pdp11

MUMPS-11
Operator's Guide

Order No. DEC-11-MMOPA-E-D

digital

MUMPS-11
Operator's Guide

Order No. DEC-11-MMOPA-E-D

digital equipment corporation · maynard, massachusetts

First Printing, October 1972
Revised: October 1973
October 1974
February 1976
May 1977

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by DIGITAL.

The postage prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL
DEC
PDP
DECUS
UNIBUS
COMPUTER LABS
COMTEX
DDT
DECCOMM
DECsystem-10
DECtape
DIBOL
EDUSYSTEM
FLIP CHIP
FOCAL
INDAC
LAB-8
DECsystem-20
MASSBUS
OMNIBUS
OS/8
PHA
RSTS
RSX
TYPESET-8
TYPESET-10
TYPESET-11

CONTENTS

			Page
FOREWORD			xi
PREFACE			xiii
ACKNOWLEDGMENT			xvii
CHAPTER	1	INTRODUCTION	1-1
	1.1	HOW THE MUMPS-11 SYSTEM IS SUPPLIED	1-1
	1.1.1	MUMPS-11 System Builder	1-2
	1.1.2	MUMPS SYSGEN, System and Library Utility Programs	1-2
	1.1.3	MUMPS Backup and Utility System	1-4
	1.1.4	Documentation	1-4
	1.2	REQUIRED MATERIALS	1-4
CHAPTER	2	BUILDING THE BASIC SYSTEM	2-1
	2.1	INTRODUCTION	2-1
	2.2	SUMMARY OF OPERATIONS	2-1
	2.3	PREOPERATIONAL INFORMATION	2-2
	2.3.1	General	2-2
	2.3.2	Reserved Devices	2-2
	2.4	ERROR CONDITIONS	2-3
	2.5	SYSTEM STARTUP WITH BOOTSTRAP ROMS	2-3
	2.5.1	Using BM792-YB Bootstrap ROM	2-3
	2.5.2	Using MR11-DB Bootstrap ROM	2-4
	2.5.3	Using BM873-YA Bootstrap ROM	2-4
	2.5.4	Using BM873-YB Bootstrap ROM	2-4
	2.5.5	Using M9301-YA or M9301-YB Bootstrap ROM (PDP-11/34)	2-5
	2.6	DETAILED DESCRIPTION OF OPERATION - MAGTAPE DISTRIBUTION	2-5
	2.6.1	Loading Procedures - MUMPS-11 System Builder	2-6
	2.6.2	User-Created Object Modules	2-8
	2.6.3	Answering SYSBLD Questions	2-9
	2.6.4	Restarting the Basic System From DECtape or Magtape	2-11
	2.7	DETAILED DESCRIPTION OF OPERATION-DISK PACK DISTRIBUTION	2-12
	2.7.1	Preoperational Information	2-13
	2.7.2	Loading Procedures - MUMPS-11 System Builder	2-13
	2.7.3	Running SYSBLD	2-14
	2.7.4	Restarting the Basic System From Disk	2-14
	2.8	SYSBLD QUESTIONS	2-15
CHAPTER	3	SYSTEM GENERATION PROCEDURES	3-1
	3.1	INTRODUCTION	3-1

CONTENTS (Cont.)

Page

3.2 ORGANIZATION 3-1
3.2.1 Program Structure 3-1
3.2.2 SYSGEN Tape 3-2
3.3 ANSWERING SYSGEN'S QUESTIONS 3-2
3.4 OPERATION 3-3
3.5 PREOPERATIONAL INFORMATION 3-3
3.5.1 Memory Size 3-3
3.5.2 Disk Memory 3-4
3.5.3 Partitions 3-4
3.5.4 I/O Devices 3-5
3.5.5 Directory Allocation 3-7
3.5.6 Restricting Disk Storage Space for Programs and Globals 3-8
3.5.7 Device Interrupt Vector Locations 3-8
3.6 STARTING PROCEDURE - MAGTAPE DISTRIBUTION 3-8
3.7 STARTING PROCEDURE - RK05, RK06 DISTRIBUTION 3-9
3.8 ERROR CONDITIONS 3-10
3.9 DETAILED DESCRIPTION OF OPERATION 3-11
3.9.1 Memory and Disk Memory Availability Questions 3-11
3.9.2 I/O Device Availability Questions 3-14
3.9.3 Device Buffer Allocation Questions 3-18
3.9.4 Partition Allocation Questions 3-20
3.9.5 Disk Inclusion and System Library Transfer 3-22
3.9.6 Modify System UCI's and PAC Questions 3-23
3.9.7 Modify Terminal Device Parameters Questions 3-25
3.9.8 Multiplexer Terminal Line Conditioning Questions 3-28
3.9.9 Modify Magtape Default Mode Questions 3-31
3.9.10 Miscellaneous System Information Questions 3-33
3.9.11 Saving The New System 3-35
3.9.12 Changing System Parameters 3-36

CHAPTER 4 OPERATOR FUNCTIONS 4-1

4.1 INTRODUCTION 4-1
4.2 PDP-11 SWITCH REGISTER USAGE AND CONSOLE BUTTON USAGE 4-1
4.2.1 Switch 0 4-2
4.2.2 Switch 4 4-3
4.2.3 Switch 6 4-3
4.2.4 Switch 12 4-3
4.2.5 Switch 13 4-4
4.2.6 Switch 15 4-4
4.3 USING THE CONSOLE TERMINAL 4-5
4.3.1 Running Mini-ODT 4-5
4.3.1.1 Operation 4-6
4.3.1.2 Opening a Location 4-6
4.3.1.3 Changing the Contents of a Location 4-6
4.3.1.4 Closing a Location Without Change 4-7
4.3.1.5 Advancing to the Next Location 4-7
4.3.1.6 Correcting Errors 4-7
4.3.2 Using the Console Terminal as a MUMPS Terminal 4-8
4.4 CREATING SYSTEM BACKUP 4-8
4.4.1 Logical Backup 4-8
4.4.2 Physical Backup 4-8

CONTENTS (Cont.)

Page

4.4.3 Saving the System Image 4-9
4.5 LOADING, STARTING AND RESTARTING THE SYSTEM 4-10
4.5.1 System Restart 4-11
4.5.2 Loading the System Into Memory from Disk 4-11
4.5.3 Restoring the System Disk from Backup Media 4-12
4.6 SYSTEM SHUTDOWN 4-13

CHAPTER 5 ERROR DETECTION AND RECOVERY 5-1

5.1 PROGRAMMING AND OPERATING SYSTEM ERRORS 5-1
5.1.1 Programming Errors 5-1
5.1.1.1 NAKED 5-2
5.1.1.2 DKSER 5-2
5.1.2 Operating System Errors (non-Garbage Collector) 5-2
5.1.2.1 DKDER 5-2
5.1.2.2 DKHER 5-2
5.1.2.3 PLDER 5-3
5.1.2.4 SWAP 5-3
5.1.2.5 SYSDG 5-3
5.1.3 Garbage Collector Errors 5-3
5.1.4 Errors Requiring DEC Assistance 5-4
5.2 MASS STORAGE HARDWARE ERRORS 5-5
5.3 ERRORS THAT CAUSE THE SYSTEM TO CRASH 5-6
5.3.1 MUMPS Interrupt Vector Utilization 5-7
5.3.1.1 Interrupt Vectors Ignored by MUMPS 5-7
5.3.1.2 Interrupts Processed by MUMPS 5-8
5.3.1.3 Hard Crash Errors Set Up by MUMPS 5-8
5.3.2 Soft Crash 5-8
5.3.3 Hard Crash 5-9

CHAPTER 6 SYSTEM UTILITY AND TEST PROGRAMS 6-1

6.1 INTRODUCTION 6-1
6.2 SYSTEM UTILITY PROGRAM DESCRIPTIONS 6-2
6.2.1 BCS - Broadcast 6-3
6.2.2 CTK - Caretaker 6-4
6.2.3 DAT - Set Date 6-5
6.2.4 DBT - Disk Block Tally 6-5
6.2.5 DMP - Disk Block Dump 6-6
6.2.6 KTR - Caretaker Reporter 6-10
6.2.7 MSP - Modify System Parameters 6-12
6.2.8 RKC - RK Copy 6-12
6.2.9 RSJ - Restore Job 6-14
6.2.10 RST - Restore Devices 6-15
6.2.11 SDP - Sequential Disk Processor Space Allocator and Deallocator 6-17
6.2.12 SIF - System Information 6-20
6.2.13 SS - System Status 6-21
6.2.13.1 Job Number 6-21
6.2.13.2 State 6-21
6.2.13.3 Devices Owned 6-23
6.2.14 SSD - System Shutdown 6-23
6.2.15 STU - System Startup 6-25
6.2.16 TIM - Set Time 6-25

CONTENTS (Cont.)

Page

6.2.17 %GP - Global Place 6-25
6.3 SYSTEM DEMO PACKAGE 6-27

APPENDIX A GLOSSARY OF TERMS A-1

APPENDIX B MUMPS CHARACTER SET B-1

APPENDIX C EXPLANATION OF MUMPS MESSAGES C-1

C.1 MUMPS PROGRAMMING ERROR MESSAGES C-2
C.2 VOLUNTARY PROGRAM TERMINATION C-4
C.3 DEBUGGING AID MESSAGE C-4
C.4 MUMPS OPERATING SYSTEM ERROR MESSAGES C-5

APPENDIX D SYMBOL USAGE D-1

APPENDIX E CONVERSION TABLES E-1

APPENDIX F ↑SYS SYSTEM GLOBAL LAYOUT F-1

APPENDIX G MAGTAPE LOADING AND UNLOADING PROCEDURES G-1

G.1 LOADING AND THREADING TAPE G-1
G.2 UNLOADING TAPE G-2

APPENDIX H NON-STANDARD TERMINAL DEVICE CONNECTION PROCEDURES H-1

H.1 INTRODUCTION H-1
H.2 PROCEDURE H-1

APPENDIX I MUMPS BACKUP AND UTILITY SYSTEM I-1

I.1 INTRODUCTION I-1
I.2 HOW TO USE MBU I-1
I.3 BEGINNER'S GUIDE TO MBU I-2
I.4 MBU COMMAND DESCRIPTIONS I-3
I.4.1 The LABEL Command I-3
I.4.2 The BACKUP Command I-4
I.4.3 The RESTORE Command I-5
I.4.4 The COPY and COMPARE Commands I-5
I.4.5 The SAVE-GLOBALS Command I-6
I.4.5.1 Redirecting a Physical Unit I-7
I.4.5.2 Restoring Globals Saved by MBU I-8
I.4.6 The FORMAT and INIT Commands I-9
I.4.7 The BOOT Command I-9
I.4.8 The SET Command I-9
I.4.9 The BB Command I-10
I.4.10 The MM Command I-11
I.5 SPECIAL DEVICES I-11
I.5.1 Special RP06 Considerations I-11
I.5.2 Special Magtape Considerations I-13
I.5.2.1 Magtape Density I-13
I.6 COMPUTING BLOCK NUMBERS I-13
I.6.1 Computing Absolute Block Number from Cylinder/Track/Sector I-14
I.6.2 Connecting To Or From Mumps Block Number I-14

CONTENTS (Cont.)

			Page
	I.7	HOW TO MAKE BOOTABLE COPIES OF MBU	I-17
	I.8	SAMPLE MBU COMMAND SUMMARY	I-18
APPENDIX J		PRESRV	J-1
	J.1	INTRODUCTION	J-1
	J.2	PRESRV COMMANDS	J-1
	J.2.1	File Transfer Command Strings	J-1
	J.2.1.1	Blocking Switch (/BL:nnn)	J-3
	J.2.1.2	Abort on LUN Error (/ER)	J-4
	J.2.1.3	Physical Format Device Switch (/FO)	J-5
	J.2.1.4	Image Copy Switch (/IM)	J-5
	J.2.1.5	Phase Encoded Tape Switch (/PE)	J-5
	J.2.1.6	Rewind at Completion (/RW)	J-5
	J.2.1.7	Verify Switch (/VE)	J-5
	J.2.2	Non-transfer Commands	J-5
	J.2.2.1	Set Vector Address Command (/VEC)	J-6
	J.2.2.2	Help Command (/HE)	J-6
	J.2.2.3	List Command (/LI)	J-6
	J.3	PRESRV OPERATOR GUIDE	J-7
	J.3.1	Starting Procedure	J-7
	J.3.1.1	Bootstrapping PRESRV	J-7
	J.3.1.2	Computing Required Blocks for Buffering	J-8
	J.3.1.3	Booting Another System or Stand-alone Program	J-9
	J.3.2	Operator Messages	J-9
	J.3.2.2	Error Messages	J-9
	J.4	EXAMPLES	J-11
	J.4.1	RK05 Disc Cartridge to Another RK05	J-11
	J.4.2	RK05 (RK11) to a TU10 (TM11) Magtape	J-11
	J.4.3	RP04, 05, 06 to Another RP04, 05, 06	J-11
	J.4.4	RH11 to a TJU16 Magtape	J-12
APPENDIX K		DSC	K-1
	K.1	OVERVIEW AND INTRODUCTION TO DSC	K-1
	K.2	INITIATING AND TERMINATING DSC	K-1
	K.3	DSC COMMAND FORMAT	K-2
	K.4	DSC - SUPPORTED DEVICES	K-3
	K.5	DATA TRANSFER	K-3
	K.6	OPTIONS	K-3
	K.6.1	Filelabel	K-3
	K.6.2	REWIND Switch (/RW)	K-4
	K.6.3	BAD BLOCK Switch (/BAD)	K-5
	K.6.4	APPEND Switch (/AP)	K-7
	K.6.5	NON-MERGE Mode Switch (/NMG)	K-7
	K.6.6	Density Switches	K-8
	K.6.6.1	1600-bpi Switch (/DENS=1600)	K-8
	K.6.6.2	SPLIT Density Switch (/DENS=800:1600)	K-9
	K.7	DSC OPERATION	K-10
	K.8	DATA TRANSFER FROM DISK	K-10
	K.9	DATA TRANSFER TO TAPE	K-11
	K.10	DATA TRANSFER FROM TAPE	K-12

CONTENTS (Cont.)

Page

K.11 DATA TRANSFER TO DISK K-13
K.12 DSC ERROR MESSAGES K-15

INDEX

TABLES

TABLE 2-1 Summary of System Build Steps - Magtape Distribution 2-2
3-1 DC11 Baud Rates 3-7
3-2 Partition Sizes 3-21
3-3 Multiplexer Terminal Speeds 3-30
5-1 System Table Error Statistics 5-5
6-1 System Utility Program Summary 6-2
6-2 Disk Block Allocation 6-17
6-3 System Test Package Programs 6-28
J-1 Device Designators J-2
J-2 Default Block Sizes J-4

FIGURES

Page

FIGURE 2-1 A Typical SYSBLD Session 2-10
2-2 SYSBLD Startup with Magtape Distribution 2-12
2-3 SYSBLD Startup with RK05 Distribution 2-14
3-1 SYSGEN Tape Logical Structure 3-2
3-2 SYSGEN Session: Memory and Disk Memory Availability 3-11
3-3 SYSGEN Session: Memory and Disk Memory Availability - RK05 or RK06 Distribution 3-12
3-4 SYSGEN Session: I/O Device Availability 3-14
3-5 SYSGEN Session: Device Buffer Allocation 3-18
3-6 SYSGEN Session: Partition Allocation 3-20
3-7 SYSGEN Session: Modify UCI's and PAC 3-23
3-8 SYSGEN Session: Modify Terminal Device Parameters 3-25
3-9 SYSGEN Session: Multiplexer Terminal Line Conditioning 3-29
3-10 SYSGEN Session: Modify Magtape Default Mode 3-31
3-11 SYSGEN Session: Miscellaneous Questions 3-33
4-1 PDP-11 Switch Register Console 4-2
4-2 Switch Register Function Assignments 4-4
4-3 PDP-11/34 Programmer's Console 4-5
6-1 Example of DBT Output 6-6
6-2 Example of DMP's Crash Analysis Report 6-8
6-3 Example of DMP's Disk Block Dump 6-9
6-4 Example of DMP's Memory Dump 6-10

CONTENTS (Cont.)

Page

6-5 Example of KTR Report 6-10
6-6 Example of RKC Dialogue 6-13
6-7 Example of RSJ Session 6-14
6-8 Example of RST Session 6-16
6-9 Example of SDP Dellocation Session 6-18
6-10 Example of SDP Allocaton Session 6-19
6-11 Example of SIF Report 6-20
6-12 Example of SS Report 6-21
6-13 Example of SSD Session 6-24
6-14 Example of STU Dialogue 6-25
F-1 SYS Nodes Built by SYSGEN F-2
F-2 SYS Layout for CTK Data F-3

FOREWORD

MUMPS-11 is an interactive, single-language, multi-user time-sharing system that allows access to a common data base. The capabilities of the system are primarily oriented towards string manipulation using a high-level language. The system relieves the user of most of the concern for programming peripheral devices or for structuring data bases in the traditional sense.

Language processing by the system is in every sense interpretive. Each line of code undergoes identical processing each time it is executed (intermediate code is not generated). The MUMPS applications programmer need not be concerned with problems associated with driving peripheral equipment or programming in assembly language. The programmer's only concern should be the development of a proper logical hierarchy for the data base, and the development of logic for the data processing requirement.

The MUMPS language is supported by a stand-alone operating system. In addition to implementing the MUMPS language and providing all operating system capabilities, the system affords the user a unique data base structure and access method. Data which is referred to symbolically is automatically stored and linked in a tree structure. The physical allocation of mass storage for the tree-structured data base is accomplished by the operating system. The data base thus created is available to other users in the system.

PREFACE

This manual provides reference information for system managers and operators of the MUMPS-11 software. It is assumed that the user has read and understood the related MUMPS-11 documentation listed below.

<u>MUMPS-11 Language Reference Manual</u> (DEC-11-MMLMA-D-DN1)

> Elements of the language, the character set, programming modes, program structure, data modes, numbers, strings, literals, constants, and variables.
>
> Expressions: how to form them and how they are evaluated.
>
> Commands: meaning, syntax, arguments, and examples of use.
>
> Functions: meaning, syntax, arguments, and examples of use.

<u>MUMPS-11 Programmer's Guide</u> (DEC-11-MMPGA-E-D)

This manual provides all information required to create, execute, and save MUMPS-11 programs.

> System Overview: MUMPS-11 hardware/software environment and functional description of the operating system.
>
> Terminal Usage: log-in/log-out procedures, terminal types, special keyboard control characters.
>
> Programming Techniques: creating programs, loading programs, storing programs, program size considerations, using system variables, conserving space.
>
> Using I/O Devices: general concepts of input/output, specific device characteristics.
>
> Library Utility Programs: functional characteristics, how to run them.

<u>MUMPS-11 Programmer's Reference Card</u> (DEC-11-MMPCA-D-D)

> Pocket reference card containing: command and function summaries, messages, symbology, character set, etc.

<u>Introduction to MUMPS-11 Language</u> (DEC-11-MMLTA-C-DN1)

> Step-by-step tutorial introduction to the MUMPS-11 Language.

<u>MUMPS-11 Release Notes</u> (DEC-11-MMGSA-B-D)

An overview of the MUMPS-11 software package containing useful hints and tips for MUMPS-11 users.

Sources of Supporting Information

The following paragraphs list applicable PDP-11 and MUMPS-11 documentation.

Standard PDP-11 Documentation

PDP-11/15, PDP-11/20, PDP-11/34, PDP-11/40, PDP-11/45, PDP-11/50, PDP-11/70 PROCESSOR HANDBOOKS

Contain information on System Architecture, Addressing Modes, Instruction Set, Programming Techniques, Arithmetic Element Operations.

PDP-11 Peripherals and Interfacing Handbook

Contains programming, information on all PDP-11 peripheral devices, including: Teletype[1], LA30/LA36 DECwriters, VT05/VT50, VT52 and VT55 video displays[2], disks, DECtape, magtape and paper tape.

MUMPS-11 Documentation

In addition to this manual, the following MUMPS documents are relevant:

MUMPS-11 Language Reference Manual

Language Fundamentals, Syntax, Operator Commands and Functions

MUMPS-11 Programmer's Guide

System Overview, I/O Devices, Using the Terminal, Programming Techniques, Library Utility Programs

PDP-11 MUMPS Programmer's Reference Card

Summary of Syntax Operators, System Variables, Commands, Functions, Error Messages and System Utility Programs

Introduction To MUMPS-11 Language

Step-by-step tutorial approach to the MUMPS-11 Language

MUMPS-11 Release Notes

An overview of the MUMPS-11 software package containing useful hints and tips for MUMPS-11 users.

[1] Teletype is a registered trademark of the Teletype Corporation.

[2] Additional information on the VT05, VT50, VT52 and VT55 video terminals is provided in DEC-00-H4AC-D, DEC-00-OVT5A-A-D, EK-VT5X-OP-001 and EK-VT55A-TM-002, respectively.

The following symbols are used throughout this manual:

Symbol	Definition
↵	Used as line or command terminator - use either Carriage RETURN or ALTMODE.
␣	Used to place spaces where desired.
CTRL	Used with special system control characters. Depress CTRL key while striking designated character.
UNDERLINING	Used with examples of user-typed terminal input.
↑U or ^U	Used as echo for CTRL/U keyboard command.
\	Used as echo for RUBOUT keyboard command.
BREAK	Used by operating BREAK key.
‖	Used to indicate that one of the enclosed items must be chosen.

ACKNOWLEDGMENT

MUMPS-11 is an integrated system comprised of an interactive programming language, a data management facility and a multi-user time-sharing executive. It was developed by Digital Equipment Corporation for the PDP-11 series of computers. Meditech Corporation contributed to the original development of MUMPS-11, and Interpretive Data Systems, Inc. assisted in the Version 4A developments.

The language is a dialect of MUMPS (Massachusetts General Hospital Utility Multi-Programming System) which was developed at the Laboratory of Computer Science at Massachusetts General Hospital and is supported by Grant HS-00240 from the National Center for Health Services Research and Development.

CHAPTER 1

INTRODUCTION

This manual provides reference information for system managers and operators using the MUMPS-11 time-sharing system.

The first three chapters are intended to help new users build and configure a MUMPS-11 system for the first time. These chapters contain procedures for building, loading, system generation and initialization. Each operation is described in a simple step-by-step fashion so that a MUMPS system can easily be brought into operation. As the user gains familiarity with the system through actual operation, more sophisticated operating techniques can be employed. Chapters 4, 5 and 6 contain reference information needed by the MUMPS system manager/operator to maintain the MUMPS system.

This chapter summarizes the contents of the manual as a whole and lists associated documents, system software, and materials required to install a MUMPS-11 system.

Chapter 2 describes how to run the MUMPS System Builder program in order to select and link together the various DEC- and user-supplied modular software components of the MUMPS-11 system into a basic system that is ready for system generation.

Chapter 3 describes the system-generation process including: how to load the System Generator program (SYSGEN), preoperational considerations, and detailed descriptions of each of SYSGEN's questions and the appropriate responses. Following the procedures in this chapter, the user specifies the particular hardware environment desired, and the MUMPS-11 system is tailored to those specifications.

Chapter 4 contains reference data on the use of the MUMPS-11 console terminal and the PDP-11 Control Console Switch Register switches. Also provided is information concerning loading and starting the system, creating system backup, and system shutdown.

Chapter 5 describes MUMPS-11 system error detection features and various recovery procedures for both hardware and software errors.

Chapter 6 describes how to use the MUMPS-11 System Utility Programs. (Library Utility Programs are described in the MUMPS-11 Programmer's Guide.) These programs provide the system manager/operator with the tools necessary to maintain the system on a day-to-day basis.

1.1 HOW THE MUMPS-11 SYSTEM IS SUPPLIED

The MUMPS-11 software kit is supplied on magtape or RK05 and RK06 cartridges. It contains the MUMPS System Builder and program object modules, MUMPS SYSGEN, the System and Library Utilities, and the MUMPS Backup and Utility System.

1.1.1 MUMPS-11 System Builder

The MUMPS-11 System Builder contains the System Build operating system and System Build programs that are necessary to generate a basic MUMPS system, and the program object modules for the MUMPS-11 operating system.

In addition to the tape containing the MUMPS SYSGEN and Utility Programs, and the tape containing the MUMPS Backup and Utility System, there are four versions of System Build tapes for use with the different disk types. Depending on which disk type is in your configuration, you can choose one of the four System Build tapes (see Table 1-1). The tape you choose will be used to create your system disk.

Table 1-1
MUMPS-11 Tape Identification

Tape Name	7-Track Magtape Kit	9-Track Magtape Kit
MUMPS System Builder (RK05 image)	X	X
MUMPS System Builder (RK06 image)		X X
MUMPS System Builder (RP02, 03 image)		X X
MUMPS System Builder (RP04, 05, 06 image)		X X
MUMPS SYSGEN and Utility Programs	X	X
MUMPS Backup and Utility System	X	X

1.1.2 MUMPS SYSGEN, System and Library Utility Programs

The MUMPS-11 SYSGEN and Utility Software consists of the programs listed below. Associated subprogram names are shown in parentheses.

System Utility Programs:

BCS	Broadcast
CTK	System Caretaker (CKO)
DAT	Date Set
DBT	Disk Block Tally (TAL, KIL)
DMP	Disk Block Dump (DCB)

%GP	Global Place (%MU)
KTR	Caretaker Reporter
MSP	Modify System Parameters (MSU, %MU, MSD, MBP, MUX, KIL, MMD)
RKC	Copy RK05 Disk Packs
RSJ	Restore Job
RST	Restore Devices
SS	System Status
SDP	Sequential Disk Processor Space Allocator (SD1, SD2, SD3, KIL, TAL)
SIF	System Information (SS)
SSD	System Shutdown
STU	System Startup (%T, %D, CTK)
SYSGEN	MUMPS System Generator (SG1 through SG8, MSU, %MU, MSD, MBP, MUX, MMD)
TIM	Time Set
TP1 through TP8	System Test Package

Library Utility Programs and Subroutines:

%D	Date Subroutine
%FD	Fast Program Directory Lister
%GD	Global Directory
%GL	Global Lister
%GR	Global File Restore
%GS	Global File Save
%GT	Global Trace
%GU	Global Utilization
%GV	Global View
%IO	I/O Device Assignment Subroutine
%IU	IN USE Message
%OD	Octal/Decimal Converter
%OP	User/Operator Communicator
%PD	Program Directory Lister
%PL	Program Load

%PS	Program Save
%T	Time Subroutine

In addition, a program editor resides in ↑%.

1.1.3 MUMPS Backup and Utility System

The MUMPS Backup and Utility System permits the creation of physical backup on any user-specified mass storage media supported by MUMPS. The MUMPS system manager may backup or save the contents of any MUMPS-created disk, tape, global file directory or individual global file. For further details, refer to Appendix I.

1.1.4 Documentation

The following subset of the documentation described in Section 1.1 is provided as part of the MUMPS-11 software package:

MUMPS-11 Language Reference Manual
MUMPS-11 Programmer's Guide
MUMPS-11 Operator's Guide
Introduction to MUMPS-11 Language
MUMPS-11 Programmer's Reference Card
MUMPS-11 Release Notes

1.2 REQUIRED MATERIALS

In order to create backup copies of a system, a number of certified DECtapes, magtapes and disk packs may be needed. However, no specific recommendations can be made as to the quantity of these materials. Such a determination depends directly upon both the user's hardware configuration and the specific backup procedures that are to be utilized. For any particular hardware configuration, several different backup procedures can be employed.

CHAPTER 2

BUILDING THE BASIC SYSTEM

2.1 INTRODUCTION

MUMPS-11 software is constructed from a number of modules that are supplied to the user as binary files (object modules). Some of these modules, like the interpreter, I/O Processor, time-sharing monitor, and others, must always be part of the system. Other modules, which implement optional system hardware/software features, such as floating point operations, additional I/O devices and terminals or sequential disk processing, can be included or omitted as desired. Special user-constructed modules for non-standard devices can also be included. This lets each user tailor the system to his exact needs for both hardware and software. This is a two-step process. The first step consists of selecting a group of modules that suit the user's hardware configuration, then linking those modules to form the Basic System. These procedures are described in this chapter. The Basic System is a special minimum capability system, the sole purpose of which is to allow operation of the MUMPS System Generator Program (SYSGEN). This is the second step in the MUMPS configuration process and is described in Chapter 3.

2.2 SUMMARY OF OPERATIONS

The procedures for building the Basic System are automated by a special program called the MUMPS System Builder, or simply SYSBLD. SYSBLD runs an interactive question/answer session with the system manager to obtain the required information to build the system. SYSBLD then uses the MUMPS SYSBLD programs, PIP, IND, and TKB to create the Basic System[1]. See Figure 2-1 for a typical SYSBLD session. When SYSBLD completes its operation, the Basic System is ready to be loaded and started, in preparation for system generation (Chapter 3). A loadable copy of the Basic System (on either magtape or disk cartridge) can be used as a backup to the system. Two copies of the system load map are output to aid the system manager in using the system and in diagnosing error conditions. One copy is placed on the system backup media, and the other is output to either the console terminal or the line printer. The principal steps in building the Basic System with magtape distribution are summarized in Table 2-1.

[1] This is a special configuration of the MUMPS System Build (MSB) Operating System to be used only for building a MUMPS-11 system.

Table 2-1
Summary of System Build Steps-Magtape Distribution

1. Mount the DSC (Disk Save and Compress copy utility) magtape on drive 0.

2. Boot the device and DSC program will come up.

3. Mount the MSB (MUMPS System Build) tape on drive 0.

4. Enter the command to copy the system image onto the system disk.

5. Boot the disk to bring up the System Builder.

6. Mount the MUMPS object tape on Drive 0, and answer the questions on the console.

7. Type @BUILD to initiate the System Builder question/answer session.

2.3 PREOPERATIONAL INFORMATION

This section describes the information required by the system manager prior to building the system.

2.3.1 General

The system manager should be familiar with the PDP-11 hardware configuration to be used. Specifically, the following information should be on hand:

- Processor type (PDP11/10, 11/20, 11/34, 11/40, 11/45, 11/50 or 11/70)
- Availability of hardware arithmetic unit(s) (EAE, EIS, or FPP)
- Memory size
- Availability and type of: magtape, data sets, DH-11, DZ-11 Multiplexers, and CPU-to-CPU devices (via DL-11 controllers)
- Whether the Sequential Disk Processor (described in the MUMPS-11 Programmer's Guide) is desired
- Whether the $M Function is desired

2.3.2 Reserved Devices

The system manager can incorporate up to four standard or non-standard I/O devices into the system as Reserved Devices. In addition to interfacing the device and its controller to the PDP-11 processor, the programmer must write a device driver program to operate the device. Design criteria and instructions for writing these device drivers are provided in Appendix I, of the MUMPS-11 Programmer's Guide. The drivers must be available at system-build time in object module (binary file) form. System managers or programmers who are to write their own drivers for reserved devices, must have a good working knowledge of the System Builder software, including the PDP-11 assembler and the System Utility Programs.

2.4 ERROR CONDITIONS

If the user discovers a typing error in a line before the terminating Carriage RETURN is typed, it can be corrected in one of two ways. One method is to use the RUBOUT key to delete single characters, beginning with the last character typed. Each time RUBOUT is typed, a character is deleted, progressing from right to left. The other method is to use the CTRL/U function to delete the entire line (press CTRL key and type a U). Errors discovered after the Carriage RETURN is typed will simply cause the question to be reissued. Logical errors, such as responding with a NO answer when a YES is desired, can be corrected only by initiating a new SYSBLD session. These are the only types of errors a user should encounter.

2.5 SYSTEM STARTUP WITH BOOTSTRAP ROMS

The step-by-step startup procedure for using the various bootstrap ROMS is described in the following sections.

2.5.1 Using BM792-YB Bootstrap ROM

The following procedures are used in starting up the BM792-YB bootstrap ROM:

1. Depress HALT.

2. Set the console switches to 173100.

3. Depress LOAD ADDRESS.

4. Enter the address of the load device in the console switches:

RK11 Disk	177406
RK611 Disk	177462
RP11 Disk	176716
TC11 DECtape	177344

5. Lift up the HALT key.

6. Press START.

2.5.2 Using MR11-DB Bootstrap ROM

The following procedures are used in starting up the MR11-DB bootstrap ROM:

1. Depress HALT.

2. Enter the address of the load device in the console switches:

RK611 Disk	173100
RK11 Disk	173110
TC11 DECtAPE	173120
TM11 Magtape	173136
RP11 Disk	173154

3. Depress LOAD ADDRESS.

4. Lift up the HALT key.

5. Press START.

2.5.3 Using BM873-YA Bootstrap ROM

The following procedures are used in starting up the BM873-YA bootstrap ROM:

1. Depress HALT.

2. Enter the address of the load device in the console switches:

RK611/RH70 Disk	173000
RK11/RH70 Disk	173010
TC11 DECtape	173030
TM11 Magtape	173050
RP11 Disk	173100
KL11/DL11	173210
TA11 Cassette	173230
PC11 Papertape	173312

3. Depress LOAD ADDRESS.

4. Lift up the HALT key.

5. Press START.

2.5.4 Using BM873-YB Bootstrap ROM

The following procedures are used in starting up the BM873-YB bootstrap ROM:

1. Depress HALT.

2. Enter the address of the load device in the console switches:

RH11/RS04 Disk (unit 0)	173000
RH11/RS04 Disk (unit in switch register)	173002
RK11 Disk (unit 0)	173030
RK11 Disk (unit in switch register)	173032

TC11 DECtape (unit 0)	173070
TM11 Magtape (unit 0)	173110
RK611 Disk	173136
RH11/RH70/TM02/TU16 Magtape (unit 0)	173150
RH11/RH70 Device Combination (unit zero)	173230
RH11/RH70 Device Combination (unit in switch register)	173232
RH11/RH70/RP04 Disk (unit 0)	173320
RH11/RH70/RP04 Disk (unit in switch register)	173322
RP11/RP02/RP03 Disk (unit 0)	173350
RP11/RP02/RP03 Disk (unit in switch register)	173352
KL11/DL11 Console Terminal Reader	173510
PC11 Papertape Reader	173620

3. Depress LOAD ADDRESS.

   For units other than zero, enter unit number in switch register.

4. Lift up the HALT key.

5. Press START.

2.5.5 Using M9301-YA or M9301-YB Bootstrap ROM (PDP-11/34)

The following procedures are used in starting up the M9301-YA or M9301-YB bootstrap ROM.

1. Press the console buttons marked CNTRL and HLT/SS simultaneously.

2. Press the console buttons marked CNTRL and BOOT simultaneously.

3. In response to the "$" prompt on the console terminal, enter one of the following:

DB - RH11 Disk	M9301-YB only
DK - RK11 Disk	
DP - RP11 (RP02/03) Disk	
DS - RS04 Disk	M9301-YB only
DT - TC11 DECtape	
MM - TM02/TU16 Magtape	M9301-YB only
MT - TM11/TU10 Magtape	
PR - PC11/PR11 Paper Tape Reader	
TT - KL11/DL11 Console Terminal Reader	

   The two alphabetic characters may be followed by an optional single octal digit for bootstrapping from units other than zero. Such alphabetic characters (and octal digit) must be terminated by a Carriage RETURN.

2.6 DETAILED DESCRIPTION OF OPERATION - MAGTAPE DISTRIBUTION

This section provides detailed procedures for building the Basic System for MSB software and MUMPS modules that have been supplied on magtape. (See Section 2.7 for disk pack distribution.)

NOTE

> All user keyboard responses are terminated by Carriage RETURN. In the examples that accompany the text, user responses are underlined.

2.6.1 Loading Procedures - MUMPS-11 System Builder

The following paragraphs describe the procedures to be followed when loading and starting the MUMPS System Builder (MSB) from magtape. As described in Chapter 1, the MSB software is supplied either on magtape (7- or 9-track), or RK05 or RK06 cartridge disk.

NOTE

> Prior to building your system, be sure to format any fresh disk packs with MBU (see Appendix I).

1. Mount the DSC (Disk Save and Compress) utility tape on the magtape drive as follows:

   A. Make sure the write enable ring is removed from the tape.

   B. Mount the tape on drive number 0.

   C. Position the tape at LOAD POINT (see Appendix G).

   D. Set the ON LINE/OFF LINE switch to ON LINE.

2. Boot the magtape by activating the hardware read-only memory (ROM) bootstrap as described in Section 2.5.

   (The copy utility DSC will now be loaded and started.)

3. Dismount the DSC tape and mount the MSB (MUMPS System Builder) tape on drive 0.

4. Copy the MSB system from magtape onto a formatted disk as shown below. If the disk has not been formatted, see Appendix I.

   A. Load a scratch disk to be used for MSB on disk drive 0.

   B. If the MSB tape is mounted on a TU10 or TS03 magtape drive, type the following command to DSC:

   ```
   DSC>MT:/VEC=224↵
   ```

   C. If the scratch disk is mounted on an RP02 or RP03 disk drive, type the following command to DSC:

   ```
   DSC>DP:/VEC=254↵
   ```

NOTE

> These two command lines must be typed (if applicable to your system configuration) due to a conflict between the vectors for these devices and for other possible system devices.

D. Type the following command line to DSC in order to transfer the proper version of MSB for your disk from the magtape.

```
                    DSC>xx:=mm:xxSYS↵

      where xx= DK for RK05
                DM for RK06
                DP for RP02,03
                DB for RP04,05,06

            mm= MT for TU10 or TS03 magtape drives
                MM for TJU16 magtape drives
```

(The tape will now be searched for the correct MSB file which will be copied to the disk. When this is completed, DSC will respond with "DSC>".)

5. Halt the system and remove the MSB tape.

6. Boot the disk in the manner described in Section 2.5. The MUMPS System Builder should announce itself as follows:

```
          ** M S B **       V4B
```

7. After several intermediate lines of printing, the user will be asked to enter the time and date with the following message:

```
>* PLEASE ENTER TIME AND DATE (HH:MM MM/DD/YY) [S]
```

The user should respond with the correct time and date.

Example:

```
                    20:55 2/6/77↵
```

8. After entering this information, mount the MUMPS object tape on drive 0.

9. MSB will request the user to enter the type of magtape device containing the MUMPS object tape. If this is a TM-11/TU10-TS03 type drive, respond with MT. If this is a TJU16/TU16 type drive, respond with MM.

10. All information from the MUMPS object tape will be copied onto the disk.

11. Once this process has completed, MSB will display the various options available to the user. At this point, the MUMPS object tape should be dismounted.

12. If the user has any user device drivers for MUMPS, they should be transferred to the MSB disk. These device drivers must be in an RSX-11M V3 object module format. Section 2.6.2 describes this procedure.

13. In order to start the build procedure for MUMPS, the user should type "@BUILD".

(Figure 2-2 shows a SYSBLD startup with magtape distribution.)

NOTE

> If MSB does not print its identification and the processor halts, an error occurred in the copy operation. Retry the procedure beginning at Step 1. If the error persists, it will be necessary to obtain a new tape.

2.6.2 User-Created Object Modules

Once MSB is loaded, any object modules of drivers for reserved I/O devices must be loaded onto the system disk. This is accomplished, using PIP as follows:

1. Mount the tape (after removing its write enable ring) containing the device driver object modules, on the appropriate drive.

2. Select a drive unit number from 0-7.

3. Set drive control switches as follows:

   A. Magtape Input

      a. Position tape at LOAD POINT (Appendix G).

      b. Set OFF LINE/ON LINE switch to ON LINE.

   B. DECtape Input

      a. Set the WRITE ENABLE/WRITE LOCK switch to WRITE LOCK.

      b. Set the LOCAL/REMOTE switch to REMOTE.

4. Transfer each module from the tape to the system disk by using one of the following command strings, as applicable:

   Magtape Input

   ```
   >PIP SY:<MTn:filnam.OBJ↵   TU10 or TS03
   >PIP SY:<MMn:filnam.OBJ↵   TJU16
   ```

   DECtape Input

   ```
   >
   >PIP SY:<DTn:filnam.OBJ↵
   ```

   where:

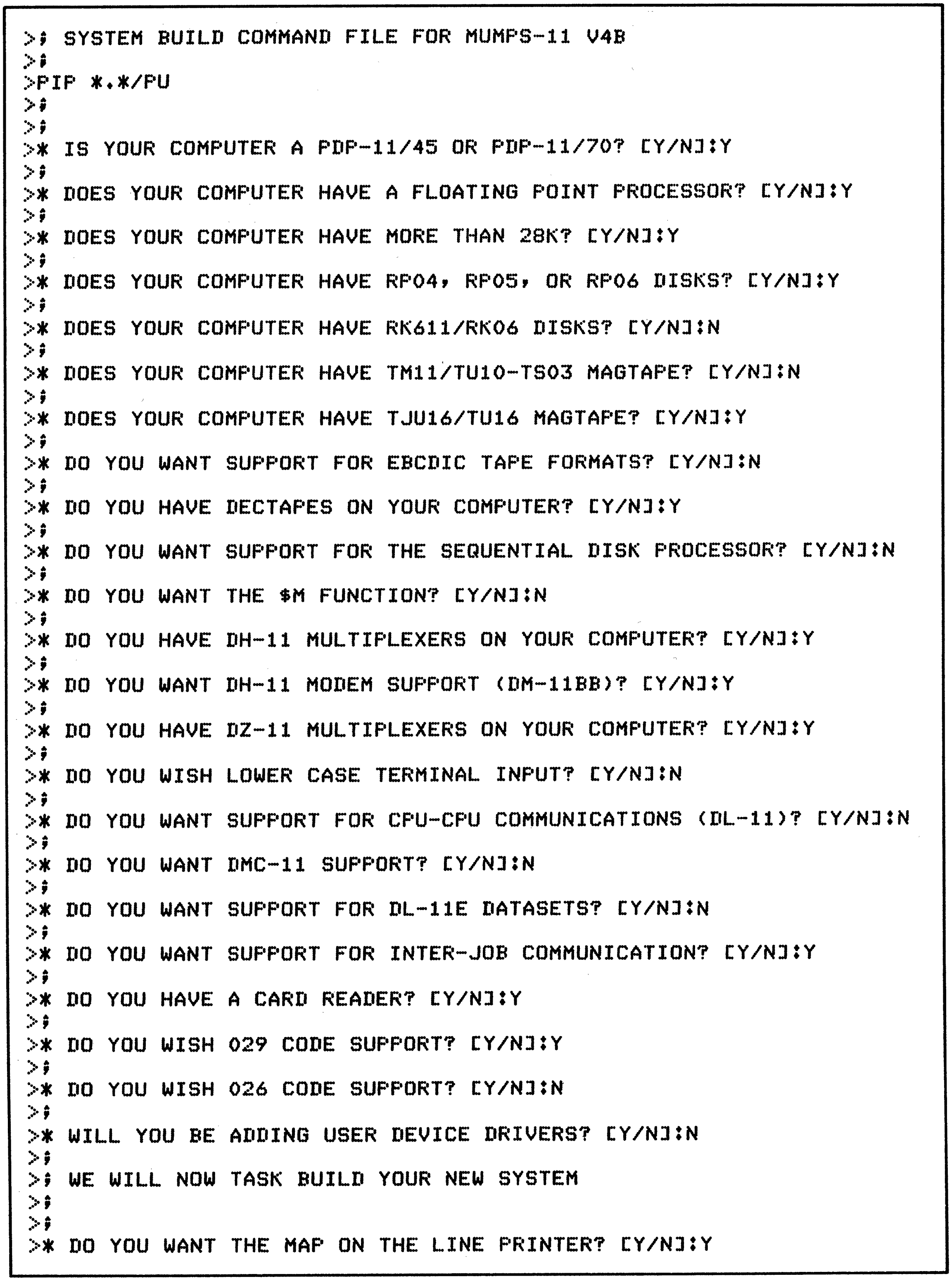
   ```
        n = unit number
   filnam = file name of object module
   ```

When all modules are transferred, terminate PIP operations by typing CTRL/Z, and carriage RETURN.

```
>↑Z↵
```

2.6.3 Answering SYSBLD Questions

Section 2.8 describes each SYSBLD question along with the appropriate responses. Answers to SYSBLD questions are either Y (for YES) or N (for NO) followed by a Carriage RETURN.



```
>; SYSTEM BUILD COMMAND FILE FOR MUMPS-11 V4B
>;
>PIP *.*/PU
>;
>;
>* IS YOUR COMPUTER A PDP-11/45 OR PDP-11/70? [Y/N]:Y
>;
>* DOES YOUR COMPUTER HAVE A FLOATING POINT PROCESSOR? [Y/N]:Y
>;
>* DOES YOUR COMPUTER HAVE MORE THAN 28K? [Y/N]:Y
>;
>* DOES YOUR COMPUTER HAVE RP04, RP05, OR RP06 DISKS? [Y/N]:Y
>;
>* DOES YOUR COMPUTER HAVE RK611/RK06 DISKS? [Y/N]:N
>;
>* DOES YOUR COMPUTER HAVE TM11/TU10-TS03 MAGTAPE? [Y/N]:N
>;
>* DOES YOUR COMPUTER HAVE TJU16/TU16 MAGTAPE? [Y/N]:Y
>;
>* DO YOU WANT SUPPORT FOR EBCDIC TAPE FORMATS? [Y/N]:N
>;
>* DO YOU HAVE DECTAPES ON YOUR COMPUTER? [Y/N]:Y
>;
>* DO YOU WANT SUPPORT FOR THE SEQUENTIAL DISK PROCESSOR? [Y/N]:N
>;
>* DO YOU WANT THE $M FUNCTION? [Y/N]:N
>;
>* DO YOU HAVE DH-11 MULTIPLEXERS ON YOUR COMPUTER? [Y/N]:Y
>;
>* DO YOU WANT DH-11 MODEM SUPPORT (DM-11BB)? [Y/N]:Y
>;
>* DO YOU HAVE DZ-11 MULTIPLEXERS ON YOUR COMPUTER? [Y/N]:Y
>;
>* DO YOU WISH LOWER CASE TERMINAL INPUT? [Y/N]:N
>;
>* DO YOU WANT SUPPORT FOR CPU-CPU COMMUNICATIONS (DL-11)? [Y/N]:N
>;
>* DO YOU WANT DMC-11 SUPPORT? [Y/N]:N
>;
>* DO YOU WANT SUPPORT FOR DL-11E DATASETS? [Y/N]:N
>;
>* DO YOU WANT SUPPORT FOR INTER-JOB COMMUNICATION? [Y/N]:Y
>;
>* DO YOU HAVE A CARD READER? [Y/N]:Y
>;
>* DO YOU WISH 029 CODE SUPPORT? [Y/N]:Y
>;
>* DO YOU WISH 026 CODE SUPPORT? [Y/N]:N
>;
>* WILL YOU BE ADDING USER DEVICE DRIVERS? [Y/N]:N
>;
>; WE WILL NOW TASK BUILD YOUR NEW SYSTEM
>;
>;
>* DO YOU WANT THE MAP ON THE LINE PRINTER? [Y/N]:Y
```

Figure 2-1 Typical System Build Session

2.6.4 Restarting the Basic System from DECtape or Magtape

When MSB completes its operation, the MUMPS Basic System resides on DECtape or magtape. Since the user may not wish to proceed with MUMPS system generation until some later time, the following paragraphs describe how to restart the Basic System using the backup tape created by MSB.

1. Mount the MSB-created backup tape on the appropriate drive (DECtape or magtape).

2. Set the drive control switches as follows:

   A. DECtape Drives

      a. Set the unit number switch to 0.

      b. Set the LOCAL/REMOTE switch to REMOTE.

      c. Set the WRITE LOCK/WRITE ENABLE switch to WRITE LOCK.

   B. Magtape Drives

      a. Set the unit number to 0.

      b. Remove the write enable ring from the tape reel.

      c. Set the OFF LINE/ON LINE switch to ON LINE.

3. For all PDP-11's (except the PDP-11/34) boot DECtape or magtape by activating the hardware read-only memory (ROM) bootstap via the SWITCH REGISTER switches, as described below:

   A. Set the console ENABLE/HALT switch to HALT.

   B. Set the console SWITCH REGISTER switches to the hardware bootstrap ROM address for DECtape or magtape (see Section 2.5).

   C. Press the LOAD ADRS switch.

   D. Set the ENABLE/HALT switch to ENABLE.

   E. Press the START switch.

4. For the PDP-11/34, boot DECtape or magtape by activating the read-only memory (ROM) bootstrap via the console buttons, as described below:

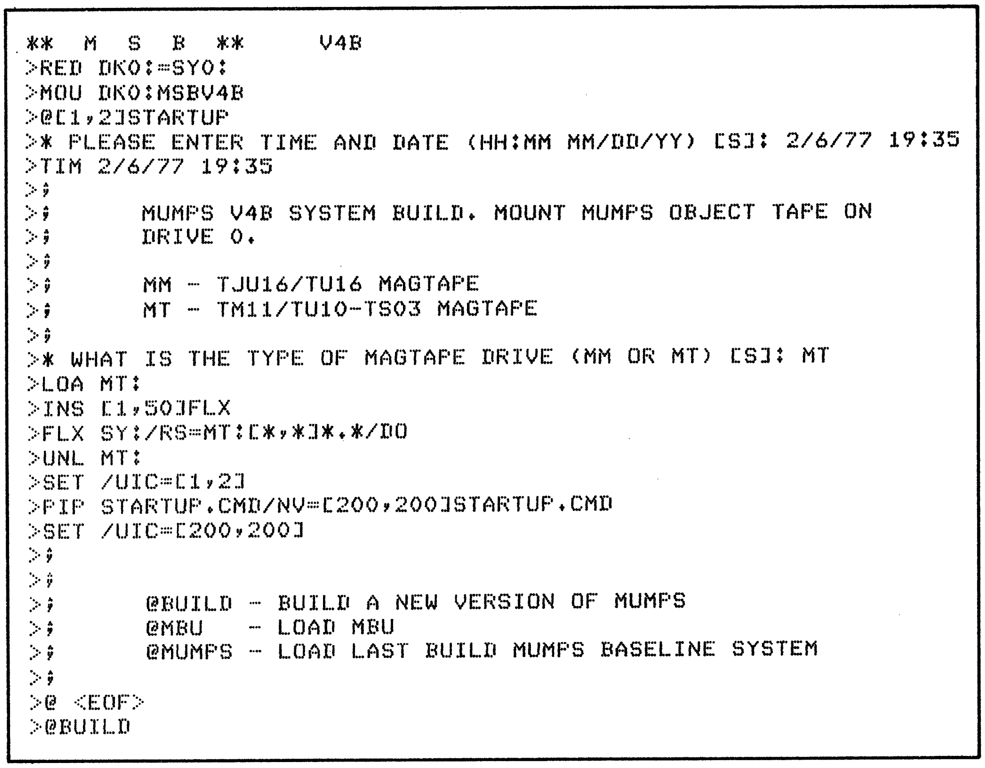
   A. Press the console buttons marked CNTRL and HLT simultaneously.

   B. Press the console buttons marked CNTRL and BOOT simultaneously.

   C. When the system prints a $, respond by typing either 'DT' for DECtape or 'MT' for magtape.

The tape should now spin, indicating that MUMPS is being loaded. When the tape halts, wait two minutes, LOWER ALL SWITCHES and then sign onto MUMPS by hitting CTRL C (press CTRL and C on the terminal simultaneously).

The MUMPS image is transferred from the tape to memory, where it is automatically started. Thereafter, MUMPS waits for the user to log-in, in preparation for system generation (see Chapter 3). If MUMPS must be halted during SYSGEN for any reason, it can be restarted using the bootstrap loader for magtape or DECtape, as described in Section 2.6.4.



```
** M S B **      V4B
>RED DK0:=SY0:
>MOU DK0:MSBV4B
>@[1,2]STARTUP
>* PLEASE ENTER TIME AND DATE (HH:MM MM/DD/YY) [S]: 2/6/77 19:35
>TIM 2/6/77 19:35
>;
>;       MUMPS V4B SYSTEM BUILD. MOUNT MUMPS OBJECT TAPE ON
>;       DRIVE 0.
>;
>;       MM - TJU16/TU16 MAGTAPE
>;       MT - TM11/TU10-TS03 MAGTAPE
>;
>* WHAT IS THE TYPE OF MAGTAPE DRIVE (MM OR MT) [S]: MT
>LOA MT:
>INS [1,50]FLX
>FLX SY:/RS=MT:[*,*]*.*/DO
>UNL MT:
>SET /UIC=[1,2]
>PIP STARTUP.CMD/NV=[200,200]STARTUP.CMD
>SET /UIC=[200,200]
>;
>;
>;       @BUILD - BUILD A NEW VERSION OF MUMPS
>;       @MBU   - LOAD MBU
>;       @MUMPS - LOAD LAST BUILD MUMPS BASELINE SYSTEM
>;
>@ <EOF>
>@BUILD
```

Figure 2-2 SYSBLD Startup with Magtape Distribution

2.7 DETAILED DESCRIPTION OF OPERATION - DISK PACK DISTRIBUTION

This section provides detailed procedures for building the Basic System for MSB software and MUMPS modules which have been supplied on disk (RK05 or RK06). (See Section 2.6 for magtape distribution.)

