

# Introduction to FEP and FRP

AA-J302B-TC

**August 1982**

This document describes the FEP (FORTRAN Enhancement Package) software packages — RT-11/FEP and RSX-11M/FEP and the FRP (FORTRAN Real-Time Package) software packages — RT-11/FRP and RSX-11M/FRP. It also describes the hardware components required to run the FEP and FRP packages. Note that FRP is FEP without the graphics (RGL/11) capability.

|                          |  |
|--------------------------|--|
| <b>OPERATING SYSTEM:</b> | RT-11, V4.0<br>RSX-11M, V4.0   |
| <b>SOFTWARE:</b>         | RT-11/FEP, V2.1<br>RSX-11M/FEP, V1.1<br>RT-11/FRP, V1.0<br>RSX-11M/FRP, V1.0 |

Software and manuals should be ordered by title and order number. In the United States, send orders to the nearest distribution center. Outside the United States, orders should be directed to the nearest DIGITAL Field Sales Office or representative.

**Northeast/Mid-Atlantic Region**

Digital Equipment Corporation  
PO Box CS2008  
Nashua, New Hampshire 03061  
Telephone:(603)884-6660

**Central Region**

Digital Equipment Corporation  
Accessories and Supplies Center  
1050 East Remington Road  
Schaumburg, Illinois 60195  
Telephone:(312)640-5612

**Western Region**

Digital Equipment Corporation  
Accessories and Supplies Center  
632 Caribbean Drive  
Sunnyvale, California 94086  
Telephone:(408)734-4915

**First Printing, October 1981**  
**Updated, January 1982**  
**Revised, August 1982**


Copyright ©, 1981, 1982, Digital Equipment Corporation. All Rights Reserved.

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 only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

|   |              |         |
|---|--------------|---------|
| DEC   | DECnet       | IAS     |
| DECUS   | DECsystem-10 | MASSBUS |
| DECSYSTEM-20  | PDT          | PDP     |
| DECwriter   | RSTS         | UNIBUS  |
| DIBOL   | RSX          | VAX     |
| EduSystem   | VMS          | VT      |
|  | RT           |         |

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.

---

# Contents

## Preface

## Section 1 Introduction

- 3 **Capabilities**
- 4 Transferring Data
- 5 Storing Data
- 5 Processing Data
- 5 Real-Time Signal Processing
- 6 Statistical Analysis
- 7 Mathematical Analysis
- 7 Synchronizing Processes
  
- 8 **Interacting with the Computer System**
- 8 **Programming Tools**
  
- 11 **RT-11 Operating System**
- 11 System Configurations
- 12 System Features
- 13 RT-11 System Utility Programs
- 14 **RSX-11M Operating System**
- 14 System Features
- 15 RSX-11M Utility Programs
- 16 **FORTRAN Language**
- 16 FORTRAN IV/RT-11
- 17 FORTRAN 77
- 18 **MACRO-11**
- 19 **REAL-11/MNC**
- 22 Instrument Bus Subroutines
- 25 Laboratory Subroutines Package
- 26 Scientific Subroutines Package
- 26 **RGL/11**
- 28 FORTRAN Debugging Technique

## Section 2 Software

---

## **Section 3 Hardware**

|    |                                    |
|----|------------------------------------|
| 33 | <b>Central Processing Units</b>    |
| 33 | PDP-11/03                          |
| 34 | PDP-11/23                          |
| 34 | <b>System Disks</b>                |
| 34 | RX02 Dual Floppy Disk              |
| 34 | RL01                               |
| 34 | RL02                               |
| 35 | <b>Terminals</b>                   |
| 35 | VT105                              |
| 35 | VT125                              |
| 36 | LA36                               |
| 36 | LA38                               |
| 36 | <b>Printers</b>                    |
| 36 | LA120 DECwriter III                |
| 36 | LA34-VA                            |
| 37 | <b>Real-Time I/O</b>               |
| 37 | IEEE Instrument Bus Interface      |
| 37 | Serial Line Units                  |
| 38 | MNC-Series Modules                 |
| 39 | Programmable Clock Module          |
| 40 | Analog-to-Digital Converter Module |
| 40 | Analog Multiplexer Module          |
| 41 | Preamplifier Module                |
| 41 | Thermocouple Preamplifier Module   |
| 42 | Digital-Input Module               |
| 43 | Digital-Output Module              |
| 44 | Digital-to-Analog Converter Module |

---

## Section 4 Documentation

- 47 **RT-11/FEP, RT-11/FRP, RSX-11M/FEP,  
and RSX-11M/FRP Manuals**
- 47 Introduction to FEP and FRP
- 47 REAL-11/MNC FORTRAN Programmer's  
Reference Manual
- 47 Scientific Subroutines Programmer's Reference Manual
- 47 Laboratory Subroutines Programmer's Reference Manual
  
- 48 **RT-11/FEP and RT-11/FRP Manuals**
- 48 RT-11/FEP and RT-11/FRP Installation and User's Guide
- 48 Instrument Bus Subroutines Programmer's  
Reference Manual
- 48 FORTRAN Debugging Technique Reference Manual
- 48 Release Notes for RT-11/FEP and RT-11/FRP  
FORTRAN Enhancement Package for MINC, Version 2.0
- 48 RT-11 Documentation Set
- 48 PDP-11 FORTRAN IV Documentation Set
  
- 49 **RSX-11M/FEP and RSX-11M/FRP Manuals**
- 49 RSX-11M/FEP and RSX-11M/FRP Installation  
and User's Guide
- 49 Release Notes for RSX-11M/FEP and RSX-11M/FRP  
FORTRAN Enhancement Package for MINC, Version 1.0
- 49 RSX-11M Documentation Set
- 49 PDP-11 FORTRAN 77 Documentation Set
- 49 RT-11/FEP and RSX-11M/FEP Graphics Manual
- 49 RGL/11 Programmer's Reference Manual

- 3 Table 1. FEP and FRP Software Components
- 19 Table 2. REAL-11/MNC Subprograms
- 22 Table 3. Instrument Bus Routines
- 25 Table 4. Laboratory Subroutine Package Routines
- 26 Table 5. Scientific Subroutines Package Routines
- 28 Table 6. FORTRAN Debugging Technique Commands
- 38 Table 7. MINC-Series Modules

## Tables



## Overview

This manual introduces the FORTRAN Enhancement Package (FEP) and the FORTRAN Real-Time Package (FRP) and the software packages that include them — RT-11/FEP and RT-11/FRP plus RSX-11M/FEP and RSX-11M/FRP. Note that FRP is FEP without graphics capability. The manual describes the packages' capabilities, software components, and documentation, as well as the hardware with which FEP and FRP are used.

## Manual Conventions

In this manual a reference to FEP includes both RT-11/FEP and RSX-11M/FEP. A reference to FRP includes both RT-11/FRP and RSX-11M/FRP.

## Intended Reader

If you are a MINC BASIC user who has recently purchased one of the FEP or FRP packages, you should read this manual before reading any others. If you are a new FEP or FRP user who has little knowledge of the MINC hardware or FEP or FRP software, you also should read this manual before any others. However, before reading the manual, you should be familiar with the FORTRAN programming language, scientific computer applications, and the concept of a computer operating system (such as RT-11 or RSX-11M). If you are unfamiliar with such basic information, read your operating system's introductory manual before reading the *Introduction to FEP and FRP*. (The introductory RT-11 manual is the *Introduction to RT-11*; the introductory RSX-11M manual is the *Introduction to RSX-11M and RSX-11M-PLUS*.)

## Related Documents

If you are an RT-11/FEP or RT-11/FRP user, read the *RT-11/FEP and RT-11/FRP Installation and User's Guide* after you complete the *Introduction to FEP and FRP*. If you are a RSX-11M/FEP or RSX-11M/FRP user, read the *RSX-11M/FEP and RSX-11M/FRP Installation and User's Guide* after you complete this manual.

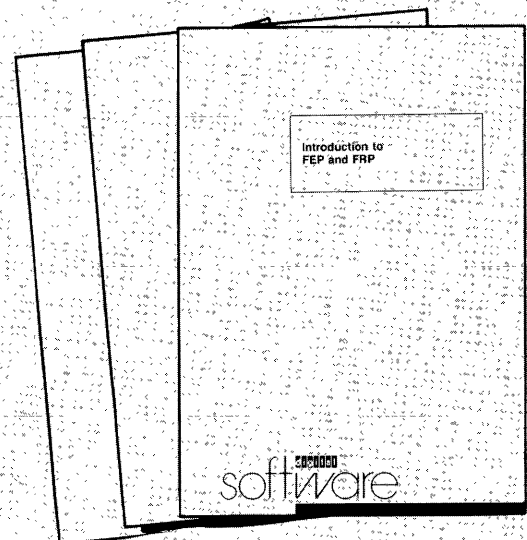
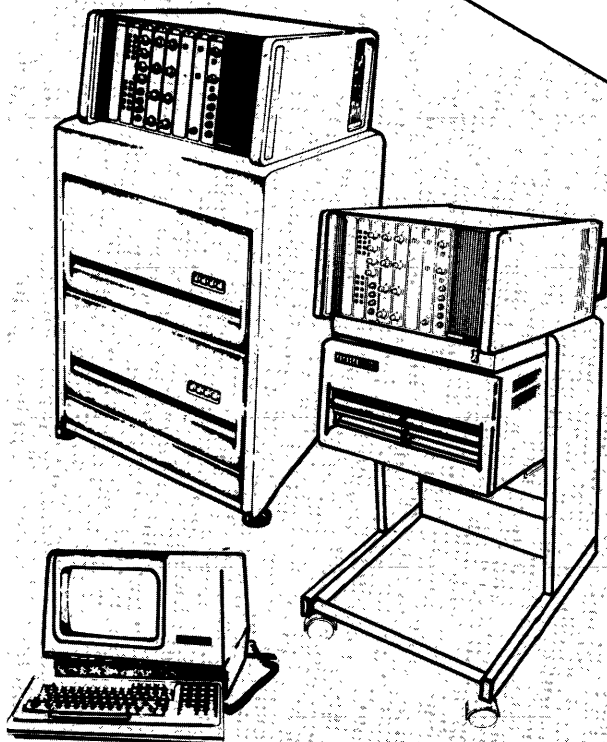




# Section 1 Introduction

FEP and FRP Software Components

| Component   | FEP   |         | FRP   |         |
|---|-------|---------|-------|---------|
|   | RT-11 | RSX-11M | RT-11 | RSX-11M |
| Programming Languages<br>FORTRAN IV<br>FORTRAN 77 | •     | •       | •     | •       |
| FORTRAN Debugging<br>FDT                          | •     |         | •     |         |
| Real-Time Data Transfers<br>REAL-11/MNC           | •     | •       | •     | •       |
| Instrument Bus Data Transfers<br>IBS              | •     |         | •     |         |
| Signal Data Processing<br>LSP                     | •     | •       | •     | •       |
| Statistics<br>SSP                                 | •     | •       | •     | •       |
| Graphics<br>RGL/11                                | •     | •       |       |         |





# Section 1 Introduction

The FORTRAN Enhancement Package (FEP) and the FORTRAN Real-Time Package (FRP) constitute a group of software components for MINC users whose needs exceed the capabilities of MINC BASIC. Because programs that use FEP or FRP laboratory software must be written in FORTRAN, users of FEP or FRP must have a greater degree of computer expertise than MINC BASIC requires. However, FORTRAN programs that use FEP or FRP laboratory software can be larger, more flexible, and faster-running than similar programs written in MINC BASIC.

