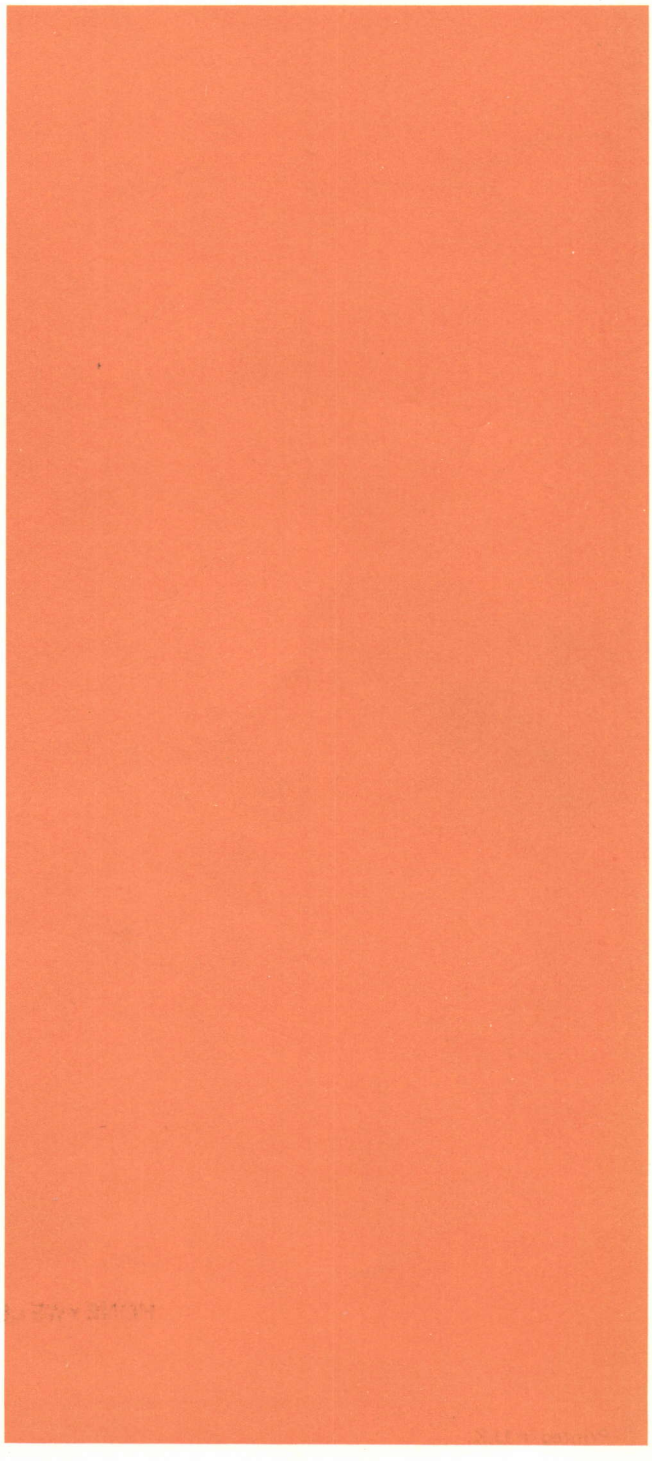
Honeywell

UTILITY PROGRAMS

SYSTEM 700

OP–16

SOFTWARE



Honeywell

UTILITY PROGRAMS

SYSTEM 700 SOFTWARE

OP–16

SUBJECT:

This Manual describes the Functions, Use and Building Procedures for Utility Programs for the OP-16 Operating System.

SPECIAL INSTRUCTIONS:

This manual forms part of the OP-16 Documentation Package and supports Software Release 0300.

DATE:

August 1975

ORDER NUMBER:

YZ50, Rev 2

DOCUMENT NUMBER:

41205813B

PREFACE

This manual describes the OP-16 Utility Programs and the building procedures required to incorporate these utilities within an Operating System. On-line and off-line operating modes are considered.

This manual forms part of the OP-16 Documentation Package (see chart) and should be read in association with the OP-16 User Guide manual which describes the basic RTX Executive. The reader is also assumed to have a basic familiarity with Series 16 assembly language programming and to have access to those manuals listed as Reference Documents.

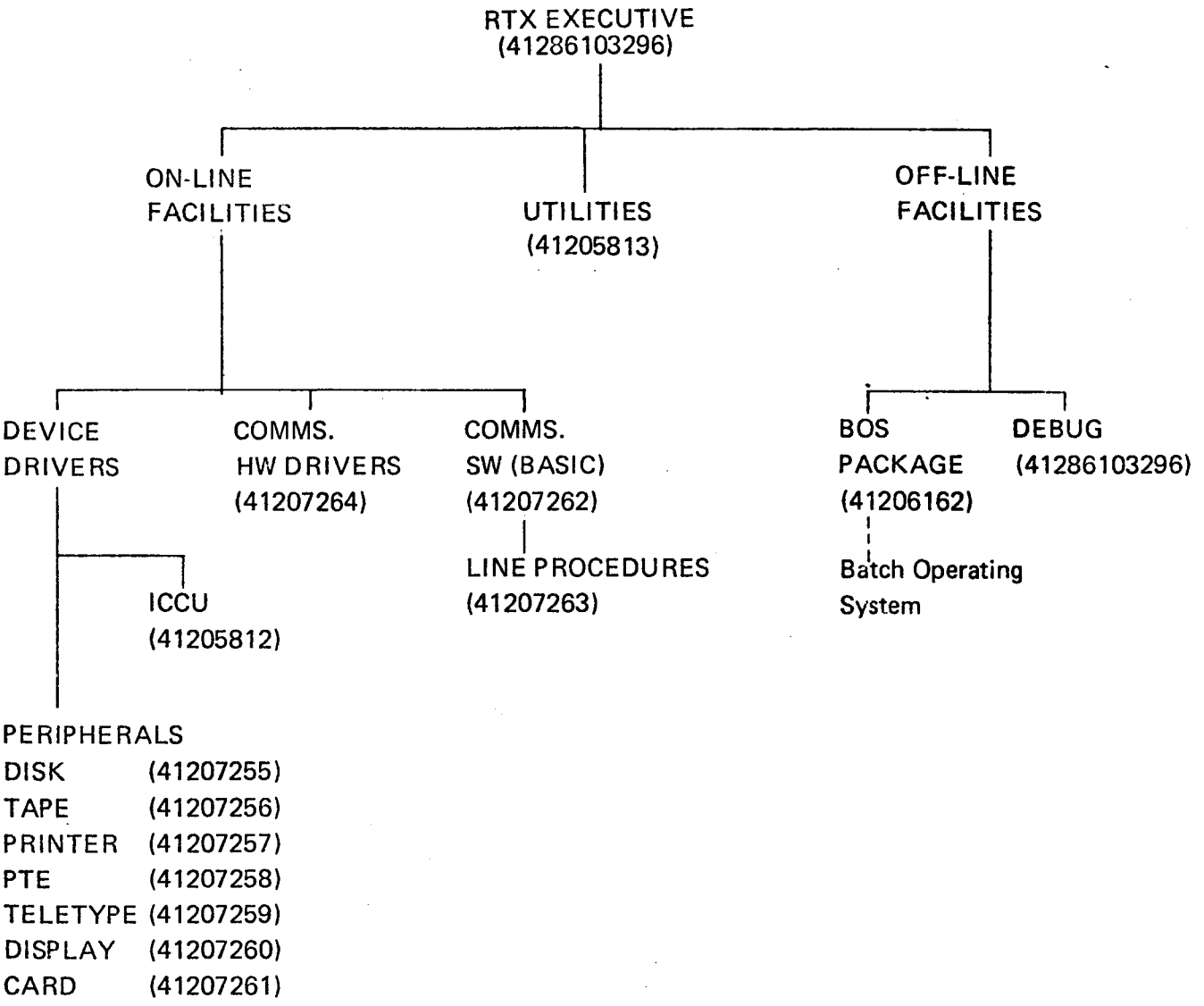
The OP-16 Utility Programs manual is structured as follows. An introductory section characterizes the two types of utility programs that can be built (off-line and on-line), describes the preconfigured basic utilities available to the user, and shows the applicability of MINX-716, an interrupt handler and supervisor program. Section II provides detailed procedures for initiating, conversing with, terminating, aborting, and editing the utility programs. Section III summarizes the system control, debugging, and transfer/verify functions currently available. Section IV, the longest part of the manual, gives step-by-step operating/programming procedures for building the OP-16 utility programs. The Batch Operating System (BOS) provides a very convenient way of building, and can be used if available.

For further information on the operating procedures of LDR-APM and PAL-AP with which the reader should be familier, consult the Series 16 Equipment Operator's Manual.

Reference Documents

BOS (Batch Operating System) programming manual	-	AB55
Series 16 Equipment Operator's manual	70130072165	BX48

OP-16 PROGRAM STRUCTURE - Supporting Documentation shown in brackets



OP-16 Documentation Package Summary

User Guide	41286103296	YY78
Utility Programs	41205813	YZ53
BOS File Package	41206162	YZ49
ICCU	41205812	YZ52
Disk Manual	41207255	YX35
Magnetic Tape Manual	41207256	YX36
Printer Manual	41207257	YX37
Paper Tape Manual	41207258	YX38
Teletype Manual	41207259	YX39
Display Manual	41207260	YX40
Card Equipment Manual	41207261	YX41
Basic Communications SW	41207262	YX42
Line Procedures	41207263	YX43
Communications HW Drivers	41207264	YX44

CONTENTS

CONTENTS (cont'd.)

ILLUSTRATIONS

TABLES

SECTION I
INTRODUCTION

PURPOSE

The Utility Programs provide a conversational interface between an operator and the OP-16 operating system, via an ASR-33 or ASR-35 teletype.

By using the program components provided, two types of utility programs can be built, each to suit a given hardware configuration and each to have all or part of the possible set of functions. The two types are:

1. Off-Line Utilities
2. On-Line Utilities

Both the off-line and on-line versions utilize the OP-16 driver programs and may be configured to be fully core-resident or, for core/mass-store systems, fully or partially mass-store-resident (Figure 1-1). All components are provided in relocatable object format.

Characteristics of the programs are summarized in Table 1-1.

ATTRIBUTES AND LIMITATIONS

The Off-Line Utility Programs assume a single-program, single-interrupt environment at any given time and do not expect program interruptions. They provide off-line

- program debugging tools;
- data transfer/verification between a variety of storage devices and external media.

When the user response on the ASR is to be followed by a carriage return, the symbol (CR) is used.

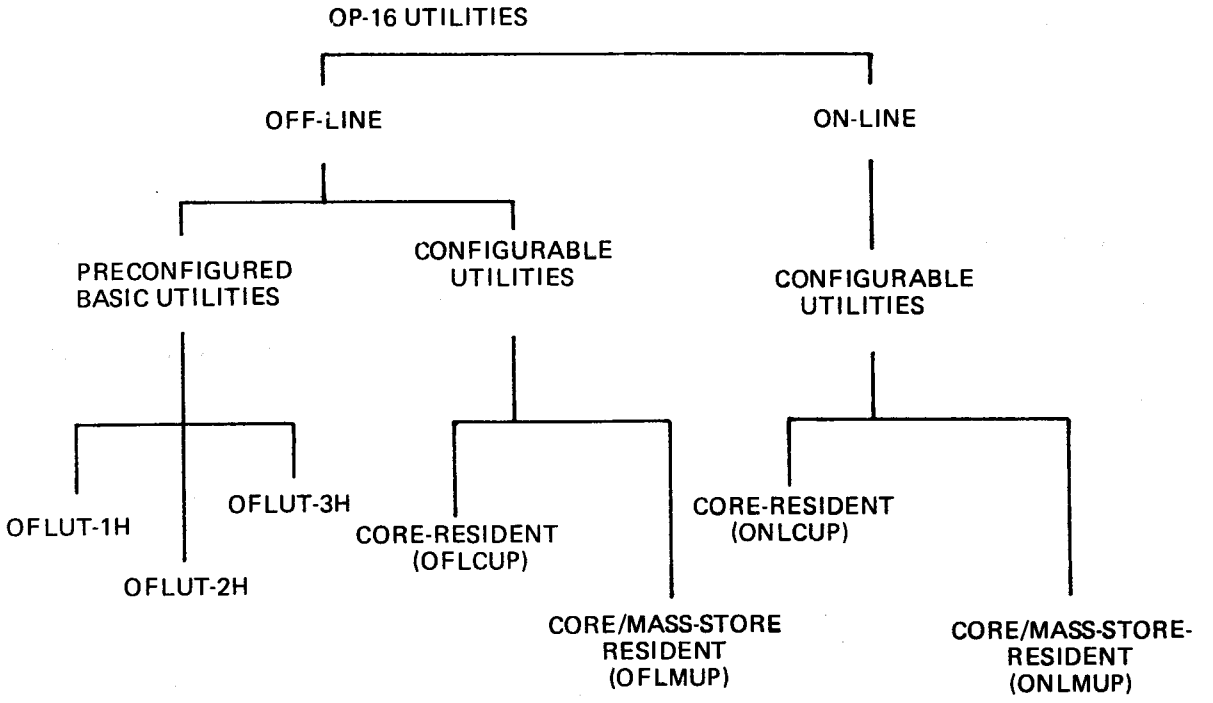


Figure 1–1. OP-16 Utility Programs

The On-Line Utility Programs run under the control of the RTX-16 Executive and provide on-line

- program debugging tools;
- data transfer/verification between a variety of storage devices and external media;
- optional operator control over the initiation and termination of periodic programs and initiation of "single shot" programs;
- control of the location and size of the core area available to the operator for on-line manipulations;
- printing and adjusting of the time of day.

The modular construction of the utility programs allows the user to include or exclude features to suit particular requirements and hardware environments. The functions which may be incorporated are summarized in Table 1-1 and discussed in detail in Section III.

SUPPORTABLE OPTIONS AND REQUIREMENTS

The required hardware and software and the supported hardware are summarized in Table 1-2.

PRECONFIGURED BASIC UTILITIES

To aid the user in building the desired utilities to meet a particular application, three library tapes are supplied, each containing I/O subroutines, I/O drivers, and functions (sequenced in their loading order) required to build an off-line core-resident utility program. The capabilities of each are described below:

OFLUT-1H — is a core-resident off-line utility which supports the ASR Keyboard/Printer and the High Speed Paper Tape Reader/Punch. It includes the functions shown in Table 1-1.

OFLUT-2H — is a core-resident off-line utility which supports the ASR Keyboard/Printer, the High Speed Paper Tape Reader/Punch, and the 20-surface or 10-surface Moving Head Disk, Option 710-47xx. It includes the functions shown in Table 1-1 and may conveniently be used to build other disk-resident utilities or other programs.

