digital

hardware

VT78 programmer's reference manual

EK-OVT78-RM-001

VT78 programmer's reference manual

EK-OVT78-RM-001

digital equipment corporation • merrimack, new hampshire

1st Edition, July 1980

The material in this manual is for informational purposes and is subject to change without notice.

The software described in this document is furnished under a license and may not be used or copied except in accordance with the terms of such license.

Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

Printed in U.S.A.

This document was typeset on DIGITAL's DECset-8000 computerized typesetting system.

CONTENTS

CONTENTS (Cont)

FIGURES

TABLES

TABLES (Cont)

PREFACE

The *VT78 Programmer's Reference Manual* is designed to familiarize programmers with the principles involved in programming the VT78 Video Data Processor. It provides the necessary information for programming the VT78 for use with its associated peripherals.

This manual does not attempt to teach fundamental programming or computer concepts. It is a reference document for VT78 programming information only.

The Introduction section describes the physical characteristics of the VT78 Video Data Processor and its WT/78 Word Terminal version. Major functional components of the VT78 are described in the System Description section. Descriptions of each software package available for use with the VT78 are provided in the Software Overview section.

The Programming section is divided into functional groups and provides instructions for using VT78 microinstructions and Input/Output Transfers (IOTs) to control data flow between the VT78 processor and display screen, keyboard, and peripheral devices. Data transfer techniques, register descriptions, command word formats, function codes, and timing considerations are included.

The following associated documents can be referenced for detailed information on the various elements of the VT78.

Document	**Document Number**
General Reference and User's Documents	
DECstation 78 Pocket Service Guide	EK-VTX78-PS
DECstation Technical Manual	EK-VTX78-TM
DP78 Synchronous Communications Unit Manual	EK-ODP78-HR
PDP-8/A Operator's Handbook	EK-8A001-OP
PDP-8/A Miniprocessor User's Manual	EK-8A002-MM
RX01/RX02 Reference Card	EK-RX102-RC
RX02 Floppy Disk System User's Guide	EK-RX02-UG
RX02 Floppy Disk System Technical Manual	EK-0RX02-TM
RX8/RX11 Floppy Disk System User's Manual	EK-RX01-OP
RX8/RX11 Floppy Disk System Maintenance Manual	EK-RX01-MM
VT78 Family Field Maintenance Print Set	MP00468-00
LQP8/78 Printer System Technical Manual	EK-OLQP8-TM
LA180 DECprinter I User's Manual	EK-LA180-OP

Document	Document Number
Software Documents	
OS/78 User's Manual	AA-5748B-TA
OS/78 Command Summary	AV-5582A-TA
MACREL/LINK User's Manual	AA-5664B-TA
OS/78 System Release Notes	AA-D927B-TA
RTS/8 User's Manual	AA-0724D-TA
COS-310 New User's Guide	AA-D758A-TA
COS-310 System Reference Manual	AA-D647A-TA
COS-310 Release Notes & Installation Guide	AA-D759B-TA
COS-310 System Reference Card	AV-D757A-TA
Word Processing Documents	
Word Processing Documentation Directory and Glossary	AV-H105B-TK
WPS-8 Word Processing System Reference Manual	AA-5267C-TA
Word Processing System List Processing User's Manual	AA-5269B-TK
WPS-8/78 V3.3 Release Notes	AA-J508A-TA
WPS-8 Preface	AV-5688C-TC
Word Processing Operator Training Manual	AA-D102B-TK
Word Processing Operator Training Manual Administrator's Guide	AA-D114B-TA
Word Processing System Communications Options Manual	AA-5264C-TK
Getting Started with the 78 Series	AA-J349A-TA
Solving Problems on the 78 Series	AA-J057A-TA
Using WPS	AA-J318A-TK
WPS Reference Manual	AA-J043A-TK
WPS Reference Card	AV-J552A-TK
DIGITAL Computer Store Documents	
DIBS Accounts Payable Operator's Guide	AA-J471A-TA
DIBS Accounts Receivable Operator's Guide	AA-J472A-TA
DIBS Payroll Operator's Guide	AA-J473A-TA
DIBS Invoicing and Inventory Control Operator's Guide	AA-J474A-TA
DIBS General Ledger Operator's Guide	AA-J475A-TA
DDS-408 Owner's Guide	EA-18487-79

INTRODUCTION

The VT78 Video Data Processor (Figure 1) is a fully integrated minicomputer and display system that offers the features of the PDP-8 and DECscope families. This includes a compatibility with a full range of software and peripheral devices. The VT78 also offers a set of problem solving tools in both single-user and multitasking applications.

The VT78 can be easily programmed and configured to perform in a wide range of data processing environments – including personal computing and fixed-function applications to real-time, multitasking operations, and networking.

In the VT78, the computer and display terminal are a single, compact unit designed to provide the maximum processing capabilities at a minimum cost. The processor and keyboard/display are interconnected so that the system can be programmed to function as an independent display unit.

The VT78 is activated by a single START switch located on the right side of the unit. The switch initiates program loading from either the floppy disk subsystem or from an externally mounted MR78 read-only memory. The display and keyboard can be tested independently of the central processing unit (CPU) and are tested automatically at system power-up under CPU control.

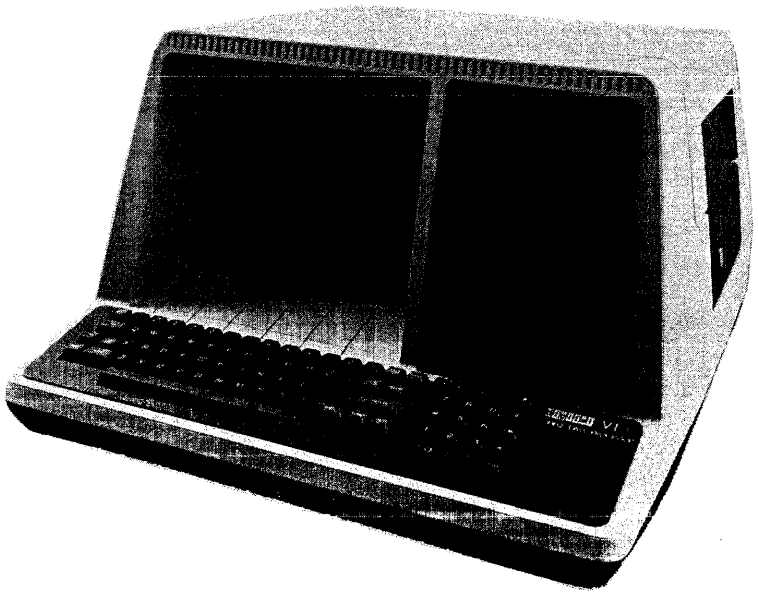


Figure 1 VT78 Video Data Processor

The word processing version of the VT78, the VT78-W, is used with a special word processing key cap set. This version is commercially labeled the WT/78 Word Terminal (See Figure 2).

Figure 2 WT/78 Word Terminal

SYSTEM DESCRIPTION

The VT78 terminal consists of three major functional components – the processor, the keyboard/video display, and the dc power supply. Figure 3 is a physical/functional block diagram of the keyboard/video display and the processor.

The processor, an LSI version of the PDP-8A microprocessor (namely the KK8-B), is a 12-bit, fixed-length word, parallel transfer computer using two's complement arithmetic. The processor contains a central processing unit (CPU), a 16K word main memory, a control memory (commonly refered to as the panel memory), an MR78 program loading device interface circuit, a real–time clock, and four peripheral I/O interfaces.

The system is started by switching on the ac power switch located at the rear of the terminal. Upon completing the power-up sequence, the diagnostics (resident in panel memory) verify that the major system components are operating correctly. If there is a malfunction within the system, the built-in diagnostics issue a halt message and provide internal status information. Two red indicators (DC OK and CPU OK), located at the rear of the terminal, also provide status information.

The control memory (panel memory) has 256 words of read/write random access memory (RAM) for variable storage and 1K words of read-only memory (ROM). The ROM contains the resident diagnostics, an MR78 loader, and an RX78 bootstrap loader.

Program loading is initiated by momentarily pressing the START switch. This causes the panel memory to select the appropriate program loading device and load its program into main memory.

There are two program loading devices available with the VT78. One is the MR78 and the other is the floppy disk drive system. The MR78 has priority over the floppy disk drive system when the START switch is pressed. Each device has its own interface. The MR78 interface is unidirectional, receiving 8-bit parallel data from the MR78 in a "papertape" form. The floppy disk drive system interface is bidirectional (that is, programs may be written or read) and permits system interface with two RX78 dual floppy disk drive systems.

Main memory has 48 N-channel MOS (NMOS) dynamic 4K RAM chips organized as 16K 12-bit words. It has three basic cycles – read/refresh, read/wait, and read/write.

The VT78 has three serial line units (SLUs). The SLU 2 interface permits the processor (using EIA signaling levels) to communicate with external RS-232-C devices at transmission rates ranging from 50 to 19,200 baud. Baud rates, word parity, word length, and stop bit selection are under program control. Framing errors and parity errors also can be detected under program control. SLU 3 can communicate with an EIA device which handles 8-bit data without parity and only one stop bit. Unlike SLU 2, this interface does not have error checking but does provide the same baud rate selection under program control. SLU 1 is similar to SLU 3 with the exception of the EIA signaling levels. It is programmable and is hardwired directly to the keyboard/video display I/O and operates at 9600 baud. The SLU interfaces support only a subset of the pins on the RS-232-C connectors. Refer to the *DECstation Technical Manual* (EK-VTX78-TM-002) for additional information.

The parallel I/O interface is a 12-bit bidirectional data port which responds to the instruction sets of the LA78 or LQP78 printers, or to the DP78 synchronous communication interface. The particular instruction set is determined by the parallel I/O device used and its associated cable.

The keyboard/video display enables the user to communicate with the processor and control all of the system peripheral devices.

Characters to be displayed on the video screen are transferred to the display via SLU 1 in ASCII format with embedded escape sequences for display control.

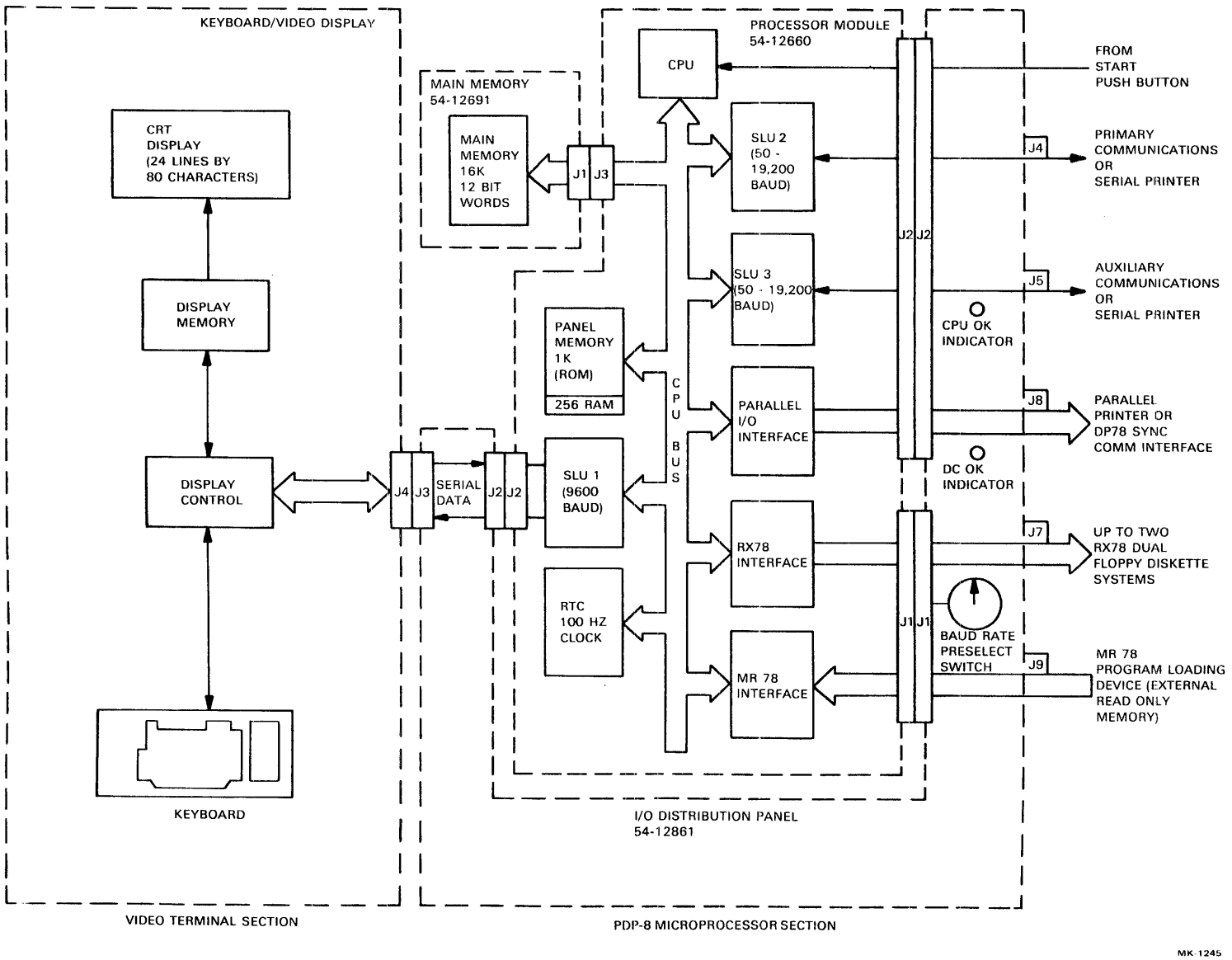


Figure 3 VT78 Physical/Functional Block Diagram

SOFTWARE OVERVIEW

There are several software packages available for use with the VT78 terminal. Each package is designed for a particular application.

OS/78 OPERATING SYSTEM

The OS/78 package provides VT78 users with both interactive and batch programming environments. OS/78 is based upon DIGITAL's proven OS/8. OS/78 maximizes the use of VT78 main memory because the resident portion of the operating system requires only 256 words of memory. Nonresident portions of the system are automatically swapped into memory from the floppy disk (diskette) system.

OS/78 is easy to use and provides the development programmer with a logical interface to program and file structures. All data files and executable programs are stored on one or more floppy disks, where they may be accessed for loading, modifying, or executing by keyboard commands.

OS/78 includes both commercial BASIC and FORTRAN IV and provides a comprehensive set of software tools and utility programs in the single user environment. For additional information refer to the OS/78 documentation listed in the Preface of this manual.

The MACREL/LINKER is provided as optional software. MACREL/LINKER is an assembly language programming system that operates with OS/78. MACREL is a macro assembler that produces relocatable modules. LINKER is a linking loader.

MACREL offers the following capabilities and features:

- Includes macros for program control,
- Allows users to create and call macro definitions,
- Allows users to create programs consisting of program "sections" locatable anywhere in memory,
- Allows users to define groups of codes as macros,
- Permits overlay and level number specification in the assembly module, and
- Allows the use of "secondary" reference symbols.

LINKER offers the following capabilities and features:

- Creates an executable image file,
- Provides a structure for program overlays,
- Links global references among sections,
- Permits the specification of overlay and level number in the assembly module, and
- Allows the use of "secondary" reference symbols.

For additional information, refer to the *MARCREL/LINK User's Manual* (AA-5664B-TA).

COS-310 OPERATING SYSTEM

COS-310 is one of Digital Equipment Corporation's DATASYSTEM 300 Series Commercial Operating Systems. It is an applications development tool for EDP users who wish to implement data management functions for small to medium-size business applications. COS-310 is a self-contained, single user, disk-resident operating

system. It provides an operations control monitor, an easy-to-learn high level programming language (DIBOL), program preparation and debugging utilities, and production utilities.

COS-310 provides software operation control through the system monitor. For memory economy, the monitor resides in two segments – one is memory-resident and the other is on the system device. The monitor includes a comprehensive set of commands that control the editing and execution of programs, and the maintenance of file directories.

Digital Equipment Corporation's Business Oriented Language (DIBOL) is built around procedural verbs that permit the programmer to arrange information for desired execution and output. These procedural statements (commands plus data) permit data manipulation, calculation of arithmetic expressions, subscription, overlaying of records, clearing of memory or buffers, file initialization, branching, tracing, program chaining, and printing overlapped with processing.

For additional information, refer to the COS-310 documentation listed in the Preface of this manual.

RTS/8 OPERATING SYSTEM

RTS/8 is DIGITAL's real-time, multitasking, operating system for use with the VT78. RTS/8 allows a VT78 to handle several tasks simultaneously. This is done by making use of otherwise idle processor cycles that occur periodically. A user-defined priority list allows the most important jobs to be processed first. As a result, programs in execution can be temporarily suspended and removed from memory (swapped out) to make room for a higher priority process.

RTS/8 includes resource management capabilities. It maintains the status of system resources to determine the combination of peripherals, input files, and run-time options used during the execution of a program. For additional information, refer to the *RTS/8 User's Manual* (AA-0724D-TA).

WPS-8/78 WORD PROCESSING SOFTWARE

Description

WPS-8/78 is DIGITAL's text processing system for use on a VT78-W terminal with dual floppy disk drives. A VT78-W is a WT/78 Word Terminal with a word processing keyboard (see Figure 2). Hard-copy output is on a LQP78 letter quality printer, an LA78 draft printer, or an LA38 desk-top printer. WPS-8/78 is a stand-alone program that loads itself from a system diskette whenever the VT78 START switch is pressed.

