Honeywell

UTILITY PROGRAMS

SERIES 16

OP-16

SOFTWARE

Honeywell

UTILITY PROGRAMS

SERIES 16

OP-16

SUBJECT:

Functions, Use, and Building Procedures of OP-16 Utility Programs within Hardware Environment Supported by OP-16 Operating System.

INCLUDES UPDATE PAGES ISSUED AS ADDENDUM A, DECEMBER 1971.

DATE:

May 1971

ORDER NUMBER:

AB63, Rev. 0 (Previously M-445)

DOCUMENT NUMBER:

700130072519

PREFACE

This reference manual for programmers contains the information required to operate the OP-16 Utility Programs through the Honeywell-supplied OP-16 Input/Output library, in conjunction with the RTX-16 Real-Time Executive. The reader is assumed to have a basic familiarity with Series 16 assembly language programming and to have read the 316/516 Programmers Reference Manual (Doc. No. 130072156) and the OP-16 Users Guide (Doc. No. 70130072404).

An introductory section characterizes the two types of utility programs that can be built (off-line and on-line), discusses the preconfigured basic utilities available to the user, and shows the applicability of MINEX, 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.

For further information on the operating procedures of LDR-APM and PAL-AP, with which the reader should be familiar, consult the Series 16 Operators Guide (Doc. No. 70130072165). All utility program components are listed in Binder Table of Contents for OP-16 Operating System (Doc. No. 70181898311).

File No.: 1A13

#AB63

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

OVERVIEW

This document describes the functions, use, and building procedures of the OP-16 Utility Programs. These programs provide a conversational interface between an operator and the OP-16 operating system via an ASR-33 or ASR-35 teletype, within the hardware environment supported by OP-16.

Using the program components provided, two types of utility programs can be built, each to suit a given hardware environment 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

OFF-LINE AND ON-LINE UTILITIES

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/verify between a variety of storage devices and external media.

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

- program debugging tools;
- data transfer/verify between a variety of storage devices and external media;
- optional operator control over the initiation and termination of periodic programs and initiations 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.

Both the off- and on-line versions utilize the standard OP-16 drivers.

Each of the two types of OP-16 utilities 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.

The functions which may be incorporated in any of the versions are summarized in Table 1-1 and are discussed in detail in Section III.

PRECONFIGURED BASIC UTILITIES

The modular construction of the utility programs allows the user to include or exclude features to suit particular requirements and hardware environments. 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-1 is a core-resident off-line utility which supports the ASR Keyboard/Printer and the High Speed Paper Tape Reader/Punch. It includes functions shown in Table 1-1.

OFLUT-2 is a core-resident off-line utility which supports the ASR Keyboard/Printer, the High Speed Paper Tape Reader/Punch, and the 20-surface (Type 4650) or the 2-surface (Type 4720) Moving Head Disk. It includes functions shown in Table 1-1. It may be conveniently used to build other disk-resident utilities or other programs.

OFLUT-3 is similar to OFLUT-2 except that it supports the Drum Storage Unit (Type 931X).

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.

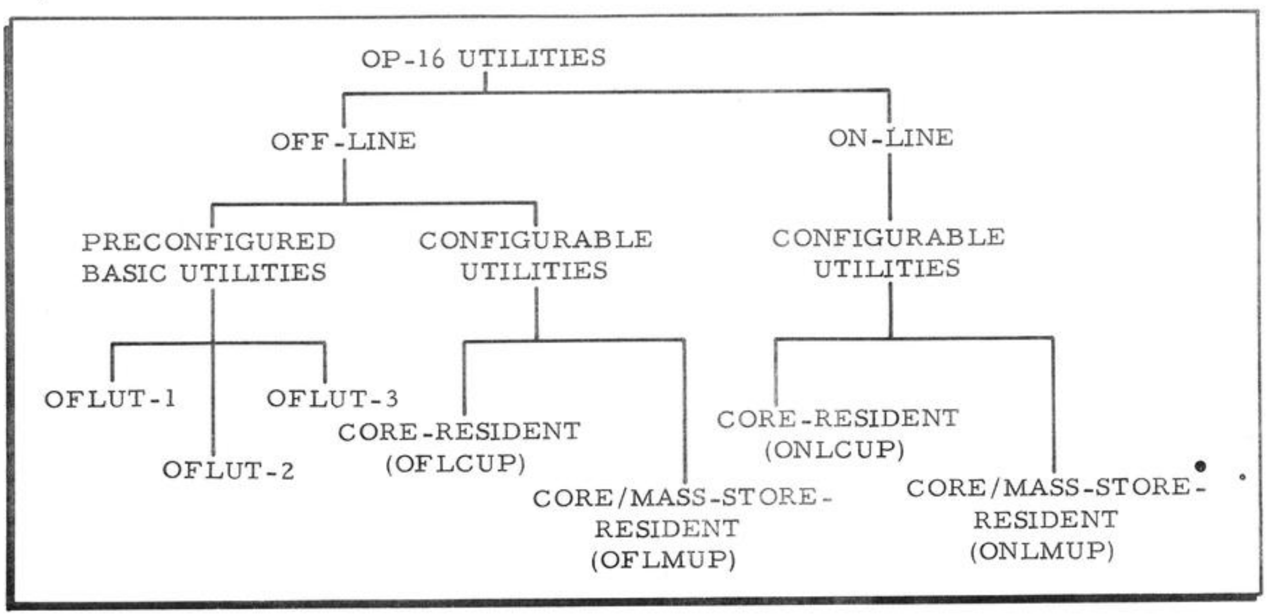


Figure 1-1. Versions of OP-16 Utility Programs

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.

MINEX

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

HARDWARE AND SOFTWARE REQUIREMENTS

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

CONVENTIONS USED IN MANUAL

The term *mass store* is used to describe the system moving head disk unit or drum storage unit. 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. The prefix I is used to indicate that decimal numbers are required, and the prefix 'I to indicate that octal numbers are required.

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

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

Table 1-2. OP-16 Utility Program Hardware, Software Requirements, and Characteristics

	ON-LINE CORE-RES. (ONLCUP)			ON-LINE CORE/MASS-STORE-RES. (ONLMUP)			OFF-LINE CORE-RES. (OFLCUP)			OFF-LINE CORE/MASS-STORE-RES. (OFLMUP)			PRECONFIGURED OFF-LINE CORE-RESIDENT					
													REQUIRED HARDWARE			REQUIRED SOFTWARE		
	REQ. HDW.	SUPPT. HDW.	REQ. SFT.	REQ. HDW.	SUPPT. HDW.	REQ. SFT.	REQ. HDW.	SUPPT. HDW.	REQ. SFT.	REQ. HDW.	SUPPT. HDW.	REQ. SFT.	OFLUT-1	OFLUT-2	OFLUT-3	OFLUT-1	OFLUT-2	OFLUT-3
Minimum hardware																		
Configuration for OP-16	X			X			X			X			X	X	X			
M50 Paper Tape Reader		X			X			X			X		X	X	X			
M52 Paper Tape Punch		X			X			X			X		X	X	X			
M931X Drum Storage Unit				X, or				X, or		X, or					X			
M4650 Moving Head Disk				X, or				X, or		X, or				X, or				
M4720 Moving Head Disk				X				X		X				X				
M4100 Magnetic Tape					X, or						X, or							
M4021 Magnetic Tape					X						X							
RTX-16 Executive			X			X												
MINEX									X			X				X	X	X
Corresponding device drivers			X			X			X			X				X	X	X
Core occupancy (sectors)	1 to 2 depending on configuration			1, independent of configuration			Up to 2, plus drivers as needed			1-1/4, plus drivers as needed						4	5	5
Maximum number of functions	15			26			10			21						5	7	7
Run under	RTX-16			RTX-16			MINEX			MINEX						MINEX	MINEX	MINEX
Structure	Contiguous			Segmented, overlaid in core			Contiguous			Segmented, overlaid in core							Contiguous	
Built by using	LDR-APM			LDR-APM, OFLUT-2 or -3, or OFLCUP			LDR-APM			LDR-APM, OFLUT-2 or -3, or OFLCUP							LDR-APM[1]	

[1]Object text is arranged on a single paper tape in the order of loading.

#AB63

SECTION II

USE

INITIATION

Once the utility programs are built in accordance with the Building Procedures (Section IV) they may be initiated by performing the steps outlined below.

