RT-11
System Generation Manual
Order No. DEC-11-ORGMA-A-D, DN1
RT-11
System Generation Manual
Order No. DEC-11-ORGMA-A-D, DN1

July 1976

ABSTRACT

This document contains the instructions necessary to bootstrap, build, and customize your RT-11 System. This version of the document incorporates Update Notice No. 1.

digital equipment corporation • maynard, massachusetts
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.

Copyright © 1976 by Digital Equipment Corporation

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

<table>
<thead>
<tr>
<th>Preface</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Overview of Software Kits</td>
<td>vii</td>
</tr>
<tr>
<td>1.1 RT-11 Software Kit</td>
<td>1-1</td>
</tr>
<tr>
<td>1.2 RT-11 Fortran IV Software Kit</td>
<td>1-3</td>
</tr>
<tr>
<td>1.3 Basic/RT-11 Software Kit</td>
<td>1-4</td>
</tr>
<tr>
<td>2. System Build Instructions</td>
<td>2-1</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>2-1</td>
</tr>
<tr>
<td>2.2 Building and Starting from Decpack Disk</td>
<td>2-2</td>
</tr>
<tr>
<td>2.2.1 RT-11</td>
<td>2-2</td>
</tr>
<tr>
<td>2.2.2 Fortran IV</td>
<td>2-5</td>
</tr>
<tr>
<td>2.2.3 Basic/RT-11</td>
<td>2-9</td>
</tr>
<tr>
<td>2.3 Building and Starting from Dectape</td>
<td>2-10</td>
</tr>
<tr>
<td>2.3.1 RT-11</td>
<td>2-10</td>
</tr>
<tr>
<td>2.3.2 Fortran IV</td>
<td>2-15</td>
</tr>
<tr>
<td>2.3.3 Basic/RT-11</td>
<td>2-20</td>
</tr>
<tr>
<td>2.4 Building and Starting from Diskette</td>
<td>2-21</td>
</tr>
<tr>
<td>2.4.1 RT-11</td>
<td>2-21</td>
</tr>
<tr>
<td>2.4.2 Fortran IV</td>
<td>2-26</td>
</tr>
<tr>
<td>2.4.3 Basic/RT-11</td>
<td>2-31</td>
</tr>
<tr>
<td>2.5 Building and Starting from Magtape</td>
<td>2-32</td>
</tr>
<tr>
<td>2.5.1 RT-11</td>
<td>2-32</td>
</tr>
<tr>
<td>2.5.2 Fortran IV</td>
<td>2-37</td>
</tr>
<tr>
<td>2.5.3 Basic/RT-11</td>
<td>2-41</td>
</tr>
<tr>
<td>2.6 Building and Starting from Cassette</td>
<td>2-42</td>
</tr>
<tr>
<td>2.6.1 RT-11</td>
<td>2-42</td>
</tr>
<tr>
<td>2.6.2 Fortran IV</td>
<td>2-44</td>
</tr>
<tr>
<td>2.6.3 Basic/RT-11</td>
<td>2-48</td>
</tr>
<tr>
<td>2.7 Building and Starting from Paper Tape</td>
<td>2-49</td>
</tr>
<tr>
<td>2.7.1 RT-11</td>
<td>2-49</td>
</tr>
<tr>
<td>2.7.2 Fortran IV</td>
<td>2-57</td>
</tr>
<tr>
<td>2.7.3 Basic/RT-11</td>
<td>2-63</td>
</tr>
<tr>
<td>3. Demonstration Package</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1 Running the RT-11 Single-Job Monitor</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2 Running the RT-11 Foreground/Background Monitor</td>
<td>3-10</td>
</tr>
<tr>
<td>3.3 Running Fortran IV</td>
<td>3-15</td>
</tr>
<tr>
<td>3.4 Running Basic/RT-11</td>
<td>3-18</td>
</tr>
<tr>
<td>4. RT-11 System Customization</td>
<td>4-1</td>
</tr>
<tr>
<td>4.1 General Building Instructions</td>
<td>4-1</td>
</tr>
<tr>
<td>4.2 Building RT-11 Systems from Decpack, Dectape, and Diskette</td>
<td>4-3</td>
</tr>
</tbody>
</table>
4.3 BUILDING DISK SYSTEMS FROM MAGTAPE (MBUILD) 4-6
4.3.1 Building RT-11 Systems with MBUILD 4-7
4.3.2 Creating Bootable Magtapes with MINIT 4-9
4.4 BUILDING DISK SYSTEMS FROM CASSETTE (CBUILD) 4-9
4.5 BUILDING DISK SYSTEMS FROM PAPER TAPE (PT BUILD) 4-12
4.5.1 Directory Extension 4-14
4.6 CUSTOMIZATION FOR SPECIAL HARDWARE 4-16
4.6.1 High Baud Rate Serial Consol Devices 4-16
4.6.2 Magtape Parity, Density, Number of Tracks 4-17
4.6.2.1 TM11 (MT.SYS) 4-17
4.6.2.2 TJIU6 (MM.SYS) 4-18
4.6.3 Specifying the Number of RPl1 Platters 4-19
4.6.4 Specifying a 50-Cycle Clock Rate 4-20
4.6.5 Interfacing RJS03/4 Disks to RT-11 4-21
4.6.6 Interfacing RP03 Disks to RT-11 4-21
4.6.7 Interfacing Card Readers to RT-11 4-23
4.6.8 Changing the Location of VT11 Floating Vectors 4-24
4.6.9 Interfacing a Second Diskette Handler 4-25
4.6.10 Reducing the Size of Text Window Displayed 4-27
4.6.11 Using "{" and "}" on the LA36 4-28
4.6.12 Setting an Upper Limit on File Size 4-28
4.6.13 Running RT-11 in Less Memory Than That Available 4-29
4.6.14 Accessing Nonsystem Disks on Single-Disk Systems 4-30
4.6.15 Reassigning Device Names for RK1 and RP11 4-31
4.6.16 Interfacing a Foreground Terminal 4-31
4.6.17 Modifying the Line Count in MACRO and CREF 4-31
4.6.18 Changing the DUMP Default Output Device 4-32
4.6.19 Using UNLOAD When a Foreground Job is Running 4-33
4.7 OPTIMIZING THE SYSTEM DEVICE 4-33
4.8 SWITCHING BETWEEN SINGLE-JOB AND FOREGROUND/BACKGROUND MONITORS 4-34

CHAPTER 5 ASSEMBLY AND LINK INSTRUCTIONS 5-1
5.1 GENERAL INSTRUCTIONS 5-1
5.2 ASSEMBLING AND LINKING THE SYSTEM FILES 5-1
5.3 ASSEMBLING AND LINKING THE UTILITIES 5-5
5.3.1 EDIT 5-5
5.3.2 MACRO 5-5
5.3.3 EXPAND 5-6
5.3.4 ASEMBL 5-6
5.3.5 CREF 5-6
5.3.6 LINK 5-7
5.3.7 LIBR 5-7
5.3.8 PIP 5-8
5.3.9 FILEX 5-8
5.3.10 SRCREM 5-8
5.3.11 DUMP 5-9
5.3.12 PATCH 5-9
5.3.13 ODT 5-9
5.3.14 BATCH 5-9
5.4 COMPILING AND LINKING PATCHO 5-10
5.5 ASSEMBLING AND BUILDING THE VT11 DISPLAY HANDLER LIBRARY (VTLIB) 5-11
5.6 ASSEMBLING AND BUILDING THE SYSTEM SUBROUTINE LIBRARY (SYSLIB) 5-11
5.7 ASSEMBLING AND LINKING MBUILD 5-17
5.8 ASSEMBLING AND LINKING THE FORTRAN IV COMPILER 5-17
5.8.1 Compiler Assembly 5-18
5.8.2 Compiler Linking 5-19
5.9 ASSEMBLING COMMON FORTRAN OTS MODULES 5-20
CONTENTS (Cont.)

5.9.1 Subscribing Modules Assembly
5.9.1.1 V2NS OTS Assembly
5.9.1.2 V2S OTS Assembly
5.10 ASSEMBLING HARDWARE DEPENDENT FORTRAN OTS MODULES
5.10.1 Bare Machine OTS Assembly
5.10.2 EIS OTS Assembly
5.10.3 FIS OTS Assembly
5.10.4 EAE OTS Assembly
5.10.5 FPU OTS Assembly
5.10.6 OTS Library Component Building
5.11 ASSEMBLING BASIC/RT-11
5.11.1 Floating Point Math Package
5.12 LINKING BASIC/RT-11
5.12.1 Linking BASIC/RT-11 with User Functions

APPENDIX A

INSTRUCTIONS FOR LOADING SOFTWARE BOOTSTRAPS

A.1 RK11 DECPACK BOOTSTRAP LOADER
A.2 TC11 DECTAPE BOOTSTRAP LOADER
A.3 RX11/RX01 DISKETTE BOOTSTRAP LOADER
A.4 MAGTAPE BOOTSTRAP LOADERS
A.5 TALL CASSETTE BOOTSTRAP LOADERS
A.6 PC11 PAPER TAPE BOOTSTRAP LOADER

APPENDIX B

FORMATTING THE RK05 DISK

TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>RT-11 Monitors</td>
</tr>
<tr>
<td>4-1</td>
<td>MBUILD Switches</td>
</tr>
<tr>
<td>4-2</td>
<td>CBUILD Switches</td>
</tr>
<tr>
<td>5-1</td>
<td>FPMP Assembly Parameters</td>
</tr>
<tr>
<td>A-1</td>
<td>RK11 Bootstrap Loader</td>
</tr>
<tr>
<td>A-2</td>
<td>TC11 Bootstrap Loader</td>
</tr>
<tr>
<td>A-3</td>
<td>RX11 Bootstrap Loader</td>
</tr>
<tr>
<td>A-4</td>
<td>TJU16 Bootstrap Loader</td>
</tr>
<tr>
<td>A-5</td>
<td>TM11 Bootstrap Loader</td>
</tr>
<tr>
<td>A-6</td>
<td>TALL CBOOT Bootstrap Loader</td>
</tr>
<tr>
<td>A-7</td>
<td>TALL QCBOOT Bootstrap Loader</td>
</tr>
<tr>
<td>A-8</td>
<td>Paper Tape Bootstrap Loader</td>
</tr>
<tr>
<td>B-1</td>
<td>RK05 Disk Formatting Program</td>
</tr>
</tbody>
</table>

V
HOW TO USE THIS MANUAL

This document introduces the RT-11 V02C, RT-11 FORTRAN V01C, and BASIC/RT-11 V01B-01 software kits to users receiving them for the first time.


Once you understand the contents of these manuals, you may build your RT-11 system according to the instructions in Chapter 2 and exercise the demonstration packages in Chapter 3. Users with special hardware considerations, or those users building nonstandard RT-11 configurations, consult Chapter 4 before building the system.

CONVENTIONS, ABBREVIATIONS, AND STANDARDS

The following are the conventions, abbreviations, and standards that are used throughout this manual.

1. All numbers are listed in octal unless otherwise indicated.

2. The following abbreviations are used:

   ALT          ALTMODE or ESCAPE key
   CTRL         CONTROL key
   CR           RETURN key
   LF           LINE FEED key
3. <CR> or <LF> indicate that the RETURN or LINE FEED key should be typed at that place in the dialogue.

4. <ALT> indicates that the ALTMODE (or ESCAPE) key should be typed at that place in the dialogue.

5. Text enclosed in square brackets, [], is optional; when including such text, do not type the square brackets unless otherwise indicated.

6. CTRL x indicates that the CONTROL key should be pressed and held down while another key, "x", is also pressed.

7. <TAB> indicates that the horizontal tab should be typed.

8. On ASR33 and ASR35 Teletype(1) terminals, special characters that are produced by holding down one key and pressing another are:

   ^  SHIFT N
   \  SHIFT L
   [  SHIFT K
   ]  SHIFT M
   <TAB>  CTRL I

9. The sample terminal dialogue provided in this document contains version numbers where they would normally appear. The version numbers given include xx's in those fields that may vary from installation to installation. The exact contents of these fields are not of interest, as long as appropriate digits appear in the area indicated in this document. The same is true for FREE CORE messages printed by any of the system programs and for FREE BLOCKS messages included in device directories.

10. Wherever necessary, computer outputs are underlined to differentiate them from user inputs.

RELATED DOCUMENTS

See the RT-11 Documentation Directory (DEC-11-ORDDA-A-D) for information concerning related documents in the RT-11 library.

(l) Teletype is a registered trademark of the Teletype Corporation.
CHAPTER 1

OVERVIEW OF SOFTWARE KITS

1.1 RT-11 SOFTWARE KIT

The basic RT-11 Software Kit is available on six media: DECPack, DECTape, diskette, magtape, cassette, and paper tape. Each kit contains user documentation and the materials necessary to build a complete RT-11 system. The components of an RT-11 Software Kit are inventoried on checklists attached to the outside of the kit. It is recommended that the user verify the contents of the package against the checklist and report any discrepancies to DIGITAL's Software Distribution Center.

The RT-11 DECTape, DECPack, and diskette kits contain "ready-to-run" RT-11 DECTape or disk Single-Job systems; copies of the masters can be mounted and bootstrapped directly without modification.

The magtape kit contains an RT-11 magtape which is used to build an RT-11 disk system; a special program (MBUILD) is used to initialize the system disk, then to copy the requisite system files and programs over to the system disk.

The cassette kit contains RT-11 system cassettes, which are used to build an RT-11 disk system; a special program (CBUILD) is used to initialize the disk, then to copy the requisite system files and programs over to the disk.

The paper tape kit consists of a special program (PTBUILD) and all system components in the form of relocatable binary object modules. PTBUILD is run to initialize the disk with a temporary monitor, linker, and paper tape handler. These tools are then used to link a complete RT-11 V02C system onto the disk.

Detailed instructions for building the system from all six media are contained in Chapter 2 of this document; general system assembly, linking, and building instructions are contained in Chapter 5. Note that the RT-11 System Reference Manual should be read thoroughly before exercising the package in Chapters 2 and 3, and the procedures in Chapters 2 and 3 should be exercised before the system is put to general use.

The contents of the RT-11 DECTape, DECPack, diskette, magtape, and cassette kits can be divided into two logical groups: system files and system programs. The system files include the monitor files and the device handlers (all with .SYS extensions), the system macro files (SYSMAC.SML, SYSMAC.8K, and VTMAC.MAC), the display support handler (VTHDLR.OBJ), and the FORTRAN system routines (SYSP4.OBJ). The monitors included are Single-Job and Foreground/Background versions.
OVERVIEW OF SOFTWARE KITS

for RK11 disk, TC11 DECtape, RX11 disk, RP02 disk, RJS03/4 disk, and RF11 disk; their names are identified in Table 1-1.

<table>
<thead>
<tr>
<th>Monitor</th>
<th>DECpack</th>
<th>DECTape</th>
<th>Diskette</th>
<th>Magtape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Job, RK11</td>
<td>MONITR.SYS</td>
<td>RKMNSJ.SYS</td>
<td>RKMNSJ.SYS</td>
<td>RKMNSJ.SYS</td>
</tr>
<tr>
<td>Foreground/ Background, RK11</td>
<td>RKMNF.B.SYS</td>
<td>RKMNF.B.SYS</td>
<td>RKMNF.B.SYS</td>
<td>RKMNF.B.SYS</td>
</tr>
<tr>
<td>Single-Job, RF11</td>
<td>RFMNSJ.SYS</td>
<td>RFMNSJ.SYS</td>
<td>RFMNSJ.SYS</td>
<td>RFMNSJ.SYS</td>
</tr>
<tr>
<td>Foreground/ Background, RF11</td>
<td>RPMNF.B.SYS</td>
<td>RPMNF.B.SYS</td>
<td>RPMNF.B.SYS</td>
<td>RPMNF.B.SYS</td>
</tr>
<tr>
<td>Single-Job, TC11</td>
<td>DTMNSJ.SYS</td>
<td>MONITR.SYS</td>
<td>*</td>
<td>DTMNSJ.SYS</td>
</tr>
<tr>
<td>Foreground/ Background, TC11</td>
<td>DTMNF.B.SYS</td>
<td>DTMNF.B.SYS</td>
<td>*</td>
<td>DTMNF.B.SYS</td>
</tr>
<tr>
<td>Single-Job, RX11</td>
<td>DXMNSJ.SYS</td>
<td>DXMNSJ.SYS</td>
<td>MONITR.SYS</td>
<td>DXMNSJ.SYS</td>
</tr>
<tr>
<td>Foreground/ Background, RX11</td>
<td>DXMNF.B.SYS</td>
<td>DXMNF.B.SYS</td>
<td>DXMNF.B.SYS</td>
<td>DXMNF.B.SYS</td>
</tr>
<tr>
<td>Single-Job, RP02</td>
<td>DPMNSJ.SYS</td>
<td>DPMNSJ.SYS</td>
<td>DPMNSJ.SYS</td>
<td>DPMNSJ.SYS</td>
</tr>
<tr>
<td>Foreground/ Background, RP02</td>
<td>DPMNF.B.SYS</td>
<td>DPMNF.B.SYS</td>
<td>DPMNF.B.SYS</td>
<td>DPMNF.B.SYS</td>
</tr>
<tr>
<td>Single-Job, RJS03 RJS04</td>
<td>DSMNSJ.SYS</td>
<td>*</td>
<td>*</td>
<td>DSMNSJ.SYS</td>
</tr>
<tr>
<td>Foreground/ Background, RJS03 RJS04</td>
<td>DSMNF.B.SYS</td>
<td>*</td>
<td>*</td>
<td>DSMNF.B.SYS</td>
</tr>
</tbody>
</table>

NOTE

The monitors available on cassette are the RK11 Single-Job (MONITR.SYS) and the RK11 Foreground/Background (RKMNF.B.SYS). The monitors available on paper tape are the RF11 Single-Job Monitor, the RF11 Foreground/Background Monitor, the RK11 Single-Job Monitor, and the RK11 Foreground/Background Monitor. See Section 2.7.1 for the paper tape monitor file names.

* Not available
OVERVIEW OF SOFTWARE KITS

The actual running version of the monitor is always named MONITR.SYS; to change monitors, the user must rename the monitor in use to a name other than MONITR.SYS, name the desired monitor MONITR.SYS, rewrite the bootstrap, and reboot the system. Additional instructions for modification of system files and their use can be found in Chapter 4.

The device handler files are named dev.SYS where dev is the 2-character device name for the device in question.

SYSMAC.SML, SYSMAC.8K, and VTMAC.MAC are the system macro libraries and are used when system macros are called from user-written assembly language programs. VTHDLR.OBJ is a set of display support routines used with assembly language programs that use the display processor (if any). SYSF4.OBJ is a set of FORTRAN callable subroutines that allow direct access to the RT-11 monitor capabilities from an RT-11 FORTRAN program; this set can be built into a library called SYSLIB.

Those files with .SAV extensions are the system programs; they can be executed directly with appropriate monitor commands. In all cases, these are the latest released versions of all programs, and they obsolete any other versions that may be in use. See RT-11 System Release Notes, Table 1, for the version numbers of the components in the V02C kit.

Recipients of RT-11 V02C paper tape kits must build all the files named above using the build process described in Section 2.7 of this document.

Users of RT-11 Version 2 who upgrade to V02C should upgrade their BASIC systems to V01B and their FORTRAN systems to V01C to take advantage of the improved reliability. Users of RT-11 Version 1 who upgrade to V02C directly (without using Version 2 or 2B) must upgrade their FORTRAN system to V01C and their BASIC system to V01B; unmodified FORTRAN V01 and BASIC V01 systems will not operate correctly under RT-11 V02C.

In addition to the basic system kit described above, RT-11 Source Kits and microfiche Listing Kits are available to assist in system development or modification.

1.2 FORTRAN IV SOFTWARE KIT

The basic RT-11 FORTRAN Software Kit is available on six media: DECPACK, DECTAPE, diskette, magtape, cassette, and paper tape. Each kit contains user documentation and the materials necessary to build a complete RT-11 FORTRAN system. The components of each package are inventoried on checklists attached to the outside of the kit. It is recommended that the user verify the contents of the package against the checklist and report any discrepancies to DIGITAL's Software Distribution Center.

The RT-11 FORTRAN DECTAPE, DECPACK, diskette, magtape, and cassette kits each contain a "ready-to-run" FORTRAN compiler and the modules necessary to build a library for each supported arithmetic option. The paper tape kit includes object modules that are used to build a running FORTRAN compiler and the appropriate library. Included in all kits is the simple demonstration program, DEMO.FOR.
OVERVIEW OF SOFTWARE KITS

All the kits include the PDP-11 FORTRAN Language Reference Manual (DEC-11-LFLRA-C-D), the RT-11/RTSF/E FORTRAN IV User's Guide (DEC-11-LRRUA-A-D), and this document. Specific instructions for building and exercising an RT-11 FORTRAN system are contained in Chapters 2 and 3 of this document; more general linking and assembly instructions are contained in Chapter 5. Before using RT-11 FORTRAN, it is important that the user be familiar with RT-11 itself and with the enclosed FORTRAN documentation.

RT-11 FORTRAN V01C requires RT-11 V02C to operate; it will not operate correctly under RT-11 V01-15, V02, or V02B.

Previous users of FORTRAN will notice the absence of LINKV2, LIBR, and PATCHO from this kit. The versions of these programs that were included in the V01-11 FORTRAN kit have since been obsoleted by those released with RT-11 V02, V02B, and V02C, and are no longer necessary in the FORTRAN kit. RT-11 V02 users will notice that OTSV2 and OTSV2S are not included in this FORTRAN kit; they were obsoleted by the files V2S and V2NS in this kit and are thus no longer necessary.

Included in this kit is the Software Performance Summary and recent editions of the Digital Software News; both documents list known software problems and solutions (if any) for PDP-11 software. These documents should be inspected carefully. The user should make the recommended modifications to the software and documents immediately; failure to do so may cause difficulty that could have been avoided by early correction of known problems.

In addition to the system kits described above, RT-11 FORTRAN Source Kits and microfiche Listing Kits are available to assist in system development or modification.

In addition to the RT-11 FORTRAN kits, there is an RT-11 FORTRAN Extensions Kit which provides FORTRAN support for optional devices such as LPS-11 and VT-11. Documentation for the use and installation of these functions can be found in the FORTRAN/RT-11 Extensions Manual (DEC-11-LRTEA-C-D), which is provided in the extensions kit.

1.3 BASIC/RT-11 SOFTWARE KIT

BASIC/RT-11 is distributed on six media: DECpack, DECTape, diskette, magtape, cassette, and paper tape. Each kit contains the BASIC/RT-11 Language Reference Manual (DEC-11-LBAC0A-D-D) and all the materials necessary to build BASIC/RT-11. The components of each package are inventoried on checklists attached to the outside of the kit. It is recommended that the user verify the contents of the software package with the checklist and report any discrepancies to DIGITAL's Software Distribution Center.

The BASIC/RT-11 DECPACK, DECTAPE, DISKETTE, MAGTAPE, and CASSETTE kits all contain the following "ready-to-run" versions of BASIC:

- BASIC.SAV: Complete BASIC, nonoverlaid; includes string support.
- BAS8K.SAV: Smallest possible version of BASIC; overlaid, no string support.

1-4 July 1976
All these versions of BASIC are ready-to-run (under RT-11 V02C); once placed on the system device, they can be started as described in Chapter 1 of the BASIC/RT-11 Language Reference Manual. The nonoverlaid version (BASIC.SAV) is linked for maximum execution speed at the expense of the extra memory necessary to keep all components resident. The overlaid version (BAS8K.SAV) is linked for maximum memory efficiency at a slight expense to execution time in response to certain commands.

Files with .OBJ extensions are relocatable object modules that can be linked together to form BASIC. They are used when customizing BASIC or when adding assembly language functions (paper tape users will use these files to build a running version of BASIC on the disk). BASICR.OBJ, BASICE.OBJ, and BASICX.OBJ are object modules that have been assembled to include the alphanumeric string capability. BASNSR.OBJ, BASNSE.OBJ, and BASNSX.OBJ are the corresponding object modules assembled without the optional string support.

To build a BASIC with all features included, use the linking instructions in Section 5.12. To build a version of BASIC without string capability, substitute BASNSR for BASICR, BASNSE for BASICE, and BASNSX for BASICX.

FPMP.OBJ is the floating point package for BASIC. FPMP.EAE, FPMP.EIS, and FPMP.FPU are customized versions of that math package and support EAE, EIS, and FPU options, respectively. To build a version of BASIC that uses one of these options, use the corresponding version of FPMP wherever FPMP is called for in the build instructions.

Instructions for building BASIC from the distribution kit can be found in Chapter 2 of this document. Users of previous releases of BASIC or RT-11 should note that the use of BASIC V01B requires RT-11 V02B or V02C for proper operation.

Finally, in addition to the BASIC kit described above, BASIC/RT-11 Source Kits and Listing Kits are available to assist in system development or modification. These may be ordered from the Software Distribution Center.

In addition to the BASIC/RT-11 kits, there is a BASIC/RT-11 Extensions Kit which provides BASIC/RT-11 support for optional devices such as LPS-11, VT11, and VT55. Documentation for the use and installation of this support can be found in the BASIC-11 Graphics Extensions User's Guide (DEC-11-LBGEA-A-D) and the BASIC-11 Lab Extensions User's Guide (DEC-11-LBEPA-A-D).
CHAPTER 2
SYSTEM BUILD INSTRUCTIONS

2.1 INTRODUCTION

This chapter contains the step-by-step information necessary to build an RT-11 system, including instructions for FORTRAN and BASIC users. The building procedures in this chapter assume familiarity with the RT-11 operating system; thus the user should have read the RT-11 System Reference Manual before beginning to build the system. In addition, FORTRAN users should be familiar with the contents of the PDP-11 FORTRAN Language Reference Manual and the RT-11/RSTS/E FORTRAN IV User's Guide. Users of BASIC/RT-11 should be familiar with the contents of the BASIC/RT-11 Language Reference Manual. There are six sections to this build procedure:

Section 2.2 Building and Starting from DECpack Disk (RK11)
Section 2.3 Building and Starting from DECTape (TC11)
Section 2.4 Building and Starting from Diskette (RX11/RX01)
Section 2.5 Building and Starting from Magtape (TJU16 or TM11)
Section 2.6 Building and Starting from Cassette (TA11)
Section 2.7 Building and Starting from Paper Tape (PC11)

After reading the general instructions that follow, proceed to the section that pertains to the medium upon which the system was distributed. Once the system has been built, proceed to Chapter 3 for a demonstration of the system.

Chapter 4 contains instructions for customizing RT-11 systems for specific hardware or unusual configurations (e.g., single-disk systems). Users with specific hardware considerations should refer to Chapter 4 after building their systems.

NOTE

No RT-11 system program ever HALTs with the expectation that the CONTINUE switch can be pressed to resume operation after corrective action has been taken. If the computer HALTs (the RUN light is off), a significant error has occurred and the entire section must be repeated from the beginning.
SYSTEM BUILD INSTRUCTIONS

If an error not explained in this manual occurs, please refer to the RT-11 System Message Manual.

If user errors occur within a section, go back to the beginning of that particular section.

User typing errors may be corrected using the standard RT-11 input editing techniques (RUBOUT and CTRL/U).

2.2 BUILDING AND STARTING FROM DECPACK DISK

2.2.1 RT-11

This section contains instructions for those who have received RT-11 on DECPack disk. The instructions describe building an RF11, RP02, or RJS03/4 disk system from RK disk or, for RK users, duplicating the master RT-11 disk, then bootstrapping the copy for use as the system disk. RK11 and RP02 users will need a blank, formatted disk for this procedure (see Appendix B for RK11 formatting procedures).

1. Set the ENABLE/HALT switch to HALT to stop any previous program that may be running. Set the ENABLE/HALT switch to ENABLE. Mount the RT-11 master disk on Unit 0, WRITE PROTECTED.

2. If the system has a hardware bootstrap capable of bootstrapping the RK11 disk, boot the disk and proceed to Step 3; otherwise, see Section A.1, then proceed with Step 3.

3. There will be a slight pause, after which RT-11 will respond with:

   RT-11SJ V02C-xx

   Type: DATE dd-mmm-yyyy<CR>
   Response: .

   where dd-mmm-yyyy is the current date in the form 27-NOV-75.

   If the master disk is being used to build an RF11 system,

   Type: ASSIGN RF1DK<CR>
   Response: .

   If the master disk is being used to build an RP02 system, mount a blank, formatted disk on Unit 0, WRITE ENABLEd.

   Type: ASSIGN RP1DK<CR>
   Response: .

   Type: ASSIGN RF1DK<CR>
   Response: .
SYSTEM BUILD INSTRUCTIONS

If the master disk is being used to build an RJS03/4 system,

Type: ASSIGN DS1DK<CR>
Response: ;

If the master is being used to build an RK11 system, mount a blank, formatted disk cartridge on Unit 1, WRITE ENABLEd.

Type: ASSIGN RK11DK<CR>
Response: ;

In any case,

Type: R PIP<CR>
Response: •

Scan the master disk for readability on this drive:

Type: RK1/K<CR>
Response: •

Each block on the disk will be read and checked for errors; the process takes about four minutes.

NOTE

If the response is anything but the above, see the RT-ll System Reference Manual, Section 4.2.12.

If the system being built is an RP02,

Type: DK1/Z/N137<CR>

If the system being built is an RK11 or a multiple platter RJS03/4 or RF11,

Type: DK1/Z/N112<CR>

Otherwise,

Type: DK1/Z<CR>

In any case,

Response: DK1/Z ARE YOU SURE?

Type: Y<CR>
Response: •

Type: DK1*,*,=RK11*,SYS/Y/X<CR>
Response: •

4. If the system being built is RK11, go to Step 5; otherwise:

Type: DK1RKMNSJ;SYS=DK1MONITR,SYS/Y/R<CR>
Response: •
SYSTEM BUILD INSTRUCTIONS

If the system being built is an RP11 system,

Type: \texttt{DKIMONIT\textasciitilde{}SYS\textasciitilde{}DKRFMNSJ\textasciitilde{}SYS/Y/R<CR>}
Response: *

Go to Step 5.

If the system being built is an RJS03/4 system,

Type: \texttt{DKIMONIT\textasciitilde{}SYS\textasciitilde{}DKDSMNSJ\textasciitilde{}SYS/Y/R<CR>}
Response: *

Go to Step 5.

If the system being built is an RP02 system,

Type: \texttt{DKIMONIT\textasciitilde{}SYS\textasciitilde{}DKDPMNSJ\textasciitilde{}SYS/Y/R<CR>}
Response: *

Go to Step 6.

5. Delete all unnecessary monitor and .SYS files from the system device as follows:

Type: \texttt{DK:xxMNSJ.SYS,xxMNF\textasciitilde{}SYS/Y/D<CR>}
Response: *

where xx is the designation of the device for any unnecessary monitors (as described in Table 1-1) as follows:

\begin{tabular}{ll}
DP & RP02 disk \\
DS & RJS03/4 disk \\
DT & DECTape \\
DX & RX11 diskette \\
RF & RP11 disk \\
RK & RK11 disk \\
\end{tabular}

Type: \texttt{DK:xx.SYS/Y/D<CR>}
Response: *

where xx is the designation of the device for any unnecessary device handler on the system as follows:

<table>
<thead>
<tr>
<th>Delete</th>
<th>If System has no</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA.SYS</td>
<td>more than 8K words of memory</td>
</tr>
<tr>
<td>CR.SYS</td>
<td>card reader</td>
</tr>
<tr>
<td>CT.SYS</td>
<td>cassette</td>
</tr>
<tr>
<td>DP.SYS</td>
<td>RP02 disk</td>
</tr>
<tr>
<td>DS.SYS</td>
<td>RJS03/4 disk</td>
</tr>
<tr>
<td>DX.SYS</td>
<td>diskette</td>
</tr>
<tr>
<td>LP.SYS</td>
<td>line printer</td>
</tr>
<tr>
<td>MM.SYS</td>
<td>TJU16 magtape</td>
</tr>
<tr>
<td>MT.SYS</td>
<td>TM11/TS03 or TM11/TU10 magtape</td>
</tr>
<tr>
<td>PP.SYS</td>
<td>paper tape punch</td>
</tr>
<tr>
<td>PR.SYS</td>
<td>paper tape reader</td>
</tr>
<tr>
<td>RF.SYS</td>
<td>RP11 disk</td>
</tr>
<tr>
<td>RK.SYS</td>
<td>RK11 disk</td>
</tr>
</tbody>
</table>

Type: \texttt{DK1/S<CR>}
Response: *
6. Type: \texttt{DK1\#\#RK1\#\#/X<CR>}
Response: \texttt{?NO\#SYS/\#BAD ACTION?} 

Type: \texttt{DK1/K<CR>}
Response: *

Each block on the disk being built will be checked for errors; the process takes about 4 minutes on an RK11, 1 minute on a single platter RF11 or RJS03/4, or about 30 minutes on an RP02.

\textbf{NOTE}

If the response is anything but the above, see the RT-11 System Reference Manual, Section 4.2.12.

Type: \texttt{DK1A=DKIMONITR, SYS/U<CR>}
Response: *

7. Remove the master disk from Unit 0 and store it in a safe place. If the new system is an RK system, remove the copy from Unit 1, label it RT-11 V02C SYSTEM DISK, then mount it in Unit 0, WRITE-ENABLEd.

If the system has a hardware bootstrap capable of bootstrapping the new system disk, boot the disk; otherwise, perform the bootstrapping procedures in the appropriate section of Appendix A.

RT-11 should bootstrap from the new system disk and type its identifying message,

\texttt{RT-11SJ V02C-xx}

If it does not, repeat this section.

8. Proceed to Section 2.2.2 if building a FORTRAN system from DECPack, Section 2.2.3 if building a BASIC system from DECPack, or consult the Table of Contents for the appropriate section if building FORTRAN or BASIC from another media. If neither FORTRAN nor BASIC will be used on the system, go to Chapter 3.

\section*{2.2.2 FORTRAN IV}

This section contains instructions for those who received RT-11 FORTRAN on DECPack disk. The instructions involve the transfer of the necessary FORTRAN-related files to the system disk and the creation of the FORTRAN library.

1. Bootstrap an RT-11 V02C system (if you have completed Section 2.2.1, the system is already booted and running).

To the running RT-11 monitor,
SYSTEM BUILD INSTRUCTIONS

Type: DATE dd-mmm-yy<CR>

where dd-mmm-yy is the current date in the form 27-NOV-75.

Response: .

Type: TIME hh:mm:ss<CR>

where hh:mm:ss is the current 24-hour time in hours:minutes:seconds.

Response: .

2. If an RK11 system is being built, mount the RT-ll FORTRAN IV system disk on Unit 1, WRITE-PROTECTED.

    Type: ASSIGN RK11DIS<CR>
    Response: .

    Go to Step 3.

If an RF11, RSJ03/4, or RP02 system is being built, mount the RT-ll FORTRAN IV system disk on Unit 0, WRITE-PROTECTED.

    Type: ASSIGN RK01DIS<CR>
    Response: .

3. Type: R PIP<CR>
    Response: *

    Type: *.*,DIS*.*,*/X<CR>
    Response: *

Dismount the RT-ll FORTRAN master disk and store it in a safe place.

    Type: CTRL C
    RESPONSE: *C

    Type: R LIBR<CR>
    Response: *

4. If the FORTRAN system will be running on a configuration that includes an EAE, proceed. Otherwise, go to Step 5.

    Type: FORLIB=UNI,OSCOM,V2NS,EAE/G<CR>
    Response: ENTRY POINT!

    Type: SERRTB<CR>
    SERRS<CR>
    <CR>

    Response: SEND2 ILL INS
    SEND2 ILL INS
    $F102 ILL INS
    $CLO2 ILL INS
    $F102 ILL INS
    $F102 ILL INS

2-6
System Build Instructions

Type: FORLIB,V2S=UNI,OTSCOM,V2S,EAE/G<CR>
Response: ENTRY POINT1

Type: SERRTB<CR>
SERRS<CR>
<CR>

Response: SEND2 ILL INS
SEND2 ILL INS
$F102 ILL INS
$CLO2 ILL INS
$F102 ILL INS
$F102 ILL INS

* 

Go to Step 9.

5. If the configuration includes a PDP-11/45 processor without FPU or a PDP-11/40 processor with EIS but without FIS, proceed. Otherwise, go to step 6.

Type: FORLIB=UNI,OTSCOM,V2NS,EIS/G<CR>
Response: ENTRY POINT1

Type: SERRTB<CR>
SERRS<CR>
<CR>

Response: SEND2 ILL INS
SEND2 ILL INS
$F102 ILL INS
$CLO2 ILL INS
$F102 ILL INS
$F102 ILL INS

* 

Type: FORLIB,V2S=UNI,OTSCOM,V2S,EIS/G<CR>
Response: ENTRY POINT1

Type: SERRTB<CR>
SERRS<CR>
<CR>

Response: SEND2 ILL INS
SEND2 ILL INS
$F102 ILL INS
$CLO2 ILL INS
$F102 ILL INS
$F102 ILL INS

* 

Go to Step 9.

6. If the configuration includes a PDP-11/40 or PDP-11/03 processor with FIS, proceed. Otherwise, go to Step 7.

Type: FORLIB=UNI,OTSCOM,V2NS,FIS/G<CR>
Response: ENTRY POINT1
SYSTEM BUILD INSTRUCTIONS

Type:

$ERRT8<CR>
$ERRS<CR>
<CR>

Response:

$SEND2 ILL INS
$SEND2 ILL INS
$SF102 ILL INS
$SCLO2 ILL INS
$SF102 ILL INS
$SF102 ILL INS

*

Type:

$ERRT8<CR>
$ERRS<CR>
<CR>

Response:

$SEND2 ILL INS
$SEND2 ILL INS
$SF102 ILL INS
$SCLO2 ILL INS
$SF102 ILL INS
$SF102 ILL INS

*

Go to Step 9.

7. If the configuration includes a PDP-11/45 processor with FPU, proceed. Otherwise, go to Step 8.

Type:

$ERRT8<CR>
$ERRS<CR>
<CR>

Response:

$SEND2 ILL INS
$SEND2 ILL INS
$SF102 ILL INS
$SCLO2 ILL INS
$SF102 ILL INS
$SF102 ILL INS

*

Type:

$ERRT8<CR>
$ERRS<CR>
<CR>

Response:

$SEND2 ILL INS
$SEND2 ILL INS
$SF102 ILL INS
$SCLO2 ILL INS
$SF102 ILL INS
$SF102 ILL INS


Go to Step 9.

8. If the configuration contains none of the above options:

   Type: FORLIB=UNI, OTSCOM, V2NS, NHD/G<CR>
   Response: ENTRY POINT1

   Type: SERRTB<CR>
   SERRS<CR>
   <CR>

   Response: SEND2 ILL INS
   SEND2 ILL INS
   SF102 ILL INS
   $CLO2 ILL INS
   SF102 ILL INS
   SF102 ILL INS

   *

   Type: FORLIB, V2S=UNI, OTSCOM, V2S, NHD/G<CR>
   Response: ENTRY POINT1

   Type: SERRTB<CR>
   SERRS<CR>
   <CR>

   Response: SEND2 ILL INS
   SEND2 ILL INS
   SF102 ILL INS
   $CLO2 ILL INS
   SF102 ILL INS
   SF102 ILL INS

   *

9. Type: CTRL C
   Response: *C

   Proceed to Section 2.2.3 if building a BASIC system from DECpack or consult the Table of Contents for the appropriate section if building BASIC from another media; otherwise, proceed to Chapter 3.

2.2.3 BASIC/RT-11

This section contains instructions for those who received BASIC/RT-11 on DECpack disk. The instructions involve placing a running version of BASIC on the system device.

1. Bootstrap an RT-11 V02C system (if you have completed Section 2.2.1 or 2.2.2, the system is already booted and running).
SYSTEM BUILD INSTRUCTIONS

To the running RT-11 monitor,

Type:     DATE dd-mmm-yy<CR>

where dd-mmm-yy is the current date in the form 27-NOV-75.

Response:  .

2. If an RK11 system is being built, mount the BASIC/RT-11 system disk on Unit 1, WRITE-PROTECTED.

Type:     ASSIGN RK11DIS<CR>
Response:  ;

If an RF11, RJS03/4, or RP02 system is being built, mount the BASIC/RT-11 system disk on RK11 Unit 0, WRITE-PROTECTED.

Type:     ASSIGN RK01DIS<CR>
Response:  ;

3. Type:     R PIP<CR>
Response:  *

Type:     9,*,DIS1:BASIC,SAV,BAS8K,SAV,DEMO,BAS/X<CR>
Response:  *

Dismount the BASIC/RT-11 master disk and store it in a safe place.

Type:     CTRL C
Response:  *C

Go to Chapter 3.

2.3 BUILDING AND STARTING FROM DECTAPE

2.3.1 RT-11

This section contains instructions for those who received RT-11 on DECTape. If DECTape is to be the system device, the user is instructed how to copy the master tape. If disk (RK11, RF11, or RP02) is to be the system device, the user is instructed how to make a disk system from the master DECTape. It is important (for the user's protection) to build the system as instructed, then to store the master in a safe place.

1. Set the ENABLE/HALT switch to HALT to stop any previous program that may be running. Set the ENABLE/HALT switch to ENABLE. Mount the RT-11 System DECTape (Tape 1 of 2, DEC-11-ORTSA-E-UC1) on Unit 0, WRITE LOCKed. Ensure that no other DECTape unit is set to 0.

2. If the system has a hardware bootstrap capable of bootstrapping the DECTape, boot the DECTape and proceed to Step 3; otherwise, see Section A.2, then proceed with Step 3.
3. The DECTape should move back and forth (for several seconds) and then the following message should be displayed on the terminal:

   RT-11SJ V02C-xx
   .

   If not, repeat this section from Step 2.

   Type: DATE dd-mmm-yy<CR>

   where dd-mmm-yy is the current date in the form 27-NOV-75.

   Response: .

   If DECTape will be the system device, proceed to Step 4. If disk will be the system device, proceed to Step 6.

4. Type: R PIP<CR>
   Response: *

   Mount a blank, formatted DECTape on an empty drive and set to Unit 1, WRITE ENABLEd.

   Type: D T11/Z<CR>
   Response: D T11/Z ARE YOU SURE?

   Type: Y<CR>
   Response: *

   Type: D T11*,*=D T01*,*/*X/Y<CR>

   (The master tape will be copied onto the blank tape. The process takes about two and one half minutes.)

   Response: *

   Type: D T11A=D T11MONITR,SYs/U<CR>

   (Both tapes will move as the system bootstrap is copied onto the tape on Unit 1. The process takes several seconds.)

   Response: *

   Type: D T11/K<CR>
   Response: *

   Each block on the DECTape being built will be checked for errors. This process takes about four and one half minutes.

   NOTE

   If the response is anything but the above, see the RT-11 System Reference Manual, Section 4.2.12.

Remove the master RT-11 System Tape (on Unit 0) and store in a safe place.

Remove the new DECTape from Unit 1 and label it as RT-11 SYSTEM TAPE 1.
5. Mount the RT-11 System DECTape (Tape 2 of 2, DEC-11-ORTSA-E-TC2) on Unit 0, WRITE LOCKed.

Mount a blank, formatted DECTape on an empty drive and set to Unit 1, WRITE ENABLEd.

Type: DT11/E<CR>
Response: DT11/E ARE YOU SURE?

TYPE: Y<CR>
Response: *

Type: DT11,*,*=-0101,*,*/#Y<CR>

The master tape will be copied onto the blank tape. The process takes about three minutes.

Response: *

Type: DT11/K<CR>
Response: *

Each block on the DECTape being built will be checked for errors. This process takes about four and one half minutes.

NOTE

If the response is anything but the above, see the RT-11 System Reference Manual, Section 4.2.12.

Remove the master RT-11 System Tape (on Unit 0) and store it in a safe place. Remove the new DECTape from Unit 1 and label it as RT-11 SYSTEM TAPE 2.

