CICS/VS PROGRAMMING II - U3682

DAY 1

BMS REVIEW AND EXTENTIONS

BMS PAGING

TEMPORARY STORAGE

DAY 2

FILE CONTROL

TRACE CONTROL

DUMP CONTROL

DAY 3

BUILT IN FUNCTIONS

TASK CONTROL

INTERVAL CONTROL

DAY 4

TEXT BUILDING

ROUTING

MESSAGE SWITCHING

JOURNALING

PROGRAM CONSIDERATIONS

CICS/VS PROGRAMMING II HANDOUT

U3682

THIS MATERIAL WAS PRODUCED FOR EDUCATIONAL PURPOSES ONLY. THE UTMOST CARE HAS BEEN TAKEN TO ENSURE THE ACCURACY OF THIS PUBLICATION. HOWEVER, NO RESPONSIBILITY IS ASSUMED FOR ANY INACCURACIES THAT MAY OCCUR. FURTHERMORE, IT SHOULD BE UNDERSTOOD THAT CHANGES MAY OCCUR THAT MAY CAUSE ALL OR PART OF THIS PUBLICATION TO BECOME OBSOLETE.

BASIC MAPPING
OVERVIEW

WHAT IS BASIC MAPPING?

WHY USE BASIC MAPPING?

HOW DO YOU USE BASIC MAPPING?

BASIC MAPPING

WHAT IS BASIC MAPPING?

- FACILITY OF CICS

- COLLECTION OF MACROS
  - WITHIN APPLICATION PROGRAM
  - EXTERNAL TO APPLICATION PROGRAM

- DESCRIBES TERMINAL DATA STREAM
  - OUTPUT
  - INPUT

- ALLOWS UTILIZATION OF TERMINAL FEATURES

MAPPING

ADVANTAGES

DEVICE INDEPENDENCE
FORMAT INDEPENDENCE

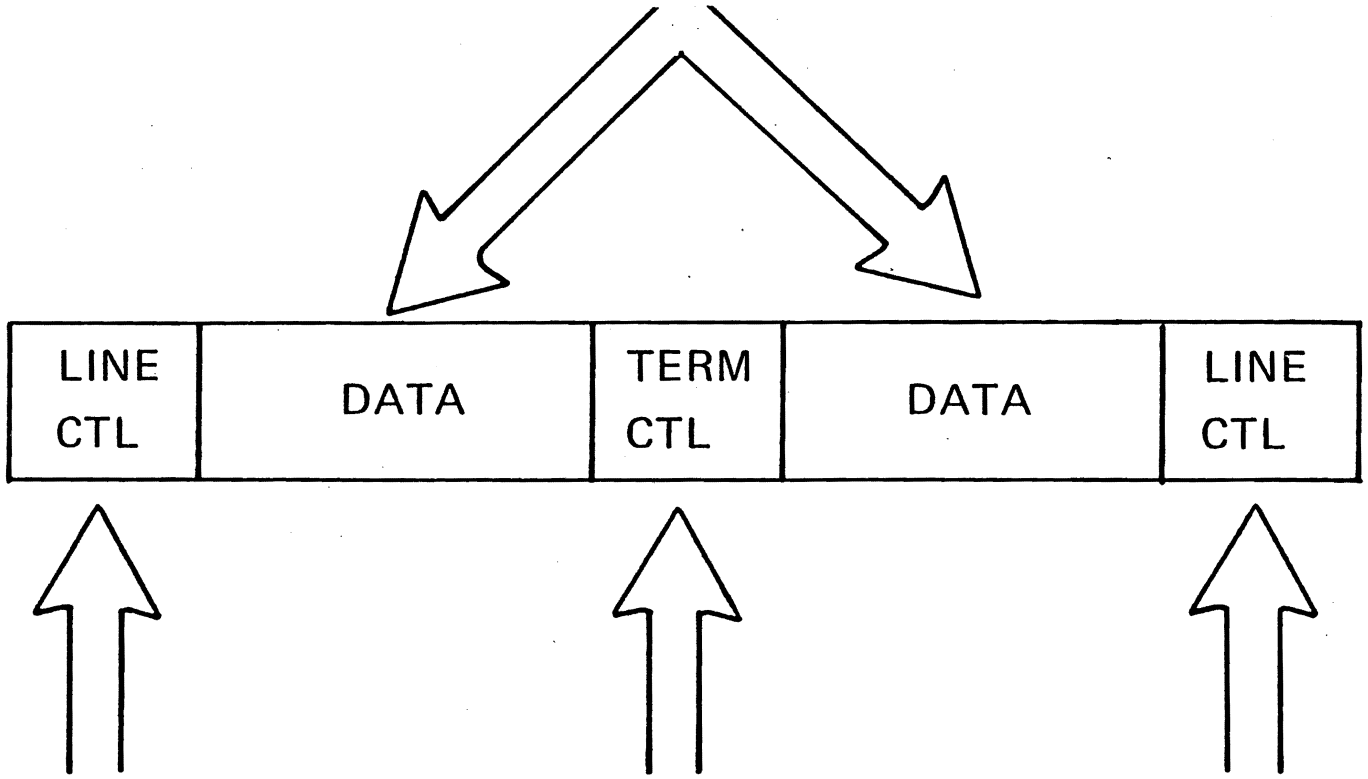
DEVICE INDEPENDENCE

DEVICE INDEPENDENCE PERMITS THE APPLICATION PROGRAM TO BE WRITTEN WITHOUT REGARD TO THE PHYSICAL CHARACTERISTICS OF EACH TERMINAL TYPE.

MAPPING

DEVICE INDEPENDENCE

SUPPLIED BY
APPLICATION
PROGRAM



LINE CTL | DATA | TERM CTL | DATA | LINE CTL

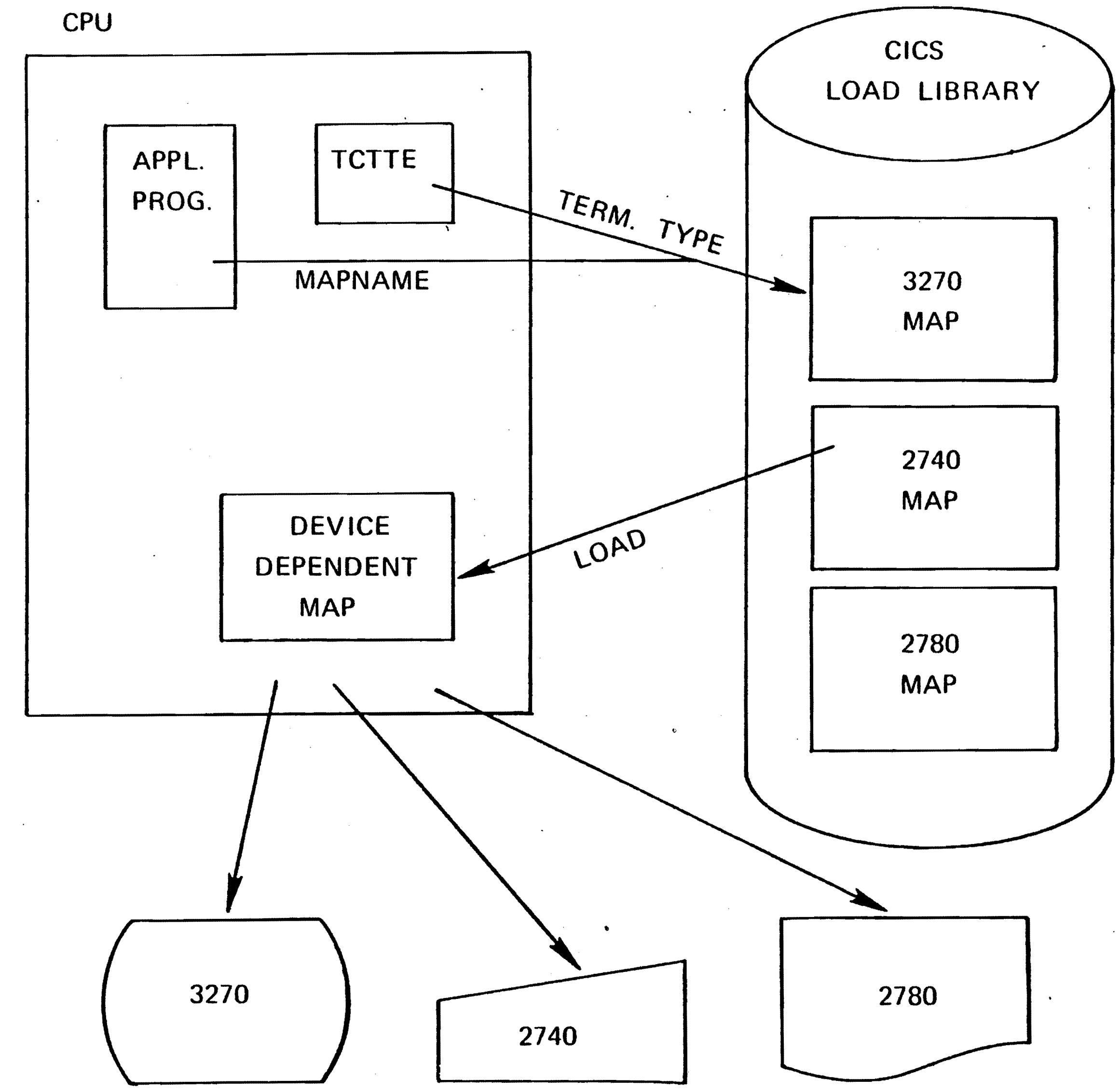
SUPPLIED
BY TCP

SUPPLIED
BY BMS

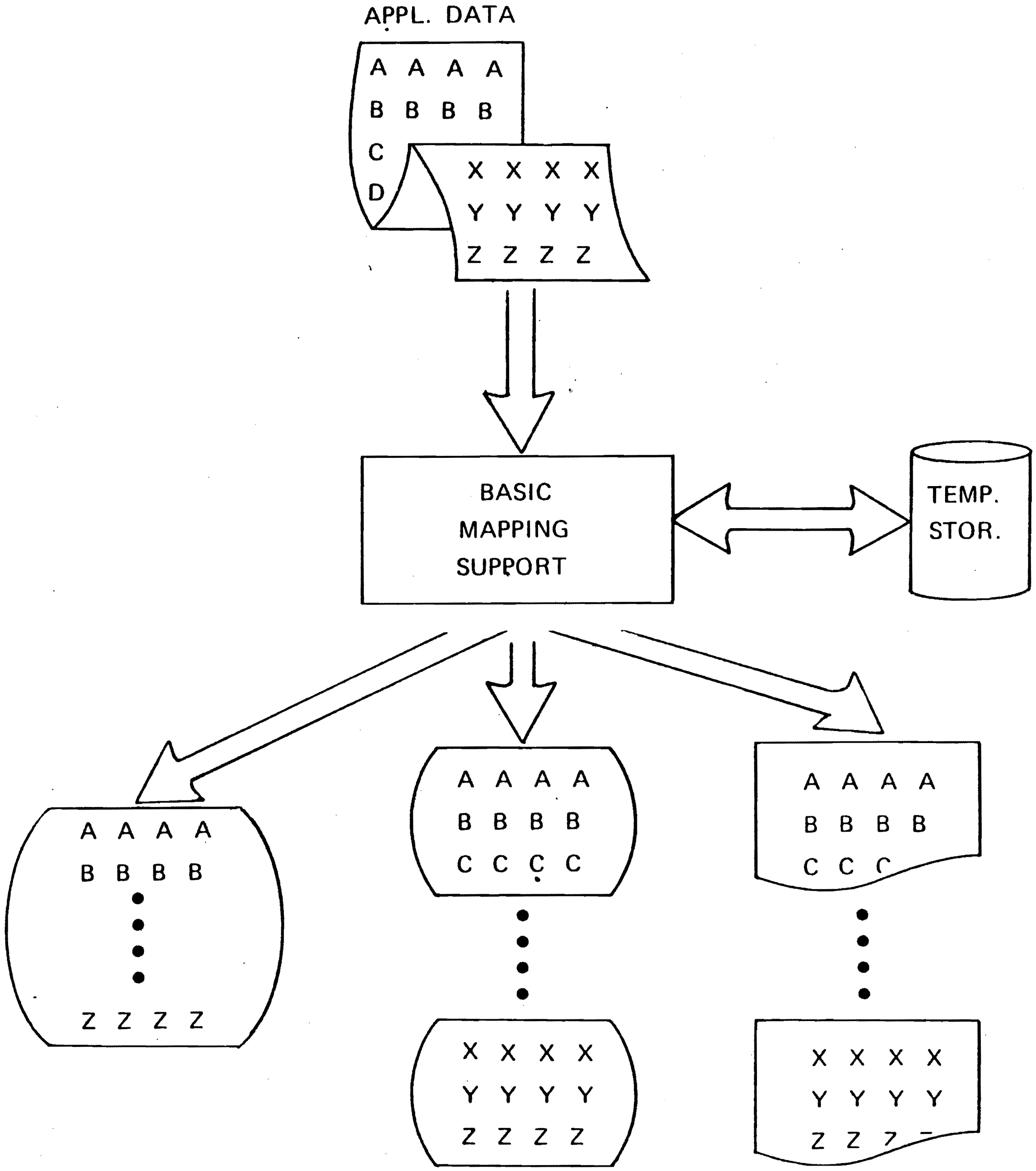
SUPPLIED
BY TCP

MAPPING

DEVICE DEPENDENT MAPS



DEVICE INDEPENDENCE

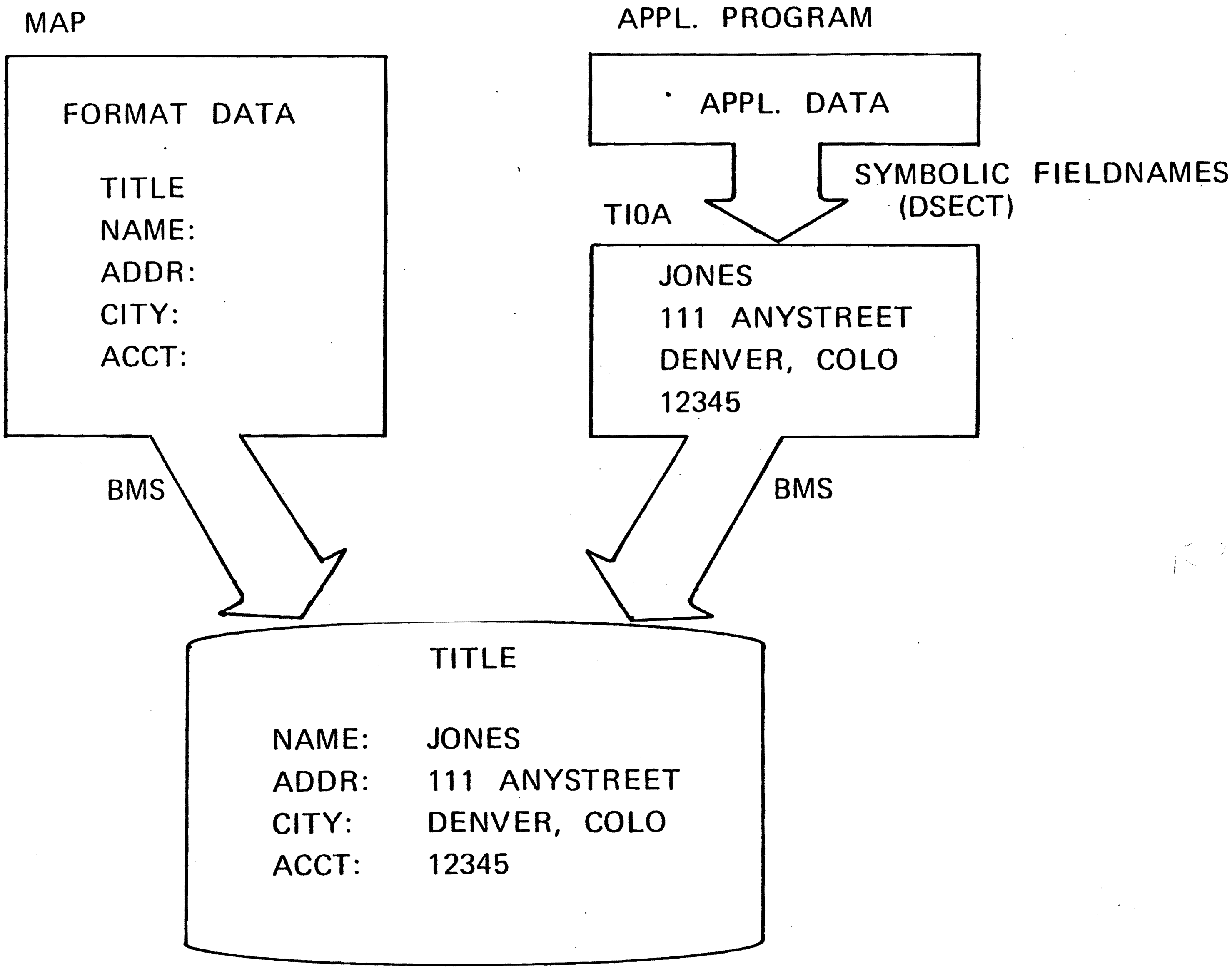


FORMAT INDEPENDENCE

FORMAT INDEPENDENCE PERMITS THE APPLICATION PROGRAM TO BE WRITTEN WITHOUT REGARD TO THE FORMAT OR PHYSICAL PLACEMENT OF FIELDS ON THE TERMINAL.

MAPPING

FORMAT INDEPENDENCE



MAP

FORMAT DATA

TITLE
NAME:
ADDR:
CITY:
ACCT:

APPL. PROGRAM

APPL. DATA

SYMBOLIC FIELDNAMES
(DSECT)

TIOA

JONES
111 ANYSTREET
DENVER, COLO
12345

BMS

BMS

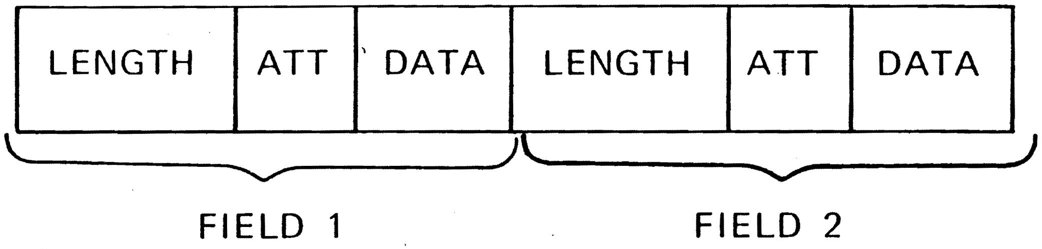
TITLE

NAME: JONES
ADDR: 111 ANYSTREET
CITY: DENVER, COLO
ACCT: 12345

MAPPING

FORMAT INDEPENDENCE

FIELD FORMAT

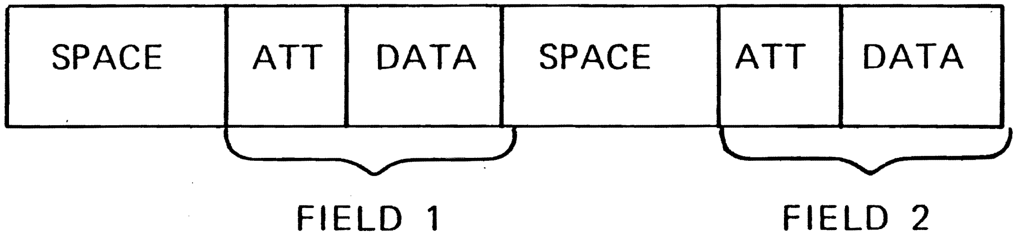


LENGTH	ATT	DATA	LENGTH	ATT	DATA

FIELD 1

FIELD 2

BLOCK FORMAT

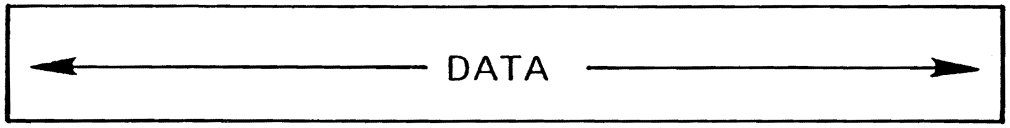


SPACE	ATT	DATA	SPACE	ATT	DATA

FIELD 1

FIELD 2

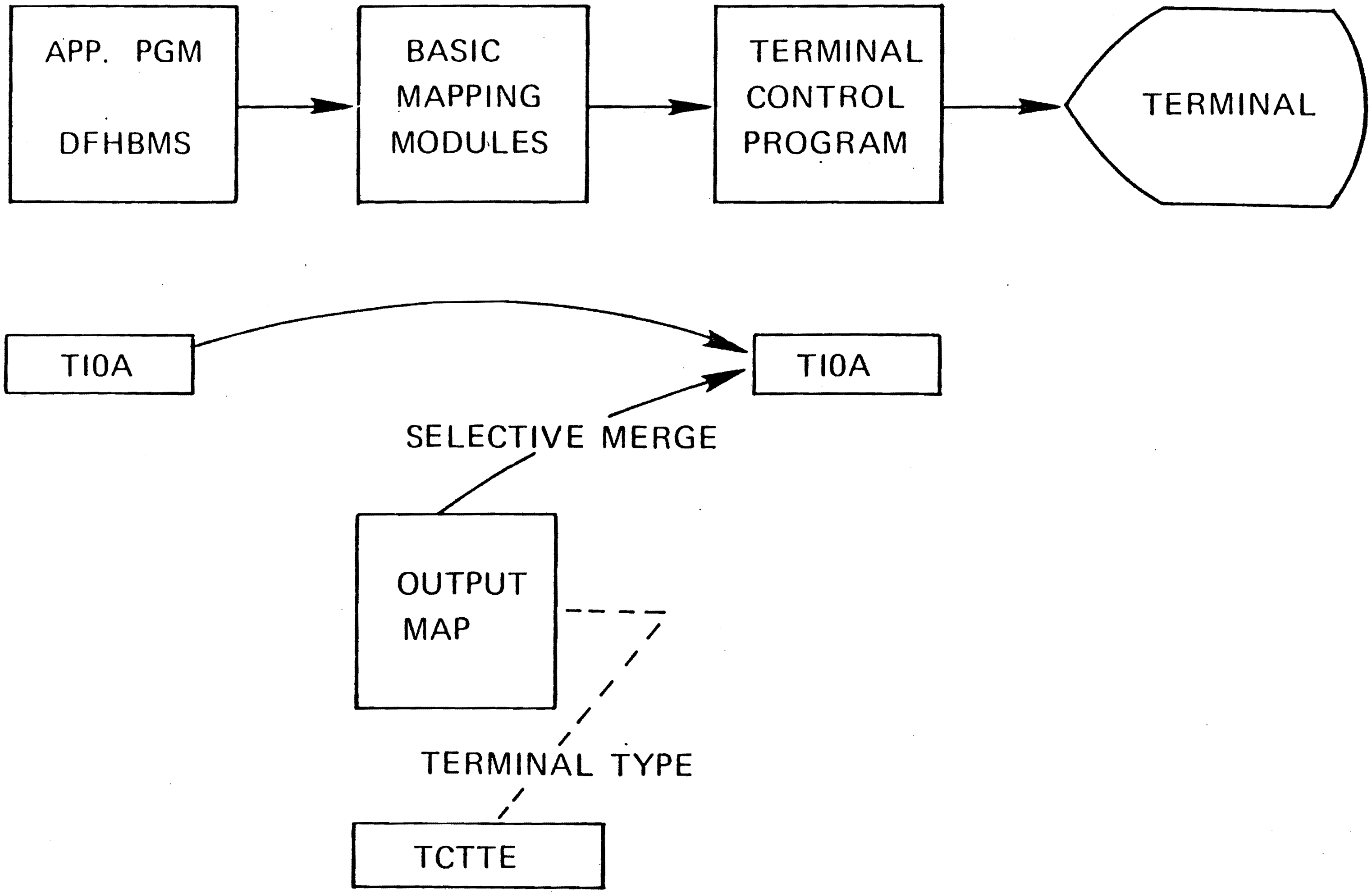
TEXT FORMAT



DATA

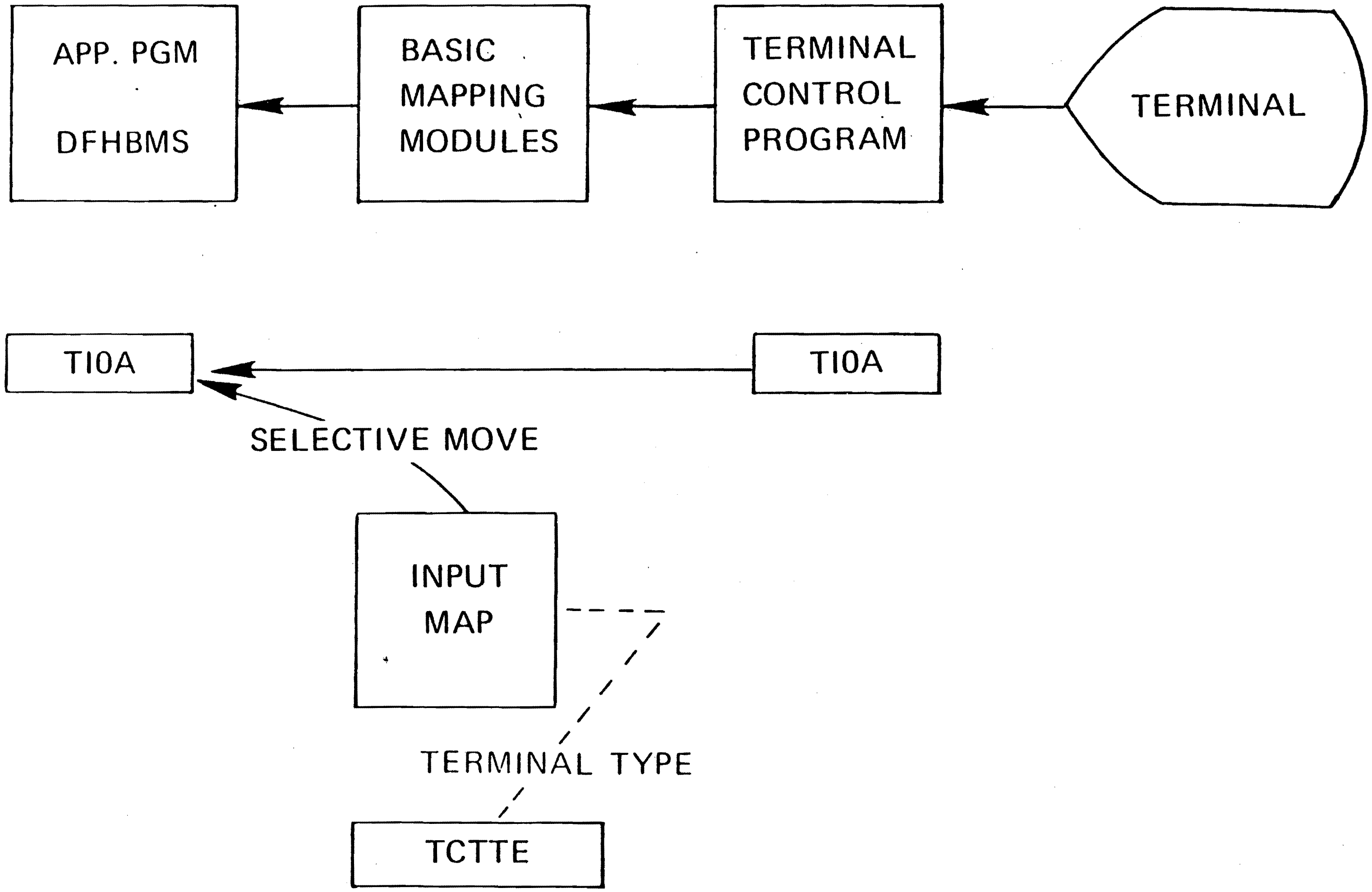
BASIC MAPPING

GENERAL FLOW OUTPUT



BASIC MAPPING

GENERAL FLOW INPUT



```
TIOABAR    EQU      10
           COPY     DFHTIOA
           COPY     MAPSETA
           ---
           L        TCTTEAR,TCAFCAAA
           DFHSC    TYPE=FREEMAIN,RELEASE=ALL
INITIAL    DFHBMS   TYPE=(ERASE,OUT),MAP=MAPA,MAPSET=MAPSETA,DATA=NO
READIN     DFHBMS   TYPE=IN,MAP=MAPA,MAPSET=MAPSETA
           L        TIOABAR,TCTTEDA
PROCESS    ---
           DFHSC    TYPE=GETMAIN,NUMBYTE=480,INITIMG=00,CLASS=USER
           L        TIOABAR,TCASCSA
           ---
WRITE      ST       TIOABAR,TCTTEDA
           DFHBMS   TYPE=OUT,MAP=MAPA,MAPSET=MAPSETA,DATA=YES
```

Facilities Control [illegible] A [illegible]

TERM

3+NFIELDS + 5 x (LGH OF FLDS)

```
01 DFHBLLDS COPY DFHBLLDS.
   02 ---
   02 TIOABAR PIC S9(8) COMP.
   02 MAPBAR PIC S9(8) COMP.
01 ---
01 DFHTIOA COPY DFHTIOA.
                                   ◄----01 CBASE PIC X(---)
01 MAPAI COPY MAPSETA.
PROCEDURE DIVISION.
      MOVE CSACDTA TO TCACBAR.

      MOVE TCAFCAAA TO TCTTEAR.
   DFHSC    TYPE=FREEMAIN,RELEASE=ALL
   DFHBMS   TYPE=(ERASE,OUT),MAP=MAPA,MAPSET=MAPSETA,DATA=NO
READIN.
   DFHBMS   TYPE=IN,MAP=MAPA,MAPSET=MAPSETA
      MOVE TCTTEDA TO TIOABAR.
      ADD 12 TO TIOABAR GIVING MAPBAR
PROCESS.
      ---
   DFHSC    TYPE=GETMAIN,CLASS=TERMINAL,INITIMG=00,        X
            NUMBYTE=480
      MOVE TCASCSA TO TIOABAR.
      ADD 12 TO TIOABAR GIVING MAPBASE.
      ---
WRITE
      MOVE TIOABAR TO TCTTEDA.
      DFHBMS TYPE=OUT,MAP=MAPA,MAPSET=MAPSETA,DATA=YES
```

```
%INCLUDE (DFHTIOA);
  2 TIOADATA CHAR (1);
%INCLUDE (MAPSETA);
———
    DFHSC     TYPE=FREEMAIN,RELEASE=ALL
    DFHBMS   TYPE=(ERASE,OUT),MAP=MAPA,MAPSET=MAPSETA,DATA=NO
READIN:
    DFHBMS   TYPE=IN,MAP=MAPA,MAPSET=MAPSETA
  TIOABAR=TCTTEDA,
  BMSMAPBR=ADDR(TIOADATA);
PROCESS:
  ———
    DFHSC     TYPE=GETMAIN,NUMBYTE=480,INITIMG=00,CLASS= USER   TCA
  TIOABAR=TCASCSA,
  BMSMAPBR=ADDR(TIOADATA);
  ———
WRITE:
  TCTTEDA=TIOABAR;
    DFHBMS   TYPE=OUT,MAP=MAPA,MAPSET=MAPSETA,DATA=YES
```

MAPPING

COPY DFHBMSCA

DFHBMPEM	3270 PRINTER END OF MESSAGE
DFHBMPNL	3270 PRINTER NEW LINE SYMBOL
DFHBMASK	AUTOSKIP
DFHBMUNP	UNPROTECTED
DFHBMUNN	UNPROTECTED AND NUMERIC
DFHBMPRO	PROTECTED
DFHBMBRY	HIGH INTENSITY
DFHBMDAR	DARK, NONPRINT
DFHBMFSE	MDT ON
DFHBMPRF	PROTECTED AND MDT ON
DFHBMASF	AUTOSKIP AND MDT ON
DFHMASB	AUTOSKIP AND HIGH INTENSITY

MAPPING

TLFIELD

COPY DFHAID

DFHENTER	ENTER KEY
DFHCLEAR	CLEAR KEY
DFHPEN	IMMEDIATELY DETECTABLE FIELD
DFHPA1	PA1 KEY
DFHPA2	PA2 KEY
DFHPA3	PA3 KEY
DFHPF1	PF1 KEY
●	●
●	●
●	●
DFHPF12	PF12 KEY

BASIC MAPPING

HOW TO USE BASIC MAPPING

DEFINE NEEDS OF OPERATOR
- HEADINGS
- DATA
  - DISPLAYABLE
  - HIGH LIGHTED
  - PEN DETECTABLE
- CURSOR POSITION

OUT

DEFINE NEEDS OF APPLICATION PROGRAM
- NO HEADINGS
- TYPE OF ACTION
- DATA PERTINENT TO ACTION

IN

out CREATE OUTPUT MAPS TO SUPPLY OPERATOR NEEDS

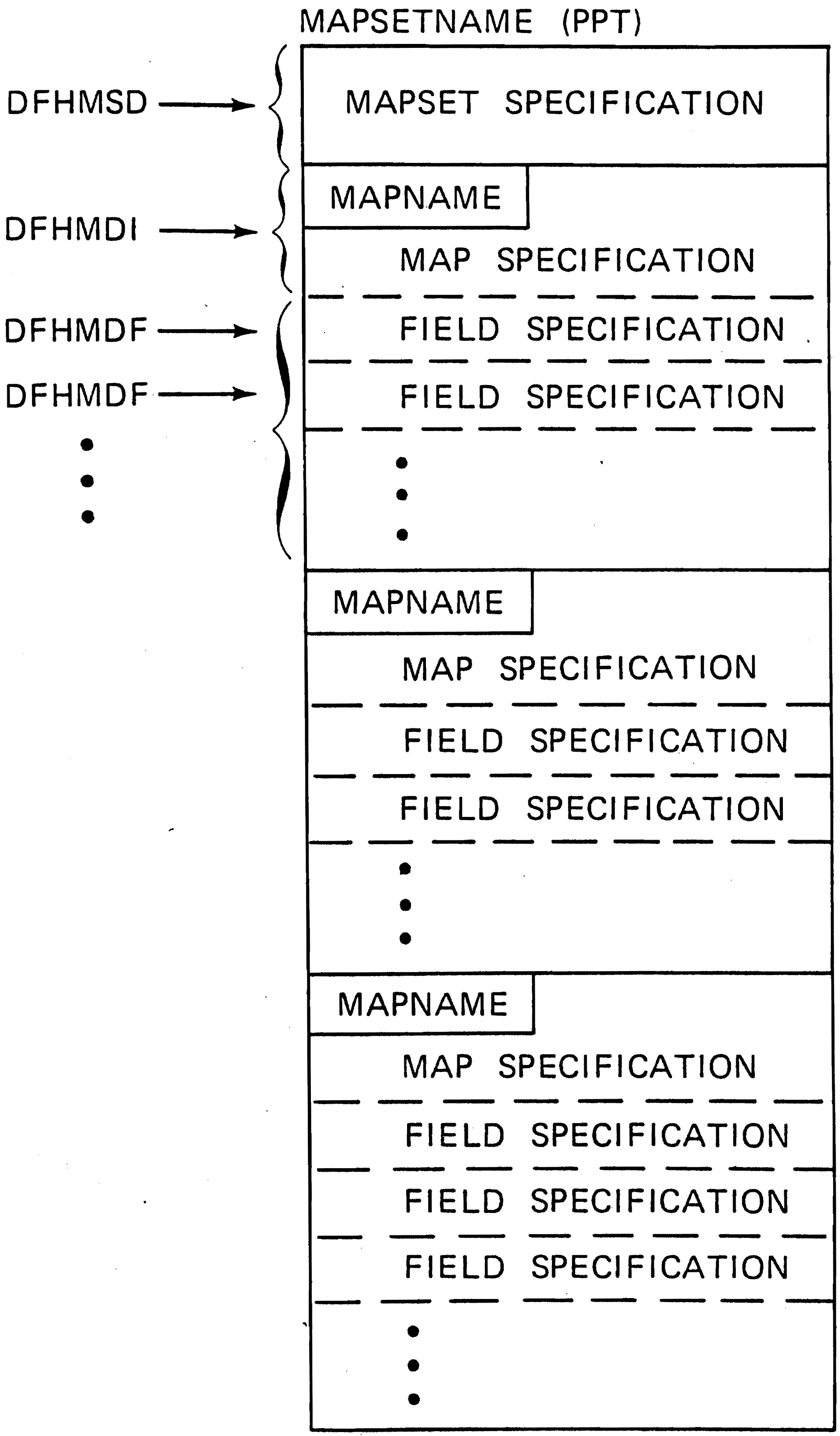
IN CREATE INPUT MAPS TO SUPPLY APPLICATION PROGRAM NEEDS

CREATE DSECTS TO MATCH MAPS (INPUT AND OUTPUT)

Can be product of map-run

MAPPING

MAPSET



MAPSETNAME (PPT)

DFHMSD → MAPSET SPECIFICATION

DFHMDI → MAPNAME

MAP SPECIFICATION

DFHMDF → FIELD SPECIFICATION

DFHMDF → FIELD SPECIFICATION

•
•
•

MAPNAME

MAP SPECIFICATION

FIELD SPECIFICATION

FIELD SPECIFICATION

•
•
•

MAPNAME

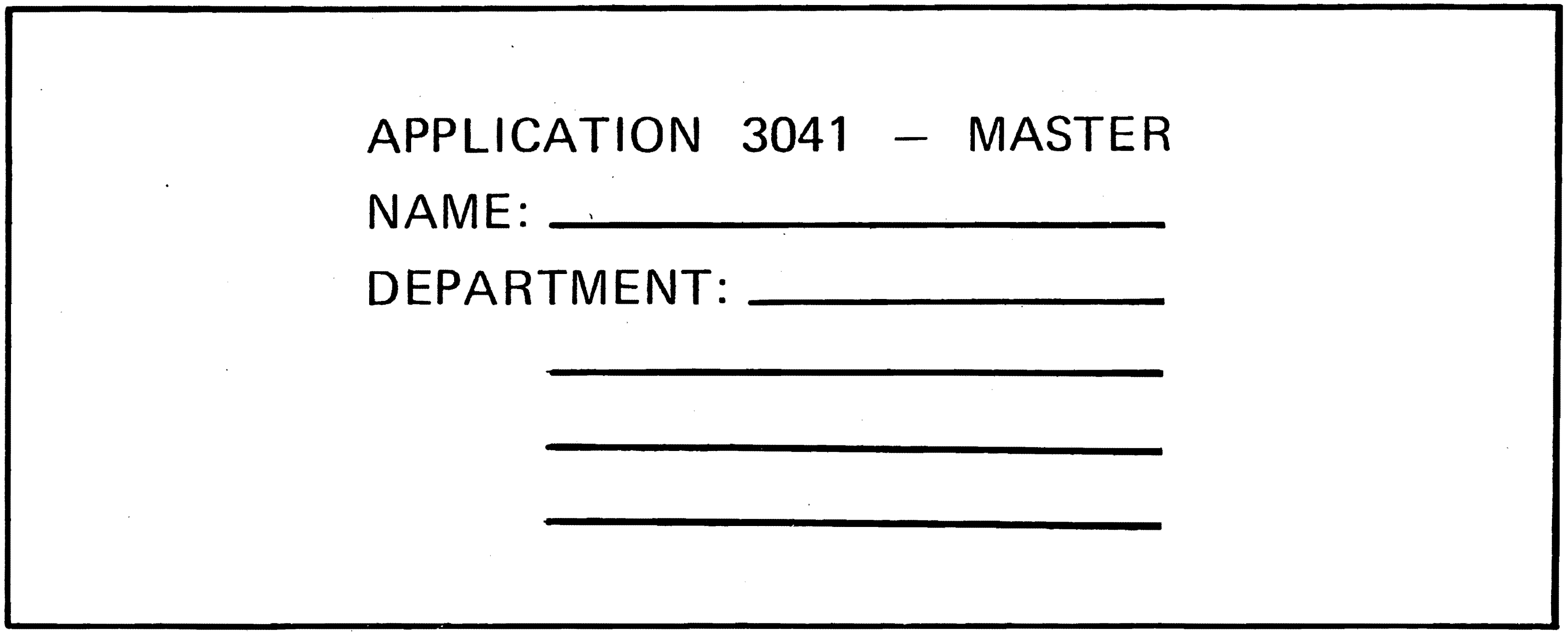
MAP SPECIFICATION

FIELD SPECIFICATION

FIELD SPECIFICATION

FIELD SPECIFICATION

•
•
•



APPLICATION 3041 – MASTER

NAME: ____________________

DEPARTMENT: ______________

____

____

____

NAME:	NAMEI (INPUT)	NAMEO (OUTPUT)
DEPARTMENT:	DEPTI (INPUT)	DEPTO (OUTPUT)

```
MAPSETA    DFHMSD    TYPE=MAP,MODE=INOUT
MAPA       DFHMDI    LINE=1,COLUMN=1,SIZE=(10,40)
           DFHMDF    POS=(1,1),LENGTH=30,ATTRB=(ASKIP,PROT),        X
                     INITIAL='APPLICATION  3041 -MASTER'
           DFHMDF    POS=(2,1),LENGTH=8,ATTRB=(ASKIP,PROT),         X
                     INITIAL='NAME:'
NAME       DFHMDF    POS=(2,9),LENGTH=20,ATTRB=(IC),                X
                     JUSTIFY=(LEFT,BLANK)
           DFHMDF    POS=(3,1),LENGTH=12,ATTRB=(ASKIP,PROT),        X
                     INITIAL='DEPARTMENT:'
                               4
DEPT       DFHMDF    POS=(3,13),LENGTH=(12)
           - - -
MAPB       DFHMDI    (define  next  map  in  mapset)
           - - -
                     (terminate  mapset)
           DFHMSD    TYPE=FINAL
```

MAPSETA — TYPE=DSECT,LANG=ASM

```
MAPAI    DS    OC
MAPAO    DS    OC
NAMEL    DS    CL2        'LENGTH'
NAMEF    DS    OC         'FLAG'
NAMEA    DS    C          'ATTRIBUTE'
NAMEI    DS    OCL20      'NAME  —  INPUT'
NAMEO    DS    CL20       'NAME  —  OUTPUT'
DEPTL    DS    CL2
DEPTF    DS    OC
DEPTA    DS    C
DEPTI    DS    OCL12
DEPTO    DS    CL12
———
MAPAE    EQU   *
         ORG   MAPAI
MAPBI    DS    OC
         ———
```

MAPSETA – TYPE=DSECT,LANG=COBOL

```
01 MAPAI
   02 NAMEL PIC S9(4) COMP.           'LENGTH'
   02 NAMEA PIC X.                    'ATTRIBUTE'
   02 FILLER REDEFINES NAMEA.
      03 NAMEF PIC X.                 'FLAG'
   02 NAMEI PIC X(20).                'NAME – INPUT'
   02 DEPTL PIC S9(4) COMP.
   02 DEPTA PIC X.
   02 FILLER REDEFINES DEPTA.
      03 DEPTF PIC X.
   02 DEPTI PIC X(12).
   ———
01 MAPAO REDEFINES MAPAI.
   02 FILLER X(3).
   02 NAMEO PIC X(20).                'NAME – OUTPUT'
   02 FILLER X(3).
   02 DEPTO PIC X(12)
   ———
01 MAPBI REDEFINES MAPAI.
```

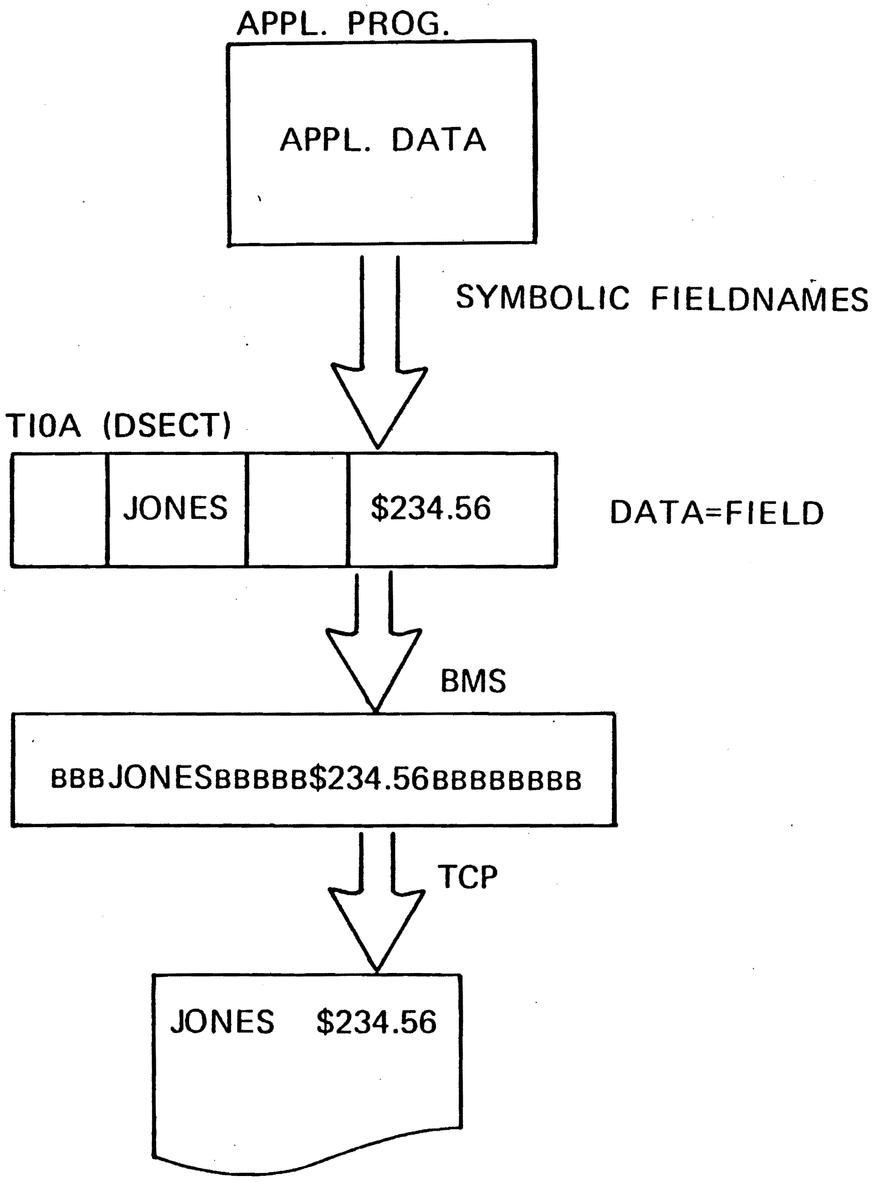
MAPSETA — TYPE=DSECT,LANG=COBOL,BASE=CBASE

```
01 MAPAI REDEFINES CBASE.
   02 NAMEL PIC S9(4) COMP.           'LENGTH'
   02 NAMEA PIC X.                    'ATTRIBUTE'
   02 FILLER REDEFINES NAMEA.
      03 NAMEF PIC X.                 'FLAG'
   02 NAMEI PIC X(20).                'NAME — INPUT'
   02 DEPTL PIC S9(4) COMP.
   02 DEPTA PIC X.
   02 FILLER REDEFINES DEPTA.
      03 DEPTF PIC X.
   02 DEPTI PIC X(12).
   ———
01 MAPAO REDEFINES MAPAI.
   02 FILLER X(3).
   02 NAMEO PIC X(20).                'NAME — OUTPUT'
   02 FILLER X(3).
   02 DEPTO PIC X(12)
   ———
01 MAPBI REDEFINES MAPAI.
```

MAPSETA – TYPE=DSECT,LANG=PL1

```
DECLARE 1 MAPAI BASED(BMSMAPBR),
   2 NAMEL FIXED BINARY (15,0),          'LENGTH'
   2 NAMEA CHAR (1),                     'ATTRIBUTE'
   2 NAMEI CHAR (20),                    'NAME – INPUT'
   2 DEPTL FIXED BINARY (15,0),
   2 DEPTA CHAR (1),
   2 DEPTI CHAR (12),
   ———
DECLARE 1 MAPAO BASED(BMSMAPBR),
   2 DFHMS1 CHAR (2),
   2 NAMEF CHAR (1),                     'FLAG'
   2 NAMEO CHAR (20),                    'NAME – OUTPUT'
   2 DFHMS2 CHAR (2),
   2 DEPTF CHAR (1),
   2 DEPTO CHAR (12),
   ———
DECLARE 1 MAPBI BASED(BMSMAPBR),
   ———
```

HARDCOPY MAPPING



APPL. PROG.