Off-line Utilities

1. Obtain the core address of location EXEC from the loading map (refer to Section IV).
2. Set the MA/SI/RUN switch to SI.
3. Master clear
4. If the program was built in the normal mode, set (P) = EXEC. If the program was built in the extended mode, set (P) = EXEC + 1.
5. Set the MA/SI/RUN switch to RUN.
6. 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.

On-line Utilities

After the RTX-16 executive is initialized as described in the OP-16 Users Guide, typing a dollar sign ($) on the ASR 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.

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 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.

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

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

TR Transfer Data

VE Verify Data

Device Names

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

Table 2-2. Device Names

Name	Description
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 MA/SI/RUN switch to SI. The program may be placed again in the "receive command" mode by repeating the steps described previously on page 2-1.

The on-line utility programs may be terminated by typing an exclamation mark (!) through 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 ($) through 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 the function, 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

<table>
<tr><th rowspan="2">Keyboard Character</th><th colspan="3">Function</th></tr>
<tr><th>On-line Utilities</th><th colspan="2">Off-line Utilities</th></tr>
<tr><td>$</td><td>Requests the utility program. When started by the EXEC, the characters SF= are typed to signify the "receive command" mode.</td><td colspan="2" rowspan="2">X</td></tr>
<tr><td>!</td><td>Causes termination of the utility program when typed at any time the utility program is expecting information.</td></tr>
<tr><td>/</td><td colspan="2">Deletes the current line and causes a new SF= to be printed.</td><td rowspan="5">When typed as part of a command line beginning with SF=.</td></tr>
<tr><td>←</td><td colspan="2">Deletes the current character.</td></tr>
<tr><td>@</td><td colspan="2">Deletes the entry in the current field.</td></tr>
<tr><td>,</td><td colspan="2">Delimits the parameter field.</td></tr>
<tr><td>(CR)</td><td colspan="2">Terminates a command line.</td></tr>
</table>

ERROR DIAGNOSTICS

Errors detected and reported by the utility programs are summarized in Table 2-4. When any of these errors is detected, the current function is terminated, an 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 Users Guide and in the corresponding OP-16 Driver Manual. After an I/O error, both the initiated function and the utility program are terminated.

Off-line Utilities

The utility program halts with the error code in the A register and the address of location CALL+1 in the B register. To continue using the utility program, it should be reinitiated 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	Checksum 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 the file contains less than the specified data.

SECTION III
FUNCTIONS

This section describes the utility functions currently available. Other functions may be provided in the future.

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 Is
CSA	Core Start Address	Octal
CEA	Core End Address	Octal
DSA	Mass Store Start Address	Octal Segment Number
DEA	Mass Store End Address	Octal Segment Number
SC	Segment Count	Octal
Unit	Magnetic Tape Unit Number	0-3
File	Magnetic Tape File Number	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 50-ms units. Interval between executions is in 50-ms 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 for the system clock. The name and base parameters are defined in Table 3-2.

DEBUGGING FUNCTIONS

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, eight 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 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 is the core address;

column 2 is the core contents;

column 3 is 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 according to 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 is the core address;

column 2 is the core contents;

column 3 is 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 according to 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    5211
    477     57      0    5255
    477     74  11147  177777
    477     75   5102       0
    477     76  11134       0
    477     77  21131  105144
```

where column 1 is the mass-store segment number;

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

column 3 is the mass-store contents;

column 4 is 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 according to 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 is the core address;

column 2 is the core contents;

column 3 is 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 was 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 record is encountered.

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

```
SF=VE MTSM 0,2,147,147
    147        0 100040 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 is the mass-store segment number;

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

column 3 is the mass-store contents;

column 4 is the corresponding magnetic tape contents.

SAMPLE PRINTOUT

A sample interchange between the user and system using some of the OP-16 utility functions is presented in Figure 3-1 (for explanation refer to the text).

```
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

LE
SF=RC 7000
  7000 177777 ,
  7001       0 /
SF=RC 12776
 12776 177762 ,
 12777 177772 ,

LE
SF=TR COPP 100,147
SF=TR PRCO
LE
SF=VE PRCO
SF=PT
0312
```

Figure 3-1. Sample Interchange Using OP-16 Utility

SECTION IV
BUILDING PROCEDURES

INTRODUCTION

This section contains detailed procedures for building OP-16 utility programs. It is assumed that the user is sufficiently familiar with the operating procedures of LDR-APM and PAL-AP, which are described in the Series 16 Equipment Operator's Guide, Order No. BX48.

A summary of the various types of utility programs is included in Section I of this manual (Figure 1-1), the functions available for each are listed in Table 1-1, and their relevant characteristics are shows in Table 1-2.

The off-line versions of the OP-16 utility program, OFLCUP and OFLMUP, are operated under the control of MINEX. These versions are relocatable and should be built without using any sector 0 locations for cross-sector indirect links. MINEX has links to several external programs, which are identified below:

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.

BUILDING PROCEDURE FOR OFLUT-1

The OFLUT-1 tape (Doc. No. 70183075521) contains 19 objects. The following simple procedure will build an OFLUT-1 utility in four sectors (YA, YB, YC, and YD) specified by the user. LDR-APM is assumed to be in core, with normal entrance at XX000.

1. Insert OFLUT-1 in the paper tape reader.
2. Set (P)= XX000 (or XX006, if the program is to support systems with more than 16K of core) and (A)=YA000. Load the first two objects into sector YA.
3. Set (P)=XX003 and (A)=YB000. Load the next two objects into sector YB.
4. Set (P)=XX003, (A)=YC000, and (B)=YC770. Load the next six objects into sector YC.
5. Set (P)=XX003, (A)=YD000, and (B)=YD260. Load the next nine objects into sector YD.
6. Obtain a memory map, (P)=XX002.
7. Use PAL-AP to punch a self-loading tape of OFLUT-1.

BUILDING PROCEDURE FOR OFLUT-2/OFLUT-3

OFLUT-2 and OFLUT-3 tapes (Doc. Nos. 70183076521 and 70183077521) both contain 21 objects. The following simple procedure will build an OFLUT-2 or OFLUT-3 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-2 or OFLUT-3 in the paper tape reader.
2. Set (P)=XX000 (or XX006, if the program is to support systems with more than 16K of core) and (A)=YA000. Load the first two objects into sector YA.
3. Set (P)=XX003 and (A)=YB000. Load the next two objects into sector YB.
4. Set (P)=XX003 and (A)=YC000. Load the next object into sector YC.
5. Set (P)=XX003, (A)=YD000, and (B)=YD770. Load the next five objects into sector YD.
6. Set (P)=XX003, (A)=YE000, and (B)=YE550. Load the next eleven objects into sector YE.
7. Obtain a memory map, (P)=XX002.
8. Refer to Appendix F.
9. Use PAL-AP to punch a self-loading tape of OFLUT-2 or -3.

BUILDING PROCEDURE FOR 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. The figures are separated for clarity.

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.
MINEX	70183045321
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	70181475321
or	
ASRD-L	70181476321
HSRD	70181500321
HSPD	70181501321
MHSD	70181774321
or	
DSUD	70181681321

Construction of OFLCUP

Perform the following steps in the order shown:

1. Load MINEX by setting (P)=XX000 (or XX006, if the program is to support systems with more than 16K of core) and (A)=YA000 and pressing START.
2. If the system uses the ASR reader/punch, proceed to step 3. Load ASRD-S by pressing START.
3. Load ROOT-C by setting (P)=XX003, (A)=YB000, and (B)=YB770 and pressing START.

4. Load TR-C by pressing START.

5. Load OFLCT by pressing START.

6. Load FCNLST-C by pressing START six times. Routines RC-C, PC-C, and PL-C are loaded.

7. Load IOD-C by pressing START.

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

9. Load OP-C by setting (P)=XX003, (A)=YC000, and (B)=YC670 and pressing START.

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

11. 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 14.

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

12. Load B2-C by pressing START.

13. Load ASRD-L by setting (P)=XX003 and (A)=YD000 and pressing START. Proceed to step 17.

14. Load B2-C by pressing START.

15. Load HSRD by setting (P)=XX003 and (A)=YD000 and pressing START.

16. Load HSPD by pressing START.

17. Load either MHSD (if the system contains a moving-head disk) or DSUD (if the system contains a drum storage unit) by setting (P)=XX003 and (A)=YE000 and pressing START.