Mount the newly created RT-11 SYSTEM TAPE 1 on Unit 0, WRITE-ENABLEd. If the system has a hardware bootstrap capable of bootstrapping the DECTape, boot the DECTape; otherwise, perform the bootstrapping procedures in Section A.2.

The new system tape (now on Unit 0) will move as the system is bootstrapped from the newly made copy. RT-11 should respond by displaying the following on the terminal:

RT-11SJ 002C-xx
.

If it does not, repeat this section from Step 1.

Proceed to Section 2.3.2 if building a FORTRAN system from DECTape, Section 2.3.3 if building a BASIC system from DECTape, or consult the Table of Contents for the appropriate section if building from another media. If neither FORTRAN nor BASIC will be used on the system, go to Chapter 3.

6. Mount the RT-11 System DECTape, Tape 2 of 2 (DEC-11-ORTSA-E-UC2), on Unit 1, WRITE LOCKed.

If the disk in use is an RK11 DECPack or RP02 disk, mount a
blank, formatted disk on Unit 0, WRITE ENABLEd. (See Appendix B for RK11 formatting procedures.)

If the system is to be built on RK11 disk,
   Type:      ASS RKIDK<CR>
   Response:  

If the system is to be built on RF11 disk,
   Type:      ASS RFIDK<CR>
   Response:  

If the system is to be built on an RP02 disk,
   Type:      ASS DPIDK<CR>
   Response:  

In any case,
   Type:      R PIP<CR>
   Response:  

If the system device will be a single platter RF11 disk,
   Type:      DK1/Z<CR>

If the system device will be RP02 disk,
   Type:      DK1/Z/N137<CR>

Otherwise,
   Type:      DK1/Z/N12<CR>

In any case,
   Response:  DK1/Z ARE YOU SURE?
   Type:      Y<CR>
   Response:  
   Type:      DK1:;*=DT01;*:/*Y/X<CR>
   Response:  
   Type:      DK1:;*=DT11;*:/*Y/X<CR>
   Response:  
   Type:      DK1DYMSJ;SYS=DK1MONITR;SYS/Y/R<CR>
   Response:  

7. If the disk being built is an RK11 disk,
   Type:      DK1MONITR;SYS=DK1RKMSJ;SYS/Y/R<CR>
   Response:  
   Go to Step 8.

If the disk being built is an RF11 disk,
   Type:      DK1MONITR;SYS=DK1RFMSJ;SYS/Y/R<CR>
SYSTEM BUILD INSTRUCTIONS

Response:  *

Go to Step 8.

If the disk being built is an RP02 disk,

Type:  Dk1MONITR,SYS=DK1DPMNSJ;SYS/Y/R<CR>
Response:  *

Go to Step 9.

8. Delete all unnecessary monitor and .SYS files from the system device as follows:

Type:  Dk1xxMNSJ;SYS,xxMNFB;SYS/Y/D<CR>
Response:  *

where xx is the designation of the device for any unnecessary monitors (as described in Table 1-1) as follows:

<table>
<thead>
<tr>
<th>Device</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>RP02 disk</td>
</tr>
<tr>
<td>DS</td>
<td>RJS03/4 disk</td>
</tr>
<tr>
<td>DT</td>
<td>DECTape</td>
</tr>
<tr>
<td>DX</td>
<td>RX11 diskette</td>
</tr>
<tr>
<td>RP</td>
<td>RF11 disk</td>
</tr>
<tr>
<td>RK</td>
<td>RK11 disk</td>
</tr>
</tbody>
</table>

Type:  Dk1xx.SYS/Y/D<CR>
Response:  *

where xx is the designation of the device for any unnecessary device handler on the system as follows:

<table>
<thead>
<tr>
<th>Device</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA.SYS</td>
<td>more than 8K words of memory</td>
</tr>
<tr>
<td>CR.SYS</td>
<td>card reader</td>
</tr>
<tr>
<td>CT.SYS</td>
<td>cassette</td>
</tr>
<tr>
<td>DP.SYS</td>
<td>RP02 disk</td>
</tr>
<tr>
<td>DS.SYS</td>
<td>RJS03/4 disk</td>
</tr>
<tr>
<td>DX.SYS</td>
<td>diskette</td>
</tr>
<tr>
<td>LP.SYS</td>
<td>line printer</td>
</tr>
<tr>
<td>MM.SYS</td>
<td>TJU16 magtape</td>
</tr>
<tr>
<td>MT.SYS</td>
<td>TM11/TS03 or TM11/TU10 magtape</td>
</tr>
<tr>
<td>PP.SYS</td>
<td>paper tape punch</td>
</tr>
<tr>
<td>PR.SYS</td>
<td>paper tape reader</td>
</tr>
<tr>
<td>RF.SYS</td>
<td>RF11 disk</td>
</tr>
<tr>
<td>RK.SYS</td>
<td>RK11 disk</td>
</tr>
</tbody>
</table>

Type:  Dk1/S<CR>
Response:  *

Type:  Dk1/K<CR>
Response:  *

Each block on the disk being built will be checked for errors. This process takes about four minutes on an RK11 or one minute on a single platter RF11 or RJS03/4.
NOTE

If the response is anything but an asterisk, see the RT-ll System Reference Manual, Section 4.2.12.

9. Type: DKL1=DK1MONITR,SYS/U<CR>
    Response: *

Type: DKL1/0<CR>
    Response: RT=115J V020-xx

If it does not, repeat this entire section.

Dismount the master DECTapes from Unit 0 and Unit 1, and store in a safe place.

Proceed to Section 2.3.2 if building a FORTRAN system from DECTape, Section 2.3.3 if building a BASIC system from DECTape, or consult the Table of Contents for the appropriate section if building from another media. If neither FORTRAN nor BASIC will be used on the system, go to Chapter 3.

2.3.2 FORTRAN IV

This section contains instructions for those who received RT-ll FORTRAN on DECTape. The instructions involve the transfer of the necessary FORTRAN-related files to the system device and the creation of the FORTRAN library.

To build a FORTRAN system, there must be at least 370 free blocks on the system device. If the system device is DECTape, single platter RFll disk, or single platter RJS03/4 disk, there are not enough free blocks available; a new system device must be created before the FORTRAN files are copied.

If the system device is RKl1, RP02, multiple platter RFll, or multiple platter RJS03/4, go to Step 1.

Otherwise, if the system device is DECTape, create a new DECTape as described in Section 2.3.1, then delete files so that FORTRAN will fit. If the system device is single platter RFll or single platter RJS03/4, files must be deleted from the disk so that FORTRAN will fit.

The following files must be on the system device; all others can be deleted.

MONITR.SYS
TT.SYS (the console terminal handler)
LP.SYS (if the system has a line printer)
DT.SYS (if the system device is not DECTape)
EDIT.SAV
LIBR.SAV
LINK.SAV
PIP.SAV

The procedure for deleting files is as follows:
SYSTEM BUILD INSTRUCTIONS

Type:  \$R PIP<CR>
Response: *

To delete .SYS files,

Type:  SY:xxxxxx.SYS/Y/D<CR>
Response: *

where xxxxxxx is the name of the system file to be deleted. Repeat this command string for each file to be deleted.

To delete system programs,

Type:  SY:xxxxxx.xxx/D<CR>
Response: *

where xxxxxxx.xxx is the name of the system program to be deleted.

When all unnecessary files have been deleted,

Type:  SY1/S<CR>
Response: *

Type:  CTRL C
Response: ^C

If the system has a hardware bootstrap capable of bootstrapping the new system device, bootstrap the device; otherwise, perform the bootstrapping procedures in the appropriate section of Appendix A.

1. Bootstrap an RT-11 V02C system. Ensure that the system device is WRITE-ENABLEd.

Type:  DATE dd-mmm-yy<CR>

where dd-mmm-yy is the current date in the form 27-NOV-75.

Response: .

Mount the RT-11 FORTRAN master DECTape(DEC-11-LRF4A-C1-UC) on Unit 1, WRITE-LOCKed.

Type:  R PIP<CR>
Response: *

Type:  SY1#: 1STFORTRAN,SAV,FORTRAN,HLP,DEMO,FOR/X<CR>
Response: *

Type:  CTRL C
Response: ^C

Type:  R LIBR<CR>
Response: *

NOTE

In the following instructions, the square brackets [] are part of the command line and must be typed.

2-16            July 1976
2. If the configuration includes an EAE, proceed. Otherwise, go to Step 3.

Type: SYIFORLIB[140]#DT1#UNI,OTSCOM,V2S,EAE/G<CR>
Response: ENTRYP OINT1

Type: $ERRT<CR>
$ERRS<CR>
<CR>

Response: SEND2 ILL INS
SEND2 ILL INS
$P102 ILL INS
$S102 ILL INS
$S102 ILL INS
$S102 ILL INS

* 

Type: SYIFORLIB,V2S[140]#DT1#UNI,OTSCOM,V2S,EAE/G<CR>
Response: ENTRYP OINT1

Type: $ERRT<CR>
$ERRS<CR>
<CR>

Response: SEND2 ILL INS
SEND2 ILL INS
$P102 ILL INS
$S102 ILL INS
$S102 ILL INS
$S102 ILL INS

* 

Go to Step 7.

3. If the configuration includes a PDP-11/45 processor without FPU or a PDP-11/40 processor with EIS but without FIS, proceed. Otherwise, go to Step 4.

Type: SYIFORLIB[140]#DT1#UNI,OTSCOM,V2S,EIS/G<CR>
Response: ENTRYP OINT1

Type: $ERRT<CR>
$ERRS<CR>
<CR>

Response: SEND2 ILL INS
SEND2 ILL INS
$P102 ILL INS
$S102 ILL INS
$S102 ILL INS
$S102 ILL INS

* 

Type: SYIFORLIB,V2S[140]#DT1#UNI,OTSCOM,V2S,EIS/G<CR>
Response: ENTRYP OINT1

Type: $ERRT<CR>
$ERRS<CR>
<CR>
SYSTEM BUILD INSTRUCTIONS

Response: $END2 ILL INS
$END2 ILL INS
$F1O2 ILL INS
$CLO2 ILL INS
$F1O2 ILL INS
$F1O2 ILL INS

* 

Go to Step 7.

4. If the configuration includes a PDP-11/40 or PDP-11/03 processor with FIS, proceed. Otherwise, go to Step 5.

Type: SYIFORLIB[140]#DT1|UNI,CTSCOM,V2NS,FIS/G<CR>
Response: ENTRY POINT

Type: $ERRTB<CR>
$ERRS<CR>
<CR>

Response: $END2 ILL INS
$END2 ILL INS
$F1O2 ILL INS
$CLO2 ILL INS
$F1O2 ILL INS
$F1O2 ILL INS

* 

Type: SYIFORLIB,V2S[140]#DT1|UNI,CTSCOM,V2S,FIS/G<CR>
Response: ENTRY POINT

Type: $ERRTB<CR>
$ERRS<CR>
<CR>

Response: $END2 ILL INS
$END2 ILL INS
$F1O2 ILL INS
$CLO2 ILL INS
$F1O2 ILL INS
$F1O2 ILL INS

* 

Go to Step 7.

5. If the configuration includes a PDP-11/45 processor with FPU, proceed. Otherwise, go to Step 6.

Type: SYIFORLIB[140]#DT1|UNI,CTSCOM,V2NS,FPU/G<CR>
Response: ENTRY POINT

Type: $ERRTB<CR>
$ERRS<CR>
<CR>

Response: $END2 ILL INS
$END2 ILL INS
$F1O2 ILL INS
SYSTEM BUILD INSTRUCTIONS

$CLO2  ILL INS
$F102  ILL INS
$F102  ILL INS

* 

Type: SYIFORLIB,V2S[140]=DT1IUNI,OTSCOM,V2S,FPU/G<CR>
Response: ENTRY POINT!

Type: $ERRTB<CR>
$ERRS<CR>

Response: $SEND2  ILL INS
$SEND2  ILL INS
$F102  ILL INS
$CLO2  ILL INS
$F102  ILL INS
$F102  ILL INS

* 

Type: SYIFORLIB,V2S[140]=DT1IUNI,OTSCOM,V2S,NHD/G<CR>
Response: ENTRY POINT!

Type: $ERRTB<CR>
$ERRS<CR>

Response: $SEND2  ILL INS
$SEND2  ILL INS
$F102  ILL INS
$CLO2  ILL INS
$F102  ILL INS
$F102  ILL INS

* 

Type: SYIFORLIB,V2S[140]=DT1IUNI,OTSCOM,V2S,NHD/G<CR>
Response: ENTRY POINT!

Type: $ERRTB<CR>
$ERRS<CR>

Response: $SEND2  ILL INS
$SEND2  ILL INS
$F102  ILL INS
$CLO2  ILL INS
$F102  ILL INS
$F102  ILL INS

* 

Go to Step 7.

6. If the configuration contains none of the above options:

Type: SYIFORLIB[140]=DT1IUNI,OTSCOM,V2S,NHD/G<CR>
Response: ENTRY POINT!

Type: $ERRTB<CR>
$ERRS<CR>

Response: $SEND2  ILL INS
$SEND2  ILL INS
$F102  ILL INS
$CLO2  ILL INS
$F102  ILL INS
$F102  ILL INS

* 

Type: SYIFORLIB,V2S[140]=DT1IUNI,OTSCOM,V2S,NHD/G<CR>
Response: ENTRY POINT!

Type: $ERRTB<CR>
$ERRS<CR>

Response: $SEND2  ILL INS
$SEND2  ILL INS
$F102  ILL INS
$CLO2  ILL INS
$F102  ILL INS
$F102  ILL INS

* 

7. Type: CTRL C
Response: +C

2-19
SYSTEM BUILD INSTRUCTIONS

Dismount the master tape from Unit 1 and store it in a safe place.

Proceed to Section 2.3.3 if building a BASIC system from DECTape or consult the Table of Contents for the appropriate section if building BASIC from another media. If BASIC will not be used on the system, proceed to Chapter 3.

2.3.3 BASIC/RT-11

This section contains instructions for those who received BASIC/RT-11 on DECTape. The instructions involve placing running versions of BASIC on the system device.

To build a BASIC system, there must be at least 80 free blocks on the system device. If the system device is a DECTape created in Section 2.3.1 or 2.3.2, there are not enough free blocks available; a new system device must be created before the BASIC files are copied.

If the system device is disk, go to Step 1. Otherwise, create a new DECTape as described in Section 2.3.1, then delete files so that BASIC will fit.

The following files must be on the system device; all others can be deleted.

MONITR.SYS  TT.SYS (the console terminal handler)  LP.SYS (if the system has a line printer)
EDIT.SAV    LIBR.SAV    LINK.SAV    PIP.SAV

The procedure for deleting files is as follows:

Type: \R PIP<CR>
Response: *

To delete .SYS files,

Type: SY:xxxxxx.SYS/Y/D<CR>
Response: *

where xxxxxx is the name of the system file to be deleted. Repeat this command string for each file to be deleted.

To delete system programs,

Type: SY:xxxxxx.xxx/D<CR>
Response: *

where xxxxxx.xxx is the name of the system program to be deleted.

When all unnecessary files have been deleted,

Type: SYI/S<CR>
Response: *
SYSTEM BUILD INSTRUCTIONS

Type: CTRL C  
Response: ⌃C

If the system has a hardware bootstrap capable of bootstrapping the new system DECTape, bootstrap the DECTape; otherwise, perform the bootstrapping procedures in Section A.2.

1. Bootstrap an RT-11 V02C system. Ensure that the system device is WRITE-ENABLEd.
   
   Type: DATE dd-mmm-yyyy<CR>
   
   where dd-mmm-yyyy is the current date in the form 27-NOV-75.
   
   Response: .

2. Mount the BASIC/RT-11 System DECTape (DEC-11-LBACA-D-UC) on Unit 1, WRITE LOCKed.

3. Type: Rحيح<CR>
   
   Response: *
   
   Type:  #DD11BASIC,SAV,BAS8K,SAV,DEMO,BAS/X<CR>
   
   Response: *

   Dismount the master DECTape and store it in a safe place.

   Type: CTRL C  
   Response: ⌃C

   Proceed to Chapter 3.

2.4 BUILDING AND STARTING ON DISKETTE

2.4.1 RT-11

This section contains instructions for those who received RT-11 on diskette. If diskette is to be the system device, the user is instructed how to copy the master diskette. If another disk (RK11, RF11, or RP02) is to be the system device, the user is instructed how to make the larger disk system from the master diskette. It is important (for the user's protection) to build the system as instructed, then store the master in a safe place.

1. Mount the RT-11 master diskette (Disk 1 of 2, DEC-11-ORTSA-E-YC1) in the left-hand drive (Unit 0). Set the ENABLE/HALT switch to Halt to stop any previous program that may be running. Set the ENABLE/HALT switch to ENABLE.

2. If the system has a hardware bootstrap capable of bootstrapping the diskette, boot the diskette and proceed to Step 3; otherwise, see Section A.3, then proceed to Step 3.

3. There is a slight pause after which the following message is displayed on the terminal:

2-21  July 1976
SYSTEM BUILD INSTRUCTIONS

RT-11SJ V02C-xx

If not, repeat this section from Step 2.

Type: DATE dd-mmm-yy<CR>

where dd-mmm-yy is the current date in the form 27-NOV-75.

Users of diskette as the system device, proceed to Step 4.
Users of other disks as the system device, proceed to Step 6.

4. Type: R PIP<CR>
   Response: *

Mount a blank diskette in Unit 1.

Type: DX11/Z<CR>
   Response: DX11 /Z ARE YOU SURE?

Type: Y<CR>
   Response: *

Type: DX11,==DX01,=/X/Y<CR>

(The system will be copied onto the blank diskette. The process takes about one minute.)

Response: *

Type: DX11A=DX11MONITR;SYS/U<CR>

(The system bootstrap will be copied onto the diskette on Unit 1.)

Response: *

Type: DX11/K<CR>
   Response: *

Each block of the diskette being built will be checked for errors. This process takes about 30 seconds.

NOTE

If the response is anything but the above, see the RT-11 System Reference Manual, Section 4.2.12.

Remove the master RT-11 System Disk and store in a safe place. Dismount the new disk from Unit 1, label it as RT-11 SYSTEM DISK 1.

5. Mount the RT-11 System Disk (Disk 2 of 2, DEC-11-ORTSA-E-YC2) on Unit 0. Mount a blank diskette in Unit 1.

Type: DX11/Z<CR>
   Response: DX11 /Z ARE YOU SURE?

Type: Y<CR>
   Response: *
SYSTEM BUILD INSTRUCTIONS

Type: DX11*;**DX21*,*/Y/X<CR>
(The system will be copied onto the blank disk. The process takes about one minute.)

Response: *

Type: DX11/K<CR>
Response: *

Each block of the disk being built will be checked for errors. This process takes about 30 seconds.

NOTE

If the response is anything but the above, see the RT-11 System Reference Manual, Section 4.2.12.

Remove the master RT-11 System Disk from Unit 0 and store it in a safe place. Remove the disk from Unit 1 and label it RT-11 SYSTEM DISK 2.

Mount the newly created RT-11 SYSTEM DISK 1 on Unit 0.

Bootstrap the new disk with the hardware bootstrap or the bootstrapping procedure in Section A.3.

RT-11 will boot off the new disk and display the following on the terminal:

RT-11SJ V02C-xx

If it does not, repeat this section from Step 1.

Proceed to Section 2.4.2 if building a FORTRAN system from diskette, Section 2.4.3 if building a BASIC system from diskette, or consult the Table of Contents for the appropriate section if building from another media. If neither FORTRAN nor BASIC will be used on the system, go to Chapter 3.


If the disk is being used to build a system for RK11 DECpack or RP02 disk, mount a blank, formatted disk on Unit 0, WRITE ENABLED. (See Appendix B for RK11 formatting procedures.)

If the system is to be built on RK11 disk,

Type: ASSIGN RK1DK<CR>
Response: *

If the system is to be built on RP11 disk,

Type: ASSIGN RP1DK<CR>
Response: *
If the system is to be built on RP02 disk,

Type: ASSIGN DP1DK]<CR>
Response: *

In any case,

Type: RIP]<CR>
Response: *

If the system device will be RP02 disk,

Type: D1/2/N137]<CR>

If the system device will be a single platter RF11 disk,

Type: D1/2]<CR>

Otherwise,

Type: D1/2/N112]<CR>

In any case,

Response: D1/2 ARE YOU SURE?

Type: Y]<CR>
Response: *

Type: D1*;*=D101*;*/Y/X]<CR>
Response: *

Type: D1*;*=D111*;*/Y/X]<CR>
Response: *

Type: D1DXMNSJ;SYS=D1MONITR;SYS/Y/R]<CR>
Response: *

7. If the disk being built is an RK11,

Type: D1MONITR;SYS=D1RKMNSJ;SYS/Y/R]<CR>
Response: *

Go to Step 8.

If the disk being built is an RF11,

Type: D1MONITR;SYS=D1RFMNSJ;SYS/Y/R]<CR>
Response: *

Go to Step 8.

If the disk being built is an RP02,

Type: D1MONITR;SYS=D1DPMNSJ;SYS/Y/R]<CR>
Response: *

Go to Step 10.

8. Delete all unnecessary monitor and .SYS files from the system device as follows:
SYSTEM BUILD INSTRUCTIONS

Type:            DK:xxMNSJ.SYS,xxMNFB.SYS/Y/D<CR>
Response:       *

where xx is the designation of the device for any unnecessary monitors as follows:

   DP     RP02 disk
   DS     RJS03/4 disk
   DT     DECTape
   DX     RX11 diskette
   RF     RF11 disk
   RK     RK11 disk

Type:            DK:xx.SYS/Y/D<CR>
Response:       *

where xx is the designation of the device for any unnecessary device handler on the system as follows:

Delete                        If System has no
BA.SYS                        more than 8K words of memory
CR.SYS                        card reader
CT.SYS                        cassette
DP.SYS                        RP02 disk
DS.SYS                        RJS03/4 disk
DT.SYS                        DECTape
LP.SYS                        line printer
MM.SYS                        TJU16 magtape
MT.SYS                        TM11/TS03 or TM11/TU10 magtape
PP.SYS                        paper tape punch
PR.SYS                        paper tape reader
RF.SYS                        RF11 disk
RK.SYS                        RK11 disk

Type:            DK1/S<CR>
Response:        *

9.  

Type:            DK1/K<CR>
Response:        *

Each block of the disk being built will be checked for errors. This process takes about four minutes for RK11 or about one minute for single platter RF11 or RJS03/4.

NOTE

If the response is anything but the above, see the RT-11 System Reference Manual, Section 4.2.12.

10. 

Type:            DK1#DKIMONTR.SYS/U<CR>
Response:        *

Type:            DK1/Q<CR>
Response:        RT-11SJ V02C-xx
.

If it does not, repeat this entire section.
Dismount the master diskettes from Units 0 and 1 and store them in a safe place.

Proceed to Section 2.4.2 if building a FORTRAN system from diskette, Section 2.3.3 if building a BASIC system from diskette, or consult the Table of Contents for the appropriate section if building from another media. If neither FORTRAN nor BASIC will be used on the system, go to Chapter 3.

2.4.2 FORTRAN IV

This section contains instructions for those who received RT-11 FORTRAN on diskette. The instructions involve the transfer of the necessary FORTRAN-related files to the system device and the creation of the FORTRAN library.

To build a FORTRAN system, there must be at least 370 free blocks on the system device. If the system device is diskette, single platter RF11 disk, or single platter RJS03/4 disk, there are not enough free blocks available; a new system device must be created before the FORTRAN files are copied.

If the system device is RK11, RP02, multiple platter RF11, or multiple platter RJS03/4, go to Step 1.

Otherwise, if the system device is diskette, create a new diskette as described in Section 2.4.1, then delete files so that FORTRAN will fit. If the system device is single platter RF11 or single platter RJS03/4, files must be deleted from the disk so that FORTRAN will fit.

The following files must be on the system device; all others can be deleted.

MONITR.SYS
TT.SYS (the console terminal handler)
LP.SYS (if the system has a line printer)
DX.SYS (if the system device is not diskette)
EDIT.SAV
LIBR.SAV
LINK.SAV
PIP.SAV

The procedure for deleting files is as follows:

Type: \R PIP<CR>
Response: *

To delete .SYS files,

Type: SY:xxxxxx.SYS/Y/D<CR>
Response: *

where xxxxxxx is the name of the system file to be deleted. Repeat this command string for each file to be deleted.

To delete system programs,

Type: SY:xxxxxx.xxx/D<CR>
SYSTEM BUILD INSTRUCTIONS

Response: *

where xxxxxx.xxx is the name of the system program to be deleted.

When all unnecessary files have been deleted,

Type: SY1/S<CR>
Response: *

Type: CTRL C
Response: *C

If the system has a hardware bootstrap capable of bootstrapping the
new system device, bootstrap the device; otherwise, perform the
bootstrapping procedures in the appropriate section of Appendix A.

1. Bootstrap an RT-11 V02C system.

Mount the RT-11 FORTRAN master diskette (DEC-11-LRF4A-C1-YC) on Unit 1.

Type: DATE dd-mmm-yy<CR>

where dd-mmm-yy is the current date in the form 27-NOV-75.

Response: .

Type: TIME hh:mm:ss<CR>

where hh:mm:ss is the current 24-hour time in the form hours:minutes:seconds.

Response: .

Type: R PIP<CR>
Response: *

Type: SY1","*DX1I, FORTRA, SAV1, FORTRA, HLP, DEMO, FOR/X<CR>
Response: *

Type: CTRL C
Response: *C

Type: R LIBR<CR>
Response: *

NOTE

In the following instructions, the square brackets [] are part of the command line and must be typed.

2. If the configuration includes an EAE, proceed. Otherwise, go to Step 3.

Type: SY1FORLIB[142]=DX1I, UNI, CTSCOM, V2NS, EAE/G <CR>
Response: ENTRY POINT
SYSTEM BUILD INSTRUCTIONS

Type: $ERRTB<CR>
$ERRS<CR>
<CR>
Response: $END2 ILL INS
$END2 ILL INS
$F102 ILL INS
$C02 ILL INS
$F102 ILL INS
$F102 ILL INS

If the system device is diskette, go to Step 7; otherwise,

Type: SYIFORLIB;V2S140J=DX11UNI,OTSCOM,V2S,EAE/G<CR>
Response: ENTRY POINTI

Type: $ERRTB<CR>
$ERRS<CR>
<CR>
Response: $END2 ILL INS
$END2 ILL INS
$F102 ILL INS
$C02 ILL INS
$F102 ILL INS
$F102 ILL INS

Go to Step 7.

3. If the configuration includes a PDP-11/45 processor without FPU or a PDP-11/40 processor with EIS but without FIS, proceed. Otherwise, go to Step 4.

Type: SYIFORLIB[140J=DX11UNI,OTSCOM,V2NS,EIS/G<CR>
Response: ENTRY POINTI

Type: $ERRTB<CR>
$ERRS<CR>
<CR>
Response: $END2 ILL INS
$END2 ILL INS
$F102 ILL INS
$C02 ILL INS
$F102 ILL INS
$F102 ILL INS

If the system device is diskette, go to Step 7; otherwise,

Type: SYIFORLIB;V2S140J=DX11UNI,OTSCOM,V2S,EIS/G<CR>
Response: ENTRY POINTI

Type: $ERRTB<CR>
$ERRS<CR>
<CR>
SYSTEM BUILD INSTRUCTIONS

Response: $SEND2 ILL INS
$SEND2 ILL INS
$SF102 ILL INS
$SCLO2 ILL INS
$SF102 ILL INS
$SF102 ILL INS

* 

Go to Step 7.

4. If the configuration includes a PDP-11/40 or PDP-11/03 processor with FIS, proceed. Otherwise, go to Step 5.

Type: SYIFORLIB[140]=DX11UNI,OTSCOM,V2NS,FIS/G<CR>
Response: ENTRY POINT1

Type: $SERRTB<CR>
$SERRS<CR>
<CR>

Response: $SEND2 ILL INS
$SEND2 ILL INS
$SF102 ILL INS
$SCLO2 ILL INS
$SF102 ILL INS
$SF102 ILL INS

* 

If the system device is diskette, go to Step 7; otherwise,

Type: SYIFORLIB,V2S[140]=DX11UNI,OTSCOM,V2S,FIS/G<CR>
Response: ENTRY POINT1

Type: $SERRTB<CR>
$SERRS<CR>
<CR>

Response: $SEND2 ILL INS
$SEND2 ILL INS
$SF102 ILL INS
$SCLO2 ILL INS
$SF102 ILL INS
$SF102 ILL INS

* 

Go to Step 7.

5. If the configuration includes a PDP-11/45 processor with FPU, proceed.

Type: SYIFORLIB[140]=DX11UNI,OTSCOM,V2NS,FPU/G<CR>
Response: ENTRY POINT1

Type: $SERRTB<CR>
$SERRS<CR>
<CR>

Response: $SEND2 ILL INS
SYSTEM BUILD INSTRUCTIONS

$END2   ILL INS
$FIO2   ILL INS
$CLO2   ILL INS
$FIO2   ILL INS
$FIO2   ILL INS

*  

If the system device is diskette, go to Step 7; otherwise,

Type: SYF0LRILB,V2S[140]=DX11UNI,OTSCOM,V2S,FPU/G<CR>
Response: ENTRY POINT1

Type: $ERRTB<CR>
$ERRS<CR>
<CR>

Response: $END2   ILL INS
$END2   ILL INS
$FIO2   ILL INS
$CLO2   ILL INS
$FIO2   ILL INS
$FIO2   ILL INS

*  

Go to Step 7.

6. If the configuration contains none of the above options:

Type: SYF0LRILB[140]=DX11UNI,OTSCOM,V2S,NHD/G<CR>
Response: ENTRY POINT1

Type: $ERRTB<CR>
$ERRS<CR>
<CR>

Response: $END2   ILL INS
$END2   ILL INS
$FIO2   ILL INS
$CLO2   ILL INS
$FIO2   ILL INS
$FIO2   ILL INS

*  

If the system device is diskette, go to Step 7; otherwise,

Type: SYF0LRILB,V2S[140]=DX11UNI,OTSCOM,V2S,NHD/G<CR>
Response: ENTRY POINT1

Type: $ERRTB<CR>
$ERRS<CR>
<CR>

Response: $END2   ILL INS
$END2   ILL INS
$FIO2   ILL INS
$CLO2   ILL INS
$FIO2   ILL INS
$FIO2   ILL INS

*
7. Type: CTRL C
   Response: *C

   Proceed to Section 2.4.3 if building a BASIC system from diskette or consult the Table of Contents for the appropriate section if building BASIC from another media. If BASIC will not be used on the system, go to Chapter 3.

2.4.3 BASIC/RT-11

This section contains instructions for those who received BASIC/RT-11 on diskette. The instructions involve placing running versions of BASIC on the system device.

To build a BASIC system, there must be at least 80 free blocks on the system device. If the system device is a diskette created in Section 2.4.1 or 2.4.2, there are not enough free blocks available; a new system device must be created before the BASIC files are copied.

If the system device is not diskette, go to Step 1. Otherwise, create a new diskette as described in Section 2.4.1, then delete files so that BASIC will fit.

The following files must be on the system device; all others can be deleted.

MONITR.SYS
TT.SYS (the console terminal handler)
LP.SYS (if the system has a line printer)
EDIT.SAV
LIBR.SAV
LINK.SAV
PIP.SAV

The procedure for deleting files is as follows:

Type: .R PIP<CR>
Response: *

To delete .SYS files,

Type: SY:xxxxxx.SYS/Y/D<CR>
Response: *

where xxxxxx is the name of the system file to be deleted. Repeat this command string for each file to be deleted.

To delete system programs,

Type: SY:xxxxxx.xxx/D<CR>
Response: *

where xxxxxx.xxx is the name of the system program to be deleted.

When all unnecessary files have been deleted,

Type: SY1/S<CR>
Response: *
SYSTEM BUILD INSTRUCTIONS

Type: CTRL C
Response: *6

If the system has a hardware bootstrap capable of bootstrapping the new system diskette, bootstrap the diskette; otherwise, perform the bootstrapping procedures in Section A.3.

1. Bootstrap an RT-11 VO2C system.

2. Mount the BASIC/RT-11 master diskette (DEC-11-LBACA-D-YC) on Unit 1.

   Type: DATE dd-mmm-yy<CR>
   
   where dd-mmm-yy is the current date in the form 27-NOV-75.
   
   Response: ;

3. Type: R PIP<CR>
   Response: *

   Type: *rittAEXL,BASIC,SAY,BAS8K,SAY,DEMO,SAY,X<CR>
   Response: *

   Dismount the master disk and store it in a safe place.

   Type: CTRL C
   Response: *6

   Proceed to Chapter 3.

2.5 BUILDING AND STARTING FROM MAGTAPE

2.5.1 RT-11

This section contains instructions for those who received RT-11 on magtape. The instructions describe building an RK11, RF11, RPO2, or RJS03/4 disk system from the master magtape. It is important (for the user's protection) to build the system as instructed, then to store the master in a safe place.

1. Set the ENABLE/HALT switch to HALT to stop any previous program that may be running. Set the ENABLE/HALT switch to ENABLE.

   Mount the RT-11 System magtape (DEC-11-ORTSA-E-MC7 for 7-track tape or DEC-11-ORTSA-E-MC9 for 9-track tape) on Unit 0, ensuring that no write ring is inserted in the back of the tape reel. If TM11, ensure that the magtape is positioned at the load point; if it is not, manually rewind the tape.

2. If the system has a hardware bootstrap capable of bootstrapping the magtape, boot the magtape and proceed to Step 3; otherwise, see Section A.4, then proceed with Step 3.

2-32

July 1976
3. The magtape will move as the magtape build program is loaded.

A program called MSBOOT should respond by displaying the following message on the terminal:

```
MSBOOT V01-xx
```

*If not, repeat this section from Step 2.

Type: `MBUILD.xxx`

where xxx indicates the type of interface and is one of the following:

- **MT1** builds RK11, RP02, or RJS03/4 from TM11/TS03 and TM11/TU10 magtapes
- **MT2** builds RF11 disk from TM11/TS03 and TM11/TU10 magtapes
- **MM1** builds RK11, RP02, or RJS03/4 from TJU16 magtape
- **MM2** builds RF11 disk from TJU16 magtape

4. The magtape will move as the specified MBUILD program is loaded. MBUILD should respond by displaying the following message:

```
MBUILD V02-xx
```

*If it does not, repeat this Section from Step 2.

5. If the system being built is an RK11 DECpack or an RP02 disk cartridge, mount a blank, formatted disk on Unit 0, WRITE ENABLED. Formatting procedures are explained in Appendix B.

If the system device will be RK11 disk,

Type: `RK1/Z/N112<CR>`
Response: `RK1 /Z ARE YOU SURE?`

Type: `Y<CR>`
Response: `*`

Go to Step 6.

If the system device will be a single platter RF11 disk,

Type: `RF1/Z<CR>`

If the system device will be multiple platter RF11 disk,

Type: `RF1/Z/N112<CR>`

In either case,

Response: `RF1 /Z ARE YOU SURE?`

Type: `Y<CR>`
Response: `*`

Go to Step 6.
If the system device will be RP02 disk,

Type:  \texttt{DP1/Z/N137}\texttt{(CR)}
Response:  \texttt{DP1 /Z ARE YOU SURE?}

Type:  \texttt{Y}\texttt{(CR)}
Response:  *

Go to Step 6.

If the system device will be a single platter RJS03/4 disk,

Type:  \texttt{DS1/Z}\texttt{(CR)}

If the system device will be a multiple platter RJS03/4 disk,

Type:  \texttt{DS1/Z/N112}\texttt{(CR)}

In either case,

Response:  \texttt{DS1 /Z ARE YOU SURE?}

Type:  \texttt{Y}\texttt{(CR)}
Response:  *

6. If the master magtape is on TM11,

Type:  \texttt{xx.*=MT0:.*/Y/X/Q/M:1000}\texttt{(CR)}

If the master magtape is on TJU16,

Type:  \texttt{xx.*=MM0:.*/Y/X/Q/M:1000}\texttt{(CR)}

where \texttt{xx} is the designation of the device to be the system device as follows:

- \texttt{DP}  RP02 disk
- \texttt{DS}  RJS03/4 disk
- \texttt{RF}  RP11 disk
- \texttt{RK}  RK11 disk

MBUILD will respond by printing the names of each system component. Type \texttt{Y}\texttt{(CR)} after the component name if you desire to transfer it to the system device. Type \texttt{<CR>} after the component name if you do not desire to transfer it to the system device.

Minimally, the system device must contain MBOOT.BOT, MSBOOT.BOT, MTINIT.SAV, the appropriate MBUILD file (selected in Step 3), the appropriate monitors (as described in Table 1-1), TT.SYS and the other necessary device handlers, and the desired system programs. The following example shows the building of a typical RK11 disk as the system device from TJU16 magtape master:

\texttt{MSBOOT;BOT?Y\texttt{(CR)}}
\texttt{MBUILD;MT1?\texttt{(CR)}}
\texttt{MBUILD;MT2?\texttt{(CR)}}
\texttt{MBUILD;MM1?Y\texttt{(CR)}}
\texttt{MBUILD;MM2?\texttt{(CR)}}
\texttt{RKMNJS;SYS?Y\texttt{(CR)}}
\texttt{RKMNFB;SYS?\texttt{(CR)}}
\texttt{RFMNJS;SYS?\texttt{(CR)}}
\texttt{RFMNFB;SYS?\texttt{(CR)}}
SYSTEM BUILD INSTRUCTIONS

NOTE

A maximum of 54 files can be transferred with this procedure. If more than 54 files are specified to be transferred, a ?COR OVR? message will appear and you must restart the selection procedure.

7. If the system device will be RK11 disk,

Response: ?REBOOT?

2-35
SYSTEM BUILD INSTRUCTIONS

Type: RKMIMONTR;SYS=RRKIRKMNSJ;SYS/Y/R<CR>
Response: *REBOOT?

Type: RKMIA=RKMIMONTR;SYS/U<CR>
Response: *

Type: RKMIO<CR>

Go to Step 8.

If the system device will be RFlI disk,

Response: *

Type: RFMONTR;SYS=RFlRFNSJ;SYS/Y/R<CR>
Response: *

Type: RFIA=RFMONTR;SYS/U<CR>
Response: *

Type: RFI/O<CR>

Go to Step 8.

If the system device will be RJS03/4 disk,

Response: *

Type: DSIMONTR;SYS=DS1DSNSJ;SYS/Y/R<CR>
Response: *

Type: DSI:A=DSIMONTR;SYS/U<CR>
Response: *

Type: DSI/O<CR>

Go to Step 8.

If the system device will be RP02 disk,

Response: *

Type: DPMONTR;SYS=DPMNSJ;SYS/Y/R<CR>
Response: *

Type: DPMIA=DPMONTR;SYS/U<CR>
Response: *

Type: DPMIO<CR>

8. RT-ll should bootstrap off the new disk and respond with its identifying message:

RT-llSJ V02C-xx

If it does not, repeat this entire section.
If it does, remove the master magtape from Unit 0 and store it in a safe place.
SYSTEM BUILD INSTRUCTIONS

Type: DATE dd-mmm-yy

where dd-mmm-yy is the current date in the form 27-NOV-75.

If RP02 is the new system device, go to Step 10.

If RK11 is the new system device,

Type: ASSIGN RK11D<CR>

Go to Step 9.

If RF11 is the new system device,

Type: ASSIGN RF11D<CR>

Go to Step 9.

If RSJ03/4 is the new system device,

Type: ASSIGN DS11D<CR>

9. Response: *

Type: RPIP<CR>
Response: *

Type: DK1/K<CR>
Response: *

Each block of the new system device will be read for errors.
This process takes about four minutes on RK11 or one minute
on a single platter RF11 or RSJ03/4.

NOTE

If the response is anything but the above, see the

Type: CTRL C
Response: *C

10. Proceed to Section 2.5.2 if building a FORTRAN system from
magtape, Section 2.5.3 if building a BASIC system from
magtape, or consult the Table of Contents for the appropriate
section if building from another media. If neither FORTRAN
nor BASIC will be used on the system, go to Chapter 3.

2.5.2 FORTRAN IV

This section contains instructions for those who received RT-11
FORTRAN on magtape. The instructions involve the transfer of the
necessary FORTRAN-related files to the system device and the creation
of the FORTRAN library.
SYSTEM BUILD INSTRUCTIONS

The system device onto which FORTRAN is being built must have at least 370 free blocks. The monitor, the appropriate magtape handler (MM.SYS or MT.SYS), TT.SYS, LP.SYS (if the configuration includes a line printer), PIP.SAV, LINK.SAV, and LIBR.SAV are the only system components necessary on the system device.

1. Bootstrap an RT-11 V02C system (if you have completed Section 2.5.1, the system is already booted and running).

   Mount the RT-11 FORTRAN master magtape (DEC-11-LRF4A-C1-MC7 for 7-track tape or DEC-11-LRF4A-C1-MC9 for 9-track tape) on Unit 0.

   Type: DATE dd-mmm-yyyy<CR>

   where dd-mmm-yyyy is the current date in the form 27-NOV-75.

   Response: .

   Type: TIME hh:mm:ss<CR>

   where hh:mm:ss is the current 24-hour time in the form hours:minutes:seconds.

   Response: .

   Type: R PIP<CR>

   Response: *

   If the master magtape is on TML11,

   Type: SYM*;*MT01;,.*/X/M1880<CR>

   If the master magtape is on TJU6,

   Type: SYM*;*MM01;,.*/X/M1880<CR>

   In either case,

   Response: *

   Type: CTRL C

   Response: C

   Type: R LIBR<CR>

   Response: *

NOTE

In the following instructions, the square brackets [] are part of the command line and must be typed.

2. If the configuration includes an EAE, proceed. Otherwise, go to Step 3.

   Type: SYFORLIB[140]*SYM1;UNI;OTSCOM;V2NS;EAE/G<CR>

   Response: ENTRY POINT

   Type: SERRT6<CR>

   SERRS<CR>

2-38  July 1976
SYSTEM BUILD INSTRUCTIONS

Response: SEND2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCL02 ILL INS
SF102 ILL INS
SF102 ILL INS

* 

Type: SYIFORLIB;V2S[140]=SYIUNI,OTSCOM,V2S,EIS/G<CR>
Response: ENTRY POINT

Type: $ERRT<CR>
$ERRS<CR>

Response: SEND2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCL02 ILL INS
SF102 ILL INS
SF102 ILL INS

* 

Go to Step 7.

3. If the configuration includes a PDP-11/45 processor without FPU or a PDP-11/40 processor with EIS but without FIS, proceed. Otherwise, go to Step 4.

Response: SEND2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCL02 ILL INS
SF102 ILL INS
SF102 ILL INS

* 

Type: SYIFORLIB[140]=SYIUNI,OTSCOM,V2NS,EIS/G<CR>
Response: ENTRY POINT

Type: $ERRT<CR>
$ERRS<CR>

Response: SEND2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCL02 ILL INS
SF102 ILL INS
SF102 ILL INS

* 

Type: SYIFORLIB;V2S[140]=SYIUNI,OTSCOM,V2S,EIS/G<CR>
Response: ENTRY POINT

Type: $ERRT<CR>
$ERRS<CR>

Response: SEND2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCL02 ILL INS
SF102 ILL INS
SF102 ILL INS

* 

Go to Step 7.
4. If the configuration includes a PDP-11/40 or PDP-11/03 processor with FIS, proceed. Otherwise, go to Step 5.

Type: \texttt{SYIFORLIB[140J=SYIUNI,OTSCom,V2NS,FIS/G}<CR>}
Response: \texttt{ENTRY POINTI}

Type: \texttt{$ERRTB<CR>}
$ERRS<CR>
<CR>

Response: 
\texttt{SEND2 ILL INS}
\texttt{SEND2 ILL INS}
\texttt{SFI02 ILL INS}
\texttt{SCL02 ILL INS}
\texttt{SFI02 ILL INS}
\texttt{SFI02 ILL INS}

* \texttt{

Type: \texttt{SYIFORLIB,V2S[140J=SYIUNI,OTSCom,V2S,FIS/G}<CR>}
Response: \texttt{ENTRY POINTI}

Type: \texttt{$ERRTB<CR>}
$ERRS<CR>
<CR>

Response: 
\texttt{SEND2 ILL INS}
\texttt{SEND2 ILL INS}
\texttt{SFI02 ILL INS}
\texttt{SCL02 ILL INS}
\texttt{SFI02 ILL INS}
\texttt{SFI02 ILL INS}

* \texttt{

Go to Step 7.

