
        CALL PLITDLI          ACCESS DATA BASES
        |
        |
        |
        CALL PLITDLI (3, ISRT, IOPCB_ OUTPUT_ MESSAGE)
        CHECK STATUS
        END DLITPLI
```

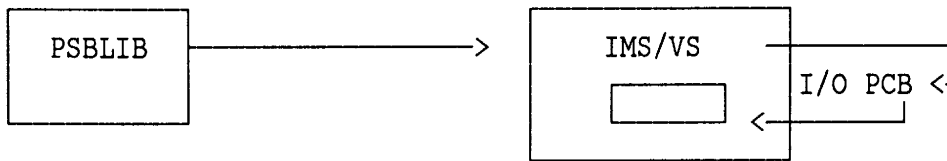
3.3 Assembler Logic Flow Using I/O PCB

```
PROGRAM CSECT
:
:
:
CALL      ASMTDLI, THREE, GU, IOPCB, INMSGAR)
OR
CALL      ASMTDLI, (GU, IOPCB, INMSGAR), VL
CHECK STATUS CODE IN IOPCB
    `QC'      NO MORE MESSAGES
:
:
CALL ASMTDLI, ACCESS DATA BASES
:
:
CALL ASMTDLI, (ISRT, IOPOCB, OTMSGAR), VL
(CHECK STATUS)
RETURN (14,12), RC=0

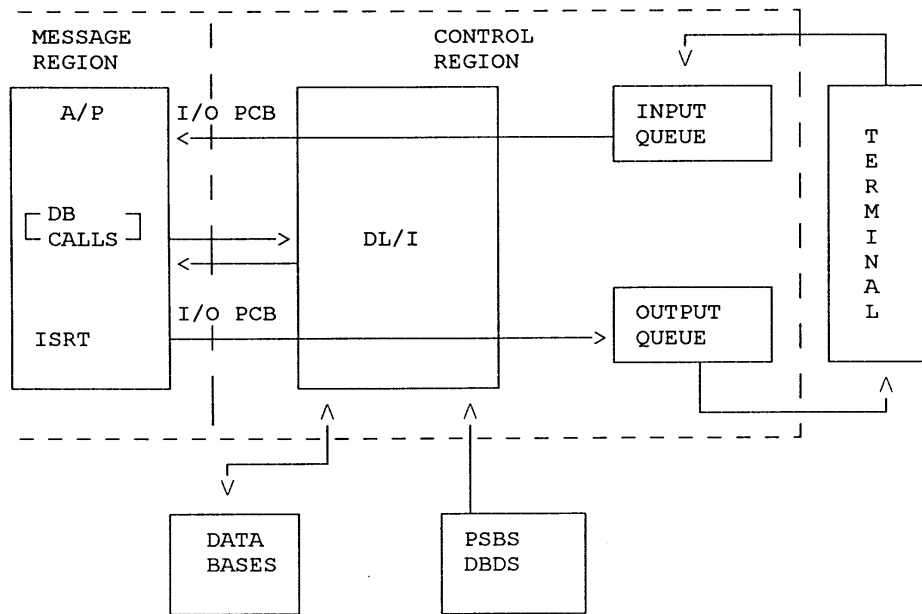
INMSGAR    DS    CLXX
OTMSGAR    DS    CLXX
```

```
(ASSEMBLER) PROGRAM    CSECT
                        ENTRY    DLITASM
                        USING A, BASE
                        DLITASM  SAVE    (14,12)
                        LR       BASE, 15
                        LR       R2,R1
                        LA       R2, 0(R2)
                        L        R4, 0(R2)
                        * NOW R4 HAS ADDRESS OF I/O PCB
```

C Created at execution time by IMS.



4 Simple Message Processing Flow



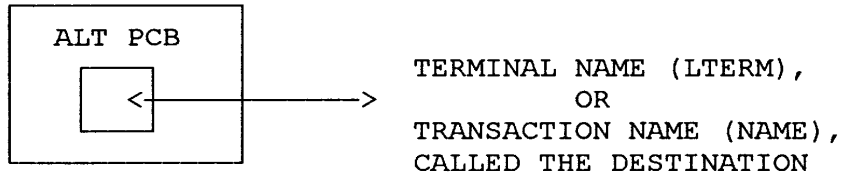
```

PCB  TYPE=DB...
.
.
.
PSBGEN ...
    
