INITIALIZATION OF THE 516 TSS SYSTEM

This paper describes the initialization procedure for the 516 TSS system (how to load the system programs into the computer and onto disk). An extensive glossary of terms is included.

1. INITIALIZATION PROCEDURE

1. Turn ON the power in the DDP-516 computer and the control terminal.
2. Ensure that the bootstrap loader listed below is stored in the computer memory ($1\text{-}17_8$).

CORE ADDRESS	ASSEMBLY CODE
1	170340
2	002001
3	131440
4	002003
5	041464
6	130440
7	002006
10	100001
11	050000
12	024000
13	101001
14	010000
15	100401
16	002003
17	042000

3. Set all the sense switches UP.
4. Set the MA/SI/RUN switch to SI or MA.
5. Depress the MASTER CLEAR pushbutton.
6. Load a 1 in the P-register.
7. Place the system deck in the hopper of the card reader. See Fig. 1A.
8. Turn the card reader ON. Depress the POWER, MOTOR, and START pushbuttons.

9. Set the MA/SI/RUN switch to RUN.
10. Depress the START pushbutton. Cards are read until the 516 TSS system deck is loaded in core.
11. Fill up the hopper of the card reader with PROGRAM SEGMENT decks.
12. Depress the MOTOR and the START pushbuttons in the SOROBAN card reader.
13. Flip SENSE SWITCH 1. Cards are read. Do not allow the hopper to become empty until the PROGRAM SEGMENTs TYPIOX, TPIOX2 and .MENU are loaded. Then, the card reader stops and the control terminal lists the names of all the PROGRAM SEGMENTs called but not loaded.
14. Wait several minutes for the bootstrapping process to complete. The P-register will change from 37777_8 to 5777_8 when it is completed.
15. Hit a key in the control terminal. It outputs the message shown below.

```
:; 516 TSS
PWD?
```

16. Set SENSE SWITCH 3 DOWN.
17. Key-in the MASTER PASSWORD. (Depress the CONTROL KEY, and key-in anyone of the passwords listed below).

    a. ZPWFZ
    b. YPWFY

    The control terminal will output the message SYS?
18. Key-in SYSSAV followed by a CARRIAGE RETURN. A copy of the system is saved on disk.
19. Repeat 15, 17.
20. Place the remainder of the PROGRAM SEGMENT decks to be loaded in the hopper of the card reader. TTYIOX, TPIOX2 and .MENU must be included.
21. Repeat 12.

22. Key-in SEGDEC followed by a CARRIAGE RETURN. Cards are read. When the hopper becomes empty, the card reader stops, and the control terminal lists the names of all the programs called but not loaded. Processing this deck takes about 10 minutes.
24. Load the password and file backup decks into the card reader hopper.
25. Repeat 12.
26. Type FLOAD followed by a CARRIAGE RETURN.
27. Cards are read creating passwords and files. (The files are associated with the preceding password).

1.1 Assignment of Sense Switches

SENSE SWITCH 1

During the initialization process it allows the user to signal when the PROGRAM SEGMENT decks are loaded. During normal system operation, it interrupts typing on the control terminal when reversed.

SENSE SWITCH 2

It allows (DOWN) or inhibits (UP) automatic rebootstrapping after system crashes.

SENSE SWITCH 3

During initialization,

1. It signals a total rebootstrap (516 TSS system decks + PROGRAM SEGMENT decks) if UP or a partial rebootstrap (516 TSS system decks) if DOWN.
2. It ignores unallocated ID's while the system is running if UP.

SENSE SWITCH 4.

It controls disposition of user errors. If the switch is DOWN, control goes to the relocatable octal package. If the switch is UP, control passes to the absolute octal package which is a system crash.

2. 2. HOW INITIALIZATION WORKS

Refer to INITIALIZATION PROCEDURE step 14.