APPL. DATA

SYMBOLIC FIELDNAMES

TIOA (DSECT)

JONES

$234.56

DATA=FIELD

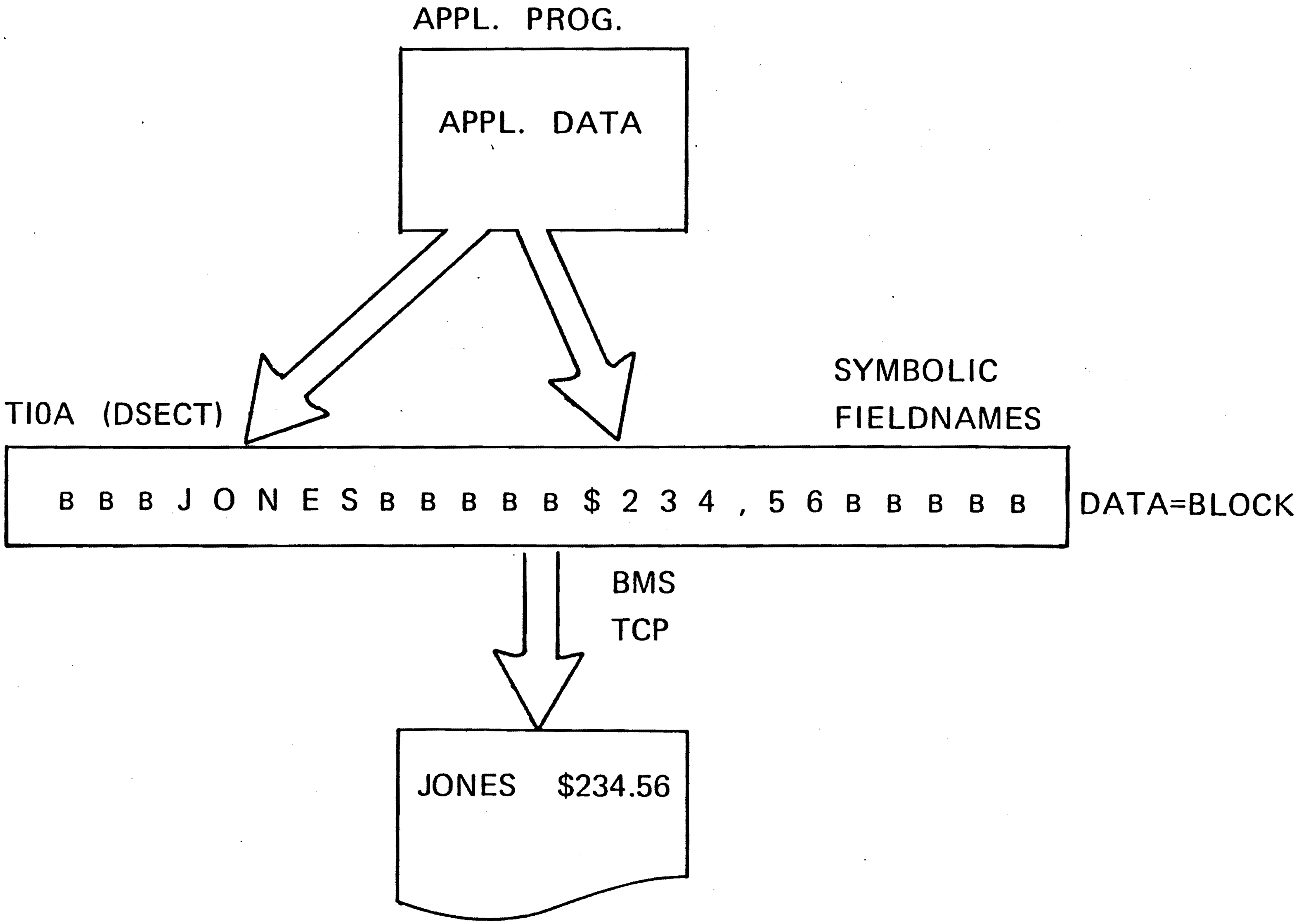
BMS

BBBJONESBBBBB$234.56BBBBBBBB

TCP

JONES $234.56

HARDCOPY MAPPING



APPL. PROG.

APPL. DATA

TIOA (DSECT)

SYMBOLIC
FIELDNAMES

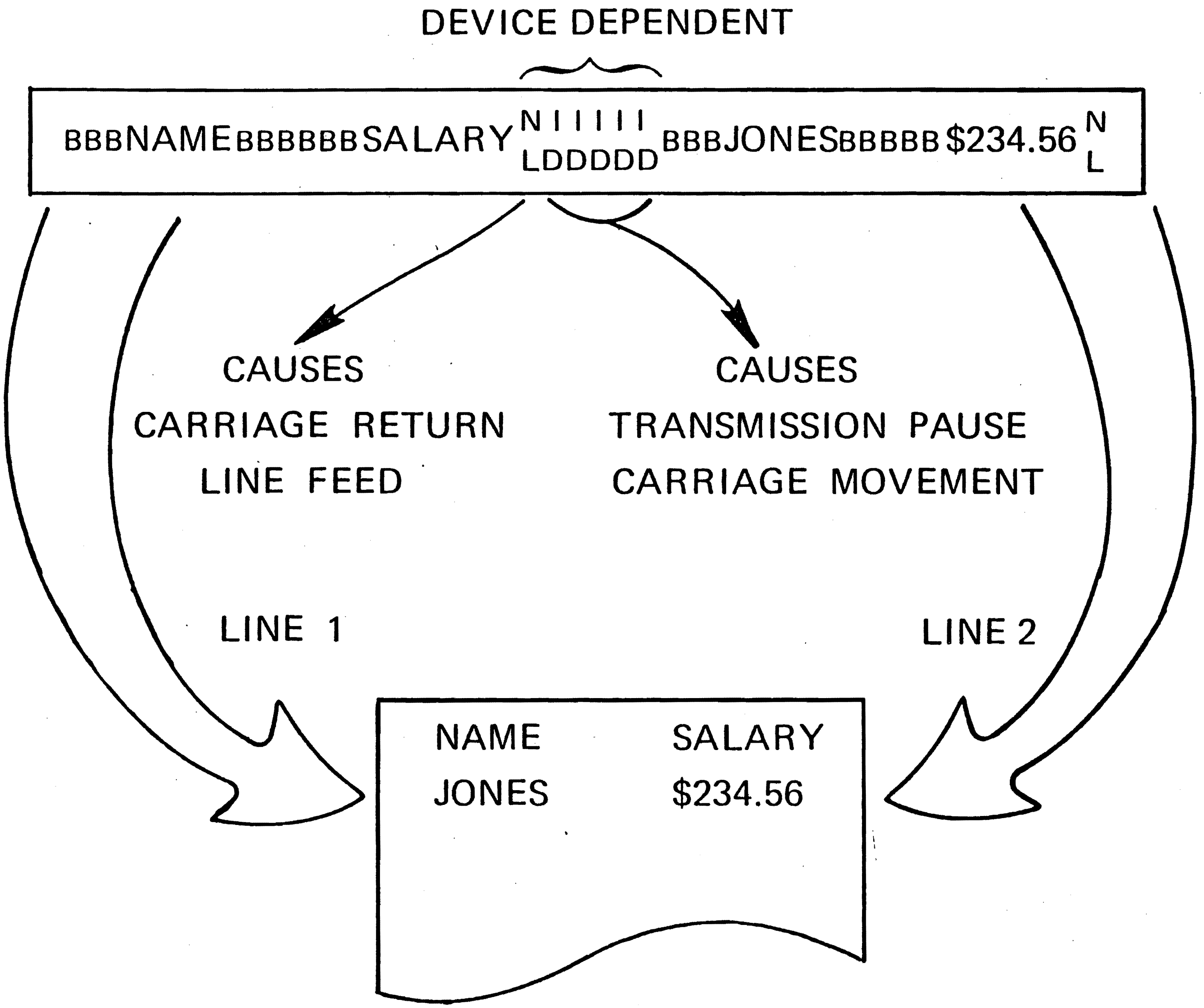
BBBJONESBBBBB$234,56BBBBB

DATA=BLOCK

BMS
TCP

JONES $234.56

LINE CONTROL



DEVICE DEPENDENT

```
                   NIIIII                        N
BBBNAMEBBBBBBSALARY      BBBJONESBBBBB $234.56
                   LDDDDD                        L
```

CAUSES
CARRIAGE RETURN
LINE FEED

CAUSES
TRANSMISSION PAUSE
CARRIAGE MOVEMENT

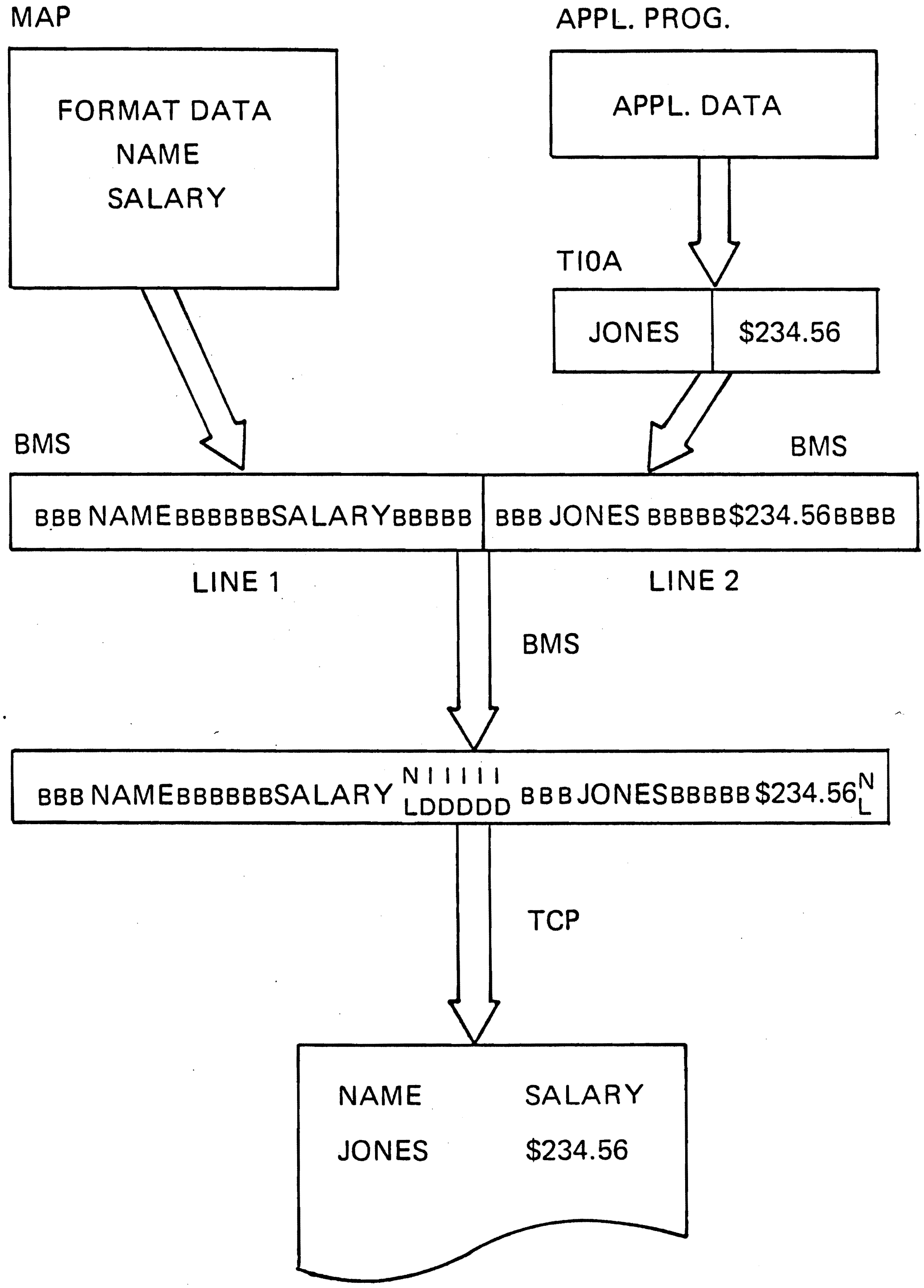
LINE 1

LINE 2

NAME	SALARY
JONES	$234.56

MAPPING

LINE CONTROL



MAP

FORMAT DATA
NAME
SALARY

APPL. PROG.

APPL. DATA

TIOA

JONES	$234.56

BMS

BMS

BBB NAMEBBBBBBSALARYBBBBB	BBB JONES BBBBB$234.56BBBB
LINE 1	LINE 2

BMS

BBB NAMEBBBBBBSALARY NL ID ID ID ID ID BBB JONESBBBBB $234.56 NL

TCP

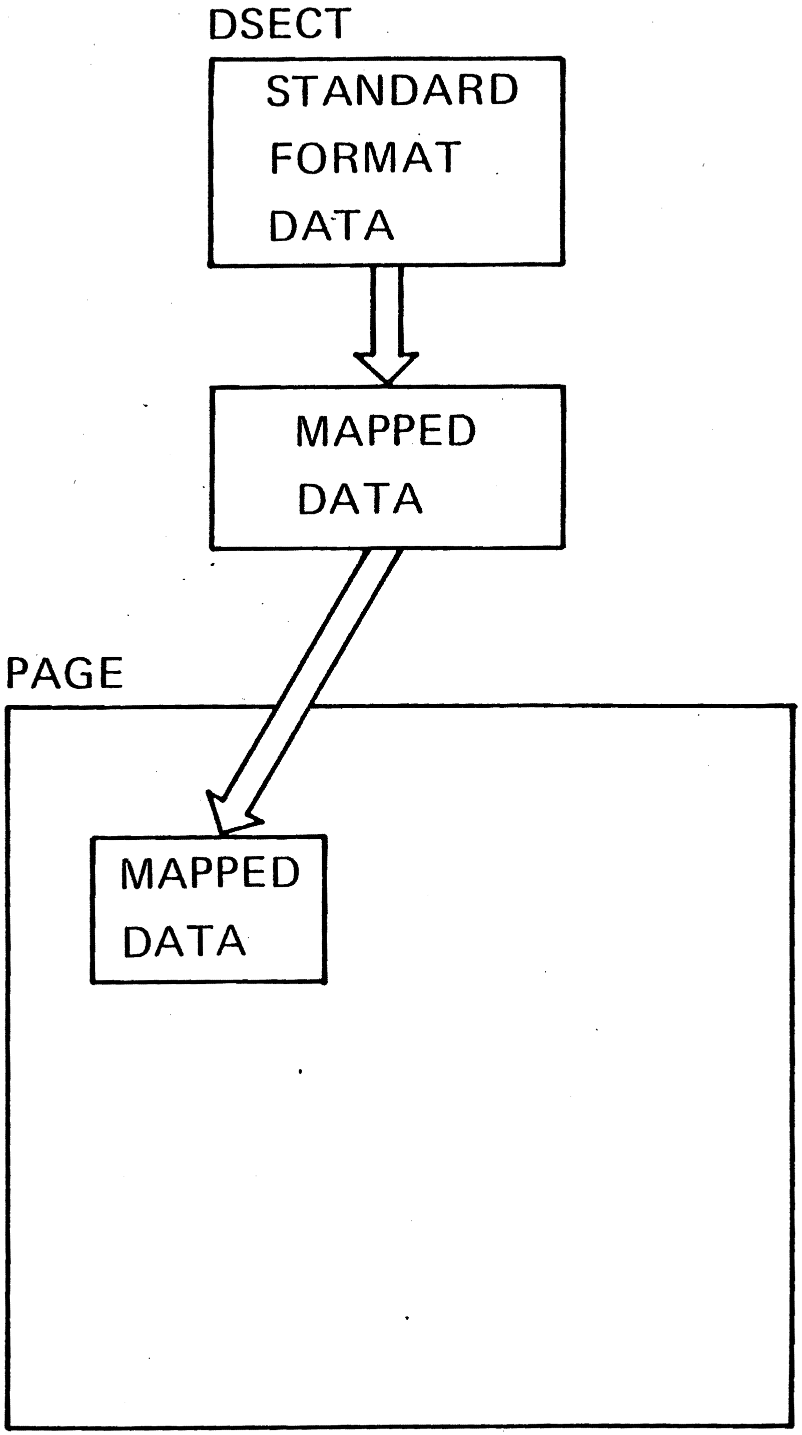
NAME SALARY
JONES $234.56

PAGING

PAGING

MAP | SIZE=(3,20)
PAGE POSITIONING

TCT
PGESIZE=(24,80)



DSECT

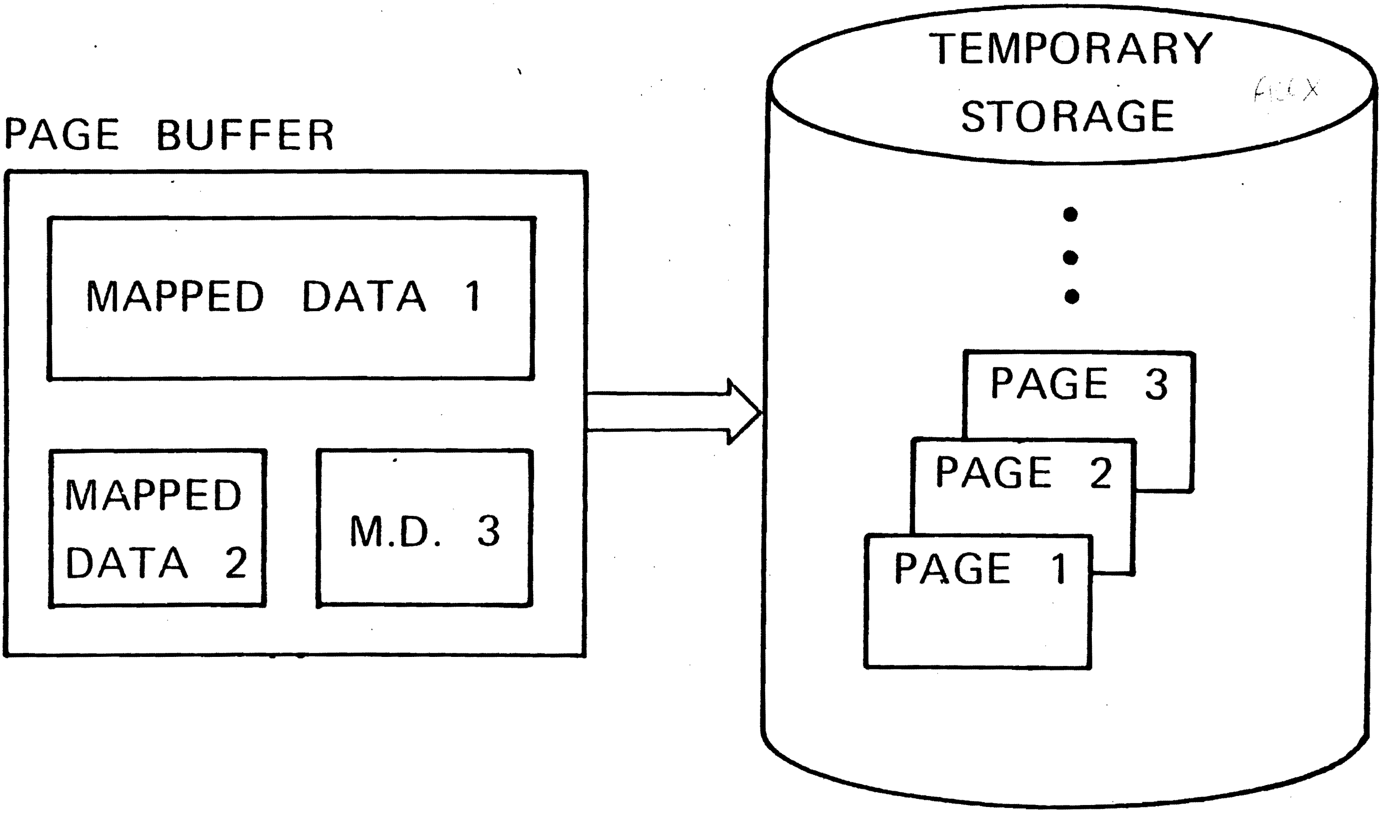
STANDARD FORMAT DATA

MAPPED DATA

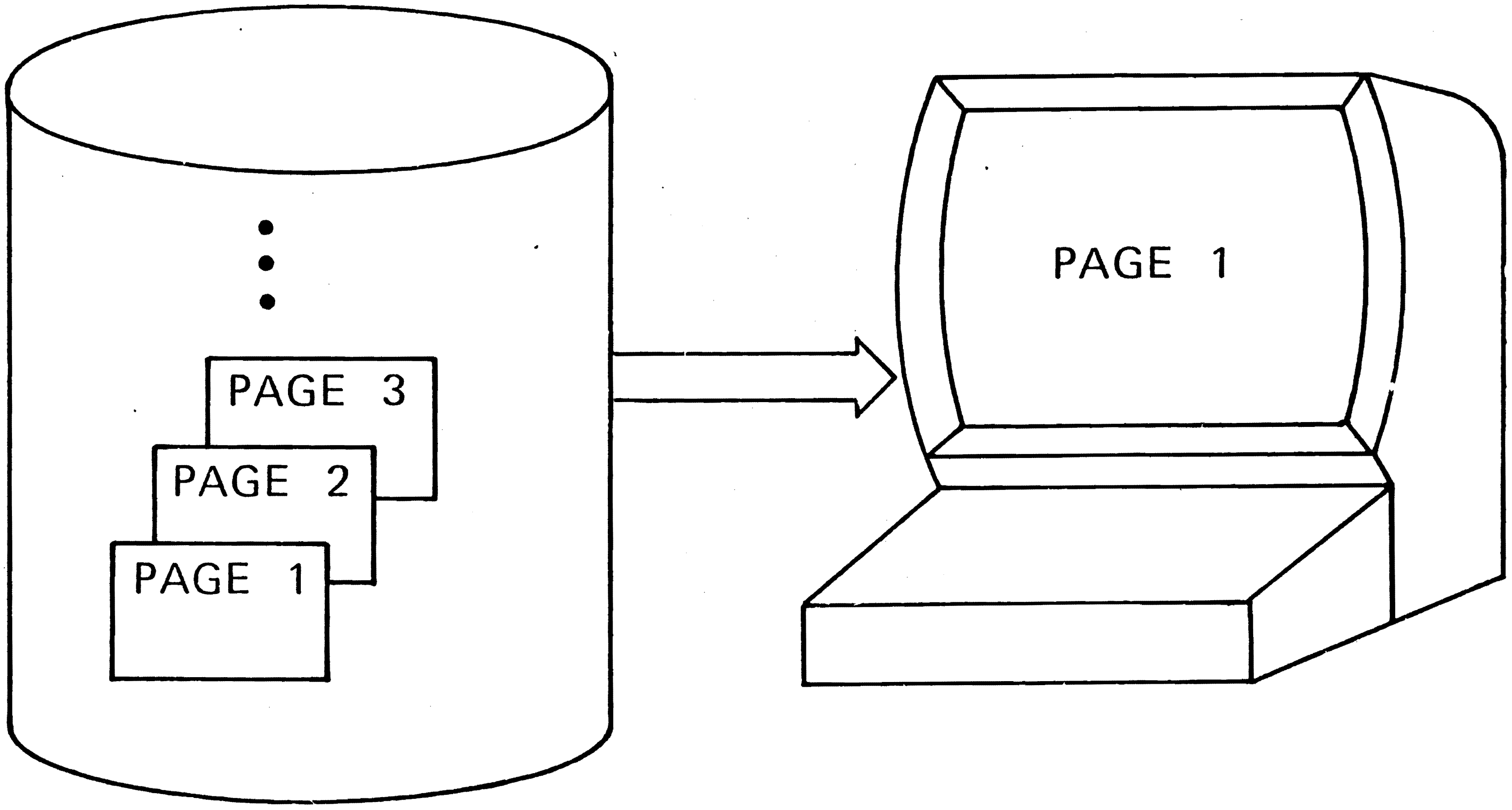
PAGE

MAPPED DATA

PAGING


PAGE BUFFER
MAPPED DATA 1
MAPPED DATA 2
M.D. 3
TEMPORARY STORAGE
AUX
PAGE 3
PAGE 2
PAGE 1

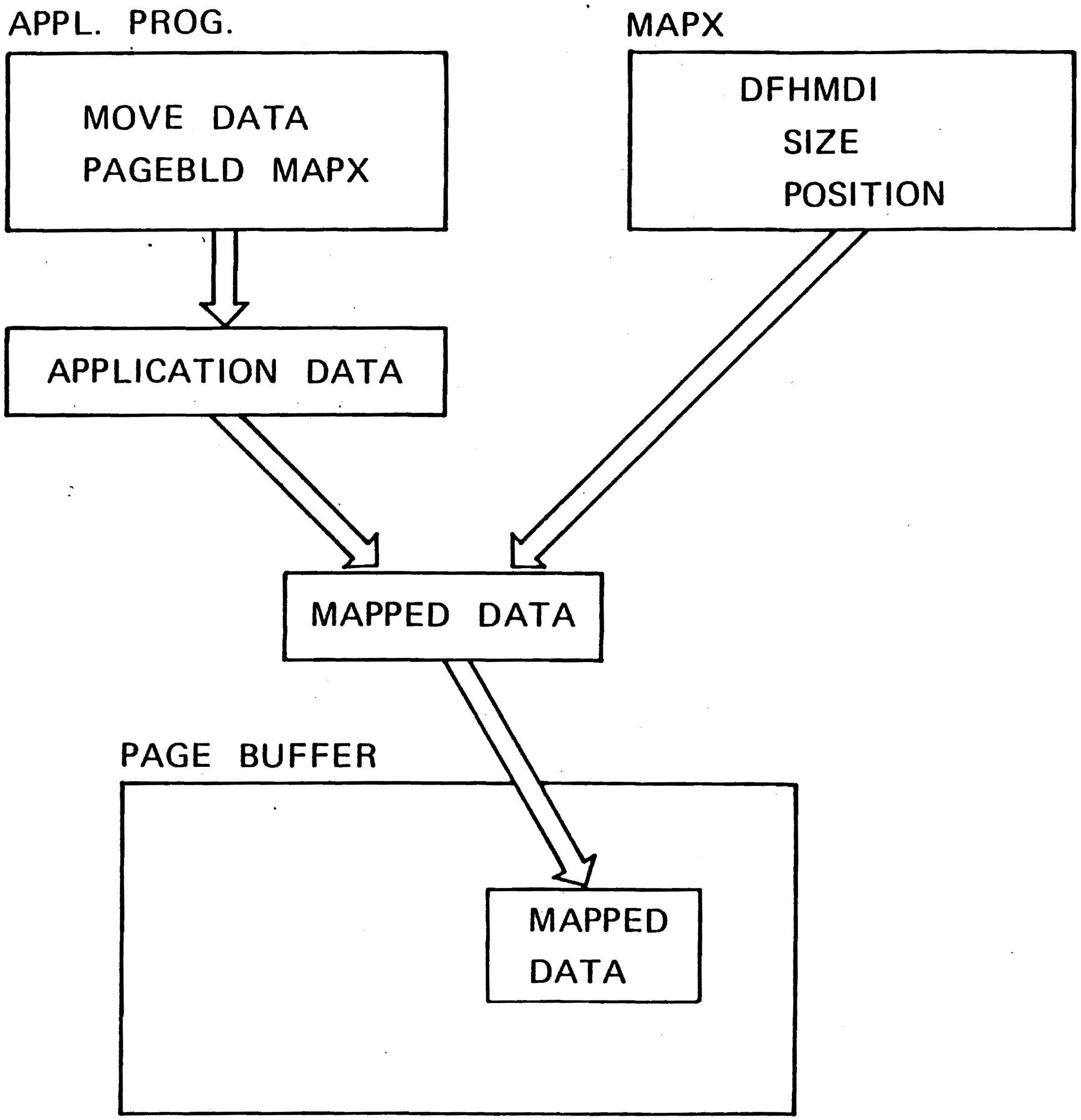
PAGING



PAGING

MAP POSITIONING

PAGING



APPL. PROG.

MOVE DATA
PAGEBLD MAPX

MAPX

DFHMDI
SIZE
POSITION

APPLICATION DATA

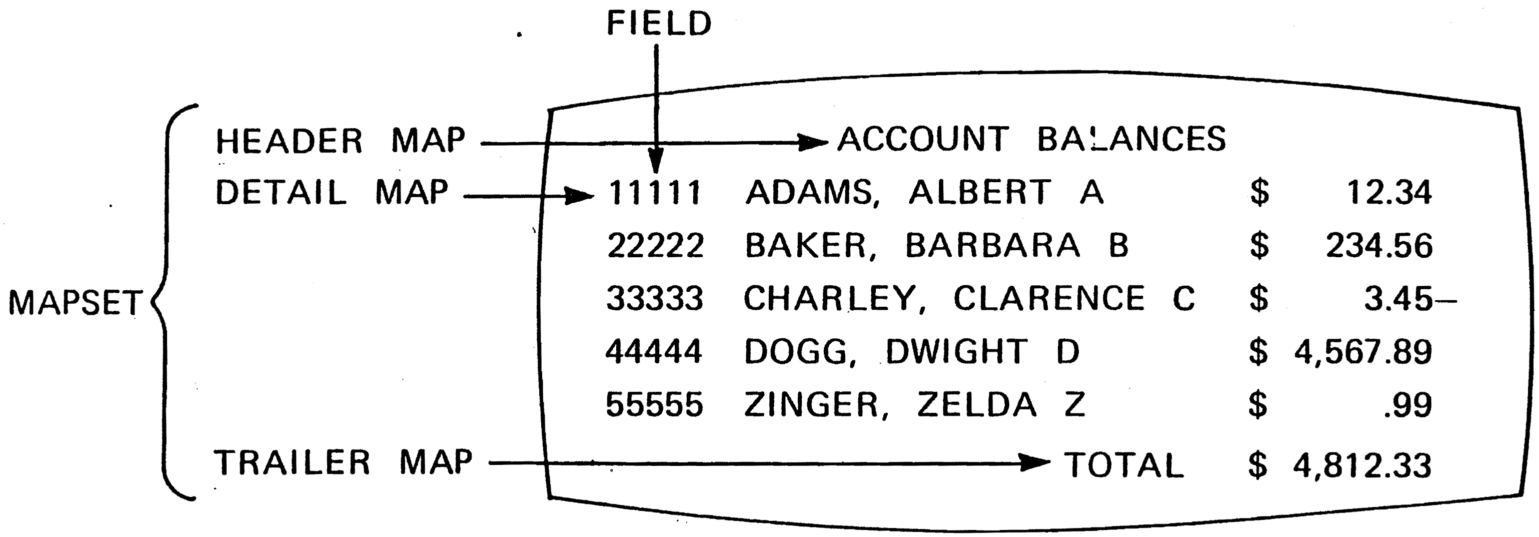
MAPPED DATA

PAGE BUFFER

MAPPED
DATA

PAGING

MAPSET



```
                              ACCOUNT  BALANCES
11111  ADAMS, ALBERT A            $     12.34
22222  BAKER, BARBARA B           $    234.56
33333  CHARLEY, CLARENCE C        $      3.45-
44444  DOGG, DWIGHT D             $  4,567.89
55555  ZINGER, ZELDA Z            $       .99
                                 TOTAL  $  4,812.33
```

```
MAPOUT      DFHMSD     TYPE=MAP,MODE=OUT
HEADING     DFHMDI     JUSTIFY=FIRST,SIZE=(1,40),HEADER=YES
TITLE       DFHMDF     POS=...
DETAIL      DFHMDI     LINE=NEXT,COLUMN=1,SIZE=(1,40)
ACCT        DFHMDF     POS=...
NAME        DFHMDF     POS=...
AMT         DFHMDF     POS=...
TRAILER     DFHMDI     JUSTIFY=LAST,SIZE=(1,40),TRAILER=YES
TOTAL       DFHMDF     POS=...
TOTAMT      DFHMDF     POS=...
            DFHMSD     TYPE=FINAL
```

BASIC MAPPING

map DFHMDI [SIZE=(line,column)]

$$\left[,\text{LINE=}\begin{Bmatrix}\text{number}\\ \underline{\text{NEXT}}\\ \text{SAME}\end{Bmatrix}\right]$$

$$\left[,\text{COLUMN=}\begin{Bmatrix}\text{number}\\ \text{NEXT}\\ \underline{\text{SAME}}\end{Bmatrix}\right]$$

$$\left[,\text{JUSTIFY=(}\left[\begin{Bmatrix}\text{LEFT}\\ \text{RIGHT}\end{Bmatrix}\right]\left[,\begin{Bmatrix}\text{FIRST}\\ \text{LAST}\end{Bmatrix}\right]\text{)}\right]$$

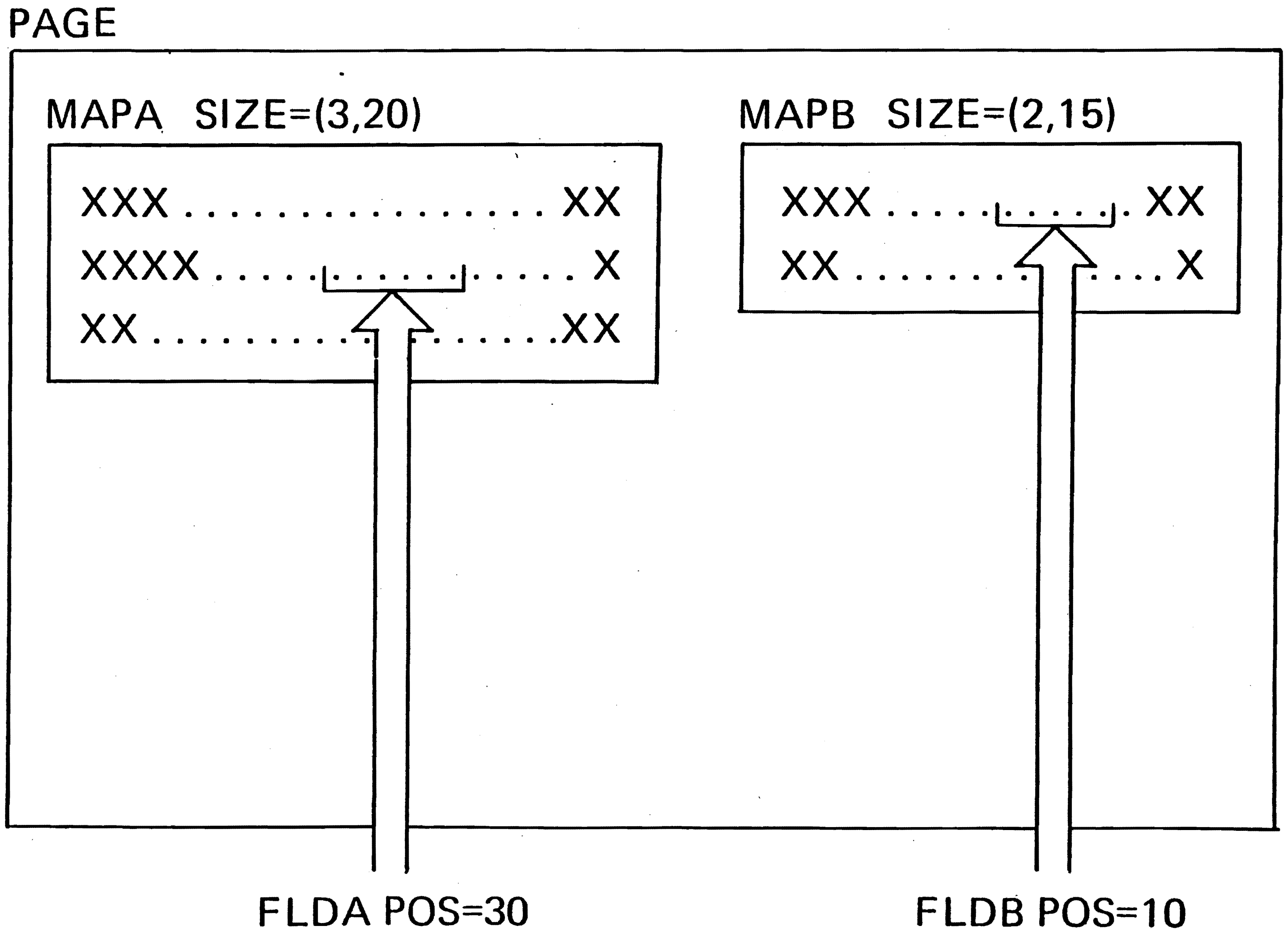
[,HEADER=YES]

[,TRAILER=YES]

$$\left[,\text{DATA=}\begin{Bmatrix}\underline{\text{FIELD}}\\ \text{BLOCK}\end{Bmatrix}\right]$$

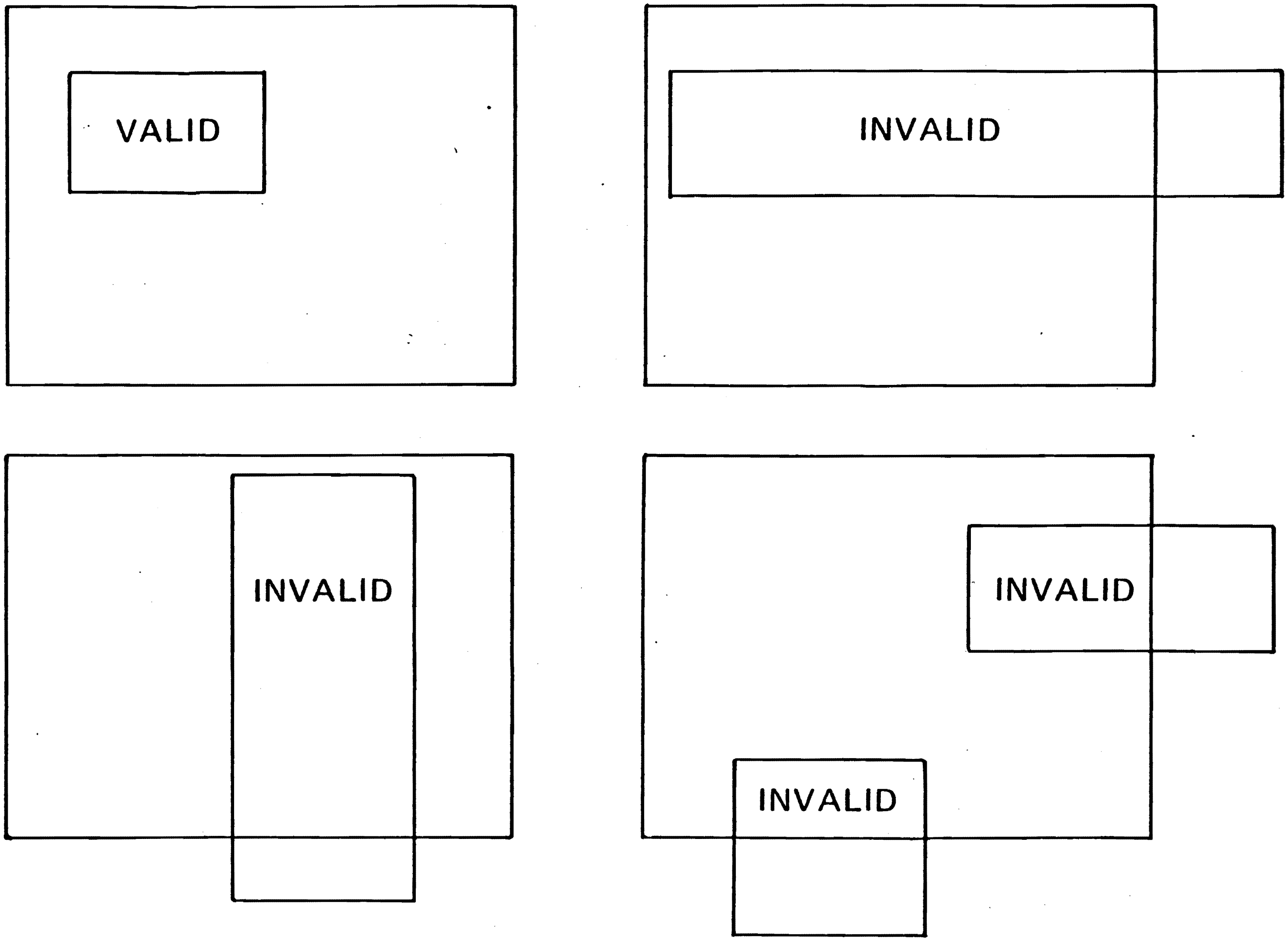
PAGING

SIZE


PAGE
MAPA SIZE=(3,20)
XXX XX
XXXX X
XX XX
MAPB SIZE=(2,15)
XXX XX
XX X
FLDA POS=30
FLDB POS=10

PAGING

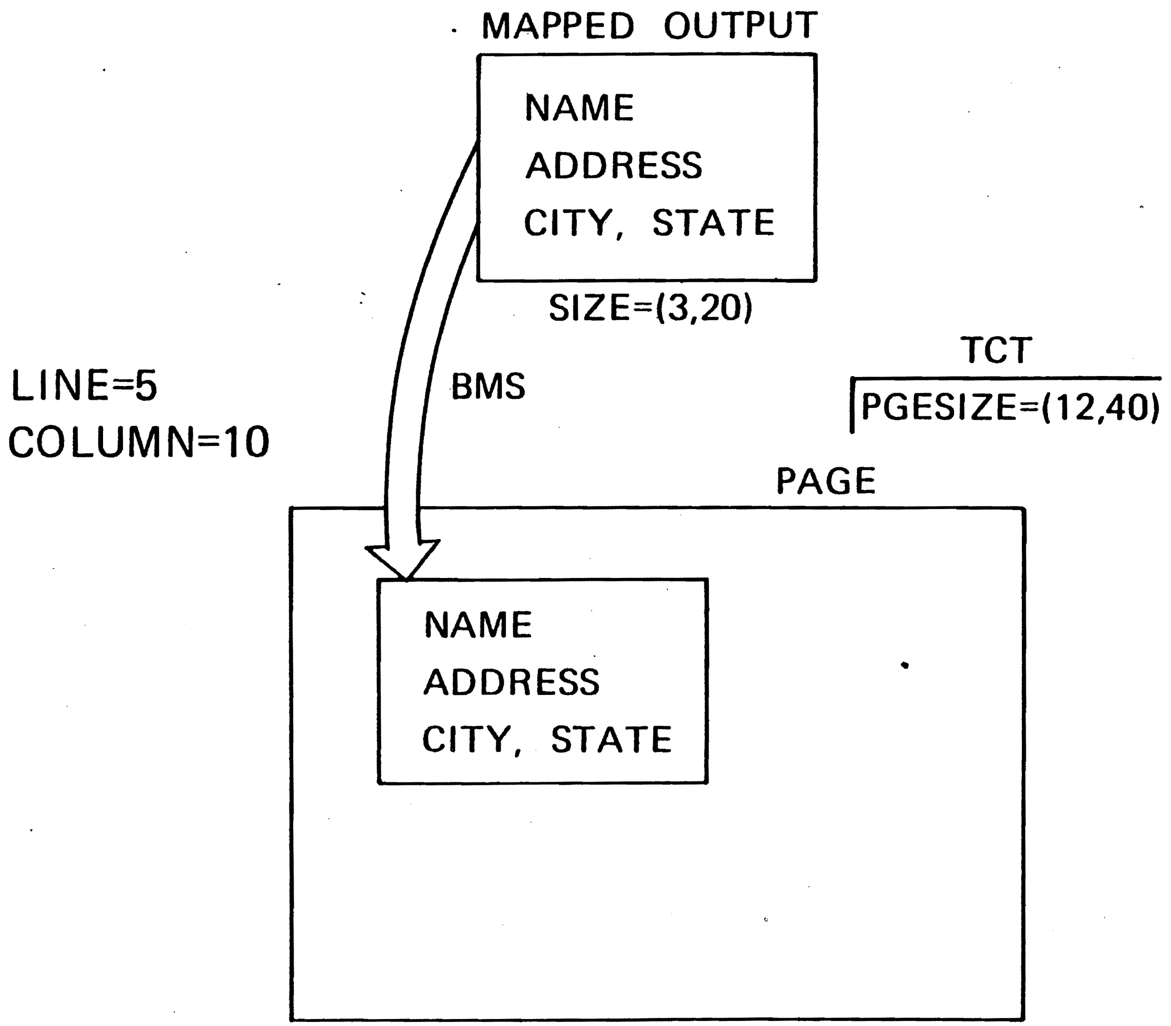
MAP POSITIONING



PAGING

MAP POSITIONING

BMS POSITIONS A MAP AT A SPECIFIED LINE AND COLUMN WITHIN A PAGE OF OUTPUT.



MAPPED OUTPUT

NAME
ADDRESS
CITY, STATE

SIZE=(3,20)

BMS

LINE=5
COLUMN=10

TCT
PGESIZE=(12,40)

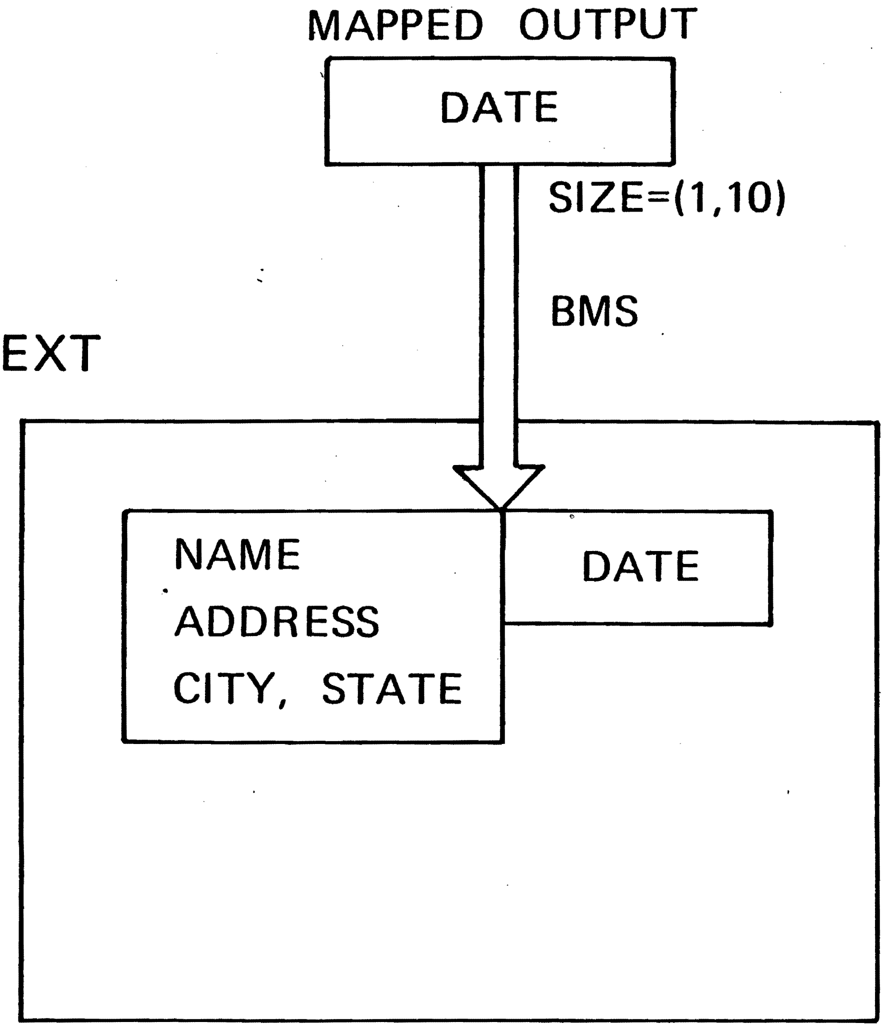
PAGE

NAME
ADDRESS
CITY, STATE

PAGING

MAP POSITIONING

BMS POSITIONS A MAP AT THE SAME LINE AND NEXT COLUMN.

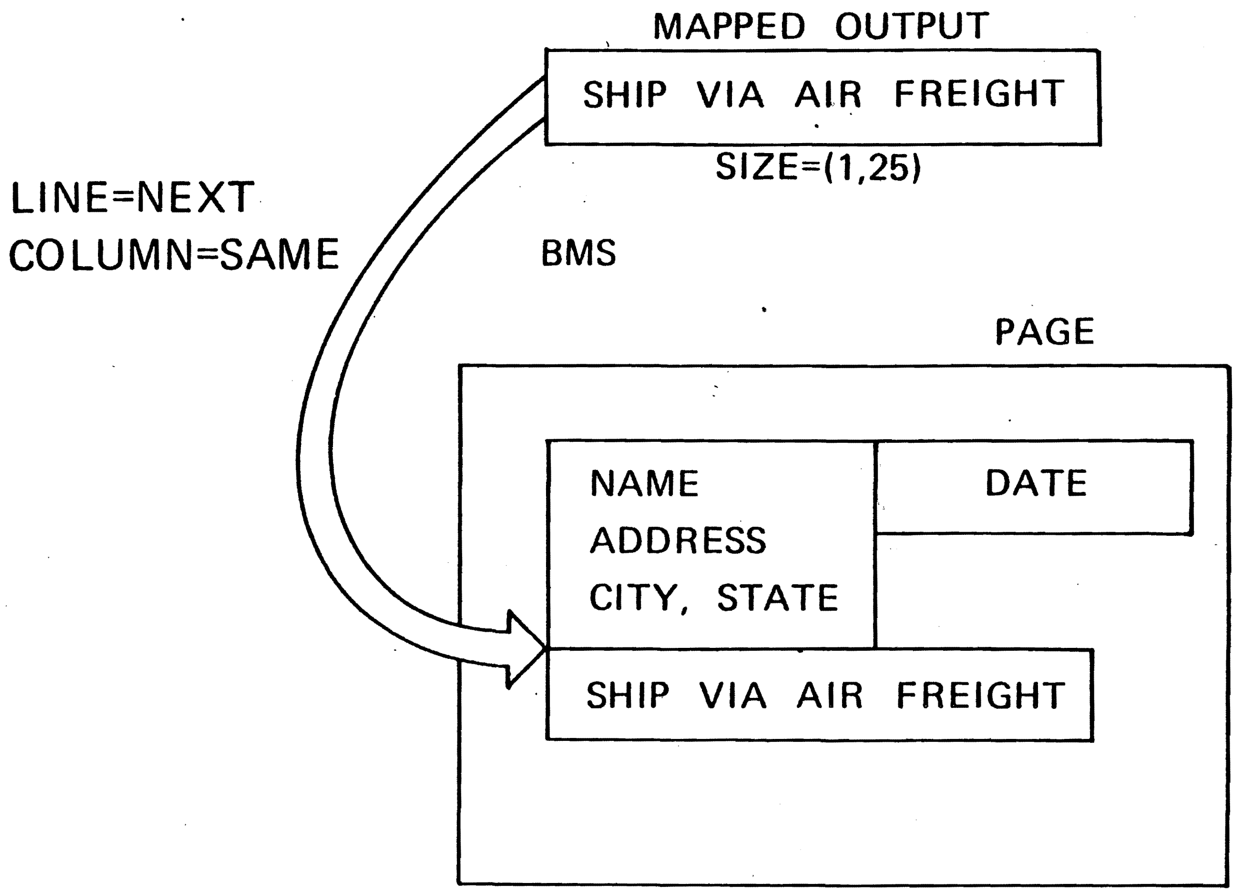


LINE=SAME
COLUMN=NEXT

PAGING

MAP POSITIONING

BMS POSITIONS A MAP AT THE NEXT FULL LINE AND SAME COLUMN.