```

5 Telecommunications PCBs

ALTPCB (FIXED) provides an application program facility for routing a message to a fixed alternate destination.

- C Contains the terminal or transaction code for the output message destination.



If a transaction name is specified, the destination will be a program.

- C Indicated at PSBGEN.

PCB TYPE=TP, LTERM=LERMB, MODIFY= NO

OR

PCB TYPE =TP, NAME=TRANSATA, MODIFY= NO

- C Always positionally the second PCB in the entry statement.

(COBOL) ENTRY 'DLITCBL' USING IO-PCB,ALTPCB....

(PL/I) DLITPLI: PROC (IOPCB_PTR,ALTPCB_PTR...)

6 ALPCB

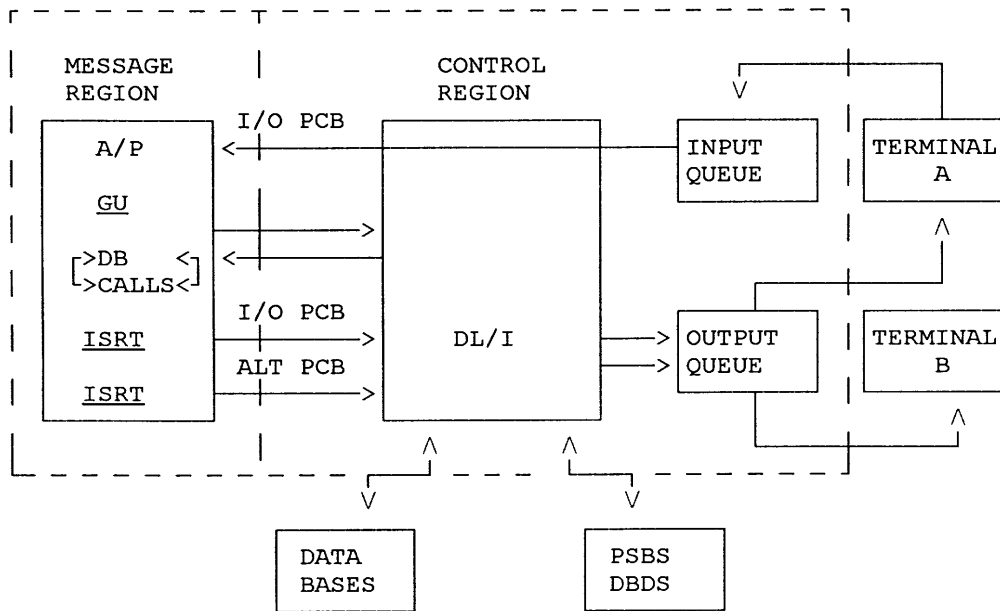
(ASSEMBLER) PROGRAM CSECT
ENTRY DLITASM
ALPCB WILL BE THE SECOND PCB IN R1

C Created during PSBGEN.

C Used in an ISRT call to IMS.

(COBOL) call `CBLTDLI' using ISRT, ALPCB, OUTPUT_MESSAGE
(PL/I) CALL PLITDLI (3, ISRT, ALPCB_PTR, OUTPUT_MESSAGE)
(ASSEMBLER) CALL ASMTDLI, (THREE, ISRT, ALPCB, OUTPUTMSG)

7 Alternate Destination Message Flow - Fixed



ALTPCB

PCB TYPE=TP, MODIFY=NO, LTERM=LTERMB
 ,
 ,
 PCB TYPE=DB
 PSBGEN ...

8 Telecommunication PCBs

ALTPCB (MODIFIABLE)

Similar to the fixed version, but with the added feature of specifying the destination.

C Indicated at PSBGEN

```
PCB TYPE=TP, LTERM=LTERMB, MODIFY = *YES*
+))),
.)))-
```

OR

```
PCB TYPE=TP, NAME=TRANSTA, MODIFY = *YES*
+))),
.)))-
```

IF THE DESTINATION IS NOT SPECIFIED WILL DEFAULT TO THE LTERM/NAME ON PCB.

C It has the same positional requirements as in the fixed ALTPCB.

C It is also created during PSBGEN.

C Indicated to IMS by first issuing a CHNG CALL and then an ISRT CALL.