2.1 Pass 1

During this pass the following events take place:

1. The DISK SYMBOL TABLE is initialized (written on disk). The table can hold 1024 4-word entries, hence it occupies 4096 words. The table is located in Track 3 Sectors 0-511 inclusive. Refer to 516-04.
2. The control teletype is set in the input mode.
3. The ring devices are initialized by sending certain commands to each one of the devices on the ring. This will allow the Awake Flag* of each device to interrupt.
4. The segment decks are loaded in the card reader. To continue SENSE SWITCH 1 is flipped.
5. The DISK ALLOCATION TABLE is initialized (bits set to ONE).
6. Segment decks are read. The data of each card is packed (4 columns become 3 words) and written on disk. When a PROGRAM CARD is read, space for the PROGRAM SEGMENT is assigned in the DISK ALLOCATION TABLE. Each bit in this table corresponds to a 64-word DISK SEGMENT. The disk address and the 3-bit size code are combined to form the 16-bit segment ID.
7. The PROGRAM SEGMENT's name and its ID number are stored in an in-core image of the DISK SYMBOL TABLE as shown below.

   NAME (first three characters)
   NAME (last three characters)
   ID#

   Steps 6 and 7 will be repeated for every PROGRAM SEGMENT that follows until the hopper is empty. Then PASS 2 begins.

---

* David R. Weller, "A Loop Communication System for I/O to a Small Multi-User Computer", MM 70-1384-1, September 29, 1970.

2.2 Pass 2

The first card image is read from disk. Its format should correspond to a PROGRAM CARD where the names of all the external PROGRAM SEGMENTs called are listed. If it is not, an error message is output at the control terminal. The in-core image of the DISK SYMBOL TABLE is searched for the ID number corresponding to each PROGRAM CARD name. Each ID found is stored in an auxiliary table. Each undefined name is typed at the control teletype and a zero is inserted in the auxiliary ID table. After the last name has been processed, all the words of the CORE SEGMENT (1400_8 words) are set to zeros. Then the contents of each DATA CARD corresponding to the PROGRAM SEGMENT are processed and stored in the CORE SEGMENT until the PROGRAM CARD image of the next PROGRAM SEGMENT is read. The size of the segment is computed from the 1st ID of the auxiliary ID table and stored in the 1st word of the segment header together with the segment type. The check sum of the PROGRAM SEGMENT is computed and put in the 2nd word of the header. The program is written on disk and a hardware check is performed to assure that it was written correctly. If it was not, it is tried again.

Once the last PROGRAM SEGMENT has been written on disk the core image of the DISK SYMBOL TABLE is written on disk in the following manner:

a. A working pointer to the beginning of the table is initialized.

b. By the use of hash coding, a DISK-SECTOR address is computed in the disk track used for the DISK SYMBOL TABLE. If the selected sector is already in use, a new one is computed by rotating the hash code (see ENTAB subr in BOOTST).

The top of core pointer is readjusted to allow the space formerly used by the in-core DISK SYMBOL TABLE and the bootstrap programs to be used by PROGRAM SEGMENTs. The ID's of the PROGRAM SEGMENTS listed in 2.2.1 are loaded into the system.

PASS 2 is finished. The system passes control to the bolted in-core IDLER program to start normal system operation (log-in).

2.2.1 Program Segments Used by the System

.LOGON	identifies the type of user (control teletype, dataphones, PDP-8 computer, DDP-316 computer, CSX Ganglion and multispeed dataphone).
.ROPAK	aid in debugging PROGRAM SEGMENTS. Refer to 516-31.
.CHARD	steps a character pointer across the segment boundaries for character string handling.
.XPWFX	contains the different MASTER PASSWORDs and their associated IDs. This allows a user to log-in.
.GLOBL	GLOBAL DIRECTORY. File containing names and pointer to FILES available to all users.
HUPEND	disconnects a THREAD if the dataphone is hung up.
ESCAPE	handles the control C escape to system level and core overload problems.
TYPIOX	handles I/O for the various devices attached to the 516 computer.