FEP and FRP include a PDP-11 operating system, a programming language, and FORTRAN-callable software for laboratory applications. Table 1 lists the components included in each of the two packages. The RGL/11 routines are not available in the FRP package. Table 1 shows the FEP and FRP components.

**Table 1. FEP and FRP Software Components**

| Component                     | FEP   |         | FRP   |         |
|-------------------------------|-------|---------|-------|---------|
|                               | RT-11 | RSX-11M | RT-11 | RSX-11M |
| Programming Languages         |       |         |       |         |
| FORTRAN IV                    | •     |         | •     |         |
| FORTRAN 77                    |       | •       |       | •       |
| FORTRAN Debugging             |       |         |       |         |
| FDT                           | •     |         | •     |         |
| Real-Time Data Transfers      |       |         |       |         |
| REAL-11/MNC                   | •     | •       | •     | •       |
| Instrument Bus Data Transfers |       |         |       |         |
| IBS                           | •     |         | •     |         |
| Signal Data Processing        |       |         |       |         |
| LSP                           | •     | •       | •     | •       |
| Statistics                    |       |         |       |         |
| SSP                           | •     | •       | •     | •       |
| Graphics                      |       |         |       |         |
| RGL/11                        | •     | •       |       |         |

MR-S-2213-82

## Capabilities

FEP and FRP capabilities include:

- Transferring data
- Storing data
- Processing data
- Synchronizing processes

---

### **Transferring Data**

The FEP and FRP software can transfer data between the MINC computer system and laboratory instruments. Data transfers can occur over any of the following MINC hardware modules:

- The optional MINC modules, which connect the computer system to a wide variety of analog or digital laboratory instruments. FEP and FRP include FORTRAN-callable software, REAL-11/MNC, that can send information to these modules and receive information from them. The types of MINC modules available include real-time clocks, analog-to-digital converters, digital-to-analog converters, and digital-input and digital-output modules. See the Hardware Section for a complete list of the optional modules.
- The MINC IEEE bus interface, which is part of the MINC hardware for RT-11/FEP and RT-11/FRP users. The MINC IEEE bus interface connects the computer system to instruments that conform to IEEE standard 488-1978. RT-11/FEP and RT-11/FRP include FORTRAN-callable software for the IEEE interface. You can write programs that collect data and status information from one or more IEEE-compatible instruments or send data and control information to those instruments by using the IEEE bus interface and RT-11/FEP and RT-11/FRP software.
- The MINC serial-ASCII controller, which connects the computer system to instruments that use serial transmission of ASCII characters in RS-232 protocol. FEP and FRP include FORTRAN-callable software, REAL-11/MNC, for communication with serial-ASCII instruments. Your programs can collect ASCII character data and status values from instruments that use RS-232 protocol and can send character information to these instruments using the serial-ASCII interface and FEP software.

---

### **Storing Data**

The FEP and FRP software include a PDP-11 operating system — either RT-11 or RSX-11M. The operating system manages all data storage on the disk media supplied with your computer system.

If you have RT-11/FEP or RT-11/FRP, the RT-11 file-management system is available to you through console commands. With these commands, you can create files for storing and editing programs or data. You can also concatenate, copy, edit, and delete files. In addition, the RT-11 operating system provides access to sequential and direct-access data files through FORTRAN READ and WRITE statements. By writing FORTRAN programs that use these statements, you can store data acquired from laboratory equipment or transmit data from RT-11 files to laboratory equipment.

If you have RSX-11M/FEP or RSX-11M/FRP, you have available the additional power of the FORTRAN 77 programming language and FILES-11, the RSX-11M file-management system. FORTRAN 77 enables you to create indexed data files. The FILES-11 file system offers enhanced protection against data loss through the use of multi-user protection and multiple file versions.

### **Processing Data**

FEP and FRP include software that perform the following types of processing:

- Real-time signal processing
- Statistical analysis
- Mathematical analysis

### **Real-Time Signal Processing**

The term “real-time” is widely used but has various meanings. However, as applied to the processing of signals (analog or digital voltages), the meaning of “real-time” is specific. A computer system for applications involving real-time signal processing must be capable of

---

transferring data and sensing and responding to outside events in a very short time. The system must also be capable of handling events from several sources at the same time, even though each source may be sending stimuli or data in rapid bursts. Both FEP and FRP include the software component REAL-11/MNC for use in real-time signal processing applications. RT-11/FEP and RT-11/FRP also include the IBS software, which can perform real-time data transfers.

Consider the following example. Suppose you want to write a program to correlate simultaneously transmitted electrocardiogram and blood-pressure signals. Suppose furthermore that your computer system will sample each signal at regular intervals over two separate channels, such that any value transmitted over one channel corresponds precisely to a value transmitted over the other channel at the same moment. Only a real-time computer system, such as a MINC, can perform this processing.

### **Statistical Analysis**

Frequently, once you have acquired and stored data from laboratory instruments, you must process that data to find relationships between variables or differences between groups of data. You may also want to process manually entered data in a similar way. The FEP and FRP software, SSP, includes FORTRAN subroutines which perform many types of statistical analysis, such as linear regression, analysis of variance, and tests of significance (t-test). FEP and FRP also include non-parametric tests such as chi-square, and an extensive set of matrix manipulation subroutines.

In the blood-pressure example, suppose you wanted to compare the cardiac performance of patients undergoing different treatments. After acquiring and storing the data, the program might evaluate each patient's heart rate and blood pressure and then compare the measurements taken from different groups of patients by using SSP subroutines in a program that performs a statistical test.

---

### **Mathematical Analysis**

In addition to statistical analysis, many problems require the application of sophisticated mathematical operations. FEP and FRP include FORTRAN-callable software, LSP, which performs such operations. These include peak-processing, fast Fourier transforms, and the production of interval histograms.

Referring again to the example, suppose the electrocardiogram signal contained noise. One way to eliminate the noise is using LSP subroutines to apply the Fourier transform to the acquired signal, eliminate the high-frequency components which correspond to the noise, and then apply the inverse Fourier transform to recover the original signal.

### **Synchronizing Processes**

You can use the MINC computer system to control external equipment, and you can also allow the computer system to be controlled by external equipment. By writing programs that enable both to occur, you can completely automate a laboratory experiment.

For example, suppose you write a program that uses REAL-11/MNC routines and controls the temperature of a chemical reaction. This program senses the temperature of the reaction through the optional MNCTP thermocouple preamplifier, which is connected to the MNCAD analog-to-digital converter. When the temperature exceeds or falls below a certain level, the program sends control information over the hardware that transmits data to a heating unit. The data-transmitting hardware is the IEEE interface, the serial ASCII interface, or the digital output module, depending on how the heating unit is connected.

At the same time, the program could receive information from other equipment that senses the state of the reaction. For example, the additional equipment could monitor such conditions as the pH level or the presence of chemical compounds significant to the experiment.

---

## **Interacting with the Computer System**

Your MINC computer system includes at least one hard-copy or video console terminal. Regardless of which type of terminal you have, it is used for interactive communication with the operating system. The operating system uses the terminal for displaying error messages and other helpful information. You use the terminal for entering applications programs and instructions to the operating system.

If you have a VT125 graphics display terminal and the optional RGL/11 graphics software, you can use the VT125's hardware graphics feature to plot data and draw geometric figures on the terminal screen.

## **Programming Tools**

Both of the operating systems available with FEP and FRP include the PDP-11 MACRO assembler. The assembler enables you to write assembly-language code that can be called by FORTRAN programs for applications in which the FORTRAN language itself does not provide needed capabilities.

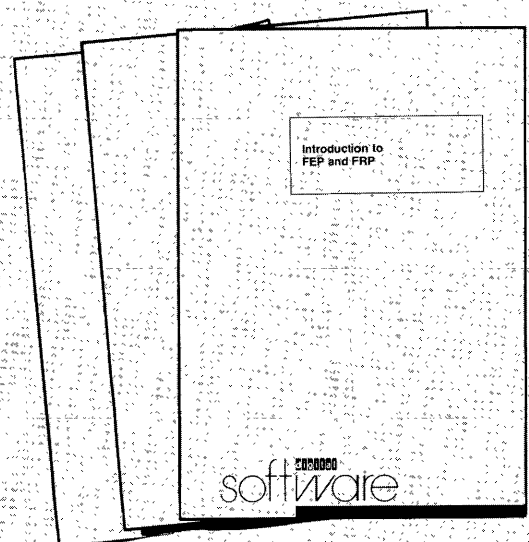
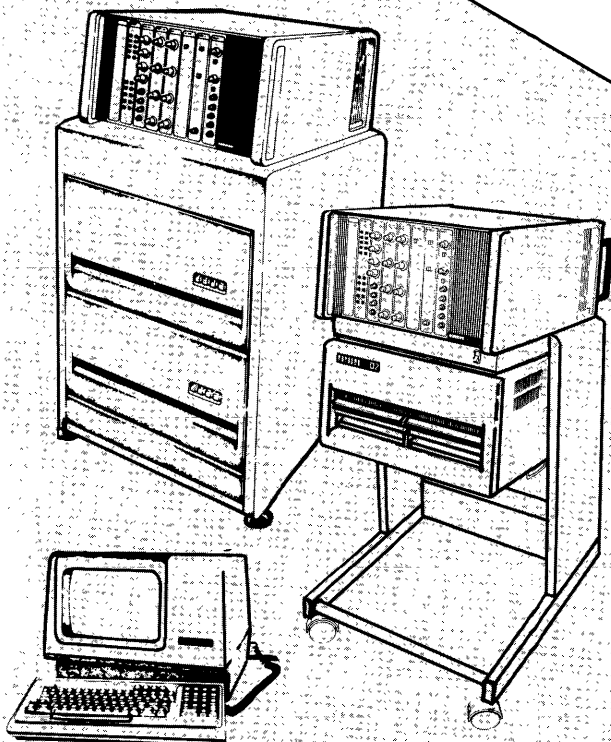
The FEP and FRP software include two debugging tools — the FORTRAN Debugging Technique (FDT, supplied only with RT-11/FEP and RT-11/FRP) and the Online Debugging Tool (ODT). FDT, a sophisticated interactive tool for debugging FORTRAN IV programs, gives you step-by-step control of your program's execution and the ability to examine and change the contents of any variable in your program during execution. ODT, which is included with your operating system, provides help in debugging both MACRO-11 programs and programs containing MACRO-11 routines. Using ODT, you can temporarily halt your program and examine or change the contents of any memory location used by the program.



# Section 2 Software

FEP and FRP Software Components

| Component   | FEP   |         | FRP   |         |
|---|-------|---------|-------|---------|
|   | RT-11 | RSX-11M | RT-11 | RSX-11M |
| Programming Languages<br>FORTRAN IV<br>FORTRAN 77 | •     | •       | •     | •       |
| FORTRAN Debugging<br>FDT                          | •     |         | •     |         |
| Real-Time Data Transfers<br>REAL-11/MNC           | •     | •       | •     | •       |
| Instrument Bus Data Transfers<br>IBS              | •     |         | •     |         |
| Signal Data Processing<br>LSP                     | •     | •       | •     | •       |
| Statistics<br>SSP                                 | •     | •       | •     | •       |
| Graphics<br>RGL/11                                | •     | •       |       |         |





## Section 2 Software

As Table 1 in Section 1 indicates, the FEP and FRP components provide a variety of software tools that fulfill laboratory programming needs. RT-11/FEP and RT-11/FRP include the single-user, real-time operating system RT-11 and the high-level programming language FORTRAN IV. RSX-11M/FEP and RSX-11M/FRP include the real-time, multiprogramming operating system RSX-11M and the high-level programming language FORTRAN 77. The MACRO-11 assembly language is supplied with RT-11/FEP, RT-11/FRP, RSX-11M/FEP and RSX-11M/FRP.

