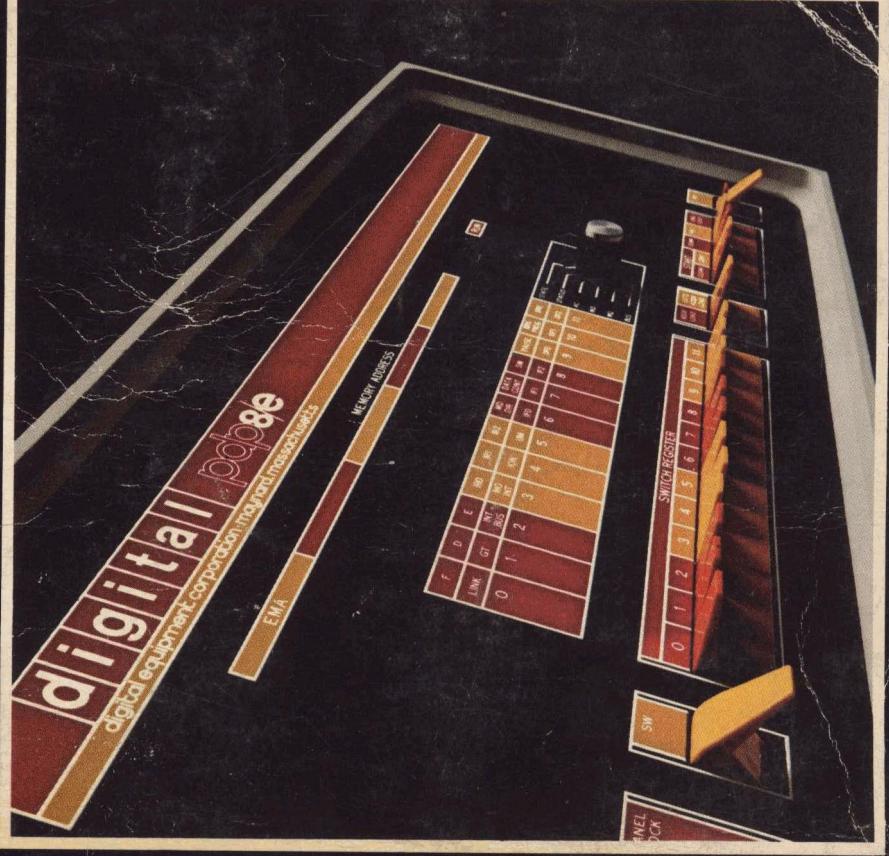


# paper & paper

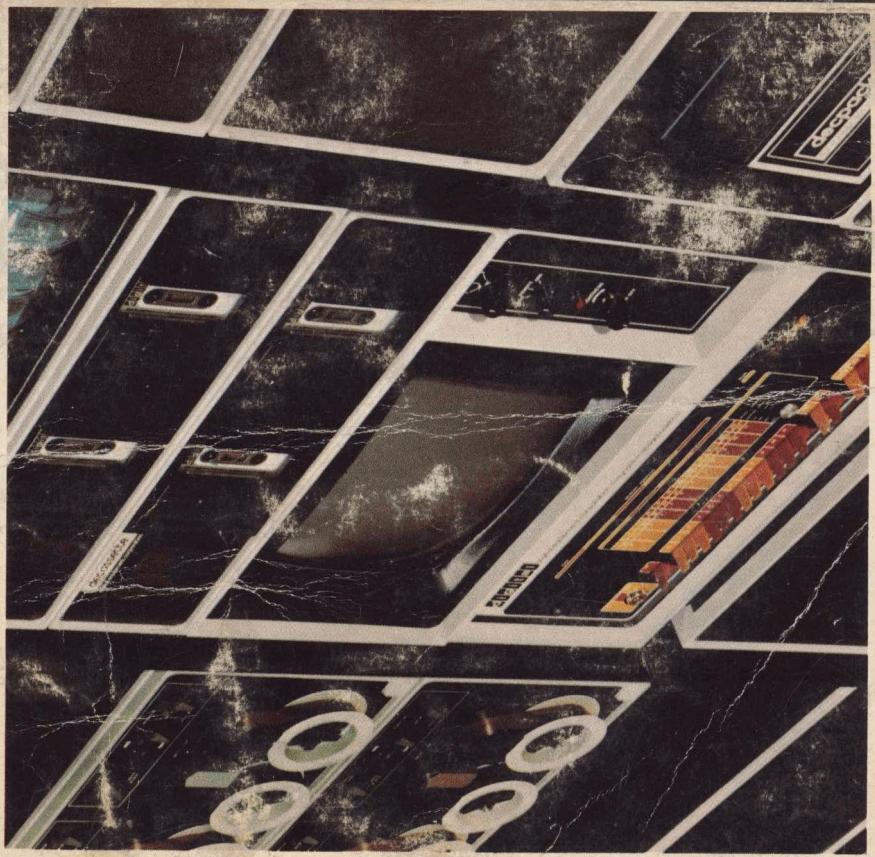
# small computer handbook



**digital** small computer handbook 1973

**digital**

DIGITAL EQUIPMENT CORPORATION, Maynard, Massachusetts Telephone: (617) 897-5111  
 • ARIZONA: Phoenix • CALIFORNIA: Sunnyvale Santa Ana Los Angeles Oakland San Diego and San Francisco (Mountain View) • COLORADO: Denver • CONNECTICUT: Meriden • DISTRICT OF COLUMBIA: Washington (Riverdale, Md.) • FLORIDA: Orlando • GEORGIA: Atlanta • ILLINOIS: Chicago • INDIANA: Indianapolis • LOUISIANA: New Orleans • MASSACHUSETTS: Cambridge and Waltham • MICHIGAN: Ann Arbor and Detroit • MINNESOTA: Minneapolis • MISSOURI: St. Louis • NEW JERSEY: Englewood Metuchen Parsippany and Princeton • NEW YORK: Centerreach (L.I.) Manhattan Syracuse and Rochester • NORTH CAROLINA: Durham Chapel Hill • OHIO: Cleveland and Dayton • OKLAHOMA: Tulsa • OREGON: Portland • PENNSYLVANIA: Philadelphia and Pittsburgh • TENNESSEE: Knoxville • TEXAS: Dallas and Houston • UTAH: Salt Lake City • WASHINGTON: Seattle • WISCONSIN: Milwaukee • ARGENTINA: Buenos Aires • AUSTRALIA: Adelaide Brisbane Melbourne Perth and Sydney • AUSTRIA: Vienna • BELGIUM: Brussels • BRAZIL: Rio de Janeiro São Paulo and Porto Alegre • CANADA: Calgary Alberta Vancouver British Columbia Ottawa and Toronto Ontario and Montreal Quebec • CHILE: Santiago • DENMARK: Copenhagen • FRANCE: Grenoble and Paris • GERMANY: Cologne Hannover Frankfurt Munich and Stuttgart • INDIA: Bombay • ITALY: Milan • JAPAN: Tokyo • MEXICO: Mexico City • NETHERLANDS: The Hague • NEW ZEALAND: Auckland • NORWAY: Oslo • PHILIPPINES: Manila • PUERTO RICO: Miramar • SPAIN: Barcelona and Madrid • SWEDEN: Stockholm • SWITZERLAND: Geneva and Zurich • UNITED KINGDOM: Birmingham Edinburgh London Manchester and Reading • VENEZUELA: Caracas



Printed in U.S.A.

018-00173 • 2546  
S 09-100

**digital**

**pdp8/e  
pdp8/m & pdp8/f  
small computer  
handbook**

**prepared  
by  
software documentation  
programming department  
digital equipment corporation**

**pdp-8 handbook series**

Copyright © 1970, 1971, 1972, 1973  
Digital Equipment Corporation

The following are registered trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

CDP	DIGITAL	LAB-8/e	RAD-8
Computer Lab	DNC	OMNIBUS	RSTS
Comtex	Flip Chip	OS/8	RSX
DEC	FOCAL	PDP	RTM
DECtape	IDAC	PHA	SABR
Dibol	Indac	PS/8	Typeset 8
	KA10	Quickpoint	Unibus

# introduction

Minicomputers, from Digital Equipment Corporation, are changing your world—in banks and hospitals, supermarkets and factories. Everywhere people are realizing that computers don't have to be large and expensive to get the job done. A Computer is no longer a multi-million dollar giant that can only survive in spotlessly clean rooms. Minicomputers are going where the job is, because they are rugged, dependable, and inexpensive.