3. GLOSSARY

DISK SECTOR	Eight words of contiguous disk space.
DISK SEGMENT	A multiple of 64 words of contiguous space.
CORE SEGMENT	Contiguous space of core. The size of this space varies from 100_8 to 1400_8 words.
PROGRAM SEGMENT	- Program of subroutine which occupies one CORE SEGMENT (as opposed to SYSTEM PROGRAM).
516 SEGMENT ASSEMBLER	- SEGMENT PROGRAM's assembler. Refer to 516-41,49.
SEGMENT DECK	Binary deck of cards obtained after assembling a SEGMENT PROGRAM. There are

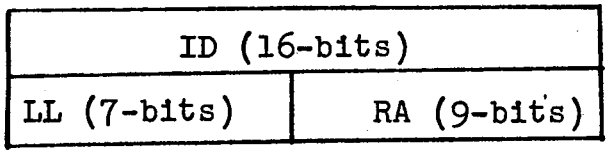
two types of binary cards in each deck: PROGRAM CARDS and DATA CARDS. The former contains the SEGMENT PROGRAM's name, its size and the names of any other external SEGMENT PROGRAM's called. The latter contains the SEGMENT PROGRAM's instructions. Refer to 516-14,20.

IDENTIFICATION NUMBER (ID) - Binary code (16-bits) that identifies a DISK SEGMENT. See Appendix A.

LOOSE LINK (LL) - Binary number (7-bits) that points to an entry of the IN-CORE SEGMENT TABLE that contains the ID and the starting physical address of every SEGMENT PROGRAM loaded in the DDP-516 core memory. Refer to 516-9.

RELATIVE ADDRESS (RA) - Address (9-bits) that points into a SEGMENT PROGRAM loaded in a CORE SEGMENT. Refer to 516-9.

VIRTUAL ADDRESS (VA) - Two-word address that allows the system to locate a word within a segment. This address consists of ID, RA and LL as shown in the figure below. Refer to 516-9. The LL field is not significant unless the segment is currently in core.



THREAD — User. These terms are used interchangeably.

THREAD ENTRY — Seven word entry in the THREAD TABLE allocated to each user. It contains pointers and data essential to a THREAD operation. Refer to 516-27.

THREAD TABLE — In-core table (42 words) where every THREAD ENTRY is stored. Six entries can be held in this table. Refer to 516-27.

THREAD SAVE — In-core segment (78 words) where all the information for the handling of a THREAD by the system is stored. This information consists of user's relocatable pointers, temporary locations and push down lists, etc. Refer to 516-9.

PUSH DOWN LIST - Buffer where the RA, ID and base address of SEGMENT PROGRAM's accessed through any form of the CALL macro are stored. This buffer is part of the THREAD SAVE. Refer to 516-19.

ROADBLOCK — Disable a THREAD ENTRY temporarily. This happens during I/O operations. However, when the I/O ends the entry is unroadblocked.

IDLER — Processing allocator of the multiprogramming system (SYSTEM PROGRAM). It steps from THREAD to THREAD processing when it can and skips over the THREADS that are roadblocked.

STRING — Chain of CORE SEGMENTS. Refer to 516-50 p. 6 description of the different types of strings.

FILE — STRING that is linked to some directory. This directory line includes the FILE name and the ID of the first DISK SEGMENT of the STRING. See Appendix B.

IN-CORE SEGMENT TABLE - A 128-word buffer that contains the ID and absolute address of all the PROGRAM SEGMENTS that are in-core. Refer to 516-5.

DISK ALLOCATION TABLE - Permanent in-core buffer, where the system keeps an account of which DISK-SEGMENTS have been used to store the PROGRAM SEGMENTS and data (including FILES). See Appendix C.

DISK SYMBOL TABLE - Disk table where the name and ID of each PROGRAM SEGMENT loaded in the system is stored. Refer to 516-6.

I/O TABLE Permanent in-core table (80 words) where a 5 word entry has been made for each device on the coaxial ring to handle their input and output operations. Refer to 516-25.

DISK-DMA QUEUE TABLE - In-core (80 words) buffer where a maximum of 20 requests can be registered for disk-to-core and core-to-disk transactions. Refer to 516-26.

GATE - SYSTEM PROGRAM used by systems programmers that allows only one THREAD to use a SEGMENT PROGRAM. The gate must be opened to make the PROGRAM SEGMENT available to other THREADS.