NOTE

> All user keyboard responses are terminated by Carriage RETURN. In the examples that accompany the text, user responses are underlined.

2.7.1 Preoperational Information

The disk pack distribution media consists of two RK05 or RK06 cartridges. One of the cartridges contains the MUMPS System Builder (MSB) operating system, the MUMPS object modules, and the MUMPS Backup and Utility System. The other cartridge contains a runnable pre-configured MUMPS system with minimal hardware support, the MUMPS SYSGEN programs, the System Library, and the Editor.

The System Manager uses the MSB cartridge to create a new basic MUMPS system, and then "merges" the new basic MUMPS system with the MUMPS cartridge during SYSGEN to produce a customized MUMPS System. This process can be repeated as frequently as desired.

CAUTION

> Do not proceed without making a copy of both disk cartridges (refer to Appendix I for instructions).

2.7.2 Loading Procedures - MUMPS-11 System Builder

The following paragraphs describe the procedures to be followed when loading and starting the MUMPS System Builder (MSB) from disk.

1. Load the disk containing the MSB software on unit 0 (write enabled).

2. For all PDP-11's, except the PDP-11/34, load and start MSB by activating the hardware read-only memory (ROM) bootstrap via the SWITCH REGISTER switches, as described below.

   A. Set the console ENABLE/HALT switch to HALT.

   B. Set the console SWITCH REGISTER switches to the appropriate address for RK11 loading (see Section 2.5).

   C. Press the LOAD ADRS switch.

   D. Set the ENABLE/HALT switch to ENABLE.

   E. Press the START switch.

3. For the PDP-11/34, load and start MSB from the system disk by using the ROM bootstrap via the console buttons, as described below:[1]

   A. Press the console buttons marked CNTRL and HLT simultaneously.

   B. Press the console buttons marked CNTRL and BOOT simultaneously.

   C. When the system prints a $, respond by typing 'DK', to boot the disk.

At this point, the MUMPS System Builder starts and identifies itself

[1] For information on the 11/34 console, and instruction set, refer to the KY11-LB Programmer's Console Maintenance Manual, EK-KY11L-MM-001, and the PDP 11/34 Processor Handbook, EB 05710 76.

at the console terminal by printing its name and version number:

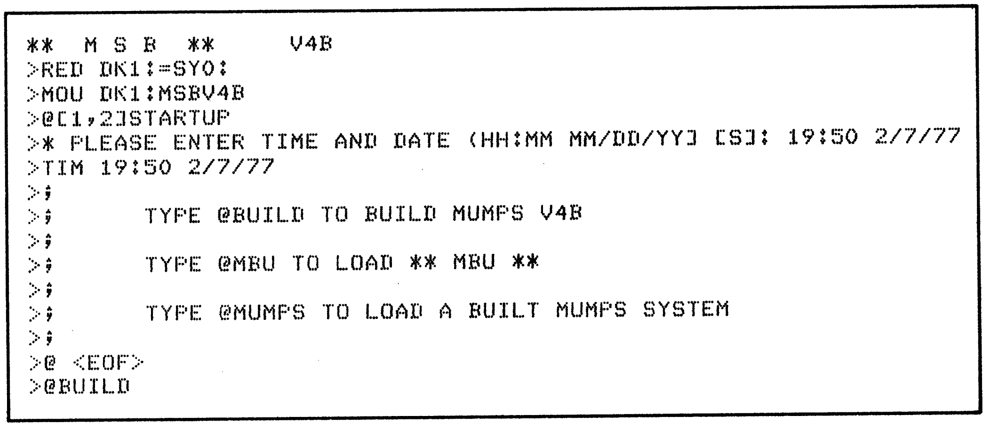
```
** M S B **     V4B
```

Respond to the time and date message in the format specified. Instructions will be displayed indicating the various options available to the user. In order to start the system build procedure, type @BUILD. Refer to Figure 2-3 for a SYSBLD Startup with RK05 distribution.

2.7.3 Running SYSBLD

Once @BUILD has been typed, SYSBLD begins its interactive question/answer session with the user. Section 2.8 describes each SYSBLD question with the appropriate responses. Once these questions have been answered, the interactive portion of SYSBLD's operation is complete and the actual system building process begins. SYSBLD calls the various MSB system programs required to perform this task. SYSBLD then prints out a message stating that the user may now boot MUMPS (for all RK disks, type: BOO MUMPS).

If the user wishes to complete the system generation at this time, the MUMPS cartridge should be mounted on drive 0, and the user should sign onto MUMPS with Control C and CALL the program SG1 (see Section 3.7). The user should ensure that all console switches are lowered before signing onto MUMPS.



```
**  M S B  **     V4B
>RED DK1:=SY0:
>MOU DK1:MSBV4B
>@[1,2]STARTUP
>* PLEASE ENTER TIME AND DATE (HH:MM MM/DD/YY] [S]: 19:50 2/7/77
>TIM 19:50 2/7/77
>;
>;       TYPE @BUILD TO BUILD MUMPS V4B
>;
>;       TYPE @MBU TO LOAD ** MBU **
>;
>;       TYPE @MUMPS TO LOAD A BUILT MUMPS SYSTEM
>;
>@ <EOF>
>@BUILD
```

Figure 2-3 SYSBLD Startup with RK05 Distribution

2.7.4 Restarting the Basic System From Disk

When MSB completes its operation, the MUMPS Basic System (MSB) resides on disk. Since the user may not wish to continue with system generation until a later time, the following paragraphs describe how to restart the Basic System using the MSB disk:

1. For all PDP-11's except the PDP-11/34, boot the MSB disk by activating the hardware read-only memory (ROM) bootstrap, as follows:

A. Set the console ENABLE/HALT switch to HALT.

B. Mount and load the MSB disk of the disk drive.

C. Set the console SWITCH REGISTER switches to the appropriate address for disk (see Section 2.5).

D. Press the LOAD ADRS switch.

E. Set the ENABLE/HALT switch to ENABLE.

F. Press the START switch.

2. For the PDP-11/34, boot the MSB disk by activating the hardware read-only memory (ROM) bootstrap, as follows:

   A. Press the console buttons marked CNTRL and HLT/SS simultaneously.

   B. Press the console buttons marked CNTRL and BOOT simultaneously.

   C. When the system prints a $, respond by typing 'DK' or 'DM' for disk.

3. MSB now resides in core. Boot MUMPS by typing:

```
@MUMPS
```

After the basic MUMPS system has been loaded, the MSB disk should be removed and the MUMPS SYSGEN program should be started as described in Section 3.7.

2.8 SYSBLD QUESTIONS

The SYSBLD program asks the user a series of questions in order to determine the configuration of the Basic System. These questions are described below. Answers to SYSBLD questions are either Y (for YES) or N (for NO).

```
IS YOUR COMPUTER A PDP 11/45 OR PDP-11/70?  [Y/N]:
```

If the MUMPS-11 system is being built to run on a PDP 11/45 or 11/70 processor, type Y; otherwise, type N. If the answer is Y, SYSBLD skips the next question.

```
DOES YOUR COMPUTER HAVE A FLOATING POINT PROCESSOR?  [Y/N]:
```

If this system has a floating-point processor, type Y; otherwise, type N.

```
DOES YOUR COMPUTER HAVE MORE THAN 28K? [Y/N]:
```

If the processor has more than 28K words of usable memory, (i.e., the system has the memory management option), type Y; otherwise, type N.

DOES YOUR COMPUTER HAVE RP04, RP05, OR RP06 DISKS? [Y/N]:

If this system has RP04, RP05 or RP06 disks, type Y; otherwise, type N. If the system has an RP04, RP05, or RP06 disk, but does not have EIS, it is an invalid configuration under MUMPS-11. Therefore, SYSBLD ceases its system build and outputs the following error message:

THE CONFIGURATION YOU HAVE SPECIFIED IS NOT SUPPORTED BY MUMPS-11. USE OF THE RH11/RJP04 DISK SYSTEM REQUIRES THE EIS INSTRUCTION SET. PDP-11/45 AND PDP-11/70 PROCESSORS HAVE THE EIS INSTRUCTIONS BUILT INTO THEM.

DOES YOUR COMPUTER HAVE RK611/RK06 DISKS? [Y/N]:

If this system has RK06 disks, type Y; otherwise, type N.

DOES YOUR COMPUTER HAVE TM11/TU10-TS03 MAGTAPE? [Y/N]:

If this system has the TM11 magnetic tape system, type Y; otherwise, type N. If the answer is Y, the following question is skipped.

DOES YOUR COMPUTER HAVE TJU16/TU16 MAGTAPE? [Y/N]:

If this system has the TJU16 magnetic tape system, type Y; otherwise, type N. If the answer is N, the following question is skipped.

DO YOU WANT SUPPORT FOR EBCDIC TAPE FORMATS? [Y/N]:

This question is only asked if the user selected a magtape option. If N is answered to this question, there is a restriction to the ASCII character set. If Y is answered, the EBCDIC translation table is included, and the user's program can use both the ASCII and the EBCDIC character sets through use of the ASSIGN command.

DO YOU HAVE DECTAPES ON YOUR COMPUTER? [Y/N]:

If the system has the TC11 DECtape option, type Y; otherwise, type N.

DO YOU WANT SUPPORT FOR THE SEQUENTIAL DISK PROCESSOR? [Y/N]:

If the MUMPS-11 system is to use the Sequential Disk Processor (SDP) software option, type Y; otherwise, type N. Refer to the MUMPS-11 Programmer's Guide for a description of the SDP.

DO YOU WANT THE $M FUNCTION? [Y/N]:

The $M function allows standard arithmetic and relational arithmetic operations to be performed on numbers outside the normal range of MUMPS numbers. If you want to use the $M function, type Y;

otherwise, type N. Refer to the MUMPS-11 Language Reference Manual for more information on the $M function.

DO YOU HAVE DH-11 MULTIPLEXERS ON YOUR COMPUTER? [Y/N]:

MUMPS-11 can support up to 65 terminals. When more than 17 terminals are to be used, the additional terminals can be operated through one, two or three DH-11 multiplexers (one DH-11 can operate a maximum of 16 terminals). If DH-11's are to be used, type Y; otherwise, type N.

DO YOU WANT DH-11 MODEM SUPPORT (DM-11BB)? [Y/N]:

This question is asked only if you answered Y to DH-11 multiplexers. This feature provides support for the DM-11 modem control hardware available with the DH-11 interface. If modem support is desired, type Y; otherwise, type N.

DO YOU HAVE DZ-11 MULTIPLEXERS ON YOUR COMPUTER? [Y/N]:

When more than 17 terminals are to be used, the additional terminals can be operated through one to six DZ-11 multiplexers (one DZ-11 can operate a maximum of eight terminals). If DZ-11's are to be used, type Y; otherwise, type N.

DO YOU WISH LOWER CASE TERMINAL INPUT? [Y/N]:

This feature provides for lower-case terminal input. If Y is typed to this question, both upper- and lower-case characters can be sent to a MUMPS program from a terminal. If N is typed to this question, all lower-case characters will be converted to upper-case on all terminals.

DO YOU WANT SUPPORT FOR CPU-CPU COMMUNICATIONS (DL-11)? [Y/N]:

If central processor-to-central-processor communication using a DL-11A controller on a local (hardware) circuit is desired, type Y; otherwise, type N. If N is typed to this question, the following question is asked.

DO YOU WANT DMC-11 SUPPORT? [Y/N]:

The DMC-11 communications controller provides for high speed, reliable transmission of data to local or remote interconnected computers over a serial synchronous link. If support for the DMC-11 is desired, type Y; otherwise, type N.

DO YOU WANT SUPPORT FOR DL-11E DATASETS? [Y/N]:

MUMPS-11 provides full support for Bell type 103A datasets when connected to the system via DL-11E communications controllers. If one or more datasets are to be used, type Y; otherwise, type N.

```
     DO YOU WANT SUPPORT FOR INTER-JOB COMMUNICATION?   [Y/N]:
```

If in-core job communication through a set of pseudo-devices is desired, type Y; otherwise, type N.

```
     DO YOU HAVE A CARD READER?   [Y/N]:
```

If a CR-11 card reader is to be used, type Y; otherwise, type N.

```
     DO YOU WISH 029 CODE SUPPORT?   [Y/N]:
```

This feature allows support for 029 formatted cards. If support is desired for these cards, type Y. If N is typed, 026 card support is assumed.

```
     DO YOU WISH 026 CODE SUPPORT?   [Y/N]:
```

This feature allows support for 026 formatted cards. If Y was typed to the previous 029 code question, this question asks if 026 code is to be supported as well. Type N if 026 code support is desired; otherwise, type N.

```
WILL YOU BE ADDING USER DEVICE DRIVERS?   [Y/N]:
```

MUMPS utilizes up to four user-specified non-standard I/O devices. Each device is operated by a user-created I/O device driver program module. The object module for each device used in the system must be transferred onto the MUMPS System Build device. If user device drivers are to be part of the MUMPS-11 configuration, type Y; otherwise, type N. If Y is answered to this question, SYSBLD prompts the user with the following question.

```
     WHAT IS THE NAME OF YOUR DEVICE MODULE?
```

User enters a six-character name, for example:

```
     XXXDRV↵
```

where XXX is the three-character name for the device module. SYSBLD then prints out:

```
     WE WILL NOW PAUSE WHILE YOU TRANSFER YOUR OBJECT MODULE ONTO THE
     SYSTEM DEVICE.
```

SYSBLD will pause while you transfer your drivers, after which time it will output the message:

```
WE WILL NOW TASK BUILD YOUR NEW SYSTEM.

DO YOU WANT THE MAP ON THE LINE PRINTER?  [Y/N]:
```

If you want the memory allocation map to be printed on the line printer, type Y; if an N is typed, the map is output to your terminal.

After this, a message prints out indicating that certain features of MUMPS were not required for your configuration. This message is normal, and always prints out at this time if any System Build questions have been answered N.

The final question SYSBLD asks is:

```
WHAT IS THE OUTPUT DEVICE?  ("MM","MT","DT", OR "") [S]:
```

This allows you to save the just-built Basic System. The output device can be TJU16 (MM), TU10 (MT), or DECtape (DT). Choosing the final option ("") indicates that you are saving your Basic System to the system disk. [S] indicates that string input is required to this question.

This completes the interactive portion of SYSBLD's operation. At this point, the actual system building process begins. SYSBLD calls the various MSB programs required to perform this task.

CHAPTER 3

SYSTEM GENERATION PROCEDURES

3.1 INTRODUCTION

This chapter describes the procedures to be followed when using the MUMPS-11 System Generator Program (SYSGEN) to tailor a MUMPS-11 system to the user's hardware/software environment. SYSGEN does not create a system, but modifies an existing one: the Basic System built as described in Chapter 2. SYSGEN can be used only with the Basic System.

Modifications to some of the parameters of an existing (SYSGENed) system can be performed using the Modify System Parameters (MSP) program described in Section 6.2.7. However, many modifications require a new SYSGEN. If a user already has an operational MUMPS system and wishes to reconfigure it, the SYSTEM MERGE option of SYSGEN can be used. A SYSTEM MERGE will tailor the basic system to the desired configuration while retaining the current data base. Otherwise, a user must perform a logical save of the data base, perform a normal SYSGEN, and restore the data base.

The MUMPS-11 Basic System is a minimum configuration system consisting of: one terminal, one partition (4K words), one User Class Identifier (UCI) and Standard Programmer Access Code (PAC).

When SYSGEN is loaded, it automatically begins an interactive dialogue with the user that covers all system aspects in a step-by-step manner. Once system generation is complete, the system manager is instructed to save the system on the disk. At this point he should create a backup copy on disk or tape. This can be done with the MUMPS Backup and Utility System (see Appendix I).

Should the system manager wish to perform a subsequent system generation, he must either build a new Basic System (see Sections 2.6.1 or 2.7.2) or load a backup copy of that system (see Appendix I) before running the SYSGEN program.

3.2 ORGANIZATION

3.2.1 Program Structure

SYSGEN, though logically a single program, is physically segmented into the following eight programs named: SG1 through SG8. These separate programs were constructed to allow SYSGEN operation in a 4K word partition. As a part of the system generation process, SYSGEN also uses five subprograms of the Modify System Parameters (MSP) program: MSU (which calls %MU), MSD, MUX, MBP and MMD.

3.2.2 SYSGEN Tape

As described in Chapter 1, SYSGEN is supplied by DEC on magtape (see Section 3.7 for RK05 and RK06 distribution). These tapes are called SYSGEN tapes and contain both the System and Library Utility Programs and the System Test Package. As shown in Figure 3-1, the layout of these tapes is essentially the same. A single exception is that the first record on magtape is a label record. The first eight entries on the tapes contain the first eight program segments of SYSGEN (SG1 - SG8); the remaining program (MSP) and the other system and Library Utility Program reside in the tape's library area.

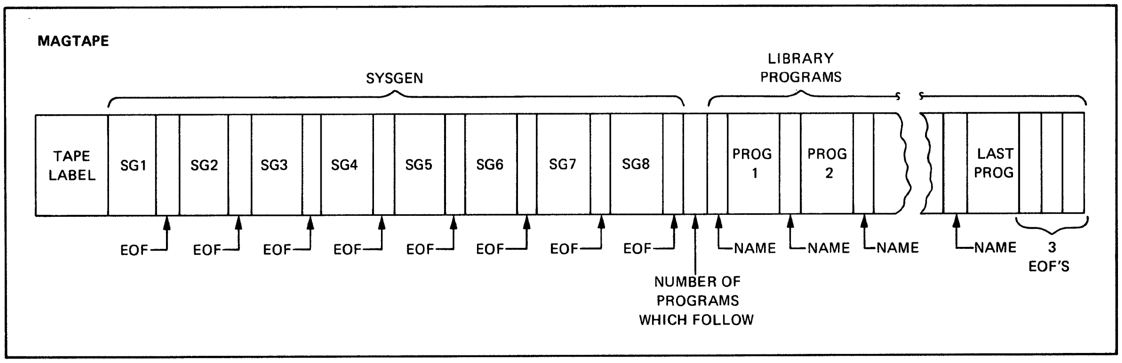


Figure 3-1 SYSGEN Tape Logical Structure

3.3 ANSWERING SYSGEN'S QUESTIONS

User responses are terminated by a Carriage RETURN. Some questions are interdependent, and SYSGEN may ask additional questions as a result of the answer to a previous question. Answers to other questions will cause SYSGEN to type a response to an answer. For example, for each UCI specified, SYSGEN will respond with the physical disk location of the associated directory. If an error is made, SYSGEN will prompt the legitimate answers, then repeat the question.

3.4 OPERATION

In its operation, SYSGEN uses the answers provided by the user to perform a variety of tasks, including:

1. Allocation of memory.
2. Setup of the Job Table, System Queues, Partition Table, Device Table, UCI Table, and the Disk Storage Allocation Table.
3. Allocation of buffer space for terminals, CPU-CPU devices and mass storage devices.
4. Setup of basic system hardware flags and status indicators.
5. Transfer of all DEC-supplied library utility programs to the system UCI.

In addition, SYSGEN builds and initializes a special system global called ↑SYS. This global is maintained by the Caretaker System Utility Program which uses it to log chronologically a variety of system status data. The Caretaker program is described in Section 6.2.2. The ↑SYS global is described in Appendix F.

3.5 PREOPERATIONAL INFORMATION

The following paragraphs describe the detailed information that should be on hand prior to beginning a SYSGEN session. In general, the user should be thoroughly familiar with the hardware configuration to be used, particularly:

1. Amount of memory available.
2. Number and type of I/O devices.
3. Type and number of disk systems.
4. Number and size of partitions.
5. Type and number of communications controllers.
6. The highest device vector address used.

Furthermore, the user should be sure that any user-written driver object modules for non-standard devices he intends to use (refer to Chapter 2) are included at System Build time. The user must also make certain that all disks to be used by MUMPS have been initialized with the MUMPS Backup and Utility System (see Appendix I). This system will take care of formatting the disk, checking for bad blocks, and setting up bit maps.

3.5.1 Memory Size

MUMPS-11 requires at least 24K words of memory and can use up to 124K words. Memory increases by a factor of 4K. For example, 24K, 32K, 64K, etc.

3.5.2 Disk Memory

There are four types of disk memory systems that can be used:

Disk Type	Disk System	No. Drives
0	RK11/RK03 or RK05 disk pack (RK)	8 drives (maximum)
1[1]	RK611/RK06 cartridge disk (RM) RF11/RS11 fixed head disk (RF) RH11/RS04 fixed head disk (RS)	8 drives (maximum) 8 drives (maximum) 8 drives (maximum)
2	RP11-C/RP02 or RP03 disk pack (RP)	8 drives (maximum)
3	RH11/RP04, RP05 or RP06 disk pack (RJ)	8 drives (maximum) [2](4 RP06 drives (maximum))

MUMPS-11 uses any one or all types, with the exception that disk types 2 and 3 cannot be combined on the same system. At least one drive of the types listed above is required. When there is more than one type of disk system, a choice must be made as to which system is to be the system device. The system device is always drive 0 of the specified disk system (i.e., RK0, RF0, RS0, RM0, RP0, RJ0). It contains: the memory image of the operating system, the global and program directories of the System UCI (UCI#1), and unless otherwise specified, the global and program directories for all other UCI's.

3.5.3 Partitions

A partition is required for each simultaneously logged-in terminal user and each STARTed program. Systems with less than 28K words can have up to 18 partitions, while systems greater than 28K words can have up to 40 partitions. The system has both standard and special size partitions. When a user logs in at a terminal, MUMPS-11 allocates a standard size partition or, if that is unavailable, any special size partition greater than or equal to the standard size. Special size partitions allow the user to allocate memory efficiently. A user can START a program in a partition of a specified size. Partitions can also be tied to specific terminals, which will be described later.

Partitions are allocated in increments of 128 memory words. Partition size can range from 128 words (minimum) to 4096 words (maximum). Size is specified by the number of 128-word increments. For example, 2 increments is a 256-word partition and 8 increments is 1024 words. At least one standard size partition must be specified and it should be at least 2K to allow the System Utility Programs to run.

Memory not used for partitions is automatically incorporated into 256-word buffers, for use by mass storage devices.

[1] Only one of these three disk types (RM, RF or RS) may be used as disk type 1.

[2] If RP06 disk drives are used, one physical RP06 is treated as two logical RP04 or 05 drives; therefore, you are limited to four RP06 drives.

3.5.4 I/O Devices

The number and types of I/O devices available, (including line printer, paper tape, card reader, DECtape, magtape, and special devices) must be determined. The controllers to which terminals are connected must be specifically identified. The console terminal is always assumed to be connected to either a KL11 or a DL11A controller. Controllers for other terminals are grouped into four classes:

Class	Type
1	DC11
2	KL11
3	DL11E, DL11W
4	DMC11
5	DH11 (16 terminal multiplexer) or DZ11 (8 terminal multiplexer)

Determine the number of terminals in each class, not including the console terminal. The relationship between device numbers for terminals (device numbers 1, 4 - 19 and 64 - 111) and the various classes of controllers is shown below:

1. Class 1 controllers, if any exist, are assigned the lowest device numbers (4 through 19).

2. Class 2 controllers, if any exist, are assigned device numbers immediately following Class 1 controllers (up to device 19).

3. Class 3 controllers, if any exist, are assigned device numbers immediately following Class 2 controllers (up to device 19).

4. Class 4 controllers, if any exist, are assigned device numbers immediately following class 3 controllers (up to device 19).

5. Class 5 controllers:

   Multiplexer No. 1, if a DH multiplexer is attached to devices 64 - 79 or if a DZ multiplexer is attached to devices 64 - 71.(1)

   Multiplexer No. 2, if a DH multiplexer is attached to devices 80 - 95 or if a DZ multiplexer is attached to devices 72 - 79.(1)

   Multiplexer No. 3, if a DH multiplexer is attached to devices 96 - 111 or if a DZ multiplexer is attached to devices 80 - 87.[1]

[1] The device number assignments for the DZ-11 multiplexer assume that no DH11's are assigned at the same time. For combined use of DH and DZ11's, see examples a and b under item 5 (class 5 controllers), and SYSGEN's multiplexer questions.

Extra Multiplexers Necessary If Using DZ's:

Multiplexer No. 4 is attached to devices 88 - 95.

Multiplexer No. 5 is attached to devices 96 - 103.

Multiplexer No. 6 is attached to devices 104 - 111.

Example:

a. If a system has two DC11's, three DL11A's, two DL11E's, three terminals on a DH11 multiplexer and three terminals on a DZ11 multiplexer, assignments are:

Device No.	Controller
4	DC11
5	DC11
6	DL11A
7	DL11A
8	DL11A
9	DL11E
10	DL11E
64	DH11 MUX #1
65	DH11
66	DH11
80	DZ11 MUX #2
81	DZ11
82	DZ11

As the above example shows, when DH11 and DZ11 multiplexers are combined, DH11's always get the lowest numbers. Because a DH11 multiplexer must have 16 devices assigned (even though only three are being utilized here), the DZ11 multiplexer (MUX # 2) is assigned to device number 80.

b. If there are: no DC11's, KL11's, DL11A's or DL11B's, three DL11E's, one DH11 with four terminals, and two DZ11's with two terminals each, assignments are:

Device No.	Controller
4	DL11E
5	DL11E
6	DL11E
64	DH11 MUX #1
65	DH11
66	DH11
67	DH11
80	DZ11 MUX #2
81	DZ11
88	DZ11 MUX #3
89	DZ11

Note that in the previous example, MUX # 2 is a DZ11, and needs only 8 device numbers (vs. 16 for DH11), so MUX # 3 must start at device number 88.

Each DC11-type controller has four selectable speeds (baud rates). Table 3-1 lists the various models of a DC11 controller along with the associated baud rates and speeds.

Table 3-1
DC11 Baud Rates

MODEL	SPEED SETTING			
	0	1	2	3
DC11-AA	110	134.5	150	300
DC11-AB	110	300	1200	1800
DC11-AC	110	150	600	1200
DC11-AD	50	110	134.5	150
DC11-AE	75	110	134.5	150
DC11-AG	134.5	150	300	1200
DC11-AH	110	134.5	600	1200
DC11-AX	110	134.5	150	(see note 1)

1. Speed 3 on a DC11-AX is a customer-specified non-standard baud rate above 600 baud.

NOTES

A. If the user intends to use data sets, MUMPS assumes that they are connected to DL11E controllers. These controllers are designed to interface to the BELL 103A Data Set.

B. It is assumed that CPU-CPU devices are connected to DL11A controllers (one for each line), and that the CPU's are local (i.e., hardwired).

3.5.5 Directory Allocation

Up to 16 UCI's can be designated. SYSGEN normally allocates directory space for programs and globals on the system disk. Storage area for programs and globals usually begins on the same disk unit as associated directories. As programs and globals increase in size and number, storage area will ultimately flow across disk unit and disk system physical boundaries. The user may locate the directories on any cylinder of any unit of any disk system. Or, he can choose not to allocate any directory areas for a UCI. This might be done to restrict the facilities available for programmer trainees.

3.5.6 Restricting Disk Storage Space For Programs And Globals

SYSGEN also gives the user the capability of limiting program and global storage to specific disk units or disk systems. This feature, when used with the directory allocation options (described in Section 3.5.5), is very useful in helping the system manager to allocate his disk storage areas for programs and globals.

3.5.7 Device Interrupt Vector Locations

Each I/O device is hardwired to a memory word location in low memory, usually below location 400 (octal). The MUMPS System Stack normally grows downward in memory until this limit is reached. Because vector space may exceed this limit in large systems, the system manager must know the highest address used as a device vector, to allow SYSGEN to set the MUMPS internal limit register appropriately.

3.6 STARTING PROCEDURE - MAGTAPE DISTRIBUTION

The following paragraphs describe the steps to be performed to load and start SYSGEN with a tape distribution media. If the MUMPS Basic System is not running, refer to Section 2.6 for loading procedures.

NOTE

> SYSGEN must be used only with the MUMPS-11 Basic System.

Perform the following steps to load and start SYSGEN:

1. Mount the SYSGEN tape on an appropriate tape drive.

2. Set the tape transport controls for magtape (7-track: DEC-11-MMSYA-A-MB7 9-track: DEC-11-MMSYA-A-MB9), as follows:

   a. Remove the write enable ring from the tape reel.

   b. Position tape at LOAD POINT.

   c. Select drive number 0 through 3.

   d. Set ON LINE/OFF LINE switch to ON LINE.

3. Begin MUMPS operation by logging in to the system at the console terminal as follows:

   a. Type CTRL/C (depress CTRL key, then type a C), or press the BREAK key. MUMPS responds by typing:

      ```
      MUMPS-11 V04B #1

      UCI:
      ```

   b. Immediately to the right of 'UCI:' type MGR (the system's default UCI code), followed by a colon (:), followed by the system's Programmer Access Code (PAC), which is: CTRL/X CTRL/X CTRL/X. (Depress the CTRL key and type X three times.)

```
UCI:MGR:CTRL/X CTRL/X CTRL/X↵
```

If no mistakes are made, MUMPS responds with its right caret (>) prompting symbol. If an error message is typed (SYNTX, etc.), repeat steps a. and b. (above). If an 'at' symbol (@) is typed rather than a >, make sure all SWITCH REGISTER switches are in the OFF (down) position, then repeat steps a. and b. above.

4. If a disk pack is to be the system device, mount a fresh pack on drive 0. Set the drive switches to: POWER ON, ON LINE and WRITE ENABLE. If other disk packs are to be used in the system being generated, they should also be installed on their drives at this time in the same way.

   Exception - If a SYSTEM MERGE is being performed, then the MUMPS data base packs should be mounted.

   NOTE

   Fresh packs must go through MBU initialization procedures, listed in Appendix I, prior to mounting and using. Fresh packs are only mounted on drives which are being added.

5. To load and start SYSGEN, type the following command line for magtape immediately to the right of the >:

```
A␣dev␣P␣5␣L␣␣D␣1↵
```

```
where:  dev = 47 for magtape drive 0
              48 for magtape drive 1
              49 for magtape drive 2
              50 for magtape drive 3
              55 for DECtape drive 0
              56 for DECtape drive 1
              57 for DECtape drive 2
              58 for DECtape drive 3
```

When loading is complete, SYSGEN identifies itself and begins its interactive dialog with the user.

3.7 STARTING PROCEDURE - RK05, RK06 DISTRIBUTION

The following paragraphs describe the steps to be performed to load and start the MUMPS SYSGEN program with an RK05 or RK06 distribution media. If the MUMPS Basic System is not already running, refer to Section 2.7 for loading procedures.

NOTE

SYSGEN and SYSTEM MERGE must be used only with the MUMPS-11 Basic System.

1. Remove the MUMPS System Builder (MSB) cartridge, mount and load the MUMPS cartridge on drive 0, POWER ON, ON LINE, and WRITE ENABLE. If other disk packs are to be used in the system being generated, they should also be installed on their drives at this time in the same way.

2. Begin MUMPS operation by lowering all switches and logging in to the system at the console terminal, as follows:

   a. Type CTRL/C (depress CTRL key, then type a C), or press the BREAK key. MUMPS responds by typing:

   ```
   MUMPS-11 V04B #1

   UCI:
   ```

   b. Immediately to the right of 'UCI:' type MGR (the system's default UCI code), followed by a colon (:), followed by the system's Programmer Access Code (PAC), which is: CTRL/X CTRL/X CTRL/X. (Depress the CTRL key and type X three times.)

   ```
    UCI:MGR:CTRL/X CTRL/X CTRL/X↵
   ```

   MUMPS should respond with its right caret (>) prompting symbol. If an error message is typed (SYNTX, etc.), repeat step a. (above). If an 'at' symbol (@) is typed rather than a >, make sure all SWITCH REGISTER switches are in the OFF (down) position, then repeat steps a. and b. above.

3. To load and start SYSGEN, type the following command line immediately to the right of the >:

   ```
   >C  SG1↵
   ```

   If no mistakes were made (see Section 3.8), SYSGEN identifies itself and begins its interactive dialogue with the user.

3.8 ERROR CONDITIONS

SYSGEN checks all user responses for correct syntax. When a syntax error is detected, a message prompting the acceptable set of answers is typed and the question is repeated. SYSGEN does not detect logical errors such as specifying more memory than is available, or specifying that parity checking is desired when, in fact, it is not.

If the user discovers an error before he types Carriage RETURN, it can be corrected by typing RUBOUTs to delete single characters or CTRL/U to delete the entire line. Errors that are found after typing Carriage RETURN cannot be corrected in this way. If the error occurred in the answer to a question before the System Library Transfer Operation, MUMPS must be reloaded from the Basic System tape or disk (see Section 2.6.4 for magtape, Section 2.7.4 for disk) and a new SYSGEN session started. If the error occurred after the System Library Transfer Operation (see Section 3.9.5), it can be corrected by running the MSP System Utility Program (see Section 6.2.7) once SYSGEN has completed operation.

Another common error is not readying a disk device when requested. A fatal error occurs when the user responds that he has readied all devices as requested, but he actually has not. There is no recovery after this type of error; the system must be reloaded (see Section 2.6.4 for magtape, Section 2.7.4 for disk) and the SYSGEN process begun again.

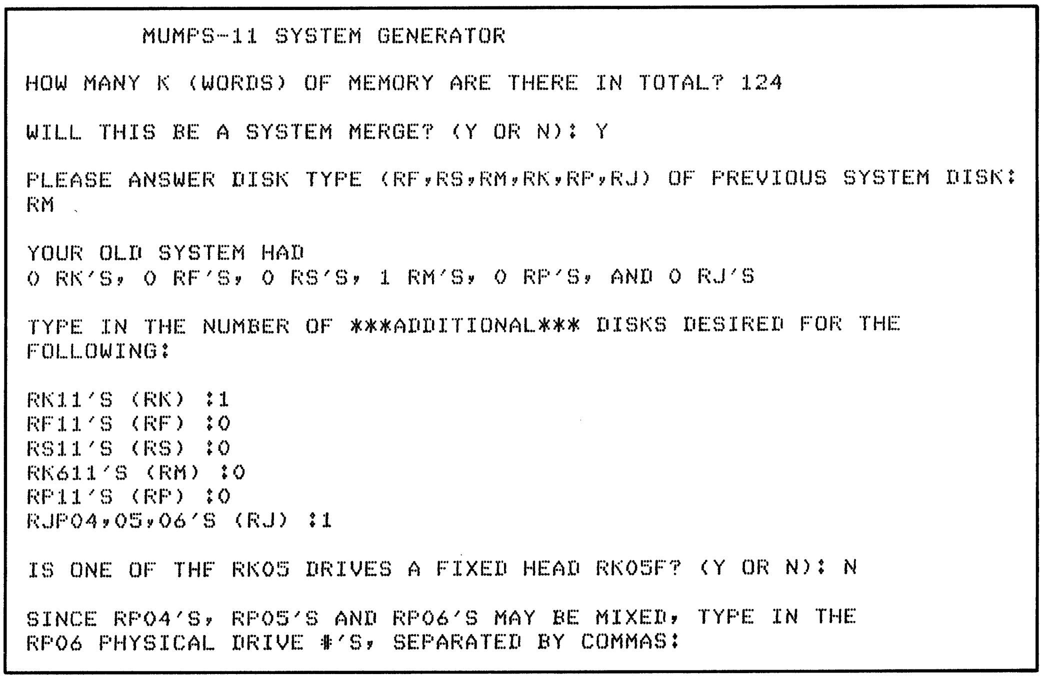
The user should also never attempt to restart SYSGEN without reloading the Basic System. This is because SYSGEN alters memory locations that cannot be reset by doing an ERASE and KILL of the user's partition. The Basic System must always be reloaded when restarting a SYSGEN to ensure the integrity of the in-core MUMPS system.

3.9 DETAILED DESCRIPTION OF OPERATION

This section describes the options available to the SYSGEN user and explains some of the planning necessary for determining an optimum configuration for a particular installation. Each 3-digit paragraph denotes a new set of questions on a common subject area, and is accompanied by a figure showing typical user response. Each 4-digit paragraph presents an individual question and a description of its meaning and use. Each answer, (Y for YES, and N for NO) must be terminated with a Carriage RETURN.

3.9.1 Memory And Disk Memory Availability Questions

This section defines the number of words in memory and the number of physical disk drives of each disk system (see Figure 3-2). At least 24K words of memory and one drive of one disk system type (RK, RF, RS, RM, RP, and RJ) must be specified.



```
        MUMPS-11 SYSTEM GENERATOR

HOW MANY K (WORDS) OF MEMORY ARE THERE IN TOTAL? 124

WILL THIS BE A SYSTEM MERGE? (Y OR N): Y

PLEASE ANSWER DISK TYPE (RF,RS,RM,RK,RP,RJ) OF PREVIOUS SYSTEM DISK:
RM

YOUR OLD SYSTEM HAD
0 RK'S, 0 RF'S, 0 RS'S, 1 RM'S, 0 RP'S, AND 0 RJ'S

TYPE IN THE NUMBER OF ***ADDITIONAL*** DISKS DESIRED FOR THE
FOLLOWING:

RK11'S (RK) :1
RF11'S (RF) :0
RS11'S (RS) :0
RK611'S (RM) :0
RP11'S (RP) :0
RJP04,05,06'S (RJ) :1

IS ONE OF THF RK05 DRIVES A FIXED HEAD RK05F? (Y OR N): N

SINCE RP04'S, RP05'S AND RP06'S MAY BE MIXED, TYPE IN THE
RP06 PHYSICAL DRIVE #'S, SEPARATED BY COMMAS:
```

Figure 3-2 SYSGEN Session: Memory and Disk Memory Availability

HOW MANY K (WORDS) OF MEMORY ARE THERE IN TOTAL? nnn↵

MUMPS uses from 24K to 124K of memory in 4K increments. Type a number (nnn) between 24 and 124 that is a multiple of 4. For example, 32, 56, 104, etc. Specifies 32K words, 56K words, 104K words, etc.

WILL THIS BE A SYSTEM MERGE (Y or N): Y↵

If the user is building this system for the first time this question should be answered N. If the user already has a running MUMPS installation and simply wants to update the MUMPS operating system/interpreter with the new version, this question should be answered Y.

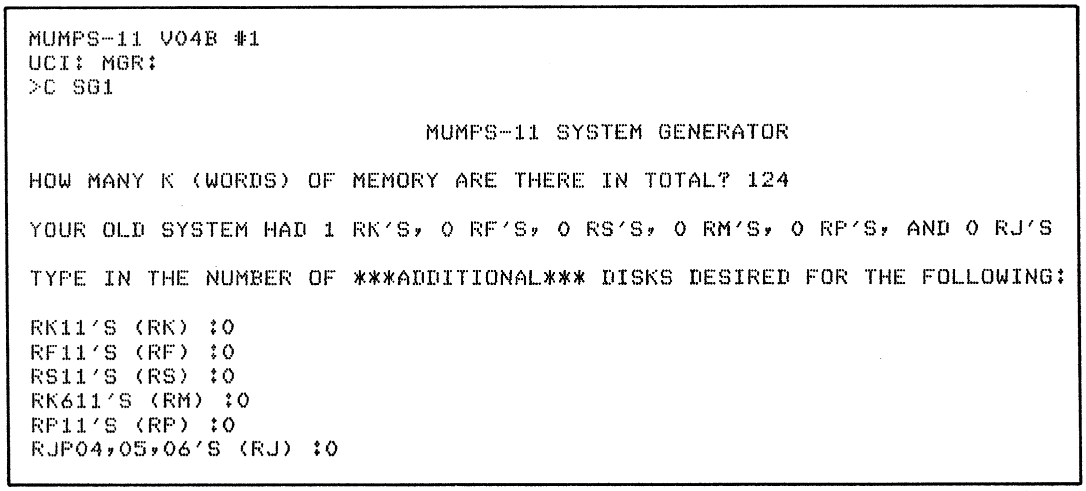
NOTE

> If the distribution media is an RK05 or RK06, this question is not asked; Y is assumed since the RK05, RK06 distribution always merges a new version of MUMPS into the MUMPS distribution system.

Answering this question with Y allows a user to replace the operating system while retaining the programs and globals on the disk. A N response will clean the disk of all programs and data.

A Y response (or an RK05 or RK06 distribution) results in a different set of questions to determine the disk configuration. These questions are illustrated for the RK05 distribution in Figure 3-3. Note that the responses to the questions of number of additional disks should be the number of disks which are being added. If the distribution media is tape and the SYSTEM MERGE option has been selected, SYSGEN will ask:

"PLEASE ANSWER DISK TYPE (RK,RF,RS,RM,RP,RJ) OF PREVIOUS SYSTEM DISK:", and the user should respond with either RK,RF,RS,RM,RP, or RJ.



```
MUMPS-11 V04B #1
UCI: MGR:
>C SG1

                              MUMPS-11 SYSTEM GENERATOR

HOW MANY K (WORDS) OF MEMORY ARE THERE IN TOTAL? 124

YOUR OLD SYSTEM HAD 1 RK'S, 0 RF'S, 0 RS'S, 0 RM'S, 0 RP'S, AND 0 RJ'S

TYPE IN THE NUMBER OF ***ADDITIONAL*** DISKS DESIRED FOR THE FOLLOWING:

RK11'S (RK) :0
RF11'S (RF) :0
RS11'S (RS) :0
RK611'S (RM) :0
RP11'S (RP) :0
RJP04,05,06'S (RJ) :0
```

Figure 3-3 SYSGEN Session: Memory Availability - RK05 Distribution

NOTE

> Before running a SYSTEM MERGE, a physical copy of every disk pack used in the system should be made. This backup is extremely important. Do not proceed from this question if that backup has not been performed.

HOW MANY DRIVES OF THE RK11 (RK) SYSTEM ARE THERE? n↵

Up to 8 drives of the RK11 disk system are allowed. Type a number (n) from 0 through 7.

HOW MANY DRIVES OF THE RK611 (RM) SYSTEM ARE THERE? n↵

Up to 8 drives of the RK611 disk system are allowed. Type a number (n) from 0 through 7. If RK611 disks are being used, the following two questions (RF or RS) do not print out. This is because only one of the three disk types (RM, RF or RS) can be selected as disk type 1 (see Section 3.5.2).

HOW MANY DRIVES OF THE RF11 (RF) SYSTEM ARE THERE? n↵

Up to 8 drives of the RF11 disk system are allowed. Type a number (n) from 0 through 7.

HOW MANY DRIVES OF THE RH11 (RS) SYSTEM ARE THERE? n↵

Up to 8 drives of the RH11 disk system are allowed. Type a number (n) from 0 through 7.

HOW MANY DRIVES OF THE RP11 (RP) SYSTEM ARE THERE? n↵

Up to 8 drives of the RP11 disk system are allowed. Type a number (n) from 0 through 7.

HOW MANY DRIVES OF THE RH11 (RJ) SYSTEM ARE THERE? n↵

Up to 8 drives of the RH11 disk system are allowed. Type a number (n) from 0 through 7.

SINCE RP02's AND RP03's MAY BE MIXED, TYPE IN THE RP02 DRIVE #'S, SEPARATED BY COMMAS: n↵

If this system has RP drives, SYSGEN asks this question to determine which RP disk drives are RP02's and which are RP03's. Enter the unit number (n) of each RP02 disk drive in the system. Separate each entry with a comma (e.g., n,n,n). If there are no RP02 disk drives, type a Carriage RETURN.

SINCE RP04'S, RP05'S AND RP06'S MAY BE MIXED, TYPE IN THE RP06 DRIVE #'S SEPARATED BY COMMAS: n↵

If this system has RP drives, SYSGEN asks this question to determine which RP disk drives are RP04's and RP05's and which are RP06's. One physical RP06 is treated as two logical RP04's or RP05's, as far as MUMPS is concerned. Enter the unit number(n) of each RP06 disk drive in the system. Separate each entry with a comma (e.g., n,n,n). If there are no RP06 disk drives, type a Carriage RETURN.

<u>WHICH TYPE OF DISK SYSTEM IS THE SYSTEM DEVICE?</u>

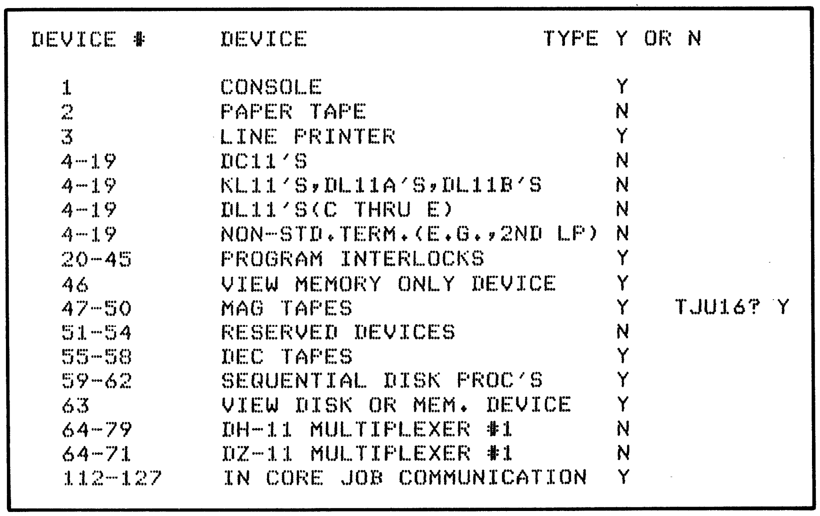
| RK
RF,RS or RM
RP
RJ |

SYSGEN asks this question when there is more than one type of disk system specified. Drive 0 of the selected system is always the system device.

3.9.2 I/O Device Availability Questions

This section asks questions about the existence of I/O devices (Figure 3-4). SYSGEN types the first two columns of a 3-column tabular listing. The first column contains the device number(s), the second column the device name and the third column the user's yes(Y) or no (N) response. Since device numbers 1, 20-45, 46 and 63 must always exist, SYSGEN itself answers Y for these devices. Therefore, the paragraphs that follow do not deal with them.

Device numbers 1, 4-19 are available for terminal-type devices that operate with the various models of DEC's single line communications controllers (DC11, DL11, KL11 or DMC11). Up to 16 devices can be specified (see Section 3.5.4). As the user responds to the questions concerning these devices, SYSGEN types out the device numbers that remain for assignment. Sixteen words of memory are allocated for each device (1-19, 64-111) specified.



```
DEVICE #      DEVICE                  TYPE Y OR N

  1           CONSOLE                      Y
  2           PAPER TAPE                   N
  3           LINE PRINTER                 Y
  4-19        DC11'S                       N
  4-19        KL11'S,DL11A'S,DL11B'S       N
  4-19        DL11'S(C THRU E)             N
  4-19        NON-STD.TERM.(E.G.,2ND LP)   N
  20-45       PROGRAM INTERLOCKS           Y
  46          VIEW MEMORY ONLY DEVICE      Y
  47-50       MAG TAPES                    Y    TJU16? Y
  51-54       RESERVED DEVICES             N
  55-58       DEC TAPES                    Y
  59-62       SEQUENTIAL DISK PROC'S       Y
  63          VIEW DISK OR MEM. DEVICE     Y
  64-79       DH-11 MULTIPLEXER #1         N
  64-71       DZ-11 MULTIPLEXER #1         N
  112-127     IN CORE JOB COMMUNICATION    Y
```

Figure 3-4 SYSGEN Session: I/O Device Availability

<u>2 PAPER TAPE</u> | Y
N |

If paper tape reader/punch exists, type Y; otherwise type N.

```
3  LINE PRINTER |Y|
                |N|↵
```

If line printer exists, type Y; otherwise type N.

```
4-19  DC11'S |Y|
             |N|↵
```

If one or more terminals use DC11 controllers, type Y; otherwise type N. If the answer is Y, SYSGEN asks for the quantity to be specified:

```
TYPE 0 - 16:  nn↵
```

Type a number (nn) from 0 through 16. A 0 answer causes SYSGEN to skip to the next question.

```
m-19  KL11'S,DL11'S,DL11B'S |Y|
                            |N|↵
```

SYSGEN types the remaining device numbers(m) to be used (m-19). If the answer is Y, SYSGEN asks for the quantity to be specified:

```
TYPE 0 - t:  nn↵
```

Type a number (nn) from 0 through t (t = total device numbers remaining). A 0 answer causes SYSGEN to skip to the next question.

```
m-19  DL11'S(C THRU E) |Y|
                       |N|↵
```

SYSGEN types the remaining device numbers(m) that can be used (m-19). If the answer is Y, SYSGEN asks for the quantity that can be specified:

```
TYPE 0 - t:  nn↵
```

Type a number (nn) from 0 through t (t = total device numbers remaining). A 0 answer causes SYSGEN to skip to the next question.

```
m-19 DMC11'S |Y|
             |N|↵
```

SYSGEN types the remaining device numbers (M) that can be used (M-19). If the answer is Y, SYSGEN asks for the quantity to be specified:

```
TYPE 0-t:  nn↵
```

Type a number (nn) from 0 through t (t = total device numbers remaining). A 0 answer causes SYSGEN to skip to the next question.

```
m-19  NON-STD.  TERM.  (E.G.2ND LP) |Y|
                                    |N|↵
```

SYSGEN types the remaining device numbers(m) that can be used (m - 19). Non-standard terminal type devices are devices that can be connected to standard DEC single line controllers (DL11,KL11 or DC11). Two examples of non-standard devices are: an additional line printer, and RT02 data entry terminal.

If devices of this type are to be used, in addition to answering Y to this question, the user must also connect the devices' interrupt vectors and I/O registers to the MUMPS terminal I/O driver as described in Appendix H.

If there are no devices of this type, type N. If the answer is Y, SYSGEN types:

```
     TYPE 0 - t:  nn↵
```

Type in the number (nn) of non-standard terminal devices in the range 0 through t (t = total device numbers remaining).

47-50 MAGTAPES | Y / N | ↵

If there are one or more magtape drives in the system, type Y; otherwise type N. If the answer is Y, SYSGEN types: TJU16? If the system will use TJU16 drives, type Y; otherwise type N.

51-54 RESERVED DEVICES | Y / N | ↵

A reserved device is any standard or non-standard I/O device controller and associated device driver combination. Examples of such devices are: card readers, magnetic tape drives (not manufactured by DEC), DECcassettes, etc. If devices of this sort are to be used, the appropriate device driver modules must have been previously designed by the user (see MUMPS-11 Programmer's Guide) and incorporated into the MUMPS Basic System at system build time (see Section 2.3.2), before answering Y to this question. If the answer is Y, SYSGEN types out the following reminder to this effect:

```
          IS YOUR OWN MODULE LINKED?
```

If the user has not done this, he must terminate SYSGEN operation and link his driver module(s) into the system, as described in Appendix I of the MUMPS-11 Programmer's Guide. Once this is accomplished, a new SYSGEN session can be initiated.

55-58 DECTAPES | Y / N | ↵

If there are one or more DECtape drives in the system, type Y; otherwise type N.

59-62 SEQUENTIAL DISK PROC'S | Y / N | ↵

The Sequential Disk Processor (SDP) device is a feature of MUMPS-11 that allows for non-standard sequential type file structuring on specified areas of the disks in the system (see MUMPS-11 Programmer's

Guide). If this feature is desired, in addition to typing Y, the user must have already incorporated the SDP module into the Basic System (see Section 2.8). If no SDP is wanted, type N. To actually reserve disk space for the SDP, the user must run the SDP System Utility Program (Chapter 6) once the new system is operational.

64-79 DH-11 MULTIPLEXER#1 Y/N ↵

MUMPS-11 allows for the installation of up to three DH-11 or up to six DZ-11 multiplexers, each of which may have up to 16 local terminals attached. If a multiplexer exists in the hardware configuration, type Y. In addition, the user must have previously incorporated the multiplexer module into the Basic System (see Section 2.8). If no multiplexer is present, type N.