MAPPED OUTPUT

SHIP VIA AIR FREIGHT

SIZE=(1,25)

LINE=NEXT
COLUMN=SAME

BMS

PAGE

NAME
ADDRESS
CITY, STATE

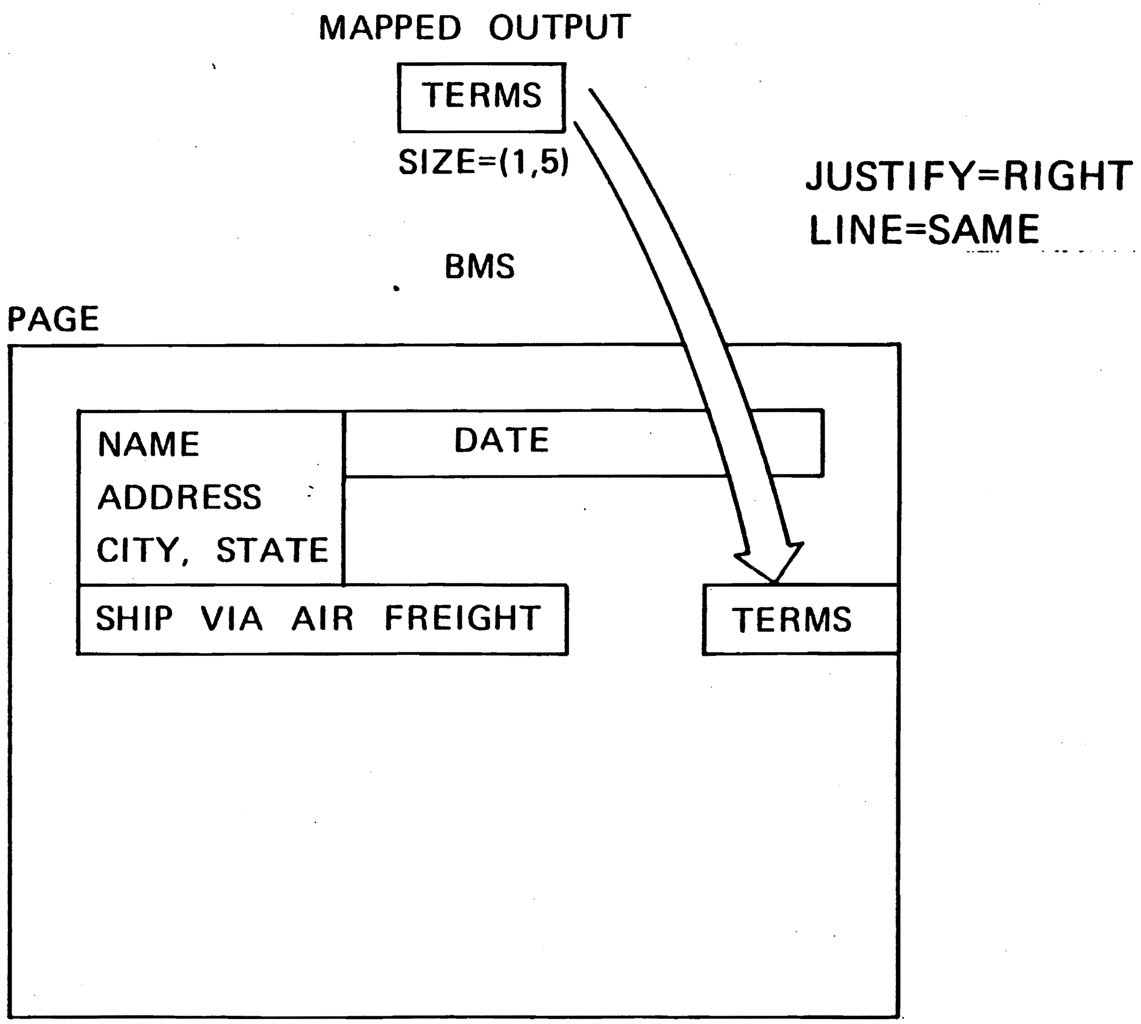
DATE

SHIP VIA AIR FREIGHT

PAGING

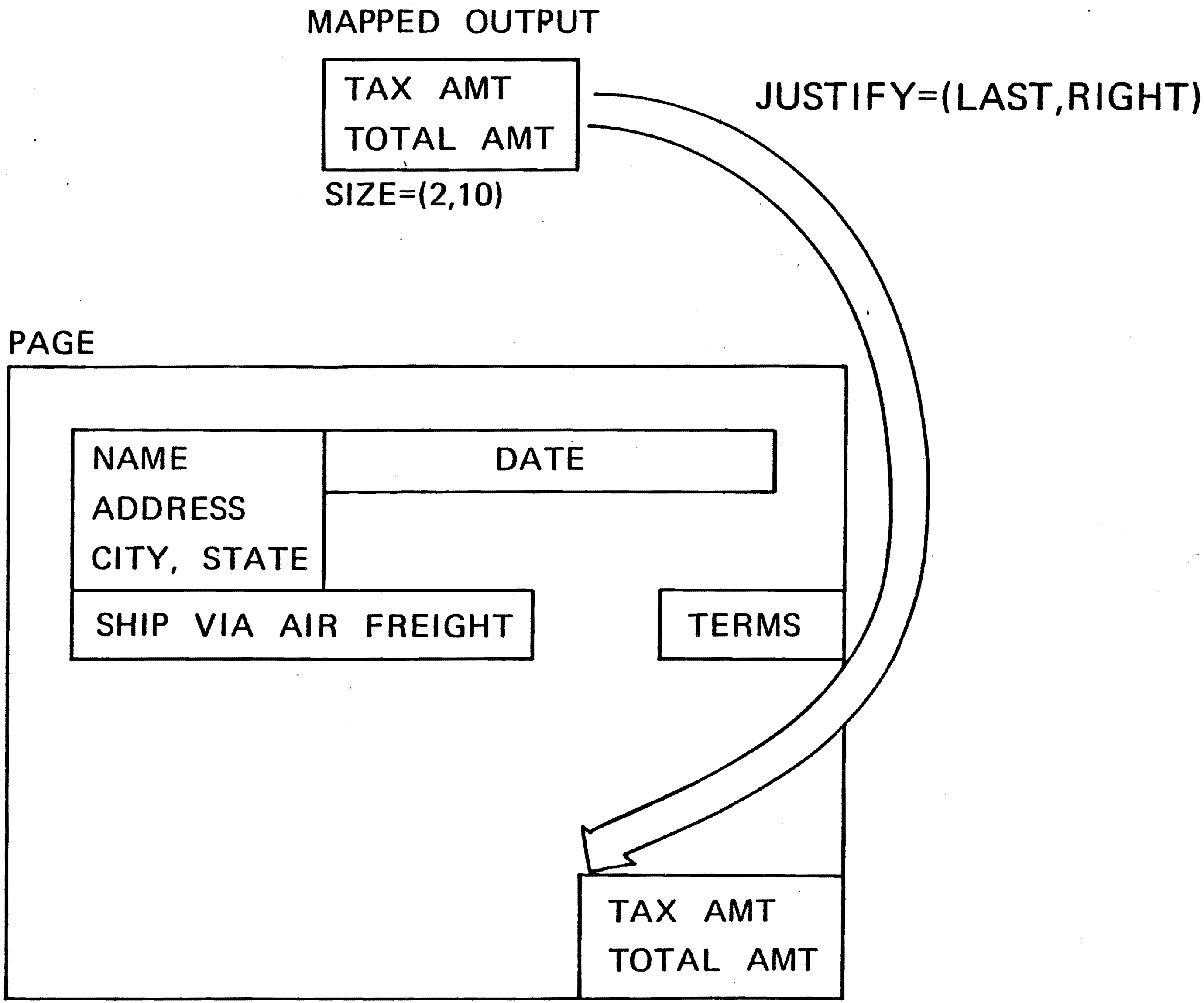
MAP POSITIONING

BMS POSITIONS A MAP JUSTIFIED TO THE LEFT OR RIGHT.



PAGING

MAP POSITIONING



MAPPED OUTPUT

TAX AMT
TOTAL AMT

SIZE=(2,10)

JUSTIFY=(LAST,RIGHT)

PAGE

NAME
ADDRESS
CITY, STATE

DATE

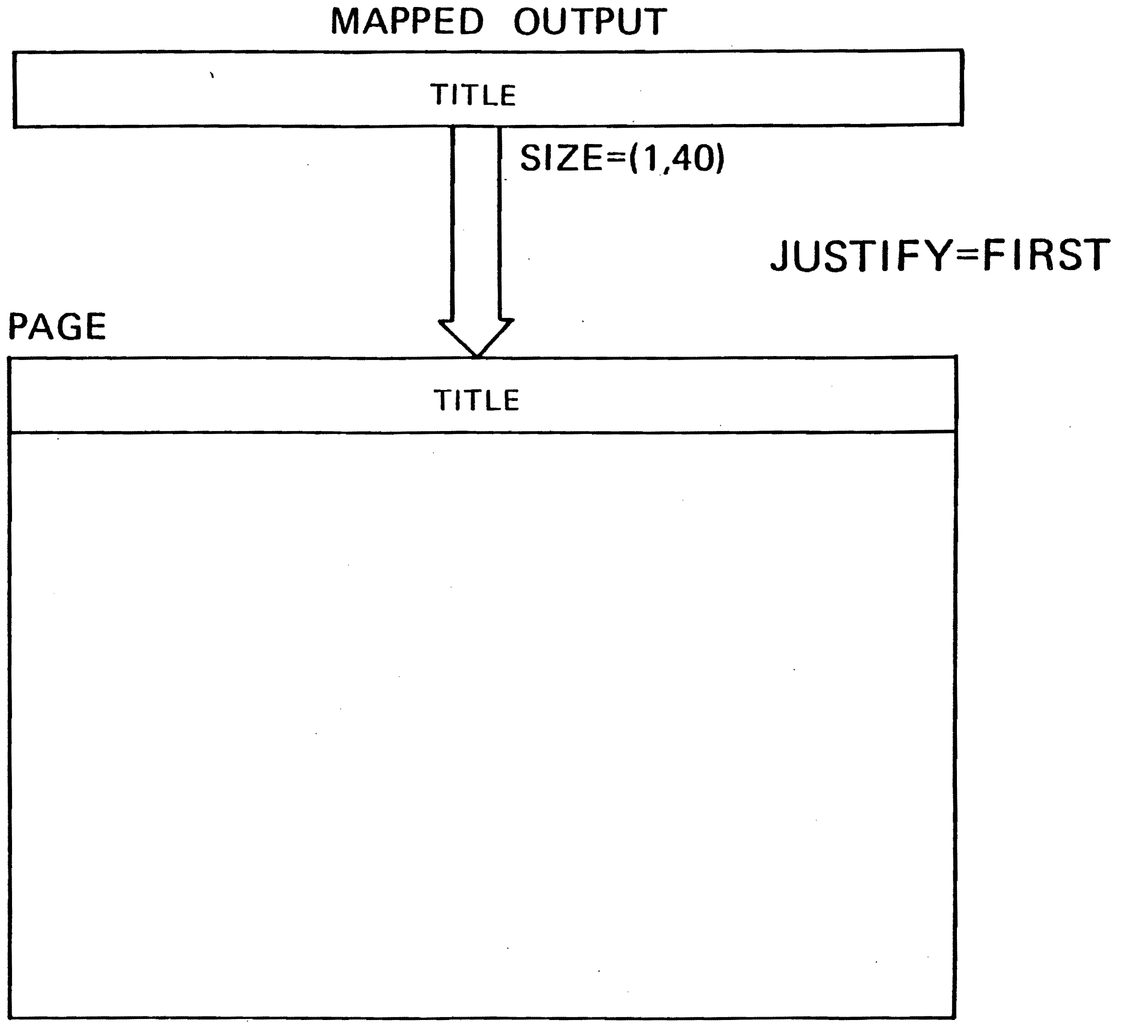
SHIP VIA AIR FREIGHT

TERMS

TAX AMT
TOTAL AMT

PAGING

MAP POSITIONING



PAGING

SUMMARY –

PAGE BUILD MAPS FROM TOP TO BOTTOM OF PAGE

PAGE BUILD MAPS FROM LEFT MARGIN AND/OR RIGHT MARGIN IN

PAGE BUILD MAPS ON SAME LINE WITH EQUAL LINE SIZE

PAGING

MAP POSITIONING

LINE.

LINE POSITION	LINE=1-240	LINE=SAME	LINE=NEXT
JUSTIFY=FIRST	LINE # ON NEW PAGE	TOP OF NEW PAGE	TOP OF NEW PAGE
JUSTIFY=LAST	BOTTOM OF PAGE	BOTTOM OF PAGE	BOTTOM OF PAGE
NEITHER	LINE NUM	SAME LINE	NEXT FULL LINE

COL

COL. POSITION	COL.=1-240	COL.=SAME	COL.=NEXT
JUSTIFY=LEFT	COLS FROM LEFT	SAME COL FROM LEFT	NEXT COL FROM LEFT
JUSTIFY=RIGHT	COLS FROM RIGHT	SAME COL FROM RIGHT	NEXT COL FROM RIGHT

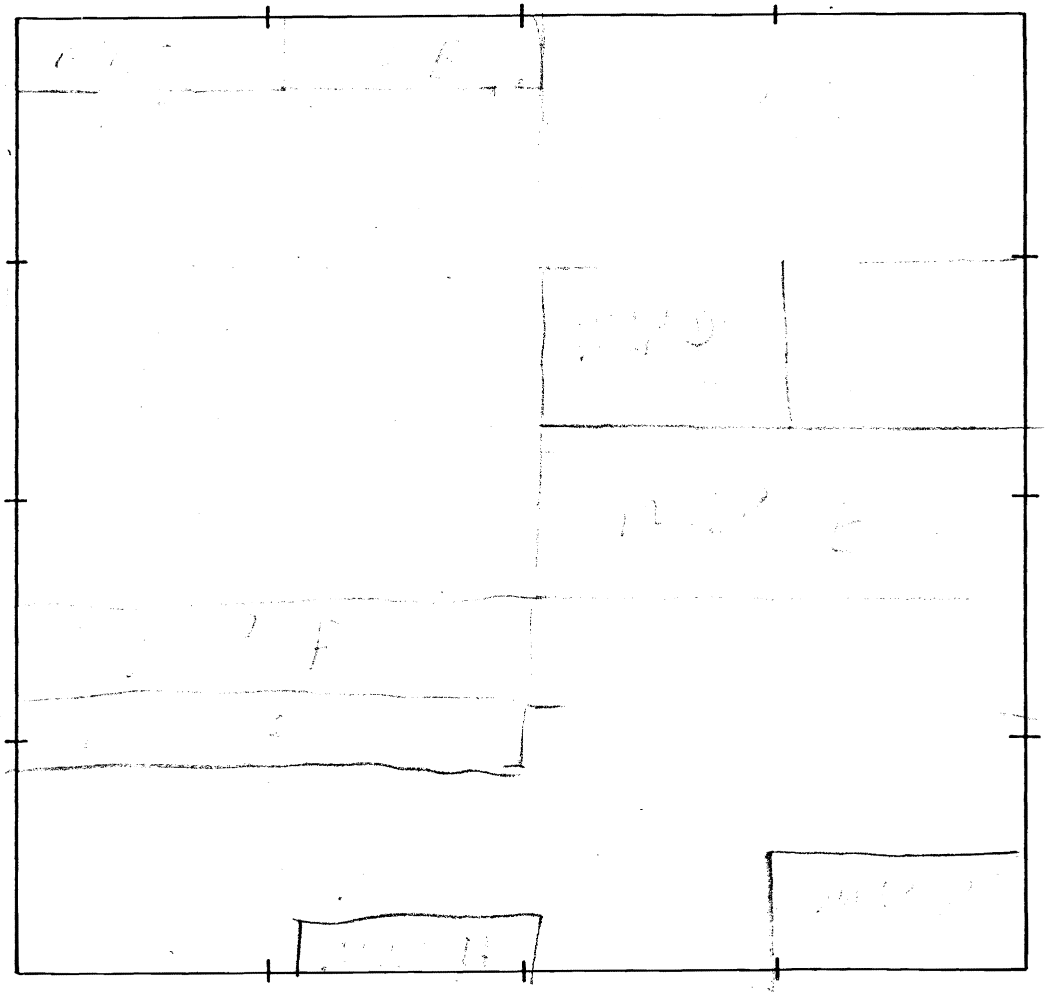
PAGE OVERFLOW	

	JUSTIFY=FIRST
	LINE=nn,COLUMN=nn IN FORMATTED AREA
	MAP WON'T FIT ON REMAINDER OF PAGE

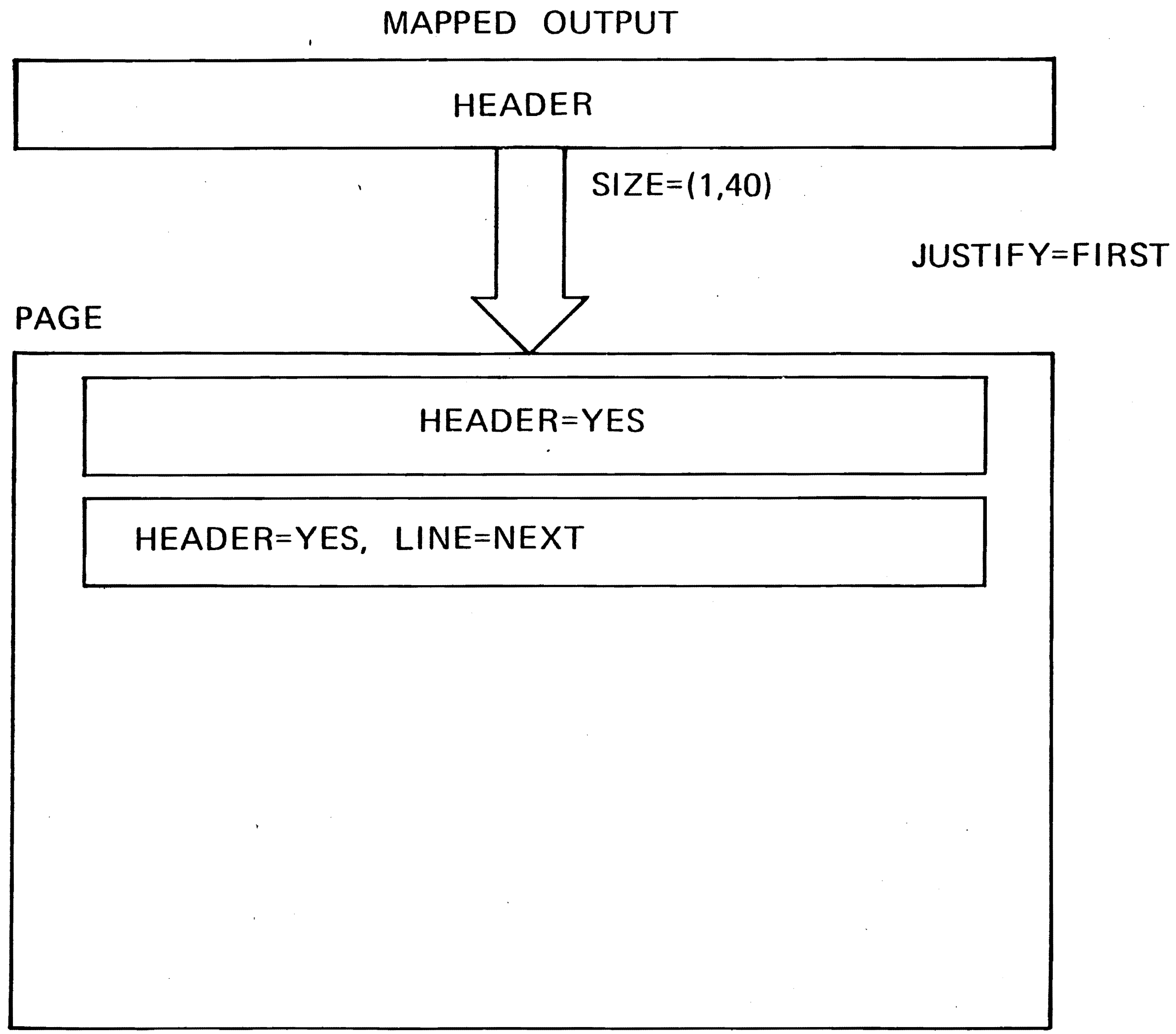
PAGEBLD

```
MAPA    SIZE=(3,20),JUSTIFY=(FIRST,RIGHT)
MAPB    SIZE=(1,10),LINE=SAME,COLUMN=11
MAPC    SIZE=(1,10),LINE=SAME,COLUMN=1
MAPD    SIZE=(2,10),JUSTIFY=RIGHT,COLUMN=21
MAPE    SIZE=(3,20),LINE=SAME
MAPF    SIZE=(1,20),LINE=SAME,COLUMN=NEXT
MAPG    SIZE=(1,20)
MAPH    SIZE=(1,10),JUSTIFY=LAST,COLUMN=11
MAPI    SIZE=(2,10),LINE=11,COLUMN=31
```

PGSIZE=(12,40)



MAP POSITIONING



MAPPED OUTPUT

HEADER

SIZE=(1,40)

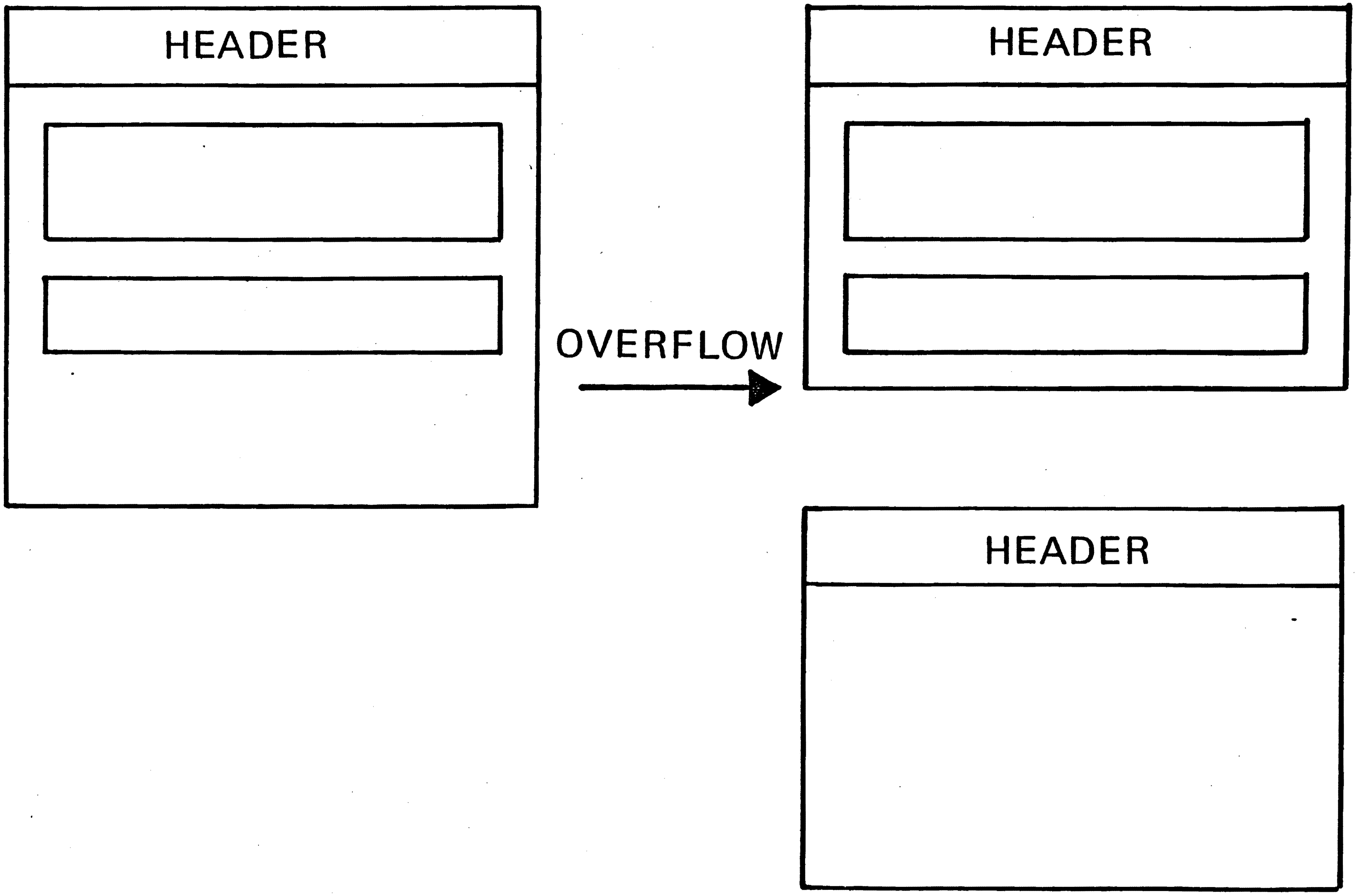
JUSTIFY=FIRST

PAGE

HEADER=YES

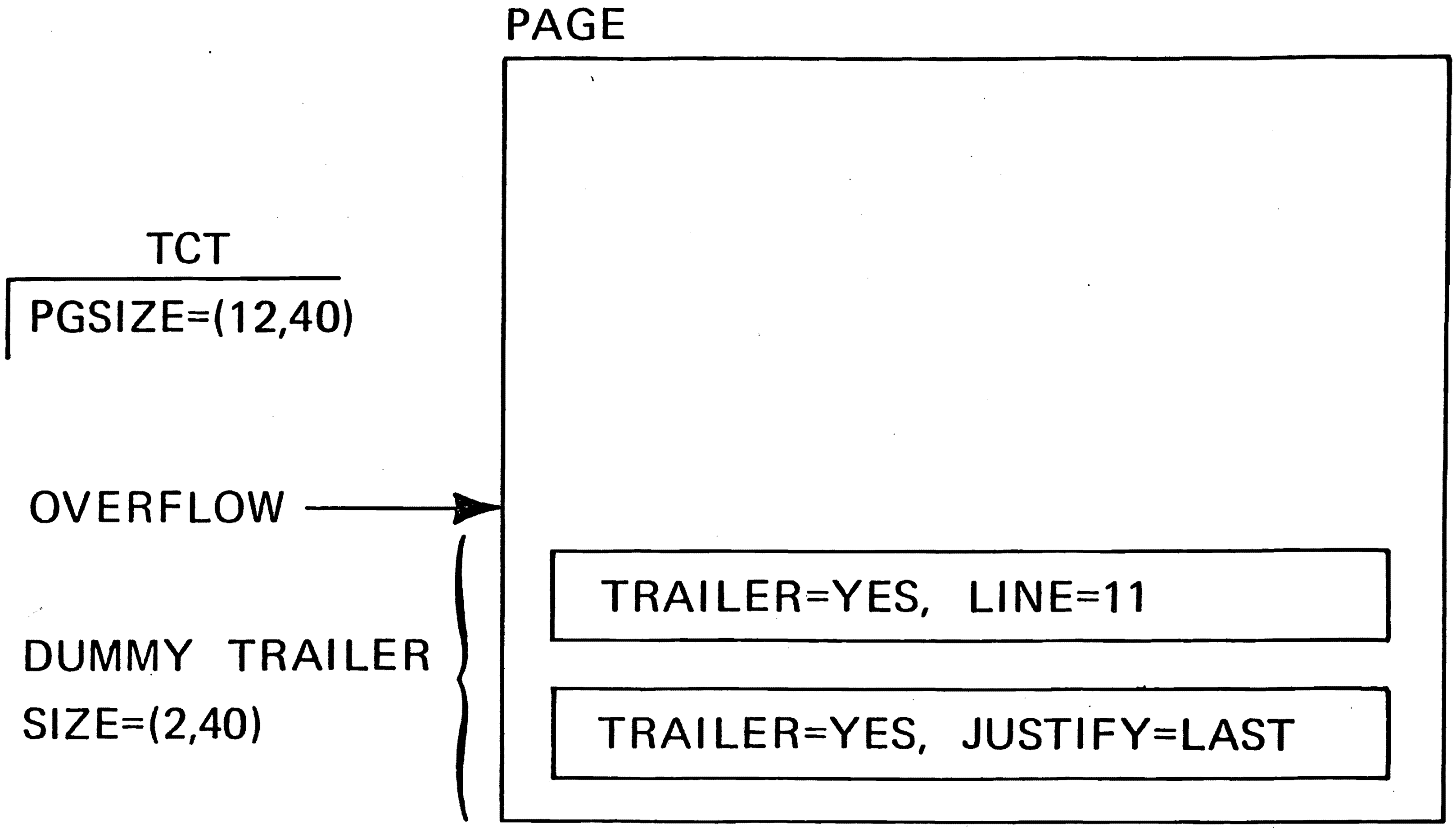
HEADER=YES, LINE=NEXT

OVERFLOW



PAGING

TRAILER



PAGE

TCT

PGSIZE=(12,40)

OVERFLOW

DUMMY TRAILER

SIZE=(2,40)

TRAILER=YES, LINE=11

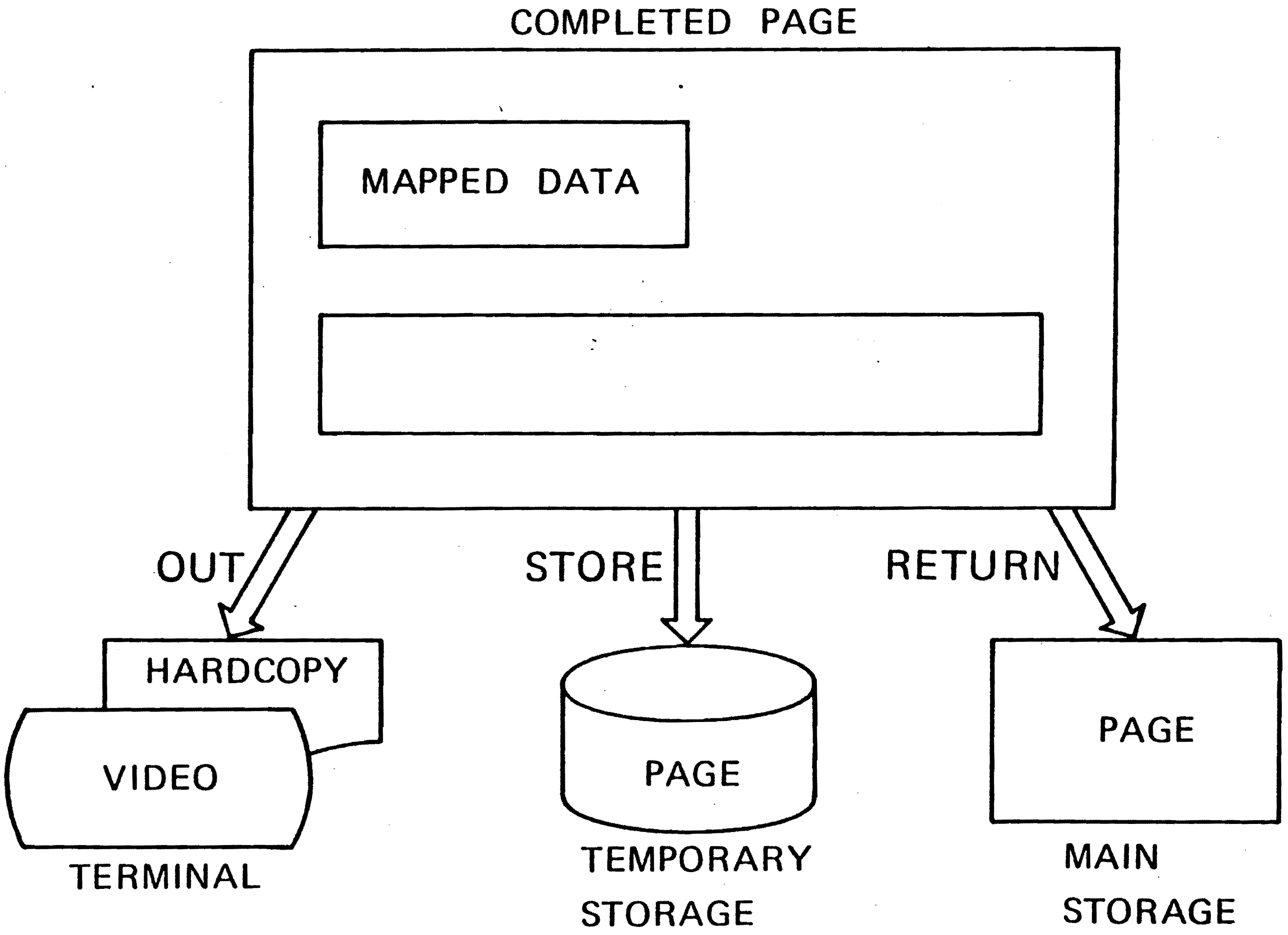
TRAILER=YES, JUSTIFY=LAST

BASIC MAPPING

```
DFHBMS       TYPE=(PAGEBLD ⎡, ⎧ OUT    ⎫⎤ [,SAVE][,ERASE])
                           ⎢  ⎨ STORE  ⎬⎥
                           ⎣  ⎩ RETURN ⎭⎦
             ⎡,DATA= ⎧ NO  ⎫⎤
             ⎣       ⎩ YES ⎭⎦
                      ONLY
              ,MAP= ⎧ map name ⎫
                    ⎩ YES      ⎭
             ⎡,MAPSET= ⎧ mapset name ⎫⎤ | ⎡,MSETADR= ⎧ symbolic address ⎫⎤
             ⎣         ⎩ YES         ⎭⎦   ⎣          ⎩ YES              ⎭⎦
             ⎡,CTRL=( [PRINT] ⎡, ⎧ L40    ⎫⎤[,FREEKB] [,ALARM] [,FRSET])⎤
             ⎢                ⎢  ⎪ L64    ⎪⎥                             ⎥
             ⎢                ⎢  ⎨ L80    ⎬⎥                             ⎥
             ⎣                ⎣  ⎩ HONEOM ⎭⎦                             ⎦
             [,OFLOW=symbolic address]  ([illegible])
             ⎡,CURSOR= ⎧ number ⎫⎤
             ⎣         ⎩ YES    ⎭⎦
             [,WRBRK=symbolic address]
             [,NORESP=symbolic address]
             [,TSIOERR=symbolic adress]
             [,INVREQ=symbolic address]
             [,RETPAGE=symbolic address]
             [,INVMPSZ=symbolic address]
             [,ERROR=symbolic address]
```

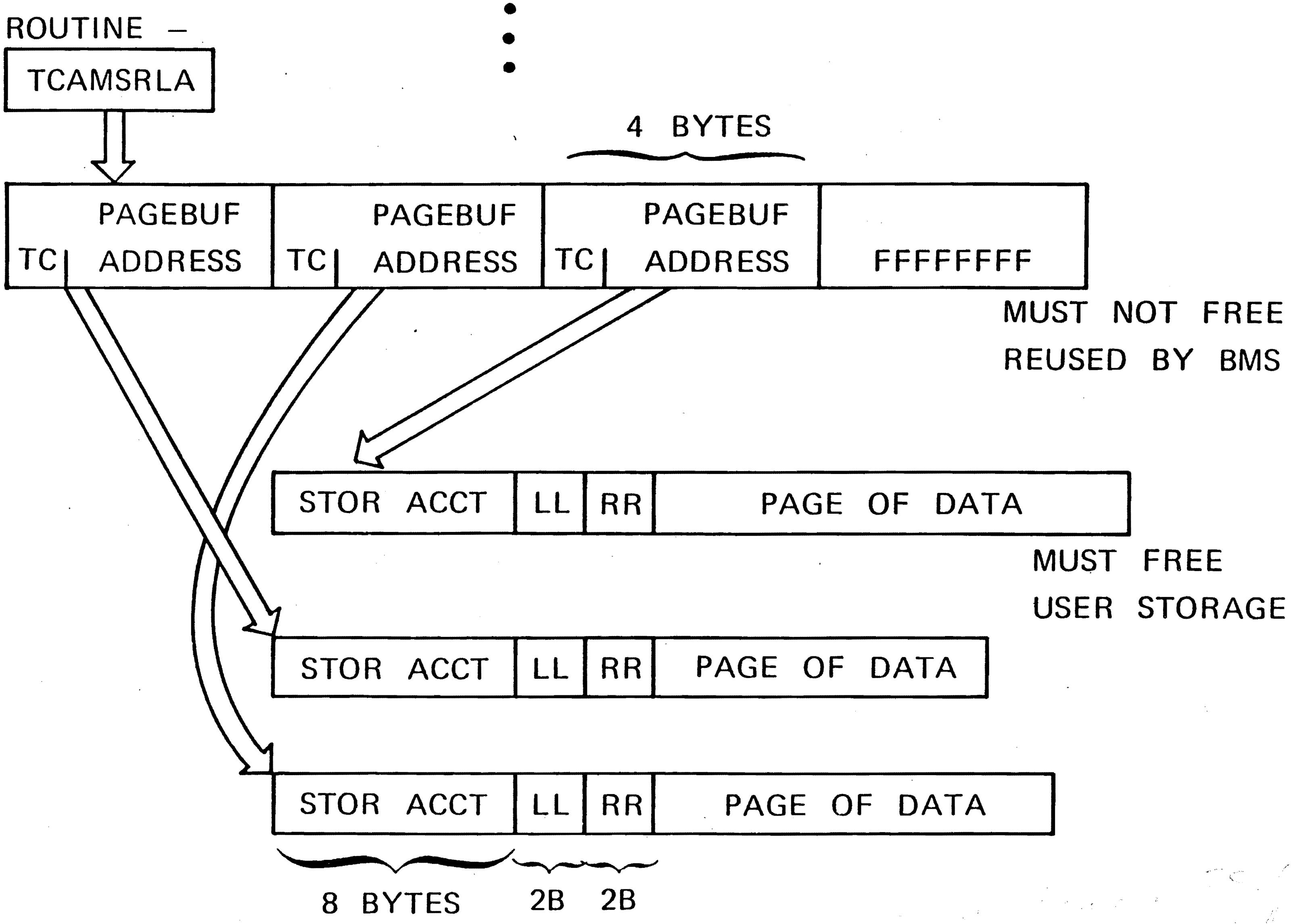
PAGING

DISPOSITION



RETURN

```
DFHBMS    TYPE=RETURN
          RETPAGE=ROUTINE
          •
          •
          •
```



ROUTINE –

TCAMSRLA

4 BYTES

TC PAGEBUF ADDRESS	TC PAGEBUF ADDRESS	TC PAGEBUF ADDRESS	FFFFFFFF

MUST NOT FREE
REUSED BY BMS

STOR ACCT	LL	RR	PAGE OF DATA

MUST FREE
USER STORAGE

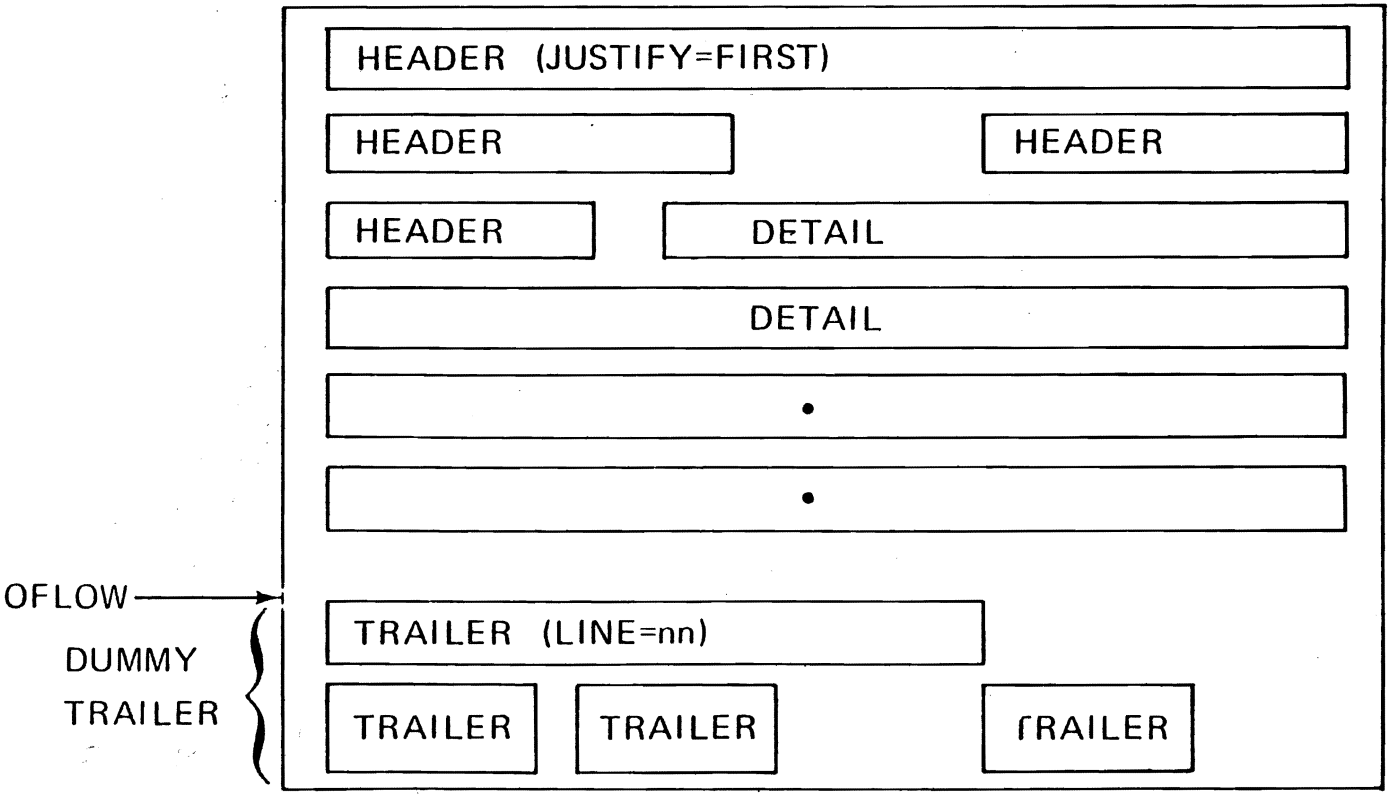
STOR ACCT	LL	RR	PAGE OF DATA

STOR ACCT	LL	RR	PAGE OF DATA
8 BYTES	2B	2B	

OVERFLOW

```
              DFHBMS      TYPE=PAGEBLD...
                          OFLOW=ROUTINE
ROUTINE—
              PAGEBLD  TRAILERS
              PAGEBLD  HEADERS
              PAGEBLD  DETAIL  CAUSING  OFLOW
```

PAGE



HEADER (JUSTIFY=FIRST)

HEADER

HEADER

HEADER

DETAIL

DETAIL

•

•

OFLOW

DUMMY TRAILER

TRAILER (LINE=nn)

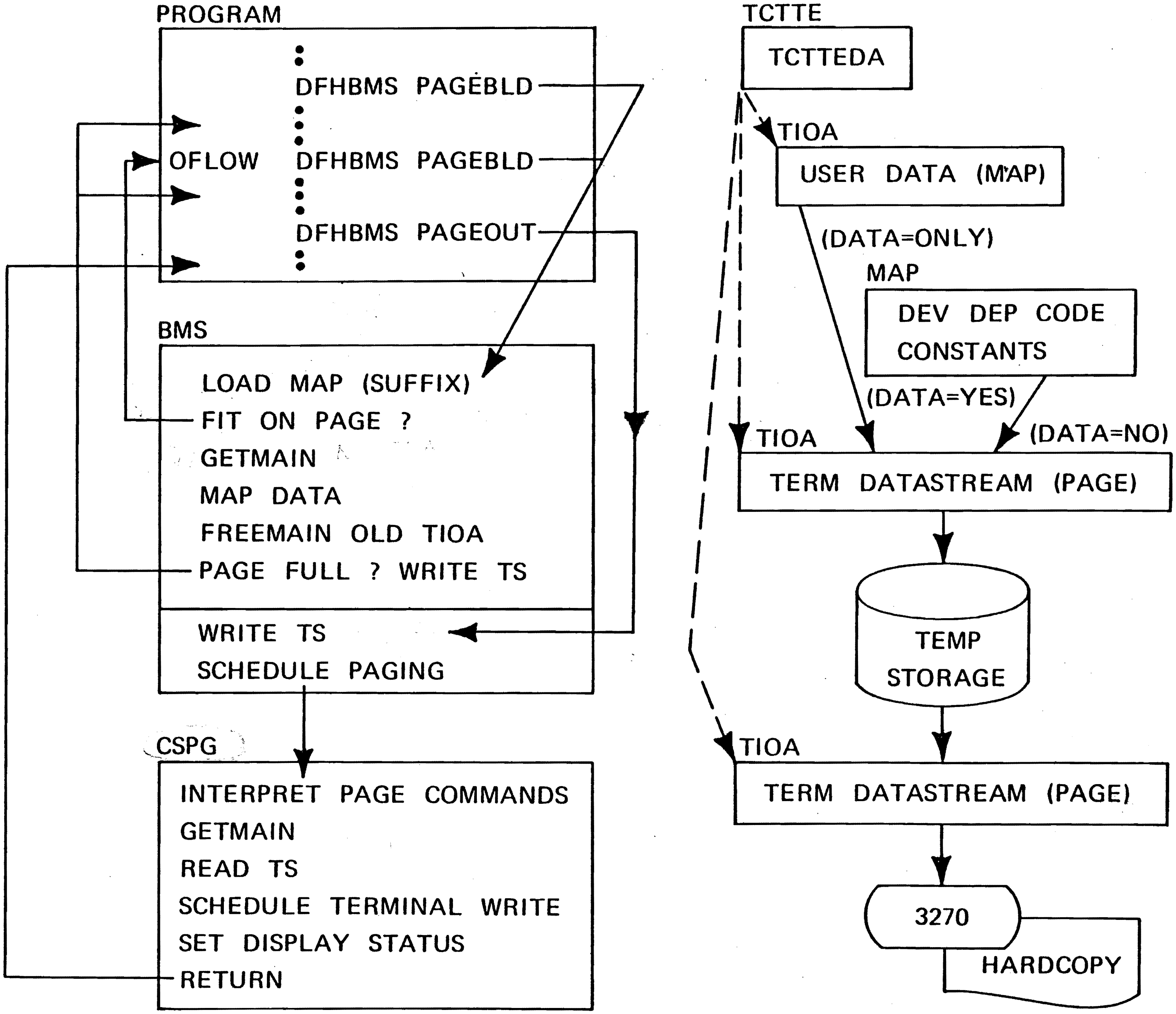
TRAILER

TRAILER

ΓRAILER

PAGE BUILDING

PAGING



PROGRAM

DFHBMS PAGEBLD

OFLOW DFHBMS PAGEBLD

DFHBMS PAGEOUT

BMS

LOAD MAP (SUFFIX)

FIT ON PAGE ?

GETMAIN

MAP DATA

FREEMAIN OLD TIOA

PAGE FULL ? WRITE TS

WRITE TS

SCHEDULE PAGING

CSPG

INTERPRET PAGE COMMANDS

GETMAIN

READ TS

SCHEDULE TERMINAL WRITE

SET DISPLAY STATUS

RETURN

TCTTE

TCTTEDA

TIOA

USER DATA (MAP)

(DATA=ONLY)

MAP

DEV DEP CODE

CONSTANTS

(DATA=YES)

(DATA=NO)

TIOA

TERM DATASTREAM (PAGE)

TEMP STORAGE

TIOA

TERM DATASTREAM (PAGE)

3270

HARDCOPY

BASIC MAPPING

```
DFHBMS      TYPE=PAGEOUT
            [,CTRL= ( [ ( PAGE     ) ]   [, ( RETAIN  ) ] ) ]
            [         [ { AUTOPAGE } ]   [  { RELEASE } ]   ]
            [,TRAILER= ( symbolic address ) ]
            [          { YES              } ]
            [,TRANSID=transaction code]
            [,WRBRK= ( symbolic address ) ]
            [        { CURRENT          } ]
            [        ( ALL              ) ]
            [,EODPURG= ( AUTO ) ]
            [          { OPER } ]
            [,NORESP=symbolic address]
            [,TSIOERR=symbolic address]
            [,RETPAGE=symbolic address]
            [,ERROR=symbolic address]
```

PAGE STATUS

PAUSE AFTER EACH PAGE

PAGING COMMANDS FOR OTHER PAGES

REQUIRED STATUS FOR VIDEO DEVICES

(OPTIONAL FOR HARDCOPY)

AUTOPAGE STATUS

NO PAUSE BETWEEN PAGES TO ACCEPT PAGING COMMANDS

NORMAL STATUS FOR 3284/3286 PRINTERS

RECOMMENDED STATUS FOR HARDCOPY

DFHBMS TYPE=PAGEOUT,CTRL=RETAIN

- STORES LAST PAGE
- PAGES WRITTEN TO TERMINAL
- RETURNS TO TRANSACTION
  - AT PAGING TERMINATION
  - UPON INPUT OF NON-PAGE COMMAND (DATA)
- TRANSACTION CONTINUES

DFHBMS TYPE=PAGEOUT,CTRL=RELEASE

- STORES LAST PAGE
- PAGE WRITTEN TO TERMINAL
- RETURNS TO CICS AFTER FIRST PAGE WRITTEN
- PAGING COMMANDS INITIATE NEW TASK

DFHBMS TYPE=PAGEOUT

- STORES LAST PAGE
- RETURNS TO TRANSACTION – – –
- PAGES WRITTEN TO TERMINAL WHEN TERMINAL AVAILABLE

ONLY WITH STORE

ONLY WITH DIRECT (NON-ROUTE) TERMINAL

2741 WRITE BREAK SUPPORT – ACTION ON 2741 ATTN KEY

WRBRK=symbadr (TYPE=OUT)

CONTROL IS PASSED TO symbadr IF . . .