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

19. Refer to Appendix F.

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

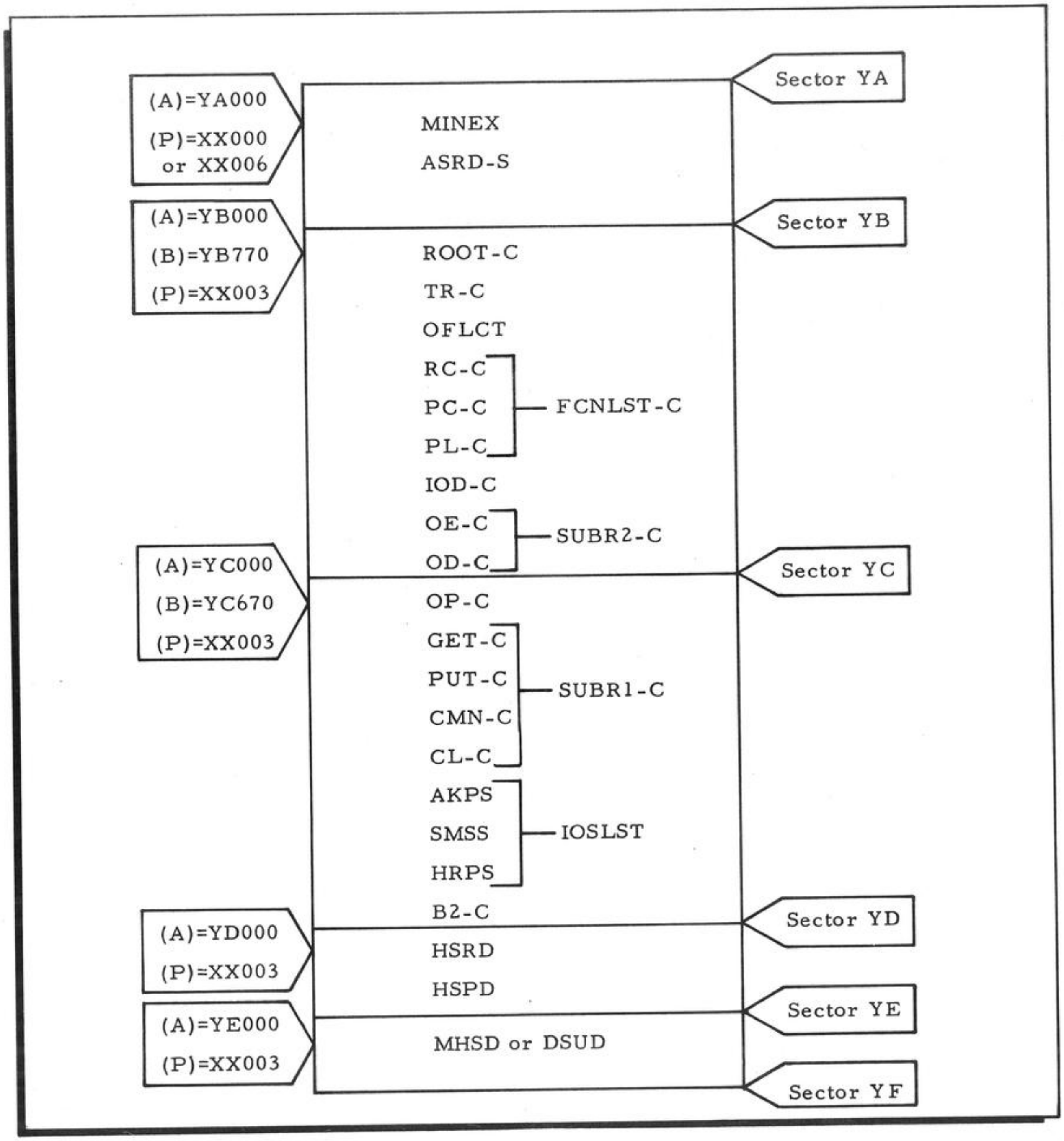


Figure 4-1a. Typical Core Layout of OFLCUP with High Speed Paper Tape Functions

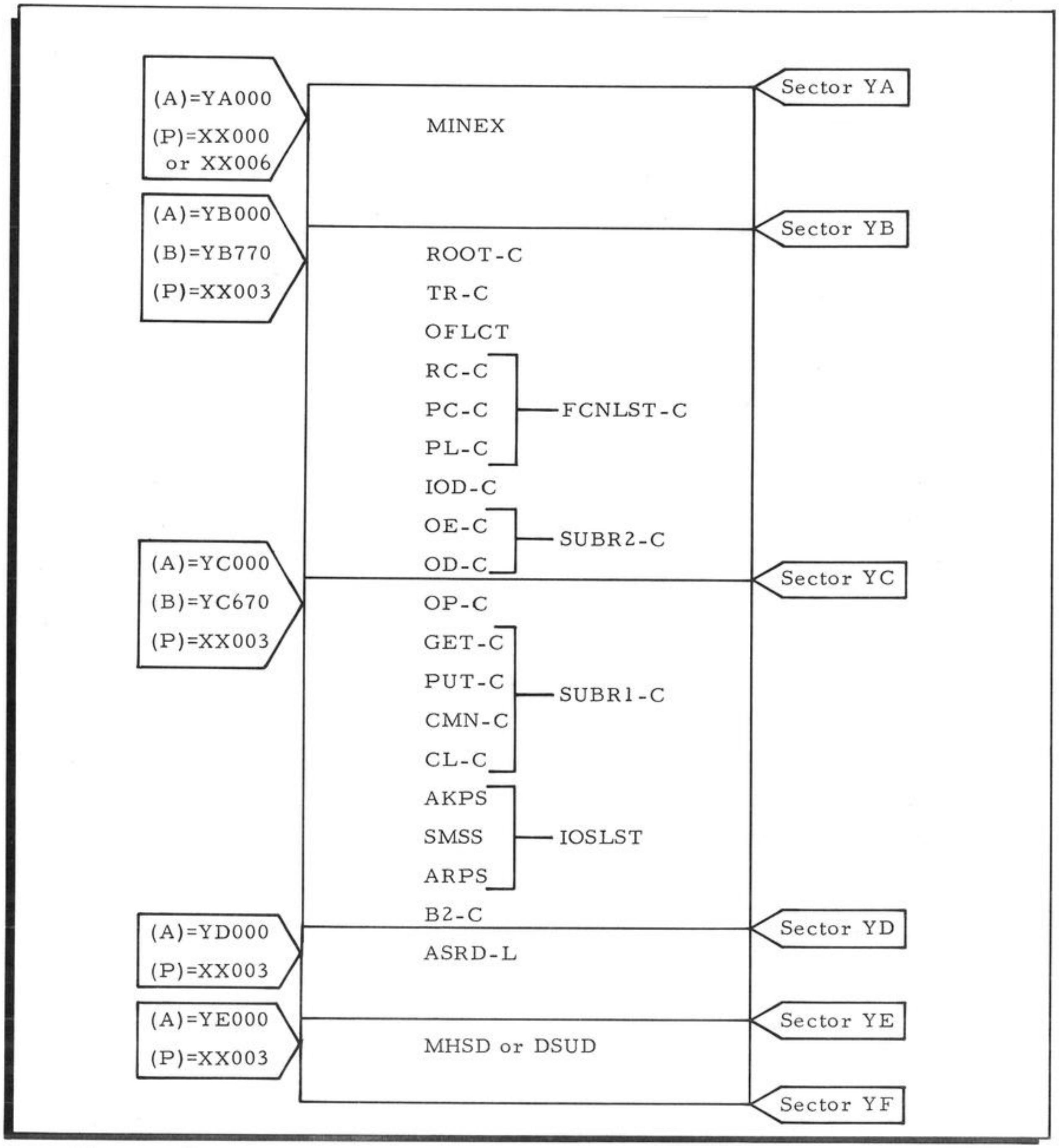


Figure 4-1b. Typical Core Layout of OFLCUP with ASR Paper Tape Functions

Part 1

*LOW	01000
*START	01000
*HIGH	05754
*NAMES	73041
*COMN	77700
*BASE	03724
*BASE	02774
EXEC	01000
SFET	01011
XCCW	01204
XMBP	01205
XPLP	01206
XMSF	01207
XLNK	01210
EROR	01211
XDCT	01212
XSPT	01213
SKT	01227
IMT	01237
DCT	01247
SPT	01256
MT	01277**
AS	01310
KB	02003
FD2	02042
ATSL	02061
ERR	02071
TERM	02076
RPRG	02100
PNME	02103
PAR3	02114
PAR4	02115
PAR5	02116
PAR6	02117
FE	02120
RT1	02120
ER	02121
PRT	02126
DRIV	02126
C1	02130
A1	02131
KBD	02133
CRLF	02140
COMA	02145
CRCK	02152
SP	02162