5. If the configuration includes a PDP-11/45 processor with FPU, proceed.

Type: \texttt{SYIFORLIB[140J=SYIUNI,OTSCom,V2NS,FPU/G}<CR>}
Response: \texttt{ENTRY POINTI}

Type: \texttt{$ERRTB<CR>}
$ERRS<CR>
<CR>

Response: 
\texttt{SEND2 ILL INS}
\texttt{SEND2 ILL INS}
\texttt{SFI02 ILL INS}
\texttt{SCL02 ILL INS}
\texttt{SFI02 ILL INS}
\texttt{SFI02 ILL INS}

* \texttt{

Type: \texttt{SYIFORLIB,V2S[140J=SYIUNI,OTSCom,V2S,FPU/G}<CR>}
Response: \texttt{ENTRY POINTI}

Type: \texttt{$ERRTB<CR>}
$ERRS<CR>
<CR>
SYSTEM BUILD INSTRUCTIONS

Response: $END2 ILL INS
     $END2 ILL INS
     $F102 ILL INS
     $GLO2 ILL INS
     $F102 ILL INS
     $F102 ILL INS

* Go to Step 7.

6. If the configuration contains none of the above options:

Type: SYIFORLIBC140J#SYIUNI,OTSCOM,V2NS,NHD/G<CR>
Response: ENTRY POINT1

Type: $ERRTG<CR>
     $ERRS<CR>
     <CR>

Response: $END2 ILL INS
     $END2 ILL INS
     $F102 ILL INS
     $GLO2 ILL INS
     $F102 ILL INS
     $F102 ILL INS

* Type: SYIFORLIB,V2S140J#SYIUNI,OTSCOM,V2S,NHD/G<CR>
Response: ENTRY POINT1

Type: $ERRTG<CR>
     $ERRS<CR>
     <CR>

Response: $END2 ILL INS
     $END2 ILL INS
     $F102 ILL INS
     $GLO2 ILL INS
     $F102 ILL INS
     $F102 ILL INS

*

7. Type: CTRL C
Response: *C

Proceed to Section 2.5.3 if building a BASIC system from magtape or consult the Table of Contents for the appropriate section if building BASIC from another media. If BASIC will not be used on the system, go to Chapter 3.

2.5.3 BASIC/RT-11

This section contains instructions for those who received BASIC/RT-11 on magtape. The instructions involve placing running versions of BASIC on the system device.
The system device onto which BASIC is being built must have at least 80 free blocks. The monitor, the appropriate magtape handler (MM.SYS or MT.SYS), and PIP.SAV are the only system components necessary on this system device.

1. Bootstrap an RT-11 V02C system (if you have completed Section 2.5.1 or 2.5.2, the system is already booted and running).

2. Mount the BASIC/RT-11 master magtape (DEC-11-LBACA-D-MC7 for 7-track tape or DEC-11-LBACA-D-MC9 for 9-track tape on Unit 0.

   Type: DATE dd-mmm-yyyy<CR>

   where dd-mmm-yyyy is the current date in the form 27-NOV-75.

   Response: *

3. Type: R PIP<CR>

   Response: *

   If the master magtape is on MT11,

   Type: *,*=MT01BASIC,SAV,BASEK,SAV,DEMO,BAS/X/H/1000<CR>

   If the master magtape is on TJ16,

   Type: *,*=MM01BASIC,SAV,BASEK,SAV,DEMO,BAS/X/H/1000<CR>

   In either case,

   Response: *

   Dismount the master magtape and store it in a safe place.

   Type: CTRL C

   Response: *C

   Proceed to Chapter 3.

2.6 BUILDING AND STARTING FROM CASSETTE

2.6.1 RT-11

This section contains instructions for those who received RT-11 on cassettes. The instructions describe the building of an RK11 disk system from cassette. RK11 users will need a blank, formatted disk for this procedure. See Appendix B for formatting procedures.

Prior to inserting any master cassette during this procedure, WRITE PROTECT the data by placing the orange tabs so that the holes are uncovered. Cassette Unit 0 is on the left and cassette Unit 1 is on the right.

1. Set the ENABLE/HALT switch to HALT to stop any previous program that may be running. Set the ENABLE/HALT switch to ENABLE.
SYSTEM BUILD INSTRUCTIONS

Mount a blank, formatted disk cartridge in Unit 0, WRITE ENABLED. (See Appendix B for formatting procedures.)

Insert the RT-11 System Cassette, Tape 1 of 5 (DEC-11-ORTSA-E-TC1), into Unit 0.

2. If the system has a hardware bootstrap capable of bootstrapping the cassette, boot the cassette and proceed to Step 3; otherwise, see Section A.5, then proceed with Step 3.

3. The cassette will move as the cassette build program is loaded.

CBUILD should respond by displaying the following message on the terminal:

```
CBUILD VERSION

If it does not, repeat this section from Step 1.

Press the rewind button on the cassette drive.
```

Type: \texttt{RK1/E/NI12<CR>}
Response: \texttt{ARE YOU SURE?}

Type: \texttt{<CR>}
Response: \texttt{Y<CR>}

Type: \texttt{RK11*/CT21*/Y/X<CR>}
Response: \texttt{REBOOT?}

4. Insert the RT-11 System Cassette, Tape 2 of 5 (DEC-11-ORTSA-E-TC2), into cassette Unit 1. WRITE PROTECT it before insertion (as described above).

Press the rewind button on the cassette drive.

Type: \texttt{RK11*/CT11*/Y/X<CR>}
Response: \texttt{REBOOT?}

5. Remove the RT-11 System Cassette, Tape 2 of 5, from Unit 1 and mount the RT-11 System Cassette, Tape 3 of 5 (DEC-11-ORTSA-E-TC3), in Unit 1, WRITE PROTECTED.

Type: \texttt{RK11*/CT11*/X<CR>}
Response: \texttt{*}

6. Remove the RT-11 System Cassette, Tape 3 of 5, from Unit 1 and mount the RT-11 System Cassette, Tape 4 of 5 (DEC-11-ORTSA-E-TC4), in Unit 1, WRITE PROTECTED.

Type: \texttt{RK11*/CT11*/X<CR>}
Response: \texttt{*}
7. Remove the System Cassette, Tape 4 of 5, from Unit 1 and mount the RT-lll System Cassette, Tape 5 of 5 (DEC-11-ORTSA-E-TC5), in Unit 1, WRITE PROTECTED.

Type: \textbackslash{RK1A=CT01;S/X/Y<CR>}
Response: \textbackslash{REBOOT?}

Type: \textbackslash{RK1A=CT01.MONITOR;SYS/U<CR>}
Response: 

8. Remove the master tapes from Units 0 and 1.

Type: \textbackslash{RK1/O<CR>}
Response: RT-llSJ V02C-xx

The system is now running from the disk. If the above message is not displayed, repeat this section.

Proceed to Section 2.6.2 if building a FORTRAN system from cassette, Section 2.6.3 if building a BASIC system from cassette, or consult the Table of Contents for the appropriate section if building from another media. If neither FORTRAN nor BASIC will be used on the system, go to Chapter 3.

2.6.2 FORTRAN IV

This section contains instructions for those who received RT-11 FORTRAN on cassette. The instructions involve the transfer of the necessary FORTRAN-related files to the system device and the creation of the FORTRAN library.

1. Bootstrap an RT-11 V02C system (if you have completed Section 2.6.1, the system is already booted and running).

Type: \textbackslash{DATE dd-mmm-yy<CR>}

where dd-mmm-yy is the current date in the form 27-NOV-75.

Response: 

Type: \textbackslash{R PIP<CR>}
Response: 

Mount the RT-11 FORTRAN IV System Cassette, Tape 1 of 4 (DEC-11-LRF4A-Cl-TC1), in cassette Unit 0. Before loading the cassette, make certain that it is WRITE PROTECTED.

Type: \textbackslash{#,*CT01;*,/X/H:1000<CR>}
Response: 

2. Remove the system cassette, Tape 1 of 4, from Unit 0 and mount the RT-11 FORTRAN IV System Cassette, Tape 2 of 4 (DEC-11-LRF4A-Cl-TC2), in Unit 0, WRITE PROTECTED.

Type: \textbackslash{#,*CT01;*,/X/H:1000<CR>}
Response: 

3. Remove the system cassette, Tape 2 of 4, from Unit 0 and mount the RT-11 FORTRAN IV System Cassette, Tape 3 of 4 (DEC-11-LRF4A-Cl-TC3), in Unit 0, WRITE PROTECTED.
4. Remove the system cassette, Tape 3 of 4, from Unit 0 and mount the RT-ll FORTRAN IV System Cassette, Tape 4 of 4 (DEC-ll-LRF4A-C1-TC4), in Unit 0, WRITE PROTECTED.

Type: \*\*CT010;**/X/M1000<CR>
Response: *

Type: \*\*CT010;**/X/M1000<CR>
Response: *

Type: CTRL C
Response: ^C

5. Store all four master cassettes in a safe place.

Type: R LIBR<CR>
Response: *

6. If the FORTRAN system will be running on a configuration that includes an EAE, proceed. Otherwise, go to Step 7.

Type: FORLIB=UNI,OTSCOM,V2NS,EAE/G<CR>
Response: ENTRY POINT

Type: SERRB<CR>
SERRS<CR>
<CR>

Response: SEND2 ILL INS
SEND2 ILL INS
SFI02 ILL INS
SCL02 ILL INS
SFI02 ILL INS
SFI02 ILL INS
*

Type: FORLIB,V2S=UNI,OTSCOM,V2S,EAE/G<CR>
Response: ENTRY POINT

Type: SERRB<CR>
SERRS<CR>
<CR>

Response: SEND2 ILL INS
SEND2 ILL INS
SFI02 ILL INS
SCL02 ILL INS
SFI02 ILL INS
SFI02 ILL INS
*

Go to Step 11.

7. If the configuration includes a PDP-11/45 processor without FPU or a PDP-11/40 processor with EIS but without FIS, proceed. Otherwise, go to Step 8.

Type: FORLIB=UNI,OTSCOM,V2NS,EIS/G<CR>
Response: ENTRY POINT

2-45 July 1976
SYSTEM BUILD INSTRUCTIONS

Type: SEND2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCLO2 ILL INS
SF102 ILL INS
SF102 ILL INS

* 

Type: FORLIB,V2S=UNI,OTSCOM,V2S,EIS/G<CR>
Response: ENTRY POINT1

Type: SEND2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCLO2 ILL INS
SF102 ILL INS
SF102 ILL INS

* 

Go to Step 11.

8. If the configuration includes a PDP-ll/40 or PDP-ll/03 processor with FIS, proceed. Otherwise, go to Step 9.

Type: FORLIB=UNI,OTSCOM,V2NS,FIS/G<CR>
Response: ENTRY POINT1

Type: SEND2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCLO2 ILL INS
SF102 ILL INS
SF102 ILL INS

* 

Type: FORLIB,V2S=UNI,OTSCOM,V2S,FIS/G<CR>
Response: ENTRY POINT1

Type: SEND2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCLO2 ILL INS
SF102 ILL INS
SF102 ILL INS

* 

2-46
Go to Step 11.

9. If the configuration includes a PDP-11/45 processor with FPU, proceed. Otherwise, go to step 10.

Type: `FORLIB=UNI,OTSCOM,V2NS,FPU/G<CR>`
Response: `ENTRY POINTI`

Type: `SERRTB<CR>`
`SERRS<CR>`

Response:
```
SEND2  ILL INS
SEND2  ILL INS
SF102  ILL INS
SC102  ILL INS
SF102  ILL INS
SF102  ILL INS
```


Type: `FORLIB,V2S=UNI,OTSCOM,V2S,FPU/G<CR>`
Response: `ENTRY POINTI`

Type: `SERRTB<CR>`
`SERRS<CR>`

Response:
```
SEND2  ILL INS
SEND2  ILL INS
SF102  ILL INS
SC102  ILL INS
SF102  ILL INS
SF102  ILL INS
```

Go to Step 11.

10. If the configuration contains none of the above options:

Type: `FORLIB=UNI,OTSCOM,V2NS,NHD/G<CR>`
Response: `ENTRY POINTI`

Type: `SERRTB<CR>`
`SERRS<CR>`

Response:
```
SEND2  ILL INS
SEND2  ILL INS
SF102  ILL INS
SC102  ILL INS
SF102  ILL INS
SF102  ILL INS
```


Type: `FORLIB,V2S=UNI,OTSCOM,V2S,NHD/G<CR>`
Response: `ENTRY POINTI`
SYSTEM BUILD INSTRUCTIONS

Type:  

Response:

11. Type:  CTRL C  
Response:  \  

Proceed to Section 2.6.3 if building a BASIC system from cassette or consult the Table of Contents for the appropriate section if building BASIC from another media. If BASIC will not be used on the system, go to Chapter 3.

2.6.3 BASIC/RT-11

This section contains instructions for those who received BASIC/RT-11 on cassettes. The instructions involve placing running versions of BASIC on the system device.

1. Bootstrap an RT-11 V02C system (if you have completed Section 2.6.1 or 2.6.2, the system is already booted and running).

   Type:  DATE dd-mmm-yy<CR>  
   where dd-mmm-yy is the current date in the form 27-NOV-75.

   Response:  .

2. WRITE PROTECT the BASIC/RT-11 master cassettes by placing the orange tabs so that the holes are uncovered.

3. Place the BASIC/RT-11 Cassette 1 of 2 (DEC-11-LBACA-D-TCL) into Unit 0.
   Press the rewind button on the cassette drive.

4. Type:  R PIP<CR>  
Response:  *  
   Type:  0,0=CPL1,0/X/M1000<CR>  
Response:  *  

5. Dismount the master cassette on Unit 0 and store it in a safe place.

   Type:  CTRL C  
Response:  *C  

Go to Chapter 3.
2.7 BUILDING AND STARTING ON PAPER TAPE

2.7.1 RT-11

This section contains instructions for those who received RT-11 on paper tape. The instructions first describe loading the Absolute Loader into memory at the 8K boundary, then detail the building of an RK11 or RF11 system from paper tape. RK11 users will need a blank, formatted disk for this procedure.

1. Check to see if the Bootstrap Loader is in memory, starting at 37744. If not, deposit it starting at 37744. (Instructions for checking and depositing the Bootstrap Loader are in Section A.6.)

Load the Absolute Loader (DEC-11-UBLA-A-PO) into the paper tape reader. Press the FEED button until the special leader is over the read head.

Set 37744 in the Switch Register. (Set switches 13 through 5 and 2 to the up (1) position. Set all other switches to the down (0) position.)

Press the LOAD ADDR switch.

Press the START switch.

The paper tape should pass through the reader and then the computer should halt.

If the system being built is an RK11 system, mount a formatted disk cartridge in Unit 0, WRITE ENABLEd. Place the paper tape labeled RK PT BUILD Tape 1 of 3 (DEC-11-ORPBA-D-PB1) in the reader.

If the system being built is an RF11 system, place the tape labeled RF PT BUILD Tape 3 of 3 (DEC-11-ORPBA-D-PB3) in the reader.

Set the starting address of the Absolute Loader, 37500, in the Switch Register. (Set switches 13 through 8 and 6 to the up (1) position. Set all other switches to the down (0) position.)

Press the LOAD ADDR switch.

Press the START switch.

The paper tape should pass through the reader and the following message should be displayed on the terminal:

PT BUILD Vxx-xx

There is a slight pause, after which PT BUILD displays the following on the terminal.

PLACE SECOND TAPE IN READER,
STRIKE ANY CHARACTER TO CONTINUE,

Place the paper tape labeled PT BUILD Tape 2 of 3 (DEC-11-ORPBA-D-PB2) in the reader. Press the FEED button until blank leader (unpunched tape preceding the punched
data) is over the read head. (PT BUILD Tape 2 of 3 serves as the second tape for both RK and RF systems.)

Strike any character on the keyboard to start the read operation.

The paper tape passes through the reader and the following message is displayed on the terminal. (There is a slight pause between reading the tape and displaying the message.)

RT-11 BUILD COMPLETE,
RT-11 V01-15  (A rudimentary V01-15 monitor is used to build V02C)
Type:       DATE dd-mmm-yy<CR>

(where dd-mmm-yy is the current date in the form 27-NOV-76.)

Response:  .

If the console terminal is an ASR or KSR Teletype, a VT50 Alphanumeric Display, a parallel LA30, or an LA36 DECwriter II, go to Step 2.

If the console terminal is a VT05 Alphanumeric Display operating at 300 baud, go to Step 2.

If the console terminal is a serial LA30 DECwriter operating at 300 baud,

Type:       D 56=5015<CR>
Response:   ,

Go to Step 2.

If the console terminal is a serial LA30 DECwriter operating at 150 baud,

Type:       D 56=2015<CR>
Response:   ,

Go to Step 2.

If the console terminal is a VT05 Alphanumeric Display operating at 2400 baud,

Type:       D 56=2012<CR>
Response:   ,

Go to Step 2.

If the console terminal is a VT05 Alphanumeric Display operating at 1200 baud,

Type:       D 56=1012<CR>
Response:   ,

Go to Step 2.

If the console terminal is a VT05 Alphanumeric Display operating at 600 baud,

Type:       D 56=412<CR>
Response:   ,

2-50
2. Type: R LINK<CR>
   Response: *

   Place the paper tape labeled OLDPIP.OBJ (DEC-11-OROPA-D-PR) in the reader. Press the FEED button until blank leader is over the read head.

   Type: OLDPIP=PR<CR>
   Response: *

   (The V01 system being used to build V02C will respond with an up-arrow or circumflex whenever it is ready to read a tape.)

   Strike any character on the keyboard to initiate the read operation.

   The paper tape should pass through the reader.

   Response: *

   Reload OLDPIP.OBJ in the reader so it can be read a second time.

   Strike any character on the keyboard to initiate the read operation.

   The paper tape should pass through the reader.

   Response: *

   Type: CTRL C
   Response: *

   Type: R OLDPIP<CR>
   Response: *

3. Place the paper tape labeled PATCH.OBJ (DEC-11-ORPAA-E-PR) in the reader. Press the FEED button until blank leader is over the read head.

   For this and all the following paper tape instructions in Step 4, the convention xxxxxx.xxx is used to represent the name on the tape label underneath the RT-11 version number. In this first example, the command below will look like PATCH.OBJ=PR:/B<CR>.

   Type: xxxxxx.xxx=PR:/B<CR>
   Response: *

   Strike any character on the keyboard to initiate the read operation.

   The paper tape should pass through the reader.

   Response: *

NOTE

During the following operations, a ?CHK SUM? message displayed during a paper tape input operation means an input error has occurred. Retry the tape if such a message is received.
SYSTEM BUILD INSTRUCTIONS

4. Repeat Step 3 for each of the following paper tapes. Note that only those system components that are needed should be transferred to the system device; see Section 4.5 for further information.

EDIT.OBJ  (DEC-11-ORTEA-E-PR1)
ODT.OBJ  (DEC-11-ORODA-E-PR)
TT.OBJ  (DEC-11-ORTTA-E-PR)
LP.OBJ  (DEC-11-ORTLA-E-PR)
PP.OBJ  (DEC-11-ORTPA-E-PR)
PREEXEC.OBJ  (DEC-11-OREXA-E-PR1)
PREPAS.OBJ  (DEC-11-OREXA-E-PR2)
SMEXEC.OBJ  (DEC-11-ORTAA-E-PR1)
SMMAC.OBJ  (DEC-11-ORTAA-E-PR2)
SMPST.OBJ  (DEC-11-ORTAA-E-PR3)
RTEXEC.OBJ  (DEC-11-ORMAA-E-PR1)
RTMAC.OBJ  (DEC-11-ORMAA-E-PR2)
RTPST.OBJ  (DEC-11-ORMAA-E-PR3)
VTCD1.OBJ  (DEC-11-ORTEA-E-PR2)
VTCD4.OBJ  (DEC-11-ORTEA-E-PR3)
VTEBDT.OBJ  (DEC-11-ORTEA-E-PR4)
LINK0.OBJ  (DEC-11-ORLLA-E-PR1)
LNKOV1.OBJ  (DEC-11-ORLLA-E-PR2)
LNKOV2.OBJ  (DEC-11-ORLLA-E-PR3)
LNKOV3.OBJ  (DEC-11-ORLLA-E-PR4)
LNKOV4.OBJ  (DEC-11-ORLLA-E-PR5)
LNKOV5.OBJ  (DEC-11-ORLLA-E-PR6)
LNK2B.OBJ  (DEC-11-ORLLA-E-PR7)
LIBR0.OBJ  (DEC-11-ORLBA-E-PR1)
LIBR1.OBJ  (DEC-11-ORLBA-E-PR2)
LIBR2.OBJ  (DEC-11-ORLBA-E-PR3)
LIBR3.OBJ  (DEC-11-ORLBA-E-PR4)
LIBR4.OBJ  (DEC-11-ORLBA-E-PR5)
DUMP.OBJ  (DEC-11-ORDMA-E-PR)
SRCCOM.OBJ  (DEC-11-OSRCA-E-PR)
FILEX.OBJ  (DEC-11-ORFLA-E-PR)
PIE.OBJ  (DEC-11-ORPPA-E-PR)
PIPL.OBJ  (DEC-11-ORPPA-E-PR2)
CREF.OBJ  (DEC-11-ORCFA-E-PR)
VTHDLR.OBJ  (DEC-11-OVTHA-E-PR)
RKBTSJ.OBJ  (DEC-11-ORBTA-E-PR4)
RKBTFB.OBJ  (DEC-11-ORBTA-E-PR2)
RFBTSJ.OBJ  (DEC-11-ORBTA-E-PR3)
RFBTFB.OBJ  (DEC-11-ORBTA-E-PR1)
RT11SJ.OBJ  (DEC-11-ORMNA-E-PR2)
RT11FB.OBJ  (DEC-11-ORMNA-E-PR1)
RK.OBJ  (DEC-11-ORKHA-E-PR)
RF.OBJ  (DEC-11-ORFHA-E-PR)
PR.OBJ  (DEC-11-ORPHA-E-PR)
MT.OBJ  (DEC-11-OMTHA-E-PR)
CT.OBJ  (DEC-11-OCAHA-E-PR)
CR.OBJ  (DEC-11-OCHRA-E-PR)
DS.OBJ  (DEC-11-ORJSA-E-PR)
MM.OBJ  (DEC-11-OTUMA-E-PR)
BA.OBJ  (DEC-11-ORBHA-E-PR)
BATCH.OBJ  (DEC-11-ORBCA-E-PR)
SYSF4.OBJ  (DEC-11-ORLIA-E-PR)

Repeat Step 3 for each of the following paper tapes, using /A in place of /B in the command string.

VTCMAC.MAC  (DEC-11-OVTMA-E-PA)
SYSMAC.SML  (DEC-11-ORSYA-E-PA1)
SYSMAC.8K  (DEC-11-ORSYA-E-PA2)

2-52
SYSTEM BUILD INSTRUCTIONS

DEMOFG.MAC (DEC-11-ORDFA-E-PA)
DEMOBG.MAC (DEC-11-ORDSA-E-PA)

Type: CTRL C
Response: C

Type: R LINK<CR>
Response: *

Type: LINK=LIBR0/B1500/C<CR>
Response: *

Type: LNKOV1/011/C<CR>
Response: *

Type: LNKOV2/011/C<CR>
Response: *

Type: LNKV2B/011/C<CR>
Response: *

Type: LNKOV3/011/C<CR>
Response: *

Type: LNKOV4/011/C<CR>
Response: *

Type: LNKOV5/011<CR>

(At this point, many ADDITIVE REF messages will appear on the terminal. These are expected and should be ignored.)

Response: *

Type: CTRL C
Response: C

Type: R LINK<CR>
Response: *

Type: LIBR=LIBR0/C<CR>
Response: *

Type: LIBR1/011/C<CR>
Response: *

Type: LIBR2/011/C<CR>
Response: *

Type: LIBR3/011/C<CR>
Response: *

Type: LIBR4/011<CR>
Response: *

Type: PATCH=PATCH<CR>
Response: *

Type: TT;SYS=TT<CR>
Response: *

2-53
SYSTEM BUILD INSTRUCTIONS

Type: PR, SYS=PR<CR>
Response: *

Type: PP, SYS=PP<CR>
Response: *

If the configuration includes a line printer,

Type: LP, SYS=LP<CR>
Response: *

If the configuration includes a card reader,

Type: CR, SYS=CR<CR>
Response: *

If the configuration includes TM11/TS03 or TM11/TU10 magtape,

Type: MT, SYS=MT<CR>
Response: *

If the configuration includes TM02/TJU16 magtape,

Type: MM, SYS=MM<CR>
Response: *

If the configuration includes RF11 disk,

Type: RF, SYS=RF<CR>
Response: UNDEF GLBLS *

If the configuration includes RJS03 or RJS04 disk,

Type: DS, SYS=DS<CR>
Response: UNDEF GLBLS *

If the configuration includes RK11 disk,

Type: RK, SYS=RK<CR>
Response: UNDEF GLBLS *

If the configuration includes cassette,

Type: CT, SYS=CT<CR>
Response: *

In any case,

Type: PIP=PIPOC<CR>
Response: *

Type: PIP1/011<CR>
Response: *

Type: EXPAND=PES, PEPAS<CR>
Response: *

Type: ASEMBL=SMEXEC, SMMAC, SMPST<CR>
Response: *
TYPE: DUMP=DUMP<CR>
Response: ⋅

TYPE: FILEX=FILEX<CR>
Response: ⋅

TYPE: SRCCOM=SRCCOM<CR>
Response: ⋅

TYPE: EDIT=VTCE01, VTCEX4, TVBEDT, EDIT<CR>
Response: ⋅

If the system includes more than 8K words of memory,

TYPE: MACRO=RTEXEC, RTMAC, RTPST<CR>
Response: ⋅

TYPE: CREF=CREF<CR>
Response: ⋅

TYPE: BATCH=BATCH<CR>
Response: ⋅

TYPE: BA, SYS*BA<CR>
Response: ⋅

If the system being built is on an RK11 disk,

TYPE: MONITR, SYS=RKBTSJ, RT11SJ, RK<CR>
Response: ⋅

Type: RKMNFB, SYS=RKBTFB, RT11FB, RK<CR>
Response: ⋅

If the system being built is on an RF11 disk,

Type: MONITR, SYS=RFBTSJ, RT11SJ, RF<CR>
Response: ⋅

Type: RFMNFB, SYS=RFBTFB, RT11FB, RF<CR>
Response: ⋅

In either case,

Type: CTRL C
Response: ⋅

Type: R OLDPIP<CR>
Response: ⋅

If the new system is on RF11 disk,

Type: RF1A=RF1MONITR, SYS/U<CR>
Response: ⋅

If the new system is on RK11 disk,

Type: RK1A=RK1MONITR, SYS/U<CR>
Response: ⋅
SYSTEM BUILD INSTRUCTIONS

5. If the system has a hardware bootstrap capable of bootstrapping the new system disk, boot the disk; otherwise, perform the bootstrapping procedures in the appropriate section of Appendix A.

RT-11 should bootstrap off the new disk and respond with its identifying message:

```
RT-11SJ V02C-xx
```

If the response is not the system identification message, repeat the entire section.

Type: DATE dd-mmm-yy
Response: 

where dd-mmm-yy is the current date in the form 27-NOV-75.

At this point, the files listed below may be deleted from the disk as they are no longer needed. To delete a disk file named xxxxxx.xxx:

Type: R PIP<CR>
Response: *

Type: xxxxxx.xxx/D<CR>
Response: *

The deletable files are:

```
OLDPIP.SAV     LIBR1.OBJ
EDIT.OBJ       LIBR2.OBJ
TT.OBJ         LIBR3.OBJ
LP.OBJ         LIBR4.OBJ
PP.OBJ         DUMP.OBJ
PREEXEC.OBJ    SRCCOM.OBJ
PREPAS.OBJ     FILEX.OBJ
SMEXEC.OBJ     PIP.OBJ
SMMCAC.OBJ     PIP1.OBJ
SMPSST.OBJ     CREF.OBJ
RTEXEC.OBJ     RKRTSJ.OBJ
RTMAC.OBJ      RRBTFB.OBJ
RTPST.OBJ      RFBTSJ.OBJ
VTCE1.OBJ      RTBTFB.OBJ
VTCE4.OBJ      R11S.J.OBJ
VTBEDT.OBJ     RT11FB.OBJ
LINK0.OBJ      RK.OBJ
LNKOV1.OBJ     RF.OBJ
LNKOV2.OBJ     PR.OBJ
LNKV2B.OBJ     MT.OBJ
LNKOV3.OBJ     CT.OBJ
LNKOV4.OBJ     CR.OBJ
LNKOV5.OBJ     DS.OBJ
LIBR0.OBJ      MM.OBJ
BATCH.OBJ      BA.OBJ
```

When all the unnecessary files have been deleted,

Type: CTRL C
Response: *C
SYSTEM BUILD INSTRUCTIONS

Paper tape users who do not have FORTRAN cannot build the program PATCHO, as PATCHO requires the FORTRAN library to link. Paper tape users who do have FORTRAN systems should proceed to Section 2.7.2 and then link PATCHO as described in Chapter 5.

Paper tape users should take special note of the program DIREXT which is described in Section 4.5.1. It can be used to extend the size of a disk that has already been built.

Proceed to Section 2.7.2 if building a FORTRAN system from paper tape, Section 2.7.3 if building a BASIC system from paper tape, or consult the Table of Contents for the appropriate section if building from another media. If neither FORTRAN nor BASIC will be used on the system, go to Chapter 3.

2.7.2 FORTRAN IV

This section contains instructions for those who received RT-ll FORTRAN on paper tape. The instructions involve the transfer of the FORTRAN-related paper tapes to the system device, then the linking of the FORTRAN files from their components. Final instructions involve creation of the FORTRAN library.

The system device onto which FORTRAN is being built must have at least 370 free blocks. The monitor, TT.SYS, LP.SYS (if the configuration includes a line printer), PIP.SAV, LINK.SAV, and LIBR.SAV are the only system components that must be on this system device.

1. Bootstrap an RT-ll V02C system (if you have completed Section 2.7.1, the system is already booted and running).

   To the running RT-ll monitor,

   Type: DATE dd-mmm-yy<CR>

   where dd-mmm-yy is the current date in the form 27-NOV-75.

   Response: .

   Type: R PIP<CR>

   Response: *

2. Place the paper tape labeled FROOT.OBJ (DEC-ll-LRF4A-C-PRI) in the reader. Press the FEED button until blank leader is over the read head.

   For this, and all the following paper tape instructions in Step 3, the convention xxxxx.xxx is used to represent the name on the tape label below the FORTRAN version number. In this first example, the command below would be FROOT.OBJ=PR:/B<CR>.

   Type: xxxxx.xxx=PR:/B<CR>

   The tape is read.

   Response: *
NOTE

A ?CHK SUM? message displayed during a paper tape input operation means an input error has occurred. Retry the operation if such a message is received.

3. Repeat Step 2 for the paper tapes labeled:

F0.OBJ  (DEC-11-LRF4A-C1-PR2)
F1.OBJ  (DEC-11-LRF4A-C1-PR3)
F2.OBJ  (DEC-11-LRF4A-C1-PR4)
F3.OBJ  (DEC-11-LRF4A-C1-PR5)
F4.OBJ  (DEC-11-LRF4A-C1-PR6)
F5.OBJ  (DEC-11-LRF4A-C1-PR7)
F6.OBJ  (DEC-11-LRF4A-C1-PR8)
F7.OBJ  (DEC-11-LRF4A-C1-PR9)
F8.OBJ  (DEC-11-LRF4A-C1-PR10)
F9.OBJ  (DEC-11-LRF4A-C1-PR11)
F10.OBJ (DEC-11-LRF4A-C1-PR12)
F11.OBJ (DEC-11-LRF4A-C1-PR13)
F12.OBJ (DEC-11-LRF4A-C1-PR14)
F13.OBJ (DEC-11-LRF4A-C1-PR15)
F14.OBJ (DEC-11-LRF4A-C1-PR16)
F15.OBJ (DEC-11-LRF4A-C1-PR17)
F16.OBJ (DEC-11-LRF4A-C1-PR18)
F17.OBJ (DEC-11-LRF4A-C1-PR19)
OTSCOM.OBJ (DEC-11-LRF4A-C1-PR20)
UNI.OBJ  (DEC-11-LRF4A-C1-PR26)
V2S.OBJ  (DEC-11-LRF4A-C1-PR27)
V2NS.OBJ (DEC-11-LRF4A-C1-PR28)

Repeat Step 2 for the following paper tapes, using /A in place of /B in the command string.

FORTRA.HLP  (DEC-11-LRF4A-C1-PA1)
DEMO.FOR  (DEC-11-LRF4A-C1-PA2)

4. If the configuration contains an EAE, repeat Step 2 for the paper tape:

EAE.OBJ  (DE-11-LRF4A-C1-PR22)

If the configuration contains a PDP-11/40 with EIS or a PDP-11/45 without FPU, repeat Step 2 for the paper tape:

EIS.OBJ  (DEC-11-LRF4A-C1-PR23)

If the configuration contains a PDP-11/40 or PDP-11/03 with FIS, repeat Step 2 for the paper tape:

FIS.OBJ  (DEC-11-LRF4A-C1-PR24)

If the configuration contains a PDP-11/45 with FPU, repeat Step 2 for the paper tape:

FPU.OBJ  (DEC-11-LRF4A-C1-PR25)

2-58  July 1976
If the configuration does not contain any of the above options, repeat Step 2 for the paper tape:

NHD.OBJ       (DEC-11-LRF4A-C1-PR21)

5. Type: CTRL C
   Response: *C

   Type: R LINK<CR>
   Response: *

   Type: FORTRAN/NROOT/C<CR>
   Response: *

   Type: F8/O11/C<CR>
   Response: *

   Type: F1/O11/C<CR>
   Response: *

   Type: F2/O11/C<CR>
   Response: *

   Type: F3/O11/C<CR>
   Response: *

   Type: F4/O11/C<CR>
   Response: *

   Type: F5/O11/C<CR>
   Response: *

   Type: F6/O11/C<CR>
   Response: *

   Type: F7/O11/C<CR>
   Response: *

   Type: F8/O11/C<CR>
   Response: *

   Type: F9/O11/C<CR>
   Response: *

   Type: F10/O11/C<CR>
   Response: *

   Type: F11/O11/C<CR>
   Response: *

   Type: F12/O11/C<CR>
   Response: *

   Type: F13/O11/C<CR>
   Response: *

   Type: F14/O11/C<CR>
   Response: *

   Type: F15/O11/C<CR>
   Response: *
SYSTEM BUILD INSTRUCTIONS

Type: \texttt{FI6/011/C<CR>}
Response: *

Type: \texttt{FI7/011<CR>}
Response: ADDITIVE REF OF WRNBAS
AT SEGMENT # 000003
ADDITIVE REF OF WRNBAS
AT SEGMENT # 000004
ADDITIVE REF OF WRNBAS
AT SEGMENT # 000004
ADDITIVE REF OF WRNBAS
AT SEGMENT # 000004
ADDITIVE REF OF WRNBAS
AT SEGMENT # 000004

*

TYPE: \texttt{CTRL C}
Response: ^C

Type: \texttt{R LBR<CR>}
Response: *

If the FORTRAN system will be running on a configuration that includes EAE, proceed. Otherwise, go to Step 6.

Type: \texttt{FORLIB=UNI,OTSCOM,V2NS,EAE/G<CR>}
Response: ENTRY POINT

Type: \texttt{SERRTB<CR> SERRS<CR> <CR>}

Response: SEND2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCL02 ILL INS
SF102 ILL INS
SF102 ILL INS

*

Type: \texttt{FORLIB,V2S=UNI,OTSCOM,V2S,EAE/G<CR>}
Response: ENTRY POINT

Type: \texttt{SERRTB<CR> SERRS<CR> <CR>}

Response: SEND2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCL02 ILL INS
SF102 ILL INS
SF102 ILL INS

*

Go to Step 10.

6. If the configuration includes a PDP-11/45 processor without FPU or a PDP-11/40 processor with EIS but without FIS, proceed. Otherwise, go to Step 7.
7. If the configuration includes a PDP-11/40 or PDP-11/03 processor with FIS, proceed. Otherwise, go to Step 8.

Go to Step 10.
SYSTEM BUILD INSTRUCTIONS

Response: SENC2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCL02 ILL INS
SF102 ILL INS
SF102 ILL INS

* 

Go to Step 10.

8. If the configuration includes a PDP-11/45 processor with FPU, proceed. Otherwise, go to Step 9.

Type: FORLIB=UNI,OTSCOM,V2NS,FPU/G<CR>
Response: ENTRY POINT1

Type: SERRTB<CR>
SERRS<CR>
<CR>

Response: SENC2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCL02 ILL INS
SF102 ILL INS
SF102 ILL INS

* 

Type: FORLIB;V2S=UNI,OTSCOM,V2S,FPU/G<CR>
Response: ENTRY POINT1

Type: SERRTB<CR>
SERRS<CR>
<CR>

Response: SENC2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCL02 ILL INS
SF102 ILL INS
SF102 ILL INS

* 

Go to Step 10.

9. If the configuration contains none of the above options:

Type: FORLIB=UNI,OTSCOM,V2NS,NHD/G<CR>
Response: ENTRY POINT1

Type: SERRTB<CR>
SERRS<CR>
<CR>

Response: SENC2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCL02 ILL INS
SF102 ILL INS
SF102 ILL INS

*
SYSTEM BUILD INSTRUCTIONS

Type: FORLIB,V2S=UN1,OYSCOM,V2S,NHD/G<CR>
Response: ENTRY POINT

Type: SERRTB<CR>
SERRS<CR>
<CR>
Response: SEND2 ILL INS
SEND2 ILL INS
SF102 ILL INS
SCL02 ILL INS
SF102 ILL INS
SF102 ILL INS
.

10. Type: CTRL C
Response: *C

Proceed to Section 2.7.3 if building a BASIC system from paper tape or consult the Table of Contents for the appropriate section if building BASIC from another media. If the system will not include BASIC, go to Chapter 3.

2.7.3 BASIC/RT-11

This section contains instructions for those who received BASIC/RT-11 on paper tape. The instructions involve placing running versions of BASIC on the system device.

The system device onto which BASIC is being built must have at least 70 free blocks. The monitor and PIP.SAV are the only system components that must be on this system device.

1. Bootstrap an RT-11 V02C system (if you have completed Section 2.7.1 or 2.7.2, the system is already booted and running).

To the running RT-11 monitor,

Type: DATE dd-mmm-yy<CR>

where dd-mmm-yy is the current date in the form 27-NOV-75.

Response: *

Type: R PIP<CR>
Response: *

2. Place the paper tape labeled DEMO.BAS (DEC-11-LBACA-D-PAl) into the paper tape reader. Press the FEED button until blank leader is over the tape head.

For this, and all the following paper tape instructions in Step 3, the convention xxxxxx.xxx is used to represent the name on the tape label below the BASIC version number. In this first example, the command below would be DEMO.BAS=PR:/A<CR>.

Type: xxxxxx.xxx=PR:/A<CR>

(the tape is read)
SYSTEM BUILD INSTRUCTIONS

Response: *

NOTE

For the following steps, a "CHK SUM?" message indicates that an input error occurred during the reading of the paper tape. Retry the operation.

3. Repeat Step 2 for the following paper tapes, using /B in place of /A in the command string.

   BASICR.OBJ  (DEC-11-LBACA-D-PR1)
   BASICE.OBJ  (DEC-11-LBACA-D-PR2)
   BASICX.OBJ  (DEC-11-LBACA-D-PR3)
   BASNSR.OBJ  (DEC-11-LBACA-D-PR4)
   BASNSE.OBJ  (DEC-11-LBACA-D-PR5)
   BASNSX.OBJ  (DEC-11-LBACA-D-PR6)
   FPMP.OBJ    (DEC-11-LBACA-D-PR7)
   BASICH.OBJ  (DEC-11-LBACA-D-PR8)

4. Type: CTRL C
Response: "C"

Type: R LINK<CR>
Response: *

Type: BASBK=BASN,FPMP/T/B1400/C<CR>

NOTE

The above command string is for a configuration that has no devices with vectors above 400. Users with 8K whose configurations have interrupt vectors above 400 should relink BASBK.SAV with a bottom address of 500, e.g.,

   BASBK=BASN,FPMP/T/B1500/C<CR>

See BASIC/RT-11 Release Notes, Section 3.6, for further information.

Response: TRANSFER ADDRESS *

Type: GO<CR>
Response: *

Type: BASNSE/011/C<CR>
Response: *

Type: BASNSX/011/C<CR>
Response: *

Type: BASICH/012<CR>
Response: *

Type: BASIC=BASICR,FPMP, BASICE, BASICX/B1400/C<CR>
Response: *
The BASIC system is now built on disk. Proceed to Chapter 3.
CHAPTER 3
DEMONSTRATION PACKAGE

3.1 RUNNING THE RT-11 SINGLE-JOB MONITOR

For purposes of this demonstration, a small program (DEMOBG.MAC) is edited, assembled, linked, and executed. When executed, DEMOBG displays a message on the terminal.

1. If the system is not running, bootstrap an RT-11 V02C system. Ensure that the system device is WRITE-ENABLEd.

   Response: RT-11 V02C-xx
   .

   Type: DATE dd-mmm-yy

   where dd-mmm-yy is the current date in the form 27-NOV-75.

   Response: .

2. If the console terminal is an ASR or KSR Teletype, a parallel LA30 DECwriter, a VT50 Alphanumeric Display, or an LA36 DECwriter II, go to Step 3.

   If the console terminal is a VT05 Alphanumeric Display operating at 300 baud, go to Step 3.

   If the console terminal is a serial LA30 DECwriter operating at 300 baud,

   Type: D 56*5015<CR>
   Response: .

   Go to Step 3.

   If the console terminal is a serial LA30 DECwriter operating at 150 baud,

   Type: D 56*2015<CR>
   Response: .

   Go to Step 3.

   If the console terminal is a serial LA30 DECwriter operating at 110 baud,

   Type: D 56*1015<CR>
   Response: .

   Go to Step 3.
If the console terminal is a VT05 Alphanumeric Display operating at 2400 baud,

Type: D 56*2012<CR>
Response: 

Go to Step 3.

If the console terminal is a VT05 Alphanumeric Display operating at 1200 baud,

Type: D 56*1012<CR>
Response: 

Go to Step 3.

If the console terminal is a VT05 Alphanumeric Display operating at 600 baud,

Type: D 56*412<CR>
Response: 

3. If the system configuration does not include a GT40 or VT11 display processor and scope or does not have more than 8K words of memory, go to Step 4. Otherwise, verify that the scope is on by turning the BRIGHTNESS knob to an adequate level.

Build the VT11 support library as follows:

Type: R LIBR<CR>
Response: *

Type: VTLIB=SY1VTWDLR<CR>
Response: *

Type: CTRL C
Response: *C

Type: CT ON<CR>

The system output will shift to the display scope. Commands are still entered at the keyboard, but echo will be on the screen.

Response: .(on screen)

4. If the system device is diskette, go to Step 5. If the system device is RFl1, RK11, RP02, or RJS03/4 disk, go to Step 6.

If the system device is DECtape, mount a blank tape on Unit 1, WRITE ENABLEd.

Type: ASSIGN DT1:1DK<CR>
Response: 

Type: R PIP<CR>
Response: *
5. If the system device is diskette, mount a blank diskette on unit 1, WRITE ENABLEd.