TRANSACTION ACTIVE – DIRECT TERMINAL INVOLVED

PAGEOUT (CTRL=RETAIN)

WRBRK=CURRENT (TYPE=STORE)

CURRENT PAGE CEASES PRINTING

NEXT PAGE BEGINS IF AUTOPAGE

WRBRK=ALL (TYPE=STORE)

CURRENT PAGE CEASES PRINTING

NO ADDITIONAL PAGES ARE SENT

REMAINING PAGES ARE PURGED

BASIC MAPPING

DFHBMS TYPE=PURGE

CSPG COMMANDS

RETRIEVAL –

P/X
↓

N	–	PAGE NUMBER N
+N	–	N PAGES FORWARD
–N	–	N PAGES BACKWARD
C	–	CURRENT PAGE
N	–	NEXT PAGE
P	–	PREVIOUS PAGE
L	–	LAST PAGE
A	–	ALL REMAINING PAGES (AUTOPAGE)
Q	–	QUERY TITLE OF REMAINING MESSAGES AND ID

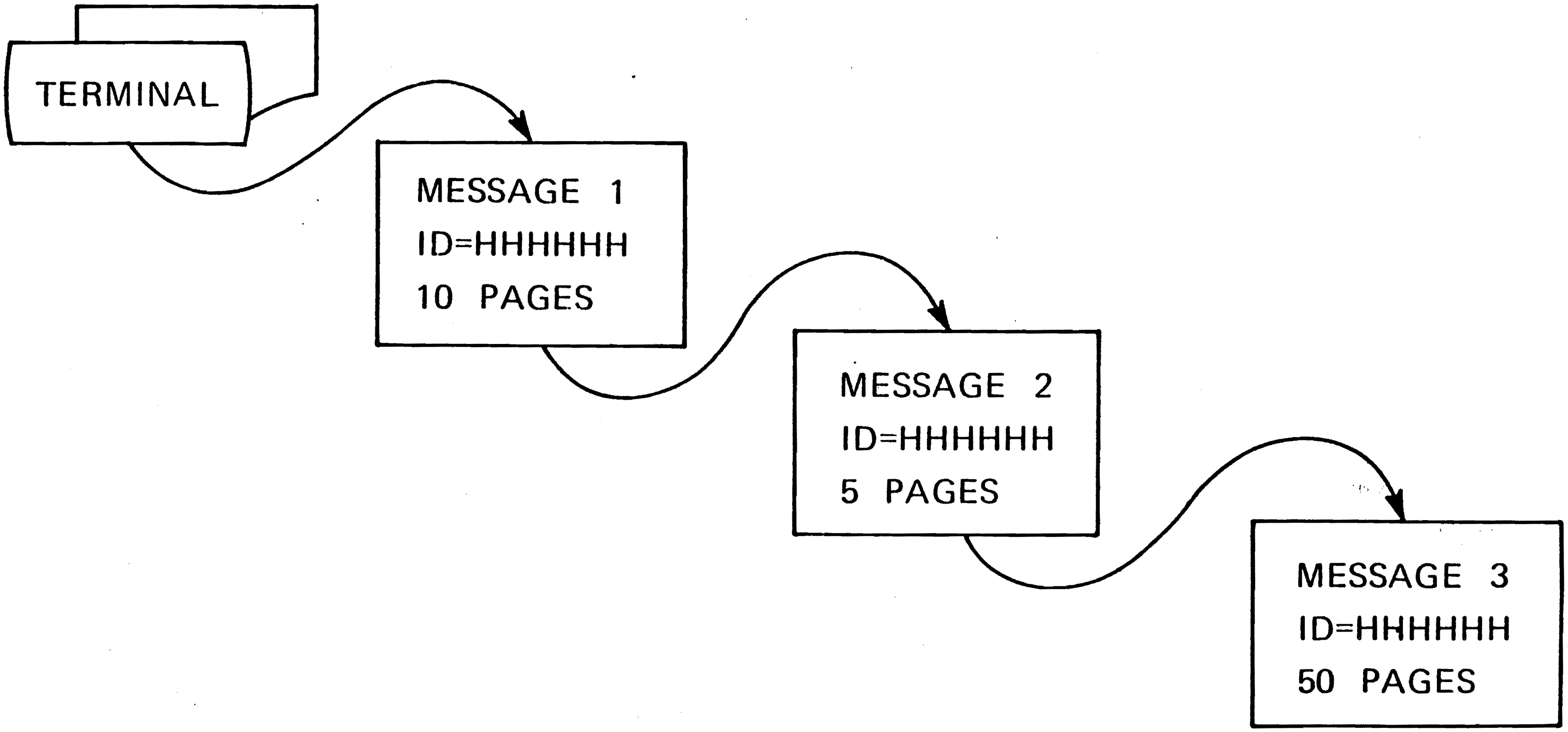
CSPG COMMANDS

CHAINING –

L/XXXX

↓

XXXX – TRANSID WHOSE PAGED OUTPUT SHOULD BE ADDED TO LOGICAL MESSAGES ALREADY CHAINED TO TERMINAL



TERMINAL

MESSAGE 1
ID=HHHHHH
10 PAGES

MESSAGE 2
ID=HHHHHH
5 PAGES

MESSAGE 3
ID=HHHHHH
50 PAGES

P/(Y)X (LOGICAL MESSAGES CHAINED TOGETHER)

N	–	LOGICAL MESSAGE N OF CHAIN
C	–	CURRENT LOGICAL MESSAGE
N	–	NEXT LOGICAL MESSAGE
P	–	PREVIOUS LOGICAL MESSAGE
L	–	LAST LOGICAL MESSAGE

P/X,HHHHHH

HHHHHH – LOGICAL MESSAGE ID (3 BYTE HEXADECIMAL)

CSPG COMMANDS

COPY –

C/XXXX

↓

XXXX – TERMINAL ID TO RECEIVE PAGE CURRENTLY DISPLAYED AS IT WAS ORIGINALLY SENT TO TERMINAL

REQUIRES ROUTING FACILITY TO BE GENERATED IN BMS

CSPG COMMANDS

TERMINATION –

T/X

↓

C – CURRENT LOGICAL MESSAGE

B – BASE LOGICAL MESSAGE AND ALL CHAINED TO IT

H – HIGHER LOGICAL MESSAGES CHAINED TO BASE, BUT NOT BASE

R – ROUTED LOGICAL MESSAGES SCHEDULED FOR IMMEDIATE DELIVERY

A – ALL LOGICAL MESSAGES (COMBINATION OF B AND R)

T/X,HHHHHH

↓

HHHHHH – LOGICAL MESSAGE HHHHHH IS PURGED
CURRENT MESSAGE IS REDISPLAYED TO OPERATOR

TEMPORARY STORAGE MANAGEMENT

- STORAGE AND RETRIEVAL OF DATA
  - DATA IDENTIFICATION
  - SEQUENTIAL OR DIRECT
- DATA TRANSFER BETWEEN TASKS
- TERMINAL PAGING
- MESSAGE ROUTING

TEMPORARY STORAGE

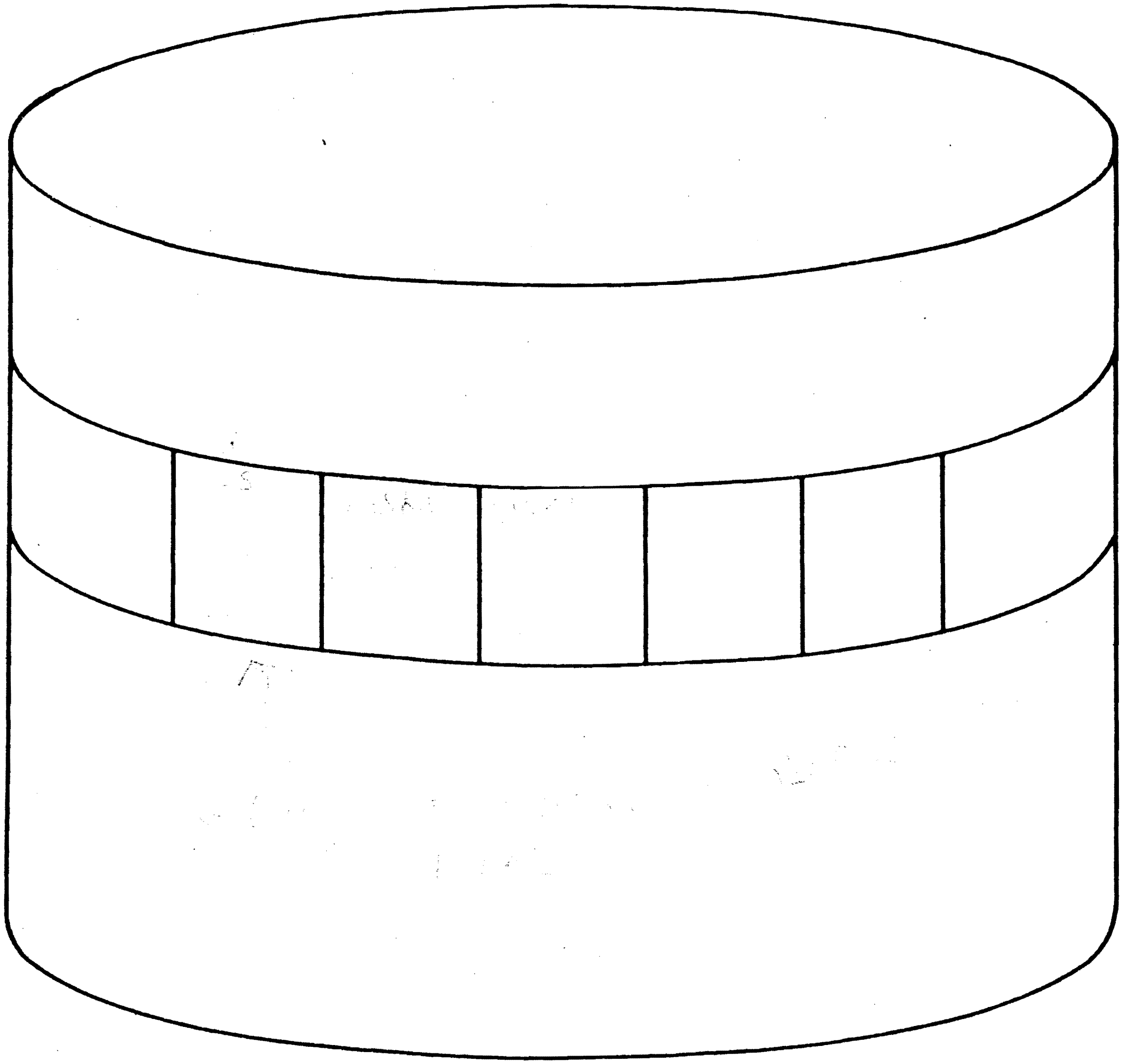
MAIN STORAGE

SHORT TERM STORAGE
SMALL AMOUNTS OF DATA
SHARED SUBPOOL
MAXIMUM RECORD – 32 K BYTES

AUXILIARY STORAGE

VSAM – ENTRY SEQUENCED DATA SET
LONGER TERM STORAGE
LARGER AREA REQUIRED
RECOVERABLE ON WARM START
MAXIMUM RECORD – CONTROL INTERVAL SIZE

TEMPORARY STORAGE

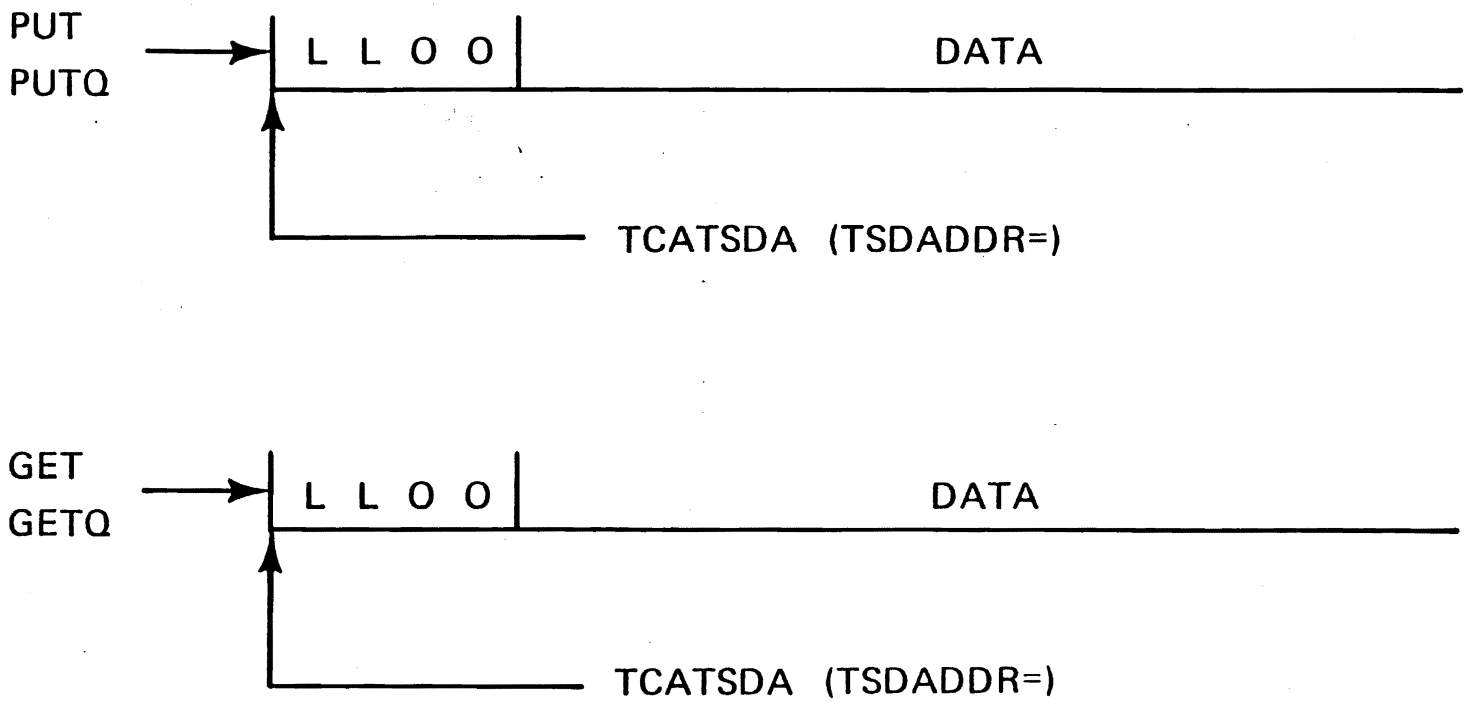


TEMPORARY STORAGE

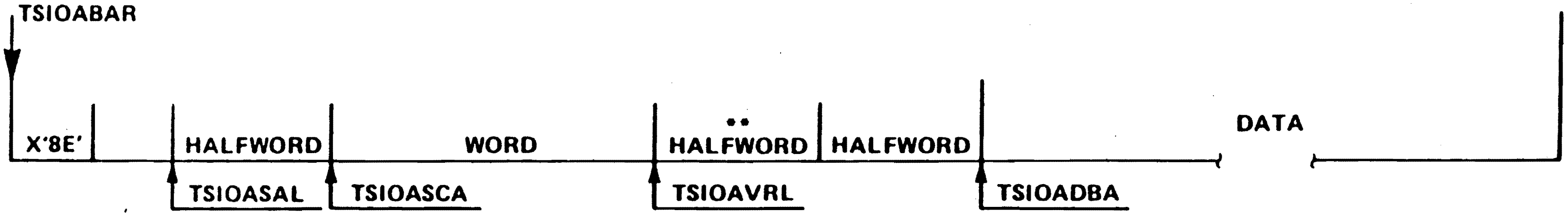
```
DFHTS          TYPE=PUT
               [,DATAID=name]
               [,TSDADDR= { symbolic address } ]
               [          { YES              } ]
               [,STORFAC= { AUXILIARY        } ]
               [          { MAIN             } ]
               [,COND=YES]
               [,NOSPACE=symbolic address]
               [,NORESP=symbolic address]
               [,IOERROR=symbolic address]
               [,INVREQ=symbolic address]
               [,ERROR=symbolic address]
```

In the STORFAC option, AUXILIARY is underlined.

```
DFHTS          TYPE=PUT,DATAID=ABC12,ERROR=CHECK
```


PUT
PUTQ
L L O O
DATA
TCATSDA (TSDADDR=)
GET
GETQ
L L O O
DATA
TCATSDA (TSDADDR=)

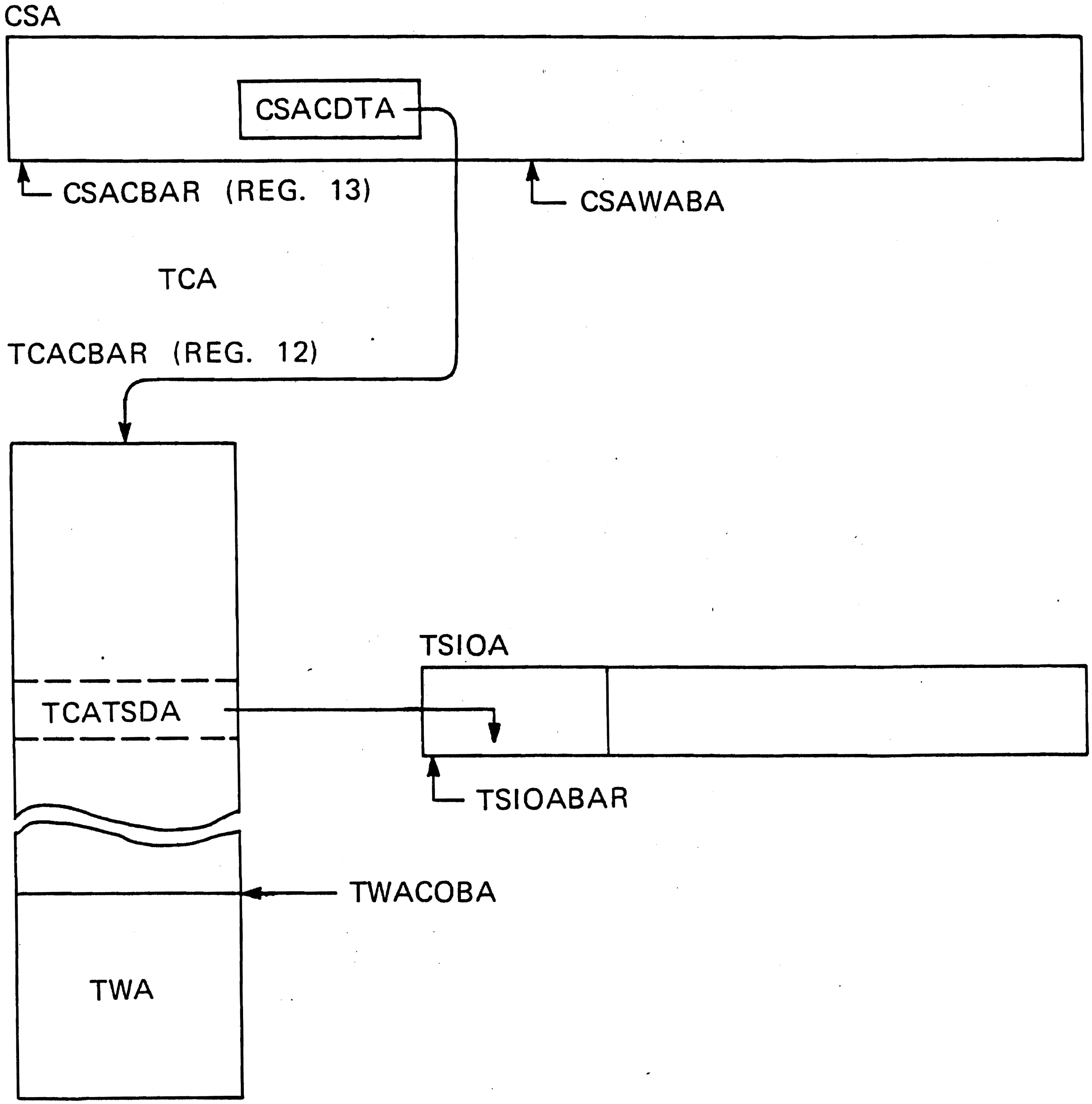
TSIOA Temporary Storage Input/Output Area (DFHTSIOA)



TSIOABAR – TSIOA Base Address Register
TSIOADBA – TSIOA Data Begin Address
TSIOASAL – TSIOA Storage Accounting – area Length

TSIOASCA – TSIOA Storage Chain Address
TSIOAVRL – TSIOA Variable Record Length (LLbb)**

TEMPORARY STORAGE



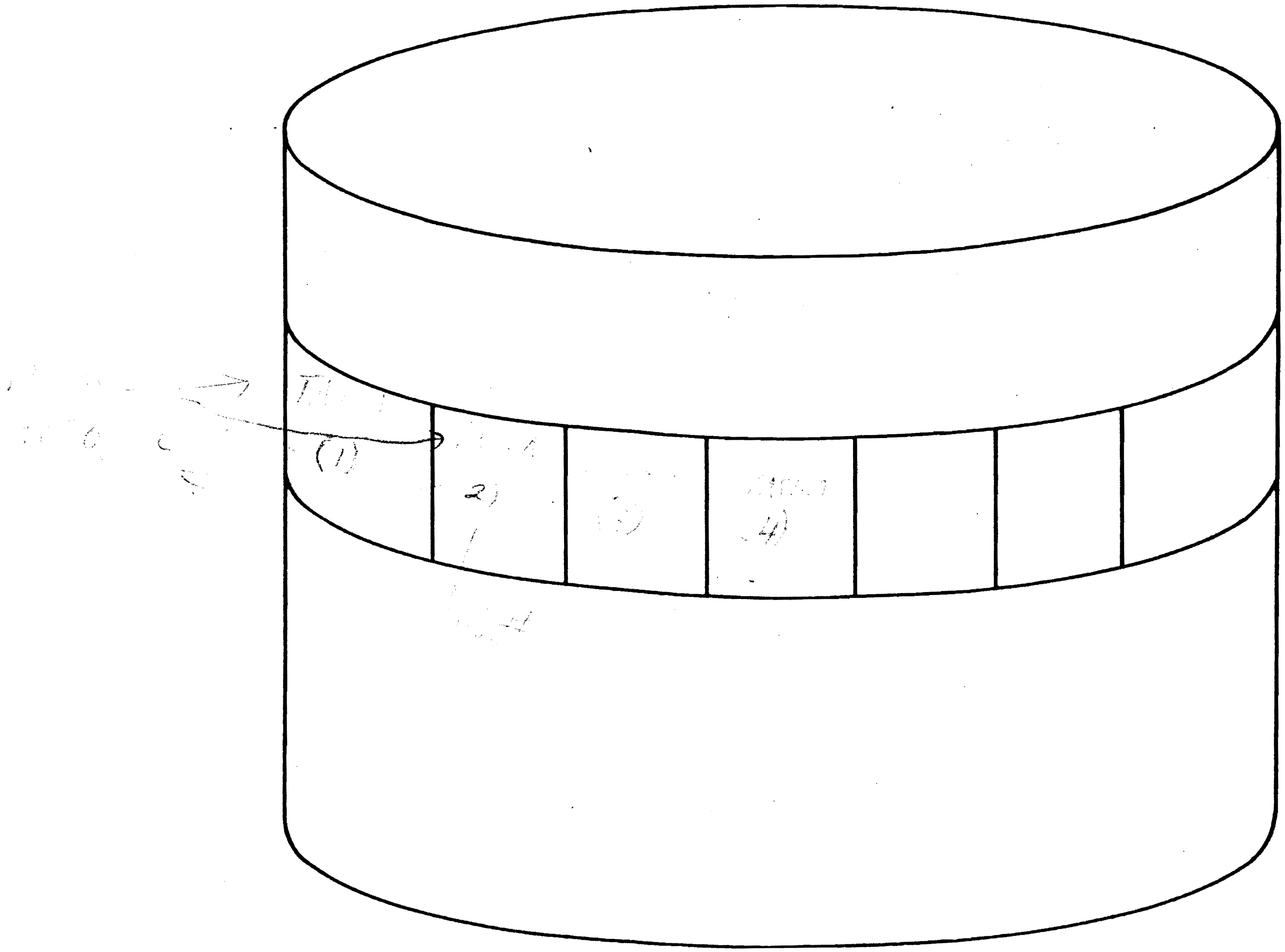
TEMPORARY STORAGE

```
DFHTS          TYPE=GET
               [,DATAID=name]
               [,TSDADDR= {symbolic address} ]
               [          {YES             } ]
               [,RELEASE={YES} ]
               [         {NO } ]
               [,NORESP=symbolic address]
               [,IDERROR=symbolic address]
               [,IOERROR=symbolic address]
               [,INVREQ=symbolic address]
               [,ERROR=symbolic address]


DFHTS          TYPE=GET,DATAID=ABC12,NORESP=GO
```

TEMPORARY STORAGE

```
DFHTS          TYPE=RELEASE
               [,DATAID=name]
               [,NORESP=symbolic address]
               [,IDERROR=symbolic address]
               [,INVREQ=symbolic address]
               [,ERROR=symbolic address]


DFHTS          TYPE=RELEASE, DATAID=ABC12
```

TEMPORARY STORAGE



TEMPORARY STORAGE

```
DFHTS        TYPE=PUTQ
            [,DATAID=name]
            [,TSDADDR= {symbolic address}]
            [          {YES             }]
            [,STORFAC= {AUXILIARY}]
            [          {MAIN     }]
            [,COND=YES]
            [,NOSPACE=symbolic address]
            [,NORESP=symbolic address]
            [,IOERROR=symbolic address]
            [,INVREQ=symbolic address]
            [,ERROR=symbolic address]
```

```
DFHTS        TYPE=PUTQ,DATAID=ABC
```

TEMPORARY STORAGE

```
DFHTS          TYPE=GETQ
               [,DATAID=name]
               [,TSDADDR= { symbolic address }]
               [          { YES              }]
               [,ENTRY= { n   }]
               [        { YES }]
               [,NORESP=symbolic address]
               [,IDERROR=symbolic address
               [,IOERROR=symbolic address]
               [,INVREQ=symbolic address]
               [,ENERROR=symbolic address]
               [,ERROR=symbolic address]
```

```
DFHTS          TYPE=GETQ,DATAID=ABC,ENTRY=15,          X
               ENERROR=NONE,NORESP=GO
```

TEMPØRARY STØRAGE

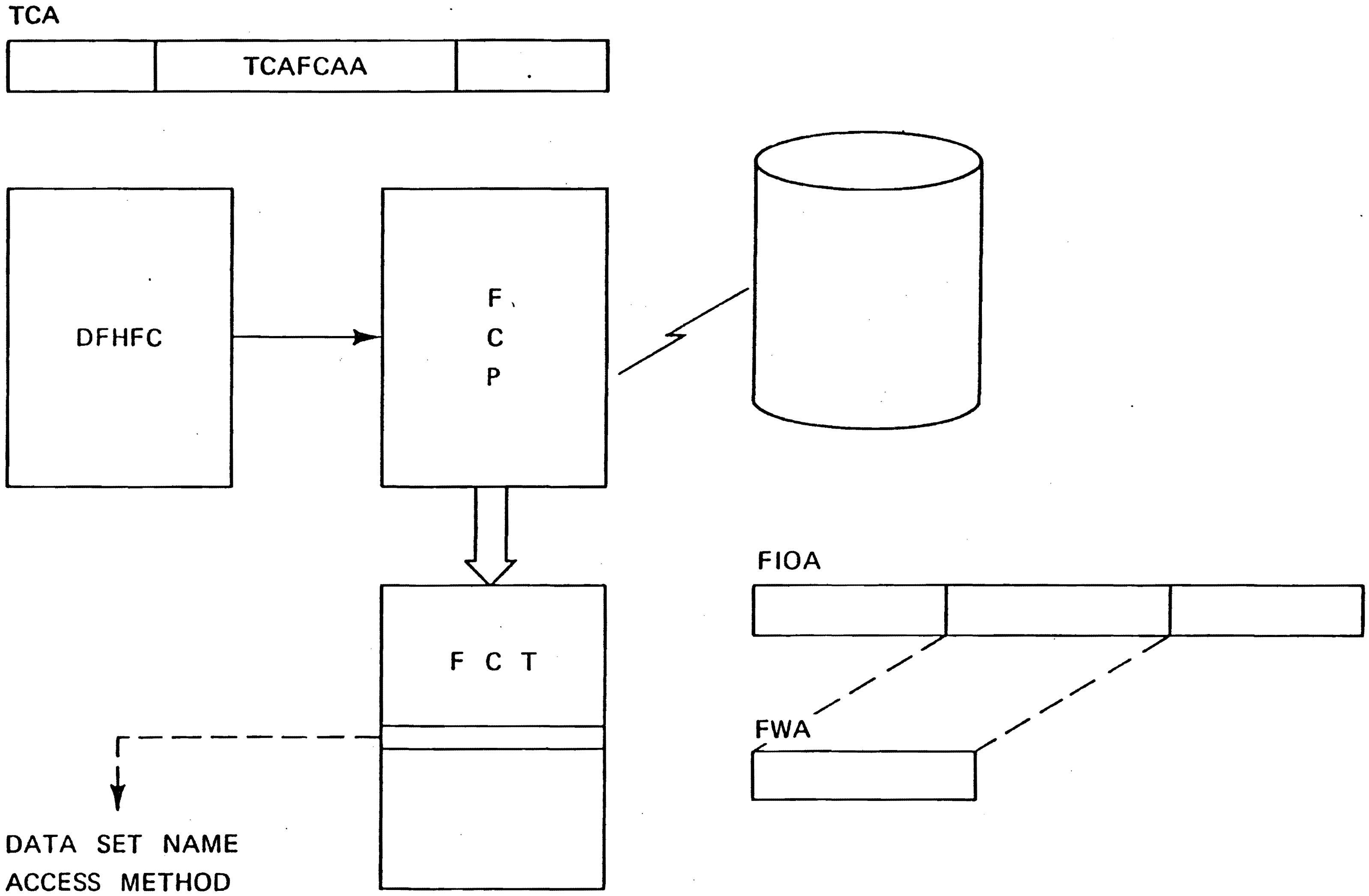
```
DFHTS          TYPE=PURGE
              [,DATAID=name]
              [,NORESP=symbolic address]
              [,IDERROR=symbolic address]
              [,INVREQ=symbolic address]
              [,ERROR=symbolic address]


DFHTS          TYPE=PURGE,DATAID=ABC
```

TEMPORARY STORAGE

```
DFHTS        TYPE=CHECK
            [,NOSPACE=symbolic address]
            [,NORESP=symbolic address]
            [,IDERROR=symbolic address]
            [,IOERROR=symbolic address]
            [,INVREQ=symbolic address]
            [,ENERROR=symbolic address]
            [,ERROR=symbolic address]


DFHTS        TYPE=CHECK,IDERROR=INVID,ENERROR=NONE
```

DATA BASE ACCESS

- STANDARD ACCESS METHODS
- EXCLUSIVE CONTROL
- INDIRECT ACCESS ([illegible])
- FILE BROWSE (Sequ[illegible])
- DL/I INTERFACE
- SEGMENTED RECORDS [illegible]



DATA SET NAME
ACCESS METHOD
DEVICE TYPE
SERVICES ALLOWED
RECORD – LENGTH, FORMAT, BLOCKING
KEY – LENGTH, POSITION

FILE I/O AREA - FILE WORK AREA

FILE I/O AREA (DFHFIOA)

READ ONLY

UNBLOCKED

UNSEGMENTED

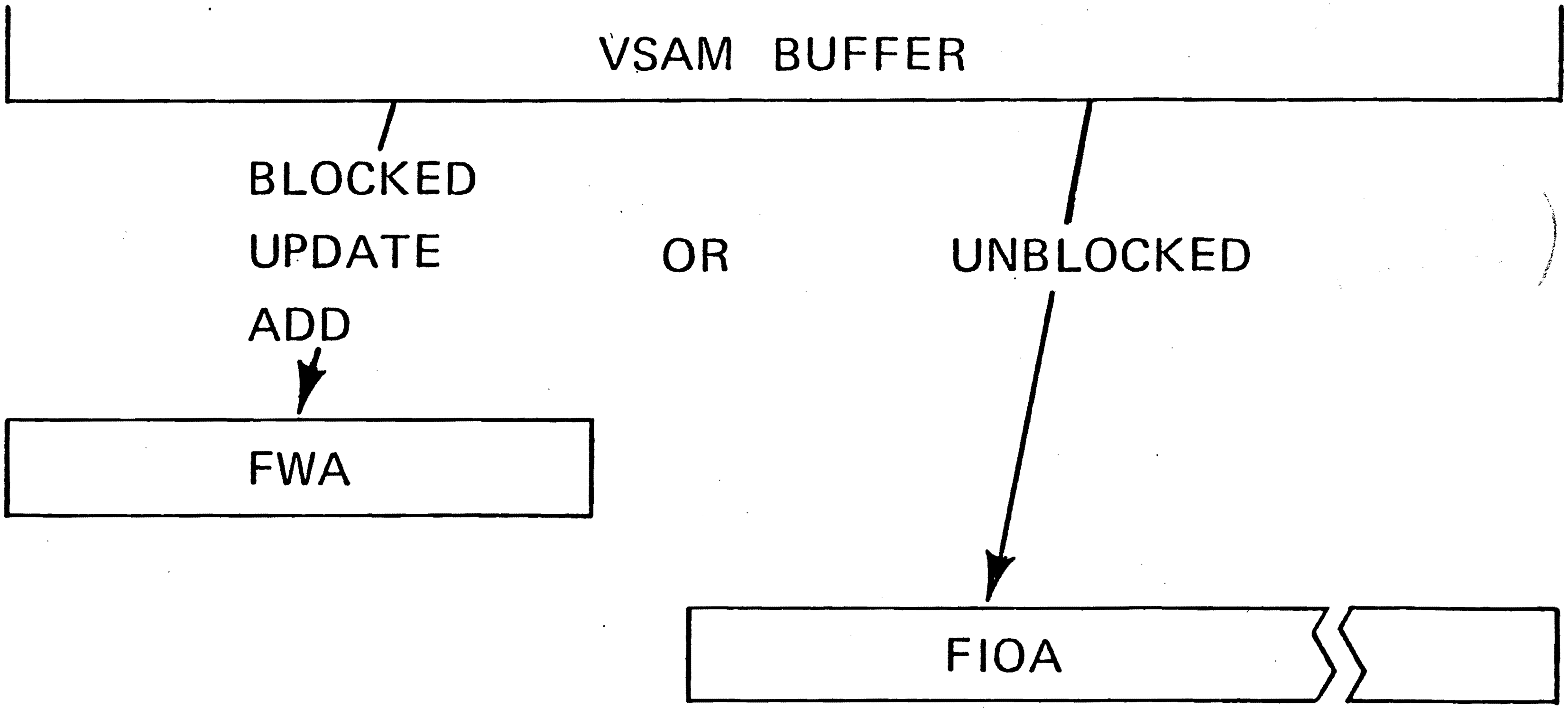
FILE WORK AREA (DFHFWADS)

UPDATE

ADDITIONS (NEW)

BLOCKED

VSAM (MOVE MODE)



VSAM BUFFER

BLOCKED
UPDATE
ADD

OR

UNBLOCKED

FWA

FIOA

RECORD IDENTIFICATION ______ RDIDADR

USER PROVIDED AREA

TWA RECOMMENDED

ISAM – RECORD KEY

VSAM – RECORD KEY

RELATIVE BYTE ADDRESS

GENERIC KEY WITH EXPLICIT LENGTH

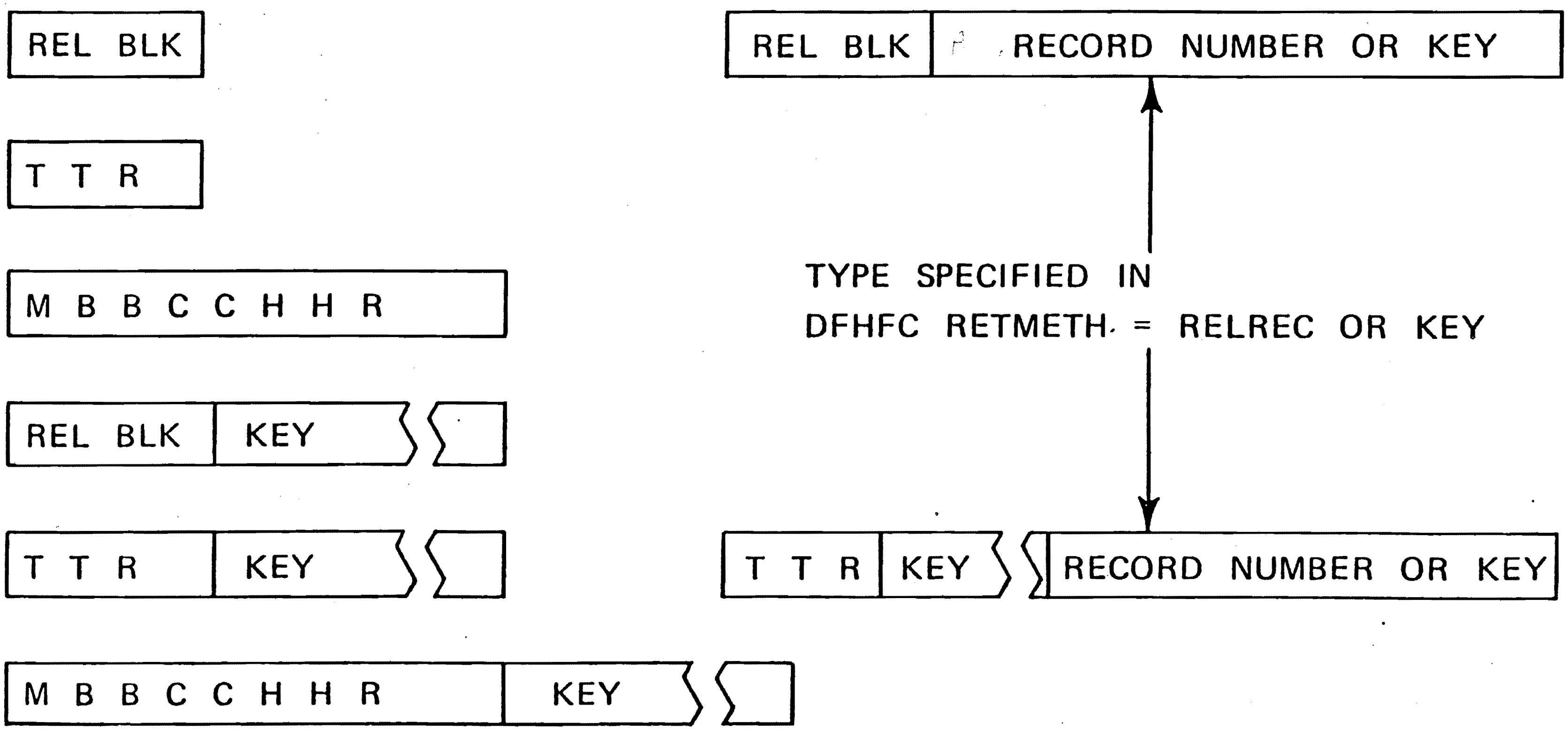
L	KEY

DAM RECORD IDENTIFICATION________RDIDADR

- UNBLOCKED – PHYSICAL RECORD SEARCH ARGUMENT
- BLOCKED – PHYSICAL RECORD SEARCH ARGUMENT
  CICS/VS DEBLOCKING ARGUMENT

PHYSICAL RECORD SEARCH

DEBLOCKING



REL BLK

T T R

M B B C C H H R

REL BLK | KEY

T T R | KEY

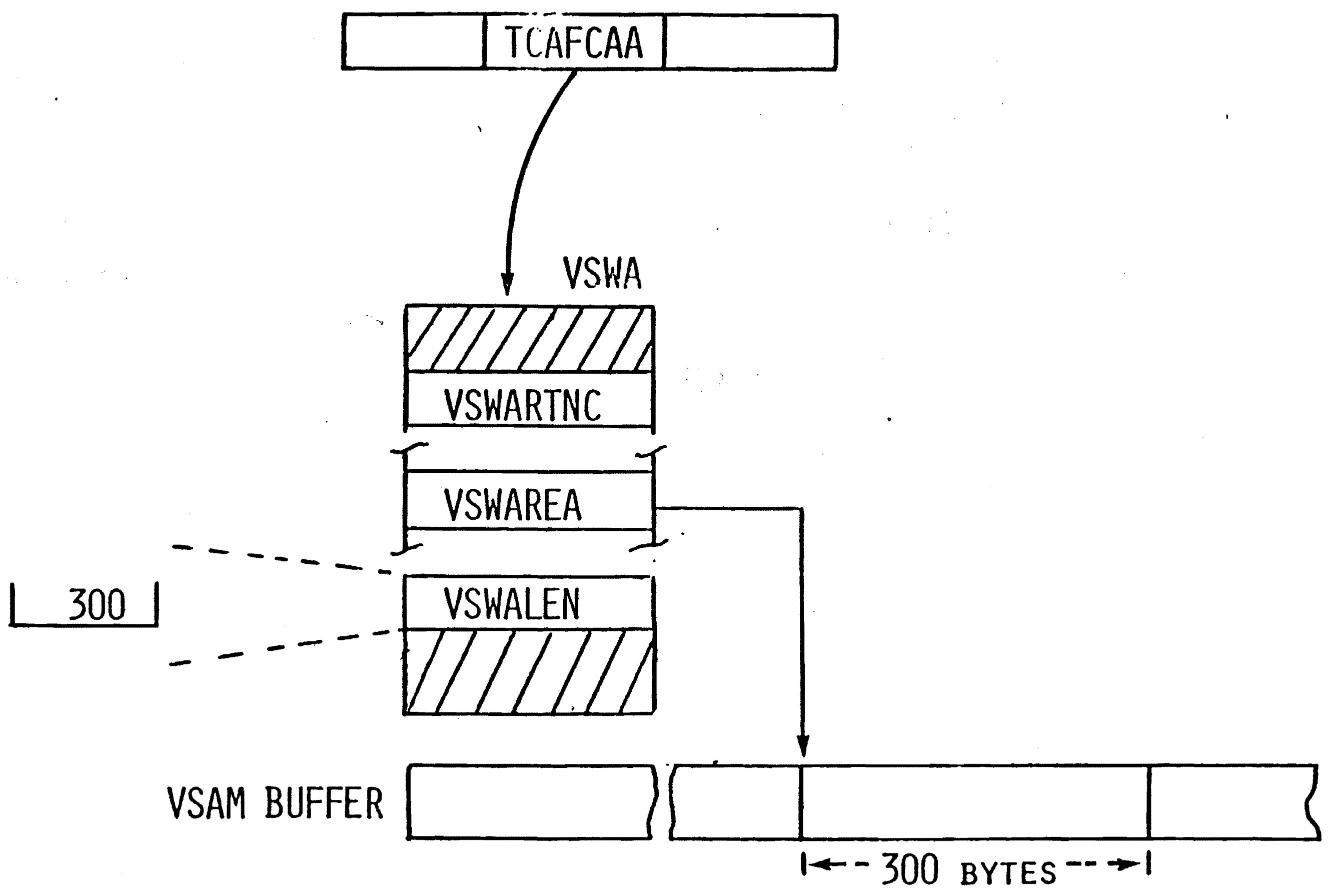
M B B C C H H R | KEY

REL BLK | RECORD NUMBER OR KEY

TYPE SPECIFIED IN
DFHFC RETMETH = RELREC OR KEY

T T R | KEY | RECORD NUMBER OR KEY

VSAM LOCATE MODE



LOCATE MODE

```
          COPY     DFHCSADS
          COPY     DFHTCADS
KEYF      DS       CL8
          ---
VSWABAR   EQU      7
          COPY     DFHVSWA
          ---
RECDBAR   EQU      8
          DSECT
          USING    *,RECDBAR
RECORD    DS       CL80
          ---
          ---
          MVC      KEYF,ACCTNO
READREC   DFHFC    TYPE=GET,DATASET=VSMSTR,RDIDADR=KEYF,        X
                   MODE=LOCATE
          L        VSWABAR,TCAFCAA
          L        RECDBAR,VSWAREA
          L        3,VSWALEN
PROCESS   ---
```

LOCATE MODE

```
01 DFHBLLDS COPY DFHBLLDS.
   02 ---
   02 VSWABAR PIC S9(8) COMP.
   02 RECDBAR PIC S9(8) COMP.
01 DFHCSADS COPY DFHCSADS.
01 DFHTCADS COPY DFHTCADS.
   02 KEYF PIC X(8).
   02 RECD-LENGTH PIC S9(8) COMP.
01 ---
01 DFHVSWA COPY DFHVSWA.
01 RECORD-DESCRIPTION.
   02 RECORD PIC X(80).
PROCEDURE DIVISION.
      ---
      MOVE ACCTNO TO KEYF.
READ-REC.
      DFHFC    TYPE=GET,DATASET=VSMSTR,RDIDADR=KEYF,        X
               MODE=LOCATE
      MOVE TCAFCAA TO VSWABAR.
      MOVE VSWAREA TO RECDBAR.
      MOVE VSWALEN TO RECD-LENGTH.
PROCESS.
      ---
```

LOCATE MODE

```
%INCLUDE (DFHCSADS);
%INCLUDE (DFHTCADS);
 2 KEYF CHAR (8),
 2 RECD_LEN FIXED BINARY (31);
%INCLUDE (DFHVSWA);
---
DECLARE 1 RECDESC BASED (RECDBAR),
 2 RECD_DESC CHAR (80);
---
READREC:
 KEYF=ACCTNO;
      DFHFC   TYPE=GET,DATASET=VSMSTR,RDIDADR=KEYF,                 X
              MODE=LOCATE
 VSWABAR=TCAFCAA;
 RECDBAR=VSWAREA;
 RECD_LEN=VSWALEN;
PROCESS:
 ---
```

FILE BROWSE

- COMPARABLE TO FILE SEARCH
- SEQUENTIAL RETRIEVAL
- CONCURRENT BROWSE
    - MULTIPLE DATA SETS
    - MULTIPLE TASKS
- ISAM - DAM - VSAM
    - VSAM BROWSE
        - MOVE OR LOCATE MODE
        - SKIP SEQUENTIAL PROCESSING

FILE BROWSE

SPECIFY STARTING POINT FOR SEARCH

```
DFHFC  TYPE=SETL,DATASET=MSTR,RDIDADR=KEYFLD
```

RETREIVE RECORDS

```
DFHFC  TYPE=GETNEXT          (RDIDADR)
```

TERMINATE RETRIEVAL

```
DFHFC  TYPE=ESETL
```

TO CHANGE STARTING POINT <u>DURING</u> BROWSE

```
DFHFC  TYPE=RESETL
```

OR

VSAM → MODIFY RDIDADR

RECORD IDENTIFICATION RDIDADR

ISAM - GENERIC KEY

PARTIAL KEY PADDED WITH BINARY ZEROS

KEY , 00000

DAM - BLOCK REFERENCE

PHYSICAL RECORD SEARCH ARGUMENT

TTR MBBCCHHR

VSAM - RELATIVE BYTE ADDRESS OR GENERIC KEY

RBA L,KEY

GENERIC KEY

- SETL (ABC)

	ISAM	VSAM
RDIDADR	C1 C2 C3 00 00 00	03 C1 C2 C3

- GETNEXT

	ISAM	VSAM
RDIDADR	C1 C2 C3 F1 F2 F3	C1 C2 C3 F1 F2 F3

- RECORD ABC123 IN FWA

ISAM	VSAM
LLBB | SMITH,JP	LLBB | SMITH,JP

FILE CONTROL

```
DFHFC         TYPE=SETL
              [,DATASET=symbolic name]
              [,RDIDADR=symbolic address]
              ┌,SEGSET= ⎧ symbolic name ⎫┐
              │         ⎨ YES           ⎬│
              └         ⎩ ALL           ⎭┘
              ┌,RETMETH= ⎧ RELREC ⎫┐ <─────────── DAM
              └          ⎩ KEY    ⎭┘       ⎫
              ┌,ARGTYP= ⎧ KEY ⎫┐            ⎪
              │         ⎪ ─── ⎪│            ⎪
              └         ⎩ RBA ⎭┘            ⎪
              ┌,SRCHTYP= ⎧ FKEQ ⎫┐          ⎪
              │          ⎪ FKGE ⎪│          ⎬ VSAM
              │          ⎨ ──── ⎬│          ⎪
              │          ⎪ GKEQ ⎪│          ⎪
              └          ⎩ GKGE ⎭┘          ⎪
              ┌,MODE ⎧ MOVE   ⎫┐            ⎪
              │      ⎪ ────   ⎪│            ⎪
              └      ⎩ LOCATE ⎭┘            ⎭
              [,NORESP=symbolic address]
              [,ERROR=symbolic address]
              [,DSIDER=symbolic address]
              [,SEGIDER=symbolic address]
              [,NOTFND=symbolic address]
              [,INVREQ=symbolic address]
              [,IOERROR=symbolic address]
              [,NOTOPEN=symbolic address]
              [,ILLOGIC=symbolic address] <─────── VSAM
```

FILE CONTROL

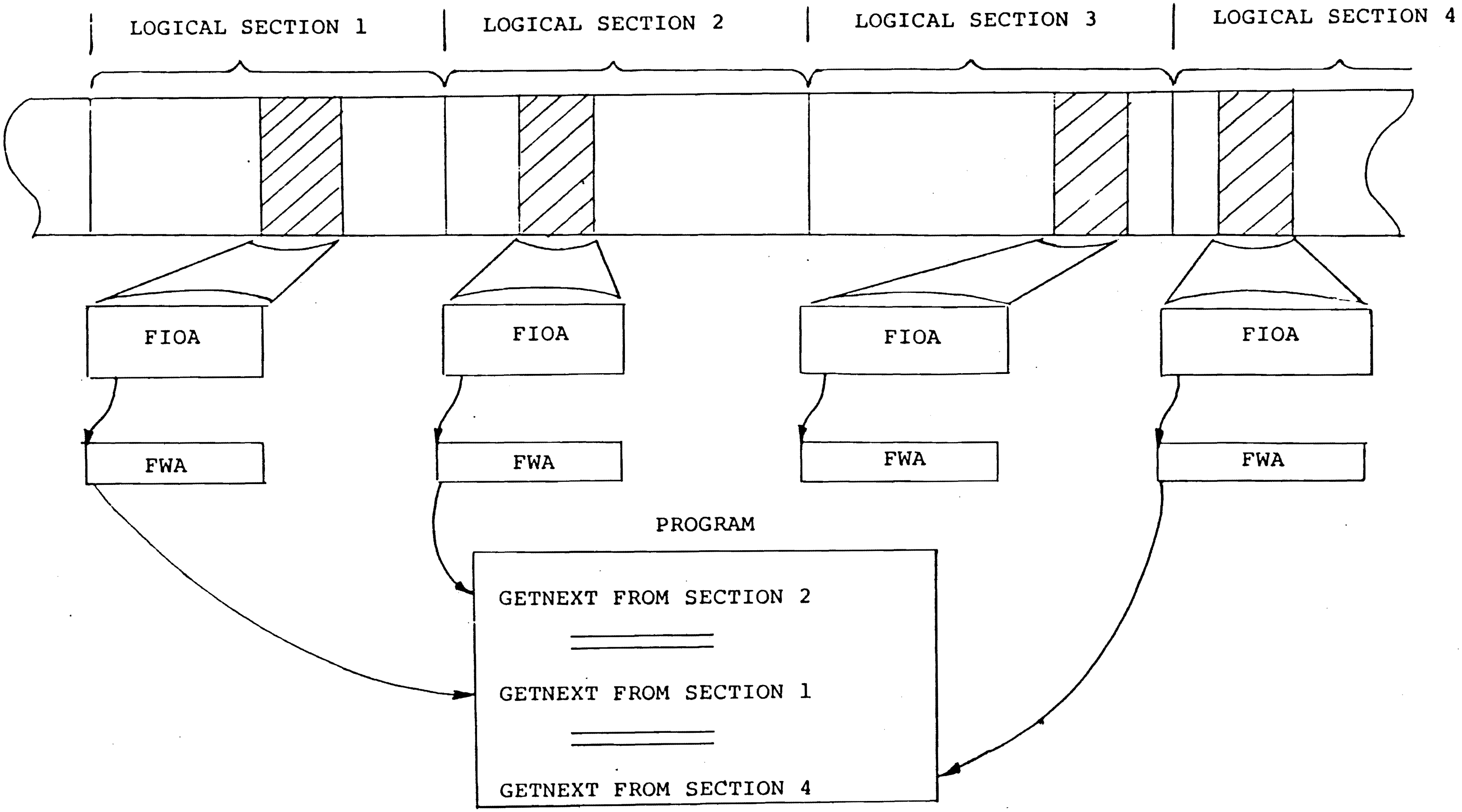
```
DFHFC      TYPE=GETNEXT
           [,SEGSET= { symbolic name } ]
           [         { YES           } ]
           [         { ALL           } ]
           [,NORESP=symbolic address]
           [,ERROR=symbolic address]
           [,SEGIDER=symbolic address]
           [,NOTFND=symbolic address]
           [,INVREQ=symbolic address]
           [,IOERROR=symbolic address]
           [,NOTOPEN=symbolic address]
           [,ENDFILE=symbolic address]
           [,ILLOGIC=symbolic address] <---------- VSAM
```