Both FEP and FRP include the REAL-11/MNC data-transfer software, the LSP signal data-processing software, the SSP statistical/mathematical software. Only FEP includes the RGL/11 graphics software. In addition, RT-11/FEP and RT-11/FRP include the IBS instrument-bus software and the FDT debugging software.

### RT-11 Operating System

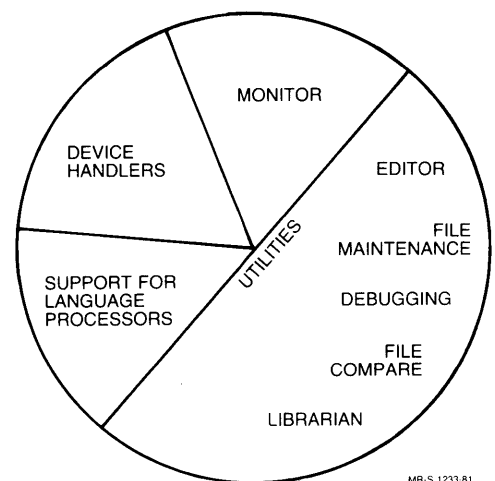
RT-11 is an interactive, single-user, real-time operating system for program development and real-time applications.

#### System Configurations

RT-11 offers three monitors: single job (SJ), foreground/background (FB), and extended memory (XM).

The SJ monitor can place only one program in memory at any given moment. SJ requires less memory than either the FB monitor or the XM monitor.

The FB monitor can simultaneously place two programs in two different parts of memory — the foreground, and the background. The FB monitor gives the foreground job priority over the background job. Therefore, if you use the FB monitor, you can execute real-time functions in the foreground while using the background for program development.



MR-S 1233-81

*Operating System Components*

---

The XM monitor is an elaboration of the FB monitor that supports systems with an 11/23 processor, an MMU (Memory Management Unit), and extended memory; that is, systems with more than 64K bytes of memory. The XM monitor can execute programs that store data and instructions in extended memory. Programs running under the XM monitor can load overlays into extended memory; this results in fast program execution.

### **System Features**

The RT-11 operating system features:

- **Ease-of-use.** The RT-11 keyboard commands are easy to use and understand.
- **Automatic command processing.** You can store a sequence of commands in an indirect-command file and then repeatedly execute that file.
- **Contiguous file structures.** The RT-11 file structure for random-access devices ensures rapid file access and minimal data-transfer times.

- **Flexible real-time input/output.** RT-11 includes two types of input/output operation:

Synchronous input/output. RT-11 suspends program processing until the input/output event is completed.

Asynchronous input/output. RT-11 starts an input/output event, and your program continues processing while the input/output event takes place.

- **Multi-terminal and remote-terminal support.** These features enable you to perform I/O operations that involve more than one terminal.
- **System-job feature.** This feature allows you to run a maximum of eight simultaneously active jobs. It is available only with the FB and XM monitors.
- **SYSLIB.** You can use SYSLIB, the RT-11 system subroutine library, to access system services from your FORTRAN programs. These services include all features of the RT-11 monitor. You can also add other

---

software, such as the FORTRAN Object Time System and REAL-11/MNC, to SYSLIB. Doing so makes linking your FORTRAN programs easier since the RT-11 linker automatically searches SYSLIB each time you link a FORTRAN program.

- **HELP.** The HELP feature provides easily accessed information about keyboard commands.

### **RT-11 System Utility Programs**

The RT-11 utility programs help you perform a variety of functions. The utilities include:

- **EDIT**, an editor for creating and modifying text files. EDIT includes character- and line-oriented commands.
- **KED** and **K52**, keypad editors for use with video terminals. Each of these easy-to-use editors enables you to modify text files by positioning a visible cursor anywhere in the file.
- **PIP**, the RT-11 peripheral interchange program. PIP enables you to transfer files between devices in your system.
- **DUP**, the RT-11 device utility program. DUP performs such utility functions as initializing devices, scanning for bad blocks, and collecting free space on devices.
- **DIR**, the RT-11 directory program. DIR displays the file directory of any file-structured device.
- **RESORC**, which displays information about monitor status and the system configuration.
- **LIBR**, the RT-11 librarian utility. LIBR creates library files of commonly used object modules.
- **ODT**, **VDT**, **PATCH**, **SIPP**, **PAT**, and **SLP**, the debugging and patching utilities, which you can use to locate, diagnose, and correct programming errors.
- Other utilities, which enable you to print the contents of a file in various formats, locate changes you have made to a file, and format disks and diskettes.

---

## **RSX-11M Operating System**

RSX-11M is an interactive, real-time, multiprogramming operating system that enables you to develop and execute multiple real-time tasks (program images). Using RSX-11M enables you to compile and debug new software while previously developed applications are being executed in real-time.

### **System Features**

The RSX-11M operating system features:

- **Multiprogramming.** RSX-11M can execute two or more tasks that reside in memory. With no user action, the operating system logically divides memory into named partitions. Each task operates in a specified partition, and program development or execution in a partition occurs independent of other system activity.
- **Event-driven priority scheduling.** RSX-11M assigns each task a priority; the scheduling of a task's execution depends on its priority.
- **Disk-based operation.** The operating system uses disks as an extension of main memory.
- **Task checkpointing.** A task that you specify as checkpointable can be interrupted and swapped from its memory partition to disk when a higher priority task requests the memory partition.
- **Dynamic memory allocation.** RSX-11M keeps a list of available memory partitions and tasks ready to execute. The operating system brings tasks into memory in order of priority until no tasks are waiting for execution or until no memory is available.

---

### **RSX-11M Utility Programs**

RSX-11M includes utility programs that help you perform a variety of functions. The utilities include:

- The EDI, EDT, and KED editors, which are used to enter and modify source and data files.
- TKB Task Builder, which creates executable tasks, links object modules, specifies tasks' attributes, creates sharable common data, and constructs task overlays.
- LBR Librarian, which creates and maintains disk-resident libraries of object modules and user-defined macros.
- PIP, which transfers files between devices on your system.
- ODT, the on-line debugger.

---

## **FORTRAN Language**

FORTRAN is the standard scientific programming language. A FORTRAN program consists of statements as well as optional comments that document the program.

The language includes two types of statements: nonexecutable and executable. Nonexecutable statements define data characteristics such as variable types, data values, and array sizes. Executable statements initiate actions such as input/output data transfers, arithmetic assignments and computations, branching, and looping.

RT-11/FEP and RT-11/FRP include FORTRAN IV/RT-11, and RSX-11M/FEP and RSX-11M/FRP include FORTRAN 77.

### **FORTRAN IV/RT-11**

FORTRAN IV/RT-11 is a high-level programming language based on the American National Standard Institute (ANSI) FORTRAN, X3.9-1966 standard. FORTRAN IV/RT-11 operates under the RT-11 Operating System and, in addition to features included in standard FORTRAN, it also provides:

- Mixed-mode arithmetic
- The BYTE data type
- ENCODE and DECODE statements
- PRINT, TYPE, and ACCEPT input/output statements
- Direct-access unformatted input/output
- List-directed input/output
- Programmer's comments at the end of each source program line
- PROGRAM statements
- OPEN and CLOSE file-access control statements

FORTRAN IV/RT-11 supports memory-resident virtual arrays on systems that include the PDP-11/23 processor with at least 64KB of memory.



The FORTRAN IV/RT-11 compiler translates a source program to a machine-language program. When you compile a program, you can specify options that reduce the size of the compiled program or increase its execution speed.

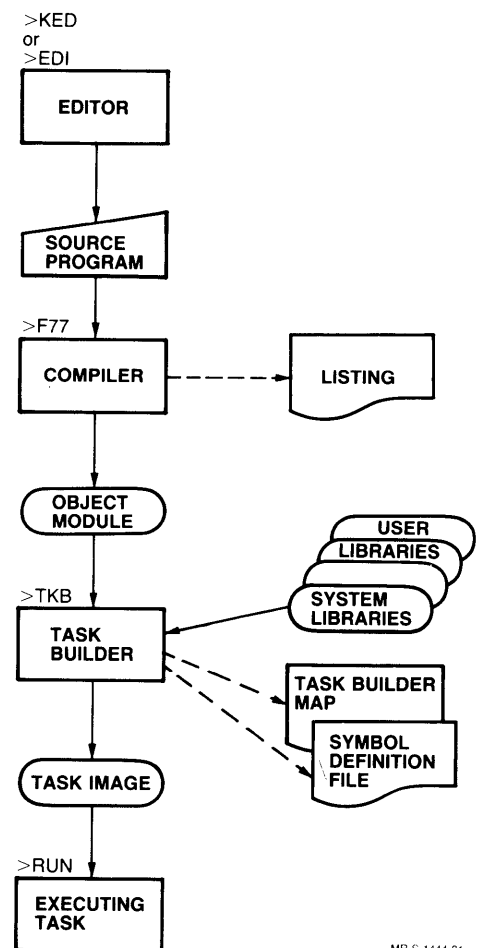
You can access SYSLIB, the RT-11 FORTRAN system subroutine library, from FORTRAN IV/RT-11 programs.

### **FORTRAN 77**

FORTRAN 77 is a high-level programming language that conforms to the ANSI-77 FORTRAN language subset standard. FORTRAN 77 operates under the RSX-11M operating system. Some features of FORTRAN 77 are:

- The CHARACTER data type
- Block-structured IF statements
- The SAVE statement
- Zero-trip DO loops
- Internal Files
- The BYTE data type
- PRINT, TYPE, and ACCEPT input/output statements
- Direct-access unformatted input/output
- Direct-access formatted input/output
- Programmer's comments at the end of each source program line
- PROGRAM statements
- The INTEGER\*4 (32-bit) data type
- The INCLUDE statement
- The PARAMETER statement

The FORTRAN 77 compiler translates a source program to an object file. As the compiler translates the source program, it makes the compiled program as fast-executing as possible.



MR-S-1444-81

*Development of a FORTRAN 77 Program*

---

## MACRO-11

The MACRO-11 assembly language uses the PDP-11 instruction set to add, compare, increment, complement, and perform other operations. Each line in a MACRO-11 source program contains a single assembly-language statement followed by a terminator such as a carriage return.

A MACRO-11 statement can contain up to four fields. The format of a statement is:

[label:] operator operand(s)[; comments]

The label field is optional and identifies a statement so that you can refer to the statement from other parts of the program.

The operator field contains one of the following:

- a PDP-11 instruction mnemonic
- a Macro call
- an assembler directive

When the operator is a PDP-11 instruction mnemonic, the assembler specifies both the instruction to be generated and the action to be performed on any operands which follow. When the operator is a macro call, the assembler inserts the appropriate code during assembly. When the operator is an assembler directive, the assembler specifies a certain function or action to be performed during assembly.

The operand is manipulated by the operator and can be an expression, number, or argument.