A menu-driven editor creates and updates documents stored on floppy diskettes. Up to 200 documents of various lengths, or up to a total of 175 pages (500 words each) can be stored on a system with dual diskette drives. (That is, 125 pages on each document diskette and 50 pages on the system diskette.) The editing capabilities allow changes without the need for retyping. Final or draft documents can be queued to a letter quality printer or to a draft printer (if so equipped). Printing and editing can be done concurrently, or the operator can create or edit documents while the system transmits documents to another system.

Media

WPS-8/78 software uses two types of diskettes, system and document.

System Diskette – The system diskette contains all WPS-8/78 programs, prestored document formatting instructions ("rulers"), "menus" and "cue cards" for automatic operator prompting, and prestored printer commands. Additional storage space (about 50 pages of text) is reserved for the "boilerplate library", "shorthand dictionary", and scratch files for transferring documents between the diskettes.

Document Diskette – Document diskettes are used for the storage of the actual text of the documents to be printed or transmitted to another computer over communications lines. Over 125 full pages of text (approximately 500 words per page) can be stored on one diskette.

An index of all documents stored on each diskette, with names assigned by the operator. is automatically recorded on the diskette. The index also indicates the size of each document and automatically notes the date and time the document was created and last modified.

WPS-8/78 document diskettes can be read and written under COS-310 (Commercial Operating System for the DEC Datasystem 310 small business computer). This facility permits the integration of business data processing applications with the word processing power of WPS-8/78.

Printer Forms Control
Printer control markers embedded within each document control a variety of factors – the printing of chapter headings at the top or bottom of every page, the positioning of automatic page numbers, and the inclusion of date and time printed.

Stored Printer Settings
Additional printer options stored with each document determine such variables as spacing, first page to print, last page to print, number of copies, pitch, page size, and whether single sheets or continuous forms are to be used.

System Features

System features include:

- Dynamic floppy diskette file allocation
- Either two or four-floppy system configurations
- Responsive menu-driven operation
- Easy-to-learn commands
- Special editing keypad
- Full editing features

    1. Cut and paste blocks of text
    2. Cursor positioning by character, tabs, word, sentence, line, paragraph, page, and section
    3. Boilerplate insert from library file
    4. Shorthand expressions
    5. Between-the-lines viewing of control characters
    6. Transpose character key
    7. Delete and rubout by sentence, line, word, and character

- Full control of tabs, margins, justification, and pagination

    1. Automatic centering of text on a line
    2. Discretionary pagination control
    3. Semi-automatic hyphenation
    4. Decimal and right-adjusted tabs

- Selectable pitch and type fonts
- Underline and overstrike (bold) printout
- Multicolumn printing
- Superscript and subscript
- Form letter merge
- Time and date stamp
- Operator statistics
- Single sheet or continuous forms printing
- User-definable keys
- Output on letter-quality printer – up to 45 characters-per-second

- Output on draft quality line printer - up to 180 characters per second
- Document storage on removable diskettes - over 125 pages of text per diskette
- "Cue card" prompting of commands - a self-teaching feature
- List processing package for automatic generation of reports and personalized form letters
- Simultaneous editing and printing - as many as eight documents may be queued to the printer
- Optional communications software package - enables the system to communicate with remotely located computers or other word processing stations

For additional information, refer to the word processing documentation listed in the Preface of this manual.

DIBS2 ACCOUNTING PACKAGES

Description
DIBS2 (Digital Integrated Business System Version 2.0) is DIGITAL's collection of five stand-alone business application packages for use on a VT78-W terminal with dual RX02 floppy disk drives. The five DIBS2 packages are:

- Accounts Payable
- Accounts Receivable
- Payroll
- Invoicing and Inventory Control
- General Ledger

The DIBS2 programs are written in DIBOL-8, DIGITAL'S Business Oriented Language, and run under the COS-310 operating system (Version 8.0).

Each DIBS2 package is a stand-alone application program that loads itself from a system diskette whenever the VT78 START switch is pressed. Once loaded, applications are selected from a master menu and all data entry to displays is from the VT78 keyboard. Hard-copy output is on an LQP78 letter quality printer, an LA78 high-speed line printer, or an LA38 desk-top printer.

Accounts Payable
This package supplies an accurate and efficient means of controlling a company's cash position by automating the computation, storage, retrieval, and reporting of the accounts payable system while retaining personal and flexible control. A printed analysis of the current cash position is always available so critical decisions can be made in time to improve profit-and-loss statements.

Features include:

- Posts invoices,
- Reports on cash requirements,
- Prints checks,
- Integrates manually drawn prepayment checks,
- Posts expenses to multiple accounts from single invoices,
- Maintains all accounts payable files automatically,
- Provides the information needed to plan and follow a discount strategy, and
- Gives statements and reports the computer's polished, professional look.

Accounts Payable does these jobs:

1. Enters and posts vouchers

   The operator types the information from invoices into the system and verifies the accuracy on the easily read terminal screen. Credits (for posted and paid vouchers) are entered as negative amounts.

The system distributes the total amount of each invoice to one or more (up to nine) general ledger accounts. After entering all vouchers, the operator can check them on a printout before automatically posting them to Accounts Payable open items.

2. Prints the cash requirements report

   This report shows how much cash will be needed at a specified date to pay vouchers outstanding at any preceding date. A voucher-by-voucher record of the discount amounts that can be taken or will be lost by deferring payment is provided.

3. Prints the aged trial balance

   Either the balance for all acounts or only selected accounts can be printed. The report shows all outstanding posted vouchers and the current amounts payable, those payable in from 31 to 60 days, from 61 to 90 days, and in more than 90 days.

4. Generates payment deferral and precheck-writing report

   The user can select the vouchers to be paid in the next check run, and the system automatically defers any that are not due, as well as any others chosen. The report shows which items will be paid, the discounts that will be taken, and the total amount of cash needed for the checks.

5. Prints computer-drawn checks

   After the system prints the checks, it prepares a check register itemizing all the checks and any prepaid vouchers entered.

6. Prints an expense cross-reference report

   An itemization of all posted invoice amounts and voucher numbers for each general ledger expense account are printed. In addition, the expense account totals which must be posted manually to the general ledger are printed.

7. Prints a vendor analysis report

   This report shows what proportion of the total payables was spent with each vendor, as well as the number of vouchers to each vendor and their average amounts.

8. Makes partial payments

   Each voucher, for which a prepayment check is to be drawn manually and for which the remaining balance will be paid by a partial payment computer-drawn check, can be identified.

Media - Two diskettes are included in the DIBS2 Accounts Payable operating system and application programs - the Accounts Payable Program Diskette, and the Accounts Payable Data File Diskette.

Information Storage Capacity -

Item	Quantity
Vendors	1,037
Open items	1,988
General ledger expense accounts	730
Daily transactions per run	300
Total general ledger distributions a month	3,498

Accounts Receivable
This package simplifies five major problem areas:

1. Recording sales
2. Applying customer payments
3. Printing statements
4. Issuing management reports
5. Training accounts receivable personnel

Reports generated by the package supply all the information needed to analyze accounts and control the income resources of a business more profitably.

Features include:

- Shows income sources,
- Shows cash position,
- Helps in planning future business,
- Improves the accuracy, efficiency, and responsiveness of an accounting system,
- Gives statements and reports the computer's polished, professional look, and
- Gives the information needed to manage the money owed to the business.

Accounts Receivable, accurately and quickly, does the following:

1. Enters and posts sales

   The operator puts each sale, credit, or debit memo into the system through the keyboard and verifies it on the video screen before the system posts it in the Sales Journal.

2. Enters and posts cash receipts

   The operator enters each customer payment, and the system records the payment and the invoice number to which it will be applied before posting it in the Cash Receipts Journal.

3. Prints statements and purges files

   Statements can be printed for all or selected customer accounts. After statements are printed, the purging program matches invoices with their credits and deletes all transactions which balance to zero.

4. Generates reports

   The following reports are generated – Receipts Work List; Detailed and Summary Aged Trial Balances; Delinquent Accounts, Credit Balance and Zero Balance Exception Registers; Monthly Sales and Cash Receipts Summary; Sales Analysis by Customer, Customer Type, State, and Sales Representative; Sales and Cash Receipts Journals; and Finance Charges.

Media – Two diskettes containing the DIBS2 Accounts Receivable Customer Master File and the Accounts Receivable Open Item File are included. A third diskette is supplied for use as a temporary file.

Information Storage Capacity –

Item	Quantity
Customers	520
Open items	4,280
Sales transactions per run	300
Cash receipts per run	300
Month-to-date sales and cash subtotals	190

Payroll
This package simplifies the total payroll function and provides records and reports that are more acceptable and understandable.

The package does the complete job quickly and accurately, saving users time for more rewarding assignments. Its flexibility allows users to tailor outputs to the company's need, while still meeting the requirements of taxing authorities, labor organizations, and a complex fringe benefit program.

Features include:

- Saves hours of clerical time in preparing and recording payroll information,
- Minimizes clerical errors and eliminates math errors,
- Accepts new deductions and other changes without costly reprinting of forms,
- Requires minimal maintenance, and
- Produces checks and reports with the computer's polished, professional look.

Payroll does the following reliably, accurately, and quickly:

1. Payroll calculations

   The operator enters time and pay rates through the keyboard and verifies each entry on the video screen. Pay categories include regular, sick, holiday, vacation, overtime, and one user-defined type. Up to five different pay rates can be used for each employee.

   An attendance edit list can be printed, if the operator prefers, to verify entries on paper rather than on the screen.

   The program calculates the gross and net pay and advanced earned income credit for each employee after the entries are verified. Following the run, the program prints the Attendance Register, Payroll Register, and Payroll Deduction Register. These can be reviewed before checks are printed.

2. Deduction calculations

   The package calculates federal, state, local, and FICA taxes. Insurance, dues, savings, and other deductions are also calculated and withheld at the specified pay periods. Up to 15 deductions can be taken, each with its own frequency code.

3. Employee Master File updates

   Amount-to-date figures are automatically updated by the package for each employee at each pay period.

4. Report generations

   The reports generated are - the Attendance Register; Payroll Register; Deduction Register; Payroll Check Register; Savings Bond Register; Payroll Attendance Distribtion Report; Monthly Union Report, Seniority List; 941 Form; Yearly W-2 Earnings and Tax Statements; and the Quarterly Earnings Report.

Media - Two diskettes containing the DIBS2 Payroll operating system and application programs are provided - the Payroll System Diskette and the Employee Data Diskette.

Information Storage Capacity –

Item	Quantity
Employees	420
Employees in each run	280
Manually drawn checks in each pay period	70

Invoicing and Inventory Control
This package simplifies and speeds up invoicing and inventory management functions while providing the material flow information needed to offer the best service at the most profitable level.

Timely, accurate information about stock levels, customer shipments, and business volume is backed up by detailed and summary sales analysis reports.

The results of sales and pricing strategies are shown in terms of volumes and profit margins.

Features include:

- Gives immediate feedback about reorder or out-of-stock items,
- Helps maintain a profitable balance between customer service levels and inventory costs,
- Improves the accuracy, efficiency, and responsiveness of an accounting system,
- Analyzes sales and prints detailed and summary reports,
- Produces statements and reports with the computer's polished, professional look, and
- Saves thousands of dollars by cutting unneeded inventory and by giving the information needed for shrewd purchasing.

Invoicing and Inventory does the following:

1. Produces invoices

   The operator enters sales information through the keyboard and verifies and edits it on the video screen. The system prints invoices and credit memos and updates the Inventory Master File. In conjunction with the DIBS2 Accounts Receivable package, the invoicing program also updates the DIBS2 Accounts Receivable Customer Master File.

2. Posts invoices

   With each invoice and credit memo it posts, the system posts in the Master Inventory File the month-to-date and year-to-date sales dollars and cost-of-sales dollars for each item in the transaction. It also posts any backorders to the Invoicing Back Order File and prints a sales journal.

3. Records and posts inventory receipts

   The operator records the receipt of inventory stock items and the cost and price for each. Entries appear on the video screen for verification and editing before the system posts them to the Inventory Master File.

4. Generates reports

   The following reports are generated – the Purchase Advisory Report, to track inventory levels and reorder points; Inventory Receiving Journal; Inventory Stock Status Report, for all or selected items; Price List; Back Order Report; and the Picking Tickets.

Media – Three diskettes containing the DIBS2 Invoicing and Inventory Control operating system and application programs are provided – the Inventory Control Master File, the Invoicing Master File, and the Back Order Master File.

NOTE

The Customer Master File and Open Item File from the DIBS2 Accounts Receivable package is also needed, as well as a right-to-use license for the DIBS2 Accounts Receivable package.

Information Storage Capacity –

Item	Quantity
Number of inventory items	1,500
Customers	520
Inventory receipts into inventory	440
Back orders	1,840
Invoice transactions per run (average of eight line items per invoice)	180

General Ledger

This package combines sound accounting principles with automated computation, storage, retrieval, and report generation to create a powerful management tool. This is a powerful but flexible tool, since it lets operators set up balance sheets and profit-and-loss statements according to their own plan.

The package meets the requirements of the most demanding accounting practices, yet it generates simple financial transaction summaries to give clear, concise reports.

Features include:

- Monitors income, expenses, and net results,
- Improves the understanding of the company's current financial position, and
- Gives the answers needed to plan the company's future.

General Ledger provides:

1. A complete chart-of-accounts

   The package includes a complete chart-of-accounts file, which can be modified to suit the company's practices.

2. Easy entry and posting of transactions into the General Ledger

   The operator enters transactions through the keyboard of the computer system, verifies them on its easily read video screen, and edits them at the keyboard or with a printout before the system posts the new transactions in the file.

3. Customized financial reports

   The system provides an Income Statement and Balance Sheet for all or selected profit centers; Changes in Financial Position; Working Trial Balance; and a Cash Flow Statement.

4. Easy year-end closings

Media – Two diskettes are provided, one containing the DIBS2 General Ledger operating system and application programs, the other, the data files.

Information Storage Capacity –

Item	Quantity
Chart-of-accounts	640
Transactions a month	850
Year-to-date entries	9,100

For additional information refer to the DIBS documentation listed in the Preface of this manual.

PROGRAMMING

CENTRAL PROCESSOR UNIT (CPU)

The processor, shown in Figure 4, is comprised of four programmable 12-bit data registers (AC, MQ, PC, and MAR), control logic, timing logic, and an arithmetic logical unit (ALU).

The ALU performs both arithmetic and logical operations such as AND, TAD, and logical OR. ALU operations involve two operands, one held in the accumulator (AC) and the other in a memory location.

The AC is a 12-bit data register used in ALU operations. Data is fetched from memory to the AC or transferred from the AC to memory. The AC can be cleared, complemented, incremented, tested, and rotated under program control.

Link (L) is a 1-bit flip-flop that serves as a high-order extension to the AC (a carry bit).

The program counter (PC) contains the address, in memory, from which the next instruction is fetched. During an instruction fetch, the PC determines the instruction address and is incremented by one. If a branch to another address in memory is requested (i.e., a JMP or JMS instruction), a new address is loaded into the PC. Several operate and IOT instructions will produce a program skip. These instructions cause the PC to be incremented by one a second time, thus skipping the next sequential instruction.

The multiplier quotient (MQ) register is a temporary register used to exchange data, OR data, or store data received from the AC.

A more detailed description of the functional operation of the CPU is provided in the *DECstation Technical Manual.*

MEMORY

There are two memories in the VT78 terminal – main memory and panel memory. Main memory (Figure 5) consists of a volatile 16K random access memory (RAM) divided into 4K blocks (fields) of memory. Each field (0-3) contains a 4K 12-bit RAM memory that is used to store a user's main program and data.

Panel memory (Figure 6) uses both read-only memory (ROM) and RAM memories. Panel memory contains 4K of memory addresses. Each address contains 12 bits of data or an instruction. The internal diagnostics, MR78 bootstrap, and disk bootstrap programs are stored in panel memory at addresses 6000 through 7777 (octal). Because these programs need several addresses in which to store register information, two pages of RAM have been provided at location 0000 to 377 8 of this field of memory. Locations 400 8 through 5777 contain no memory.