```
(COBOL) CALL 'CBLTDLI' USING CHNG, ALTPCB, LTERMD
```

OR

```
CALL 'CBLTDLI' USING CHNG, ALTPCB, TRANSTB
```

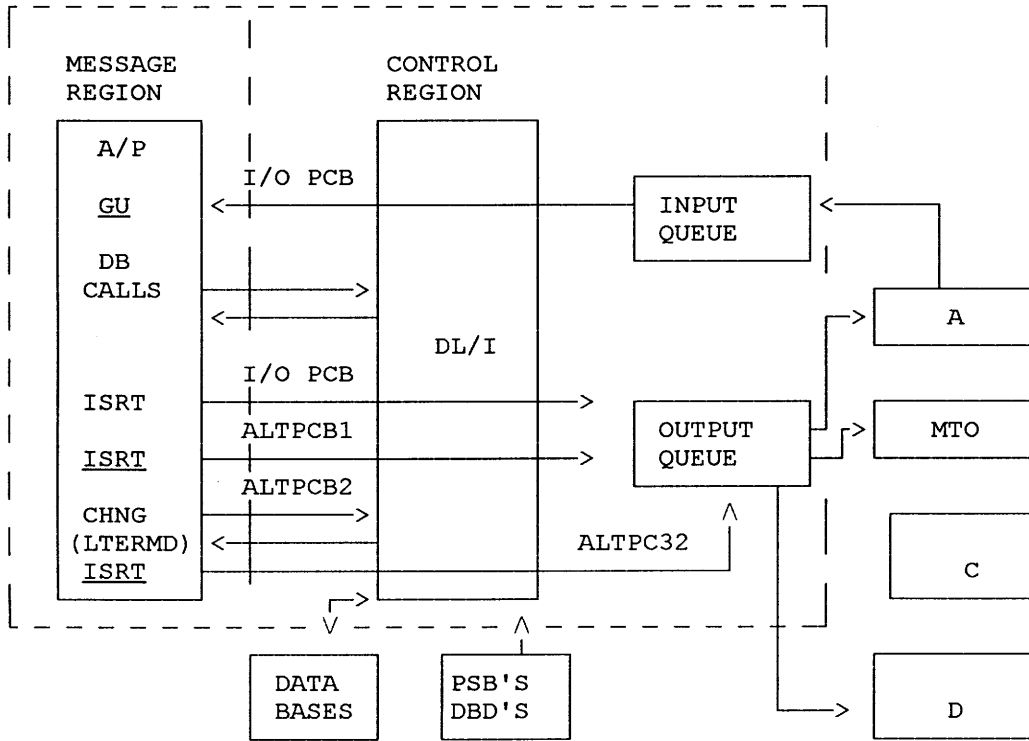
```
CALL 'CBLTDLI' USING ISRT, ALTPCB, OUTPUT-MESSAGE
```

9 ALTPCB

(PL/1) CALL PLITCLI (3, CHNG, ALTPCB_PTR, LTERMD)
 OR
 CALL PLITDLI (3, CHNG, ALTPCB_PTR, TRANSTB)
 CALL PLITDLI (3, ISRT, ALTPCB_PTR, OUTPUT-MESSAGE)

(ASSEMBLER) CALL ASMTDLI, (CHNG, ALTPCB, LTERMD), VL
 OR
 CALL ASMTDLI, (CHNG, ALTPCB, TRANSTB), VL
 CALL ASMTDLI, (ISRT, ALTPCB, OTMSGAR), VL

10 Alternate Destination Message Flow - Modifiable



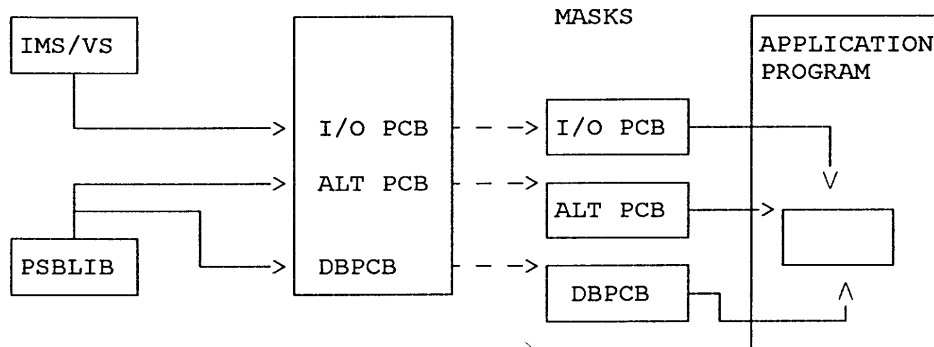
```