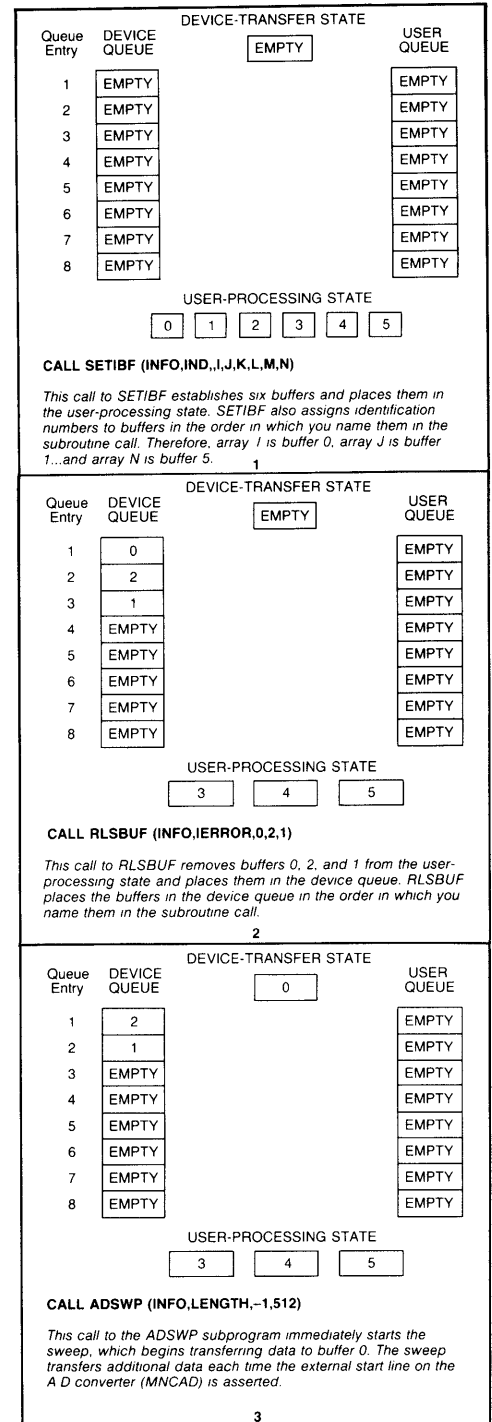
The comment is optional and is a string of ASCII characters. Comments are useful in source listings for documentation and debugging.

## REAL-11/MNC

The REAL-11/MNC software consists of subprograms you can call from FORTRAN IV or FORTRAN 77 programs to communicate with laboratory instruments. FORTRAN programs can call REAL-11/MNC subprograms to perform analog-to-digital data transfers, digital-to-analog data transfers, digital input and output data transfers, buffer management, bit operations, real-time clock operations, serial input/output, and utility functions. Table 2 lists the REAL-11/MNC subprograms.

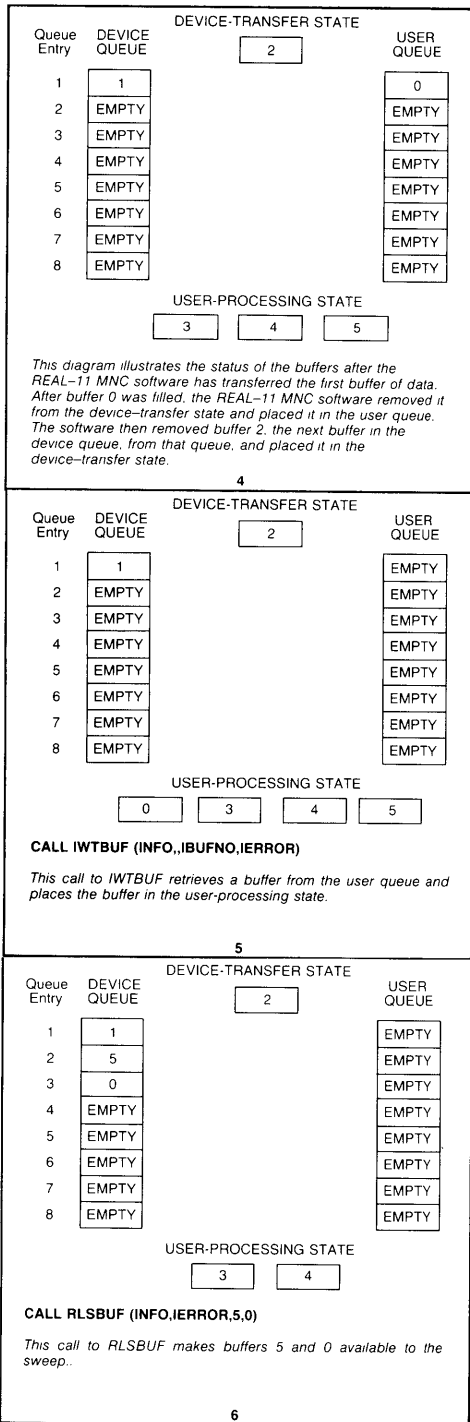
**Table 2. REAL-11/MNC Subprograms**

| Type                                    | Subprogram Function   |
|---|---|
| <i>Analog-to-Digital Data Transfers</i> |   |
| ADFAST                                  | Initiates a high-speed A/D sampling sweep through the A/D converter and through any dual multiplexers and preamplifier connected to the A/D converter.    |
| ADINP                                   | Returns one word as input from the A/D converter and through any dual multiplexers and preamplifier connected to the A/D converter.                       |
| ADSUM                                   | Initiates a high-speed, cumulative A/D sampling sweep through the A/D converter and any dual multiplexer and preamplifier connected to the A/D converter. |
| ADSWP                                   | Initiates an A/D sampling sweep through the A/D converter and through any dual multiplexers and preamplifier connected to the A/D converter.              |
| MADSTS                                  | Reads the status bits of an A/D channel.  |
| <i>Digital-to-Analog Data Transfers</i> |   |
| AOUT                                    | Transfers a single digital value to a D/A-converter channel for conversion to an analog signal.   |
| DASWP                                   | Initiates output sweeps of digital values for conversion to analog signals.   |



MR-S-943-81

Real-Time Processing



**Table 2. (Cont.)**

*Type Subprogram Function*

*Digital-Input Data Transfers*

- DIGO** Selects the digital-input module input bits that define a sweep-start event.
- DINP** Accepts a single 16-bit word from a digital-input module.
- DISWP** Initiates a digital-input sampling sweep through a digital-input module.
- MDISET** Sets the operating mode and the bits of the stimulus-bit register of a digital-input module.

*Digital-Output Data Transfers*

- DOSWP** Initiates a digital-output sweep through a digital-output module.
- DOUT** Transfers a single 16-bit word through a digital-output module.

*Buffer Management*

- IBFSTS** Returns status of buffers being used in a sweep.
- IGTBUF** Returns the number of the next buffer available from the user queue and dequeues the buffer. IGTBUF does not block execution of the calling program.
- INXTBF** Enables you to specify the buffer in the device queue that the Interrupt Service Routines will use.
- IWTBUF** Returns the number of the next buffer available from the user queue and dequeues the buffer. IWTBUF blocks execution of the calling program.
- RLSBUF** Releases buffers to the device queue for use in a sweep.
- RMVBUF** Removes a buffer from the device queue.
- SETIBF** Initializes a sweep-information array for each sweep and assigns numbers to the buffers used in the sweep.

*Real-Time Clock Operations*

- CLOCKA** Sets the rate of the primary real-time clock.
- CLOCKB** Gives control of the secondary real-time clock and the software clock.

---

**Table 2. (Cont.)**

| <i>Type</i>                  | <i>Subprogram Function</i>  |
|------------------------------|---|
| GTHIST                       | Starts a sampling sweep to measure the elapsed time between specified external events and generates a time-interval histogram.  |
| ICLOKB                       | Returns the contents of the 16-bit software clock as an unsigned integer value.   |
| RCLOKB                       | Returns the contents of the 16-bit software clock as a floating-point number.   |
| XRATE                        | Computes clock-rate and clock-preset values for a specified dwell.  |
| <i>Serial I/O Operations</i> |   |
| CIN                          | Initiates the receipt of serial ASCII messages transmitted by a device to the computer.   |
| COUT                         | Initiates the transfer of serial ASCII messages from the computer to a device.  |
| <i>Utility Functions</i>     |   |
| CAD2FP                       | Converts A/D input values to floating-point numbers.  |
| FLT16                        | Converts unsigned 16-bit binary integers to positive floating-point numbers.  |
| GETADR                       | Returns the addresses of subprograms that you can use as completion routines.   |
| ITEMP                        | Converts A/D data into temperature values.  |
| KAD2DI                       | Converts A/D input values to normalized double-precision integers.  |
| KBCD2B                       | Converts 4-digit unsigned Binary Coded Decimal numbers to positive 16-bit binary integers.                                      |
| KB2BCD                       | Converts positive 16-bit binary integers to 4-digit unsigned Binary Coded Decimal numbers.                                      |
| SCANBT                       | Tests all bits from right to left in a specified word. Clears the first bit encountered and indicates the position of that bit. |
| SETBT                        | Sets or clears a bit in a specified 16-bit word.  |
| STPSWP                       | Stops a sampling sweep.   |
| TESTBT                       | Determines if a bit in a specified word is set or clear.  |

## Instrument Bus Subroutines

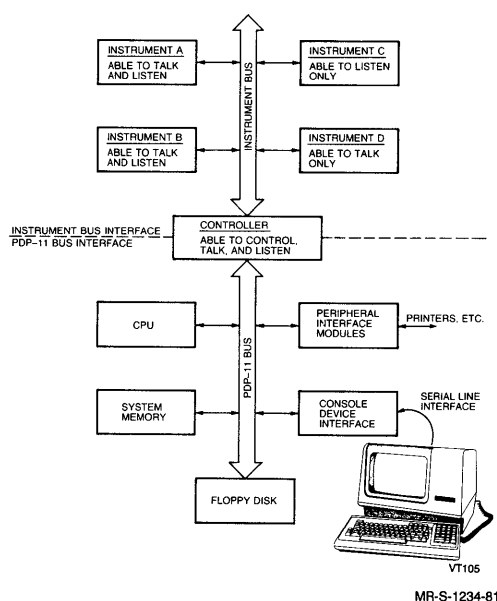
The FORTRAN Instrument Bus Subroutines (IBS) consist of software you can use to transfer messages between the computer and instruments on the IEEE Standard 488-1978 General Purpose Instrument Bus.

The subroutines enable you to:

- Send and receive messages between the computer and instruments.
- Transfer messages between instruments.
- Conduct serial and parallel polls.
- Handle service requests from instruments.
- Perform control functions.
- Perform error checking.

You can include the routines only in FORTRAN IV programs running under RT-11. RSX-11M/FEP and RSX-11M/FRP do not include the Instrument Bus Subroutines. Table 3 lists the Instrument Bus Routines.