You should know about minicomputers. The PDP-8/E Story shows our computers at work; designing, producing and testing new computers, saving time and money. Other industries, such as oil refineries and automobile manufacturers, are also using the power and speed of computers to produce better products. Minicomputers are not just for big business; hospitals, schools, laboratories and factories are using minis just as effectively. New and old companies are exploring minicomputers.

How large a computer should you buy? Most enterprises begin small. After the computer requirements are completely defined, a decision is then made to either continue with the existing system or to expand. The basic PDP-8/E can be expanded without having to sacrifice your initial investment.

Right now, there are more than 30,000 minicomputers serving in almost every field of endeavor and embracing every discipline known to man. The PDP-8/E and PDP-8/M are DEC's newest models of the PDP-8 family. We invite you to explore the advantages of owning this small machine with big ideas.

This edition of the *Small Computer Handbook* presents all the information you will want to help verify that a PDP-8/E computer system provides the optimum solution to your data processing problem. Chapter 2 is a PDP-8/E system introduction that describes the unique internal design and packaging scheme of the PDP-8/E and presents a representative selection of supported software products. Chapters 3 and 4 are a definitive machine language programming reference that affords complete instructions for programming the processor and performing data input/output. Chapters 5, 6 and 7 describe the wide variety of PDP-8/E peripherals and options. Chapter 8 presents information and suggestions intended to facilitate site preparation and installation planning, while Chapters 9 and 10 offer detailed technical information on interfacing and logic design to assist the digital engineer in developing custom hardware for new applications.

## THE COMPANY

In a little over thirteen years, Digital has become a major force in the electronics industry. The company has grown from three employees and 8,500 square feet of production floor space in a converted woolen mill in Maynard, Massachusetts, to an international corporation employing more than 11,000 people with well over two million square feet of floor space in more than 60 manufacturing, sales, and service facilities around the world. In addition to the corporate headquarters in Maynard, Massachusetts, other manufacturing facilities are located in Westfield and Westminster, Massachusetts. Internationally and outside the continental United States, Digital has manufacturing plants in England, Canada and Puerto Rico.

From its beginnings as a manufacturer of digital modules, the company has now grown to the point where it is the world's largest manufacturing supplier of logic modules and the third largest computer-manufacturer, by number of installations, in the industry. Digital's rise as a leader in the electronics industry began in 1957 with the introduction of the company's line of electronic circuit modules. These solid-state modules were used to build and test other manufacturers' computers. Two years later, Digital introduced its first computer, the PDP-1. The PDP-1 heralded a new concept for the industry—the small, on-line computer. And the PDP-1 was inexpensive—it sold for \$120,000 while competitive machines with similar capabilities were selling at over \$1 million. But the PDP-1 was more than a data processor; more than just a tool to manipulate data. It was a system that could be connected to all types of instrumentation and equipment for on-line, real-time monitoring control, and analysis. It was a system with which people and machines could interact.

Also, in 1958, Digital introduced the Systems Modules, high-quality, low-cost, solid-state, digital logic circuits on a single printed circuit card.

Today, electronic modules like the ones Digital introduced are used in most electronic equipment, from computers to television sets.

In 1965, Digital announced the first of the FLIP CHIP® module lines. These highly reliable modules include cards for internal computer logic, interfacing, control and analog-to-digital conversion.

In 1963, Digital Equipment Corporation introduced the PDP-5 computer, predecessor of the PDP-8 series. This was followed by the first PDP-8 in 1964, and since that time DEC has added the PDP-8/S, the PDP-8/I, and PDP-8/L. Over this seven year period, considerable improvement has been made, many options have been developed, over 60 peripherals and a variety of programs developed. As each new application need arises, Digital Equipment engineering responds with new equipment; each time further increasing the capability of the PDP-8 Family and making available a wider range of equipment.