Part 2

P1	02164
P2	02165
CSA	02165
P3	02166
CEA	02166
S1	02167
S2	02170
S3	02171
LL	02172
HL	02173
AP	02177
AK	02200
IDN	02210
B1	02210
ODN	02211
TR	02250
FT	02312
RC	02354
PC	02410
PL	02470
RL	02503
ODK	02536
IDK	02546
VDK	02557
OE	02650
OD	02674
OP	03000
IP	03071
VP	03127
GET	03274
PUT	03306
CMN	03322
CL	03336
AKS	03352
APS	03357
SMS	03400
PRS	03424
PPS	03431
B2	03446
CKSM	03506
EOF	03507
PR	04006
PP	04320
SM	05006
MR	

Figure 4-2. Typical OFLCUP Memory Map

BUILDING PROCEDURE FOR 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, shown in Figure 4-5, and only loading the desired functions, or by generating a new version of FT-C and including only the desired functions.

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 as examples only.

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.

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	70183007321

Construction of ONLCUP

Perform the following steps in the order shown:

1. Load ROOT-C by setting (P)=XX000 (or XX006, if the program is to support systems with more than 16K of core), (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 routines 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 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 <u>must not</u> be used for the ASR device.
2. Coordination <u>may</u> be used for the ASR driver to prevent conflicts with other programs which call the ASR driver.

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

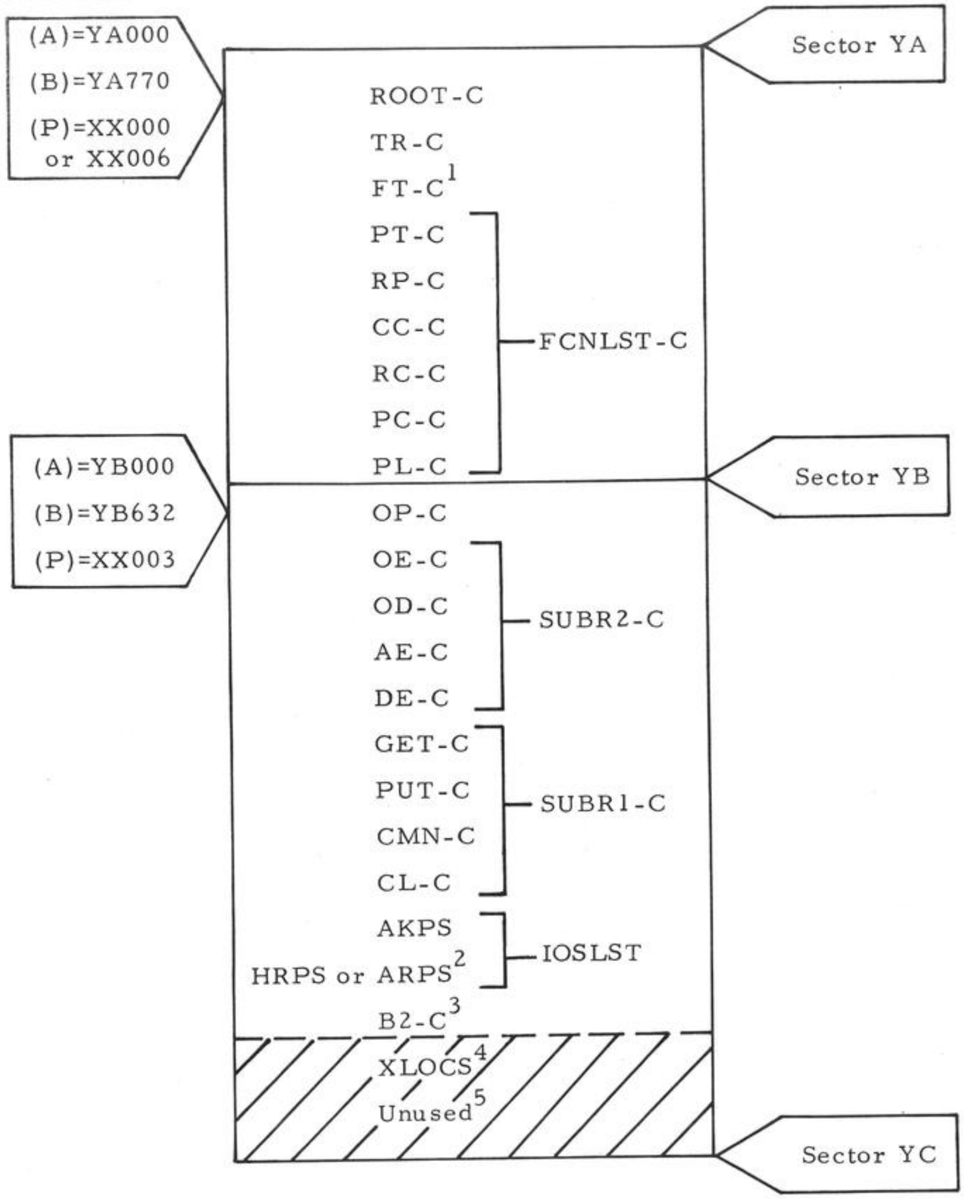
Component	Size	Other Utility Components Required
ROOT-C	168	TR-C[1], FT-C (or OFLCT), AKPS, XLOCS
TR-C[1]	34	None
FT-C[2]	44	None
OFLCT[2]	34	None
PT-C	76	PUT-C, DE-C
RP-C	24	OE-C, AE-C
CC-C	38	AE-C, DE-C
RC-C	28	OE-C, CL-C, OD-C
PC-C	48	OE-C, OD-C
PL-C	38	OE-C, OD-C
OP-C	188	OE-C, CL-C, OD-C, HRPS[3] (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	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.

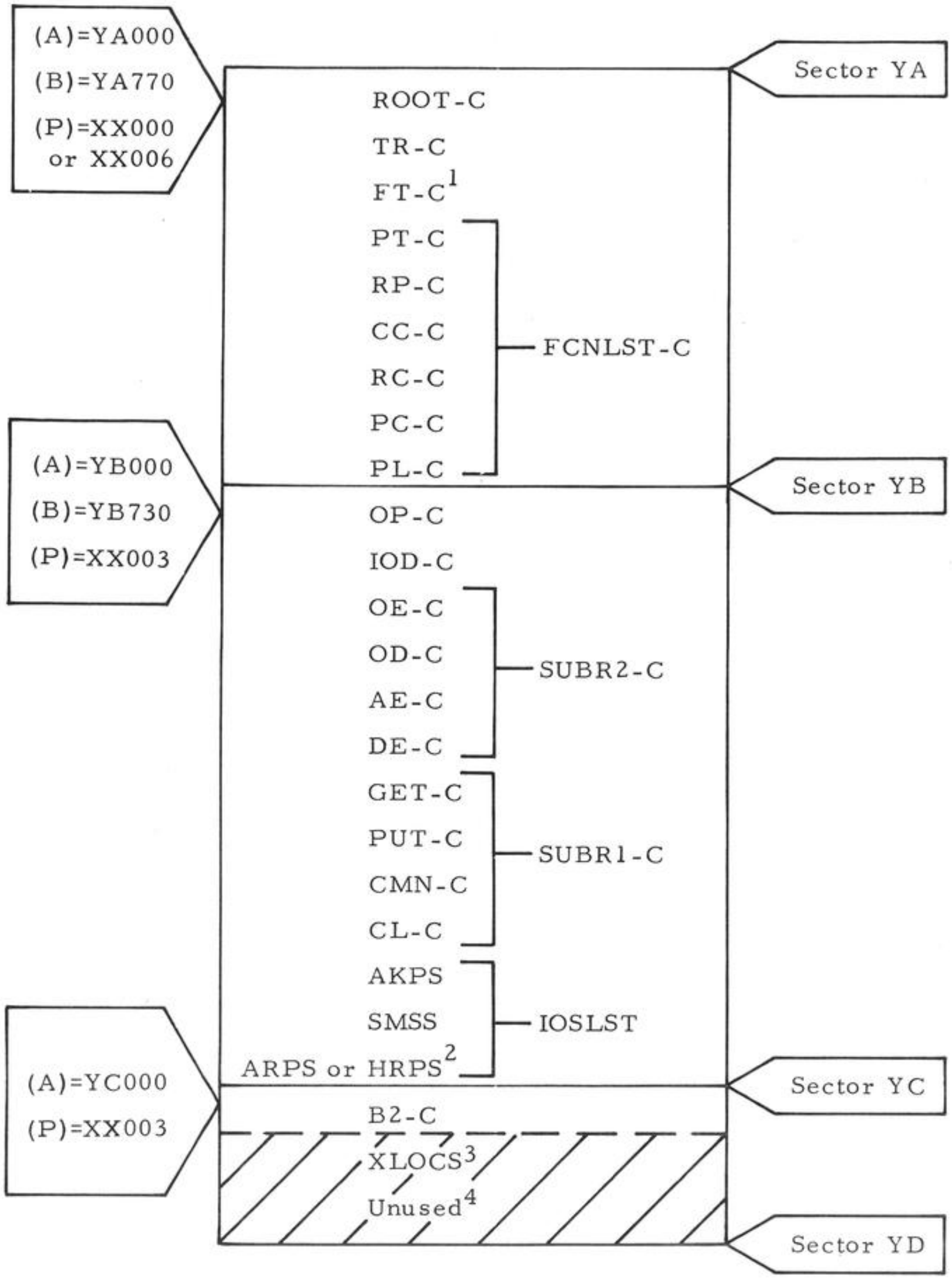
[2] The layout takes into account the size differential between HRPS and ARPS.