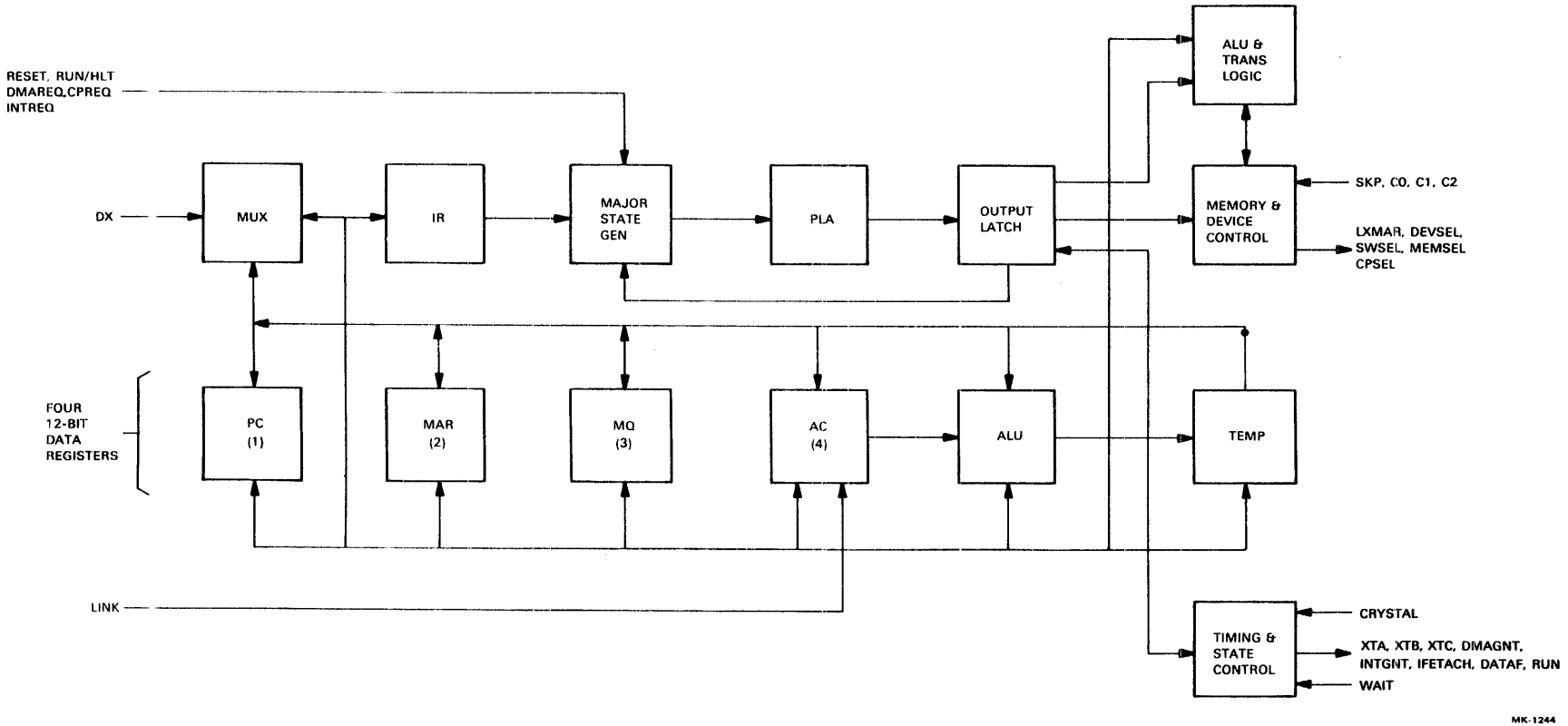


Figure 4 CPU Block Diagram

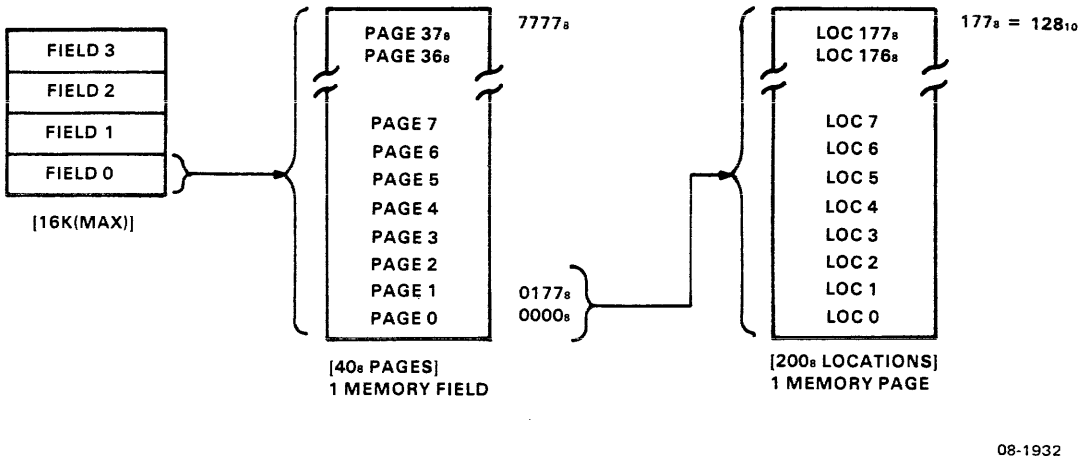


Figure 5 Main Memory

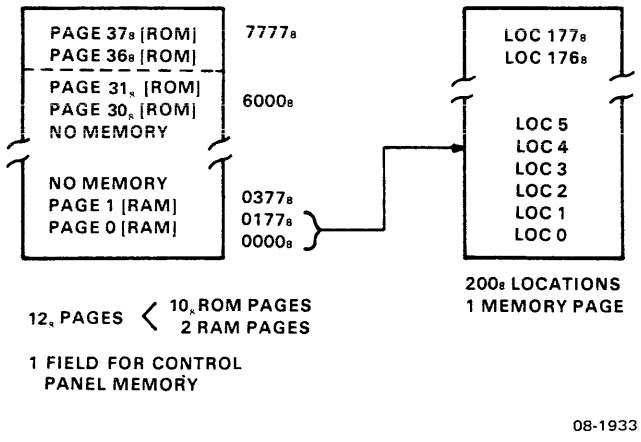


Figure 6 Panel Memory

Panel Memory Operation

The CPU is controlled by the panel memory routines at power-up during normal operation. Panel memory contains 256 words of random access memory (RAM) and 1K words of read-only memory (ROM). The routines contained in panel memory include the self-test diagnostics and bootstrap loader programs for both the MR78 and the RX78.

Figure 7 is a flow diagram of the VT78 panel memory routines. Follow this figure as you read the following panel memory discussion.

Panel memory can be entered in four different ways. The four methods are shown in the upper left corner of the flow diagram. Upon initial power-up, the VT78 enters the panel memory routines. Panel memory is entered each time the VT78 START switch is pressed. Once a program is running, detection of either a halt or a special instruction requesting panel memory causes the VT78 to enter the panel memory routines.

The routine first determines if a legal entry was made when panel memory is entered. An illegal entry causes the CPU OK indicator to be extinguished. If the entry to panel memory was legal, the routine determines which of the four conditions caused the entry.

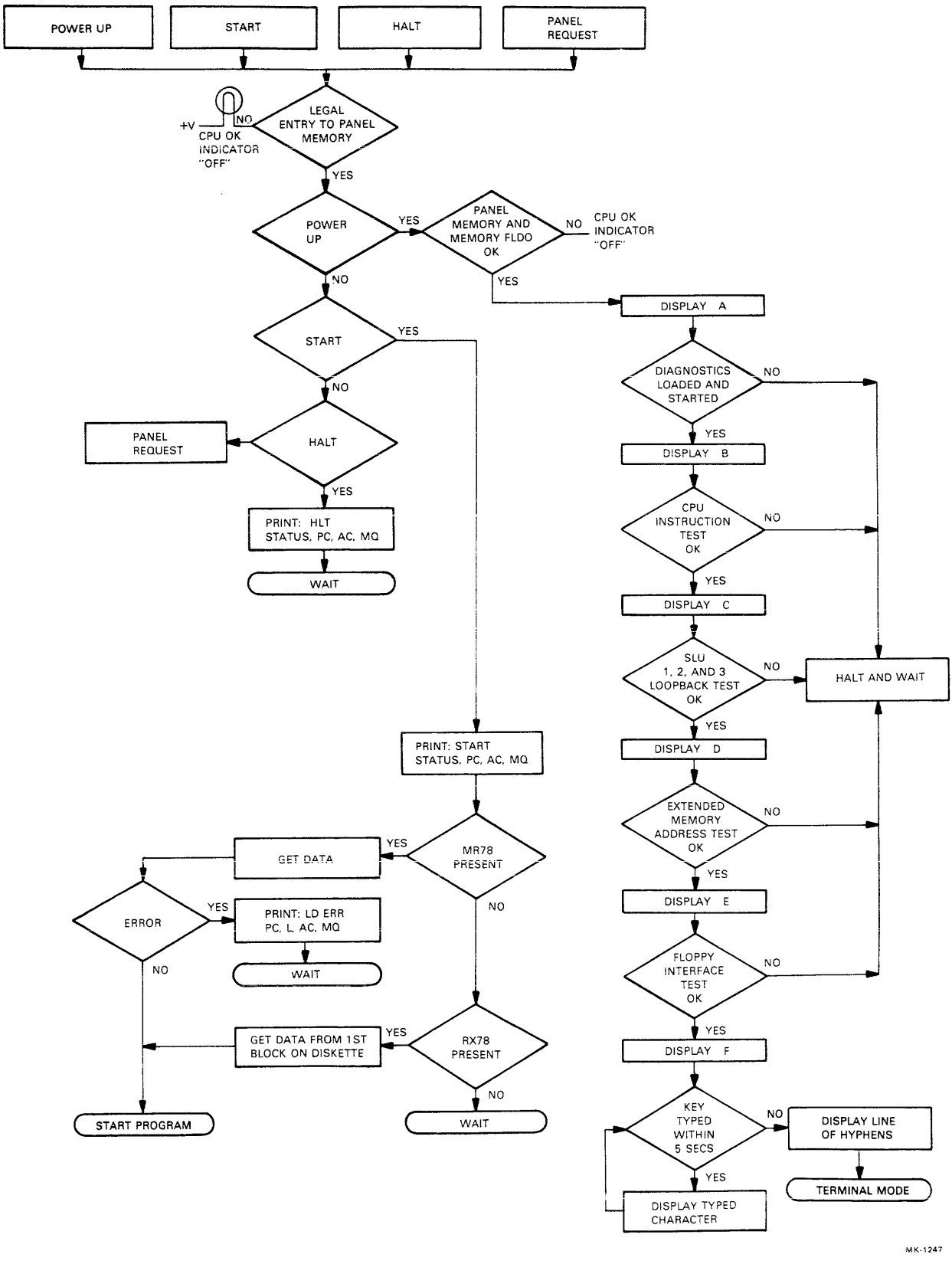


Figure 7 Panel Memory Routines Flow Diagram

Power-up causes the panel memory to perform the resident diagnostics. The diagnostic routine displays the letters A through F to indicate a successful completion of each test.

The first test (Test A) checks panel memory itself and a portion of the main memory. A test failure causes the CPU OK light to be extinguished, indicating an error condition. A satisfactory test causes the letter A to be displayed on the video screen and the next test is run.

Test B loads the diagnostics from panel memory into main memory and starts the routine. If a fault is detected, the routine halts. The indication that test B failed is the letter A displayed on the video screen. A halt message may or may not be displayed. If test B passes, the letter is displayed along with the letter A, so the entire display shows the letters AB.

Satisfactory completion of test B allows test C to begin. Test C runs a CPU instruction test to verify operation of the processor. If test C fails, the video screen only displays the letters AB. A halt message may also be displayed. Completion of test C displays the letters ABC and allows the next test to begin.

Test D is a loopback test for the three serial line units (SLUs) on the VT78. A loopback connector is not necessary for this test to be run. A failure is indicated by the letters ABC. A good test is shown by the letters ABCD.

Test E checks the remainder of memory that was not checked in test A. A good test adds the letter E to the video display, thus displaying ABCDE.

The next test checks the floppy interface circuitry. The test is run even if no RX78 is attached. If the test is successful, the letter F is added to the video screen so that the entire display shows the letters ABCDEF.

Completion of test F causes the VT78 to enter the keyboard mode. For five seconds after the letter F is displayed, any key typed on the keyboard is displayed directly on the video screen. The timer is reset each time a key is typed. As many keys can be typed as is desired as long as five seconds is not allowed to pass between typed keys. A line of hyphens is displayed once the timer counts five seconds after the last keystroke and the keyboard mode is terminated.

The VT78 enters the terminal mode once the line of hyphens is displayed. During this time, the VT78 functions as an Input/Output video data terminal. Any character typed, after F but before the line of hyphens, on the keyboard is transmitted to SLU 1. Any character received on the receive line of SLU 1 is displayed. Provisions made to loop the transmit line to the receive line allow you to check the operation of all video circuitry and SLU 1. The VT78 remains in the terminal mode indefinitely. The usual method of terminating the terminal mode is to attempt to load a program by pressing the VT78 START switch.

Pressing the VT78 START switch causes a legal entry to the panel memory routines. Figure 7 shows what action is taken if a start is detected.

First, a start message is displayed on the video screen. After the start message is displayed, the panel memory determines if an MR78 is present. If the MR78 is attached, the program contained in it is loaded into main memory. A check is made to verify that the program was loaded correctly and the program is then started. A load error message is displayed if an error is detected in the loading.

The VT78 attempts to load a program from the RX78. If an MR78 is not attached, the program is started once it is loaded. The VT78 stops and waits for further action on the operator's part if no RX78 is attached. Normally this involves powering down the VT78, attaching an MR78 or RX78, and pressing the START switch again.

The last two methods of entering the panel memory routines are a panel request and a halt. A panel request is a software request to enter some portion of the panel memory routines. The program can request that a special part of the panel memory routines, for instance the self-test diagnostics, be run before continuing with the program.

An entry is made into panel memory any time a halt is detected in the program. Figure 7 shows that a halt causes a halt message to be printed and then the VT78 simply waits for further action.

INSTRUCTIONS

Memory Reference Instructions (MRIs)

The MRIs cause the computer to operate on the contents of a memory location or to use the contents of a memory location to operate on the AC.

Figure 8 shows the MRI format. Bits 0 through 2 specify the function (op code) to be executed while the lower order bits, 3 through 11, give the operand address. Table 1 lists the mnemonics for the six MRIs, their octal codes, and the operations performed.

Table 1 MRI Operation Codes

OP Code	Mnemonic	Time (μs)	Operation*	Description
0	AND	5.92 8.8 9.4	D I A	The content of the effective location is ANDed with the contents of the AC. The result is deposited into the AC (link and MQ register are unchanged).
1	TAD	5.92 8.8 9.47	D I A	The contents of the effective location are added to the contents of the AC using 2's complement arithmetic. The result is deposited into the AC. The link bit is set (1); otherwise, link and MQ register are unchanged.
2	ISZ	9.47 12.4 13.0	D I A	The content of the effective location is incremented, then tested for a zero. If the result is zero, the next sequential instruction is skipped; otherwise, the next sequential instruction is fetched. The AC, link, and MQ register are unchanged.
3	DCA	6.5 9.47 10.0	D I A	The contents of the AC are deposited into the effective location. The AC is then cleared. Link and MQ register are unchanged.

Note: *D = Direct
I = Indirect
A = Autoindexing

Table 1 MRI Operation Codes (Cont)

OP Code	Mnemonic	Time (μs)	Operation*	Description
4	JMS	6.5 9.47 10.0	D I A	The contents of the PC are stored at the effective location and the effective location, + 1, is loaded into the PC. This is a jump to subroutine instruction (link, AC, and MQ are unchanged).
5	JMP	5.92 8.8 9.47	D I A	The effective address is loaded into the PC (link, AC, and MQ are unchanged).

Note: *D = Direct
I = Indirect
A = Autoindexing

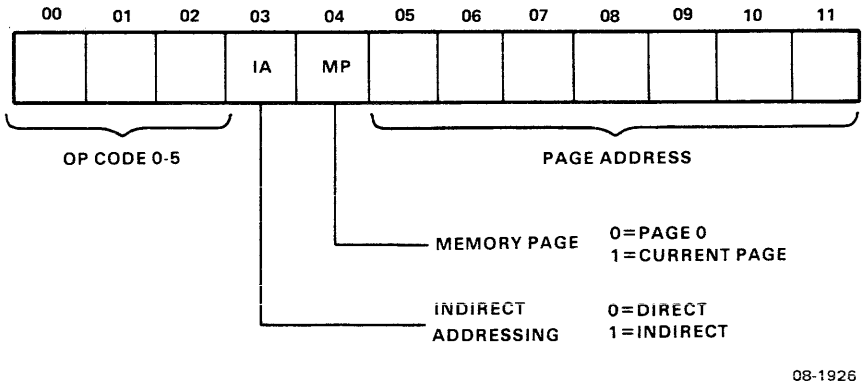


Figure 8 MRI Format

Bits 3 through 11 specify an address on one of two pages of memory. The page of memory is determined by bit 4. If bit 4 is a 0, the address specified will be in location 0000 to 0177 of page 0. If bit 4 is a 1, the location will be in the current page (the page on which the instruction is located). Bits 5 through 11 determine the absolute address on the page. For example, if an instruction is located at address 503 and bits 3 through 11 equal 364, the operand address will be in location 164 of the current page or at address 564.

After an address has been determined, the processor examines bit 3 to determine if this address is the operand address (direct addressing) or a pointer to the operand address (indirect or deferred addressing). If bit 3 equals 0, the processor operates on the address specified. If bit 3 equals 1, the processor uses the data from this page address as the 12-bit address of the operand. Notice that although the page addressing method restricts direct addressing to a maximum of 256 locations, indirect addressing allows access to all 4096 locations in the current memory field. For example, if the instruction is located at address 503 and instruction bits 3 through 11 equal 764, and if the contents of location 564 were 3015, the effective address would be 3015.

Autoindexing locations 10–17 are treated in a special manner. If an indirect reference is made through one of these addresses, the address is incremented before being used. No facility for skipping is provided when a location has been incremented to zero. To use autoindexing locations, the memory page (MP) bit must be cleared, even if the instructions are on page zero.

Operate Microinstructions

The operate microinstructions are identified by an op code of 7 (bits 0–2 = 111). Operate instructions do not specify a memory address; instead, the remaining nine bits specify the operation to be performed.

The operate instructions are subdivided into three major classes – Group I, Group II, and Group III – as described in the following paragraphs. Group I microinstructions provide manipulation of the AC and the link. Group II provides testing of the AC and link. Group III allows manipulation of the MQ.

Group I Microinstructions

Figure 9 shows the format of a Group I microinstruction.