Throughout the life span of the PDP-8 Family, DEC has developed more than 1,000 programs for a wide variety of applications. New programs are constantly in development by Digital's Programming Department and the PDP-8 Users. This means that each PDP-8/E user will have a wide variety of programs immediately available to him.

# additional information

This *Small Computer Handbook* is one volume of a large library of technical information available free or at nominal cost to assist engineers, technicians, programmers and managers in developing and maintaining a minicomputer installation.

The *PDP-8/E Maintenance Manual* is a detailed hardware reference that relates each PDP-8/E instruction to the hardware circuitry that executes the operation. The manual is published in three volumes to show a clear distinction between the basic system circuitry described in Volume 1 (DEC-8E-HR1B-D), internal bus options described in Volume 2 (DEC-8E-HR2B-D) and external bus options described in Volume 3 (DEC-8E-HR3B-D). All three volumes include extensive illustrations from the PDP-8/E print set (available separately) as well as flowcharts, timing diagrams and full functional descriptions of the various logic circuits. The Maintenance Manual is an indispensable reference for design engineers and programmers who require knowledge of the system architecture or detailed hardware information.

*Introduction to Programming* is a companion document to the *Small Computer Handbook* that serves as a textbook covering most areas of assembly language programming. The 1973 edition of *Introduction to Programming*, available in September, includes revised background material on number systems, binary/octal arithmetic, floating-point and ASCII operations, logical processes and flowcharting. Subsequent Chapters introduce PDP-8/E processor, EAE and extended memory instructions while presenting techniques for implementing common data processing operations in PAL assembly language. *Introduction to Programming* is suitable for use as a classroom teaching aid or a self-study text; it assumes no background in computer science.

More than 60 input/output devices may be interfaced with any PDP-8/E. This handbook contains brief descriptions of most I/O and processor options, nearly all of which may be supplied with the requisite supporting software. Users who intend to program their own device handlers or modify the software drivers for supported options should consult the appropriate device maintenance manual. Maintenance manuals afford complete programming instructions as well as hardware descriptions and full operating instructions. They are updated frequently to reflect engineering changes and incorporate the experience of PDP-8/E users in the field. Option bulletins and brochures, available without charge from local DEC sales offices, provide another excellent source of information. Most option bulletins contain concise, up-to-date specifications and a handy instruction summary, along with a description of important manual controls and indicators. Print sets and engineering specifications are also available for most device options.

#### **ERROR REPORTING**

If you find any errors in this handbook, or if you have any questions or comments concerning the clarity or completeness of this handbook, please direct your remarks to:

Digital Equipment Corporation  
Software Information Service, Building 3-4  
Maynard, Massachusetts 01754

# contents

## CHAPTER 1 PDP-8/E STORY

## CHAPTER 2 SYSTEM INTRODUCTION

PDP-8/E BASIC SYSTEM .....	2-1
KK8-E CENTRAL PROCESSING UNIT .....	2-3
M3800 MAJOR REGISTERS MODULE .....	2-3
Accumulator (AC) .....	2-3
Multiplier Quotient Register (MQ) .....	2-5
Program Counter (PC) .....	2-5
Central Processor Memory Address Register (CPMA) .....	2-5
Memory Buffer Register (MB) .....	2-5
Data Gates and Adders .....	2-5
M8310 MAJOR REGISTERS CONTROL MODULE .....	2-6
Link .....	2-6
Major Register Control Circuits .....	2-6
Major State Generator .....	2-6
Instruction Register (IR) .....	2-6
M8320 BUS LOADS MODULE .....	2-6
M8330 TIMING GENERATOR MODULE .....	2-6
M849 RFI SHIELD MODULE .....	2-7
KC8-EA PROGRAMMERS CONSOLE .....	2-7
M8650 ASYNCHRONOUS DATA CONTROL MODULE .....	2-7
MEMORY SYSTEM .....	2-8
G227 XY Driver and Current Source Module .....	2-8
G619 Planar Stack Board .....	2-8
G104 Sense/Inhibit Module .....	2-8
INTERFACING .....	2-9
DIFFERENCES BETWEEN THE PDP-8/E AND ITS PREDECESSORS .....	2-10
PDP-8/F AND PDP-8/M .....	2-12