PASSWORD Code that allows a user to gain access to a directory of files.

MASTER PASSWORD - PASSWORD that allows a user to gain access to a file where all the user's PASSWORDS are stored.

USER DIRECTORY - File containing all the users file names and pointers to the associated files.

GLOBAL DIRECTORY - File containing names and pointers to FILES <u>available to all</u> users.

APPENDIX A

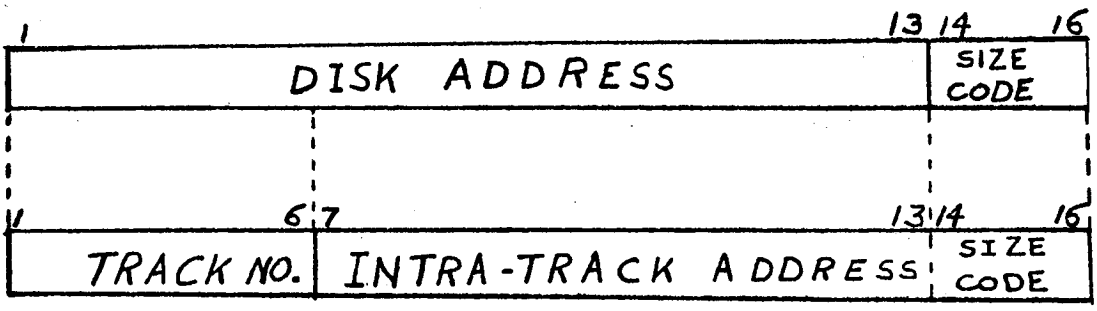
ID-NUMBER

The identification number of a SEGMENT PROGRAM is a 16-bit number that contains:

1. A 3-bit code for the DISK SEGMENT's size.

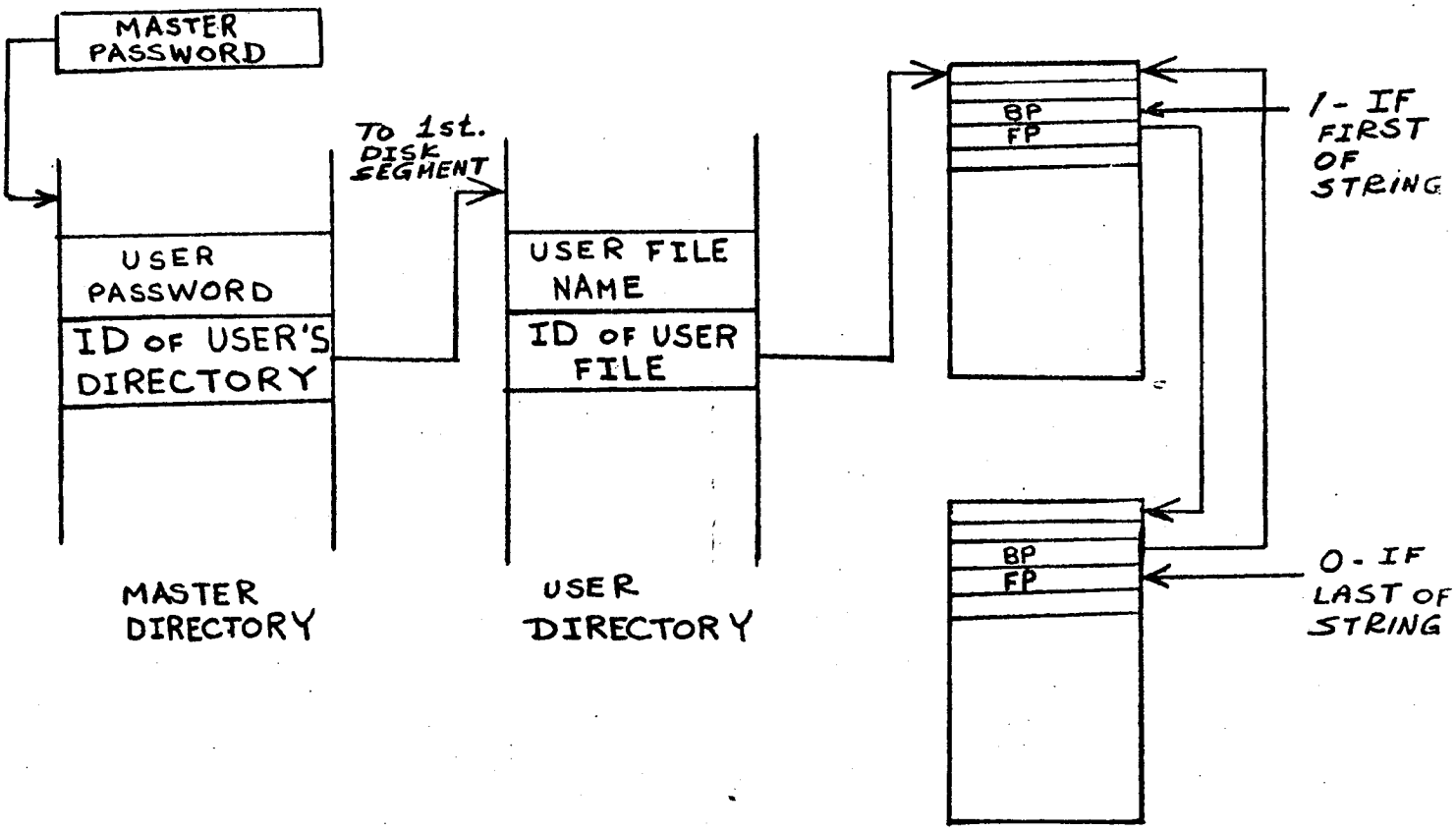
CODE	SIZE	
	OCTAL	DECIMAL
0	100	64
1	200	128
2	300	192
3	400	256
4	600	384
5	1000	512
6	1200	640
7	1400	768