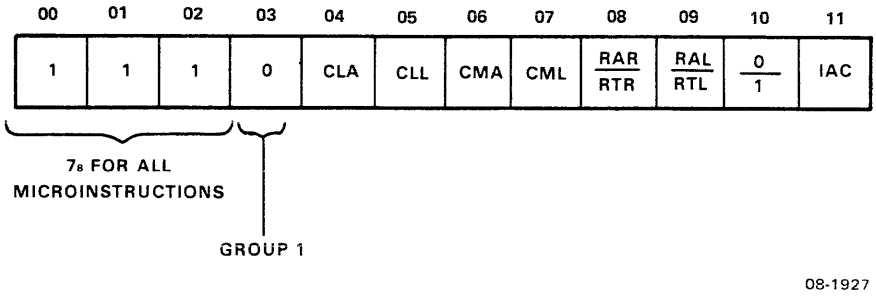


Figure 9 Group I Microinstruction Format

The Group I microinstructions allow the programmer to manipulate the AC and the link bit. The functions cause the AC to be cleared, complemented, incremented, rotated to the left or right one bit, rotated to the left or right two bits, or byte swapped. The link can be cleared, complemented, or shifted right or left with the AC. Any combination of logically nonconflicting bits can be used unless bits 8 and 9 are both set.

Table 2 lists the Group I microinstructions, their assigned mnemonics, and their operations.

Table 2 Group I Microinstructions

Octal	Mnemonic	Sequence*	Times (μs)	Description
7000	NOP	1	5.92	No operation.
7001	IAC	3	5.92	Increment AC by 1. Carry out of AC0 complements link.
7002	BSW	4	8.84	Byte swap. Right six bits are exchanged with left six bits (AC0 exchanged with AC6; AC1 with AC7; etc). Link is not changed.

Table 2 Group I Microinstructions (Cont)

Octal	Mnemonic	Sequence*	Time (μs)	Description
7004	RAL	4	8.88	Rotate AC left one bit. AC0 rotated into link. Link rotated to AC bit 11.
7006	RTL	4	8.88	Rotate AC left two bits. AC1 rotated into link. Link rotated to AC bit 10.
7010	RAR	4	8.88	Rotate AC right one bit. AC11 rotated into link. Link rotated to AC bit 0.
7012	RTR	4	8.88	Rotate AC right two bits AC10 rotated into link. Link rotated to AC bit 1.
7020	CML	2	5.92	Complement link.
7040	CMA	2	5.92	Complement AC.
7041	CIA	2,3	5.92	Complement AC and increment AC. (Forms the 2's complement.)
7100	CLL	1	5.92	Clear link (L=0).
7104	CLL RAL	1,4	8.88	Clear link then rotate AC and link left.
7106	CLL RTL	1,4	8.88	Clear link then rotate AC and link twice left.
7110	CLL RAR	1,4	8.88	Clear link then rotate AC and link right.
7112	CLL RTR	1,4	8.88	Clear link then rotate AC and link twice right.
7120	STL	1,2	5.92	Set the link (clear, then complement link).
7200	CLA	1	5.92	Clear the AC (AC=0000).

*Logical Sequence

1 = CLA, CLL
2 = CMA, CML
3 = IAC
4 = RAR, RAL, RTR, RTL, BSW

Table 2 Group I Microinstructions (Cont)

Octal	Mnemonic	Sequence*	Time (μs)	Description
7201	CLA IAC	1,3	5.92	Clear the AC then increment the AC (AC=0001).
7204	GLK	1,4	8.88	Get the link (AC is cleared; the AC and link are rotated left).
7240	STA	1,2	5.92	Set the AC (AC=7777).
7300	CLA CLL	1	5.92	Clear the AC and link.

*Logical Sequence

1 = CLA, CLL
2 = CMA, CML
3 = IAC
4 = RAR, RAL, RTR, RTL, BSW

Group II Microinstructions

Group II microinstructions allow the programmer to test the data in the AC and/or the link using the skip instructions. For Group II microinstructions, bit 3 must be a 1 and bit 11 must be a 0. Figure 10 shows the format of a Group II microinstruction.

A skip microinstruction requiring bit 8 to equal zero cannot be programmed with instructions requiring that bit 8 equals a 1.

Table 3 lists the Group II microinstructions, their assigned mnemonics, and their operations.

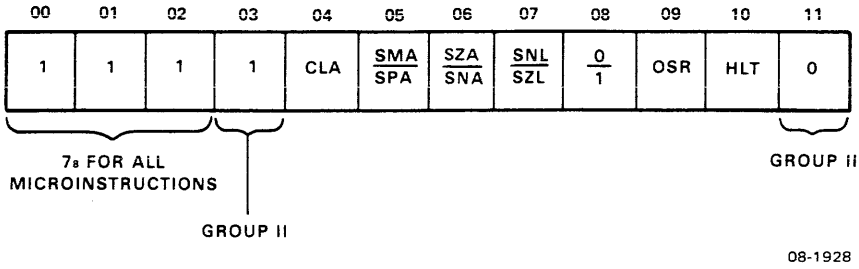


Figure 10 Group II Microinstruction Format

Table 3 Group II Microinstructions

Octal	Mnemonic	Sequence*	Time (μs)	Description
7400	NOP	1	5.92	No operation (See Group I microinstruction).
7402	HLT	4	5.92	HALT - Stops main memory processing. The VT78 displays a series of status words defining the state of the program and waits for the operator to press the START button.
7404	OSR	3	8.88	Clear AC. Load AC with preselect and MR78 status. Bits 8-11 = Preselect Bits 3- 5 = Number of binary program Bit 0 = MR78 is present.
7410	SKP	1	5.92	SKIP (PC is incremented by one, next instruction is skipped).
7420	SNL	1	5.92	Skip on nonzero link.
7430	SZL	1	5.92	Skip on zero link.
7440	SZA	1	5.92	Skip on zero AC.
7450	SNA	1	5.92	Skip on nonzero AC.
7460	SZA SNL	1	5.92	Skip on zero AC, or skip on nonzero link or both.
7470	SNA SZL	1	5.92	Skip if AC is nonzero and link is zero.
7500	SMA	1	5.92	Skip on minus AC. [If AC(0) = 1, skip.]
7510	SPA	1	5.92	Skip on positive AC. [If AC(0) = 0, skip.]
7520	SMA SNL	1	5.92	Skip on minus AC or skip nonzero link or both.
7530	SPA SZL	1	5.92	Skip if AC is positive and link is zero.

Table 3 Group II Microinstructions (Cont)

Octal	Mnemonic	Sequence*	Time (μs)	Description
7540	SMA SZA	1	5.92	Skip on minus AC or skip on zero AC or both.
7550	SPA SNA	1	5.92	Skip if AC is positive and nonzero.
7560	SMA SZA SNL	1	5.92	Skip on minus AC or skip on zero AC or skip on nonzero link or all.
7570	SPA SNA SZL	1	5.92	Skip if link is zero and AC is positive and nonzero.
7600	CLA	2	5.92	Clear AC.
7604	LAS	1,3	8.88	OR with preselect and MR78 status. Bits 8–11 = Preselect Bits 3– 5 = Number of binary program Bit 0 = MR78 is present
7640	SZA CLA	1,2	5.92	Skip on zero AC then clear AC.
7650	SNA CLA	1,2	5.92	Skip on nonzero AC then clear AC.
7700	SMA CLA	1,2	5.92	Skip on minus AC then clear AC.
7710	SPA CLA	1,2	5.92	Skip on positive AC then clear AC.

*Logical Sequence

1 = When bit 8 is 0 – either SMA, SZA or SNL
When bit 8 is 1 – Both SPA, SNA and SZL
2 = CLA
3 = OSR
4 = HLT

Group III Microinstructions

Figure 11 shows the format of a Group III microinstruction.

These microinstructions allow the programmer to manipulate the MQ register. Bits 3 and 11 must be set (1) for Group III microinstructions. Although bits 6, 8, 9, and 10 are not used, it is recommended that they always remain 0 so that code generated on this system will be compatible to computers that are equipped with extended arithmetic elements.

Table 4 lists the Group III microinstructions, their assigned octal numbers, and their operations.

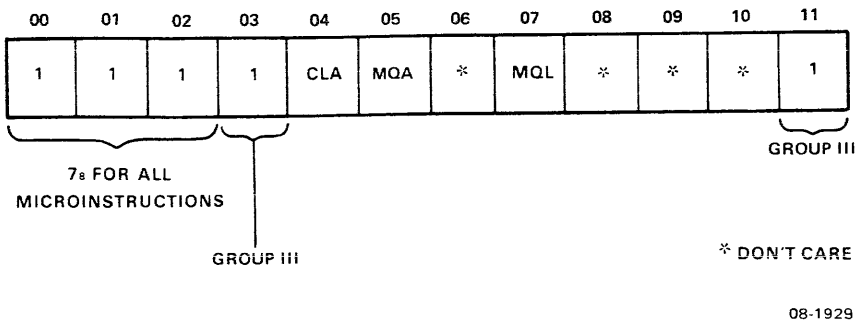


Figure 11 Group III Microinstruction Format

Table 4 Group III Microinstructions

Octal	Mnemonic	Sequence*	Time (μs)	Description
7401	NOP	3	5.92	No operation.
7421	MQL	2	5.92	MQ register load. AC is loaded into MQ, AC is then cleared.
7501	MQA	2	5.92	MQ register is loaded into AC. MQ is ORed with AC; results loaded in AC.
7521	SWP	3	5.92	Swap MQ and AC.
7601	CLA	1	5.92	Clear AC.
7621	CAM	3	5.92	Clear AC and clear MQ.
7701	ACL	3	5.92	Clear AC, then load MQ into AC.
7721	CLA SWP	3	5.92	Load MQ into AC, clear MQ.

* Logical Sequence

1 = CLA
2 = MQA, MQL
3 = SWP

I/O TRANSFERS (IOTs)

The I/O instructions (0-2) are identified by op code 6. The control bits (9-11) define the function to be performed. Bits 3-8 represent the device selection code that specifies which device is intended. These six bits will accom-

modate a maximum of 64 I/O devices. Some of these device codes are used for CPU I/O and memory extensions. All instruction times are 10 μs. Figure 12 shows the complete IOT instruction format.

Extended Memory Instruction IOTs

Memory extension is necessary because the maximum number of memory locations that can be defined by 12 bits is 4096 addresses. The VT78 system uses 16K memory; therefore, an additional two bits are needed to define any addresses above 4K. Memory extension instructions use registers to define the bits of addressing. These registers are the instruction field (IF), data field (DF), instruction buffer (IB), and the save field (SF) registers.

A list of the memory instructions and a brief description of each is provided in Table 5.

The IF is a 2-bit register that serves as an extension to the PC. Instructions and directly addressed operands are obtained from the IF.

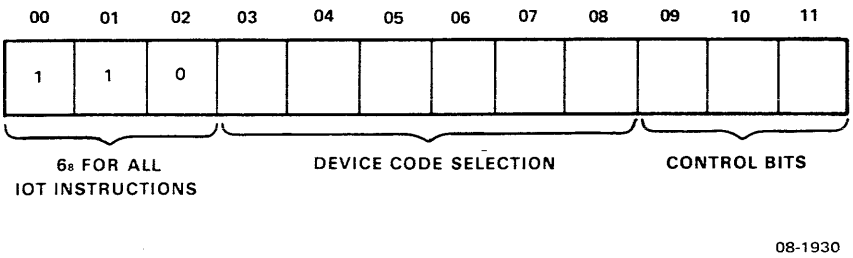


Figure 12 IOT Instruction Format

Table 5 Memory Extension Instructions

Octal	Mnemonic	Time (μs)	Description
62N1	CDF	10	Change the data field to N (N may be 0–7).
62N2	CIF	10	At the next JMP or JMS, change instruction field to N. Inhibit interrupts until change occurs (N may be 0–3).
62N3	CDF CIF	10	Combined CDF and CIF.
6214	RDF	10	Read the data field (inclusive OR) into bits 7 and 8 of AC. Note: Bit 6 will always be zero.
6224	RIF	10	Read the instruction field (inclusive OR) into bits 7 and 8 of AC. Bit 6 will always be zero.
6234	RSB	10	Read (inclusive OR) the ISF and DSF into bits 7 and 8 and bits 10–11 of the AC, respectively.
6244	RMF	10	Restore memory fields to their state prior to last interrupt by loading DSF into DF and ISF into IB and inhibiting interrupts. At the time of next JMP or JMS, IB is loaded into IF and the interrupt inhibit is removed.

The IB is a 2-bit register that serves as a holding register whenever a change instruction field (CIF) or equivalent instruction is given. When a CIF instruction is requested, the new field is loaded into the IB and interrupts are inhibited. At the next JMP or JMS, the IB is loaded into the IF and interrupts are allowed. For JMP instructions, the entire instruction cycle takes place in the old field; the new field is not selected until the next instruction fetch. In the case of JMS instructions, the execute phase (depositing the old PC) takes place in the new field. The double buffering of the IF allows the programs to JMS or JMP to any address in the new memory field.

The DF is a 3-bit register that serves as an extension to the memory address register. Indirect AND, TAD, ISZ, and DCA instructions access the current data field.

The save field register is a 4-bit register consisting of two halves – the instruction save field (ISF) and the data save field (DSF). Upon recognition of an interrupt, the contents of the IF and DF are loaded into the ISF and DSF, respectively. The IB, IF, and DF are cleared, allowing field 0 to be entered for interrupt handling. The save field is the field in which to return after the interrupt has been processed and the restore memory field (RMF) instruction has been given.

Central Processor Unit IOTs

The central processor IOTs (Table 6) control the interrupt system and panel memory.

If the interrupt system is enabled, the CPU will respond to an interrupt request by storing its PC in location 0, field 0, disabling the interrupt system. The interrupt request line being driven low by any interface will cause an interrupt. The program must then enter a skip chain to determine what I/O device caused the interrupt and proceed to service that device. After servicing, the interrupts can be turned on again.

Table 6 CPU I/O Instructions

Octal	Mnemonic	Time (μs)	Function
6000	SKON	10	Skip if interrupt system is enabled. Turn the interrupt off.
6001	ION	10	If the CPU is in panel mode, exit panel mode after fetching next instruction. If the CPU is fetching from main memory, turn interrupt system on after next instruction fetch.
6002	IOF	10	Turn the interrupt off.
6003	SRQ	10	Skip if interrupt bus is low. **NOTE** **Due to the manner that the memory extension control inhibits interrupts, this IOT will not skip if a CIF has been issued and a JMP or JMS has not been encountered.**

Table 6 CPU I/O Instructions (Cont)

Octal	Mnemonic	Time (μs)	Function
6004	GTF	10	Get Flags Bit 0 = Link Bit Bit 1 = 0 Bit 2 = 1 if interrupt bus (low) Bit 3 = 0 Bit 4 = interrupt enable flip-flop Bit 5 = 0 Bit 6 = 0 Bit 7 = ISF1 Bit 8 = ISF2 Bit 9 = 0 Bit 10 = DSF1 Bit 11 = DSF2
6005	RTF	10	Restore flags. Bit 0 is shifted left into the link, bits 7 and 8 are loaded into IB, bits 10 and 11 are loaded into DF, and interrupts are enabled.
6007	CAF	10	Clear the AC, link, and interrupt. Clear all peripheral flags, and set all interrupt enable flip-flops (except LQP78 if used).
6071	PRS	10	Read panel status into the AC. Then clear panel status. Bit 0 = Power turn on. Bit 1 = START switch. Bit 2 = CPU halted. Bit 3 = IOT 6073 (PRQ) was fetched in main memory.
6072	PST	10	Complement the CPU's run flip-flop.
6073	PRQ	10	Panel request – enter panel mode, set bit 3 of the panel status word. The PRQ processor in panel memory retrieves the word following the PRQ instruction as an address in panel memory and jumps to that address. (Refer to Table 7.)
6074	PER	10	Turn OFF the CPU OK light (used in conjunction with internal diagnostics to indicate condition of CPU – normally on).

Table 7 Panel Request IOTs

Address	Function	Restrictions
6002	Initiates dumb terminal mode (VT78 terminal functions as a conventional terminal, via SLU 2, with a host computer).	
6003	Initiates start routine. Produces same results as pressing terminal START switch.	
6200	Run floppy disk drive bootstrap.	Interrupt must be off.
6201	Run MR78 loader.	Interrupt must be off.

Real-Time Clock

The real-time clock interrupts the processor every 100 Hz if the interrupt enable is set. A skip instruction causes the program to skip an instruction if the clock flag is set. After testing and skipping on the flag, the programmer must clear the flag.

Table 8 Real-Time Clock Instructions

Mnemonic	Octal Code	Function
CLLE	6135	Load the interrupt enable from the AC 11. If AC 11 = 1, set interrupt enable. If AC 11 = 0, clear interrupt enable.
CLCL	6136	Clear clock flag.
CLSK	6137	Skip on clock flag.

RX78 Floppy Disk Drive Interface

Overview – The floppy disk drive interface transfers data between the VT78 terminal and up to two dual floppy disk drive systems. Figure 13 is a block diagram of the RX78 interface circuits. Three types of RX78 floppy disk drive systems are supported:

- The RX78-P with single density RX01 drives,
- The RX78-R with double density RX02 drives, and
- The RX78-U with double density RX02 drives switched to RX01 mode.

Using the instructions listed in Table 9, the CPU sends information through the CPU bus to the floppy disk interface requesting a desired operation. This information includes the selection of a particular disk drive and

function. The functions which the CPU selects are determined by the function code set up in the primary control word (that is, the Load Command (LCD) – 6751).

A program protocol summary is provided in Table 10 as a programming aid.