Table 1-1. Functions Included in OP-16 Utility Programs

Function		Available on						
Description	Command[1]	ONLMUP	ONLCUP	OFLMUP	OFLCUP	OFLUT-1	OFLUT-2	OFLUT-3
Print Time	PT	X	X					
Replace Time	RT	X	X					
Request Program	RP	X	X					
Connect Clock	CC	X	X					
Disconnect Clock	DC	X	X					
Replace Core	RC	X	X	X	X	X	X	X
Print Core	PC	X	X	X	X	X		
Fill Core	FC	X		X				
Search Core	SC	X		X				
Print Core Protection Limits	PL	X	X	X	X			
Replace Core Protection Limits	RL	X	X	X	X			
Transfer Data — Core to Paper Tape Punch	TR COPP	X	X	X	X	X	X	X
Transfer Data — Paper Tape Reader to Core	TR PRCO	X	X	X	X	X	X	X
Verify Data — Paper Tape Reader to Core	VE PRCO	X	X	X	X	X	X	X
Transfer Data — Core to System Mass Store	TR COSM	X	X	X	X		X	X
Transfer Data — System Mass Store to Core	TR SMCO	X	X	X	X		X	X
Verify Data — System Mass Store to Core	VE SMCO	X	X	X	X		X	X
Transfer Data — System Mass Store to P.T. Punch	TR SMPP	X		X				
Transfer Data — P.T. Reader to System Mass Store	TR PRSM	X		X				
Verify Data — P.T. Reader to System Mass Store	VE PRSM	X		X				
Transfer Data — Core to Magnetic Tape	TR COMT	X		X				
Transfer Data — Magnetic Tape to Core	TR MTCO	X		X				
Verify Data — Magnetic Tape to Core	VE MTCO	X		X				
Transfer Data — System Mass Store to Mag. Tape	TR SMMT	X		X				
Transfer Data — Mag. Tape to System Mass Store	TR MTSM	X		X				
Verify Data — Mag. Tape to System Mass Store	VE MTSM	X		X				

[1]The commands shown in this table identify the type of function. When demanding most functions, the operator must enter additional parameters as described in Section III.

OFLUT-3H is similar to OFLUT-2H except that it supports the Fixed Head Disk, Option 710-451x.

Each of the above preconfigured basic utilities is punched in object form on a single paper tape which contains all necessary program components in their appropriate loading order. The programs may be loaded anywhere in core via LDR-APM. Once the off-line utility program is built, it may be punched out by PAL-AP to obtain a self-loading paper tape.

MINX-716

All OP-16 utility programs utilize the standard OP-16 I/O drivers which are designed to run under the interrupt and priority control of the RTX-16 Executive. In order to use the on-line drivers off-line, a small interrupt handler and supervisor program (MINX-716) is provided as a component of the off-line utility programs. MINX-716 allows only sequential execution of programs and enables only one interrupt at a time.

DEFINITION OF TERMS

In this manual, the term mass store is used to describe the system disk peripheral. The word segment refers to a 128-word block of core or mass store. The term segmented program is used to describe a program which contains a core-resident portion (called the root) and one or more mass-store-resident portions (called overlays) which are brought into core only when required.

Octal numbers are preceded by an apostrophe or followed by the word octal; e.g., '57 = 57 (octal).

Angle brackets < > enclose items which are variable in nature.

Parentheses ( ) indicate the contents of a given register. For instance, (A) = '333 means that the A register contains octal 333.

When the user response in the ASR is to be followed by a carriage return, the symbol (CR) is used.

Table 1-2. OP-16 Utility Programs; Hardware/Software Prerequisites

SYSTEM CONFIGURATION	USER CONFIGURABLE UTILITIES												PRECONFIGURED UTILITIES (OFF-LINE CORE RESIDENT)					
	ON-LINE CORE RESIDENT(ONLCUP)			ON-LINE CORE/MASS STORE RES.(ONLMUP)			OFF-LINE CORE RESIDENT(OFLCUP)			OFF-LINE CORE/MASS STORE RES.(OFLMUP)			REQUIRED HARDWARE			REQUIRED SOFTWARE		
	REQ HW	SUPP HW	REQ SW	REQ HW	SUPP HW	REQ SW	REQ HW	SUPP HW	REQ SW	REQ HW	SUPP HW	REQ SW	OFLUT -1H	OFLUT -2H	OFLUT -3H	OFLUT -1H	OFLUT -2H	OFLUT -3H
Minimum HW configuration for OP-16 [1]	X			X			X			X			X	X	X			
710-5010 PTR		X			X			X			X		X	X	X			
710-52XX PTP		X			X			X			X		X	X	X			
710-451X FHD				X				X		X					X			
710-47XX MHD				X				X		X				X				
710-4150 MT					X						X							
710-4021 MT					X						X							
MINX-716 EXEC.			X			X												
MINEX			X			X												
MINEX Device Drivers									X			X				X	X	X
CORE OCCUPIED (sectors)	1 or 2 depending on configuration			1, independant of configuration			Up to 2, plus drivers as needed			Between 1 and 4, plus drivers as needed						4	5	5
MAX. NUMBER OF FUNCTIONS	15			26			10			21						5	7	7
RUN UNDER	RTX-16			RTX-16			MINX-716			MINX-716						MINX -716	MINX -716	MINX -716
STRUCTURE	Contiguous			Segmented, overlaid in core			Contiguous			Segmented, overlaid in core							Contig-uous	
BUILT BY USING	BOS or LDR-APM			BOS, LDR-APM, OFLUT-2H/3H, or OFLUP			BOS or LDR-APM			BOS, LDR-APM, OFLUT-2H/3H, or OFLCUP							BOS or LDR-APM	

NOTES: 1. Minimum HW configuration to run OP-16

2. Object text is arranged on a single paper tape in the order of loading

SECTION II

USE

CALLING AND INITIALIZATION

This Section describes the initialisation and use of the OP-16 Utility Programs. It is assumed that the Utilities have been built in accordance with the building procedures described in Section IV.

Off-Line Utilities

1. Ensure that the utility program is in core.
2. Obtain the core address of location MINX from the loading map (refer to Section IV).
3. Set the STOP/RUN switch to STOP.
4. Master clear.
5. Set (Y) = MINX.
6. Set the STOP/RUN switch to RUN.
7. Depress the START pushbutton.

At this point the program goes into the "receive command" mode, causing SF= to be typed on the ASR. It remains in this mode until a legitimate command is received (see Conversation).

On-Line Utilities

1. Ensure that the utility program is in core.
2. Initialize the RTX-16 Executive as described in the OP-16 User Guide manual (see Preface).
3. Type a dollar sign ($) on the ASR. This places the utility program in the "receive command" mode, causing SF= to be typed on the ASR. It remains in this mode until a legitimate command is received (see Conversation).

CONVERSATION

The operator may answer SF= either by typing in any of the legitimate commands (refer to Section III) or by terminating the conversation. Once the command line is completed by a carriage return, the program performs the selected function and, after its completion, returns to the "receive command" mode, causing SF= to be typed again on the ASR. At this time the user may enter a subsequent command. The general command line formats, function mnemonics, and device names are described here; parameters are discussed in detail in Section III.

Command Line Formats

The two formats used in a system command line are presented below. The user may supply information for two fields in format (a) and for three fields in format (b).

(a) SF = <FN > Δ <parameters > (CR)

(b) SF = <FN > Δ <IDOD > Δ <parameters > (CR)

where SF= is output by the program to indicate the "receive command" mode

FN is a two-character function mnemonic entered by the user

Δ indicates a single space output by the program to divide the command line into fields

ID is a two-character input device name entered by the user

OD is a two-character output device name entered by the user

(CR) indicates a carriage return entered by the user.

Function Mnemonics

The 11 function mnemonics used with format (a) of the command line are presented in Table 2-1.

Two function mnemonics are used with format (b) of the command line:

TR Transfer Data

VE Verify Data

Table 2-1. Function Mnemonics

Mnemonic	Description
PT	Print Time
RT	Replace Time
RP	Request Program
CC	Connect Clock
DC	Disconnect Clock
RC	Replace Core
FC	Fill Core
SC	Search Core
PC	Print Core
PL	Print Core Protection Limits
RL	Replace Core Protection Limits

Device Names

Five device names are used for the variables ID and OD in format (b) of the command line. The allowable codes for these device names are presented in Table 2-2 and Appendix B.

Table 2-2. Device Names

Mnemonic Code	Devices
CO	Core
SM	System Mass Store
PR	Paper Tape Reader
PP	Paper Tape Punch
MT	Magnetic Tape

TERMINATION

The off-line utility programs may be terminated either during their execution or after the completion of the last function simply by setting the STOP/RUN switch to STOP. The program may be placed again in the "receive command" mode

by repeating the steps given in Calling and Initialization of off-Line Utilities, see page 2-1.

On-line utility programs may be terminated by typing an exclamation mark (!) on the keyboard and, if necessary, terminating the field with a carriage return (CR). At this time control is returned to the RTX-16 Executive and the utility program remains dormant until requested again by typing a dollar sign ($) on the ASR keyboard. If the on-line utility program is mass-store-resident, the core area it occupied during execution becomes available at this time for other programs.

ABORTION OF FUNCTIONS

In some cases the execution of the selected function may take some time (print core, read tape, punch tape, etc.) and the operator may wish to terminate it before it runs to completion. By setting sense switch 3 on the computer control panel, the following functions may be aborted: print core (PC), search core (SC), and all transfer (TR) and verify (VE) functions, except transfer between core and system mass store (TR COSM and TR SMCO). After abortion of these functions, the utility program goes into the "receive command" mode and displays SF= on the ASR.

COMMAND EDIT FEATURES

The utility programs recognize three control characters for command editing when typed as part of the command line: a slash (/), a commercial 'at' (@) and a back arrow (←). The use of these and other control characters is summarized in Table 2-3.

Table 2-3. Keyboard Control Functions

Keyboard Character	Function
$	(On-line utilities only). Requests the utility program. When started by the EXEC, the characters SF= are typed to signify the "receive command" mode.
!	(On-line utilities) Causes termination of the utility program when typed at any time the utility program is expecting information. (Off-line utilities) Causes SF= to be retyped.
/	Deletes the current line and causes a new SF= to be printed.
←	*Deletes the current character.
@	*Deletes the entry in the current field.
,	*Delinits the parameter field.
(CR)	*Terminates a command line.
	*These characters are applicable only to a command line beginning with SF=

ERROR DIAGNOSTICS

Errors detected and reported by the utility programs are indicated by the messages shown in Table 2-4 and Appendix B. When any of these errors is detected, the current function is terminated, the error message is printed, and the utility program returns to the "receive command" mode after printing SF=.

I/O errors detected by the device drivers are reported as follows.

On-Line Utilities

The on-line Error Print program prints the appropriate I/O error message as described in the OP-16 User Guide and appropriate Driver manual (see Preface). After an I/O error, both the initiated function and the utility program are terminated.

The utility program halts with the error code in the A register and the address of location CALL+1 in the B register. The X register will contain either the 2 - character name of the calling program or a negative number (in the case of calls for EP3). To continue using the utility program, it must be reinitialized as described on page 2-1.

Table 2-4. Utility Program Messages

Message	Meaning
SF=	System is ready to receive function and other parameters from the user.
FE	Format or function error: parameter entered incorrectly.
LE	Limit error: attempt to modify core outside core protection limits.
HE	Header error: incorrect header record for the particular type of paper tape or magnetic tape transfer.
CE	Check sum error detected during a paper tape function.
ME	Magnetic tape error: magnetic tape driver could not complete the requested operation, because of a hardware malfunction.
ET	End of tape: end of tape mark encountered on magnetic tape.
EF	End of file: end of file mark encountered on magnetic tape, indicating that the file contains less than the specified data.
E100KB	The utility program was unable to complete the requested function and has been terminated.

SECTION III
UTILITY FUNCTIONS

INTRODUCTION

This section describes the utility functions available to the user. The functions are considered as three groups:

1. System Control Functions
2. Debugging Functions
3. Transfer and Verification Functions

Table 3-1 lists some common abbreviations used to simplify the parameter field descriptions. Other abbreviations have been described in Section II.

Table 3-1. Parameter Abbreviations

Abbreviation	Description	Variable
CSA	Core Start Address given in	Octal
CEA	Core End Address given in	Octal
DSA	Mass Store Start Address given as	Octal Segment Number
DEA	Mass Store End Address given as	Octal Segment Number
SC	Segment Count given in	Octal
Unit	Magnetic Tape Unit Number any number	0 to 3
File	Magnetic Tape File Number given in	Octal

SYSTEM CONTROL FUNCTIONS

Print Time

SF=PT$_{\Delta}$(CR)

prints the system time in decimal hours and minutes, based on the 24-hour clock.

Replace Time

SF=RT$_{\Delta}$<hours>, <minutes> (CR)

replaces the system time with the decimal hours and minutes specified.

Request Program

SF=RP $_{\Delta}$ <name> (CR) or SF=RP $_{\Delta}$ <name>, <parameter> (CR)

requests the execution of a program and optionally passes an octal parameter to it. The program name can be any two ASCII characters.

Connect Clock

SF=CC $_{\Delta}$ <name>, <initial>, <interval>, <base> (CR)

connects a program to the system clock for automatic periodic execution. The name, initial, interval, and base parameters are described in Table 3-2.

Table 3-2. Parameters for Clock Calls

Parameter		Meaning
Name		Name of program to be connected (two ASCII characters).
Initial		Time (decimal) until first execution (see Base parameter).
Interval		Interval (decimal) between subsequent executions (see Base parameter).
Base		Base frequency as follows:
	0	Time until first execution is an absolute time of day in minutes. Thereafter, interval between executions is in minutes.
	1	Time until first execution is in 50ms units. Interval between executions is in 50ms units.
	2	Time until first execution is in seconds. Interval between executions is in seconds.
	3	Time until first execution is in minutes. Interval between executions is in minutes.

Disconnect Clock

SF=DC $_{\Delta}$ <name>, <base> (CR)

disconnects a program from the system clock. The name and base parameters are defined in Table 3-2.

Replace Core

SF=RC $_{\Delta}$ <CSA> (CR)

prints and optionally replaces the contents of an area of core, starting at a specified address, depending on the user's response. Table 3-3 lists the possible responses and the corresponding action taken. A typical printout is as follows:

```
SF = RC    10000
10000      123456 111111
10001      123456 ,
10002      123456
10003      123456 /
```

Table 3-3. Responses for Replace Core Function

Response	Action
, (CR)	Contents of the current address remain unchanged and the next higher address is examined.
<value> (CR)	The octal value specified is stored in the current address and the next higher address is examined.
(CR)	Zero is stored in the current address and the next higher address is examined.
/ (CR)	The function is terminated.

Print Core

SF=PC $_{\Delta}$ <CSA> , <CEA> (CR) or SF=PC $_{\Delta}$ <CSA> (CR)

prints the contents of a specified number of core locations (nine per line) in octal. If the core end address (CEA) is omitted, only a single location will be printed. A typical printout is shown below:

```
    SF = PC 7000,7777
7000   7160     5000    11005     5022     3007     7162     5023    11134
7010 105005    11133    25005   105005    11135     5024    11122    21117
7020  25005   103005        4        1   140723    11274     7154   105026
7030  11134    25026   105026    11133    25026   105026   101400     3043
7040 140100    11026   100000    25026    11135     5051    11122    21117
```

Fill Core

SF=FC $_{\Delta}$ <CSA> , < CEA > , <value > (CR)

fills a specified block of core with a given octal value.

Search Core

SF=SC $_{\Delta}$ <CSA > , <CEA > , <value > , <mask > (CR)

searches a specified area of core for a given octal value under a mask. If a mask of zero is specified, the entire word is tested. Whenever a match is found, the address and its contents are printed as shown below:

```
SF=SC 7000,7777,7000,7000
  7000    7160
  7005    7246
  7026    7154
  7074  177777
  7107  107154
```

Print Limits

SF=PL $_{\Delta}$ (CR)

prints the core protection limits. Utility functions are prevented from modifying core locations outside these limits. The fourth and fifth words of Table XSPT in the OP-16 Configuration Module, or Table SPT in MINX-716, contain the low and high limits, respectively.