[3] Buffer B2-C is a BSS 128, but only 34 words are used in this example. The indirect links starting at YB632 are not destroyed.

[4] XLOCS consists entirely of EQU pseudo-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



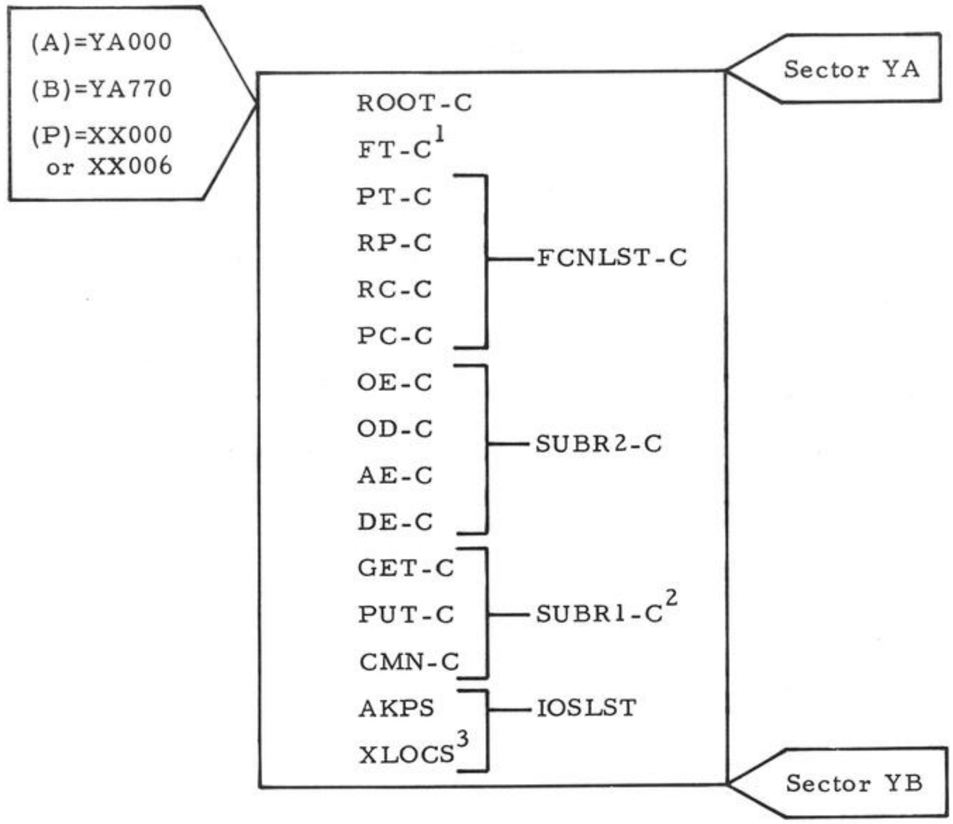
[1] In this example, FT-C includes all those functions shown in Figure 4-5 (Sheets 1 to 3).

[2] The layout takes into account the size differential between HRPS and ARPS.

[3] XLOCS 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 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

Part 1		Part 2		Part 3	
*LOW	00000	KB	05003	TR	05250
*START	05000	FD2	05042	FT	05312
*HIGH	06756	ATSL	05061	ODK	05351**
*NAMES	72635	ERR	05071	IDK	05354**
*COMN	77700	TERM	05076	VDK	05363**
*BASE	06664	RPRG	05100	PT	05366
*BASE	05777	PNME	05103	RT	05413
XSTR	01000	PAR3	05114	RP	05502
XLNK	01001	PAR4	05115	CC	05532
XPFP	01002	PAR5	05116	DC	05562
XMIL	01003	PAR6	05117	RC	05600
XSEC	01004	FE	05120	PC	05634
XMIN	01005	RT1	05120	PL	05714
XHR	01006	ER	05121	RL	05727
XDAY	01007	PRT	05126	OP	06000
XMSD	01010	DRIV	05126	IP	06071
XPIC	01011	C1	05130	VP	06127
XJBS	01012	A1	05131	OE	06274
XPLP	01013	KBD	05133	OD	06320
XPSP	01014	CRLF	05140	AE	06360
XCCW	01015	COMA	05145	DE	06376
XEPP	01016	CRCK	05152	GET	06430
EROR	01016	SP	05162	PUT	06442
XMBP	01017	P1	05164	CMN	06456
XPIF	01020	P2	05165	CL	06472
XMSF	01021	CSA	05165	AKS	06506
XICF	01022	P3	05166	APS	06513
XSSA	01023	CEA	05166	PRS	06534
XPLT	01025	S1	05167	PPS	06541
XPCT	01026	S2	05170	B2	06556
XCUT	01027	S3	05171	CKSM	06616
XIDT	01030	LL	05172	EOF	06617
XID1	01031	HL	05173		
XID2	01032	AP	05177	MR	
XIVT	01033	AK	05200		
XIRT	01034	IDN	05210		
XDCT	01035	B1	05210		
XSPT	01036	ODN	05211		
XOPT	01037				

Figure 4-4. Typical ONLCUP Memory Map

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

                                     SUBR   FT
                                     REL
                              *
                              *      FUNCTION TABLE
                              *             WORD 1  2 CHARACTER FUNCTION NAME
                              *             WORD 2  START ADDRESS
                              *
00000     150324              FT     BCI    1,PT                 PRINT TIME
00001     0 000000                   XAC    PT
00002     151324                     BCI    1,RT                 REPLACE TIME
00003     0 000000                   XAC    RT
00004     151320                     BCI    1,RP                 REQUEST PROGRAM
00005     0 000000                   XAC    RP
00006     141703                     BCI    1,CC                 CONNECT CLOCK
00007     0 000000                   XAC    CC
00010     142303                     BCI    1,DC                 DISCONNECT CLOCK
00011     0 000000                   XAC    DC
00012     151303                     BCI    1,RC                 REPLACE CORE
00013     0 000000                   XAC    RC
00014     150303                     BCI    1,PC                 PRINT CORE
00015     0 000000                   XAC    PC
00016     151314                     BCI    1,RL                 REPLACE LIMITS
00017     0 000000                   XAC    RL
00020     150314                     BCI    1,PL                 PRINT LIMITS
00021     0 000000                   XAC    PL
00022     152322                     BCI    1,TR                 TRANSFER DATA
00023    -0 000027                   DAC*   TT                   POINTER TO TRANSFER TABLE
00024     153305                     BCI    1,VE                 VERIFY DATA
00025    -0 000044                   DAC*   VT                   POINTER TO VERIFY TABLE
00026     000000                     OCT    0                    END OF FUNCTION TABLE
                                     EJCT
```

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

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

                        *
                        *  TRANSFER TABLE
                        *         WORD 1  2 CHARACTER INPUT DEVICE NAME
                        *         WORD 2  2 CHARACTER OUTPUT DEVICE NAME
                        *         WORD 3  START ADDRESS
                        *
00027     150322        TT   BCI  2,PRCO                PAPER TAPE TO CORE
00030     141717
00031     0 000000           XAC  IP
00032     141717             BCI  2,COPP                CORE TO PAPER TAPE
00033     150320
00034     0 000000           XAC  OP
00035     141717             BCI  2,COSM                CORE TO MASS STORE
00036     151715
00037     0 000000           XAC  ODK
00040     151715             BCI  2,SMCO                MASS STORE TO CORE
00041     141717
00042     0 000000           XAC  IDK
00043     000000             OCT  0                     END OF TRANSFER TABLE
                             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
DAP-16 MOD 2     REV. B    10-20-70
```