Table 9 Floppy Disk Drive Interface Instruction Set

Octal	Mnemonic	Function*
6750	SEL	Select RX78 drive pair A if AC 11 = 0 or drive pair B if AC 11 = 1.
6751	LCD	Load command, clear AC. XDR required for second byte if 8-bit mode.
6752	XDR	Transfer data register (RXDB) to/from AC.
6753	STR	Skip on transfer request (TR) flag, clear flag.
6754	SER	Skip on error flag, clear flag.
6755	SDN	Skip on done flag, clear flag.
6756	INTR	SET/CLEAR interrupt enable (AC 11 = 1 for enable, AC 11 = 0 for disable).
6757	INIT	Initialize.

*Operating time for each function is 10.2 μs.

The floppy disk drive system interface allows two modes of data transfer between the VT78 and the floppy disk drive system – 8-bit and 12-bit word lengths.

For each mode of data transfer, there can be either single density or double density storage of data for the RX02, and single density only for the RX01. For each drive system using double density recording, up to 512K 8-bit bytes of data or 256K 12-bit words can be stored and retrieved. For each drive system using single density recording, up to 256K 8-bit bytes of data or 128K 12-bit words can be stored and retrieved.

The DEVSEL/CON logic decodes instruction bits (3 through 11) to (1) select a floppy disk drive system to transmit or receive data from the VT78 terminal and (2) control all of the functions to be performed. Bit 7 selects the floppy disk drive unit. Drive 0 is selected when bit 7 equals 0. Drive 1 is selected when bit 7 equals 1. Bits 9 through 11 select the function to be performed.

Bit 4 is the maintenance bit which can be used to check the RX78 interface during on-line and off-line conditions. The on-line condition occurs when the cable connecting the VT78 terminal and the RX78 floppy disk drive system is connected. The off-line condition exists when this cable is disconnected.

While the maintenance bit is set (bit 4 equals one), data transfers between the microprocessor and the floppy disk drive system are inhibited. When bit 4 equals zero, the RUN flip-flop may be set producing RUN L or RUN 1 L (depending on the floppy disk drive system selected). The RUN L or RUN 1 L signal initiates communication between the interface and the appropriate disk drive. The RUN flip-flop is clocked in the command transfer mode when LCD is issued or in the data transfer mode to or from the drive when XDR is issued.

Bit 5 is the data transfer mode bit. When bit 5 equals 0, the 12-bit mode is selected. When bit 5 equals 1, the 8-bit mode is selected. The VT78(KK8-B) initializes into 12-bit mode.

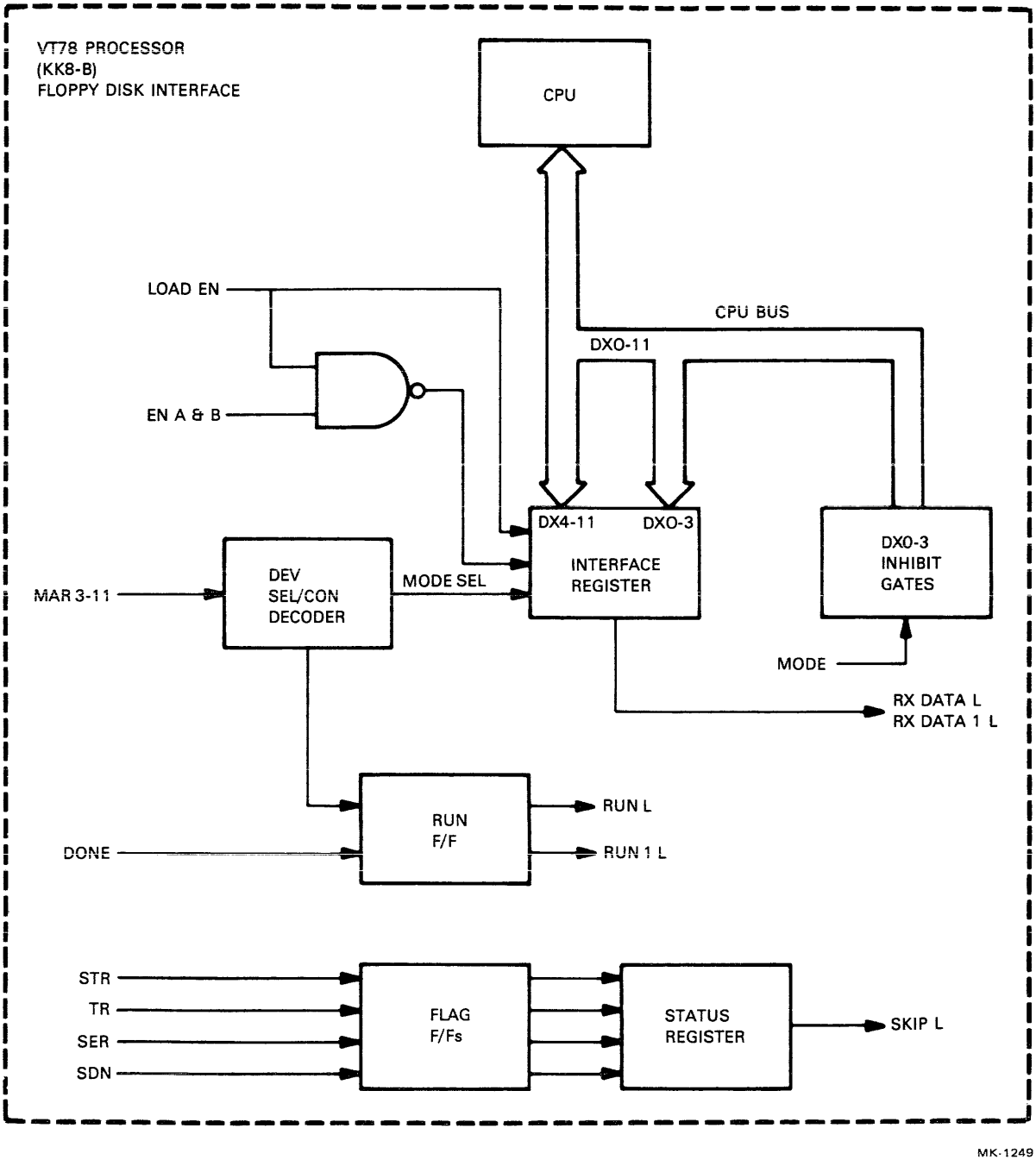


Figure 13 Floppy Disk Drive System Interface Block Diagram

In the 12-bit mode, 64 (128 if double density) words are written in a diskette sector, requiring two sectors (one sector if double density) to store one page of information. The diskette capacity in this mode is 128,128 12-bit words (1,001 pages), or 256,256 12-bit words (2,002 pages) if double density. In the 8-bit transfer mode, 128 (256 if double density) 8-bit words are written in each sector. Diskette capacity is 256,256 (512,512 if double density) 8-bit words, which is a 33 percent increase in diskette capacity over the 12-bit mode. The 8-bit mode must be used for generating IBM compatible diskettes. Since the 12-bit mode does not fully pack the sectors with data, the

Table 10 Floppy Disk Drive Interface Program Protocol Summary

Operation	Program Sequence and Description
Read or Write Sector	1. Test for done (only if first command since INIT) 2. (CLA) TAD command 3. LCD **NOTE** **If 8-bit mode, move command bits 0-3 to AC 8-11, then wait for TR, XDR.** 4. Wait for TR (STR) 5. TAD sector #, XDR 6. Wait for TR 7. TAD track #, XDR 8. Wait for done (SDN) 9. Check errors
Set Media Density (RX02 only)	1. Test for done (only if first command since INIT) 2. Load command (DEN set, DD; clear, SD) 3. Wait for TR 4. Load "1" (111_8) into RXDB 5. Wait for done (allow 15 seconds) 6. Check errors
Read ERREG	1. (CLA) TAD command 2. LCD 3. Wait for done 4. XDR (ERREG is now in AC)
Read Status (RXES)	(If previous command was not Read ERREG) 1. Test for done 2. Read RXES (RXDB) (If last command was Read ERREG on RX8; or if current DRV RDY condition required, or current DRV DEN on RX02) 1. Test for done 2. Load command 3. Wait for done 4. Read RXES (RXDB) **NOTE** **Interrupts occur on done (If INT ENB) but not on TR.**

Table 10 Floppy Disk Drive Interface Program Protocol Summary (Cont)

Operation	Program Sequence and Description
Fill or Empty Buffer (Programmed data transfer)	1. Test for done (only if first command since INIT) 2. (CLA) TAD command 3. LCD 4. If 12-bit mode wait for TR (STR). If 8-bit mode, move command bits 0-3 to AC 8-11, wait for TR and XDR, then wait for TR. 5. FILLBUF: TAD from memory, XDR EMPTBF: XDR, DCA to memory 6. TR or done? 7. If TR and not done, back to step 5 8. If done and not TR, check errors

hardware puts in extra 0s. Data transfer requests occur 23 μs after the previous request was serviced for 12-bit mode (18 μs for the 8-bit mode). There is no maximum time between the transfer request from the floppy disk drive system and servicing that request by the processor. This allows the data transfer to and from the floppy disk drive system to be interrupted without the loss of data.

When data is to be transferred to the floppy disk drive system from the CPU, a request is made by the CPU to fill the RX78 buffer. The RX78 responds by setting the transfer request (TR) flag requesting the first data word to be loaded into the buffer. The XDR command loads the data word from the microprocessor accumulator (AC) into the interface register.

The interface register consists of two 8-bit, edge-triggered universal shift registers used to temporarily store data during transfers between the microprocessor and the floppy disk drive system. Data is parallel loaded into the shift register when LOAD EN is asserted low.

The next XDR IOT then causes LOAD EN to be asserted high. LOAD EN H is ANDed with EN A H and EN B H to initiate the transfer of the data word (in serial form) from the interface register to the sector buffer in the floppy disk drive system.

Upon receipt of the entire data word, the floppy disk drive system issues another TR requesting the next data word. The transfer process is then repeated until the sector buffer is loaded (for single density recording, 64 data transfers are required for 12-bit mode or 128 transfers for 8-bit mode; for double density recording, 128 data transfers are required for 12-bit mode or 256 data transfers for 8-bit mode).

After the sector buffer is filled, the floppy disk drive system sets the done flag indicating that the function is complete.

SKIP L is asserted if any of the skip commands [skip on transfer request (STR), skip on error (SER), or skip on done (SDN)] are decoded by the DEV SEL/CON circuit and the corresponding flag is asserted.

When data is to be retrieved from the floppy disk drive system to the microprocessor, XDR causes LOAD EN, ENA, and ENB to be asserted high, shifting the first word into the interface register. The TR flag is set with the first word in the interface register. This flag denotes that a request for a data transfer from the floppy disk drive system to the microprocessor has been made. After the flag has been tested and cleared, the word is transferred to the microprocessor accumulator by the next XDR command.

After transferring the next data word into the interface register, the TR flag is set again and the transfer process is repeated until the entire contents of the floppy disk drive system's buffer register have been transferred, emptying the buffer register. The done flag is then set indicating the end of the transfer function.

To record or retrieve data, the drive performs implied seeks. Given an absolute sector address, the drive locates the desired sector and performs the indicated function, including automatic head position verification and hardware calculation and verification of the cyclic redundancy check (CRC) character. The CRC character that is read and generated is compatible with IBM 3740 equipment.

Instruction Set Descriptions – The floppy disk drive interface instruction set is listed in Table 9 and described in the following paragraphs.

Select Command (SEL) - 6750
This command determines which dual floppy drive system will be selected. When AC 11 = 0, dual floppy drive system A is selected. When AC 11 = 1, dual floppy drive system B is selected.

Load Command (LCD) - 6751
This command transfers the contents of AC to the interface register and clears AC. If AC bit 4 is zero, the floppy disk drive system begins to execute the function specified in AC 8, 9, and 10 (Figure 14) on the drive specified by AC 7. A new function cannot be initiated unless the floppy disk drive system has completed the previous function.

The DRV SEL bit (bit 7) selects one of the two floppy disk drives to perform the I/O function. When AC 7 = 0, drive 0 is selected. When AC 7 = 1, drive 1 is selected.

The 8/12 bit mode (bit 5) selects the length of the data word. When AC 5 = 0, the 12-bit mode is selected. When AC 5 = 1, the 8-bit mode is selected. The VT78 initializes into 12-bit mode.

The density bit (bit 3) indicates the recording density for the function to be performed (0 = single, 1 = double). Double density can only be used with an RX78-R (that is, RX02 drives).

The maintenance bit (bit 4) can be used to diagnose the interface under off-line and on-line conditions. The off-line condition exists when the cable is disconnected from the floppy disk drive system. The on-line condition exists when the cable is connected.

In the on-line mode, if an LCD IOT is issued when AC 4 = 1, the maintenance flip-flop is set causing the assertion of RUN on the following XDR to be inhibited. All data register transfers are then forced into AC. The maintenance bit allows the interface register to be written and read for maintenance checks. The maintenance flip-flop is cleared by initialize or by an LCD IOT when AC 4 = 0.

In the off-line mode, the contents of the interface buffer cannot be guaranteed immediately following the first LCD IOT, which sets the maintenance flip-flop. However, successive LCD IOTs guarantee the content of the interface register. The content of the interface register can then be verified by using the XDR IOT to transfer the content into AC.

The maintenance flip-flop also sets the skip flags, which remain set as long as the maintenance flip-flop is set. Skipping does not clear the flags as long as the maintenance flip-flop is set. Setting and then clearing the maintenance flip-flop leaves the skip flags in a set condition. The skip IOTs can then be issued to determine whether or not a large protion of the interface skip logic is working correctly.

The maintenance flip-flop can be used to test the INIT IOT. This is accomplished by setting the maintenance flip-flop and then clearing it to generate the flags. INIT IOT is then executed. If execution of INIT IOT is successful, all of the flags and the interrupt enable flip-flop should be cleared if previously set.

The maintenance flip-flop can be used to test the INIT IOT. This is accomplished by setting the maintenance flip-flop and then clearing it to generate the flags. INIT IOT is then executed. If execution of INIT IOT is successful, all of the flags and the interrupt enable flip-flop should be cleared if previously set.

In the on-line mode, use of the maintenance bit should be restricted to writing and reading the interface register. The same procedure described to write and read the interface register in the off-line mode should be implemented in the on-line mode. Exiting from the on-line maintenance bit mode should be finalized by an initialize to the floppy disk drive system.

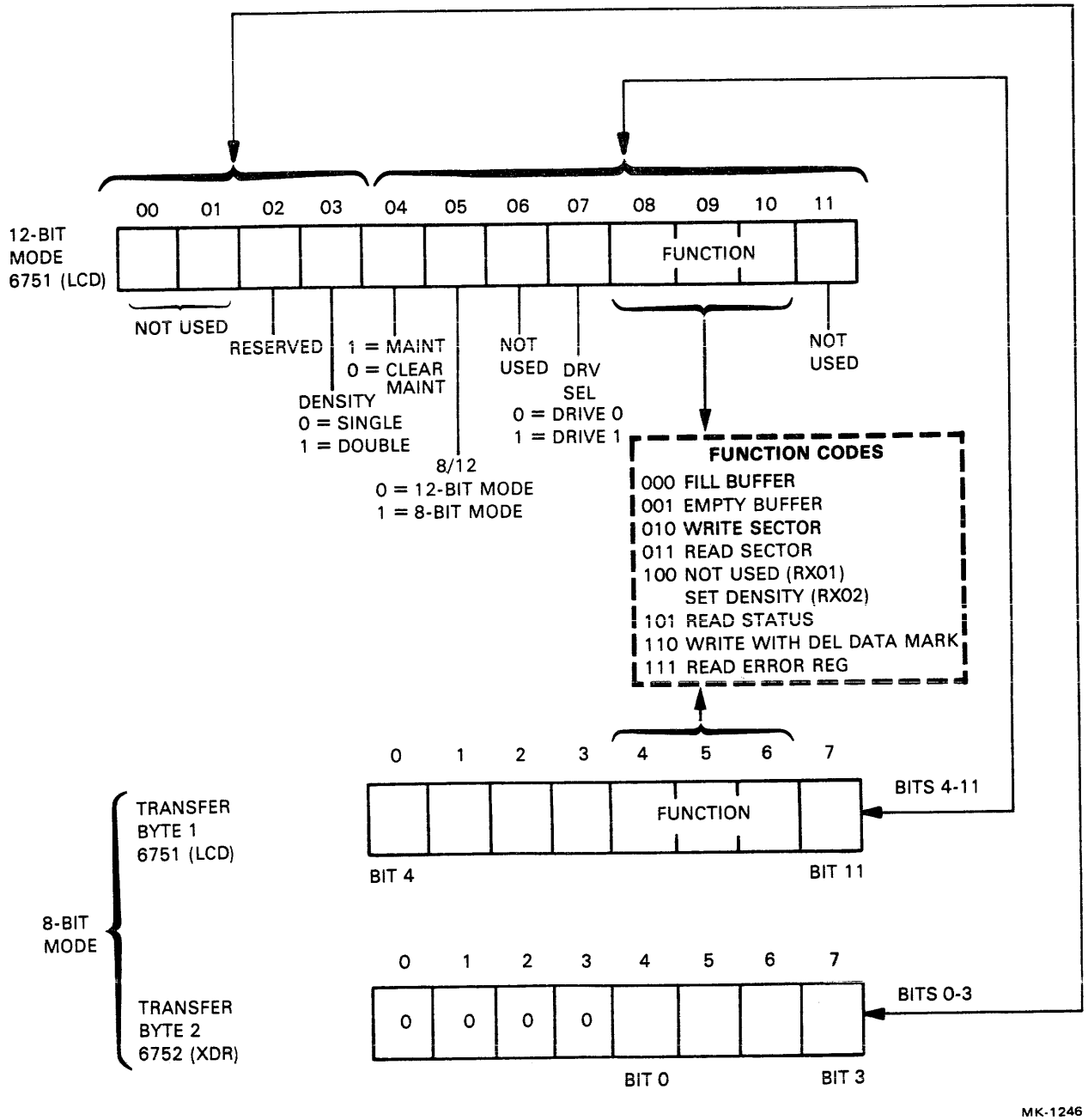


Figure 14 Floppy Disk Drive Interface Load Command Word (LCD) Format for the Command and Status Register RX2CS

Transfer Data (XDR) - 6752
With the maintenance flip-flop cleared, this instruction operates as follows. A word is transferred between the AC and the interface register. The direction of the transfer is governed by the floppy disk drive system OUT L line. The length of the word transferred is determined by the mode selected (8 or 12-bit). When done is negated,

executing this instruction indicates to the floppy disk drive system that the last data word supplied by the floppy disk drive system has been accepted by the processor, and the floppy disk drive can proceed; or that the data or address word requested by the floppy disk drive system has been provided by the processor and the floppy disk drive system can proceed.