**Table 3. Instrument Bus Routines**



| Type                     | Routine Function   |
|--------------------------|--|
| <i>Data Transmission</i> |  |
| IBARCV                   | Receives a message string asynchronously from selected instruments.                |
| IBASND                   | Sends a message string asynchronously to selected instruments.                     |
| IBAXFR                   | Transfers a message string asynchronously between instruments.                     |
| IBFREE                   | Transfers messages from a talker to listeners with no monitoring by the IB driver. |
| IBRCVS                   | Returns the termination status of a receive or transfer operation.                 |
| IBRECV                   | Receives messages from a specified talker.   |
| IBSEND                   | Sends a message from the computer to specified listeners.                          |

### Instrument Bus Configuration

---

**Table 3. (Cont.)**

| <i>Type</i>              | <i>Routine Function</i>   |
|--------------------------|---|
| <i>Data Transmission</i> |   |
| IBSEOI                   | Sends a message from the computer to specified listeners and asserts the end or identify line of the IEEE bus during transmission of the last character in the message. |
| IBTERM                   | Defines data-message terminators.   |
| IBWAIT                   | Waits for asynchronous transmission to conclude.  |
| IBXFER                   | Transfers a message from a talker to a listener with monitoring by the IB driver.   |
| <i>Serial Polling</i>    |   |
| IBSPL                    | Conducts a serial poll.   |
| IBSTS                    | Checks the status of a single instrument.   |
| <i>Parallel Polling</i>  |   |
| IBPPD                    | Disables the parallel-poll response of selected instruments previously enabled for a parallel poll.   |
| IBPPE                    | Enables selected instruments to respond to parallel polls.  |
| IBPPL                    | Conducts a parallel poll.   |
| IBPPU                    | Disables the parallel-poll response of all instruments previously enabled for a parallel poll.  |
| <i>Service Requests</i>  |   |
| IBDEV                    | Supplies the IB driver with a list of all instruments that can generate service requests on the IBV11-A/IB11 unit currently being handled.                              |
| IBSRQ                    | Enables the IB driver to handle service requests.   |
| IBSRQF                   | Determines if any instrument is requesting service.   |
| <i>Control</i>           |   |
| IBCMD                    | Executes programmed commands.   |
| IBDCL                    | Clears the instrument-dependent part of each bus instrument.  |

---

**Table 3. (Cont.)**

| <i>Type</i>           | <i>Routine Function</i>   |
|-----------------------|---|
| <i>Control</i>        |   |
| IBGET                 | Triggers the operation of selected listeners.                                 |
| IBGTL                 | Puts selected instruments in the local state.                                 |
| IBIFC                 | Clears all interfaces.  |
| IBLLO                 | Disables return-to-local buttons of all instruments on the bus.               |
| IBLNR                 | Checks for valid listeners.   |
| IBRDA                 | Puts all bus instruments in the local state.                                  |
| IBREMO                | Determines if remote control is on or off and turns remote control on or off. |
| IBREN                 | Enables all bus-instruments to be put in the remote state.                    |
| IBSDC                 | Clears the instrument-dependent part of selected instruments.                 |
| IBTIMO                | Specifies the timeout value.  |
| IBUNIT                | Selects the IBV11-A/IB11 unit number.   |
| IBUNL                 | Removes listeners from the IEEE bus.  |
| IBUNT                 | Removes talkers from the IEEE bus.  |
| <i>Error Handling</i> |   |
| IBERRF                | Reads the error flag.   |
| IBSTER                | Specifies error-handling characteristics.                                     |



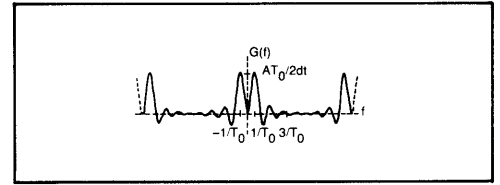
## Laboratory Subroutines Package

The Laboratory Subroutines Package (LSP) is a set of eight subroutines that analyze laboratory data.

FORTTRAN IV and FORTTRAN 77 programs can call the subroutines, which are written in MACRO-11 assembly language. You do not need to know MACRO-11 to use the subroutines. Table 4 lists the LSP routines.

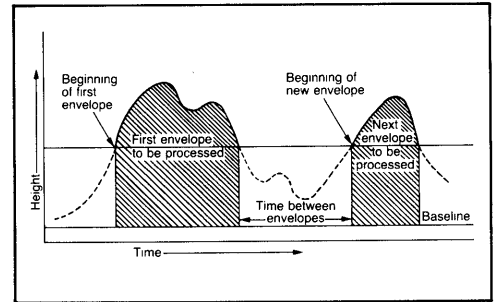
**Table 4. Laboratory Subroutine Package Routines**

| <i>Routine</i> | <i>Function</i>  |
|----------------|--|
| CORREL         | Computes the correlation function.   |
| FFT            | Approximates the analytical or continuous Fourier transform.   |
| HISTI          | Counts the number of observations that fall into predefined categories; in other words, HISTI generates a histogram. |
| NVELOP         | Detects peaks in discontinuous segments of waveform data.  |
| PEAK           | Detects peaks in waveform data.  |
| PHAMPL         | Converts complex numerical values to phase angles and amplitudes.  |
| POWRSP         | Determines the power spectrum for a set of Fourier coefficients.   |
| RHISTI         | Counts the number of observations that fall into numerical intervals.  |



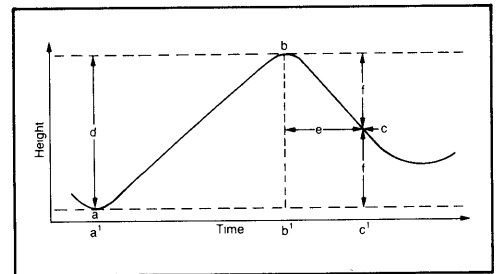
MR-S-1238-81

*Output of FFT*



MR-S-636-81

*Envelope Processing*



MR-S-625-81

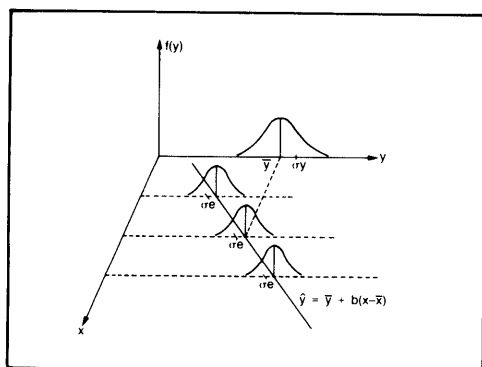
*Peak Processing*

---

## Scientific Subroutines Package

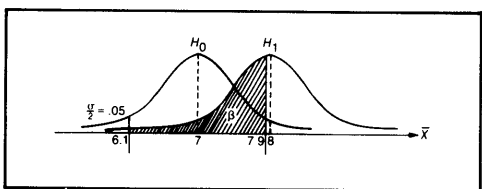
The Scientific Subroutines Package (SSP) contains over 100 statistical and mathematical subroutines that FORTRAN IV and FORTRAN 77 programs can call. All subroutines are written in fully commented FORTRAN and use single-precision values, but you can modify a subroutine to include double-precision arithmetic. Table 5 lists the SSP routines.

**Table 5. Scientific Subroutines Package Routines**



*Regression Analysis*

MR-S-1236-81



*Hypothesis Testing*

MR-S-1235-81

### *Statistical Routines*

- Data screening
- Elementary statistics
- Correlation
- Multiple linear regression
- Polynomial regression
- Canonical correlation
- Analysis of variance
- Discriminant analysis
- Factor analysis
- Time series
- Nonparametric tests
- Random number generator

### *Mathematical Routines*

- Matrix operations
- Integration and differentiation
- Fourier analysis
- Mathematical functions
- Linear equations
- Nonlinear equations
- Roots of a polynomial
- Polynomial operations

## RGL/11

RGL/11 is a library of subroutines for picture drawing and data plotting on the VT125 video terminal. The VT125 displays four shades of gray; for color displays, you can connect a color monitor to the VT125. The RGL/11 routines are callable from FORTRAN IV and FORTRAN 77 programs. It is not available in the FRP package.

---

The picture drawing routines enable you to:

- Draw – circular arcs  
boxes  
circles  
lines  
connected lines  
polygons
- Display – graphics text  
markers (point, cross, diamond, etc.)  
data points  
error messages
- Specify – the terminal's color attributes shading  
text height and width  
viewable limits of data values

The data-plotting routines enable you to create and display two-dimensional graphs. You can specify the characteristics of the graph paper by defining the:

- Types of X and Y axes (linear or logarithmic).
- Number of tick marks on the X and Y axes.
- Color of grid lines.
- Labels (alphanumeric) on the X and Y axes.
- Label (numeric) on the tick marks.

After specifying the graph-paper characteristics, you can plot data. RGL/11 includes two data-plotting types: static and dynamic.

For static plots, the X-axis and Y-axis coordinates of each point to be plotted are stored. The subroutine PDATA displays all data points.

The PPOINT subroutine adds data points one by one to the graph.

For both static and dynamic plots you can specify the marker and color of the data points.

You can also display the X and Y coordinates of selected data points by moving the graphics cursor to the data point.

---

## **FORTRAN Debugging Technique**

The FORTRAN Debugging Technique (FDT) is an interactive tool useful for debugging FORTRAN IV/RT-11 programs. FDT is not available for RSX-11M/FEP or RSX-11M/FRP.

FDT includes three types of commands:

- Program control
- Information transfer
- FDT control

Program-control commands control the execution of a FORTRAN program. Information-transfer commands display and modify the value of a variable or array element. FDT-control commands control sequences of FDT commands. Table 6 lists the FDT routines.

**Table 6. FORTRAN Debugging Technique Commands**

| <i>Type</i>     | <i>Command</i> | <i>Function</i>  |
|-----------------|----------------|--|
| Program Control | CONTINUE       | Resumes program execution after any FDT pause.                               |
|                 | PAUSE          | Defines an FDT pause.  |
|                 | RESET          | Removes an FDT pause.  |
|                 | START          | Starts execution of the program.   |
|                 | STEP           | Executes the program in step mode.   |
|                 | STOP           | Stops the debugging session.   |
|                 | WATCH          | Executes an FDT pause when the value of a variable or array element changes. |