80-95 DH-11 MULTIPLEXER #2 Y/N ↵

If the response to the previous question was Y, this question is asked to determine the presence of a second multiplexer. If a second DH11 is included in the hardware configuration, type Y; otherwise type N.

96-111 DH-11 MULTIPLEXER #3 Y/N ↵

If a third DH-11 is not included, type N and the following set of questions will be asked.

64-71 DZ-11 MULTIPLEXER #1 Y/N ↵

This question is typed only if less than three DH-11's were selected. If only one DH-11 was selected previously, up to four DZ-11's can now be added; if two DH-11's were selected previously, up to two DZ-11's can now be added.

72-79 DZ-11 MULTIPLEXER #2 Y/N ↵

This is only asked if the answer to "64-71 DZ-11 MULTIPLEXER #1" was "Y". If only one DH-11 was selected previously, up to three more DZ-11's can now be added; if two DH-11's were selected previously, only one more DZ-11 can now be added.

Depending on the user's answers to the previous DH-11 and DZ-11 questions, one of the questions in each set below will be asked.

80-87 DZ-11 MULTIPLEXER #1
OR
80-87 DZ-11 MULTIPLEXER #3

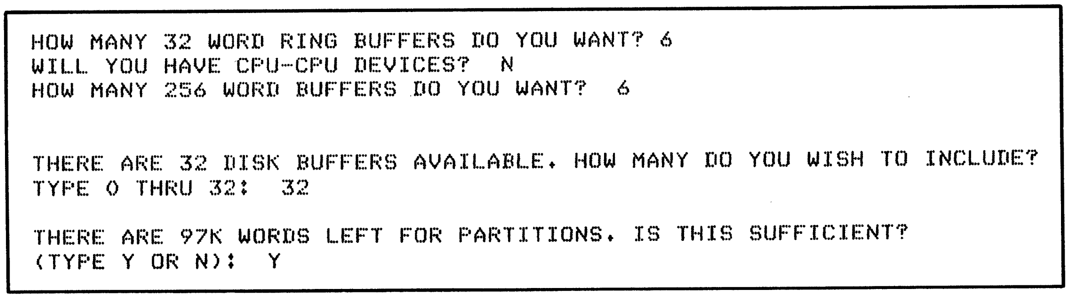
```
88-95 DZ-11 MULTIPLEXER #2
            OR
88-95 DZ-11 MULTIPLEXER #4


96-103 DZ-11 MULTIPLEXER #1
            OR
96-103 DZ-11 MULTIPLEXER #3
            OR
96-103 DZ-11 MULTIPLEXER #5


104-111   DZ-11 MULTIPLEXER #2
                OR
104-111   DZ-11 MULTIPLEXER #4
                OR
104-111   DZ-11 MULTIPLEXER #6
```

112-127 JOB COMMUNICATION | Y / N | ↵

Job communication allows jobs to pass information to each other through pseudo devices. If this module has been incorporated in the Basic System, type Y; otherwise type N.

3.9.3 Device Buffer Allocation Questions

This section (see Figure 3-5) contains questions that deal with the allocation of memory buffer space for the various I/O devices and the disk memory system as established by questions covered in Sections 3.9.1 and 3.9.2.



```
HOW MANY 32 WORD RING BUFFERS DO YOU WANT? 6
WILL YOU HAVE CPU-CPU DEVICES?  N
HOW MANY 256 WORD BUFFERS DO YOU WANT?   6


THERE ARE 32 DISK BUFFERS AVAILABLE. HOW MANY DO YOU WISH TO INCLUDE?
TYPE 0 THRU 32:  32

THERE ARE 97K WORDS LEFT FOR PARTITIONS. IS THIS SUFFICIENT?
(TYPE Y OR N):  Y
```

Figure 3-5 SYSGEN Session: Device Buffer Allocation

HOW MANY 32 WORD RING BUFFERS DO YOU WANT? nn↵

All terminal type devices (device numbers 1, 4 - 19, and 64 - 111) utilize buffers on a shared, first come, first served basis. These buffers, called ring buffers, require 32 words of memory each. Job communication does not use these ring buffers.

The number of 32-word ring buffers is determined by the total number of terminals that are to be simultaneously active (ASSIGNed) in the new system, including paper tape and line printer. Type a number between 1 and 64.

WILL YOU HAVE CPU-CPU DEVICES? $\left|\begin{matrix}\text{Y}\\\text{N}\end{matrix}\right|$ ↵

This question does not print out unless CPU-CPU devices were selected in the previous (I/O Device Availability) section. CPU-CPU devices, although serviced by MUMPS as terminal devices (device numbers 1, 4 - 19 and 64-111), do not share buffers with the other devices. Each device is assigned a single 44-word buffer for its own use. These devices are assumed to be DL11E controllers, each of which is directly connected (hardwired) to another central processor.

If there are one or more CPU-CPU devices in the system, type Y; otherwise, type N. If the response is Y, SYSGEN types:

HOW MANY 44 WORD BUFFERS? nn↵

Enter the number (nn) of 44-word buffers - one for each device (0 - 16). If 0 is typed, SYSGEN allocates none and skips to the next question.

WILL YOU BE USING DMC-11 COMMUNICATION CONTROLLERS? $\left|\begin{matrix}\text{Y}\\\text{N}\end{matrix}\right|$ ↵

This question does not print out unless DMC-11 Controllers were selected in the previous (I/O Device Availability) section. If you are using one or more DMC-11 communication controllers, type Y; otherwise, type N.

HOW MANY 256 WORD BUFFERS DO YOU WANT? nn↵

Mass storage type devices include DECtape (device numbers 55-58), Magtape (device numbers 47-50), Sequential Disk Processor (device numbers 59-62), and the VIEW Disk or Memory Device (device number 63). Each of these devices uses a single 256-word buffer when operating.

Buffers which are not being used reside in a common pool and are allocated to mass storage devices as needed. A job which requests a buffer when none are available will hang until a buffer becomes available. Enter the number (nn) of 256-word mass storage device buffers desired. For example, if there are two DECtape drives and one magtape drive, four buffers may be desired, or perhaps as few as two. (The VIEW command uses a buffer.) At least one buffer should always be allocated to enable system programs to use the VIEW command.

THERE ARE 32 DISK BUFFERS AVAILABLE. HOW MANY DO YOU
WISH TO INCLUDE? TYPE 0 THROUGH 32: nn↵

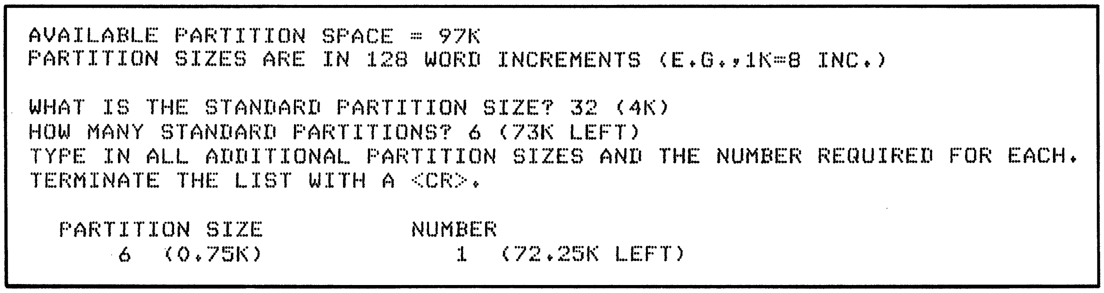
The Basic MUMPS-11 System contains two 256-word buffers that are used for all disk I/O operations. Most significantly, these buffers are used to store disk blocks of global data. As the number of concurrently active globals in a system increases, the average global access response time will also increase unless the number of disk buffers in the system is increased. This is because fewer logical disk accesses can be made before a physical disk access is required to service another global. Up to 32 additional disk buffers can be specified to reduce global access time. Enter the number (nn) of 256-word buffers desired. SYSGEN then responds:

```
THERE ARE nn. nnK WORDS LEFT FOR PARTITIONS.  IS THIS
SUFFICIENT?

TYPE Y OR N: | y |
             | n |↲
```

If the number of words of memory remaining after buffer allocation (nn.nnK) is sufficient, type a Y to continue; otherwise, type an N to return to the previous question.

3.9.4 Partition Allocation Questions

This section deals with questions about the allocation of memory for both standard and non-standard size partitions (see Section 3.5.3 and Figure 3-6). At least one standard partition must be allocated. A maximum of 18 partitions, for systems with less than 28K of memory, or 40 partitions, for systems with more than 28K of memory, can be allocated. Partition size is specified in terms of the number of 128-word increments desired. Table 3-2, following, shows partition size in 128-word increments versus partition size (words). Though any size can be specified from 1 (128 words) to 32 (4096 words), a highly recommended size for standard partitions is 16 (2048 words). This allows operation of the DEC-supplied System Library Programs, and it is a good general size for MUMPS applications programs. In addition, it is recommended that a 0.75K-word partition be allocated for use by the System Caretaker program (CTK), described in Section 6.2.2. At the completion of this section, SYSGEN allocates unused memory as 256-word buffers.



```
AVAILABLE PARTITION SPACE = 97K
PARTITION SIZES ARE IN 128 WORD INCREMENTS (E.G.,1K=8 INC.)

WHAT IS THE STANDARD PARTITION SIZE? 32 (4K)
HOW MANY STANDARD PARTITIONS? 6 (73K LEFT)
TYPE IN ALL ADDITIONAL PARTITION SIZES AND THE NUMBER REQUIRED FOR EACH.
TERMINATE THE LIST WITH A <CR>.

  PARTITION SIZE          NUMBER
      6  (0.75K)             1  (72.25K LEFT)
```

Figure 3-6 SYSGEN Session: Partition Allocation

Table 3-2
Partition Sizes

128-word Increments	Size (words)
1	128
2	256
3	384
4	512
5	640
6	768
7	896
8	1024
9	1152
10	1280
11	1408
12	1536
13	1664
14	1792
15	1920
16	2048
17	2176
18	2304
19	2432
20	2560
21	2688
22	2816
23	2944
24	3072
25	3200
26	3328
27	3456
28	3584
29	3712
30	3840
31	3968
32	4096

SYSGEN begins this section by informing the user of the amount of memory (nnK) available for partitions:

```
        AVAILABLE PARTITION SPACE = nnK
```

followed by the message:

```
        PARTITION SIZES ARE IN 128 WORD INCREMENTS(E.G., 1K = 8 INC.)
```

WHAT IS THE STANDARD PARTITION SIZE? nn↵

Enter a number which is between 1 and 32 (see Table 3-2). SYSGEN types the partition's size to the right of the answer.

HOW MANY STANDARD PARTITIONS? nn↵

Enter a number (nn) between 1 and m, where m is the result of dividing the amount of memory available for partitions by the standard partition size (in words). If the user types an answer that would cause allocation of more memory than is available, SYSGEN types:

```
nnK SPACE EXCEEDED.
```

and allows the user to try again.

Immediately to the right of the user's answer, SYSGEN types the amount of memory remaining for non-standard size partition allocation. Before typing the next question and if there is any memory space remaining, SYSGEN types:

```
TYPE IN ALL ADDITIONAL PARTITION SIZES AND THE NUMBER
REQUIRED FOR EACH.  TERMINATE THE LIST WITH A <CR>.
```

```
PARTITION SIZE  NUMBER
```

SYSGEN types the column headings shown above. Under each heading the user can enter the sizes of non-standard partitions. Typing a Carriage RETURN (only) terminates the list. SYSGEN types out the amount of memory remaining to the right of each entry in the NUMBER column.

3.9.5 Disk Inclusion and System Library Transfer

At this point SYSGEN is about to include all disk devices in the system and, unless the distribution media is RK05 or RK06, subsequently transfer the System Library Programs from the SYSGEN tape to the system disk. In so doing, it enters the name of each program in the program directory of UCI No. 1 (system manager's UCI). To this end, SYSGEN types out the following message:

```
PREPARE FOR DISK INCLUSION.
READY ALL DISKS (MOUNT FORMATTED, INITIALIZED PACKS, WRITE ENABLED,
ETC.)
WHEN ALL ARE READY, TYPE <CR>.
```

Before typing Carriage RETURN, the user must:

1. Mount all disk packs that were specified as being in this system. (These packs must have been previously formatted for PDP-11 use, and initialized by the MUMPS Backup and Utility System described in Appendix I.)

2. Set all appropriate disk drive function switches to: POWER ON, ON LINE, WRITE ENABLE (or WRITE PROTECT OFF).

Once Carriage RETURN is typed, SYSGEN proceeds with disk inclusion procedures. SYSGEN checks each disk to be sure that it has been initialized by the MUMPS Backup and Utility System. This system performs bad block checking and creates the MUMPS bits maps. If any disk has not been initialized, SYSGEN will abort.

Upon completion, the following question is asked (unless the distribution media is the RK05 or RK06):

```
DO YOU WANT THE EDITOR TO BE LOADED TO ↑%? |Y|↵
                                           |N|
```

if the Editor (see Chapter 4 of the MUMPS-11 Programmer's Guide) is to be inserted into the ↑% global, type Y. The following message is then output.

THE SYSTEM LIBRARY WILL NOW BE TRANSFERRED TO THE SYSTEM DISK.

SYSGEN types out the name of each program transferred.

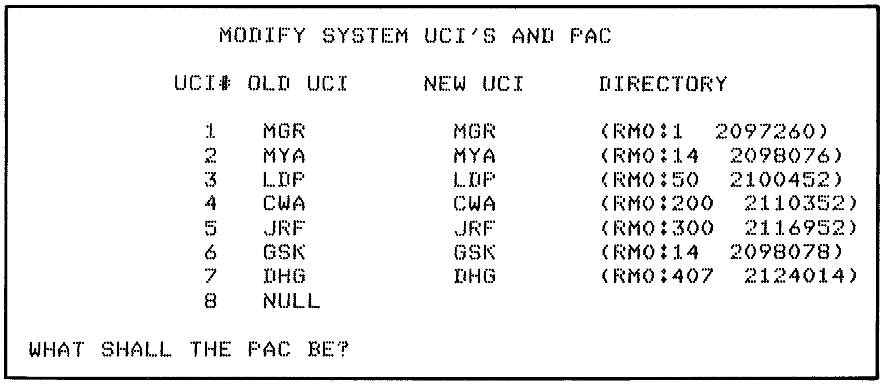
NOTE

This terminates the operation of SYSGEN proper. SYSGEN now transfers control to the Modify System Parameters program (MSP) to continue the system generation process. Logical errors that occur in answering the questions which follow can be corrected by running MSP after the current session has been completed.

3.9.6 Modify System UCI's and PAC Questions

The questions in this section (see Figure 3-7) deal with the current User Class Identifier (UCI) codes, the Programmer Access Code (PAC), and new UCI codes. A user can alter the current UCI or PAC, and specify new UCI codes. In addition, the user can specify: the actual disk cylinder for the program/global directory area of each UCI, that no directory be allocated at all, or that MSP allocate space for the directories. In the last case, MSP will allocate directories on the system disk. Up to 16 UCI's can be specified.

The dialog in this section is in tabular form. The user enters responses in a left-to-right, column-by-column manner. MSP enters the UCI's that currently exist in the system in the OLD UCI column. If there are none, the column entry is NULL. The paragraphs that follow describe the responses that the user can make in the remaining two columns.



```
                MODIFY SYSTEM UCI'S AND PAC

            UCI# OLD UCI      NEW UCI      DIRECTORY

              1   MGR          MGR         (RMO:1   2097260)
              2   MYA          MYA         (RMO:14   2098076)
              3   LDP          LDP         (RMO:50   2100452)
              4   CWA          CWA         (RMO:200   2110352)
              5   JRF          JRF         (RMO:300   2116952)
              6   GSK          GSK         (RMO:14   2098078)
              7   DHG          DHG         (RMO:407   2124014)
              8   NULL

WHAT SHALL THE PAC BE?
```

Figure 3-7 SYSGEN Session: Modify UCI's and PAC

NEW UCI

Under this column, the user can replace the current UCI by typing a new UCI name or, if NULL, enter a new UCI name. UCI names must be three characters long. The first character must be either alphabetic or the % symbol. In addition, the following alternative responses (underlined) can be typed.

Carriage RETURN to retain the old UCI/directory status and continue the listing sequence.

N (Carriage RETURN) to retain the old UCI, without allocating any directory, and continue the listing sequence.

Any of the above responses following a NULL UCI entry terminates the listing sequence and causes MSP to skip the next question.

DIRECTORY

Under this column, the user can specify the existence and location of user program and global directory storage for each UCI. If a directory is allocated, MSP types the directory's location to the right of the user's last response in the parenthetical form:

(dkn:c b)

where: dkn = Disk device mnemonic and unit number
(e.g., RK2, RF6, RP0, RJ3, etc.)
RS4 or
RM1

c = Disk unit cylinder number

b = MUMPS block number

The user can type one of the following responses (underlined):

Carriage RETURN to request MSP to allocate a directory on the system disk.

N (Carriage RETURN) to prevent directory allocation but continue the listing sequence.

RKn:c(Carriage RETURN) to allocate the directory on RK11 disk unit n (0 - 7), cylinder number c (0 - 199)

RFn or (Carriage RETURN) to allocate the directory on RF or
RS
RSn disk unit n (0 - 7)

RMn:c(Carriage RETURN) to allocate the directory on RK06 disk unit n(0-7), cylinder number c(0-407)

RPn:c(Carriage RETURN) to allocate the directory on RP11 disk unit n (0 - 7), cylinder number c (0 - 199 for RP02 or 0 - 399 for RP03)

RJn:c(Carriage RETURN) to allocate the directory on RH11 disk unit n (0 - 7), cylinder number c (0 - 407)

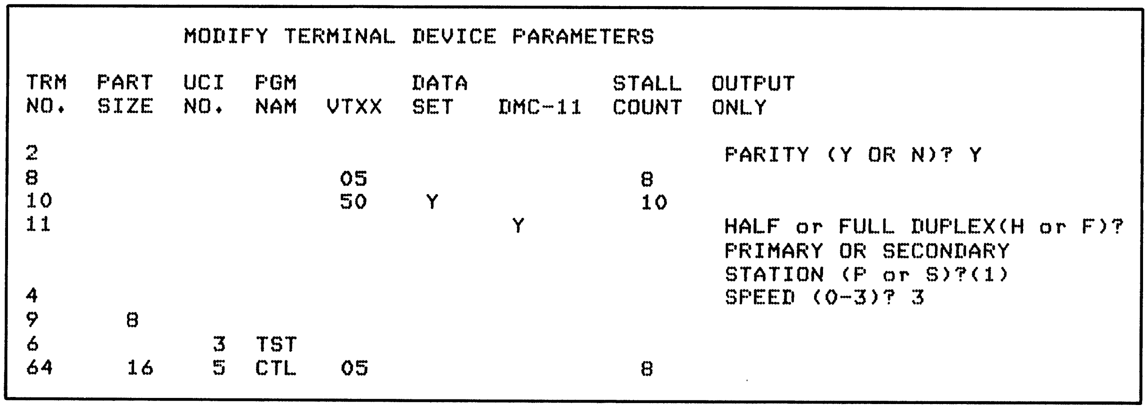
MSP continues the listing sequence, unless prematurely terminated by the user, until 16 UCI's have been specified. MSP then proceeds to the next question in the series.

WHAT SHALL THE PAC BE? |xxx ↵| ↵

The Programmer Access Code (PAC) allows a user to operate at a MUMPS terminal in Direct Mode. The PAC can be any three printing or non-printing characters in the MUMPS character set except CTRL/C, CTRL/O and CTRL/U. To retain the current PAC, type Carriage RETURN. To create a new PAC, type in the new PAC (xxx) followed by a Carriage RETURN.

3.9.7 Modify Terminal Device Parameters Questions

Proper responses to the questions in this section (see Figure 3-8) allow the user to establish and modify operating parameters for terminal type devices (1, 4 - 19 and 64 - 111). The dialog is in a table format consisting of nine columns. The user enters responses in a left-to-right, column-by-column manner.



```
           MODIFY TERMINAL DEVICE PARAMETERS

TRM  PART  UCI  PGM        DATA         STALL  OUTPUT
NO.  SIZE  NO.  NAM  VTXX  SET   DMC-11 COUNT  ONLY

2                                               PARITY (Y OR N)? Y
8                     05                  8
10                    50     Y            10
11                                 Y            HALF or FULL DUPLEX(H or F)?
                                                PRIMARY OR SECONDARY
                                                STATION (P or S)?(1)
4                                               SPEED (0-3)? 3
9      8
6            3  TST
64     16    5  CTL   05                  8
```

Figure 3-8 SYSGEN Session: Modify Terminal Device Parameters

Default values for each parameter (Basic System) are as follows:

The terminal is not tied to a partition, program or UCI.

The terminal is a hard-copy device.

The terminal is not a CPU-CPU device.

The terminal is not connected to a data set (the appropriate modules must be linked into the Basic System at system build time. See Chapter 2).

The stall count (number of filler characters) is 0.

[1] This question only prints out if the previous question is answered H.

The terminal has both input and output capabilities.

Papertape parity generation and checking are not performed.

DC11 controllers are set to the lowest speed (see Table 3-1).

TRM
NO.
nn↵
↵

Enter the number of the terminal (nn) for which parameter change is desired. The number must be in the range 1,4 - 19 or 64 - 111. To terminate the listing sequence, type Carriage RETURN. MSP skips to the next set of questions (see Section 3.9.8).

PART
SIZE
nn↵
↵

If the terminal is to be tied to a specific size partition (i.e., tied terminal), enter the size (nn). A terminal that is not tied to a partition is allocated a standard partition or larger (if necessary) at log-in time. Note that the size must be one that was specified under the questions covered in Section 3.9.4. If the terminal is not to be tied, type Carriage RETURN.

UCI
NO.
nn↵
↵

A proper response to this question allows a terminal to be tied to a particular User Class Identifier code (UCI). When a terminal is tied to a UCI it must be tied to a program also (see comments under following question entitled PGM NAM). This restricts the user to programs and globals residing in the program and global directories associated with that UCI; log in via other UCI's is not allowed. To tie the terminal to a UCI, enter the UCI's number (nn). The UCI number is determined by its place in the order of UCI's entered. The System UCI is UCI number 1, the fifteenth UCI code entered is UCI number 15, etc. If a UCI number is specified, MSP advances to the next column. If a terminal is not to be tied to a UCI, type Carriage RETURN (MSP skips the next column). To untie a terminal that has previously been tied, type a space when MSP asks for the UCI number, and another space when MSP asks for the program name. If only a Carriage RETURN is hit, MSP assumes that no changes will be made to the modified terminal.

PGM
NAM
xxx↵
↵

A proper response to this question allows the user to tie a terminal to a specific program (xxx). MSP expects an answer in this column only if a UCI was specified in the previous question. When a terminal is tied to a program, only that program is available to users of that terminal. Program operation begins when a user types CTRL/C or presses the BREAK key. To tie the terminal to a program, enter the program's name. Otherwise type Carriage RETURN to skip to the next column. If the terminal is already tied to a UCI, an answer to this question is required.

VTxx

05↵
50↵
52↵
55↵
↵

If the terminal is a VT05 video terminal, type 05; if it is VT50, type 50; if it is a VT52, type 52; if it is a VT55, type 55; otherwise, type Carriage RETURN.

CPU
DEV

Y↵
↵

This question will only print out if CPU-CPU devices were selected in the I/O Device Availability section. The CPU-CPU device is used for local (hardwired) interprocessor communication. Only DL11A controllers can be used this way. Therefore, the user should be sure that the terminal specified is connected to a DL11A controller. If the terminal is to be a CPU-CPU device, type a Y; otherwise, type a Carriage RETURN. Once a terminal is set up to be a CPU-CPU device, it cannot be changed via MSP; instead, the user must system-generate a new system.

DMC-11

Y↵
↵

This question will only print out if DMC-11 controllers were selected in the I/O Device Availability section. The DMC-11 communications controller is used for local or remote interconnection of PDP-11 computers in network applications. If one or more DMC-11's are to be used, type a Y; otherwise, type a Carriage RETURN.

DATA
SET

Y↵
↵

If the terminal is to interface to a BELL type 103A Dataset (supported by the DL11-E, DH11 or DZ11 interfaces), type Y; otherwise type Carriage RETURN.

STALL
COUNT

n↵
↵

When terminals operate at data rates of 300 baud or greater, certain of their functions such as Carriage RETURN, FORM Feed and LINE FEED cannot be performed within the interval allotted to the other characters and functions. This can cause character displacement or total loss for characters that immediately follow one of these functions. To prevent this from happening, MUMPS can wait a specified number of character times after one of these functions before sending out the next character. This is accomplished simply by sending out a number of non-printing RUBOUT character codes (octal code 177). The number of RUBOUTS is specified by the stall count value.

Stall counts for applicable DEC terminals are:

Serial	LA30	(300 baud) =	10
	LA36	(300 baud) =	10
	VT05	(300 baud) =	1
	VT05	(600 baud) =	4
	VT05	(1200 baud) =	8
	VT05	(2400 baud) =	20

If the device is to operate at 300 baud or greater, enter the stall count (nn). Some of the newer devices, such as the VT50, VT52 and VT55 Video Terminals, have zero stall counts. In these cases, type a Carriage RETURN to advance to the next column.

OUTPUT
ONLY

| Y↵ |
| ↵ |

If the device is to be used for output only, (e.g., a second line printer or an RT01) type Y; otherwise type Carriage RETURN.

PARITY (Y OR N)? | Y / N | ↵

A proper response to this question allows the user to enable or disable parity generation for paper tape I/O. Even parity is generated on output and validated during input. This question is asked only when device number 2 is specified in TRN NO. column. The question is typed to the right of the "output only" column. If parity is desired, type Y; otherwise type N.

SPEED (0 - 3)? n↵

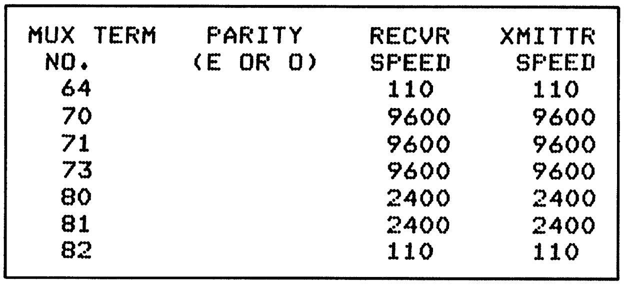
A proper response to this question allows the user to set the speed (baud rate) of the various models of DC11 controllers. The relationship between the speed value (0 - 3) and the actual baud rate is shown in Table 3-1 in the beginning of the Chapter. This question is output only when a device number that is associated with a DC11 (see Section 3.9.2) is specified in the TRM NO. column. Enter a number (n) between 0 and 3.

3.9.8 Multiplexer Terminal Line Conditioning Questions

This section (see Figure 3-9) allows the user to establish and modify operating parameters for terminals (devices 64-111) connected to DH11 or DZ11 multiplexers. These questions are asked only if a multiplexer was specified (see Section 3.9.2). The information obtained in these questions is in addition to that covered in Section 3.9.7, specifically:

> address(es) of the multiplexer status register(s) and interrupt vector
>
> whether parity operation and checking are to be performed
>
> I/O line speed (baud rate)

The address questions appear only when MSP is running under the control of SYSGEN. The remaining questions request that terminal operating parameters be specified. The dialog for this question is in table format consisting of four columns. Responses are entered in a left-to-right, column-by-column manner.



```
MUX TERM   PARITY    RECVR   XMITTR
 NO.      (E OR O)   SPEED    SPEED
  64                  110      110
  70                 9600     9600
  71                 9600     9600
  73                 9600     9600
  80                 2400     2400
  81                 2400     2400
  82                  110      110
```

Figure 3-9 SYSGEN Session: Multiplexer Terminal Line Conditioning

Default values for each parameter (Basic System) are as follows:

1. The terminal has no parity checking
2. The terminal is not connected to the system (i.e., RECVR SPEED = 0, XMITTR SPEED = 0)

Note that the multiplexer must be specified as being part of the Basic System at system build time (see Section 2.7).

```
MULTIPLEXER n VECTOR ASSIGNED STARTING AT aaa(OCTAL)
IS THIS ADDRESS CORRECT? |Y|↵
                         |N|
```

The interrupt vectors for DH11's and DZ-11's are normally connected to standard locations in low memory reserved for this purpose, and SYSGEN assumes that this is the case. However, in cases where there are very large non-standard hardware configurations, an exception to this rule may exist. When this message is output, the n specifies the multiplexer number (1, 2, 3, 4, 5 or 6), and aaa specifies the address (octal) of the first word of the multiplexer interrupt vector area. This address is the first vector following the last vector used by the last DC11, KL11 or DL11 specified (see Section 3.9.2). Vectors are assigned contiguously beginning at address 300 (octal). If this address is correct, type Y; otherwise, type N. If N is typed, SYSGEN asks for the correct vector address, as in the following example:

```
MULTIPLEXER 1 VECTOR ASSIGNED STARTING AT 000310 (OCTAL)
IS THIS ADDRESS CORRECT? N
TYPE IN CORRECT VECTOR ADDRESS IN OCTAL (#<500): 320

TYPE IN THE OCTAL ADDRESS OF MUXn STATUS REGISTER: aaaaaa↵
```

Each DH11 or DZ11 multiplexer has a status register in the PDP-11's external memory page. There are no standard addresses for these registers, therefore the user must obtain this information from his DIGITAL field service representative.

Enter the address aaaaaa of the status register of multiplexer n (1 or 2). The address must be in the range: 160000<aaaaaa<177774.

```
TYPE IN OCTAL ADDRESS OF MUX1 STATUS REGISTER (#>160000):  160040
```

MUX TERM
NO.
| nn↵ |
| ↵ |

If a parameter change is required for a specific terminal, enter the number of the terminal (nn). The number must be in the range 64-79 if one DH11 exists, 64-95 if two exist, 64-111 if three exits; or, 64-71 if one DZ11 exists, 64-79 if two DZ11 exist, 64-78 if three exist, 64-95 if four exist, 64-103 if five exist, or 64-111 if six exist. Note that if DH and DZ multiplexers are combined, DH multiplexers will be assigned the lowest numbers. To terminate the listing sequence, type Carriage RETURN to cause MSP to skip to the next set of questions.

PARITY
| E |↵
| O |

A proper response to this question allows the user to select I/O parity generation and checking for the terminal. Type an E if even parity is desired, or O if odd parity is desired, or a Carriage RETURN if no parity checking is desired.

RECVR
SPEED
nnn↵

Enter a number (nnn) that is the baud rate for input from the specified terminal. The acceptable inputs are listed in Table 3-3 below. A baud rate of 0 effectively disconnects a terminal's receiver from the multiplexer and is a method of disabling currently nonexistent or broken terminals.

XMITTR
SPEED
nnn↵

Certain terminals (e.g., VT05, serial LA30) allow selectable speeds and some (e.g., VT05) allow split input/output speeds. This question allows the user to specify the Baud rate for output to the specified terminal. Acceptable responses are shown in Table 3-3 following. A 0 response disconnects the terminal's transmitter from the multiplexer.

Table 3-3
Multiplexer Terminal Speeds

0	50	75	110	134.5	150	200
300	600	1200	1800	2400	4800	9600

(Except for 0, all entries are baud rates.)

3.9.9 Modify Magtape Default Mode Questions

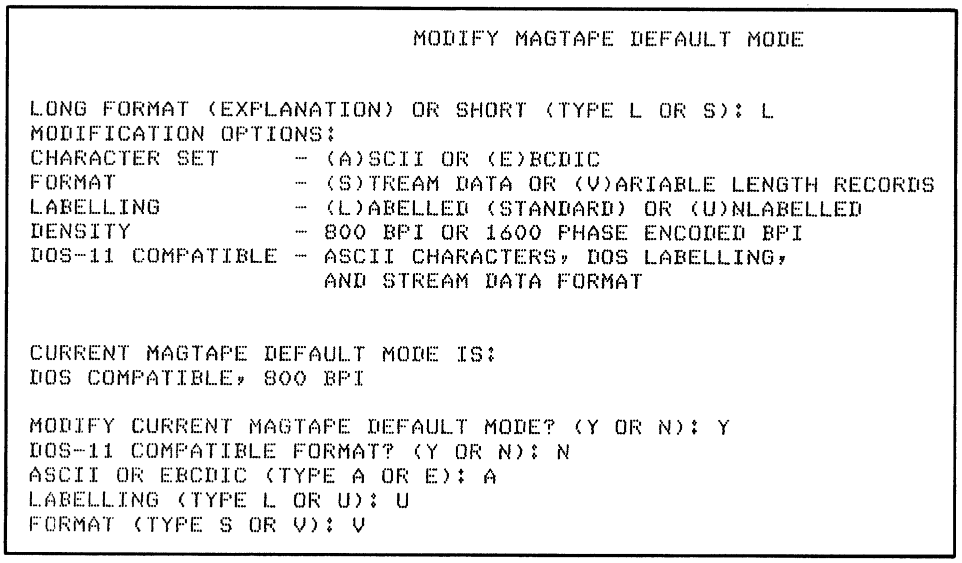
This section (see Figure 3-10) allows the user to establish and modify the tape format to be used for all magtape I/O [1] These questions are asked only if TM11's or TJU16's have been specified (see Section 3.9.2). The format specified may be either DOS-11 compatible (ASCII characters, DOS labelling, and Stream Data format) or any combination of the following:

Character Set	-	ASCII or EBCDIC
Labelling	-	Standard labelling or unlabelled
Format	-	Stream data or variable length records.

For the TJU16, the user must also specify the tape density desired (800 BPI or 1600 Phase Encoded BPI). The density for the TM11 is automatically set to 800 BPI.

NOTE

> If the EBCDIC character set is selected, the user must be sure that the EBCDIC module was linked during the SYSTEM BUILD procedure described in Chapter 2.



```
                     MODIFY MAGTAPE DEFAULT MODE

LONG FORMAT (EXPLANATION) OR SHORT (TYPE L OR S): L
MODIFICATION OPTIONS:
CHARACTER SET      - (A)SCII OR (E)BCDIC
FORMAT             - (S)TREAM DATA OR (V)ARIABLE LENGTH RECORDS
LABELLING          - (L)ABELLED (STANDARD) OR (U)NLABELLED
DENSITY            - 800 BPI OR 1600 PHASE ENCODED BPI
DOS-11 COMPATIBLE  - ASCII CHARACTERS, DOS LABELLING,
                     AND STREAM DATA FORMAT


CURRENT MAGTAPE DEFAULT MODE IS:
DOS COMPATIBLE, 800 BPI

MODIFY CURRENT MAGTAPE DEFAULT MODE? (Y OR N): Y
DOS-11 COMPATIBLE FORMAT? (Y OR N): N
ASCII OR EBCDIC (TYPE A OR E): A
LABELLING (TYPE L OR U): U
FORMAT (TYPE S OR V): V
```

Figure 3-10 SYSGEN Session: Modify Magtape Default Mode

After identifying itself, this session begins operation by requesting that the user specify whether a description of the various modification options is desired or not.

[1] The Magtape Format Mode may be temporarily modified by any user via the ASSIGN command, as explained in the MUMPS-11 Language Reference Manual (DEC-11-MMLMA-D-D).

LONG FORMAT (EXPLANATION) OR SHORT (TYPE L OR S): $\left|\begin{matrix}L\\S\end{matrix}\right|$ ↵

Thereafter, the program outputs the current Magtape Default Mode if MSP (Modify System Parameters Program) is running under the control of SYSGEN; the default mode will be DOS-11 compatible and proceeds with its interactive dialogue, as follows:

MODIFY CURRENT MAGTAPE DEFAULT MODE? (Y OR N): $\left|\begin{matrix}Y\\N\end{matrix}\right|$ ↵

If the current default parameters are satisfactory, type N; otherwise, type Y.

NOTE

> All remaining questions are skipped if N is typed to the MODIFY CURRENT MAGTAPE DEFAULT MODE question.

DOS-11 COMPATIBLE FORMAT ? (Y OR N): $\left|\begin{matrix}Y\\N\end{matrix}\right|$ ↵

Type Y if the ASCII character set, DOS labelling, stream data format are desired; otherwise, type N. A Y causes the next three questions to be skipped.

ASCII or EBCDIC (TYPE A OR E): $\left|\begin{matrix}A\\E\end{matrix}\right|$ ↵

Type A if the ASCII character set will be used; type E if EBCDIC translation is desired. An E will cause ASCII characters to be translated to EBCDIC on output and EBCDIC characters to be translated to ASCII on input. E should not be selected unless the EBCDIC module was linked during the SYSTEM BUILD. If the EBCDIC module was linked, the user may still select the ASCII character set.

LABELLING (TYPE L OR U): $\left|\begin{matrix}L\\U\end{matrix}\right|$ ↵

If L is typed in response to this question, ANSI standard labelling will be used if the ASCII character set has been specified (see previous question). IBM standard EBCDIC labels will be used if the EBCDIC character set has been specified. Type U if no labels are desired.

FORMAT (TYPE S OR V): $\left|\begin{matrix}S\\V\end{matrix}\right|$ ↵

Type S if stream data format is desired. On output, characters are sequentially packed into the buffer; a CR-LF is translated to a LF, and an argument of a TYPE command is not split across a block boundary. On input, LF's, FF's, and HT's (write end of file labels) are treated as string delimiters, and CR's are ignored. TYPE V to select Variable Length logical records (ANSI standard "D" format). If V is entered, each argument of a TYPE command corresponds to a logical record which can be read as a single argument of a READ command.

DENSITY: aaaa↵

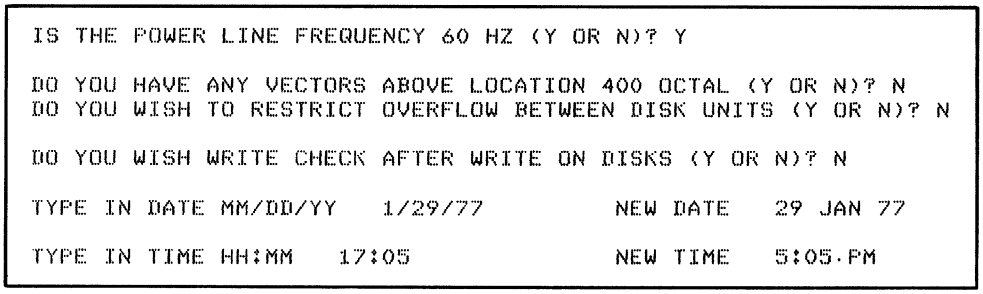
This question is asked only if TJU16's are incorporated into the system. The number (aaaa) entered must be either 800 or 1600 (Phase Encoded) BPI.

NOTE

The Magtape Default Mode selected applies to all magtape drives.

3.9.10 Miscellaneous System Information Questions

This section (see Figure 3-11) contains questions covering a variety of subject areas, as follows:



```
IS THE POWER LINE FREQUENCY 60 HZ (Y OR N)? Y

DO YOU HAVE ANY VECTORS ABOVE LOCATION 400 OCTAL (Y OR N)? N
DO YOU WISH TO RESTRICT OVERFLOW BETWEEN DISK UNITS (Y OR N)? N

DO YOU WISH WRITE CHECK AFTER WRITE ON DISKS (Y OR N)? N

TYPE IN DATE MM/DD/YY    1/29/77          NEW DATE   29 JAN 77

TYPE IN TIME HH:MM    17:05               NEW TIME   5:05.PM
```

Figure 3-11 SYSGEN Session: Miscellaneous Questions

IS THE POWER LINE FREQUENCY 60 HZ (N OR <CR>)? | N↵ ↵ |

The power line frequency directly affects the timing interval (or time between ticks) of the MUMPS internal clock. At 60 HZ, one tick = 16.6 milliseconds. At 50 HZ, one tick = 20 milliseconds. Since the system's time sharing interval (standard time slice value) is based on ticks, the power frequency directly affects the amount of CPU time allotted to each active job (i.e., terminal user or program) in the system. In response to this question, the user can type:

N (Carriage RETURN) to specify a 50 HZ power line frequency.

DO YOU HAVE ANY VECTORS ABOVE LOCATION
400 OCTAL (Y OR N)? |Y|
|N|↵

A proper response to the next two questions allows the user to set the software's system stack limit register. All device interrupt vectors use low memory addresses. Each vector normally occupies four consecutive locations. Large configurations, however, may have vectors which use memory space above location 400 (octal). If the last used vector base address is 370 (octal) or less, type N. (SYSGEN skips the next question). If the hardware is wired with vectors above 400 (octal) type Y.

TYPE IN OCTAL ADDRESS OR FIRST LOCATION
ABOVE LAST VECTOR: aaa↵

This question is output if the response to the last question was Y. The system needs to know the address to be used as the low limit of the system stack. Enter the octal address (aaa) of the first free memory word above the last interrupt vector used by the hardware. Low memory vector words are grouped four words per device.

Example: If a second line printer has been installed with interrupt vector words at 520 (octal), the user would type in 530.

DO YOU WISH TO RESTRICT OVERFLOW BETWEEN
DISK UNITS (Y OR N)?: |Y|
|N|↵

Unless otherwise specified, MUMPS allows program and global storage areas to cross physical disk unit and disk system boundaries. This means that all physical disks are normally treated as one logical disk. This option allows the user to override normal operation. This option is most often selected when the user wishes to relegate globals and programs belonging to specific UCIs to particular disk units. To accomplish this, the user must have previously allocated his global and program directories to the desired disks (see Section 3.9.6, question on Directory). If this restriction is desired, type Y (MSP skips the next question); otherwise type N.

DO YOU WISH TO RESTRICT OVERFLOW BETWEEN
DISK SYSTEMS (Y OR N)?: |Y|
|N|↵

This question is similar to the last except that the restriction is now between disk systems (i.e., RK, RF, RS, RM, RP and RJ) rather than disk units. Answer Y or N as desired.

DO YOU WISH WRITE CHECK AFTER WRITE ON
DISK UNITS (Y OR N)?: |Y|
|N|↵

A proper response to this question allows the user to validate data being output to the disk. MUMPS normally verifies disk-stored data during input only. This means that any errors that are generated during output are not detected until an attempt to input is made. At this point it is usually too late to correct the error. This question allows the user to specify that MUMPS validate all data output to the disk at the time of the output operation. This allows the program performing the output to effect a graceful recovery without subsequent data loss. When this option is selected, the time required for each physical write operation is increased by at least one disk revolution time. If this feature is desired, type Y; otherwise type N.

TYPE IN DATE MM/DD/YY mm/dd/yy↵

MSP calls the DATE SET (DAT) System Program so that the user will have a record of the system's creation date. Enter the current date (mm/dd/yy). A system's creation date can be examined using the %D Library Utility Program immediately after reinitializing the system from the disk, or from other backup media.

TYPE IN TIME HH:MM hh:mm↵

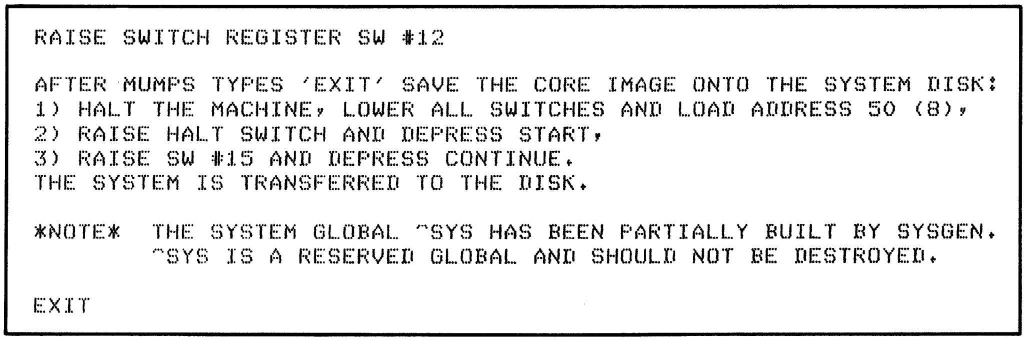
MSP calls the TIME SET (TIM) System Program so that the user will have a record of the system's time of creation. Enter the current time (24 hour notation, hh:mm). A system's time of creation can be examined by using the %T Library Program immediately after the system is loaded into memory from the disk or from other backup media.

NOTE

If a system merge, RK05 or RK06 distribution is being executed, the Disk Block Tally program (DBT) will be called at this point.

3.9.11 Saving The New System

After all questions have been satisfactorily answered, the new system residing in memory must be written out onto the save area of the system disk. To accomplish this, MSP types out the set of instructions (see Figure 3-12) which the user must follow. Once the system has been saved, a backup copy of the system disk should be made using the Backup and Utility Program described in Appendix I.



```
RAISE SWITCH REGISTER SW #12

AFTER MUMPS TYPES 'EXIT' SAVE THE CORE IMAGE ONTO THE SYSTEM DISK:
1) HALT THE MACHINE, LOWER ALL SWITCHES AND LOAD ADDRESS 50 (8),
2) RAISE HALT SWITCH AND DEPRESS START,
3) RAISE SW #15 AND DEPRESS CONTINUE.
THE SYSTEM IS TRANSFERRED TO THE DISK.

*NOTE*  THE SYSTEM GLOBAL ^SYS HAS BEEN PARTIALLY BUILT BY SYSGEN.
        ^SYS IS A RESERVED GLOBAL AND SHOULD NOT BE DESTROYED.

EXIT
```

Figure 3-12 SYSGEN Session: Saving the New System

3.9.12 Changing System Parameters

The user can alter system parameters covered by MSP sections of this chapter (see Sections 3.9.6 through 3.9.11 but excluding question in Section 3.9.7 on CPU-CPU device) simply by running MSP by itself (see Chapter 6). Changes to parameters 3.9.1 through 3.9.5 can only be made by initiating a new SYSGEN session (see Section 3.6).

CHAPTER 4

OPERATOR FUNCTIONS

4.1 INTRODUCTION

This chapter contains reference information on the use of the SWITCH REGISTER switches(1), used by all PDP-11 machines except for the PDP-11/34. Figure 4-1 shows a typical control console for a PDP-11 which uses these switches[1]. Figure 4-2 shows the function assignments for the individual switches. Reference information is also provided on the use of the control buttons for the Programmer's Console of the PDP-11/34 (see Figure 4-3)[2]. Also contained in this chapter is information on the console terminal, as well as procedures for loading and starting, restoring, restarting, saving and shutting down the system.

While MUMPS is running, various control operations can take place at the user terminals, the console terminal, the PDP-11 SWITCH REGISTER or at the Programmer's Console (PDP-11/34). Users can log in to the system from various terminals, run existing programs, and create and modify programs and globals residing within their respective UCI's. In addition, various Library Utility Programs can be run and Library Globals accessed.

The console terminal may be used as a regular MUMPS terminal or as an operating terminal for the system's Mini-ODT (Octal Debugging Technique) program. The SWITCH REGISTER switches and the Programmer's Console buttons control the operation of various functions in the operating system to facilitate daily operations and periodic maintenance.

4.2 PDP-11 SWITCH REGISTER AND CONSOLE BUTTON USAGE

The SWITCH REGISTER switches on the PDP-11 determine both the function of the console terminal and the operational status of the system in general. The position of all SWITCH REGISTER switches for normal MUMPS operation is OFF (down).

Table 4-1 shows the equivalent values of the SWITCH REGISTER switches for the PDP-11/34. To have the equivalent of two or more switches being raised simultaneously, on the PDP-11/34, simply add the octal values of those switches together and key in the result on the console. For example, to disable both garbage collection and log-ins, key in the octal value 120.

[1] For further information, refer to the specific processor handbook needed. These are listed in the Introduction, page 1-2.

[2] For further information, refer to the PDP-11/34 Processor Handbook.

NOTE

> To key in the equivalent of two or more switches on the PDP-11/34, be sure to add the octal values together. If this adding procedure is not followed, the first value keyed in will then be overwritten by the next value keyed in.

The paragraphs which follow describe the function and usage of PDP-11 SWITCH REGISTER switches in greater detail. A switch is set to 1 when it is in the raised position and to 0 when in the lowered position.

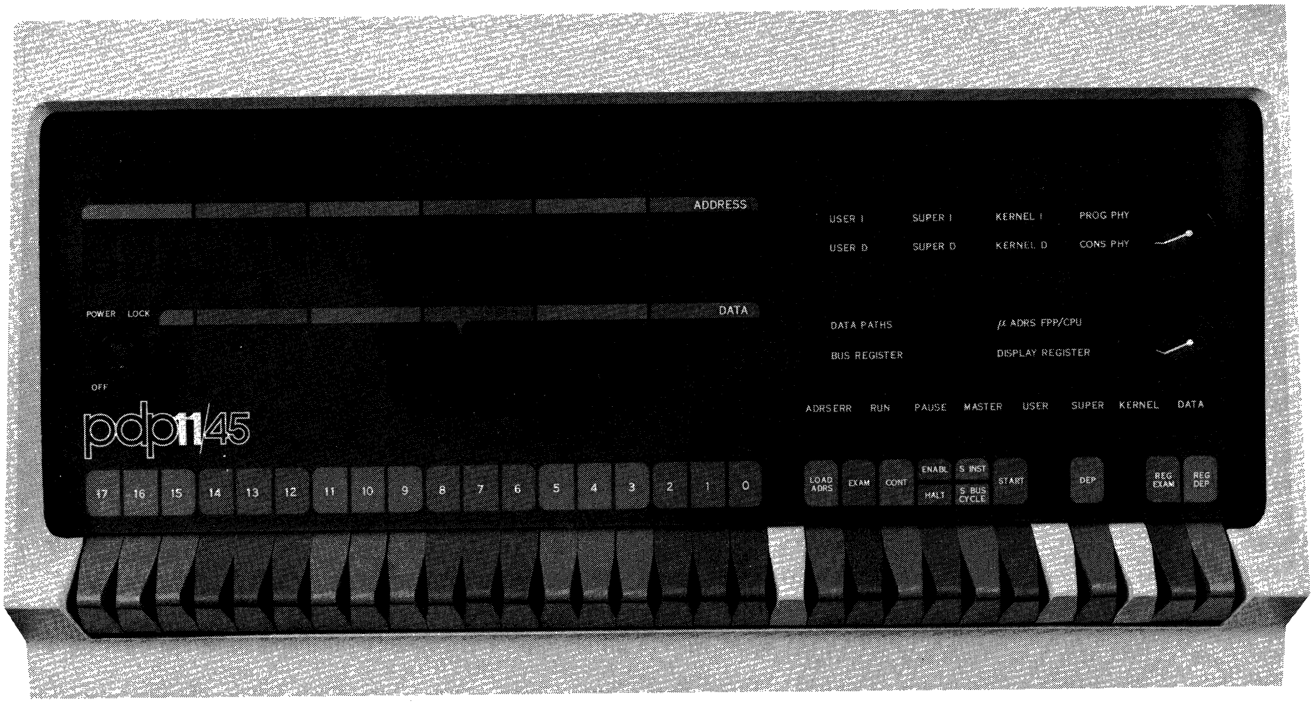


Figure 4-1 PDP-11 Switch Register Console

4.2.1 Switch 0

This switch controls the use of the console terminal. When switch 0 is set to 0, the console terminal operates as a normal MUMPS terminal. Any user can operate from the terminal under any of the legitimate UCI codes. When switch 0 is set to 1, the console terminal operates independently of the MUMPS interpreter under control of the Mini-ODT program which is embedded in the operating system. Mini-ODT can run concurrently with other regular MUMPS system operations. The system need not be shut down. The program permits the direct examination and modification of memory locations using octal numbers. Operational features of Mini-ODT are discussed in Section 4.3.1.

4.2.2 Switch 4

This switch controls the operation of the system's Garbage Collector routine. This routine is used to deallocate disk blocks that have been discarded as the result of the modification of global files and programs. In addition, the routine continually updates copies of certain memory resident tables which are part of the core image of the system contained on the system disk. When the switch is set to 0, the Garbage Collector operates; setting the switch to 1 disables the Garbage Collector. This switch should always be set to 0, except as directed by the SSD System Utility Program.

4.2.3 Switch 6

This switch is used to effect a normal system close. Once the switch has been set to 1, all users, including the console terminal user, attempting to log in to the system will receive the message:

'SYSTEM IS DOWN'

Furthermore, as current users (including the console terminal user) complete their terminal sessions, they too will be unable to log in. This switch does not interact with switch 0 and has no effect upon the operation of Mini-ODT. User system application programs should check this switch periodically in order to initiate their own shutdown procedures when switch 6 is set to 1.