FILE CONTROL

```
DFHFC          TYPE=RESETL
               [,SEGSET= { symbolic  name }]
               [         { YES            }]
               [         { ALL            }]
               [,ARGTYP= { KEY }]                   }
               [         { RBA }]                   }
               [,SRCHTYP=  { FKEQ }  ]              }
               [           { FKGE }  ]              }   VSAM
               [           { GKEQ }  ]              }
               [           { GKGE }  ]              }
               [,NORESP=symbolic  address]
               [,ERROR=symbolic  address]
               [,SEGIDER=symbolic  address]
               [,NOTFND=symbolic  address]
               [,INVREQ=symbolic  address]
               [,IOERROR=symbolic  address]
               [,NOTOPEN=symbolic  address]
               [,ILLOGIC=symbolic  address]<------------ VSAM
```

FILE CONTROL

```
DFHFC        TYPE=ESETL
             [,NORESP=symbolic address]
             [,ERROR=symbolic address]
             [,INVREQ=symbolic address]
             [,ILLOGIC=symbolic address ] ←——— VSAM
```


LOGICAL SECTION 1
LOGICAL SECTION 2
LOGICAL SECTION 3
LOGICAL SECTION 4
FIOA
FIOA
FIOA
FIOA
FWA
FWA
FWA
FWA
PROGRAM
GETNEXT FROM SECTION 2
GETNEXT FROM SECTION 1
GETNEXT FROM SECTION 4

MULTIPLE BROWSE

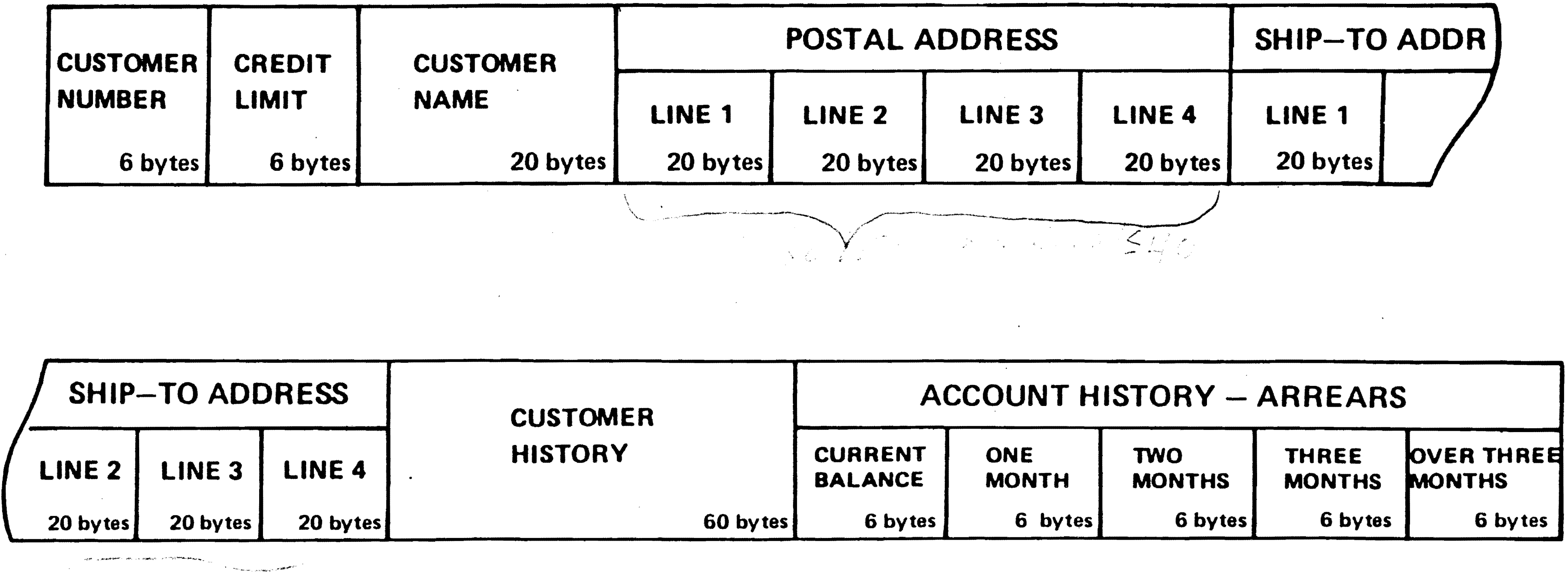
```
DFHTCADS   (TWA)

          SAVE ADDRESS FWA1
          SAVE ADDRESS FWA2

START BROWSE

          DFHFC  TYPE=SETL,DATASET=FILE1,RDIDADR=KEY1
          SAVE TCAFCAA IN FWA1 SAVE ADDRESS
          ---
          DFHFC  TYPE=SETL,DATASET=FILE2,RDIDADR=KEY2
          SAVE TCAFCAA IN FWA2 SAVE ADDRESS

RETRIEVE RECORDS

          RESTORE TCAFCAA WITH FWA1 OR FWA2
          DFHFC  TYPE=GETNEXT

TERMINATE FILE BROWSE

          RESTORE TCAFCAA WITH FWA1
          DFHFC  TYPE=ESETL
          ---
          RESTORE TCAFCAA WITH FWA2
          DFHFC  TYPE=ESETL
```

FILE CONTROL

```
DFHFC        TYPE=CHECK
             [,NORESP=symbolic address]
             [,ERROR=symbolic address]
             [,DSIDER=symbolic address]
             [,SEGIDER=symbolic address]
             [,NOTFND=symbolic address]
             [,DUPREC=symbolic address]
             [,INVREQ=symbolic address]
             [,IOERROR=symbolic address]
             [,DUPDS=symbolic address]
             [,NOSPACE=symbolic address]
             [,NOTOPEN=symbolic address]
             [,ENDFILE=symbolic address]
             [,ILLOGIC=symbolic address] ←——— VSAM
```

FILE CONTROL

```
DFHFC         TYPE=RELEASE
              [,NORESP=symbolic address]
              [,ERROR=symbolic address]
              [,INVREQ=symbolic address]
              [,IOERROR=symbolic address]
              [,ILLOGIC=symbolic address] ←——— VSAM
```



CUSTOMER NUMBER	CREDIT LIMIT	CUSTOMER NAME	POSTAL ADDRESS				SHIP–TO ADDR
			LINE 1	LINE 2	LINE 3	LINE 4	LINE 1
6 bytes	6 bytes	20 bytes	20 bytes	20 bytes	20 bytes	20 bytes	20 bytes

SHIP–TO ADDRESS			CUSTOMER HISTORY	ACCOUNT HISTORY – ARREARS				
LINE 2	LINE 3	LINE 4		CURRENT BALANCE	ONE MONTH	TWO MONTHS	THREE MONTHS	OVER THREE MONTHS
20 bytes	20 bytes	20 bytes	60 bytes	6 bytes	6 bytes	6 bytes	6 bytes	6 bytes

SEGMENTED RECORDS

V.9.21

variable format length bytes 4 bytes	ROOT SEGMENT 15 bytes			CUSTOMER NAME SEGMENT 12 bytes		POSTAL ADDRESS SEGMENTS						ACCOUNT HISTORY SEGMENT 30 bytes				
						LINE 1 13 bytes		LINE 2 9 bytes		LINE 3 7 bytes						
** var. lngth cntl LLb̸b̸	customer number	credit limit	segment presence indicators	*L	customer name	*L	postal address line 1	*L	postal address line 2	*L	postal address line 3	account history arrears				
												cur bal	1 mth	2 mth	3 mth	over 3 mth

with SEGMENTS

FILE CONTROL TABLE (FCT)

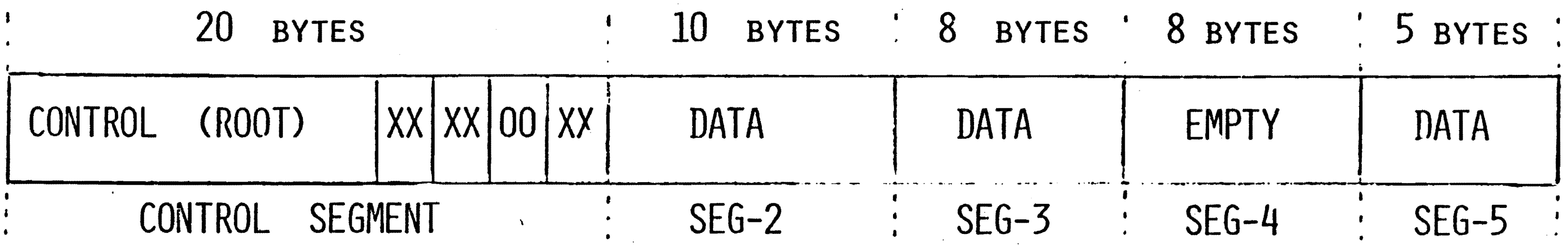
- CONTROL SEGMENT DEFINITION (HEADER)
    - LENGTH - OFFSET TO SEGMENT INDICATORS - TYPE OF INDICATORS
- SEGMENT DEFINITIONS (EACH SEGMENT IN RECORD)
    - LENGTH - SEGMENT NAME - CHARACTERISTICS
- SEGMENT SET DEFINITIONS
    - LOGICAL GROUPS OF SEGMENTS

DATASET=CUSTOMER	DATA SET SPECS			
SEGMENT=ROOT	LENGTH=15	BITINDICS	START=12	LNG=3
SEGMENT=NAME	VARIABLE	20 BYTES (MAX)	BYTE ALIGN	
SEGMENT=ADDR1	"	20 "	" "	
SEGMENT=ADDR2	"	20 "	" "	
SEGMENT=ADDR3	"	20 "	"	
SEGMENT=ADDR4	"	20 "	"	
SEGMENT=SHIP1	"	20 "	"	
SEGMENT=SHIP2	"	20 "	"	
SEGMENT=SHIP3	"	20 "	"	
SEGMENT=SHIP4	VARIABLE	20 "	BYTE ALIGN	
SEGMENT=HISTORY	FIXED	60 BYTES	WORD ALIGN	
SEGMENT=ARREARS	FIXED	60 BYTES	WORD ALIGN	

SEGSET=NAMEADDR (ROOT) NAME ADDR1 ADDR2 ADDR3 ADDR4

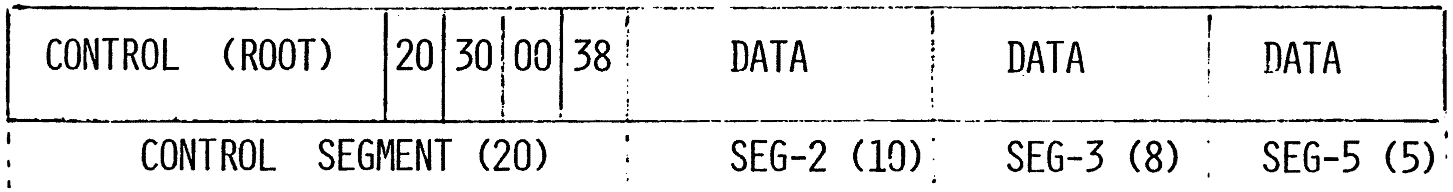
SEGSET=ACCOUNT (ROOT) NAME ARREARS

- APPLICATION - CREATES SEGMENTED RECORD



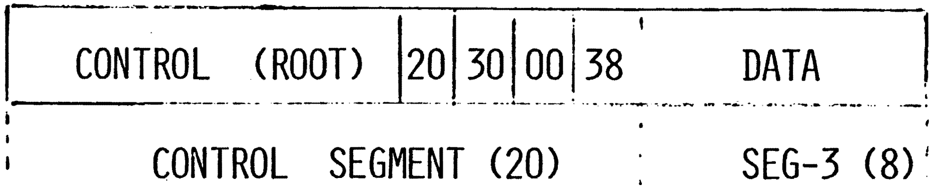
20 BYTES					10 BYTES	8 BYTES	8 BYTES	5 BYTES
CONTROL (ROOT)	XX	XX	00	XX	DATA	DATA	EMPTY	DATA
CONTROL SEGMENT					SEG-2	SEG-3	SEG-4	SEG-5

- CICS/VS - MODIFY DISPLACEMENT INDICATORS AND COMPRESS SEGMENTS



CONTROL (ROOT)	20	30	00	38	DATA	DATA	DATA
CONTROL SEGMENT (20)					SEG-2 (10)	SEG-3 (8)	SEG-5 (5)

- APPLICATION - REQUESTS "SEG-3"



CONTROL (ROOT)	20	30	00	38	DATA
CONTROL SEGMENT (20)					SEG-3 (8)

FILE CONTROL TABLE DEFINES

ROOT AND SEGMENTS 2-9

SEGMENT SET SEGA AS SEGMENTS 2,6,AND 9

RECORD ON DASD CONTAINS

ROOT	SEG2	SEG3	SEG9

DFHFC TYPE=GET,DATASET=MSTR,SEGSET=SEGA

FWA

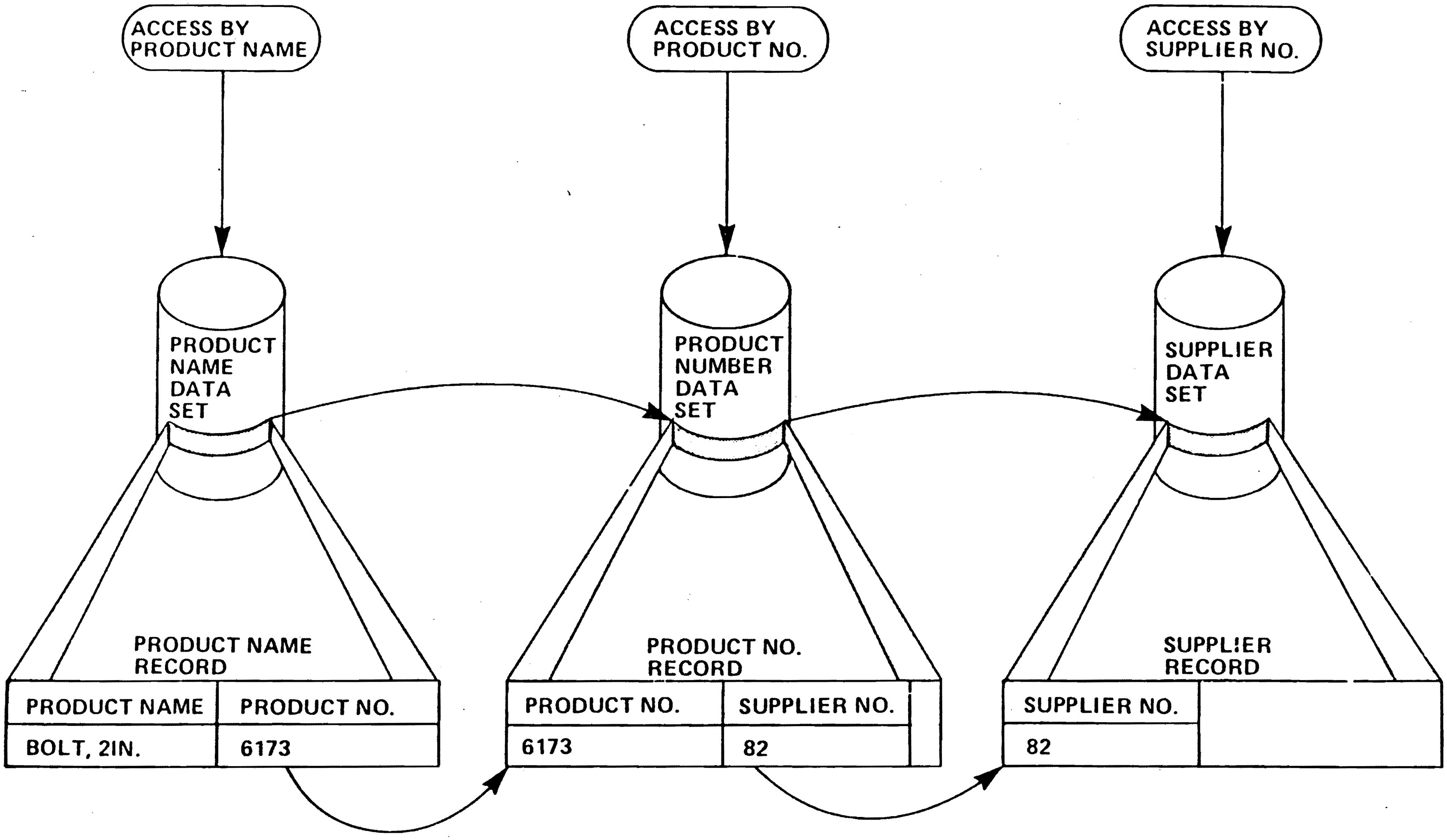
ROOT	SEG2	EMPTY	SEG9

DFHFC TYPE=GET,DATASET=MSTR,TYPOPER=UPDATE,SEGSET=ALL

FWA

ROOT	SEG2	SEG3	EMPTY		SEG9

11



PRODUCT NAME	PRODUCT NO.
BOLT, 2IN.	6173

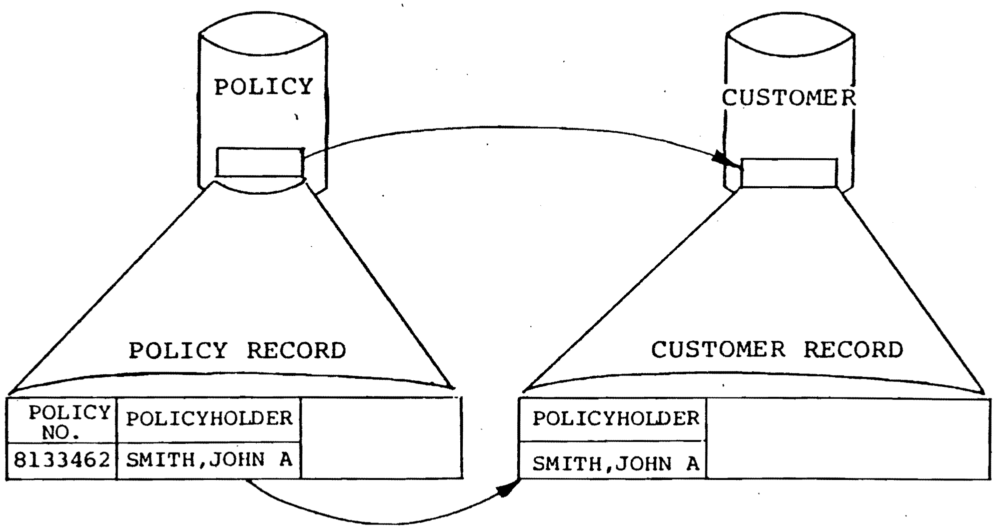
PRODUCT NO.	SUPPLIER NO.
6173	82

SUPPLIER NO.	
82	

V.9.27

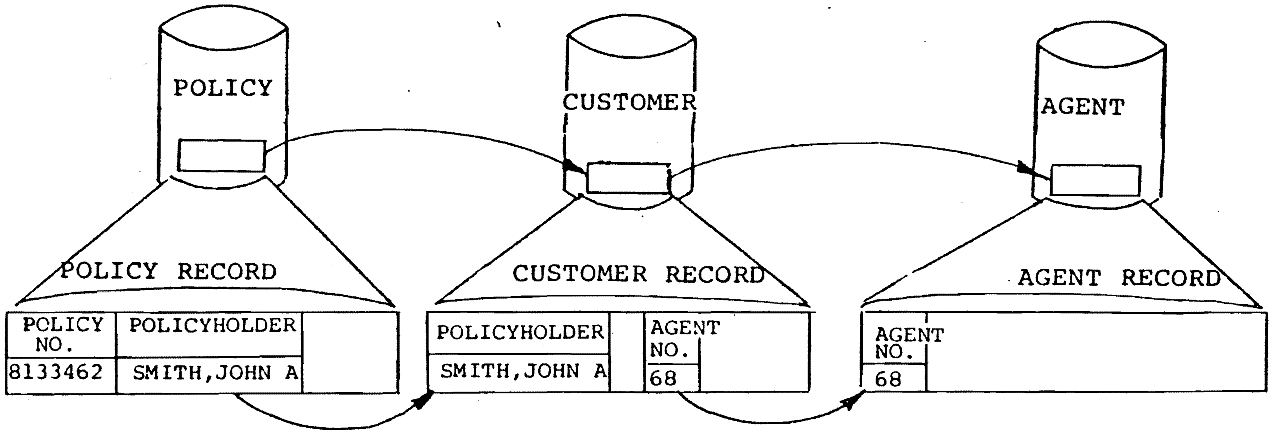
NDIRECT ACCESS

INDIRECT ACCESS



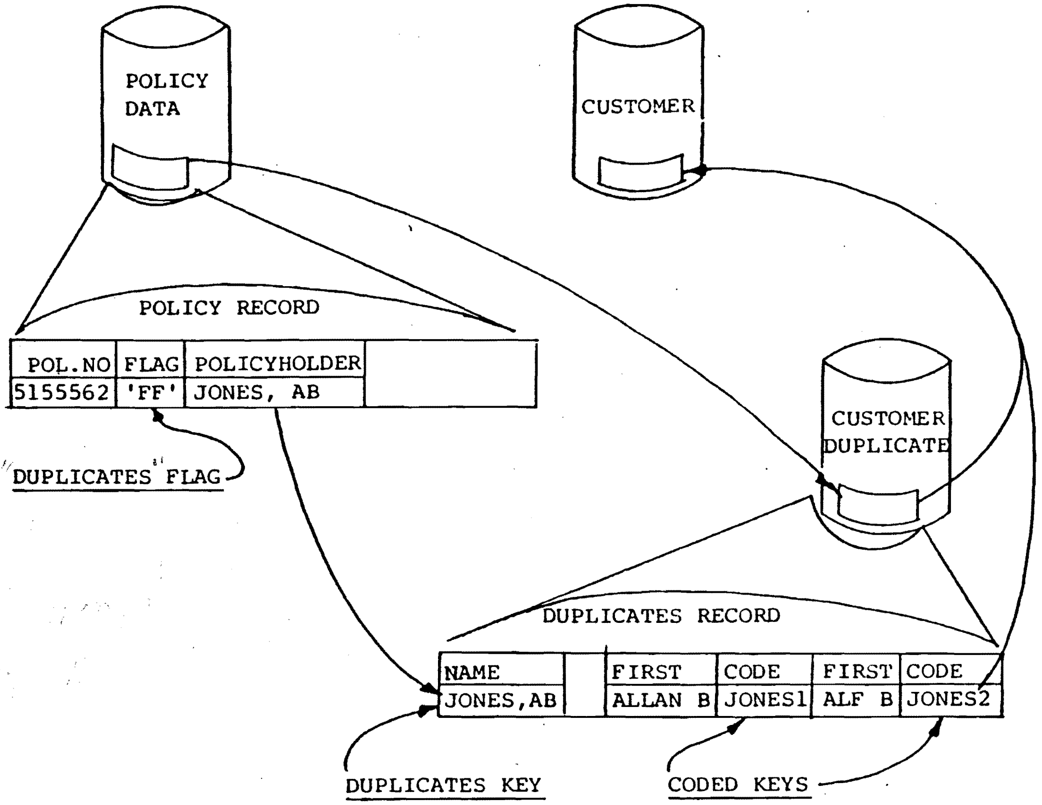
```
DFHFC   TYPE=GET,DATASET=CUSTOMER,INDEX=POLICY,RDIDADR=KEYFLD,NORESP=PROCESS
```

INDIRECT ACCESS



```
DFHFC   TYPE=GET,INDEX=POLICY,DATASET=AGENT,RDIDADR=KEYFLD,NORESP=PROCESS
```

INDIRECT ACCESS



POL.NO	FLAG	POLICYHOLDER	
5155562	'FF'	JONES, AB	

NAME		FIRST	CODE	FIRST	CODE
JONES,AB		ALLAN B	JONES1	ALF B	JONES2

```
DFHFC   TYPE=GET,DATASET=CUSTOMER,INDEX=POLICY,RDIDADR=KEYF,NORESP=PROCESS,     X
        DUPDS=DUPLICATE
```

VSAM MASS INSERT

```
          DFHFC  TYPE=GETAREA,TYPOPER=MASSINSERT,     *
                ≀
READ  INPUT•••••
                ≀
          DFHFC  TYPE=PUT,TYPOPER=NEWREC,             *
                ≀
IF  NOT  END–OF–INPUT,  GO  TO  READ–INPUT•••••
                ≀
          DFHFC  TYPE=RELEASE,                        *
                ≀
```

VSAM GROUP DELETE

```
DFHFC  TYPE=DELETE,                              *
         SRCHTYP=GKEY,                           *
         RDIDADR=PARTKEY,                        *
```

TRACE CONTROL

DEBUGGING AID FOR APPLICATION DEVELOPMENT

SELECTIVE RECORDING OF CICS/VS MANAGEMENT REQUESTS

RECORD OF USER SPECIFIED ENTRIES

SIZE OF TRACE TABLE SPECIFIED IN SIT (TRT=NN)

DUMMY TRACE – not [illegible]

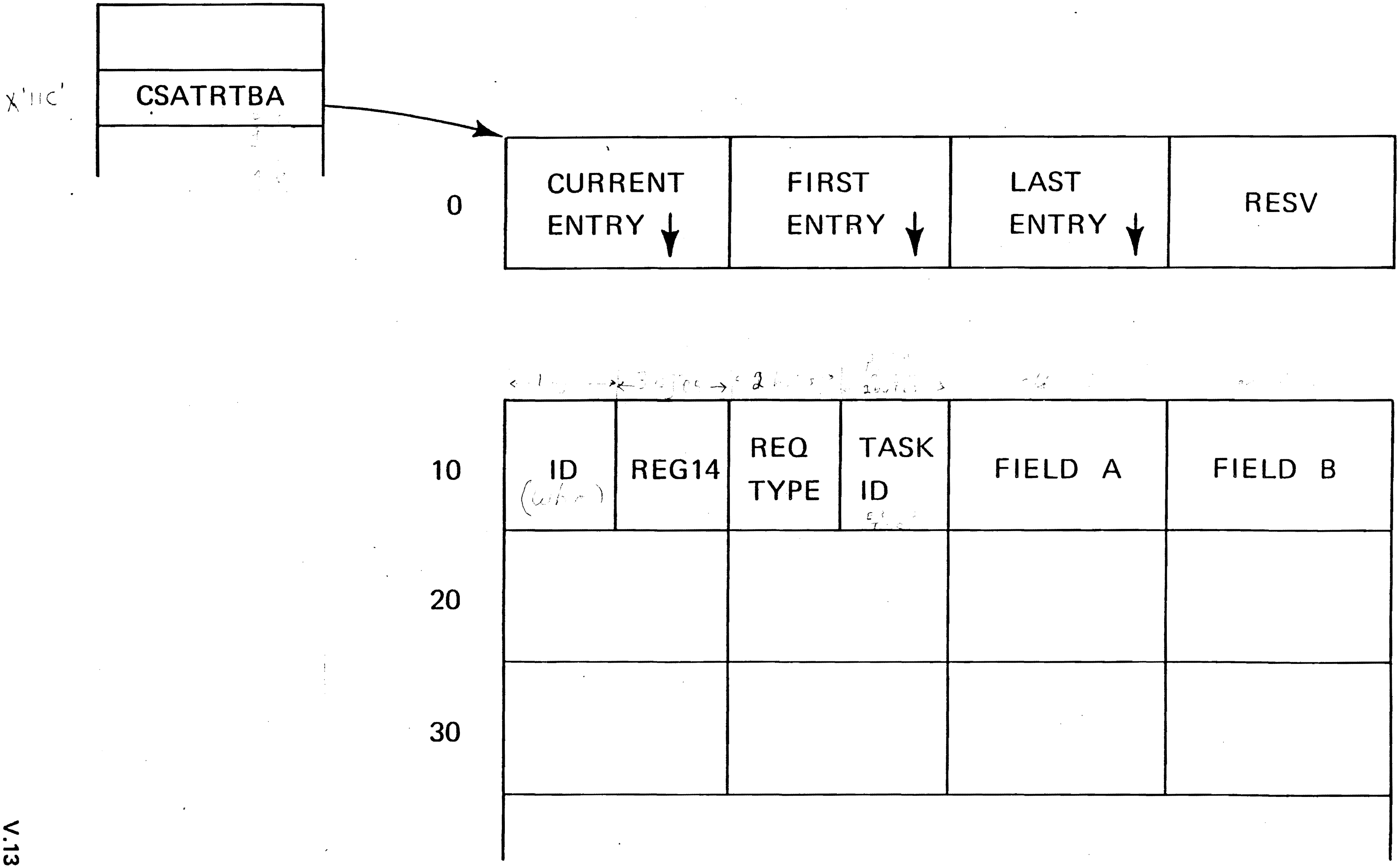
AUXILIARY TRACE FOR MORE EXTENSIVE ANALYSIS.

→ more than that the wrap around Trace table.

16 BYTE ENTRIES

WRAP-AROUND TABLE

FIRST ENTRY IS 'CONTROL'



TRACE IDENTIFICATION

00 – C7	0 – 199	USER ENTRIES
C8 – E5	200 – 229	SYSTEM ENTRIES
E6 – EF	230 – 239	F E ENTRIES
FO – FD	240 – 253	TRACE CONTROL ENTRIES (CICS/VS)
FE	254	TRACE ON
FF	255	TRACE OFF

TRACE CONTROL

TR

```
DFHTR     TYPE=ENTRY
          ┌             ┌ SYSTEM ┐ ┐
          │ ,STYPE=     │ USER   │ │
          └             └ FE     ┘ ┘
          ,ID=number
          ┌           ┌ symbol   ┐ ┐   │   ┌            ┌ register   ┐ ┐
          │ ,DATA1=   └ (symbol) ┘ │   │   │ ,RDATA1=   └ (register) ┘ │
          └                        ┘   │   └                           ┘
          ┌           ┌ symbol   ┐ ┐   │   ┌            ┌ register   ┐ ┐
          │ ,DATA2=   └ (symbol) ┘ │   │   │ ,RDATA2=   └ (register) ┘ │
          └                        ┘   │   └                           ┘
          ┌             ┌ HBIN    ┐ ┐
          │ ,DATA1TP=   │ FBIN    │ │
          │             │ CHAR    │ │
          │             │ PACK    │ │
          └             └ POINTER ┘ ┘
          ┌             ┌ HBIN    ┐ ┐
          │ ,DATA2TP    │ FBIN    │ │
          │             │ CHAR    │ │
          │             │ PACK    │ │
          └             └ POINTER ┘ ┘
```

```
DFHTR     TYPE=ENTRY,STYPE=USER,ID=110,         X
          DATA1=FLDA,DATA1TP=CHAR
```

```
CUSTOMER INFORMATION CONTROL SYSTEM STORAGE DUMP    CODE=ASRA   TASK=PARX   DATE=76.147   TIME=10.33.363       PAGE 003

TRACE TABLE                     ADDRESS  099800  TO  099E4F      LENGTH  000650



000040  F10BCB0A EA00E3C3 000001E0 800AEED0  C80AC162 0000E3C3 00088800 8A0401E8   *1.....TC........H.A...TC.......Y*    099840
000060  F10BCB44 E100E3C3 00000038 800AEED0  C80AC162 0000E3C3 00088000 81000040   *1.....TC........H.A...TC....... *    099860
000080  F00B429E 4000E3C3 40000000 000AF194  F20AA360 0200004C D7C1D9E3 D7C7D4C6   *0... .TC .....1.2.......PARTPGMF*    099880
0000A0  F10AAC22 8800004C 000002E2 010AEED0  C80AC162 0000004C 00091000 88001800   *1..........S....H.A.............*    0998A0
0000C0  F00AAC76 2000004C FF000000 00000000  F00AAE38 4000004C 80000000 000AB6D0   *0...............0... ...........*    0998C0
0000E0  FD000000 003A00DF 003A00E1 0000002C  F00AAFCE 2000004C 21000000 00000000   *................0...............*    0998E0
000100  F10AB160 8C00004C 00090240 010AEED0  C80AC162 0000004C 000889F0 8C090248   *1.......... ....H.A........0....*    099900
000120  F60AB4FE 4000004C 00088980 E2E3C1E3  F00A62FA 4000004C 20000000 000AB6D0   *6... .......STAT0... ...........*    099920
000140  F50AB4FE 6000004C D7C1D9E3 E2404040  F10A91FA 9D00004C 0008001C 010AEED0   *5.......PARTS   1...............*    099940
000160  C80AC162 0000004C 00088C40 9D080028  F10A903A 8F00004C 00080060 010AEED0   *H.A........ ....1...............*    099960
000180  C80AC162 0000004C 00088C70 8F080068  F00A7F0C 4000004C 80000000 00088C80   *H.A.............0... ...........*    099980
0001A0  F10A903A 8F00004C 0008006A 010AEED0  C80AC162 0000004C 00088CE0 8F080078   *1...............H.A.............*    0999A0
0001C0  F10A91E2 4000004C 00088C40 010AEED0  C90AC24C 0000004C 00088C40 9D080028   *1..S ...... ....I.B......... ....*    0999C0
0001E0  F10A91E2 4000004C 00088C70 010AEED0  C90AC24C 0000004C 00088C70 8F080068   *1..S ............I.B.............*    0999E0
000200  F2088C08 6000004C C1E2D9C1 00000000  F40AA55C FE00004C 00000000 C1E2D9C1   *2.......ASRA....4...........ASRA*    099A00
```

PGM CHK.

```
        CUSTOMER INFORMATION CONTROL SYSTEM  -  T R A C E   U T I L I T Y   P R O G R A M
        SELECTION PARAMETERS ARE
   TIME OF DAY  HEXADECIMAL DISPLAY OF TRACE ENTRY   CHAR-DISPLAY       TRACE TYPE
AUXILIARY TRACE ACTIVATED AT -  15:51:45.94

     15:51:51.41    F0142 8F8 4000E3C3 40000000 0013EAA0   0..8..TC........   KCP WAIT
     15:51:51.41    D0138F8E  0500E3C3 00000000 00000000   ......TC........   KCP DISPATCH
     15:51:51.41    F01448D8  1100E3C3 0113DE00 C9D5D8C1   0..Q..TC....INQA   KCP CDL-ATACH
     15:51:51.42    F1138310  EA00E3C3 001001E4 8013DE00   1.....TC...U....   SCP
     15:51:51.42    C813A96E  0000E3C3 00143800 8A9401F8   H.....TC.......Y   SCP
     15:51:51.43    F113834A  F100E3C3 000000C8 8013DE00   1.....TC...H....   SCP
     15:51:51.43    C813A96E  0000E3C3 00143000 81000000   H.....TC........   SCP
     15:51:51.43    D0138F66  0600002C E2C5D8C1 C9D5D8C1   ........SEQAINQA   KCP CREATE
     15:51:51.43    D0138F8E  0500E3C3 00000000 00000000   ......TC........   KCP DISPATCH
     15:51:51.44    F01428F8  4000E3C3 40000000 0013EAA0   0..8..TC........   KCP WAIT
     15:51:51.44    D0138F8E  0500002C 00000000 00000000   ................   KCP DISPATCH
     15:51:51.44    F213981A  0200002C C3E2C9D5 D8D9E8C1   2.......CSINQRYA   PCP XCTL
     15:51:51.46    F11398C6  8C00002C 00190250 0113DE00   1..F............   SCP
     15:51:51.46    C813A96F  0000002C 001439F0 80190258   H..........0....   SCP
     15:51:51.50    F5139F14  8000002C C3E4E2E3 D6D4D940   5.......CUSTOMR.   FCP GET
     15:51:51.54    F4139F14  FE00002C 00000000 C3E4C7E3   4...........CUGT   DCP
     15:51:51.54    F019C924  4000002C 80000000 0019CA60   0.I.............   KCP WAIT
     15:51:51.60    D0138F8E  0500002C 00000000 00000000   ................   KCP DISPATCH
     15:51:51.60    F019C924  4000002C 80000000 0019CA60   0.I.............   KCP WAIT
     15:51:51.63    D0138F8E  0500002C 00000000 00000000   ................   KCP DISPATCH
     15:51:51.64    F019C924  4000002C 80000000 0019CA60   0.I.............   KCP WAIT
     15:51:51.66    D0138F8E  0500002C 00000000 00000000   ................   KCP DISPATCH
```

TIME STAMP

TRACE TYPE

Run 'DFHTRP'

DUMP CONTROL

- SELECTIVE STORAGE DUMPS
  - SYSTEM TABLES AND AREAS
  - TRANSACTION STORAGE
  - SPECIFIC STORAGE
  - PROGRAM STORAGE

FIFO

- DUMP REQUEST QUEUEING

- DUMP DATA SET
  - SEQUENTIAL DATA SET
  - TAPE OR DASD

- CICS PROVIDED DUMP UTILITY PROGRAM

DFHDUP p

- VALUABLE DEBUGGING AID
  - TESTING ENVIRONMENT
  - ABNORMAL TASK OPERATION

```
CUSTOMER INFORMATION CONTROL SYSTEM STORAGE DUMP    CODE=ASRA    TASK=PARX    DATE=76.147    TIME=10.33.363        PAGE 001

                                                           PSW AT ENTRY TO ABEND  078D0004 C008EC4C
REGS 14-4          500AA55C   000AB23E   00000208   00000000            0008FE20   0008F300   0008E808
REGS 5-11          0008FE28   0008899F   000BE990   00000000            00089408   0008E9F0   0008F2C8

TASK CONTROL AREA (USER AREA)  ADDRESS  088880  TO  0889EF       LENGTH  000170
000000    00088800 0009CD78 010AEED0 00000000    00000000 0009CD60 80402100 043100A0   *........................ ......*    088880
000020    4009CC5A 92D5F3F0 00000208 0009CD60    5009C60A A009C73C 00088300 8009CC1A   * ....N30..........F...G.........*    0888A0
000040    00000080 000BE990 00000000 00089408    000ABDF4 0008F2C8 4009CC5A 40089A60   *......Z............4..2H ... ...*    0888C0
000060    0008FE20 0008899F 0008FE28 00089A68    500A7104 000A8104 0008881C 00089A60   *..............................*    0888E0
000080    FE009AD0 D7C1D9E3 E2404040 C1E2D9C1    078D0004 C008EC4C 00000000 00087816   *....PARTS   ASRA................*    088900
0000A0    500AA55C 000AB23E 00000208 00000000    0008FE20 0008F300 0008E808 0008FE28   *.......................3...Y.....*    088920
0000C0    0008899F 000BE990 00000000 00089408    0008E9F0 0008F2C8 00000000 00000000   *......Z...........Z0..2H........*    088940
0000E0    00000000 00000000 24F00000 00000000    00000000 00000000 000A70DA 00000000   *.........0.....................*    088960
000100    00270000 D7C1D9E7 6BC4C9E2 D76BE2D1    F0F1F000 00000000 E3F0F0F1 F04040F1   *....PARX.DISP.SJ010.....T0010  1*    088980
000120    F0F3F3F3 F6F34000 00000000 00000000    00000000 00000000 00000000 00000000   *033363 .........................*    0889A0
000140    00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000   *..............................*    0889C0
000160    00000000 00000000 8A0401E8 00089AD0                                          *...........Y....               *    0889E0

TASK CONTROL AREA (SYSTEM AREA)ADDRESS  088800  TO  08887F       LENGTH  000080
000000    8A0401E8 00089AD0 00089A30 00089B50    0000005C 000ABDF4 00088000 00000000   *...Y...................4........*    088800
000020    000BE4E0 00000000 00000000 00000000    00000000 010AA158 00000000 00089220   *..U.............................*    088820
000040    00089210 00000000 00000000 00000000    00000000 00000000 00000000 00000000   *................................*    088840
000060    00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000   *................................*    088860

 ASRA  REGS  0 THRU 15         ADDRESS  09CCF8  TO  09CD37       LENGTH  000040
000000    00000208 00000000 0008FE20 0008F300    0008E808 0008FE28 0008899F 000BE990   *..............3...Y...........Z.*    09CCF8
000020    00000000 00089408 0008E9F0 0008F2C8    00088A08 00089220 4008EC20 000AB23E   *..........Z0..2H........ .......*    09CD18
```

DUMP CONTROL

```
DFHDC        TYPE=TRANSACTION
             ,DMPCODE= { value }
                       { YES   }
```

COMMON DUMP CONTENTS

TCA, TWA, CSA, TRACE TABLE, REGISTERS, TCTTE, OR DCTE

```
DFHDC        TYPE=TRANSACTION,DMPCODE=ABC5
```

DUMP CONTROL

```
DFHDC        TYPE=CICS      ([illegible])
             ,DMPCODE= { value }
                       { YES   }


DFHDC        TYPE=COMPLETE
             ,DMPCODE= { value }
                       { YES   }


DFHDC        TYPE=PARTIAL
             ,LIST=(  [TERMINAL] [,PROGRAM] [,TRANSACTION] [,SEGMENT])
             ,DMPCODE= { value }
                       { YES   }
```

COMMON DUMP CONTENTS

TCA, TWA, CSA,   TRACE TABLE, REGISTERS, TCTTE, OR DCTE

SEGMENT DUMP

ADDRESS OF AREA TO TCADCSA

LENGTH OF AREA TO TCADCNB

DFHDC TYPE=PARTIAL,LIST=SEGMENT

DFHDC TYPE=PARTIAL,LIST=(SEGMENT,PROGRAM)

```
01 DFHBLLDS COPY DFHBLLDS.
   02 ...
   02 SAACBAR PIC S9(8) COMP.
01 ...
01 DFHSAADS COPY DFHSAADS.

     MOVE 'DCAF' TO TCADCDC.
     MOVE FWACBAR TO SAACBAR,TCADCSA.
     MOVE SAASAD TO TCADCNB.
     GO TO DUMP-RTN.

DUMP-RTN.
      DFHDC TYPE=PARTIAL,LIST=SEGMENT,DMPCODE=YES
```

```
%INCLUDE DFHSAADS;

   SAACBAR=FWACBAR;
   TCADCNB=SAASAD;
   TCADCSA=FWACBAR;

      DFHDC TYPE=PARTIAL,LIST=SEGMENT,DMPCODE=DCAF
```

BUILT-IN FUNCTIONS

BASIC FUNCTIONS

TABLE SEARCH

PHONETIC CONVERSION

VERIFY DATA FIELD

EDIT DATA FIELD

BIT MANIPULATION ([illegible])

INPUT FORMATTING

WEIGHTED RETRIEVAL (lik FISTER)

BUILT–IN FUNCTIONS

```
DFHBFTCA     ⎡OPTION= ⎧BASIC ⎫⎤
             ⎣        ⎩WTRET ⎭⎦
```

[illegible]

[illegible]

[illegible]

ASSEMBLER EXAMPLE

```
          COPY        DFHTCADS
AREA      DS          CL20
          DFHBFTCA    OPTION=BASIC
```

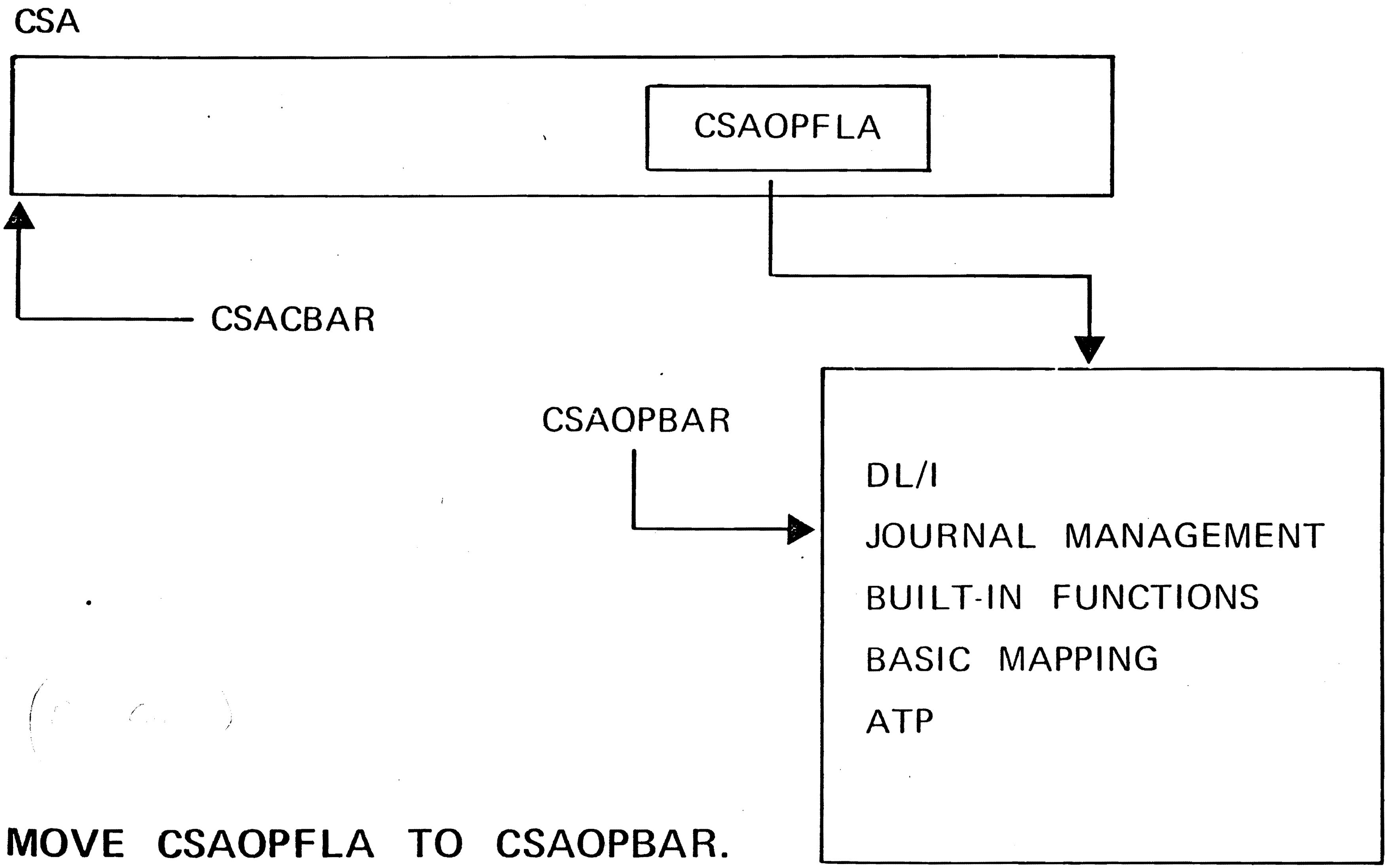
COBOL EXAMPLE

```
          01          DFHTCADS COPY DFHTCADS.
                      02 AREA PIC X(20).
          DFHBFTCA    OPTION=BASIC
```

PL/I EXAMPLE

```
          % INCLUDE (DFHTCADS);
             2 AREA CHAR (20);
          DFHBFTCA    OPTION=BASIC
```

CSA OPTIONAL FEATURES LIST



CSA

CSAOPFLA

CSACBAR

CSAOPBAR

DL/I

JOURNAL MANAGEMENT

BUILT-IN FUNCTIONS

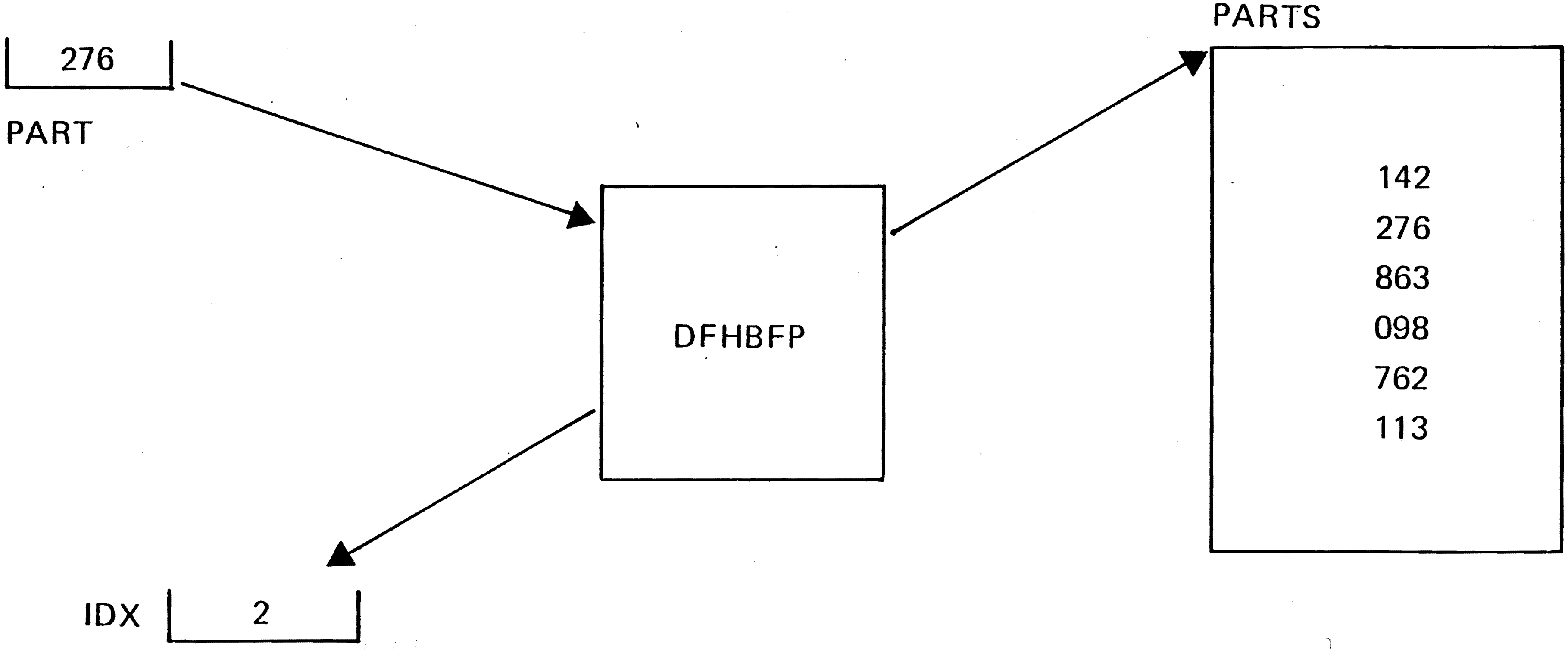
BASIC MAPPING

ATP

MOVE CSAOPFLA TO CSAOPBAR.