A data transfer (XDR) from the AC always leaves the AC unchanged. If operation is in the 8-bit mode, AC bits 0–3 are transferred to the interface register but are ignored by the floppy disk drive system. Transfers into the AC are 12-bit jam transfers when in the 12-bit mode. When in the 8-bit mode, the 8-bit word is ORed into AC 4–11, and AC 0–3 remain unchanged. When the floppy disk drive system is done, this instruction can be used to transfer the status word from the interface register to the AC. The selected mode controls this transfer as indicated previously.

Skip On Transfer Request (STR) - 6753
This instruction causes the next instruction to be skipped if the TR flag has been set by the floppy disk drive system and clears the flag. The TR flag should be tested prior to transferring data or address words with the XDR instruction to ensure the data or address has been received or transferred, or after an LCD instruction to ensure the command is in the interface register. In cases where an XDR follows an LCD, the flag only has to be tested once between the two instructions.

Skip On Error (SER) - 6754
This instruction causes the next instruction to be skipped if the error flag is set by an error condition in the floppy disk drive system and clears the flag. An error also causes the done flag to be set.

The floppy disk drive system error status register contains the current error and status conditions of the selected drive. This read-only register can be accessed by the read status function (101). The RXES (or RX2ES) is also available in the interface register upon completion of any function. The RXES (or RX2ES) is accessed by the XDR instruction. The meaning of the error bits is listed in Figure 15.

Skip On Done (SDN) - 6755
This instruction causes the next instruction to be skipped if the done flag is set by the floppy disk drive indicating the completion of a function or detection of an error condition. If the done flag is set, it is cleared by the SDN instruction. This flag interrupts if interrupts are enabled.

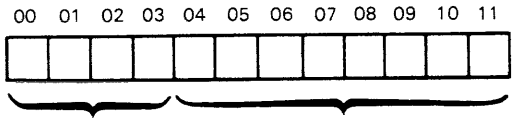
Interrupt Enable (INTR) - 6756
This instruction enables interrupts by the done flag if AC 11 = 1. It disables interrupts if AC 11 = 0.

Initialize (INIT) - 6757
The initialize instruction initializes the floppy disk drive by moving the head position mechanism of drive 1 (if drive 1 is available) to track 0. It reads track 1, sector 1, or drive 0. It zeros the error and status register and sets the done flag upon successful completion of initialize. Up to 1.8 seconds can elapse before the floppy disk drive system returns to the done state.

Register Descriptions – Only one physical register (the interface register) exists in the floppy disk drive interface as shown in Figure 13, but it may represent one of the six RX78 registers described in the following paragraphs, according to the protocol of the function in progress. Upon completion of any function, the contents of the floppy disk drive system error status register (RXES or RX2ES) are loaded into the interface register.

Command and Status Register RX2CS
The command word (primary control word) is loaded into the interface register (RX2CS register) by an LCD instruction if in 12-bit mode, or by an LCD followed by an XDR instruction if in 8-bit mode. Command word bit functions are shown in Figure 14 and are described in the Load Command (LCD) – 6751 description.

Error Register ERREG
Specific error codes can be accessed using the LCD instruction read error register function code (111). Error codes are in octal and are described in Figure 15.



CODE	MEANING
0010	Drive 0 failed to see home on Initialize.
0020	Drive 1 failed to see home on Initialize.
*0030	Found Home when stepping in 10 tracks for Init.
0040	Tried to access a track greater than 76.
0050	Home was found before desired track was reached.
*0060	Self diagnostic error (Init).
0070	Desired sector could not be found after looking at 52 headers (2 revolutions).
00110	More than 40 us and no SEP clock seen.
0120	A preamble could not be found.
0130	Preamble found but no ID mark found within allowable time span.

CODE	MEANING
*0140	CRC Error on a header, Err not asserted.
0150	The header track adress of a good header does not compare with the desired track.
0160	Too many tries for an IDAM (identifies header).
0170	Data AM not found in allotted time.
0200	CRC error on reading the sector from the disk. No code appears in the ERREG.
*0210	Parity Error on a word from I/F to Controller.
†0220	R/W electronics failed maintenance mode test.
†0230	Word count overflow.
†0240	Density Error.
†0250	Wrong key word for set media density command.

*RX01 ONLY †RX02 ONLY

MK-1248

Figure 15 Floppy Disk Drive Error Register (ERREG)

Track Address Register RX2TA
This register is loaded to indicate on which of the 77 (0 - 76) tracks a given function is to operate. It can be addressed only under the protocol of the function in progress. Bits 0-3 are unused and ignored by the control (see Figure 16).

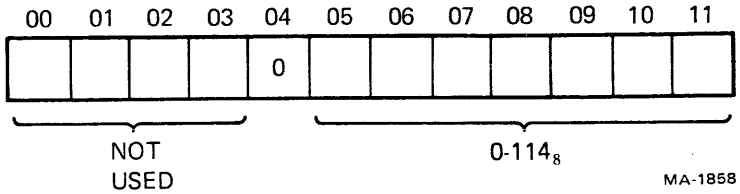


Figure 16 Floppy Disk Drive Track Address Register (RX2TA)

Sector Address Register RX2SA
This register is loaded to indicate on which of the 26 (1-26) sectors a given function is to operate. It can be addressed only under the protocol of the function in progress. Bits 0-3 are unused and ignored by the control (see Figure 17).

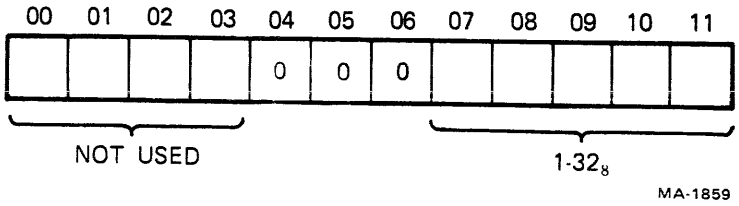


Figure 17 Floppy Disk Drive Sector Address Register (RX2SA)

Data Buffer Register RX2DB
All information transferred to and from the floppy media passes through this register and is addressable only under the protocol of the function in progress. The length of data transfer is either 8 or 12 bits, depending on the state of bit 5 of the command storage register RX2CS when the Load Command (LCD) IOT is issued (see Figure 18).

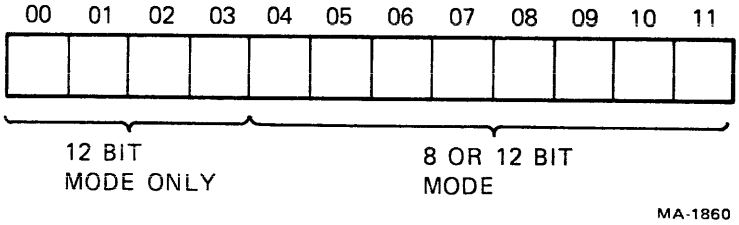
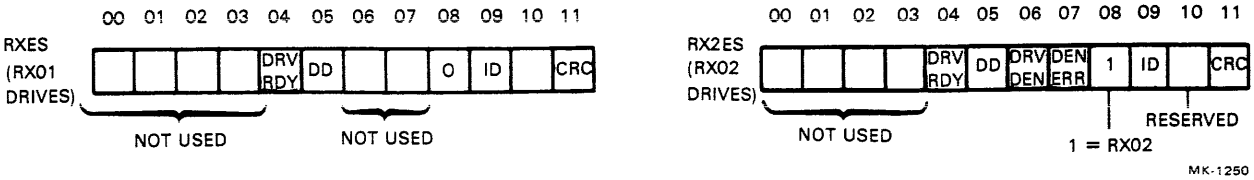


Figure 18 Floppy Disk Drive Data Buffer Register (RX2DB)

Error and Status Register RXES or RX2ES
The RX2ES (or RXES) contains the current error and status conditions of the selected drive. This read-only register can be accessed by using the LCD instruction read status function code (101). The RX2ES (or RXES) is also available in the interface register upon completion of any function. The RX2ES (or RXES) is accessed by the XDR instruction. The meaning of the error bits is given in Figure 19.

Function Code Descriptions – The floppy disk drive function codes that are set up in bits 8, 9, and 10 of the Load Command (LCD) instruction are listed in Figure 14 and described in the following paragraphs.

Fill Buffer (000)
This function loads the sector buffer in 12-bit mode with 64, 12-bit words for single density or 128, 12-bit words for double density. In the 8-bit mode, the buffer is loaded with 128, 8-bit bytes for single density or 256, 8-bit bytes for double density. This instruction only loads the sector buffer. In order to complete the transfer of the diskette, another function, write sector, must be performed. The buffer may also be read back using the empty buffer function code (001) in order to verify the data.



Bit Number	Description
11 *	CRC Error - The cyclic redundnacy check at the end of the header or data field has indicated an error. The header or data must be considered invalid. It is suggested that the data transfer be retried up to ten times, as most data errors are recoverable.
11 †	CRC Error - The cyclic redundancy check at the end of the data field has indicated an error. The data must be considered invalid. It is suggested that the data transfer be retried up to 10 times, as most data errors are recoverable (soft).
10 *	Parity Error - When status bit 10 = 1, a parity error has been detected on command and address information being transferred to the floppy disk drive system from the interface. Upon detection of a parity error, the current function is terminated, the RXES status word is moved to the interface register, and the error and done flags are set. The function can be retried to determine if the parity error is a soft or hard error. A parity error indication means that there is a problem in the interface cable between the floppy disk drive system and the interface.
10 †	Reserved.
9 *†	Initialize Done - This bit indicates completion of the Initialize routine. It can be asserted due to floppy disk drive system power failure, system power failure, or programmable or bus Initialize. This bit is not available within the RX2ES (or RXES) from a read status function.
8 †	RX02 - This bit is asserted if an RX02 system is being used.
7 †	DEN ERR - This bit indicates that the density of the function does not agree with the drive density. Upon detection of this error the control terminates the operation and asserts error and done.
6 †	DRV DEN - This bit indicates the density of the diskette in the drive selected (0 = single, 1 = double).
5 *†	Deleted Data (DD) - In the course of reading data, a deleted data mark was detected in the identification field. The data following is collected and transferred normally, as the deleted data mark has no further significance within the floppy disk drive system. Any alteration of files or actual deletion of data due to this mark must be accomplished by user software. This bit is set if a successful or unsuccessful write deleted data function is performed.
4 *†	Drive Ready - This bit is asserted if the unit currently selected exists, is properly supplied with power, has a diskette installed properly, has its door closed, and has a diskette up to speed.

NOTE

1. **This bit is only valid for either drive when retrieved via a read status function or for drive 0 upon completion of an Initialize.**
2. **If the error bit was set in the RX2CS but error bits are not set in the RX2ES (or RXES), specific error conditions can be accessed using an LCD instruction read error register function code (111) followed by an XDR instruction.**

* RX01
† RX02

Figure 19 Floppy Disk Drive Error and Status Register (RXES or RX2ES)

Upon decoding the fill buffer function, the floppy disk drive system sets the transfer request (TR) flag, signaling a request for the first data word. The TR flag must be tested and cleared by the host processor with the STR instruction prior to each successive XDR IOT. The data word can then be transferred to the interface register by means of the XDR IOT.

The floppy disk drive system next moves the data word from the interface register to the sector buffer and sets the TR flag as a request for the next data word. The sequence above is repeated, until the sector buffer has been loaded (64 data transfers for 12-bit mode or 128 data transfers for 8-bit mode). After the 64th (or 128th) word has been loaded into the sector buffer, the RX2ES (or RXES) contents are moved to the interface register, and the floppy disk drive system sets the done flag to indicate the completion of the function. Therefore, it is unnecessary for the processor to keep a count of the data transfers.

Any XDR command, after done is set, loads the RXES or RX2ES status word into the AC. The sector buffer must be completely loaded before the floppy disk drive system sets done and recognizes a new command. An interrupt now occurs if Interrupt Enable is set.

Empty Buffer (001)
This function moves the contents of the sector buffer to the processor. Upon decoding this function the error and status register (RXES or RX2ES) bits are cleared and the TR flag is set with the first data word in the interface register. This TR flag signifies the request for a data transfer from the buffer to the processor. The flag must be tested and cleared; then the word can be moved to the AC by an XDR command.

The direction of transfer for an XDR command is controlled by the floppy disk drive system. The TR flag is set, again, with the next word in the interface register. The above sequence is repeated until all words or bytes have been transferred, thus emptying the sector buffer. The done flag is then set after the RX2ES or RXES is moved in the interface register to indicate the end of the function. An interrupt now occurs if Interrupt Enable is set.

NOTE
The empty buffer function does not destroy the contents of the sector buffer.

Write Sector (010)
This function transfers the contents of the sector buffer to a specific track and sector on the diskette. Upon decoding this function, the floppy disk drive system clears the RX2ES (or RXES) and sets the TR flag, signifying a request for the sector address. The TR flag must be tested and cleared before the binary sector address can be loaded into the interface register by means of the XDR command. The sector address must be within the limits 1-32_8.

The TR flag is set, signifying a request for the track address. The TR flag must be tested and cleared; then the binary track address may be loaded into the interface register by means of the XDR command. The track address must be within the limits 0-114_8.

The floppy disk drive system tests the supplied track address to determine if it is within the allowable limits. If it is not, the RX2ES (or RXES) is moved to the interface register, the error and done flags are set, and the function is terminated.

If the track address is legal, the floppy disk drive system moves the head of the selected drive to the selected track, locates the requested sector, transfers the contents of the sector buffer and a CRC character to that sector, and sets done. Any errors encountered in the seek operation cause the function to cease, the RX2ES (or RXES) to be loaded into the interface register, and the error and done flags to be set. If no errors are encountered, the RX2ES (or RXES) is loaded into the interface register and only the done flag is set.

NOTE
The write sector function does not destroy the contents of the sector buffer.

Read Sector (011)
This function moves a sector of data from a specified track and sector to the sector buffer. Upon decoding this function, the floppy disk drive system clears RX2ES (or RXES) and sets the TR flag, signifying the request for the sector address. The flag must be tested and cleared. The sector address is then loaded into the interface register by means of the XDR command. The TR flag is set, signifying a request for the track address. The flag is tested and cleared by the processor and the track address is then loaded into the interface register by an XDR command.

The legality of the track address is checked by the floppy disk drive system. If illegal, the error and done flags are set, the contents of RX2ES (or RXES) are moved to the interface register, and the function is terminated. Otherwise, the floppy disk drive system moves the head to the specified track, locates the specified sector, transfers the data to the sector buffer, and computes and checks CRC for data. If no errors occur, the done flag is set with the RX2ES (or RXES) in the interface register. If an error occurs anytime during the execution of the function, the function is terminated by setting the error and done flags with RX2ES (or RXES) in the interface register. A detection of a CRC error results in RX2ES (or RXES) bit 11 being set. If a deleted data mark is encountered at the beginning of the desired data field, RX2ES (or RXES) bit 5 is set.

Set Media Density (100) For RX02 Only
This function causes the entire diskette to be reassigned to a new density. The density bit (bit 3 RX2CS) indicates the new density of the diskette. The control reformats the diskette by writing new data address marks (double or single density) and zeroing out all data fields on the diskette. Before executing the command the control looks for a protective key word of 01001001 (ASCII'I').

The control starts at sector 1, track 0, and reads the header information. It then starts a write operation, writing the new data address mark and data field as well as CRC characters. If the header information is damaged, the control aborts the operation and asserts the done and error flags.

This operation takes about 15 seconds and should not be interrupted. If for any reason the operation is interrupted, an illegal diskette is generated which may have data marks of both densities. This diskette should again be completely reformatted.

Read Status (101)
Upon decoding this function, the floppy disk drive system moves RXES (or RX2ES) to the interface register and sets the done flag. RXES (or RX2ES) can then be read by the XDR instruction.

NOTE

The average time for this function is 250 ms. Excessive use of this function will result in substantially reduced throughput.

For RX02 drives, this function updates the drive ready and drive density status of the selected drive, clears the INIT DONE bit, updates the Unit Sel, possibly sets the density error bit, and leaves the remainder of the RX2ES unchanged. The drive density is updated by loading the head on the selected drive (without changing head and reading position) with the first header and data mark that randomly appears under the head. The control then generates the appropriate number of shift pulses which transfer the RX2ES (error and status) register over the interface. Upon completion of RX2ES transfer, the control asserts done to complete the operation.

Write Deleted Data Sector (110)
This function is identical to the write data function except that a deleted data mark is written prior to the data field rather than the normal data mark. RX2ES (or RXES) bit 5 (Deleted Data) is set in the interface register upon completion of the function.

Read Error Register Function (111)
The read error register function can be used to retrieve explicit error information upon detection of the error flag. Upon receiving this function, the floppy disk drive system moves an octal error code to the interface register and sets done. The interface register can then be read via an XDR command and the code interrogated to determine which type of failure occurred. Error code descriptions are given in Figure 15.