4.2.4 Switch 12

This switch controls the granting of partitions. When switch 12 is set to 0, each time a user presses his BREAK key, types CTRL/C or STARTs a program, MUMPS grants a partition -- even when Switch 6 is set to 1. When Switch 12 is set to 1, the system is effectively closed. MUMPS outputs an @ symbol when a user attempts to log in.

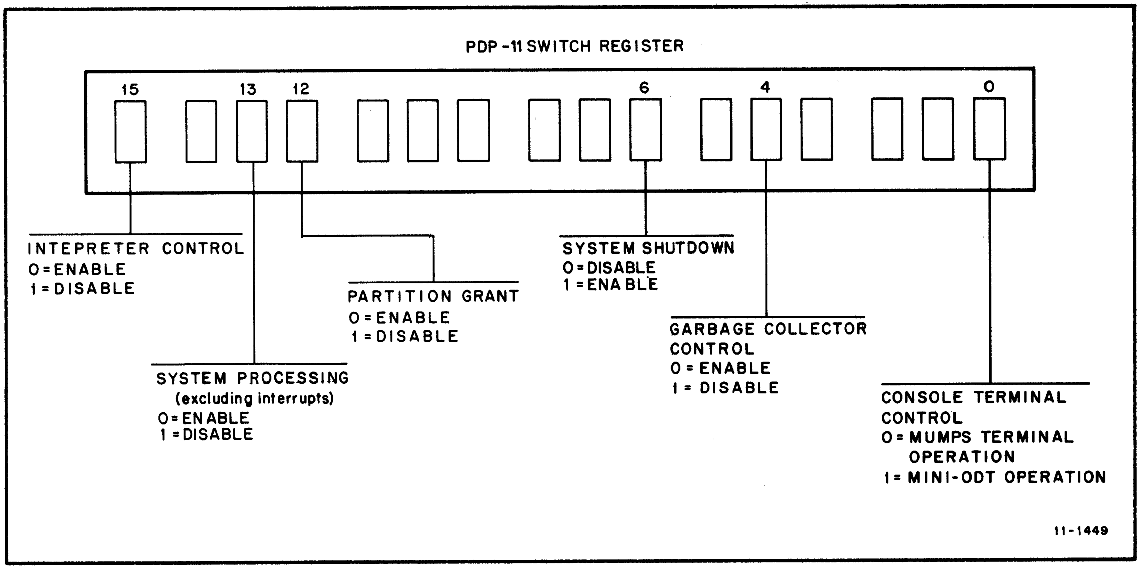


Figure 4-2 Switch Register Function Assignments

4.2.5 Switch 13

This switch controls the entire operation of the system. When set to 1, all MUMPS-11 operation ceases except for the processing of hardware interrupts. Mini-ODT operation is not affected.

4.2.6 Switch 15

This switch, when set to 1, disables operation of the interpreter for all jobs, except the job currently executing a disk I/O task. This job is also disabled upon task completion. The operation of Mini-ODT or the 'Garbage Collector' is not affected, however.

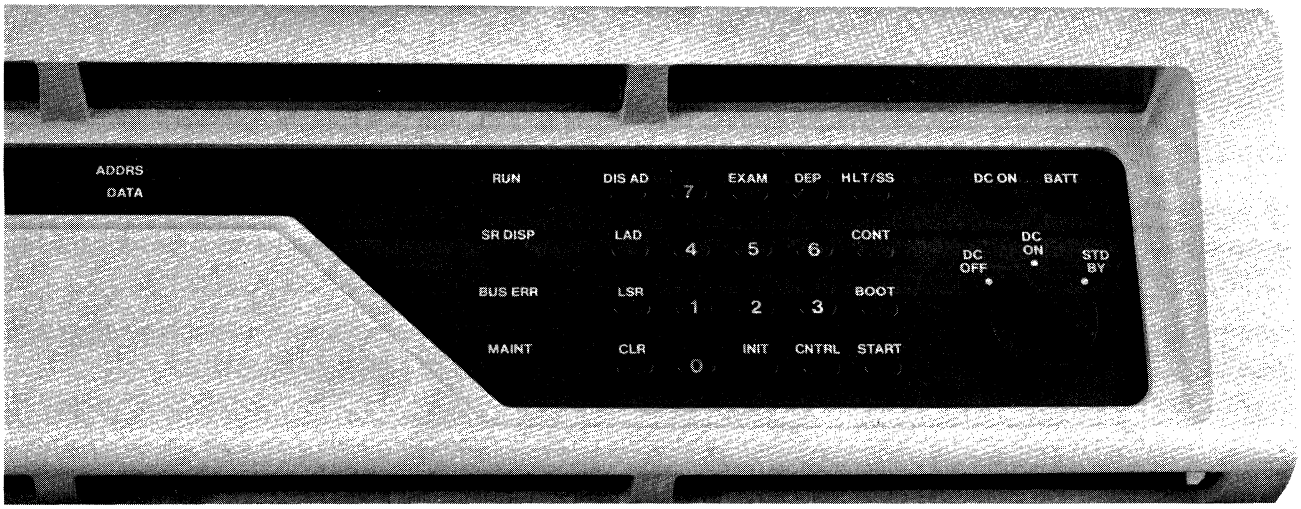


Figure 4-3 PDP-11/34 Programmer's Console

4.3 USING THE CONSOLE TERMINAL

The console terminal has four main functions in the MUMPS system:

- It is used to operate Mini^ODT (octal debugging routine).
- It is used for 'Garbage Collector' error reporting (see Section 5.1.3).
- It is used for running the privileged system utilities MSP,BCS and SSD (see Chapter 6).
- It is used as a regular MUMPS terminal.

4.3.1 Running Mini-ODT

Mini-ODT is an octal debugging program embedded in the MUMPS-11 operating system. It permits examination and modification of PDP-11 memory locations. The routine is interrupt driven and operates at the same time MUMPS is running. Mini-ODT can only be used at the console terminal. Log-in is not required. Figure 4-4 is a typical Mini-ODT session (user responses are underlined and are terminated by ALTMODE).

Mini-ODT allows the operator to:

1. Open the location specified by an octal address.
2. Print the contents of that location (in octal).
3. Modify the contents of that location (in octal).
4. Close that location and open the next sequentially higher location.

Memory locations can be accessed either as 16-bit words (Word Mode) or as 8-bit bytes (Byte Mode), as the user requires.

4.3.1.1 Operation - Mini-ODT is entered by setting SWITCH REGISTER switch 0 to 1.

NOTE

> If SWITCH REGISTER switch 0 is set to 1 while a MUMPS job is performing an output operation to the console terminal, the job's output is suspended. To recover control of the terminal, set Switch 0 to OFF, then type <CR> or CTRL/C.

Once the switch is set to 1, operation can begin. Log-in is unnecessary. The user can type up to 6 numeric characters to specify the desired address. Words are accessed by specifying even addresses (e.g., 03722) while byte addresses can be either odd or even.

The mode of operation, either byte or word, is set by prefixing, suffixing, or inserting either a W (for word mode) or a B (for byte mode) into the desired address. All subsequent requests are assumed to be in the mode last specified. Thus, typing in an odd address while in word mode produces the '?' error message.

The paragraphs below describe the procedures to be followed when using Mini-ODT. Input from the user is represented by AAA (address) or MMM (modified contents), and output from Mini-ODT by NNN.

4.3.1.2 Opening a Location - To open a location for examination or modification, type the desired octal address terminated by an ALTMODE. Mini-ODT responds by typing a slash (/) followed by the octal contents of that location. Thus:

Word Mode:

```
AAAAAA(ALTMODE)/NNNNNN
```

Byte Mode:

```
AAAAAA(ALTMODE)/NNN
```

Mini-ODT then waits to permit the user to either modify or close the location.

4.3.1.3 Changing the Contents of a Location - To change the contents of an opened location, enter up to 6 octal digits (3 if in byte mode) followed by ALTMODE.

```
AAAAAA(ALTMODE)/NNNNNN MMMMMM(ALTMODE)
```

Mini-ODT responds by entering the change into memory and closing that location.

4.3.1.4 Closing a Location Without Change - To close a location but not change it, type ALTMODE immediately after the type-out of its contents:

```
          B4352(ALTMODE) / 053 (ALTMODE)
```

4.3.1.5 Advancing to the Next Location - To examine the next higher location in sequence, type ALTMODE.

Mini-ODT types:

```
          AAAAAA/NNNNNN
         if in Word Mode

            AAAAAA/NNN
         if in Byte Mode
```

To advance to any other location (not in sequence), enter the desired addressing mode, if it is to be changed, and a new address as shown in the previous Section 4.3.1.2.

4.3.1.6 Correcting Errors - In a non-numeric character, if a decimal number, or a byte address (odd address when operating in a word mode) is entered, Mini-ODT ignores the entry and types a question mark (?).

```
                    2?
          W207(ALTMODE)?
```

If a non-numeric character or decimal number is entered as the new contents of a location, Mini-ODT ignores the entry, types a '?' and waits either for the new contents to be entered or for the location to be closed.

```
          B2234(ALTMODE)/003 3?
```

Also, the same condition results if, when in byte mode, a number larger than three digits is input. If an error is detected, the entire string is ignored.

```
1020/  000000               Examine word at 1020
150006/  000000  3452       Change the contents of word at 150006
B1104/   077                Examine five successive bytes,
001105/  000                changing the third and fourth
001106/  000  66
001107/  000  34
001110/  000
W45342/  000000             Examine word at 45342
045344/  000000             Examine next successive word
7144/  116700  113700       Examine three successive words,
007146/  172310             changing the first and third
007150/  116000  777
44445/  ?                   Attempt to examine a byte in Word Mode
                            (odd address)
44446/  000002              Correction for above
B67X?                       Wrong address, typed an "X" to
                            discard the line.
5701/  336  1? 17           Open correct address and change contents
                            to 17.
5701/  17
```

Figure 4-4 Typical Mini-ODT Session

4.3.2 Using the Console Terminal as a MUMPS Terminal

When SWITCH REGISTER switch 0 is set to 0, the console terminal operates as a regular MUMPS terminal. Users can log in under any legal UCI and create, run, modify, and file programs, as described in the MUMPS-11 Programmer's Guide.

4.4 CREATING SYSTEM BACKUP

The system manager of each MUMPS installation should create and retain a duplicate copy of the contents of each disk in his system. This can save considerable time if a system crash (failure) occurs, whether it is caused by hardware or software. The number of copies to be made and the frequency with which they are created depend in large part upon the amount of system activity on the disk. It is not uncommon for an installation to create a backup copy of the system at the close of each day's activity. There are two basic types of backup; physical backup and logical backup.

4.4.1 Logical Backup

Logical backup allows an individual user to save specified programs and globals listed in the directory of a particular UCI while the system is running.

This is normally accomplished by each UCI user via the %GS and %PS Library Utility Programs described in the MUMPS-11 Programmer's Guide. These programs are particularly useful when creating a user's private backup or when it is necessary to install programs or globals into another MUMPS-11 system. These programs do not, however, save the system image or the physical disk data structure.

4.4.2 Physical Backup

The MUMPS Backup and Utility System (MBU) allows the MUMPS system manager to backup or save the contents of any MUMPS-created disk, tape, global file directory, or individual global file. The backup medium may be magnetic tape, DECtape or any disk type supported by MUMPS. This program allows the operator to later restore the MUMPS master volumes from the backups, in the event that programming errors or a system crash lead to the loss of important information.

MBU provides the following MUMPS-related functions:

1. Backup the contents of any MUMPS-structured disk volume onto disk, magtape or DECtape.

2. Restore a MUMPS disk volume from a backup thus created.

3. Save the contents of an entire MUMPS global directory, or of individual global files, on disk, magtape or DECtape in a manner which will enable efficient restoring later, should data become lost due to hardware or software problems.

4. "Initialize" or "Format" a disk. This operation automatically checks for unusable areas on the disk (bad blocks) and prevents the MUMPS system from trying to use these areas.

5. "Label" a disk. This operation allows the user to give each disk volume a unique "name", and furthermore, to declare whether or not the given volume is to be used as a "master" or as a "backup" volume. A master is a volume containing the data in its original form, as used by MUMPS. A backup is any volume created by use of the MUMPS Backup Program BACKUP command. Additional special functions, such as special handling of bad or suspected bad blocks, setting magtape densities, or specifying non-standard interrupt vector or status register addresses, are described in Appendix I.

4.4.3 Saving the System Image

The MUMPS-11 operating system contains a special save routine that is used to save the system memory image on the system disk. When started, the routine transfers a copy of the system memory to the save area of the system disk. This copy is used to load memory prior to startup (see Section 4.5.2). The save routine can be used only when MUMPS is not running.

For all PDP-11's except the 11/34, use the Console Control Switches to perform the following steps:

1. Make sure there is no activity on system (run SSD)

2. Press HALT

3. Set SWITCH REGISTER Switches to 50(8)

4. Press LOAD ADRS

5. Raise HALT

6. Press START

7. Raise SWITCH REGISTER Switch 15

8. Press CONTINUE.

For the PDP-11/34, use the Console Control Buttons to perform the following steps:

1. Make sure there is no activity on the system (run SSD)

2. Press and hold CNTRL, while pressing HLT/SS

3. Set Display Register to 40(8) by pressing the set of numbered buttons on the console
4. Press LAD
5. Press and hold CNTRL, while pressing HLT/SS
6. Press and hold CNTRL, while pressing START
7. Key in the octal value 100000 (see Table 4-1)
8. Press and hold CNTRL, while pressing CONT.

Then set all SWITCH REGISTER Switches to 0. (For the PDP-11/34, zero out the Display Register.)

Table 4-1
PDP-11/34 Values for SWITCH REGISTER
Function Assignments

Function	Switch	Octal Value for 11/34
Mini-ODT	0	1
Garbage Collector	4	20
Disable Log-in	6	100
Disable Partition Grants	12	10000
Disable Entire System	13	20000
Disable Interpretor	15	100000

When the operation is complete, the system is automatically restarted. Thereafter, users can log in and continue normal operation.

4.5 LOADING, STARTING AND RESTARTING THE SYSTEM

The paragraphs that follow describe various procedures for loading, starting, and restarting the MUMPS-11 system software after system generation. Procedures for loading and starting the Basic (un-SYSGENed) System are provided in Sections 2.6.1 and 2.7.2. The MUMPS system software is loaded, started or restarted in several ways, depending upon the following circumstances:

- If MUMPS is currently in memory, it can be started using the system's internal restart routine.
- If MUMPS is not in memory but is on the system disk, it can be loaded and started via the PDP-11's Read Only Memory (ROM) Bootstrap Loader.
- If MUMPS is neither in memory nor on the system disk, it can be restarted by loading it onto the system disk from backup media via the Backup and Utility System (MBU). Thereafter, it can be loaded into memory via the toggle switches (or

buttons, if using the PDP-11/34) by the ROM Bootstrap Loader. The following sections describe these procedures in detail.

4.5.1 System Restart

The MUMPS-11 operating system contains a special routine to restart the copy of the MUMPS operating system residing in memory. Restart is often used to start up the system after it has been shut down or to restart the system after a system failure. When a restart after a failure is successful, only those users currently being serviced by the system run the risk of loosing their data and ASSIGNED devices. (Chapter 5 describes system error detection and recovery procedures.)

For all PDP-11's except the PDP-11/34, restart is performed using the Console Control Switches, as follows:

1. Press HALT
2. Set SWITCH REGISTER Switches to 40(8)
3. Press LOAD ADRS
4. Raise HALT
5. Press START.

For the PDP-11/34, restart is performed using the Console Control Buttons (refer to Figure 4-3) as follows:

1. Press and hold CNTRL, while pressing HLT/SS
2. Set Display Register to 40(8) by pressing the set of numbered buttons on the console
3. Press LAD
4. Press and hold CNTRL, while pressing HLT/SS
5. Press and hold CNTRL, while pressing START.

Then set all SWITCH REGISTER Switches to 0. (For the PDP-11/34, zero out the Display Register.)

After restart, users can log into the system in the normal way (described in the MUMPS-11 Programmer's Guide). If restart occurs after a system error, some devices may have become inaccessible to users. Appropriate procedures are provided in Chapter 5 to remedy this situation.

4.5.2 Loading The System Into Memory From Disk

The MUMPS Operating System, after system generation, resides as a memory image at a known location (system save area) of the system disk. This is the normal medium on which the working copy of the MUMPS system resides. When MUMPS has been shut down due to problems which might have caused alteration of the system's memory image, the system is loaded and started from disk, as follows:

1. Make sure all disks used by MUMPS are loaded, switched on, and on line. The system disk must be on drive 0.

2. Set the drive control ON LINE/OFF LINE to ON LINE, and the drive control WRITE ENABLE/WRITE LOCK to WRITE ENABLE.

3. Set the SWITCH REGISTER switches to the ROM Bootstrap address for the disk being used (see Chapter 2, Section 2.5)

4. Operate the Console Controls for the specific processor being used, as follows.

   A. For all PDP-11's except the PDP-11/34, operate the Console Control Switches in the following sequence:

      a. Press HALT

      b. Press LOAD ADR

      c. Raise HALT

      d. Press START

   B. For the PDP-11/34, operate the Console Control Buttons in the following sequence:

      a. Press and hold CNTRL, while pressing HLT/SS

      b. Press LAD

      c. Press and hold CNTRL, while pressing HLT/SS

      d. Press and hold CNTRL, while pressing START.

      Then set all SWITCH REGISTER switches to 0. (For the PDP-11/34, zero out the Display Register.)

      If loading is successful, MUMPS prints

      ```
      MUMPS-11 Vnn     RELOAD

                  (nn = version number)
      ```

      at the console terminal, then runs the system Startup Program (refer to Chapter 6).

If loading is not successful, repeat steps 1 through 5 above. If loading is still unsuccessful, a backup copy of the system, created by the Backup and Utility System (MBU), must be used (see Section 4.4). If no system backup exists, the user must rebuild the system (as described in Chapter 2) then re-SYSGEN the system (Chapter 3) and restore all logically saved programs and globals using the %PL and %GR programs.

4.5.3 Restoring the System Disk from Backup Media

The restart and disk bootstrap procedures previously described require that the operating system be resident either in memory (for restart) or on the system disk (for bootstrap). If, through some hardware or software malfunction, the system disk is damaged (e.g., by a head crash), these procedures cannot be used. Either a new system must be created (see Chapters 2 and 3) or a backup copy of the system must be obtained to restore the system to its normal operating configuration and condition. MBU, described in Section 4.2, is used not only to create physical data backup of any disk but also to restore the data. By creating and maintaining recent backup copies of disks, via MBU, the system manager can restore a system in a relatively short time. Otherwise, time-consuming system build (SYSBLD) and generation (SYSGEN) sessions are necessary simply to recreate the operating environment, and user-created programs and global data must also be restored or recreated. Detailed operating procedures for MBU are provided in Appendix I.

4.6 SYSTEM SHUTDOWN

Periodically, the system manager must terminate MUMPS system operation to perform various hardware and software maintenance tasks or to remedy some error condition. A shutdown may not necessitate an actual system halt (pressing the HALT switch). The system may only need to be shut down to time-sharing users. Such is the case when certain System Utility Programs (e.g., MSP,DBT, etc.) are to be run by the system manager. On the other hand, the system must be halted completely when MBU is to be run. Because of the complexity of system operations in the MUMPS timesharing environment, system shutdown must be performed in an organized, step-by-step manner.

Before halting the system, all users, including the console terminal user, must log out. Furthermore, the Garbage Collector routine must also have completed its operation. The system must never be halted during time-sharing operation, for this can cause terminal users much difficulty and wreak havoc upon the disk file structure. Specifically, the following can occur:

- Users can lose all partition-resident data and programs.
- Disk system management and space allocation data can be lost, resulting in unknown states for programs and global data.

The normal way to bring the system to the point where the processor can be HALTed, is by using the SSD (System Shutdown) System Utility Program. SSD tells the system manager the exact procedures to be followed in closing down a MUMPS-11 System, including the various console SWITCH REGISTER switches to be used (refer to Table 4-1 for the Equivalent Console Button Values), and how to deal with uncooperative users. In addition, it also runs the SS and DBT programs upon request. Further information on SSD is provided in Chapter 6.

CHAPTER 5

ERROR DETECTION AND RECOVERY

This chapter contains information to assist the system manager in recognizing and dealing with MUMPS-11 software and hardware error conditions.

There are three major classes of errors that can occur in a MUMPS-11 system:

1. Programming and Operating System Errors

2. Mass Storage Hardware Errors

3. System Crash Errors

The following paragraphs describe in detail the errors in each class.

5.1 PROGRAMMING AND OPERATING SYSTEM ERRORS

Most of the errors in this class are job-related and are reported by MUMPS either as typed messages to the appropriate terminals or as interrupts, via the $E System Variable, to specific user programs. These errors are described in Appendix C.

Errors described in paragraphs C.1, C.2 and C.3 of Appendix C are general programming errors. These errors usually occur during program development and result from faulty programming technique (e.g., insufficient partition space for a program or its variables, incorrect syntax, etc.). Recovery from errors of this type is straightforward (except for NAKED and DKSER errors) and further explanation is unnecessary.

The operating system errors described in paragraph C.4 are more serious and deal with the viability of the operating system itself. Recovery from errors of this type can be difficult and time consuming. A thorough understanding of the operating system and the application programs can be very helpful. In some cases, assistance from a DIGITAL Software Specialist may be necessary. Operating system errors are divided into two groups: those that are caused by the operating system in response to the operation of a specific job, and Garbage Collector errors, which are system-caused and job-independent.

5.1.1 Programming Errors

The paragraphs which follow provide information on Programming Errors for which recovery is most difficult, and supplement the information in Appendix C.

5.1.1.1 NAKED

This error indicates an illegal naked access and is often caused by a user-created, time-sharing conflict in the application software. For example, assume program A KILL's parts of a global that program B is currently referencing via naked syntax. When this happens, the system's Garbage Collector clears user B's disk pointers, thereby preventing further naked references and causing the NAKED message to be output to him.

5.1.1.2 DKSER

This error normally indicates there was an illegal use of the VIEW command by attempting to:

1. Access a disk block number less than 1.
2. Access a disk block number higher than is available on the disk.
3. Reference a block on a nonexistent disk drive.

DKSER can also occur, however, during a global reference. This is usually caused by the detection of an invalid disk block number in a pointer. This means that either the user (via VIEW) or the operating system itself has altered the pointer incorrectly. The system manager should call his DIGITAL Software Specialist for assistance if the error appears to be the fault of the operating system.

5.1.2 Operating System Errors (non-Garbage Collector)

The paragraphs which follow provide information on Operating System Errors for which recovery is most difficult, and supplement the information in Appendix C.

5.1.2.1 DKDER

This error is a variation of the DKHER error described below. Though the cause of this error is the same as for the DKHER error (disk hardware failure), the circumstances of its occurrence are different. DKDER is generated during an output operation to the disk that was initiated by the operating system (as opposed to the user). DKDER occurs when the system determines that the current disk block must be written out (e.g., user's program was swapped out, a continuation block was needed, a different global level was accessed).

5.1.2.2 DKHER

This error occurs when a disk hardware error is detected while a job is attempting to read or write disk data. Though the request is accepted, it cannot be completed. Hardware errors can be caused by the following: head crash, disk off line, a write-protect violation, inability to read previously written block, or power failure.

5.1.2.3 PLDER

This message indicates that the program being CALLed, STARTed, OVERLAYed, or LOADed was FILEd incorrectly by the system. The user must refile a backup copy of the program. Recurrence of the error indicates a disk data structure problem.

5.1.2.4 SWAP

Whenever a job's time slice expires, the job becomes inactive, and the system starts up another job. During the job's inactivity, the contents of the system stack are saved in the inactive job's partition overhead area (see MUMPS-11 Programmer's Guide, Appendix G). This error occurs when the system stack cannot fit on the user stack because of insufficient partition space.

5.1.2.5 SYSDG

This error is caused by the detection of a discrepancy between the memory resident Disk Storage Allocation Table (DSKSAT) and the disk resident bit maps. This usually results from indiscriminate or disorganized system shutdown procedures. The DBT program (Chapter 6) should be run to correct the problem.

5.1.3 Garbage Collector Errors

The Garbage Collector is a routine which resides within the operating system. It is used to recover deallocated disk storage blocks for the system. As a part of its operation, the Garbage Collector performs error checking to determine the validity of its operations. Appropriate error messages are output in the form GARBn, where 'n' is the message number. All error messages are output to the console terminal. Because these errors can have a serious impact on the system, output of these error messages takes precedence over any other output operation at the console terminal. Further, the occurrence of any of these errors indicates that the integrity of the data base is questionable. The system manager should initiate appropriate data base recovery procedures.

There are two types of errors, as shown below, Read Errors and Write Errors:

Read Errors:

GARB0	Error on reading data block[1]
GARB2	Error on reading bit map
GARB4	Block being collected is itself a bit map. DANGEROUS ERROR

Write Errors:

GARB1	Error while writing a data block
GARB3	Error while writing a bit map

[1] This error will occur when starting the system via the ROM bootstrap loader if the system disk is set to WRITE LOCK.

When Read Errors occur, normal system operation continues after the error message is output. The console terminal user, however, will have to re-initialize this terminal by typing CTRL/C (if a MUMPS terminal operation was in progress) or ALTMODE (if a Mini-ODT operation was in progress). If the error is a Write Error, the disk is locked out and is unavailable for further use. To regain use of the disk, clear the Disk I/O Bound Queue using Mini-ODT as shown below:

1. Set SWITCH REGISTER switch 0 to 1

2. Examine the contents of location 54(8):

   <u>54 (ALTMODE)</u>/NNNNNN <u>(ALTMODE)</u>

3. Using the contents of 54 as an address, examine the contents of NNNNNN:

   <u>NNNNNN (ALTMODE)</u>/MMMMMM <u>(ALTMODE)</u>

4. Add 32(8) to MMMMMM yielding XXXXXX. Open byte location XXXXXX and change its contents to 0.

   <u>BXXXXXX (ALTMODE)</u>/377 0 <u>(ALTMODE)</u>

NOTE

> If the Garbage Collector is reporting errors continuously at a rapid rate, it must be disabled before proceeding with recovery procedures. To do this, set SWITCH REGISTER switch 4 to 1 (raised position).

5.1.4 Errors Requiring DEC Assistance

The Programming and Operating System Errors shown below often require the assistance of a DIGITAL Software specialist:

- DBDGD
- DKSER
- DSKDG
- GARB4
- SYSER

When these errors occur, the user should run both the Disk Block Tally (DBT) and Caretaker Reporter (KTR) programs. The reports produced by DBT and KTR should be saved for use by DEC personnel in diagnosing the problem. After the programs have completed operation, the system should be shut down by using the System Shutdown Program (SSD). Refer to Chapter 6 for descriptions of the System Utility Programs.

5.2 MASS STORAGE HARDWARE ERRORS

In addition to reporting the hardware errors discussed in Section 5.1, and Appendix C, Section C.4, MUMPS also maintains an internal log of some of these errors. This includes running totals for DECtape, magtape, disk and bus errors. Examples of logged errors are: disk hardware failure, DECtape OFF LINE, magtape read error, etc. This data is stored in locations that are part of the System Table (Table 5-1). The occurrence of these errors is normally transparent to the individual jobs (users), because the operating system always makes a number of attempts to successfully perform the operation that caused the error. Each unsuccessful attempt is logged in the System Table by incrementing the appropriate table entry.

The number of retries for any particular device is fixed (refer to Chapter 3 in the MUMPS-11 Programmer's Guide). Once this limit is exceeded, the appropriate operating system error message is generated (see Section C.4). In addition to mass storage error data, the system also maintains a count of the number of illegal hardware interrupts. The occurrence of the error messages may be due to:

1. Operation of a device that is not logically part of the current system (not SYSGENed into the system)

2. Interrupt conditions explicitly ignored by MUMPS-11 (see Section 5.3)

3. An unstable hardware condition

The hardware error statistics contained in the following System Table are not automatically reported to the system manager. This task is performed by the Caretaker System Utility program (CTK) and the Caretaker Reporter program (KTR), described in Chapter 6. The CTK program is designed to run as a background job in its own 0.75K word partition. It operates at 5-second intervals taking error statistics from the System Table and storing them as chronological data in the special ↑SYS system global described in Appendix F. The KTR program reports the data stored in ↑SYS by CTK and provides a chronological report as the system's hardware error statistics. It is strongly suggested that the system manager include CTK and KTR as part of his application system. Though these errors are not fatal to the operation system, they can cause user program termination. A high frequency of these errors indicates a general hardware problem that should be remedied. KTR should be run periodically to monitor both the system error activity and the size of the ↑SYS global.

Table 5-1
System Table Error Statistics

Mnemonic	Location	Use
ILGINT	$V($V(44)+16)	count of illegal interrupts
MTERRS	$V($V(44)+102)	count of magtape errors
DTERRS	$V($V(44)+104)	count of DECtape errors
RKERRS	$V($V(44)+106)	count of RK11 errors
	$V($V(44)+108)	disk address of latest error
	$V($V(44)+110)	always 0
	$V($V(44)+112)	error (status word)

Table 5-1 (Cont.)
System Table Error Statistics

Mnemonic	Location	Use
RFERRS	$V($V(44)+114)	count of RF11 errors
	$V($V(44)+116)	low-order of disk address of latest error
	$V($V(44)+118)	high-order of disk address of latest error
	$V($V(44)+120)	error (status word)
RPERRS	$V($V(44)+122)	count of RH11/RJP04,05,06 errors
	$V($V(44)+124)	address of latest errors bits 0-3 : sector # bits 8-12: track #
	$V($V(44)+126)	bits 0-8 : cylinder # bits 10-12: drive #
	$V($V(44)+128)	error (status word)
		TJU16 Hardware Register
HTCST	$V($V(44)+192)	control status register
HTCCM	$V($V(44)+194)	command register
HTCWC	$V($V(44)+196)	word count register
HTCBA	$V($V(44)+198)	bus count register
HTCDT	$V($V(44)+200)	data register
		TJU16 Hardware Register
MTCS1	$V($V(44)+220)	control status 1 register
MTCS2	$V($V(44)+222)	control status 2 register
MTDS	$V($V(44)+224)	drive status register
MTER	$V($V(44)+226)	error register
MTTC	$V($V(44)+228)	tape control register
PARERR	$V($V(44)+232)	Group 0 11/70 cache error count
	$V($V(44)+233)	Group 1 11/70 cache error count
	$V($V(44)+234)	11/70 parity LOW error register
	$V($V(44)+236)	11/70 parity HIGH error register
	$V($V(44)+238)	11/70 parity memory system error register
	$V($V(44)+240)	11/70 parity job hung #
RMERRS	$V($V(44)+306)	count of RK611/RK06 errors
	$V($V(44)+308)	address of latest error bits 0-3: sector # bits 8-12: track #
	$V($V(44)+310)	address of latest error bits 0-8: cylinder # bits 10-12: drive #
	$V($V(44)+312)	error register (status word)

5.3 ERRORS THAT CAUSE THE SYSTEM TO CRASH

Errors in this category are often the most serious that can occur in the MUMPS system. These are errors that cause the termination of all MUMPS job processing. There are two types of system crashes: soft crashes and hard crashes. A soft crash is caused by one of several hardware-detected error conditions. This results in a shutdown of the operating system that leaves the processor running and Mini-ODT operable. A hard crash is caused by an undetected error condition and results in a processor halt. Occurrence of either type of crash indicates an abnormal hardware or software operation that may require the assistance of DIGITAL service personnel. Anyone diagnosing errors of this type should have a working knowledge of MUMPS internal structure and operation, as well as the PDP-11 hardware. In particular, the individual must comprehend the PDP-11's interrupt processing system and its utilization of low memory for interrupt vectors.[1]

5.3.1 MUMPS Interrupt Vector Utilization

MUMPS uses the TRAP instruction for all MUMPS program and system errors except for GARB0 through GARB4. TRAPs employed are 104776 (octal) through 104732 (octal), and correspond to returned $E values of 0 through -.38. The TRAP instruction vector (address 34 (octal)) is set up to transfer control to the MUMPS Error Processor in the Interpreter. Furthermore, memory location 0 is set up to contain the TRAP instruction (104735(octal)) for the NOTSY error. Undesired transfers of control to location 0 generate an operating system error rather than create a crash condition. Specifically, this scheme allows system crashes to be avoided when system object modules or MACRO-11 global references (i.e., assembly language context) are missing. The LINK program (during system build) stores a zero in the undefined global reference. For example, referencing a DECtape (e.g., A 55) when the DECtape driver module is not part of the system, produces a NOTSY error.

Other low memory addresses with special use include:

Octal Locations 40,42 — Contain a JUMP to the system restart code in Exec.

Octal Locations 44,46 — Contain a JUMP instruction to transfer control to the system's "Write Crash Block" Routine.

Octal Location 54 — Contains a pointer to the base of the System Table.

Octal Location 56 — Contains a "BR ." MACRO-11 instruction which implements a MUMPS soft crash condition. This enables the use of Mini-ODT after a system crash. All crash errors, at the completion of their processing, transfer control to location 56 (octal).

5.3.1.1 Interrupt Vectors Ignored by MUMPS

Illegal interrupts are logged, but otherwise ignored by MUMPS. All unused interrupt vectors, i.e., low memory addresses up through location 700 (octal) with no associated devices, are set up to be ignored by MUMPS should an interrupt occur. These interrupt vectors are set up to contain a pointer to the hardware error statistics routine in the operating system. This routine increments the illegal interrupt entry register in the System Table (Table 5-1). Normal system operation continues after the error condition is logged.

[1] This information can be found in the applicable PDP-11 Processor Handbook.

5.3.1.2 Interrupts Processed by MUMPS

The following interrupts are processed by MUMPS and result in a soft crash:

- Bus Timeout Error
- Illegal Instruction Error
- Memory Parity Error
- Segmentation Error
- Power Fail

The first four interrupts above result in a small amount of system checking. The hardware priority is then lowered (to enable Mini-ODT) and control is transferred to location 56 (octal). In addition, a power failure causes an immediate trap to 56 (octal).

5.3.1.3 Hard Crash Interrupts Set Up by MUMPS

The following interrupts are set up to HALT the processor at the interrupt location, should an interrupt occur.

- EMT Trap Instruction
- Programming Interrupt Request
- Op Code 000003 with T-bit set
- Floating Point Exception

Under normal MUMPS operation these interrupts should never occur. If one of the above does occur, the CPU or MUMPS software is not functioning properly. Refer to procedures in Section 5.3.3.

5.3.2 Soft Crash

As mentioned earlier, a soft crash is caused by hardware or software errors that are detected by the PDP-11 processor. When a soft crash occurs, all MUMPS operations, except Mini-ODT, cease; the Control Console RUN lamp remains ON, and the processor ADDRESS REGISTER displays 56 (octal).

Location 56 contains a "BR ." MACRO-11 instruction that causes the processor to execute continuously that single instruction. This causes the job currently running (in the RUNQ) to be lost and the job performing disk I/O (DSIOBQ) to be suspended.

NOTE

> Users with PDP-11/10 processors must depress the HALT switch on the console to observe whether a Soft Crash is in effect. Pressing the CONT switch resumes operation.

When a soft crash occurs, the system manager should first HALT the processor, then save the Crash Block. The Crash Block is a reserved disk/memory area that can contain vital system crash data. This includes the contents of all system queues, general registers RO through R7, and the System Stack. This Crash Block data is obtained by activating a special save routine embedded within MUMPS. This information facilitates in determining the cause of the crash.

The save operation is initiated by starting the processor at location 44 (octal), as follows:

1. Press HALT
2. Set the SWITCH REGISTER switches to 44 (octal)
3. Press LOAD ADRS
4. Raise HALT
5. Press START

When the Crash Block is saved, the processor returns control to location 56. The system manager should next attempt to resume MUMPS operation by performing a system restart (see Section 4.5.1). This will allow all jobs, except the job in the RUNQ, to resume operation without data loss. The job in the RUNQ will be lost and the I/O devices owned by it at the time of the crash will be locked out of the system. Once the system is restarted, the SIF or SS System Utility Programs should be run to obtain the current system status data. Thereafter the DMP program should be run to obtain a Crash Block Analysis Report. The reports obtained from these programs should be saved to allow error diagnosis by a DIGITAL Software Specialist, should his services be required. Lost jobs and locked-out devices can be restored by reloading the system from the disk via the ROM bootstrap loader.

5.3.3 Hard Crash

A hard crash is the unconditional termination of all processor operations. This type of error condition occurs when the system inadvertently causes the processor to execute a HALT instruction (machine code 000000). This condition is caused by a hardware or software malfunction that results in the modification of the contents of one or more memory locations that contain the operating system. This error condition is identified by the abrupt termination of all system operations and the extinguishing of the processor's RUN light. Possible causes of this error condition include:

- Improper use of VIEW command
- Hardware failure
- Operating System failure
- HALT Console Switch being pressed
- Interrupts described under Section 5.3.1.3.

When a hard crash occurs, the system manager should obtain the following information for use in diagnosing the cause of the error.

- The address displayed in the processor's ADDRESS REGISTER (lights)

- Contents of registers R0 through R6 (word locations 777700 through 777706 (octal))

- Contents of the first five memory word locations (ascending order) beginning with the address contained in R6

Once this data is recorded, the system manager may wish to attempt either to restart the system or to reload the system as discussed in Chapter 4.

CHAPTER 6

SYSTEM UTILITY AND TEST PROGRAMS

6.1 INTRODUCTION

The System Utility and Test Programs are a group of privileged access utility programs, available only to the system manager via the System UCI (UCI #1). These programs provide the facilities necessary for the day-to-day maintenance, modification and basic testing of the MUMPS operating system. The programs reside in the Program Directory of the System UCI (UCI #1) and, unlike Library Utility Programs, these programs can only be accessed from terminals logged in under the System UCI.

In general, program operation can be initiated in one of two ways:

1. A program can be loaded at log-in time by entering the name of the program after the UCI code. For example:

   Typing: 'SYS:NAM' in response to the sign-in directive loads and starts the program called NAM from the UCI SYS.

2. A program can be CALLed after log-in. For example:

   Typing: 'C␣NAM' also loads and starts the program called NAM.

The programs are completely interactive and prompt the desired set of responses. Unless otherwise specified, all user responses are terminated with a Carriage RETURN.

General error checking is performed by all programs. If a user response is not one that the program expects, most of the programs will output the range or type of expected responses and then retype the question. If the user detects an error before the Carriage RETURN is typed, individual characters can be deleted using RUBOUT, or the entire line can be deleted with a CTRL/U. System errors, such as those caused by defective hardware or a device not ready, are terminal and end the program's operation. Generally it is necessary to restart the entire program.

Several of the programs (DBT,MSP,RKC,SDP,%GP) can run only if no other job is running. In such cases, the program will instruct the user to set one or more SWITCH REGISTER switches to effect a system shutdown.

When the program has determined that all other users have finished, it continues operation. Some of these programs can be operated only from the console terminal; if loading is attempted from other terminals, an appropriate message is output.

6.2 SYSTEM UTILITY PROGRAM DESCRIPTIONS

The paragraphs that follow describe the functional and operational characteristics of the System Utility programs. Table 6-1 lists and summarizes the operation of each.

Table 6-1
System Utility Program Summary

Program Name	Description
BCS	Broadcast Allows the system operator/manager to send a message to all or selected active user terminals.
CTK	System Caretaker Monitors system hardware error status and stores the data in the ↑SYS system global.
DAT	Date Set Sets the $D System variable to the current date.
DBT	Disk Block Tally Calculates the number of disk blocks available for each disk unit, (logical and physical), updates Disk Storage Allocation Table, tallies and reports errors.
DMP	Disk Block Dump Provides octal dumps of specified disk blocks or a report on the contents of the system's crash block.
KTR	Caretaker Reporter Interprets and reports system error data stored by CTK in the ↑SYS System Global.
MSP	Modify System Parameters Alters UCI codes, terminal device, multiplexer, and timing and disk parameters, magtape default mode.
RKC	RK Disk Copy Used with RK05 distribution to physically copy one RK05 cartridge to another RK05.
RSJ	Restore Job Restores wait queue and I/O hung jobs to the system.
RST	Restore Terminal Releases terminal devices owned by a job.

Table 6-1 (Cont.)
System Utility Program Summary

Program Name	Description
SDP	Sequential Disk Processor Space Allocator/Deallocator Allocates and deallocates disk space used by the system's Sequential Disk Processor.
SIF	System Information Provides the partition starting address and system status (calls SS program) and system partition size/assignments for each job.
SS	System Status Provides information about the current system users, their job status and device ownership.
SSD	System Shutdown Instructs the operator how and when the system can be HALTed
STU	System Startup Initializes system after a bootstrap start from the disk.
TIM	Time Set Sets the $T system variable to the current time.
%GP	Global Place Positions a global on a specified disk and cylinder.

6.2.1 BCS - Broadcast

The Broadcast program allows the system operator to send messages to all or specified active terminals (excluding CPU-CPU devices) from the console terminal. The user can broadcast either a message of his choice or the standard warning message:

```
<<SYSTEM SHUTDOWN IN n MINUTES--PLEASE LOG OFF>>
```

Once running, BCS types the time, then requests the user to specify either the S option (shutdown message) or the M option (operator-specified message). It then requests the user to select the A option (output message to all terminals) or the S option (output message to specified terminals). If output is to specific terminals, BCS reques<s the numbers of the terminals to which output is to be directed. Each number must be separated from the next by a comma (,).

Once the options and terminals are selected, BCS either outputs the shutdown message, if selected, or types two left angle brackets on the console terminal and waits for the user to type in a message.

6.2.2 CTK - Caretaker

The Caretaker is a program that retrieves and logs the data contained in various PDP-11 hardware error status registers. CTK stores the data starting at node 18 of the special system global ↑SYS that was created by SYSGEN.

Appendix F shows the complete structure of ↑SYS. Data collected by CTK includes:

- Date and time when the file was begun.
- Date and time of the last entry.
- Number of illegal interrupts.
- Number of bus errors.
- DECtape error status.
- Magtape error status (either TM11 or TJU16).
- Disk system error status.

CTK requires at least a 0.75K-word partition for operation. It consists of the main program CTK and the subprogram CK0. CTK is normally STARTed from the console terminal. A typical command string to begin operation is:

```
U⎵0⎵S⎵CTK(6)⎵H⎵1⎵A⎵0↵

 1      2     3   4
```

1. Release principal I/O device for use by CTK.
2. START CTK in a 0.75K partition.
3. HANG for one second to allow CTK to finish its I/O to the principal I/O device.
4. Reassign principal I/O device (CTK does no further I/O).

Once running, CTK allows the user to specify either the I (Initialize) or C (Continue) option. The I option causes CTK to wipe out previously saved ↑SYS data before running. The C option instructs CTK to continue entering data following the data entered the last time CTK ran. Once STARTed, CTK runs on 5-second intervals. The data saved in ↑SYS is interpreted and reported by using the KTR (Caretaker Reporter) program described in paragraph 6.2.6.

Because CTK is continually adding data to ↑SYS, the user must periodically reinitialize ↑SYS to avoid using excessive disk space. This can be done by using the I option. The old data can be saved by using the %GP Library Utility Program described in the MUMPS-11 Programmer's Guide.

NOTE

If SWITCH REGISTER switch 6 is used to shut down the system, CTK must be restarted.

6.2.3 DAT - Set Date

This program allows the user to set the $D System Variable to the current calendar date. When it is loaded and running, DAT requests the user to enter the date. The user should enter the date using the mm/dd/yy format (e.g., 7/4/76 = July 4 76). Once entered, DAT outputs the new contents of $D (via the %D Library Utility Program) for user verification. DAT performs error checking on input to determine correct value range and form.

6.2.4 DBT - Disk Block Tally

DBT is a program that assists the user in determining the integrity of the MUMPS disk data structure. DBT consists of the main program DBT and the subprogram TAL and KIL. The program provides diagnostic data, remedial assistance and storage allocation information. SWITCH REGISTER switch #4 must be set to 0 to allow the Garbage Collector to run. Specifically, it:

- Verifies each bit map's summary count of available blocks by comparing it to the calculated number of available blocks.
- Reports the summary count, the calculated count, and the number of blocks allocated for sequential disk processor use, as well as the error status for each bit map in each disk system.
- Removes bit maps with count discrepancies from the Disk Storage Allocation Table (DSKSAT) to prevent propagation of additional errors.
- Updates the memory resident Disk Storage Allocation Table (DSKSAT).
- Enables the system's Garbage Collector routine to run which in turn causes a new image of the DSKSAT to be saved on the system disk.

DBT allows the user to specify both the device on which its report is printed and whether the long or short form report is desired. The user can request DBT to tally any or all disk units in the system. Figure 6-1 shows a typical DBT report. DBT does not allow other jobs to be running during its operation.

```
>C DBT


               DISK BLOCK TALLY
DATE   29 AUG 73                TIME   7:55AM

ENTER TERMINAL # (1,4-19,64-111) FOR TALLY OUTPUT: 1
RAISE SWITCH REGISTER SW #6

DO YOU WANT LONG FORM (RESULTS PER MAP) TALLY OUTPUT,
OR SHORT FORM (RESULTS PER UNIT)? TYPE L OR S. L
DO YOU WANT TO TALLY ALL DISKS (Y OR N)? Y




       DISK   MAP SUMMARY    CALCULATED    AVAILABLE      # OF
       /UNIT     COUNT     GLOBAL BLOCKS  SDP BLOCKS     ERRORS

MAP0   RK/0      2039         2039           0
MAP1   RK/0      0            0              2400
MAP0   RK/1      0            0              2400
MAP1   RK/1      0            0              2400
       RK        2039         2039           7200          0



DISK BLOCK TALLY FINISHED.
```

Figure 6-1 Example of DBT Output

6.2.5 DMP - Disk Block Dump

This program allows the user to dump specified blocks of any disk unit onto the line printer. Further, DMP can also perform an analysis of the system's crash block (described in Chapter 5) and output a special report describing the status of the MUMPS-11 System at the time the crash occurred. DMP consists of the main program DMP and the subprogram DCB.

The program begins operation by requesting the user to specify the desired option:

```
                                     | Y↵ |
DUMP CRASH BLOCK (Y OR N)   ?        | N↵ |
                                     |  ↵ |
```

The user can type:

Y↵ to request a crash block analysis report to be output to a specified device

N↵ to request that specified blocks be dumped in octal to the line printer

↵ to request that the crash block be dumped in octal to the line printer

If N is typed, DMP requests the user to specify the desired mode:

Mode	Description
A	ASCII mode causes each byte to be output as an ASCII character
B	Byte mode causes the octal value of each byte to be output
H	Hexadecimal mode causes each word to be output in hexadecimal radix
O	Octal mode causes the octal value of each word to be output

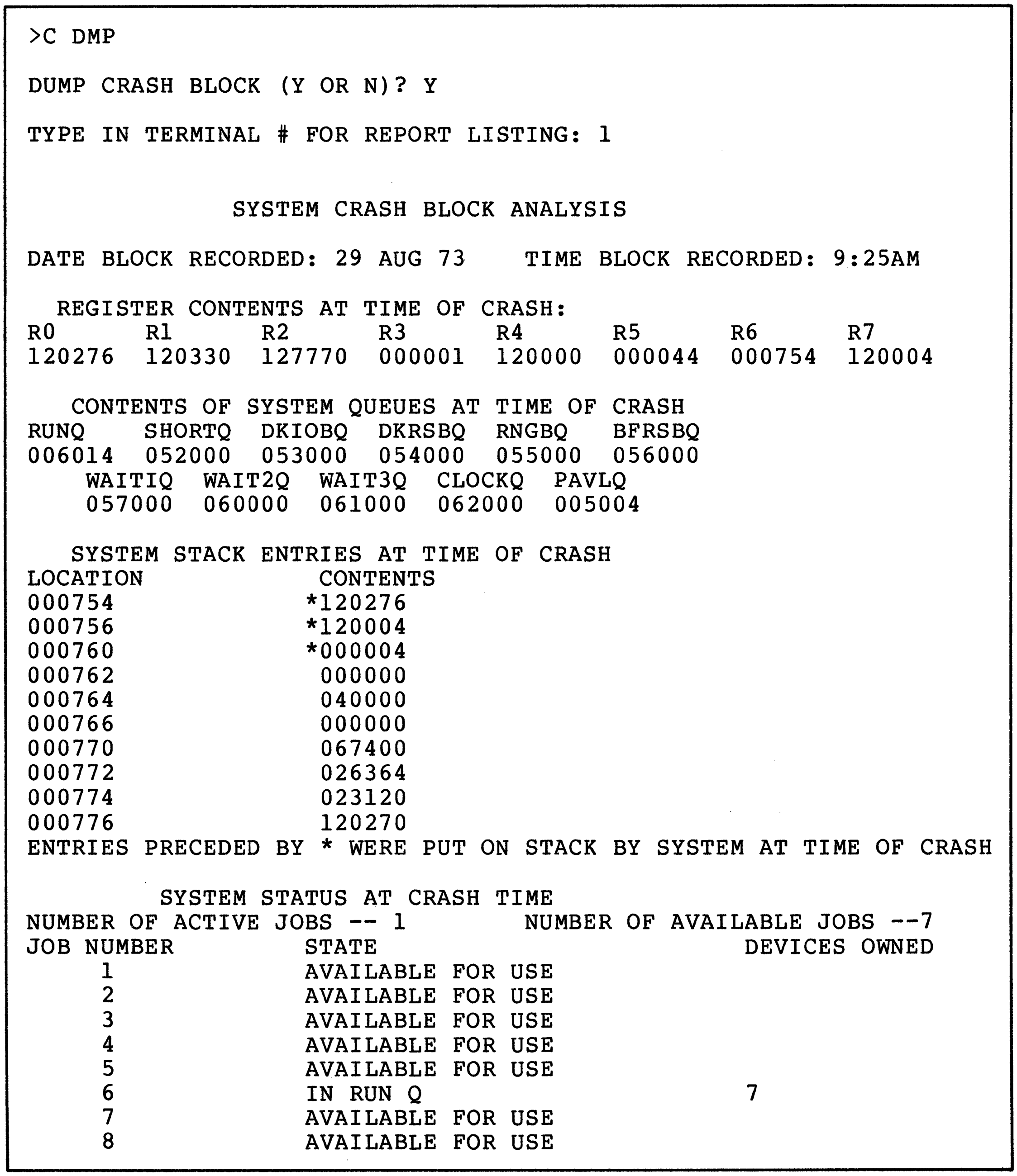
Any one or all of the modes can be specified in any combination. When more than one output mode is selected, commas must be used to separate each mode option. If no response is made within 20 seconds, the crash block is automatically output to the line printer in octal mode. Figure 6-2 shows a typical session requesting a crash block analysis report. Figure 6-3 shows a typical session requesting a disk block dump of RK disk unit 0, block 0 in all modes. Figure 6-4 shows a typical session requesting a memory dump from memory locations 1200 to 1300 (octal). The relative block address (octal) precedes each line of 16 words.

Before branching to location 56 on an error (location 56 contains a "BR ." instruction), the system usually saves certain information on the system stack. This information consists of the current instruction address when the trap occurred, the then-current PSW, and, finally, a sequence of words which depends on the type of error that occurred.