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

BUILDING PROCEDURE FOR ONLMUP

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:

Tape Name	Document No.
IOSLST	70183001521
BASIC	70183006321
XLOCS	70183007321
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 has been assumed 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)=XX000 (or XX006, if the program is to support systems with more than 16K of core) and (A)=YA000 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 in the system, move IOSLST forward by hand, bypassing routine HRPS. Then

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

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   05154
*NAMES  22705
*COMN   26700
AKS     05000
APS     05005
SMS     05026
PRS     05052
PPS     05057
KB      05077
RPRG    05117
PNMF    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. 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 by pressing START. This will resolve the references to RTX-16 Executive locations XLNK, XSPT, and EROR and complete the loading of ROOT.

6. Use OFLCUP (entered at location EXEC 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:

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

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)=XX000 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

LC
```

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:

```
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
```

11. To load single overlays from the FCNLST1 and FCNLST2 tapes (e.g., RP or TRCOSM), set (P)=XX000 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)=XX000 and (A)=OLAP and pressing START.

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

```
*LOW    05140
*START  05140
*HIGH   05506
*NAMES  23031
*COMN   26700
VEMTSM  05154
OV1     05165

LC
```

The symbol OV1 indicates 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)=XX000 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.

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 for the ASR driver to prevent conflicts with other programs which call the ASR driver.

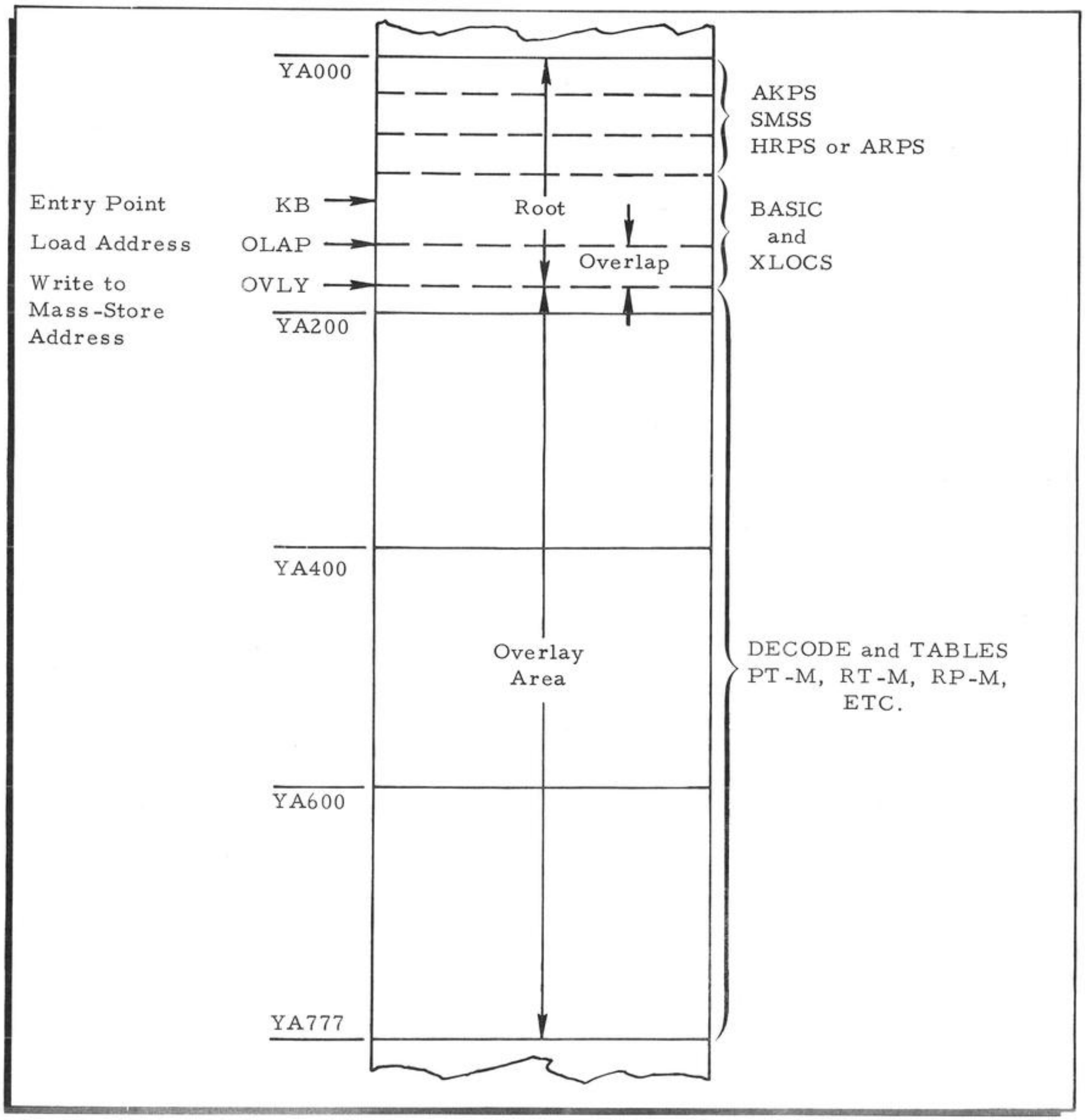


Figure 4-6. Typical Core Layout of ONLMUP

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 (500)

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

```
* NAME: TABLES          DOC 70183009000  REV B

                             SUBR   FT
                             REL
                       *
                       *     FUNCTION TABLE
                       *            WORD 1 BITS 1-16  2 CHARACTER FUNCTION NAME
                       *            WORD 2 BITS 1-4   OVERLAY SIZE IN SEGMENTS
                       *            WORD 2 BITS 5-16  RELATIVE SEGMENT NUMBER
                       *
00000   150324         FT    BCI    1,PT              PRINT TIME
00001   010002               OCT    10002
00002   151324               BCI    1,RT              REPLACE TIME
00003   010003               OCT    10003
00004   151320               BCI    1,RP              REQUEST PROGRAM
00005   010004               OCT    10004
00006   141703               BCI    1,CC              CONNECT CLOCK
00007   010005               OCT    10005
00010   142303               BCI    1,DC              DISCONNECT CLOCK
00011   010006               OCT    10006
00012   151303               BCI    1,RC              REPLACE CORE
00013   020007               OCT    20007
00014   150303               BCI    1,PC              PRINT CORE
00015   020011               OCT    20011
00016   143303               BCI    1,FC              FILL CORE
00017   010013               OCT    10013
00020   151703               BCI    1,SC              SEARCH CORE
00021   020014               OCT    20014
00022   150314               BCI    1,PL              PRINT LIMITS
00023   010016               OCT    10016
00024   151314               BCI    1,RL              REPLACE LIMITS
00025   010017               OCT    10017
00026   152322               BCI    1,TR              TRANSFER DATA
00027  -0 000033             DAC*   TT                POINTER TO TRANSFER TABLE
00030   153305               BCI    1,VE              VERIFY DATA
00031  -0 000072             DAC*   VT                POINTER TO VERIFY TABLE
00032   000000               OCT    0                 END OF FUNCTION TABLE
                             EJCT
```

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