---

**Table 6. (Cont.)**

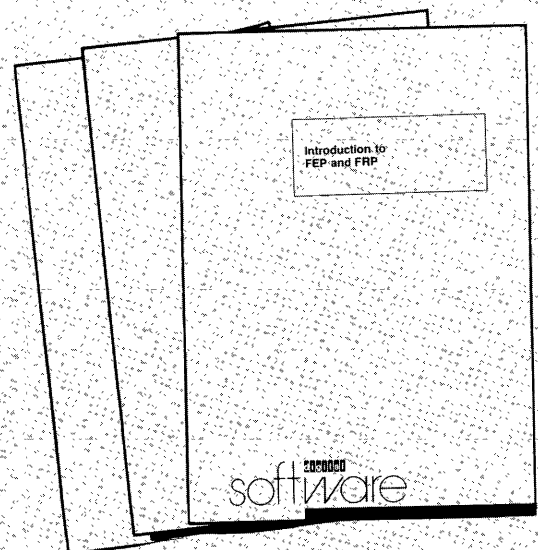
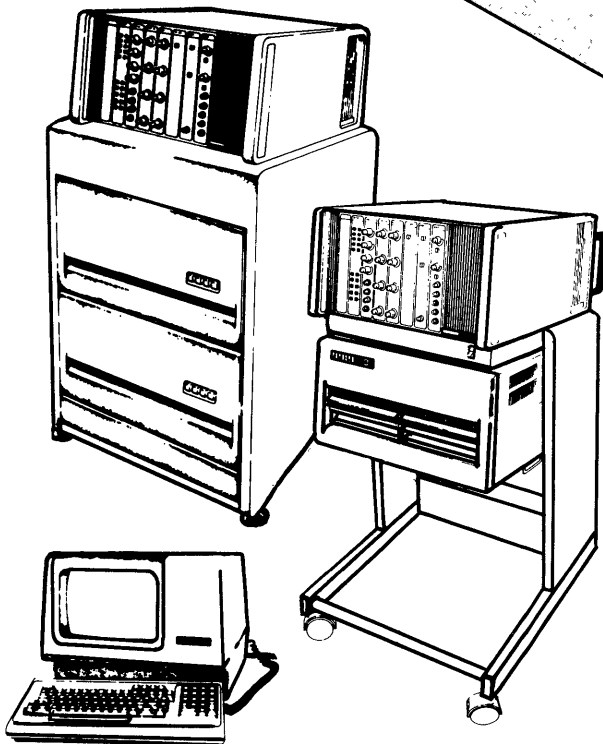
| <i>Type</i>          | <i>Command</i> | <i>Function</i>   |
|----------------------|----------------|---|
| Information Transfer | ACCEPT         | Assigns values to variables.  |
|                      | DIMENSION      | Associates a name with a FORTRAN array.                               |
|                      | ERASE          | Cancels name and location associations.                               |
|                      | NAME           | Associates a name with a location.                                    |
|                      | TYPE           | Displays values of variables and/or text on the terminal.             |
| FDT Control          | GOTO           | Executes an unconditional branch in a sequence of FDT commands.       |
|                      | IF             | Executes an FDT command if the expression in an IF statement is true. |
|                      | MACRO          | Defines, executes, or deletes a sequence of FDT commands.             |
|                      | WHAT           | Displays the current status of FDT.                                   |



# Section 3 Hardware

FEP and FRP Software Components

| Component                               | FEP   |         | FRP   |         |
|---|-------|---------|-------|---------|
|   | RT-11 | RSX-11M | RT-11 | RSX-11M |
| Programming Languages<br>FORTRAN IV     | •     |         | •     |         |
| FORTRAN 77                              |       | •       |       | •       |
| FORTAN Debugging<br>FDT                 | •     |         | •     |         |
| Real-Time Data Transfers<br>REAL-11/MNC | •     | •       | •     | •       |
| Instrument Bus Data Transfers<br>IBS    | •     |         | •     |         |
| Signal Data Processing<br>LSP           | •     | •       | •     | •       |
| Statistics<br>SSP                       | •     | •       | •     | •       |
| Graphics<br>RGL/11                      | •     | •       |       |         |







You can execute the FEP software on the MINC-11 family of hardware, which includes:

- The PDP-11/03 and PDP-11/23 central-processing units
- RX02 floppy disks and RL01 and RL02 hard disks
- The VT105 and VT125 video terminals as well as the LA36 and LA38 printing terminals
- The LA120 and LA34 printers
- The IEEE-488 instrument bus interface
- Serial-ASCII transfer controllers
- The MNC-series laboratory modules

RT-11/FEP and RT-11/FRP can run on any of the MINC-11 processors and disks. RSX-11M/FEP and RSX-11M/FRP require a MINC/DECLAB23.

### Central Processing Units

#### PDP-11/03

The PDP-11/03 (LSI-11/2) includes 60K bytes of fast dynamic MOS memory, eight general-purpose registers, fast interrupt response, and the EIS/FIS performance-enhancement chip. The PDP-11/03 incorporates the full PDP-11 instruction set of over 400 commands.

The Extended Instruction Set (EIS) provides hardware integer multiply and divide, and direct multiple shifting.

The Floating-Point Instruction Set (FIS) offers significant speed improvements over software floating-point routines because it enables the processor to operate directly on single-precision 32-bit data types using floating-point arithmetic, including floating add, subtract, multiply, and divide. The EIS/FIS functions are performed automatically and are transparent to the user. The MNCFA option kit allows you to upgrade a MINC-11 system to a MINC-23.

---

### **PDP-11/23**

The MINC/DECLAB23 is based on the PDP-11/23 processor (LSI-11/23) and incorporates the full PDP-11 instruction set of over 400 commands. This PDP-11/23 includes the FPU floating-point processor, which allows floating-point operations to be executed five to ten times faster than equivalent software operations.

The PDP-11/23 in the MINC/DECLAB23 system has 64K bytes of fast dynamic MOS memory. Memory can be expanded in 64K byte increments to a total of 256K bytes.

The processor includes true memory management. The basis of this capability is the Memory Management Unit (MMU) which provides memory relocation, segmentation, and protection. These features allow more efficient use of memory, simplify handling large memory applications, and provide protection for both user and system programs.

### **System Disks**

#### **RX02 Dual Floppy Disk**

The RX02 dual-floppy-disk drive provides the system with a million characters of random-access memory. The dual-drive system allows you to copy information that is particularly valuable onto a second disk for safekeeping.

#### **RL01**

The RL01 cartridge disk offers you much faster access time than is available with RX02s as well as greatly increased storage capacity. An RL01 can store 5.2 megabytes of data.

#### **RL02**

The RL02 cartridge disk is similar to the RL01 but its storage capacity is double the capacity of the RL01.

---

## **Terminals**

### **VT105**

This interactive video terminal provides both graphing and alphanumeric capabilities. The VT105 can display 24 lines of 80 upper- and lower- case characters. This terminal also can display 24 lines of 132 characters to reproduce data that might normally be output on a 132-column line printer.

A split-screen scrolling feature allows the CRT to operate as two separate displays. One portion of the screen can be used for alphanumeric text while the other is being used for graphic displays.

The VT105 features smooth scrolling, which allows information to move smoothly upward for the length of the full screen, or for a portion of the screen when the split-screen feature is being used.

Important information can be underlined or displayed in boldface, reverse-video, or flashing characters. Combinations of these modes are also possible. The  $7 \times 9$  dot matrix characters can be set to double width for easier viewing.

The keyboard is separate from the CRT so that it can be positioned where it is easiest to use. It provides a full 128-character ASCII set in standard typewriter format, a separate numeric keypad in calculator format, and a special-function EDIT keypad that makes editing programs simple.

### **VT125**

The VT125 is a graphics display terminal that includes bit map architecture and the alphanumeric capability of the VT105 terminal. The VT125 executes ReGIS (Remote Graphics Instruction Set) commands, which allow you to create and store pictures with ASCII text commands. You can connect a VT125 to an LA34-VA graphics printer to make hard copies of VT125 graphics displays. Additional higher level language support is provided by RGL/11 in FEP-11.

---

### **LA36**

The LA36 DECwriter II is a 96-character-ASCII, upper- and lower-case printer that provides such capabilities as fast operation and 30-character-per-second printing. The LA36 accepts 1- to 6-part forms and standard 132-column line printer paper. It has a wide variety of options and accessories.

### **LA38**

The LA38 DECwriter IV is a low-cost, desk-top, microprocessor-driven printing terminal with a printing speed of up to 30 characters per second. Its operating features include horizontal tabs, horizontal margins, and a choice of four character sizes and six line spacings. In addition, it contains a numeric keypad and offers tractor feed.

## **Printers**

### **LA120 DECwriter III**

This is a pedestal-mounted hard-copy terminal with a printing speed of 180 characters per second. It features an LED display, horizontal and vertical tabbing, left/right/top/bottom margins, an EIA interface, bidirectional smart printing, eight font sizes, six line spacings, a numeric keypad, font expansion and compression capability, non-volatile forms memory, and a 1K character buffer.

### **LA34-VA**

The LA34-VA is a low-cost, desk-top, receive-only, hard-copy terminal that includes two print modes — one for printing text characters and the other for printing graphics displays from a video terminal.

In graphics mode, the LA34-VA processes data at 30 characters per second. The graphics mode is available only with FEP. In text mode, it prints at 320 columns or 960 dots per second. This terminal can copy any figure displayed on a video terminal.

---

The LA34-VA features selectable baud rates ranging from 110 to 9600, adjustable character widths, adjustable line spacing. It can print the full 128-character-ASCII upper/lower case set in a standard nine-by-seven dot-matrix font.

## **Real-Time I/O**

### **IEEE Instrument Bus Interface**

Up to 14 laboratory instruments and testing devices can be supported with the IEEE-488 instrument bus interface. The bus is an asynchronous, 8-bit, bidirectional data path that transfers data between instruments and computer memory.

Any instrument connected to the bus can function as a "talker," or a "listener," or both. As a talker, an instrument can send data or commands to any other instrument on the bus. As a listener, an instrument can receive commands or data from any other instrument on the bus.

### **Serial Line Units**

The MINC computer system includes one serial-ASCII transfer controller. This device contains three channels that are routed to the rear of the MINC chassis and are identified as serial line units or SLUs. A fourth channel, also routed to the rear of the chassis, is identified by the term CONSOLE and is dedicated to the system terminal. These SLUs support serial transfers of 8-bit ASCII code between the system and the terminal, a printer, and two other devices. The serial-ASCII transfer controller does not support the use of modem control lines; therefore, the only type of modem you can use with an SLU is an acoustic coupler.

SLU2 is often assigned to an optional 300-baud printer such as the LA36. SLU0 AND SLU1 can be used with a variety of monitor and control instruments that can accept and send serial ASCII data. The MINC computer system can accommodate additional serial ASCII transfer controllers.

---

### **MNC-Series Modules**

Since most laboratory devices cannot be connected to an IEEE bus interface, DIGITAL has developed eight specialized input/output modules for the MINC-11 family of hardware. There are slots in the MINC chassis where up to eight of these optional modules can be plugged in.

**Table 7. MNC-Series Modules**

| <i>Type</i> | <i>Function</i>             |
|-------------|-----------------------------|
| MNCAA       | Digital-to-Analog Converter |
| MNCAD       | Analog-to-Digital Converter |
| MNCAG       | Preamplifier                |
| MNCAM       | Analog Multiplexer          |
| MNCDI       | Digital Input               |
| MNCDO       | Digital Output              |
| MNCKW       | Programmable Clock          |
| MNCTP       | Thermocouple Preamplifier   |

Each module is protected in its own metal enclosure. Each one can be removed from the MINC chassis and stored on a shelf or table without damaging the electronics. In a large laboratory where there are several MINCs, the modules can be stored in a central location and used by lab personnel as needed.

A second feature of the modules is the interlocks provided with them. These interlocks ensure that the modules are always installed correctly. For example, since analog signals can be passed from a multiplexer module to the analog-to-digital converter, the interlocks ensure that the modules can be installed only one way — the right way — to permit interaction.

Third, each module is designed to accept its own 24-line connector block. These connectors are labeled and keyed so that the user cannot plug a connector into the wrong module.

The connector can be left attached to the instrument so that when it is time to use the instrument again, the connector can be plugged right into the MINC module. There is no need to constantly attach, remove, and reattach lines to screw terminals.

In addition, the top of the MINC chassis is hinged to facilitate plugging the connectors into the top of the appropriate modules.