2. The track-sector address where the DISK SEGMENT begins.



APPENDIX B

FILES' STRUCTURE



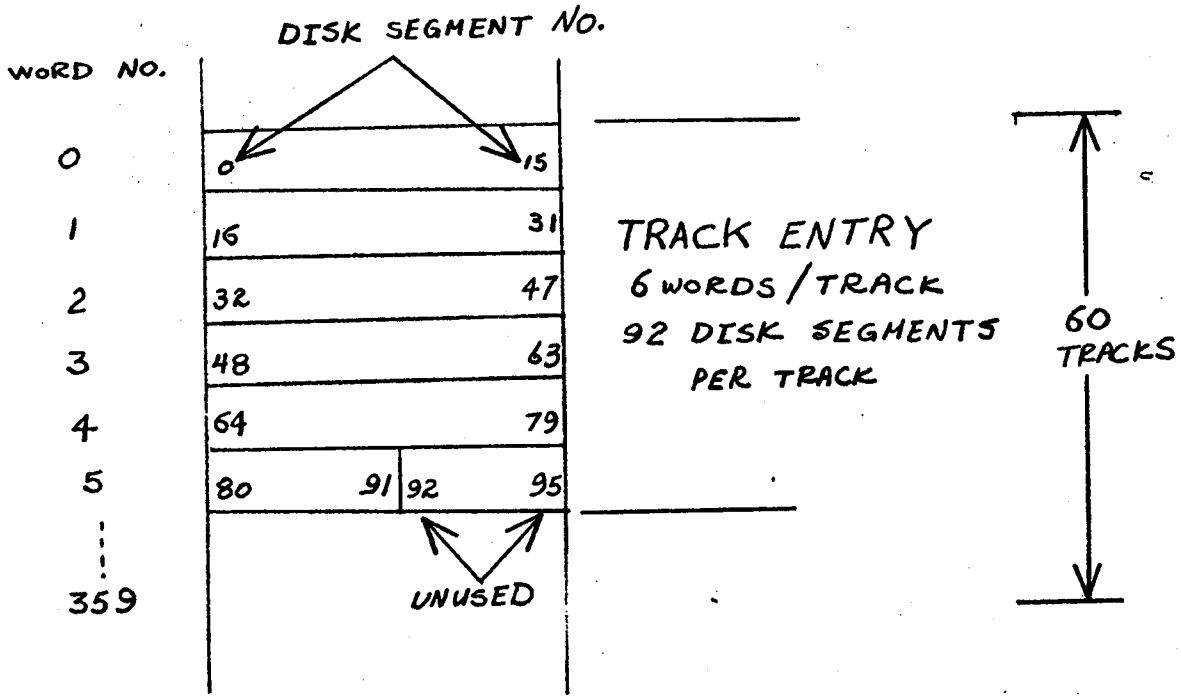
BP - BACKWARD POINTER. ID OF THE PRECEEDING SEGMENT.
FP - FORWARD POINTER. ID OF THE FOLLOWING SEGMENT FOUND.