Type: \texttt{ASSIGN DX11D<CR>}
Response: *

Type: \texttt{R PIP<CR>}
Response: *

Type: \texttt{DX11/E<CR>}
Response: \texttt{DX11/E ARE YOU SURE?}

Type: \texttt{Y<CR>}
Response: *

Type: \texttt{CTRL C}
Response: *

6. Display the directory of the system device on the terminal.

Type: \texttt{R PIP<CR>}
Response: *

Type: \texttt{SY1/F<CR>}
Response: \texttt{MONITOR,SYS}
\texttt{DT,SYS}
\texttt{LP,SYS}
\texttt{PP,SYS}
\texttt{PR,SYS}
\texttt{TT,SYS}
\texttt{PATCH,SAV}
\texttt{EDIT,SAV}
\texttt{MACRO,SAV}
\texttt{EXPAND,SAV}
\texttt{ASEMBL,SAV}
\texttt{SYMSAV,SML}
\texttt{LINK,SAV}
\texttt{ODT,OBJ}
\texttt{PIP,SAV}
* 

\textbf{NOTE}

Depending on the medium from which RT-11 is built and the individual system configuration, the preceding directory will vary. As long as a directory is printed, its exact content is not of concern.
7. Use the Text Editor to modify the demonstration program, DEMOGB.MAC.

Type: CTRL C
Response: *C

Type: R EDIT<CR>
Response: *

Type: EBSY|DEMOB|MAC<ALT>R<ALT><ALT>
Response: *

Type: G1<TAB>,ASCII<ALT><ALT>
Response: *

Type: GAD<ALT><ALT>
Response: *

Type: EX<ALT><ALT>
Response: *

Assemble DEMOGB.MAC with EXPAND and ASEMBL and obtain a listing.

Type: R EXPAND<CR>
Response: *

Type: DEMOGB=SY|DEMOB|G<CR>
Response: ERRORS DETECTED: 0

Type: CTRL C
Response: *C

Type: R ASEMBl<CR>
Response: *

If a line printer is available, go to Step 9.

8. If running on an 8K DECtape-, RFile-, or Diskette-based system,

Type: DEMOGB=DEMOBG<CR>
Response: ERRORS DETECTED: 0
FREE CORE: xxx WORDS

Type: ,TTI=DEMOBG<CR>
Response: (see following)

All other systems,

Type: DEMOBG,TTI=DEMOBG<CR>
Response: (see following)
DEMONSTRATION PACKAGE

MAIN. RT-11 MACRO VS02-09 PAGE 1

1 : RT-11 MACRO EXPAND VS02-02
24 DEMOBG MAC
25 DEMONSTRATION PROGRAM TO PRINT DEMONSTRATION MESSAGE, THEN
26 RING BELL IF FG JOE SENDS A MESSAGE.
27
28 .MCALL .V2...REGDEF, RCVDC, PRINT
29 .MACRO .V2...
30 . V2=1
31 .ENDM
32 .MACRO .PRINT .ADD
33 .IF NB .ADD
34 MOV .ADD,%0
35 .ENDDC
36 .ENDDM
37 .MACRO .RCVDC .AREA, .BUFF, .WCNT, .CRTN
38 .IF NB .AREA
39 MOV .AREA,%0
40 MOV %13000,(0)
41 .ENDC
42 .IF NB .BUFF
43 MOV .BUFF,(0)
44 .ENDC
45 .IF NB .WCNT
46 MOV .WCNT, (0)
47 .ENDC
48 .IF NB .CRTN
49 MOV .CRTN, (0)
50 .ENDC
51 .ENDDM
52 .MACRO .REGDEF
53 .R0=%0
54 .R1=%1
55 .R2=%2
56 .R3=%3
57 .R4=%4
58 .R5=%5
59 .SF=%6
60 .PC=%7
61 .ENDM
62
63 .REGDEF
64 00000 R0=%0
65 00001 R1=%1
66 00002 R2=%2
67 00003 R3=%3
68 00004 R4=%4
69 00005 R5=%5
70 00006 SF=%6
71 00007 PC=%7
72
73 00000 START: RCVDC #AREA, #BUFFER, #400, #MSGIN, #POST REQUEST FOR MESS
74 IF NB #AREA
75 MOV #AREA,%0
76
77 00000 012700 .MOV #AREA,%0
78 00004 012710 .MOV #13000,(0)
79
80 013000

013000
ENDC
. IF NB #BUFFER
81 00810 012760
008236' 000004
MOV #BUFFER.4.(0)
82
83
84 00816 012760
000040
000000
ENDC
85
86 00824 012760
000004
000010
IF NB #MSGIN
MOV #MSGIN.8.(0)
87
88
89 00832 104375
PRINT #MSG
PRINT DEMONSTRATION MESSAGE
EMT "0375
90
91 00834 012700
000112
IF NB #MSG
MOV #MSG.%0
92
93
94 00840 104351
BR EMT "0351
95
96
97
98 00844 012700
000110
COMPLETION ROUTINE ENTERED WHEN FG SEBS MESSAGE
00810
99 00844 012700
000110
IF NB #BELL
MOV #BELL.%0
100
101
102 00850 104351
EMT "0351
103
104
105 00852 012700
000226
IF NB #AREA
MOV #AREA.%0
106
107
108
109
107 00856 012710
001000
MOV #13000.(0)
106
107
108
109
110
111
112
111
112
113
114
115
116
117
118
119
120 00862 012760
000236
000004
IF NB #BUFFER
MOV #BUFFER.4.(0)
118
120
121 0110 007 BELL:
BYTE 7,200 MESSAGE THAT RINGS BELL
0111 200
122
123
124 MSG:
ASCII /RT-11 DEMONSTRATION PROGRAM/
0112 122
DEMONSTRATION PACKAGE

0114 055
0115 061
0116 061
0117 040
0120 104
0121 105
0122 115
0123 117
0124 116
0125 123
0126 124
0127 122
0130 101
0131 124
0132 111
0133 117
0134 116
0135 040
0136 120
0137 122
0140 117
0141 107
0142 122
0143 101
0144 115
124 0145 015
125 0146 012
125 0147 111

.ASCII /* IF INCORRECTLY EDITED, THIS IS THE LAST LINE. */
0150 106
0151 040
0152 111
0153 116
0154 102
0155 117
0156 122
0157 122
0160 105
0161 103
0162 124
0163 114
0164 131
0165 040
0166 105
0167 104
0170 111
0171 124
0172 105
0173 104
0174 054
0175 124

.MAIN RT-11 MACRO V502-09 PAGE 1+

0176 110
0177 111
0200 123
0201 040
0202 111
0203 123
0204 040
0205 124
0206 110
0207 105
0210 040
0211 114
0212 101
DEMONSTRATION PACKAGE

0213 123
0214 124
0215 040
0216 114
0217 111
0220 116
0221 105
0222 056
0223 015
0224 012
0225 000
0226 000236
127 ASCII /WELL DONE./
128 .BYTE 15, 12
129 .BYTE 0
130 .RCVDC PACKET AREA
131 AREA: = +10
132 .RCVDC MESSAGE AREA
133 BUFFER:
134 MAIN RT-11 MACRO VS02-09 PAGE 1+
135 SYMBOL TABLE

AREA 000226R BELL 000110R BUFFER 000236R
MSG 000112R MSGIN 000044R FC 00000007
R0 =%000000 R1 =%000001 R2 =%000002
R3 =%000003 R4 =%000004 R5 =%000005
SP =%000006 START 000000R

ERRORS DETECTED: 0
FREE CORE: 20488. WORDS

DEMOBG.TT:=DEMOBG

ERRORS DETECTED: 0
FREE CORE: 20488. WORDS

Response: *

Go to Step 10.

9. Ensure that the line printer is on and set to on-line. If running on an 8K DECTape-, RFll-, or Diskette-based system,

Type: DEMOBG*DEMOBG<CR>
Response: ERRORS DETECTED = 0
FREE CORE: xxx WORDS
*

Type: ;LPI*DEMOBG<CR>
Response: *(The listing on the printer is similar to the response for Step 8, except that the FREE CORE message appears only once.)

All other systems,

Type: DEMOBG;LPI*DEMOBG<CR>
Response: ERRORS DETECTED: 0
FREE CORE: xxx WORDS
*(The listing on the printer is similar to the response for Step 8, except that the FREE CORE message appears only once.)
10. Type: CTRL C
    Response: *C

Link the program DEMOBG.

Type: R LINK<CR>
Response: *

Type: SY:DEMOBG*DEMOBG<CR>
Response: *

Type: CTRL C
response: *C

Execute the demonstration program.

Type: R DEMOBG<CR>
Response: RT-11 DEMONSTRATION PROGRAM,
             IF INCORRECTLY EDITED, THIS IS THE LAST LINE,
             WELL DONE.

Type: CTRL C
CTRL C
Response: *C
          *C
          *

Type: CT OFF<CR>
Response: *

If the file was incorrectly edited, the procedure may be repeated, although this is not necessary for successful continuation. If repeating the editing procedure, first,

Type: R PIP<CR>
Response: *

Type: DEMOBG,MAC=DEMOBG,BAK/R<CR>
Response: *

Type: CTRL C
Response: *C

Then go to Step 7.

Otherwise, if the system will be running the Foreground/Background Monitor, proceed to Section 3.2. If the system will be running FORTRAN IV but will not be running F/B, proceed to Section 3.3. If the system will be running BASIC but will not be running F/B or FORTRAN IV, proceed to Section 3.4. Otherwise the demonstration is complete.

Before continuing to use the system, make all patches and corrections documented in the Digital Software News and Software Performance Summary, and note the restrictions documented in the RT-11 System Release Notes manual. In addition, you may wish to permanently customize the system for special hardware; instructions for customizing the system are in Section 4.6.
3.2 RUNNING THE RT-11 FOREGROUND/BACKGROUND MONITOR

For the purposes of this demonstration, a second program (DEMOFG.MAC) is assembled, linked for the foreground, and executed in conjunction with DEMOBG. This portion of the demonstration requires 16K words of memory and a clock to run. DEMOFG is a small foreground program that sends a message every two seconds to DEMOBG, running in the background, telling it to ring the terminal bell. Besides printing the terminal message used in Section 3.1, DEMOBG recognizes these messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every two seconds, this demonstration will execute other programs in the background besides DEMOBG. Only when DEMOBG is active, however, is the circuit complete and messages successfully received and honored. During those periods when DEMOBG is not running, DEMOFG will enter the messages in the message queue; when DEMOBG is started, all the messages queued since the last forced exit will be dequeued immediately, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell is rung every two seconds, as each current message is sent and honored.

1. To the running RT-11 Single-Job Monitor,

   Type: \texttt{R PIP<CR>}
   
   Response: \texttt{*}

   If running an RK11 system,
   
   Type: \texttt{SYIRKMNSJ;SYS=SYMONITR;SYS/Y/R<CR>}
   
   If running an RF11 system,
   
   Type: \texttt{SYIRFMNSJ;SYS=SYMONITR;SYS/Y/R<CR>}
   
   If running a DECTape system,
   
   Type: \texttt{SYIRDTMNSJ;SYS=SYMONITR;SYS/Y/R<CR>}
   
   If running an RJS03/4 system,
   
   Type: \texttt{SYIRDMSNSJ;SYS=SYMONITR;SYS/Y/R<CR>}
   
   If running an RP11/RP02 system,
   
   Type: \texttt{SYIRDPMNSJ;SYS=SYMONITR;SYS/Y/R<CR>}
   
   If running a diskette system,
   
   Type: \texttt{SYIRDXMNSJ;SYS=SYMONITR;SYS/Y/R<CR>}

   In any case, the response is:
   
   Response: \texttt{?REBOOT?}
   
   Type: \texttt{SYIMONITR;SYS=SYIRKMNB;SYS/Y/R<CR>}
   
   If running an RK11 system,
   
   Type: \texttt{SYIMONITR;SYS=SYIRFMNB;SYS/Y/R<CR>}
   
   If running an RF11 system,
If running an RPl1/RP02 system,
Type:  SYMONITR;SYS*SYIDPMNFB;SYS/Y/R<CR>
If running a diskette system,
Type:  SYMONITR;SYS*SYIDXMNFB;SYS/Y/R<CR>
If running an RJS03/4 system,
Type:  SYMONITR;SYS*SYIDSMNFB;SYS/Y/R<CR>
If running a DECTape system,
Type:  SYMONITR;SYS*SYIDTMNFB;SYS/Y/R<CR>
In any case, the response is:
Response:  ?REBOOT?
*  
Type:  SYIA*SYMONITR;SYS/U<CR>
Response:  *
Type:  SYS/O<CR>
Response:  RT-11FB V02C-xx
.

The F/B Monitor is now running.

If the console terminal is an ASR or KSR Teletype, a VT50 Alphanumeric Display, an LA36 DECwriter II, or a parallel LA30 DECwriter, go to Step 2.

If the console terminal is a VT05 Alphanumeric Display operating at 300 baud, go to Step 2.

If the console terminal is a serial LA30 DECwriter operating at 300 baud,
Type:  D 56*5015<CR>
Response:  ;
Go to Step 2.

If the console terminal is a serial LA30 DECwriter operating at 150 baud,
Type:  D 56*2015<CR>
Response:  ;
Go to Step 2.

If the console terminal is a serial LA30 DECwriter operating at 110 baud,
Type:  D 56*1015<CR>
Response:  ;
Go to Step 2.

3-11
If the console terminal is a VT05 Alphanumeric Display operating at 2400 baud,

Type: \texttt{D 56#2012<CR>}
Response: 

Go to Step 2.

If the console terminal is a VT05 Alphanumeric Display operating at 1200 baud,

Type: \texttt{D 56#1012<CR>}
Response: 

Go to Step 2.

If the console terminal is a VT05 Alphanumeric Display operating at 600 baud,

Type: \texttt{D 56#412<CR>}
Response: 

2. If the system configuration does not include a GT40 or VT11 display processor and scope, go to Step 3. Otherwise, verify that the scope is on by turning the \textbf{BRIGHTNESS} knob to an adequate level.

Type: \texttt{G7 ON<CR>}

The system output will shift to the display scope. Commands are still entered at the keyboard, but echo will be on the screen.

Response: \texttt{(on screen)}

3. If the system device is diskette, go to Step 4. If the system device is RF11, RK11, RJS03/4, or RP02, go to Step 5.

If the system device is DECTape, mount a blank, formatted DECTape on Unit 1, WRITE ENABLEd.

Type: \texttt{ASSIGN DT11/2<CR>}
Response: 

Type: \texttt{R PIP<CR>}
Response: *

Type: \texttt{DT11/2<CR>}
Response: \texttt{DT11/2 ARE YOU SURE?}

Type: \texttt{Y<CR>}
Response: *

Type: \texttt{CTRL C}
Response: \texttt{C}

Go to Step 5.

4. If the system device is diskette, mount a blank diskette on Unit 1, WRITE ENABLEd.

Type: \texttt{ASSIGN DX11/2<CR>}
Response: 

3-12
DEMONSTRATION PACKAGE

Type: R PIP<CR>
Response: *

Type: DX11/2<CR>
Response: DX11/2 ARE YOU SURE?

Type: Y<CR>
Response: *

Type: CTRL C
Response: *C

5. Enter the date on which this demonstration is being run.

Type: DATE dd-mmm-yy<CR>

where dd-mmm-yy is the current date in the form 27-NOV-75.

Response: .

Enter the time of day.

Type: TIME hh:mm:ss<CR>

where hh:mm:ss is the current hour, minutes, and seconds in the form 13:12:50 (1:12 p.m.).

6. Assemble the foreground demonstration program, DEMOFG.MAC.

Type: R MACRO<CR>
Response: *

Type: DEMOFG=SYIDEMOFG<CR>
Response: ERRORS DETECTED: 0
FREE CORE xxxxx. WORDS *

Type: CTRL C
Response: *C

7. Link DEMOFG for the foreground.

Type: R LINK<CR>
Response: *

Type: SYIDEMOFG=DEMOFG/R<CR>
Response: *

Type: CTRL C
Response: *C

8. Start DEMOFG as the foreground job.

Type: FRUN SYIDEMOFG<CR>
Response: F> FOREGROUND DEMONSTRATION PROGRAM, SENDS A MESSAGE TO THE BACKGROUND PROGRAM "DEMOBG" EVERY 2 SECONDS, TELLING IT TO RING THE BELL.
B>
DEMOFG is now running and queueing the messages for DEMOBG every two seconds.

9. Execute DEMOBG and receive the messages.

Type: \texttt{R DEMOBG<CR>}
(The bell will ring quickly several times, then will ring once every two seconds.)

Response: \texttt{RT-11 DEMONSTRATION PROGRAM}
\texttt{IF INCORRECTLY EDITED, THIS IS THE LAST LINE, WELL DONE.}

10. Execute PIP in the background to get a directory listing.

Type: \texttt{CTRL C}
CTRL C (the bell will stop)

Response: \texttt{•C}
\texttt{•C}

Type: \texttt{R PIP<CR>}
Response: \texttt{•}

Type: \texttt{/L<CR>}
Response: \texttt{dd-mmm-yy}
(The directory of the device DK: is printed on the terminal. Its exact contents are not of consequence.)

Type: \texttt{CTRL C}
Response: \texttt{•C}

11. Rerun DEMOBG to collect all the foreground messages queued while PIP was running.

Type: \texttt{R DEMOBG<CR>}
(The bell will ring several times in rapid succession, then will begin ringing once every two seconds.)

Response: \texttt{RT-11 DEMONSTRATION PROGRAM}
\texttt{IF INCORRECTLY EDITED, THIS IS THE LAST LINE, WELL DONE.}

Type: \texttt{CTRL C}
CTRL C
Response: \texttt{•C}
\texttt{•C}

(The bell will stop ringing.)

12. Stop the foreground program and remove it from memory.

Type: \texttt{CTRL/F}
Response: \texttt{F>}

Type: \texttt{CTRL C}
CTRL C
Response: \texttt{•C}
\texttt{•C}
\texttt{B>}

3-14
DEMONSTRATION PACKAGE

Type: UNL PG<CR>
Response: 

Type: GT OFF<CR>
Response: 

Proceed to Section 3.3 if running FORTRAN IV, or to Section
3.4 if running BASIC; otherwise, the demonstration is
complete.

Before continuing to use the system, make all patches and
corrections documented in the Digital Software News and
Software Performance Summary, and note the restrictions
documented in the RT-II System Release Notes. In addition,
you may wish to permanently customize the system for special
hardware; instructions for customizing the system are in
Section 4.6.

3.3 RUNNING FORTRAN IV

This section contains instructions for compiling the sample program
(DESCFOR), linking it, and executing it. The program is a simple
FORTRAN program to calculate a Fibonacci Series (each term is the sum
of the preceding two terms) based on user inputs.

1. Bootstrap the FORTRAN system device created in Chapter 2;
   enter the date this demonstration is being run.

   Type: DATE dd-mmm-yy<CR>

   where dd-mmm-yy is the current date in the form 27-NOV-75.

   Response: 

   If the configuration does not include a line printer, proceed
to Step 2. Otherwise,

   Type: ASS LP1<CR>
   Response: 

2. If the system device is diskette or disk, go to Step 3.

   If the system device is DECTape, mount a blank tape on Unit
   1, WRITE ENABLEd.

   Type: ASSIGN DT11<CR>
   Response: 

   Type: R PIP<CR>
   Response: *

   Type: DT11/Z<CR>
   Response: DT11/Z ARE YOU SURE?

   Type: Y<CR>
   Response: 

   Type: CTRL C
   Response: 

3-15
3. Compile the FORTRAN program DEMO.FOR as follows:

   Type:     R FORTRAN<CR>
   Response: *

   Type:     DEMO,TT1#DEMO<CR>

If the configuration includes a line printer, the source listing is printed on the printer. If not, the listing appears on the console terminal.

Response:

```
0001   WRITE (5,1000)
0002   WRITE (5,1010)
0003   READ (5,1060) LENGTH
0004   IF (LENGTH)=150 125,150 1A LENGTH OF ZERO MEANS FINISHED
0005   STOP
0006   WRITE (5,1020)
0007   READ (5,1060) ITERM1
0008   WRITE (5,1030)
0009   READ (5,1060) ITERM2
0010   IF (LENGTH)=-3 200,250;250
0011   WRITE (5,1040) LENGTH
0012   QOTO 100
0013   IF (LENGTH)=90 300,300,200
0014   WRITE (5,1050)
0015   WRITE (5,1060) ITERM1
0016   WRITE (5,1060) ITERM2
0017   LENGTH = LENGTH +2
0018   ITNEW = ITERM1 * ITERM2
0019   ITERM1 = ITERM2
0020   ITERM2 = ITNEW
0021   WRITE (5,1060) ITNEW
0022   LENGTH = LENGTH -1
0023   IF (LENGTH) 100,100,400
0024   FORMAT ('0PROGRAM TO GENERATE A FIBONACCI SERIES')
0025   FORMAT ('0HOW MANY TERMS DO YOU WANT GENERATED? ')
0026   FORMAT ('0WHAT IS THE FIRST TERM? ')
0027   FORMAT ('0WHAT IS THE SECOND TERM? ')
0028   FORMAT ('15,' TERMS DOES NOT REALLY MAKE SENSE,')
0029   FORMAT ('0THE REQUESTED SERIES IS')
0030   FORMAT (110)
END
```
FORTRAN IV STORAGE MAP

NAME OFFSET ATTRIBUTES

LENGTH 000334 INTEGER*2 VARIABLE
ITEM1 000336 INTEGER*2 VARIABLE
ITEM2 000340 INTEGER*2 VARIABLE
ITNEW 000342 INTEGER*2 VARIABLE

Type: CTRL C
Response: "C"

Type: ASSIGN<CR>
Response: ""

4. Link the program with the FORTRAN library, FORLIB.OBJ, as follows:

Type: R LINK<CR>
Response: ""

Type: DEMO=DEMO/F<CR>
Response: ""

Type: CTRL C
Response: "C"

5. Execute the program.

Type: R DEMO<CR>
Response: PROGRAM TO GENERATE A FIBONACCI SERIES
HOW MANY TERMS DO YOU WANT GENERATED?

Type: 10<CR>
Response: WHAT IS THE FIRST TERM?

Type: 2<CR>
Response: WHAT IS THE SECOND TERM?

Type: 4<CR>
Response: THE REQUESTED SERIES IS:
2
4
6
10
16
26
42
68
110
178

HOW MANY TERMS DO YOU WANT GENERATED?

Type: 10<CR>
Response: WHAT IS THE FIRST TERM?

Type: 17<CR>
Response: WHAT IS THE SECOND TERM?

Type: =3<CR>
Response: THE REQUESTED SERIES IS:
17
-3
44
11
25
36
61
97
150
255

Response: HOW MANY TERMS DO YOU WANT GENERATED?
Type: 0<CR>
Response: STOP --

Proceed to Section 3.4 if running BASIC; otherwise the demonstration is complete.

Before continuing to use the system, make all patches and corrections documented in the Digital Software News and Software Performance Summary, and note the restrictions documented in the RT-ll System Release Notes. In addition, you may wish to permanently customize the system for special hardware; instructions for customizing the system are in Section 4.6.

Before using FORTRAN, carefully read the RT-ll FORTRAN Release Notes manual and note the restrictions and clarifications for RT-ll FORTRAN V01C.

3.4 RUNNING BASIC/RT-ll

The following instructions are for running the simple BASIC program, DEMO.BAS, which is the same program as DEMO.FOR, written in BASIC. It calculates a Fibonacci series (each term is the sum of the previous two) based on user inputs.

1. To the running RT-ll monitor,
   Type: R BAS8K<CR>
   Response: BASIC V01B-81
   Type: <CR>
   Response: READY

2. Load the demonstration program into memory.
   Type: OLD "DEMO,BAS"<CR>
   Response: READY

3. List the program.
   Type: LIST<CR>
   Response:
DEMO 27 NOV 75  BASIC V01B-01

10 REM BASIC PROGRAM TO GENERATE N TERMS OF A FIBONACCI SERIES,
20 REM THE FIRST TWO TERMS OF WHICH ARE SPECIFIED BY THE USER,
30 REM
40 REM PRINT THE IDENTIFYING MESSAGE
50 PRINT "PROGRAM TO GENERATE A FIBONACCI SERIES"
60 REM
70 REM GET THE LENGTH AND FIRST TWO TERMS OF THE SERIES
80 PRINT "HOW MANY TERMS DO YOU WANT GENERATED?"
90 INPUT L
100 IF L<0 THEN 130
110 REM IF USER REQUESTS 0 TERMS, TERMINATE EXECUTION
120 STOP
130 PRINT "WHAT IS THE FIRST TERM?"
140 INPUT T1
150 PRINT "WHAT IS THE SECOND TERM?"
160 INPUT T2
170 REM MAKE SURE L IS NOT NEGATIVE OR TOO LARGE
180 IF L<1 THEN 200
190 IF L<100 THEN 220
200 PRINT "THE TERMS DO NOT REALLY MAKE SENSE."
210 GO TO 80
220 REM PRINT THE FIRST TWO TERMS OF THE SERIES
230 PRINT "THE REQUESTED SERIES IS"
240 PRINT T1
250 PRINT T2
260 L=L-2
270 REM CALCULATE NEXT TERM AND PRINT IT
280 N=T1+T2
290 T1=T2
300 T2=N
310 PRINT N
320 REM DETERMINE IF SERIES IS FINISHED. IF SO, DO NEXT ONE.
330 L=L-1
340 IF L<0 THEN 80
350 GO TO 200
360 END

READY

4. Execute the program.

Type:  RUN<CR>
Response: DEMO dd-mm-yy BASIC V01B-01
          PROGRAM TO GENERATE A FIBONACCI SERIES
          HOW MANY TERMS DO YOU WANT GENERATED?

Type:  10<CR>
Response: WHAT IS THE FIRST TERM?

Type:  2<CR>
Response: WHAT IS THE SECOND TERM?

Type:  4<CR>
Response: \textbf{THE REQUESTED SERIES IS}: \begin{align*}
2 \\
4 \\
6 \\
10 \\
16 \\
26 \\
42 \\
68 \\
110 \\
178 \\
\end{align*}
\textbf{HOW MANY TERMS DO YOU WANT GENERATED?}

Type: \texttt{CTRL C} \\
Response: \texttt{C} \\

Type: \texttt{RE<CR>} \\
Response: \texttt{READY}

5. BASIC/RT-11 may be run in immediate mode, described in Chapter 4 of the \textit{BASIC/RT-11 Language Reference Manual}. Statements without line numbers are executed as soon as they are entered.

Type: \texttt{FOR I=1 TO 5;PRINT I;SQR(I);NEXT I<CR>} \\
Response: \begin{align*}
1 & 1 \\
2 & 1.41421 \\
3 & 1.73205 \\
4 & 2 \\
5 & 2.23607 \\
\end{align*}

Type: \texttt{CTRL C} \\
Response: \texttt{C} \\

The demonstration is complete.

Before continuing to use the system, make all patches and corrections documented in the \textit{Digital Software News} and \textit{Software Performance Summary}, and note the restrictions documented in the \textit{RT-11 System Release Notes}. In addition, you may wish to permanently customize the system for special hardware; instructions for customizing the system are in Section 4.6.

Before using BASIC, carefully read the \textit{BASIC/RT-11 Release Notes} manual and note the restrictions and clarifications for \textit{BASIC/RT-11 V01B}. 
CHAPTER 4
RT-11 SYSTEM CUSTOMIZATION

4.1 GENERAL BUILDING INSTRUCTIONS

RT-11 is designed so that the monitor and device handlers that comprise the system are files on the system device. These files (called system files) all have the extension .SYS and can be transferred between devices just like any other RT-11 file.

The running version of the monitor must be named MONITR.SYS; other versions of the monitor may reside on the system device, but they must be named something other than MONITR.SYS. The handlers for the system must be named xx.SYS, where xx is the device mnemonic as used in command strings. For example, the high-speed reader handler must be named PR.SYS. There may be many versions of a given handler on the system device, but the one that is in use must be named as above.

Once copies of the system files have been obtained, the procedure for building an RT-11 system consists of the following basic steps:

1. Initializing the target device with an RT-11 directory.
2. Transferring the appropriate monitor file to the target device and giving it the name MONITR.SYS.
3. Transferring the appropriate handler files to the target device.
4. Writing the appropriate bootstrap on the target device.
5. Transferring the desired system components (EDIT, LINK, etc.) to the target device.
6. Check the target device with the PIP /K switch to read all the blocks and verify that all the data written is good.

After step 4 above, the target device may be bootstrapped and the remainder of the build procedure may be carried out while executing the system from the new (and perhaps faster) system device. Because the above build steps involve standard RT-11 file operations, system programs are used for the build procedure. When building from DECTape or disk, PIP is used; from magtape, MBUILD is used; from cassette, CBUILD is used; and from paper tape, LINK and PIP are used. RT-11 V02C can be bootstrapped WRITE-PROTECTed and will run PIP WRITE-PROTECTed as well. System building should always be carried out with the master WRITE-PROTECTed.
RT-11 SYSTEM CUSTOMIZATION

Once the system is built, any necessary patches to the system should be installed as noted in the Digital Software News, Software Performance Summary, and RT-11 System Release Notes. Then the system should be customized for specific hardware, if any, as detailed in Section 4.6. When all patches are made and the system customized, the new system device should be backed-up immediately.

The following files on the distribution media have special significance:

MONITR.SYS
On a DECTape master this is a Single-Job DECTape Monitor; on a DECPack or TAll cassette master, this is a Single-Job RK11 Disk Monitor; on diskette, this is the Single-Job RX11 Monitor. When the volume is booted, this file is the monitor used. (Not available on magtape or paper tape.)

DTMNSJ.SYS
This is a Single-Job DECTape Monitor, used for building DECTape-based Single-Job systems (not available on cassette, diskette, or paper tape).

DTMNFBS.YS
This is a Foreground/Background DECTape Monitor and is used when it is desired to run the F/B system from DECTape (not available on cassette, diskette, or paper tape).

RKMNSJ.SYS
This a Single-Job RK11 Monitor. This file is called MONITR.SYS on RK11 and cassette masters.

RKMNFBS.YS
This is a Foreground/Background RK11 Monitor. If an RK11 F/B system is desired, this file becomes MONITR.SYS on the disk.

RFMNSJ.SYS
This is a Single-Job RF11 Monitor (not available on cassette).

RFMNFB.SYS
This is a Foreground/Background RF11 Monitor (not available on cassette).

DXMNSJ.SYS
This is a Single-Job RX11 Monitor (not available on cassette or paper tape).

DXMNFB.SYS
This is a Foreground/Background RX11 Monitor (not available on cassette or paper tape).

DPMNSJ.SYS
This is a Single-Job RP11/RP02 Monitor (not available on cassette or paper tape).

DPMNFBS.YS
This is a Foreground/Background RP11/RP02 Monitor (not available on cassette or paper tape).

DSMNSJ.SYS
This is a Single-Job RJS03/4 Monitor (not available on cassette, DECTape, diskette, or paper tape).

DSMNFB.SYS
This is a Foreground/Background RJS03/4 Monitor (not available on cassette, DECTape, diskette, or paper tape).
This is the RX11/RX01 diskette handler, which allows RT-11 systems running from a disk other than the RX to access a diskette (not available on cassette or paper tape).

This is the RPI1/RP02 disk handler; it allows systems based on another device to access the RP02 (not available on cassette or paper tape).

This is an RP11/RS11 device handler; it allows RT-11 systems running from other devices to access the RP11 disk (not available on cassette).

This is an RK11 device handler which allows RT-11 to read and write RK11 disks while running from an alternate system device.

This is a TC11 DECTape handler which allows RT-11 to read and write DECTape while running disk systems (not available on cassette or paper tape).

This is the RJS03/4 fixed-head disk handler.

This is the TM11/TS03 or TM11/TU10 magtape handler.

This is the TJU16 magtape handler.

This is the TA11 cassette handler.

This is the PC11 high-speed reader handler.

This is the PC11 high-speed punch handler.

This is the general terminal handler for the S/J Monitor. The F/B terminal handler is included in the MONITR.SYS file.

This is the line printer (LP11, LS11, LV11) handler.

This is the CR11 card reader handler.

This is the BATCH run-time handler.

4.2 BUILDING RT-11 SYSTEMS FROM DECTAPE, DECPACK, AND DISKETTE

On DECTape, DECPack, and diskette, RT-11 is distributed as a ready-to-run Single-Job system. When bootstrapped (as described in the RT-11 System Reference Manual, Chapter 2), the system is running and may be used to build other DECTape and disk systems.

To build another DECTape or disk system from a running DECTape or disk system, the following set of commands to PIP can be used. These commands assume that the device to build has been assigned the logical device name DK1:. (For example, the command ASSIGN DT0:DK1 assigns the logical name DK1 to DECTape unit 0.)
Command | Explanation
--- | ---
`R PIP<CR>`  
`DK1/Z<CR>`  
`DK1/I ARE YOU SURE?Y<CR>`  
`DK1/A*DK01/S<CR>`  
`DK1/A=DK1/MONITR;SYS/U<CR>`  
`DK1/K<CR>` | Initialize the new DECTape or disk.

Copy all files from DK0 to DK1.
Write the hardware bootstrap on DK1.
Check for bad blocks.

DK1 is now a ready-to-run copy of the system device.

In the preceding example, the system files were copied to DK1: via the /S option in PIP; the /S option makes the directory on DK1 exactly the same as the directory on DK0: In actuality, any method of copying the files to the new device would have sufficed. For example, the command:

`DK1/A=DK01/S`

could be replaced by:

`DK1/Z/N112`  
`DK1/A=DK01/*Y/X`  

or

`DK1/MONITR, SYS=MONITR;SYS/Y`  
`DK1/LP, SYS=LP, SYS/Y`

etc., until all desired files are transferred.

When a disk is initialized for RT-11, the PIP /Z/N switches can be used to determine the maximum number of files that can be entered in the directory. (See the RT-11 System Reference Manual, Section 4.2.7 for details.) The PIP /Z command allows the following approximate number of files per disk:

*DK:/Z = DK:/Z/N:4 = 280 files allowed
*DK:/Z/N:10 = 560 files allowed
*DK:/Z/N:37 = 2170 files allowed

The procedures given in this document are for single platter disks; if more platters are on the system, the disk initialization command can be changed.

For example, to build an RK11 disk system from a running DECTape system, use the following commands:

<table>
<thead>
<tr>
<th>Commands</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| `R PIP<CR>`  
`RK1/Z/N112<CR>`  
`RK1/Z ARE YOU SURE?Y<CR>`  
`RK1/A=DE1/S<CR>`  
`RK1/DYMNSJ, SYS=RK1/MONITR, SYS/Y/R<CR>` | Initialize the disk.

Copy the DECTape files onto disk.
 Rename the DECTape monitor on the disk to an appropriate name. |
RT-11 SYSTEM CUSTOMIZATION

Command                                      Explanation

*.RKIMONITR,SYSPRIRKMNJSY/Y/R<CR>             Rename the disk monitor on
_                                                 the disk to MONITR.SYS (in
_                                                 this case, the Single-Job
_                                                 Monitor).

_.RKIA=RKIMONITR,SYSU<CR>                     Write the system bootstrap
_                                                 on the disk.

_                                                 Check for bad blocks.

The RKII disk may now be bootstrapped as described in the RT-11 System
Reference Manual, Chapter 2, or with the PIP /O switch. The original
DECTape MONITR.SYS file (on the disk) must be renamed to another name
before the RK monitor is named to MONITR.SYS, or it will be
automatically deleted when the RK monitor rename is performed.

As another example, to build an RFII disk system from a running RKII
system, use the following commands:

Command                                      Explanation

_.RPI<CR>                                      Initialize the disk to
_                                                 allow room for 280 files.
_.RF1/2<CR>                                    Copy all files from the
_                                                 running system to the new
_                                                 system.

_.RF1/YES<CR>                                  Rename the RK monitor on
_                                                 the RF to an appropriate
_                                                 name.

_.RF1RIKMNJSY=RF1MONITR,SYSY/R<CR>             Rename the desired RF
_                                                 monitor on the RF disk to
_                                                 MONITR.SYS (in this case,
_                                                 the F/B Monitor).

_.RF1RFMNFB,SYSY/R<CR>                         Write the system bootstrap
_                                                 on the RF disk.

_                                                 Check for bad blocks.

The RFII disk may now be bootstrapped.

Note that in the above examples, all files were copied from the
running system to the system being built. Although this is a common
practice, it is not necessary, and is usually not possible when the
running system is disk and the target device is DECTape or a smaller
disk. Only the following elements must be transferred:

1. A monitor file
2. The handlers for the desired devices
3. The hardware bootstrap
4. Those programs and files that will be used with the new
   system.

When building a system for a configuration that contains only LA36 and
diskette or DECTape, it is wise to avoid transferring any handler
files that are not required, since they require space on the system
device, yet serve no purpose. Note that the system device handler is
built into the monitor, therefore the file DT.SYS need not be
transferred to a DECTape system since both DTMNSJ.SYS and DTMNFB.SYS
contain the DECTape handler.
Users of 8K machines may choose to build systems without the files MACRO.SAV, BA.SYS, and BATCH.SAV, while EXPAND.SAV and ASEMBL.SAV can be eliminated from Single-Job systems with 16K or more of memory.

Although the system is distributed with 12 monitor files, the system only requires 1 monitor for day-to-day operation. An application would seldom need more than two (the Single-Job and Foreground/Background Monitors for a particular system device often reside on the same volume for convenient switching).

The following example details the building of a working diskette-based system from another diskette. This system configuration includes diskette, console terminal, and 16K words of memory.

<table>
<thead>
<tr>
<th>Command</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>@R PIP&lt;CR&gt;</td>
<td>Initialize the disk to be built.</td>
</tr>
<tr>
<td>$DX1f/Z&lt;CR&gt;</td>
<td>Copy the RX11 Single-Job Monitor to the new system.</td>
</tr>
<tr>
<td>DX1Y/Z ARE YOU SURE?Y&lt;CR&gt;</td>
<td>Copy the RX11 F/B monitor to the new system.</td>
</tr>
<tr>
<td>-DX1MONITR,SYS=DX01MONITR,SYS/Y/X&lt;CR&gt;</td>
<td>Copy the console terminal and BATCH handlers.</td>
</tr>
<tr>
<td>-DX1DMNF,SY=DX01DMNF,SY3/Y/X&lt;CR&gt;</td>
<td>Copy the BATCH compiler and the Editor.</td>
</tr>
<tr>
<td>-DX1ITT,SY=BA,SYS/Y/X&lt;CR&gt;</td>
<td>Copy other needed system files to the new system.</td>
</tr>
<tr>
<td>-DX1BATCH,SAV,EDIT;SAV/X&lt;CR&gt;</td>
<td>Write the system bootstrap onto the new system device.</td>
</tr>
<tr>
<td>-DX1MACRO,SAV,CREF,SAV/X&lt;CR&gt;</td>
<td>Check for bad blocks.</td>
</tr>
<tr>
<td>-DX1LINK,SAV,PIP,SAV/X&lt;CR&gt;</td>
<td>Bootstrap the new system device.</td>
</tr>
<tr>
<td>-DX1SYSMAC,SML/X&lt;CR&gt;</td>
<td></td>
</tr>
<tr>
<td>-DX1MONITR,SYS/U&lt;CR&gt;</td>
<td></td>
</tr>
<tr>
<td>-DX1f/K&lt;CR&gt;</td>
<td></td>
</tr>
<tr>
<td>-DX1f/O&lt;CR&gt;</td>
<td></td>
</tr>
</tbody>
</table>

4.3 BUILDING RT-11 SYSTEMS FROM MAGTAPE (MBUILD)

The magtape on which RT-11 is distributed is a bootable magtape that contains RT-11 files. The following files on the system magtape have special significance:

- **MBOOT.BOT**: Used to read the secondary bootstrap MSBOOT.BOT into memory.
- **MSBOOT.BOT**: Used to load an RT-11 file (normally MBUILD) from magtape.
- **MBUILD.MT1**: Used to build RK11, RJS03/4, or RP11/RP02 disk systems from TM11/TS03 or TM11/TU10 magtape.
- **MBUILD.MT2**: Used to build RP11 disk systems from TM11/TS03 or TM11/TU10 magtape.
- **MBUILD.MM1**: Used to build RK11, RJS03/4, and RP11/RP02 disk systems from TJU16 magtape.
RT-11 SYSTEM CUSTOMIZATION

MBUILD.MM2 Used to build RT11 disk systems from T1JU16 magtape.

MTINIT.SAV Used to initialize a bootable magtape. MTINIT writes the bootstrap on the magtape and zeroes the tape. MTINIT is used as part of the procedure to backup a system device on a bootable magtape.

The remaining .SYS files are the monitor and handler files as described in Section 4.1. Step-by-step instructions for building RT-11 with MBUILD are contained in Section 2.5.1.

4.3.1 Building RT-11 Systems with MBUILD

MBUILD is the special system-build version of PIP which is loaded via MSBOOT. This program is used to initialize the disk and transfer the remaining files from magtape to disk. To build an RT-11 system from magtape, the bootable magtape is booted via a hardware or software bootstrap which reads MBOOT.BOT into memory. MBOOT.BOT then automatically reads the secondary bootstrap MSBOOT.BOT (which must be the first file on the tape) into memory.

MSBOOT.BOT types its version number, then prints an asterisk and waits for the user to enter the file name of the MBUILD file to be used. As noted above, the file name entered depends on the type of disk being built and the type of magtape interface (TM11 or T1JU16). When the file name is entered, MSBOOT searches the magtape for the file, reads it into memory, and starts it.

MBUILD prints its version number, then prints an asterisk. In response to the asterisk, the user enters an I/O specification in the standard RT-11 PIP format. The various operations that can be performed by MBUILD are summarized in Table 4-1; they are a subset of PIP commands. If no switch is specified, MBUILD assumes that the operation is a file transfer in image mode; the files are not concatenated.

NOTE

If more than 50 files at a time are to be transferred to the disk, do not use the *.* construction to perform the operation. Use MBUILD to transfer the necessary magtape handlers, monitors, and PIP.SAV; then use PIP to transfer the remaining files.
## Table 4-1
### MBUILD Switches

<table>
<thead>
<tr>
<th>Switch</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>/C</td>
<td>Used with another switch to cause only files with the current date (as designated using the monitor DATE command) to be included in the specified operation.</td>
</tr>
<tr>
<td>/E</td>
<td>Lists the entire specified directory, including unused spaces, on the terminal.</td>
</tr>
<tr>
<td>/F</td>
<td>Lists the short specified directory (file names only) on the terminal.</td>
</tr>
<tr>
<td>/I</td>
<td>Transfers files in image mode.</td>
</tr>
<tr>
<td>/L</td>
<td>Lists the specified directory on the terminal; this directory includes the file names and dates for each entry.</td>
</tr>
<tr>
<td>/M:n</td>
<td>Used to prevent the magtape from rewinding between each file involved in the operation; n should be a large positive number.</td>
</tr>
<tr>
<td>/N:n</td>
<td>Used with /Z to specify the number of directory segments (n) to allocate to the disk directory.</td>
</tr>
<tr>
<td>/O</td>
<td>Boots the RT-ll system from the specified disk.</td>
</tr>
<tr>
<td>/Q</td>
<td>When used with another switch, causes MBUILD to print each file name that is eligible for a wild card operation and to ask for a confirmation of its inclusion in the operation. Typing a Y causes the named file to be included in the operation; typing anything else excludes the file. The command line is not processed until the user has confirmed each file in the operation.</td>
</tr>
<tr>
<td>/R</td>
<td>Renames the specified file.</td>
</tr>
<tr>
<td>/U</td>
<td>Copies the bootstrap from the specified file into the boot blocks of the specified volume.</td>
</tr>
<tr>
<td>/X</td>
<td>When combined with wild card (*) file specification, causes all files that satisfy conditions to be individually transferred.</td>
</tr>
<tr>
<td>/Y</td>
<td>Causes system files and .BAD files to be operated on by the command specified. Attempted modifications of .SYS or .BAD files without /Y are not done and cause the message ?NO .SYS/.BAD ACTION? to be printed.</td>
</tr>
<tr>
<td>/Z[:n]</td>
<td>Zeroes (initializes) the directory of the specified disk; n is used to allocate extra words per directory entry. When used with /N, the number of directory segments for entries may be specified.</td>
</tr>
</tbody>
</table>
4.3.2 Creating Bootable Magtapes with MTINIT

MTINIT is a small program used to initialize a magtape so that it can be bootstrapped from RT-11. This program writes the bootstrap on the magtape and zeroes the magtape. MTINIT is used primarily to create a bootable magtape backup of the system disk.