Following are examples of some errors for which a column of words are stacked:

11/70 parity error:	Address when interrupted old PSW memory system error register low address error high address error
11/45 parity error: 40	address when interrupted old PSW bank number of parity error
bus errors:	address when interrupted old PSW address when interrupted

Note that on a parity error, the last stacked word will be a small number (<16), whereas a bus error will result in a much larger last word, which is also the same as the second previous word.



```
>C DMP

DUMP CRASH BLOCK (Y OR N)? Y

TYPE IN TERMINAL # FOR REPORT LISTING: 1


              SYSTEM CRASH BLOCK ANALYSIS

DATE BLOCK RECORDED: 29 AUG 73    TIME BLOCK RECORDED: 9:25AM

  REGISTER CONTENTS AT TIME OF CRASH:
R0      R1      R2      R3      R4      R5      R6      R7
120276  120330  127770  000001  120000  000044  000754  120004

   CONTENTS OF SYSTEM QUEUES AT TIME OF CRASH
RUNQ    SHORTQ  DKIOBQ  DKRSBQ  RNGBQ   BFRSBQ
006014  052000  053000  054000  055000  056000
    WAITIQ  WAIT2Q  WAIT3Q  CLOCKQ  PAVLQ
    057000  060000  061000  062000  005004

   SYSTEM STACK ENTRIES AT TIME OF CRASH
LOCATION            CONTENTS
000754             *120276
000756             *120004
000760             *000004
000762              000000
000764              040000
000766              000000
000770              067400
000772              026364
000774              023120
000776              120270
ENTRIES PRECEDED BY * WERE PUT ON STACK BY SYSTEM AT TIME OF CRASH

         SYSTEM STATUS AT CRASH TIME
NUMBER OF ACTIVE JOBS -- 1        NUMBER OF AVAILABLE JOBS --7
JOB NUMBER         STATE                          DEVICES OWNED
     1             AVAILABLE FOR USE
     2             AVAILABLE FOR USE
     3             AVAILABLE FOR USE
     4             AVAILABLE FOR USE
     5             AVAILABLE FOR USE
     6             IN RUN Q                       7
     7             AVAILABLE FOR USE
     8             AVAILABLE FOR USE
```

Figure 6-2 Example of DMP's Crash Analysis Report

```
>C DMP

DUMP DISK BLOCKS OR MEMORY (D OR M)?D

DUMP CRASH BLOCK (Y OR N)? N

ANSWER THE FOLLOWING APPROPRIATELY:
DISK? RK  UNIT? 0  BLOCK? 0

TYPE MODES (A[SCII],B[YTES],H[EX],O[CTAL]), SEPARATED BY COMMAS:A,B,H,O
OUTPUT DEVICE #: 1
OCTAL CONTENTS OF DISK BLOCK # 0        MODES: O          RJ DISK,  UNIT O,
000000:    -   @     -   @   -   A   -   @   -   B    -   C        -   -   -
         021 300   013 340 035 301 000 300 020 102  025 303  377 000 024 021
            11C0      0BE0    1DC1    00C0    1024     15C3     FF00    1411
          010700    005740  016701  000300  010102   012703   177400  012021

000020:    -   -     -   ]   -   R   -   -   -   -    E   B    -   J   -   -
         012 203   005 375 000 162 000 030 000 005  145 302  000 312 024 200
            0A83      05FD    0072    0018    0005     65C2     00CA    1480
          005203    002775  000162  000030  000005   062702   000312  012200

000040:    -   -     -   -   -   -   -   -   -        -   H    -       -   P
         024 201   024 221 024 211 024 210 022 240  213 310  200 376 013 320
            1481      1491    1489    1488    12A0     8BC8     80FE    0BD0
          012201    012221  012211  012210  011240   105710   100376  005720

000060:    -   -     -   -   -   ^   %   R   -   -    -   -    -   C       T
         200 002   000 000 001 376 045 322 000 003  003 030  025 303 377 164
            8002      0000    01FE    25D2    0003     0318     15C3    FF74
          100002    000000  000776  022722  000003   001430   012703  177564
```

Figure 6-3 Example of DMP'S Disk Block Dump

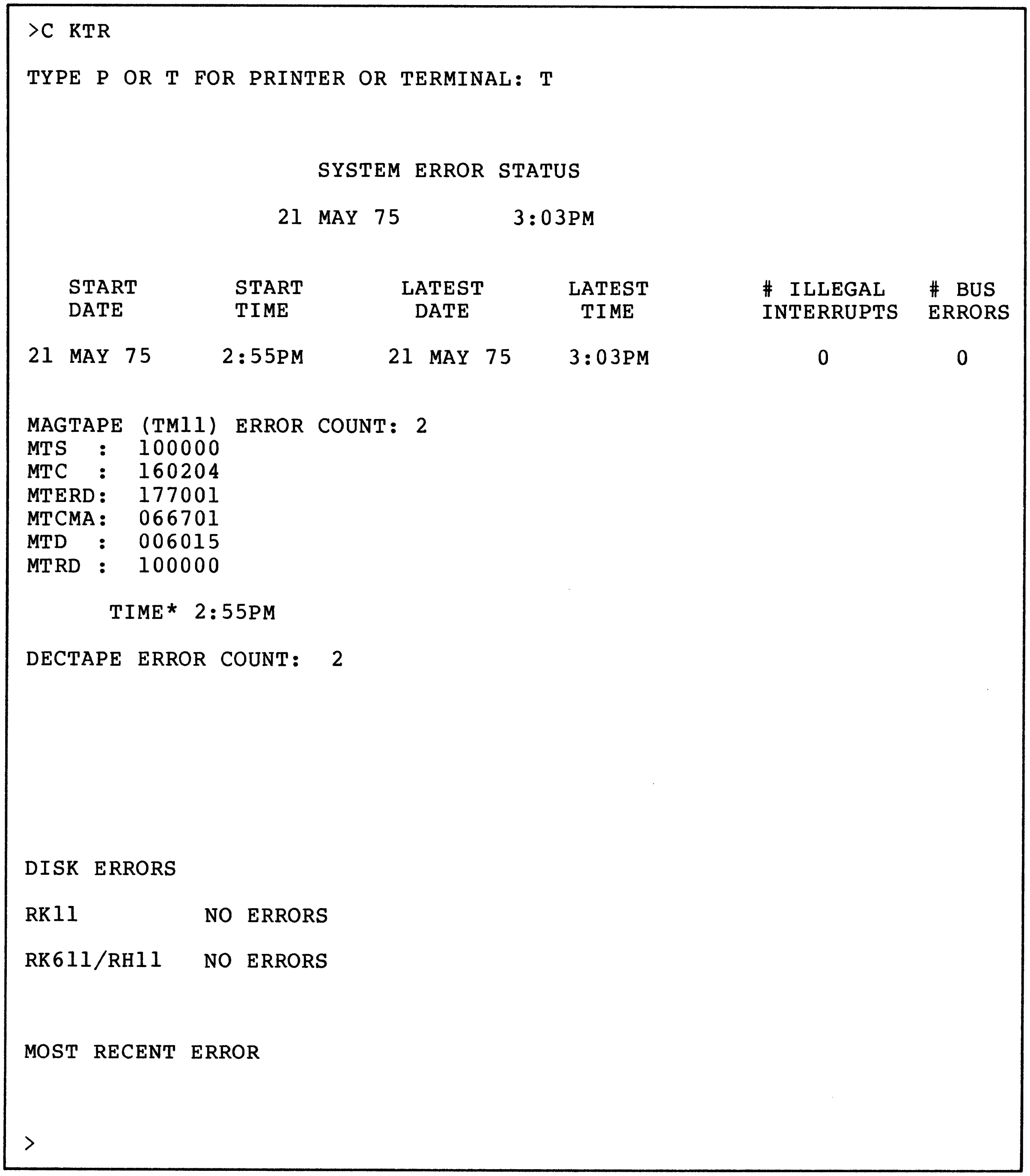
```
>C DMP
DUMP DISK BLOCKS OR MEMORY (D OR M)? M
LOWER LIMIT(OCTAL)= 1200
UPPER LIMIT(OCTAL)= 1300
TYPE MODES (A[SCII],B[YTES],H[EX],O[CTAL], SEPARATED BY COMMAS:O
OUTPUT DEVICE #: 1
OCTAL DUMP OF MEMORY FROM 1200(OCTAL) TO 1300(OCTAL)          MODES: O
001200:  002360  002236  001676  107260  002076  111020  111126  112000
001220:  000000  000000  077540  105060  106060  177400  036017  002003
001240:  000404  001057  113212  001200  112000  116000  172540  177400
001260:  140060  160000  000154  177540  111400  000117  005140  047507
001300:  000104  000000  000000  000000  000000  000000  000000  000000
END OF DUMP
```

Figure 6-4 Example of DMP'S Memory Dump

6.2.6 KTR - Caretaker Reporter

The Caretaker Reporter interprets and formats the error data stored by the CTK program in the special system global ↑SYS (see Section 6.2.2).

The program allows the user to select either the line printer or the terminal as the output device. Figure 6-5 shows a typical KTR report. KTR operates only if nodes 18 through 60 of the ↑SYS global (described in Appendix F) contain data. This means that CTK must have run at some time prior to KTR's operation.



```
>C KTR

TYPE P OR T FOR PRINTER OR TERMINAL: T


                     SYSTEM ERROR STATUS

                  21 MAY 75        3:03PM


   START       START       LATEST      LATEST        # ILLEGAL   # BUS
   DATE        TIME         DATE        TIME         INTERRUPTS  ERRORS

21 MAY 75     2:55PM      21 MAY 75    3:03PM            0          0


MAGTAPE (TM11) ERROR COUNT: 2
MTS   :  100000
MTC   :  160204
MTERD:   177001
MTCMA:   066701
MTD   :  006015
MTRD :   100000

      TIME* 2:55PM

DECTAPE ERROR COUNT:  2






DISK ERRORS

RK11         NO ERRORS

RK611/RH11   NO ERRORS


MOST RECENT ERROR


>
```

Figure 6-5 Example of KTR Report

6.2.7 MSP - Modify System Parameters

The MSP Program allows the system manager/operator to alter many system parameters without having to perform a new system generation via SYSGEN. The program consists of the main program MSP and subprogram modules MSU, %MU, MUX, MBP, MMD, and KIL. MSP permits the user to modify any of the four basic system parameter groups:

- UCI Data and PAC.
- Terminal Device Data (type, speed, tied or untied, etc.).
- Multiplexer Line Conditioning (parity, speed, etc.).
- Basic System Parameters (time slice, stack limit, line frequency, and disk control).
- Magtape Default Mode.

The program can be run only at the console terminal. It does not allow other users to be running at the same time. MSP performs part of the system generation process; detailed descriptions of its questions and appropriate user responses can be found in Chapter 3.

When MSP completes its operation, it requests the user to save the newly modified system on its system disk by starting up the system's save routine from location 50(8). Thereafter, the user should make a backup copy of the system disk with the Backup program, (see Chapter 4).

6.2.8 RKC - RK Copy

This utility is used in the RK05 distribution to physically copy one RK05 to another. This physical copy takes about 5 to 10 minutes.

PHYSICAL COPY FROM RK05 DRIVE:

> Enter the appropriate drive number. A null response will result in an exit from this program.

PHYSICAL COPY TO RK05 DRIVE:

Enter the appropriate drive number. An invalid drive number will result in the reasking of the previous question.

LOAD THE DRIVES WITH THE DESIRED PACKS WHEN THEY ARE READY, TYPE <CR>

Mount the desired packs on the selected drives. When they are ready, type a carriage-return.

Figure 6-6 illustrates the use off RKC.

```
RAISE SWITCHES 4 & 6
PHYSICAL COPY FROM RK05 DRIVE: 0

                TO RK05 DRIVE: 1

LOAD THE DRIVES WITH THE DESIRED PACKS
WHEN THEY ARE READY, TYPE:     READY
                               READY

*** FINAL CHECK ***
COPY RK DRIVE 0 TO RK05 DRIVE 1
IF THIS IS CORRECT, TYPE :     YES
                               YES

COPY STARTED

COPY COMPLETED

RESTORE THE MASTER PACKS TO THE APPROPRIATE DRIVES
WHEN THEY ARE READY, TYPE:     READY
                               READY

LOWER SWITCHES 4 & 6
IF DESIRED, CALL CTK TO RESTART THE CARETAKER

>
```

Figure 6-6 Example of RKC Dialogue

The program prints the pertinent information as a final check, and then asks if the information is correct. Any response other than YES will abort the copy without any disc transfers, and control will be returned to the first question. If the user answers YES, the copy is performed. After the copy is completed, the message

ALL DONE

is typed.

NOTE

> The program RKC can only be run from the console. All other users should be signed off before running RKC.

6.2.9 RSJ - Restore Job

This program allows the system manager to restore a job, and the I/O devices owned by it, to the system. RSJ restores jobs that are reported by the System Status Program (SS) to be in a Wait Queue or in an I/O Hung state. Jobs that are "in transition" cannot be restored. The program is particularly useful for releasing jobs that are "lost" during a system crash, jobs that are accidentally STARTed, and jobs that are inactive because a user failed to log out.

Once running, the program identifies itself and asks the user if he wishes to run the SS program to determine the number of the job to be restored. RSJ then either runs SS or requests the number of the job to be restored. When restoration is complete, RSJ prints a message to this effect then halts. Figure 6-7 shows an example of a typical session at the terminal using RSJ.

```
>C RSJ

UTILITY TO RESTORE A USER-SPECIFIED JOB
AND RELEASE ALL DEVICES USED BY THE JOB

DO YOU WISH TO RUN SYSTEM STATUS
TO DETERMINE YOUR JOB NUMBER? (Y OR N): Y



                    SYSTEM STATUS

DATE--10 OCT 74                         TIME--2:55PM
NUMBER OF ACTIVE JOBS -- 3          NUMBER OF AVAILABLE JOBS --6
JOB NUMBER          STATE                         DEVICES OWNED
     1              IN CLOCK Q
     2              AVAILABLE FOR USE
     3              IN RUN Q                          1
     4              TERMINAL I/O HUNG                 4
     5              AVAILABLE FOR USE
     6              AVAILABLE FOR USE
     7              AVAILABLE FOR USE
     8              AVAILABLE FOR USE
     9              AVAILABLE FOR USE



TYPE IN JOB NUMBER TO BE RESTORED:4
YOUR JOB IS BEING RESTORED
>
```

Figure 6-7 Example of RSJ Session

6.2.10 RST - Restore Devices

This program restores to the system the I/O devices that were owned by the jobs in the Disk I/O Bound Queue and the Run Queue. In addition, the program can release any device owned by a job in a normally running system. However, this should be limited to terminal type devices that are not currently active.

DECtape and magtape devices can also be released, but to do so may endanger the device's file structure or data. Further, a job's Principal I/O Device should never be released, for this denies a user the ability to communicate with his job. Once this is done, the job can be stopped only by using the Restore Job Program (RSJ).

After identifying itself, RST begins operation by requesting the user to specify whether an explanation of the program's options is desired.

```
                                                      | L |
LONG FORMAT (EXPLANATION) OR SHORT  (TYPE L OR S)     |   | ↵
                                                      | S |
```

Then the program requests the option to be used:

```
          | ALL |
          | RUN |
OPTION?   | DSK | ↵
          | nn  |
```

The user can type:

ALL — to return all devices owned by the jobs in the Run Queue and Disk I/O Bound Queue at the time of a system crash.

RUN — to return all devices owned by the job in the Run Queue at the time of a system crash.

DSK — to return all devices owned by the job in the Disk I/O Bound Queue at the time of a system crash.

nn — to return one or more devices owned by the job with the number (nn). If this option is selected, RST requests the number(s) of the device(s):

```
             | ALL          |
DEVICE(S)?   |              | ↵
             | nn  {,...}   |
```

The user can type:

ALL — to return all devices owned by the job except its Principal I/O Device.

nn — to return one or more selected devices.

When the program completes the requested operation, it prints the message:

```
          REQUESTED DEVICES RETURNED
```

then calls the System Status Program (SS) to present the current device status. Figure 6-8 shows an example of a typical RST session.

```
>C RST


UTILITY TO RETURN DEVICES TO SYSTEM
LONG FORMAT (EXPLANATION) OR SHORT? (TYPE L OR S) L

OPTION(S) TO BE INPUT:
ALL   - RETURNS DEVICES OWNED BY JOBS IN RUN QUEUE AND DISK I/O
        BOUND QUEUE SUBSEQUENT TO SYSTEM CRASH
RUN   - RETURNS DEVICES AFTER CRASH FROM JOB IN RUN QUEUE AT CRASH TIME
DSK   - RETURNS DEVICES AFTER CRASH FROM JOB IN DISK I/O BOUND QUEUE
        AT CRASH TIME
NN    - RETURNS DEVICES FROM JOB NUMBER (NN)
        PROMPTS A QUESTION AS TO WHICH DEVICES FROM THE STATED JOB
        SHOULD BE RETURNED. TYPE IN DEVICE NUMBERS, SEPARATED
        BY COMMAS (<CR> ENDS LIST), OR TYPE 'ALL' TO GET BACK ALL
        DEVICES OWNED BY THAT JOB.
OPTION? 2

DEVICE(S)? ALL

REQUESTED DEVICES RETURNED



                    SYSTEM STATUS
DATE--10 OCT 74                         TIME--3:02PM
NUMBER OF ACTIVE JOBS -- 4         NUMBER OF AVAILABLE JOBS --5
JOB NUMBER          STATE                          DEVICES OWNED
     1              IN CLOCK Q
     2              TERMINAL I/O HUNG
     3              IN RUN Q                       1
     4              AVAILABLE FOR USE
     5              AVAILABLE FOR USE
     6              AVAILABLE FOR USE
     7              AVAILABLE FOR USE
     8              AVAILABLE FOR USE
     9              TERMINAL I/O HUNG              4


 >
```

Figure 6-8 Example of RST Session

6.2.11 SDP - Sequential Disk Processor Space Allocator and Deallocator

SDP provides the user with the ability to allocate and deallocate disk blocks to be used by the MUMPS Sequential Disk Processor. The program consists of the main program SDP and subprograms SD1, SD2, SD3, KIL and TAL. During allocation, SDP reserves unused global' and program storage blocks available to MUMPS. Blocks that contain global or program data (as opposed to SDP data) cannot be allocated. The deallocation process is the reverse of allocation. SDP allocates and deallocates blocks on disk units in multiples of the number of blocks available to a MUMPS bit map for a given disk type. The user specifies the disk type, the disk unit, the bit map at which allocation or deallocation is to begin, and the number of maps to be processed. The program validates the format and content of each request so that the existing global data base or sequential data base will not be adversely affected.

If a request is granted, the user is told specifically which unit and how many blocks have been affected. If a request is denied, the user is told specifically which MUMPS bit map or sequential data base is the cause. SDP accepts multiple requests for multiple disk types. It provides a summary for each disk type at the end of each session, showing the number of disk blocks affected. Table 6-2 shows the number of blocks available for each type of disk unit.

Table 6-2
Disk Block Allocation

Disk Type	Disk	No. of Units	Maps Per Unit	Blocks Per Map	Blocks Per Unit Type
0	RK11	0-7	2	2400	4800
1	RF11	0-7	1	1024	1024
1	RS11	0-7	1	2048	2048
1	RK611	0-7	51	528	26,928
2	RP11 (RP02,03)	0-7	200/400	200/200	40,000/80,000
3	RH11 (RP04,05,06)	0-7	408	418	170,544

Since SDP's operation involves manipulation of the system's file structure, it cannot be run while other jobs are running. Once loaded, it begins an interactive session with the user. Each command line must be terminated with a Carriage RETURN. Multiple commands can be entered on the same line by separating them with commas.

The correct format for entering a command is:

```
     disk type:  unit no.:  map beginning:  no.  of maps
```

Where: disk type = RK, RF, RS, RM, RP, or RJ

unit no. = 0-7

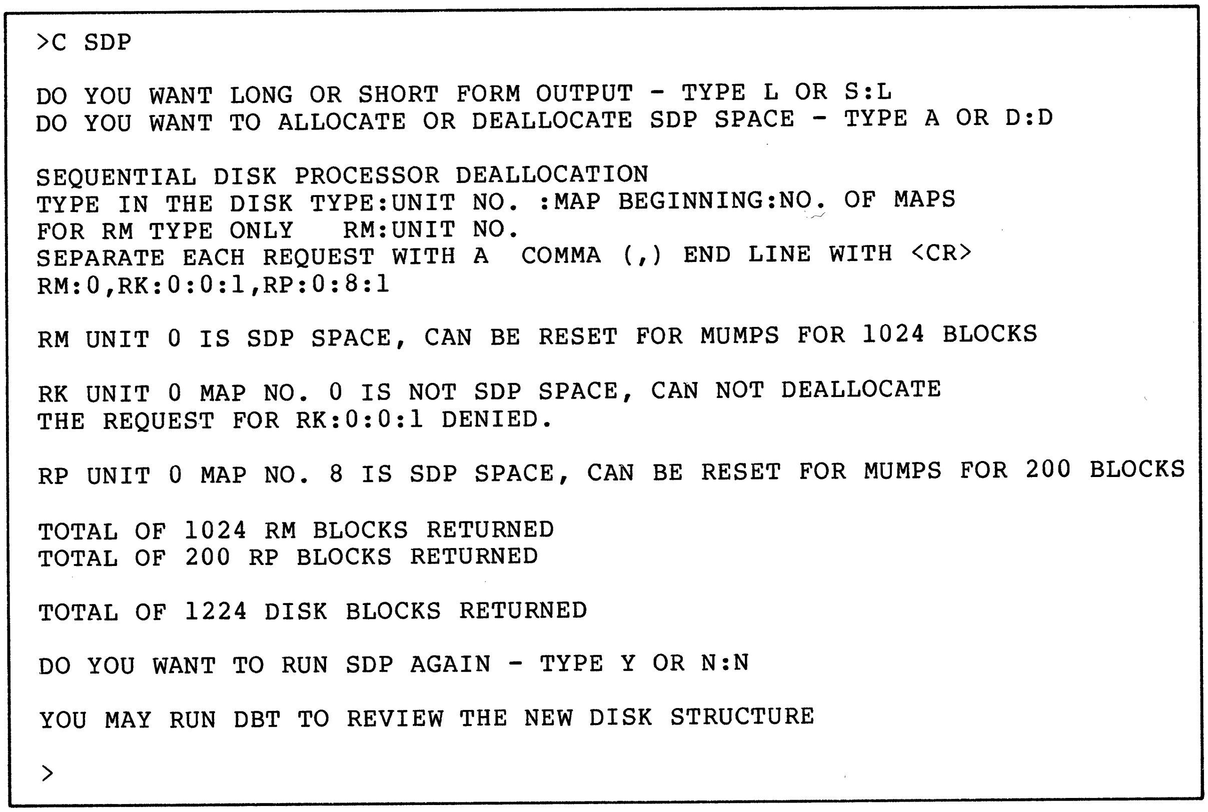
map beginning = map number of first map in a contiguous series

no. of maps = the number of contiguous maps in the series

NOTE

> Because an RM disk unit has only one bit map per unit, the "map beginning" and "no. of maps" entries should not be specified.

Figures 6-9 and 6-10 show typical SDP deallocation and allocation sessions.



```
>C SDP

DO YOU WANT LONG OR SHORT FORM OUTPUT - TYPE L OR S:L
DO YOU WANT TO ALLOCATE OR DEALLOCATE SDP SPACE - TYPE A OR D:D

SEQUENTIAL DISK PROCESSOR DEALLOCATION
TYPE IN THE DISK TYPE:UNIT NO. :MAP BEGINNING:NO. OF MAPS
FOR RM TYPE ONLY   RM:UNIT NO.
SEPARATE EACH REQUEST WITH A  COMMA (,) END LINE WITH <CR>
RM:0,RK:0:0:1,RP:0:8:1

RM UNIT 0 IS SDP SPACE, CAN BE RESET FOR MUMPS FOR 1024 BLOCKS

RK UNIT 0 MAP NO. 0 IS NOT SDP SPACE, CAN NOT DEALLOCATE
THE REQUEST FOR RK:0:0:1 DENIED.

RP UNIT 0 MAP NO. 8 IS SDP SPACE, CAN BE RESET FOR MUMPS FOR 200 BLOCKS

TOTAL OF 1024 RM BLOCKS RETURNED
TOTAL OF 200 RP BLOCKS RETURNED

TOTAL OF 1224 DISK BLOCKS RETURNED

DO YOU WANT TO RUN SDP AGAIN - TYPE Y OR N:N

YOU MAY RUN DBT TO REVIEW THE NEW DISK STRUCTURE

>
```

Figure 6-9 Example of SDP Deallocation Session

```
>C SDP

DO YOU WANT LONG OR SHORT FORM OUTPUT - TYPE L OR S:L
DO YOU WANT TO ALLOCATE OR DEALLOCATE SDP SPACE - TYPE A OR D:A

SEQUENTIAL DISK PROCESSOR ALLOCATION
TYPE IN THE DISK TYPE:UNIT NO. :MAP BEGINNING:NO. OF MAPS
FOR RK TYPE ONLY   RM:UNIT NO.
SEPARATE EACH REQUEST WITH A COMMA (,) END WITH A <CR>
RM:0,RP:0:13:1,RK:0:0:1

SPACE FOR SDP ALREADY ALLOCATED ON RM UNIT 0

SPACE FOR SDP CAN BE GIVEN AS REQUESTED RP UNIT 0 MAP NO. 13 FOR 200 BLOCKS

CANNOT ALLOCATE MAP 0 OF SYSTEM DISK!

TOTAL OF 200 RP BLOCKS GIVEN

TOTAL OF 200 DISK BLOCKS GIVEN

DO YOU WANT TO RUN SDP AGAIN - TYPE Y OR N:N

YOU MAY RUN DBT TO REVIEW THE NEW DISK STRUCTURE

>
```

Figure 6-10 Example of SDP Allocation Session

6.2.12 SIF - System Information

This program provides both system partition and system status information. The partition data is obtained from the system's Partition Available Table while system status data is provided by the SS program which is called by SIF. Both system status and partition information is reported in a tabular format via the currently ASSIGNed I/O device. Figure 6-11 shows a typical SIF report. The user should refer to Section 6.2.13 for a description of the System Status report. The partition information part of the report is self-explanatory.

```
>C SIF

                     SYSTEM INFORMATION


                       SYSTEM STATUS

DATE--28 MAY 75                          TIME--11:49AM
NUMBER OF ACTIVE JOBS -- 2          NUMBER OF AVAILABLE JOBS --9
JOB NUMBER           STATE                             DEVICES OWNED
      1              IN CLOCK Q
      2              AVAILABLE FOR USE
      3              AVAILABLE FOR USE
      4              IN RUN Q                          1 4
      5              AVAILABLE FOR USE
      6              AVAILABLE FOR USE
      7              AVAILABLE FOR USE
      8              AVAILABLE FOR USE
      9              AVAILABLE FOR USE
      10             AVAILABLE FOR USE
      11             AVAILABLE FOR USE


          STANDARD PARTITION SIZE  8192 BYTES

JOB NUMBER           PARTITION SIZE      START ADDRESS
      1              1536 BYTES          133632
      2              8192 BYTES          188416
      3              8192 BYTES          188224
      4              8192 BYTES          172032
      5              8192 BYTES          163840
      6              8192 BYTES          155648
      7              4096 BYTES          151552
      8              4096 BYTES          147456
      9              4096 BYTES          143360
      10             4096 BYTES          139264
      11             4096 BYTES          135168
>
```

Figure 6-11 Example of SIF Report

6.2.13 SS - System Status

This program provides status information on all jobs in the system. The job information is obtained from the various system tables and queues in a "snapshot" manner. The data is interpreted and subsequently reported on the currently ASSIGNed I/O device. The report itself is in tabular form, consisting of three columns of data. Figure 6-12 shows a typical example of an SS report.

```
MUMPS INITIALIZER

THE SYSTEM WAS LAST SAVED ON 30 JAN 77 AT 7:25 PM

TYPE IN DATE MM/DD/YY  1/30/77        NEW DATE  30 JAN 77
TYPE IN TIME HH:MM  19:26             NEW TIME  7:26 PM
EXIT
```

Figure 6-12 Example of SS Report

The following paragraphs describe the responses that can appear in each column.

6.2.13.1 JOB NUMBER

This number identifies a user-initiated task (e.g., logging in, STARTing a program). There is a corresponding job number for each partition in the system.

6.2.13.2 STATE

Entries in this column specify the status of each job. The user should remember that when this information is printed out, it is not necessarily the system's status at the time of printing, for a certain amount of time is required to compile the data. There are fourteen possible job status states:

DISK I/O DONE — The job has just completed a logical disk access. A job having this status has the highest execution priority.

DISK I/O BOUND — The job is currently performing a logical disk access (e.g., accessing a global node).

DISK RESOURCE BOUND — The job is currently waiting to use the disk.

RING BUFFER RESOURCE BOUND Q — The job is currently waiting to obtain a 64-character ring buffer. If several jobs are listed with this status, it is likely that a new SYSGEN should be performed to allocate more buffer space.

256-WORD BUFFER RESOURCE BOUND Q
The job is currently waiting to obtain a buffer in preparation for DECtape, magtape, VIEW or SDP I/O.

HI PRIORITY WAIT Q
The job is currently waiting to run (next in priority after Disk I/O jobs are done). Jobs in this queue are given one CPU time slice.

MED PRIORITY WAIT Q
The job is currently waiting to run (next in priority after high priority jobs). Jobs in this queue are given two CPU time slices.

LOW PRIORITY WAIT Q
The job is currently waiting to run (next in priority after med-priority jobs). Jobs in this queue are given three CPU time slices.

IN CLOCK Q
The job is waiting for a time period to expire (e.g., timed READ, HANG or other significant event).

TERMINAL I/O HUNG
The job is currently waiting for completion of I/O operation to a terminal device (devices 1,4-19, 64-111).

CARD READER I/O HUNG
The job is currently waiting for completion of I/O operation to a card reader (devices 51-54).

DECTAPE I/O HUNG
The job is currently waiting for DECtape I/O to complete.

MAGTAPE I/O HUNG
The job is currently waiting for magtape I/O to complete.

IN TRANSITION
The job was not found in any queue.

AVAILABLE FOR USE
This message is self-explanatory.

6.2.13.3 DEVICES OWNED

This column lists the numbers of the I/O devices owned (ASSIGNed) by the job.

6.2.14 SSD - System Shutdown

In order to assure that data base integrity is maintained, it is important that the MUMPS-11 System is HALTed only after all jobs have signed off. An abrupt system HALT can cause user files and, more importantly, the disk data structure to be seriously degraded. SSD aids the operator in performing an orderly system shutdown by providing the step-by-step procedures to bring the system to a quiescent state with all users signed off.

SSD can be run only from the system's Console Terminal; an attempt to run from another terminal results in an appropriate message. Once SSD is running, it types a series of messages that specify the procedures the operator must follow. Figure 6-13 shows a typical operating session. If other users' jobs are running, the program then outputs a system status report (SS program) so that the operator can tell exactly which users' jobs are still running. If requested, SSD will update this report at specified intervals. Once all users have signed off, SSD will run the DBT program, if requested, to allow the operator to validate all disk structures. When SSD has completed its operation, it types EXIT, then stops. The system is ready for shutdown and the operator can now press the console's HALT switch.

```
>C SSD


         * * * S Y S T E M   S H U T   D O W N * * *

DATE -- 20 MAY 75                        TIME -- 12:16PM


NOTE:  IF TERMINAL USERS DO NOT COOPERATE IN GETTING OFF  THE  SYSTEM,
RAISE SWITCH REGISTER SW #15 AND WAIT 30 SECONDS, OR UNTIL THE ADDRESS
REGISTER LIGHTS HAVE  SETTLED.   THEN  RAISE  SW  #4.   AFTER  SEVERAL
SECONDS  (GARBAGE  COLLECTOR  FINISHES  UP  IF  THERE  WERE  BLOCKS TO
COLLECT) YOU MAY HALT THE MACHINE.

RAISE SWITCH REGISTER SW #6

RUN DISK BLOCK TALLY (Y OR N)?Y



               DISK BLOCK TALLY
DATE  20 MAY 75               TIME  12:16PM

ENTER TERMINAL # (0-19,64-95) FOR TALLY OUTPUT: 0
DO YOU WANT LONG FORM (RESULTS PER MAP) TALLY OUTPUT,
OR SHORT FORM (RESULTS PER UNIT)?  TYPE L OR S.  L
DO YOU WANT TO TALLY ALL DISKS (Y OR N)? Y



       DISK  MAP SUMMARY  CALCULATED   AVAILABLE    # OF
      /UNIT     COUNT    GLOBAL BLOCKS SDP BLOCKS   ERRORS

MAP0   RK/0     1698        1698         0
MAP1   RK/0     2053        2053         0
       RK       3751        3751         0            0



DISK BLOCK TALLY FINISHED


RAISE SWITCH REGISTER SW #4 TO DISABLE GARBAGE COLLECTOR


SYSTEM SHUTDOWN COMPLETED AT 12:17PM ON 20 MAY 75


YOU MAY HALT THE MACHINE AFTER 'EXIT'
IF PHYSICAL DISK SAVE(S) DESIRED, BOOT BACKUP PROGRAM.


EXIT
```

Figure 6-13 Example of SSD Session

6.2.15 STU - System Startup

STU is an initialization program that automatically runs whenever MUMPS is loaded from the disk and started using the bootstrap loader (see Section 4.5.2). Once running, the program prints the date and time of the last system save operation (see Section 4.4.3), calls the Date and Time routines (%D and %T), starts the system Caretaker program (CTK), then halts. If the user does not want CTK to run, it can be stopped by raising SWITCH REGISTER SWITCH 12 for 10 seconds.

Since program operation is inherent in system initialization via the bootstrap loader, the system library must always contain a program called STU. Users not wishing to use this program should create their own version even if it consists of a HALT instruction only.

Figure 6-14 shows an example of STU operation. The user's responses are underlined.

```
MUMPS INITIALIZER

THE SYSTEM WAS LAST SAVED ON 5 NOV 74 AT 10:15AM


TYPE IN DATE MM/DD/YY  11/6/74           NEW DATE   6 NOV 74
TYPE IN TIME HH/MM   10:05               NEW TIME  10:05AM
EXIT
```

Figure 6-14 Example of STU Dialogue

6.2.16 TIM - Set Time

This program allows the user to set the $T system variable to the current time of day. When it is loaded and running, TIM requests the user to enter the time. The user should enter the time in 24-hour form (e.g., 07:30 = 7:30 A.M., 15:00 = 3:00 P.M., etc.). Once the new time is entered, TIM outputs the new contents of $T (using the %T Library Utility Program) to allow verification. TIM performs error checking on input to determine correct value range and form.

6.2.17 %GP - Global Place

This program allows the user to position a new global on a selected physical disk type and unit. In the case of RK11, RP11 or RH11 disk systems, the disk cylinder can also be selected (RK611 disks are not organized into cylinders). Its operation consists of:

1. entering the global's name in the global directory of the current UCI, and

2. allocating the first block of the global on the specified disk unit and cylinder, if applicable.

Since the program's operation involves manipulation of the system's file structure, %GP does not allow other jobs to be running at the same time. Furthermore, the user must have access to the System UCI code to operate %GP. The program consists of the main program %GP and the subprogram %MU.

Once running, the program identifies itself, then requests the user to enter the system's UCI code. When the code is validated, %GP asks the user to quiet the system by using one or more specified SWITCH REGISTER switches. After all other users are logged out of the system, the program begins the global placement session with the question:

```
                        |xxx↵|
          GLOBAL?     ↑ |YYY↵|
                        |↵   |
```

The user can type:

A new global name (xxx)
to create a new global not currently residing under the current UCI.

An old global name (YYY)
to request the deletion of an existing global and the creation of a new global of the same name.

↵ to terminate program operation.

If an old global name is entered, the program responds:

```
          ↑YYY ALREADY EXISTS!!

                                                        |Y|
          DO YOU WISH TO KILL THIS GLOBAL?  (TYPE Y OR N): | |↵
                                                        |N|
```

The user can type:

Y to delete the present global and create a new global with this name.

N to restart %GP so that another name can be used.

%GP then requests the name of the type of disk where the global is to reside:

```
                        |RK↵|
                        |RF↵|
                        |RS↵|
          DISK TYPE?    |RM↵|
                        |RP↵|
                        |RJ↵|
                        |↵  |
```

The user can type:

RK to specify the RK03 or RK05 on the RK11

RF to specify the RS11 on the RF11

RS to specify the RS04 on the RH11

RM to specify the RK06 on the RK611

RP to specify either the RP02 or RP03 disk on the RP11 controller

RJ to specify either the RP04, RP05 or RP06 disk on the RH11 controller

CARRIAGE RETURN (↵)	to specify automatic allocation of the global to the system disk (the next two questions are skipped).

Thereafter, the program requests the unit number of the specified disk.

UNIT? n↵

The user can type any number (n) in the range 0 through 7 that specifies a disk unit that exists in the system.

If the disk that was selected is an RK11, an RP11, or an RH11 (RJ), %GP requests the number of the cylinder where the global is to reside:

CYLINDER? | nnn
↵ |

The user can type:

cylinder number (nnn)	to specify the disk cylinder where the global is to begin. %GP will attempt to place the first block of the global on this cylinder, or if full on the next (higher) cylinder.[1]
↵	to request block placement on cylinder zero (0) or if full, on the next available cylinder.

%GP reports the results of each successful placement session in the following form:

(dkn:nn mmm)

where:	dkn = disk type and unit number (e.g., RK03)
	nn = cylinder number
	mmm = address of the disk block allocated as the global's first block.

6.3 SYSTEM DEMO PACKAGE

The system demo package is a group of eight separate test programs (TP1 through TP8) that are designed to assist the system manager in verifying the integrity of the MUMPS-11 operating system and language. These programs are particularly useful for verifying the system after a SYSGEN session or after field service hardware or software maintenance.

As shown in Table 6-3, each program performs a group of tests related to a specific aspect of the system or language. The tests can be run either singly or in any desired combination. The TP7 program runs all programs in the test package, except TP5 and TP6, automatically. The

[1] Refer to the PDP-11 Peripherals and Interfacing Handbook for further information.

system test package is a first-level check to assure that a MUMPS hardware/software system is operational. It should be run when a system is installed and the output should be saved for subsequent checks if the system seems to be malfunctioning.

Table 6-3
System Test Package Programs

Program Name	Description
TP1	MUMPS-11 Language Functions Test Verifies the operation of all functions in the language. (Creates ↑A global)
TP2	System Timings Test Performs and reports timing measurements for arithmetic, global and symbol table routines. (Creates ↑A global)
TP3	MUMPS-11 Expression Evaluate Test Verifies the operation of all expression evaluating tasks. (Creates globals: ↑A,↑B,↑C,↑D,↑E,↑T)
TP4	Line Printer/Terminal Test Verifies operation of the line printer or any specified terminal.
TP5	DECtape Test Verifies all DECtape functions and error conditions.
TP6	TM11 Magnetic Tape Test Verifies all magtape functions and error conditions.
TP7	System Exerciser Utility Runs all other test programs except TP5 and TP6 and performs its own abbreviated TM11 magtape and DECtape tests.
TP8	Global Test Builds a test global (↑A) to check global routine operation.

APPENDIX A

GLOSSARY OF TERMS

Array

An array, which can consist of either local or global variables, is a group of subscripted variables that have a common identifier.

Binary Operator

A binary operator is an operator that requires two operands (expression elements).

Boolean Valued Expression

A Boolean valued expression (bve) is an expression, which, when evaluated, produces either a True (-0.01) or False (0) result.

Command

A command is the principal algorithmic component of the MUMPS Language. MUMPS commands consist of a set of keywords that characterize actions. (e.g., GOTO, SET, HALT, RUN,etc.)

Concatenation

Concatenation is the process of linking together two or more string data elements to form a single string. Concatenation is a string expression operation that is designated by the commercial "at" sign (@).

Constant

A constant is a quantity within the range of legal MUMPS numbers (±21474836.47) explicitly stated in an argument to a command or as an operand in an expression.

Data Base

Data base is that body of disk-stored information residing in global arrays.

Direct Mode

Direct mode is that mode of system operation which enables the programmer to:

1. enter commands and/or functions for immediate execution

2. create or modify steps of a user's program

Directory

A directory is a disk resident table which can contain the names and disk starting addresses of either programs or global files. Each User Class Identifier in a MUMPS-11 system is associated with two directories; a program directory, and a global directory.

Double Numeric Quantity

This term refers to MUMPS numbers whose absolute values lie in the range (±) 21474836.47 which are stored by the operating system in two consecutive words. (See also Single Numeric Quantity.)

Expression

An expression is any legal combination of operands (elements) and operators. Legal expression elements include: literals, constants, variables, subexpressions, and function references. An expression may consist of a single element, an element/operator combination or a series of element/operator combinations.

Expression Element

An expression element is the operand component of a MUMPS expression. An expression element may be a constant, a simple variable, a literal, a local subscripted variable, a global variable, a function reference, or a subexpression.

Floating Point Numeric

A 4-word floating point number in the range $+0.14*10^{38}$ to $+1.7*10^{38}$. The MUMPS $M function allows floating point numbers to be used with the operators + - * / <> =. A floating point number may be stored only as a local variable which is not the name of an associated array (i.e., pointer variables are excluded) or as a global variable.

Function

A function is a MUMPS expression component that invokes an algorithm, the result of which is an expression element (operand). Each MUMPS function is assigned a unique mnemonic, the first character of which is the dollar sign ($) symbol.

Global

A global is a tree-structured data file stored in the common data base on the disk. Globals comprise an external system of symbolically referenced arrays.

Global Variable

A global variable is a subscripted variable that forms an element (or node) of a global array.

Identifier

An identifier is a name consisting of one to three alphanumeric characters. The first character must be either an alphabetic character or the percent (%)

symbol. Identifiers are used as names for variables, programs, library (or system) programs, and globals. The percent symbol is reserved for naming library programs and globals, though any local variable can use percent as the first character of its name.

IF Switch

The IF switch is a logical switch that resides in the Program Vector area in each user's partition. This switch is set to the logical result of the last executed IF statement, either True (-0.01) or False (0). Note that an IF without arguments or an ELSE only tests the logical value of the IF Switch and does not change it.

Indirect Mode

Indirect mode is that mode of system operation in which the steps of a stored program are executed. In this mode of operation, commands cannot be entered from the terminal and programs cannot be created or modified.

Indirect Reference

An indirect reference is a feature of the language that permits a string variable to represent a command's argument or argument list. In operation, the string value of the variable is taken as the argument or argument list. The indirection symbol, back arrow (←) or underscore (_), must precede the variable reference.

Job

A job is any user activity which requires the use of a partition. For example, logging or STARTing a program are jobs.

Library Program

This term refers to those programs that are listed in the Program Directory of the System UCI (UCI#1) and have a percent symbol (%) as the first character of their names. Programs residing in the system in this way can be run by any user regardless of UCI.

Literal

A literal is an element of the language that permits the explicit representation of character strings in expressions, commands, and function arguments by delimiting them with quotation marks (" "). Literals may not contain:

quotation marks
Carriage RETURN
ALT MODE
RUBOUT (DEL)
CTRL O
CTRL C
CTRL U
NUL
Line Feed
Form Feed
Vertical Tab
CTRL S and Q

Local Variable

A local variable is a variable that resides in the partition of the program that created it (as opposed to a global variable).

Naked Reference

The naked reference is a feature that provides an abbreviated method for accessing global variables to reduce disk access time. This permits subsequent references to a global to be made simply by specifying an up-arrow (↑) followed by one or more subscripts. The variable name is assumed from the last global reference in which a name was explicitly stated. The first subscript in the naked reference replaces last subscript in the previous reference (either naked or complete). Using the naked reference reduces disk access time since the search for the specified node begins at the subscipting level attained by the last global reference rather than at the global directory level.

Node

A node is a global array element addressed by a subscript.

Numbers

Numbers in MUMPS are signed fixed-point quantities in the range ±21474836.47. Decimal fractions greater than two places are truncated to two places.

Numeric Valued Expression

A numeric valued expression (nve) is an expression which, when evaluated, produces a numeric result.

Operator

An operator is a component of a MUMPS expression that invokes an algorithm to perform either arithmetic, string, or Boolean manipulators. (See binary operator and unary operator.)

Part Number

A part number is the integer portion of a step number and is used to refer collectively to all steps having a common integer base.

Partition

A partition is the memory area within which a job resides. A partition is allocated to a job either at terminal log-in time or upon execution of the START command. A partition contains both program and local variable storage areas as well as program state information necessary for timesharing operation.

Pattern Verification

Pattern verification is a feature of the language which permits evaluation of text strings for the occurrence of desired combination of alphabetic, numeric, and punctuation characters.

Pattern verification is specified by the "?" operator followed by Pattern Specification Codes (psc).

Principal I/O Device

This term refers to the keyboard

terminal that initiated the job. This is the device to which control returns when an error message is to be output or when an ASSIGN␣O command is issued.

Program Name

A program name is an identifier that is associated with a particular program. System library program names must use the percent symbol (%) as the first character.

Programmer Access Code

The Programmer Access Code (PAC) is a 3-character code, created at system generation time, that allows the terminal user to enter direct mode.

Queue

A queue is an ordered list in which the first item to be entered is the first item to be removed (first-in-first-out sequence).

Run Queue

The run queue is a system queue which contains the number of the job currently executing in its time slice. This queue is effectively a one-entry queue.

Secondary Storage

This term refers to all I/O devices which are not used to contain the global data base (non-disk), (i.e., paper tape, magtape, or DECtape).

Single Numeric Quantity

This term refers to MUMPS numbers in the range ±327.67 which are stored by the operating system in one 16-bit word. (See also Double Numeric Quantity.)

Sparse Array

A sparse array refers to the method of storage allocation used for local and global arrays in which space is allocated only as variables are explicitly defined (unlike other languages which require dimension or size statements for preallocation of storage).

Step Number

A step number is a number used to identify each line of a MUMPS program. A step number must be in the range 0.01 - 327.67, excluding all numbers in this range that are integers.

String

A string is a contiguous combination of any of the ASCII characters. (132 characters maximum).

String Concatenation

See Concatenation.

String Valued Expression

A string valued expression (sve) is an expression which produces a string result upon evaluation.

Subexpression

A subexpression is an expression element that consists of any legitimate expression enclosed in parentheses.

Subscripts

A subscript is a numeric valued expression or expression element which is appended to a local or global variable name to uniquely identify specific elements of an array. Subscripts are enclosed in parentheses. Multiple subscripts must be separated by commas.

Subscripted Variable

A subscripted variable is a variable to which a subscript is affixed (see Subscript and Variable). Both global and local variables are forms of subscripted variables.

System Program

A system program is a program either supplied by DIGITAL or created by the MUMPS user which is used to assist the MUMPS system owner in the operational maintenance of the system. System programs normally reside under the protection of the System UCI (UCI #1).

System Queues

This term refers to the set of queues used by the MUMPS Operating System to control the allocation of system resources (see Run Queue and Wait Queue).

System UCI

The System User Class Identifier (UCI) code is that UCI code assigned to the first entry in the system's UCI table. The Program and Global Directories associated with the System UCI are used to contain both system and library programs and globals.

System Variable

A system variable is a variable that is permanently defined within the operating system. These variables provide system and control information to all programs. The first character of a system variable is always a dollar sign ($). System variables are maintained and modified by the operating system and/or system manager only.

Time Slice

This term refers to the period of time allocated by the operating system to process a particular partition's program. This term is synonymous with 'timesharing interval'.

Unary Operator

A unary operator is an operator that requires a single operand (expression element).

User Class Identifier (UCI)

A UCI is a 3-character code used at terminal log-in time to permit access to the group of programs and global files with which it is associated. When used with the Programmer Access Code, the UCI allows these programs to be modified and new programs to be created.

Variable — A variable is the symbolic representation of a logical storage location. Specific types include local, global, simple, and subscripted variables. Variables are symbolically referenced by means of identifiers.

Wait Queues — The wait queues are a group of system queues which contain the numbers of the jobs awaiting service by the operating system.

APPENDIX B

MUMPS CHARACTER SET

The following table shows, with the corresponding octal and decimal equivalents, the 128-character set of 7-bit ASCII code used by MUMPS for data, command, and control purposes. In addition, the order of the character set as shown establishes the MUMPS collating sequence used by the system's Expression Evaluator when establishing string relationships.

For command and control purposes, MUMPS uses the 64-character graphic subset. The system also uses the control codes shown in brackets ([ ]). These codes may not be used as input data. The NUL, code 000, is used internally as a logical end-of-message and cannot be used. Characters shown in braces({}) are part of the 1963 ASCII Character Set and may appear in the character set of some terminals.

All characters may be used for data input and output except for those mentioned above. The system does not perform any character conversion. It is the programmer's responsibility to perform all upper/lower-case letter conversions or mappings which are required for the particular application.

CHARACTER SET

Octal Code	Decimal Code	Character
000	000	NUL
001	001	SOH (Backspace)+
002	002	STX (Forward space)+
003	003	ETX (CTRL C)* (Write tape mark).
004	004	EOT (Write Block)
005	005	ENQ (Rewind)+
006	006	ACK (Read block)+
007	007	BELL (Read Label)+
010	008	BS*(Write header label)+
011	009	HT (Write EOF label)+
012	010	LF
013	011	VT
014	012	FF
015	013	CR
016	014	SO
[017	015	SI(CTRL O)*]
020	016	DLE
021	017	DC1
022	018	DC2
023	019	DC3

+ Dagger denotes the control function for magtape devices.

* Asterisk denotes the control function for MUMPS terminal, if different from specified or other use.

CHARACTER SET (Cont.)

Octal Code	Decimal Code	Character
024	020	DC4
[025	021	NAK (CTRL U)*]
026	022	SYN
027	023	ETB
030	024	CAN
031	025	EM
032	026	SUB
[033	027	ESC (ALTMODE)*]
034	028	FS
035	029	GS
036	030	RS
037	031	US
040	032	Space
041	033	!
042	034	'
043	035	#
044	036	$
045	037	%
046	038	&
047	039	'
050	040	(
051	041	)
052	042	*
053	043	+
054	044	,
055	045	-
056	046	.
057	047	/
060	048	0
061	049	1
062	050	2
063	051	3
064	052	4
065	053	5
066	054	6
067	055	7
070	056	8
071	057	9
072	058	:
073	059	;
074	060	<
075	061	=
076	062	>
077	063	?
100	064	@
101	065	A
102	066	B
103	067	C
104	068	D
105	069	E
106	070	F
107	071	G
110	072	H
111	073	I
112	074	J
113	075	K
114	076	L
115	077	M
116	078	N

CHARACTER SET (Cont.)

Octal Code	Decimal Code	Character
117	079	O
120	080	P
121	081	Q
122	082	R
123	083	S
124	084	T
125	085	U
126	086	V
127	087	W
130	088	X
131	089	Y
132	090	Z
133	091	[
134	092	\
135	093	]
136	094	^ {↑}
137	095	— {←}
140	096	`
141	097	a
142	098	b
143	099	c
144	100	d
145	101	e
146	102	f
147	103	g
150	104	h
151	105	i
152	106	j
153	107	k
454	108	l
155	109	m
156	110	n
157	111	o
160	112	p
161	113	q
162	114	r
163	115	s
164	116	t
165	117	u
166	118	v
167	119	w
170	120	x
171	121	y
172	122	z
173	123	{
174	124	¦
175	125	} (ALTMODE)*
176	126	~ (ALTMODE)*
177	127	DEL (RUBOUT)+

* Asterisk denotes the control function for MUMPS terminals, if different from specified or other use.

+ Dagger denotes the control function for magtape devices.

APPENDIX C

EXPLANATION OF MUMPS MESSAGES

When execution of a MUMPS program is terminated by either an error, a CTRL C, or by pressing the BREAK key, the program executive outputs a short message to indicate the reason for termination. This message is followed by the number of the Step being executed and the program name, unless the error occurred while in Direct Mode. The error message format is:

```
?message>spn␣pnam
```

MUMPS messages are categorized as follows:

1. MUMPS Programming Error Messages - these result from errors associated with programming problems (either incorrect language syntax or semantic misunderstandings).

2. Voluntary Program Termination Message - there is only one message of this type.

3. Debugging Aid Message - this indicates that a BREAK command has been encountered in the program.

4. Operating System Error Messages - these result from various troubles which are detected by the operating system and which are beyond the control of the MUMPS application programmer.

MUMPS errors are considered terminal unless the user's program SETS the $E System Variable for application program control of error processing. The programmer may SET $E to a Step or Part number (S␣$E=spn) to which control will go if an error occurs (except GARB0 - GARB4 errors are reported only on the console terminal and do not terminate a running job). When $E is set to an spn and an error occurs, the system transfers control to the spn and resets $E to an index in the range 0 through -0.38, which indicates the type of error encountered (e.g., 0 = INRPT, -0.01 = MXNUM - see below). The number of the Step that contains the error is entered in the $W System Variable. The system also cancels all currently active DO, FOR, and CALL commands. It is the user's responsibility to reset $E to an spn if he wishes to control further error processing; otherwise, error processing reverts to system control.

If an error occurs and $E is not set by the programmer, the action taken by the system depends on the mode in which the user signed on at log-in. If the programming access code (PAC) was used, control is returned to Direct Mode after the error message is output. Otherwise, the job is aborted after typing the error and 'EXIT' messages, and the terminal is automatically logged-out.