NOTE

Care should be exercised in the use of this function. The program must perform this function before a read status because the error register is always modified by a read status function.

Power Fail – There is no actual function code associated with power fail. When the floppy disk drive system senses a loss of power, it unloads the head and aborts all controller action. All status signals are invalid while power is low.

When the floppy disk drive system senses the return of power, it removes done and begins a sequence to:

1. Move drive 1 head position mechanism to track 0,
2. Clear any active error bits,
3. Read sector 1 of track 1 of drive 0 into the buffer, and
4. Set the Initialize Done bit of the RX2ES (or RXES), after which done is again asserted.

There is no guarantee that information written at the time of a power failure is retrievable. However, all other information on the diskette remains unaltered.

INIT IOT is a method of aborting an incomplete function.

Error Recovery – There are two error indications given by the floppy disk drive system. The read status function assembles the current contents of the RX2ES (or RXES) which can be sampled to determine errors. The read error register function can also be used to retrieve explicit error information.

The results of the read status function or the read error register function are in the interface register when done sets, indicating the completion of the function. The XDR IOT must be issued to transfer the contents of the interface register to the processor's AC.

NOTE

A read status function is not necessary if the DRV RDY bit is not going to be interrogated because the RX2ES (or RXES) is in the interface register at the completion of every function.

The error codes for the read error register function are presented in Figure 15.

MR78 Interface

The MR78 interface is available for interfacing the optional MR78 device with theVT78 terminal via the connector on the I/O distribution panel.

The MR78 device, described in detail in the *DECstation Technical Manual*, contains a programmed set of ROMs and is accessed using the IOTs listed in Table 11. Data from the MR78 is in binary paper tape format. The data is checksummed and, if an error is encountered, it is reported on the display as a HLT with status information. The MR78 interface port does not provide for interrupts.

Table 11 MR78 IOTs

Octal	Mnemonic	Function
6011	RSF	Skip if a character is available.
6016	RRB RFC	OR the character presently at the port into the AC (4–11), clear the character available flag, and obtain a new character. Then set character available flag when next character arrives.

Serial Line Unit 1 (SLU 1)
SLU 1 is hardwired to the keyboard/video display. The IOTs associated with SLU 1 are listed in Table 12.

Table 12 SLU 1 IOTs

Octal	Mnemonic	Time (μs)	Description
6030	KCF	10	Clear the input flag.
6031	KSF	10	Skip if input flag set.
6032	KCC	10	Clear input flag, clear AC.
6034	KRS	10	OR content of input buffer into AC.
6035	KIE	10	Load content of AC(11) into interrupt enable flip-flop.
6036	KRB	10	Load content of input buffer into AC, then clear input flag.
6037	KLB	10	Load content of AC(11) into loopback flip-flop. NOTE **This IOT is used for internal diagnostics. Setting the loopback flip-flop disconnects all SLUs from their connector and connects the outputs back to the inputs of each SLU including the video terminal SLU 1.**

Table 12 SLU 1 IOTs (Cont)

Octal	Mnemonic	Time (μs)	Description
6040	SPF	10	Set output flag enable.
6041	TSF	10	Skip if output flag is set and flag enable is set.
6042	TCF	10	Clear output flag enable.
6043	TSB	10	Set the baud rate. (See Table 13.)
6044	TLS	10	Same as 6046.
6045	TSK	10	Skip if SLU 1 interrupt enable is set and the input flag is set; or if the SLU 1 interrupt enable flag is set and output flag enable are both present.
6046	TLS	10	Load the content of AC 4–11 into the SLU transmitter and send the character out over the serial line. Set the output flag enable flip-flop. As soon as the new character is loaded into the transmitter, set the output flag.

Serial Line Unit 2 (SLU 2)

SLU 2 is a full-duplex EIA port with programmable baud rate selection. SLU 2 is different from the other SLUs because it has programmable bit detection, programmable data terminal ready, character length, stop bit selection, and parity generation. Error bit detection is accomplished with K1E1 IOT. The remaining programmable options are done with KMD1 IOT. The device input code of SLU 2 is 30, the output code is 31. The IOTs associated with SLU 2 are listed in Table 14.

Table 13 Baud Rates

AC 8–11	Baud Rate	AC 8–11	Baud Rate
0	50	10	1800
1	75	11	2000
2	110*	12	2400*
3	134.5	13	3600
4	150*	14	4800*
5	300*	15	7200
6	600*	16	9600*
7	1200*	17	19200

*Available preselect switch settings for SLU 2.

Table 14 SLU 2 IOTs

Octal	Mnemonic	Description
6300	KCF1	Clear input flag.
6301	KSF1	Skip if input flag is set.
6302	KCC1	Clear input flag, clear AC.
6303	-	-
6304	KRS1	OR contents of input buffer into AC.
6305	KIE1	Load content of AC10 into status enable flip-flop and load contents of AC11 into interrupt enable flip-flop. (See Table 15.)
6306	KRB1	Load contents of input buffer into AC, then clear input flag.
6307	KMD1	Set the operating mode of SLU 2. (See Table 16.)
6310	SPF1	Set output flag enable.
6311	TSF1	Skip if output flag is set and flag enable is set.
6312	TCF1	Clear output flag enable.
6313	TSB1	Set baud rate according to Table 13. Normally, this instruction is preceded by an LAS instruction, which obtains a bit pattern from the rotary switch on the I/O distribution panel.
6314	-	This IOT is the same as 6316 (TLS1).
6315	TSK1	Skip if SLU 2 interupt enable is set and if the input flag is set or if the output flag and output flag enable are both present.
6316	TLS1	Load the contents of AC 4–11 into the SLU transmitter and send the character over the serial EIA line. As soon as a new character is loaded into transmitter set output flag.

Table 15 Loading Contents of AC into Status Enable and Interrupt Enable Flip-Flops

AC Bit	Function
AC 10	If AC(10) = 1, during KIE1 IOT, error bits are enabled and four extra bits are read into the AC along with the incoming character. These are: AC0 Set if any error exists. AC1 Set if parity error has occurred. AC2 Set if framing error has occurred. A framing error is caused by an invalid stop bit and usually signifies either a break character has been transmitted or the incoming line is open. AC3 Set if overrun error has occurred. An overrun error occurs when a new character arrives before the last one was serviced by the program.

Table 16 Setting SLU 2 Operating Modes

AC Bit	Function
	NOTE **If the AC bits defined in this table are set, the associated functions will be performed (SLU 2 only) with the KMD1 IOT.**
6	If AC(6) = 0, set data terminal ready; If AC(6) = 1, clear data terminal ready.
7	If AC(7) = 0, generate and check parity. If AC(7) = 1, inhibit parity generation and checking and clear parity error bit.
8	If AC(8) = 0, select 1 stop bit. If AC(8) = 1, select 1.5 stop bits for a 5-element character. Select 2.0 stop bits for all other lengths.
9, 10	Select number of elements per character length. 9 10 Elements per character 0 0 5 0 1 6 1 0 7 1 1 8
11	If AC 11 = 0 and AC 7 = 0, generate and check for odd parity. If AC 11 = 1 and AC 7 = 0, generate and check for even parity.

Serial Line Unit 3 (SLU 3)

SLU 3 has a programmable baud rate, but is hardwired for 8-bit words. It has no parity or error detection capabilities and is always operated in the full-duplex mode.

The IOTs associated with SLU3 are listed in Table 17.

Table 17 SLU 3 IOTs

Octal	Mnemonic	Description
6320	KCF2	Clear input flag.
6321	KSF2	Skip if input flag is set.
6322	KCC2	Clear input flag and clear AC.
6323	-	-
6324	KRS2	OR content of input buffer into AC.
6325	KIE2	Load contents of AC(11) into interrupt enable flip-flop.
6326	KRB2	Load contents of input buffer into AC then clear input flag.
6330	SPF2	Set output flag enable.
6331	TSF2	Skip if output flag is set and flag enable is set.
6332	TCF2	Clear output flag enable.
6333	TSB2	Set baud rate according to Table 13.
6334	-	Same as 6336 (TLS2).
6335	TSK2	Skip if SLU3 interrupt enable is set and the input flag is set or the output flag and output flag enable are both present.
6336	TLS2	Load the contents of AC 4–11 into the SLU transmitter and send the character over the serial EIA line. As soon as a new character can be loaded into the SLU transmitter, set the output flag.

Parallel I/O

The parallel I/O permits 12-bit data to be transferred between the processor accumulator and peripheral devices. It has 12 bidirectional lines. Two device codes are available for this interface – device code 66 and 50. The choice between device codes is made by the logic level present at one pin of the parallel I/O connector. Device code 66 is generally used to generate LA78 printer IOTs. Device code 50 is used for the LQP78 printer. The parallel I/O may also be used as a general-purpose 12-bit parallel I/O port using either set of IOT instructions and the signal line OUT to allow input of externally supplied data.

The parallel I/O IOTs for the LA78 printer and the LQP78 printer are listed and described in Tables 18 and 19, respectively. The parallel I/O IOTs for the DP78 sychronous communications interface are listed in the *DP78 Synchronous Communications Unit Manual* listed in the Preface of this manual.

Table 18 Parallel IOTs - LA78 Printer

Octal	Mnemonic	Time (μs)	Description
6660	PSSF	10	Set the print flag (AC unchanged).
6661	PSKF	10	Skip on flag (if the character RDY flag is set the next sequential instruction is skipped).
6662	PCLF	10	Clear flag (the character RDY flags are cleared - AC not changed. Status register not affected).
6663	-	10	Not used.
6664	PSTB	10	Load printer buffer. AC (0:11) are transferred to the interface buffer register. Approximately 200 ns later, character strobe is issued. AC is not changed.
6665	PCIE	10	Set or clear interrupt enable* if AC11(0) interrupt enable is cleared. If AC11(1) interrupt enable is set, AC is not changed.
6666	PCLF PSTB	10	Load buffer and clear flags (combination 6664 and 6662).
6667	PRDB	10	Read data. If the OUT line is high, the contents of the previous buffer load (6664 or 6666) are read back into the AC. If OUT is low, a 12-bit word supplied by the external device is loaded into the AC.

*After CAF or power on interrupts are enabled.

Table 19 Parallel IOTs - LQP78 Printer

Octal	Mnemonic	Time (μs)	Description
6500	LQSK	10	Skip on done flag. If any of the following conditions exist, skip the next instruction: the transition from high to low of "character ready," "paper ready," "carriage ready," "printer ready," or the "check" line has taken place since this IOT was last issued.
6501	LQRB	10	Read buffer if the OUT line is high. The contents of the output register are read back into the AC. If the OUT line is low, a 12-bit word supplied by the external device is read into the AC.

Table 19 Parallel IOTs - LQP78 Printer (Cont)

Octal	Mnemonic	Time (μs)	Description
6502	LQMP	10	Move paper. AC (0:11) are sent to the output register and, 200 ns later, a "paper strobe" pulse is issued. The AC and the paper ready flag are cleared.
6503	LQMC	10	Move carriage. AC (0:11) are sent to the output register and, 200 ns later, a "carriage strobe" pulse is issued. The AC and the carriage ready flag are cleared.
6504	LQPC	10	Print a character. AC (0:11) are sent to the output register and, 200 ns later, a "character strobe" is issued. The AC and the character ready flag are cleared.
6505	LQRS	10	Read status and clear done flag. The state of the lines listed below are stored in the AC, then the done flag is cleared. ACO = 1 Printer Ready 1 = Character Ready 2 = Carriage Ready 3 = Paper Ready 4 = 0 5 = 0 6 = Check AC 7 = 0 AC 8 = 0 AC 9 = 0 AC 10 = Lift ribbon AC 11 = Interrupt enable
6506	LQLS	10	Write status and set done. AC (10:11) are sent to the interface to set or clear certain control bits AC 10(1) - Set lift ribbon AC 10(0) - Clears lift ribbon AC 11(1) - Sets interrupt enable* AC 11(0) - Clears interrupt enable.
6507	LQRE	10	Restore and clear done flag. Restore strobe is issued for approximately 200 ns. The done flag is cleared. AC is not affected.

*After CAF and power up, interrupts are disabled.

KEYBOARD/VIDEO DISPLAY

The keyboard/video display (Figure 20) enables the user to communicate with the processor and control all of the system peripheral devices.

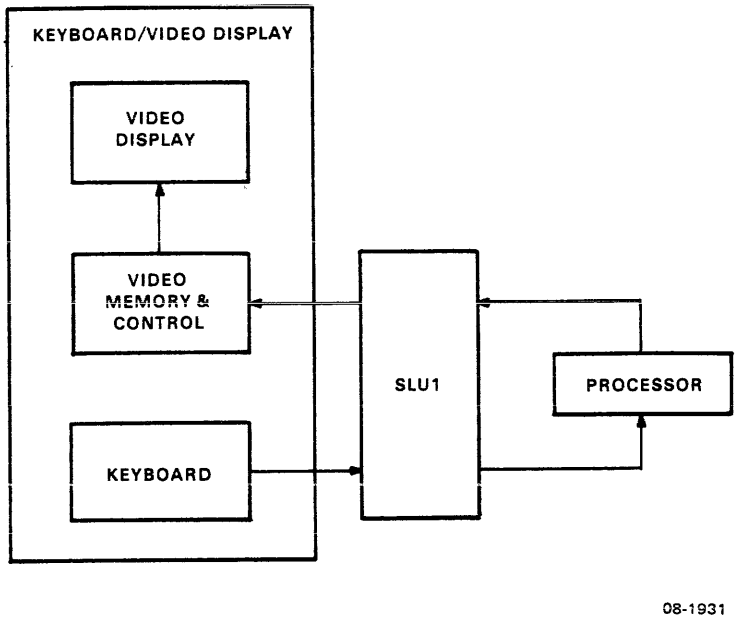


Figure 20 Keyboard/Video Display Block Diagram

The keyboard/video display features full-duplex operation with 8-bit ASCII characters. These characters are transmitted or received as serial data coded with a start bit, eight data bits, and a stop bit.

The keyboard/video display is hard-wired to SLU 1 (part of the processor), which provides the interface between the keyboard/video display and the processor.

Keyboard

The keyboard transmits codes to the processor using a set of I/O transfer instructions (device code 03). Some keyboard keys transmit one or more codes to the processor when typed. Others, such as the shift key, are control keys and send no codes.

The code transmitting keys cause the keyboard/video display to make a clicking sound to verify that a keystroke has been sent to the processor. The code is transmitted in the order that the keys are pressed. If three keys are pressed simultaneously, the first two key codes will be transmitted immediately but the third will not be transmitted until the first two keys are released.

All alphabetic keys transmit in uppercase and lowercase code. Uppercase is transmitted when a key is typed while the SHIFT or the CAPS LOCK key is down. (The CAPS LOCK key does not affect codes transmitted by keys other than the alphabetic keys.) The uppercase and lowercase key codes are listed in Table 20.

The numeric and symbol key codes are listed in Table 21.

Table 20 Alphabetic Key Codes

Key	Uppercase Code*	Lowercase Code*
A	101	141
B	102	142
C	103	143
D	104	144
E	105	145
F	106	146
G	107	147
H	110	150
I	111	151
J	112	152
K	113	153
L	114	154
M	115	155
N	116	156
O	117	157
P	120	160
Q	121	161
R	122	162
S	123	163
T	124	164
U	125	165
V	126	166
W	127	167
X	130	170
Y	131	171
Z	132	172

*All codes are expressed in octal. Note that there are no 8s or 9s in the octal system.

Table 21 Numeric/Symbol Key Codes

Key	Neither SHIFT Key Down	Either or Both SHIFT Keys Down
1	061	041 (!)
2	062	100 (@)
3	063	043 (#)
4	064	044 ($)
5	065	045 (%)
6	066	136 (^)
7	067	046 (&)
8	070	052 (*)
9	071	050 (()
0	060	051 ())
-	055	137 (_)
=	075	053 (+)
[	133	135 (])
;	073	072 (:)
'	047	042 (")
,	054	074 (<)
.	056	076 (>)
	057	077 (?)

The function keys transmit control codes to the processor. They cause an action to occur in the keyboard/video display such as tab, line feed, etc., if the processor echos these codes back to the keyboard/video display.

The CTRL key is used in conjunction with other keys on the keyboard to produce control codes in the range of 000-037. When held down it alters the code normally produced by a typed key by forcing the two high-order bits of the code to zero. Table 22 lists the codes transmitted when the control key is held down. The original code is included for comparison.

Table 22 Control Key Codes

Code Originally Transmitted	With The CTRL Key Down	
	Code Actually Transmitted	Special Name
040 or 100 or 140	000	NUL
041 or 101 or 141	001	SOH
042 or 102 or 142	002	STX
043 or 103 or 143	003	ETX
044 or 104 or 144	004	EOT
045 or 105 or 145	005	ENQ
046 or 106 or 146	006	ACK
047 or 107 or 147	007	BELL
050 or 110 or 150	010	BS
051 or 111 or 151	011	HT
052 or 112 or 152	012	LF
053 or 113 or 153	013	VT
054 or 114 or 154	014	FF
055 or 115 or 155	015	CR
056 or 116 or 156	016	SO
057 or 117 or 157	017	SI
060 or 120 or 160	020	DLE
061 or 121 or 161	021	DC1 or XON
062 or 122 or 162	022	DC2
063 or 123 or 163	023	DC3 or XOFF
064 or 124 or 164	024	DC4
065 or 125 or 165	025	NAK
066 or 126 or 166	026	SYN
067 or 127 or 167	027	ETB
070 or 130 or 170	030	CAN
071 or 131 or 171	031	EM
072 or 132 or 172	032	SUB
073 or 133 or 173	033	ESC
074 or 134 or 174	034	FS
075 or 135 or 175	035	GS
076 or 136 or 176	036	RS
077 or 137 or 177	037	US

The ESC key is used to select a command rather than the characters noted on the character keys. For example, if the keyboard/video display receives code 102, it will display a B on the video screen. If it receives 033, or the ESC key is pressed when the letter B is typed, the code preceding 102 will initiate a command rather than display the character. The letter B will not be displayed on the video screen. If a control code is sent to the keyboard/video display between the ESC and the final character, the function specified by the control code will be performed when the control code is received. The function specified by the escape sequence will be performed when the final character is received.