The system disk being used to create the magtape must contain the following files: MBOOT.BOT, MSBOOT.BOT, MTINIT.SAV, the appropriate MBUILD file, the appropriate monitors, the appropriate device handlers, and the desired system programs.

To call MTINIT from the RT-11 system device, type:

```
R MTINIT<CR>
```

in response to the dot printed by the Keyboard Monitor. MTINIT prints:

```
MTINIT Vxx-xx
```

*  

and the user responds:

```
MTn:=<CR>
```

where n is the magtape unit number on which the bootable magtape is to be created. MTINIT rewinds the magtape on MTh and prints:

```
MTn:/ZERO/BOOT - ARE YOU SURE?
```

the user responds,

```
Y<CR>
```

When MTINIT has written the bootstrap and zeroed the magtape, it returns to the keyboard monitor which prompts the user with a dot at the terminal.

PIP can then be used to transfer the remaining files to the magtape. Files must be copied to the magtape in the following order:

1. MSBOOT.BOT
2. MBUILD.xxx files
3. Monitor files
4. Device handlers
5. Programs

4.4 BUILDING RT-11 SYSTEMS FROM CASSETTE (CBUILD)

On cassette, RT-11 is distributed as a series of RT-11 files on several cassettes, each cassette labeled DEC-11-ORTSA-E-TCn. The following files on the system cassettes have special significance:
CBUILD.SYS

CBUILD is the special system-build version of PIP which is loaded via the cassette bootstrap. This program is used to initialize the disk and transfer the remaining files from cassette to disk. CBUILD.SYS is used to build RT11 systems only.

MONITR.SYS


The remaining .SYS files are the monitor and handler files as described in Section 4.1. Step-by-step instructions for building RT-11 with CBUILD are contained in Section 2.6.1.

CBUILD is designed to allow cassettes to serve as backup devices. CBUILD can be used to transfer files between standard cassettes and other RT-11 devices, delete cassette files, and list cassette directories. Cassette to cassette transfers, however, are not allowed.

To build an RT-11 system from cassette, bootstrap an RT-11 cassette (with the file CBUILD.SYS as the first file) on Unit 0 and perform the following operations:

In response to the asterisk printed by CBUILD, type an I/O specification in the standard RT-11 format.

CBUILD accepts up to six input files and three output files. The contents of the input file are transferred to the output file in image mode.

The various operations that can be performed by CBUILD are summarized in Table 4-2. If no switch is specified, CBUILD assumes that the operation is a file transfer in image mode; the files are not concatenated.

<table>
<thead>
<tr>
<th>Switch</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>/A</td>
<td>Transfers files in ASCII mode (nulls are ignored).</td>
</tr>
<tr>
<td>/B</td>
<td>Transfers files in formatted binary mode.</td>
</tr>
<tr>
<td>/D</td>
<td>Deletes the file specified from the output cassette. The /D switch is valid only if the output device is a cassette. For example:</td>
</tr>
<tr>
<td></td>
<td>_CT1:OFIL.E MAC/D</td>
</tr>
<tr>
<td></td>
<td>deletes OFILE.MAC from the cassette on drive 1.</td>
</tr>
<tr>
<td>/E</td>
<td>Lists the entire directory including spaces called *EMPTY which result from deleted files. For example:</td>
</tr>
<tr>
<td></td>
<td>_CT0:/*E</td>
</tr>
<tr>
<td></td>
<td>lists the directory of the cassette mounted on unit 0 including *EMPTY entries.</td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>Switch</th>
<th>Meaning</th>
</tr>
</thead>
</table>
| /F     | Prints the short directory of the specified cassette (file names only). For example:  
  `@LP1=CT11/F`  
  prints the directory (file names only) of cassette unit 1 on the line printer. |
| /G     | Copies a file and ignores input errors. When copying from a cassette to another device, input errors in data and the header are ignored. |
| /I or none | Transfers files in image mode. For example:  
  `@CT10FA,FB,FC=FD,FE,FF/I`  
  transfers files FD, FE, and FF from device DK to cassette unit 0 as files FA, FB, FC. |
| /L     | Reads the input cassette directory and writes it on the output device. The directory includes file names and dates for each entry. Notice that in this case the input file itself is not transferred, only the directory. The /L switch is valid only if the input device is a cassette. |
| /M     | Reads multi-volume cassette files. |
| /N     | Used with /Z to specify the number of directory blocks to allocate on the disk or DECtape. The number of directory blocks governs how many files can be stored on a volume. |
| /O     | Boots the RT-11 system from the specified disk or DECtape unit 0. For example:  
  `@RK1/O`  
  `@RK1/0`  
  `@RK1/O` |
| /Q     | Reads after write; writes a block then reads that block and checks for write errors. |
| /U     | Transfers the system bootstrap to the specified device. For example:  
  `@DK1A=CT1IMONITR,SYS/U`  
  transfers the bootstrap in the file MONITR.SYS from cassette unit 1 to the boot blocks of the device DK. The file name A is a dummy specification. |
| /V     | Displays the version number of CBUILD. |
| /X     | When combined with wild card (*) file specification, causes all files that satisfy conditions to be individually transferred. |

(continued on next page)
RT-11 SYSTEM CUSTOMIZATION

Table 4-2 (Cont.)
CBUILD Switches

<table>
<thead>
<tr>
<th>Switch</th>
<th>Meaning</th>
</tr>
</thead>
</table>
| /Z     | Clears the directory of the specified device. The message:  

**ARE YOU SURE?**

is displayed. Enter Y and a carriage return to zero the directory; any other response causes CBUILD to ignore the command. /Z allows specification of extra words in the directory entry when used to zero a disk or DECtape. For example:

```
DT$1/Z12
```

provides two extra words per directory entry. A value given to the /Z switch when used to zero a cassette has no effect.

Use the following set of commands (or their equivalent) to build the RK11 disk system.

<table>
<thead>
<tr>
<th>Commands</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| #RK1/E/N12<CR>  
**ARE YOU SURE?**  
Y<CR> |

Write system files on disk. Repeat this command for each cassette necessary to build the desired system.

| #RK1A=CTBIMONTR,SYS/U<CR>  
#RK1/O<CR> |

Write bootstrap on disk. Bootstrap the disk system.

RT-11 is then running from the RK11 disk and may be used in the normal fashion to copy any other desired files from cassette to disk. Note that no devices other than disk or cassette can be used until their handler files have been added to the disk and the system has been rebooted.

4.5 BUILDING DISK SYSTEMS FROM PAPER TAPE (PT BUILD)

RT-11 is distributed as object modules on paper tape. Two of the tapes (PT BUILD Tapes 1 and 2) are used to place a rudimentary VOL-15 Monitor and Linker on the disk. The disk system is then started and the Linker is used to link OLDPIP from paper tape onto the disk. Once linked, OLDPIP is used to copy the remaining paper tapes onto the disk, where they can be linked to complete the system.

The following paper tapes have special significance:

- **DEC-11-ORPBA-D-PB1** This paper tape is for RK11 systems only.
- RK PT BUILD Tape 1
- **DEC-11-ORPBA-D-PB2** This paper tape is for both RK11 and RP11 systems.
- PT BUILD Tape 2

4-12
RT-11 SYSTEM CUSTOMIZATION

DEC-11-ORPBA-D-PB3  This paper tape is for RF11 systems only.
RF PT BUILD Tape 1

To build an RT-11 system from paper tape, perform the following operations:

1. Load the Bootstrap Loader (see Section A.6) at 37744, then use it to load the Absolute Loader.

2. Using the Absolute Loader, load the appropriate PT BUILD Tape 1 for the system device desired. It self starts and prints:

   PT BUILD Version number

3. There is a 10-15 second pause, after which PT BUILD prints:

   PLACE SECOND TAPE IN READER;
   STRIKE ANY CHARACTER TO CONTINUE.

4. Place the tape PT BUILD Tape 2 into the reader, then strike any character to start the tape.

   There is a slight pause, after which the following is printed:

   DISK BUILD COMPLETE,
   RT-11 V01-15

During the build procedure, only a WRITE FAILED error message has exact meaning. If encountered, check to ensure that the system disk is not WRITE-PROTECTED. Any other error indicates a user error or hardware problem.

5. Link OLDPIP as follows:

   \R\LINK<CR>
   *OLDPIP=PRI
   ***

   For each occurrence of the prompting ^, place the tape OLDPIP (DEC-11-OROPA-D-PR) in the reader, then strike a character to read the tape. Type CTRL C when the second * is printed.

6. Run OLDPIP to copy the remaining tapes onto the system disk (use the /B switch for all files but SYSMAC.SML, SYSMAC.8K, and VTMAC.MAC, which require /A.) Unlike the V02C system when built, OLDPIP prompts with an up-arrow or circumflex prior to reading each tape, and proceeds when a character is struck at the keyboard.

   Section 2.7.1 lists the tapes distributed with RT-11. If a specific component is not required, it is not necessary to transfer the corresponding tapes.

PREXEC.OBJ and PREPAS.OBJ are the object modules for EXPAND. SMEXEC.OBJ, SMMAC.OBJ, and SMPST.OBJ are linked for ASEMBL, while RTEXEC.OBJ, RTMAC.OBJ, and RTPST.OBJ are the MACRO object modules. PIP.OBJ is the object module for PIP.
RT-11 SYSTEM CUSTOMIZATION

VTHDLR, after a pass by the Librarian, becomes the display handler. CT.OBJ through MM.OBJ are linked into handlers of the same name; RPBTJF.OBJ through RTL1SJ.OBJ are the RFL1 and RK11 monitor components. SYSF4.OBJ becomes SYSLIB after a pass through the Librarian; BA.OBJ and BATCH.OBJ are the BATCH system components.

EDIT.OBJ and VTL EDT.OBJ make the Editor, while LINK0 through LNKOV5 are the components for the V02C Linker. DUMP.OBJ, SRCOM.OBJ, FILEX.OBJ, and CREP.OBJ are linked for programs of the same name, while LIBR0 through LIBR4 become the Librarian. PAT0 through R50ASC are linked with RT-11 FORTRAN (if available) to make PATCHO.

7. Run LINK to generate the V02C .SYS files and program .SAV files as described in Chapter 5. Linking the new V02C Linker will generate ADDITIVE REF messages, which should be ignored. Once the V02C Linker is built, it should be used for subsequent linking operations.

8. Rename the desired monitor file to MONITOR.SYS, then copy the bootstrap with the PIP /U switch.

9. Reboot the system with a hardware bootstrap or use the bootstrapping procedures in the appropriate section of Appendix A. The .OBJ files (except for ODT, SYSLIB, and VTLIB) can be deleted as well as OLDPIP.SAV, since they are no longer required.

4.5.1 Directory Extension

When a device is zeroed, additional directory segments can be specified by use of the PIP /N switch. Once files are written on the device, however, the directory size cannot be changed without using the /Z option in PIP, which destroys existing directory information.

Although it is recommended that, whenever possible, the PIP /Z/N combination be used to determine directory size, the following program may be used to change the size of a directory on-line without reinitializing the disk. It can be used if there is no other alternative, and will be of special interest to paper tape users who cannot specify the size of their disk directory at system build time. The program should be used only as a last resort, and only on a disk which has been thoroughly backed-up. It should be entered as instructed in the comments. Instructions for its use are also in the comments.

```plaintext
#DIRECT-PROGRAM TO ADD SEGMENTS TO AN RT-11 DIRECTORY
#TO USE, THE FIRST ENTRY IN THE DIRECTORY MUST BE <UNUSED>, AND ITS LENGTH
#MUST BE TWICE AS MANY BLOCKS AS THE NUMBER OF SEGMENTS THAT ARE TO
#BE ADDED. THE FIRST ENTRY MAY BE MADE <UNUSED> SIMPLY BY CALLING
#PIP TO MOVE THE FIRST FILE ON THE DEVICE. ONCE THE UNUSED AREA HAS BEEN
#CREATED, DIRECT IS RUN BY TYPING:
#
# .R DIRECT
#THE RESPONSE: *
#
# A COMMAND LINE IS ENTERED OF THE FORM "DEV;/N:H" WHERE DEV IS
# THE DEVICE Whose DIRECTORY IS TO BE EXPANDED AND "H" IS
# THE NUMBER OF SEGMENTS (NOT BLOCKS) TO ADD TO THE DIRECTORY.
```
RT-11 SYSTEM CUSTOMIZATION

FOR EXAMPLE:

*RK1/N14

WILL ADD 12 (DECIMAL) SEGMENTS TO THE
DIRECTORY ON RK0, FOR WHICH THE FIRST
FILE MUST BE AN <UNUSED> AREA OF AT
LEAST 24 BLOCKS

THERE ARE TWO POSSIBLE ERROR MESSAGES:

?HARD I/O ERROR?
WILL OCCUR IF THE DIRECTORY FOR THE
DEVICE CANNOT BE READ OR WRITTEN WITHOUT
HARDWARE ERROR.

?ILLEGAL COMMAND?
WILL OCCUR ON ANY OF THE FOLLOWING CONDITIONS
1) INAPPROPRIATE COMMAND STRING
2) FIRST ENTRY IN DIRECTORY NOT <UNUSED>
3) REQUESTED NUMBER OF SEGMENTS WOULD
CAUSE TOTAL TO EXCEED ALLOWABLE 31(10)
4) SIZE OF <UNUSED> AREA TOO SMALL TO
ACCOMODATE REQUESTED DIRECTORY INCREASE

AND THE PROGRAM IS RESTARTED.

TO CREATE DIREXT, TYPE THIS TEXT INTO THE FILE "DIREXT.MAC" WITH THE EDITOR.

TO ASSEMBLE: .R MACRO
*DIREXT=DIREXT

TO LINK: .R LINK
*DIREXT=DIREXT

CALL ..V1,.CSIGEN,.READW,.WRITW,.EXIT,.PRINT
..V1,
R0=%0
R1=1
R2=2
R3=3
R4=4
R5=5
SP=6

ILLCMD: .PRINT $MSG1 ;TYPE "?ILLEGAL COMMAND?"
START: .CSIGEN $DSPACE,$DEXT,0 ;GET COMMAND STRING
DEC (SP)+ ;ANY SWITCHES?
BNE ILLCMD ;NO-BAD COMMAND
CMP (SP)+,$101516 ;NIM ON FIRST INPUT FILE?
BNE ILLCMD ;NO-BAD COMMAND
MOV (SP)+,R1 ;NUMBER OF SEGS TO ADD INTO R1
MOV $6,R2 ;BLOCK FOR CURRENT SEG IN R2
MOV $DIRBUF,R3 ;R3 POINTS INTO DIRECTORY BUFFER
,READW 3,R3,#256,R2 ;READ FIRST DIRECTORY SEG
BCS HERR ;READ ERROR
CMP 12(R3),#1000 ;FIRST ENTRY <UNUSED>?
BNE ILLCMD ;NO-BAD COMMAND
MOV (R3),R4 ;NUMBER OF SEGS NOW IN DIR INTO R4
ADD R1,R4 ;TOTAL SEGS FOR NEW DIR IN R4
MOV R4,R5 ;REMEMBER NEW SEG TOTAL
CMP R4,#37 ;IS NEW TOTAL TOO LARGE?
BHI ILLCMD ;YES-BAD COMMAND
ASL R1 ;# OF BLOCKS NEEDED FOR NEW SEG
CMP R1,22(R3) ;<UNUSED> LARGE ENOUGH TO TAKE NEW SEG?
BGT ILLCMD ;NO-BAD COMMAND
ADD R1,10(R3) ;UPDATE FILE START ADDRESS
SUB R1,22(R3) ;REDUCE <UNUSED> SIZE

4-15
RT-11 SYSTEM CUSTOMIZATION

LOOP: MOV R4, (R3) ; UPDATE NUMBER OF SEG TO NEW TOTAL
.WRITW 3+R3,*256,#R2 ; WRITE SEG BACK OUT
BCS HERR ; WRITE ERROR
DEC R5 ; ALL SEG UPDATED?
BLE DONE ; YES
TST (R2)+ ; NO-ADD 2 TO BLOCK NUMBER
.READW 3+R3,*256,#R2 ; READ NEXT SEG
BCC LOOP ; AND UPDATE
HERR: .PRINT MSG92 ; PRINT *?HARD I/O ERROR?*
DONE: .EXIT
MS91: .ASCIZ *?ILLEGAL COMMAND?*
MS92: .ASCIZ *?HARD I/O ERROR?*
.EVEN
DEXT: 0,0,0,0
DIRBUF: .=.+1000
DSPACE: .END START

4.6 CUSTOMIZATION FOR SPECIAL HARDWARE

This section contains instructions on patching various RT-11 system components to customize your system for specific hardware. The patching instructions contain certain mnemonics in place of actual values. For example, the mnemonic $HSIZE is used in place of the actual value for the handler size. The actual values for these mnemonics are listed in RT-11 System Release Notes, Table 2.

4.6.1 High Baud Rate Serial Console Devices

The serial LA30 (LA30S) requires that filler characters follow each carriage return; the 600, 1200, and 2400 baud VT05's require that filler characters follow each line feed. RT-11 has established a mechanism by which any number of fills may follow any character. The byte at location 56 (octal) contains the character to be followed by fillers and the byte at location 57 (octal) contains the number of null fills to be used. These locations are initially set to zero, which results in no fillers being generated (normal operation for LT33, LA30P, LA36, VT50, and VT52).

Depending on the terminal, modify the locations as follows:

<table>
<thead>
<tr>
<th>Location 56</th>
<th>Location 57</th>
<th>Resulting Word (octal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA30S 110 baud</td>
<td>015(8)</td>
<td>002(8)</td>
</tr>
<tr>
<td>LA30S 150 baud</td>
<td>015(8)</td>
<td>004(8)</td>
</tr>
<tr>
<td>LA30S 300 baud</td>
<td>015(8)</td>
<td>012(8)</td>
</tr>
<tr>
<td>VT05 600 baud</td>
<td>012(8)</td>
<td>001(8)</td>
</tr>
<tr>
<td>VT05 1200 baud</td>
<td>012(8)</td>
<td>002(8)</td>
</tr>
<tr>
<td>VT05 2400 baud</td>
<td>012(8)</td>
<td>004(8)</td>
</tr>
</tbody>
</table>

The proper octal word can be changed permanently in the monitor by using PATCH to modify locations 56 and 57 in the monitor file. For example:

4-16
PATCH Version number

FILE NAME--
*MONITR.SYS/M<CR>
*56\015<LF>
57\014<CR> (Fill after <CR> with 4 nulls)
*E

Once the change is made, the bootstrap recopied with the PIP /U switch, and the monitor rebooted with the PIP /O switch, all programs that use the monitor for console I/O will operate correctly.

4.6.2 Magtape Parity, Density, Number of Tracks

4.6.2.1 TM11 (MT.SYS) -- The RT-11 TM11 magtape handler is distributed such that it will correctly handle both 7- and 9-track drives without modification. It does so at 800 bpi, using the TM11 dump mode for 7-track drives. Seven-track drives can also be written (in hardware mode only; see the RT-11 System Reference Manual, Appendix H) at 200, 556, and 800 bpi (non-dump mode) by modifying the handler as described below.

To alter the magtape density used by the handler, the following patches must be made:

1. Patch the handler density word (location PARDEH in MT.SYS).
2. When changing from 800 bpi 9- or 7-track dump mode to 200, 556, or 800 bpi 7-track, you must patch the handler to default to hardware mode (location HW in MT.SYS) and patch the $STAT entry in the monitor for MT.SYS (location MTSTAT in MONITR.SYS) so that the special device bit (bit 12) is off (zero).
3. When changing to 800 bpi 9- or 7-track dump mode from 200, 556, or 800 bpi 7-track, you must patch the handler to default to software mode and patch the $STAT entry in the monitor for MT.SYS so that the special device bit is on (one).

NOTE

In the 200, 556, and 800 bpi 7-track modes, with the $STAT table patched so that bit 12 is off, the MT handler will not get control when a .LOOKUP, .ENTER, .CLOSE, or .DELETE is issued for MT. Doing a .LOOKUP, .ENTER, or .CLOSE in this state simply opens or closes a channel associated with MT. The magtape is not rewound for any of these functions in this state; the only operations that are passed to the magtape handler are .READ, .WRITE, and special functions (.SPFUN).
The following table describes the patches for the various magtape densities:

<table>
<thead>
<tr>
<th>Location</th>
<th>200 bpi 7-track</th>
<th>556 bpi 7-track</th>
<th>800 bpi 7-track</th>
<th>7-track dump mode (800 bpi) or 800 bpi 9-track</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARDEN (MT.SYS)</td>
<td>0</td>
<td>20000</td>
<td>40000</td>
<td>60000</td>
</tr>
<tr>
<td>HW (MT.SYS)</td>
<td>377</td>
<td>377</td>
<td>377</td>
<td>0</td>
</tr>
<tr>
<td>MTSTAT (MONITR.SYS)</td>
<td>2011</td>
<td>2011</td>
<td>2011</td>
<td>12011</td>
</tr>
</tbody>
</table>

For example, to cause the MT handler to write 7-track tapes at 556 bpi:

```
.R PATCH<CR>

PATCH Version number

FILE NAME--
*MT.SYS<CR>
*PARDEN/60000 20000<CR>
*HW /00000 377<CR>
*F

FILE NAME--
*MONITR.SYS/M<CR>
*MTSTAT/12011 2011<CR>
*E

.R PIP<CR>
*SY:A=MONITR.SYS/U<CR>
*SY:/O<CR>
```

NOTE

See RT-11 System Release Notes, Table 2, for the exact addresses of PARDEN, HW, and MTSTAT.

4.6.2.2 **TJU16 (MM.SYS)** -- The TJU16 allows five possible tape modes in file-structured operation. Since the user may wish to transfer between tapes written in various modes, the TJU16 handler includes a table that defines the default mode for each magtape unit (MM0 to MM7). The user can change the default mode for a particular unit simply by patching that unit's entry in the mode table. The table is located at UNIMOD in the magtape handler and is eight words long. Each word contains the mode for a given unit (0-7).

The default mode for a particular unit may also be changed dynamically under program control. A nonfile-structured LOOKUP with a file count between 1 and 5 causes a mode change for the unit accessed. The following list shows the possible modes and corresponding table value for patching and file count for dynamic modification:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Table Value</th>
<th>File Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 bpi</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>556 bpi</td>
<td>400</td>
<td>2</td>
</tr>
<tr>
<td>800 bpi (odd parity)</td>
<td>1000</td>
<td>3</td>
</tr>
<tr>
<td>800 bpi (even parity)</td>
<td>1400</td>
<td>4</td>
</tr>
<tr>
<td>1600 bpi (phase-encoded)</td>
<td>2000</td>
<td>5</td>
</tr>
</tbody>
</table>

4-18  
July 1976
RT-11 SYSTEM CUSTOMIZATION

Default for all drives is initially 800 bpi even parity. The handler can distinguish between 7- and 9-track drives, so the user need not concern himself with the type of drive in use.

For example, to cause unit 1 to write in 1600 bpi mode (the first word of the table, location UNIMOD, is mode for unit 0; location UNIMOD+2 is mode for unit 1):

```
_.R PATCH<CR>

PATCH Version number

FILE NAME--
*MM.SYS<CR>
*UNIMOD+2/1000 2000<CR>
*E
_.R PIP<CR>
*SY:/O<CR>
```

NOTE

When dynamically modifying the mode with a nonfile-structured LOOKUP, position the magtape at the beginning of the tape (BOT). See RT-11 Release Notes, Table 2, for the exact address of UNIMOD.

4.6.3 Specifying the Number of RF11 Platters

RT-11 is distributed with fixed-head disk support initialized for one platter. To allow RT-11 to make use of more than one platter, the device size table in the various monitor files must be modified as follows:

<table>
<thead>
<tr>
<th>Number of Platters</th>
<th>New Value of Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2000</td>
</tr>
<tr>
<td>2</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>6000</td>
</tr>
<tr>
<td>4</td>
<td>10000</td>
</tr>
</tbody>
</table>

For example, to modify the RF11 F/B Monitor for three RF/RS11 platters, type:

```
_.R PATCH

PATCH Version number

FILE NAME--
*RFMNFB.SYS/M<CR>
*RF$1Z/2000 6000<CR>
*E
_.R PIP<CR>
*SY:/O<CR>
```
NOTE

For all monitors, the address to modify for RF disk is RFSIZ. The address of RFSIZ for the current version of the monitor can be found in RT-11 System Release Notes, Table 2.

Once the above change has been made, zeroing the disk (using the PIP /Z switch) will adjust the directory to the appropriate size. If the system is already running off fixed-head disk as the system device and the disk cannot be zeroed without destroying the system, compressing the disk (with the PIP /S switch) will automatically re-adjust the directory size.

4.6.4 Specifying a 50-Cycle Clock Rate

RT-11 is distributed with the Keyboard Monitor TIME command calculations based on a 60-cycle clock rate. To cause the TIME command to base calculations on a 50-cycle clock rate, modify the monitor such that bit 5 (40(octal)) is set in the monitor configuration word (see the RT-11 System Reference Manual, Section 9.2.6). For example:

For the F/B Monitors:

_.R PATCH<CR>

PATCH Version number

FILE NAME--
*MONITR.SYS/M<CR>
*CONFIG/1 41<CR>
*E
_.R PIP<CR>
*SY:A=MONITR.SYS/U<CR>
*SY:/O<CR>

NOTE

See RT-11 System Release Notes, Table 2 for the exact address of CONFIG in the current monitor.

For the S/J Monitors:

_.R PATCH<CR>

PATCH Version number

FILE NAME--
*MONITR.SYS/M<CR>
*CONFIG/0 40<CR>
*E
4.6.5 Interfacing RJS03/4 DISKS TO RT-11

RT-11 is distributed with the monitor device tables for RJS03/4 disk initialized for RJS03. To allow complete use of all the space available on an RJS04 disk, modify the device size table in the monitor as follows:

<table>
<thead>
<tr>
<th>Location</th>
<th>RJS03 Value</th>
<th>RJS04 Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSSIZ</td>
<td>2000</td>
<td>4000</td>
</tr>
</tbody>
</table>

For example, to modify the RT11 DS monitor to support an RJS04 disk rather than RJS03, type:

```
.R PATCH

PATCH Version number

FILE NAME--
*MONITR.SYS/M<CR>
*DSSIZ/2000        4000<CR>
*E

.R PIP<CR>
*SY:A=MONITR.SYS/U<CR>
*SY:/O<CR>
```

NOTE

See RT-11 System Release Notes, Table 2, for the exact address of DSSIZ in the current monitor.

Once the above change has been made, zeroing the disk (using the PIP /Z switch) will adjust the directory to the appropriate size. If the system is already running from fixed-head disk as the system device and the disk cannot be zeroed without destroying the system, compressing the disk (with the PIP /S switch) will automatically re-adjust the directory size.

4.6.6 Interfacing RP03 Disks to RT-11

The RP02 support provided in the distribution kit is initialized for RP02 only. The RT-11 file structure can accommodate a maximum of 64000(10) blocks. The 40000(10)-block RP02 cartridge, therefore, can be accommodated as a single logical unit, while an RP03 cannot.

The RT-11 RP02 support can easily be altered, however to accommodate RP03s as follows:
1. Each RP03 drive must be considered as two logical units of 40000 blocks each; in essence, a single RP03 drive looks like two RP02 drives to the system. The cartridge on physical unit n is accessed as logical DPn and DPn+4; thus, drive 0 is referenced as DP0: and DP4:, drive 1 is DP1: and DP5:, etc. Note that although an RP03 is physically one device, it is two separate devices to the system. Each logical unit has its own complete directory and data space.

2. The DP.SYS handler must be patched to change location RP23 from 404 (octal) to 1404 (octal).

3. The DP monitors must be patched to alter location RP23 from 404 (octal) to 1404 (octal).

For example, to allow RT-11 to support RP03s:

```
*R PATCH<CR>

PATCH Version number

FILE NAME--
*DP.SYS<CR>
*RP23/____404____1404<CR>
*E

*R PATCH<CR>

PATCH Version number

FILE NAME--
*DPMNFB.SYS/M<CR>
*RP23/____404____1404<CR>
*E

*R PATCH<CR>

PATCH Version number

FILE NAME--
*DPMNSJ.SYS/M<CR>
*RP23/____404____1404<CR>
*E

*R PIP<CR>
*S<CR>
```

NOTE

See RT-11 System Release Notes, Table 2, for the actual location of RP23 in each monitor and in DP.SYS.

Note that a maximum of four RP03s can be supported on the system. RP02s and RP03s can be mixed as long as the total number of units (physical drives) on the system does not exceed four. If the system contains only RP02s, the above changes must not be made and the system can support as many as eight units.
4.6.7 Interfacing Card Readers to RT-11

Although the CR11 device handler is included in the RT-11 V02C kits, this device is not included in the monitor device tables due to lack of available space. To install the card reader in the V02C system, the user must examine Table 3 in RT-11 System Release Notes, select a device that is not used on the system, and replace that device with CR. Four tables must be patched: the handler size table, the device size table, the physical name table, and the device status word table. The corresponding table values for the CR driver are:

```
handler size: 1326
device size: 0
physical name: 12620 (.RAD50 /CR/)
status word: 40014
```

The table entries to be patched are:

```
$HSIZE + octal offset for driver to be replaced
$DVSIZ + octal offset for driver to be replaced
$NAME + octal offset for driver to be replaced
$STAT + octal offset for driver to be replaced
```

**NOTE**

See RT-11 System Release Notes, Table 2, for the exact addresses of $HSIZE, $DVSIZ, $NAME, and $STAT.

For example, suppose the PC11 reader/punch is not used on the system. In this case, the PR and PP drivers are not needed. The PR driver is selected to be replaced by the CR driver. The PR driver's table index is 5, which is equivalent to an octal byte offset of 10 into each table. The following patches can then be applied to the appropriate monitors.

For the Single-Job Monitors:

```
.R PATCH<CR>

PATCH Version number

FILE NAME--
*MONITR.SYS/M<CR>
*BASE:0<CR>
*0,13634/174 1326<CR>
*0,13670/0 0<CR>
*0,16500/63320 12620<CR>
*0,16534/40007 40014<CR>
*E

.R PIP<CR>
*SY:A=MONITR.SYS/U<CR>
*SY:/O<CR>
```
For the F/B Monitors:

```
._R PATCH<CR>
PATCH Version number
FILE NAME--
*MONITR.SYS/M<CR>
*BASE;0R<CR>
*0,14566/174_1326<CR>
*0,14622/0_0<CR>
*0,17640/63320_12620<CR>
*0,17674/40007_40014<CR>
*E
._R PIP<CR>
*SY:A=MONITR.SYS/U<CR>
*SY:/O<CR>
```

NOTE

See RT-11 System Release Notes, Table 2 for the actual address of BASE.

4.6.8 Changing the Location of VT11 Floating Vectors

The floating vector region on the PDP-11 is situated in locations 300 to 476; VT11 display processor vectors are normally located at 320 to 332. However, the VT11 display vectors may be forced to float by the addition of other devices. For example, VT11 device vectors may be changed if a DLL device (a communications device) is added.

In such a case, a patch to the RT-11 monitor (at GTVECT), to reflect the new location of the VT11 display vectors, is necessary. The value of the VT11 display stop vector (called NEW in the following example) must be determined by consulting the configuration data for the particular installation.

For the F/B Monitors:

```
._R PATCH<CR>
PATCH Version number
FILE NAME--
*MONITR.SYS/M<CR>
*GTVECT/320.NEW<CR>
*E
._R PIP<CR>
*SY:A=MONITR.SYS/U<CR>
*SY:/O<CR>
```

NOTE

See RT-11 System Release Notes, Table 2, for the exact address of GTVECT in the current monitor.
RT-11 SYSTEM CUSTOMIZATION

For example, if NEW is 340 and the value of GTVECT for the F/B Monitor is 37354:

```
.R PATCH<CR>

PATCH Version number

FILE NAME--
*MONITR.SYS/M<CR>
*37354/320_340<CR>
*E

.R PIP<CR>
*SY:A=MONITR.SYS/U<CR>
*SY:/O<CR>
```

This patch enables the text scroller (GT ON), the display file handler, EDIT, BASIC/GT, and FORTRAN/GT to function properly on the system without further patching.

4.6.9 Interfacing a Second Diskette Handler

RT-11 V02C is distributed with only two diskettes supported. If additional diskettes are needed, an additional device handler for each two extra diskettes must be installed into the RT-11 monitor. A system with an additional diskette handler must have at least 16K words of memory.

Perform the following operations to install a second diskette handler in the system.

1. Use PIP to create a copy of the floppy handler:

```
.R PIP<CR>
*DY.SYS=DX.SYS/Y/X<CR>
**C
```

2. Use PATCH to modify the new handler (DY.SYS) as follows:

```
.R PATCH<CR>

PATCH Version number

FILE NAME--
*DY.SYS<CR>
*1000/264_xxx<CR>
*DXIOP/177170_yyyyyy<CR>
*E

.R PIP<CR>
*SY:/O<CR>
```

where xxx is a 3-digit value for the interrupt vector address for the second diskette controller and yyyyyy is a 6-digit value for the I/O page address of the second diskette controller.

NOTE

See RT-11 System Release Notes, Table 2, for the actual address of DXIOP.
3. Use PATCH to insert the new handler into the system tables in place of another handler. Four tables must be patched: handler size, device size, physical name, and device status word. The corresponding table values for the DY driver are:

- handler size: 670
- device size: 756
- physical name: 16350 (.RAD50 /DY/)
- status word: 102022

The table entries to be patched are:

- $HSIZE + octal offset for driver to be replaced
- $DVSIZ + octal offset for driver to be replaced
- $PNAME + octal offset for driver to be replaced
- $STAT + octal offset for driver to be replaced

NOTE

See RT-11 System Release Notes, Table 2, for the exact addresses of $HSIZE, $DVSIZ, $PNAME, and $STAT.

For example, suppose that the TU60 cassette handler is not used on the system. Since the CT driver is not needed, it is selected to be replaced by the DY handler. The CT driver's table index is 10, which is equivalent to an octal byte offset of 22 into each table. The following patches can then be applied to the appropriate monitors:

For the Single-Job Monitors:

```
.R PATCH<CR>

PATCH Version number

FILE NAME--
*MONITR.SYS/M<CR>
*BASE:0R<CR>
*0,13646/3710  670<CR>
*0,13702/0  756<CR>
*0,16512/12740 16350<CR>
*0,16546/12013 102022<CR>

.R PIP<CR>
*SY:A=MONITR.SYS/U<CR>
*SY:/O<CR>
```

For the F/B Monitors:

```
.R PATCH<CR>

PATCH Version number

FILE NAME--
*MONITR.SYS/M<CR>
*BASE:0R<CR>
*0,14600/3710  670<CR>
*0,14634/0  756<CR>
*0,17652/12740 16350<CR>
*0,17706/12013 102022<CR>
```

4-26    July 1976
RT-11 SYSTEM CUSTOMIZATION

.R PIP<CR>
*BY:A=MONITR.SYS/U<CR>
*BY:/O<CR>

NOTE

See RT-11 System Release Notes, Table 2, for the exact address of BASE.

4. The diskettes on the second controller may now be referenced as DY0 and DY1. The monitor ASSIGN command may be used to assign a user logical name such as DX2 or DX3 as follows:

ASSIGN DY0IDX2<CR>
ASSIGN DY1IDX3<CR>

When the name DX2 is used, operations will be performed on DY0; when the name DX3 is used, operations will be performed on DY1. Note that the ASSIGN command is temporary and must be reissued each time the system is bootstrapped.

4.6.10 Reducing the Size of Text Window Displayed

The Editor is constructed in such a way that when the scope is in use, the window into the buffer and the scrolled commands lines are separate "pictures". On rare occasions, if the text window around the cursor contains long lines and several line feeds (or form feed characters), the window can "overflow" onto the scrolled editing commands, making that portion of the screen difficult to read.

In most applications, this problem does not occur; when it does, the obscure lines can be seen by advancing the cursor several lines to bring them into clear view.

If the problem is troublesome for a particular application, it can be removed by reducing the size of the window displayed as follows:

.R PATCH<CR>

PATCH Version number

FILE NAME--
*EDIT.SAV<CR>
*EBASE;OR<CR>
*0,DSARG/___12___n<CR>
*E

.R PIP<CR>
*SY:/O<CR>

where n is the number of lines to be displayed above and below the cursor; n should be smaller than the value currently in DSARG to eliminate the problem.

NOTE

See RT-11 System Release Notes, Table 2, for the exact addresses of EBASE and DSARG.
4.6.11  Using "{" and "~" Characters on LA36

The LA36 contains two characters ("{} and "~") which generate ASCII codes 175 (octal) and 176 (octal). Because many older terminals generate 175 (octal) and 176 (octal) for ALTMODE (or ESCAPE), RT-11 EDIT treats 175 (octal) and 176 (octal) as ALTMODE, making the new characters impossible to insert as text.

The Editor may be patched as follows to remove the special-case check for these characters so that they may be used on an LA36 or other terminal capable of handling them.

```
.R PATCH<CR>

PATCH Version number

FILE NAME--
*EDIT.SAV<CR>
*ALTTST/  175  33<CR>
*ALTTST+6/ 176  33<CR>
*E

.R PIP<CR>
*SY:/O<CR>
```

**NOTE**

See RT-11 System Release Notes, Table 2, for the actual address of ALTTST.

Once EDIT is altered in the preceding manner, "{}" and "~" can be used normally.

4.6.12  Setting an Upper Limit on File Size

RT-11 is distributed such that the maximum size of a file allocated in a general .ENTER request is half the largest space, or the entire second largest space available, whichever is larger. This is satisfactory for most applications and should be left unchanged. It is possible that certain applications require that an upper limit be set on the size of a file; for these applications (and these only), the following change can be made.

For Single-Job Monitors:

```
.R PATCH<CR>

PATCH Version number

FILE NAME--
*MONITR.SYS/M<CR>
*BASE;OR<CR>
*0,MAXBLK/177777 n<CR>
*E
```

where n is an octal number of blocks defining the maximum file size for a general .ENTER.
RT-11 SYSTEM CUSTOMIZATION

.R PIP<CR>
*SY:A=MONITR.SYS/U<CR>
*SY:/O<CR>

NOTE
See RT-11 System Release Notes, Table 2, for the actual values of BASE and MAXBLK.

For Foreground/Background Monitors:

.R PATCH<CR>

PATCH Version number

FILE NAME--
*MONITR.SYS/M<CR>
*BASE;0<CR>
*0,MAXBLK/177777n<CR>
*E

where n is an octal number of blocks defining the maximum file size for a general .ENTER.

.R PIP<CR>
*SY:A=MONITR.SYS/U<CR>
*SY:/O<CR>

4.6.13 Running RT-11 in Less Memory Than That Available

The V02C monitors have bootstraps which allow the system to run in less memory than is available on the system (e.g., RT-11 can be bootstrapped to run in the lower 12K or 8K of a 16K machine). Most applications require that RT-11 make use of all memory available, and the system is distributed such that it automatically does so.

If (and only if) an application requires that RT-11 run in less memory than is available, the following change can be made.

For all monitors:

.R PATCH<CR>

PATCH Version number

FILE NAME--
*MONITR.SYS/M<CR>
*BHALT/4070<CR>
*E

.R PIP<CR>
*SY:A=MONITR.SYS/U<CR>

NOTE
BHALT is a value obtained from Table 2 of RT-11 System Release Notes.
RT-11 SYSTEM CUSTOMIZATION

Once the change has been made and a new system bootstrap written on the device (with the PIP /U switch), a halt occurs whenever the system is booted.

At this point, set the switch register to one of the following values and press CONTINUE; the bootstrap operation completes for the specified memory size.

- 40000 = 8K
- 60000 = 12K
- 100000 = 16K
- 120000 = 20K
- 140000 = 24K
- 160000 = 28K
- >160000 = Use all available memory.

4.6.14 Accessing Nonsystem Disks on Single-Disk Systems

Source disks and other nonsystem disks can be accessed on single-disk systems with more than 8K words of memory as follows:

1. Boot the system disk and enter the date.
2. LOAD the handler for the device to which the desired files will be transferred.
3. Run PIP. When the prompting * appears, dismount the system disk.
4. Mount the source disk, WRITE LOCKed, in place of the system disk.
5. Transfer the desired files to the backup device (such as magtape or cassette).
6. Remount the system disk.
7. Type CTRL C to return to the monitor.

PIP always keeps the USR resident and, with the handler loaded, the system disk is not required if no other devices are referenced.

For example to transfer RT-11 sources from the source disk to TMLl magtape on a single-disk system:

1. Boot system and enter the date.
2. Type: LOAD M<CR>
Response: *
Type: R PIP<CR>
Response: *
3. Dismount system disk, mount source disk.
4. Type: M<CR>
5. When done, remount system disk.

The sources can now be manipulated from magtape.
4.6.15 Reassigning Device Names for RK11 and RF11

Users of other DIGITAL operating systems will notice that RT-11 uses the controller names (RK and RF) rather than the more common user-level names (DK and DF) for these devices. This is due to the fact that RT-11 uses the name DK to refer to the default storage device, which may not necessarily be the RK11.

If you find this situation annoying, the device names can be reassigned with the monitor ASSIGN command, as follows:

\ASSIGN RK1DK<CR>
\ASSIGN RF1DF<CR>

Note, however, that when DK is reassigned in this manner, all default storage goes to the device name DK, and you may not wish to use the physical device RK as the default storage device.

4.6.16 Interfacing a Foreground Terminal

Applications for the F/B Monitor frequently require that the foreground program dialogue appear on a separate terminal, independent of the console terminal.

To facilitate development of these applications, or any others requiring multiple terminals, a source (KB.MAC) for a device-independent terminal handler has been included in the distribution kit. The source is provided for user convenience and can be used as a model for handler development, modified to meet specific needs, or assembled and used as is to provide support for a second terminal.

Documentation for the use of KB.MAC is contained in the comments at the beginning of the source; a listing appears in Appendix B of the RT-11 Software Support Manual.

4.6.17 Modifying the Line Count in MACRO and CREF

RT-11 MACRO and CREF set the number of lines printed per listing page at 60. This line count is satisfactory for applications with line printers that use paper 10.5 inches long. Applications that use paper of a different size (e.g., 8.5 inches long) and applications without line printers should modify MACRO and CREF as follows.

For MACRO:

\R PATCH<CR>

PATCH Version number

FILE NAME--
*MACRO.SAV<CR>
*MACR1;OR<CR>
*0,12366/ 74  n<CR>
"E

\R PIP<CR>
"SY:/O<CR>

where n is the new line count specified in octal.
RT-11 SYSTEM CUSTOMIZATION

For CREF:

```
,R PATCH<CR>

PATCH Version number

FILE NAME--
  *CREF.SAV<CR>
  *CREF1;OR<CR>
  *0,3122/  74    n<CR>
  *E

,R PIP<CR>
  *SY:/O<CR>
```

where n is the new line count specified in octal.

NOTE

See RT-11 System Release Notes, Table 2, for the actual addresses of MACR1 and CREF1.

4.6.18 Changing the DUMP Default Output Device

The DUMP utility program uses LP as its default output device. Systems that do not have line printers will want to change the default device, normally to TT.

DUMP.SAV can be easily altered to change the default output device as follows:

1. Patch to change location LP to the .RAD50 code for the new default output device.

2. Patch to change location MSGO+1 to the .ASCII code for the first letter of the new default output device.

3. Patch to change location MSGO+2 to the .ASCII code for the second letter of the new default output device.

For example, to change the DUMP default output device to TT (the console terminal),

```
,R PATCH<CR>

PATCH Version number

FILE NAME--
  *DUMP.SAV<CR>
  *LP/  46600  10040<CR>         [.RAD50 for TT]
  *MSGO+1\114  124<CR>             [.ASCII for T]
  *MSGO+2\120  124<CR>             [.ASCII for T]
  *E

,R PIP<CR>
  *SY:/O<CR>
```
NOTE
See RT-11 System Release Notes, Table 2, for the actual values of LP and MSGO.
See RT-11 System Release Notes, Table 3, for the .ASCII and .RAD50 codes for all RT-11 devices.