Replace Limits

SF=RL $_{\Delta}$ < low limit > , <high limit > (CR)

replaces the core protection limits with those specified.

TRANSFER AND VERIFY FUNCTIONS

Transfer - Core to Mass Store

SF=TR $_{\Delta}$ COSM $_{\Delta}$ <CSA > , < DSA > , <SC > (CR)

transfers a specified number of segments from core to the mass store.

Transfer - Mass Store to Core

SF=TR $_{\Delta}$ SMCO $_{\Delta}$ <CSA> , <DSA> , <SC> (CR)

transfers a specified number of segments from the mass store into core.

Verify - Mass Store Against Core

SF=VE $_{\Delta}$ SMCO $_{\Delta}$ <CSA> , <DSA> , <SC> (CR)

verifies a specified number of segments on the mass store against core. Differences are printed on the ASR in the following format:

```
SF=VE SMCO 5000,477,1
  5000    5172       0
  5005    5257       0
  5025       4    5155
  5026    5230       0
  5052    5243       0
  5057    5245       0
  5074  177777   11147
  5075       0    5102
  5076       0   11134
  5077  105144   21131
```

where column 1 shows the core address

column 2 shows the core contents

column 3 shows the corresponding mass store contents.

Transfer - Core to Paper Tape

SF=TR $_{\Delta}$ COPP $_{\Delta}$ <CSA> , <CEA> (CR)

transfers a specified area of core to paper tape using the format described in Appendix C.

Transfer - Paper Tape to Core

SF=TR $_{\Delta}$ PRCO $_{\Delta}$ (CR)

transfers information from paper tape into the area of core from which it was punched.

SF=TR $_{\Delta}$ PRCO $_{\Delta}$ <CSA> , <CEA> (CR)

transfers information from paper tape into another specified area of core. Information punched from the mass store may also be transferred to core by use of this form of

the command. The transfer terminates either when the core area is filled or when the end-of-tape record is encountered.

Verify - Paper Tape Against Core

SF=VE $_{\Delta}$ PRCO $_{\Delta}$ (CR)

verifies information from paper tape against the area of core from which it was punched.

SF=VE $_{\Delta}$ PRCO $_{\Delta}$ <CSA> , <CEA> (CR)

verifies information from paper tape against another specified area of core. Information punched from the mass store may also be verified against core by use of this form of the command. The verification terminates either at the core end address (CEA) or when the end-of-tape record is encountered.

In both cases, differences are printed on the ASR in the following format:

```
SF=VE PRCO 5000,5177
  5057    5245    5255
  5134  150322  150320
  5145      42      41
  5146       1    5363
  5147    5514    5363
  5160     533     527
  5170    5472    5343
  5173    5514    5363
  5174   21336   21271
  5175  100040   17362
```

where column 1 shows the core address

column 2 shows core contents

column 3 shows the corresponding tape contents.

Transfer - Mass Store to Paper Tape

SF=TR $_{\Delta}$ SMPP $_{\Delta}$ <DSA> , <DEA> (CR)

transfers a specified number of segments from the mass store to paper tape using the format described in Appendix C.

Transfer - Paper Tape to Mass Store

SF=TR $_{\Delta}$ PRSM $_{\Delta}$ (CR)

transfers information from paper tape to the mass-store segments from which it was punched.

SF=TR $_{\Delta}$ PRSM $_{\Delta}$ <DSA> , <DEA> (CR)

transfers information from paper tape to other specified mass-store segments. Information punched from core may also be transferred to the mass store by use of this form of the command. The transfer terminates either at the end address (DEA) or when the end-of-tape record is encountered.

Verify - Paper Tape Against Mass Store

SF=VE $_{\Delta}$ PRSM $_{\Delta}$ (CR)

verifies information from paper tape against the mass-store segments from which it was punched.

SF=VE $_{\Delta}$ PRSM $_{\Delta}$ <DSA> , <DEA> (CR)

verifies information from paper tape against other specified mass-store segments. Information punched from core may also be verified against the mass store by use of this form of the command. The verification terminates either at the end address (DEA) or when the end-of-tape record is encountered. In both cases differences are printed on the ASR in the following format:

```
SF=VE PRSM 477,477
   477      0      0   5172
   477      5      0   5174
   477     25   5155      4
   477     26      0   5166
   477     52      0   5811
   477     57      0   5255
   477     74  11147 177777
   477     75   5102      0
   477     76  11134      0
   477     77  21131 105144
```

where column 1 shows the mass-store segment number

column 2 shows the relative word within the segment (0-127)

column 3 shows the mass-store contents

column 4 shows the corresponding tape contents.

Transfer - Core to Magnetic Tape

SF=TR $_{\Delta}$ COMT $_{\Delta}$ <unit> , <file> , <CSA> , <CEA> (CR)

transfers a specified area of core to magnetic tape using the format described in Appendix C.

Transfer - Magnetic Tape to Core

SF=TR $_{\Delta}$ MTCO $_{\Delta}$ <unit> , <file> (CR)

transfers information from magnetic tape into the area of core from which it was written.

SF=TR $_{\Delta}$ MTCO $_{\Delta}$ <unit> , <file> , <CSA> , <CEA> (CR)

transfers information from magnetic tape into another specified area of core. The transfer terminates either at the core end address (CEA) or when the end-of-file record is encountered.

Verify - Magnetic Tape Against Core

SF=VE $_{\Delta}$ MTCO $_{\Delta}$ <unit> , <file> (CR)

verifies information from magnetic tape against the area of core from which it was written.

SF=VE $_{\Delta}$ MTCO $_{\Delta}$ <unit> , <file> , <CSA> , <CEA> (CR)

verifies information from magnetic tape against another specified area of core. The verification terminates either at the core end address (CEA) or when the end-of-file record is encountered.

In both cases, differences are printed on the ASR in the following format:

```
SF=VE MTCO 0,1,5000,5177
  5000    5177    5172
  5005    5237    5174
  5026    5173    5166
  5145      27      24
  5146       1       4
  5147    5422    5466
  5160     551     545
  5166    3173  121162
  5167       0       1
  5170  177770    5446
```

where column 1 shows the core address

column 2 shows the core contents

column 3 shows the corresponding magnetic tape contents.

Transfer - Mass Store to Magnetic Tape

SF=TR $_{\Delta}$ SMMT $_{\Delta}$ <unit> , <file> , <DSA> , <DEA> (CR)

transfers a specified number of segments from the mass store to magnetic tape.

Transfer - Magnetic Tape to Mass Store

SF=TR $_{\Delta}$ MTSM $_{\Delta}$ <unit> , <file> (CR)

transfers information from magnetic tape to the mass-store segments from which it is written.

SF=TR $_{\Delta}$ MTSM $_{\Delta}$ <unit> , <file> , <DSA> , <DEA> (CR)

transfers information from magnetic tape to other specified mass-store segments. The transfer terminates either at the end address (DEA) or when the end-of-file record is encountered.

Verify - Magnetic Tape Against Mass Store

SF=VE $_{\Delta}$ MTSM $_{\Delta}$ <unit> , <file> (CR)

verifies information from magnetic tape against the mass-store segments from which it was written.

SF=VE $_{\Delta}$ MTSM $_{\Delta}$ <unit> , <file> , <DSA> , <DEA> (CR)

verifies information from magnetic tape against other specified mass-store segments. The verification terminates either at the end address (DEA) or when the end-of-file is encountered.

In both cases, differences are printed on the ASR in the following format:

```
SF = VE MTSM 0,8,147,147
     147       0    1000040   177777
     147       1       3204        0
     147       2      25705        0
     147       3       3314        0
     147       4       5743        0
     147       5     100040        0
     147       6       3221   105702
     147       7       5710   141206
     147      10     100040    11746
     147      11       3215   105746
```

where column 1 shows the mass-store segment number

column 2 shows the relative word within the segment (0-127)

column 3 shows the mass-store contents

column 4 shows the corresponding magnetic tape contents.

SAMPLE PRINTOUT

A sample interchange between the user and system using some of the above-mentioned OP-16 utility functions is presented below.

```
SF = PT
0000
SF = RT 3,9
SF = PT
0309
SF = PL
     3654 3655
SF = RL 7000,12777
SF = PL
        7000 12777
SF = RC 6777                          (Limit Error - See Appendix B)

LE
SF = RC 7000
     7000 177777 ,
     7001      0 /
SF = RC 12776
     12776 177762 ,
     12777 177772 ,                   (Limit Error - See Appendix B)
LE
SF = TR COPP 100,147
SF = TR PRCO

LE
SF = VE PRCO                          (Limit Error - See Appendix B)
SF = PT
0312
```

SECTION IV
UTILITY BUILDING PROCEDURES

INTRODUCTION

This section describes the building procedures for OP-16 utility programs. Reference is made to the Series 16 Equipment Operators manual (see preface) for detailed procedural information and the use of the loader program LDR-APM and the punch and load program PAL-AP. It will be appreciated that depending on the user's computer installation, LDR-APM may be used to load object programs into core by means of the ASR teletype, high speed paper tape reader, magnetic tape unit, disk unit or card reader. Core output, in the form of self-loading system tape is obtained via the punch and load program PAL-AP.

This section also gives details for building certain utilities with the aid of BOS, the Batch Operating System. This facility assumes the availability of a disk unit within the user's computer installation (or access to a disk system), and that initially, all files to be called by the BOS Loader ($LO) are first established on disk. Attention is drawn to the File Name convention, whereby an object code for a program XY-Z is referred to in BOS as OXYZ.

A summary of the various types of utility programs is included in Section I of this manual, together with a list of available functions (see Table 1-1).

Off-line versions of the OP-16 utility program (OFLCUP and OFLMUP) are operated under the control of MINX-716. These versions are relocatable and should be built without using any sector 0 locations for cross-sector indirect links. MINX-716 has links to the following external programs:

Link Name	External Program
KB	Utility Program
AS	ASR Driver
PR	Paper Tape Reader Driver
PP	Paper Tape Punch Driver
SM	System Mass-Store Driver
MT	Magnetic Tape Driver

The external programs identified above must be core-resident. During the building procedure for off-line versions, the KB and AS links must always be satisfied. The other external program links can remain unsatisfied during loading if the device is not included in the utility program. For example, the SM link need not be included if the utility does not include the mass-store functions.

OFLUT-1H

The OFLUT-1H tape (Doc. No. 41286639-000A) contains 19 objects. The following simple procedure will build an OFLUT-1H utility in five sectors (YA, YB, YC, YD and YE) specified by the user. LDR-APM is assumed to be in core, with normal entrance at XX000.

1. Insert OFLUT-1H in the paper tape reader.
2. Set (P) = XX006 and (A) = YA000. Load the first object program into sector YA.
3. Set (P) = XX003 and (A) = YB000. Load the next object into sector YB.
4. Set (P) = XX003 and (A) = YC000. Load the next two objects into sector YC.
5. Set (P) = XX003, (A) = YD000, and (B) = YD770. Load the next six objects into sector YD.
6. Set (P) = XX003, (A) = YE000, and (B) = YE700. Load the next nine objects into sector YE.
7. Obtain a memory map, (P) = XX002.
8. Use PAL-AP to punch a self-loading tape of OFLUT-1H.

OFLUT-2H/OFLUT-3H

OFLUT-2H and OFLUT-3H tapes (Doc. Nos. 41286640-000A and 41206106-514A) each contain 21 objects. The following simple procedure will build an OFLUT-2H or OFLUT-3H utility in six sectors (YA, YB, YC, YD, YE, and YF) specified by the user. LDR-APM is assumed to be in core, with normal entrance at XX000.

1. Insert OFLUT-2H or OFLUT-3H in the paper tape reader.
2. Set (P) = XX006 and (A) = YA000. Load the first object program into sector YA.
3. Set (P) = XX003 and (A) = YB000. Load the next object into sector YB.
4. Set (P) = XX003 and (A) = YC000. Load the next two objects into sector YC.
5. Set (P) = XX003 and (A) = YD000. Load the next object into sector YD.
6. Set (P) = XX003, (A) = YE000, and (B) = YE770. Load the next five objects into sector YE.
7. Set (P) = XX003, (A) = YF000, and (B) = YF550. Load the next eleven objects into sector YF.
8. Obtain a memory map, (P) = XX002.
9. Refer to Appendix F.
10. Use PAL-AP to punch a self-loading tape of OFLUT-2H or -3H.

OFLCUP

Program Structure

Two typical core layouts for OFLCUP are presented in Figures 4-1a and 4-1b. The procedure for building both is described below.

OFLCUP is constructed in consecutive sectors beginning at location YA000. It is assumed that LDR-APM is in core, with normal entrance at XX000, where XX represents the highest sector occupied by the loader.

The following tapes are required by this procedure:

Tape Name	Document No.
MINX-716	41205679
ROOT-C	70183081321
TR-C	70183082321
OFLCT	70183084321
FCNLST-C	70183071521
IOD-C	70183092321
OP-C	70183091321
SUBR2-C	70183074521
SUBR1-C	70183072521
IOSLST	70183001521
B2-C	70183101321
ASRD-S or ASRD-L	70181475321 70181476321
HSRD-H	41286258-001 -02
HSPD-H	41286259-001 -03
MHSD or DDAM-C	70181774321 41204915

Construction of OFLCUP

Perform the following steps in the order shown:

1. Load MINX-716 by setting (P) = XX006 and (A) = YA000 and pressing START.
2. If the system uses the ASR reader/punch, proceed to step 3. If not, load ASRD-S by setting (P) = XX003 and (A) = YB000, and pressing START.
3. If the system contains a fixed-head disk, load DDAM-C by pressing START.
4. Load ROOT-C by setting (P) = XX003, (A) = YC000, and (B) = YC770 and pressing START.
5. Load TR-C by pressing START.
6. Load OFLCT by pressing START.
7. Load FCNLST-C by pressing START six times. Routines RC-C, PC-C, and PL-C are loaded.

8. Load IOD-C by pressing START.

9. Load SUBR2-C by pressing START four times. Routines OE-C and OD-C are loaded.

10. Load OP-C by setting (P) = XX003, (A) = YD000, and (B) = YD670 and pressing START.

11. Load SUBR1-C by pressing START four times. Routines GET-C, PUT-C, CMN-C, and CL-C are loaded.

12. Load routines AKPS and SMSS from IOSLST by pressing START twice.

    If the system uses the high speed reader/punch, load routine HRPS from IOSLST by pressing START. Then remove the tape from the reader and proceed to step 15.

    If the system uses the ASR reader/punch, move IOSLST forward by hand to bypass routine HRPS. Then press START to load routine ARPS.

13. Load B2-C by pressing START.

14. Load ASRD-L by setting (P) = XX003 and (A) = YE000 and pressing START. Proceed to step 18.

15. Load B2-C by pressing START.

16. Load HSRD-H by setting (P) = XX003 and (A) = YE000 and pressing START.

17. Load HSPD-H by pressing START.

18. Load MHSD (if the system contains a moving-head disk) by setting (P) = XX003 and (A) = YF000 and pressing START.

19. Obtain a memory map. A typical memory map is presented in Figure 4-2 for YA=1 and YE=5.

20. Refer to Appendix F.

21. Use PAL-AP to punch a self-loading tape of OFLCUP.

BOS Construction of OFLCUP

The following procedure indicates how OFLCUP can be constructed under BOS. The file names for OFLCUP obey the convention described in the Introduction to this section. The system described corresponds to Figure 4-1a.