Each of the messages is explained on the pages which follow:

C.1 MUMPS PROGRAMMING ERROR MESSAGES

Message	$E Index	Meaning
CMMND	-0.15	Indicates illegal use of a command: 1. Command is undefined in the language 2. An argument has been omitted where required.
DIVER	-0.19	Indicates an attempt to perform division by zero.
DKSER	-0.04	Indicates (if not a system software error (C.4)), an attempt to: 1. use the VIEW command to access a block number larger than the size of the referenced disk, or a nonexistent disk; or 2. use the disk (e.g., creating global variables, issuing the FILE, LOAD, etc., commands) under a UCI that has no associated directories.
FRACT	-0.08	Indicates that a fractional number was encountered when the process being executed was expecting a integer number. Also involved when a Step number has no fractional part.
FUNCT	-0.07	Indicates that the function is undefined in the language.
LBOV	-0.14	Indicates an attempt to input or output a line greater than 132 characters.
$MERR	-0.36	Indicates that an error occurred in $M processing. 1. exponent overflow 2. exponent underflow 3. division by 0 4. illegal trap instruction (system error)
MINIM	-0.03	Indicates that a number has more than two digits following the decimal point.
MINUS	-0.12	Indicates that a negative or zero number was encountered when a positive number was expected. For example, MUMPS causes a MINUS error if the user references a subscripted variable with a negative subscript. Only positive subscripts are allowed, except when using the $HIGH function.

EXPLANATION OF MUMPS MESSAGES

Message	$E Index	Meaning
MODER	-0.23	1. An nve was encountered where an svl was expected, or vice versa. 2. Argument to $TEXT is not numeric. 3. Argument to $VIEW is not numeric.
MXNUM	-0.01	Indicates that the value of a number has exceeded the integer bounds set by the MUMPS system. The maximum value for a number is ±21474836.47.
MXSTR	-0.02	Indicates that the string has exceeded maximum length allowed (132 characters).
NAKED	-0.29	Indicates that the present user attempted to reference a global variable using "naked" syntax: 1. prior to any full syntax reference; or 2. after another user KILLed the global variable.
NODEV	-0.13	Indicates an attempt to ASSIGN a nonexistent device or the use of an illegal device number.
NOPGM	-0.28	Indicates that reference is made to a program name that does not exist in the program directory for this UCI and is not in the Directory of Library (%) Programs.
NOTSY	-0.34	Indicates that the referenced device or function is not in the system (it may not have been linked at system generation).
NXMEM	-0.05	Indicates that nonexistent Memory was referenced in the VIEW command or in the $VIEW function.
PGMOV	-0.24	Indicates that there is insufficient space available in the partition. This can be caused by: 1. too many program steps in the program being created via the terminal or in the program being loaded; (LOAD, CALL and OVERLAY commands) 2. too many local variables; 3. expression or subscript nesting too deep.
PROT	-0.06	Indicates that an attempt was made to use either the VIEW Command or the $VIEW Function from a non-Library (%) Program or when not logged in under the System UCI. Also indicates that the MODIFY command issued from Indirect Mode specified an spn smaller than the current spn.

Message	$E Index	Meaning
SBSCR	-0.09	Indicates illegal subscript usage: - subscript out of range; - negative subscript.
SPNER	-0.17	Indicates that an illegal or nonexistent Step or Part number was used.
STKOV	-0.10	Indicates that the available stack space is used up. Generally indicates nesting is too deep in DO or CALL statements.
STKUN	-0.11	Indicates execution of the Overlay command from Direct Mode (stack underflow).
SYMOV	-0.16	Indicates that Symbol Table Overflow occurred on an attempt to create or change a local variable.
SYNTX	-0.27	Indicates that the current Step being executed has an error in syntax. Syntax errors include illegal punctuation, illegal use of operators, illegal use of parentheses, as well as errors encountered in editing a Step. Syntax errors comprise a great majority of errors made in the MUMPS system and usually the user will be able to determine the exact cause of the error by merely looking at the Step concerned.
UNDEF	-0.21	Indicates a reference to an undefined local or global variable.

C.2 VOLUNTARY PROGRAM TERMINATION

Message	$E Index	Meaning
INRPT	0	Indicates an interruption of the program execution, caused by typing CTRL C or pressing the BREAK key.

C.3 DEBUGGING AID MESSAGE

Message	$E Index	Meaning
?n BREAK	None	Indicates that program control has reached a BREAK command at Step n. BREAK commands are used to interrupt execution of the program for debugging purposes. The GO command may be typed to resume operation.

C.4 MUMPS OPERATING SYSTEM ERROR MESSAGES

Message	$E Index	Meaning
GARB0	None	Disk error while reading a data block.
GARB1	None	Disk error while writing a data block.
GARB2	None	Disk error while reading a bit map.
GARB3	None	Disk error while writing a bit map.
GARB4	None	Disk error, an attempt to deallocate a bit map or data block not yet allocated.

NOTE

The above errors are disk errors detected by the system's Garbage Collector routine. The message is output to the console terminal. GARB1 and GARB3 result in suspension of all disk I/O until system restart. Notify system manager.

Message	$E Index	Meaning
DBDGD	-0.31	Indicates a data base degradation. The system attempted to read a block that was not actually allocated. Notify system manager.
DKDER	-0.33	Indicates that a disk I/O occurred on an attempt to write a global data buffer. The error is not given until the write is actually attempted.
DKFUL	-0.26	Indicates that there is no more room on the disk for global or program storage. Caused by SET and FILE commands. Notify system manager.
DKHER	-0.20	Indicates disk hardware error. Notify system manager.
DKSER	-0.04	Indicates that disk block pointers in the global data base are referencing nonexistent or invalid disk blocks (see also conditions listed under C.1). Notify system manager.
DSKDG	-0.18	Indicates disk degradation. Attempt was made to allocate bit map for data storage. The system corrects the bit map subsequent to this error. Notify system manager.
DTERR	-0.30	Indicates DECtape hardware or operator error. Common causes are: 1. not set to ON LINE 2. not set to WRITE ENABLE 3. unit number not selected.

EXPLANATION OF MUMPS MESSAGES

Message	$E Index	Meaning
LPERR	-0.38	Indicates a line printer hardware error. Common causes are: 1. device off line 2. out of paper 3. yoke open 4. power off.
MTERR	-0.37	Indicates magtape hardware or operator error as determined by the current contents of the $A System Variable. The system generates this error only if the user SET the $E System Variable.
PLDER	-0.35	Indicates that the system cannot retrieve the program being LOADed, CALLed, or STARTed. The FILE command did not complete writing the program. The user must re-FILE the backup copy of the program.
SWAP	-0.32	Indicates: 1. that the previous swap-out overflowed the user partition stack. The error is not reported until the next swap-in. 2. imminent system stack overflow. May be caused by faulty programming techniques, for example: 1.10 F I=1:1:1000D 2 2.10 D 1
SYSDG	-0.25	Indicates that the table in main memory which represents the bit maps on a physical disk unit (Disk Storage Allocation Table) does not correspond to the block allocation specified by the disk's bit maps. The Disk Block Tally Utility Program allows recovery from this error. Notify system manager.
SYSER	-0.22	Indicates system stack underflow on swapout. Notify system manager.

APPENDIX D

SYMBOL USAGE

The following special symbols are used by MUMPS in addition to the logical operators described in Chapter 2.

Symbol Definition

Number sign is used as a format control character to initiate a Page Feed or a FORM FEED on an output device.

! Exclamation point is used as a format control character to initiate a Carriage RETURN/LINE FEED sequence on an output device.

? Question mark is multiply defined:

1. as an output format control character for terminals, line printer and paper-tape punch. It is followed by an nve to indicate the number of spaces to tabulate in from the absolute left margin (e.g., ?5=5 spaces from the left margin).
2. as an expression operator, it is followed by a Pattern Specification Code (psc).
3. as the first character printed when a BREAK command or error interrupts a program's execution.

, Comma is used as the term separator in an argument list.

␣ Space is multiply defined:

1. A command followed immediately by two spaces indicates the command has no arguments;
2. One space separates a command from its arguments, or the last argument of a preceding command from the next command on the line.

: Colon is multiply defined:

1. used as a delimiter for field separation in the argument of FOR, MODIFY, and ASSIGN commands.
2. used to indicate the presence of an optional expression appended to a command or the argument of a command (where allowed).

SYMBOL USAGE

Symbol Definition

3. used to indicate the presence of an optional bve appended to a command (:bve may not be appended to FOR, ELSE or IF commands). If the bve is true, the command is executed. If the bve is false, control is passed to the next command on the line or the next line (whichever is applicable). The "next command on the line" is identified by skipping to the second space following the bve. If a bve is appended to a command, no argument of that command may contain a space (i.e., a string literal enclosed in quotes).

; Semicolon is used as a delimiter to indicate the remainder of a line is a comment.

> Right caret is the prompting character used by MUMPS-11 when operating in Direct Mode to signal to the user that the system is ready to accept a command; that is, commands and functions may be entered for immediate execution, or program steps may be entered for program execution.

$ Dollar sign is multiply utilized:

1. precedes the first character of a system variable.
2. precedes the first character of a function name.

% Percent sign is used as the first character of a library program or library global name.

"" Quotation marks are used to delimit literals.

← or _ Back arrow or underscore is used to specify the indirection operation used for command argument replacement.

↑ or ^ Up-arrow or up-caret precedes a global variable reference.

APPENDIX E

CONVERSION TABLES

2^x IN DECIMAL

x	2^x	x	2^x	x	2^x
0.001	1.00069 33874 62581	0.01	1.00695 55500 56719	0.1	1.07177 34625 36293
0.002	1.00138 72557 11335	0.02	1.01395 94797 90029	0.2	1.14869 83549 97035
0.003	1.00208 16050 79633	0.03	1.02101 21257 07193	0.3	1.23114 44133 44916
C.004	1.00277 64359 01078	0.04	1.02811 38266 56067	0.4	1.31950 79107 72894
0.005	1.00347 17485 09503	0.05	1.03526 49238 41377	0.5	1.41421 35623 73095
0.006	1.00416 75432 38973	0.06	1.04246 57608 41121	0.6	1.51571 65665 10398
0.007	1.00486 38204 23785	0.07	1.04971 66836 23067	0.7	1.62450 47927 12471
0.008	1.00556 05803 98468	0.08	1.05701 80405 61380	0.8	1.74110 11265 92248
0.009	1.00625 78234 97782	0.09	1.06437 01824 53360	0.9	1.86606 59830 73615

$10^{\pm n}$ IN OCTAL

10^n	n	10^{-n}	10^n	n	10^{-n}
1	0	1.000 000 000 000 000 000 00	112 402 762 000	10	0.000 000 000 006 676 337 66
12	1	0.063 146 314 631 463 146 31	1 351 035 564 000	11	0.000 000 000 000 537 657 77
144	2	0.005 075 341 217 270 243 66	16 432 451 210 000	12	0.000 000 000 000 043 136 32
1 750	3	0.000 406 111 564 570 651 77	221 411 634 520 000	13	0.000 000 000 000 003 411 35
23 420	4	0.000 032 155 613 530 704 15	2 657 142 036 440 000	14	0.000 000 000 000 000 264 11
303 240	5	0.000 002 476 132 610 706 64	34 327 724 461 500 000	15	0.000 000 000 000 000 022 01
3 641 100	6	0.000 000 206 157 364 055 37	434 157 115 760 200 000	16	0.000 000 000 000 000 001 63
46 113 200	7	0.000 000 015 327 745 152 75	5 432 127 413 542 400 000	17	0.000 000 000 000 000 000 14
575 360 400	8	0.000 000 001 257 143 561 06	67 405 553 164 731 000 000	18	0.000 000 000 000 000 000 01
7 346 545 000	9	0.000 000 000 104 560 276 41			

$n \log_{10} 2$, $n \log_2 10$ IN DECIMAL

n	$n \log_{10} 2$	$n \log_2 10$	n	$n \log_{10} 2$	$n \log_2 10$
1	0.30102 99957	3.32192 80949	6	1.80617 99740	19.93156 85693
2	0.60205 99913	6.64385 61898	7	2.10720 99696	23.25349 66642
3	0.90308 99870	9.96578 42847	8	2.40823 99653	26.57542 47591
4	1.20411 99827	13.28771 23795	9	2.70926 99610	29.89735 28540
5	1.50514 99783	16.60964 04744	10	3.01029 99566	33.21928 09489

ADDITION AND MULTIPLICATION TABLES

Addition

Multiplication

Binary Scale

$$\begin{aligned} 0+0 &= 0 \\ 0+1 = 1+0 &= 1 \\ 1+1 &= 10 \end{aligned}$$

$$\begin{aligned} 0 \times 0 &= 0 \\ 0 \times 1 = 1 \times 0 &= 0 \\ 1 \times 1 &= 1 \end{aligned}$$

Octal Scale

0	01	02	03	04	05	06	07
1	02	03	04	05	06	07	10
2	03	04	05	06	07	10	11
3	04	05	06	07	10	11	12
4	05	06	07	10	11	12	13
5	06	07	10	11	12	13	14
6	07	10	11	12	13	14	15
7	10	11	12	13	14	15	16

1	02	03	04	05	06	07
2	04	06	10	12	14	16
3	06	11	14	17	22	25
4	10	14	20	24	30	34
5	12	17	24	31	36	43
6	14	22	30	36	44	52
7	16	25	34	43	52	61

MATHEMATICAL CONSTANTS IN OCTAL SCALE

$\pi = 3.11037\ 552421_8$	$e = 2.55760\ 521305_8$	$\gamma = 0.44742\ 147707_8$
$\pi^{-1} = 0.24276\ 301556_8$	$e^{-1} = 0.27426\ 530661_8$	$\ln \gamma = -\ 0.43127\ 233602_8$
$\sqrt{\pi} = 1.61337\ 611067_8$	$\sqrt{e} = 1.51411\ 230704_8$	$\log_2 \gamma = -\ .0.62573\ 030645_8$
$\ln \pi = 1.11206\ 404435_8$	$\log_{10} e = 0.33626\ 754251_8$	$\sqrt{2} = 1.32404\ 746320_8$
$\log_2 \pi = 1.51544\ 163223_8$	$\log_2 e = 1.34252\ 166245_8$	$\ln 2 = 0.54271\ 027760_8$
$\sqrt{10} = 3.12305\ 407267_8$	$\log_2 10 = 3.24464\ 741136_8$	$\ln 10 = 2.23273\ 067355_8$

POWERS OF TWO

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
4 398 046 511 104	42	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
8 796 093 022 208	43	0.000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
17 592 186 044 416	44	0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
35 184 372 088 832	45	0.000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
140 737 488 355 328	47	0.000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
281 474 976 710 656	48	0.000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625
562 949 953 421 312	49	0.000 000 000 000 001 776 356 839 400 250 464 677 810 668 945 312 5
1 125 899 906 842 624	50	0.000 000 000 000 000 888 178 419 700 125 232 338 905 334 472 656 25
2 251 799 813 685 248	51	0.000 000 000 000 000 444 089 209 850 062 616 169 452 667 236 328 125
4 503 599 627 370 496	52	0.000 000 000 000 000 222 044 604 925 031 308 084 726 333 618 164 062 5
9 007 199 254 740 992	53	0.000 000 000 000 000 111 022 302 462 515 654 042 363 166 809 082 031 25
18 014 398 509 481 984	54	0.000 000 000 000 000 055 511 151 231 257 827 021 181 583 404 541 015 625
36 028 797 018 963 968	55	0.000 000 000 000 000 027 755 575 615 628 913 510 590 791 702 270 507 812 5
72 057 594 037 927 936	56	0.000 000 000 000 000 013 877 787 807 814 456 755 295 395 851 135 253 906 25
144 115 188 075 855 872	57	0.000 000 000 000 000 006 938 893 903 907 228 377 647 697 925 567 626 953 125
288 230 376 151 711 744	58	0.000 000 000 000 000 003 469 446 951 953 614 188 823 848 962 783 813 476 562 5
576 460 752 303 423 488	59	0.000 000 000 000 000 001 734 723 475 976 807 094 411 924 481 391 906 738 281 25
1 152 921 504 606 846 976	60	0.000 000 000 000 000 000 867 361 737 988 403 547 205 962 240 695 953 369 140 625
2 305 843 009 213 693 952	61	0.000 000 000 000 000 000 433 680 868 994 201 773 602 981 120 347 976 684 570 312 5
4 611 686 018 427 387 904	62	0.000 000 000 000 000 000 216 840 434 497 100 886 801 490 560 173 988 342 285 156 25
9 223 372 036 854 775 808	63	0.000 000 000 000 000 000 108 420 217 248 550 443 400 745 280 086 994 171 142 578 125
18 446 744 073 709 551 616	64	0.000 000 000 000 000 000 054 210 108 624 275 221 700 372 640 043 497 085 571 289 062 5
36 893 488 147 419 103 232	65	0.000 000 000 000 000 000 027 105 054 312 137 610 850 186 320 021 748 542 785 644 531 25
73 786 976 294 838 206 464	66	0.000 000 000 000 000 000 013 552 527 156 068 805 425 093 160 010 874 271 392 822 265 625
147 573 952 589 676 412 928	67	0.000 000 000 000 000 000 006 776 263 578 034 402 712 546 580 005 437 135 696 411 132 812 5
295 147 905 179 352 825 856	68	0.000 000 000 000 000 000 003 388 131 789 017 201 356 273 290 002 718 567 848 205 566 406 25
590 295 810 358 705 651 712	69	0.000 000 000 000 000 000 001 694 065 894 508 600 678 136 645 001 359 283 924 102 783 203 125
1 180 591 620 717 411 303 424	70	0.000 000 000 000 000 000 000 847 032 947 254 300 339 068 322 500 679 641 962 051 391 601 562 5
2 361 183 241 434 822 606 848	71	0.000 000 000 000 000 000 000 423 516 473 627 150 169 534 161 250 339 820 981 025 695 800 781 25
4 722 366 482 869 645 213 696	72	0.000 000 000 000 000 000 000 211 758 236 813 575 084 767 080 625 169 910 490 512 847 900 390 625

OCTAL-DECIMAL CONVERSION

OCTAL-DECIMAL INTEGER CONVERSION TABLE

0000 to 0777 (Octal)	0000 to 0511 (Decimal)

Octal	Decimal
10000	4096
20000	8192
30000	12288
40000	16384
50000	20480
60000	24576
70000	28672

	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

1000 to 1777 (Octal)	0512 to 1023 (Decimal)

	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

OCTAL-DECIMAL INTEGER CONVERSION TABLE (continued)

	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

2000 to 2777 (Octal)	1024 to 1535 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

3000 to 3777 (Octal)	1536 to 2047 (Decimal)

OCTAL-DECIMAL INTEGER CONVERSION TABLE (continued)

4000 to 4777 (Octal)	2048 to 2559 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

5000 to 5777 (Octal)	2560 to 3071 (Decimal)

	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

OCTAL-DECIMAL INTEGER CONVERSION TABLE (continued)

	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

6000 to 6777 (Octal)	3072 to 3583 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	1022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

7000 to 7777 (Octal)	3584 to 4095 (Decimal)

OCTAL-DECIMAL FRACTION CONVERSION TABLE

Octal	Decimal	Octal	Decimal	Octal	Decimal	Octal	Decimal
.000	.000000	.100	.125000	.200	.250000	.300	.375000
.001	.001953	.101	.126953	.201	.251953	.301	.376953
.002	.003906	.102	.128906	.202	.253906	.302	.378906
.003	.005859	.103	.130859	.203	.255859	.303	.380859
.004	.007812	.104	.132812	.204	.257812	.304	.382812
.005	.009765	.105	.134765	.205	.259765	.305	.384765
.006	.011718	.106	.136718	.206	.261718	.306	.386718
.007	.013671	.107	.138671	.207	.263671	.307	.388671
.010	.015625	.110	.140625	.210	.265625	.310	.390625
.011	.017578	.111	.142578	.211	.267578	.311	.392578
.012	.019531	.112	.144531	.212	.269531	.312	.394531
.013	.021484	.113	.146484	.213	.271484	.313	.396484
.014	.023437	.114	.148437	.214	.273437	.314	.398437
.015	.025390	.115	.150390	.215	.275390	.315	.400390
.016	.027343	.116	.152343	.216	.277343	.316	.402343
.017	.029296	.117	.154296	.217	.279296	.317	.404296
.020	.031250	.120	.156250	.220	.281250	.320	.406250
.021	.033203	.121	.158203	.221	.283203	.321	.408203
.022	.035156	.122	.160156	.222	.285156	.322	.410156
.023	.037109	.123	.162109	.223	.287109	.323	.412109
.024	.039062	.124	.164062	.224	.289062	.324	.414062
.025	.041015	.125	.166015	.225	.291015	.325	.416015
.026	.042968	.126	.167968	.226	.292968	.326	.417968
.027	.044921	.127	.169921	.227	.294921	.327	.419921
.030	.046875	.130	.171875	.230	.296875	.330	.421875
.031	.048828	.131	.173828	.231	.298828	.331	.423828
.032	.050781	.132	.175781	.232	.300781	.332	.425781
.033	.052734	.133	.177734	.233	.302734	.333	.427734
.034	.054687	.134	.179687	.234	.304687	.334	.429687
.035	.056640	.135	.181640	.235	.306640	.335	.431640
.036	.058593	.136	.183593	.236	.308593	.336	.433593
.037	.060546	.137	.185546	.237	.310546	.337	.435546
.040	.062500	.140	.187500	.240	.312500	.340	.437500
.041	.064453	.141	.189453	.241	.314453	.341	.439453
.042	.066406	.142	.191406	.242	.316406	.342	.441406
.043	.068359	.143	.193359	.243	.318359	.343	.443359
.044	.070312	.144	.195312	.244	.320312	.344	.445312
.045	.072265	.145	.197265	.245	.322265	.345	.447265
.046	.074218	.146	.199218	.246	.324218	.346	.449218
.047	.076171	.147	.201171	.247	.326171	.347	.451171
.050	.078125	.150	.203125	.250	.328125	.350	.453125
.051	.080078	.151	.205078	.251	.330078	.351	.455078
.052	.082031	.152	.207031	.252	.332031	.352	.457031
.053	.083984	.153	.208984	.253	.333984	.353	.458984
.054	.085937	.154	.210937	.254	.335937	.354	.460937
.055	.087890	.155	.212890	.255	.337890	.355	.462890
.056	.089843	.156	.214843	.256	.339843	.356	.464843
.057	.091796	.157	.216796	.257	.341796	.357	.466796
.060	.093750	.160	.218750	.260	.343750	.360	.468750
.061	.095703	.161	.220703	.261	.345703	.361	.470703
.062	.097656	.162	.222656	.262	.347656	.362	.472656
.063	.099609	.163	.224609	.263	.349609	.363	.474609
.064	.101562	.164	.226562	.264	.351562	.364	.476562
.065	.103515	.165	.228515	.265	.353515	.365	.478515
.066	.105468	.166	.230468	.266	.355468	.366	.480468
.067	.107421	.167	.232421	.267	.357421	.367	.482421
.070	.109375	.170	.234375	.270	.359375	.370	.484375
.071	.111328	.171	.236328	.271	.361328	.371	.486328
.072	.113281	.172	.238281	.272	.363281	.372	.488281
.073	.115234	.173	.240234	.273	.365234	.373	.490234
.074	.117187	.174	.242187	.274	.367187	.374	.492187
.075	.119140	.175	.244140	.275	.369140	.375	.494140
.076	.121093	.176	.246093	.276	.371093	.376	.496093
.077	.123046	.177	.248046	.277	.373046	.377	.498046

OCTAL-DECIMAL FRACTION CONVERSION TABLE (continued)

Octal	Decimal	Octal	Decimal	Octal	Decimal	Octal	Decimal
.000000	.000000	.000100	.000244	.000200	.000488	.000300	.000732
.000001	.000003	.000101	.000247	.000201	.000492	.000301	.000736
.000002	.000007	.000102	.000251	.000202	.000495	.000302	.000740
.000003	.000011	.000103	.000255	.000203	.000499	.000303	.000743
.000004	.000015	.000104	.000259	.000204	.000503	.000304	.000747
.000005	.000019	.000105	.000263	.000205	.000507	.000305	.000751
.000006	.000022	.000106	.000267	.000206	.000511	.000306	.000755
.000007	.000026	.000107	.000270	.000207	.000514	.000307	.000759
.000010	.000030	.000110	.000274	.000210	.000518	.000310	.000762
.000011	.000034	.000111	.000278	.000211	.000522	.000311	.000766
.000012	.000038	.000112	.000282	.000212	.000526	.000312	.000770
.000013	.000041	.000113	.000286	.000213	.000530	.000313	.000774
.000014	.000045	.000114	.000289	.000214	.000534	.000314	.000778
.000015	.000049	.000115	.000293	.000215	.000537	.000315	.000782
.000016	.000053	.000116	.000297	.000216	.000541	.000316	.000785
.000017	.000057	.000117	.000301	.000217	.000545	.000317	.000789
.000020	.000061	.000120	.000305	.000220	.000549	.000320	.000793
.000021	.000064	.000121	.000308	.000221	.000553	.000321	.000797
.000022	.000068	.000122	.000312	.000222	.000556	.000322	.000801
.000023	.000072	.000123	.000316	.000223	.000560	.000323	.000805
.000024	.000076	.000124	.000320	.000224	.000564	.000324	.000808
.000025	.000080	.000125	.000324	.000225	.000568	.000325	.000812
.000026	.000083	.000126	.000328	.000226	.000572	.000326	.000816
.000027	.000087	.000127	.000331	.000227	.000576	.000327	.000820
.000030	.000091	.000130	.000335	.000230	.000579	.000330	.000823
.000031	.000095	.000131	.000339	.000231	.000583	.000331	.000827
.000032	.000099	.000132	.000343	.000232	.000587	.000332	.000831
.000033	.000102	.000133	.000347	.000233	.000591	.000333	.000835
.000034	.000106	.000134	.000350	.000234	.000595	.000334	.000839
.000035	.000110	.000135	.000354	.000235	.000598	.000335	.000843
.000036	.000114	.000136	.000358	.000236	.000602	.000336	.000846
.000037	.000118	.000137	.000362	.000237	.000606	.000337	.000850
.000040	.000122	.000140	.000366	.000240	.000610	.000340	.000854
.000041	.000125	.000141	.000370	.000241	.000614	.000341	.000858
.000042	.000129	.000142	.000373	.000242	.000617	.000342	.000862
.000043	.000133	.000143	.000377	.000243	.000621	.000343	.000865
.000044	.000137	.000144	.000381	.000244	.000625	.000344	.000869
.000045	.000141	.000145	.000385	.000245	.000629	.000345	.000873
.000046	.000144	.000146	.000389	.000246	.000633	.000346	.000877
.000047	.000148	.000147	.000392	.000247	.000637	.000347	.000881
.000050	.000152	.000150	.000396	.000250	.000640	.000350	.000885
.000051	.000156	.000151	.000400	.000251	.000644	.000351	.000888
.000052	.000160	.000152	.000404	.000252	.000648	.000352	.000892
.000053	.000164	.000153	.000408	.000253	.000652	.000353	.000896
.000054	.000167	.000154	.000411	.000254	.000656	.000354	.000900
.000055	.000171	.000155	.000415	.000255	.000659	.000355	.000904
.000056	.000175	.000156	.000419	.000256	.000663	.000356	.000907
.000057	.000179	.000157	.000423	.000257	.000667	.000357	.000911
.000060	.000183	.000160	.000427	.000260	.000671	.000360	.000915
.000061	.000186	.000161	.000431	.000261	.000675	.000361	.000919
.000062	.000190	.000162	.000434	.000262	.000679	.000362	.000923
.000063	.000194	.000163	.000438	.000263	.000682	.000363	.000926
.000064	.000198	.000164	.000442	.000264	.000686	.000364	.000930
.000065	.000202	.000165	.000446	.000265	.000690	.000365	.000934
.000066	.000205	.000166	.000450	.000266	.000694	.000366	.000938
.000067	.000209	.000167	.000453	.000267	.000698	.000367	.000942
.000070	.000213	.000170	.000457	.000270	.000701	.000370	.000946
.000071	.000217	.000171	.000461	.000271	.000705	.000371	.000949
.000072	.000221	.000172	.000465	.000272	.000709	.000372	.000953
.000073	.000225	.000173	.000469	.000273	.000713	.000373	.000957
.000074	.000228	.000174	.000473	.000274	.000717	.000374	.000961
.000075	.000232	.000175	.000476	.000275	.000720	.000375	.000965
.000076	.000236	.000176	.000480	.000276	.000724	.000376	.000968
.000077	.000240	.000177	.000484	.000277	.000728	.000377	.000972

OCTAL-DECIMAL FRACTION CONVERSION TABLE (continued)

Octal	Decimal	Octal	Decimal	Octal	Decimal	Octal	Decimal
.000400	.000976	.000500	.001220	.000600	.001464	.000700	.001708
.000401	.000980	.000501	.001224	.000601	.001468	.000701	.001712
.000402	.000984	.000502	.001228	.000602	.001472	.000702	.001716
.000403	.000988	.000503	.001232	.000603	.001476	.000703	.001720
.000404	.000991	.000504	.001235	.000604	.001480	.000704	.001724
.000405	.000995	.000505	.001239	.000605	.001483	.000705	.001728
.000406	.000999	.000506	.001243	.000606	.001487	.000706	.001731
.000407	.001003	.000507	.001247	.000607	.001491	.000707	.001735
.000410	.001007	.000510	.001251	.000610	.001495	.000710	.001739
.000411	.00101C	.000511	.001255	.000611	.001499	.000711	.001743
.000412	.001014	.000512	.001258	.000612	.001502	.000712	.001747
.000413	.001018	.000513	.001262	.000613	.001506	.000713	.001750
.000414	.001022	.000514	.001266	.000614	.001510	.000714	.001754
.000415	.001026	.000515	.001270	.000615	.001514	.000715	.001758
.000416	.001029	.000516	.001274	.000616	.001518	.000716	.001762
.000417	.001033	.000517	.001277	.000617	.001522	.000717	.001766
.000420	.001037	.000520	.001281	.000620	.001525	.000720	.001770
.000421	.001041	.000521	.001285	.000621	.001529	.000721	.001773
.000422	.001045	.000522	.001289	.000622	.001533	.000722	.001777
.000423	.001049	.000523	.001293	.000623	.001537	.000723	.001781
.000424	.001052	.000524	.001296	.000624	.001541	.000724	.001785
.000425	.001056	.000525	.001300	.000625	.001544	.000725	.001789
.000426	.001060	.000526	.001304	.000626	.001548	.000726	.001792
.000427	.001064	.000527	.001308	.000627	.001552	.000727	.001796
.000430	.001068	.000530	.001312	.000630	.001556	.000730	.001800
.000431	.001071	.000531	.001316	.000631	.001560	.000731	.001804
.000432	.001075	.000532	.001319	.000632	.001564	.000732	.001808
.000433	.001079	.000533	.001323	.000633	.001567	.000733	.001811
.000434	.001083	.000534	.001327	.000634	.001571	.000734	.001815
.000435	.001087	.000535	.001331	.000635	.001575	.000735	.001819
.000436	.001091	.000536	.001335	.000636	.001579	.000736	.001823
.000437	.001094	.000537	.001338	.000637	.001583	.000737	.001827
.000440	.001098	.000540	.001342	.000640	.001586	.000740	.001831
.000441	.001102	.000541	.001346	.000641	.001590	.000741	.001834
.000442	.001106	.000542	.001350	.000642	.001594	.000742	.001838
.000443	.001110	.000543	.001354	.000643	.001598	.000743	.001842
.000444	.001113	.000544	.001358	.000644	.001602	.000744	.001846
.000445	.001117	.000545	.001361	.000645	.001605	.000745	.001850
.000446	.001121	.000546	.001365	.000646	.001609	.000746	.001853
.000447	.001125	.000547	.001369	.000647	.001613	.000747	.001857
.000450	.001129	.000550	.001373	.000650	.001617	.000750	.001861
.000451	.001132	.000551	.001377	.000651	.001621	.000751	.001865
.000452	.001136	.000552	.001380	.000652	.001625	.000752	.001869
.000453	.001140	.000553	.001384	.000653	.001628	.000753	.001873
.000454	.001144	.000554	.001388	.000654	.001632	.000754	.001876
.000455	.001148	.000555	.001392	.000655	.001636	.000755	.001880
.000456	.001152	.000556	.001396	.000656	.001640	.000756	.001884
.000457	.001155	.000557	.001399	.000657	.001644	.000757	.001888
.000460	.001159	.000560	.001403	.000660	.001647	.000760	.001892
.000461	.001163	.000561	.001407	.000661	.001651	.000761	.001895
.000462	.001167	.000562	.001411	.000662	.001655	.000762	.001899
.000463	.001171	.000563	.001415	.000663	.001659	.000763	.001903
.000464	.001174	.000564	.001419	000664	.001663	.000764	.001907
.000465	.001178	.000565	.001422	.000665	.001667	.000765	.001911
.000466	.001182	.000566	.001426	.000666	.001670	.000766	.001914
.000467	.001186	.000567	.001430	.000667	.001674	.000767	.001918
.000470	.001190	.000570	.001434	.000670	.001678	.000770	.001922
.000471	.001194	.000571	.001438	.000671	.001682	.000771	.001926
.000472	.001197	.000572	.001441	.000672	.001686	.000772	.001930
.000473	.001201	.000573	.001445	.000673	.001689	.000773	.001934
.000474	.001205	.000574	.001449	.000674	.001693	.000774	.001937
.000475	.001209	.000575	.001453	.000675	.001697	.000775	.001941
.000476	.001213	.000576	.001457	.000676	.001701	.000776	.001945
.000477	.001216	.000577	.001461	.000677	.001705	.000777	.001949

APPENDIX F

↑SYS SYSTEM GLOBAL LAYOUT

↑SYS is a special global that contains both system configuration data and hardware error statistics. The global is created during system generation by the SYSGEN program and is under the control of the System UCI (UCI Number 1). The first seventeen nodes of ↑SYS (Figure F-1) contain system configuration data obtained during system generation. The 0.01 through 0.07 and 10 through 17 nodes are used for temporary storage by SYSGEN. Nodes 18 through 64 (Figure F-2) contain system hardware error statistics compiled and updated by the System Caretaker Program (CTK). The data in these nodes are reported to the user by the Caretaker Reporter Program (KTR).

↑SYS SYSTEM GLOBAL LAYOUT

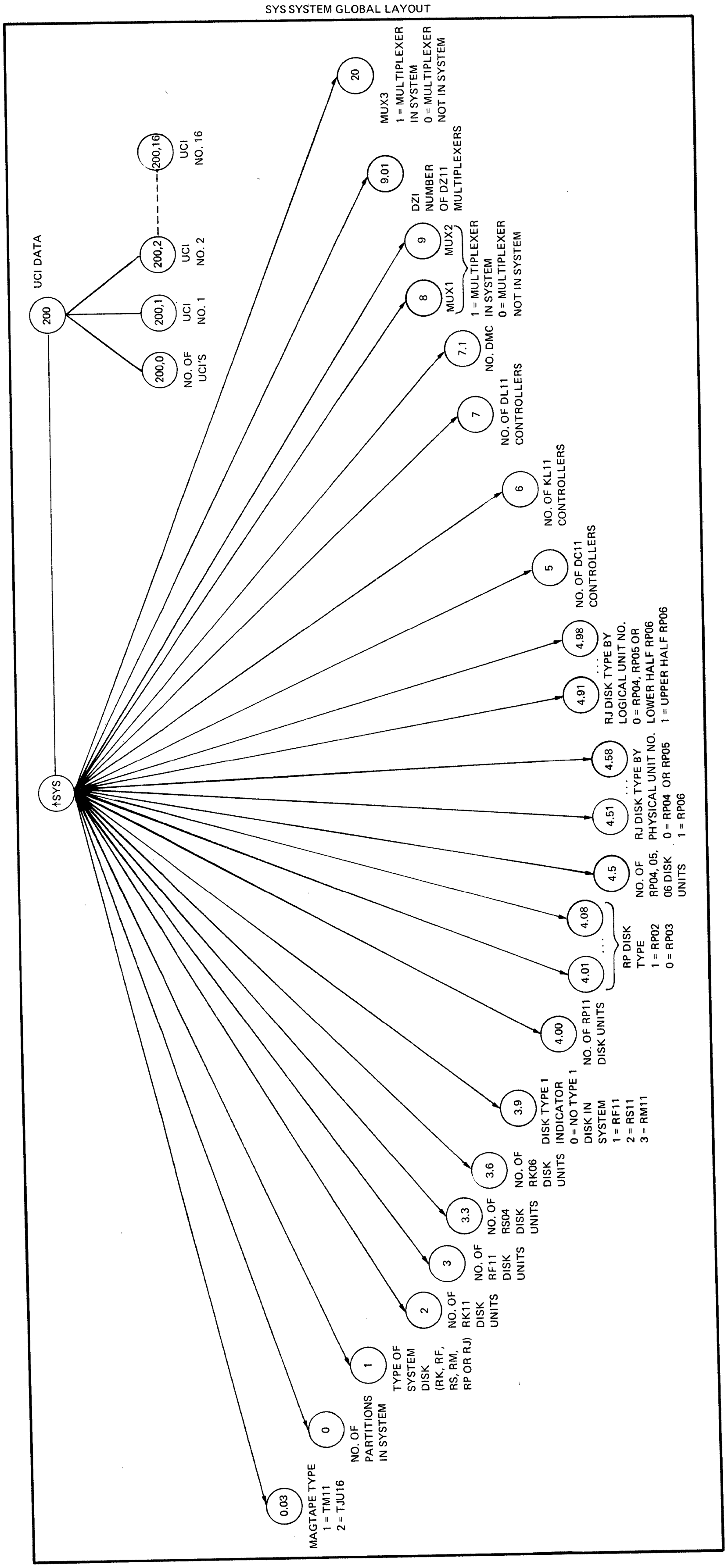

SYS SYSTEM GLOBAL LAYOUT
↑SYS
200
UCI DATA
200,0
NO. OF UCI'S
200,1
UCI NO. 1
200,2
UCI NO. 2
200,16
UCI NO. 16
0.03
MAGTAPE TYPE 1 = TM11 2 = TJU16
0
NO. OF PARTITIONS IN SYSTEM
1
TYPE OF SYSTEM DISK (RK, RF, RS, RM, RP OR RJ)
2
NO. OF RK11 DISK UNITS
3
NO. OF RF11 DISK UNITS
3.3
NO. OF RS04 DISK UNITS
3.6
NO. OF RK06 DISK UNITS
3.9
DISK TYPE 1 INDICATOR 0 = NO TYPE 1 DISK IN SYSTEM 1 = RF11 2 = RS11 3 = RM11
4.00
NO. OF RP11 DISK UNITS
4.01
4.08
RP DISK TYPE 1 = RP02 0 = RP03
4.5
NO. OF RP04, 05, 06 DISK UNITS
4.51
4.58
RJ DISK TYPE BY PHYSICAL UNIT NO. 0 = RP04 OR RP05 1 = RP06
4.91
4.98
RJ DISK TYPE BY LOGICAL UNIT NO. 0 = RP04, RP05 OR LOWER HALF RP06 1 = UPPER HALF RP06
5
NO. OF DC11 CONTROLLERS
6
NO. OF KL11 CONTROLLERS
7
NO. OF DL11 CONTROLLERS
7.1
NO. DMC
8
9
MUX1
MUX2
1 = MULTIPLEXER IN SYSTEM 0 = MULTIPLEXER NOT IN SYSTEM
9.01
DZI NUMBER OF DZ11 MULTIPLEXERS
20
MUX3 1 = MULTIPLEXER IN SYSTEM 0 = MULTIPLEXER NOT IN SYSTEM

Figure F-1 ↑SYS Nodes Built By SYSGEN

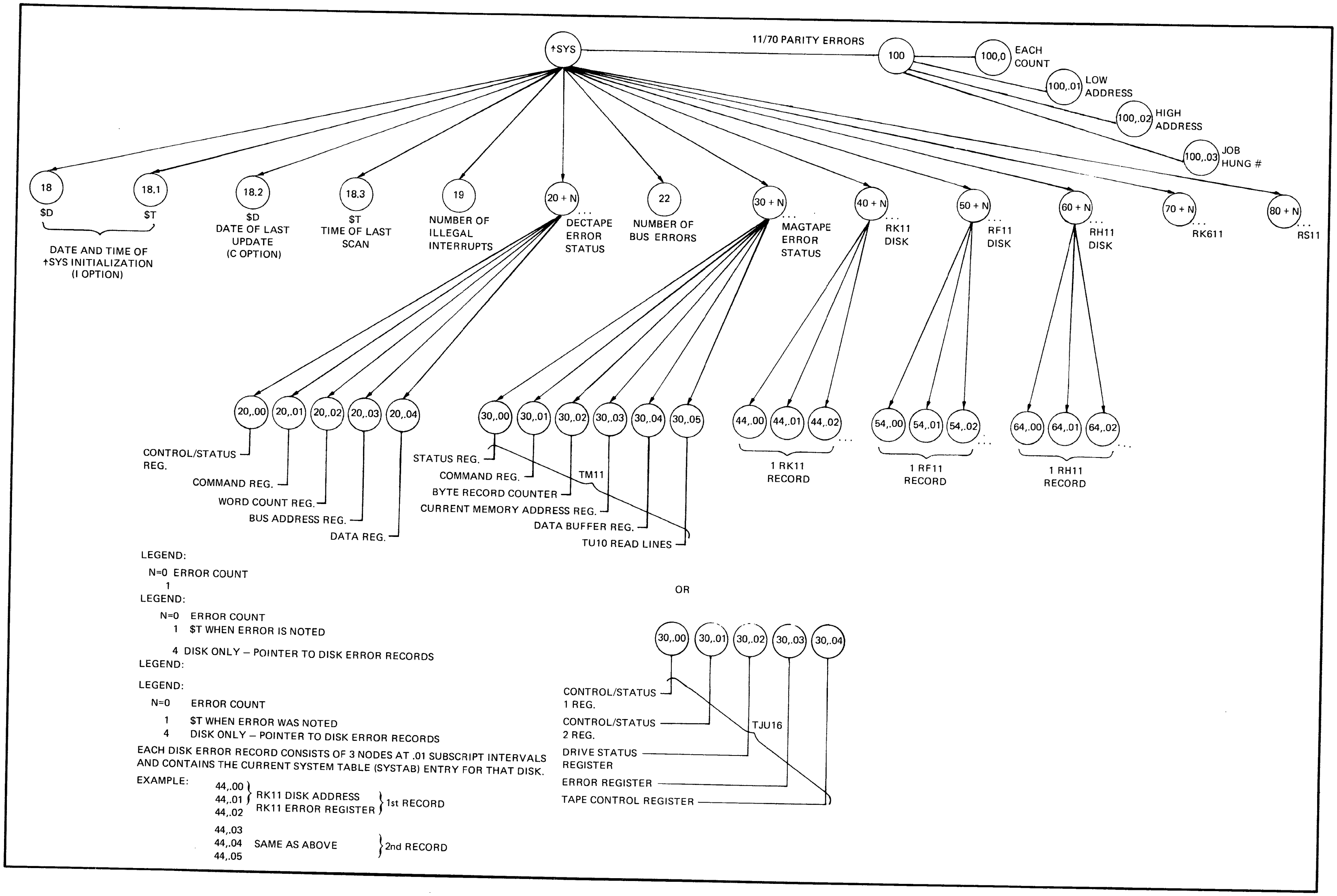


Figure F-2 ↑SYS Layout for CTK Data

APPENDIX G

MAGTAPE LOADING AND UNLOADING PROCEDURES

G.1 LOADING AND THREADING TAPE

Use the following procedure to mount and thread the tape:

Step — Procedure

1 Apply power to the transport by depressing PWR ON switch.

2 Ensure that the LOAD/BR REL switch is in the center position (this applies the brakes).

3 Place a write enable ring in the groove on the file reel if data is to be written on the tape.

Ensure there is no ring in the groove if data on the tape is not to be erased or written over.

4 Mount the file reel onto the lower hub with the groove facing towards the back. Ensure that the reel is firmly seated against the flange of the hub.

5 Install the take-up reel (top) as described in Step 4.

6 Move LOAD/BR REL switch to the BR REL position.

7 Unwind tape from the file reel and thread the tape over the tape quides and head assembly.

8 Wind about five turns of tape onto the take-up reel.

9 Set the LOAD/BR REL switch to the LOAD position to draw tape into the vacuum columns.

10 Select FWD and press START to advance the tape to Load Point. When the BOT marker is sensed, tape motion stops, the FWD indicator goes out, and the LOAD PT indicator comes on.

NOTE

If tape motion continues for more than 10 seconds, press STOP, select REV (reverse) and press START. The tape should move to the BOT marker (Load Point) before stopping.

G.2 UNLOADING TAPE

To unload the tape, proceed as follows:

1. Press OFF-LINE switch if the transport has been operating in the on-line mode.
2. Press STOP switch and select REW.
3. Press START switch. The tape should rewind until the BOT marker is reached.
4. Press the LOAD/BR REL switch to release the brakes.
5. Gently hand wind the file reel in a counterclockwise direction until all of the tape is wound onto the reel.

NOTE

When handwinding the tape, do not jerk the reel. This can stretch or compress the tape which could cause irreparable damage.

6. Remove the file reel from the hub assembly.

APPENDIX H

NON-STANDARD TERMINAL DEVICE CONNECTION PROCEDURES

H.1 INTRODUCTION

This appendix describes the procedures for connecting non-standard terminal devices to the system's terminal I/O driver (IOD). Non-standard terminal devices are any devices that can be connected to standard DEC single-line communication controllers, such as models DL11, KL11 and DC11. Two examples of DIGITAL-manufactured non-standard terminal devices are the RT02 data entry terminal and a second LP11 or LS11 line printer.

Before performing these procedures, each device and its associated controller must be connected into the system's hardware by a DEC Field Service representative. Further, the user must obtain the addresses of the interrupt vector and I/O registers for each device from the DEC representative.

H.2 PROCEDURE

1. Perform System Generation (Chapter 3) and specify the existence and characteristics of the non-standard terminal device(s) being used (see 3.8.2.6 and 3.8.7).

2. After System Generation (SYSGEN) finishes its operation, do not save the system as instructed in 3.8.10; instead, perform the following steps:

   a. Determine the base address of the device's device descriptor buffer (DDB)(1) from the formula:

      DDB Base Address = C(SYSTAB+6)+(32*(dn-1)

      Where:

      C = "contents of"

      SYSTAB = Base Address of MUMPS-11 System Table[1]

      dn = Device Number

   b. Using Mini-ODT (Chapter 4)

      1. Enter the base address of the DDB in the first word of the device's interrupt vector (VECTOR+0).

[1] Refer to Appendix F in the MUMPS-11 Programmer's Guide for further information.

   2. Enter the constant 300(8) (PSW value) in the second word of the vector (VECTOR+2).

   3. Enter the address of the device's receiver status register in DDB+4 of the device descriptor buffer. If the device has no receiver, enter a dummy relative address. This is calculated by subtracting 4 from the address of the device's transmitter status register.

   4. Enter the initial values for the device's status registers in DDB+6 (receiver register) and DDB+10 (transmitter register). For example, the initial value for the receiver status register of a DL11A controller that is connected to a terminal's keyboard would be 100(8). If a transmit-only device is used, the controller's receiver status register must contain a 0.

3. When all the steps above are completed, the device is connected to the system and the new system should be saved as described in Section 3.8.10.

APPENDIX I

MUMPS BACKUP AND UTILITY SYSTEM

I.1 INTRODUCTION

The MUMPS Backup and Utility System (MBU) is a bootable, stand-alone system. MBU allows you to back up or save the significant data from MUMPS disks so that in the event of a system failure, the disks can be restored to their former states. MBU also allows you to save individual globals for later restoring via the MUMPS %LR Logical Restore program. In addition, MBU performs the following important utility functions:

1. labels disks, DECtapes and magtapes, for indentification purposes and for prevention of inadvertent destruction of important data;

2. formats and tests disks; also initializes disks to be used in a MUMPS environment;

3. makes exact image copies of magtape, DECtape and disk volumes to allow you to back up non-MUMPS data;

4. allows you to directly allocate or deallocate individual blocks on a MUMPS disk;

5. performs automatic handling of "bad" (known to be unusable) blocks. "Bad" blocks are any blocks which cause errors during formatting or testing, in addition to blocks which you declare explicitly to be bad.

Instructions for creating a bootable image of MBU on DECtape or magtape are contained in Section I.7.

I.2 HOW TO USE MBU

Once loaded into memory, MBU begins executing automatically. Thereafter you need only answer the question it asks in order to have it perform the operations desired. If you are in doubt as to the way in which a particular question should be answered, you may type the single character "H" (followed by a Carriage RETURN), and you will receive a list of sample answers to the question, showing the format in which the answer is expected. Note that if you type "H" in response to the very first question MBU asks, you will receive an "MBU Command Summary" -- a brief description of all MBU commands, explaining the use of each. A sample MBU command summary is reproduced in Section I.8 of this Appendix.

I.3 BEGINNERS' GUIDE TO MBU

This section is intended for operators who are not yet familiar with MBU and who wish to perform simple operations, such as labeling disks, backing up disks to magtape, making image copies of disks, DECtapes or magtapes, or formatting disks. The section consists entirely of an output listing from an MBU session during which the operator actually performed these functions. Comments have been added to explain the operator's intentions. (Note: All comments are in small letters and are enclosed in parentheses. The computer output itself, as printed at the operator's console, is in CAPITAL LETTERS: everything typed by the operator is in UNDERLINED CAPITAL LETTERS; the rest was typed automatically by MBU.)

For more detailed explanations of the commands shown here, and for descriptions of the remaining MBU commands, see Section I.4.

```
     (start of sesson:  operator boots MBU, from disk, DECtape or
     magtape;  it prints introductory message and asks for first
     command . . . operator will put a label on the disk  mounted
     on RK05 (RK) unit zero)

          MUMPS BACKUP UTILITY . . . V01.2
          TYPE  H  FOR HELP AT ANY TIME

YOU MAY TYPE  H  NOW, FOR A SUMMARY
OF ALL MBU COMMANDS AVAILABLE.
(36. BUFFERS AVAILABLE)

COMMAND? LABEL
DEVICE & UNIT? RK0
(NO MBU LABEL CURRENTLY ON VOLUME)
DO YOU WISH TO PUT A NEW LABEL ON VOLUME (Y OR N) ? Y
MASTER LABEL (MA) OR BACKUP LABEL (BA) ? MA
NEW LABEL: "RK DISK 1, 2/1/77"
OPERATION COMPLETE.

     (operator now wishes to check the labels on RP04 (RJ)  units
     zero and one)

COMMAND? LABEL
DEVICE & UNIT? RJ0
MASTER LABEL CURRENTLY ON VOLUME IS: "MUMPS SYS 2/13"
DO YOU WISH TO PUT A NEW LABEL ON VOLUME (Y OR N) ? N
OPERATION COMPLETE.

COMMAND? LABEL RJ1
MASTER LABEL CURRENTLY ON VOLUME IS: "RJ DATA DISK"
DO YOU WISH TO PUT A NEW LABEL ON VOLUME (Y OR N) ? N
OPERATION COMPLETE.

     (operator is now going to back up the contents of  disk  RK0
     to TU16 magtape unit MM2)

COMMAND? BACKUP
MASTER UNIT TO BE BACKED UP? RK0
LABEL: "RK DISK 1, 2/1/7"
ENTIRE VOLUME (EN) OR ONLY MUMPS BLOCKS-IN-USE (IU) ? EN
BACKUP UNIT TO BE WRITTEN TO? MN2
LABEL: "MY RK SYS BACKUP"
(NO MBU LABEL CURRENTLY ON VOLUME)
DO YOU WISH TO PROCEED, RELABELING THIS VOLUME (Y OR N) ? Y
VERIFY WHEN DONE (Y OR N) ? Y
** BEGIN BACKUP
** BEGIN VERIFY
** END VERIFY
OPERATION COMPLETE.
```

(operator will now check label on magtape MM2)

```
COMMAND? LA MM2
BACKUP LABEL CURRENTLY ON VOLUME IS: "MY RK SYS BACKUP"
BACKUP VOL. 1 OF RK DISK "RK DISK 1, 2/1/77"
DO YOU WISH TO PUT A NEW LABEL ON VOLUME (Y OR N) ? N
OPERATION COMPLETE.
```

(now the operator will format a new disk on RK unit 1, and make an image-copy of the RK0 on this new disk)

```
COMMAND? FORMAT
DEVICE & UNIT? RK1
FORMAT RK1    ARE YOU SURE (Y OR N) Y
** BEGIN FORMATTING
** END FORMATTING
DO YOU WISH TO EXECUTE TEST COMMAND (Y OR N) ? Y
TEST RK1    ARE YOU SURE (Y OR N) ? Y
TEST PATTERN 1 (Y OR N) ? Y
** BEGIN TEST
** END TEST
TEST PATTERN 2 (Y OR N) ? N
TEST PATTERN 3 (Y OR N) ? N
TEST PATTERN 4 (Y OR N) ? N
DO YOU WISH TO INITIALIZE AS A MUMPS VOLUME (Y OR N) ? N
BACKUP LABEL CURRENTLY ON VOLUME IS: "*MBU*"
DO YOU WISH TO PUT A NEW LABEL ON VOLUME (Y OR N) ? N
OPERATION COMPLETE.

COMMAND? COPY
MASTER ("FROM") UNIT? RK0
MASTER LABEL CURRENTLY ON VOLUME IS: "RK DISK 1, 2/1/77"
BACKUP ("TO") UNIT? RK1
BACKUP LABEL CURRENTLY ON VOLUME IS: "*MBU*"
COPY "SPARE" BLOCKS TOO (Y OR N) ? Y
COMPARE WHEN DONE (Y OR N) ? Y
** BEGIN COPY
(NOTE: LABEL BLOCK WILL NOT BE COPIED)
** BEGIN COMPARE
(NOTE: LABEL BLOCK WILL NOT BE COMPARED)
** END COMPARE
OPERATION COMPLETE.
```

I.4 MBU COMMAND DESCRIPTIONS

I.4.1 The LABEL Command

The LABEL command is used to place a new MBU label on a volume or to examine the MBU label already on it. There are two types of MBU labels. A "master" label should be placed on a volume if it contains important data which you do not wish inadvertently destroyed. MBU will help you protect the volume by: (a) not allowing you to use a master volume as the "BACKUP" or "TO" volume in any "BACKUP", "COPY" or "SAVE" operation, and (b) informing you that there is a master label on the volume if you are about to FORMAT or INITIALIZE the volume or, you may place a "backup" label on a volume if it does not contain important data and you are labeling it for identification purposes only, or if it is to be used as the "BACKUP" or "TO" volume in a "BACKUP", "COPY" or "SAVE" operation.