If a FILE NAME is not found in the USER's DIRECTORY, it is looked up in the GLOBAL DIRECTORY.

APPENDIX C

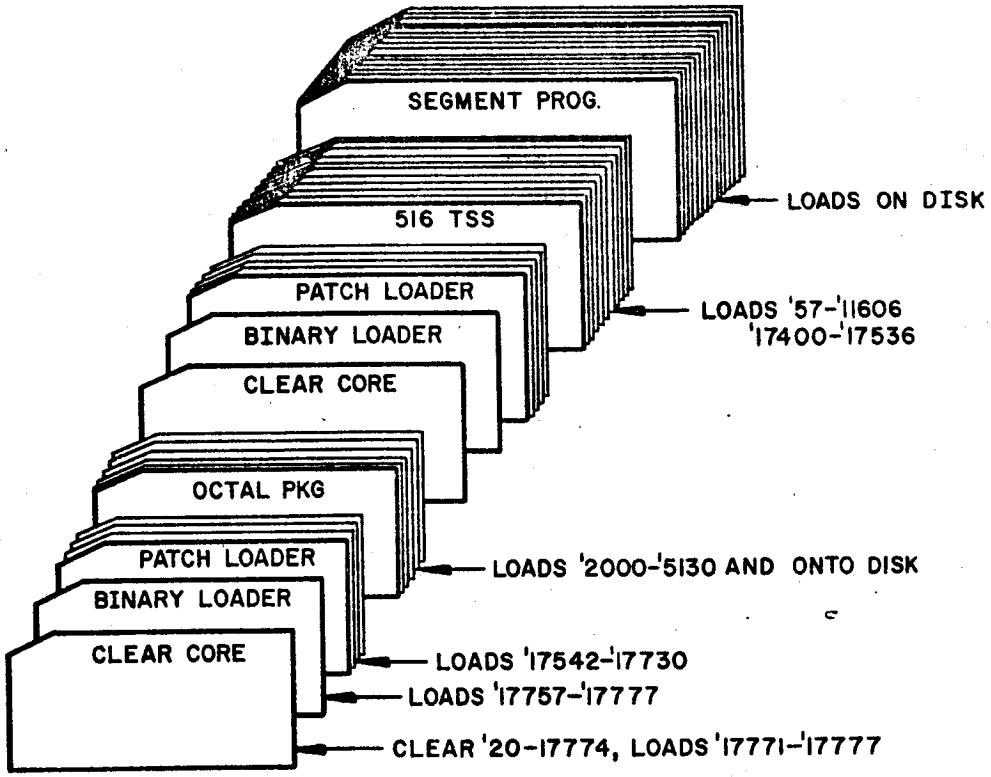
DISK ALLOCATION TABLE

Indicates the status of each DISK SEGMENT (used or free).