1. Establish a source file containing the following records:
   *X, *A = YA000, OMINX, *G, OASRDS, ODDAMC
   *G, *B = YC770, OROOTC, OTRC, OOFLCT, ORCC, OPCC
   OPLC, OIODC, OOEC, OODC

*G, *B = YD670, OOPC, OGETC, OPUTC, OCMNC, OCLC, OAKPS, OSMSS, OHRPS, OB2C
*G, OHSRDH, OHSPDH
*M = MAP, *R = OFLCUP, YA000

2. Issue the following BOS commands:
$LO *W = <Workfile above>
$OU MAP, SO (there will be certain unsatisified references but such routines are not used)
$OU OFLCUP, BO (to contain a paper tape of OFLCUP).

NOTE

For an explanation of the BOS interpretation of the load instructions, refer to the BOS manual (see preface).

ONLCUP

Program Structure

ONLCUP may be configured at load time to support from one to fifteen functions. Configuration is achieved either by using the supplied component FT-C (Figure 4-5) and only loading the desired functions, or by generating a new version of FT-C including only the desired functions.

It should be noted that ONLCUP is designed to be loaded separately from the main OP-16 system, as certain cross-references are duplicated in EXEC-716 and the utility modules. This can be done, using LDR-APM, by reinitializing between utility and main system load or, under BOS, by use of the *I parameter of $LO; refer to the BOS manual (see Preface).

The core layout will vary with each configuration and must be performed prior to loading. Table 4-1 contains information on size and components required. If the program exceeds one sector, the user must desectorize it by choosing appropriate bases in each sector for cross-sector indirect links.

Three typical core layouts for ONLCUP are presented in Figures 4-3a through 4-3c. The procedure for building the first is described below. The other two are provided only as examples.

ONLCUP is constructed in consecutive sectors beginning at location YA000. It is assumed that LDR-APM is in core, with normal entrance at XX000, where XX represents the highest sector occupied by the loader.

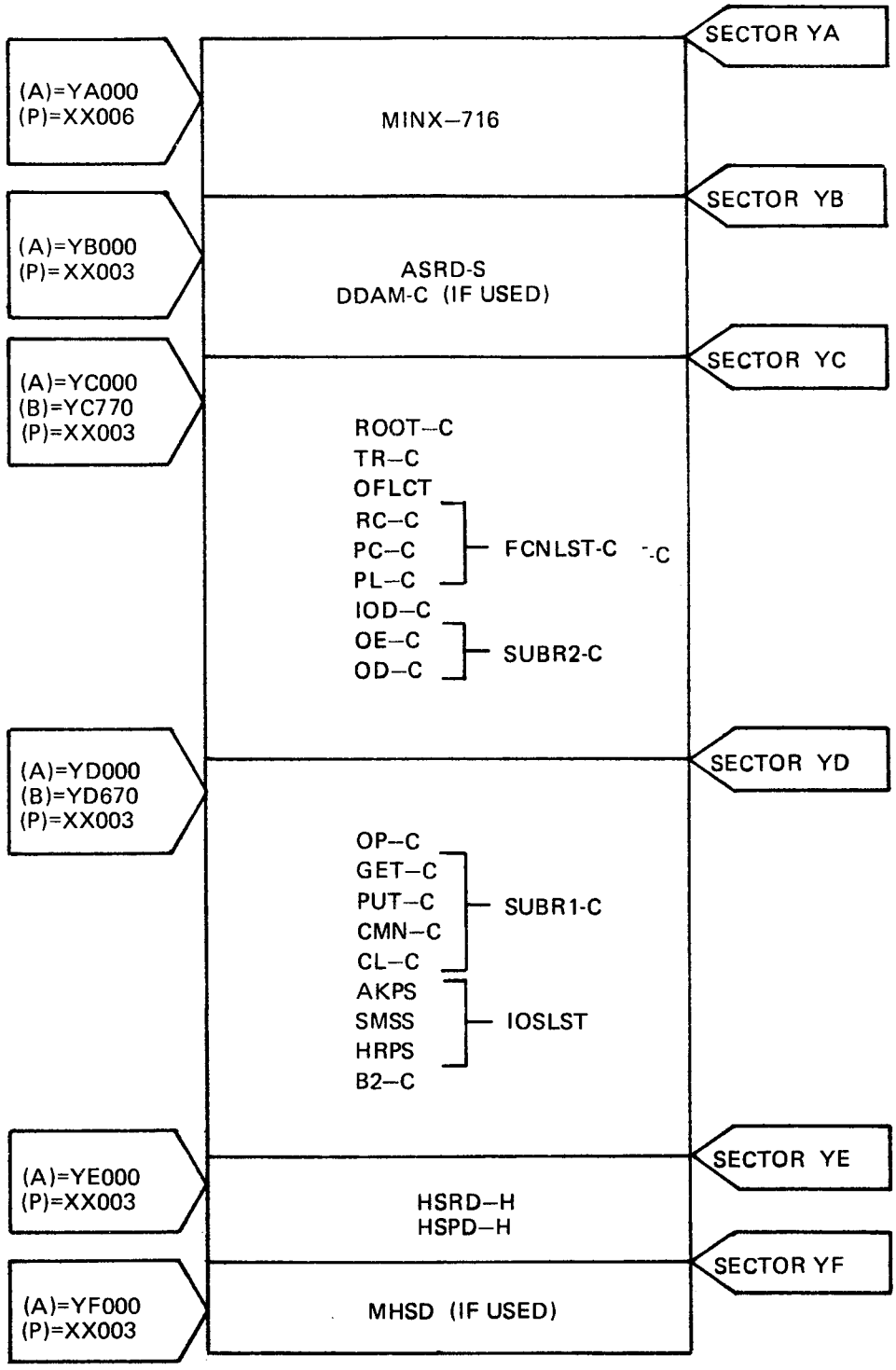


Figure 4–1a. Typical Core Layout of OFLCUP with High Speed Paper Tape Functions

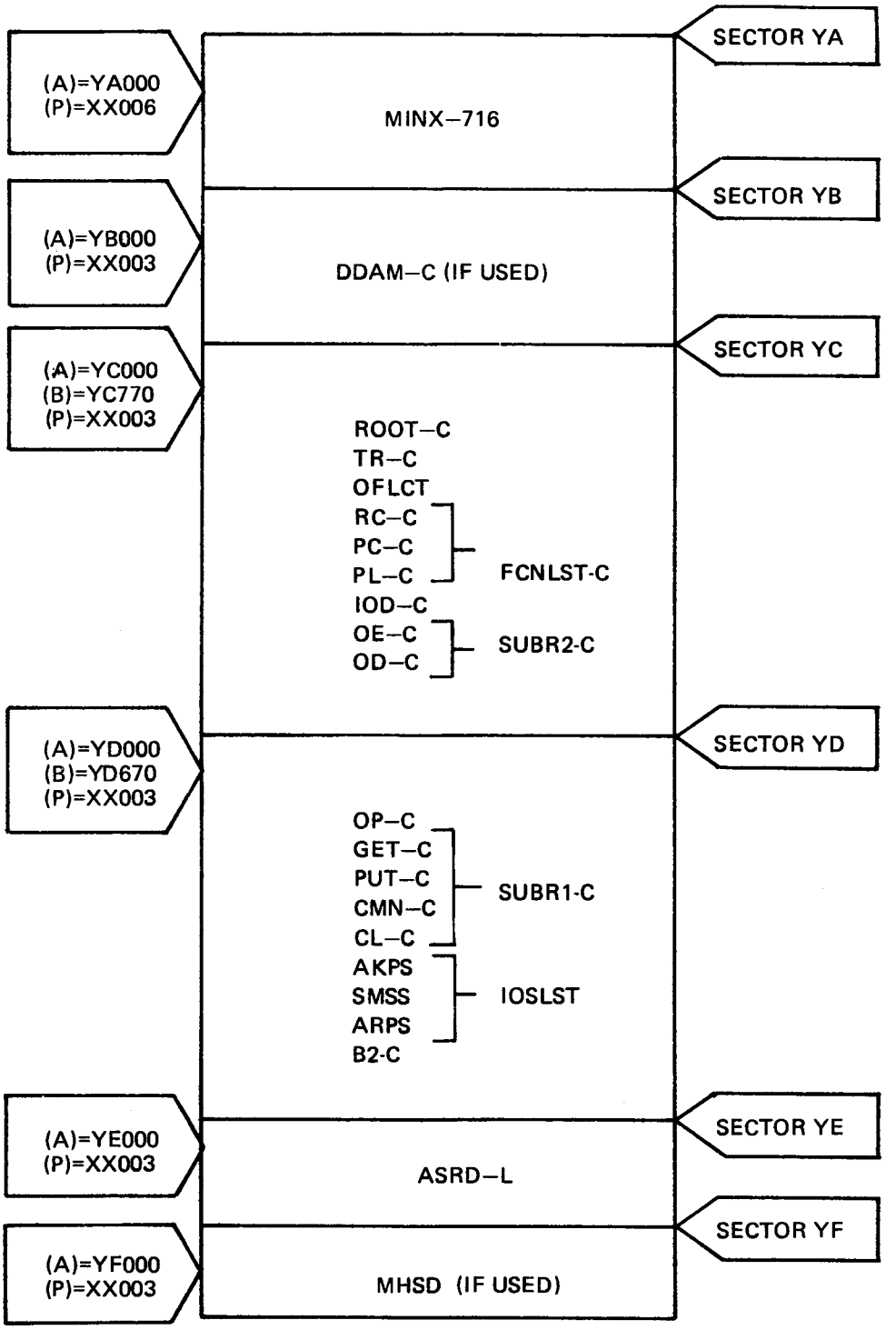


Figure 4–1b. Typical Core Layout of OFLCUP with ASR Paper Tape Functions

```
*BOTM   01000
*LOW    01000
*START  01000
*HIGH   05726
*TOP    05726
*NAMES  25143
*COMN   37000
*BASE   04724
*BASE   03774
MINX    01000
XLNK    01411
EROR    01412
EPM3    01413
XSPT    01415
CHL1    01466
CHL2    01504
MT      01527**
CHL3    01532
CHL4    01533
AS      02006
ASF1    02343**
ASF2    02344**
SM      02370
KB      03003
FD2     03042
ATSL    03061
ERR     03071
TERM    03076
RPRG    03100
PNME    03103
PAR3    03114
PAR4    03115
PAR5    03116
PAR6    03117
FE      03120
RT1     03120
ER      03121
PRT     03126
DRIV    03126
C1      03130
A1      03131
KBD     03133
CRLF    03140
COMA    03145
CRCK    03152
SP      03162
P1      03164
P2      03165
CSA     03165
P3      03166
CEA     03166
S1      03167
S2      03170
S3      03171
LL      03172
HL      03173
AP      03177
AK      03200
IDN     03210
B1      03210
ODN     03211
TR      03250
FT      03312
RC      03354
PC      03410
PL      03470
RL      03503
ODK     03536
IDK     03546
VDK     03557
OE      03650
OD      03674
OP      04000
IP      04071
VP      04127
GET     04274
PUT     04306
CMN     04322
CL      04336
AKS     04352
APS     04357
SMS     04400
PRS     04424
PPS     04431
B2      04446
CKSM    04506
EOF     04507
PR      05006
C$OP    05356
C$IP    05356
PP      05366
```

Figure 4-2. Typical OFLCUP Memory Map

The following tapes are required by this procedure:

Tape Name	Document No.
ROOT-C	70183081321
TR-C	70183082321
FT-C	70183083321
FCNLST-C	70183071521
OP-C	70183091321
SUBR2-C	70183074521
SUBR1-C	70183072521
IOSLST	70183001521
B2-C	70183101321
XLOCS-716	41204887

Construction of ONLCUP

Perform the following steps in the order shown:

1. Load ROOT-C by setting (P) = XX006, (A) = YA000, and (B) = YA770 and pressing START.

2. Load TR-C by pressing START.

3. Load FT-C by pressing START.

4. Load FCNLST-C by pressing START six times. Routines PT-C, RP-C, CC-C, RC-C, PC-C and PL-C are loaded.

5. Load OP-C by setting (P) = XX003, (A) = YB000, and (B) = YB632 and pressing START.

6. Load SUBR2-C by pressing START four times. Routines OE-C, OD-C, AE-C, and DE-C are loaded.

7. Load SUBR1-C by pressing START four times. Routines GET-C, PUT-C, CMN-C, and CL-C are loaded.

8. Load routine AKPS from IOSLST by pressing START twice. Routine SMSS is bypassed.

   If the system uses the high speed reader/punch, load routine HRPS from IOSLST by pressing START. Then remove the tape from the reader.

   If the system uses the ASR reader/punch, move IOSLST forward by hand to bypass routine HRPS. Then press START to load routine ARPS.

9. Load B2-C by pressing START.

10. Load XLOCS-716 by pressing START.

11. Obtain a memory map. A typical map for YA=5 and YB=6 is presented in Figure 4-4. Note that address KB is the program start address to be inserted in word 2 of the XPLT table entry for this program (e.g., 5003).

12. Use PAL-AP to punch a self-loading tape of ONLCUP.

The following statements apply with regard to the coordination of ONLCUP:

1. Coordination must not be used for the ASR device.

2. Coordination may be used for the ASR driver to prevent conflicts with other programs which call the ASR driver.

BOS Construction on ONLCUP

The following procedure indicates how ONLCUP can be constructed under BOS. The file names for ONLCUP obey the conventions described in the Introduction of this section.

1. Establish a source file containing the following records:
*X, *A = YA000, *B = YA770, OROOTC, OTRC, OFTC, OPTC, ORPC
OCCC, ORCC, OPCC, OPLC
*P = YB000, *B = YB632, OOPC, OOEC, OODC, OAEC, ODEC
OGETC, OPUTC, OCMNC, OCLC, OAKPS, OHRPS
OB2C, OXLOCS, *M = MAP, *R = ONLCUP, 1000

2. Perform the following BOS commands:
$LO *W = <Workfile above>
$OU MAP, SO
$OU ONLCUP, BO (to obtain a paper tape of ONLCUP)

NOTE

For an explanation of the BOS interpretation of the load instructions, refer to the BOS manual (see Preface).

It is possible to construct a BOS work file which loads both the OP-16 main system and the On-Line Utilities, provided that the load command *I (Initialize) separates main load from the utilities load. An example of such a work file together with notes on its construction, will be found in the OP-16 User Guide manual, see Section VII SYSTEM BUILDING, para- System Building with the Aid of BOS.