```
                          * NAME: TABLES          DOC 70183009000  REV B


                          *
                          *   TRANSFER TABLE
                          *           WORD 1 BITS 1-16  2 CHARACTER INPUT DEVICE NAME
                          *           WORD 2 BITS 1-16  2 CHARACTER OUTPUT DEVICE NAME
                          *           WORD 3 BITS 1-4   OVERLAY SIZE IN SEGMENTS
                          *           WORD 3 BITS 5-16  RELATIVE SEGMENT NUMBER
                          *
00033     141717          TT  BCI     2,COSM            CORE TO MASS STORE
00034     151715
00035     010020              OCT     10020
00036     151715              BCI     2,SMCO            MASS STORE TO CORE
00037     141717
00040     010021              OCT     10021
00041     141717              BCI     2,COPP            CORE TO PAPER TAPE
00042     150320
00043     010024              OCT     10024
00044     150322              BCI     2,PRCO            PAPER TAPE TO CORE
00045     141717
00046     020025              OCT     20025
00047     151715              BCI     2,SMPP            MASS STORE TO PAPER TAPE
00050     150320
00051     020031              OCT     20031
00052     150322              BCI     2,PRSM            PAPER TAPE TO MASS STORE
00053     151715
00054     020033              OCT     20033
00055     141717              BCI     2,COMT            CORE TO MAG TAPE
00056     146724
00057     020041              OCT     20041
00060     146724              BCI     2,MTCO            MAG TAPE TO CORE
00061     141717
00062     020043              OCT     20043
00063     151715              BCI     2,SMMT            MASS STORE TO MAG TAPE
00064     146724
00065     020051              OCT     20051
00066     146724              BCI     2,MTSM            MAG TAPE TO MASS STORE
00067     151715
00070     020053              OCT     20053.
00071     000000              OCT     0                 END OF TRANSFER TABLE
                              EJCT
```

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

```
                                   * NAME: TABLES          DOC 70183009000  REV B

0098                               *
0099                               *    VERIFY TABLE
0100                               *         WORD 1 BITS 1-16  2 CHARACTER INPUT DEVICE NAME
0101                               *         WORD 2 BITS 1-16  2 CHARACTER OUTPUT DEVICE NAME
0102                               *         WORD 3 BITS 1-4   OVERLAY SIZE IN SEGMENTS
0103                               *         WORD 3 BITS 5-16  RELATIVE SEGMENT NUMBER
0104                               *
0105 00072    151715               VT   BCI  2,SMCO            MASS STORE AGAINST CORE
     00073    141717
0106 00074    020022                    OCT  20022
0107 00075    150322                    BCI  2,PRCO            PAPER TAPE AGAINST CORE
     00076    141717
0108 00077    020027                    OCT  20027
0109 00100    150322                    BCI  2,PRSM            PAPER TAPE AGAINST MASS STORE
     00101    151715
0110 00102    020035                    OCT  20035
0111 00103    146724                    BCI  2,MTCO            MAG TAPE AGAINST CORE
     00104    141717
0112 00105    020045                    OCT  20045
0113 00106    146724                    BCT  2,MTSM            MAG TAPE AGAINST MASS STORE
     00107    151715
0114 00110    020055                    OCT  20055
0115 00111    000000                    OCT  0                 END OF VERIFY TABLE
0116                                    END

FT       000000     TT       000033     VT       000072

0000 WARNING OR ERROR FLAGS
DAP-16 MOD 2     REV. B      10-20-70
```

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

BUILDING PROCEDURE FOR 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 used to build OFLMUP 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:

Tape Name	Document No.
IOSLST	70183001521
BASIC	70183006321
DECODE	70183008321
OFLMT or User-generated OFLMT tape	70183048321
MINEX	70183045321
ASRD-S	70181475321
or	
ASRD-L	70181476321
MHSD	70181744321
or	
DSUD	70181681321
HSRD	70181500321
HSPD	70181501321
MTUD	70181622321

Construction of OFLMUP

It is assumed 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

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

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)=XX000 (or XX006, if the program is to support systems with more than 16K of core) 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:

```
*LOW    05000
*START  05000
*HIGH   05154
*NAMES  22705
*COMN   26700
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 MINEX by setting (P)=XX003 and (A)=YA000 and pressing START. Obtain a memory map, which should appear as follows for YA000=4000:

```
*LOW    04000
*START  05000
*HIGH   05154
*NAMES  73545
*COMN   77700
EXEC    04000
SFET    04011
XCCW    04204
XMBP    04205
XPLP    04206
XMSF    04207
XLNK    04210
EROR    04211
XDCT    04212
XSPT    04213
SKT     04227
IMT     04237
DCT     04247
SPT     04256
AS      04267**
PR      04271**
PP      04273**
SM      04275**
MT      04277**
AKS     05000
APS     05005
SMS     05026
PRS     05052
PPS     05057
KB      05077
RPRG    05117
PNME    05122
PAR3    05133
PAR4    05134
PAR5    05135
OLAP    05140
OVLY    05154

MR
```

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

7. Load the mass-store driver (MHSD, DSUD, 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 by setting (P)=XX003 and (A)=YE000, and pressing START.

9. Load HSPD 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 OFLMUP. Obtain a memory map. A typical memory map is presented on the following page for YC000=6000, YD000=7000, YE000=10000, and YF000=11000.

```
*LOW     04000
*START   05000
*HIGH    10332
*NAMES   73545
*COMN    77700
EXEC     04000
SFET     04011
XCCW     04204
XMBP     04205
XPLP     04206
XMSF     04207
XLNK     04210
EROR     04211
XDCT     04212
XSPT     04213
SKT      04227
IMT      04237
DCT      04247
SPT      04256
MT       04310
AKS      05000
APS      05005
SMS      05026
PRS      05052
PPS      05057
KB       05077
RPRG     05117
PNME     05122
PAR3     05133
PAR4     05134
PAR5     05135
OLAP     05140
OVLY     05154
SM       06006
PR       07006
PP       07320
AS       10006

LC
```

NOTE: The unresolved references MTI1, MTI2, ASF1, and ASF2 are links to Fortran "caps" and may be ignored.

13. Refer to 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.

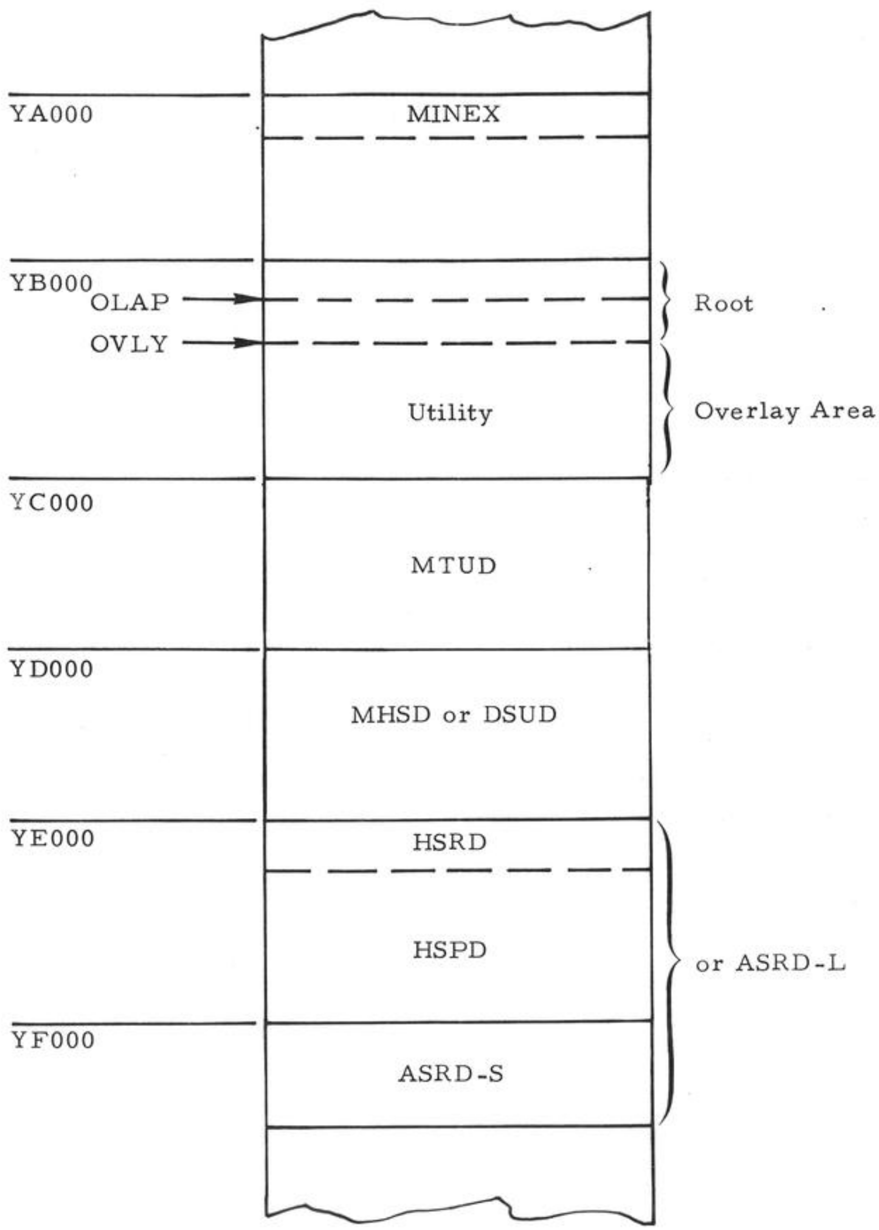


Figure 4-9. Typical Core Layout for OFLMUP.