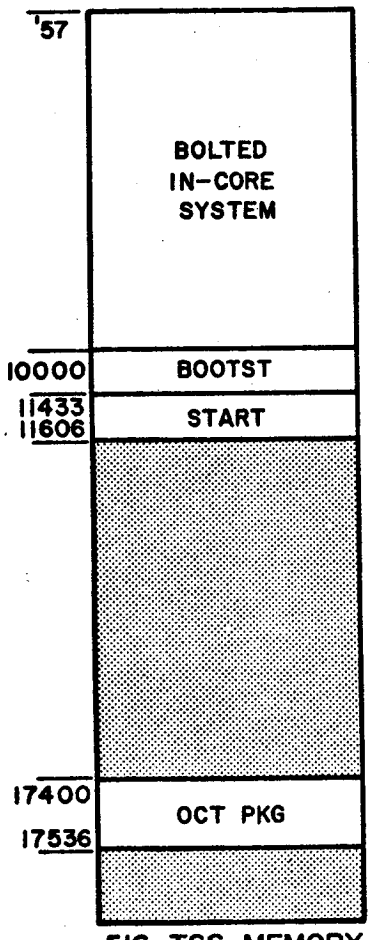


It occupies 360 (dec) words; the first four tracks of the disk are dedicated to the system and are excluded from the table.

A copy of this table is kept on disk in the area designated SYSTEM STATE.

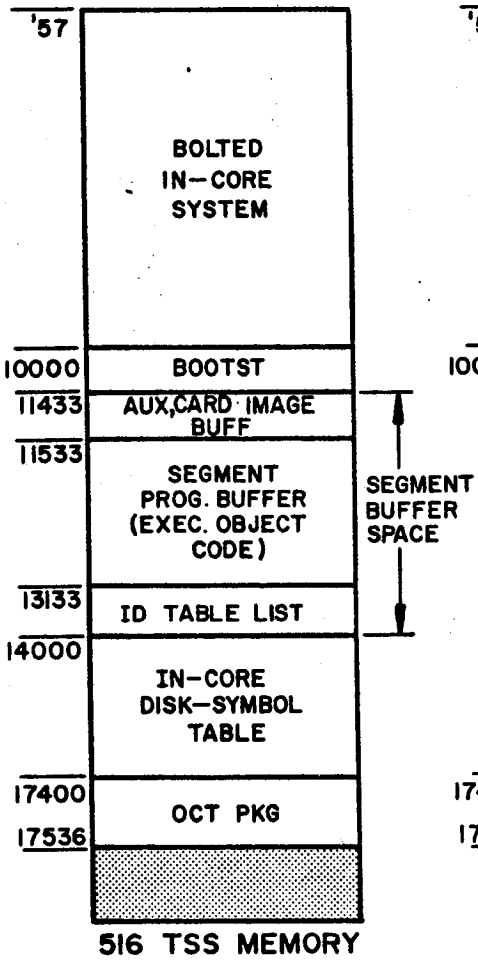


516 TSS DECKS
FIG IA



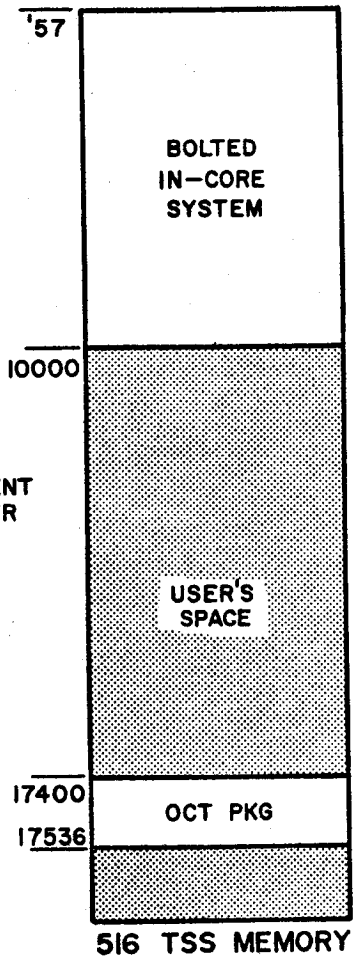
516 TSS MEMORY MAP

BEFORE LOADING ANY SEGMENT PROG
FIG. IB



516 TSS MEMORY MAP

DURING PASS 1 AND PASS 2
FIG. IC



516 TSS MEMORY MAP

DURING NORMAL OPERATION
FIG. ID