TABLE SEARCH

(NO FUNCTION TABLE)



PART

276

DFHBFP

PARTS

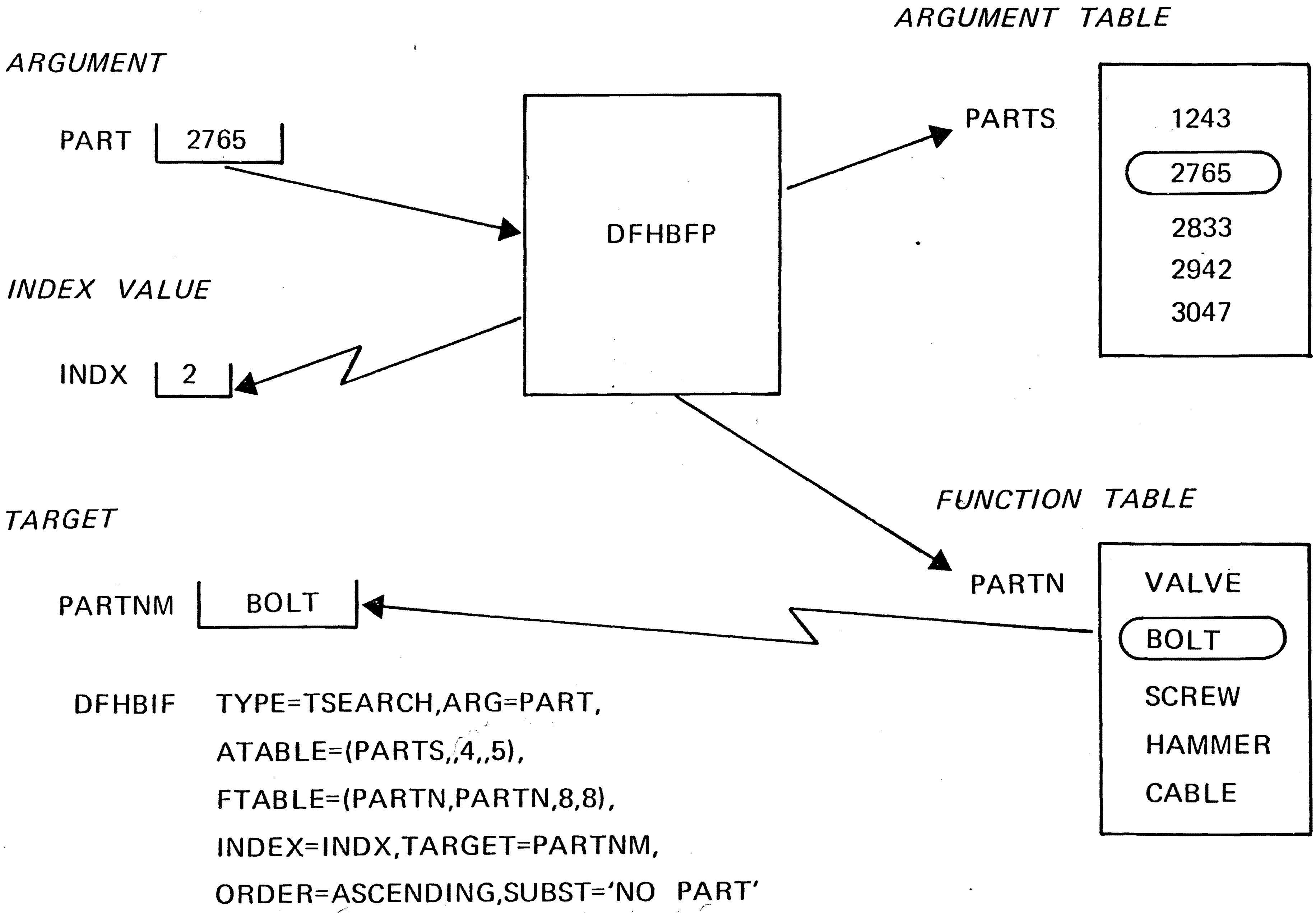
142
276
863
098
762
113

IDX

2

```
DFHBIF   TYPE=TSEARCH,
         ARG=PART,
         ATABLE=(PARTS,,3,,6),
         INDEX=IDX
```

TABLE SEARCH



```
DFHBIF    TYPE=TSEARCH,ARG=PART,
          ATABLE=(PARTS,,4,,5),
          FTABLE=(PARTN,PARTN,8,8),
          INDEX=INDX,TARGET=PARTNM,
          ORDER=ASCENDING,SUBST='NO  PART'
```

(BINARY SEARCH) [illegible]

TABLE 1

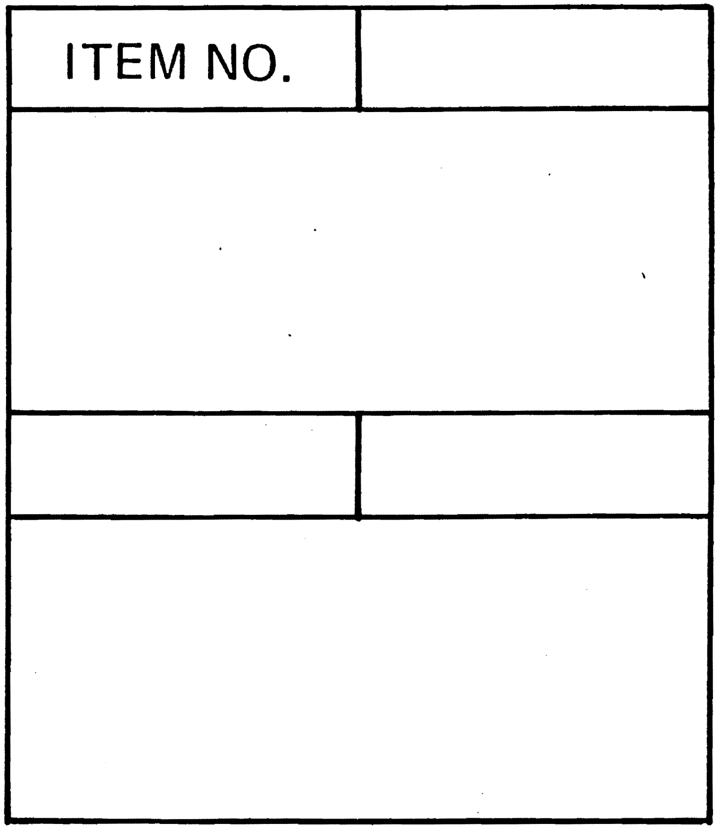
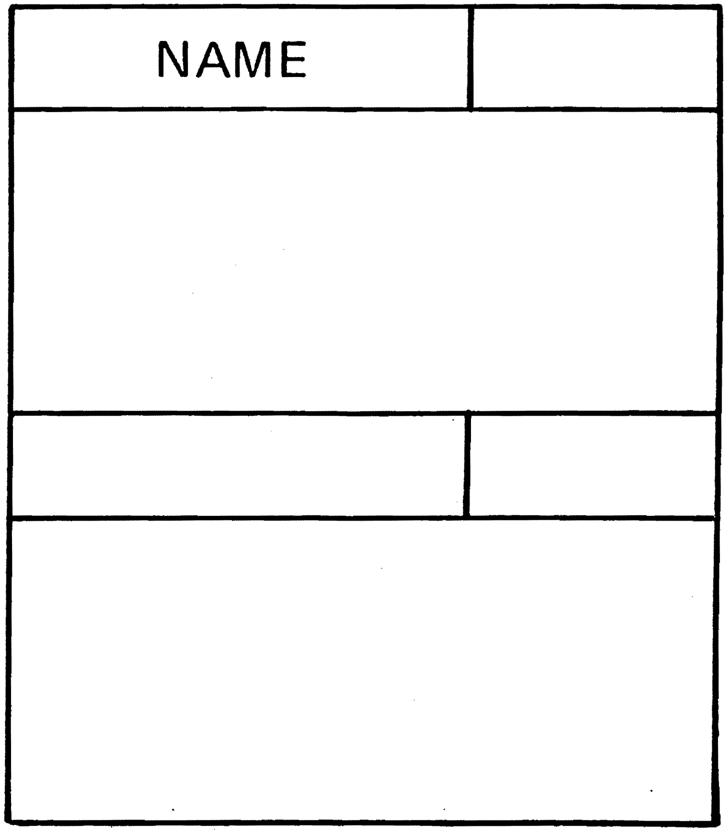


TABLE 2



ATABLE = (ATBL,AFLD,ATBLN,AFLDLN,ENTRIES)

FTABLE = (FTBL,FFLD,FTBLN,FFLDLN)

EMPL. NO.		LOC		POS		NAME	
31467		027		131		SMITH	

ATABLE = (ATBL,AFLD,ATBLN,AFLDLN,ENTRIES)

FTABLE = (FTBL,FFLD,FTBLN,FFLDLN)

[illegible]

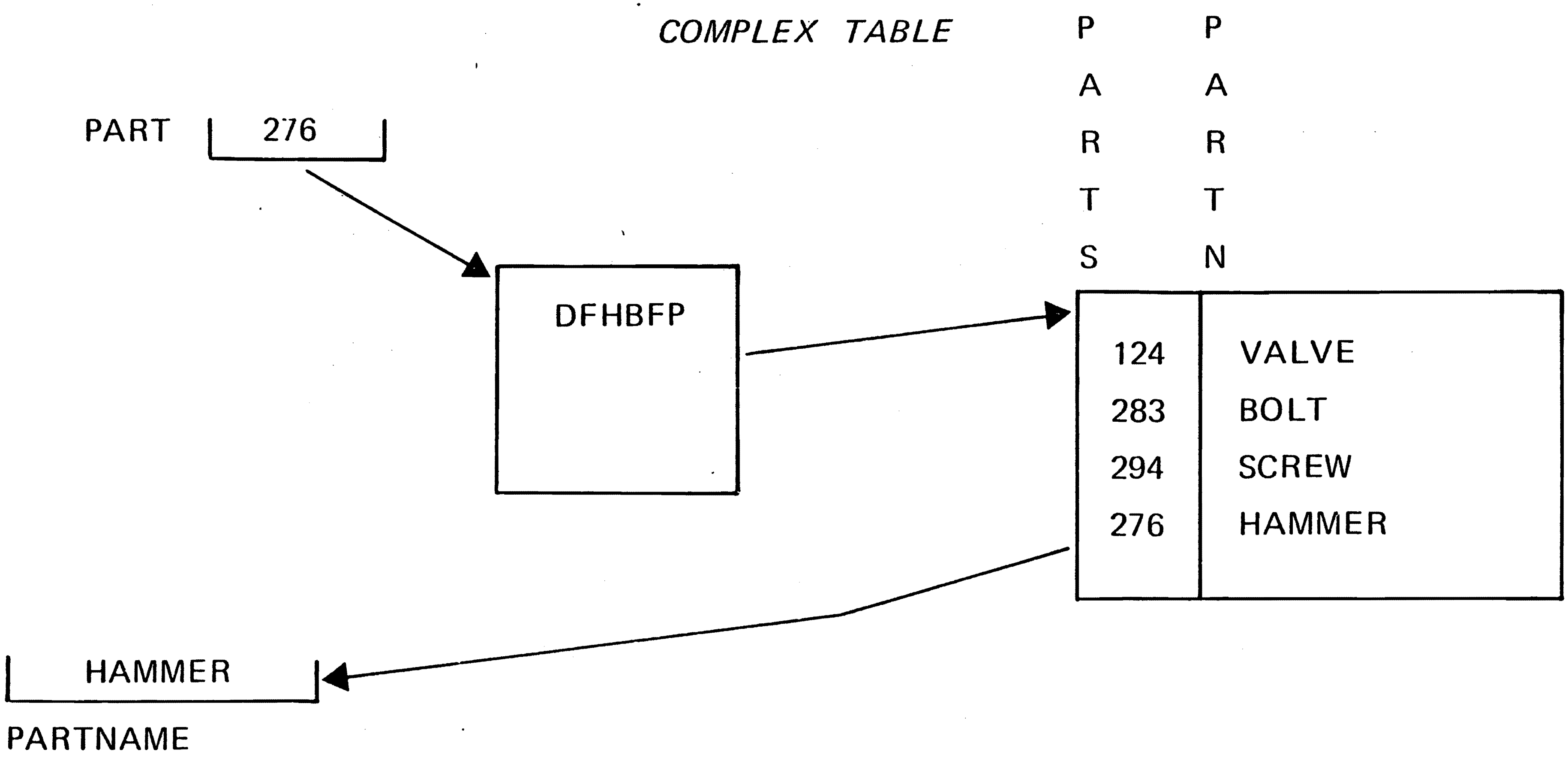
TABLE SEARCH

```
DFHBIF      TYPE=TSEARCH
          [,ARG=symbolic address]
          [,TARGET=symbolic address]
          ┌,ATABLE=( [symbolic address1] ┌ , ┌ symbolic address2 ┐┐ ┐
          │                              └   └ YES               ┘┘ │
          │          [,numeric value1] ┌ , ┌ numeric value2 ┐┐      │
          │                            └   └ YES            ┘┘      │
          └          [,numeric value3])                             ┘
          ┌,FTABLE=( ┌ ┌ symbolic address1 ┐┐  ┌ , ┌ symbolic address2 ┐┐ ┐
          │          └ └ YES               ┘┘  └   └ YES               ┘┘ │
          │          ┌ , ┌ numeric value1 ┐┐  ┌ , ┌ numeric value2 ┐┐ )   │
          └          └   └ YES            ┘┘  └   └ YES            ┘┘     ┘
          ┌,ORDER= ┌ ASCENDING  ┐┐
          └        └ DESCENDING ┘┘
          ┌,SUBST= ┌ symbolic address ┐┐ │ [,NOMATCH=symbolic address]
          └        └ 'literal value'  ┘┘ │
          [,INDEX=symbolic address]
          [,RANGE=YES
          [,ERROR=symbolic address]
```

[illegible]

[illegible]

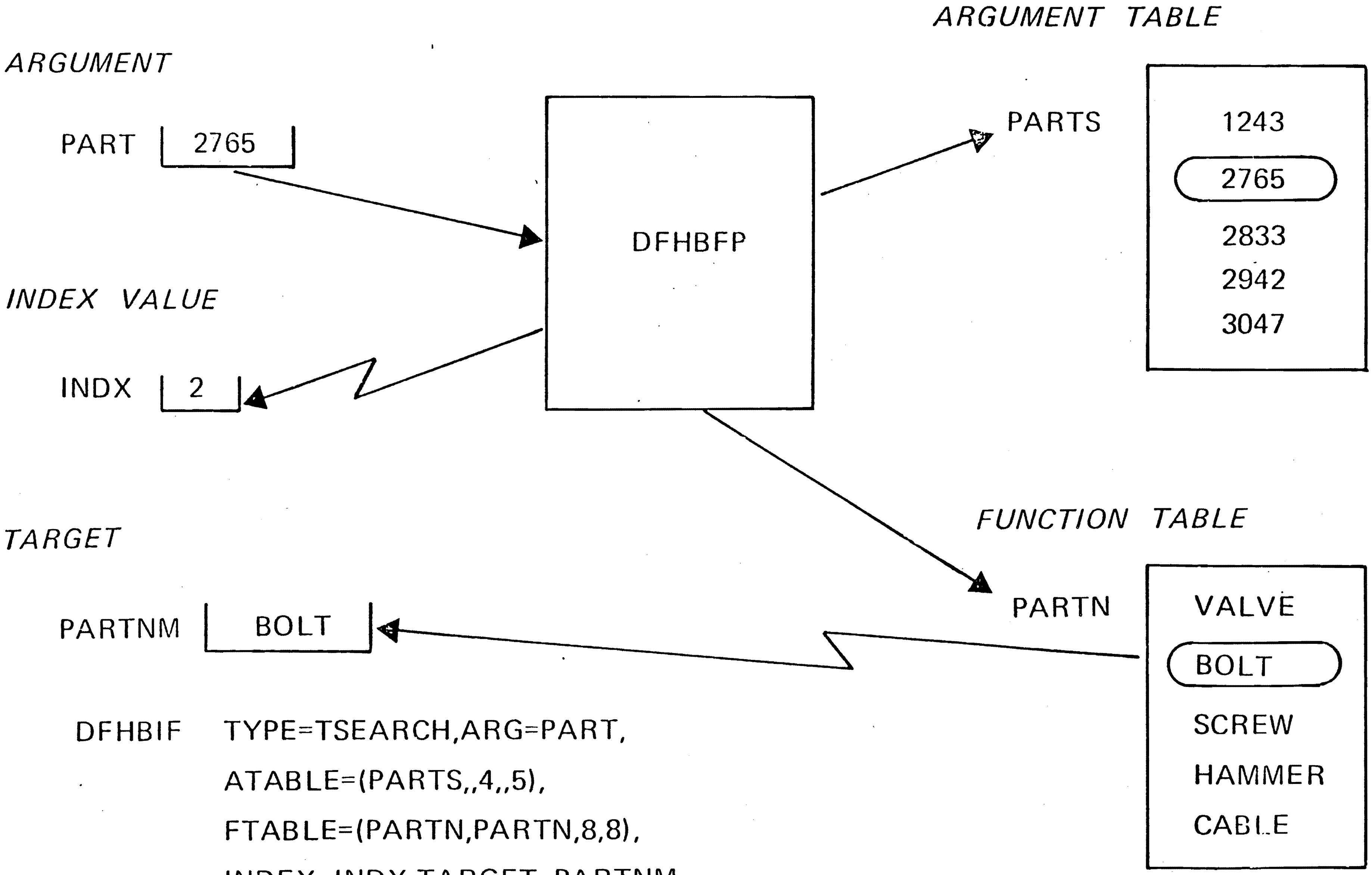
TABLE SEARCH



PARTS	PARTN
124	VALVE
283	BOLT
294	SCREW
276	HAMMER

```
DFHBIF   TYPE=TSEARCH,
         ARG=PART,
         TARGET=PARTNAME,
         ATABLE=(PARTS,,13,3,4),
         FTABLE=(,PARTN,,10),
         NOMATCH=NOTFND
```

TABLE SEARCH



ARGUMENT

PART 2765

INDEX VALUE

INDX 2

TARGET

PARTNM BOLT

DFHBFP

ARGUMENT TABLE

PARTS

1243
2765
2833
2942
3047

FUNCTION TABLE

PARTN

VALVE
BOLT
SCREW
HAMMER
CABLE

```
DFHBIF   TYPE=TSEARCH,ARG=PART,
         ATABLE=(PARTS,,4,,5),
         FTABLE=(PARTN,PARTN,8,8),
         INDEX=INDX,TARGET=PARTNM,
         ORDER=ASCENDING,SUBST='NO  PART'
```

PHONETIC CONVERSION

NAMES THAT SOUND ALIKE

SMITH
SMYTHE
SMITTHE
SMYTH

Mc ALILLY
MAC ALILLY
Mc ALLILE
MAC ALLILEY

PRODUCE THE SAME PHONETIC KEY

S 5 3 0	SMITHE
M 2 4 0	Mc ALLILY

PHONETIC CONVERSION

1	B P F V
2	C G J K S Q X Z
3	D T
4	L
5	M N
6	R
NO VALUE	A E I O U W Y H
	numbers or special characters

PHONETIC CONVERSION

```
DFHBIF     TYPE=PHONETIC
          [,FIELD=symbolic address]
          [,ERROR=symbolic address]
```

PHONETIC CONVERSION SUBROUTINE

Batch version

ASSEMBLER LANGUAGE:

```
CALL DFHPHN, (lang,name,phon)
```

ANS COBOL:

```
CALL 'DFHPHN' USING lang name phon.
```

PL/I:

```
CALL DFHPHN (lang,name,phon);
```

FIELD VERIFICATION

CHECK FIELD FOR

- ALL ALPHA (A – Z or b)
- ALL NUMERIC (X'F0' – X'F9' WITH TRAILING MINUS OR CR)
- ALL PACKED FIELD

ANY COMBINATION OF ABOVE

FIELD VERIFY

```
DFHBIF          TYPE=FVERIFY
               [,FIELD=symbolic address]
               [,LENGTH= { symbolic address } ]
               [         { numeric value    } ]
               [,ALPHA=symbolic address]
               [,NUMERIC=symbolic address]
               [,PACKED=symbolic address]
```

FIELD DE-EDIT

STRIP NON-NUMERIC CHARACTERS FROM A FIELD

DETECTS TRAILING '–' OR 'CR' AND PLACES NEGATIVE ZONE OVER LAST DIGIT

RIGHT JUSTIFIES REMAINING DIGITS

ZERO PADDING TO LEFT

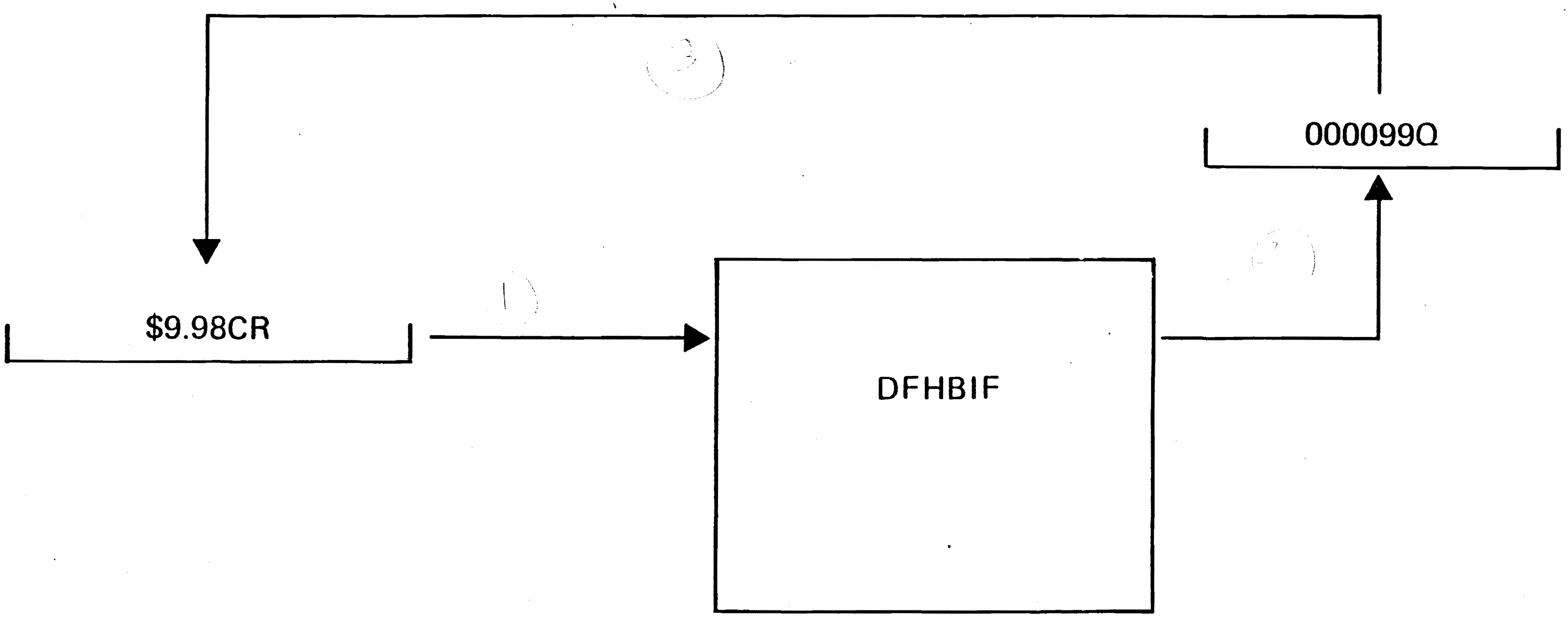
'A' TO 'F' ZONE ALLOWED ON RIGHTMOST BYTE

FIELD EDIT

```
DFHBIF          TYPE=DEEDIT
               [,FIELD=symbolic address
               [,LENGTH= { symbolic address } ]
               [         { numeric value    } ]
```

FIELD DE-EDIT

```
DFHBIF   TYPE=DEEDIT,
         FIELD=
         LENGTH=
```



RESULTANT VALUE RETURNED IN INPUT FIELD

BIT MANIPULATION

```
DFHBIF        TYPE= ( BITSETON   )
                    ) BITSETOFF  (
                    { BITFLIP    }
                    ( BITEST     )
             [,FIELD=symbolic address]
             [,BIT= { symbolic address }]
             [      { value            }]
             [,BITON=symbolic address]
             [,BITOFF=symbolic address]
```

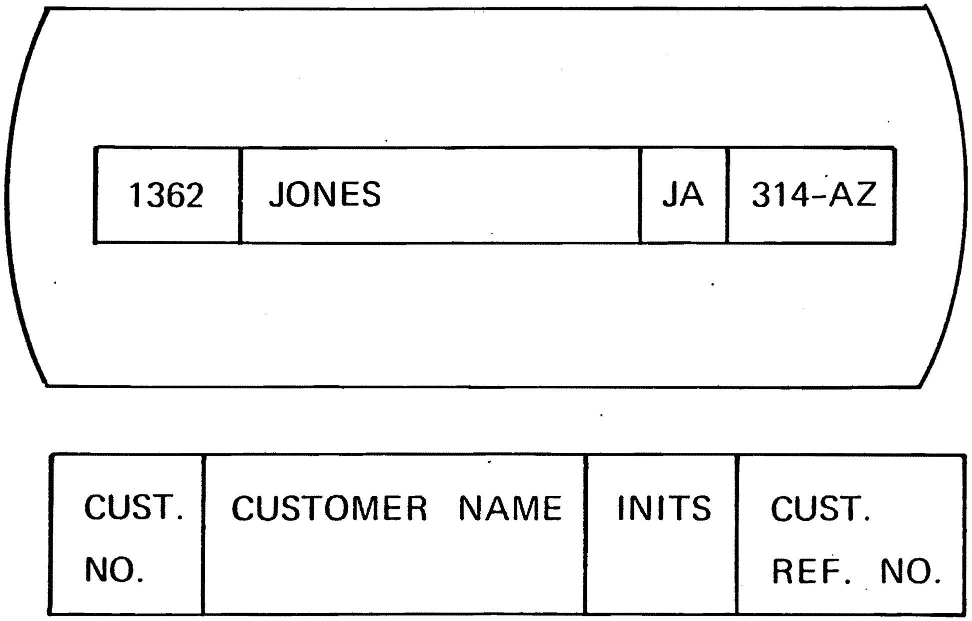
INPUT FORMATTING

ALLOWS FLEXIBLE INPUT FORMAT TO BE USED BY TERMINAL OPERATOR

FIXED FORMAT	JONES J A
POSITIONAL	JONES,J,A
KEYWORD	LAST=JONES,FI=J,MI=A

PROGRAM WORKS WITH FIXED FORMAT

FIXED FORMAT



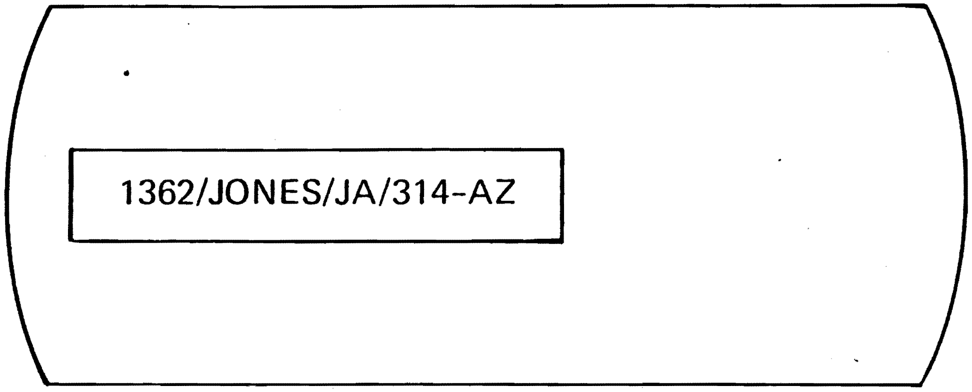
1362	JONES	JA	314-AZ

FORMAT ENTERED AT TERMINAL

CUST. NO.	CUSTOMER NAME	INITS	CUST. REF. NO.

FORMAT PRESENTED TO PROGRAM

POSITIONAL FORMAT



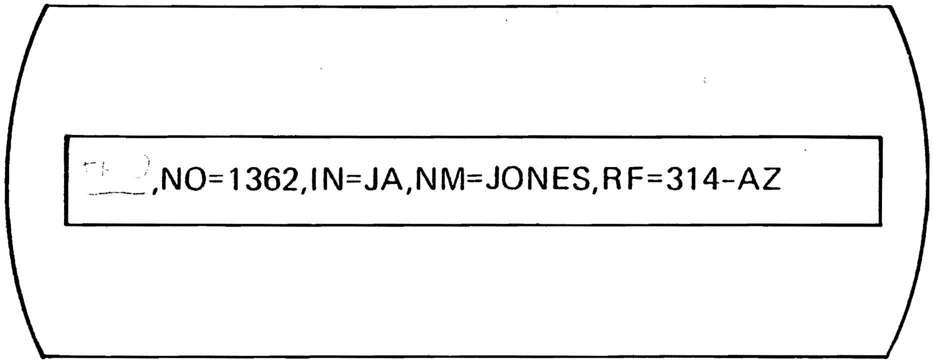
1362/JONES/JA/314-AZ

FORMAT ENTERED AT TERMINAL

CUST. NO.	CUSTOMER NAME	INITS	CUST. REF. NO.

AFTER PROCESSING BY CICS/VS INPUT FORMATTING

KEYWORD FORMAT



[illegible],NO=1362,IN=JA,NM=JONES,RF=314-AZ

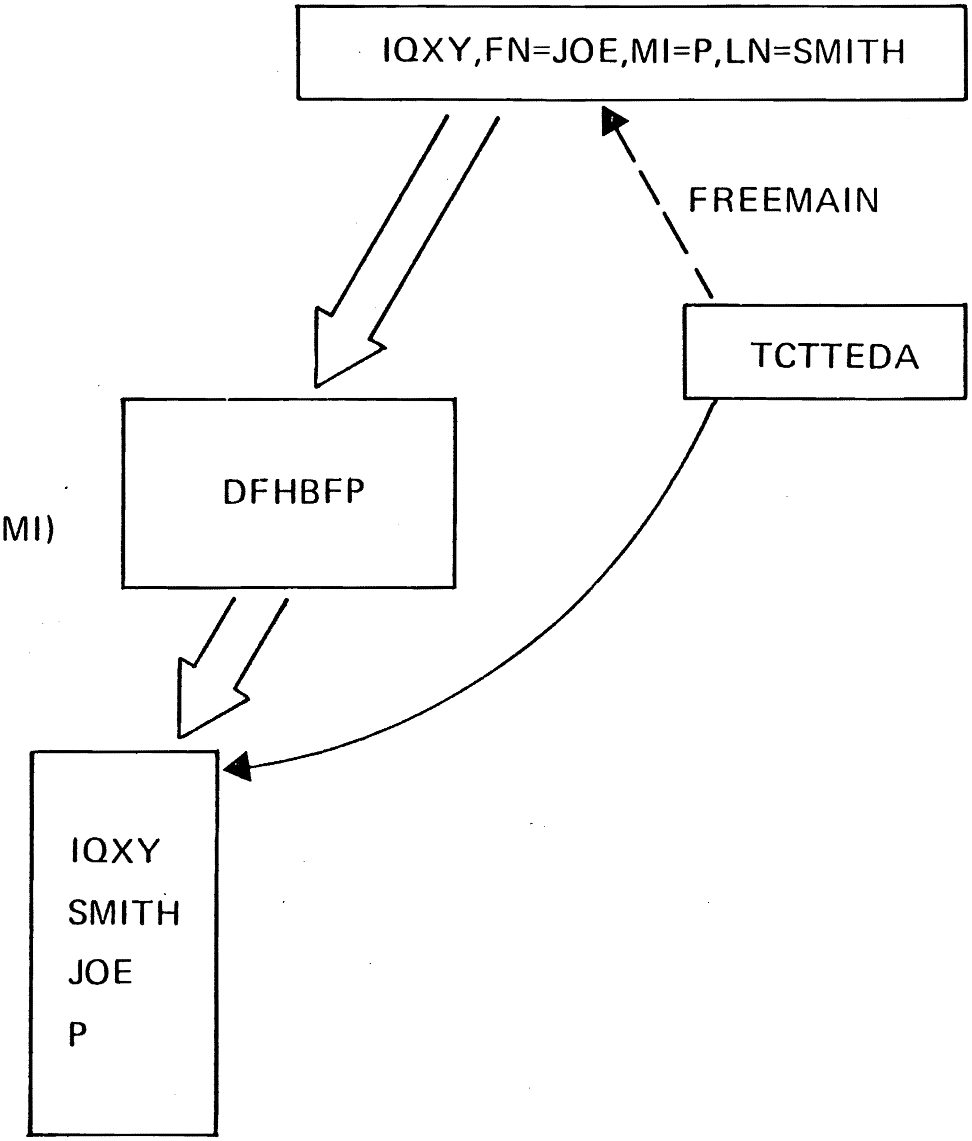
FORMAT ENTERED AT TERMINAL

CUST. NO.	CUSTOMER NAME	INITS	CUST. REF. NO.

AFTER PROCESSING BY CICS/VS
INPUT FORMATTING

INPUT FORMATTING

```
DFHBIF  TYPE  =  DEFLDNM,
        NAMES  =  (TRN,LN,FN,MI)


DFHBIF  TYPE  =  INFORMAT,
        FIELDS  =  (TRNCD,LAST,FIRST,MI)


DFHTIOA
          TRNCD
          LAST
          FIRST
          MI
```



IQXY,FN=JOE,MI=P,LN=SMITH

FREEMAIN

TCTTEDA

DFHBFP

IQXY
SMITH
JOE
P

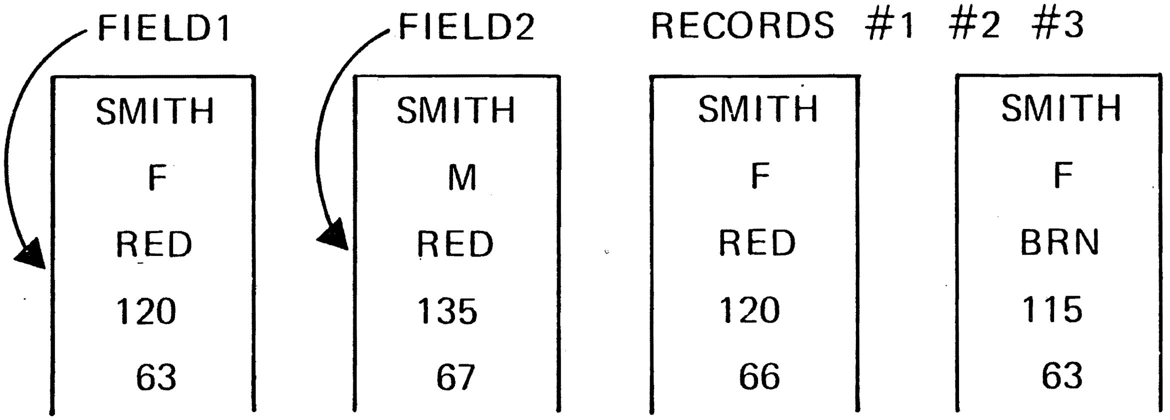
INPUT FORMATTING

```
DFHBIF        TYPE=DEFLDNM
             ,NAMES=(keyword[,keyword,...])
             ,LABEL=symbolic address

DFHBIF        TYPE=INFORMAT
             ,FIELDS=(symbolic address [,symbolic address,...])
             [,NAMES=   {symbolic address}]
             [          {YES             }]
             [,LENGTH=   {symbolic address}]
             [           {numeric value   }]
             [,ERROR=symbolic address]
```

WEIGHTED RETRIEVAL

FIELD1	*MATCH*	*NOMATCH*	*RANGE*
(NAME,20,C)	140	60	–
(SEX,1,C)	10	50	–
(HAIR,3,C)	30	5	–
(WGT,3,Z)	30	10	(P,15)
(HGT,2,Z)	50	20	(U,2)

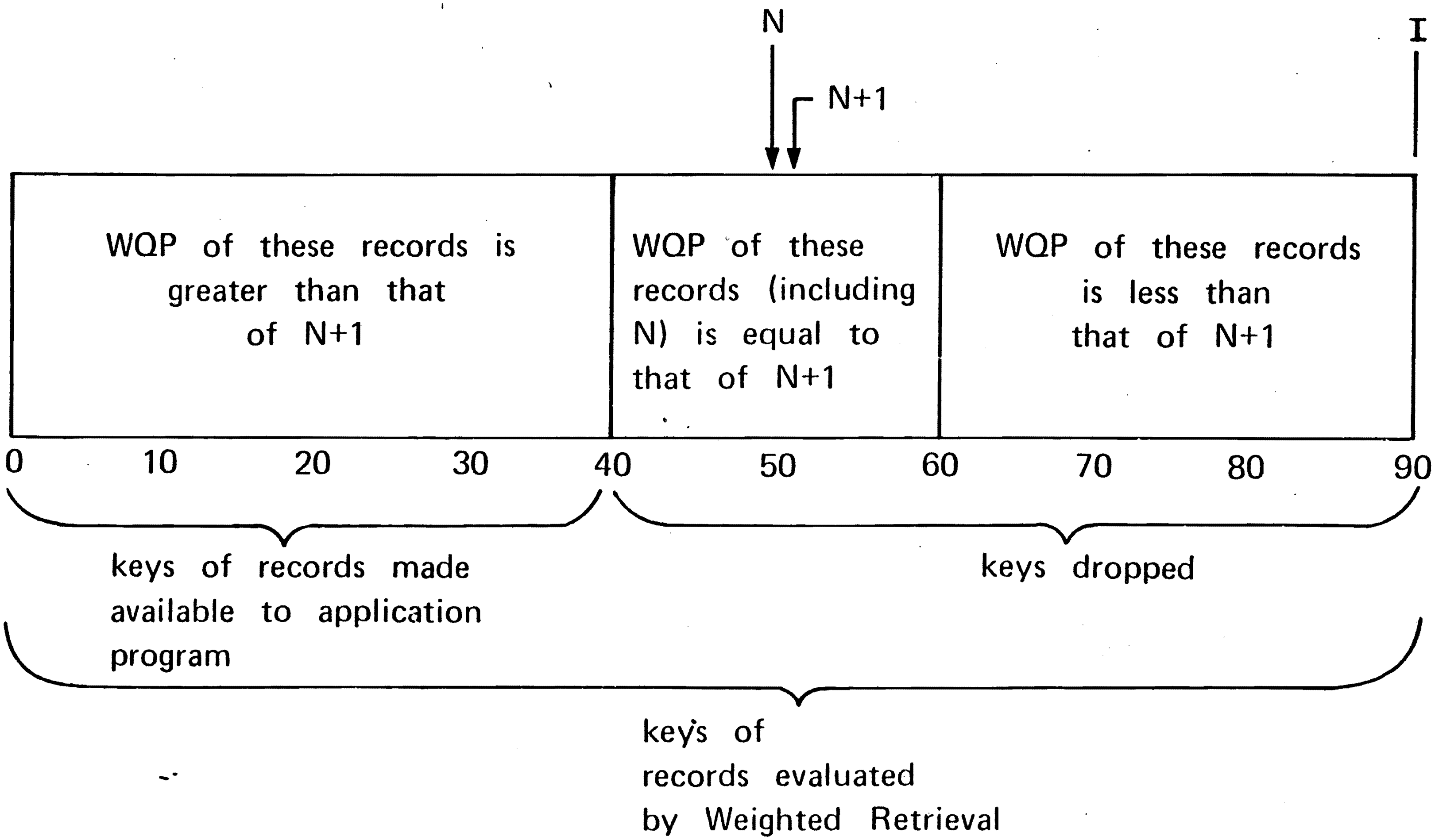
TOTAL COUNTER VALUE 260



	FIELD1	FIELD2	RECORDS #1 #2 #3	
	SMITH	SMITH	SMITH	SMITH
	F	M	F	F
	RED	RED	RED	BRN
	120	135	120	115
	63	67	66	63
ACCEPTABILITY		50%	73%	87%
$\frac{\text{MATCH - NOMATCH}}{\text{TOTAL}}$		130/260	190/260	225/260

V.8.27

WEIGHTED RETRIEVAL



WQP of these records is greater than that of N+1

WQP of these records (including N) is equal to that of N+1

WQP of these records is less than that of N+1

keys of records made available to application program

keys dropped

key's of records evaluated by Weighted Retrieval

USING WEIGHTED RETRIEVAL

DEFINE VSAM WORK AREA

COPY DFHVSWA

DEFINE VSAM RECORD DESCRIPTION

USER DSECT

INITIATE RETRIEVAL FUNCTION

DFHBIF TYPE = WTRETST

DEFINE SELECTION CRITERIA

DFHBIF TYPE = WTRTPARM

RETRIEVE SELECTED RECORDS

DFHBIF TYPE = WTRETGET

RELEASE ACQUIRED STORAGE

DFHBIF TYPE = WTRETREL

WEIGHTED RETRIEVAL

```
DFHBIF          TYPE=WTRETST
               [,DATASET=symbolic name]
               [,RDIDADR=symbolic address]
               ┌,INPUTNO=  ⎧symbolic address⎫┐
               │           ⎨numeric value   ⎬│
               └           ⎩YES             ⎭┘
               ┌,INPUTST=  ⎧symbolic address⎫┐
               │           ⎨numeric value   ⎬│
               └           ⎩YES             ⎭┘
               [,INPUTPC=( [suboperand1][,suboperand2]) ]
               ┌,NRECDS=   ⎧symbolic address⎫┐
               │           ⎨numeric value   ⎬│
               └           ⎩YES             ⎭┘
               [,NORESP=symbolic address]
               [,DSIDER=symbolic address]
               [,NOTOPEN=symbolic address]
               [,NOTFND=symbolic address]
               [,INVREQ=symbolic address]
               [,IOERROR=symbolic address]
               [,OFLOW=symbolic address]
               [,ILLOGIC=symbolic address]
```

WEIGHTED RETRIEVAL

```
DFHBIF        TYPE=WTRTPARM
             [,FIELD1=( [symbolic address][,numeric value][,char]) ]
             [,FIELD2=( [symbolic address1][,symbolic address2]) ]
             [,NULL=  ( symbolic address  ) ]
             [        { character value   } ]
             [        ( YES               ) ]
             [,MATCH=    { symbolic address } ]
             [           { numeric value    } ]
             [,NOMATCH=    { symbolic address } ]
             [             { numeric value    } ]
             [,RANGE=(suboperand1,suboperand2[,suboperand3]) ]
```

RANGE	VALUE	EQUAL
UNITS		
(U, 5)	165	160 to 170
(U, 5, 10)		155 to 170
PERCENT		
(P, 20)	165	132 to 198
VALUE		
(V, 190, 160)		160 to 190

DFHBIF TYPE = WTRETGET

AFTER WTRETST

TCAWRAA → VSWA

AFTER WTRETGET

VSWAREA → RECORD

VSWALEN CONTAINS LENGTH

WEIGHTED RETRIEVAL

```
DFHBIF          TYPE=WTRETGET
               [,NORESP=symbolic address]
               [,ENDFILE=symbolic address]
               [,NOTOPEN=symbolic address]
               [,NOTFND=symbolic address]
               [,INVREQ=symbolic address]
               [,IOERROR=symbolic address]
               [,OFLOW=symbolic address]
               [,ILLOGIC=symbolic address]
```

WEIGHTED RETRIEVAL

```
DFHBIF          TYPE=WTRETCHK
               [,NORESP=symbolic address]
               [,DSIDER=symbolic address]
               [,NOTOPEN=symbolic address]
               [,NOTFND=symbolic address]
               [,INVREQ=symbolic address]
               [,ENDFILE=symbolic address]
               [,IOERROR=symbolic address]
               [,OFLOW=symbolic address]
               [,ILLOGIC=symbolic address]
```

WEIGHTED RETRIEVAL

```
DFHBIF        TYPE=WTRETREL
             [,NORESP=symbolic address]
             [,INVREQ=symbolic address]
             [,ILLOGIC=symbolic address]
```

KC

TASK MANAGEMENT

MULTITASKING

TASK ORIGINATION PCT

TASK TERMINATION

STALL PROTECTION

RUNAWAY TASK PROTECTION

TASK SYNCHRONIZATION

RESOURCE SYNCHRONIZATION

PRIORITY CHANGE

TASK CONTROL

CREATE A NEW TASK (ATTACH) [illegible]

[illegible] → TERMINAL ORIENTED

- ONLY ONE TASK PER TERMINAL
- PRIORITY CALCULATED
- TERMINAL NOTIFIED OF ABEND

[illegible]

→ NON TERMINAL ORIENTED

- FACILITY MUST BE PROVIDED
- PRIORITY FROM PCT

"ORIGINATING" TASK NOT NOTIFIED OF APCT or OTHER ABENDS

(not [illegible])

K

K

TASK CONTROL

```
DFHKC          TYPE=ATTACH
               [,FCADDR=symbolic  address]
               [,TRANSID=name]
```

TASK CONTROL

```
DFHKC        TYPE=CHAP
             [,PRTY=priority value]
```

Task [illegible]

RELINQUISH CONTROL TO HIGHER PRIORITY TASK

- LONG RUNNING TASK — LENGTHY CALCULATIONS

SYNCHRONIZE A TASK WITH THE COMPLETION OF EVENT

- SINGLE EVENT = ECB
- MULTIPLE EVENTS
- FOUR BYTE ECB — OPERATING SYSTEM COMPATIBLE

TASK CONTROL

```
DFHKC          TYPE=WAIT
              ,DCI= ( SINGLE )
                    { LIST   }
                    ( DISPat )
              [,ECADDR=symbolic  address]
```

[illegible] Serialization

TASK CONTROL

```
DFHKC          TYPE=ENQ
              [,QARGADR=symbolic address]
              [,QARGLNG=number]

DFHKC          TYPE=DEQ
              [,QARGADR=symbolic address]
              [,QARGLNG=number]
```

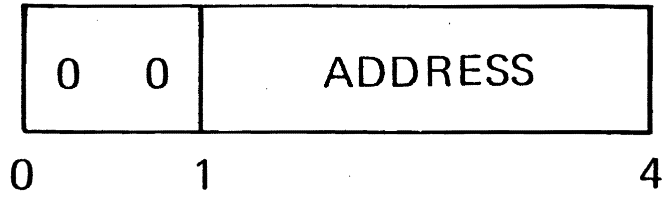
✳ possibility for Deadly Embrace
[Detection
Convention

(CICS [illegible])