See Section I.3 for examples demonstrating the use of the MBU LABEL command.

Notes:

1. When you are asked to specify an MBU label, you specify a string of up to 20 characters, enclosed in quotes, like this: "THE MBU LABEL"

2. When an MBU label is placed on a volume, what is actually written is an entire block (256. computer words) of information, including:

   a. the label string you specified,

   b. information identifying the label as either a master or a backup label,

   c. the volume's bad-block table, if there are any known, unusable blocks on the volume, and

   d. if the volume is a backup, a complete record of the master volume from which it came, including the master's disk-type, (RK, RJ, RM, etc.) and its 20 character label string.

3. When an MBU label is written on a disk, it is written near the end of the spare-block area of the disk. Whereas MUMPS (and most other operating systems) never use the spare-block area, you may safely place an MBU label on a MUMPS disk, even if it already contains data. (The actual disk addresses (block numbers) where labels are written for the various types of disks can be found in Table I-4.)

I.4.2 The BACKUP Command

The BACKUP command allows you to back up or save, the contents of any single disk volume, on any backup medium of your choice -- disk, DECtape or magtape. If the information being backed up will not fit on one backup volume, MBU will ask you to mount a second volume, and so on, until all the required information has been backed up. MBU will automatically number the backup volumes sequentially (starting with the number 1), and these volume numbers may later be inspected at any time using the LABEL command. When restoring from a multi-volume backup, MBU will, of course, check the backup volume numbers and inform you if you have mounted a backup volume out of sequence.

Most of the questions MBU asks in connection with the BACKUP command are extremely straightforward. The following two, however, require detailed explanations.

(1) ENTIRE VOLUME (EN) OR ONLY MUMPS BLOCKS-IN-USE (IU) ?

If you respond EN to this question, MBU will backup the entire master volume you have specified -- that is, all blocks of the specified master disk except the spare blocks. Note that the spare blocks are never used by MUMPS, so typing EN guarantees that all blocks ever accessible to MUMPS will be backed up.

If you respond IU to the above question, MBU will back up only (a) those blocks from the master volume that are specifically indicated as in-use (allocated) in the MUMPS bit maps on the volume, plus (b) any blocks contained in SDP areas on the volume.

IU is particularly useful when the master disk being backed up contains a significant number of free, or unused blocks. In this case, backing up only the in-use blocks will result in a considerable saving of space on the backup volume(s) with a corresponding saving of time.

You should, of course, not respond IU to the above question if the disk you are backing up is not a MUMPS disk.

(2) VERIFY WHEN DONE (Y OR N) ?

This question is asked by MBU just before the backup operation actually begins. If you respond Y (YES), then after the backup operation is complete, MBU will check all information it has written on the backup volume(s) to make certain that it is in fact identical to the information from the master that was to be backed up. It is recommended that you always answer Y to this question when backing up important data.

An example demonstrating how the BACKUP command is used is shown in Section I.3.

I.4.3 The RESTORE Command

The RESTORE command is used to restore a disk volume that was backed up using the BACKUP command. The following sample MBU session demonstrates how the RESTORE command is used.

```
COMMAND? RESTORE
MASTER UNIT TO BE RESTORED TO? RK5
LABEL: "RK DISK 1, 2/1/77"
BACKUP UNIT TO BE RESTORED FROM? MT4
LABEL: "MY RK SYS BACKUP"
VERIFY WHEN DONE (Y OR N) ? Y
** BEGIN RESTORE
** BEGIN VERIFY
** END VERIFY
OPERATION COMPLETE.
```

I.4.4 The COPY And COMPARE Commands

The COPY command is used to make an exact image-copy of a disk, DECtape or magtape volume on another disk (of the exact same type), DECtape or magtape volume, respectively. The copy may then be used in place of the original.

For DECtapes, all blocks will be copied. For magtapes, everything up to and including the first occurrence of two consecutive end-of-file marks (tape marks) will be copied. In both of these cases, you will be asked to specify a MASTER ("FROM") UNIT and a BACKUP ("TO") UNIT. The "FROM" unit is the original you wish to make a copy of -- if possible, you should write-protect it. The "TO" unit is the one that will be written on.

When the volumes are disks, you are asked the additional question:

COPY "SPARE" BLOCKS TO (Y OR N) ?

r& the disk being copied is a MUMPS disk, or if you are certain the spare-block area is not being used, you may answer N to this question. If you answer Y to this question, the following rule holds true: if either disk contains a non-blank MBU label, then the last 5 spare blocks, which include the label block and bad-block table, will not be copied. This is to prevent the "TO" volume from acquiring an incorrect bad-block table.

Just before copying actually begins, you will be asked the question:

COMPARE WHEN DONE (Y OR N) ?

If you answer Y, then after the copying is completed, all blocks that have been copied will be compared to see that they were copied correctly. An example showing how the COPY command is used is given in Section I.3.

Notes:

1. If either volume contains known bad blocks (or, in other words, has anything in its bad-block table), you will be asked the additional question:

   ERROR MESSAGES EVEN FOR KNOWN BAD BLOCKS (Y OR N) ?

   If you answer Y to this question, MBU will inform you of errors even if they occur in reading or writing known bad blocks (where errors are expected).

2. The COMPARE command may be issued independently without first executing a COPY command -- you may type COMPARE in response to MBU's "ND?" prompt. The questions you will be asked are similar to those asked by the COPY command.

I.4.5 The SAVE-GLOBALS Command

The SAVE-GLOBALS (or SAVE) command allows you to back up individual globals, or all globals, from one or more specific MUMPS UCI's to either magtape or DECtape. The globals may later be restored using the MUMPS %LR Logical Restore utility. When using the SAVE command, you specify as the "SYSTEM MASTER DEVICE & UNIT" the unit on which your MUMPS system resides. For additional master units, you specify any other MUMPS disks in your system on which portions of the globals to be backed up may reside. For the "BACKUP UNIT", you specify the magtape or DECtape unit that is to hold the backed up globals.

You may either:

1. save all globals from a given MUMPS UCI, or

2. type in explicitly the names of the globals you wish to save from that UCI.

When MBU has determined which globals you wish to save from a given UCI, it saves them, then it asks:

```
NEXT UCI (TYPE <CR> WHEN DONE) ?
```

thereby allowing you to save globals from as many UCI's as you wish in one SAVE operation. If the information being saved will not fit on one backup volume, MBU will request a second volume, and so on, until all the required globals have been saved.

Following is a sample MBU session showing how you would save globals ↑A, ↑B and ↑C from the system manager's UCI (assume UCI code = "SYS"), and all globals from UCI "HAL". Assume your MUMPS system resides on unit RK0, with additional units RK1 and RJ0 having been defined as part of the system during your MUMPS SYSGEN. The backup medium will be magtape.

```
COMMAND? SAVE
SYSTEM MASTER DEVICE & UNIT? DK0
LABEL: "MUMPS SYS - 1/17/77"
NEXT MASTER DEVICE & UNIT (TYPE <CR> WHEN DONE) ? DK1
(NO MBU LABEL CURRENTLY ON VOLUME)
NEXT MASTER DEVICE & UNIT (TYPE <CR> WHEN DONE) ? RJ0
MASTER LABEL CURRENTLY ON VOLUME IS:  "**MOUNT ON RJ0 **"
NEXT MASTER DEVICE & UNIT (TYPE <CR> WHEN DONE) ? (operator typed
Carriage RETURN)
BACKUP UNIT? MT1
LABEL:  "3/1/77 GLOBALS"
(LABEL BLOCK UNREADABLE)
DO YOU WISH TO PROCEED, RE-LABELING THIS VOLUME (Y OR N) ? Y

NEXT UCI (TYPE <CR> WHEN DONE) ? SYS
TYPE  H  FOR HELP
NEXT GLOBAL? ↑A
NEXT GLOBAL? ↑B
NEXT GLOBAL? ↑C
NEXT GLOBAL?    (operator typed Carriage RETURN)
** BEGIN SAVE
↑A SAVED
↑B SAVED
↑C SAVED
** END SAVE

NEXT UCI (TYPE <CR> WHEN DONE) ? HAL
TYPE  H  FOR HELP
NEXT GLOBAL? ALL
** BEGIN SAVE
↑H1 SAVED
↑ABC SAVED
↑MAX SAVED
↑LST SAVED
** END SAVE

NEXT UCI (TYPE <CR> WHEN DONE) ?   (operator typed Carriage RETURN)
OPERATION COMPLETE.
```

I.4.5.1 Redirecting A Physical Unit - With most MBU commands, you are free to mount a volume to be used with that command on any available unit of the correct type. For example, if you wish to back up an RK06 disk, you may mount the disk cartridge on drive RM0, or RM1, or RM2, and so on. With the SAVE-GLOBALS command, however, all disks must be mounted on exactly the units they would be mounted on if you were running your MUMPS system. This allows MBU to correctly follow the disk block pointers, which tell which blocks on which disks contain the various parts of the globals you are saving.

When a disk drive that you would normally need for saving globals is "down" (inoperative), you can inform MBU that the disk you would normally mount on that drive is (today) mounted on some other drive. You do this with the SET ... PHYSU= ... command. Following is an example that demonstrates the use of this command. Assume that you are going to save globals, as in the example in Section I.4.5, but drive RK0 is "down". You might mount that disk on unit RK3 instead. You would then proceed as follows:

```
COMMAND? SET RK0 PHYSU=RK3
RK0 PHYSU=RK3
OPERATION COMPLETE.

COMMAND? SAVE
SYSTEM MASTER DEVICE & UNIT? DK0
(REDIRECTED TO DK3  )
LABEL:  "MUMPS SYS - 1/17/77"
```

The remainder of the dialogue would be identical to that shown in Section I.4.5.

Note that any time you specify unit DK0 or RK0, MBU will remind you that it has been redirected to unit DK3.

Later, if you wished to redirect RK0 references back to physical unit RK0, you could issue the command:

```
SET RK0 PHYSU=RK0
```

I.4.5.2 Restoring Globals Saved by MBU - Globals that were saved using MBU are not restored using MBU. You restore them while running MUMPS by calling the %LR Logical Restore utility. Before restoring, you log in under the UCI to which you wish to restore globals. Note that %LR restores globals only to the UCI you are currently logged in under, even if the globals were originally saved from different UCI. %LR will ask you simple questions to determine which magtape or DECtape unit you wish to restore globals from, and what the label on the tape should be. (When you are asked to type in the label, you may enclose it in quotes if you wish, but you are not required. If the label %LR finds on the tape is not the one you typed in, %LR will stop, telling you what label it did find on the tape. Call %LR again, and this time mount the correct tape or specify the correct label.)

%LR conducts a simple question-and-answer session to determine which globals, from which UCI's on the tape, you wish to restore to the UCI you are currently logged in under. You need not remember exactly what you saved on the tape, because %LR will tell you of each new UCI and global as it encounters it on the tape, and will allow you to select, via simple yes-and-no questions, which globals you wish to restore. %LR will also give you the option of restoring any global under a different name than it had when you saved it.

Just before restoring, %LR asks you if you wish to "kill" the present contents of the global before restoring. If you do not, then the effect is a "merge" -- all elements of the global that were saved on the tape are restored by %LR, but any other elements of that global array that were already defined in your MUMPS system before %LR was called, will remain defined after the restore is complete.

%LR will terminate automatically with an appropriate message when it encounters the end of the data saved on your tape. You may stop its execution any time you wish, however, by typing CONTROL-C.

I.4.6 The FORMAT And INIT Commands

The FORMAT command allows you to "format" a brand new disk volume, and then, optionally, to test it for bad blocks and/or initialize it. Initializing consists of writing MUMPS bit maps in the appropriate postions on the volume, to prepare the volume for use as a MUMPS disk.

The INIT command allows you to execute the second and third operations (test and initialize) without having to format the volume first. This command is used when the disk has already been formatted at some time in the past.

An example showing the questions you will be asked when formatting, testing or initializing a disk volume, is given in Section I.3.

Notes:

1. MBU can format the following types of disks: RP02, RP03, RP04, RP05, RP06 and RK05. Note that RK06 disks are delivered to customers already formatted.
2. At the termination of any FORMAT or INIT command, MBU will write a backup label named "*MBU*" on the volume, if there was no label on it to begin with; otherwise, MBU will rewrite the label that was there. Note that formatting, testing or initializing a volume causes any other information previously on the volume to be lost.
3. The INIT command can also be used to initialize a brand new magtape. In this case, initializing consists of simply writing a backup label named "*MBU*" on the tape. It is recommended that any brand new magtape (one that has never been written on) be initialized with the INIT command before being used with any other MBU commands.

I.4.7 The BOOT Command

The BOOT command is used to exit from MBU by bootstrapping a new system, such as MUMPS. Normally you will boot the new system from unit 0 of the desired device type. Following is an example showing how the BOOT command is issued:

```
COMMAND? BOOT
DEVICE & UNIT? DK0
BOOT DK0     ARE YOU SURE (Y OR N) ? Y
(the system residing on disk DK0 is read into memory, and control is
transferred to it)
```

I.4.8 The SET Command

The SET command is used:

1. to modify a device's VECTOR or CSR address,
2. to set a magtape unit's density to other than 800 bits per inch,

3. to set a magtape unit's blocking factor to other than 8, or

4. to declare what actual physical unit is to be associated with a given DEVICE & UNIT name.

Examples of valid SET commands are given in the "MBU Command Summary" reproduced in Section I.8. Further information can be found in the notes following the description of the SET command given there, and also in Sections I.4.5.1 and I.5.1.

I.4.9 The BB Command

The BB command is used to examine a disk's bad block table, or to add or delete entries in it. When specifying a block number to be added to or deleted from the table, you specify the absolute block number (not the MUMPS block number) of the block you wish to reference. The absolute block number of the first block on any disk is 0. If you need help computing absolute block numbers, refer to Section I.6.

When entering a block number, you may enter it as either a decimal or an octal number. You indicate a decimal number by following it with a period, like this: 102.

You indicate an octal number by preceding it with a number-sign, or pound-sign, like this: #146.

In the following example, the operator adds block number 146 octal to the bad block table on disk RJ1, and deletes block number 1025 (decimal):

```
COMMAND? BB
DEVICE & UNIT? RJ1
TYPE  H  FOR HELP
TYPE  EX  WHEN DONE, TO EXIT
ACTION? LIST

RJ1 BAD-BLOCK TABLE

#277 (191.)
#2001 (1025.)
#2020 (1040.)

ACTION? INSERT #146
ACTION? DELETE 1025.
ACTION? LIST

RJ1 BAD-BLOCK TABLE

#146 (102.)
#277 (191.)
#2020 (1040.)

ACTION? EXIT
OPERATION COMPLETE.
```

I.4.10 The MM Command

The MM command may be used to directly allocate (declare "in-use") or deallocate (declare "not-in-use") specific blocks on a MUMPS disk, or to allocate all known bad blocks on the disk. Allocating a block that was not previously allocated has the effect of making the block inaccessible to MUMPS, since MUMPS will assume that any allocated block is in use for some purpose. Deallocating blocks is something you will probably never have occasion to do.

Since the only use you will normally have for this command is to allocate known bad blocks -- so that MUMPS will not attempt to use them -- a special action, "ALLOC BB", is provided to allow you to perform this operation without having to know the block numbers of the bad blocks. The following example shows how you would allocate all the bad blocks on disk RJ1:

```
COMMAND? MM
DEVICE & UNIT? RJ1
TYPE  H  FOR HELP
TYPE  EX  WHEN DONE, TO EXIT
ACTION? ALLOC BB
* FINISHED *

ACTION? EXIT
OPERATION COMPLETE.
```

If for some reason you need to allocate or deallocate other blocks on the disk, type H for help where indicated, and you will receive a message showing you how to do this.

Note: Any time you initialize a disk as a MUMPS disk (which results in writing the initial MUMPS bit maps on the disk), then, when initializing is complete, MBU looks at the disk's bad block table. If there are any known bad blocks on the volume, MBU asks you:

```
ALLOCATE BAD BLOCKS ( Y OR N) ?
```

You should normally answer Y to this question.

I.5 SPECIAL DEVICES

I.5.1 Special RP06 Considerations

(You need only refer to this section if you have one or more RP06 drives in your system. Note that an RP05 drive is, to MUMPS or MBU, indistinguishable from an RP04 drive.)

A disk mounted on an RP06 drive is treated by MUMPS V4B as if it were, in fact, two RP04 disks. If, for example, your configuration consisted of an RP04 drive as physical drive 0, another RP04 drive as physical drive 1, and an RP06 drive as physical drive 2, then the following would hold true within your MUMPS system:

1. when you referenced RJ unit 0, you would be referencing the RP04 disk mounted in physical drive 0;

2. when you referenced RJ unit 1, you would be referencing the RP04 disk mounted in physical drive 1;

3. when you referenced RJ unit 2, you would be referencing the first, or "low" half of the RP06 disk mounted on physical drive 2;

4. when you referenced RJ unit 3, you would be referencing the second, or "high" half of the RP06 disk mounted on physical drive 2.

As a result of this division of one physical RP06 unit into two logical units, MBU must provide you with two separate ways to reference RP06 units, for sometimes you will want to reference the entire RP06 disk (for example, while formatting), and sometimes you will want to reference only a particular half at a time.

To be able to reference the halves individually, you must issue two "SET ... PHYSU= ..." commands to MBU -- to tell it the names by which you will be referencing the two halves. Given the configuration described above, you would issue the following two SET commands to MBU:

```
SET RJ2 PHYSU=RJ2/L
SET RJ3 PHYSU=RJ/H
```

This tells MBU that hereafter the DEVICE & UNIT name RJ2 is to be taken as a reference to the low half of physical RJ06 unit RJ2, and that the DEVICE & UNIT name RJ3 is to be taken as a reference to the high half of physical RP06 unit RJ2. It would be necessary to issue these commands, for example, if you were planning to save globals from your system, or if you wished to back up one half of your RP06 without backing up the other half.

When you wish to reference an entire RP06 unit, you specify the device name as RB (or DB), instead of RJ (or DJ). Here is how you might format a new RP06 disk on physical drive 2:

```
COMMAND? FORMAT
DEVICE & UNIT? RJ2
MUST USE DEVICE NAME DB OR RB WHEN REFERENCING ENTIRE RP06
DEVICE & UNIT? DB2
FORMAT DB2    ARE YOU SURE (Y OR N)? Y
** BEGIN FORMATTING
```

etc.

If you later wished to back up only the high half of this disk, you could do so as follows:

```
COMMAND? SET RJ3 PHYSU=RJ2/H
RJ3 PHYSU=RJ2/H
OPERATION COMPLETE.

COMMAND? BACKUP
MASTER UNIT TO BE BACKED UP? RJ3
(REDIRECTED TO RJ2/H)
LABEL:
```

etc.

Notes:

1. An RP06 disk has only one MBU label and only one bad block table.

2. You should use the RB (or DB) form of the device name whenever referencing an RP06 in any of the following commands: LABEL, FORMAT, INIT, MM, BB, BOOT, COPY, COMPARE.

I.5.2 Special Magtape Considerations

Whenever you attempt to put a label on a magtape, using the LABEL command, or use a magtape as the "BACKUP" unit in a BACKUP or SAVE-GLOBALS command, MBU will first check to see if there is a bootable copy of MBU on the magtape. If there is, MBU will not write over the bootable copy of MBU, but will, instead, skip over it before writing the label and/or any data to be backed up. When attempting to read the label or restore data from the magtape, MBU (and the %LR Logical Restore utility) will again skip over the bootable copy of MBU, before attempting to read the label or data.

This means that you can, if you wish, back up data to, and later restore data from, the same magtape from which you boot MBU. This is especially convenient for user who have only one magtape drive.

If you ever wish to erase the bootable copy of MBU from a magtape, you may do so by using the MBU "INIT" command.

I.5.2.1 Magtape Density - MBU allows densities of 800 BPI for TM11/TU10 (MT) magtapes, and 800 or 1600 BPI for TJU16/TU16 (MM) magtapes. Densities of 200 and 556 BPI are not supported.

I.6 COMPUTING BLOCK NUMBERS

MUMPS and MBU store information on disks in "disk blocks". A disk block holds 256 (decimal) computer words of information. There are three ways one can specify a particular disk block on a disk:

1. by its cylinder, track and sector numbers,

2. by its "absolute" block number: the block at cylinder 0, track 0, sector 0 is absolute block 0; the block at cylinder 0, track 1, sector 0 is absolute block #m, if m is the number of blocks per track on the disk; the block at cylinder 1, track 0, sector 0 is absolute block #m*n, if there are n tracks per cylinder on the disk; or

3. by its "MUMPS" block number: the MUMPS block number is always equal to the absolute block number plus some fixed constant which depends on the device type and MUMPS unit number.

By using Tables I-1 and I-2, and the formulas given in Sections I.6.1 and I.6.2, you should be able to easily convert one type of block number to another.

I.6.1 Computing Absolute Block Number From Cylinder/Track/Sector

If m is the number of disk blocks per track for a given disk type, and n is the number of tracks per cylinder, then the block at cylinder c, track t, sector s, has absolute block number A, given by:

$$A = ((c * n) +t) * m + s$$

You can find the values to use for n and m, in Table I-1.

Notes:

1. Any time MBU asks you to specify a block number, you should specify the absolute block number.

2. If the disk in question is an RS04, use this formula instead:

   $$A = t * m + (s/2)$$

   (The value of s should be even.) On an RS04, each disk block occupies two sectors, and there is only one cylinder.

I.6.2 Converting To Or From Mumps Block Number

If you know the absolute block number, A, for a given disk block on a given disk, then you can determine the "MUMPS" block number, M, by using the following formula:

$$M = u * 262144. + K + A$$

The constant K depends on the disk type, and is found in Table I-2; and u is the unit number (0 to 7) by which you would refer to this disk in your MUMPS system (MUMPS logical unit number).

If you already know the MUMPS block number, M, and you wish to compute the absolute block number, A, use this formula:

$$A = M - K - (u * 262144.)$$

Table I-1
Constants For Use With Formulas in Section I.6.1

Disk Type	Blks/Trk	Trks/Cyl	Tot # Clys	Tot # Cyls
	(m)	(n)	Accessible to MUMPS	Counting Spares
RK05 (RK)	12.	2.	200.	203.
RS04 (RS)	32.	64.	1.	1.
RP02 (RP)	10.	20.	200.	203.
RP03 (RP)	10.	20.	400.	406.
RK06 (RM)	22.	3.	408.	411.
RP04 (RJ)	22.	19.	408.	411.
RP05 (RJ)	identical to RP04			
RP06 (RJ or RB)	22.	19.	815.	815.

Table I-2
Constants For Use With Formulas in Section I.6.2

Disk Type	Constant "K" Decimal Value	(Octal Value)
RK05 (RK)	0.	(#0)
RS04 (RS)	2097152.	(#10000000)
RK06 (RM)	2097152.	(#10000000)
RP02 (RP)	4194304.	(#20000000)
RP03 (RP)	4194304.	(#20000000)
RP04 (RJ)	6291456.	(#30000000)
RP05 (RJ)	identical to RP04	
RP06 (RJ) low half:	6291456.	(#30000000)
RP06 (RJ) high half:	6120912.	(#27262732)

Table I-3
Multiples of 262144.

	Decimal Value	(Octal Value)
0 * 262144.	0.	(#0)
1 * 262144.	262144.	(#1000000)
2 * 262144.	524288.	(#2000000)
3 * 262144.	786432.	(#3000000)
4 * 262144.	1048576.	(#4000000)
5 * 262144.	1310720.	(#5000000)
6 * 262144.	1572864.	(#6000000)
7 * 262144.	1835008.	(#7000000)

Table I-4
MBU Device Statistics

Device Type	Tot # Blks Accessible to MUMPS	Tot # Blks Accessible to MBU
RK05 (RK)	4800. (#11300)	4872. (#11410)
RS04 (RS)	2048. (#4000)	2048. (#4000)
RK06 (RM)	26928. (#64460)	27104. (#64740)
RP02 (RP)	40000. (#116100)	40600. (#117230)
RP03 (RP)	80000. (#234100)	81200. (#236460)
RP04 (RJ)	170544. (#515060)	171798. (#517426)
RP05 (RJ)	identical to RP04	
RP06 (RJ) low:	170544. (#515060)	170544. (#515060)
RP06 (RJ) high:	170096. (#514160)	170126. (#514216)
RP06 (RB) total:	N/A	340670. (#1231276)
DECtape (DT)	578. (#1102)	578. (#1102)
TM11/TU10 Magtape (MT)	N/A	N/A
TJu16/TU16 Magtape(MM)	N/A	N/A

Note: On DECtape and magtape, the label block is the first block on the volume. On any disk except RS04, the label block address is exactly 5 less than the total number of blocks accessible to MBU. (An RS04 disk cannot have a label.)

Table I-5
MBU Default VECTOR and CSR Addresses

Device Name	VECTOR Address	CSR Address
RK	#220	#177404
RS	#204	#172040
RM	#210	#177440
RP	#254	#176714
RJ or RB	#254	#176700
DT	#214	#177340
MT	#224	#172522
MM	#224	#172440

I.7 HOW TO MAKE BOOTABLE COPIES OF MBU

Your MUMPS System Builder (MSB) disk contains a bootable copy of MBU. If you received your MUMPS V4B system via disk distribution (RK05 or RK06), then one of the disk cartridges you received in your kit is the MUMPS System Builder disk. If you received your MUMPS V4B system via magtape distribution, then you had to create a MUMPS System Builder disk according to the instructions in Chapter 2 of this manual.

You can boot MBU directly from your MSB disk by first booting the MSB disk and then typing:

```
    >@MBU
```

You can create a bootable copy of MBU on DECtape (DT), TM11/TU10 magtape (MT), or TJU16/TU16 magtape (MM) by first booting the MSB disk and then typing the command:

```
    >@MBUBLD
```

MBUBLD will type the following:

```
     * ENTER TAPE DEVICE (DT, MT, OR MM) [S]:
```

Enter the two character name, DT, MT or MM, for the kind of tape you are using. MBUBLD will ask you to mount your tape on drive 0 of the appropriate device type, and will proceed to place a bootable copy of MBU on this drive. You can make further copies of that tape, if you wish, using MBU itself.

I.8 SAMPLE MBU COMMAND SUMMARY

MBU COMMAND SUMMARY

IN RESPONSE TO "COMMAND? " TYPE ONE OF THE FOLLOWING:

LA[BEL] - TO EXAMINE THE MBU LABEL CURRENTLY ON A VOLUME, OR TO PLACE A NEW MBU LABEL ON IT.

BA[CKUP]- TO "BACK UP" A SINGLE DISK VOLUME. THE VOLUME TO BE BACKED UP NEED NOT BE MUMPS-STRUCTURED, BUT IT MUST CONTAIN A VALID MBU "MASTER" LABEL. THE BACKUP MEDIUM MAY BE ANY DISK, DECTAPE OR MAGTAPE.

RE[STORE]- TO RESTORE THE DATA BACKED UP IN A PREVIOUS "BACKUP" OPERATION, TO THE MASTER VOLUME FROM WHICH IT CAME.

FO[RMAT]- TO FORMAT A DISK VOLUME, AND THEN, IF DESIRED, TO EXECUTE "TEST" AND/OR "INIT" FUNCTIONS. CAUTION: FORMATTING DESTROYS ALL INFORMATION PREVIOUSLY ON VOLUME.

IN[IT] - TO "TEST" A VOLUME AND/OR "INITIALIZE" IT AS A MUMPS VOLUME. "TESTING" CONSISTS OF WRITING AND RE-READING ALL SECTORS OF THE VOLUME USING SEVERAL DIFFERENT BIT PATTERNS, INFORMING OPR OF ALL ERRORS, AND RECORDING ANY BLOCKS INVOLVED IN ERRORS IN THE VOLUME'S BAD-BLOCK TABLE. "INITIALIZING" CAUSES MUMPS BITMAPS TO BE WRITTEN IN THE APPROPRIATE SECTORS ON THE VOLUME, SO THAT THE VOLUME MAY LATER BE USED IN A MUMPS ENVIRONMENT. CAUTION: TESTING AND/OR INITIALIZING DESTROYS ALL INFORMATION PREVIOUSLY ON VOLUME.

SA[VE-GLOBALS]- TO SAVE ALL OR SELECTED GLOBALS FROM ANY MUMPS VOLUME OR VOLUME-SET, ON MAGTAPE OR DECTAPE, IN A FORMAT FROM WHICH THE DATA CAN LATER BE RESTORED USING THE MUMPS %LR (LOGICAL RESTORE) UTILITY.

BB - TO LIST AND/OR MODIFY A VOLUME'S BAD BLOCK TABLE.

MM - TO DIRECTLY ALLOCATE OR DEALLOCATE ONE OR MORE SPECIFIC BLOCKS ON A MUMPS-STRUCTURED VOLUME, OR TO ALLOCATE ALL BLOCKS CURRENTLY CONTAINED IN THE VOLUME'S BAD-BLOCK TABLE.

BO[OT] - TO BOOT FROM ANY DESIRED DISK, DECTAPE OR MAGTAPE. (THE SYSTEM BEING BOOTED NEED NOT BE MUMPS.)

COP[Y] - TO MAKE AN EXACT IMAGE COPY OF ANY VOLUME, SUCH THAT THE COPY MAY THEN BE USED IN PLACE OF THE ORIGINAL. NOTE: IF THE VOLUMES ARE DISKS AND EITHER VOLUME CONTAINS A NON-BLANK MBU LABEL, THE LAST 5 "SPARE" BLOCKS WILL NOT BE COPIED. THIS WILL PREVENT THE "TO" VOLUME'S LABEL BLOCK (AND BAD-BLOCK TABLE) BEING CHANGED.

COM[PARE]-TO COMPARE TWO VOLUMES OF THE SAME TYPE TO VERIFY THAT ONE IS IN FACT AN EXACT IMAGE COPY OF THE OTHER. APPROPRIATE MESSAGES WILL IDENTIFY ANY BLOCKS WHICH ARE NOT IDENTICAL. (IF THE VOLUMES ARE DISK AND EITHER VOLUME CONTAINS A NON-BLANK MBU LABEL, THE LAST 5 "SPARE" BLOCKS WILL NOT BE COMPARED.)

SE[T] - TO CHANGE ANY OF THE FOLLOWING SYSTEM INFORMATION:
ANY DEVICE'S VECTOR OR CSR ADDRESS,
ANY MAGTAPE UNIT'S DENSITY,
ANY MAGTAPE UNIT'S BLOCKING FACTOR, OR
THE ACTUAL PHYSICAL UNIT TO BE ASSOCIATED WITH A GIVEN DEVICE & UNIT NAME.

```
EXAMPLES OF VALID "SET" COMMANDS:
   SET RP0 VEC=#254      ;SET RP02/03 VECTOR ADR TO OCTAL 254
   SET DK0 CSR=#177404   ;SET RK CSR ADR TO OCTAL 177404
   SET MT1 BLK=12.       ;SET MT1 BLOCKING FACTOR TO 12 DECIMAL
   SET MM2 DEN=1600      ;SET MM UNIT 2'S DENSITY = 1600 BPI
   SET RK5 PHYSU=RK3     ;RE-DIRECT ANY REFERENCE TO UNIT "RK5"
                         ;TO ACTUAL PHYSICAL UNIT RK3.
   SET RJ6 PHYSU=RJ4/L   ;(USED WITH RP06 DRIVES ONLY) RE-DIRECT
                         ;ANY REFERENCE TO UNIT "RJ6" TO THE LOW
                         ;HALF OF PHYSICAL UNIT RJ4 (DB4).  (AN
                         ;RP06 IS TREATED BY MUMPS AS 2 SEPARATE
                         ;"RJ" UNITS.  MBU ALLOWS BACKING UP OF
                         ;EACH HALF SEPARATELY.)
```

NOTES:

A "MAGTAPE BLOCKING FACTOR" IS THE NUMBER OF 256. WORD (512. BYTE) BLOCKS THAT WILL BE CONCATENATED TO FORM ONE PHYSICAL BLOCK ON THE TAPE. NORMALLY, THIS NUMBER SHOULD NOT EXCEED ONE-HALF THE NUMBER OF BUFFERS AVAILABLE. THE DEFAULT MAGTAPE BLOCKING FACTOR IS 8.

IN ALL COMMANDS, OCTAL NUMBERS ARE SPECIFIED USING A PRECEDING NUMBER SIGN (#), DECIMAL NUMBERS WITH A FOLLOWING PERIOD (.). THE ONLY EXCEPTIONS ARE NUMBERS IN SET DENSITY COMMANDS, WHICH ARE ALWAYS ASSUMED DECIMAL, AND UNIT NUMBERS, WHICH ARE ALWAYS IN THE RANGE 0 TO 7.

SPECIFYING A DEVICE NAME & UNIT NUMBER:

WHEN YOU ARE ASKED TO SPECIFY A UNIT, YOU SHOULD SPECIFY BOTH THE DEVICE NAME AND THE UNIT NUMBER,
LIKE THIS: RK1

SPECIFYING MBU LABELS:

AN MBU LABEL CONSISTS OF UP TO 20 CHARACTERS, ENCLOSED IN QUOTES, LIKE THIS: "THE MBU LABEL"
LABELS ARE PLACED ON VOLUMES BY USE OF THE MBU LABEL COMMAND, PLACE A BACKUP LABEL ON A VOLUME IF YOU WILL BE USING THAT VOLUME TO SAVE GLOBALS, OR TO BACK UP OTHER VOLUMES. PLACE A MASTER LABEL ON IT IF IT IS, OR IS TO BECOME, A MUMPS VOLUME, OR IF IT CONTAINS IMPORTANT DATA AND YOU DO NOT WISH IT TO BE INADVERTENTLY USED AS A BACKUP. IF YOU EVER WISH TO SPECIFY NO LABEL, OR A BLANK LABEL, INDICATE IT LIKE THIS: ""

CHARACTER CONVENTIONS:

YOU MAY TYPE ESCAPE OR ALTMODE IN RESPONSE TO ANY QUESTION BEING ASKED, TO ABORT THE OPERATION THEN IN PROGRESS.

YOU MAY TYPE CONTROL-C WHILE ANY NORMAL DISK OR TAPE I/O IS IN PROGRESS (SUCH AS DURING A BACKUP, RESTORE, SAVE-GLOBALS, COPY, COMPARE OR INIT COMMAND), TO CHECK ON THE PROGRESS OF THE COMMAND, AND (OPTIONALLY) TO ABORT IT.

YOU MAY TYPE H IN RESPONSE TO ANY QUESTION. YOU WILL USUALLY RECEIVE A SAMPLE ANSWER TO THE QUESTION BEING ASKED, SHOWING THE FORMAT IN WHICH THE ANSWER IS EXPECTED.

APPENDIX J

PRESRV

J.1 INTRODUCTION

The Preserve program (PRESRV) is a complete stand-alone core-resident system; MUMPS must be shut down before starting PRESRV. Its principal use with MUMPS-11 is the physical formatting of discs to create copies of disk and tape volumes. This copy capability is the basis for MUMPS-11 physical backup operations. PRESRV can make copies in several formats, only one of which is needed for MUMPS-11 operations. The type of copy mode which should be used for MUMPS-11 is called image mode, which is used to copy all of the blocks from one medium to another. In effect, image copies are duplications.

The following sections describe how to use PRESRV.

Note that only the features necessary for making MUMPS-11 image copies are described in this manual. For a detailed description of all of PRESRV's capabilities, refer to Appendix A in the *RSX-11M System Generation Manual* (DEC-11-OMGIA-A-D).

J.2 PRESRV COMMANDS

PRESRV acts on user requests which are in either of the two following forms:

1. File transfer command string

2. Non-transfer commands

File transfer command strings are used to specify and initiate a file transfer, whereas non-transfer commands are used to modify the interrupt vector address for a device or task for information. Before a transfer involving an RP04, 05 or 06 and/or a TJU16 is executed, the user must modify the vector address for the RP04, 05, 06 and/or the TJU16.

J.2.1 File Transfer Command Strings

PRESRV file transfer command strings have the following form:

DFV$_1$:filename[/switch$_1$]...[/switch$_n$]=DEV$_2$:filename[/switch$_1$]...[/switch$_n$]

DEV(1) is the output device designator, DEV(2) is the input device designator, and each /switch indicates one of the legal PRESRV switch options. FILENAME is used only to identify files on magtape. When a tape is produced by PRESRV, the user must specify a label of up to six characters for the tape to be produced.

The switches are used to specify the transfer format and to control the transfer. An image mode transfer is assumed if the device designators are of the same type (e.g., both are RK05 cartridge discs). Otherwise, the "/M" switch must be specified.

Following are three examples of file transfer command strings and a description of their actions. The PRESRV program prompts the user with the phrase "PRE>", and the user's responses are underlined in the following examples.

```
PRE>MT1:ABC/BL:12.=DK1:/IM
```

An image copy of the RK05 cartridge disc on drive 1 is placed on the TU10 magtape drive 1 and is assigned the filename ABC. The file is transferred 12 blocks at a time.

```
PRE>DK1:=DK0:
```

An image copy of the drive 0 RK05 cartridge disc is placed on drive 1.

```
PRE>DK1:/FO/VE=DK0:
```

The drive 1 disc is physically formatted, an image copy of the drive 0 disc is placed on the drive 1 disc, and the accuracy of the transfer is verified.

Whenever a disc file is transferred to magtape, a copy of PRESRV is placed at the beginning of the tape just prior to the image copy of the disc file. Thus, to later transfer the file from the magtape back to the disc, PRESRV can be booted from the tape which contains the image file. Section J.4 contains an example of such a transfer.

Table J-1 contains a list of the device designators.

Following is a list of PRESRV copy switches which are of interest to MUMPS-11 users. These switches are further described in the following sections.

Copy Switch	Effect
/BL:nnn	modify blocking factor
/ER	abort on LUN error
/FO	physically format a disc
/IM	image copy
/PE	phase-encoded option (1600 bpi) of a TJU16
/RW	rewind at completion
/VE	verify the accuracy of the transfer

Table J-1
Device Designators

Device Designator	Device Medium
DF:	An RF11 disk controller with one RS11 disk unit (one platter).
DKn: (n = 0 to 7)	An RK11 disk controller with one to eight RK03 or RK05 drive units. DK: is equivalent to DK0:.
DPn: (n = 0 to 7)	An RP11-C disk controller with one to eight RP03 disk drive units. DP: is equivalent to DP0:.
DBn: (n = 0 to 7)	An RH11 disk controller with one to eight RP04 disk drive units. DB: is equivalent to DB0:.
DTn: (n = 0 to 7)	One TC11 control unit with up to four TU56 dual transports or eight DECtape drive units.
MTn:filename (n = 0 to 7)	One TM11 control unit with up to eight TU10 magtape transports. Each transport is designated by a unique unit number selectable between zero and seven. A filename (from one to six alphanumeric characters) must be specified for disk to magtape transfers but not for magtape to copy operations.
MMn:filename (n = 0 to 7)	One TJU16 control unit with up to eight TU16 magtape transports. A filename (from one to six alphanumeric characters) must be specified for disk to magtape transfers but not for magtape to magtape copy operations.

J.2.1.1 Blocking Switch (/BL:nnn) - Blocking can be imposed on either tape or disk-type devices. A block is a group of 256-word units which represents the amount of data that will be transferred by a single read or write issued by PRESRV.

Each device has a default blocking factor which is used by PRESRV if no blocking switch is specified, and if buffer space is available to perform the default blocking. The defaults, shown in Table J-2 are overridden in one of two ways:

1. If a copy is image mode, PRESRV will, for tapes, use a blocking factor which uses all available buffering space.

2. The /BL switch can be used on either input or output file specifiers to establish a blocking factor.

Table J-2
Default Block Sizes

Device	Default block size
DK	12
DF	8
DP	10
DT	2
DB	22
MT	1
MM	1

The switch is of the form:

```
/BL:nnn
```

where:

nnn is the number of 256(10)-word block in each record.

The /BL switch can apply to either input or output file specifiers.

Example:

```
MT1:ABQRS/BL:20.=DB0:/BL:11./IM
```

NOTE

1. For image copies, the blocking factor specified with a disk or *DECtape must be a submultiple of the number of blocks on the volume (e.g., 4800 for DK).

2. If buffer space permits, use a large blocking factor for DECtape to improve operational efficiency.

Blocking calculations used to match available buffering space to blocking specifications for a file specifier appear in Section J.

J.2.1.2 Abort on LUN Error (/ER) - If the /ER switch is attached to a file specifier, an unrecoverable I/O error will cause the termination of the current PRESRV operation. PRESRV prompts with

```
PRE>
```

After issuing an error message.

/ER applies to either input or output file specifiers.

Example:

```
DT1:A/ER=DK:
```

J.2.1.3 Physical Format Device Switch (/FO) - This switch directs that the output device be physically formatted before any transfer is made. The switch may be used to format RK03/05, RP02/RP03, and RP04 05 and 06 disks.

Example:

```
     PRE>DK1:/FO/VE=DK0:
```

J.2.1.4 Image Copy Switch (/IM) - Image mode copy produces an output format which is identical to the input format. If both the input and output devices are identical and neither has a switch, the transfer will automatically be made in image mode. However, if the devices are not identical, the Image Copy Switch should be used on the disk (or DECtape for *DECtape - magtape transfers) aide of the PRESRV command.

```
     PRE>MT0:ABC=DK1:/IM
```

J.2.1.5 Phase Encoded Tape Switch (/PE) - This switch indicates that magtape volumes to be written or read on a TU16 drive are to be treated in phase-encoded mode (1600 bpi). This switch may appear on either input or output file specifiers.

Example:

```
     MM0:/VE/PE=MM1:/PE
```

J.2.1.6 Rewind at Completion (/RW) - The /RW switch will cause PRESRV to rewind any physical tape that has completed the specified transfer. The /RW applies to either input or output file specifiers.

Example:

```
     MT1:FILEABC/RW=DP0:IM
```

J.2.1.7 Verify Switch (/VE) - The /VE switch will cause a second pass over the input and output media so that every record written is read back and compared with the corresponding record on the input device. A verification failure is declared if the comparison fails. The verify switch applies only to output file specifiers.

Example:

```
     DK0:/VE=DK1:
```

J.2.2 Non-transfer Commands

PRESRV non-transfer commands allow the user to ask for information or modify the interrupt vector address for a device. Three non-transfer commands are of interest to MUMPS users, and these three are described below.

J.2.2.1 Set Vector Address Command (/VEC) - The following is a list of the default interrupt vector addresses for each of the devices supported by PRESRV:

Device	Vectors
DB	320
DF	204
DK	220
DP	254
DT	214
MM	300
MT	224

If the default vector address does not correspond to the actual hardware configuration in use, the /VEC command can be used to override the default value.

This command is of the form:

```
     dev:/VEC=nnnnnn
```

where:

```
        dev:  = The device-unit whose vector address is to be changed.
     nnnnnn = The new octal vector address.
```

Examples:

```
     PRE>MM0:/VEC=224
     PRE>DB0:/VEC=254
```

J.2.2.2 Help Command (/HE) - The Help command, whose format is

```
     /HE or /HELP
```

summarizes PRESRV commands, switches, and defaults.

J.2.2.3 List Command (/LI) - Information about the file which is present on a magtape or DECtape can be obtained by observing the following sequence (for magtape).

```
RSX-11S V01 BL01
VOLUME PRESERVATION PROGRAM - X02
FOR HELP TYPE /HE
52. BLOCKS AVAILABLE FOR BUFFERING

PRE>MT0:/LI

LAB:ABC     FRM:DK0 VOL 1 BLK:10. RECSIZ:5120. 29-APR-74 00:00:01
PRE>
```

After printing the information about the file, a message

```
     TAPE LABEL READ ERROR
```

may be printed. That warning message can be ignored.

The date and time on the listing will be meaningless.

J.3 PRESRV OPERATOR GUIDE

J.3.1 Starting Procedures

J.3.1.1 Bootstrapping PRESRV -

1. Mount the PRESRV tape on the appropriate drive (DECtape or magtape)

2. Set the tape transport controls as follows:

   For DECtape

   a. Set the unit number to 0.

   b. Set the WRITE ENABLE/WRITE LOCK switch to WRITE LOCK.

   c. Set LOCAL/REMOTE switch to REMOTE.

   For magtape:

   a. Set the drive number to 0.

   b. Remove the write enable ring.

   c. Position tape at LOAD POINT.

   d. Set ON LINE/OFF LINE switch to ON LINE.

3. Load and start the program by activating the hardware read only memory (ROM) bootstrap via the SWITCH REGISTER switches as described below.

   a. Set the console ENABLE/HALT switch to HALT.

   b. Set the console SWITCH REGISTER switches to the appropriate hardware bootstrap address.

   c. Press the LOAD ADRS switch.

   d. Set the ENABLE/HALT switch to ENABLE.

   e. Press the START switch.

The tape spins and PRESRV is loaded into memory. When loading is complete, PRESRV is started and identifier itself at the console terminal:

```
RSX-11S VOLUME PRESERVATION PROGRAM - Vnn
FOR HELP TYPE /HE
nn. BLOCKS AVAILABLE FOR BUFFERING
```

Following this message PRESRV prompts with

```
PRE>
```

and is ready to accept commands. The PRESRV tape can be removed at this time inasmuch as the entire PRESRV program resides in core.

J.3.1.2 Computing Required Blocks for Buffering - In the display, nn is the number of 256(10)-word blocks available for data transfer and verification operations. The user must use nn to determine if a given PRESRV media copy has sufficient buffering space. The space required depends on blocking factors and use of the verification switch. Default block sizes for devices supported by PRESRV are given in Table J-1.

The example presented below illustrates the method for determining the blocks required for PRESRV copy operations. The rule is to establish the blocks required for:

Input
Output
Verification

Example 1:

```
PRE>MT0:ABC = DK0:/IM
```

In this example, the following number of blocks are required for buffering.

Input	12.	(DK default block size)
Output	1.	(MT default block size)
Verification	0.	(No verification specified)
Total	13.	

Example 2:

```
PRE>DK0:/VE = DK1:
```

Copy unit DK1 to unit DK0:. Each record written to DK0: will be read and compared to it in-memory image.

Input	12.	(DK default block size)
Output	12.	(DK default block size)
Verification	12.	(Re-read block from DK0:)
Total	36.	

Example 3:

```
PRE>MM0: = MM1:
```

Image copy MM1: to MM0:. The image copy mode will use all available memory buffer space in creating block on MM0. If any input record is too large the transfer will be aborted.

Example 4:

```
PRE>MM0:/VE = MM1:
```

Image copy MM1: to MM0:. Since verification is requested, one-half of available memory buffer space will be used to form blocks to be written to MM0:. If any input record is too large, the transfer will be aborted.

J.3.1.3 Booting Another System or Stand-alone Program - The operator should halt the processor and boot the other system.

J.3.2 Operator Messages

J.3.2.1 Action Messages - During the course of PRESRV's operation, it may be necessary to take certain actions. At such time, PRESRV will issue promoting messages to the user's terminal. These messages are as follows:

```
**BEGIN VERIFICATION**

MOUNT OUTPUT VOLUME     AND TYPE CR>

MOUNT INPUT VOLUME      AND TYPE CR>

OUTPUT VOLUME WRITE-LOCKED. TYPE CR WHEN READY>

VOLUME FAULTY - "R" TO RETRY, "C" TO CONTINUE>
```

The VOLUME FAULTY message is issued whenever a verification error or a write error occurs and the /ER switch has not been specified.

NOTE

> If PRESRV is ready to accept input, and the user types "ALTMODE", the current operation will be aborted.

J.3.2.2 Error Messages - ERROR IN READING COMMAND

There is some problem with terminal input or a -Z typed illegally.

Action: Retype command

EXPECTED EOF NOT FOUND

During verification of a tape image copy, an EOF mark was read from the input volume, but not from the output volume.

Action: Retry the operation

FATAL ERROR ON INPUT DEVICE

An error has occurred while reading the input volume during a tape image copy.

Action: Retry the operation

FATAL ERROR ON OUTPUT DEVICE

An error has occurred while reading or writing the output volume during a tape image copy.

Action: Retry operation

ILLEGAL BLOCK COUNT

The number of blocks specified with the /BL switch is too high for the available buffer space or it is 0.

Action: Retry with valid block count

INPUT DEVICE ERROR nnnnn nnnnn

An error has occurred in attempting to read data. The number provided is the octal logical block number when the error was detected.

Action: None

INPUT VOLUME nn. OUT OF SEQUENCE

The number of the input volume does not match the number of the required volume.

Action: Mount proper volume and proceed

INSUFFICIENT BUFFER SPACE

A record (or all records) on an input tape has been found to be larger than the available buffer space. Alternatively, transfer specification requires more blocks than are available.

Action: Abort (or retry in second case)

LUN ASSIGNMENT ERROR

A device and unit combination specified in a command string is either not represented in the current PRESRV system or is not a valid device for PRESRV operations.

OUTPUT DEVICE ERROR nnnnn nnnnn

An error has occurred reading or writing data. Retry, continue, or abort. The number provided is the octal logical block number when the error was detected.

Action:

SYNTAX ERROR

Input not recognizable

Action: Retype command

VERIFICATION ERROR nnnnn nnnnn

The data read back from the output volume does not match what was written. The number provided is the octal logical block number of the output device when the error was detected.

Action: Retry, continue or abort.

J.4 EXAMPLES

This section provides examples of how to provide backup of discs using PRESRV. The user's responses are underlined. If verification of the accuracy of the transfer is not desired, the /VE switch may be deleted in the following examples.

J.4.1 RK05 Disc Cartridge to Another RK05

PRESRV is first booted (see Section J.3.1.1 for boot instructions). The following sequence then occurs to transfer the contents of disk 0 to disc 1.

```
     PRE>DK1:/VE - DK0:
     PRE>
```

J.4.2 RK05 (RK11) to a TU10 (TM11) Magtape

PRESRV is first booted and then the following sequence occurs.

```
     PRE>MT0:ABC/BL:12./VE = DK0:/IM

     MOUNT OUTPUT VOLUME 1.  AND TYPE CR>

     PRE>
```

Note that the volume number (1) does not imply that the magtape is on drive 1. The volume number specifies the sequence of tapes should more than one tape be required to perform backup.