Table 4-1. Summary of Core-Resident Utility Components

Component	Size	Other Utility Components Required
RCOT-C	168	TR-C[1], FT-C (or OFLCT), AKPS, XLOCS
TR-C[1]	34	None
FT-C[2]	44	None

Table 4-1 (Cont). Summary of Core-Resident Utility Components

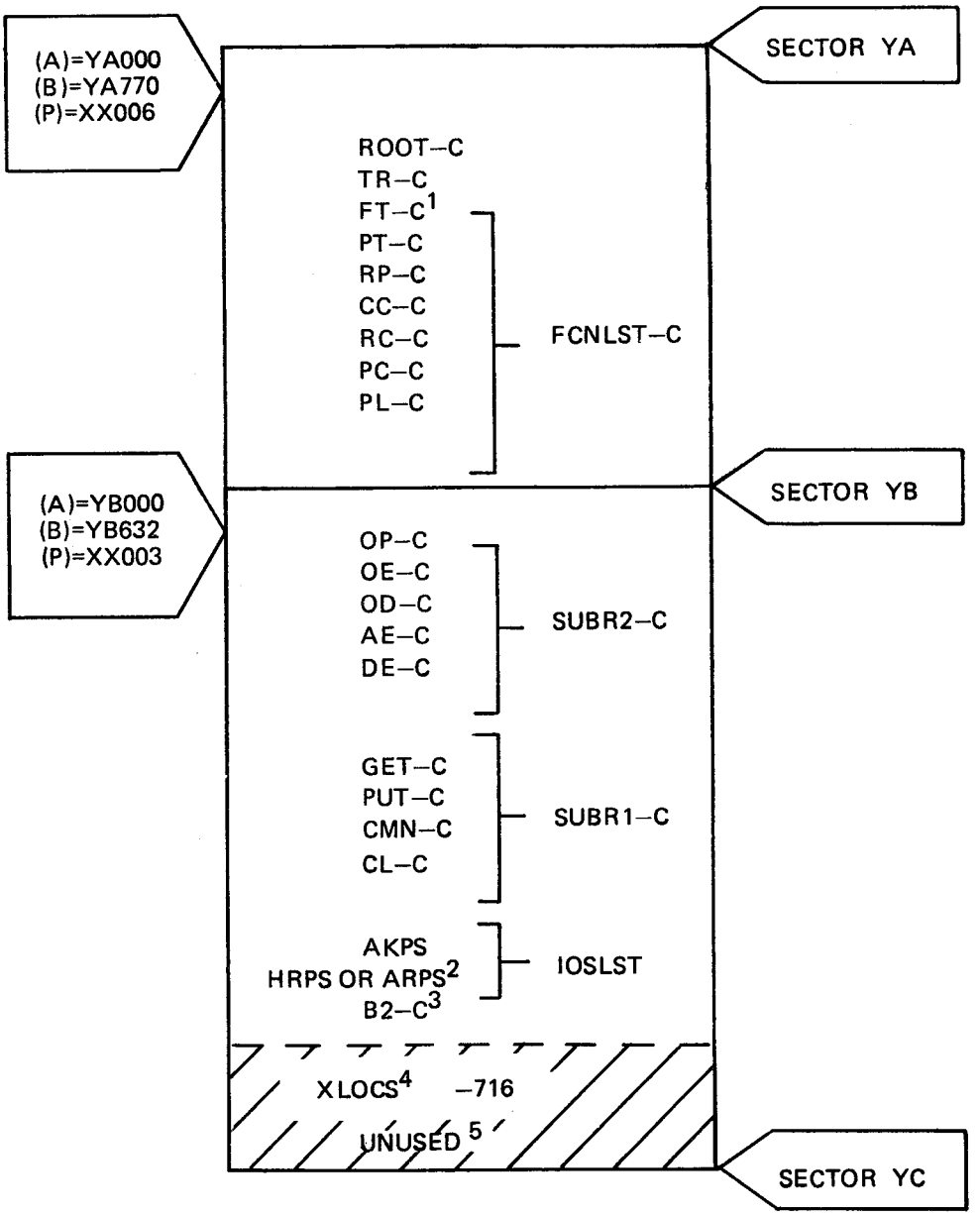
Component	Size	Other Utility Components Required
OFLCT[2]	34	None
PT-C	76	PUT-C, DE-C
RP-C	23	OE-C, AE-C
CC-C	39	AE-C, DE-C
RC-C	28	OE-C, CL-C, OD-C
PC-C	48	OE-C, OD-C
PL-C	38	OE-C, OD-C
OP-C	188	OE-C, CL-C, OD-C, HRPS[3] (or ARPS[3]), B2-C[4]
IOD-C	74	OE-C, CL-C, OD-C, SMSS, B2-C[4]
OE-C	20	GET-C
OD-C	32	PUT-C
AE-C	14	GET-C
DE-C	26	GET-C
GET-C	10	CMN-C
PUT-C	12	CMN-C
CMN-C	12	None
CL-C	12	None
AKPS	22	None
SMSS	20	None
HRPS[3]	18	None
ARPS[3]	28	None
B2-C[4]	34-128	None
XLOCS-716	0	None

[1]TR-C is required only if transfer (TR) and verify (VE) functions are to be included.

[2]FT-C and OFLCT are the on-line and off-line function tables, respectively; the sizes indicated are as supplied.

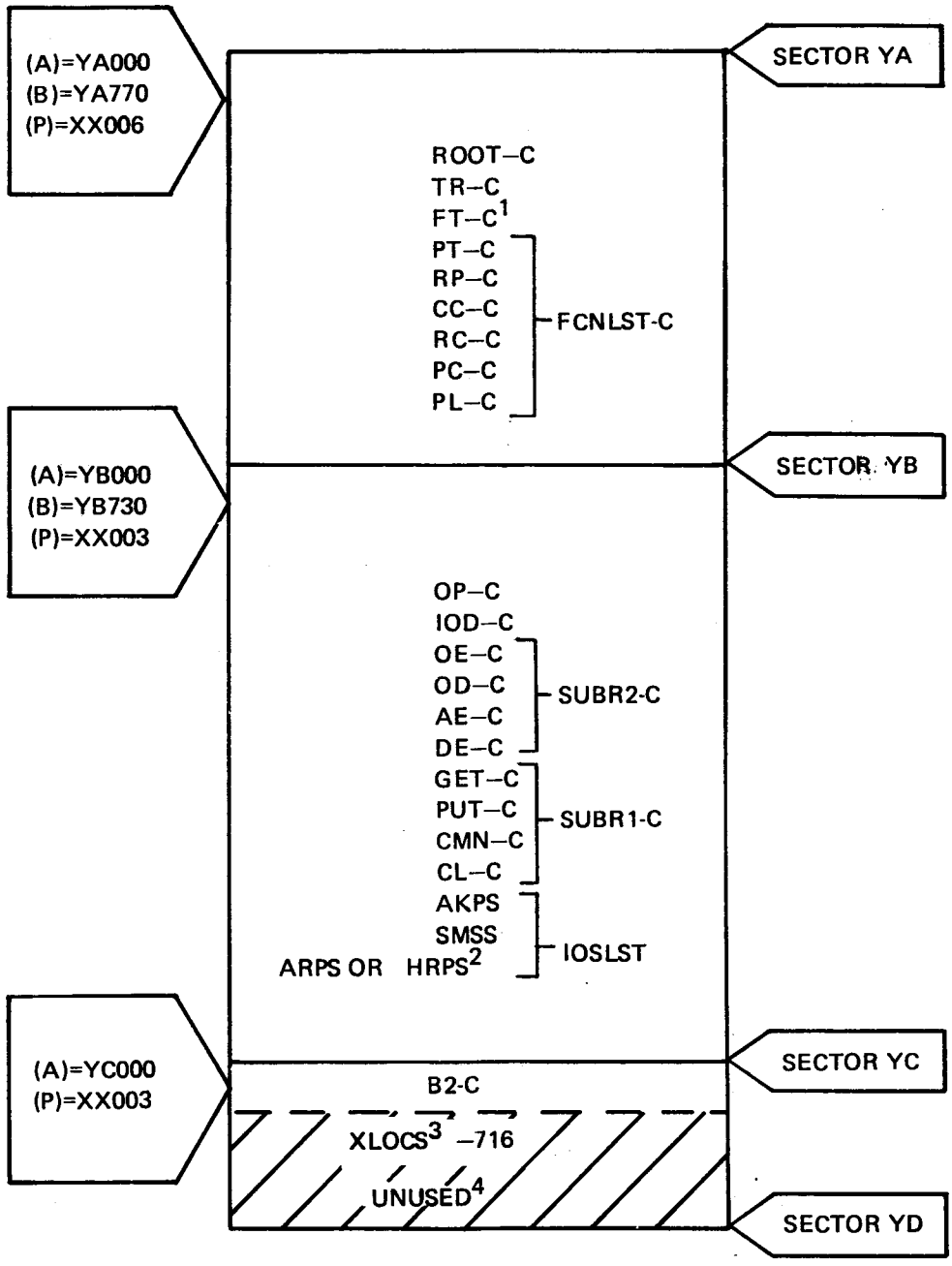
[3]HRPS and ARPS are mutually exclusive.

[4]B2-C is a BSS 128; IOD-C uses all 128 words, whereas OP-C uses only 34 words. Since a BSS does not alter the contents of core, B2-C may be as small as 34 words or as large as 128 words.



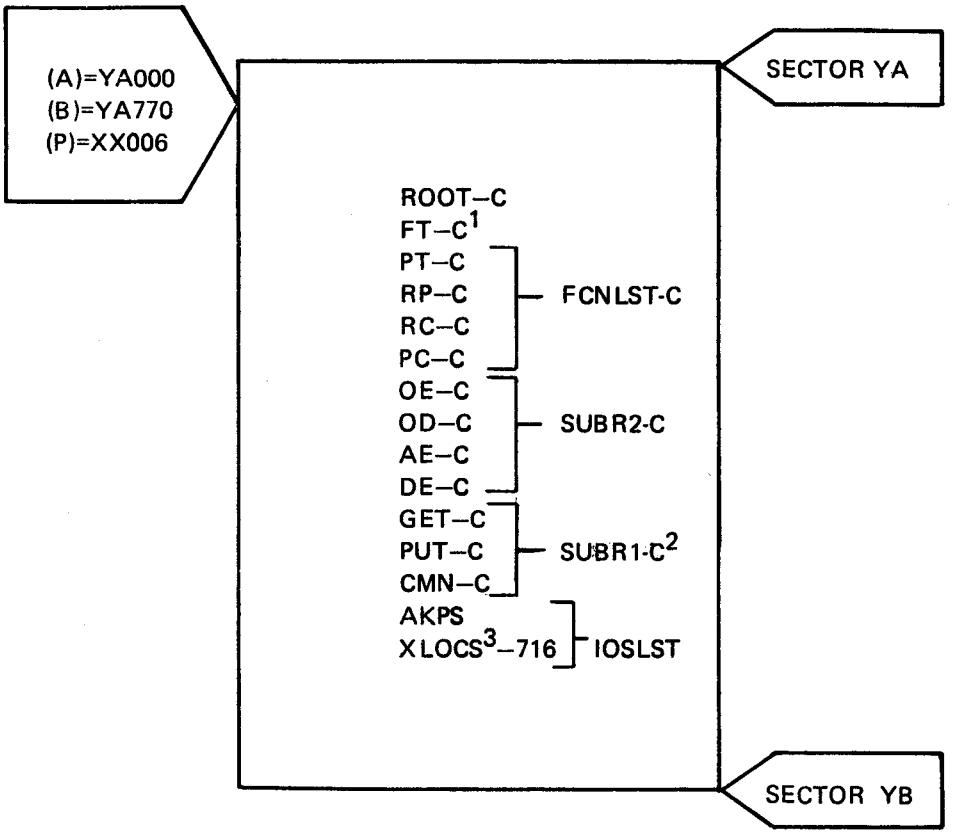
1. In this example, FT–C includes all those functions shown in Figure 4-5 (sheets 1 to 3); disk routines IOD–C and SMSS are not loaded.
2. The layout takes into account the size differential between HRPS and ARPS.
3. Buffer B2-C is a BSS 128, but only34 words are used in this example. The indirect links starting at YB632 are not destroyed.
4. XLOCS -716 consists entirely of EQU speudo-operations and does not add to the size of the program.
5. Locations 664 through 777 of sector YB are unused.

Figure 4-3a. Typical Core Layout of ONLCUP for Two Sectors



1. In this example, FT–C includes all those functions shown in Figure 4-5 (Sheets 1 to 3).
2. The layout takes into account the size differential between HRPS and ARPS.
3. XLOCS –716 consists entirely of EQU pseudo-operations and does not add to the size of the program.
4. Locations 200 through 777 of sector YC are unused.

Figure 4-3b. Typical Core Layout of ONLCUP for Three Sectors



1. In this example, FT–C consists only of functions PT, RT, RP, RC, and PC.
2. CL–C is omitted from SUBR1-C; location CL from the map must be replaced with a NOP instruction ('101000), where CL, for example, is YA441; there will be no protection from core modification by the RC function.
3. XLOCS 716 consists entirely of EQU pseudo-operations and does not add to the size of the program.

Figure 4-3c. Typical Core Layout of ONLCUP for One Sector

```
*BOTM   05000
*LOW    01000
*START  05000
*HIGH   06756
*TOP    06664
*NAMES  24571
*COMN   37000
*BASE   06664
*BASE   05776
XSTR    01000
XLNK    01001
XINT    01002
XMIL    01003
XSEC    01004
XMIN    01005
XHR     01006
XDAY    01007
XMSD    01010
XPIC    01011
XJBS    01012
XPLP    01013
XPSP    01014
XCCW    01015
XEPP    01016
EROR    01016
XLOD    01017
XPEP    01020
XSYS    01021
XICF    01022
COF1    01023
CIF1    01023
MTI1    01023
LPI1    01023
ASF1    01023
ROCA    01023
XSSA    01023
COF2    01024
CIF2    01024
MTI2    01024
LPI2    01024
ASF2    01024
ROCB    01024
XHLT    01024
XPLT    01025
XPCT    01026
XCUT    01027
XLAS    01030
XID1    01031
XID2    01032
XIVT    01033
XIVS    01034
XDCT    01035
XSPT    01036
XPET    01037
XCLC    01040
XHPT    01041
EPM3    01042
XILT    01044
KB      05003
FD2     05042
ATSL    05061
ERR     05071
TERM    05076
RPRG    05100
PNME    05103
PAR3    05114
PAR4    05115
PAR5    05116
PAR6    05117
FE      05120
RT1     05120
ER      05121
PRT     05126
DRIV    05126
C1      05130
A1      05131
KBD     05133
CRLF    05140
COMA    05145
CRCK    05152
SP      05162
P1      05164
P2      05165
CSA     05165
P3      05166
CEA     05166
S1      05167
S2      05170
S3      05171
LL      05172
HL      05173
AP      05177
AK      05200
IDN     05210
B1      05210
ODN     05211
TR      05250
FT      05312
ODK     05351**
IDK     05354**
VDK     05363**
PT      05366
RT      05413
RP      05502
CC      05532
DC      05562
RC      05602
PC      05636
PL      05716
RL      05731
OP      06000
IP      06071
VP      06127
OE      06274
OD      06320
AE      06360
DE      06376
GET     06430
PUT     06442
CMN     06456
CL      06472
AKS     06506
APS     06513
PRS     06534
PPS     06541
B2      06556
CKSM    06616
EOF     06617
```