TASK CONTROL

STORAGE ADDRESS

TCATCQA



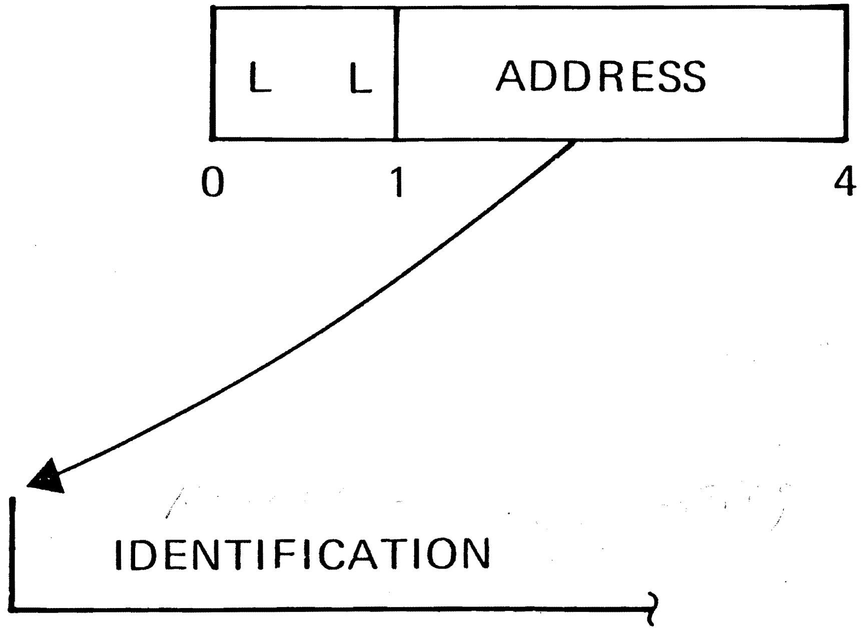
0 0

ADDRESS

0 1 4

SYMBOLIC NAME

TCATCQA



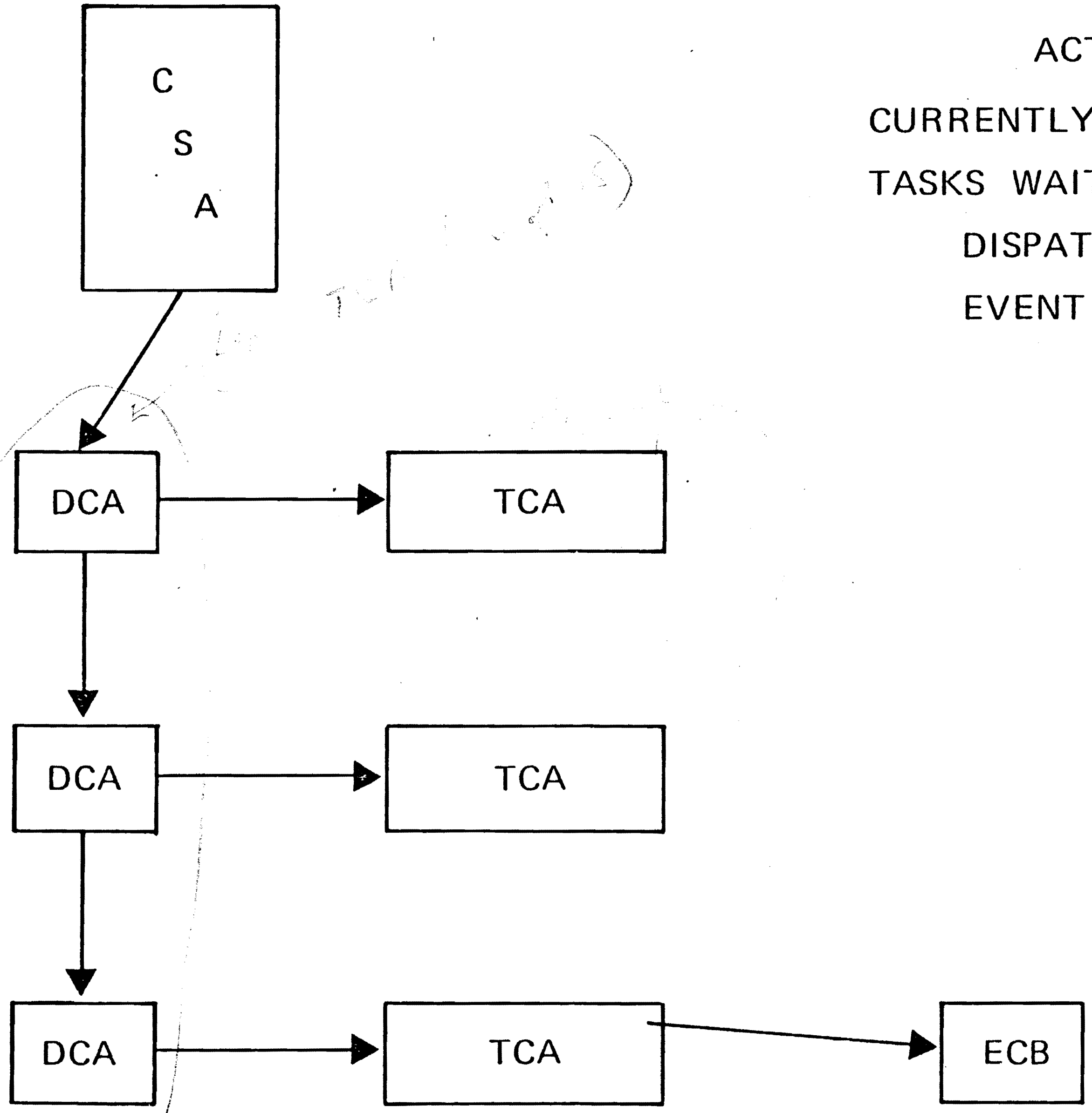
L L

ADDRESS

0 1 4

IDENTIFICATION

ACTIVE DCA CHAIN



ACTIVE TASKS

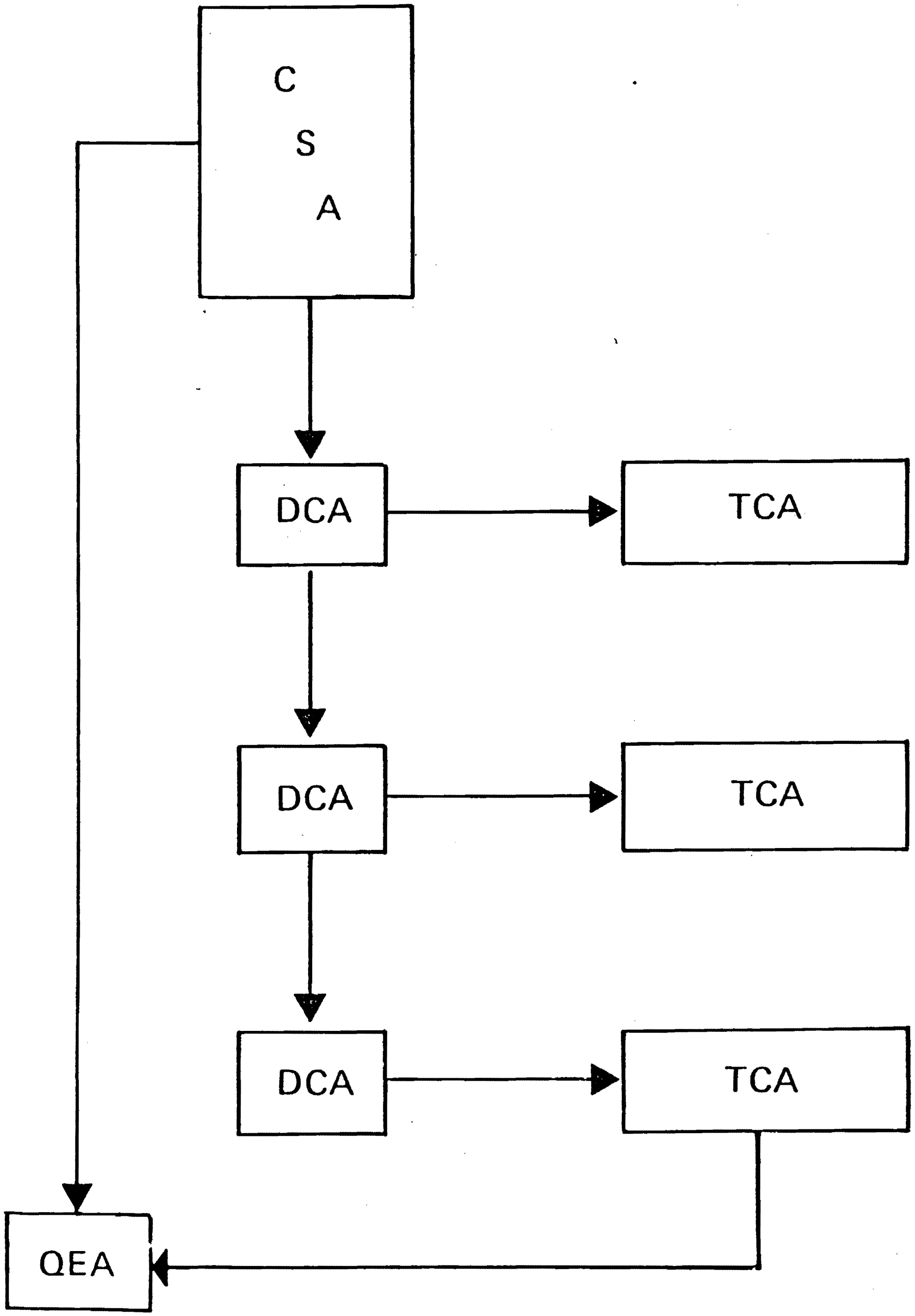
CURRENTLY DISPATCHED TASK

TASKS WAITING FOR – –

DISPATCHING

EVENT COMPLETION

SUSPENDED DCA CHAIN

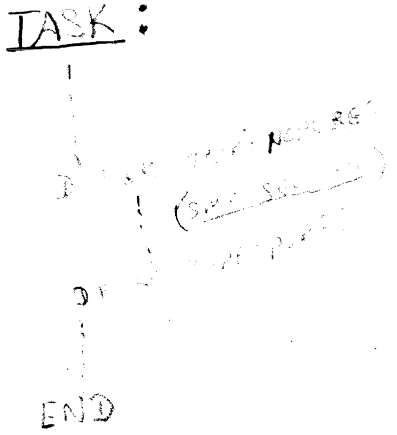


TASKS WAITING FOR – –

- TEMPORARY STORAGE
- MAIN STORAGE
- TERMINAL I/O
- INTERVAL CONTROL
- ENQUEUED RESOURCE

TASK CONTROL

DFHKC	TYPE=PURGE
DFHKC	TYPE=NOPURGE



TIME MANAGEMENT

CICS/VS EXIT TIME INTERVAL

SYSTEM STALL PROTECTION ([illegible] Not VS)

RUNAWAY TASK PROTECTION

TIME OF DAY

TIME DEPENDENT TRANSACTION SYNCHRONIZATION

WAIT–POST–CANCEL

AUTOMATIC TIME-ORDERED TRANSACTION INITIATION

INTERVAL CONTROL

```
DFHIC          TYPE=GETIME
              [,FORM=    { BINARY }]
              [          { PACKED }]
              [,TIMADR= { symbolic address }]
              [         { YES              }]
              [,NORESP=symbolic address]
              [,INVREQ=symbolic address]
              [,ERROR=symbolic address]
```

TIME ORIENTED TASK SYNCHRONIZATION

WAIT – DELAY THE PROCESSING OF A TASK

TIME OF DAY

RESUME AT SPECIFIED TIME

INTERVAL

RESUME UPON EXPIRATION

POST – SIGNAL EXPIRATION OF A SPECIFIED TIME

TIME OF DAY

"POST" ECB AT SPECIFIED TIME

INTERVAL

"POST" ECB UPON EXPIRATION

INTERVAL CONTROL

```
DFHIC      TYPE=WAIT
           [,INTRVAL= { numeric value } ]  |  [,TIME= { numeric value } ]
           [          { YES           } ]     [       { YES           } ]
           [,REQID= { name } ]
           [        { YES  } ]
           [,NORESP=symbolic address]
           [,INVREQ=symbolic address]
           [,EXPIRD=symbolic address]
           [,ERROR=symbolic address]
```

INTERVAL CONTROL

```
DFHIC        TYPE=POST
            ┌,INTRVAL= ⎧ numeric value ⎫┐ | ┌,TIME=⎧ numeric value ⎫┐
            └          ⎩ YES           ⎭┘   └      ⎩ YES           ⎭┘
            ┌,REQID=  ⎧ name ⎫┐
            └         ⎩ YES  ⎭┘
            [,NORESP=symbolic address]
            [,INVREQ=symbolic address]
            [,EXPIRD=symbolic address]
            [,ERROR=symbolic address]
```

AUTOMATIC TIME ORIENTED TASK INITIATION

REQUEST INITIATION OF ANOTHER TASK AT SOME FUTURE TIME

INITIATE – WITHOUT DATA

PUT – PASS DATA TO TASK USING TEMPORARY STORAGE

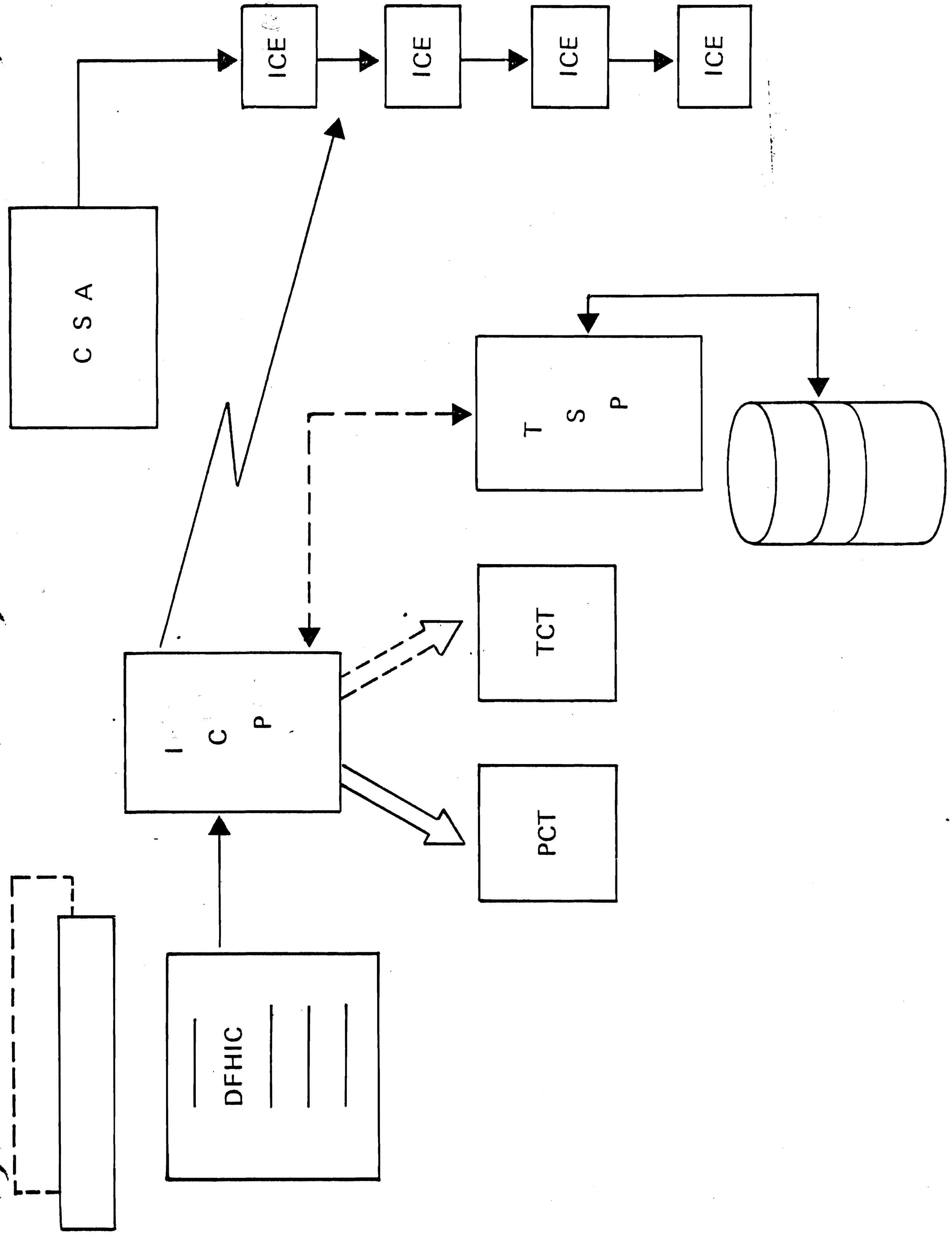
QUEUES TIME ORDERED REQUEST

VALIDATES REQUESTED FACILITIES

GET – RETREIVE EXPIRED TIME-ORDERED DATA

TERMINAL ORIENTED TASK

NON TERMINAL TASK


ICE
ICE
ICE
ICE
C S A
T S P
I C P
TCT
PCT
DFHIC

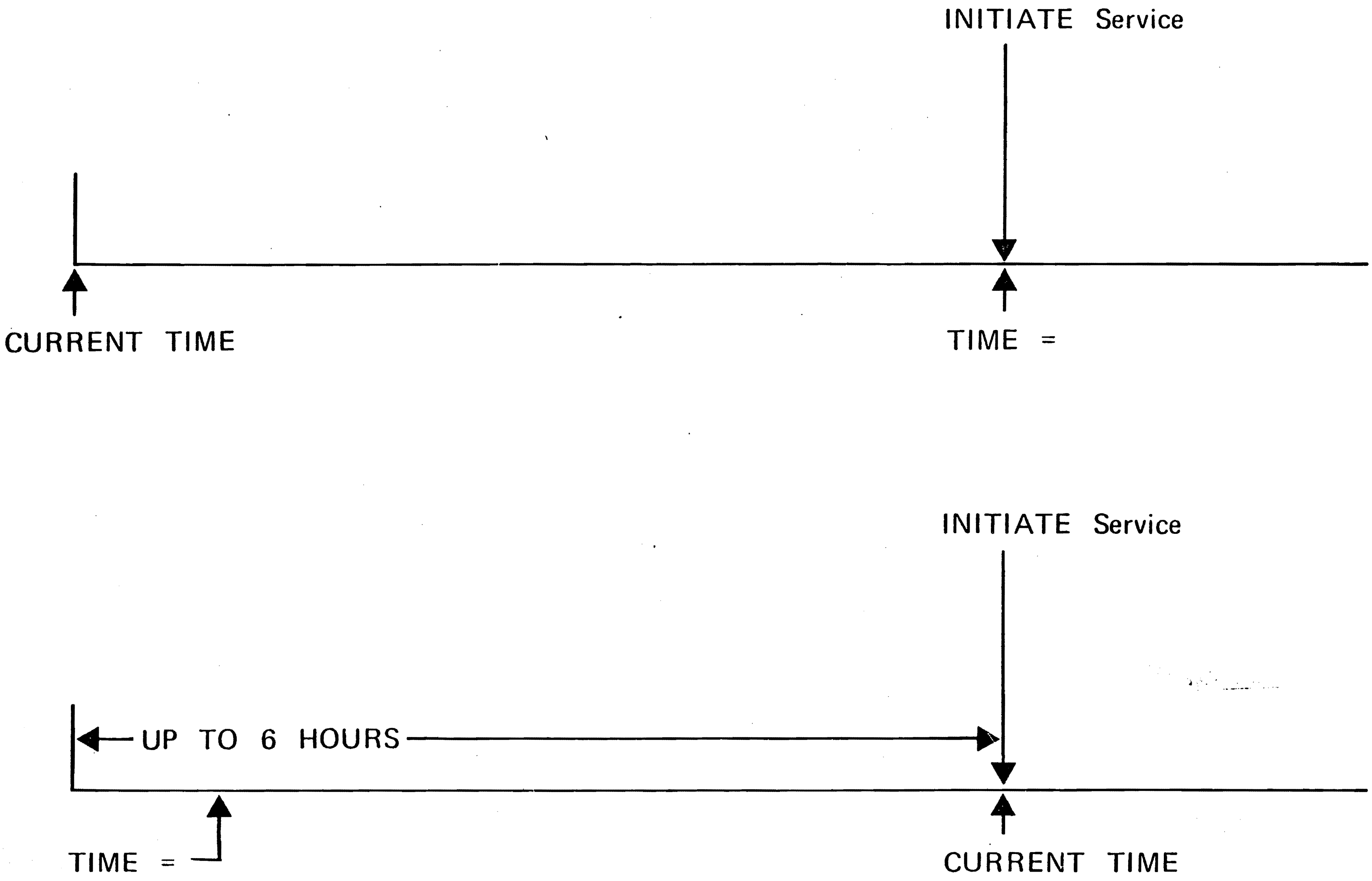
INTERVAL CONTROL

```
DFHIC        TYPE=INITIATE
             ┌,INTRVAL= ┌numeric value┐┐  |  ┌,TIME= ┌numeric value┐┐
             └          └YES          ┘┘     └       └YES          ┘┘
             ┌,REQID=   ┌name┐┐
             └          └YES ┘┘
             [,TRANSID=name]
             ┌,TRMIDNT= ┌name┐┐
             └          └YES ┘┘
             [,NORESP=symbolic address]
             [,INVREQ=symbolic address]
             [,TRNIDER=symbolic address]
             [,TRMIDER=symbolic address]
             [,ERROR=symbolic address]
```

INTERVAL CONTROL

```
DFHIC          TYPE=PUT
              [,INTRVAL= {numeric value}]  |  [,TIME= {numeric value}]
              [          {YES          }]     [       {YES          }]
              [,REQID= {name}]
              [        {YES }]
              [,TRANSID=name]
              [,TRMIDNT= {name}]
              [          {YES }]
              [,ICDADDR= {symbolic address}]
              [          {YES             }]
              [,NORESP=symbolic address]
              [,INVREQ=symbolic address]
              [,TRNIDER=symbolic address]
              [,TRMIDER=symbolic address]
              [,IOERROR=symbolic address]
              [,ERROR=symbolic address]
```

INTERVAL CONTROL



INTERVAL CONTROL

```
DFHIC          TYPE=GET
              ⎡,ICDADDR= ⎧symbolic address⎫⎤
              ⎣          ⎩YES             ⎭⎦
              [,NORESP=symbolic address]
              [,INVREQ=symbolic address]
              [,NOTFND=symbolic address]
              [,ENDDATA=symbolic address]
              [,IOERROR=symbolic address]
              [,TSINVLD=symbolic address]
              [,ERROR=symbolic address]
              [[illegible]]
```

INTERVAL CONTROL

```
DFHIC          TYPE=RETRY
              [,NORESP=symbolic address]
              [,INVREQ=symbolic address]
              [,NOTFND=symbolic address]
              [,IOERROR=symbolic address]
              [,ERROR=symbolic address]
```

INTERVAL CONTROL

```
DFHIC          TYPE=CANCEL
               [,REQID= { name }]
               [        { YES  }]
               [,NORESP=symbolic address]
               [,INVREQ=symbolic address]
               [,NOTFND=symbolic address]
               [,ERROR=symbolic address]
```

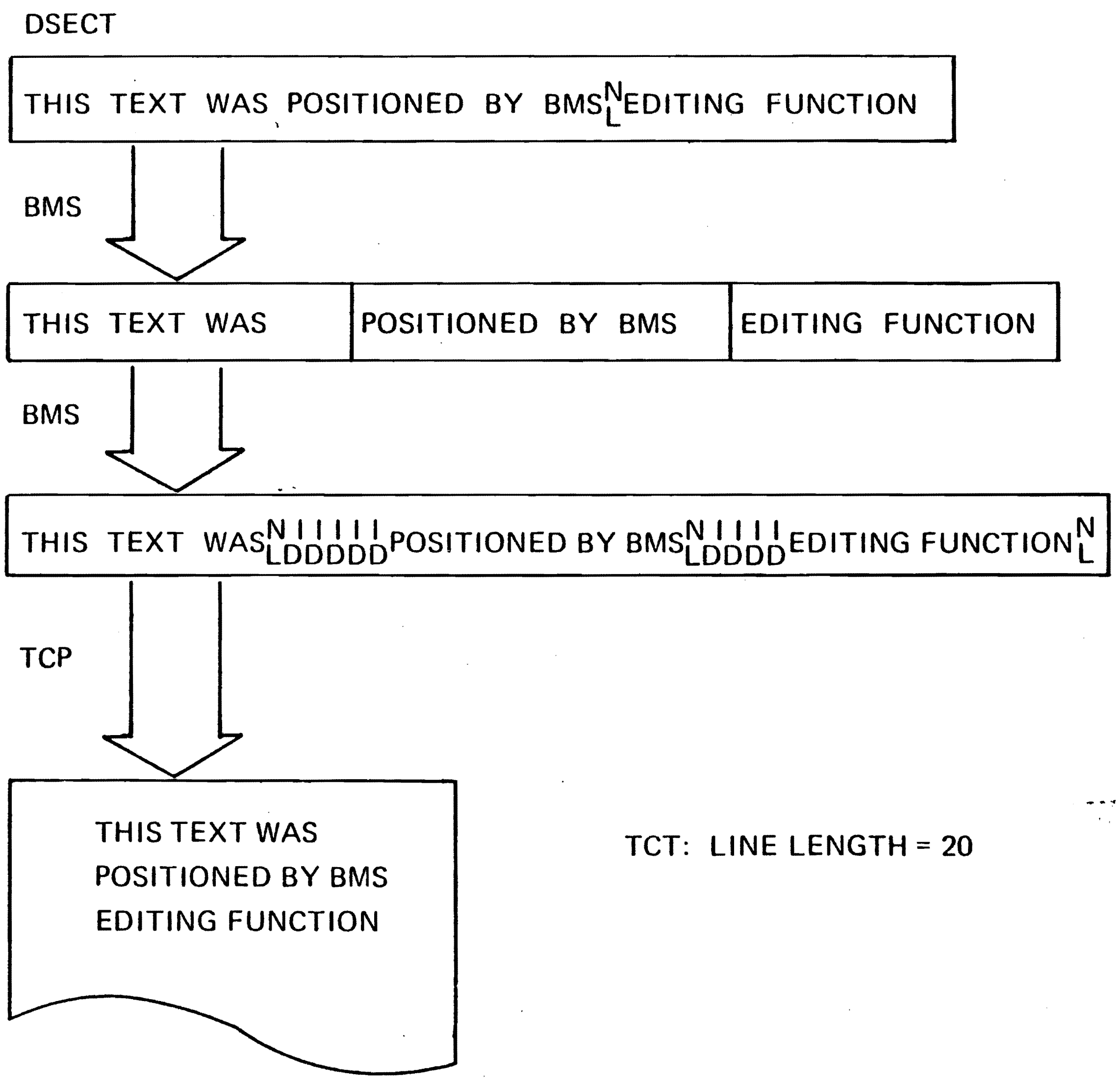
INTERVAL CONTROL

```
DFHIC          TYPE=CHECK
               [,NORESP=symbolic address]
               [,INVREQ=symbolic address]
               [,EXPIRD=symbolic address]
               [,TRNIDER=symbolic address]
               [,TRMIDER=symbolic address]
               [,NOTFND=symbolic address]
               [,ENDDATA=symbolic address]
               [,ERROR=symbolic address]
               [,TSINVLD=symbolic address]
```

MAPPING

TEXT DATA

TEXT DATA MAY BE HANDLED BY BMS WITH *NO* MAPS



DSECT

THIS TEXT WAS POSITIONED BY BMS$\substack{N\\L}$EDITING FUNCTION

BMS

THIS TEXT WAS	POSITIONED BY BMS	EDITING FUNCTION

BMS

THIS TEXT WAS$\substack{N\,I\,I\,I\,I\,I\\L\,D\,D\,D\,D\,D}$POSITIONED BY BMS$\substack{N\,I\,I\,I\,I\\L\,D\,D\,D\,D}$EDITING FUNCTION$\substack{N\\L}$

TCP

THIS TEXT WAS
POSITIONED BY BMS
EDITING FUNCTION

TCT: LINE LENGTH = 20

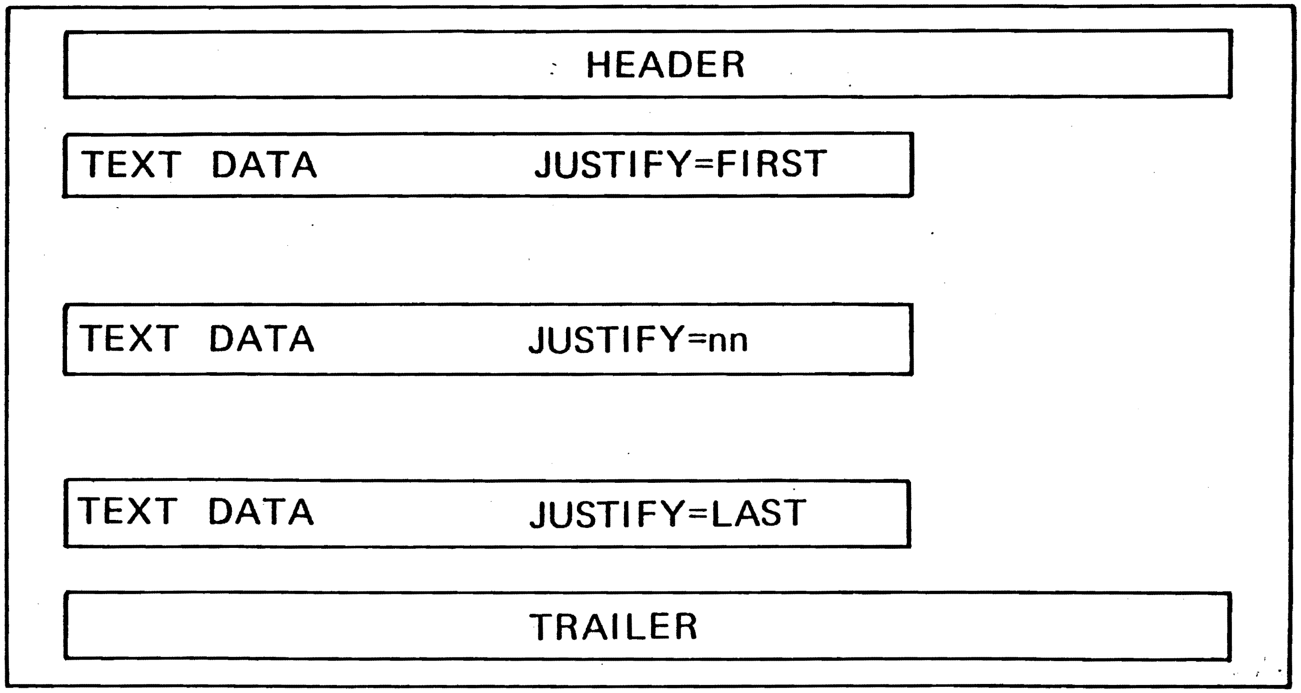
BASIC MAPPING

```
DFHBMS    TYPE=(TEXTBLD ┌, ( OUT     ) ┐ [,SAVE][,ERASE])
                        │  { STORE   } │
                        └  ( RETURN  ) ┘
          ┌,HEADER=  ( symbolic address ) ┐
          └          ( YES              ) ┘
          ┌,TRAILER= ( symbolic address ) ┐
          └          ( YES              ) ┘
          ┌,JUSTIFY= ( FIRST ) ┐
          │          { LAST  } │
          │          { nnn   } │
          └          ( YES   ) ┘
          ┌,CTRL=( [PRINT] ┌, ( L40    ) ┐[,FREEKB][,ALARM]) ┐
          │                │  { L64    } │                  │
          │                │  { L80    } │                  │
          └                └  ( HONEOM ) ┘                  ┘
                               ‾‾‾‾‾‾
          ┌,CURSOR=  ( number ) ┐
          └          ( YES    ) ┘
          [,WRBRK=symbolic address]
          [,NORESP=symbolic address]
          [,TSIOERR=symbolic address]
          [,INVREQ=symbolic address]
          [,RETPAGE=symbolic address]
          [,ERROR=symbolic address]
```

JUSTIFY

BMS POSITIONS TEXT JUSTIFIED FIRST, LAST, OR AT A SPECIFIED LINE NUMBER WITHIN A PAGE BUFFER.

PAGE



HEADER

TEXT DATA JUSTIFY=FIRST

TEXT DATA JUSTIFY=nn

TEXT DATA JUSTIFY=LAST

TRAILER

JUSTIFY=YES

LINE NUMBER	1-240	⟶ TCAMSJ
JUSTIFY FIRST	254	
JUSTIFY LAST	255	

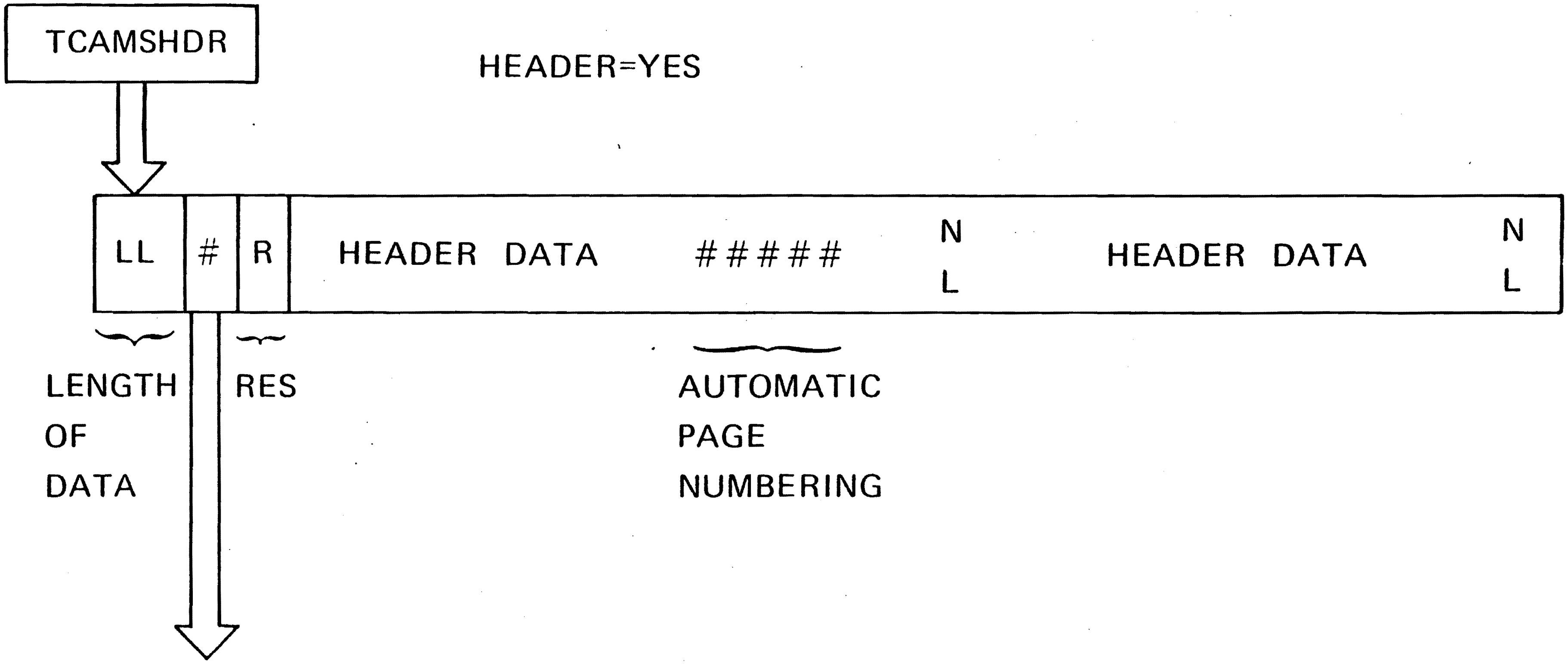
HEADER

TEXTBLD AUTOMATICALLY PLACES HEADER, IF SPECIFIED, AT BEGINNING OF EACH PAGE OF OUTPUT.

MULTIPLE LINE HEADERS CAN BE PROVIDED BY EMBEDDING NL (X'15') CHARACTERS IN HEADER DATA.

AUTOMATIC PAGE NUMBERING IS PROVIDED.

HEADER



PAGE NUMBER LOCATOR

B – AUTOMATIC PAGE NUMBERING NOT USED

N – ANY LOCATOR VALUE EXCEPT X'0C' X'15' X'17' X'26'
BMS SCANS HEADING DATA FOR THIS VALUE AND
REPLACES 1–5 BYTES WITH PAGE NUMBER.

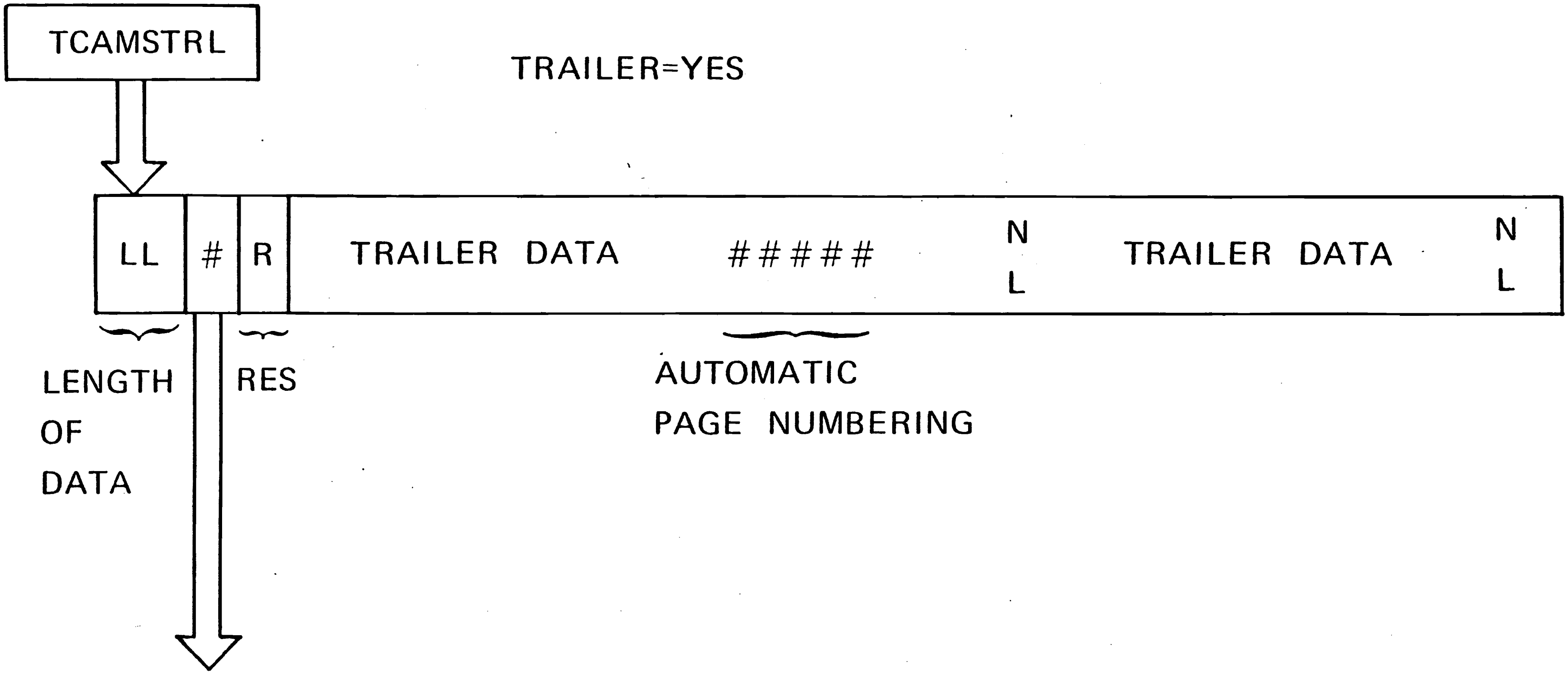
TRAILER

TEXTBLD AUTOMATICALLY PLACES TRAILER, IF SPECIFIED, AT THE BOTTOM OF EACH PAGE OF OUTPUT.

MULTIPLE LINE TRAILERS CAN BE PROVIDED BY EMBEDDING NL (X'15') CHARACTERS IN TRAILER DATA.

AUTOMATIC PAGE NUMBERING IS PROVIDED.

TRAILER



TCAMSTRL

TRAILER=YES

LL | # | R | TRAILER DATA ##### NL TRAILER DATA NL

LENGTH OF DATA

RES

AUTOMATIC PAGE NUMBERING

PAGE NUMBER LOCATOR

B – AUTOMATIC PAGE NUMBERING NOT USED

N – ANY LOCATOR VALUE EXCEPT X'0C' X'15' X'17' X'26'
BMS SCANS TRAILER DATA FOR THIS VALUE AND
REPLACES 1–5 BYTES WITH PAGE NUMBER.

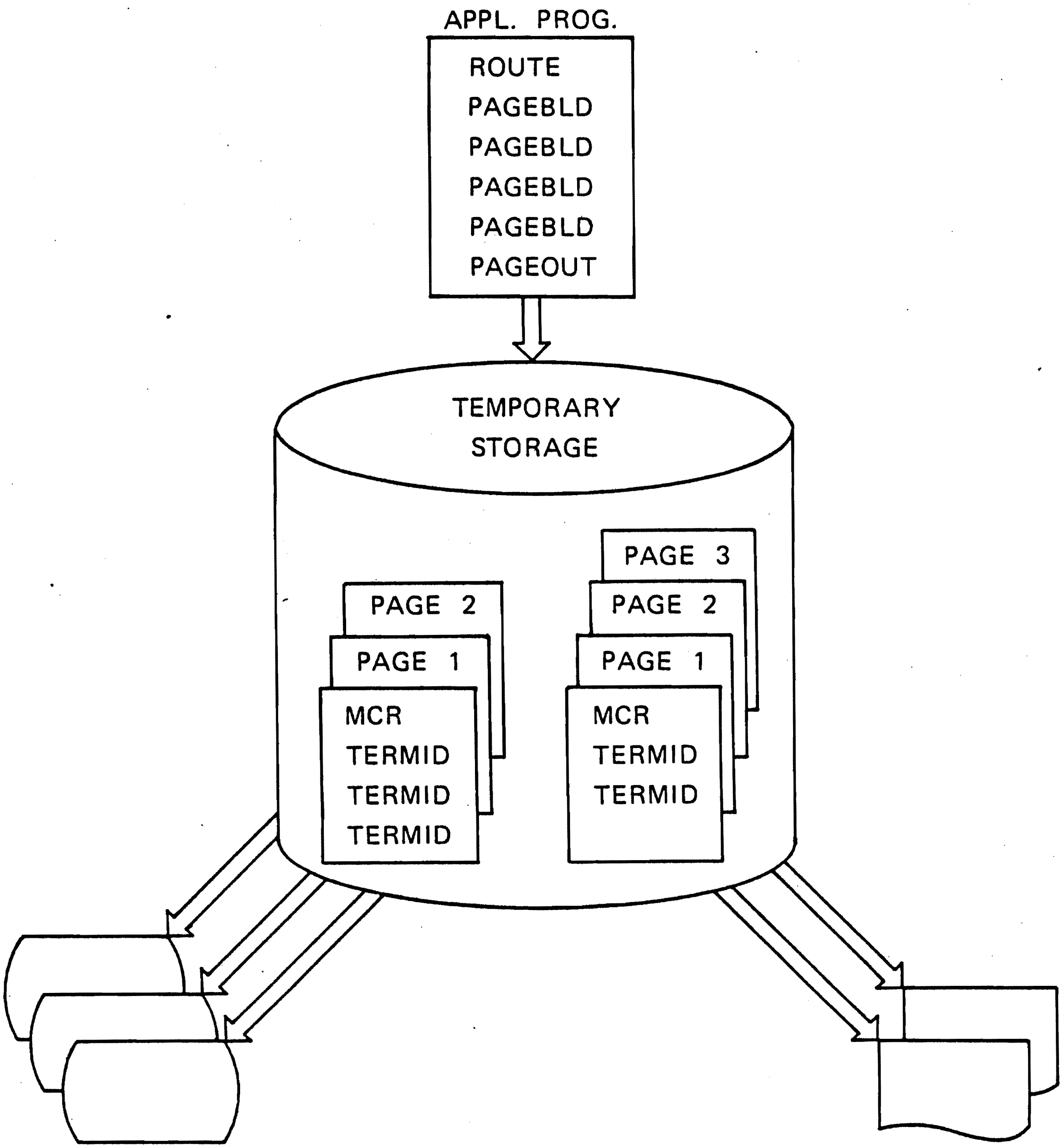
BASIC MAPPING

```
DFHBMS       TYPE=PAGEOUT
             [,CTRL= ( [ { PAGE     } ]   [, { RETAIN  } ] ) ]
             [         [ { AUTOPAGE } ]   [  { RELEASE } ]   ]
             [,TRAILER= { symbolic address } ]
             [          { YES              } ]
             [,TRANSID=transaction code]
             [,WRBRK= { symbolic address } ]
             [        { CURRENT          } ]
             [        { ALL              } ]
             [,EODPURG= { AUTO } ]
             [          { OPER } ]
             [,NORESP=symbolic address]
             [,TSIOERR=symbolic address]
             [,RETPAGE=symbolic address]
             [,ERROR=symbolic address]
```

ROUTING

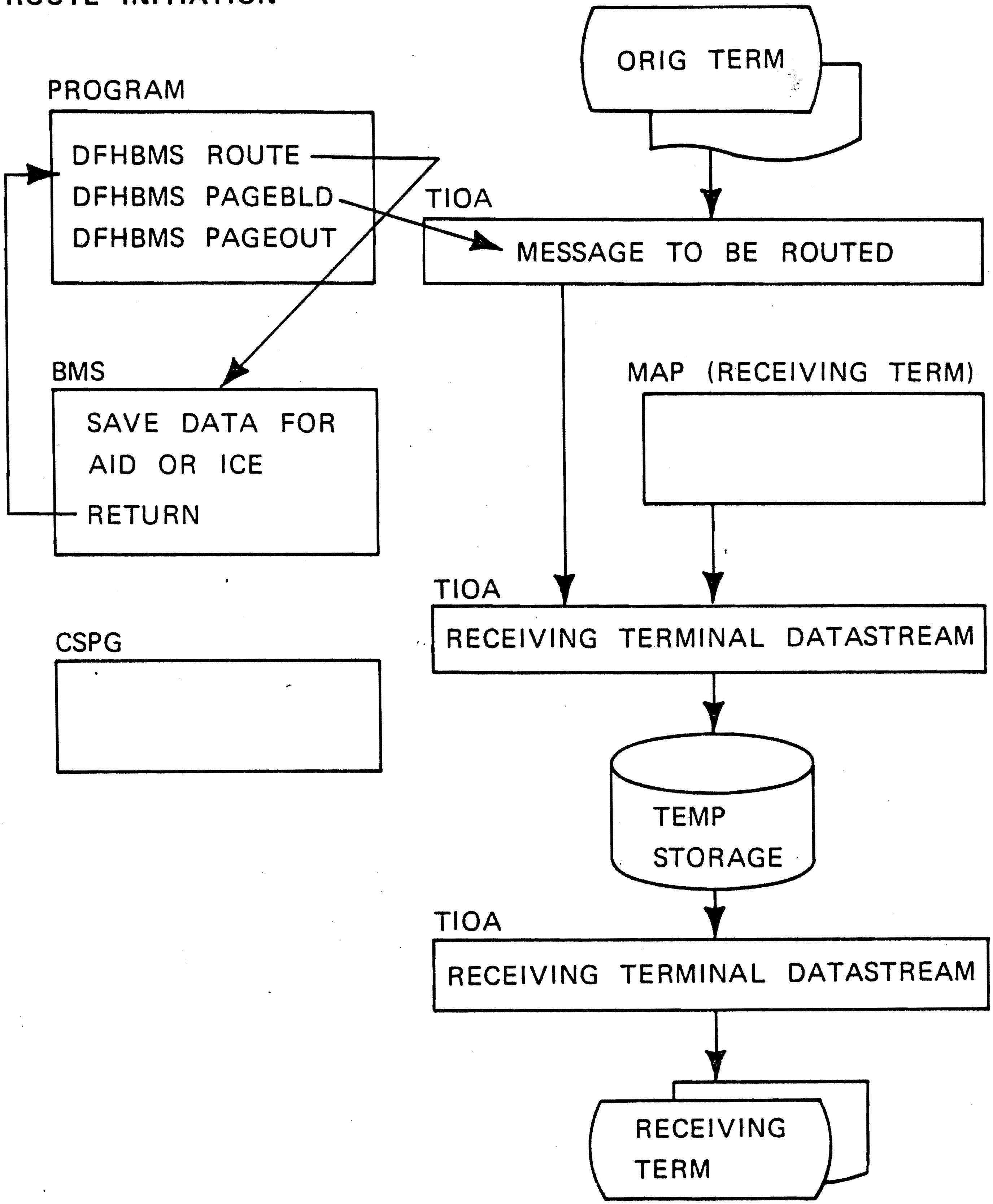
INTRODUCTION

ROUTING BUILDS A LOGICAL MESSAGE FOR EACH DIFFERENT TERMINAL TYPE. A LIST OF TERMINALS TO RECEIVE THE LOGICAL MESSAGE IS PLACED ON TEMPORARY STORAGE WITH TERMINAL TYPE PAGES.