## CHAPTER 3 SYSTEM OPERATION

PROGRAMMER'S CONSOLE OPERATION .....	3-1
MEMORY ORGANIZATION .....	3-5

<b>MEMORY AND PROCESSOR INSTRUCTIONS .....</b>	<b>3-7</b>
Memory Reference Instructions .....	3-8
The Housekeeping Instruction .....	3-9
<b>AUGMENTED INSTRUCTIONS .....</b>	<b>3-10</b>
The Input/Output Transfer Instruction .....	3-10
The Operate Instruction.....	3-11
<b>GROUP 1 MICROINSTRUCTIONS .....</b>	<b>3-11</b>
<b>GROUP 2 MICROINSTRUCTIONS .....</b>	<b>3-13</b>
<b>GROUP 3 MICROINSTRUCTIONS .....</b>	<b>3-16</b>
<b>INSTRUCTION EXECUTION AND TIMING .....</b>	<b>3-17</b>
Memory Timing .....	3-19
FETCH Major State .....	3-19
DEFER Major State .....	3-20
EXECUTE Major State .....	3-21

#### **CHAPTER 4 INPUT/OUTPUT OPERATIONS**

<b>PROGRAMMED DATA TRANSFER .....</b>	<b>4-1</b>
Peripheral Device Status .....	4-2
Programmed Data Transfer Timing Constraints .....	4-3
<b>PROGRAM INTERRUPT TRANSFERS .....</b>	<b>4-4</b>
Interrupt System Operation .....	4-4
Interrupt System Hardware Components .....	4-9
<b>DATA BREAK TRANSFERS .....</b>	<b>4-11</b>
Current Address (CA) Register .....	4-11
Word Count (WC) Register .....	4-11
Data Break Priority .....	4-12
Data Break Transfers .....	4-12
Single-Cycle Data Break .....	4-14
Three-Cycle Data Break .....	4-18

#### **CHAPTER 5 PROCESSOR OPTIONS**

<b>MECHANICAL EXPANSION OPTIONS .....</b>	<b>5-1</b>
<b>KE8-E EXTENDED ARITHMETIC ELEMENT .....</b>	<b>5-1</b>
<b>MEMORY EQUIPMENT OPTIONS .....</b>	<b>5-12</b>
<b>REAL TIME CLOCK OPTIONS .....</b>	<b>5-22</b>
<b>KP8-E POWER FAIL DETECT .....</b>	<b>5-29</b>

#### **CHAPTER 6 INTERNAL I/O OPTIONS**

<b>CONSOLE TELEPRINTERS .....</b>	<b>6-1</b>
<b>CRT DISPLAYS .....</b>	<b>6-13</b>

X/Y PLOTTER OPTIONS .....	6-25
LINE PRINTER OPTIONS .....	6-29
DATA COMMUNICATIONS EQUIPMENT OPTIONS .....	6-34
CARD READER OPTIONS .....	6-61
RK8-E DISK CARTRIDGE SYSTEM .....	6-67
OMNIBUS MAGNETIC TAPE OPTIONS .....	6-75
LABORATORY PERIPHERALS .....	6-96
DB8-E INTERPROCESSOR BUFFER .....	6-107

## **CHAPTER 7 EXTERNAL I/O OPTIONS**

EXTERNAL BUS INTERFACE CONTROL OPTIONS .....	7-1
RANDOM ACCESS DISK DEVICES .....	7-5
MAGNETIC TAPE OPTIONS .....	7-18
DATA ACQUISITION PERIPHERALS .....	7-39
VW01 WRITING TABLET .....	7-57
FPP-12 FLOATING POINT PROCESSOR .....	7-65
DATA ENTRY TERMINALS .....	7-77
DW08-A I/O CONVERSION PANEL .....	7-80