### Programmable Clock Module

In addition to the 60 Hz line-frequency clock that is included with the MINC computer system, a programmable crystal clock (MNCKW) is also available. The MNCKW has three major functions:

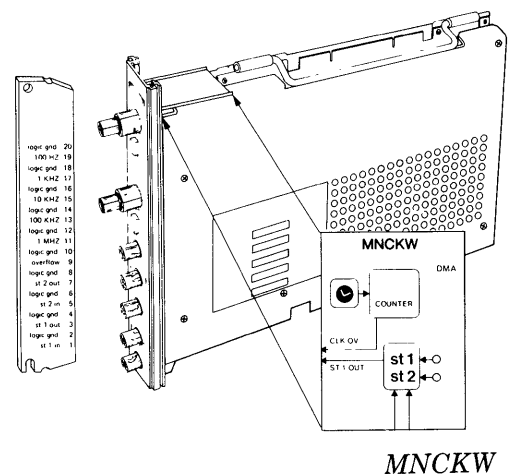
1. To control analog or digital transfer rates; for example, the sweep sampling rate in an EEG data-acquisition experiment.
2. To create (in conjunction with a digital-output module) precisely timed intervals or events, such as stimulus presentations in psychology experiments.
3. To measure intervals and events; for example, measuring poststimulus intervals in a neurophysiology experiment.

Two programmable clocks can be used: a primary clock for controlling one of the three types of operations just described and a secondary clock for keeping a high-resolution time base.

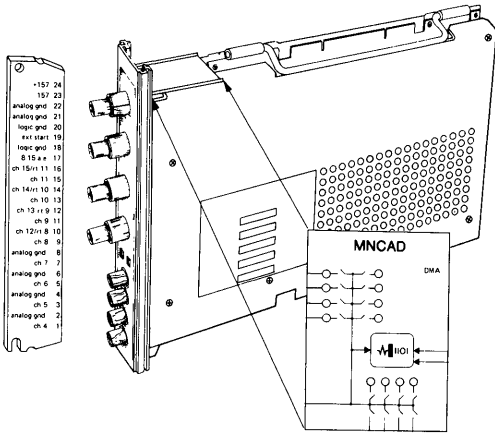
On the front panel of the clock there are two easy-to-use BNC input connectors for the Schmitt triggers. Input signals may range between  $-30$  and  $+30$  volts.

One Schmitt trigger input is used either as an external frequency input (used in place of the crystal clock or line-frequency rate) or as a sense line to recognize "stimulus" events when measuring poststimulus times (PSTs).

The second Schmitt trigger can be used to start the clock upon an external event, or as a sense line for measuring PSTs, or as the event being measured when acquiring data to create time-interval histograms (TIHs).



MNCKW



MNCAD

### Analog-to-Digital Converter Module

The A/D module (MNCAD) will support as many as 16 single-ended channels, or eight single-ended and four quasidifferential channels (useful where the signal cable must be run over long distances or travel through electrically noisy environments). If additional lines are needed you can plug in one or more analog multiplexer modules.

The analog-to-digital converter features input protection from excessive external voltages, and has an input range of  $-5.12$  volts to  $+5.12$  volts, with a resolution of one part in 4096, enabling it to detect differences as small as 2.5 millivolts.

A sample-and-hold feature of the preamplifier module (MNCAG) keeps the measured voltage constant while the analog-to-digital conversion is taking place. This feature makes it possible to use the A/D module with rapidly changing input voltages.

The A/D module also features a direct connection to the primary programmable clock module so that input sampling during a sweep will occur at very precise, clock-controlled intervals.

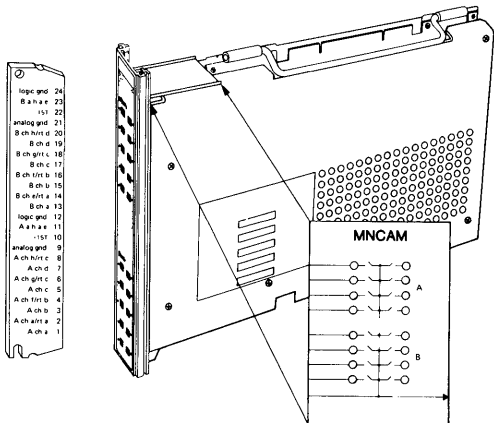
Four switches on the front panel allow input on each of the first four A/D channels to come from one of the four front panel BNC connectors, one of the four front-panel potentiometers, or one of four internal test signals.

### Analog Multiplexer Module

The analog multiplexer module (MNCAM) is designed so that expansion occurs in two groups of channels, each of which may be selected independently to provide eight single-ended channels, or four quasidifferential channels.

Thus, a single multiplexer module will increase the capacity of the A/D converter to 32 single-ended channels, 24 single-ended and four quasidifferential channels, 16 single-ended and eight quasidifferential channels, or eight single-ended and 12 quasidifferential channels.

Three multiplexers will provide the user with a total of 64 single-ended channels. And as many as seven multiplexers can be used with a single A/D module to give a total of 56 quasidifferential and eight single-ended channels.



MNCAM

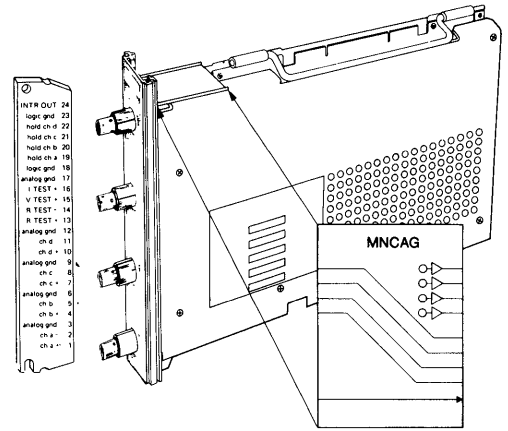


### Preamplifier Module

The preamplifier module (MNCAG) provides four true differential input channels. High common-mode rejection minimizes electrical noise interference with low-level signals, such as those found with gas or liquid chromatographs.

The four front-panel switches select the type of input signal to be measured on each channel. Since these switches may be set to measure voltage, resistance, or current, the preamplifier module may be interfaced to almost any type of transducer or sensing device. Amplification can be under program control, or the user can select fixed gains of 0.5, 5, 50, or 500 by using the controls on the front panel. The amplification of each channel can be controlled individually. Each of the four channels has a sample-and-hold circuit, which may be triggered by an external TTL signal.

By using several preamplifier modules with synchronized sample-and-hold circuits, signals from a multitrack analog tape recorder may be digitized without any time distortion (skew) of the input signals across all input channels.



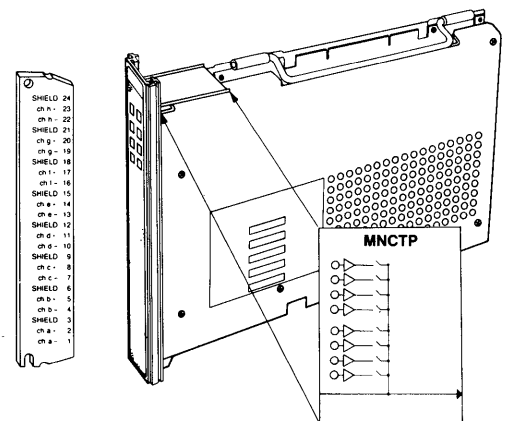
MNCAG

### Thermocouple Preamplifier Module

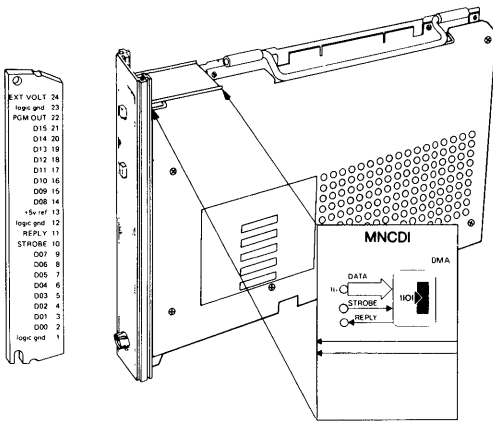
The thermocouple preamplifier module (MNCTP) provides differential-mode amplification of thermocouple signals. The MNC chassis can accommodate a maximum of seven thermocouple preamplifiers, each of which contains eight channels and an eight-channel internal multiplexer.

The thermocouple preamplifier is compatible with the following National Bureau of Standards thermocouple types: J, K, E, T, S, R, and B. The REAL-11/MNC software, in conjunction with the thermocouple preamplifier, automatically sets the gain to utilize the full range of the A/D converter.

The front panel of the thermocouple preamplifier contains eight monitor lights. When a light is lit, the REAL-11/MNC software is accessing the channel with which the light is associated.



MNCTP



MNCDI

### Digital-Input Module

The digital-input module (MNCDI) performs three different functions. One function is to allow a computer program to sample all 16 input lines under clock control at regular intervals. This type of input is useful for measuring the duration of from one to 16 separate events.

A second mode of operation allows an external device to send a signal to the module's "strobe" input. When the module receives this signal, the current state of all 16 input lines is stored internally; this information is then passed to the program. When the program receives the data, a handshaking signal is sent back to the external device to tell it to send the next item of data. This mode is particularly useful for instrumentation that sends readings to the MINC in the form of multiple BCD-encoded digits.

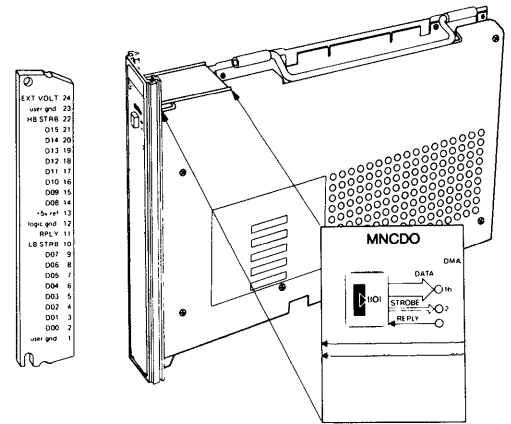
In the third mode, the program sends the digital-input module information telling it to which of the 16 input lines it should be sensitive. It may be sensitive to one or more lines at a particular time. When the module senses an input event on a "sensitive" line, it stores the current state of all 16 lines and causes the program to accept the data. In this way the digital-input module can be used to sense external events from limit switches in mechanical apparatus, or responses made by human or animal subjects in psychology experiments.

### Digital-Output Module

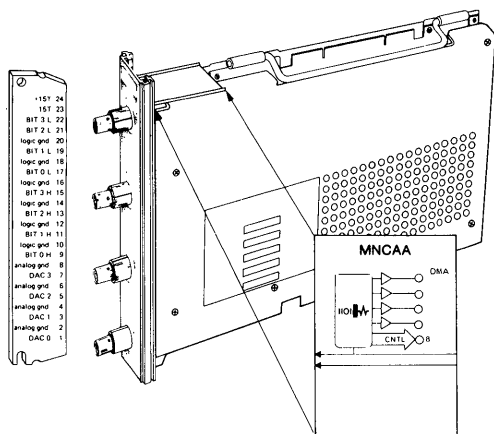
The digital-output module (MNCDO) outputs digital signals on 16 separate lines. Normally, outputs are TTL-compatible, with a ground level representing a logically true condition and a +5 voltage representing a logically false condition. If, however, an external-supply voltage between 5 and 30 volts is applied to the module, then a logically false condition can be represented by this voltage. Because a device connected to an output line may present a load of up to 140 mA, lamps and relays can be directly controlled by the module.