Figure 4-4. Typical ONLCUP Memory Map

```
                                * NAME: FT-C          DOC 70183083000  REV A

0031                                 SUBR  FT
0032                                 REL
0033                            *
0034                            *    FUNCTION TABLE
0035                            *         WORD 1  2 CHARACTER FUNCTION NAME
0036                            *         WORD 2  START ADDRESS
0037                            *
0038 00000   150324        FT    BCI   1,PT              PRINT TIME
0039 00001   0 000000            XAC   PT
0040 00002   151324              BCI   1,RT              REPLACE TIME
0041 00003   0 000000            XAC   RT
0042 00004   151320              BCI   1,RP              REQUEST PROGRAM
0043 00005   0 000000            XAC   RP
0044 00006   141703              BCI   1,CC              CONNECT CLOCK
0045 00007   0 000000            XAC   CC
0046 00010   142303              BCI   1,DC              DISCONNECT CLOCK
0047 00011   0 000000            XAC   DC
0048 00012   151303              BCI   1,RC              REPLACE CORE
0049 00013   0 000000            XAC   RC
0050 00014   150303              BCI   1,PC              PRINT CORE
0051 00015   0 000000            XAC   PC
0052 00016   151314              BCI   1,RL              REPLACE LIMITS
0053 00017   0 000000            XAC   RL
0054 00020   150314              BCI   1,PL              PRINT LIMITS
0055 00021   0 000000            XAC   PL
0056 00022   152322              BCI   1,TR              TRANSFER DATA
0057 00023   -0 000027           DAC*  TT                POINTER TO TRANSFER TABLE
0058 00024   153305              BCI   1,VE              VERIFY DATA
0059 00025   -0 000044          DAC*  VT                POINTER TO VERIFY TABLE
0060 00026   000000             OCT   0                 END OF FUNCTION TABLE
0061                                 EJCT
```

Figure 4-5 (Sheet 1 of 3). Format for FT-C (Function, Transfer, and Verify Tables)

```
                                  * NAME: FT-C          DOC 70183083000  REV A

0062                              *
0063                              *     TRANSFER TABLE
0064                              *           WORD 1  2 CHARACTER INPUT DEVICE NAME
0065                              *           WORD 2  2 CHARACTER OUTPUT DEVICE NAME
0066                              *           WORD 3  START ADDRESS
0067                              *
0068 00027    150322        TT    BCI   2,PRCO             PAPER TAPE TO CORE
     00030    141717
0069 00031    0 000000            XAC   IP
0070 00032    141717              BCI   2,COPP             CORE TO PAPER TAPE
     00033    150320
0071 00034    0 000000            XAC   OP
0072 00035    141717              BCI   2,COSM             CORE TO MASS STORE
     00036    151715
0073 00037    0 000000            XAC   ODK
0074 00040    151715              BCI   2,SMCO             MASS STORE TO CORE
     00041    141717
0075 00042    0 000000            XAC   IDK
0076 00043    000000              OCT   0                  END OF TRANSFER TABLE
0077                              EJCT
```

Figure 4-5 (Sheet 2 of 3). Format for FT-C (Function, Transfer, and Verify Tables)

```
                                  * NAME: FT-C          DOC 70183083000  REV A

0078                              *
0079                              *     VERIFY TABLE
0080                              *           WORD 1  2 CHARACTER INPUT DEVICE NAME
0081                              *           WORD 2  2 CHARACTER OUTPUT DEVICE NAME
0082                              *           WORD 3  START ADDRESS
0083                              *
0084 00044    150322      VT      BCI     2,PRCO              PAPER TAPE AGAINST CORE
     00045    141717
0085 00046    0 000000            XAC     VP
0086 00047    151715              BCI     2,SMCO              MASS STORE AGAINST CORE
     00050    141717
0087 00051    0 000000            XAC     VDK
0088 00052    000000              OCT     0                   END OF VERIFY TABLE
0089                              END

FT      000000      TT      000027      VT      000044

0000 WARNING OR ERROR FLAGS
£DA 41285326-614 H
```

Figure 4-5 (Sheet 3 of 3). Format for FT-C (Function, Transfer, and Verify Tables)

Program Structure

Before performing this procedure, the user should become familiar with the program structure presented in Figure 4-6 and the mass-store layout presented in Figure 4-7. The mass-store layout is filled out for the full complement of 26 functions and is used to derive the component TABLES listed in Figure 4-8. For configurations consisting of less than a full complement of functions, the user must write a new version of TABLES and assemble it in the format shown in Figure 4-8.

The following tapes are required by this procedure (see Appendix E):

Tape Name	Document No.
IOSLST	70183001521
BASIC	70183006321
XLOCS-716	41204887
DECODE	70183008321
TABLES	70183009321
FCNLST1	70183000521
FCNLST2	70183010521

Construction of ONLMUP

It is assumed that LDR-APM is in core, with a normal entrance at XX000, where XX represents the highest sector occupied by the loader. Since OFLCUP[1] with core/mass-store functions is required in this procedure, it is assumed that the user has an OFLCUP tape available.

Perform the following steps in the sequence shown:

1. Load OFLCUP with core/mass-store functions.
2. Force-load the routines from IOSLST, starting at the beginning of sector YA:
   a. Load routine AKPS by setting (P) = XX006 and (A) = YA000 and pressing START.

[1]Wherever OFLCUP is referred to in this procedure, OFLUT-2H or OFLUT-3H may be substituted.

b. Load routine SMSS by setting (P) = XX004 and (A) = 0 and pressing START.

c. If the high speed reader/punch is utilized in the system, load routine HRPS by setting (P) = XX004 and (A) = 0 and pressing START. Remove IOSLST from the reader.

If the ASR reader/punch is utilized in the system, move IOSLST forward by hand, bypassing routine HRPS. Then load routine ARPS by setting (P) = XX004 and (A) = 0 and pressing START.

3. Load BASIC by pressing START.

4. Obtain a memory map. A typical memory map for YA000 = 5000 is shown below.

```
*LOW    05000
*START  05000
*HIGH   0514
*NAMES  22705
*COMN   26700
AKS     05000
AKS     05005
SMS     05026
PRS     05052
PPS     05057
KB      05077
RPRG    05117
PNMF    05122
PAR3    05133
PAR4    05134
PAR5    05135
OLAP    05140
ALNK    05143**
ASPT    05144**
EROR    05145**
OVLY    05154

MR
```

Note the addresses contained on the memory map for the following locations:

a. KB - program start address, inserted in word 2 of the XPLT table entry for this program.

```
*XPLT TABLE ENTRY:
KB    BCI    1,KB          PROGRAM NAME
      OCT    5077          START
      OCT    0             STATUS
      OCT    4245          OPTION
      OCT                  COORDINATION
      OCT    477           MASS STORE
```

b. OLAP - load address for all overlays.

c. OVLY - write to mass-store address for single overlay functions.

5. Load XLOCS-716 by pressing START. This will satisfy the calls to RTX-16 Executive locations XLNK, XSPT, and EROR and complete the loading of ROOT.

6. Use OFLCUP (entered at location MINX of the OFLCUP memory map) to write the ROOT to the mass store. A typical entry for this transfer is shown below, with (CSA) = 5000, (DSA) = 477, and (SC) = 1:

This information is used to complete words 4 and 6 of the XPLT table entry for this program (see step 4).

7. Load DECODE by setting (P) = XX006 and (A) = OLAP and pressing START.

8. Load TABLES by pressing START. This completes the loading of the decode overlay.

9. Obtain a memory map, which should appear as follows:

```
*LOW      05140
*START    05140
*HIGH     05442
*NAMES    23031
*COMN     26700
FD        05154
FT        05330
```

10. Use OFLCUP to write the decode overlay to mass store. A typical entry for (CSA) = 5154 (OVLY or FD), (DSA) = 500, and (SC) = 2 is presented below:

The segment number (500) used is the base segment number, for all overlays, which is entered in word 6 of table XSPT in the RTX-16 Configuration Module. A typical XSPT table is shown below:

```
*XSPT TABLE:
XSPT OCT  1023      POWER FAIL
     OCT  0         32K EXTENSION
     OCT  0         CORE LOW LIMIT
     OCT  77777     CORE HIGH LIMIT
     OCT  500       BASE SEGMENT
```

11. To load single overlays from the FCNLST1 and FCNLST2 tapes (e.g., RP or TRCOSM), set (P) = XX006 and (A) = OLAP, and press START. Obtain a memory map, which should appear as follows:

```
*LOW    05140
*START  05140
*HIGH   05340
*NAMES  23037
*COMN   26700
RP      05154

LC
```

The memory map shows that the overlay has been loaded properly at OLAP. Its name in this case is RP, for Request Program function. The address alongside the name should be the same as OVLY shown in step 4.

12. Use OFLCUP to write the function overlay to the mass store, using (CSA) = OVLY (or RP for the above example).

Refer to Figure 4-7 to obtain the size and absolute segment number for (SC) and (DSA), respectively. A sample of the OFLCUP entry using these values is shown below:

```
SF=TR COSM 5154,504,1
```

13. To load a function with multiple overlays from the FCNLST2 tape (e.g., VEMTSM), perform steps 14 through 20.

14. Load the first overlay of the function by setting (P) = XX006 and (A) = OLAP and pressing START.

15. Obtain a memory map, which should appear as follows:

```
*LOW    05140
*START  05140
*HIGH   05506
*NAMES  23031
VEMTSM  05154
OV1     05165

LC
```

The symbol OV1 indicates that the function has more than one overlay.

16. Use OFLCUP to write the first overlay to mass store from OVLY (or VEMTSM, as shown in step 15). Refer to Figure 4-7 to obtain the size and absolute segment number. These values are used in a sample OFLCUP entry, shown below:

```
SF=TR COSM 5154,555,2
```

17. Load the second overlay by setting (P) = XX006 and (A) = OLAP and pressing START.

18. Obtain a memory map, which should appear as follows:

```
*LOW    05140
*START  05140
*HIGH   05766
*NAMES  23031
*COMN   26700
VEMTSM  05154
OV2     05165

LC
```

The symbol OV2 indicates the second overlay of the function.

19. Use OFLCUP to write the second overlay to the mass store from location OV2 shown in step 18. Refer to Figure 4-7 to obtain the size and absolute segment number. A sample OFLCUP entry is shown below:

```
SF=TR COSM 5165,557,1
```

20. Repeat steps 17 through 19 for the third and any subsequent overlays. A memory map for the third overlay is presented below.

```
*LOW    05140
*START  05140
*HIGH   05324
*NAMES  23031
*COMN   26700
VEMTSM  05154
OV3     05165

LC
```

The OFLCUP entry used to write the third overlay to the mass store is shown below:

```
SF=TR COSM 5165,560,1
```

NOTE 1: There may be unresolved references to RTX-16 Executive locations (e.g., XLNK, XSPT, EROR, XHR, XMIN, and XSEC) after loading some overlays. These are resolved by loading XLOCS-716.

NOTE 2: The following statements apply with regard to the coordination of ONLMUP:

1. Coordination must be present.
2. Coordination must not be used for the ASR device.
3. Coordination may be used with those programs which call the ASR driver in order to prevent conflict.

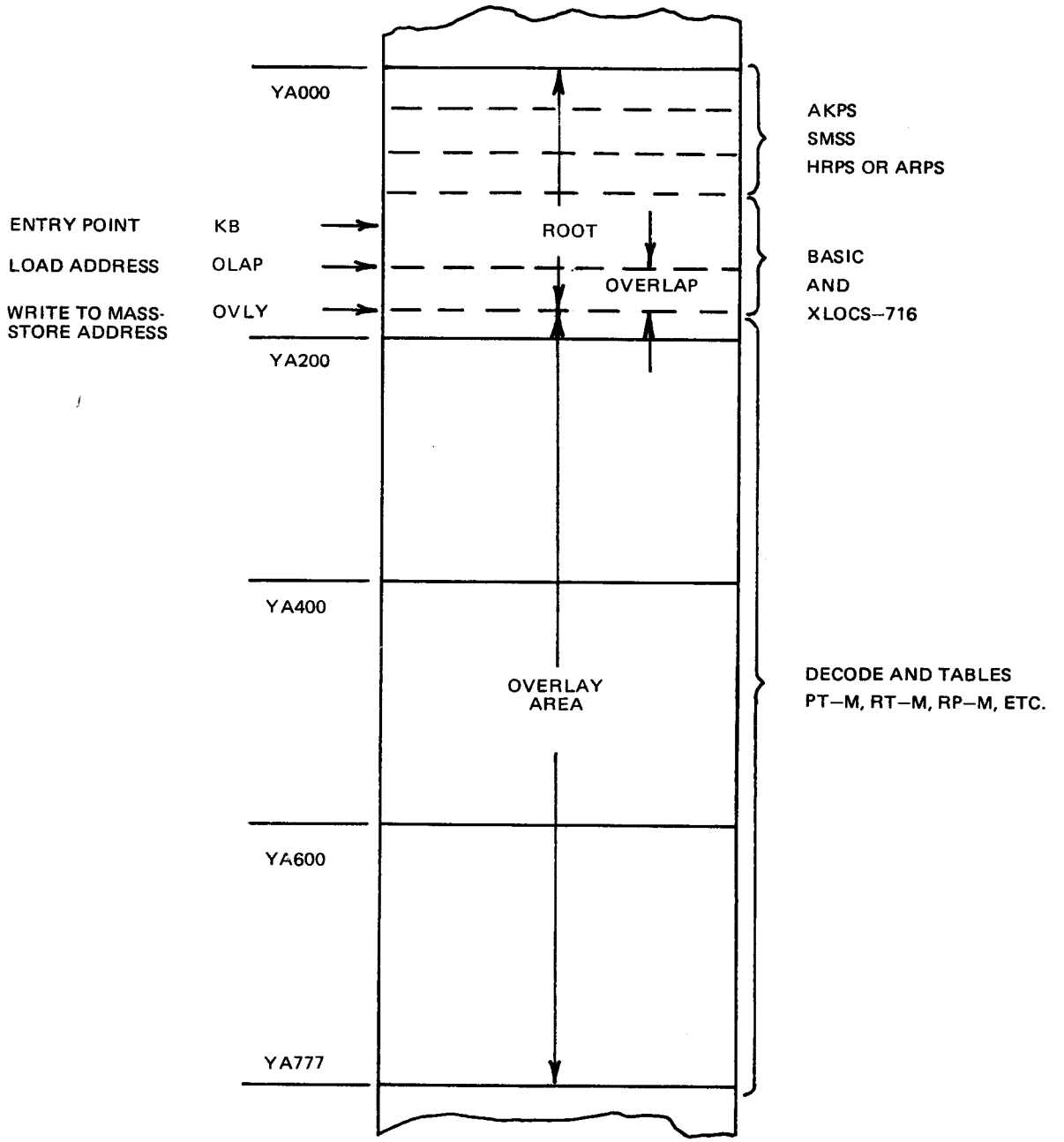


Figure 4-6. Typical Core Layout of ONLMUP

BOS Construction of ONLMUP

This operation may best be performed in the following way. It is assumed that a binary file, OFLCUP, occupying locations different from those to be used for ONLMUP, has already been created.

1. ? $LO *X, *A = YA000, OAKPS, *F, OSMSS, *F, OHRPS, *O
   ?? OBASIC, OXLOCS, *M = MAP1, *R = RSLT1, *Q
   ? $OU MAP1, SO (Note values of KB, OLAP, OVLY)
   ? $MA RSLTA = RSLT1, OFLCUP
   ? $EX RSLTA

2. Use RSLTA, created above, to write the ROOT to the mass store. A typical entry for this transfer is shown below, with (CSA) = 5000, (DSA) = 477, and (SC) = 1:

```
SF = TR COSM 5000,477,1
```

3. Return to BOS by typing ! (Return) in response to SF=

4. ? $LO *X, *A = <OLAP > , ODCODE, OTBLES, *H, *R = RSLT2
   (Memory Map will be printed)
   ? $MA RSLTB = RSLT2, OFLCUP
   ? $EX RSLTB

5. Use RSLTB to write the decode overlay to mass store. A typical entry for (CSA) = 5154 (OVLY or FD), (DSA) = 500, and (SC) = 2 is presented below:

```
SF = TR COSM 5154, 500, 2
```

   The segment number (500) used is the base segment number, for all overlays, which is entered in word 6 of table XSPT in the RTX-16 Configuration Module. A typical XSPT table is shown below:

```
*XSPT TABLE:
XSPT OCT  0          J-BASE
XPFP OCT  1023       POWER FAIL
     OCT  0          32K EXTENSION
     OCT  0          CORE LOW LIMIT
     OCT  77777      CORE HIGH LIMIT
     OCT  500        BASE SEGMENT
```

6. For single overlays, e.g. RP-M, the load should be as follows:
   ? $LO *X, *A = <OLAP > , ORPM, *H, *R = RSL
   (Memory Map will be printed)
   ? $MA RSLTN = RSL, OFLCUP
   ? $EX RSLTN

7. Use RSLTN to write the function overlay to the mass store, using (CSA) = OVLY (or RP for the above example).

   Refer to Figure 4-7 to obtain the size and absolute segment number for (SC) and (DSA), respectively. A sample of the OFLCUP entry using these values is shown below:

   ```
   SF = TR COSM 5154, 504, 1
   ```

8. For multiple overlays, e.g. VEMTSM, load as in step 6, but for each overlay, returning to BOS after each write to disk.