4.6.19 Using UNLOAD When a Foreground Job is Running

The keyboard monitor .UNLOAD command cannot be used to unload handlers from a background job when a foreground job is running. The following patch allows users to reclaim the space occupied by .LOADed background job handlers while a foreground job is running. When this patch is made, users have no protection against the possibility of .UNLOADing handlers required by the foreground job. If a handler required by a foreground job is inadvertently .UNLOADed, an ?M-BAD FETCH error on a .FETCH or an ?M-NO DEV error on a .LOOKUP, .ENTER, .READ, or .WRITE will occur.

The correction to the RT-11 V02C Foreground/Background Monitor is:

>_R PATCH<CR>
PATCH Version number
FILE NAME--
*MONITR.SYS/M<CR>
*BASE.OR<CR>
*0,41046/____1403____403<CR>
*E

>_R PIP<CR>
*SY:A=MONITR.SYS/U<CR>
*SY:/O<CR>

NOTE
See RT-11 System Release Notes, Table 2, for the exact address of BASE.

4.7 OPTIMIZING THE SYSTEM DEVICE

When building RT-11 systems, performance can be optimized by proper placement of .SYS files on the system device.

Optimal file placement is:

MONITR.SYS
Most frequently used handler
.
.
Least frequently used handler
SYSMAC.SML (if many assembly operations are performed)
RT-ll SYSTEM CUSTOMIZATION

Most frequently used program

Least frequently used program

Considerations for the above placements are:

1. Positioning the monitor immediately after the directory optimizes device motion during monitor swapping operations.

2. Positioning PIP.SAV immediately after the monitor ensures that it will not move when the device is compressed or files are deleted.

3. Positioning the handlers and programs in descending order related to frequency of usage reduces the access times for those files.

4. In systems that will be used for frequent assembly operations, placing SYSMAC.SML near the beginning of the device improves assembler performance.

Diskette and DECTape users can also conserve time and space by placing only those files needed on the system disk or DECTape. Users of 8K systems need not place files such as MACRO.SAV, SYSMAC.SML, CREF.SAV and BA.SYS on the system device, since these files cannot be used in 8K systems.

4.8 SWITCHING BETWEEN SINGLE-JOB AND FOREGROUND/BACKGROUND MONITORS

For an application that requires frequent switching between the F/B and Single-Job Monitors, use the following procedure:

1. Both monitors reside on the same volume, the one running is named MONITRSYS and the other is called xyyyyyy.SYS. (The actual name xyyyyyy is not significant.)

2. When a change-over is desired, PIP is used to:
   a. Preserve the running monitor by renaming it to yyyyyy.SYS
   b. Rename the desired monitor to MONITRSYS
   c. Write the new bootstrap from the new MONITRSYS file
   d. Reboot the system

For example, assume an RTll system is running the Single-Job Monitor; the Foreground/Background Monitor on the system is named RKMNFBSYS. The following commands are used to switch from Single-Job to Foreground/Background:

```
DR PIP<CR>
@DMNSJ,SYSS=MONITRSYS/Y/R<CR>
FRBRTOS<CR>
@MONITRSYS=DMNFBSYS/Y/R<CR>
@FRBRTOS<CR>
@MX=MONITRSYS/U<CR>
@MX=SYS/U<CR>
```

Preserve the S/J Monitor.

Activate the F/B Monitor by renaming it to MONITRSYS.

Write the new bootstrap.

Reboot the system.

4-34
The following is a faster method of switching between monitors; this method may be used with all system devices except RXll diskette and DECTape. This method presents no danger if the user forgets to rename the monitor or does the rename in the wrong order.

This example shows the switching of monitors on an RKll system currently running the foreground/background monitor. Before using this method, ensure that both monitor files are present on the disk and named RKMNSJ.SYS and RKMNF.B.SYS. The monitor file MONITR.SYS is also present.

```
R PIP<CR>
#MONITR,SYS=RKMNSJ,SYS/X/Y<CR>
?REBOOT?
#A=MONITR,SYS/U<CR>
#RK1/O<CR>
```

Create a copy of the RKll S/J Monitor.

Write the new bootstrap.

Reboot the system.
CHAPTER 5
ASSEMBLY AND LINK INSTRUCTIONS

5.1 GENERAL INSTRUCTIONS

All RT-11 components, except MACRO, ASEMBl, and the monitors, require 16K words of memory to be assembled. MACRO and ASEMBl require 20K words; the monitors require 24K words. RT-11 MACRO is used as the assembler, and RT-11 LINK is used as the Linker in all cases. All assemblies (except ODT) and all links (except where otherwise noted) should be error free.

Throughout this chapter, the following conventions are used:

1. Default extensions are not explicitly specified. For all the source files, the extensions are .MAC. The assembler output is .OBJ and Linker output is .SAV.

2. The system macro library, SYSMAC.SML, must be on the system device during all assemblies given below.

3. In the example command strings, the sources are kept on a logical device SRC:, binary output is to device BIN:, and listing and map files are output to LST:. In actual practice, any appropriate device can be used. The assembly and link operations were run as a BATCH stream; command lines in the following sections were taken from the output of the BATCH stream, which accounts for lack of prompting asterisks and periods in some cases.

4. The example command strings were executed on a 28K computer and the FREE CORE error messages reflect that fact. The actual number of free memory words in each V02C installation will vary, and is not important.

All RT-11 system assembling and linking operations are normal operations, and the command strings in the descriptions below can be altered to take full advantage of all RT-11 MACRO and LINK command features.

5.2 ASSEMBLING AND LINKING THE SYSTEM FILES

The result of the operations below is the 12 monitors and 14 handler files. (The BATCH run-time handler is assembled and linked in Section 5.3.14.) The UNDEF GLBLS messages resulting from RK.SYS, RF.SYS, DX.SYS, DS.SYS, DP.SYS, and DT.SYS linking are expected. The reason for this undefined global is to prevent the accidental linking of the
wrong bootstrap and system devices when linking a monitor. An
undefined global will result if, e.g., an RF boot is linked with an RK
driver. The monitor files are named as described in Table 1-1.

R MACRO
*BIN;RT11SJ,LST;RT11SJ=SRC;KMON,USR,RMONSJ,KMOVLY/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 6144, WORDS

*BIN;RT11FB,LST;RT11FB=SRC;BFDEF,KMON,USR,RMONFB,KMOVLY/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 4329, WORDS

*BIN;RKBTSJ,LST;RKBTSJ=SRC;BSTRAP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 12016, WORDS

*BIN;RFBTSJ,LST;RFBTSJ=SRC;RFSYS,BSTRAP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 12008, WORDS

*BIN;DTBTSJ,LST;DTBTSJ=SRC;DTSYS,BSTRAP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11979, WORDS

*BIN;DPBTSJ,LST;DPBTSJ=SRC;DPSYS,BSTRAP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11936, WORDS

*BIN;DDBTSJ,LST;DDBTSJ=SRC;DSSYS,BSTRAP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 12024, WORDS

*BIN;RKBTFB,LST;RKBTFB=SRC;BFDEF,BSTRAP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11988, WORDS

*BIN;RFBTFB,LST;RFBTFB=SRC;RFSYS,BFDEF,BSTRAP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11980, WORDS

*BIN;DTBTFB,LST;DTBTFB=SRC;DTSYS,BFDEF,BSTRAP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11972, WORDS

*BIN;DPBTFB,LST;DPBTFB=SRC;DPSYS,BFDEF,BSTRAP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11951, WORDS

*BIN;DDBTFB,LST;DDBTFB=SRC;DSSYS,BFDEF,BSTRAP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11908, WORDS

*BIN;DSBTFB,LST;DSBTFB=SRC;DSSYS,BFDEF,BSTRAP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11996, WORDS
ASSEMBLY AND LINK INSTRUCTIONS

*BINiDP, LSTiDP = SRCiDP/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 12824, WORDS

*BINiDX, LSTiDX = SRCiDX/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 12788, WORDS

*BINiRK, LSTiRK = SRCiRK/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 12965, WORDS

*BINiRF, LSTiRF = SRCiRF/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 12966, WORDS

*BINiDT, LSTiDT = SRCiDT/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 12968, WORDS

*BINiTT, LSTiTT = SRCiTT/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 12927, WORDS

*BINiLP, LSTiLP = SRCiLP/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 12945, WORDS

*BINiPR, LSTiPR = SRCiPR/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 13047, WORDS

*BINiPP, LSTiPP = SRCiPP/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 13044, WORDS

*BINiCR, LSTiCR = SRCiCR/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 12581, WORDS

*BINiMT, LSTiMT = SRCiMT/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 11869, WORDS

*BINiMM, LSTiMM = SRCiTUDEF, MT/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 11473, WORDS

*BINiCT, LSTiCT = SRCiCT/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 11970, WORDS

*BINiDS, LSTiDS = SRCiDS/C/NiTTMiCND
ERRORS DETECTED: 0
FREE CORE: 12992, WORDS
R LINK
BINIRKMNFB, SYS, LSTIRKMNFB=BINIRKBTFB, RT11FB, RK
BINIRKMSJ, SYS, LSTIRKMSJ=BINIRKBTSJ, RT11SJ, RK
BINIRFMNFBS, SYS, LSTIRFMMFB=BINIRFBTFB, RT11FB, RF
BINIRFMNSJ, SYS, LSTIRFMNSJ=BINIRFBTSJ, RT11SJ, RF
BINIDTMNFB, SYS, LSTIDTMNFBS=BINIDTBTFB, RT11FB, DT
BINIDTMNSJ, SYS, LSTIDTMNSJ=BINIDTBTSJ, RT11SJ, DT
BINIDPMNFBS, SYS, LSTIDPMNFBS=BINIDPBTFB, RT11FB, DP
BINIDPNSJ, SYS, LSTIDPMNSJ=BINIDPBTSJ, RT11SJ, DP
BINIDXMNFBS, SYS, LSTIDXMNFBS=BINIDXBTFB, RT11FB, DX
BINIDXNSJ, SYS, LSTIDXNSJ=BINIDXTSJ, RT11SJ, DX
BINIDSMNFBS, SYS, LSTIDSMNFBS=BINIDSTFB, RT11FB, DS
BINIDSMNSJ, SYS, LSTIDSMNSJ=BINIDSBTSJ, RT11SJ, DS
BINIDP, SYS, LSTIDP=BINIDP
UNDEF GLBLS
BINIDX, SYS, LSTD1X=BINIDX
UNDEF GLBLS
BINIRK, SYS, LSTIRK=BINIRK
UNDEF GLBLS
BINIRF, SYS, LSTIRF=BINIRF
UNDEF GLBLS
BINIDT, SYS, LSTIDT=BINIDT
UNDEF GLBLS
BINIDT, SYS, LSTIDT=BINIDT
UNDEF GLBLS
BINITT, SYS, LSTITT=BINITT
BINILP, SYS, LSTILP=BINILP
ASSEMBLY AND LINK INSTRUCTIONS

BIN:PR, SYS, LST:PR=BIN:PR
BIN:PP, SYS, LST:PP=BIN:PP
BIN:MT, SYS, LST:MT=BIN:MT
BIN:MM, SYS, LST:MM=BIN:MM
BIN:CT, SYS, LST:CT=BIN:CT
BIN:DS, SYS, LST:DS=BIN:DS
UNDEF GLBL

5.3 ASSEMBLING AND LINKING THE UTILITIES

5.3.1 EDIT

R MACRO
*BIN:VTCED1, LST:VTCED1=SRC:EDITDF, VTCAL1/C/N:TTM:CND
ERRORS DETECTED: 0
FREE CORE: 15083, WORDS

ERRORS DETECTED: 0
FREE CORE: 14592, WORDS

*BIN:VTBEDT, LST:VTBEDT=SRC:EDITDF, VTBASE/C/N:TTM:CND
ERRORS DETECTED: 0
FREE CORE: 14594, WORDS

*BIN:EDIT, LST:EDIT=SRC:VTMAC, EDIT/C/N:TTM:CND
ERRORS DETECTED: 0
FREE CORE: 11198, WORDS

R LINK
BIN:EDIT, LST:EDIT=BIN:VTCED1, VTCED4, VTBEDT, EDIT

5.3.2 MACRO

R MACRO
*BIN:RTEXEC, LST:RTEXEC=SRC:RTPAR, RPARAM, RCIOCH, RTEXEC/C/N:TTM:CND
ERRORS DETECTED: 0
FREE CORE: 12182, WORDS

ERRORS DETECTED: 0
FREE CORE: 7603, WORDS

5-5
ASSEMBLY AND LINK INSTRUCTIONS

*BIN;RTYPE,LST;RTYPE=SRC;RTYPE=PAR;RST/NTTM;CND/C
ERRORS DETECTED: 0
FREE CORE: 14687, WORDS

R LINK
BIN;MACRO,LST;MACRO=BIN;RTEXEC,RTMAC,RTYPE

5.3.3 EXPAND

R MACRO
*BIN;RTEXEC,LST;RTEXEC=SRC;RTEXEC=PAR;PPARAM,PCIOCH,RTEXEC/C/NTTM;CND
ERRORS DETECTED: 0
FREE CORE: 13586, WORDS

*BIN;RPASS,LST;RPASS=SRC;RPASS=PAR;PPARAM,PCIOCH,RPASS/C/NTTM;CND
ERRORS DETECTED: 0
FREE CORE: 13322, WORDS

R LINK
BIN;EXPAND,LST;EXPAND=BIN;RTEXEC,RPASS

5.3.4 AEMBL

R MACRO
*BIN;RTEXEC,LST;RTEXEC=SRC;SMEXEC,PARAM,RCIOCH,RTEXEC/C/NTTM;CND
ERRORS DETECTED: 0
FREE CORE: 12424, WORDS

*BIN;SMMAC,LST;SMMAC=SRC;SMMAC=PARAM,RCIOCH,MACRO3,MACRO5/C/NTTM;CND
ERRORS DETECTED: 0
FREE CORE: 13590, WORDS

*BIN;SMST,LST;SMST=SRC;SMST=PARAM,PS/C/NTTM;CND
ERRORS DETECTED: 0
FREE CORE: 14683, WORDS

R LINK
BIN;AEMBL,LST;AEMBL=BIN;RTEXEC,SMMAC,SMST

5.3.5 CRF

R MACRO
*BIN;CREF,LST;CREF=SRC;CREF/C/NTTM;CND
ERRORS DETECTED: 0
FREE CORE: 13221, WORDS

R LINK
BIN;CREF,LST;CREF=BIN;CREF
5.3.6 LINK

R MACRO
*BINILINK0, LSTILINK0=SRCILINK0/C/N1TM1CND ERRORS DETECTED: 0 FREE CORE: 12433, WORDS
*BINILINK4, LSTILINK4=SRCILINK4/C/N1TM1CND ERRORS DETECTED: 0 FREE CORE: 12896, WORDS
*BINILINKV1, LSTILINKV1=SRCILINKV1/C/N1TM1CND ERRORS DETECTED: 0 FREE CORE: 13685, WORDS
*BINILNKV2, LSTILINKV2=SRCILNKV2/C/N1TM1CND ERRORS DETECTED: 0 FREE CORE: 13635, WORDS
*BINILNKV3, LSTILINKV3=SRCILINKV3/C/N1TM1CND ERRORS DETECTED: 0 FREE CORE: 14017, WORDS
*BINILNKV4, LSTILINKV4=SRCILINKV4/C/N1TM1CND ERRORS DETECTED: 0 FREE CORE: 13870, WORDS
*BINILNKV5, LSTILINKV5=SRCILINKV5/C/N1TM1CND ERRORS DETECTED: 0 FREE CORE: 13899, WORDS

R LINK
BINILINK, LSTILINK=BINILINK0/C/B1500
BINILINK0/O11/C
BINILINK1/O11/C
BINILINK2/O11/C
BINILINK3/O11/C
BINILINK4/O11/C
BINILINK5/C11

5.3.7 LIBR

R MACRO
*BINILIBR0, LSTILIBR0=SRCILIBR0/C/N1TM1CND ERRORS DETECTED: 0 FREE CORE: 12853, WORDS
*BINILIBR1, LSTILIBR1=SRCILIBR1/C/N1TM1CND ERRORS DETECTED: 0 FREE CORE: 12992, WORDS
*BINILIBR2, LSTILIBR2=SRCILIBR2/C/N1TM1CND ERRORS DETECTED: 0 FREE CORE: 13200, WORDS
*BINILIBR3, LSTILIBR3=SRCILIBR3/C/N1TM1CND ERRORS DETECTED: 0 FREE CORE: 13449, WORDS

5-7 July 1976
ASSEMBLY AND LINK INSTRUCTIONS

*BINLIB4,LSTLIB4=SRC4LIB4/C/NTM1CN
ERRORS DETECTED: 0
FREE CORE: 13365, WORDS

R LINK
BINLIB4,LSTLIB4=BINLIB4/C
BINLIB1/011/C
BINLIB2/011/C
BINLIB3/011/C
BINLIB4/011

5.3.8 PIP

R MACRO
*BINPIP,LSTPIP=SRC1PIP/C/NTM1CN
ERRORS DETECTED: 0
FREE CORE: 11769, WORDS

*BIN1PIP1,LST1PIP1=SRC1PIP1/C/NTM1CN
ERRORS DETECTED: 0
FREE CORE:

R LINK
BIN1PIP,LST1PIP=BIN1PIP/C
BIN1PIP1/011

5.3.9 FILEX

R MACRO
*BIN:FILEX,LST:FILEX=SRC:FILEX/C/NTM1CN
ERRORS DETECTED: 0
FREE CORE: 11391, WORDS

R LINK
BIN:FILEX,LST:FILEX=BIN:FILEX

5.3.10 SRCCOM

R MACRO
*BIN:SRCCOM,LST:SRCCOM=SRC:SRCCOM/C/NTM1CN
ERRORS DETECTED: 0
FREE CORE: 13698, WORDS

R LINK
BIN:SRCCOM,LST:SRCCOM=BIN:SRCCOM
5.3.11  DUMP

R MACRO
*BIN=DUMP, LST=DUMP=SRC=DUMP/C/NI/TTM1/CND
ERRORS DETECTED: 0
FREE CORE: 13986, WORDS

R LINK
BIN=DUMP, LST=DUMP=BIN=DUMP

5.3.12  PATCH

R MACRO
*BIN=PATCH, LST=PATCH=SRC=PATCH/C/NI/TTM1/CND
ERRORS DETECTED: 0
FREE CORE: 13989, WORDS

R LINK
BIN=PATCH, LST=PATCH=BIN=PATCH

5.3.13  ODT

ODT assembles with one error (a Z error which flags an ODT instruction
that is machine dependent). The error is necessary and should be
ignored.

R MACRO
*BIN=ODT, LST=ODT=SRC=ODT/C/NI/TTM1/CND
ERRORS DETECTED: 1
FREE CORE: 14159, WORDS

5.3.14  BATCH

R MACRO
*BIN=BA, LST=BA=SRC=BA/C/NI/TTM1/CND
ERRORS DETECTED: 0
FREE CORE: 12238, WORDS

*BIN=BATCH, LST=BATCH=SRC=BATCH/C/NI/TTM1/CND
ERRORS DETECTED: 0
FREE CORE: 5551, WORDS

R LINK
BIN=BA, SYS=LST=BA=BIN=BA

BIN=BATCH, LST=BATCH=BIN=BATCH
5.4 COMPILING AND LINKING PATCHO

PATCHO is written in FORTRAN, and requires RT-11 FORTRAN IV for compilation and linking. The warning messages should be ignored.

```
R FORTRA
*BINIPAT0;LSTIPAT0=SRCIPAT2/S/P/N15/R120
[,...MAIN,] ERRORS 000, WARNINGS 002
*BINIPAT1;LSTIPAT1=SRCIPAT1/S/P
*BINIPAT2;LSTIPAT2=SRCIPAT2/S/P
*BINIPAT3;LSTIPAT3=SRCIPAT3/S/P
[LINBLK ] ERRORS 000, WARNINGS 002
*BINIPAT4;LSTIPAT4=SRCIPAT4/S/P
[LIST ] ERRORS 000, WARNINGS 004
*BINIPAT5;LSTIPAT5=SRCIPAT5/S/P
*BINIPAT6;LSTIPAT6=SRCIPAT6/S/P
[DCUMP ] ERRORS 000, WARNINGS 003

R MACRO
*BINIRAD50;LSTIRAD50=SRClIRAD50/C/NITTMICN0
ERRORS DETECTED 0
FREE CORE 15148, WORDS

*BINIR50ASC;LSTIR50ASC=SRCIR50ASC/C/NITTMICN0
ERRORS DETECTED 0
FREE CORE 15149, WORDS

*BINICMFCV;LSTICMFCV=SRCICMFCV/C/NITTMICN0
ERRORS DETECTED 0
FREE CORE 15142, WORDS
```

The following linking instructions require that the FORTRAN library, FORLIB.OBJ, be available on the system device. FORLIB.OBJ is available as part of the RT-11 FORTRAN software kit. The UNDEF GLBLS message from the link should be ignored.

```
R LINK
BINIFATCHC;LSTIPATCHO=BINIPAT0;CMFCV/F/I/C
BINIPAT1;IRAD50/O11/C
BINIPAT3;R50ASC/O11/C
BINIPAT2/O12/C
BINIPAT4/O12/C
BINIPAT5/O12/C
BINIPAT6/O12

LIBRARY SEARCH:
$SHOMET
UNDEF:GLBLS
```
ASSEMBLY AND LINK INSTRUCTIONS

5.5  ASSEMBLING AND BUILDING THE VT11 DISPLAY HANDLER LIBRARY (VTLIB)

R MACRO
*BIN*VTCA11, LST*VTCA11=SRC*VTCA11/C/N11TM1CND
ERRORS DETECTED: 0
FREE CORE: 15087, WORDS

*BIN*VTCA2L, LST*VTCA2L=SRC*VTCA2L/C/N11TM1CND
ERRORS DETECTED: 0
FREE CORE: 15151, WORDS

*BIN*VTCA3L, LST*VTCA3L=SRC*VTCA3L/C/N11TM1CND
ERRORS DETECTED: 0
FREE CORE: 15122, WORDS

*BIN*VTCA4L, LST*VTCA4L=SRC*VTCA4L/C/N11TM1CND
ERRORS DETECTED: 0
FREE CORE: 14612, WORDS

*BIN*VTBASEL, LST*VTBASEL=SRC*VTBASEL/C/N11TM1CND
ERRORS DETECTED: 0
FREE CORE: 14784, WORDS

R PIP
BIN*VTHDLRL, OBJ=BIN*VTCA1L, OBJ, VTCA2L, OBJ, VTCA3L, OBJ/B
BIN*VTHDLRL, OBJ=BIN*VTHDLRL, OBJ, VTCA4L, OBJ, VTBASEL, OBJ/B

R LIBR
BIN*VTLIBL=BIN*VTHDLRL

5.6  ASSEMBLING AND BUILDING THE SYSTEM SUBROUTINE LIBRARY (SYSLIB)

R MACRO
*BIN*LEN, SBJ, LST*LEN=SRC*LEN/C/N11TM1CND
ERRORS DETECTED: 0
FREE CORE: 15168, WORDS

*BIN*TRIM, SBJ, LST*TRIM=SRC*TRIM/C/N11TM1CND
ERRORS DETECTED: 0
FREE CORE: 15176, WORDS

*BIN*STRPAD, SBJ, LST*STRPAD=SRC*STRPAD/C/N11TM1CND
ERRORS DETECTED: 0
FREE CORE: 15161, WORDS

*BIN*VERIFY, SBJ, LST*VERIFY=SRC*VERIFY/C/N11TM1CND
ERRORS DETECTED: 0
FREE CORE: 15112, WORDS

*BIN*INSERT, SBJ, LST*INSERT=SRC*INSERT/C/N11TM1CND
ERRORS DETECTED: 0
FREE CORE: 15159, WORDS

*BIN*CONCAT, SBJ, LST*CONCAT=SRC*CONCAT/C/N11TM1CND
ERRORS DETECTED: 0
FREE CORE: 15149, WORDS

*BIN*REPEAT, SBJ, LST*REPEAT=SRC*REPEAT/C/N11TM1CND
ERRORS DETECTED: 0
FREE CORE: 15161, WORDS

5-11
ASSEMBLY AND LINK INSTRUCTIONS

*BINLIAADDR,SBJ,LSTLIAADDR=SRC1IAADDR/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15183, WORDS

*BINLINDEX,SBJ,LSTINDEX=SRC1INDEX/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15157, WORDS

*BINLTRANS'L,SBJ,LSTTRANS'L=SRC1TRANS'L/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15163, WORDS

*BINLSCOMP,SBJ,LSTSCOMP=SRC1SCOMP/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15155, WORDS

*BINLSUBSTR,SBJ,LSTSUBSTR=SRC1SUBSTR/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15163, WORDS

*BINLSCOPY,SBJ,LSTSCOPY=SRC1SCOPY/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15169, WORDS

*BINLCLOSEC,SBJ,LSTCLOSEC=SRC1CLOSEC/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15182, WORDS

*BINLICMKT,SBJ,LSTICMKT=SRC1ICMKT/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15182, WORDS

*BINLICSTAT,SBJ,LSTICSTAT=SRC1ICSTAT/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15182, WORDS

*BINLGTIM,SBJ,LSTGTIM=SRC1GTIM/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15182, WORDS

*BINLIDELET,SBJ,LSTIDELET=SRC1IDELET/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15175, WORDS

*BINLIDSTAT,SBJ,LSTITSTAT=SRC1IDSTAT/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15191, WORDS

*BINLIENTER,SBJ,LSTIENTER=SRC1IENTER/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15174, WORDS

*BINLIFREEC,SBJ,LSTITFREEC=SRC1IFREEC/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15167, WORDS

*BINLIGETC,SBJ,LSTIIGETC=SRC1IGETC/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15161, WORDS

*BINLTJB,SBJ,LSTGTJB=SRC1GTJB/C/N1TTM1CN.D
ERRORS DETECTED: 0
FREE CORE: 15182, WORDS
ASSEMBLY AND LINK INSTRUCTIONS

*BIN\LOOKUP, SBJ, LST*LOOKUP=SRC\LOOKUP/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15174, WORDS

*BIN\WAIT, SBJ, LST*WAIT=SRC\WAIT/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15162, WORDS

*BIN\PRINT, SBJ, LST*PRINT=SRC\PRINT/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15163, WORDS

*BIN\PURGE, SBJ, LST*PURGE=SRC\PURGE/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15182, WORDS

*BIN\CTRL, SBJ, LST*CTRL=SRC\CTRL/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15182, WORDS

*BIN\REOPEN, SBJ, LST*REOPEN=SRC\REOPEN/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15182, WORDS

*BIN\RESUME, SBJ, LST*RESUME=SRC\RESUME/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15182, WORDS

*BIN\ISAVES, SBJ, LST*ISAVES=SRC\ISAVES/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15178, WORDS

*BIN\SUSPEND, SBJ, LST*SUSPEND=SRC\SUSPEND/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15182, WORDS

*BIN\ITLOCK, SBJ, LST*ITLOCK=SRC\ITLOCK/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15182, WORDS

*BIN\ITINR, SBJ, LST*ITINR=SRC\ITINR/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15182, WORDS

*BIN\ITOUR, SBJ, LST*ITOUR=SRC\ITOUR/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15182, WORDS

*BIN\ITWAIT, SBJ, LST*ITWAIT=SRC\ITWAIT/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15183, WORDS

*BIN\UNLOCK, SBJ, LST*UNLOCK=SRC\UNLOCK/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15192, WORDS

*BIN\WAIT, SBJ, LST*WAIT=SRC\WAIT/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15179, WORDS

*BIN\LOCK, SBJ, LST*LOCK=SRC\LOCK/C\NITM\CND
ERRORS DETECTED: 0
FREE CORE: 15192, WORDS

5-13
ASSEMBLY AND LINK INSTRUCTIONS

*BIN: TIMSUB, SBJ, LST: TIMSUB=SRC: TIMSUB/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15164, WORDS

*BIN: ISLEEP, SBJ, LST: ISLEEP=SRC: ISLEEP/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15178, WORDS

*BIN: JTIME, SBJ, LST: JTIME=SRC: JTIME/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15181, WORDS

*BIN: IUNTIL, SBJ, LST: IUNTIL=SRC: IUNTIL/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15186, WORDS

*BIN: IREAD, SBJ, LST: IREAD=SRC: IREAD/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15166, WORDS

*BIN: ISDAT, SBJ, LST: ISDAT=SRC: ISDAT/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15170, WORDS

*BIN: ISPYN, SBJ, LST: ISPYN=SRC: ISPYN/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15153, WORDS

*BIN: IRITES, SBJ, LST: IRITES=SRC: IRITES/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15162, WORDS

*BIN: IRCVD, SBJ, LST: IRCVD=SRC: IRCVD/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15170, WORDS

*BIN: CVTTIM, SBJ, LST: CVTTIM=SRC: CVTTIM/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15177, WORDS

*BIN: DIV0, SBJ, LST: DIV0=SRC: DIV0/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15164, WORDS

*BIN: R50ASC, SBJ, LST: R50ASC=SRC: R50ASC/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15145, WORDS

*BIN: ISPY, SBJ, LST: ISPY=SRC: ISPY/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15179, WORDS

*BIN: TIMASC, SBJ, LST: TIMASC=SRC: TIMASC/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15165, WORDS

*BIN: IFETCH, SBJ, LST: IFETCH=SRC: IFETCH/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15155, WORDS

*BIN: IQSET, SBJ, LST: IQSET=SRC: IQSET/C/N: TTM: CND
ERRORS DETECTED: 0
FREE CORE: 15167, WORDS

5-14
ASSEMBLY AND LINK INSTRUCTIONS

*BIN:JICVT,SBJ,LST:JICVT=SRC:JICVT/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15181, WORDS

*BIN:JICVT,SBJ,LST:JICVT=SRC:JICVT/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15161, WORDS

*BIN:JFCX,SBJ,LST:JFCX=SRC:JFCX/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15153, WORDS

*BIN:JFLT,SBJ,LST:JFLT=SRC:JFLT/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15137, WORDS

*BIN:JMOV,SBJ,LST:JMOV=SRC:JMOV/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15175, WORDS

*BIN:IREADF,SBJ,LST:IREADF=SRC:IREADF/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15177, WORDS

*BIN:CMPLT,SBJ,LST:CMPLT=SRC:CMPLT/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 14955, WORDS

*BIN:IRAD50,SBJ,LST:IRAD50=SRC:IRAD50/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15148, WORDS

*BIN:CHAIN,SBJ,LST:CHAIN=SRC:CHAIN/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15174, WORDS

*BIN:RCHAIN,SBJ,LST:RCHAIN=SRC:RCHAIN/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15175, WORDS

*BIN:SECNDS,SBJ,LST:SECNDS=SRC:SECNDS/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15170, WORDS

*BIN:JCMPP,SBJ,LST:JCMPP=SRC:JCMPP/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15167, WORDS

*BIN:ILUN,SBJ,LST:ILUN=SRC:ILUN/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15159, WORDS

*BIN:IASIGN,SBJ,LST:IASIGN=SRC:IASIGN/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15092, WORDS

*BIN:TIME,SBJ,LST:TIME=SRC:TIME/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15178, WORDS

*BIN:JJCVT,SBJ,LST:JJCVT=SRC:JJCVT/C/N:TTM:ICND
ERRORS DETECTED: 0
FREE CORE: 15183, WORDS
5.7 ASSEMBLING AND LINKING MBUILD

R MACRO
*BINIMBOOT, LSTIMBOOT=SRCIMBOOT/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 16435, WORDS

*BINIMSBUILD, LSTIMSBUILD=SRCIMSBUILD/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 16247, WORDS

*BINIMTINIT, LSTIMTINIT=SRCIMTINIT/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 15377, WORDS

*BINIMBUILD, LSTIMBUILD=SRCIMBUILD, PIP/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 13471, WORDS

*BINIMBDL1, LSTIMBDL1=SRCIMBDL1, PIP1/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 12582, WORDS

R LINK
BINIMBOOT, BOT, LSTIMBOOT=BINIMBOOT
BINIMSBUILD, BOY, LSTIMSBUILD=BINIMSBUILD
BINIMTINIT, LSTIMTINIT=BINIMTINIT
BINIMBUILD, LSTIMBUILD=BINIMBDL1, MBUILD

5.8 ASSEMBLING AND LINKING THE FORTRAN IV COMPILER

This section provides assembly and linking instructions for the RT-11 FORTRAN Compiler. This information applies only to those users who received the source versions of the Compiler.
ASSEMBLY AND LINK INSTRUCTIONS

All assemblies listed in this section were actually run as a BATCH stream under the RT-ll BATCH Compiler. BIN, LST, and SRC are logical device names used for the binary output device, listing output device, and source input device, respectively. The user can employ any legal MACRO assembly command that suits the need; if no listing is desired, the list file specification may be omitted entirely.

All assemblies require that the system macro file SYSMAC.SML be present on the system device.

5.8.1 Compiler Assembly

Below is an example of the Compiler assembly procedure.

$JOB/TIME/RT11
$RT11
00148116
TTY10

R MACRO
*BINIFROOT, LSTIFROOT*SRCIFORTHD, FROOT/C/NCTTIMCND
ERRORS DETECTED: 0
FREE CORE: 14486, WORDS

*BINIF0, LSTIF0*SRCIFORTHD, F0/C/NCTTIMCND
ERRORS DETECTED: 0
FREE CORE: 13057, WORDS

*BINIF1, LSTIF1*SRCIFORTHD, F1, F1S/C/NCTTIMCND
ERRORS DETECTED: 0
FREE CORE: 13507, WORDS

*BINIF2, LSTIF2*SRCIFORTHD, F2/C/NCTTIMCND
ERRORS DETECTED: 0
FREE CORE: 13594, WORDS

*BINIF3, LSTIF3*SRCIFORTHD, F3, F3S/C/NCTTIMCND
ERRORS DETECTED: 0
FREE CORE: 13502, WORDS

*BINIF4, LSTIF4*SRCIFORTHD, F4, F4S/C/NCTTIMCND
ERRORS DETECTED: 0
FREE CORE: 13479, WORDS

*BINIF5, LSTIF5*SRCIFORTHD, F5, F5S/C/NCTTIMCND
ERRORS DETECTED: 0
FREE CORE: 13454, WORDS

*BINIF6, LSTIF6*SRCIFORTHD, F6/C/NCTTIMCND
ERRORS DETECTED: 0
FREE CORE: 13741, WORDS

*BINIF7, LSTIF7*SRCIF7/C/NCTTIMCND
ERRORS DETECTED: 0
FREE CORE: 13624, WORDS

*BINIF8, LSTIF8*SRCIF8/C/NCTTIMCND
ERRORS DETECTED: 0
FREE CORE: 13525, WORDS
5.8.2 Compiler Linking

Below is an example of the Compiler linking procedures.

$JOB/TIME/RT11
SR11
0110855
TTY10
R LINK
BINIFORTRAN,LSTFORTRAN=BINIFROOT/C
BINIF0/011/C
BINIF1/011/C
BINIF2/011/C
BINIF3/011/C
BINIF4/011/C
BINIF5/011/C

5-19
ASSEMBLY AND LINK INSTRUCTIONS

BINIF6/011/C
BINIF7/011/C
BINIF8/011/C
BINIF9/011/C
BINIF10/011/C
BINIF11/011/C
BINIF12/011/C
BINIF13/011/C
BINIF14/011/C
BINIF15/011/C
BINIF16/011/C
BINIF17/011

$EOJ

TIME
01119129

5.9 ASSEMBLING COMMON FORTRAN OTS MODULES

This section provides assembly instructions for the RT-ll FORTRAN Object Time System (OTS) library modules that are hardware independent and are therefore common to all OTS libraries. This information applies only to those users who received the source versions of OTS.

Below is an example of the assembly procedures for the common OTS library modules.

$JOB/TIME/RT11

$RT11

23155153

TTYIO

R PIP
BINIF MAC=SRC1V2DEF,PRE,FINIT,PRE,OTSH,PRE,FBLOCK,PRE,ERRORS,PRE/A

R MACHO
*BINIABS,ALL,LST:ABS=BINIF,SRCLABS/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINIABS,ALL,LST:ABS=BINIF,SRCLABS/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11189, WORDS

*BINICABS,ALL,LST:ICABS=BINIF,SRCLICABS/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINIFLOAT,ALL,LST:FLOAT=BINIF,SRCLFLOAT/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11385, WORDS

5-20
ASSEMBLY AND LINK INSTRUCTIONS

*BINIDIM, ALL, LSTIDIM=BINIF, SRCIDIM/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11381, WORDS

*BINIDIM, ALL, LSTIDIM=BINIF, SRCIDIM/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11381, WORDS

*BINICEXP, ALL, LSTICEXP=BINIF, SRCICEXP/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11357, WORDS

*BINICSIN, ALL, LSTICSIN=BINIF, SRCICSIN/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11309, WORDS

*BINITANH, ALL, LSTITANH=BINIF, SRCITANH/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11301, WORDS

*BINICONJG, ALL, LSTICONJG=BINIF, SRCICONJG/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11393, WORDS

*BINIVFIX, ALL, LSTIVFIX=BINIF, SRCIVFIX/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11385, WORDS

*BINIDBLLE, ALL, LSTIDBLE=BINIF, SRCIDBLE/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINIREAL, ALL, LSTIREAL=BINIF, SRCIREAL/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINIAIMAG, ALL, LSTIAIMAG=BINIF, SRCIAIMAG/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINICMPLX, ALL, LSTICMPLX=BINIF, SRCICMPLX/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINIINT, ALL, LSTIINT=BINIF, SRCIINT/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11385, WORDS

*BINIAMOD, ALL, LSTIAMOD=BINIF, SRCIAMOD/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11361, WORDS

*BINIMOD, ALL, LSTIMOD=BINIF, SRCIMOD/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11361, WORDS

*BINIMOD, ALL, LSTIMOD=BINIF, SRCIMOD/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11361, WORDS

*BINIMAX0, ALL, LSTIMAX0=BINIF, SRCIMAX0/C/N1TTMICN
ERRORS DETECTED: 0
FREE CORE: 11385, WORDS
ASSEMBLY AND LINK INSTRUCTIONS

*BINIAMN0, ALL, LSTIAMN0 = BINIF, SRCIAMN0/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11357, WORDS

*BINIMN0, ALL, LSTIMN0 = BINIF, SRCIMIN0/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11385, WORDS

*BINIDI MIN1, ALL, LSTIDMIN1 = BINIF, SRCIDMIN1/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11358, WORDS

*BINISIGN, ALL, LSTISIGN = BINIF, SRCISIGN/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINISIGN, ALL, LSTISIGN = BINIF, SRCISIGN/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11385, WORDS

*BINIDSIGN, ALL, LSTIDSIGN = BINIF, SRCIDSIGN/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINICSORT, ALL, LSTICSORT = BINIF, SRCICSORT/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11149, WORDS

*BINISNGL, ALL, LSTISNGL = BINIF, SRCISNGL/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11185, WORDS

*BINIENDERR, ALL, LSTIENDERR = BINIF, SRCIENDERR/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11029, WORDS

*BINICONV5, ALL, LSTICONV5 = BINIF, SRCICONV5/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11371, WORDS

*BINIWAIT, ALL, LSTIWAIT = BINIF, SRCIWAIT/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11298, WORDS

*BINIEOF, ALL, LSTIEOF = BINIF, SRCIEOF/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 10806, WORDS

*BINIFNEG, ALL, LSTIFNEG = BINIF, SRCIFNEG/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11361, WORDS

*BINIXCI, ALL, LSTIXCI = BINIF, SRCIXCI/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11133, WORDS

*BINIRETD, ALL, LSTIRETD = BINIF, SRCIRETD/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11162, WORDS

*BINIFW, ALL, LSTIFW = BINIF, SRCIFW/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 10809, WORDS
ASSEMBLY AND LINK INSTRUCTIONS

*BIN1IFR, ALL, LST1IFR=BIN1F, SRC1IFR/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 10821, WORDS

*BIN1IBR, ALL, LST1IBR=BIN1F, SRC1IBR/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 11165, WORDS

*BIN1IBW, ALL, LST1IBW=BIN1F, SRC1IBW/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 11165, WORDS

*BIN1FCHNL, ALL, LST1FCHNL=BIN1F, SRC1FCHNL/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 10817, WORDS

*BIN1CMPF, ALL, LST1CMPF=BIN1F, SRC1CMPF/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 11293, WORDS

*BIN1CMPO, ALL, LST1CMPO=BIN1F, SRC1CMPO/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 11295, WORDS

*BIN1INITIO, ALL, LST1INITIO=BIN1F, SRC1INITIO/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 10825, WORDS

*BIN1UIO, ALL, LST1UIO=BIN1F, SRC1UIO/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 10688, WORDS

*BIN1REWIND, ALL, LST1REWIND=BIN1F, SRC1REWIND/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 11019, WORDS

*BIN1DUMPLA, ALL, LST1DUMPLA=BIN1F, SRC1DUMPLA/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 11285, WORDS

*BIN1CLOSE, ALL, LST1CLOSE=BIN1F, SRC1CLOSE/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 10924, WORDS

*BIN1VTRAN, ALL, LST1VTRAN=BIN1F, SRC1VTRAN/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 11129, WORDS

*BIN1GETFIL, ALL, LST1GETFIL=BIN1F, SRC1GETFIL/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 10837, WORDS

*BIN1EOL, ALL, LST1EOL=BIN1F, SRC1EOL/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 11021, WORDS

*BIN1CALL, ALL, LST1CALL=BIN1F, SRC1CALL/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BIN1ISNLN, ALL, LST1ISNLN=BIN1F, SRC1ISNLN/C, N1TTMICN1D
ERRORS DETECTED: 0
FREE CORE: 11164, WORDS
ASSEMBLY AND LINK INSTRUCTIONS

*BINIPMOV#ALL, LST1IPMOV#BIN#F, SRC1IPMOV/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11365, WORDS

*BINIADD#ALL, LST1IADD#BIN#F, SRC1IADD/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11373, WORDS

*BINIPADD#ALL, LST1IPADD#BIN#F, SRC1IPADD/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11377, WORDS

*BINISUB#ALL, LST1ISUB#BIN#F, SRC1ISUB/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11373, WORDS

*BINIPSUB#ALL, LST1IPSUB#BIN#F, SRC1IPSUB/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11377, WORDS

*BININCR#ALL, LST1INCR#BIN#F, SRC1INCR/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11377, WORDS

*BININEG#ALL, LST1INEG#BIN#F, SRC1INEG/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11393, WORDS

*BINITESTS#ALL, LST1TESTS#BIN#F, SRC1TESTS/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11361, WORDS

*BINICMP#ALL, LST1ICMP#BIN#F, SRC1ICMP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11373, WORDS

*BINIPCMPS#ALL, LST1IPCMPS#BIN#F, SRC1IPCMPS/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11377, WORDS

*BINIBRAS#ALL, LST1BRAS#BIN#F, SRC1BRAS/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11362, WORDS

*BININXT#ALL, LST1NXT#BIN#F, SRC1NXT/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11351, WORDS

*BININX#ALL, LST1NX#BIN#F, SRC1NX/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11351, WORDS

*BININXT2#ALL, LST1NXT2#BIN#F, SRC1NXT2/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11351, WORDS

*BININXT3#ALL, LST1NXT3#BIN#F, SRC1NXT3/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11351, WORDS

*BININXT4#ALL, LST1NXT4#BIN#F, SRC1NXT4/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11379, WORDS

*BINIAIF#ALL, LST1IAIF#BIN#F, SRC1IAIF/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11390, WORDS