## **CHAPTER 8 INSTALLATION PLANNING AND DELIVERY**

PRE-DELIVERY PLANNING AND SITE PREPARATION .....	8-1
Processor Options .....	8-3
Cabinet Options .....	8-5
I/O Cabling Requirements .....	8-8
TEMPERATURE AND HUMIDITY .....	8-9
FACTORS AFFECTING SYSTEM RELIABILITY .....	8-11
EXTENDED OPERATION UNDER EXTREME CONDITIONS .....	8-11
ELECTRICAL CONSIDERATIONS .....	8-13
General Power Requirements .....	8-14
Power Failure .....	8-18
Ground Requirements .....	8-18
AC POWER FACILITY INSTALLATION AND TESTING .....	8-18
Installation Procedure for PDP-8/M and Teletype .....	8-20
Equipment Support Services .....	8-27
Maintenance and Service Options .....	8-27
Service Agreement .....	8-27

Service Contract Coverage .....	8-28
Eligibility for Service Contract Coverage .....	8-28
Depot Repair .....	8-28
<b>CHAPTER 9 THE OMNIBUS INTERFACING</b>	
INTRODUCTION .....	9-1
SECTION 1 OMNIBUS DESCRIPTION .....	9-3
BUS STRUCTURE .....	9-3
BUS SPECIFICATIONS .....	9-4
SYSTEMS CONFIGURATION .....	9-4
RELATIONSHIP OF THE EXTERNAL BUS TO THE OMNIBUS .....	9-6
OMNIBUS SIGNALS .....	9-6
SECTION 2 HOW TO CHOOSE THE TYPE OF I/O TRANSFER .....	9-29
DATA TRANSFER TYPES .....	9-29
INTERFACING TO THE PROCESSOR .....	9-29
DATA TRANSFER RATES .....	9-32
DEVICE CODES .....	9-32
<b>SECTION 3 DESIGNING BASIC PROGRAMMED I/O INTERFACE</b>	
CONTROL CIRCUITS .....	9-34
DEVICE SELECTION CIRCUIT .....	9-34
OPERATIONS DECODER .....	9-34
FLAG LOGIC .....	9-36
INTERRUPT REQUEST .....	9-36
OUTPUT BUFFER .....	9-36
INPUT BUFFER .....	9-36
I/O CONTROL .....	9-39
INPUT/OUTPUT TIMING FOR PROGRAMMED I/O INTERFACES .....	9-39
EXTENDED I/O.....	9-39
SECTION 4 DESIGNING A BASIC DATA BREAK INTERFACE .....	9-41
BREAK ADDRESS .....	9-42
DATA PATHS .....	9-42
STATUS REGISTER .....	9-42
BREAK PRIORITIES .....	9-42
TRANSFER DIRECTION AND LOADING LOGIC .....	9-42

DATA BREAK INTERNAL LOGIC AND TIMING .....	9-42
BASIC ONE-CYCLE DATA BREAK INTERFACE .....	9-43
TIMING FOR SAMPLE DATA BREAK INTERFACE .....	9-47
THREE-CYCLE DATA BREAKS .....	9-47
DESIGN CHECK LIST FOR SINGLE CYCLE DATA BREAK INTERFACE .....	9-51
SECTION 5 GENERAL DESIGN AND CONSTRUCTION GUIDELINES .....	9-52
INTERFACE DESIGN OPTIONS .....	9-52
ETCHED CIRCUIT LAYOUT AND CONSTRUCTION RULES .....	9-52
GENERAL CABLE RULES AND SUGGESTIONS .....	9-54
DEC SUPPLIED INTERFACE CABLES .....	9-54
CABLING RULES .....	9-54
INTERFACE TIMING CRITERIA .....	9-56
General Timing Rules .....	9-56
Interrupt Timing .....	9-56
Timing Example .....	9-56
Timing Requirements For Data Break Facilities .....	9-58
Timing and Break Priorities .....	9-58
GENERAL PROPAGATION DELAY GUIDELINES .....	9-58
2-input NAND Gate Delay .....	9-59
Flip-Flop Propagation Delays .....	9-59
J-K Flip-Flops .....	9-60
One-Shot Delays .....	9-60
MAXIMUM OPERATING FREQUENCY .....	9-60
LOADING RULES .....	9-62
Device Selection Inputs .....	9-63
Skip and Interrupt Request Lines .....	9-63
Electrical Considerations of Driving a Line .....	9-63
GROUNDED .....	9-63
TESTING TECHNIQUES .....	9-63
Initial Checkout .....	9-63
System Test .....	9-63
Final Testing .....	9-64
PROGRAMMING RULES .....	9-64
DESIGN CHECKLIST .....	9-64
SECTION 6 PDP-8/E INTERFACE HARDWARE .....	9-66
W966 & W967 MOUNTING BOARDS .....	9-67
H851 EDGE CONNECTOR .....	9-68