OFLMUP

Program Structure

Before building OFLMUP, the user should study the program structure presented in Figure 4-9 and the mass-store layout for ONLMUP presented in Figure 4-7. If the desired configuration contains a full complement of 21 functions, the mass-store layout for OFLMUP is identical to that of ONLMUP, except that functions PT, RT, RP, CC, and DC are not included. The component OFLMT was derived by use of the OFLMUP mass-store layout.

If the desired OFLMUP configuration contains less than the full complement of functions, the user must generate a mass-store layout by including only the desired functions and correcting the relative and absolute segment number. Figure 4-7 can be used as a guide. Then a new version of OFLMT must be written and assembled, using the listing for component TABLES, presented in Figure 4-8, as a guide.

The following tapes are required by this procedure (see Appendix E):

Tape Name	Document No.
IOSLST	70183001521
BASIC	70183006321
DECODE	70183008321
OFLMT or User-generated OFLMT tape	70183048321
MINX-716	41205679321

NAME/FUNCTION	SIZE IN SEGMENTS	RELATIVE SEGMENT NUMBER	ABSOLUTE SEGMENT NUMBER
ROOT	1	N/A	477
FD	2	0	500
PT	1	2	502
RT	1	3	503
RP	1	4	504
CC	1	5	505
DC	1	6	506
RC	2	7	507
PC	2	11	511
FC	1	13	513
SC	2	14	514
PL	1	16	516
RL	1	17	517
TRCOSM	1	20	520
TRSMCO	1	21	521
VESMCO	2	22	522
TRCOPP	1	24	524
TRPRCO	2	25	525
VEPRCO	2	27	527
TRSMPP	2	31	531
TRPRSM	2	33	533
VEPRSM1	2	35	535
VEPRSM2	2	37	537
TRCOMT	2	41	541
TRMTCO	2	43	543
VEMTCO1	2	45	545
VEMTCO2	2	47	547
TRSMMT	2	51	551
TRMTSM	2	53	553
VEMTSM1	2	55	555
VEMTSM2	1	57	557
VEMTSM3	1	60	560

BASE SEGMENT NUMBER

THE RELATIVE SEGMENT NUMBER PLUS THE SIZE IN SEGMENTS IS EQUAL TO THE NEXT RELATIVE SEGMENT NUMBER. THE ABSOLUTE SEGMENT NUMBER IS EQUAL TO THE BASE SEGMENT NUMBER PLUS THE RELATIVE SEGMENT NUMBER.

Figure 4-7. Mass-Store Layout for Mass-Store Utilities

```
      SUBR  FT                                                          0033
      REL                                                               0034
*                                                                       0035
*     FUNCTION TABLE                                                    0036
*           WORD 1 BITS 1-16  2 CHARACTER FUNCTION NAME                 0037
* NAME: TABLES          DOC 70183009000  REV F                          0001
*           WORD 2 BITS 1-4   OVERLAY SIZE IN SEGMENTS                  0038
*           WORD 2 BITS 5-16  RELATIVE SEGMENT NUMBER                   0039
*                                                                       0040
FT    BCI   1,PT              PRINT TIME                                0041
      OCT   10002                                                       0042
      BCI   1,RT              REPLACE TIME                              0043
      OCT   10003                                                       0044
      BCI   1,RP              REQUEST PROGRAM                           0045
      OCT   10004                                                       0046
      BCI   1,CC              CONNECT CLOCK                             0047
      OCT   10005                                                       0048
      BCI   1,DC              DISCONNECT CLOCK                          0049
      OCT   10006                                                       0050
      BCI   1,RC              REPLACE CORE                              0051
      OCT   20007                                                       0052
      BCI   1,PC              PRINT CORE                                0053
      OCT   20011                                                       0054
      BCI   1,FC              FILL CORE                                 0055
      OCT   10013                                                       0056
      BCI   1,SC              SEARCH CORE                               0057
      OCT   20014                                                       0058
      BCI   1,PL              PRINT LIMITS                              0059
      OCT   10016                                                       0060
      BCI   1,RL              REPLACE LIMITS                            0061
      OCT   10017                                                       0062
      BCI   1,TR              TRANSFER DATA                             0063
      DAC*  TT                POINTER TO TRANSFER TABLE                 0064
      BCI   1,VE              VERIFY DATA                               0065
      DAC*  VT                POINTER TO VERIFY TABLE                   0066
      OCT   0                 END OF FUNCTION TABLE                     0067
      EJCT                                                              0068
```

Figure 4-8 (Sheet 1 of 3). Format for TABLES (Function, Transfer, and Verify Tables)

```
*
*     TRANSFER TABLE                                                    0069
*                                                                       0070
*          WORD 1 BITS 1-16  2 CHARACTER INPUT DEVICE NAME              0071
*          WORD 2 BITS 1-16  2 CHARACTER OUTPUT DEVICE NAME             0072
*          WORD 3 BITS 1-4   OVERLAY SIZE IN SEGMENTS                   0073
*          WORD 3 BITS 5-16  RELATIVE SEGMENT NUMBER                    0074

* NAME: TABLES          DOC 70183009000  REV B                          0001

*                                                                       0075
TT   BCI   2,COSM          CORE TO MASS STORE                           0076
     OCT   10020                                                       0077
     BCI   2,SMCO          MASS STORE TO CORE                           0078
     OCT   10021                                                       0079
     BCI   2,COPP          CORE TO PAPER TAPE                           0080
     OCT   10024                                                       0081
     BCI   2,PRCO          PAPER TAPE TO CORE                           0082
     OCT   20025                                                       0083
     BCI   2,SMPP          MASS STORE TO PAPER TAPE                     0084
     OCT   20031                                                       0085
     BCI   2,PRSM          PAPER TAPE TO MASS STORE                     0086
     OCT   20033                                                       0087
     BCI   2,COMT          CORE TO MAG TAPE                             0088
     OCT   20041                                                       0089
     BCI   2,MTCO          MAG TAPE TO CORE                             0090
     OCT   20043                                                       0091
     BCI   2,SMMT          MASS STORE TO MAG TAPE                       0092
     OCT   20051                                                       0093
     BCI   2,MTSM          MAG TAPE TO MASS STORE                       0094
     OCT   20053                                                       0095
     OCT   0               END OF TRANSFER TABLE                        0096
     EJCT                                                              0097
```

Figure 4-8 (Sheet 2 of 3). Format for TABLES (Function, Transfer, and Verify Tables)

```
*                                                                       0098
*     VERIFY TABLE                                                      0099
*           WORD 1 BITS 1-16  2 CHARACTER INPUT DEVICE NAME             0100
*           WORD 2 BITS 1-16  2 CHARACTER OUTPUT DEVICE NAME            0101
*           WORD 3 BITS 1-4   OVERLAY SIZE IN SEGMENTS                  0102
*           WORD 3 BITS 5-16  RELATIVE SEGMENT NUMBER                   0103
*                                                                       0104
VT    BCI   2,SMCO            MASS STORE AGAINST CORE                   0105
      OCT   20022                                                       0106
      BCI   2,PRCO            PAPER TAPE AGAINST CORE                   0107
      OCT   20027                                                       0108
      BCI   2,PRSM            PAPER TAPE AGAINST MASS STORE             0109
      OCT   20035                                                       0110
      BCI   2,MTCO            MAG TAPE AGAINST CORE                     0111
* NAME: TABLES        DOC 70183009000  REV B                            0001

      OCT   20045                                                       0112
      BCI   2,MTSM            MAG TAPE AGAINST MASS STORE               0113
      OCT   20055                                                       0114
      OCT   0                 END OF VERIFY TABLE                       0115
      END                                                               0116
```


Figure 4-8 (Sheet 3 of 3). Format for TABLES (Function, Transfer, and Verify Tables)

Tape Name	Document No.
ASRD-S or ASRD-L	70181475321 70181476321
MHSD or DDAM-C	70181744321 41204915321
HSRD-H	41286258321
HSPD-H	41286259321
MTUD	70181622321

Construction of OFLMUP

It is assumed that the user has an OFLCUP[1] tape available and has loaded LDR-APM in core, with a normal entrance at XX000. The XX represents the highest sector occupied by the loader. To construct OFLMUP, the user must perform the following steps in the sequence shown:

1. Load OFLCUP with core/mass-store functions.
2. Force-load the following routines from IOSLST, beginning at sector YB000:
   a. Load routine AKPS by setting (P) = XX006 and (A) = YB000 and pressing START.
   b. Load routine SMSS by setting (P) = XX004 and (A) = 0 and pressing START.
   c. If the high speed reader/punch is utilized in the system, load routine HRPS by setting (P) = XX004 and (A) = 0 and pressing START. Remove IOSLST from the reader.

      If the ASR reader/punch is utilized, move IOSLST forward by hand, bypassing routine HRPS. Then load routine ARPS by setting (P) = XX004 and (A) = 0 and pressing START.
3. Load BASIC by pressing START. This completes the loading of the Root.
4. Obtain a memory map. A typical memory map for YB000 = 5000 is presented below:

[1]Wherever OFLCUP is referred to in this procedure, OFLUT-2H or OFLUT-3H may be substituted.

```
*BOTM   05000
*LOW    05000
*START  05000
*HIGH   05154
*TOP    05154
*NAMES  26001
*COMN   37000
AKS     05000
APS     05005
SMS     05026
PRS     05052
PPS     05057
KB      05077
RPRG    05117
PNME    05122
PAR3    05133
PAR4    05134
PAR5    05135
OLAP    05140
XLNK    05143**
XSPT    05144**
EROR    05145**
OVLY    05154

MR
```

Note the addresses contained on the memory map for the following locations:

a. OLAP - load address for all overlays.

b. OVLY - write to mass-store address for single overlay functions.

5. Load MINX-716 by setting (P) = XX003 and (A) = YA000 and pressing START. Obtain a memory map, which should appear as follows for YA000 = 4000:

```
*BOTM   04000
*LOW    04000
*START  05000
*HIGH   05154
*TOP    05154
*NAMES  25677
*COMN   37000
MINX    04000
XLNK    04411
EROR    04412
EPM3    04413
XSPT    04415
CHL1    04466
AS      04500**
CHL2    04504
PR      04515**
PP      04522**
MT      04527**
```

```
CHL3    04[illegible]2
CHL4    04[illegible]3
SM      04[illegible]4**
AKS     05[illegible]
APS     05[illegible]
SMS     05[illegible]
PRS     05[illegible]
PPS     05[illegible]
KB      05077
PPRC    05117
PMME    05122
PAR3    05[illegible]3
PAR4    05[illegible]4
PAR5    05[illegible]5
PLAF    05140
OVLY    05154

MR
```

6. Load MTUD by setting (P) = XX003 and (A) = YC000, and pressing START.

7. Load the mass-store driver (MHSD, DDAM-C, etc.) associated with the system by setting (P) = XX003 and (A) = YD000, and pressing START. If ARPS was loaded in step 2, proceed to step 11.

8. Load HSRD-H by setting (P) = XX003 and (A) = YE000, and pressing START.

9. Load HSPD-H by pressing START.

10. Load ASRD-S by setting (P) = XX003 and (A) = YF000, and pressing START. Proceed to step 12.

11. Load ASRD-L by setting (P) = XX003 and (A) = YE000, and pressing START.

12. This completes the core-resident portion of OFLMIP. Obtain a memory map. A typical memory map is presented below for YC000 = 6000, YD000 = 7000, YE000 = 10000, and YF000 = 11000.

```
*BOTM   04000
*LOW    04000
*START  05000
*HIGH   11362
*TOP    11361
*NAMES  25625
*COMN   37000
MINX    04000
XLNK    04411
```

NOTE: The unresolved references MT11, MT12, ASF1, and ASF2 are linked to Fortran "caps" and may be ignored. Appendix F deals with XDCT.

```
EROR    04412          MT      06006
EPM3    04413          XDCT    06502**
XSPT    04415          MTI1    06555**
CHL1    04466          MTI2    06556**
CHL2    04504          SM      07006
CHL3    04532          PR      10006
CHL4    04533          C$OP    10356
AKS     05000          C$IP    10356
APS     05005          PP      10366
SMS     05026          AS      11006
PRS     05052          ASF1    11343**
PPS     05057          ASF2    11344**
KB      05077
RPRG    05117
PNME    05122          MR
PAR3    05133
PAR4    05134
PAR5    05135
OLAP    05140
OVLY    05154
```

13. Initialize as in Appendix F.

14. Use PAL-AP to punch a self-loading tape of the core-resident portion of OFLMUP.

15. The mass-store-resident portion of OFLMUP is constructed following steps 7 through 20 of the ONLMUP procedure, except for the following differences:

    a. Either OFLMT or a user-generated OFLMT is used in place of TABLES.

    b. The on-line routines from FCNLST1 are not loaded.

    Note that ONLMUP and OFLMUP may use the same set of overlays, provided they are built in the same sector.

BOS Construction of OFLMUP

It is assumed that a binary file, OFLCUP, occupying locations different from those to be used for OFLMUP, has already been created.

1.
```
? $LO *X, *A = YB000, OAKPS, *F, OSMSS, *F, OHRPS, *O
?? OBASIC, *M = MAP1, *P = YA000, OMINX, *M = MAP2
?? *P = YC000, OMTUD, *P = YD000, ODDAMC
?? *P = YE000, OHSRDH, OHSPDH, *P = YF000, OASRDS
?? *M = MAP3, *R = OFLMUP, YA000, *Q
```

2. Output all memory maps produced.

3. Refer to Appendix F, and use Debug and Trace ($DT) to make any necessary changes.

4. The above creates the core-resident portion of OFLMUP. The mass-store-resident portion may be constructed using steps 4-8 of the BOS ONLMUP procedure, with the following differences:

   - Either OFLMT or a user-generated version is used in place of TABLES.
   - The on-line overlays from FCNLST1 are not loaded.