5-24
ASSEMBLY AND LINK INSTRUCTIONS

*BIN1FM6VR, ALL, LSTIFMOV8*BIN1F, SRC1FM6VR/C, N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11385, WORDS

*BIN1FMV5, ALL, LSTIFMOV5*BIN1F, SRC1FMV5/C, N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BIN1FMV6, ALL, LSTIFMOV6*BIN1F, SRC1FMV6/C, N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BIN1FMV7, ALL, LSTIFMOV7*BIN1F, SRC1FMV7/C, N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BIN1FMV8, ALL, LSTIFMOV8*BIN1F, SRC1FMV8/C, N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11389, WORDS

*BIN1FMV9, ALL, LSTIFMOV9*BIN1F, SRC1FMV9/C, N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11365, WORDS

*BIN1LOADS, ALL, LSTILLOADS*BIN1F, SRC1LOADS/C, N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11367, WORDS

*BIN1IDMOV, ALL, LSTIDMOV*BIN1F, SRC1IDMOV/C, N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11393, WORDS

*BIN1IDMOV1, ALL, LSTIDMOV1*BIN1F, SRC1IDMOV1/C, N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11405, WORDS

*BIN1IDMOV2, ALL, LSTIDMOV2*BIN1F, SRC1IDMOV2/C, N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11406, WORDS

*BIN1IDMOV3, ALL, LSTIDMOV3*BIN1F, SRC1IDMOV3/C, N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11369, WORDS

*BIN1IDMOV4, ALL, LSTIDMOV4*BIN1F, SRC1IDMOV4/C, N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS
*BINIDMOV5,ALL,LSTIDMOV5=BINIF,SRCIDMOV5/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11381, WORDS

*BINIDMOV6,ALL,LSTIDMOV6=BINIF,SRCIDMOV6/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11365, WORDS

*BINIDMOV7,ALL,LSTIDMOV7=BINIF,SRCIDMOV7/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINILMOVS,ALL,LSTILMOVS=BINIF,SRCILMOVS/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11385, WORDS

*BINIBITID,ALL,LSTIBITID=BINIF,SRCIBITID/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11345, WORDS

*BINILTEST,ALL,LSTILTEST=BINIF,SRCILTEST/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11393, WORDS

*BINILNOTS,ALL,LSTILNOTS=BINIF,SRCILNOTS/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11389, WORDS

*BINILCMPSP,ALL,LSTILCMPSP=BINIF,SRCILCMPSP/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11385, WORDS

*BINILCMPSP,ALL,LSTILCMPSP=BINIF,SRCILCMPSP/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11393, WORDS

*BINILPCMPS,ALL,LSTILPCMPS=BINIF,SRCILPCMPS/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINIQVEC,ALL,LSTIQVEC=BINIF,SRCIQVEC/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINIQVCP,ALL,LSTIQVCP=BINIF,SRCIQVCP/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINIQVEC,ALL,LSTIQVEC=BINIF,SRCIQVEC/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINISUBR,ALL,LSTISUBR=BINIF,SRCISUBR/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11161, WORDS

*BINILCMPSI,ALL,LSTILCMPSI=BINIF,SRCILCMPSI/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11401, WORDS

*BINICONV6,ALL,LSTICONV6=BINIF,SRCICONV6/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11378, WORDS
ASSEMBLY AND LINK INSTRUCTIONS

**BINIXFF**, ALL, LSTIXFF=BINIF, SRCIXFF/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 1129, WORDS

**BINIXDD**, ALL, LSTIXDD=BINIF, SRCIXDD/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11199, WORDS

**BINIRETS**, ALL, LSTIRETS=BINIF, SRCIRETS/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11149, WORDS

**BINIAFRET**, ALL, LSTIAFRET=BINIF, SRCIAFRET/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11383, WORDS

**BINIRETDSF**, ALL, LSTIRETDSF=BINIF, SRCIRETDSF/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11179, WORDS

**BINITCMPLY**, ALL, LSTITCMPLY=BINIF, SRCITCMPLY/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11149, WORDS

**BINITIRARY**, ALL, LSTITIRARY=BINIF, SRCTITARY/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11125, WORDS

**BINIFUD**, ALL, LSTIFUD=BINIF, SRCFUD/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11197, WORDS

**BINIIOMVS**, ALL, LSTIIOMVS=BINIF, SRCIOMVS/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11333, WORDS

**BINIIOMVR**, ALL, LSTIIOMVR=BINIF, SRCIOMVR/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11389, WORDS

**BINILMOVPR**, ALL, LSTILMOVPR=BINIF, SRCLMOVPR/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

**BINIGOTO**, ALL, LSTIGOTO=BINIF, SRCIGOTO/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11185, WORDS

**BININPUTREC**, ALL, LSTINPUTREC=BINIF, SRCHIPUTREC/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 10768, WORDS

**BINIRAN**, ALL, LSTIRAN=BINIF, SRCIRAN/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11169, WORDS

**BINIRANDU**, ALL, LSTIRANDU=BINIF, SRCIRANDU/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11169, WORDS

**BINIFIND**, ALL, LSTIFIND=BINIF, SRCHIPFIND/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 10809, WORDS

5-27
ASSEMBLY AND LINK INSTRUCTIONS

*BINOPEN, ALL, LSTOPEN=BIN1F, SRC=OPEN/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 10694, WORDS

*BINVCTRAN, ALL, LSTVCTRAN=BIN1F, SRC=VCTRAN/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 11149, WORDS

*BINCONVI, ALL, LSTCONVI=BIN1F, SRC=CONVI/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 11275, WORDS

*BINCONVL, ALL, LSTCONVL=BIN1F, SRC=CONVL/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 11388, WORDS

*BINIDATE, ALL, LSTIDATE=BIN1F, SRC=IDATE/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 10835, WORDS

*BINIDATE, ALL, LSTIDATE=BIN1F, SRC=DATE/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 10993, WORDS

*BINSETERR, ALL, LSTSETERR=BIN1F, SRC=SETERR/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 10965, WORDS

*BINASSIGN, ALL, LSTASSIGN=BIN1F, SRC=ASSIGN/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 10681, WORDS

*BINIRIO, ALL, LSTIRIO=BIN1F, SRC=IRIO/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 10709, WORDS

*BINCLS, ALL, LSTCLS=BIN1F, SRC=CLS/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 11033, WORDS

*BINAMAX1, ALL, LSTAMAX1=BIN1F, SRC=AMAX1/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 11329, WORDS

*BIN14K, SIZ, LST14K=BIN1F, SRC=14K/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 11406, WORDS

*BIN18K, SIZ, LST18K=BIN1F, SRC=18K/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 11406, WORDS

*BIN12K, SIZ, LST12K=BIN1F, SRC=12K/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 11406, WORDS

*BIN16K, SIZ, LST16K=BIN1F, SRC=16K/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 11406, WORDS

*BIN120K, SIZ, LST120K=BIN1F, SRC=120K/C/NITTMCND
ERRORS DETECTED: 0
FREE CORE: 11406, WORDS

5-28
*BIN124K,SIZ,LST:124K=BINIF,SRC124K/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11406, WORDS

*BIN128K,SIZ,LST:128K=BINIF,SRC128K/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 10765, WORDS

*BINISIMRT,UNI,LST:1SIMRT=BINIF,SRC:ISIMRT/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 10971, WORDS

*BINIPR,ALL,LST:IPR=BINIF,SRC:IPR/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11397, WORDS

*BINICNEG,ALL,LST:ICNEG=BINIF,SRC:ICNEG/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11377, WORDS

*BINICOPY,ALL,LST:ICOPY=BINIF,SRC:ICOPY/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11389, WORDS

*BINIGETREC,ALL,LST:IGETREC=BINIF,SRC:IGETREC/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 10725, WORDS

*BINIFCALL,ALL,LST:IFCALL=BINIF,SRC:IFCALL/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11401, WORDS

*BINIPAUSE,ALL,LST:PAUSE=BINIF,SRC:PAUSE/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11260, WORDS

*BINIBACKSP,ALL,LST:BACKSP=BINIF,SRC:BACKSP/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 10973, WORDS

*BINIOBJFMT,ALL,LST:OBJFMT=BINIF,SRC:OBJFMT/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 10558, WORDS

*BINIRWBLK,ALL,LST:RWBLK=BINIF,SRC:RWBLK/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 10773, WORDS

*BINICLOG,ALL,LST:ICLOG=BINIF,SRC:ICLOG/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11381, WORDS

*BINISTOP,ALL,LST:ISTOP=BINIF,SRC:ISTOP/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11047, WORDS

*BINIEXIT,ALL,LST:IEXIT=BINIF,SRC:IEXIT/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11395, WORDS

*BINIENCODE,ALL,LST:ENCODE=BINIF,SRC:ENCODE/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 10809, WORDS
5.9.1 Subscripting Modules Assembly

The FORTRAN library modules used for subscripting purposes are grouped into two categories, V2NS and V2S. V2NS modules (Section 5.9.1.1) are the normal library modules used to link debugged programs for production purposes or when execution time is critical. V2S modules (Section 5.9.1.2) constitute a special version of the library containing subscript checking to determine whether an array subscript reference is within the bounds of the program and are used for debugging or running in a Foreground/Background environment.

5.9.1.1 V2NS OTS Assembly

Below is an example of the assembly procedures for the V2NS modules.

$JOB/RT11/TIME ASSEMBLE V2NS MODULES
TTY10

R PIP
SRC:F,MAC=SRC:FINIT,PRE,OTSWA,PRE,FBLCK,PRE,ERRORS,PRE,V2DEF,PRE/A

R MACRO
*BIN:IVEC,V2N,LST:IVEC,VLN=SRC:IVEC,C/NITMICN
ERRORS DETECTED: 0
FREE CORE: 11385, WORDS

*BIN:FVEC,V2N,LST:FVEC,VLN=SRC:FVEC,C/NITMICN
ERRORS DETECTED: 0
FREE CORE: 11373, WORDS

*BIN:DVEC,V2N,LST:DVEC,VLN=SRC:DVEC,C/NITMICN
ERRORS DETECTED: 0
FREE CORE: 11371, WORDS

*BIN:VEC,V2N,LST:VEC,VLN=SRC:VEC,C/NITMICN
ERRORS DETECTED: 0
FREE CORE: 11385, WORDS
5.9.1.2  V2S OTS Assembly

Below is an example of the assembly procedures for the V2S modules.

SJOB/RT11/TIME ASSEMBLE V2S MODULES
TTY10

R PIP
SRC.F, MAC.SRC.FININIT, PRE, OTSWA, PRE, F8LOCK, PRE, ERRORS.PRE, V2SDEF.PRE/A
ASSEMBLY AND LINK INSTRUCTIONS

R MACRO
*BINIVEC,V28,LSTIVEC,VLS*SRC1F,IVEC/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 10941, WORDS

*BINIVECP,V28,LSTIVECP,VLS*SRC1F,IVECP/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 11365, WORDS

*BINIDVEC,V28,LSTIDVEC,VLS*SRC1F,DVEC/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 11363, WORDS

*BINILVEC,V28,LSTILVEC,VLS*SRC1F,LVEC/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 11371, WORDS

*BINIVEP,V28,LSTIVEP,VLS*SRC1F,IVEP/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 11371, WORDS

*BINIFVEP,V28,LSTIFVEP,VLS*SRC1F,FVEP/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 11365, WORDS

*BINIDVEP,V28,LSTIDVEP,VLS*SRC1F,DVEP/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 11363, WORDS

*BINILVEP,V28,LSTILVEP,VLS*SRC1F,LVEP/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 11371, WORDS

*BINIFPVEC,V28,LSTIFPVEC,VLS*SRC1F,FPVEC/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 11365, WORDS

*BINIPVEC,V28,LSTIPVEC,VLS*SRC1F,IPVEC/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 11379, WORDS

*BINIDPVEC,V28,LSTIDPVEC,VLS*SRC1F,DPVEC/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 11375, WORDS

*BINILPVEC,V28,LSTILPVEC,VLS*SRC1F,LPVEC/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 11379, WORDS

*BINISHORT,E28,LSTISHORT,VLS*SRC1F,SHORT/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 9602, WORDS

*BINIERRS,V28,LSTIERRS,VLS*SRC1F,ERRS/C/NITM1CND
ERRORS DETECTED: 0
FREE CORE: 9422, WORDS
5.10 ASSEMBLING HARDWARE DEPENDENT FORTRAN OTS MODULES

Certain OTS library modules are hardware dependent and therefore need to be selectively assembled. These modules include bare machine (Section 5.10.1), EIS (Section 5.10.2), FIS (Section 5.10.3), EAE (Section 5.10.4), and FPU (Section 5.10.5). Refer to the section applicable to your hardware configuration.

5.10.1 Bare Machine OTS Assembly

The following is an example of the assembly procedures for the hardware dependent modules on a machine with none of the optional arithmetic hardware extensions.

```
$JOB/TIME/RT11
$RT11
23144142
TTY10
R PIP
BIN1F,MAC=SRC1V2DEF,PREF,FINIT,PREF,OTSWA,PREF,FBLOCK,PREF,ERRORS,PREF/A
R MACHO
*BINIOTI,NHD,LST1OTI,NHL=BIN1F,SRClOTI/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 10293,WORDS

*BINIIATAN,NHD,LST1IATAN,NHL=BIN1F,SRClIATAN/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11276,WORDS

*BINIXI,NHD,LST1IXI,NHL=BIN1F,SRClIIXI/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11133,WORDS

*BINICONVF,NHD,LSTICONVF,NHL=BIN1F,SRClICONVF/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11036,WORDS

*BINICONV2,NHD,LSTICONV2,NHL=BIN1F,SRClICONV2/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11137,WORDS

*BINICMUL,NHD,LSTICMUL,NHL=BIN1F,SRClCMUL/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11328,WORDS
```

5-33
ASSEMBLY AND LINK INSTRUCTIONS

*BINICDIV, NHD, LSTICDIV, NHL*BINIF, SRCICDIV/C/NHTMICND
ERRORS DETECTED: 0
FREE CORE: 11085, WORDS

*BINIDMUL, NHD, LSTIDMUL, NHL*BINIF, SRCIDMUL/C/NHTMICND
ERRORS DETECTED: 0
FREE CORE: 11069, WORDS

*BINIDADD, NHD, LSTIADD, NHL*BINIF, SRCIADD/C/NHTMICND
ERRORS DETECTED: 0
FREE CORE: 10981, WORDS

*BINIDDIV, NHD, LSTIDDIV, NHL*BINIF, SRCIDDIV/C/NHTMICND
ERRORS DETECTED: 0
FREE CORE: 11077, WORDS

*BINIXDI, NHD, LSTIXDI, NHL*BINIF, SRCIXDI/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11133, WORDS

*BINICONV1, NHD, LSTICONV1, NHL*BINIF, SRCICONV1/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11305, WORDS

*BINICONV3, NHD, LSTICONV3, NHL*BINIF, SRCICONV3/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11361, WORDS

*BINICADD, NHD, LSTICADD, NHL*BINIF, SRCICADD/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11337, WORDS

*BINIALOG, NHD, LSTIALOG, NHL*BINIF, SRCIALOG/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11093, WORDS

*BINIDEXP, NHD, LSTIDEXP, NHL*BINIF, SRCIDEXP/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11053, WORDS

*BINIEXP, NHD, LSTIEXP, NHL*BINIF, SRCIEXP/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11143, WORDS

*BINIDSQRT, NHD, LSTIDSQRT, NHL*BINIF, SRCIDSQRT/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11157, WORDS

*BINISIN, NHD, LSTISIN, NHL*BINIF, SRCISIN/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11312, WORDS

*BINIDSIN, NHD, LSTIDSIN, NHL*BINIF, SRCIDSIN/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11272, WORDS

*BINIDATAN, NHD, LSTIDATAN, NHL*BINIF, SRCIDATAN/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11272, WORDS

*BINIAINT, NHD, LSTIAINT, NHL*BINIF, SRCIAINT/C/NITMICND
ERRORS DETECTED: 0
FREE CORE: 11367, WORDS
ASSEMBLY AND LINK INSTRUCTIONS

*BINIDINT, NHD, LST, DINT, NHL=BIN1F, SRC=DINT/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11367, WORDS

*BINIADD, NHD, LST, IADD, NHL=BIN1F, SRC=IADD/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11347, WORDS

*BINISORT, NHD, LST, ISORT, NHL=BIN1F, SRC=ISORT/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11153, WORDS

*BINIFMUL, NHD, LST, IFMUL, NHL=BIN1F, SRC=IFMUL/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11121, WORDS

*BINIFDIV, NHD, LST, IFDIV, NHL=BIN1F, SRC=IFDIV/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11097, WORDS

*BINIADDM, NHD, LST, IADDM, NHL=BIN1F, SRC=IADDM/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11347, WORDS

*BINIADDP, NHD, LST, IADDP, NHL=BIN1F, SRC=IADDP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11347, WORDS

*BINIDLOG, NHD, LST, IDLOG, NHL=BIN1F, SRC=IDLOG/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11085, WORDS

*BINIFADD, NHD, LST, IFADD, NHL=BIN1F, SRC=IFADD/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11057, WORDS

*BINIXFI, NHD, LST, IXFI, NHL=BIN1F, SRC=IXFI/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11133, WORDS

*BINICONV4, NHD, LST, ICONV4, NHL=BIN1F, SRC=ICONV4/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11185, WORDS

*BINIMUL, NHD, LST, IMUL, NHL=BIN1F, SRC=IMUL/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11149, WORDS

*BINIDIV, NHD, LST, IDIV, NHL=BIN1F, SRC=IDIV/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11141, WORDS

*
$EOJ

TIME
23:54:17

5-35
5.10.2 EIS OTS Assembly

Below is an example of the assembly procedures for the hardware dependent modules on a machine with the EIS hardware option.

$JOB/TIME/RT11

$RT11

2315157

TTYIO

R PIP
BINIF,MAC=SRC1V2DEF,PRE,FINIT,EIS,OTSA,PRE,FBLOCK,PRE,ERRORS,PRE/A

R MACRO
*BINIOTI,EIS,LISTOITI,EIL*BINIF,SRC1OTI/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 10285, WORDS

*BINIIATAN,EIS,LISTIIATAN,EIL*BINIF,SRC1ATAN/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11268, WORDS

*BINIXII,EIS,LISTIXII,EIL*BINIF,SRC1IXII/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11141, WORDS

*BINICONVF,EIS,LISTICONVF,EIL*BINIF,SRC1CONVF/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11036, WORDS

*BINICONV2,EIS,LISTICONV2,EIL*BINIF,SRC1CONV2/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11121, WORDS

*BINICMUL,EIS,LISTICMUL,EIL*BINIF,SRC1CMUL/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11320, WORDS

*BINICDIV,EIS,LISTICDIV,EIL*BINIF,SRC1CDIV/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11077, WORDS

*BINIDMUL,EIS,LISTIDMUL,EIL*BINIF,SRC1DMUL/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11077, WORDS

*BINIDADD,EIS,LISTIDADD,EIL*BINIF,SRC1DADD/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 10977, WORDS

*BINIDDIV,EIS,LISTIDDIV,EIL*BINIF,SRC1DDIV/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11069, WORDS

*BINIIXDI,EIS,LISTIXDI,EIL*BINIF,SRC1IXDI/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11125, WORDS

*BINICONV1,EIS,LISTICONV1,EIL*BINIF,SRC1CONV1/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11377, WORDS

5-36
ASSEMBLY AND LINK INSTRUCTIONS

*BINICONV3,EIS,LSTICONV3,EIL=BIN1F,SRCICONV3/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11353, WORDS

*BINICADD,EIS,LSTICADD,EIL=BIN1F,SRClCADD/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11329, WORDS

*BINIALOG,EIS,LSTIALOG,EIL=BIN1F,SRClALOG/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11085, WORDS

*BINIDEXP,EIS,LSTITDEXP,EIL=BIN1F,SRClDEXP/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11065, WORDS

*BINIEXP,EIS,LSTIEXP,EIL=BIN1F,SRClEXP/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11105, WORDS

*BINIDSQRT,EIS,LSTITDSQRT,EIL=BIN1F,SRClDSQRT/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11149, WORDS

*BINISIN,EIS,LSTISIN,EIL=BIN1F,SRClSIN/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11304, WORDS

*BINIDSIN,EIS,LSTITDSIN,EIL=BIN1F,SRClDSIN/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11264, WORDS

*BINIDATAN,EIS,LSTITDATAN,EIL=BIN1F,SRClDATAN/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11264, WORDS

*BINIAutowired,EIS,LSTITAutowired,EIL=BIN1F,SRClAutowired/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11367, WORDS

*BINIAutowired,EIS,LSTITAutowired,EIL=BIN1F,SRClAutowired/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11367, WORDS

*BINIAutowired,EIS,LSTITAutowired,EIL=BIN1F,SRClAutowired/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11367, WORDS

*BINIAutowired,EIS,LSTITAutowired,EIL=BIN1F,SRClAutowired/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11367, WORDS

*BINIAutowired,EIS,LSTITAutowired,EIL=BIN1F,SRClAutowired/C/N1TTM1CND
ERRORS DETECTED: 0
FREE CORE: 11367, WORDS

5-37
5.10.3 **FIS OTS Assembly**

Below is an example of the assembly procedures for the hardware dependent modules on a machine with the FIS hardware option.

```plaintext
$JOB/TIME/RT11
$RT11
23:25:12

TTVIO
R PIP
BINIF, MAC=SRC:V2DEF, PRE, FINIT, FIS, OTS=NA, PRE, FBLOCK, PRE, ERRORS, PRE/A
R MACHO
*BINIOTI, FIS, LST:OTI, FIL=BINIF, SRC:OTI/C/NITM:CNDE
ERRORS DETECTED: 0
FREE CORE: 10261, WORDS
*BINIATAN, FIS, LST:ATAN, FIL=BINIF, SRC:ATAN/C/NITM:CNDE
ERRORS DETECTED: 0
FREE CORE: 11264, WORDS
*BINIXII, FIS, LST:IXII, FIL=BINIF, SRC:IXII/C/NITM:CNDE
ERRORS DETECTED: 0
FREE CORE: 11137, WORDS
```
ASSEMBLY AND LINK INSTRUCTIONS

*BINICONVF,FIS,LSTICONVF,FIL*BINIF,SRCICONVF/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11032, WORDS

*BINICONV2,FIS,LSTICONV2,FIL*BINIF,SRCICONV2/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11117, WORDS

*BINICMUL,FIS,LSTICMUL,FIL*BINIF,SRCICMUL/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11316, WORDS

*BINICDIV,FIS,LSTICDIV,FIL*BINIF,SRCIDIV/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11073, WORDS

*BINIDMUL,FIS,LSTIDMUL,FIL*BINIF,SRCIDMUL/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11073, WORDS

*BINIDADD,FIS,LSTIDADD,FIL*BINIF,SRCIDADD/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 10973, WORDS

*BINIDDIV,FIS,LSTIDDIV,FIL*BINIF,SRCIDDIV/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11065, WORDS

*BINIXDI,FIS,LSTIXDI,FIL*BINIF,SRCIXDI/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11121, WORDS

*BINICONV1,FIS,LSTICONV1,FIL*BINIF,SRCICONV1/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11316, WORDS

*BINICONV3,FIS,LSTICONV3,FIL*BINIF,SRCICONV3/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11349, WORDS

*BINICADD,FIS,LSTICADD,FIL*BINIF,SRCICADD/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11325, WORDS

*BINIALOG,FIS,LSTIALOG,FIL*BINIF,SRCIALOG/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11081, WORDS

*BINIDEXP,FIS,LSTIDEXP,FIL*BINIF,SRCIDEXP/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11041, WORDS

*BINIEXP,FIS,LSTIEXP,FIL*BINIF,SRCIEXP/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11081, WORDS

*BINIDSQRT,FIS,LSTIDSQRT,FIL*BINIF,SRCIDSQRT/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11146, WORDS

*BINISIN,FIS,LSTISIN,FIL*BINIF,SRCISIN/C/NITMICOND
ERRORS DETECTED: 0
FREE CORE: 11300, WORDS
ASSEMBLY AND LINK INSTRUCTIONS

*BINIDSIN,FIS,LISTIDSIN,FILEBINIF,SRCDSIN/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11260, WORDS

*BINIDATAN,FIS,LISTIDATAN,FILEBINIF,SRCIDATAN/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11260, WORDS

*BINIAINT,FIS,LISTIAINT,FILEBINIF,SRCIAINT/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11363, WORDS

*BINIDINT,FIS,LISTIDINT,FILEBINIF,SRCIDINT/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11363, WORDS

*BINIADDA,FIS,LISTIADDA,FILEBINIF,SRCIADDA/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11165, WORDS

*BINISORT,FIS,LISTISORT,FILEBINIF,SRCISORT/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11145, WORDS

*BINIFMUL,FIS,LISTIFMUL,FILEBINIF,SRCIFMUL/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11165, WORDS

*BINIFDIV,FIS,LISTIFDIV,FILEBINIF,SRCIFDIV/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11165, WORDS

*BINIADDM,FIS,LISTIADDM,FILEBINIF,SRCIADDM/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11341, WORDS

*BINIADDP,FIS,LISTIADDP,FILEBINIF,SRCIADDP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11341, WORDS

*BINIDLOG,FIS,LISTIDLOG,FILEBINIF,SRCIDLOG/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11073, WORDS

*BINIFADD,FIS,LISTIFADD,FILEBINIF,SRCIFADD/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11141, WORDS

*BINIFXI,FIS,LISTIFXI,FILEBINIF,SRCIFXI/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11121, WORDS

*BINICONV4,FIS,LISTICONV4,FILEBINIF,SRCICONV4/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11173, WORDS

*BINIIMUL,FIS,LISTIIMUL,FILEBINIF,SRCIIMUL/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11161, WORDS

5-40
ASSEMBLY AND LINK INSTRUCTIONS

ERRORS DETECTED: 0
FREE CORE: 11161, WORDS

*SEOJ

TIME
23:34:36

5.10.4 EAE OTS Assembly

Below is an example of the assembly procedures for the hardware dependent modules on a machine with the EAE hardware option.

$JOB/TIME/RT11

$RT11

23:03:03

TTY10

R PIP
BIN,F,MAC=SRC1V2DEF,PRE,FINIT,EAE,OTSWA,PRE,FBLOCK,PRE,ERRORS,PRE/A

R MACH
*BINIOTI,EAE,LST:OTI,EAL=BIN,F,SRG:OTI/C,N:TTMICND
ERRORS DETECTED: 0
FREE CORE: 10285, WORDS

*BINIATAN,EAE,LST:ATAN,EAL=BIN,F,SRG:ATAN/C/N:TTMICND
ERRORS DETECTED: 0
FREE CORE: 11272, WORDS

*BINI XII,EAE,LST: XII,EAL=BIN,F,SRG: XII/C/N:TTMICND
ERRORS DETECTED: 0
FREE CORE: 11129, WORDS

*BINI CONVF,EAE,LST: CONVF,EAL=BIN,F,SRG: CONVF/C/N:TTMICND
ERRORS DETECTED: 0
FREE CORE: 11032, WORDS

*BINI CONV2,EAE,LST: CONVF,EAL=BIN,F,SRG: CONVF/C/N:TTMICND
ERRORS DETECTED: 0
FREE CORE: 11117, WORDS

*BINI CMUL,EAE,LST: CMUL,EAL=BIN,F,SRG: CMUL/C/N:TTMICND
ERRORS DETECTED: 0
FREE CORE: 11324, WORDS

*BINI CDIV,EAE,LST: CDIV,EAL=BIN,F,SRG: CDIV/C/N:TTMICND
ERRORS DETECTED: 0
FREE CORE: 11081, WORDS

*BINI DMUL,EAE,LST: DMUL,EAL=BIN,F,SRG: DMUL/C/N:TTMICND
ERRORS DETECTED: 0
FREE CORE: 11081, WORDS

*BINI DADD,EAE,LST: DADD,EAL=BIN,F,SRG: DADD/C/N:TTMICND
ERRORS DETECTED: 0
FREE CORE: 10981, WORDS
ASSEMBLY AND LINK INSTRUCTIONS

*BINIDDIV.EAE,LSTIDDIV.EAL=BINIF.SRC=IDDIV/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11073, WORDS

*BINIDXI.EAE,LSTDXI.EAL=BINIF.SRC=IDXI/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11129, WORDS

*BINICONV1.EAE,LSTICONV1.EAL=BINIF.SRC=ICONV1/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11381, WORDS

*BINICONV3.EAE,LSTICONV3.EAL=BINIF.SRC=ICONV3/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11333, WORDS

*BINICADD.EAE,LSTICADD.EAL=BINIF.SRC=CADD/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11089, WORDS

*BINIALOG.EAE,LSTIALOG.EAL=BINIF.SRC=IALOG/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11049, WORDS

*BINIDEEXP.EAE,LSTIDEEXP.EAL=BINIF.SRC=IDEEXP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11109, WORDS

*BINIEXP.EAE,LSTIEXP.EAL=BINIF.SRC=EXP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11109, WORDS

*BINIDSQRT.EAE,LSTIDSQRT.EAL=BINIF.SRC=IDSQRT/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11153, WORDS

*BINISIN.EAE,LSTISIN.EAL=BINIF.SRC=SIN/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11308, WORDS

*BINJDSIN.EAE,LSTDJSIN.EAL=BINIF.SRC=JSIN/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11268, WORDS

*BINIDATAN.EAE,LSTIDATAN.EAL=BINIF.SRC=DATAN/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11268, WORDS

*BINIIANT.EAE,LSTIANT.EAL=BINIF.SRC=IANT/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11363, WORDS

*BINIIDINT.EAE,LSTIIDINT.EAL=BINIF.SRC=IIDINT/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11363, WORDS

*BINIADD.D.EAE,LSTIADD.D.EAL=BINIF.SRC=ADD.D/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11343, WORDS

*BINISOR.T.EAE,LSTISOR.T.EAL=BINIF.SRC=SOR.T/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11149, WORDS
ASSEMBLY AND LINK INSTRUCTIONS

*BINIFMUL, EAE, LSTIFMUL, EAL = BINIF, SRC1FMUL/C/N1TM1CND
ERRORS DETECTED: 0
FREE CORE: 11077, WORDS

*BINIFDIV, EAE, LSTIFDIV, EAL = BINIF, SRC1FDIV/C/N1TM1CND
ERRORS DETECTED: 0
FREE CORE: 11089, WORDS

*BINIADDM, EAE, LSTIADDM, EAL = BINIF, SRC1ADDM/C/N1TM1CND
ERRORS DETECTED: 0
FREE CORE: 11343, WORDS

*BINIADDP, EAE, LSTIADDP, EAL = BINIF, SRC1ADDP/C/N1TM1CND
ERRORS DETECTED: 0
FREE CORE: 11343, WORDS

*BINILDLOG, EAE, LSTILDLOG, EAL = BINIF, SRC1DLOG/C/N1TM1CND
ERRORS DETECTED: 0
FREE CORE: 111081, WORDS

*BINIFADD, EAE, LSTIFADD, EAL = BINIF, SRC1FADD/C/N1TM1CND
ERRORS DETECTED: 0
FREE CORE: 11053, WORDS

*BINIXFI, EAE, LSTIXFI, EAL = BINIF, SRC1XFI/C/N1TM1CND
ERRORS DETECTED: 0
FREE CORE: 11129, WORDS

*BINICONV4, EAE, LSTICONV4, EAL = BINIF, SRC1CONV4/C/N1TM1CND
ERRORS DETECTED: 0
FREE CORE: 11181, WORDS

*BINIMUL, EAE, LSTIMUL, EAL = BINIF, SRC1IMUL/C/N1TM1CND
ERRORS DETECTED: 0
FREE CORE: 11157, WORDS

*BINIDIV, EAE, LSTIDIV, EAL = BINIF, SRC1IDIV/C/N1TM1CND
ERRORS DETECTED: 0
FREE CORE: 11165, WORDS

*
$EOJ

TIME
23:12:15

5.10.5 FPU OTS Assembly

Below is an example of the assembly procedures for the hardware dependent modules on a machine with the FPU hardware option.

$JOB TIME/RT11
$RT11
23:134149
TTY10
R PIP
BINIF, MAC SRC1V2DEF, PRE, FINIT, FPU, OTSWA, PRE, FBLOCK, PRE, ERRORS, PRE/A

5-43
ASSEMBLY AND LINK INSTRUCTIONS

R MACRO
*BINIOTI,FPU,LSTIOTI,FPL=BINIF,SRCIOTI/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 10237, WORDS

*BINIATAN,FPU,LSTIATAN,FPL=BINIF,SRCIATAN/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11281, WORDS

*BINIXII,FPU,LSTIXII,FPL=BINIF,SRCIXII/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11133, WORDS

*BINICONVF,FPU,LSTICONVF,FPL=BINIF,SRCICONVF/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11028, WORDS

*BINICONV2,FPU,LSTICONV2,FPL=BINIF,SRCICONV2/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11134, WORDS

*BINICMUL,FPU,LSTICMUL,FPL=BINIF,SRCICMUL/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11325, WORDS

*BINICDIV,FPU,LSTICDIV,FPL=BINIF,SRCICDIV/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11101, WORDS

*BINIDMUL,FPU,LSTIDMUL,FPL=BINIF,SRCIDMUL/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11149, WORDS

*BINIDADD,FPU,LSTIDADD,FPL=BINIF,SRCIDADD/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11069, WORDS

*BINIDDIV,FPU,LSTIDDIV,FPL=BINIF,SRCIDDIV/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11149, WORDS

*BINIXDI,FPU,LSTIXDI,FPL=BINIF,SRCIXDI/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11137, WORDS

*BINICONV1,FPU,LSTICONV1,FPL=BINIF,SRCICONV1/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11374, WORDS

*BINICONV3,FPU,LSTICONV3,FPL=BINIF,SRCICONV3/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11357, WORDS

*BINICADD,FPU,LSTICADD,FPL=BINIF,SRCICADD/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11326, WORDS

*BINIALOG,FPU,LSTIALOG,FPL=BINIF,SRCIALOG/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11129, WORDS

*BINIDEXP,FPU,LSTIDEXP,FPL=BINIF,SRCIDEXP/C/NITTMICND
ERRORS DETECTED: 0
FREE CORE: 11101, WORDS
ASSEMBLY AND LINK INSTRUCTIONS

*BIN1EXP,FPU,LST1EXP,FPL=BIN1,FSRC1EXP/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11121, WORDS

*BIN1DSQRT,FPU,LST1DSQRT,FPL=BIN1,FSRC1DSQRT/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11149, WORDS

*BIN1SIN,FPU,LST1SIN,FPL=BIN1,FSRC1SIN/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11325, WORDS

*BIN1DSIN,FPU,LST1DSIN,FPL=BIN1,FSRC1DSIN/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11325, WORDS

*BIN1DATAN,FPU,LST1DATAN,FPL=BIN1,FSRC1DATAN/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11281, WORDS

*BIN1ANT,FPU,LST1ANT,FPL=BIN1,FSRC1ANT/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11365, WORDS

*BIN1DINT,FPU,LST1DINT,FPL=BIN1,FSRC1DINT/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11362, WORDS

*BIN1ADDA,FPU,LST1ADDA,FPL=BIN1,FSRC1ADDA/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11349, WORDS

*BIN1DSQRT,FPU,LST1DSQRT,FPL=BIN1,FSRC1DSQRT/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11419, WORDS

*BIN1FMUL,FPU,LST1FMUL,FPL=BIN1,FSRC1FMUL/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11165, WORDS

*BIN1FDIV,FPU,LST1FDIV,FPL=BIN1,FSRC1FDIV/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11161, WORDS

*BIN1ADDM,FPU,LST1ADDM,FPL=BIN1,FSRC1ADDM/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11161, WORDS

*BIN1ADDP,FPU,LST1ADDP,FPL=BIN1,FSRC1ADDP/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11349, WORDS

*BIN1DLOG,FPU,LST1DLOG,FPL=BIN1,FSRC1DLOG/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11129, WORDS

*BIN1FADD,FPU,LST1FADD,FPL=BIN1,FSRC1FADD/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11141, WORDS

*BIN1XF1,FPU,LST1XF1,FPL=BIN1,FSRC1XF1/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11137, WORDS
ASSEMBLY AND LINK INSTRUCTIONS

*BINICONV4,FPU,LSTICONV4,FPL=BINIF,SRCKCONV4/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11169, WORDS

*BINIMUL,FPU,LSTIMUL,FPL=BINIF,SRCKIMUL/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11157, WORDS

*BINIDIV,FPU,LSTIDIV,FPL=BINIF,SRCKIDIV/C/N1TMCN
ERRORS DETECTED: 0
FREE CORE: 11157, WORDS

*
$EOJ

TIME
23143154

5.10.6 OTS Library Component Building

This section contains the final OTS library preparation. The FORTRAN
object modules have been concatenated, using PIP, into larger common
object modules to simplify the library build process.

$JOB/RT11/TIME CONCATENATE F4 OTS OBJ'S INTO LARGER FILES
$RT11
00140148

TYI0

R PIP
BIN1OTSOM,OBJ=BIN1*,ALL/B
BIN1FPU,OBJ=BIN1*,FPU/B
BIN1FIS,OBJ=BIN1*,FIS/B
BIN1EAE,OBJ=BIN1*,EAE/B
BIN1EIS,OBJ=BIN1*,EIS/B
BIN1NHD,OBJ=BIN1*,NHD/B
BIN1UNI,OBJ=BIN1*,UNI,*,SZ/B
BIN1V2S,OBJ=BIN1SHORT,E2S,*,V2S/B
BIN1V2N8,OBJ=BIN1SHORT,E2N,*,V2N/B
$EOJ

TIME
00143112

Refer to the appropriate section of Chapter 2 for specific library
build procedures.

Section 2.2.2 building from DECpack
Section 2.3.2 building from DECTape
Section 2.4.2 building from diskette
Section 2.5.2 building from magtape
Section 2.6.2 building from cassette
Section 2.7.2 building from paper tape
5.11 ASSEMBLING BASIC/RT-11

This section provides assembly instructions for BASIC/RT-11; this information applies only to those users who received the source versions of BASIC/RT-11. A 16k system is required to assemble BASIC. The source program of BASIC/RT-11 consists of three source files:

BASICL.MAC  
BASICH.MAC  
FPMP.MAC

It is necessary to create the .MAC files BASICR, BASICE, and BASICX which consist of only one line of code each. They specify the conditionals necessary to assemble BASICL into three object modules: BASICR.OBJ, BASICE.OBJ, and BASICX.OBJ.

1. Create the object modules using EDIT. To the running RT-11 monitor,

Type:  
Response: R EDIT<CR>

Type: EWBASICR.MAC<ALT><ALT>
Response: *

Type: IBASICR=1<CR>
<ALT>EX<ALT><ALT>
Response: ;

Type: R EDIT<CR>
Response: *

Type: EWBASICE.MAC<ALT><ALT>
Response: *

Type: IBASICE=1<CR>
<ALT>EX<ALT><ALT>
Response: ;

Type: R EDIT<CR>
Response: *

Type: EWBASICX.MAC<ALT><ALT>
Response: *

Type: IBASICX=1<CR>
<ALT>EX<ALT><ALT>
Response: ;

2. If any other options are desired, include the conditionals for them in these files. For example:

$NOSTR=1 ;NO STRINGS  
$LONGER=1 ;LONGER ERROR MESSAGES  
$NOVF=1 ;NO VIRTUAL MEMORY FILES  
$NOPOW=1 ;NO POWER-FAIL OPTION  
$STKSIZE=n ;STACK SIZE IN BYTES (DEFAULT IS 200 (OCTAL) BYTES)

NOTE

BASIC uses part of the user area for its working stack. See BASIC/RT-11 Release Notes, Section 3.9, for further information.
ASSEMBLY AND LINK INSTRUCTIONS

If BASIC is to run on an 8K system, the $NOSTR conditional must be specified.

For example, to create a BASIC with no strings, no virtual memory files, and a stack size of 300 (octal) bytes, create the BASICR, BASICE, and BASICX files using the EDIT program, as follows:

Type: R EDIT<CR>
Response: *

Type: EWBASICR,MAC<ALT><ALT>
Response: *

Type: IBASICR=1<CR>
$NOSTR=1<CR>
$NOVF=1<CR>
$STKSZ=300<CR>
<ALT>EX<ALT><ALT>
Response: *

Type: R EDIT<CR>
Response: *

Type: EWBASICE,MAC<ALT><ALT>
Response: *

Type: IBASICE=1<CR>
$NOSTR=1<CR>
$NOVF=1<CR>
$STKSZ=300<CR>
<ALT>EX<ALT><ALT>
Response: *

Type: R EDIT<CR>
Response: *

Type: EWBASICX,MAC<ALT><ALT>
Response: *

Type: IBASICX=1<CR>
$NOSTR=1<CR>
$NOVF=1<CR>
$STKSZ=300<CR>
<ALT>EX<ALT><ALT>
Response: *

3. To assemble BASIC,

Type: R MACRO<CR>
Response: *

Type: BASICR=BASICR,BASICL<CR>
Response: *

Type: BASICE=BASICE,BASICL<CR>
Response: *

Type: BASICX=BASICX,BASICL<CR>
Response: *
ASSEMBLY AND LINK INSTRUCTIONS

Type: BASIC=BASIC<CR>
Response: *

Type: FPMP=FPMP<CR>
Response: *

Type: CTRL C
*C

The preceding instructions produce five object modules:

BASICR.OBJ BASIC Root section
BASICE.OBJ BASIC Edit overlay
BASICX.OBJ BASIC EXECution overlay
FPMP.OBJ Floating Point Math Package
BASICH.OBJ BASIC High section, with once-only code and optional functions

5.11.1 Floating Point Math Package

Assembly of the FPMP source file produces a standard FPMP for BASIC. This standard FPMP runs on any PDP-11 but does not make use of special arithmetic hardware; it includes all of the routines needed for the full complement of BASIC arithmetic functions. A non-standard FPMP may be specified, as outlined in Table 5-1.

1. To assemble the Floating Point Math Package with conditionals, it is necessary to use the EDIT program to either insert the conditionals in the beginning of the FPMP.MAC file or create a new file, FPMP.MAC which will be assembled with FPMP.MAC. For example, to create the FPMP with the SIN, COS, and SQR functions and EAE hardware but without the ATN function, create the file FPMP.MAC with EDIT as follows:

Type: R EDIT<CR>
Response: *

Type: EWFPMPC,MAC<ALT><ALT>
Response: *

Type: [MIN*1<CR>
EAE=1<CR>
CND$37=1<CR>
CND$41=1<CR>
<ALT>EX<ALT><ALT>

Response: *

2. Assemble BASIC with MACRO as follows:

Type: R MACRO<CR>
Response: *

Type: BASICR=BASICR,BASICL<CR>
Response: *

5-49
### Table 5-1
**FPMP Assembly Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIN</td>
<td>undefined</td>
<td>Define to eliminate code for BASIC functions SIN, COS, ATN, and SQR. When linked, the functions are listed as undefined references. However, when executed by a BASIC program, they produce a ?UFN (UNDEFINED FUNCTION) error.</td>
</tr>
<tr>
<td>FPU</td>
<td>undefined</td>
<td>Define to assemble a version for the PDP-11/45 FPU hardware.</td>
</tr>
<tr>
<td>EAE</td>
<td>undefined</td>
<td>Define to assemble for the EAE hardware.</td>
</tr>
<tr>
<td>MULDIV</td>
<td>undefined</td>
<td>Define to assemble for the PDP-11/40 extended instruction set (EIS) or the PDP-11/45 processor.</td>
</tr>
</tbody>
</table>

If MIN is defined, then the following parameters may be specified to include the SIN, COS, ATN, and SQR functions, selectively.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CND$37</td>
<td>1</td>
<td>Define (only if MIN is specified) to include the code for the SIN and COS functions.</td>
</tr>
<tr>
<td>CND$39</td>
<td>1</td>
<td>Define (only if MIN is specified) to include the code for the ATN function.</td>
</tr>
<tr>
<td>CND$41</td>
<td>1</td>
<td>Define (only if MIN is specified) to include the code for the SQR function.</td>
</tr>
</tbody>
</table>

**Type:** BASICE=BASIC,BASICL<CR>
**Response:** *  
**Type:** BASICX=BASICX,BASICL<CR>
**Response:** *  
**Type:** FPMP=FPMPG,FPMP<CR>
**Response:** *  
**Type:** CTRL C
**Response:** *C

### 5.12 LINKING BASIC/RT-11

The five object modules (BASICR, BASICE, BASICX, FPMP, BASICH) may be linked with or without an overlay structure. The overlay option has the advantage that sections of BASIC that are not required at the same time occupy the same core space alternately when they are used; the
disadvantage is that BASIC will run somewhat slower and I/O time will be spent when switching overlay segments in and out of core. When BASIC is linked to run in an 8K system, it must use the overlay option and the $NOSTR conditional must be used during assembly.