ALT PCB1      PCB TYPE=TP, MODIFY=NO, LTERM=MTOTERM
ALT PCB2      PCB TYPE=TP, MODIFY=YES, LTERM=LTERMC
               .
               .
               .
               PCB TYPE=DB
               PSBGEN ..., CMPAT=YES
    
```

11 PCB Masks

When an application program is scheduled, IMS passes to the program an address for each PCB within a program PSB.

This address is the beginning point of a PCB mask and represents the communication link between DL/1 and the application program.



Declared at program entry:

(COBOL) ENTRY 'DLITCBL' USING IOPCB, ALTPCB, DBPCB

(PL/I) DCL 'PLITDLI' ENTRY
 DCL (IOPCB_PTR, ALTPCB_PTR, DBPCB_PTR)

12 Layout of Telecommunications

↓ ALTPCB ↑	SOURCE/DESTINATION	8	↓ I/O PCB ↑
	RESERVED IMS	2	
	STATUS CODE	2	
CURRENT DATE	4		
CURRENT TIME	4		
INPUT MESSAGE SEQUENCE	4		
MOD	8		
USER IDENTIFICATION	8		

- C I/O PCB, the area for all inputs (SOURCE).
- C Destination, for ALTPCBS, maybe fixed or modifiable.
- C Status code, the most important field to an application programmer.

12.1 PCB Mask - Example

COBOL

```

01      IO_PCB.
02      LTERM-NAME      PICTURE X(8)
02      IMS-RESERVE     PICTURE XX.
02      STATUS-CODE    PICTURE XX.
02      JULIAN-DATE     PICTURE S9(7)      COMP-3.
02      TIME-OF-DAY    PICTURE S9(7)      COMP-3.
02      MSG-SEQ        PICTURE S9(7)      COMP.
02      MOD-NAME       PICTURE X(8).
02      USER-ID        PICTURE X(8).
    
```

PL/1

```

DECLARE 1 IO_PCB (IOPCB_PTR)
2      IO_TERM      CHARACTER (8).
2      IO_RESERVE   CHARACTER (2).
2      IO_STATUS    CHARACTER (2).
2      IO_JULIAN    FIXED DECIMAL (7).
2      IO_TIME      FIXED DECIMAL (7).
2      IO_MSG_SEQ   FIXED BINARY (31).
2      IO_MODNAME   CHARACTER (8).
2      IO_USERID    CHARACTER (8).
    
```

ASSEMBLER

LTERM	DSECT	LAYOUT OF PCB MASK
RESERVED	DS CL8	
STATUS	DS CL2	
JULIAN	DS CL2	
TIME	DS PL4	
MSG	DS F	
MODNAME	DS CL8	
USERID	DS CL8	