ROUTING

ROUTE INITIATION



PROGRAM

DFHBMS ROUTE

DFHBMS PAGEBLD

DFHBMS PAGEOUT

BMS

SAVE DATA FOR AID OR ICE

RETURN

CSPG

ORIG TERM

TIOA

MESSAGE TO BE ROUTED

MAP (RECEIVING TERM)

TIOA

RECEIVING TERMINAL DATASTREAM

TEMP STORAGE

TIOA

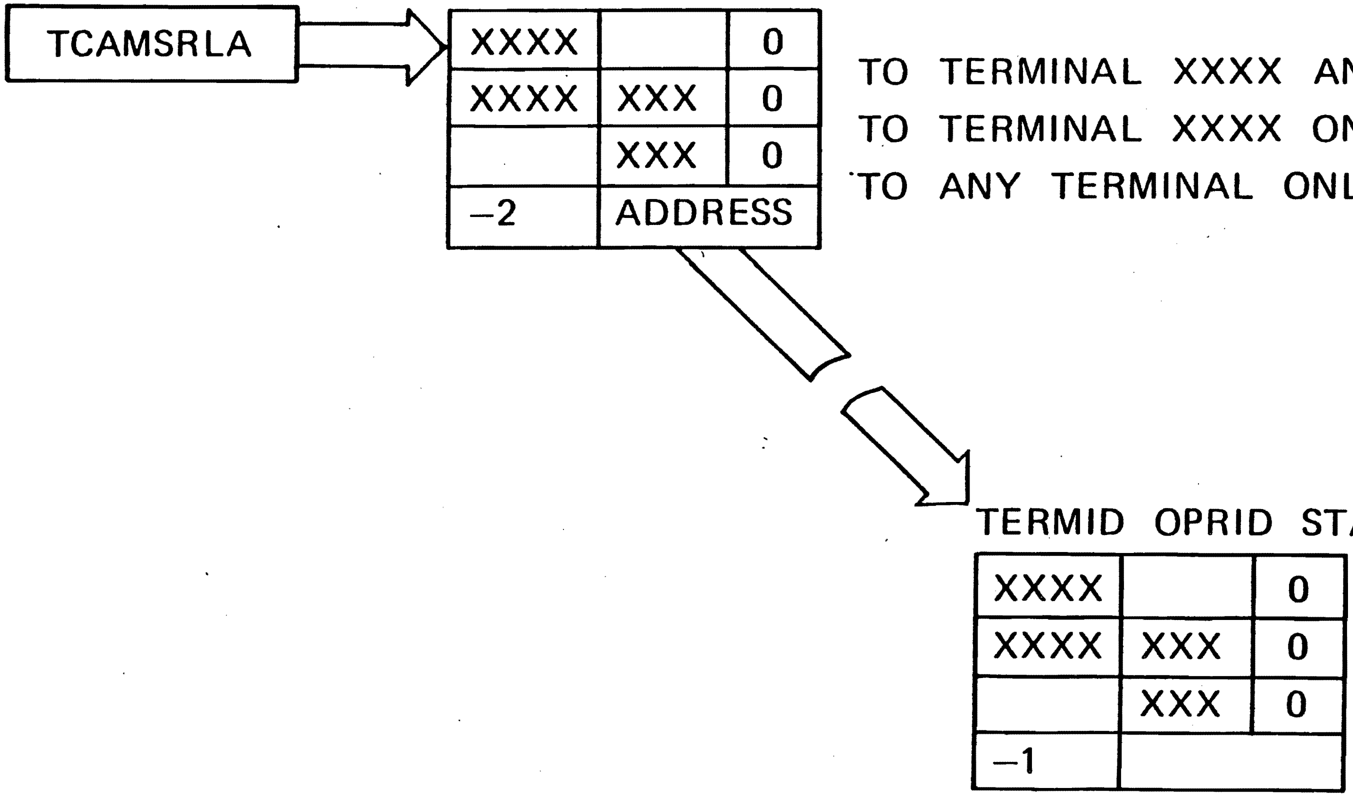
RECEIVING TERMINAL DATASTREAM

RECEIVING TERM

BASIC MAPPING

```
DFHBMS        TYPE=ROUTE
             ┌,LIST=    ⎧symbolic address⎫┐
             │          ⎨YES             ⎬│
             └          ⎩ALL             ⎭┘
             ┌,INTRVAL=⎧numeric value⎫┐  |  ┌,TIME= ⎧numeric value⎫┐
             └         ⎩YES          ⎭┘     └       ⎩YES          ⎭┘
             ┌,OPCLASS=⎧decimal value,...⎫┐
             └         ⎩YES              ⎭┘
             ┌,TITLE=  ⎧symbolic address⎫┐
             └         ⎩YES             ⎭┘
             ┌,ERRTERM=⎧termid⎫┐
             │         ⎨ORIG  ⎬│
             └         ⎩YES   ⎭┘
              [,NORESP=symbolic address]
              [,INVET=symbolic address]
              [,RTEFAIL=symbolic address]
              [,RTESOME=symbolic address]
              [,ERROR=symbolic address]
```

ROUTING



ROUTE LIST

TCAMSRLA

TERMID	OPRID	STATUS
XXXX		0
XXXX	XXX	0
	XXX	0
−2	ADDRESS	

TO TERMINAL XXXX ANY OPERATOR

TO TERMINAL XXXX ONLY OPERATOR XXX

TO ANY TERMINAL ONLY OPERATOR XXX

TERMID	OPRID	STATUS
XXXX		0
XXXX	XXX	0
	XXX	0
−1		

```
DFHURLDS (URLBAR)
   URLTRMID   DS   CL4    TERMINAL ID
              DS   CL2    RESERVED
   URLOPID    DS   CL3    OPERATOR ID
   URLTSF     DS   B      STATUS FLAG
              DS   CL6    RESERVED
```

ROUTING

STATUS FLAG

USER ROUTE LIST STATUS FLAG –

X'80'	ENTRY SKIPPED
X'40'	INVALID TERMINAL ID
X'20'	TERMINAL NOT SUPPORTED BY BMS
X'10'	OPERATOR NOT SIGNED ON
X'08'	OPERATOR SIGNED ON UNSUPPORTED TERMINAL

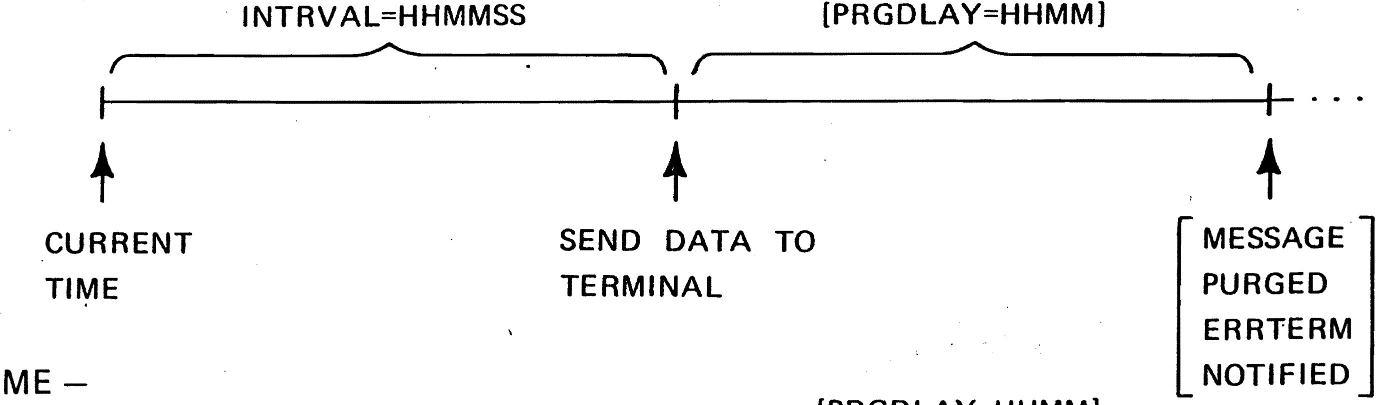
ROUTING

OPERATOR CLASS

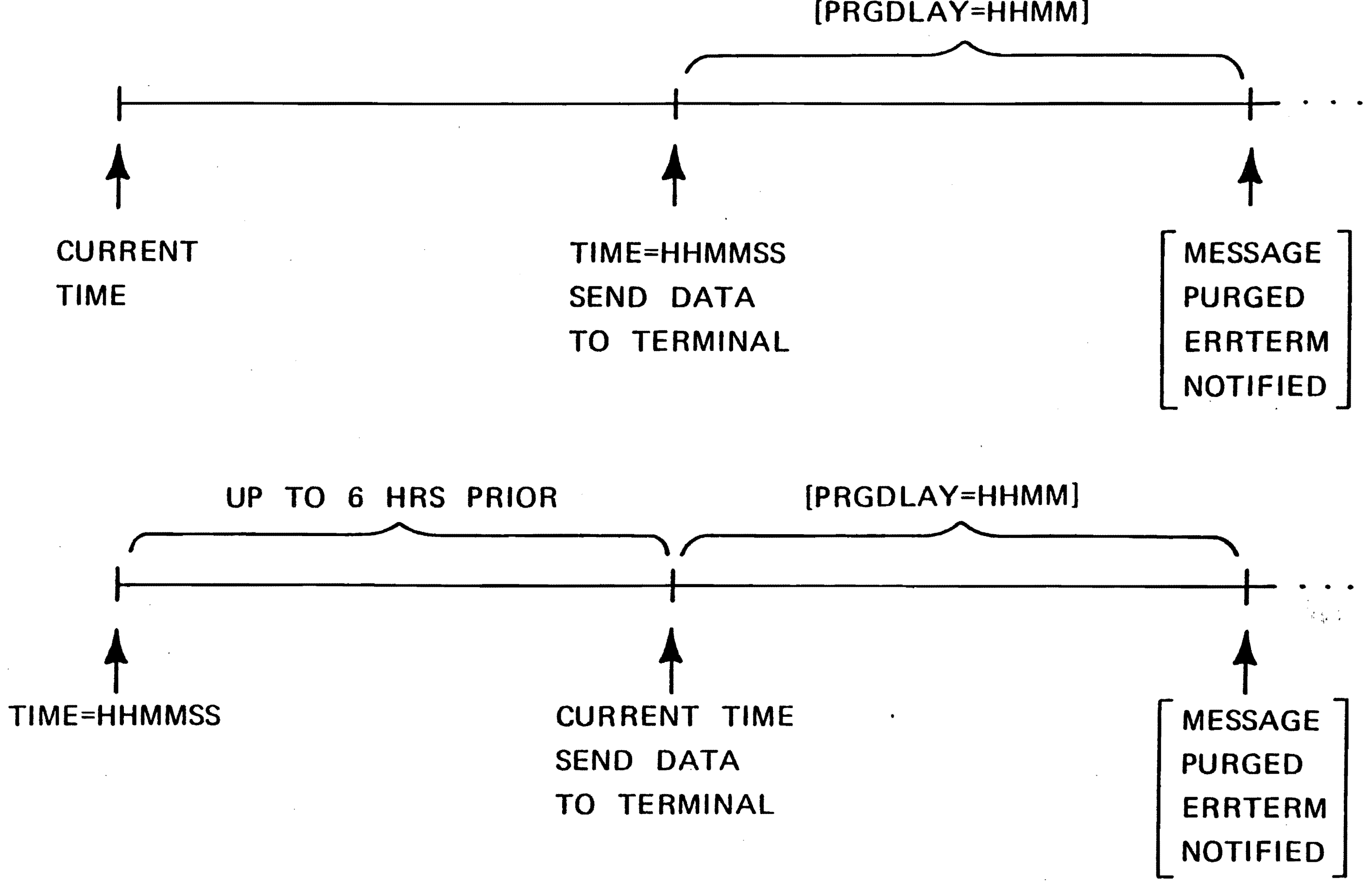
```
DFHBMS   TYPE=ROUTE,OPCLASS=N

DFHBMS   TYPE=ROUTE,OPCLASS=N,LIST=ALL

DFHBMS   TYPE=ROUTE,OPCLASS=N,LIST=SYBADDR
```

ROUTING

INTRVAL –

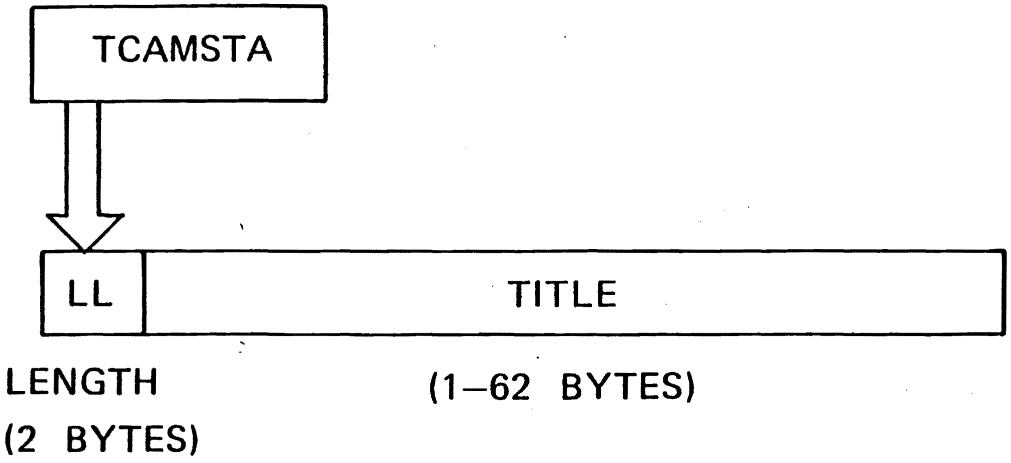


TIME –


[PRGDLAY=HHMM]
CURRENT TIME
TIME=HHMMSS SEND DATA TO TERMINAL
MESSAGE PURGED ERRTERM NOTIFIED
UP TO 6 HRS PRIOR
[PRGDLAY=HHMM]
TIME=HHMMSS
CURRENT TIME SEND DATA TO TERMINAL
MESSAGE PURGED ERRTERM NOTIFIED

ROUTING

TITLE



TCAMSTA

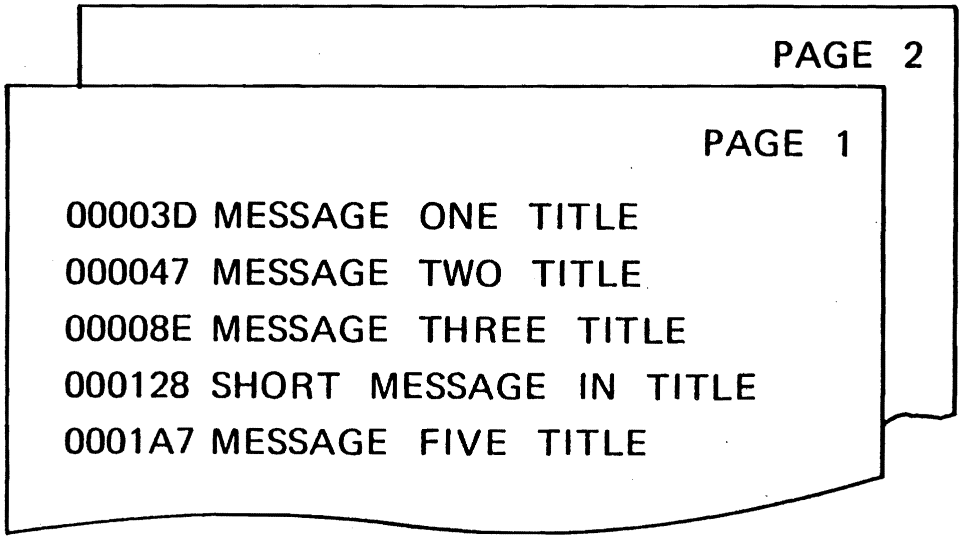
LL

TITLE

LENGTH
(2 BYTES)

(1–62 BYTES)

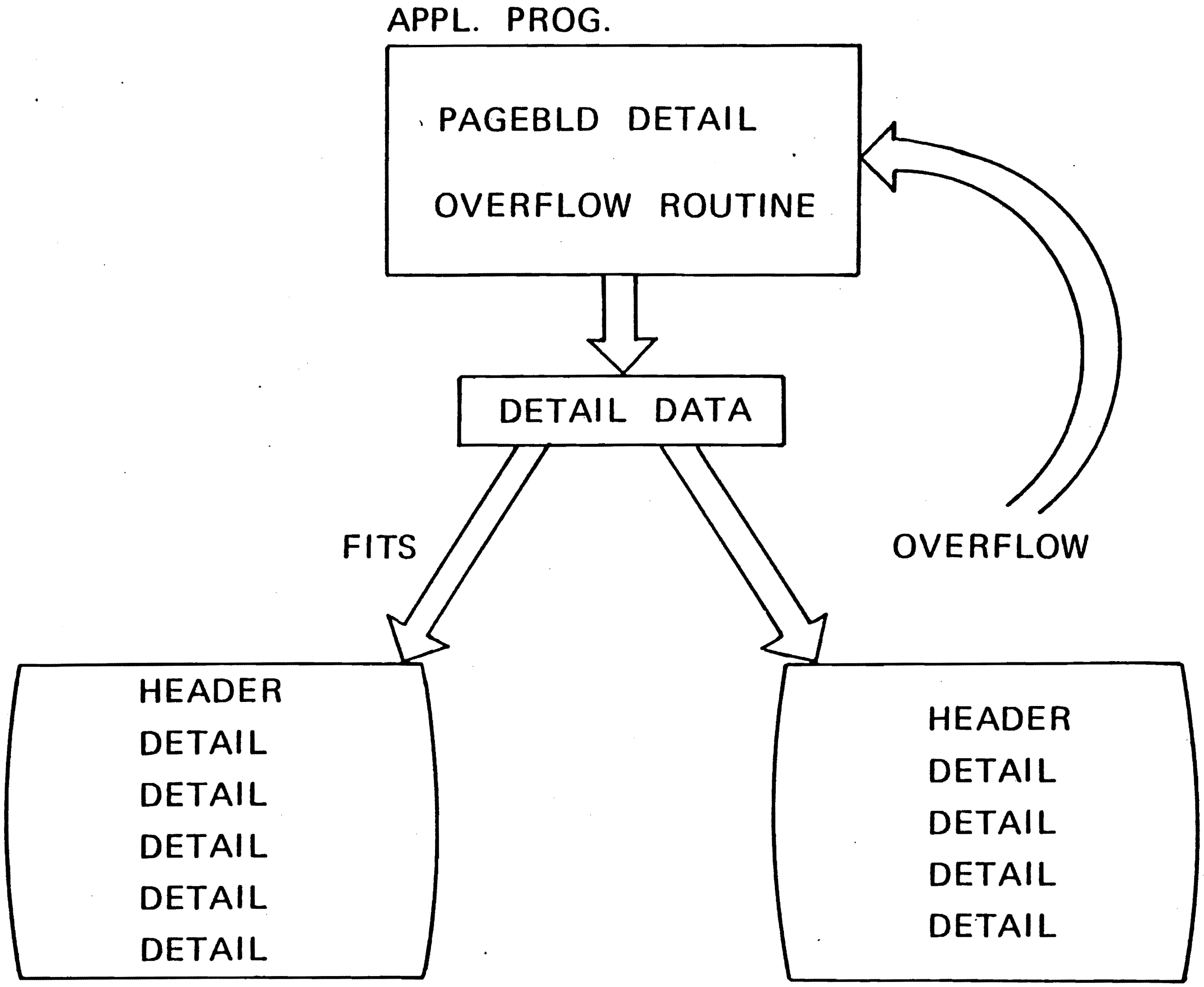
LENGTH INCLUDES 2 BYTE LENGTH FIELD.



```
                                    PAGE 2

                                 PAGE 1

00003D MESSAGE  ONE  TITLE
000047 MESSAGE  TWO  TITLE
00008E MESSAGE  THREE  TITLE
000128 SHORT  MESSAGE  IN  TITLE
0001A7 MESSAGE  FIVE  TITLE
```

ROUTING

OVERFLOW



APPL. PROG.

PAGEBLD DETAIL

OVERFLOW ROUTINE

DETAIL DATA

FITS

OVERFLOW

HEADER
DETAIL
DETAIL
DETAIL
DETAIL
DETAIL

HEADER
DETAIL
DETAIL
DETAIL
DETAIL

ROUTING

ROUTE	ROUTED LOGICAL MESSAGE
PAGEBLD	
PAGEBLD	
PAGEBLD	
PAGEOUT	
TEXTBLD	DIRECT LOGICAL MESSAGE
TEXTBLD	
TEXTBLD	
TEXTBLD	
ROUTE	ROUTED LOGICAL MESSAGE
PAGEBLD	
PAGEBLD	
PAGEBLD	
RETURN	

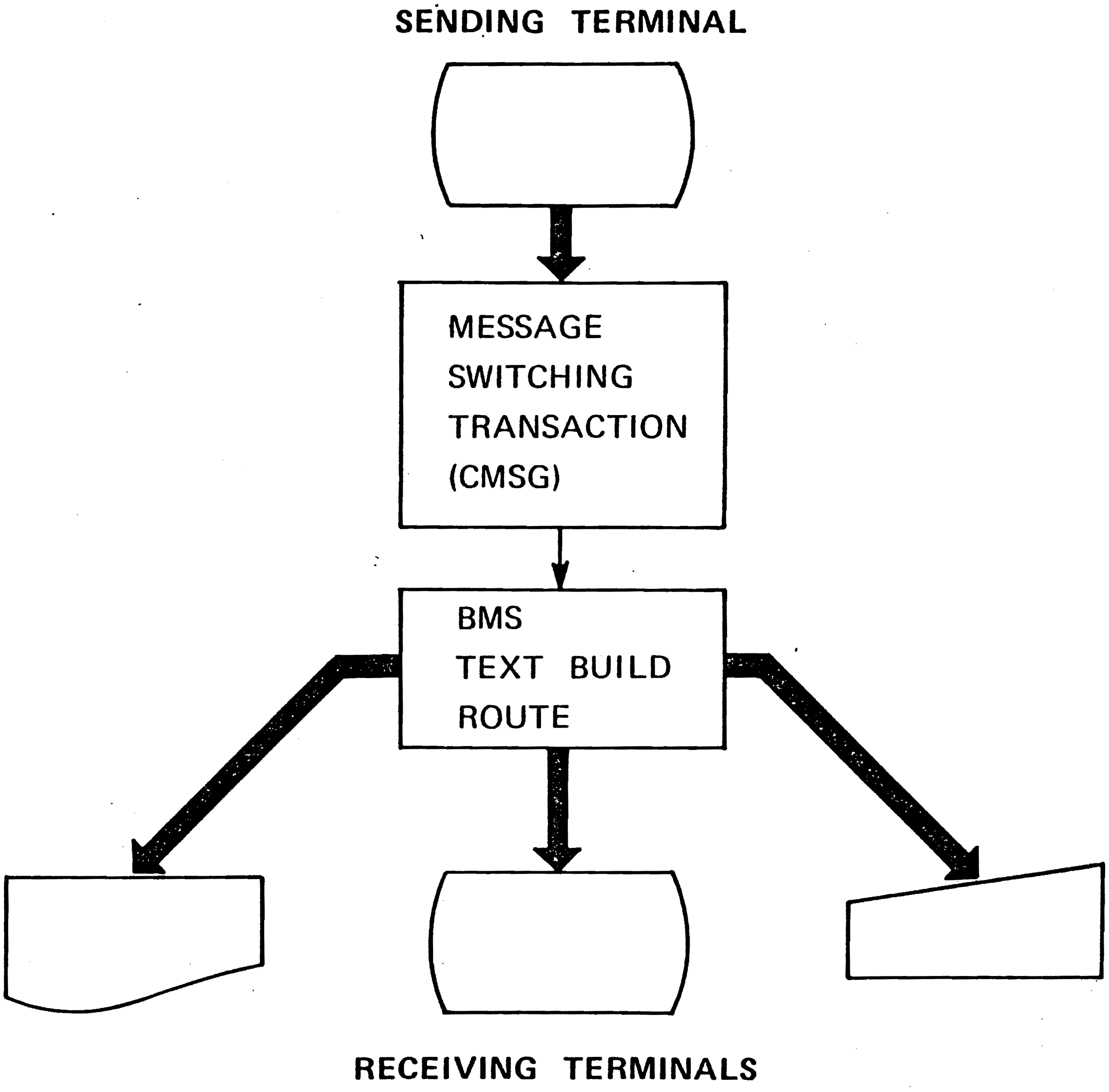
ROUTING

DIRECT/ROUTE INTERMIX

OUT	DIRECT TERMINAL
IN	DIRECT TERMINAL
ROUTE	
PAGEBLD	
PAGEBLD	
OUT	DIRECT TERMINAL
IN	DIRECT TERMINAL
PAGEBLD	
PAGEBLD	
PAGEBLD	
OUT	DIRECT TERMINAL
IN	DIRECT TERMINAL
PAGEBLD	
PAGEBLD	
OUT	DIRECT TERMINAL
PAGEOUT	ROUTED TERMINALS

THE DIRECT TERMINAL MAY BE ONE OF THE TERMINALS ON THE ROUTE LIST.

MESSAGE SWITCHING



SENDING TERMINAL

MESSAGE SWITCHING TRANSACTION (CMSG)

BMS TEXT BUILD ROUTE

RECEIVING TERMINALS

MESSAGE SWITCHING TRANSACTION

PSEUDO-CONVERSATIONAL

MESSAGES AND ROUTING MAY SPAN MULTIPLE INPUTS

REPLACE BY RE-ENTERING

ABBREVIATED KEYWORDS

BACKSPACE HONORED

MESSAGE SWITCHING

```
CMSG   [MSG=]  'MESSAGE'
       [,HEADING]
        ,ROUTE= ⎧ ([TERMID] [/OPRID] , . . . )                               ⎫
                ⎨ (,TERMLST , . . . )                                        ⎬
                ⎪ (,TERMLST , . . . ,± [TERMID]  [/OPRID] , . . . )          ⎪
                ⎩ ALL                                                        ⎭
       [,OPCLASS=(N,N, . . . )]
       [,TIME=VALUE]
       [,DATE=VALUE]
       [,ERRTERM= ⎧ TERMID ⎫ ]
                  ⎩ ORIG   ⎭
       [,ID=(TITLE)]
       [,CANCEL]
       [,SEND]
```

MESSAGE SWITCHING

MESSAGE TEXT –

```
MSG='MESSAGE'
M='MESSAGE'
'MESSAGE'
```

QUOTES WITHIN TEXT MUST BE ENTERED AS TWO QUOTES –

```
MSG='DON''T FORGET DOUBLE QUOTES'
```

EXAMPLES –

```
CMSG  MSG='GOOD MORNING',...
CMSG  M='GOOD MORNING',...
CMSG  'GOOD MORNING',...
                      E
CMSG  'THIS IS AN     O
                      B
                  E
EXAMPLE OF AN     O
                  B
MULTIPLE INPUT MESSAGE',...
```

MESSAGE SWITCHING

THE MESSAGE CAN BE STAMPED WITH –

TIME OF ORIGINATION
DATE OF ORIGINATION
ORIGINATING TERMINAL

EXAMPLES –

HEADING=YES
H=YES
HEADING
H

PREVIOUS HEADING REQUEST CAN BE IGNORED –

HEADING=NO
H=NO

MESSAGE SWITCHING

DESTINATIONS ARE FLEXIBLE –

ROUTE=(⇧)

XXXX	TERMINAL ID
/YYY	OPERATOR ID
.ZZ	TERMINAL LIST TABLE SUFFIX
+	ADD ENTRY TO TLT
–	DELETE ENTRY FROM TLT
ALL	ALL TERMINALS

OPCLASS=(N,N,. . .)

MESSAGE SWITCHING

EXAMPLES OF DESTINATIONS –

R=TRM1	A SPECIFIC TERMINAL
R=/OP1	A SPECIFIC OPERATOR IF SIGNED ON
R=TRM1/OP1	A SPECIFIC TERMINAL WHEN OP1 SIGNS ON
O=(2)	ANY OPR WITH OPCLASS 2 SIGNED ON
R=TRM1,O=(2)	A SPECIFIC TERMINAL WHEN OPR WITH OPCLASS 2 SIGNS ON
R=.L1	DESTINATIONS IN TLT DFHTLTL1
R=(.L1,+TRM1,–/OP1)	DESTINATIONS IN TLT WITH MODIFICATIONS
R=ALL	ALL TERMINALS

MESSAGE SWITCHING

MESSAGE DELIVERY MAY BE DELAYED –

TIME=HHMM	TIME OF DELIVERY ON 24 HOUR CLOCK ASSUMES SAME DATE UNLESS DATE SPECIFIED
TIME=+HHMM +MM +M	TIME OF DELIVERY RELATIVE TO CURRENT TIME
DATE=YY,DDD MM/DD MM/DD/YY	DATE OF DELIVERY ASSUMES CURRENT TIME UNLESS TIME SPECIFIED
DATE=+D	DATE OF DELIVERY RELATIVE TO CURRENT DATE

MESSAGE SWITCHING

A TERMINAL MAY BE NOTIFIED IF MESSAGE NOT DELIVERED WITHIN A SPECIFIC TIME INTERVAL.

$$\text{ERRTERM=}\left\{\begin{matrix}\text{TERMID}\\\text{ORIG}\end{matrix}\right\}$$

TIME INTERVAL SPECIFIED AT SYSGEN.

PRGDLAY=HHMM

CSMT IS ALWAYS NOTIFIED WHEN MESSAGE IS PURGED.

MESSAGE SWITCHING

THE MESSAGE CAN HAVE AN IDENTIFYING TITLE TO BE DISPLAYED BY THE OPERATOR WHEN 'P/Q' ENTERED.

ID=(TITLE)

MESSAGE SWITCHING

TRANSACTION MAY BE CANCELLED AS FOLLOWS –

CANCEL	LAST 6 CHARACTERS OF INPUT
CMSG	FIRST 4 CHARACTERS OF INPUT STARTS A NEW MESSAGE SWITCHING TRANS.

UPON COMPLETION OF INPUT OPERATOR ENTERS –

SEND

S

MESSAGE SWITCHING

EXAMPLES –

```
CMSG  'GOOD MORNING',R=ALL,S
CMSG  'GOOD MORNING',R=(.G1),T=0900,S
CMSG  'GOOD MORNING',R=(.G2,+TRM3),S
CMSG 'SUPERVISOR''S MEETING AT 3:00 PM',0=(4,5),S
CMSG 'SUPERVISOR''S MEETING IN 5 MINUTES',T=1455,0=(4,5),S
CMSG 'MR. L. SPERIK, ARRIVING 3:15 PST. PLEASE MAKE
        RESERVATION AT HYATT HOUSE.NL G. BRUINDE',R=PADC,H,S
CMSG  'SHUTDOWN IN 5 MINUTES',R=ALL,T=1655,S
```

JOURNALING

- BE PREPARED FOR RECOVERY BEFORE THE NEED ARISES.
- PROVIDE A RECORD OF TRANSACTION STATUS (SYSTEM STATUS) BEFORE ERRORS CAN OCCUR.
- HAVE STATUS RECORD READILY ACCESSIBLE.
- ORDER RECORDS CHRONOLOGICALLY; RECORD EVENTS AS THEY HAPPEN.
- RECORD IN PREDETERMINED AND RIGIDLY CONTROLLED FORMAT.

USE OF JOURNALS

RECOVERING FROM THE EFFECTS OF:

A USER ENTERING WRONG DATA

SYSTEM GOES DOWN

BAD PROGRAM POLLUTES DATA

LOSS OF DATA

JOURNAL MANAGEMENT

JCP. JCT

CREATION AND MANAGEMENT OF JOURNAL FILES

AUTOMATIC JOURNALING

APPLICATION SERVICE REQUESTS

RECORD OF DATA BASE CHANGES

SYSTEM ACTIVITY LOG

FACILITATE RECONSTRUCTION

FCT

AUTOMATIC JOURNALING

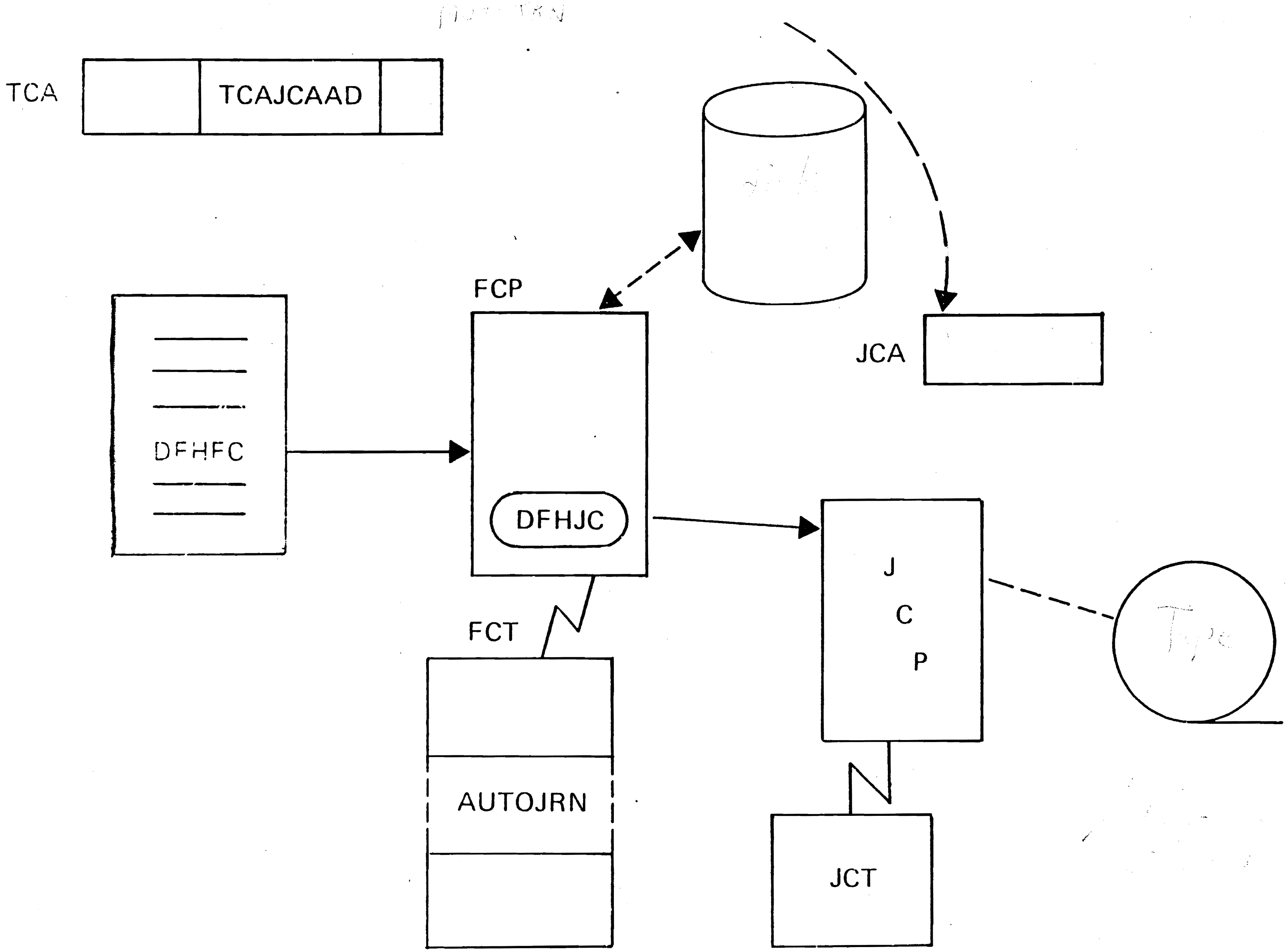
FILE CONTROL REQUESTS JOURNALING

FILE CONTROL TABLE DEFINES

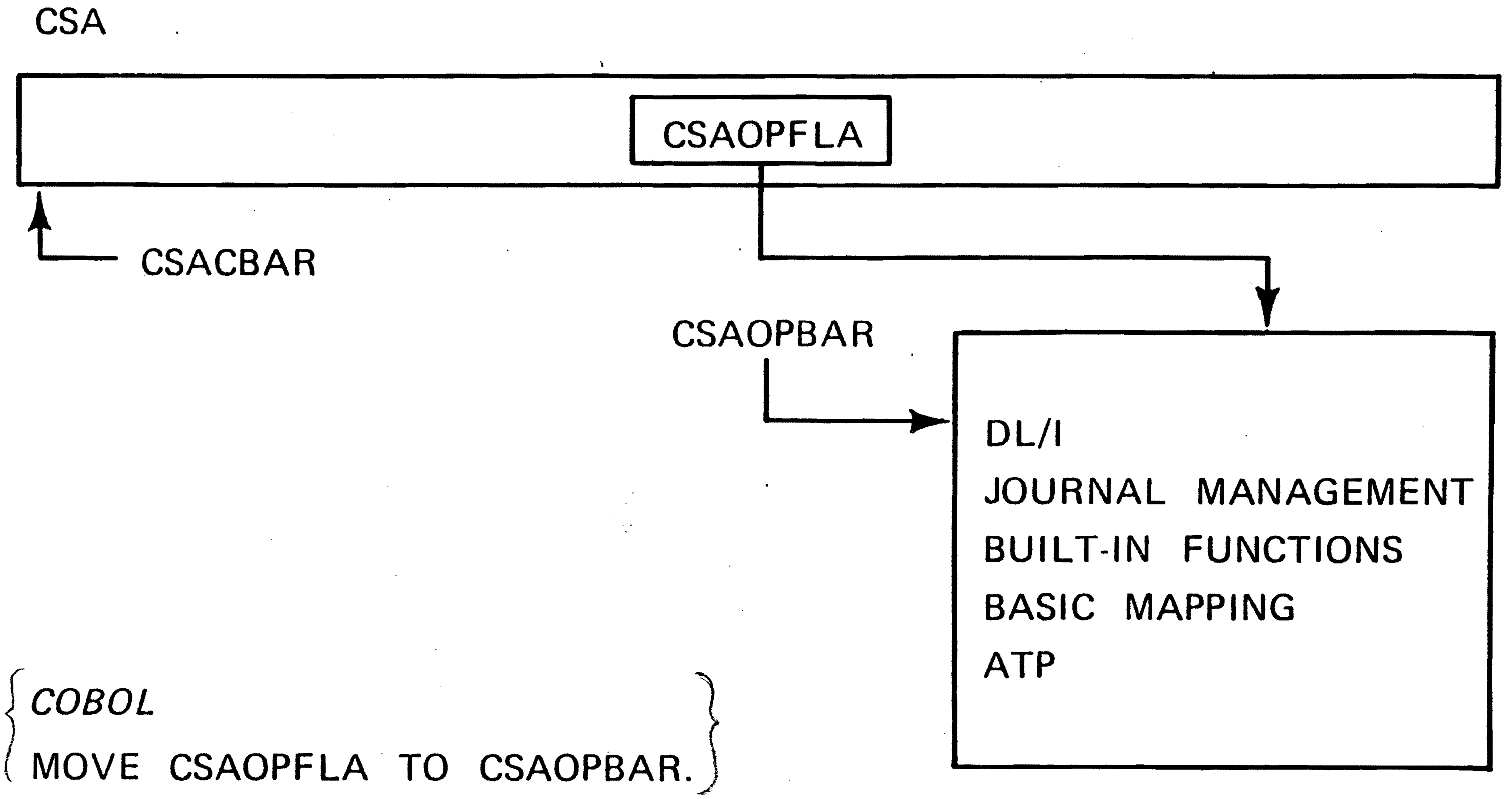
- WHICH JOURNAL TO USE
- TYPE OF REQUESTS TO BE JOURNALED
  - ALL, READ ONLY, READ FOR UPDATE, WRITE NEW, WRITE UPDATE
- MAY SPECIFY THAT . . .
  - BEFORE IMAGES OF RECORDS FOR UPDATE IN PLACE
  - RECORD ID OF ADDITIONS TO FILE
  - COPY OF RECORDS DELETED FROM FILE

. . . BE LOGGED TO A JOURNAL DATASET

AUTOMATIC JOURNALING



CSA OPTIONAL FEATURES LIST



CSA

CSAOPFLA

CSACBAR

CSAOPBAR

DL/I

JOURNAL MANAGEMENT

BUILT-IN FUNCTIONS

BASIC MAPPING

ATP

COBOL

MOVE CSAOPFLA TO CSAOPBAR.

THE APPLICATION PROGRAM CAN – – – –

- ACQUIRE THE TASK'S JOURNAL CONTROL AREA
- CREATE A JOURNAL RECORD AND WAIT FOR OUTPUT
- CREATE A JOURNAL RECORD BUT RETAIN CONTROL
- WAIT FOR OUTPUT OF A JOURNAL RECORD

USING JOURNAL SERVICES

DEFINE JOURNAL CONTROL AREA (JCA)

COPY DFHJCADS

OBTAIN A JOURNAL CONTROL AREA

DFHJC TYPE = GETJCA

REQUEST JOURNAL SERVICES — — — DFHJC

SYNCHRONOUS

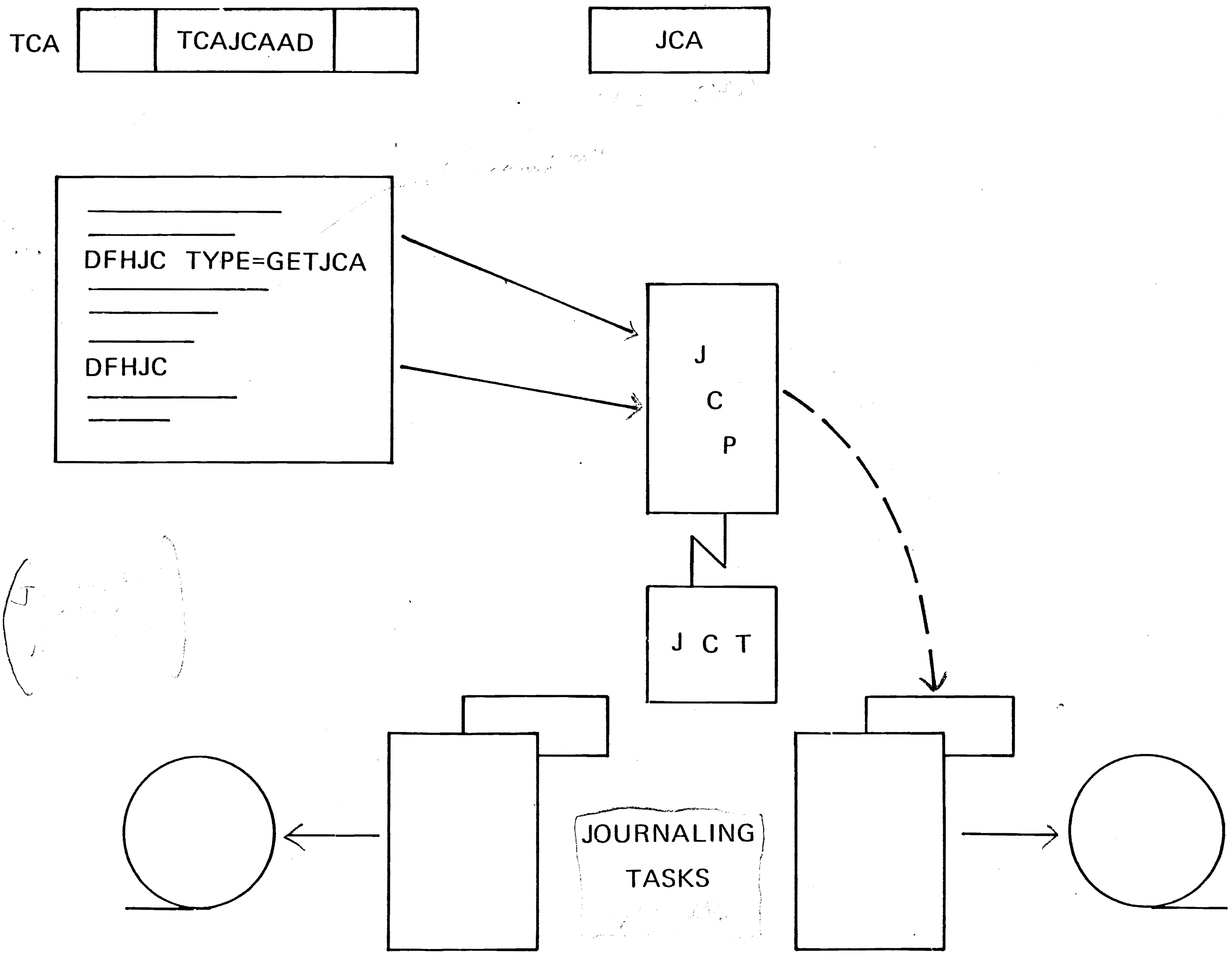
PUT or (WRITE,WAIT)

ASYNCHRONOUS

WRITE

SYNCHRONIZE

WAIT



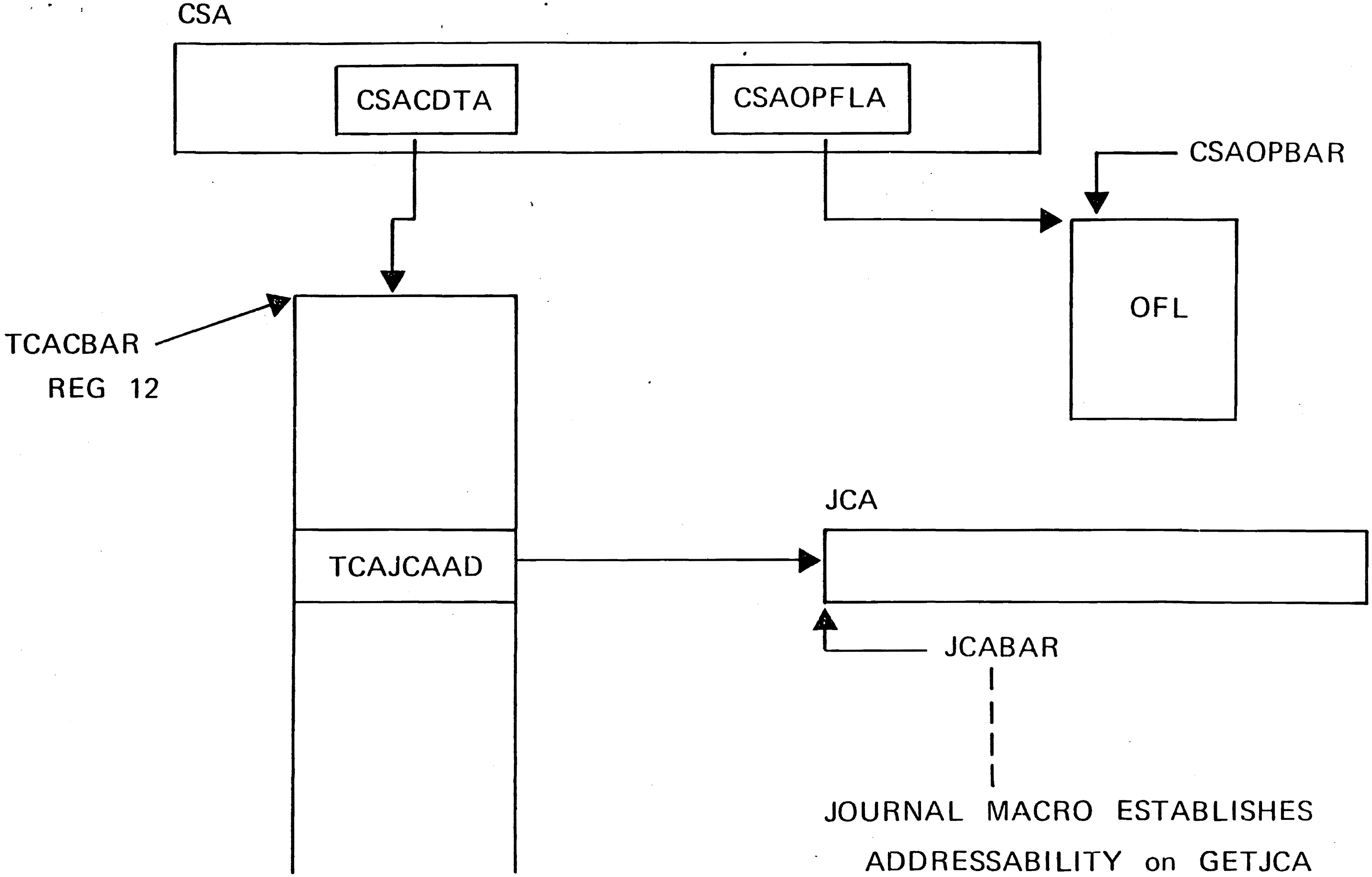
TCA
TCAJCAAD
JCA
DFHJC TYPE=GETJCA
DFHJC
J C P
J C T
JOURNALING TASKS

JOURNAL CONTROL

DFHJC TYPE=GETJCA

[illegible]

JOURNAL MANAGEMENT



JOURNAL MACRO ESTABLISHES ADDRESSABILITY on GETJCA

JOURNAL CONTROL

```
DFHJC        TYPE= { PUT         }
                   { (WRITE,WAIT)}
             ,JFILEID= { nn     }
                       { SYSTEM }
                       { YES    }
             [,JTYPEID= { nnnn } ]
             [          { YES  } ]
             [,JCDADDR= { symbolic address } ]
             [          { YES              } ]
             [,JCDLGTH= { decimal value } ]
             [          { YES           } ]
             [,PFXADDR= { symbolic address } ]
             [          { YES              } ]
             [,PFXLGTH= { decimal value } ]
             [          { YES           } ]
             [,NORESP=symbolic address]
             [,IDERROR=symbolic address]
             [,LERROR=symbolic address]
             [,IOERROR=symbolic address]
             [,NOTOPEN=symbolic address]
             [,INVREQ=symbolic address]
```

JOURNAL CONTROL

```
DFHJC          TYPE=WRITE
               ,JFILEID= ( nn      )
                         { SYSTEM  }
                         ( YES     )
              [,JTYPEID=  { nnnn }]
              [           { YES  }]
              [,JCDADDR= { symbolic address }]
              [          { YES              }]
              [,JCDLGTH= { decimal value }]
              [          { YES           }]
              [,PFXADDR= { symbolic address }]
              [          { YES              }]
              [,PFXLGTH= { decimal value }]
              [          { YES           }]
              [,STARTIO=  { YES }]
              [           { NO  }]
              [,COND=    { (YES,symbolic address) }]
              [          {  NO                    }]
              [,NORESP=symbolic address]
              [,IDERROR=symbolic address]
              [,LERROR=symbolic address]
              [,NOTOPEN=symbolic address]
              [,INVREQ=symbolic address]
```

JCILEN

JOURNAL CONTROL

```
DFHJC       TYPE=WAIT
            ,JFILEID= ( nn     )
                      { SYSTEM }
                      ( YES    )
           [,NORESP=symbolic address]
           [,IDERROR=symbolic address]
           [,IOERROR=symbolic address]
           [,NOTOPEN=symbolic address]
           [,INVREQ=symbolic address]
```

JOURNAL CONTROL

```
DFHJC          TYPE=CHECK
              [,NORESP=symbolic address]
              [,IDERROR=symbolic address]
              [,LERROR=symbolic address]
              [,IOERROR=symbolic address]
              [,NOTOPEN=symbolic address]
              [,INVREQ=symbolic address]
```

LOGICAL UNIT OF WORK (LUW)

- ONE TERMINAL CONVERSATION
- A COMPLETED DATA BASE UPDATE SEQUENCE
- AN ENTIRE CICS TASK

PROTECTED RESOURCE

- A RESOURCE WHICH HAS BEEN ENQUED UPON IN ORDER TO PROVIDE EXCLUSIVE USE, OR CONTROL.

SYNCH POINT

- A POINT AT WHICH A LOGICAL UNIT OF WORK HAS BEEN COMPLETED AND AT WHICH PROTECTED RESOURCES MAY BE FREED.
- IMPLICITLY DEFINED AS A CICS/VS TASK COMPLETION
- EXPLICITLY DEFINED AS AN APPLICATION FUNCTION UPON COMPLETION OF A LOGICAL UNIT OF WORK.

SYNC POINT RECORD IS WRITTEN TO A SPECIAL PURPOSE JOURNAL (SYSTEM LOG) UPON COMPLETION OF A LOGICAL UNIT OF WORK.

- DEFINES THE EXTENT OF BACKOUT FOR EMERGENCY RESTART.
  - INDICATES THAT THE WORK HAS BEEN COMPLETED
  - UPDATES AND MODIFICATIONS TO THIS POINT DO NOT NEED TO BE BACKED OUT IN THE EVENT OF SYSTEM FAILURE.
- SYNC POINT INDICATES TO CICS/VS THAT IT MAY FREE PROTECTED RESOURCES AND DEFERRED PROCESSING CAN BE INITIATED.
  - DEQUEING OF FILE CONTROL RECORDS
  - TRANSIENT DATA PURGE – IF QUEUE DEFINED AS RECOVERABLE