The digital-output module is used to perform three principal functions:

1. It allows a computer program to turn on or off individual electronic circuits such as indicator or warning lamps.
2. If several output signals are utilized in parallel, apparatus such as digitally controllable signal attenuators can be controlled by the MINC. Also, in this mode, a series of 16-bit words can be transmitted to an external device at regularly spaced intervals under clock control. This is particularly useful in controlling a device such as a speech synthesizer.
3. The computer system can transmit a series of 16-bit words at a rate determined by an external device. Each time the computer system transmits a word the system also transmits a data "ready" pulse. The external device can use this pulse to recognize the presence of transmitted data and then transmit a data-accepted signal to the computer system. By sending data-accepted signals to the computer system, the external device controls the rate at which information is delivered.



MNCDO



MNCAA

### Digital-to-Analog Converter Module

The digital-to-analog converter module (MNCAA) supports four separate analog output channels which can be used to control X/Y plotters, CRTs, and laboratory apparatus such as voltage-controlled oscillators.

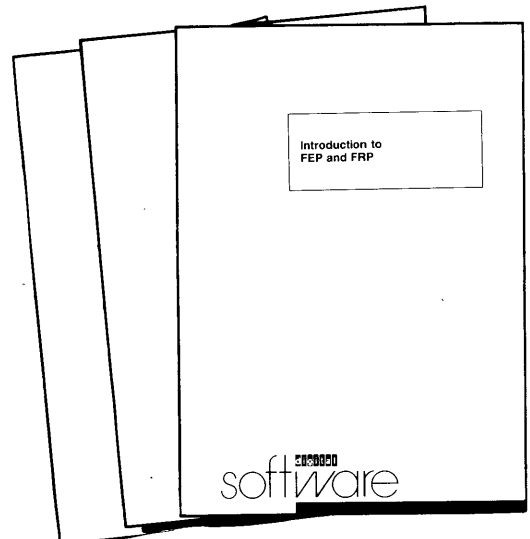
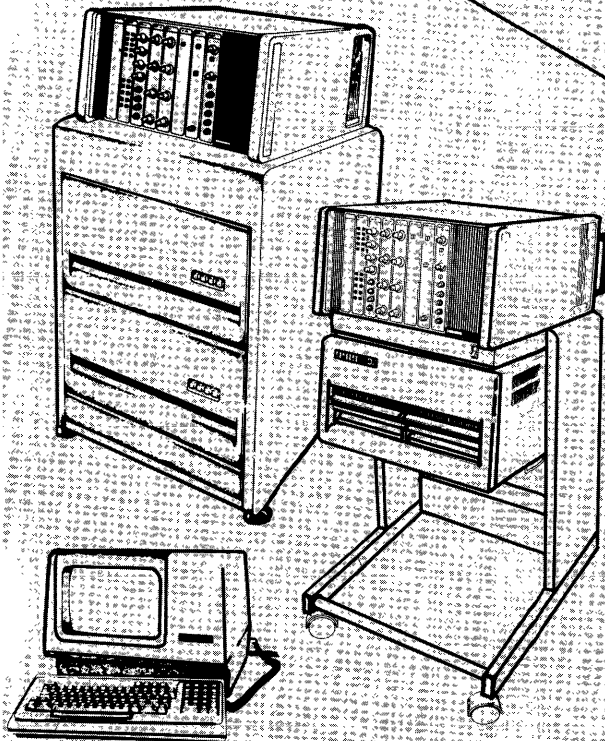
Each channel can be switch-selected for one of three bipolar voltage ranges ( $\pm 2.26$  volts,  $\pm 5.12$  volts, and  $\pm 10.24$  volts) or one of two unipolar ranges (0 to 5.12 volts and 0 to 10.24 volts). Each analog output channel has a resolution of one part in 4096, allowing voltage changes as small as 1.25 millivolts to be generated.

The D/A module also features four TTL-compatible output signals that can be used to control the up/down position of a pen on an X/Y plotter, the beam-intensity circuit on a refresh CRT, or the erase circuit on a storage-tube CRT. Each TTL signal is output on two separate lines. Because one line represents logically true signals as ground level voltage, the digital-to-analog module can control a device regardless of its signal-polarity requirements.

# Section 4 Documentation

FEP and FRP Software Components

| Component                               | FEP   |         | FRP   |         |
|---|-------|---------|-------|---------|
|   | RT-11 | RSX-11M | RT-11 | RSX-11M |
| Programming Languages<br>FORTRAN IV     | •     |         | •     |         |
| FORTRAN 77                              |       | •       |       | •       |
| FORTRAN Debugging<br>FDT                | •     |         | •     |         |
| Real-Time Data Transfers<br>REAL-11/MNC | •     | •       | •     | •       |
| Instrument Bus Data Transfers<br>IBS    | •     |         | •     |         |
| Signal Data Processing<br>LSP           | •     | •       | •     | •       |
| Statistics<br>SSP                       | •     | •       | •     | •       |
| Graphics<br>RGL/11                      | •     | •       |       |         |





---

## Section 4 Documentation

FEP and FRP documentation consist of manuals which are common to both RT-11 and RSX-11M and manuals which are unique to each operating system.

### **RT-11/FEP, RT-11/FRP, RSX-11M/FEP, and RSX-11M/FRP Manuals**

The following manuals are common to all packages.

#### **Introduction to FEP and FRP**

This document introduces the following topics:

- FEP and FRP capabilities
- FEP and FRP software
- Hardware required to run FEP and FRP
- FEP and FRP documentation

#### **REAL-11/MNC FORTRAN Programmer's Reference Manual**

This reference manual includes the following topics:

- Overview of MNC series modules
- REAL-11/MNC concepts and capabilities
- REAL-11/MNC subprograms
- Sample FORTRAN programs

#### **Scientific Subroutines Programmer's Reference Manual**

This manual describes the statistical and mathematical subroutine library. It also includes the installation and verification procedures.

#### **Laboratory Subroutines Programmer's Reference Manual**

This manual describes each of the eight laboratory subroutines.

---

## **RT-11/FEP and RT-11/FRP Manuals**

The following manuals are unique to RT-11/FEP and RT-11/FRP.

### **RT-11/FEP and RT-11/FRP Installation and User's Guide**

This manual describes the installation procedures for RT-11, FORTRAN IV, and RT-11/FEP. It also discusses program development techniques and system precautions for REAL-11/MNC.

### **Instrument Bus Subroutines Programmer's Reference Manual**

This document describes IEEE bus operations, system installation and operation procedures, and instrument bus subroutine programming techniques.

### **FORTRAN Debugging Technique Reference Manual**

This manual describes the FDT commands and the FDT installation procedures.

### **Release Notes for RT-11/FEP and RT-11/FRP FORTRAN Enhancement Package for MINC, Version 2.0**

The Release Notes describe the components of the Distribution Kit, list known problems and restrictions, and explain how to report software or documentation problems.

### **RT-11 Documentation Set**

These manuals document RT-11.

### **PDP-11 FORTRAN IV Documentation Set**

These manuals document FORTRAN IV.



---

## **RSX-11M/FEP and RSX-11M/FRP Manuals**

The following manuals are unique to RSX-11M/FEP and RSX-11M/FRP.

### **NOTE**

The manuals assume the system is set to accept MCR rather than DCL commands.

### **RSX-11M/FEP and RSX-11M/FRP Installation and User's Guide**

This manual describes the installation procedures for FEP. It also discusses program-development techniques and system precautions for REAL-11/MNC.

### **Release Notes for RSX-11M/FEP and RSX-11M/FRP FORTRAN Enhancement Package for MINC, Version 1.0**

The Release Notes describe the components of the Distribution Kit, list known problems and restrictions, and explain how to report software or documentation problems.

### **RSX-11M Documentation Set**

These manuals document RSX-11M

### **PDP-11 FORTRAN 77 Documentation Set**

These manuals document FORTRAN 77.

### **RT-11/FEP and RSX-11M/FEP Graphics Manual**

### **RGL/11 Programmer's Reference Manual**

This manual is supplied with RT-11/FEP and RSX-11M/FEP but not included in FRP. The manual describes how to use the RGL/11 software for plotting data and drawing figures.



Analog multiplexer, 40  
Analog-to-digital converter, 7, 40

Central processing units, 33

Data plotting, 26  
Digital input module, 42  
Digital output module, 43  
Digital-to-analog converter, 44  
Documentation, 47

EIS, 33  
Extended Instruction Set, 33

FDT, 8, 28  
FEP components, 3  
File management, 5  
FILES-11, 5  
FIS, 33  
Floating-Point Instruction Set, 33  
FORTRAN Debugging Technique, 8, 28  
FORTRAN Enhancement Package (FEP), 3  
FORTRAN language, 16  
FORTRAN IV/RT-11, 11, 16  
FORTRAN Real-Time Package (FRP), 3  
FORTRAN 77, 5, 16, 17  
Fourier transform, 7  
FRP components, 3

IEEE bus interface, 4, 37  
Instrument bus subroutines, 22

Laboratory Subroutine Package, 25  
LA34-VA, 36  
LA36 DECwriter II, 36  
LA38 DECwriter IV, 36  
LA120 DECwriter III, 36  
LSI-11/2, 33  
LSI-11/23, 34  
LSP, 7, 25

MACRO-11, 8, 18  
Mathematical analysis, 7  
MINC BASIC, 3  
MINC/DECLAB23, 33  
MINC-11, 33  
MINC-23, 33

---

MNCAA digital-to-analog converter, 44  
MNCAD analog-to-digital converter, 7, 40  
MNCAG preamplifier, 41  
MNCAM analog multiplexer, 40  
MNCDI digital input, 42  
MNCDO digital output, 43  
MNCFA option kit, 33  
MNCKW programmable clock, 39  
MNCTP thermocouple preamplifier, 7, 41  
MNC-series modules, 38

ODT, 8  
Online Debugging Tool, 8

PDP-11/03, 33  
PDP-11/23, 34  
Picture drawing, 26  
Preamplifier, 41  
Printers, 36  
Processing data, 5  
Programmable clock, 39

Real-time I/O, 37  
Real-time signal processing, 5  
REAL-11/MNC, 4, 19  
ReGIS, 35  
RGL/11, 3, 8, 26  
RL01 cartridge disk, 34  
RL02 cartridge disk, 34  
RS-232 protocol, 4  
RSX-11M operating system, 14  
RSX-11M/FEP, 3  
RSX-11M/FRP, 3  
RT-11 operating system, 11  
RT-11/FEP, 3  
RT-11/FRP, 3  
RX02 dual floppy disk, 34

Scientific Subroutines Package, 26  
Serial-ASCII controller, 4, 37  
SSP, 6, 26  
Statistical analysis, 6  
Storing data, 5  
Synchronizing processes, 7  
System disks, 34

---

VT105, 35  
VT125, 8, 26, 35

Terminals, 35  
Thermocouple preamplifier, 7, 41  
Transferring data, 4



**READER'S COMMENTS**

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

---

---

---

---

---

---

---

---

---

---

Did you find errors in this manual? If so, specify the error and the page number.

---

---

---

---

---

---

---

---

---

---

Please indicate the type of reader that you most nearly represent.

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Other (please specify) \_\_\_\_\_

Name \_\_\_\_\_ Date \_\_\_\_\_

Organization \_\_\_\_\_ Telephone \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_  
or Country

----- Do Not Tear - Fold Here and Tape -----

**digital**



No Postage  
Necessary  
if Mailed in the  
United States



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

**SOFTWARE PUBLICATIONS**  
200 FOREST STREET MR01-2/L12  
MARLBOROUGH, MA 01752

----- Do Not Tear - Fold Here and Tape -----