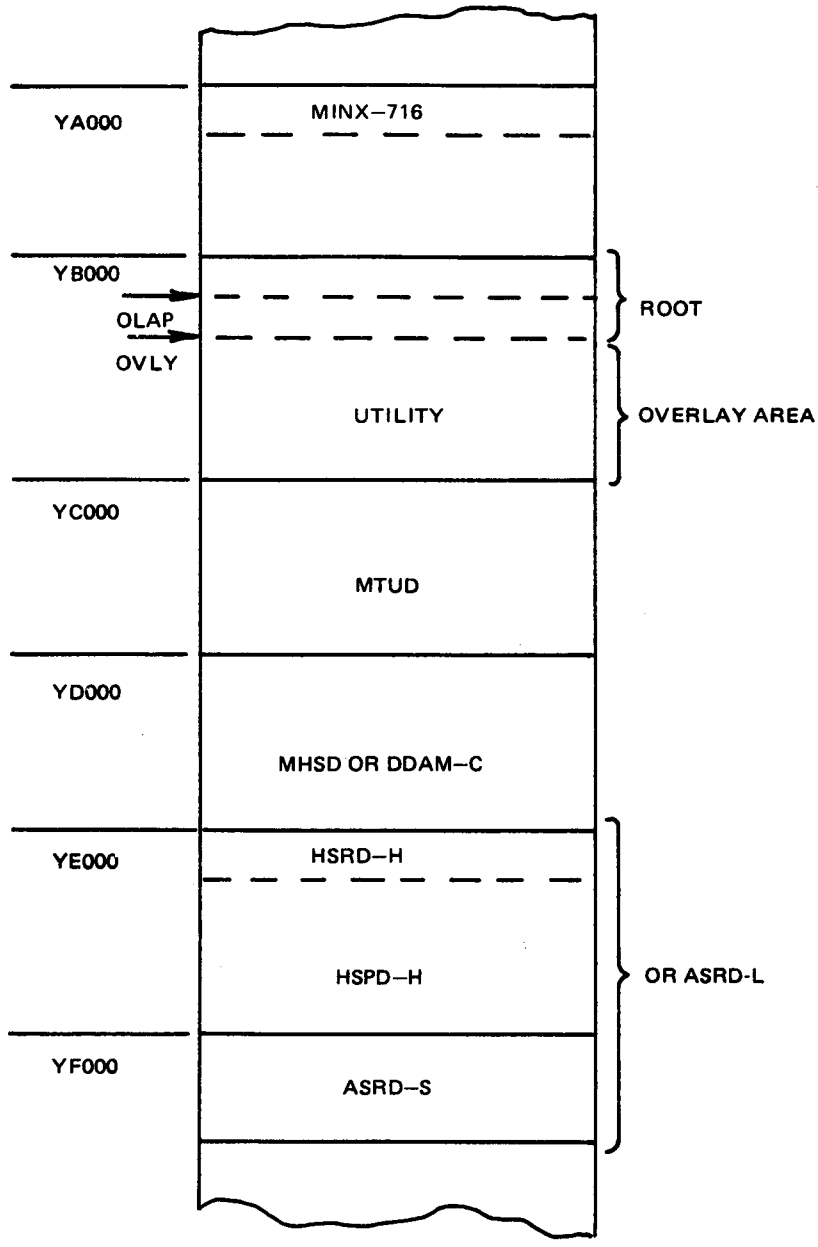


Figure 4-9. Typical Core Layout for OFLMUP

APPENDIX A

KEYBOARD FUNCTIONS

The formats for the keyboard commands of the OP-16 utility programs are presented below. Further explanations are given in Section III.

SF=VE MTCO <UNIT> , <FILE> , <CSA> , <CEA> (CR)

SF=TR COSM <CSA> , <DSA> , <SC> (CR)

SF=TR SMCO <CSA> , <DSA> , <SC> (CR)

SF=VE SMCO <CSA> , <DSA> , <SC> (CR)

SF=TR SMMT <UNIT> , <FILE> , <DSA> , <DEA> (CR)

SF=TR MTSM <UNIT> , <FILE> (CR)

SF=TR MTSM <UNIT> , <FILE> , <DSA> , <DEA> (CR)

SF=VE MTSM <UNIT> , <FILE> (CR)

SF=VE MTSM <UNIT> , <FILE> , <DSA> , <DEA> (CR)

SF=PT (CR)

SF=RT <HOURS> , <MINUTES> (CR)

SF=RP <NAME> (CR)

SF=RP <NAME> , <PARAMETER> (CR)

SF=CC <NAME> , <INITIAL> , <INTERVAL> , <BASE> (CR)

SF=DC <NAME> , <BASE> (CR)

SF=RC <CSA> (CR)

SF=PC <CSA> , <CEA> (CR)

SF=PC <CSA> (CR)

SF=FC <CSA> , <CEA> , <VALUE> (CR)

SF=SC <CSA> , <CEA> , <VALUE> , <MASK> (CR)

SF=PL (CR)

SF=RL <LOW LIMIT> , <HIGH LIMIT> (CR)

SF=TR COPP <CSA> , <CEA> (CR)

SF=TR PRCO (CR)

SF=TR PRCO <CSA> , <CEA> (CR)

SF=VE PRCO (CR)

SF=VE PRCO <CSA> , <CEA> (CR)

SF=TR SMPP <DSA> , <DEA> (CR)

SF=TR PRSM (CR)

SF=TR PRSM <DSA> , <DEA> (CR)

SF=VE PRSM (CR)

SF=VE PRSM <DSA> , <DEA> (CR)

SF=TR COMT <UNIT> , <FILE> , <CSA> , <CEA> (CR)

SF=TR MTCO <UNIT> , <FILE> (CR)

SF=TR MTCO <UNIT> , <FILE> , <CSA> , <CEA> (CR)

SF=VE MTCO <UNIT> , <FILE> (CR)

APPENDIX B

KEYBOARD MESSAGES AND DEVICE NAMES

The keyboard messages of the OP-16 utility programs are presented in Table B-1.

Table B-1. Keyboard Messages

Message	Meaning
SF=	System is ready to receive function and other parameters from the user.
FE	Format or function error: parameter entered incorrectly.
LE	Limit error: attempt to modify core outside core protection limits.
HE	Header error: incorrect header record for the particular type of paper tape or magnetic tape transfer.
CE	Checksum error detected during a paper tape function.
ME	Magnetic tape error: magnetic tape driver could not complete the requested operation because of hardware malfunction.
ET	End-of-tape: end-of-tape mark encountered on magnetic tape.
EF	End-of-file: end-of-file mark encountered on magnetic tape, indicating the file contains less than the specified data.
E100KB	The utility program was unable to complete the requested function and has been terminated.

The present device codes used with the OP-16 utility programs are presented in Table B-2.

Table B-2. Device Codes

Mnemonic	Device Name
CO	Core
PP	Paper Tape Punch
PR	Paper Tape Reader
SM	System Mass Store
MT	Magnetic Tape

APPENDIX C
DATA FORMATS

Data formats for paper tape, magnetic tape, and mass store are presented below:

Paper Tape

Standard binary 4/6/6 format

Three types of record - Header, Data, and EOM

Header record - 4 words

- Word 1 - Source (0 = core, 1 = mass store)
- Word 2 - Start address
- Word 3 - End address
- Word 4 - Checksum of words 1-3

Data record - 33 words (fixed length)

- 32 words of data
- 1 word checksum of 32 data words

EOM record - Standard end-of-tape record

Magnetic Tape

Three-character binary 6/6/4 format

Three types of record - Header, Data, and EOF

Header record - 3 words

- Word 1 - Source (0 = core, 1 = mass store)
- Word 2 - Start address
- Word 3 - End address

Data record - 128 words

EOF record - Standard end-of-file record

Mass Store

128-word records or segments

APPENDIX D

OCTAL/DECIMAL CONVERSION

OCTAL TO DECIMAL

This is best performed by repeated multiplication by 8.

Example: To convert 6154_8 to decimal

6 1 5 4		
8		
4 9	i.e.	6 x 8 + 1 (next digit)
8		
3 9 7	i.e.	49 x 8 + 5 (next digit)
8		
3 1 8 0	i.e.	397 x 8 + 4

$6154_8 = 3180_{10}$

DECIMAL TO OCTAL

This is best performed by repeated division by 8, noting any remainders.

Example: To convert 3975_{10} to octal

8)3 9 7 5		
8) 4 9 6	rem.	7
8) 6 2	rem.	0
7	rem.	6

Take remainders in reverse order, i.e. 7607.

$3975_{10} = 7607_8$

APPENDIX E

COMPONENTS AND ROUTINES REQUIRED BY EACH UTILITY

Component	Routine	ONLMUP	OFLMUP
MINX-716 and Device Drivers	-		X
IOSLST	AKPS	X	X
	SMSS	X	X
	HRPS	X	X
	ARPS	X	X
BASIC	-	X	X
XLOCS-716	-	X	
DECODE	-	X	X
TABLES	-	X	
OFLMT	-		X
FCNLST1	PT-M	X	
	RT-M	X	
	RP-M	X	
	CC-M	X	
	DC-M	X	
FCNLST2	RC-M	X	X
	PC-M	X	X
	FC-M	X	X
	SC-M	X	X
	PL-M	X	X
	RL-M	X	X
	TRCOSM-M	X	X
	TRSMCO-M	X	X
	VESMCO-M	X	X
	TRCOPP-M	X	X
	TRPRCO-M	X	X
	VEPRCO-M	X	X
	TRSMPP-M	X	X
	TRPRSM-M	X	X
	VEPRSM1-M	X	X
	VEPRSM2-M	X	X
	TRCOMT-M	X	X
	TRMTCO-M	X	X

Component	Routine	OMLMUP	OFLMUP
FCNLST2 (cont)	VEMTCO1-M	X	X
	VEMTCO2-M	X	X
	TRSMMT-M	X	X
	TRMTSM-M	X	X
	VEMTSM1-M	X	X
	VEMTSM2-M	X	X
	VEMTSM3-M	X	X
ROOT-C	-	X	X
FT-C	-	X	
OFLCT	-		X
FCNLST-C	PT-C	X	
	RP-C	X	
	CC-C	X	
	RC-C	X	X
	PC-C	X	X
	PL-C	X	X
SUBR1-C	GET-C	X	X
	PUT-C	X	X
	CMN-C	X	X
	CL-C	X	X
SUBR2-C	OE-C	X	X
	OD-C	X	X
	AE-C	X	
	DE-C	X	
TR-C	-	X	X
B2-C	-	X	X
OP-C	-	X	X

Document numbers for each component are listed in the appropriate paragraphs of Section IV.

APPENDIX F

OFF-LINE UTILITY INITIALIZATION

The off-line utilities may have to be initialized with certain user-supplied information before a self-loading tape is punched.

Referring to the utility memory map, note the addresses of locations CHL1, CHL2, CHL3 and CHL4. These may require change as follows:

Base segment for Overlays not O	CHL1 set to base segment
Using moving-head disk	CHL2 set to 1, CHL4 set to '125252
Magnetic Tape present	Loc XDCT set to CHL3-6
Magnetic Tape not on DMC channel 2	CHL3 set to DMC channel

```
*BOTM   01000
*LOW    01000
*START  01000
*HIGH   01552
*TOP    01552
*NAMES  26007
*COMN   37000
MINX    01000
XLNK    01411
EROR    01412
EPM3    01413
XSPT    01415
KB      01461**
CHL1    01466
AS      01500**
CHL2    01504
PR      01515**
PP      01522**
MT      01527**
CHL3    01532
CHL4    01533
SM      01534**

MR
```

Figure F-1. Single of Memory Map of MINX-716

```
0414                                *
0415                                *     CONFIGURATION TABLES FOR CP-16 UTILITIES
0416                                *     THOSE LOCATIONS MARKED *** MAY HAVE TO BE CHANGED TO
0417                                *     SUIT PARTICULAR CONFIGURATIONS.
0418                                *     NOTE THAT ONLY 7-TRACK MAG. TAPE IS CATERED FOR.
0419                                *
0420                                *     SPECIAL PARAMETER TABLE
0421 00461   0 000000   SPT   XAC   KP                  UTILITY PROGRAM
0422 00462   000000           BSZ   2                   NOT USED
0423 00464   000001           CCT   1                   LOW CORE PROTECTION LIMIT
0424 00465   077777           CCT   77777               HIGH CORE PROTECTION LIMIT
0425 00466   000000     CHL1  CCT   0                   *** BASE SEGMENT FOR OVERLAYS
0426                                *
0427                                *     INTERRUPT DEFINITION TABLE
0428 00467   177772     IDT   DEC   -6                  COUNT OF NUMBER OF DEVICES
0429 00470   0 000477         DAC   L01                 POINTER TO DEVICE LIST
0430 00471   0 000504         DAC   L02
0431 00472   0 000514         DAC   L03
0432 00473   0 000521         DAC   L04
0433 00474   0 000526         DAC   L05
0434 00475   0 000533         DAC   L06
0435                                *
0436                                *     ASR TELETYPE
0437 00476   34 0404           SKS   '404                SKS INSTRUCTION
0438 00477   000004     L01   HEX   4                   DEVICE CODE
0439 00500   0 000000         XAC   AS                  DRIVER START ADDRESS
0440 00501   74 0020          SMK   '20                 SMK INSTRUCTION
0441 00502   000040           CCT   40                  MASK BIT
0442                                *
0443                                *     MOVING HEAD DISK
0444 00503   34 0125           SKS   '125
0445 00504   125252     L02   CCT   125252              *** THIS IS SET ONLY TO AVOID 451X CONFLICT
0446 00505   0 000000         XAC   SM
0447 00506   74 0020          SMK   '20
0448 00507   010000           CCT   10000
0449 00510   000001           DEC   1                   CHANNEL NUMBER
0450 00511   000000           BSZ   1                   NOT USED
```

Figure F-2. MINX-716 Tables

```
0451 00512   000024          DEC   20                    20-SURFACE DISK
0452                       *
0453                       *     HIGH SPEED PAPER TAPE READER
0454 00513   34 0401        SKS   '401
0455 00514   000002    L03  HEX   2
0456 00515   0 000000       XAC   PR
0457 00516   74 0020        SMK   '20
0458 00517   000200         CCT   200
0459                       *
0460                       *     HIGH SPEED PAPER TAPE PUNCH
0461 00520   34 0402        SKS   '402
0462 00521   000003    L04  HEX   3
0463 00522   0 000000       XAC   PP
0464 00523   74 0020        SMK   '20
0465 00524   000100         CCT   100
0466                       *
0467                       *     MAGNETIC TAPE
0468 00525   34 0410        SKS   '410
0469 00526   000007    L05  HEX   7
0470 00527   0 000000       XAC   MT
0471 00530   74 0020        SMK   '20
0472 00531   100000         SKP                         BIT 1 MASK IT
0473 00532   000002    CHL3 CCT   2                     *** CHANNEL NUMBER
0474                       *
0475                       *     451X FIXED HEAD DISK (716 DEVICE)
0476 00533   000001    L06  HEX   1                     ***
0477 00534   0 000000       XAC   SM
0478 00535   14 1522        CCP   '1522
0479 00536   14 1622        CCP   '1622
0480                       *
0481                       *     PROGRAM NAME TABLE
0482                       *
0483 00537   177773    PNT  DEC   -5                    COUNT OF ENTRIES
0484 00540   140723         BCI   1,AS
0485 00541   000004         HEX   4
0486 00542   151715         BCI   1,SM
```

Figure F-2. MINX-716 Tables

```
0487 00543   000001          HEX   1
0488 00544   150322          ECI   1,PT
0489 00545   000002          HEX   2
0490 00546   150320          ECI   1,PP
0491 00547   000003          HEX   3
0492 00550   146724          ECI   1,MT
0493 00551   000007          HEX   7
```

Figure F-2. MINX-716 Tables

Honeywell

HONEYWELL INFORMATION SYSTEMS

Printed in U.K.

YZ50 Rev. 2