The tape contents can be restored to the disc by booting PRESRV from the tape which contains the tape image, and by then following the sequence below:

```
     PRE>DK0:/VE = MT0:ABC/BL:12.

     MOUNT INPUT VOLUME 1.   AND TYPE CR>

     PRE>
```

J.4.3 RP04, 05, 06 to Another RP04, 05, 06

PRESRV is first booted, and then the following sequence is observed to transfer the contents of disc 0 to disc 1.

```
     PRE>DB:/VEC=254

     PRE>DB1:/VE=DK0

     PRE>
```

J.4.4 RH11 to a TJU16 Magtape

PRESRV is booted, and the following sequence occurs.

```
PRE>DB:/VEC=254

PRE>MM:/VEC=224

PRE>MM0:ABC/BL:22./PE/VE=DB0/IM

MOUNT OUTPUT VOLUME 1. AND TYPE CR>
```

Several tapes are required when transferring large discs to magtapes. When PRESRV requests a new tape, that tape should be mounted (at load point) on the same unit as the former tape.

To restore the contents of a set of tapes created on a TJU16 to an RJP04, PRESRV should be booted from Volume 1, and then the following sequence should be observed.

```
PRE>DB:/VEC=254

PRE>MM:/VEC=224

PRE>DB0:/VE=MM0:ABC/BL:22./PE

MOUNT INPUT VOLUME 1. AND TYPE CR>
```

Several tapes are required when transferring magtapes to large discs. When PRESRV requests the next tape, that tape should be mounted (at load point) on the same unit as the Former Tape.

APPENDIX K

DSC

K.1 OVERVIEW AND INTRODUCTION TO DSC

The DSC (Disk Save and Compress) utility program copies the contents of Files-11 disks to tape or disk for backup and storage, decreases the amount of disk space required for data storage, and transfers data from bad blocks to accessible ones. DSC also provides a means of concatenating files and extensions into contiguous data blocks, as well as reducing the required number of retrieval pointers and file extensions.

DSC transfers and compresses data from a Files-11 formatted disk to either disk or tape by copying only blocks allocated to active files. Data files randomly scattered over the disk are accessed and written to a new medium without the intervening spaces. As a result, individual files are closer together, and the space available for new files is grouped together eliminating wasted space between files and improving access efficiency.

The contents of a disk can be written to another disk of the same size or can be transcribed to magnetic tape for permanent storage. If the contents of one disk are transferred to a disk with a larger storage capacity, the new disk is formatted using the attributes of the original disk. The increase in available storage space is thus negated.

DSC can also be used to recover from a hardware malfunction that renders a portion of the disk unreadable. The file, including what can be read from the bad block contents, is written to accessible blocks on a new disk where they may be restored.

K.2 INITIATING AND TERMINATING DSC

The DSC utility is included as part of the software package for RSX-11M, RSX-11D, and IAS operating systems. DSC may be initiated by entering either of the following commands:

```
MCR> DSC

MCR> DSC command string
```

To return control to the operating system after all DSC commands have been completed, type CTRL Z.

A stand-alone version of DSC may also be used under RSX-11M by bootstrapping from tape. The proper tapes to be used are those supplied with the operating system software package or, the first volume of a set of tapes which were created by DSC at a density of 800 bytes per inch may also be used.

K.3 DSC COMMAND FORMAT

DSC commands are entered in the format:

```
outdev:[filelabel][/switch] = indev:[filelabel][/switch]
```

where:

outdev: lists the physical device(s) to which data are transferred. The format of outdev: is AAnn: where AA are the ASCII charcters that specify the device mnemonic, and nn is the 1- or 2-digit (octal) unit number, and the colon is the device name terminator. When more than one device is listed, the device symbols must appear in succession, separated by commas, before any other command parameters are specified. (See Section 2.2.)

[filelabel] is the identifier assigned to the disk contents contained in a tape file on a tape volume or volume set created by DSC. A filelabel must be specified when the output device is magnetic tape, and is optional with other devices. (See Section 1.5.1.)

[/switch] is one or more optional switches described in Sections 1.5.2 through 1.5.5.2.

indev: lists the physical device(s) from which data are copied in the format AAnn: (described in outdev: above).

[filelabel] is the optional tape file identifier detailed in [filelabel] above and in Section 1.5.1.

[/switch] is one or more optional switches described in Sections 1.5.2 through 1.5.5.2.

Consider the following example of a DSC command string:

```
DSC> MM01:.MM2:SYSFILE/DENS=1600=DB1:
```

In this example, the contents of the RP04 disk pack DR1: are written to two TU16 tape drives MM01:, and MM2:. The resultant tape volume set recorded at a density of 1600 bytes per inch (bpi) contains all the data files from DB1: in the single tape file named "SYSFILE". Note that the tape cannot be used to bootstrap the stand-alone version of DSC.

K.4 DSC-SUPPORTED DEVICES

The following devices may be used with DSC:

Device Mnemonic	Device Type	Device Class
DM	RK11/RK06 cartridge disk pack	Block-structured
DB	RH11/RP04,05,06 and RH70/RP04,05,06 disk pack	Block-structured
DK	RK11/RK05 cartridge disk	Block-structured
DF	RF11/RS11 fixed head disk	Block-structured
DP	RP11/RP02/RP03 disk pack	Block-structured
DS	RH11/RS03/RS04 and RH70/RS03/RS04 fixed head disk	Block-structured
MM	RH11/TM02/TU16 and RH70/TM02/TU16 9-track magnetic tape	Tape
MT	TM11/TU10 7- or 9-track magnetic tape and TM11/TS03 9-track magnetic tape	Tape

K.5 DATA TRANSFER

DSC performs four distinct data transfers in its operation. Data are actually transferred from disk to disk, disk to tape, or tape to disk, but these operations consist of

1. copying data from a disk,
2. writing data to tape,
3. copying data from tape, and
4. writing the data back to disk.

Note that DSC cannot be used to transfer data from tape to tape.

K.6 OPTIONS

One option and five optional switches may be used with DSC commands. The formats and usages of these options are discussed in the following sections.

K.6.1 Filelabel

The filelabel identifies the data copied from a disk and stored on a set of tape volumes. A filelabel must be specified when the output medium for a DSC operation is magnetic tape.

The filelabel consists of from 1 to 12 alphanumeric characters of which the first nine are used by DSC to identify the tape file containing a disk's contents. The filelabel must follow the device specification and precede any switches. The filelabel is terminated by one of the following:

1. An option switch
2. An equal sign (indicating the end of the output side of the command)
3. A carriage return (indicating the end of the command string)

Note in this example:

```
DSC> MM01:,MM02:SYSFILE=DB1:
```

that a file is created on a tape labelled SYSFILE which contains all the data copied from the specified disk, DB1:. The filelabel may optionally be used when restoring data to disk. If the filelabel is entered as part of the input specification, the tape is searched for a file with the same name as the specified filelabel. That file of disk contents is transferred to disk when it is located. If no filelabel is specified, the first file of disk contents located is transferred.

In the following example:

```
DSC>DB1:=MM01:,MM-2:SYSFILE
```

the first tape volume specified, MM01:, is searched for a disk content file named SYSFILE. If the header for SYSFILE is located in the first volume, the data transfer is made. If the header is not found, an appropriate error message is issued, and the operation is terminated. If the command were entered without the filelabel, the first file of disk contents located on the tape on drive MM01:, which may or may not be SYSFILE, would be transcribed.

K.6.2 REWIND Switch (/RW)

The REWIND switch directs DSC to automatically rewind the associated magnetic tapes before beginning any read or write operation.

If entered as part of the input specification, the /RW switch causes the tape to be rewound before the copy operation begins. If a filelabel is specified with the /RW switch, DSC searches for the specified file of disk contents from the beginning of the first tape volume. If no filelabel is specified the first file encountered on the tape is copied.

When used as part of the output specification, the REWIND switch causes the copy operation to begin writing data at the beginning of the rewound tape. Thus, starting at the beginning of the tape, DSC overlaps whatever data were contained on the tape. If no REWIND switch is entered with the output specification, DSC begins writing to the tape following the first end of file block encountered.

NOTE

> The REWIND switch can only be used in conjunction with magnetic tapes. If used with any other device, an error message will be issued.

Note that in the following command:

```
     DSC> MM01:  SYSFILE/RW = DB1:
```

the tape of drive MM01: is rewound to its beginning before any data are written, and any data contained on the tape are overlaid. The data files originally contained in DB1: are written to a single tape file identified as SYSFILE.

In this example:

```
     DSC> DB1:=MM01:,MM02:SYSFILE/RW
```

A disk is restored using a tape created by a previous DSC command. The first tape volume is rewound and searched for a previously created DSC file labelled SYSFILE. If the file is found, it is transcribed. If it is not found, an error message is issued.

K.6.3 BAD BLOCK Switch /BAD

The /BAD switch is used in conjunction with output disks to allow manual entry of bad block information.

There are three variations of the /BAD switch allowing a choice between manually entered bad blocks supplementing the bad block file present on the disk, no bad block data at all, or manually entered bad blocks exclusively. The format for the /BAD switch and its options is:

```
     /BAD=MAN                  MAN
     /BAD=NOAUTO       or   /BAD=NOAUTO
     /BAD=MAN:NOAUTO   or        MAN:NOAUTO
```

where:

MAN   allows manual entry of bad blocks data which are appended to the bad block file present on the disk

NOAUTO   causes DSC to bypass all bad block information for the disk, resulting in no applicable bad block data

MAN:NOAUTO   uses only manually entered bad block data.

If either MAN or MAN:NOAUTO are entered as options, DSC prompts

```
     DSC>BAD=
```

and awaits the entry of bad block data by the user. Bad blocks are entered by responding

```
     DSC>BAD = n[,m]
```

where:

n   is the octal logical block number address of the first manually entered bad block on the disk

m   is the number of blocks that are to be included in the manually entered bad block file. (default of 1)

NOTE

> Several sets of bad blocks may be entered in this manner. DSC continues prompting
>
>   DSC>BAD=
>
> until a null line (a carriage return only) is entered.

Note in the following example:

```
     DSC>DB1:/BAD=MAN:NOAUTO=MM01:,MM02:/SYSFILE/RW

     DSC>BAD=00702,7
     DSC>BAD=00644,2
     DSC>BAD=04057
     DSC>BAD=
     DSC>
```

the output disk, DB1:, is restored from the tape file "SYSFILE" contained on MM01: and MM02:, using as bad block descriptors only those blocks specified by the user. In response to DSC's prompts for bad blocks following the /BAD = MAN:NOAUTO switch in the command string, only the following blocks which were entered in the example above will not have data allocated to them:

```
     00702
     00703
     00704
     00705
     00706
     00707
     00710
     00644
     00645
     04057
```

Compare the above example with the following example:

```
     DSC>DB1:/BAD=NOAUTO=DB0:
```

which transfers data to the lowest LBNs on device DB1:, regardless of the contents of the resident bad block file.

NOTE

> The bad block file contained on a disk used with any of the /BAD= switches is not overlaid or destroyed; it is either supplemented (/BAD=MAN) or ignored (/BAD=NOAUTO;/BAD=MAN:NOAUTO). The resident bad block file, if ignored during a DSC operation, cannot be accessed until another DSC operation is performed on the disk. Only at that time can the resident bad block file be restored.

K.6.4 APPEND Switch (/AP)

The /AP switch causes DSC to begin writing another file to a tape that contains an entire DSC-created disk content file. DSC searches the first specified output tape for two consecutive tape marks preceded by a DSC end-of-file (EOF) block. If the tape is the only volume that was used by DSC to previously record an entire disk content file, the command is accepted and the new file is begun at the end of the first file. If the tape is the second or a subsequent volume of a DSC created file, the following error message is returned.

```
OUTPUT TAPE AAnn:IS NOT THE ONLY REEL IN ITS SET
```

If the following command string is entered, accepted, and processed:

```
DSC> MM01:,MM02:SYSFILE/RW/AP=DB1:
```

the disk files on DB1: will be appended to the DSC-created file contained on the first magnetic tape volume specified, MM01:. In this operation, the first tape, MM0-1:, is rewound to its beginning, and is searched for the end of a previous DSC file contained on that tape. If an EOF is found for a DSC-created file and the whole file is contained on the single volume, the new file, SYSFILE, is appended to the tape. If necessary SYSFILE can extend to additional volumes.

The operation is terminated with an error message if one of the following conditions exists:

1. The file ending on the first specified volume was begun on another tape volume

2. The file began on the volume specified, but extended to other volumes

3. No file ended on the volume.

NOTE

The /AP switch may only be used with output tape devices. Specification of /AP in any other situation will result in an error message and termination of the command.

K.6.5 NON-MERGE Mode Switch (/NMG)

The NON-MERGE mode switch is entered as part of the output specification to override the default MERGE mode of copying data. In MERGE mode, DSC copies data from disk, an entire file, including extensions, at a time. In this mode, all sections of an individual file are copied in sequential order, and are then written to the output medium before the next file is accessed.

NOTE

> Only files written in MERGE mode may be accessed through the RMS RESTORE utility.

The NON-MERGE mode switch affects the order in which data are accessed. If the /NMG switch is specified in the command string, file sections are accessed and written in the order of their assigned file numbers. Each section of a file is accessed and written separately, as if it were a complete file. The retrieval pointers in the file headers and extension headers are updated, if necessary, so that the file sections are correctly mapped. Thus, the /NMG switch does not destroy the linkage between file sections.

While /NMG is specified as part of the output description when copying data from a disk, it is not necessary to specify the switch when writing from tape to disk.

Note in this example which uses the /NMG switch:

```
     DSC>MM01:,MM02:SYSFILE/RW/NMG=DS1:
```

That the contents of the disk DB1: are written in NON-MERGE mode to the beginning of the tapes in the output tape set MM01:, and MM02:. While data compression will occur on each file section when a complete disk is restored from the tape set, extensions of a file will not necessarily be in contiguous data blocks.

NOTE

> The resulting tape cannot be accessed through the RESTORE utility.

In this example of a MERGE mode operation:

```
     DSC>DB2:=DB1:
```

all of the files on DB1: are transcribed to DB2:. The files and file extensions are written sequentially and result in contiguous data blocks for each individual file in the lower numbered blocks on the disk. Unused blocks have higher block numbers.

K.6.6 Density Switches

Two option switches are provided for TU16 tape drives to override the default storage density of 800 bytes per inch (bpi). They are discussed in the following two sections. Tape drives other than TU16s may be used with DSC, but only TU16 drives may use the density switches. All tape drives operate at a default density of 800 bytes per inch.

K.6.6.1 1600-bpi Switch (/DENS = 1600) - The 1600-bpi switch directs the TU16 drive to operate as either input or output device at a density of 1600 bytes per inch. All volumes written or read by the device are accessed at this density. If a tape is written at 1600 bpi, it must be read at the same density.

Tapes recorded at a density of 1600 bpi cannot be used to bootstrap the system. A bootstrap block recorded in this density is inaccessible by the hardware. If the tape is to be used to bootstrap the system, specify the split density option or use the default density of 800 bpi.

The following example illustrates the usage of the 1600 bpi switch:

```
     DSC> MM01:,MM02:SYSFILE/RW/AP/DENS=1600=DB1:
```

The tapes created by this example, MM01: and MM02: are written at 1600 bytes per inch. The resulting tapes, even though they contain all the necessary information, cannot be used to bootstrap the stand-alone version of DSC. The same 1600 bpi switch must be specified when the tapes on MM01: and MM02: are used to restore the disk.

K.6.6.2 SPLIT Density Switch (/DENS=800:1600) - The SPLIT Density switch directs the TU16 drive to read or write the entire volume set, except for the first two blocks on each volume of the set, at 1600 bytes per inch. The first two blocks contain the volume header information, and the bootstrap block and DSC image; they are accessed at a density of 800 bpi. This split in the recording density allows DSC to access the bootstrap block and bring a stand-alone version of itself into memory, should it be required.

NOTE

> Although the first two blocks of each tape volume are recorded at 800 bpi when the SPLIT density switch is specified, only the first volume of the set can be used to bootstrap the system.

Consider the following example:

```
     DSC> MM01:,MM02:SYSFILE/RW/DENS=800:1600=DB1:
```

The tape created on drive MM01: in this example can be used to bootstrap DSC. The first two blocks of each volume are recorded at 800 bpi, but only the first volume may be used to bootstrap the DSC utility. The remainder of the first volume and of all subsequent volumes are recorded at 1600 bpi. If this tape were used to restore the disk contents to the disk medium, the SPLIT Density switch would have to be identified as part of the input specification.

NOTE

> The two density switches, DENS=1600 and DENS=800:1600, may only be used with TU16 tape drives. If specified with a disk device, the switches will halt the operation and cause an error message to be returned. If used with tape drives other than TU16s, the switches are ignored and do not affect the default density of 800 bpi. If either switch is specified when the tapes are created on a TU16 tape drive, the same switch must

be specified when the tapes are used as input. If both switches are specified in the same command, the SPLIT Density switch will be used and no error will be issued.

K.7 DSC OPERATION

The sections within this chapter detail the methods of DSC operation in each of the four stages of data transfer.

K.8 DATA TRANSFER FROM DISK

After the DSC command specifying a disk copy operation is entered and accepted, the DSC utility begins the data transfer by scanning the input disk to ensure that it is in Files-11 format. DSC begins transcribing data by copying an approximation of the disk index files. Because these files are updated to reflect the status and location of blocks on a new disk device, the Index File Bit Map, the Storage Bit Map File, and the Pad Block File will not be the same on the output disk as they were on the input disk. Only the data necessary for the reconstruction of these files is transcribed.

If the transfer is being made under MERGE mode, the file headers indicated as active in the Index File are accessed to locate the next active prime header of a data file. That header, the blocks it maps, and all extension headers and their related blocks included in the same file are then transferred to the output medium before the next file header is accessed from the Index File. This operation continues, each time writing a complete file, until all active files have been transferred.

In NON-MERGE mode each file header, including extension file headers, is treated as if it maps all the blocks included in the file. Thus, files and related file extensions are not consolidated and written at once as in MERGE mode, but are accessed and written individually as they are located.

During this operation, DSC examines only those blocks allocated to active files. Unallocated and bad blocks listed in the Bad Block File that were formerly interspersed throughout the input disk are ignored and bypassed by DSC's operation, resulting in compressed, contiguous data blocks on the output disk. When DSC restores the data to a disk medium, these blocks occupy the lowest Logical Block Numbers, bypassing any blocks listed in the Bad Block File, and the storage space available for new data is grouped in a continuous area of the highest LBNs.

If, during the copy operation, DSC accesses a data file that contains a section written on a bad block, the data on the bad section, however garbled, are written as they are read. When the file is restored to disk, the blocks will contain the same garbled data, but will be accessible to allow restoration by the user. A message identifying the location of areas of this type will be displayed on the console.

To copy a disk, DSC performs the following three steps:

1. Verifies that the disk is online, is identified correctly and is in Files-11 format.

2. Transcribes disk Index Files.

3. Copies data files.

NOTE

> Only one disk may be specified as the input device in any one DSC operation.
>
> Output device may be tape or disk.

K.9 DATA TRANSFER TO TAPE

When a magnetic tape drive is specified as the output device in a DSC operation, the entire contents of the input disk are written through it onto tape. This data transfer usually involves more than one reel of tape, and may utilize more than one tape drive.

Tapes created by DSC serve as a backup of the disk contents, but cannot be used by themselves, except through the RESTORE utility as discussed in Section 1.5.4. DSC tapes can only be efficiently used by copying them back to a disk medium, which restores and reformats the disk. Although the tapes contain many individual data files that are copied from the input disk, they can be treated as though they contain a single file - the file of the disk's contents.

When the operation of copying the disk's contents to tape begins, DSC allows transcription to more than one tape. Therefore, the first data block written to tape is a header which contains the volume name (obtained from the filelabel), and the relative volume number. This header identifies the tape volume set and the place within that set represented by the volume, assuring that when the disk restore operation is begun, the tapes will be loaded in order, and as a set.

If the system disk itself requires maintenance, or if for some other reason DSC cannot be directly invoked, a bootstrappable stand-alone version of the utility is provided on all tape volume sets created by DSC. The second block of data on the first volume of the set is a bootstrap loader and DSC image.

Subsequent contents of the tapes include the data required to reconstruct disk directory files, maps and pointers, and the actual data files copied from the disk.

In order to initiate the tape creation process, the tape devices must be on line and correctly identified, and a filelabel must be specified. Multiple online tape drives may be specified in the following format:

```
AAnn(0):, AAnn(1):,...AAnn(7):filelabel = indev:
```

as in the following example:

```
MM00:,MM1:,MM04:,MM2:SYSFILE=DR1:
```

As shown, each device identifier is followed by a colon, and the devices are separated by commas. The last device is followed by the filelabel. Only one type of device, 7 or 9 track tape drives can be specified in a single DSC command. Up to eight drives may be specified per command, but an individual tape drive may be specified only once.

If the number of tape volumes required exceeds the number of tape drives available, the system allows replacement of tapes on the drives specified in round robin fashion:

```
MM1 MM2 MM3
```

until the data transfer is complete.

DSC performs the following four steps when magnetic tapes are created:

1. Verifies that multiple output devices are on line and are identified correctly.
2. Creates a volume header that identifies volume number and volume set.
3. Records a bootstrap loader and DSC image.
4. Transcribes data.

NOTE

> Input device must be a disk.
>
> Valid options are
>
> /RW
> /AP
> /DENS=1600
> /DENS=800:1600
> /NMG

K.10 DATA TRANSFER FROM TAPE

Tapes created by DSC can only be used by DSC to reconstruct a disk. When the tapes are mounted and the tape drives are correctly specified as the input devices, the tape contents are sequentially accessed and written to the output disk. The directory files are created and updated as the data are transferred.

Tape drives specified as input devices must be on line and be identified correctly. The tape volumes must be mounted so as to be accessed in the correct order. Options that affect the physical format of the tapes, specifically the Density switches, must be specified as they were when the tape was created.

DSC performs the following four operations when copying data from tape:

1. Verifies that the tape drives are on line and are identified correctly.
2. Accesses tape volumes in order.
3. Creates directory files.
4. Transfers data files.

NOTE

> One must use the same density switches that were used when the tape was created. If a filelabel is specified, only the disk contents in the tape file identified by that filelabel will be transferred. If no filelabel is specified, only the first disk content file encountered will be transferred. Valid options are:
>
> /RW
> /DENS=1600
> DENS=800:1600
> filelabel

K.11 DATA TRANSFER TO DISK

No DSC operation is complete until the data files are restored to disk medium. Tape backup files are inaccessible (except through the RMS RESTORE utility) until they are written to a disk. Data involved in a disk-to-disk transfer are only available after the newly formatted disk has transcribed all the input.

To receive input, a disk must be in Files-11 format, on line, and correctly identified. Only one disk can be specified as the output device in any single operation. The disk must be large enough to contain all the data involved in the transfer, and it must have an up-to-date bad block file to ensure accessibility of the data being written to the disk.

DSC requires that the bad block identification file be updated immediately preceding the operation. This serves to eliminate the possibility of data being written to inaccessible blocks. If desired, the bad block file may be supplemented or overridden by manual bad block entry through the /BAD switches.

The copy operation begins with an examination of the output disk to ensure that it is in Files-11 format, and that it can contain all the data being transferred. The number of blocks being transferred from the input device(s) is compared to the number of blocks available on the output disk. An appropriate error message will be generated if not enough blocks are available.

The Index and Storage bit map files are constructed on the disk from data supplied by the input medium.

At this point in the operation, the transcription of the data files begins. File headers are updated to reflect the new size and location of the files. This updating is required since blocks that were previously scattered are now copied to a contiguous set of blocks, beginning at the lowest Logical Block Numbers available on the disk. If the original disk was copied in MERGE mode, file headers and their contents, and associated file extension headers and the extensions they map are all written to a contiguous series of blocks. If the /NMG switch was specified in the original disk copy operation, all the data files are written to the lowest LBNs available, but each file section is written in the order it was accessed.

Compression of files in this manner is beneficial when a file header's retrieval pointers are almost used up. Since DSC reformats a disk so that large numbers of contiguous blocks are allocated to a single data file, the number of retrieval pointers required to map the location and length of the file contents can be significantly reduced. If the original data transfer is made in MERGE mode, it is possible to reduce the number and size of file extensions and extension headers.

When the DSC operation is concluded, the allocated blocks occupy the lowest LBNs available on the disk. Blocks that are available for use generally have higher LBNs, and are in a contiguous section as well.

It should be noted that DSC, when writing to a disk, begins transcribing data onto the lowest LBN possible. Data present on the disk in this area are overlaid by the new data. Therefore, DSC cannot be used to transfer several smaller disks' contents onto a single, larger disk. Each copy operation will wipe out whatever previously occupied the blocks being used.

DSC performs the following procedures when creating a disk:

1. Verifies that the disk has an up-to-date bad block register.
2. Verifies that the disk is in Files-11 format, is on line, and is identified correctly.
3. Verifies that the disk has enough available blocks to contain all the data involved in the transfer.
4. Creates index and directory files.
5. Transcribes data files.

NOTE

> Any data originally contained on the specified output disk is overlaid.
>
> The /NMG switch can be specified with an output disk, but may only be used when the input device is also a disk.
>
> The /BAD switches may be used to override or supplement bad block data or the output disk.

K.12 DSC ERROR MESSAGES

DSC prints out many messages on the console that provide information on errors, problems, or potential problems encountered in the DSC operations. Each message is prefixed with:

```
DSC--
```

to identify itself as a DSC error message. In most cases, the above prefix is received in conjunction with a message identification type:

```
DSC -- *WARNING* error messge text
```

or:

```
DSC -- *FATAL* error message text
```

followed by a diagnostic error message. In the first example above, where a *WARNING* is received, the DSC operation in effect at the time will continue after the message is received. If a *FATAL* message is issued, the current DSC operation is aborted and DSC prompts for a new command line.

In some cases a message will be received that is neither a *WARNING* nor a *FATAL* message. Usually these are instructions to DSC users who must comply with the instructions to continue the operation.

The messages themselves are listed below in alphabetical order. The text of the message is shown first, followed by the meaning of the message, and concluded with instructions on what should be done to alleviate the condition.

ERROR MESSAGE	MEANING	SUGGESTED ACTION
AAnn:IS WRITE LOCKED. INSERT WRITE RING AND HIT BETWEEN	Output tape on drive AAnn:cannot be written on until a write enable ring is inserted.	Insert ring on specified tape and list the carriage return key.
BAD DATA BLOCK ON AAnn: FILE ID, FILE #, FILE SEQ. VBN X	Data block specified on the reported device contains data that is erroneous.	Examine the data block, correct if possible.
CANNOT DETERMINE DENSITY OF TAPE ON AAnn:	The input tape on the specified tape drive cannot be used for the reason that it was recorded at an indeterminate density.	Make sure the tape is the one needed for DSC copy operation.
CONFLICTING DEVICE TYPES	Device types, such as disk and tape drive, are specified on the same side of the command string.	Check command string, and change so that only one device type is specified on a command string side.
DEVICE AAnn: IS NOT FILES-11	The specified device is not in Files-11 format.	Use another device that is in Files-11 format.
DEVICE AAnn: IS NOT IN SYSTEM	The specified device is not contained as one of the peripheral devices in system being used.	Check the device identifier and re-enter.
DIRECTIVE ERROR	Internal error.	Retry the operation.
DUPLICATE DEVICE NAME	The same device was specified more than once.	The re-enter command string with device specified only once.
EXPECTED EXTENSION HEADER NOT PRESENT ON AAnn: - FILE ID, FILE # FILE SEQ.#	Tape read error causes specified file to be lost.	Retry if preceded by I/O warning messages, else tape is bad.

ERROR MESSAGE	MEANING	SUGGESTED ACTION
FAILED TO ALLOCATE HOME BLOCK ON AAnn:	Due to too many bad blocks on the output, home block cannot be created.	Replace specified device.
FAILED TO FIND HOME BLOCK ON AAnn:	Read error from disk meaning that disk is either not Files-11 disk is bad, or home block is bad.	Retry; change disk drives, or change disk.
FAILED TO MAP OUTPUT FILE on AAnn:-FILE ID,#,SEQ.#	Inconsistency in writing specified file to disk. File is lost.	Retry operation.
FAILED TO READ FILE EXTENSION HEADER ON AAnn: File ID, #, SEQ.#	File extension header not found when reading from input disk.	Usually result of preceding I/O error, or else disk is inconsistent.
FILE HEADER OUT OF PHASE ON AAnn:AFTER FILE ID, FILE#, SEQ.#	Same as above.	Retry the operation.
FILELABEL IS TOO LONG	A filelabel of more than 12 characters has been specified.	Check filelabel and re-enter the command.

NOTE

The following input or output error messages are received in conjunction with DSC error messages specified as I/O errors. The DSC messages demonstrate the effect the I/O error has on the DSC operation. The following I/O messages point out the input or output condition that causes the error.

DEVICE NOT READY	Device is not mounted.	Mount the device.
PARITY ERROR ON DEVICE	Parity error has occurred.	Retry, try on new device drive.

ERROR MESSAGE	MEANING	SUGGESTED ACTION
DEVICE WRITE LOCKED	Disk drive is write locked.	Unwrite lock the drive.
DATA OVERRUN	Tape error-out of position or incorrect format.	Retry the operation.
PRIVILEGE VIOLATION	Device is mounted as Files-11.	RSX-11M users: Unmount the disk: IAS and other system users: dismount the disk, mount as foreign.
BAD BLOCK NUMBER	Block does not exist on disk; a DSC error has occurred; or the disk is bad.	Retry, changing disk and/or disk drive.
BAD BLOCK ON DEVICE	Device malfunction; or tape incompatibility.	Retry the operation.
FATAL HARDWARE ERROR	Hardware malfunction.	Retry; call DIGITAL Field Service.
END OF FILE DETECTED	Tape position lost.	Retry the operation.
END OF VOLUME DETECTED	Tape position lost.	Retry the operation.
DEVICE OFFLINE	Device not in the system.	Check command device specifications.
BLOCK CHECK	Parity error.	Retry on new drive.
INSUFFICIENT POOL SPACE	Operating system is overloaded.	Retry the operation.

ERROR MESSAGE	MEANING	SUGGESTED ACTION
HANDLER NOT RESIDENT	Device handler is not loaded.	Load the appropriate device handler.
UNKNOWN SYSTEM ERROR	Undefinable I/O error.	Retry the operation.
END OF TAPE DETECTED	Tape position lost.	Retry the operation.

NOTE

The above messages are received with DSC error messages specified as I/O errors.

I/O ERROR IN INPUT TAPE LABELS ON AAnn:	I/O error (which follows) occurred when reading, specified tape.	Retry on another tape drive.
I/O ERROR ON INPUT DEVICE AAnn:	Following I/O error may cause additional errors.	Warning only. No action necessary unless other errors occur.
I/O ERROR ON OUTPUT DEVICE AAnn:	Following I/O error message explains why device specified cannot be written to.	Correct error and retry.
I/O ERROR READING BOOT OR HOME BLOCK ON AAnn:	Problem reading disk, I/O error message follows.	Retry on new disk drive.
I/O ERROR READING FILE HEADER ON AAnn:AFTER FILE ID, #, SEQ.#	Following I/O error specifies error reading tape. Specified file is lost.	Retry entire operation.
I/O ERROR READING FILE HEADER ON AAnn:FILE ID, #, SEQ.#	When writing to disk, error in reading file header. File is lost. Following I/O error tells why.	Warning message only, entire operation should be retried.

ERROR MESSAGE	MEANING	SUGGESTED ACTION
I/O ERROR READING INDEX FILE BITMAP ON AAnn:	I/O error follows, occurred while reading disk.	Retry operation.
I/O ERROR READING INDEX FILE DATA FROM AAnn:	I/O error follows, occurred while reading disk.	Retry operation.
I/O ERROR READING SUMMARY DATA FROM AAnn:	I/O error occurred while reading tape to restore index and storage map files.	Retry another tape drive, or retry entire operation.
I/O ERROR WRITING FILE HEADER ON AAnn:FILE ID, #, SEQ.#	I/O error occurred while writing file header.	Retry operation.
I/O ERROR WRITING INDEX FILE BITMAP ON AAnn:	I/O error occurred while writing index file.	Retry operation.
I/O ERROR WRITING STORAGE MAP FILE ON AAnn:	I/O error occurred while writing storage file.	Retry operation.
ILLEGAL SWITCH	The command string contains a switch that cannot be used.	Re-enter with correct switch.
INDEX FILE ALLOCATION FAILURE on AAnn:	Error occurred because of too many bad blocks on output disk.	Retry with new disk.
INDEX FILE DATA NOT PRESENT on AAnn:	Error reading tape, or bad tape.	Retry on new drive, or retry the entire operation.
INPUT DATA OUT OF PHASE ON AAnn:FILE ID,#SEQ#,VBN, EXPECTED X FOUND Y	Tape position lost while reading specified file and location. Some data blocks may be lost.	Check extent of error, retry.
INPUT DEVICE AAnn:FILE ID, #, FILE HEADER CHECKSUM ERROR	Error reading specified file from disk due to bad header contents. File is lost.	Redo operation.

ERROR MESSAGE	MEANING	SUGGESTED ACTION
INPUT DEVICE AAnn:FILE ID,#, SEQ# FILE NUMBER CHECK	File is lost due to faulty file header read from disk.	Redo operation.
INPUT DEVICE AAnn:FILE ID,#, IS DELETED	File partially deleted or flagged is accessed, but not transcribed.	Warning only. No action required.
INPUT DEVICE AAnn:FILE ID, #, IS NOT PRESENT	Cannot find file while reading input disk.	Change disk or disk drive.
INPUT DEVICE AAnn:file id, #, SEGMENT NUMBER CHECK	File section linkage is broken. Specified file is lost.	Retry the operation.
INPUT DEVICE AAnn:FILE ID, # WILL BE SYNCHRONIZED	File section linkage is broken. Position lost while reading input tape. Some files will be lost. (Other error messages follow).	Retry the operation. Follow action indicated by subsequent error messages.
INPUT FILE OUT OF PHASE ON AAnn:FILE ID, #,SEQ #, EXPECTED X, FOUND Y.	Position lost while reading tape. Remainder of "X" and all of "Y" are lost.	Retry operation.
INPUT TAPE AAnn:IS A CONTINUATION TAPE	Tape out of sequence.	Re-enter command string, specify first volume first.
INPUT TAPE AAnn:IS NOT ANSI FORMAT	Input tape is in incorrect format.	Check and change input tape.
INPUT TAPE AAnn:RESYNCHRONIZED AT FILE ID, #, SEQ.#	Tape position lost, some data before specified file is lost.	Redo the operation.
INPUT TAPE ON AAnn:MUST BE AT BOT	Specified tape not at load point.	Remount, start at load point.

ERROR MESSAGE	MEANING	SUGGESTED ACTION
INVALID BAD BLOCK DATA ON AAnn:	Bad block file on output disk is invalid.	Run BAD utility on disk, or get new disk and retry the command.
MIXED TAPE DRIVE TYPES	A 7- and/or 9-track tape drive. were both specified in the same command string.	Specify only one. tape drive type.
MOUNT REEL X ON AAnn: AND HIT RETURN	Self explanatory.	Follow the directions displayed.
NO BAD BLOCK DATA ON AAnn:	No bad block file exists on the specified output disk.	Run BAD on disk, retry command.
NON-DATA BLOCK ENCOUNTERED ON AAnn:FILE ID, #, VBN#	Tape position lost while reading data. Specified file is lost.	Retry operation on new tape drive.
OUTPUT DEVICE AAnn:FULL (FILE ID, #)	Self explanatory.	Mount new device.
OUTPUT DEVICE AAnn:IS A DIAGNOSTIC PACK. DO NOT USE IT!	Self explanatory.	Get new device Re-enter command.
OUTPUT DISK AAnn:IS NOT BOOTABLE	Block number 0 is bad.	No action required, warning only.
OUTPUT FILE HEADER FULL ON AAnn:FILE ID, #	Inconsistent file header data due to too many bad blocks, File is lost.	Redo operation with new output disk.
OUTPUT FILE HEADER ON AAnn: NOT MAPPED-FILE ID, #	Consistency problem writing to disk. File is lost.	Redo operation, new disk may be needed.
OUTPUT TAPE AAnn:IS A CONTINUATION TAPE	Illegal append. Cannot append to other than first volume of a set.	Reenter command, change output tape.

ERROR MESSAGE	MEANING	SUGGESTED ACTION
OUTPUT TAPE AAnn:IS FULL	Cannot append to specified tape.	Re-enter command, change output tape.
OUTPUT TAPE AAnn:IS NOT A DSC TAPE	Illegal append.	Re-enter command, do not specify /AP switch.
OUTPUT TAPE AAnn:IS NOT ANSI FORMAT	Illegal append.	Re-enter command, do not specify /AP switch.
OUTPUT TAPE AAnn: IS NOT THE ONLY REEL IN ITS SET	Illegal append.	Re-enter command specify new output tape.
OUTPUT TAPE ON AAnn:IS NOT AT BOT	Tape not at load point.	Remount the tape or set at load point and re-enter the command.
SYNTAX ERROR	Error in command string format.	Check command, re-enter correctly.
TAPE FILE "FILENAME" NOT FOUND ON AAnn:	Specified filelabel not found.	Check filelabel, re-enter command.
TOO MANY DEVICES	More than eight devices have been specified on one side of the command.	Re-enter command string with a maximum of eight devices per side.
UNDEFINED ERROR	Error encountered, unable to identify the error.	Retry.

ERROR MESSAGE	MEANING	SUGGESTED ACTION
VOLUME SUMMARY DATA NOT PRESENT ON AAnn:	Incomplete data from specified input tape.	Retry the operation.
WRONG INPUT TAPE ON AAnn: EXPECTING FILELABEL, FOUND FILELABEL	Input tapes specified out of sequence.	Re-enter command with tapes correctly ordered.

INDEX

Allocation of,
  device buffers, 3-18
  directory, 3-7, 3-24
  disk blocks, 6-5 through 6-9, 6-17 through 6-19
Arithmetic conversion tables, E-1 through E-13
ASCII code, 2-16, B-1 through B-3
Associated documents, xi, xii, 1-4

Backslash (\), xiii
Backup and Utility system, 1-4, 4-8, I-1 through I-20
Backup copies, 4-8, 4-9
  system disk, 2-12, 4-11 through 4-13
Basic System Build, 2-1, 2-2, 2-10, 2-13
  answering questions, 2-9, 2-15 through 2-19
  loading System Builder, 2-6 through 2-8, 2-14
  object modules, 2-8
  overview, 1-2
  running SYSBLD, 2-14
Basic System configuration, 3-1
Baud rates, 3-7, 3-28, 3-31
BCS (Broadcast) System Utility program, 6-2, 6-3
Bell-type 103A datasets, 2-17
BREAK (Break key symbol), xiii
Buffer allocation for devices, 3-18
Building the Basic System, 2-1, 2-2, 2-6 through 2-8, 2-10, 2-13, 2-14

Carriage return, xv, 3-2
Character deletion, 2-3, 3-10
Character set, B-1 through B-3
Clock timing, 3-35
Closing system, 4-3
Command terminator, xiii
Connection procedures,
  non-standard terminal devices, H-1, H-2
Console terminal, 3-5, 4-1, 4-2
  function, 4-5
Controllers, 3-5 through 3-7
Conversion tables, E-1 through E-13

CPU device, 2-18, 3-7, 3-19, 3-28
Crash condition, 5-6 through 5-9
Creation date of system, 3-35
Creation time of system, 3-35
CTK (Caretaker) System Utility program, 5-5, 6-2, 6-4, 6-10, F-1
CTRL (command symbol), xiii
CTRL-U or U, xiii, 2-3

Datasets, Bell-type 103A, 2-17, 3-27
Date of system creation, 3-35
DAT (Date Set) System Utility program, 6-2, 6-5
DBDGD error, 5-4
DBT (Disk Block Tally) System Utility program, 6-2, 6-5, 6-6
Deallocating disk blocks, 6-17, 6-18
Debugging, 4-5 through 4-7
Debugging aid message, C-4
DECtape Test (TP5), 6-28
Default values for terminal parameters, 3-27, 3-28
Deletion of character or line, 2-3, 3-10
Device assignment, 3-5 through 3-7
Device availability for I/O, 3-14 through 3-18
Device buffer allocation, 3-18 through 3-20
Device driver object modules, loading of, 2-6 through 2-8
Device driver program, 2-2, 2-3
Device interrupt vector locations, 3-8
Device parameters, terminal, 3-25 through 3-28
Devices, I/O, 2-2, 2-3, 3-5, 3-14
Devices, reserved, 2-2, 2-3
DH-11 Multiplexer - see Multiplexer
Directory allocation, 3-7, 3-24
Disk-block allocation/de-allocation, 4-3, 6-5 through 6-9, 6-17 through 6-19
Disk hardware failure, 5-2
Disk inclusion, SYSGEN, 3-22, 3-23

Disk memory, 3-4, 3-11
Disk pack as system device, 3-9
Disk packs, 1-4, 3-4,
  distribution, 2-12, 2-13
DKDER error, 5-2
DKHER error, 5-2
DKSER error, 5-2, 5-4
%D Library Utility program, 3-35
DMC-11 Controller, 3-19, 3-27
DMP (Disk Block Dump) System Utility program, 6-2, 6-6 through 6-10
Documentation included in MUMPS-11 software package, 1-4
Documents, associated, xi, xii, 1-4
Drivers, - see Device driver
Drives, number permitted, 3-11 through 3-13
DSKDG error, 5-4
Dump program (DMP), 6-2, 6-6 through 6-10
DZ-11 Multiplexer - see Multiplexer

EIS Extended Instruction Set, 2-16
Error checking, utility programs, 6-1 through 6-28
Error conditions, 2-3
  SYSGEN, 3-10, 3-11
Error detection and recovery, 5-1 through 5-10
Error messages, C-1 through C-4
  operating system, C-5, C-6
Errors,
  bus timeout, 5-8
  Garbage Collector, 5-3, 5-4
  hard crash, 5-6 through 5-10
  hardware, 5-2, 5-5 through 5-9
  illegal instruction, 5-8
  logical, 2-3
  memory parity, 5-8
  operating, 5-1 through 5-3
  programming, 5-1, 5-2
  segmentation, 5-8
  soft crash, 5-6 through 5-9
  software, 5-2, 5-8, 5-9
  SYSBLD, 2-3
  system table, 5-5, 5-6
  typing, 2-3
Expression Evaluation Test (TP3), 6-28

Fixed head disk, 3-4
Floating-Point Processor, 2-15

Garbage Collector errors, 5-3, 5-4
Garbage Collector routine, 4-1, 4-3
GARB0 error, 5-3
GARB1 error, 5-3
GARB2 error, 5-3
GARB3 error, 5-3
GARB4 error, 5-3, 5-4
Global Test (TP8), 6-28
Glossary of terms, A-1 through A-7
%GP (Global Place) System Utility program, 6-3, 6-25 through 6-27

Hard crash, 5-6 through 5-10
Hardware configuration, 2-2

Inactive jobs, 6-14
Initialization program (STU), 6-25
Interpreter disabled, 4-4
Interrupt vectors, 3-29, 5-7, 5-8
I/O devices, 2-1, 2-2, 3-5
  availability of, 3-14 through 3-18

Job number, 6-21
Jobs lost or inactive, 6-14
Job status, 6-21, 6-22

KTR (Caretaker Reporter) System Utility program, 5-5, 6-2, 6-10, 6-11, F-1

Language Functions Test (TP1), 6-28
Library Utility programs, 1-3, 1-4, 3-22, 3-23
Line deletion, 2-3, 3-10, 3-11
Line Printer/Terminal Test (TP4), 6-28
Line terminator, xiii
Loading,
  object modules, 2-8, 2-9
  system, 4-10 through 4-12
  System Builder, 2-6 through 2-8, 2-13, 2-14

Loading procedure,
  Magtape, 2-6 through 2-8, 3-8, 3-9, G-1
  SYSGEN, 3-8 through 3-10
Logical backup, 4-8
Logical errors, 2-3
Log-in, 2-6 through 2-8, 3-9, 3-10
Lost jobs, 6-14

Magnetic Tape Test (TP6), 6-28
Magtape loading and unloading, 2-6 through 2-8, 3-8, 3-9, G-1, G-2
Magtape transport controls, 3-8
Materials required for system, 1-4
Mathematical conversion tables, E-1 through E-13
MBU (MUMPS Backup and Utility System), 1-4, 4-8, I-1 through I-20
Memory availability, SYSGEN, 3-11, 3-12
Memory buffer space, 3-18 through 3-20
Memory size, 3-4
Merge, System, 2-13, 3-11, 3-12
Messages from operator, 6-2, 6-3
$MONEY ($M) function, 2-16
Mini-ODT, 4-1 through 4-7
Miscellaneous SYSGEN information, 3-33 through 3-35
Modules (binary files), 2-1
MSB (MUMPS-11 System Builder), 2-1, 2-2, 2-6 through 2-8, 2-10, 2-13, 2-14
MSP (Modify System Parameters) Utility program, 6-11, 6-12
Multiplexers, 2-17, 3-5, 3-6
  availability, 3-17, 3-18
  operating parameters, 3-28 through 3-30

NAKED error, 5-2
Nodes, F-1 through F-3
Non-standard terminal device connection procedures, H-1, H-2

Object modules, 2-1
  user created, 2-8, 2-9
Octal debugging program (mini-ODT), 4-5 through 4-7
Operating system errors, 5-1 through 5-4, C-1, C-5, C-6
Operation of SYSGEN, 3-3
Operations summary, 2-1, 2-2
Operator functions, 4-1 through 4-13
Optional modules, 2-1
Overflow between disk units, 3-34
Overflow, program and global disk, 3-7, 3-8

PAC (Program Access Code) modification, 3-23, 3-25
Parity generation,
  for multiplexer, 3-30
  for paper tape I/O, 3-28
Partitions, 3-4
  allocation, 3-4, 3-20 through 3-22
  data, 6-20
  granting, 4-3
  sizes, 3-4, 3-5, 3-18, 3-20 through 3-22, 3-25
PDP-11 documentation, xii
PDP-11 processor, 2-2
  hardware configuration, 2-2
  switch register, 4-1 through 4-4
Physical backup, 4-8, 4-9
PLDER error, 5-3
Power line frequency, 3-33
PRESERV, J-1 through J-12
Program initiation, 6-1
Programming errors, 5-1 through 5-4
  messages, C-1 through C-4

Questions, SYSBLD, 2-15 through 2-19
Questions, SYSGEN operation, 3-3
  buffer allocation, 3-18
  device availability, 3-14 through 3-18
  memory availability, 3-11, 3-12
  miscellaneous information, 3-33 through 3-35
  modify terminal, 3-25 through 3-28
  modify UCI and PAC, 3-23 through 3-25
  multiplexer, 3-28 through 3-30
  partition allocation, 3-4, 3-20 through 3-22

Read errors, 5-3, 5-4
Reserved devices, 2-2, 2-3

Restart of system, 4-10, 4-11
Restoring system disk from backup, 4-8, 4-9, 4-11 through 4-13
> (right caret or right angle bracket) prompting symbol, 3-9, 3-10
RKC (RK Disk Copy) System Utility program, 6-2, 6-12 through 6-14
RK distribution, 2-12, 2-13
ROM bootstrap-loader starting addresses, 2-3 through 2-5
RSJ (Restore Job) System Utility program, 6-2, 6-14
RST (Restore Terminal Devices) System Utility program, 6-2, 6-15, 6-16
RUBOUT key, xiii, 2-3, 3-10

Save crash block, 5-9
Saving system image, 2-12, 2-19, 3-35, 4-9, 4-10
Shutdown of system, 4-13, 6-23, 6-24
SDP (Sequential Disk Processor) System Utility program, 2-16, 6-3, 6-17 through 6-19
SIF (System Information) Utility program, 6-3, 6-20
Soft crash, 5-6 through 5-9
Software documentation package, 1-4
Software, 1-1 through 1-3, 2-1
Space (symbol used to indicate), xiii
SS (System Status) Utility program, 6-3, 6-21 through 6-23
SSD (System Shutdown) Utility program, 4-13, 6-3, 6-23, 6-24
Stack growth, 3-8
Stack limit register, 3-34
Stall counts for DIGITAL terminals, 3-27, 3-28
Starting addresses, ROM bootstrap loader, 2-3 through 2-5
Starting,
  Basic System, 2-6, 2-14
  SYSGEN, 3-8 through 3-10
  the system, 4-10
Status information, 6-21, 6-22
Stopping system operation, 4-3
Storage areas for directories, 3-7
STU (System Startup) Utility program, 6-3, 6-25
SWAP error, 5-3
Switch register usage, PDP-11, 4-1 through 4-4
Switch settings, 3-10, 4-10, 5-4
Symbol usage, xiii, D-1, D-2
SYSBLD (System Build) program, 2-1
  examples, 2-1, 2-2, 2-6 through 2-8, 2-10, 2-13, 2-14
  questions, 2-9, 2-15 through 2-19
SYSDG error, 5-3
SYSER error, 5-4
SYSGEN (System Generation), 3-1
  error conditions, 3-10, 3-11
  loading procedure, 3-8 through 3-10
  operation, 3-3, 3-11
  organization, 3-1
  start, 3-8 through 3-10
  tape distribution, 3-2
SYSGEN questions,
  answering, 3-3
  buffer allocation, 3-18
  device availability, 3-14 through 3-18
  memory availability, 3-11, 3-12
  miscellaneous information, 3-33, through 3-35
  modify terminal, 3-25 through 3-28
  modify UCI and PAC, 3-23 through 3-25
  multiplexer, 3-17, 3-18, 3-28 through 3-30
  partition allocation, 3-4, 3-20 through 3-22
↑SYS system global, F-1 through F-3
System crash errors, 5-6, 5-7
System Demo Package, 6-27, 6-28
System device, 3-4, 3-14 through 3-16
System Exerciser Utility (TP7), 6-28
System generation procedures, 3-1 through 3-36
System library transfer, 3-22, 3-23
System merge, 2-13, 3-11, 3-12
System parameters, changing, 3-36, 6-12
System stack, 3-8, 3-34
System table error statistics, 5-5, 5-6
System Test Package, 6-27, 6-28
System Timings Test (TP2), 6-28 - for other system components, please refer to the specific subject.
System Utility and Test Programs, 6-1 through 6-28

Tape identification, 1-2
Tape logical structure, SYSGEN, 3-2
Terminal device parameters, 3-25 through 3-28
Terminals,
  baud rates, 3-7, 3-28, 3-30
  connection procedures for non-standard devices, H-1
  stall counts for, 3-27, 3-28
  tied, 3-26
Terminal-type devices, 3-14 through 3-16
Termination message, C-1, C-4
Terminator symbol (↵), xiii
Terms, glossary of, A-1 through A-7
Test package, 6-27, 6-28
Test programs, 6-1, 6-27, 6-28
Tied terminals, 3-26
Time of system creation, 3-35
TIM (Time Set) System Utility program, 3-36, 6-3, 6-25
TM-11 Magnetic Tape Test (TP6), 6-28
TP1 through TP8 (system test package), 6-27, 6-28
Typing errors, 2-3

UCI (User Class Identifier), 6-1
  for tied terminal, 3-26
  modification, 3-23
Underlining, xiii
Unloading procedures, Magtape, G-2
Up arrow (↑) to indicate CTRL command, xiii
User Device Driver, 2-18
Utility programs, 1-2, 1-3,
  summary, 6-2, 6-3

VT05, 50, 52, 55 Terminals, 3-27
VIEW command, illegal use of, 5-2

Write check, 3-34, 3-35
Write errors, 5-3, 5-4

MUMPS-11
Operator's Guide
DEC-11-MMOPA-E-D

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. Problems with software should be reported on a Software Performance Report (SPR) form. If you require a written reply and are eligible to receive one under SPR service, submit your comments on an SPR form.

Did you find errors in this manual? If so, specify by page.

____
____
____
____
____
____

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

____
____
____
____
____
____

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

____
____
____
____
____
____

Please indicate the type of user/reader that you most nearly represent.

- □ Assembly language programmer
- □ Higher-level language programmer
- □ Occasional programmer (experienced)
- □ User with little programming experience
- □ Student programmer
- □ Non-programmer interested in computer concepts and capabilities

Name ________________________________ Date ____________________

Organization ____

Street ____

City ____________________ State ____________ Zip Code ____________
or
Country

Please cut along this line.

-- Fold Here --

-- Do Not Tear - Fold Here and Staple --

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Software Documentation
146 Main Street ML5-5/E39
Maynard, Massachusetts 01754

digital

digital equipment corporation

Printed in U.S.A.