APPENDIX A
KEYBOARD FUNCTIONS

The formats for the 26 keyboard commands of the OP-16 utility program are presented below:

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 CODES

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

Table B-1. Keyboard Messages

Message	Meaning
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 program are presented in Table B-2. Device names which could be added in the future are also identified.

Table B-2. Device Codes

Mnemonic	Device Name	Present/Future
CO	Core	Present
PP	Paper Tape Punch	
PR	Paper Tape Reader	
SM	System Mass Store	
MT	Magnetic Tape	
CR	Card Reader	Future
LP	Line Printer	
CP	Card Punch	

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 TO DECIMAL CONVERSION

6		5		4		3		2		1	
OCT	DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT	DEC
0	0	0	0	0	0	0	0	0	0	0	0
1	32768	1	4096	1	512	1	64	1	8	1	1
2	65536	2	8192	2	1024	2	128	2	16	2	2
3	98304	3	12288	3	1536	3	192	3	24	3	3
4	131072	4	16384	4	2048	4	256	4	32	4	4
5	163840	5	20480	5	2560	5	320	5	40	5	5
6	196608	6	24576	6	3072	6	384	6	48	6	6
7	229376	7	28672	7	3584	7	448	7	56	7	7

EXAMPLES:

1. Convert 6154_8 to Decimal.

```
6 = 3072
1 =   64
5 =   40
4 = +  4
    ----
    3180
```

$3180_{10} = 6154_8$

2. Convert 3180_{10} to Octal.

```
 3180
-3072 = 6
 ----
  108
-  64 = 1
 ----
   44
-  40 = 5
 ----
    4
-   4 = 4
 ----
    0
```

$6154_8 = 3180_{10}$

APPENDIX E

COMPONENTS AND ROUTINES REQUIRED BY EACH UTILITY

Component	Routine	ONLMUP	OFLMUP	
MINEX and Device Drivers	-		X	
IOSLST	AKPS	X	X	
	SMSS	X	X	
	HRPS	X	X	
	ARPS	X	X	
BASIC	-	X	X	
XLOCS	-	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	
	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	

Component	Routine	ONLCUP	OFLCUP
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
MINEX and Device Drivers	–		X
IOD-C	–	X	X
IOSLST	AKPS	X	X
	SMSS	X	X
	HRPS	X	X
	ARPS	X	X
XLOCS	–	X	

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 SKT, IMT, DCT, and SPT. These are the start addresses of four tables in the utility component, MINEX, which contain user-dependent information. Figure F-1 shows a sample memory map for MINEX, loaded in sector 1, with the relevant locations underlined. Figure F-2 shows the structure and contents of the tables. Items marked with an asterisk may have to be modified, as indicated, to describe the user's configuration. This may be done by means of the control panel or, in the case of the core-resident utilities (OFLCUP, OFLUT -1, -2, -3), by starting up and using the utility itself.

The addresses of the items to be modified are calculated as follows:

Address	Example
SKT + 1	'1227 + 1 = '1230
IMT + 1	'1237 + 1 = '1240
DCT	'1247
DCT + 1	'1247 + 1 = '1250
DCT + 6	'1247 + 6 = '1255
SPT + 5	'1256 + 5 = '1263

```
*LOW      01000
*START    01000
*HIGH     01302
*NAMES    73655
*COMN     77700
EXEC      01000
SFET      01011
XCCW      01204
XMBP      01205
XPLP      01206
XMSF      01207
XLNK      01210
EROR      01211
XDCT      01212
XSPT      01213
SKT       01227
IMT       01237
DCT       01247
SPT       01256
KB        01265**
AS        01267**
PR        01271**
PP        01273**
SM        01275**
MT        01277**

MR
```

Figure F-1. Sample of Memory Map of MINEX

```
                              *
                              *    SKS TABLE
     00227    100000          SKT  SKP                 NOT USED
     00230    34 0125              SKS    '125        *MASS-STORE: '70125 FOR MHD/'70422 FOR DSU
     00231    34 0401              SKS    '401         HIGH SPEED PAPER TAPE READER
     00232    34 0402              SKS    '402         HIGH SPEED PAPER TAPE PUNCH
     00233    34 0404              SKS    '404         ASR TELETYPE
     00234    100000               SKP                 NOT USED
     00235    100000               SKP                 NOT USED
     00236    34 0410              SKS    '410         MAGNETIC TAPE UNIT
                              *
                              *    INTERRUPT MASK TABLE
     00237    000000          INT  OCT    0            NOT USED
     00240    010000               OCT    10000       *MASS-STORE: '10000 FOR MHD/'400 FOR DSU
     00241    000200               OCT    200          HIGH SPEED PAPER TAPE READER
     00242    000100               OCT    100          HIGH SPEED PAPER TAPE PUNCH
     00243    000040               OCT    40           ASR TELETYPE
     00244    000000               OCT    0            NOT USED
     00245    000000               OCT    0            NOT USED
     00246    100000               OCT    100000       MAGNETIC TAPE UNIT
                              *
                              *    DUMMY XDCT TABLE
     00247    000024          DCT  OCT    24          *MASS-STORE TYPE: '24 FOR 20-SURFACE MHD/
                              *                        2 FOR 2-SURFACE MHD/CAPACITY OF DSU IN
                              *                        SEGMENTS
     00250    000001               OCT    1           *ASR TYPE: 1 FOR 33/0 FOR 35
     00251    000000               OCT    0            NOT USED
     00252    000000               OCT    0            NOT USED
     00253    000000               OCT    0            NOT USED
     00254    000000               OCT    0            NOT USED
     00255    000002               OCT    2           *CHANNEL NUMBER FOR MAGNETIC TAPE UNIT
                              *
                              *    DUMMY XSPT TABLE
     00256    000000          SPT  OCT    0            NOT USED
     00257    000000               OCT    0            NOT USED
     00260    000000               OCT    0            NOT USED
     00261    000001               OCT    1            LOW LIMIT FOR CORE MODIFICATION
     00262    077777               OCT    77777        HIGH LIMIT FOR CORE MODIFICATION
     00263    000010               OCT    10          *BASE SEGMENT NUMBER FOR UTILITY OVERLAYS
```

Figure F-2. MINEX Tables

HONEYWELL INFORMATION SYSTEMS

Publications Remarks Form*

TITLE:	SERIES 16 OP-16 UTILITY PROGRAMS	ORDER No.:	AB63, REV. 0
		DATED:	MAY, 1971

ERRORS IN PUBLICATION:

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION:

(Please Print)

FROM: NAME ______________________ DATE: ______________

COMPANY ______________________

TITLE ______________________

____

____

*Your comments will be promptly investigated by appropriate technical personnel, action will be taken as required, and you will receive a written reply. If you do not require a written reply, please check here. ☐

CUT ALONG LINE

CUT ALONG LINE

FIRST CLASS
PERMIT NO. 39531
WELLESLEY HILLS,
MASS. 02181

Business Reply Mail
Postage Stamp Not Necessary if Mailed in the United States

POSTAGE WILL BE PAID BY:

HONEYWELL INFORMATION SYSTEMS
60 WALNUT STREET
WELLESLEY HILLS, MASS. 02181

ATTN: PUBLICATIONS, MS 050

Honeywell