If the keyboard/video display receives ESC ESC from the processor, the second ESC will not cancel the escape sequence, but will continue to be ready to interrupt the next character in the escape sequence.

Typing the BREAK key forces the serial data output line of the keyboard/video display to the zero state for as long as the key is held down. The BREAK function is provided for users with software written to operate in the half-duplex mode. The keyboard/video display normally operates in the full-duplex mode so there is usually no need for the BREAK function.

The 19-key auxiliary keypad is provided for applications requiring heavy use of the numeric keys. In addition to the 10 numeric keys, the keypad has a decimal point key, four cursor move keys, three blank keys, and an ENTER key.

In the normal mode of operation, the decimal point key and the numeric keys transmit the same code as the decimal point key and the numeric keys on the main keyboard. The processor cannot distinguish between them. The ENTER key transmits the same code as the RETURN key.

In response to a command from the processor, the keyboard/video display enters the alternate-keypad mode and the ENTER, decimal point, and numeric keys transmit a unique escape sequence. This allows the processor to distinguish between main keyboard entries and auxiliary keypad entries. It also provides the processor with 12 user-definable keys to use for individual applications. Table 23 lists the codes transmitted by these keys.

Table 23 Keypad Numeric Key Codes

Key	Code(s) Transmitted To Processor	
	Key Not in Alternate-Keypad Mode	Alternate-Keypad Mode
0	060	030 077 160 (ESC ? p)
1	061	033 077 161 (ESC ? q)
2	062	033 077 162 (ESC ? r)
3	063	033 077 163 (ESC ? s)
4	064	033 077 164 (ESC ? t)
5	065	033 077 165 (ESC ? u)
6	066	033 077 166 (ESC ? v)
7	067	033 077 167 (ESC ? w)
8	070	033 077 170 (ESC ? x)
9	071	033 077 171 (ESC ? y)
.	056	033 077 156 (ESC ? n)
ENTER	015	033 077 115 (ESC ? M)

Note: None of the keys on the auxiliary keypad are affected by pressing the SHIFT, CAPS LOCK, or CTRL keys.

The seven remaining keys on the keypad are the four cursor move keys and three blank keys. The cursor move keys transmit an escape sequence to the processor. If the processor echos these codes back to the keyboard/video display, the cursor moves one character position up, down, left, or right, depending on the typed key. The three blank keys transmit user-defined escape sequences. The user can define the meaning of each key to fit a particular application. Table 24 lists the codes transmitted by these seven keys.

Table 24 Keypad Control Key Codes

Keypad Key	Code(s) Transmitted to Processor
Left blank key	033 120 (ESC P)
Center blank key	033 121 (ESC Q)
Right blank key	033 122 (ESC R)
Up-arrow key	033 101 (ESC A)
Down-arrow key	033 102 (ESC B)
Right-arrow key	033 103 (ESC C)
Left-arrow key	033 104 (ESC D)

None of the keys on the auxiliary keypad are affected by pressing the SHIFT, CAPS LOCK, or CTRL keys.

The REPEAT key is used in conjunction with other keys. It does not transmit a code. Any key that transmits a code to the processor will transmit that code repeatedly if pressed while the REPEAT key is down.

The SCROLL key also performs a local function. It is used to request more data from the processor when the keyboard/video display is in the hold screen mode. Typing SCROLL will add one line of characters to the display screen. Typing SHIFT SCROLL will add a screenful of characters.

The COPY key is employed only when using the optional copier peripheral device.

Table 25 lists the function keys, their code, and the keyboard/video display action taken if the code is echoed back to the keyboard/video display.

Display

The display portion of the keyboard/video display has no programmable features. Refer to the *DECstation Technical Manual* for a detailed functional desciption of the display portion of the keyboard/video display.

Basic Cursor Movement Commands

Line Feed (LF - 012) - The LF command moves the cursor down one character position to the same column of the line below. If the cursor was on the bottom line of the screen to begin with, it will remain where it was, but all of the information on the screen will move up one line. The information previously on the top line will be lost from the screen and a new bottom line will appear. This process is referred to as an upward scroll.

Table 25 Function Key Codes

Key	Code Sent	Action
RETURN	015	Carriage Return function.
LINE FEED	012	Line Feed function.
BACK SPACE	010	Backspace (Cursor Left) function.
TAB	011	Tab function.
Space bar	040	Deposit a space on the screen, erasing what was there before.
DELETE	177	Nothing.
ESC (SEL)	033	Interpret the next character from the processor as a command, rather than displaying it.

Cursor Down (ESC B - 033 102) - This command moves the cursor in the same manner as the LF command. If the cursor was on the bottom line of the screen to begin with, it will remain where it was, and no scroll will occur.

Reverse Line Feed (ESC 1 - 033 111) - The reverse line feed command moves the cursor up one character position to the same column of the line above. If the cursor was on the top line to begin with, it will remain where it was, but all of the information on the screen will move down one line. The information that was previously on the bottom line of the screen will be lost and a new blank line will appear at the top line. This process is referred to as a downward scroll.

Cursor Up (ESC A - 033 101) - The cursor up command moves the cursor up one character position to the same column of the line above. If the cursor was on the top line to begin with, it will remain where it was, and no scroll will occur.

Space (040) - The space command erases the character at the cursor position and the cursor moves one column to the right. If the cursor was at the end of the line to begin with, it will not move.

Space can be viewed as a command to erase one character or simply as a displayable character.

Programs written for teleprinters that position the carriage (or cursor) by returning it to the left of the line and spacing it over to the desired column may have to be rewritten. The cursor right command should be used instead of space to move the cursor to a certain column, over data that has already been written on the screen.

Cursor Right (ESC C - 033 103) - This command causes the cursor to move one column to the right. If the cursor was at the end of the line to begin with, it will not move. No character on the screen will be erased.

Cursor Left or Backspace (BS 010) or (ESC D - 033 104) - This command moves the cursor one column to the left. If the cursor was at the start of the line to begin with, it will not move. No character on the screen will be erased.

NOTE

Even though the keyboard/video display responds to the backspace code it is impossible to produce a composite character on the screen by backspacing and overprinting one character on another. If this is attempted, the overprinted character will vanish from the screen because only one character can occupy a character position at any time.

Advanced Cursor Movement Commands

Carriage Return (CR 015) - The carriage return command moves the cursor to the start (leftmost column) of the line it was in. If it was there to begin with, it will remain there.

Cursor Home (ESC H - 033 110) - This command moves the cursor to the home position (the character position at the upper left corner of the screen). If the cursor was there to begin with, it will remain there.

TAB (TAB 011) - The tab command moves the cursor to the right until it reaches a horizontal tab stop. The cursor remains in the same line it was in. If columns are numbered from 1 (leftmost column) to 80 (rightmost column), the TAB stops will be fixed in columns 9, 17, 25, 33, 41, 49, 57, 65, and 73. If the cursor was at a tab stop to begin with, it will move right to the next tab stop. If the cursor was in columns 73–79, it will move right one column. If the cursor was in column 80, it will not move.

Direct Cursor Addressing (ESC Y - 033 131) - The next code after ESC Y that the processor sends to the keyboard/video display will not be displayed but will be interpreted as specifying one of the lines on the screen. The character the keyboard/video display receives after that will not be displayed but will be interpreted as specifying one of the columns on the screen. The cursor will be moved to the character position at the specified line and column. The complete direct cursor addressing command has the form ESC Y line # column # and consists of four characters from the processor. Control codes or other escape sequences should not be embedded in this string of four characters. Doing so will produce unspecified results.

For line #, the processor sends octal code 040 to specify the top line of the screen, 041 to specify the line below the top line, and so forth. Octal code 067 specifies the bottom line. If line # does not specify a line that exists on the screen, the keyboard/video display will not move the cursor vertically if the vertical parameter is out of bounds. A direct cursor addressing command with the first parameter greater than 067 can be issued to the keyboard/video display to move the cursor arbitrarily in the horizontal direction without the flickering of the video that the direct cursor addressing command can cause.

For column #, the processor sends octal code 040 to specify the leftmost column in a line and 157 to specify the rightmost column. If column # is greater than 157 and does not specify a column that exists on the screen, the cursor will be moved to the rightmost column.

Screen Erasure Commands

Erase To End-Of-Line ((ESC K - 033 113) - This command erases all of the information at the cursor position and rightward to the end of the line. Spaces are deposited at those character positions. If the cursor is at the rightmost column of a line, the character at the cursor position will be the only character to be erased. If the cursor is at the leftmost column of a line, the entire line will be erased.

Erase To End-Of-Screen (ESC J - 033 112) - This command erases all of the information from the cursor position to the end of the screen. The erase to end-of-screen command does what the erase to end-of-line command does and also erases the information in every line below the cursor. If the cursor is at the lower right corner of the screen, one character will be erased. If the cursor is at the home position of the screen, all of the information on the screen will be erased.

In addition to the screen erasure commands, the space command can be thought of as a single-character screen erasure command.

Hold-Screen Mode of Operation

The hold-screen mode of operation allows the operator to control the rate at which data enters and leaves the screen. This is important because the keyboard/video display can operate at such fast speeds that data from the processor might remain on the screen for only a few seconds before it scrolls up and off the top of the screen to make way for new data.

Whenever the keyboard/video display cannot process data from the processor, it automatically transmits control code XOFF (023). When it is ready again it transmits XON (021). The keyboard/video display depends on the processor to suspend its transmission promptly when the processor receives XOFF from the keyboard/video display and resumes transmission where it left off upon receiving XON. When software places the keyboard/video display in the hold-screen mode of operation, the keyboard/video display will refuse to perform scrolls. If the processor commands the display to scroll by sending the keyboard/video display a LF (012) when the cursor is on the bottom line, the keyboard/video display will place the LF in a silo (first in, first out memory) to be executed later, and send XOFF to the processor. XOFF means the keyboard/video display is not ready for more data from the processor because the keyboard/video display assumes that the operator is not ready for more.

The SCROLL key is pressed when the operator wants to see more data. The keyboard/video display then processes the LF character out of the silo. When this happens a scroll occurs. The keyboard/video display then takes any other characters from the silo that may have arrived from the processor before the processor responded to the XOFF and suspended its output. Each character in the silo is displayed on the screen or, in the case of commands, executed exactly as if it had just been received - unless it is another LF causing another scroll. If the keyboard/video display encounters an LF in the silo, it will stop processing characters out of the silo until the operator types the SCROLL key again.

If the keyboard/video display processes all the characters in the silo without finding an LF, it will transmit XON to the processor to notify it that the keyboard/video display is again ready to receive characters. It will display all the characters and execute all the commands until it is again ordered to perform a scroll. Then it will send XOFF, store the LF in the silo, and wait for the operator to press the SCROLL key again.

If the processor ignores the XOFF signal, it might completely fill the silo. Then, rather than allow data to be lost, the keyboard/video display will perform the scroll it was commanded to perform despite the hold-screen mode. It will remove the characters from the silo and interpret them, reducing the backlog. However, the keyboard/video display will not exit hold-screen mode.

The SCROLL key is typed to request that another line be admitted to the screen. The keyboard/video display translates this request into start and stop commands (XON and XOFF) and sends them to the processor in such a manner that just enough data comes to the keyboard/video display to satisfy the operator's request for one more line.

The operator can type the SCROLL key with the SHIFT key down to request a new screenful of data. As with the unshifted SCROLL request, the keyboard/video display begins to process characters again and sends XON to the processor when the characters that accumulated in the silo have all been

processed. But the shifted scroll request tells the keyboard/video display to allow an entire screenful of new data to enter the screen before shutting off the transmission from the processor.

Enter Hold-Screen Mode (ESC - 033 133) - When the keyboard/video display enters hold-screen mode, data will not be allowed to scroll off the screen without permission from the operator by use of the SCROLL key. After entering the hold-screen mode, the first command that would cause a scroll to occur will not be processed immediately, and the keyboard/video display will send XOFF to the processor.

The hold-screen mode will remain in effect until the exit hold-screen mode command disables that feature.

Exit Hold-Screen Mode (ESC / - 033 134) - When the keyboard/video display exits the hold-screen mode, data will be allowed to scroll off the screen if it has to to make room for new data coming from the processor.

Alternate-Keypad Mode of Operation

The auxiliary keypad is capable of transmitting the codes listed in Table 23. There are two sets of codes that the software can select by issuing these commands - enter alternate-keypad mode and exit alternate-keypad mode.

Enter Alternate-Keypad Mode (ESC = - 033 075) - The alternate-keypad mode enables the numeric keys, decimal point key, and ENTER key to transmit unique escape sequences, allowing software to distinguish between them and keys on the main keyboard, and to assign its own meaning to each key.

The alternate-keypad mode will not be in effect until the processor issues this command. Once enabled, it will remain in effect until the processor uses the exit alternate-keypad mode instruction.

Exit Alternate-Keypad Mode (ESC) - 033 076) - When the exit alternate-keypad mode is initiated the numeral, decimal point, and ENTER keys transmit codes are indistinguishable from the codes transmitted by the numeric, decimal point, and RETURN keys on the main keyboard. Applications that do not need to redefine the meanings of these twelve keys will work correctly allowing the operator to use the keypad for entry of numeric data.

Graphics Mode of Operation

There are 33 special symbols that can be displayed on the screen of the keyboard/video display. These symbols can be entered on the screen only by placing the keyboard/video display in the graphics mode. Normally, codes 136-176 stand for lowercase letters and symbols. In the graphics mode, each code in this range will specify that one of the special symbols be placed on the screen. Table 26 describes the appearance of the keyboard/video display special symbols that are displayed on the screen in the graphics mode.

Codes 040-135 are unaffected. The symbols represented can be placed on the screen whether or not the keyboard/video display is in the graphics mode.

The keyboard/video display uses the control codes to mark the position of the special symbols in its memory. Therefore, the special symbols and the lowercase letters can coexist on the screen. The special symbols will remain on the screen where they were entered even if the keyboard/video display is taken out of the graphics mode.

Enter Graphics Mode (ESC F - 033 106) - This command causes codes 136-176 to be converted to special symbols before being placed on the screen. This remains true until the keyboard/video display receives the exit graphics mode command.

Exit Graphics Mode (ESC G - 033 107) - The exit graphics mode signal causes codes 136-176 to resume their standard (ASCII) meanings.

Table 26 Graphic Mode Symbols

When This Code Received	The Screen Will Display	
	In Graphics Mode	Not In Graphics Mode
136	blank	^
137	blank	_
140	reserved	`
141	solid rectangle	a
142	1/	b
143	3/	c
144	5/	d
145	7/	e
146	degrees	f
147	plus or minus	g
150	right arrow	h
151	ellipsis	i
152	divide by	j
153	down arrow	k
154	bar at scan 0	l
155	bar at scan 1	m
156	bar at scan 2	n
157	bar at scan 3	o
160	bar at scan 4	p
161	bar at scan 5	q
162	bar at scan 6	r
163	bar at scan 7	s
164	subscript 0	t
165	subscript 1	u
166	subscript 2	v
167	subscript 3	w
170	subscript 4	x
171	subscript 5	y
172	subscript 6	z
173	subscript 7	{
174	subscript 8	|
175	subscript 9	}
176	paragraph	~

Uses of Special Symbols

Codes 154–163 – These codes cause eight horizontal lines at various scans within the character positon to be displayed. These bars can be used to print a bargraph on the screen with more accuracy than would be possible using only minus signs and underlines.

Codes 142–154 – (1/, 3/, 5/, and 7/) are used preceding the subscripts to form fractions. In particular, the fractions 1/8, 1/4, 3/8, 1/2, 5/8, 3/4, and 7/8 can be formed using these four symbols and the subscripts.

Code 151 – The ellipsis appears as three dots in the character position (...). The spacing of these three dots is such that several of these symbols placed adjacent to one another in the screen will produce a smooth line of dots.

Codes 136–140 – These codes are reserved for future use.

INDEX

VT78 PROGRAMMER'S REFERENCE MANUAL
EK-OVT78-RM-001

Reader's Comments

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgement is it complete, accurate, well organized, well written, etc? Is it easy to use? _____

What features are most useful? _____

What faults or errors have you found in the manual? _____

Does this manual satisfy the need you think it was intended to satisfy? _____

Does it satisfy *your* needs? _____ Why? _____

☐ Please send me the current copy of the *Technical Documentation Catalog*, which contains information on the remainder of DIGITAL's technical documentation.

Name _____ Street _____
Title _____ City _____
Company _____ State/Country _____
Department _____ Zip _____

Additional copies of this document are available from:

Digital Equipment Corporation
444 Whitney Street
Northboro, MA 01532

Attention: Communications Services (NR2/M15)
Customer Services Section

Order No. **EK-OVT78-RM-001**

Fold Here

Do Not Tear – Fold Here and Staple

digital

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MERRIMACK, NH

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Educational Services Development & Publishing
Continental Blvd. (MK1/2M26)
Merrimack, N.H. 03054