1. To link BASIC without overlays,

Type: \texttt{R LINK<CR>}
Response: *

Type: \texttt{BASIC,BASIC+<CR>,FPMP,BASICE,BASICX,BASIC/B1408<CR>}
Response: ;

\textbf{NOTE}

The above command string is for a configuration that has no devices with vectors above 400. User's with 8K whose configurations have interrupt vectors above 400 should relink BASIC with a bottom address of 500. See \texttt{BASIC/RT-11 Release Notes}, Section 3.6, for further information.

2. To link BASIC with overlays,

Type: \texttt{R LINK<CR>}
Response: *

Type: \texttt{BAS8K,BAS8K+BASICR,FPMP/T/B1400/C<CR>}
Response: \texttt{TRANSFER ADDRESS =}

\textbf{NOTE}

The above command string is for a configuration that has no devices with vectors above 400. User's with 8K whose configurations have interrupt vectors above 400 should relink BASIC with a bottom address of 500. See \texttt{BASIC/RT-11 Release Notes}, Section 3.6, for further information.

Type: \texttt{GO<CR>}
Response: *

Type: \texttt{BASICE/011/C<CR>}
Response: *

Type: \texttt{BASICX/011/C<CR>}
Response: *

Type: \texttt{BASIC/B1408<CR>}
Response: *

5.12.1 \textbf{Linking BASIC/RT-11 with User Functions}

The System Function Table address used by the CALL statement to link the user's assembly language routines must be set in the first word of the BASICR control section.
The source code for the System Function Table and the actual function routines must be broken into two separate source files. The source file FUN1 consists of the System Function Table definition, with this general outline:

Function Entry Points

.GLOBL FN1, FN2
.CSECT BASICR
.WORD FUNTAB

.CSECT FUN1  
FUNTAB: (function table entries for FN1,FN2,...)

The source file FUN2 consists of the code for the function routines, with the following general outline:

.GLOBL FN1,FN2,...  
.CSECT FUN2

FN1:  
FN2:  
(the user function routines)

1. To link BASIC with the user functions in a non-overlay system, type the following to the running RT-11 monitor:

   Type:  R LINK<CR>
   Response:  *

   Type:  BASIC=BASICR,FPMP,BASICE,BASICX/B1500/C<CR>
   Response:  *

   Type:  FUN1,FUN2[,GETARG],BASICH<CR>
   Response:  *

GETARG is the general argument interface module listed in Appendix H of the BASIC/RT-11 Language Reference Manual.

2. In an overlay system, there are two possible ways to link BASIC with the user functions.

If the user function routines contain no data that must be preserved from one function call to the next, that is, if the code for the routines may be refreshed at the beginning of each function call, then the routines may be incorporated into the execution overlay by using the following commands to the Linker:

   Type:  BASIC,BASIC=BASICR,FPMP,FUN1/T/B1400/C<CR>
   Response:  TRANSFER ADDRESS =

NOTE

The above command string is for a configuration that has no devices with vectors above 400. User's with 8K whose configurations have interrupt vectors above 400 should relink BASIC with a bottom address of 500. See BASIC/RT-11 Release Notes, Section 3.6, for further information.
ASSEMBLY AND LINK INSTRUCTIONS

Type: GO<CR>
Response: *

Type: BASICE/O11/C<CR>
Response: *

Type: BASICX, FUN2[, GETARG]/O11/C<CR>
Response: *

Type: BASICH/O12<CR>

In this case, the function routines (in the module FUN2) occupy space in the first overlay segment that is normally unused, since the Edit overlay segment (BASICE) is about 250 words longer in the 8K no-string system than the Execution overlay segment (BASICX). These first 250 words of storage are free in this case.

3. If FUN2 may not be read in anew whenever it is used, enter the following to the Linker:

Type: BASIC=BASICR, FPMP, FUN1, FUN2/T/BI400/C<CR>
Response: TRANSFER ADDRESS =

NOTE

The above command string is for a configuration that has no devices with vectors above 400. Users with 8K whose configurations have interrupt vectors above 400 should relink BASIC with a bottom address of 500. See BASIC/RT-11 Release Notes, Section 3.6, for further information.

Type: GO<CR>
Response: *

Type: BASICE/O11/C<CR>
Response: *

Type: BASICX[, GETARG]/O11/C<CR>
Response: *

Type: BASICH/O12<CR>

Three additional object modules (FPMP.FPU, FPMP.EAE, FPMP.EIS) allow BASIC/RT-11 to be linked for special arithmetic hardware.

Optional Hardware Replace FPMP.OBJ with
EAE hardware FPMP.EAE
PDP-11/40 extended processor or PDP-11/45 processor FPMP.EIS
PDP-11/45 FPU hardware FPMP.FPU
APPENDIX A

INSTRUCTIONS FOR LOADING SOFTWARE BOOTSTRAPS

A.1 RK11 DECPACK BOOTSTRAP LOADER

1. Deposit the basic RK11 disk bootstrap loader into memory as follows:
   a. Set the ENABLE/HALT switch to HALT, then set the first address, 001000, in the Switch Register. (Set switch 9 to the up (1) position and all others to the down (0) position.)
   b. Press the LOAD ADDR switch.
   c. Set the proper contents for Table A-1 in the Switch Register and lift the DEP switch.
   d. Repeat Step c until all the instructions have been deposited.

2. Verify that the bootstrap program has been deposited properly as follows:
   a. Set the starting address in the Switch Register as in Step 1a above.
   b. Press the LOAD ADDR switch.
   c. Display the contents of that address in the Data Register by pressing the EXAM switch.
   d. Compare the number in the Data Register with the value in Table A-1.
   e. If they are the same, repeat Step 2c until all words have been examined.
   f. If not the same, repeat Step 1.

3. Set the starting address, 001000, in the Switch Register as in Step 1a above, then press the LOAD ADDR switch.

4. Set the ENABLE/HALT switch to ENABLE, then press the START switch.
A.2 TC11 DECTAPE BOOTSTRAP LOADER

1. Deposit the basic DECTape bootstrap loader into memory as follows:

   a. Set the ENABLE/HALT switch to HALT, then set the first address, 001000, in the Switch Register. (Set switch 9 to the up (1) position and all others to the down (0) position.

   b. Press the LOAD ADDR switch.

   c. Set the proper contents from Table A-2 in the Switch
INSTRUCTIONS FOR LOADING SOFTWARE BOOTSTRAPS

Register and lift the DEP switch.

d. Repeat Step c until all the instructions have been deposited.

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>001000</td>
<td>012700</td>
<td>Set switches 12, 10, 8, 7, and 6 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001002</td>
<td>177344</td>
<td>Set switches 15 through 9, 7, 6, 5, and 2 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001004</td>
<td>012710</td>
<td>Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001006</td>
<td>177400</td>
<td>Set switches 15 through 8 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001010</td>
<td>012740</td>
<td>Set switches 12, 10, 8, 7, 6, and 5 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001012</td>
<td>004002</td>
<td>Set switches 11 and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001014</td>
<td>005710</td>
<td>Set switches 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001016</td>
<td>100376</td>
<td>Set switches 15, and 7 through 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001020</td>
<td>012710</td>
<td>Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001022</td>
<td>000003</td>
<td>Set switches 1 and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

(continued on next page)
### Table A-2 (Cont.)
**TC11 Bootstrap Loader**

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>001024</td>
<td>105710</td>
<td>Set switches 15, 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001026</td>
<td>100376</td>
<td>Set switches 15, and 7 through 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001030</td>
<td>012710</td>
<td>Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001032</td>
<td>000005</td>
<td>Set switches 2 and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001034</td>
<td>105710</td>
<td>Set switches 15, 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001036</td>
<td>100376</td>
<td>Set switches 15 and 7 through 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001040</td>
<td>005007</td>
<td>Set switches 11, 9, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

2. Verify that the bootstrap program has been deposited properly as follows:
   
   a. Set the starting address in the Switch Register as in Step 1a above.
   
   b. Press the LOAD ADDR switch.
   
   c. Display the contents of that address in the Data Register by pressing the EXAM switch.
   
   d. Compare the number in the Data Register with the value in Table A-2.
   
   e. If they are the same, repeat Step 2c until all words have been examined.
   
   f. If not the same, repeat Step 1.

3. Set the starting address 001000 in the Switch Register as in Step 1a above, then press the LOAD ADDR switch.

4. Set the ENABLE/HALT switch to ENABLE, then press the START switch.
INSTRUCTIONS FOR LOADING SOFTWARE BOOTSTRAPS

A.3 RX11/RX01 DISKETTE BOOTSTRAP LOADER

1. If the computer is a PDP-11V/03, perform the following; if the computer is a PDP-11/03 or LSI-11, see the LSI-11, PDP-11/03 User's Manual (EK-LSI11-TM-002); otherwise, go to Step 2.

   a. Put all three switches in the up position.

   b. Move the DC ON/OFF switch down and up.

   c. Response: $  
      Type: DX<CR>

   The bootstrapping procedure is complete.

2. Deposit the basic diskette bootstrap loader into memory as follows:

   a. Set the ENABLE/HALT switch to HALT, then set the first address, 001000, in the Switch Register. (Set switch 9 to the up (1) position and all others to the down (0) position.)

   b. Press the LOAD ADDR switch.

   c. Set the proper contents from Table A-3 in the Switch Register and lift the DEP switch.

   d. Repeat Step 2c until all the instructions have been deposited.

Table A-3

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>001000</td>
<td>012702</td>
<td>Set switches 12, 10, 8, 7, 6, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001002</td>
<td>1002n7</td>
<td>(n=4 for unit 0 and n=6 for unit 1) Set switches 15, 7, 2, 1, and 0 to the up (1) position and all others to the down (0) position. For unit 0, set switch 5 to the up (1) position; for unit 1, set switches 5 and 4 to the up (1) position.</td>
</tr>
<tr>
<td>001004</td>
<td>012701</td>
<td>Set switches 12, 10, 8, 7, 6, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

(continued on next page)
### Table A-3 (Cont.)
#### RX12 Bootstrap Loader

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>001006</td>
<td>177170</td>
<td>Set switches 15 through 9, and 6 through 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001010</td>
<td>130211</td>
<td>Set switches 15, 13, 12, 7, 3, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001012</td>
<td>001776</td>
<td>Set switches 9 through 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001014</td>
<td>112703</td>
<td>Set switches 15, 12, 10, 8, 7, 6, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001016</td>
<td>000007</td>
<td>Set switches 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001020</td>
<td>010100</td>
<td>Set switches 12 and 6 to the up (1) position. Set all others to the down (1) position.</td>
</tr>
<tr>
<td>001022</td>
<td>010220</td>
<td>Set switches 12, 7, and 4 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001024</td>
<td>000402</td>
<td>Set switches 8 and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001026</td>
<td>012710</td>
<td>Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001030</td>
<td>000001</td>
<td>Set switch 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001032</td>
<td>006203</td>
<td>Set switches 11, 10, 7, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001034</td>
<td>103402</td>
<td>Set switches 15, 10, 9, 8, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

(continued on next page)
Table A-3 (Cont.)
RX11 Bootstrap Loader

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>001036</td>
<td>112711</td>
<td>Set switches 15, 12, 10, 8, 7, 6, 3, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001040</td>
<td>111023</td>
<td>Set switches 15, 12, 9, 4, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001042</td>
<td>030211</td>
<td>Set switches 13, 12, 7, 3, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001044</td>
<td>001776</td>
<td>Set switches 9 through 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001046</td>
<td>100756</td>
<td>Set switches 15, 8, 7, 6, 5, and 3, 2, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001050</td>
<td>103766</td>
<td>Set switches 15, 10 through 4, 2, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001052</td>
<td>105711</td>
<td>Set switches 15, 11, 9 through 6, 3, and 0 to the up position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001054</td>
<td>100771</td>
<td>Set switches 15, 8 through 3, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001056</td>
<td>005000</td>
<td>Set switches 11 and 9 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001060</td>
<td>022710</td>
<td>Set switches 13, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001062</td>
<td>000240</td>
<td>Set switches 7 and 5 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001064</td>
<td>001347</td>
<td>Set switches 9, 7, 6, 5, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

(continued on next page)
3. Verify that the bootstrap program has been deposited properly as follows:
   a. Set the starting address in the Switch Register as in Step 2a above.
   b. Press the LOAD ADDR switch.
   c. Display the contents of that address in the Data Register by pressing the EXAM switch.
   d. Compare the number in the Data Register with the value in Table A-3.
   e. If they are the same, repeat Step 3c until all words have been examined.
   f. If not the same, repeat Step 2.

3. Set the starting address 001000 in the Switch Register as in Step 2a above, then press the LOAD ADDR switch.

4. Set the ENABLE/HALT switch to ENABLE, then press the START switch.

A.4 MAGTAPE BOOTSTRAP LOADERS

1. Deposit the basic magtape bootstrap into memory as follows:
   a. Set the ENABLE/HALT switch to HALT, then set the first address, 010000, in the Switch Register. (Set switch 12 to the up (1) position and all others to the down (0) position.)
b. Press the LOAD ADDR switch.

c. Set the proper contents for TJU16 magtape (Table A-4) or for TM11 magtape (Table A-5) in the Switch Register and lift the DEP switch.

d. Repeat Step c until all the instructions have been deposited.

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>010000</td>
<td>012700</td>
<td>Set switches 12, 10, 8, 7, and 6 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010002</td>
<td>172440</td>
<td>Set switches 15 through 12, 10, 8, and 5 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010004</td>
<td>012710</td>
<td>Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010006</td>
<td>000021</td>
<td>Set switches 4 and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010010</td>
<td>012760</td>
<td>Set switches 12, 10, and 8 through 4 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010012</td>
<td>001300</td>
<td>Set switches 9, 7, and 6 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010014</td>
<td>000032</td>
<td>Set switches 4, 3, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010016</td>
<td>012760</td>
<td>Set switches 12, 10, and 8 through 4 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010020</td>
<td>177777</td>
<td>Set all switches to the up (1) position.</td>
</tr>
<tr>
<td>010022</td>
<td>000006</td>
<td>Set switches 2 and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

(continued on next page)
### INSTRUCTIONS FOR LOADING SOFTWARE BOOTSTRAPS

Table A-4 (Cont.)
TJU16 Bootstrap Loader

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>010024</td>
<td>012720</td>
<td>Set switches 12, 10, 8, 7, 6, and 4 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010026</td>
<td>000031</td>
<td>Set switches 4, 3, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010030</td>
<td>105760</td>
<td>Set switches 15, 11, and 9 through 4 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010032</td>
<td>000010</td>
<td>Set switch 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010034</td>
<td>100375</td>
<td>Set switches 15, 7 through 2, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010036</td>
<td>012710</td>
<td>Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010040</td>
<td>177000</td>
<td>Set switches 15 through 9 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010042</td>
<td>012740</td>
<td>Set switches 12, 10, and 8 through 5 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010044</td>
<td>000071</td>
<td>Set switches 5, 4, 3, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010046</td>
<td>032710</td>
<td>Set switches 13, 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010050</td>
<td>100200</td>
<td>Set switches 15 and 7 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010052</td>
<td>001775</td>
<td>Set switches 9 through 2 and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

(continued on next page)
### Instructions for Loading Software Bootstraps

#### TJUL6 Bootstrap Loader

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>010054</td>
<td>100007</td>
<td>Set switches 15, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010056</td>
<td>022760</td>
<td>Set switches 13, 10, and 8 through 4 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010060</td>
<td>001000</td>
<td>Set switch 9 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010062</td>
<td>000014</td>
<td>Set switches 3 and 2 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010064</td>
<td>001403</td>
<td>Set switches 9, 8, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010066</td>
<td>000005</td>
<td>Set switches 2 and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010070</td>
<td>000167</td>
<td>Set switches 6, 5, 4, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010072</td>
<td>177704</td>
<td>Set switches 15 through 6 and 2 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010074</td>
<td>005007</td>
<td>Set switches 11, 9, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>Location</td>
<td>Contents</td>
<td>Like This</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td>010000</td>
<td>012700</td>
<td>Set switches 12, 10, 8, 7, and 6 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010002</td>
<td>172524</td>
<td>Set switches 15 through 12, 10, 8, 6, 4, and 2 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010004</td>
<td>005310</td>
<td>Set switches 11, 9, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010006</td>
<td>012740</td>
<td>Set switches 12, 10, and 8 through 5 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010010</td>
<td>060011</td>
<td>Set switches 14, 13, 3, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010012</td>
<td>105710</td>
<td>Set switches 15, 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010014</td>
<td>100376</td>
<td>Set switches 15 and 7 through 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010016</td>
<td>005710</td>
<td>Set switches 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010020</td>
<td>100767</td>
<td>Set switches 15, 8 through 4, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010022</td>
<td>012710</td>
<td>Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010024</td>
<td>060003</td>
<td>Set switches 14, 13, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

(continued on next page)
Table A-5 (Cont.)
TM11 Bootstrap Loader

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>010026</td>
<td>105710</td>
<td>Set switches 15, 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010030</td>
<td>100376</td>
<td>Set switches 15 and 7 through 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010032</td>
<td>005710</td>
<td>Set switches 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010034</td>
<td>100777</td>
<td>Set switches 15 and 8 through 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>010036</td>
<td>005007</td>
<td>Set switches 11, 9, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

2. Verify that the bootstrap program has been deposited properly as follows:

a. Set the starting address, 010000, in the Switch Register as in Step 1a above.

b. Press the LOAD ADDR switch.

c. Display the contents of that address in the Data Register by pressing the EXAM switch.

d. Compare the number in the Data Register with the value in the appropriate table (A-4 or A-5).

e. If they are the same, repeat Step 2c until all words have been examined.

f. If not the same, repeat Step 1.

3. If TM11 magtape is being used, ensure that the magtape is positioned at the load point; if it is not, manually rewind the magtape.

4. Set the starting address, 010000, in the Switch Register as in Step 1a above, then press the LOAD ADDR switch.

5. Set the ENABLE/HALT switch to ENABLE, then press the START switch.
A.5  TALL CASSETTE BOOTSTRAP LOADER

Two cassette bootstraps are available: CBOOT (Table A-6) is the standard version, QCBOOT (Table A-7) is a shorter version that may optionally be loaded. QCBOOT does not provide some of the error checking and handling that the longer CBOOT does, but it allows a faster means of manually booting the system.

1. Deposit the cassette software bootstrap into memory as follows:
   a. Set the first address, 001000, in the Switch Register. (Set switch 9 to the up (1) position and all others to the down (0) position.)
   b. Press the LOAD ADDR switch.
   c. Set the proper contents for CBOOT (Table A-6) or QCBOOT (Table A-7) in the Switch Register and lift the DEP switch.
   d. Repeat Step c until all values have been deposited.

Table A-6
Tall CBOOT Bootstrap Loader

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>001000</td>
<td>012700</td>
<td>Set switches 12, 10, 8, 7, and 6 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001002</td>
<td>177500</td>
<td>Set switches 15 through 8, and 6 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001004</td>
<td>005010</td>
<td>Set switches 11, 9, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001006</td>
<td>010701</td>
<td>Set switches 12, 8, 7, 6, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001010</td>
<td>062701</td>
<td>Set switches 14, 13, 10, 8, 7, 6, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001012</td>
<td>000052</td>
<td>Set switches 5, 3, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

(continued on next page)
Table A-6 (Cont.)
TALL CBOOT Bootstrap Loader

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>001014</td>
<td>012702</td>
<td>Set switches 12, 10, 8, 7, 6, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001016</td>
<td>000375</td>
<td>Set switches 7 through 2 and 0 to the up position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001020</td>
<td>112103</td>
<td>Set switches 15, 12, 10, 6, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001022</td>
<td>112110</td>
<td>Set switches 15, 12, 10, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001024</td>
<td>100413</td>
<td>Set switches 15, 8, 3, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001026</td>
<td>130310</td>
<td>Set switches 15, 13, 12, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001030</td>
<td>001776</td>
<td>Set switches 9 through 1 in the up (1) position. Set all others in the down (0) position.</td>
</tr>
<tr>
<td>001032</td>
<td>105202</td>
<td>Set switches 15, 11, 9, 7, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001034</td>
<td>100772</td>
<td>Set switches 15, 8 through 3, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001036</td>
<td>116012</td>
<td>Set switches 15, 12, 11, 10, 3, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001040</td>
<td>000002</td>
<td>Set switch 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>001042</td>
<td>120337</td>
<td>Set switches 15, 13, 7, 6, and 4 through 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001044</td>
<td>000000</td>
<td>Set all switches to the down (0) position.</td>
</tr>
<tr>
<td>001046</td>
<td>001767</td>
<td>Set switches 9 through 4, and 2 through 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001050</td>
<td>000000</td>
<td>Set all switches to the down (0) position.</td>
</tr>
<tr>
<td>001052</td>
<td>000755</td>
<td>Set switches 8 through 5, 3, 2, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001054</td>
<td>005710</td>
<td>Set switches 11, 9 through 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001056</td>
<td>100774</td>
<td>Set switches 15, and 8 through 2 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001060</td>
<td>005007</td>
<td>Set switches 11, 9, and 2 through 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001062</td>
<td>017640</td>
<td>Set switches 12 through 7 and 5 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001064</td>
<td>002415</td>
<td>Set switches 10, 8, 3, 2, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001066</td>
<td>112024</td>
<td>Set switches 15, 12, 10, 4, and 2 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>
## Table A-7
### TALL QCBOOT Bootstrap Loader

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>001000</td>
<td>012700</td>
<td>Set switches 12, 10, 8, 7, and 6 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001002</td>
<td>177500</td>
<td>Set switches 15 through 8, and 6 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001004</td>
<td>005010</td>
<td>Set switches 11, 9, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001006</td>
<td>010701</td>
<td>Set switches 12, 8, 7, 6, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001010</td>
<td>062701</td>
<td>Set switches 14, 13, 10, 8, 7, 6, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001012</td>
<td>000034</td>
<td>Set switches 4, 3, and 2 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001014</td>
<td>112102</td>
<td>Set switches 15, 12, 10, 6, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001016</td>
<td>112110</td>
<td>Set switches 15, 12, 10, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001020</td>
<td>032710</td>
<td>Set switches 13, 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001022</td>
<td>100240</td>
<td>Set switches 15, 7, and 5 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001024</td>
<td>001775</td>
<td>Set switches 9 through 2 and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

(continued on next page)
### Table A-7 (Cont.)
**Tall QCBOOT Bootstrap Loader**

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>001026</td>
<td>100001</td>
<td>Set switches 15 and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001030</td>
<td>005007</td>
<td>Set switches 11, 9, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001032</td>
<td>005202</td>
<td>Set switches 11, 9, 7, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001034</td>
<td>100770</td>
<td>Set switches 15 and 8 through 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001036</td>
<td>116012</td>
<td>Set switches 15, 12, 11, 10, 3, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001040</td>
<td>000002</td>
<td>Set switch 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001042</td>
<td>000766</td>
<td>Set switches 8 through 4, 2, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001044</td>
<td>017775</td>
<td>Set switches 12 through 2, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001046</td>
<td>002415</td>
<td>Set switches 10, 8, 3, 2, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

2. Verify that the bootstrap is properly in memory as follows:
   a. Set the first address in the Switch Register as in Step 1a above.
   b. Press the LOAD ADDR switch.
   c. Display the contents of that address in the Data Register by pressing the EXAM switch.
   d. Compare the number in the Data Register with the value in the appropriate table (A-6 or A-7).
   e. If they are the same, repeat Step 2c until all words have been examined.
INSTRUCTIONS FOR LOADING SOFTWARE BOOTSTRAPS

f. If not the same, repeat Step 1.

3. Set the starting address in the Switch Register as in Step 1a above.

4. Press the LOAD ADDR switch.

5. Press the rewind button on the cassette drives.

6. Press the START switch.

A.6 PC11 PAPER TAPE BOOTSTRAP LOADER

1. Checking the Bootstrap Loader

   a. Set the ENABLE/HALT switch to HALT to stop any previous program that may be running.

   b. Set the ENABLE/HALT switch to ENABLE.

   c. Set the first address, 37744, in the Switch Register. (Set switches 13 through 5 and switch 2 to the up (1) position; set all other switches to the down (0) position.)

   d. Press the LOAD ADDR switch.

   e. Press the EXAM switch. The data is displayed in the Data Register.

   f. Compare the data displayed to the value given in Table A-8.

   g. If they are the same, repeats Steps e through f until the entire Bootstrap Loader has been checked. If they are not the same, go to Step 2 and enter the loader correctly. If all locations are the same as in Table A-8, the Bootstrap Loader is correctly in memory and Step 2 can be bypassed.

2. Deposit the Bootstrap Loader into memory as follows:

   a. Set the first address, 37744, in the Switch Register as in Step 1c above.

   b. Press the LOAD ADDR switch.

   c. Set the proper contents from Table A-8 in the Switch Register and lift the DEP switch.

   d. Repeat Step 2c until all the instructions have been deposited.
## INSTRUCTIONS FOR LOADING SOFTWARE BOOTSTRAPS

### Table A-8
**Paper Tape Bootstrap Loader**

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>37744</td>
<td>016701</td>
<td>Set switches 12, 11, 10, 8, 7, 6, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>37746</td>
<td>000026</td>
<td>Set switches 4, 2, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>37750</td>
<td>012702</td>
<td>Set switches 12, 10, 8, 7, 6, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>37752</td>
<td>000352</td>
<td>Set switches 7, 6, 5, 3, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>37754</td>
<td>005211</td>
<td>Set switches 11, 9, 7, 3, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>37756</td>
<td>105711</td>
<td>Set switches 15, 11, 9, 8, 7, 6, 3, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>37760</td>
<td>100376</td>
<td>Set switches 15 and 7 through 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>37762</td>
<td>116162</td>
<td>Set switches 15, 12, 11, 10, 6, 5, 4, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>37764</td>
<td>000002</td>
<td>Set switch 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>37766</td>
<td>037400</td>
<td>Set switches 13 through 8 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>37770</td>
<td>005267</td>
<td>Set switches 11, 9, 7, 5, 4, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

(continued on next page)
### Table A-8 (Cont.)
**Paper Tape Bootstrap Loader**

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>37772</td>
<td>177756</td>
<td>Set switches 15 through 5, 3, 2, and 1 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>37774</td>
<td>000765</td>
<td>Set switches 8 through 4, 2, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>37776</td>
<td>177550</td>
<td>Set switches 15 through 8, 6, 5, and 3 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>

Repeat Step 1 to verify that the Bootstrap Loader has been correctly deposited.
APPENDIX B

FORMATTING THE RK05 DISK

The following instructions detail the procedure for formatting the RK05 DECPack disk for use on the RK11.

1. Mount the disk to be formatted in Unit 0. The following formatting procedure will work only on Unit 0.

2. Set the ENABLE/HALT switch to HALT to stop any previous program which may be running.

3. Deposit the formatting program into memory as follows:
   a. Set the first address, 001000, in the Switch Register. (Set switch 9 to the up (1) position; set all others to the down (0) position.)
   b. Press the LOAD ADDR switch.
   c. Set the proper contents from Table B-1 in the Switch Register and lift the DEP switch.
   d. Repeat Step c until all the values have been deposited.

4. Verify that the formatting program is properly in memory as follows:
   a. Set the starting address in the Switch Register as in Step 3a above.
   b. Press the LOAD ADDR switch.
   c. Display the contents of that address in the Data Register by pressing the EXAM switch.
   d. Compare the number in the Data Register with the value in Table B-1.
   e. If they are the same, repeat Step 4c until all words have been examined.
   f. If they are not the same, repeat from Step 1.

5. Set the starting address in the Switch Register as in Step 3a above, then press the LOAD ADDR switch.

6. Set the ENABLE/HALT switch to ENABLE.
7. Verify that the disk mounted in Unit 0 is the one to be formatted. If so, WRITE ENABLE Unit 0.

8. Press the START switch. Let the program execute for 60 seconds, then set the ENABLE/HALT switch to HALT to stop the program.

The disk is now formatted and ready for use.

<table>
<thead>
<tr>
<th>Location</th>
<th>Contents</th>
<th>Like This</th>
</tr>
</thead>
<tbody>
<tr>
<td>001000</td>
<td>012737</td>
<td>Set switches 12, 10, 8, 7, 6, and 4 through 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001002</td>
<td>006003</td>
<td>Set switches 11, 10, 1, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001004</td>
<td>177404</td>
<td>Set switches 15 through 8 and 2 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001006</td>
<td>105737</td>
<td>Set switches 15, 11, 9 through 6, and 4 through 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001010</td>
<td>177404</td>
<td>Set switches 15 through 8 and 2 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001012</td>
<td>100375</td>
<td>Set switches 15, 7 through 2, and 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001014</td>
<td>000137</td>
<td>Set switches 6 and 4 through 0 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
<tr>
<td>001016</td>
<td>001000</td>
<td>Set switch 9 to the up (1) position. Set all others to the down (0) position.</td>
</tr>
</tbody>
</table>
INDEX

Altering magtape density, 1
TJUL6 magtape, 4-18
TM11, 4-17
ASEMLB
assembling, 5-6
linking, 5-6
Assembling,
ASEMLB, 5-6
BASIC/RT-11, 5-48
BATCH, 5-9
CREF, 5-6
DUMP, 5-9
EDIT, 5-5
EXPAND, 5-6
FILEX, 5-8
FORTRAN, 5-18
LIBR, 5-7
LINK, 5-7
MACRO, 5-5
MBUILD, 5-17
ODT, 5-9
PATCH, 5-9
PATCHO, 5-9
PIF, 5-8
SRCCOM, 5-8
SYSLIB, 5-11
system files, 5-1
VTLIB, 5-11
Assembly instructions, 5-1

\ (backslash) character, viii
BA.OBJ, 4-14
Bare machine OTS assembly, 5-33
BAS8K.SAV, 1-4
BASICE.MAC, 5-47
BASICE.OBJ, 1-5
BASICR.MAC, 5-47
BASICR.OBJ, 1-5
BASIC.MAC, 5-47
BASICX.MAC, 5-47
BASICX.OBJ, 1-5
BASNS1.OBJ, 1-5
BASNSX.OBJ, 1-5
BATCH,
assembling, 5-9
linking, 5-9
BATCH.OBJ, 4-14
BIN:., 5-18
Bootable magtapes, 4-9
Bootstrap loader,
see software bootstraps
Bootstraps, software, A-1
see also software bootstraps
} (brace), 4-28
Building FORLIB, 2-60
from cassette, 2-45
from DECpack disk, 2-6
from DEXTape, 2-17
Building from cassette,
BASIC/RT-11, 2-48
FORTRAN, 2-44
RT-11, 2-42, 4-9
Building from DEXTape disk, 2-2
BASIC/RT-11, 2-9
FORTRAN IV, 2-5
RT-11, 2-2, 4-4
Building from DEXTape,
BASIC/RT-11, 2-20
FORTRAN IV, 2-15
RT-11, 2-10, 4-3
Building from diskette,
BASIC/RT-11, 2-31
FORTRAN IV, 2-26
RT-11, 2-21, 4-6
Building from magtape,
BASIC/RT-11, 2-41
FORTRAN IV, 2-37
RT-11, 2-32, 4-6
Building from paper tape,
BASIC/RT-11, 2-63
FORTRAN IV, 2-57
RT-11, 2-49, 4-12

BASIC/RT-11 (cont.),
PPMP assembly parameters, 5-50
linking, 5-50, 5-51
LPS functions, 1-5
LV-11 package, 1-5
overlaid versions, 1-5
nonoverlaid versions, 1-5
software kit overview, 1-4
VT11 support, 1-5
BASIC.SAV, 1-4
BASICX.MAC, 5-47
BASICX.OBJ, 1-5
BASNS1.OBJ, 1-5
BASNSX.OBJ, 1-5
BATCH,
assembling, 5-9
linking, 5-9
BATCH.OBJ, 4-14
BIN:., 5-18
Bootable magtapes, 4-9
Bootstrap loader,
see software bootstraps
Bootstraps, software, A-1
see also software bootstraps
} (brace), 4-28
Building FORLIB, 2-60
from cassette, 2-45
from DECpack disk, 2-6
from DEXTape, 2-17
Building from cassette,
BASIC/RT-11, 2-48
FORTRAN, 2-44
RT-11, 2-42, 4-9
Building from DEXTape disk, 2-2
BASIC/RT-11, 2-9
FORTRAN IV, 2-5
RT-11, 2-2, 4-4
Building from DEXTape,
BASIC/RT-11, 2-20
FORTRAN IV, 2-15
RT-11, 2-10, 4-3
Building from diskette,
BASIC/RT-11, 2-31
FORTRAN IV, 2-26
RT-11, 2-21, 4-6
Building from magtape,
BASIC/RT-11, 2-41
FORTRAN IV, 2-37
RT-11, 2-32, 4-6
Building from paper tape,
BASIC/RT-11, 2-63
FORTRAN IV, 2-57
RT-11, 2-49, 4-12

Index-1  July 1976
INDEX (Cont.)

Building RT-11
see RT-11, building
Building SYSLIB, 5-17
Building VTLIB, 5-11

Card reader, installing, 4-23
Cassette,
building from, 2-42, 4-9
<CR> (Carriage return), viii
CBOOT, A-14
CBUILD, 1-1, 2-48, 4-9
switches, 4-10
Changing DUMP default output
device, 4-32
Clock rate, 50-cycle, 4-20
Common OTS modules, assembling,
5-20
Compiling PATCHO, 5-10
Concatenating OTS modules,
5-46
Console devices, serial, 4-16
CR11 card reader, 4-23
CREF,
assembling, 5-6
linking, 5-6
modifying line count in, 4-31
CREF.OBJ, 4-14
Customization,
special hardware, 4-16
system, 4-1
50-cycle clock rate, specifying,
4-20

DECPack disk, building from,
2-2 - 2-10, 4-4
DECTape, building from, 2-10 -
2-21, 4-3
Default output device, DUMP, 4-32
DEMO.BAS, 3-18
DEMOBG.MAC, 3-1
DEMOPG.MAC, 3-10
DEMO.FOR, 1-3, 3-15
Demonstration program,
BASIC/RT-11, 3-18
foreground/background monitor,
3-10
FORTRAN IV, 3-15
single-job monitor, 3-1
Density, altering magtape, 4-17
Device handlers, 1-3
deleting from system device,
2-4, 2-14, 2-25, 2-35
Directory extension program,
4-14
Diskette,
building from, 2-21 - 2-32
second handler, 4-25

Documentation,
conventions, vii
instructions, vii
DUMP,
assembling, 5-9
default output device, 4-32
linking, 5-9
DUMP.OBJ, 4-14

EAEO OTS assembly, 5-41
EDIT, 3-4
assembling, 5-5
linking, 5-5
reducing size of text window
displayed, 4-27
EDIT.OBJ, 4-14
EIS OTS assembly, 5-36
EXPAND,
assembling, 5-6
linking, 5-6
Extensions kit, FORTRAN, 1-4

File size, upper limit, 4-28
FILEX,
assembling, 5-8
linking, 5-8
FILEX.OBJ, 4-14
FIS OTS assembly, 5-38
Floating point math package,
BASIC/RT-11, 5-49
Floating vector locations, 4-24
Foreground terminal, installing,
4-31
Foreground/Background Monitor,
4-34
FORLIB,
assembling, 5-30
building, 2-60
building from cassette, 2-45
building from DECPack disk,
2-6
building from DECTape, 2-17
building from diskette, 2-27
building from paper tape,
2-57
building on magtape system,
2-38
FORLIB.V2NS, 5-30
Formatting RK05 disk, B-1
FORTRAN IV, 1-3
building from cassette, 2-44
building from DECPack disk,
2-5
building from DECTape, 2-15
building from diskette, 2-26
building from magtape, 2-37
INDEX (Cont.)

FORTRAN IV, (Cont.)
building from paper tape, 2-57
building the library, 5-46
demonstration program, 3-15
extensions kit, 1-4
linking object modules, 2-59
software kit overview, 1-3

FORTRAN IV, assembling
bare machine, 5-33
custom OTS, 5-20
EAE OTS, 5-41
ETS OTS, 5-36
FIS OTS, 5-38
FPU OTS, 5-33
hardware dependent, 5-33
subscripting modules, 5-30
V2NS OTS, 5-30

FORTRAN IV compiler,
assembling, 5-18
linking, 5-19

FPMP assembly parameters, 5-50
FPMP.EAE, 1-5
FPMP.EIS, 1-5
FPMP.FPU, 1-5
FPMP.OBJ, 1-5
FPMP.MAC, 5-47
FPU OTS assembly, 5-43

General building instructions,
2-1, 4-1
GT40 display processor, 3-2

Hardware dependent OTS modules
assembly, 5-33
High-baud rate serial console
device, 4-16

Initializing
DECTapes, 4-4
disks, 4-4
Installing
card reader, 4-23
foreground terminal, 4-31
second diskette handler, 4-25
Interfacing
RJS03/4 disks, 4-21
RP03 disks, 4-21

KB.MAC, 4-31
Kit overview, 1-1

LA36, 4-28
Left bracket, [, viii
<LF> (line feed), viii
LIBR, 1-4
assembling, 5-7
linking, 2-53, 5-7
Libraries, macro, 1-3
LINK, 4-14
assembling, 5-7
linking, 2-53, 5-7
Link instructions, 5-1
Linker, see LINK
Linking,
ASEMBL, 5-6
BASIC/RT-11, 5-50, 5-51
BATCH, 5-9
CREF, 5-6
DUMP, 5-9
EDIT, 5-5
EXPAND, 5-6
FILEX, 5-8
FORTRAN IV compiler, 5-19
LIBR, 2-53, 5-7
LINK, 2-53, 5-7
MACRO, 5-5
MBUILD, 5-17
ODT, 5-9
PATCH, 5-9
PATCHO, 5-10
PIP, 5-8
SRCCOM, 5-8
system files, 5-4
Linking BASIC/RT-11,
with overlays, 5-51
with user functions, 5-51 - 5-53
without overlays, 5-51
LINKV2, 1-4
Loader, bootstrap,
see software bootstraps
LST:, 5-18

MACRO,
assembling, 5-5
libraries, 1-3
linking, 5-5
modifying line count in, 4-31
Magtape,
altering density, 4-17
bootable, 4-9
bootstraps, A-8 - A-13
building from, 2-32 - 2-42
4-6
MBOOT.BOT, 4-6
MBUILD, 1-1, 2-33
assembling, 5-17
file transfer limitation, 4-7
linking, 5-17
switches, 4-8

Index-3
INDEX (Cont.)

MBUILD.MM1, 2-33, 4-6
MBUILD.MM2, 2-33, 4-7
MBUILD.MTL, 2-33, 4-6
MBUILD.MT2, 2-33, 4-6
Modifying line count in
CREF, 4-31
MACRO, 4-31
Modules, object, 2-52
Monitor files,
   deleting from system device,
      2-4, 2-14, 2-25, 2-34
MSBOOT.BOT, 2-33, 4-6
MTINIT.SAV, 4-7, 4-9

Object modules, 2-52,
   deleting from system device,
      2-56
ODT,
   assembling, 5-9
   linking, 5-9
OLDPIP, 2-51, 4-12, 4-13
Optimizing system device, 4-33
OTSV2, 1-4
OTSV2S, 1-4
Overview, software kit
   see Software kit overview

Paper tape
   bootstrap, A-19
   building from 2-49 - 2-65,
      4-12
PATCH
   assembling, 5-9
   linking, 5-9
PATCHO, 1-4, 4-14
   compiling, 5-10
   linking, 5-10
PATCH.OBJ, 2-51
PC11 paper tape bootstrap, A-19
PIP,
   assembling, 5-8
   linking, 5-8
   switches, 4-4
PIPIB.OBJ, 4-13
Platter specification, RF11 disk, 4-19
PREPAS.OBJ, 4-13
PREEXEC.OBJ, 4-13
PT BUILD, 1-1, 2-49, 4-12

QCBOOT, A-17

RF11 disk, platter specification,
   4-19

RFBTBF.OBJ, 4-14
Right bracket ], viii
RJS03/4 disks, interfacing, 4-21
RK05 disk, formatting, B-1
RK11 DECpack
   bootstrap, A-1
   building from, 2-2 - 2-10
RP03 disks, interfacing, 4-21
RT-11,
   building 8K systems, 4-6
   building from cassette, 2-42
   building from DECpack disk,
      2-2
   building from DECtape, 2-10
   building from diskette, 2-21
   building from magtape, 2-32
   building from paper tape, 2-49
   foreground/background monitor,
      3-10
   monitors, 1-2, 3-1, 4-2
   software kit overview, 1-1
   using less memory than available,
      4-29
RT11JS.OBJ, 4-14
RTEXEC.OBJ, 4-13
RTMAC.OBJ, 4-13
RTFPST.OBJ, 4-13
RX11/RX01 Diskette bootstrap,
   bootstrap, A-5
   building from, 2-21, 4-6

Second diskette handler,
   installing, 4-25
Serial console devices, 4-16
Single-disk systems, accessing
   nonsystem disks on, 4-30
Single-job monitor, 4-34
   demonstration program, 3-1
SMEXEC.OBJ, 4-13
SMMAC.OBJ, 4-13
SMFPSB.OBJ, 4-13
Software bootstraps,
   paper tape, A-20
   RK11 DECpack, A-2
   RK11 diskette, A-5
   TALL cassette, A-14, A-17
   TC11 DECtape, A-3
   TJU16 magtape, A-9
   TM11 magtape, A-12
Software kit overview,
   BASIC/RT-11, 1-4
   FORTRAN IV, 1-3
   RT-11, 1-1
   Special hardware, customization
      for, 4-16

Index-4
INDEX (Cont.)

Specifying 50-cycle clock rate, 4-20
SRC:; 5-18
SRCCOM,
    assembling, 5-8
    linking, 5-8
SRCCOM.OBJ, 4-14
Subscripting modules assembly, 5-30
Switches,
    CBUILD, 4-10
    MBUILD, 4-8
Switching between Single-Job and Foreground/Background monitors, 4-34
SYSP4.OBJ, 1-1, 1-3, 4-14
SYSLIB, 1-3, 4-14
    assembling, 5-11
    building, 5-17
SYSMAC.8K, 1-1, 1-3, 4-13
SYSMAC.SML, 1-3, 4-13
System,
    customization, 4-1
    device, optimizing, 4-33
    files, 1-1
    macro libraries, 1-3
    programs, 1-1, 1-3
System files,
    assembling, 5-1
    linking, 5-4
System subroutine library, see SYSLIB

TAG cassette bootstrap, A-14
CBOOT, A-14
QCBOOT, A-17
<TAB> character, viii
TC11 DECtape bootstrap, A-2
TJU16 magtape,
    altering density, 4-18
    bootstrap loader, A-9
TM11 magtape,
    altering density, 4-17
    bootstrap loader, A-12

.UNLOAD command, 4-33
    using with foreground active, 4-33
Uparrow, ‡, viii
Upper limit, file size, 4-28
User functions, BASIC/RT-11, 5-51

V2S OTS assembly, 5-31
Vector locations, floating, 4-24
VT11,
    changing floating vector locations, 4-24
    display handler library, 5-11
    display processor, 3-2
VTDLR.OBJ, 1-1, 1-3
VTLEDT.OBJ, 4-14
VTLIB, assembling, 5-11
    building, 5-11
VTMAC.MAC, 1-1, 1-3, 4-13
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________________________