MODULE HOLDERS .....	9-68
30-GAUGE INSULATED WIRE .....	9-68
<b>CHAPTER 10 I/O EXPANSION TECHNIQUES</b>	
SECTION 1 POSITIVE I/O BUS INTERFACING TECHNIQUES .....	10-1
THE NATURE OF THE EXTERNAL BUS .....	10-2
EXTERNAL BUS SIGNALS .....	10-3
APPLICATION .....	10-8
Programmed I/O Transfer .....	10-8
Program Interrupt Transfers .....	10-13
Data Break Transfers .....	10-13
INTERFACING TECHNIQUES .....	10-17
PDP-8/I and 8L-Type Peripherals .....	10-18
Customer Peripherals .....	10-20
DEC Logic Module Interfacing .....	10-20
Customer Designed Interfaces .....	10-28
Restrictions and Criteria .....	10-30
Cooling .....	
Signal Terminating .....	
Timing Criteria .....	
CABLING RULES AND SUGGESTIONS .....	10-31
SECTION 2 OMNIBUS INTERFACING USING "OFF THE SHELF" MODULES .....	10-34
OMNIBUS SIGNAL SUMMARY .....	10-34
THE BUILDING BLOCK APPROACH .....	10-34
M783 BUS DRIVERS .....	10-34
M784 BUS RECEIVERS .....	10-36
M785 BUS TRANSCEIVER .....	10-37
H9190 M935 KIT .....	10-38
H803 CONNECTOR BLOCK .....	10-38
M935 BUS CONNECTOR .....	10-39
H9190 MOUNTING PANEL .....	10-39
H019 MOUNTING BAR .....	10-40
INTERFACE EXAMPLE PAPER TAPE READER .....	10-42
INPUT/OUTPUT TRANSFER (IOT) INSTRUCTION USAGE .....	10-42
SYSTEM OPERATION .....	10-42
M-SERIES MODULE SUMMARY .....	10-43
K-SERIES MODULE SUMMARY .....	10-50

# List of illustrations

Figure	Page
2-1 PDP-8/E Basic System Block Diagram .....	2-4
3-1 PDP-8/E Programmer's Console .....	3-1
3-2 PDP-8/E Memory Organization .....	3-6
3-3 Memory Reference Instruction Format .....	3-8
3-4 Jump Instruction Format .....	3-10
3-5 IOT Instruction Format .....	3-10
3-6 Group 1 Operate Microinstructions .....	3-11
3-7 Group 2 Operate Microinstructions .....	3-13
3-8 Group 3 Operate Microinstructions .....	3-16
3-9 Major State Flow Diagram .....	3-18
3-10 PDP-8/E Memory Timing Diagram .....	3-19
3-11 FETCH Major State .....	3-20
3-12 DEFER Major State .....	3-21
3-13 EXECUTE Major State .....	3-22
4-1 Program to Print ASCII Characters .....	4-4
4-2 Program Interrupt Flow of Events .....	4-6
4-3 Simplified Interrupt Service Routine .....	4-10
4-4 Single-Cycle Data Break Simplified Block Diagram .....	4-15
4-5 Single-Cycle Data Break Flow Chart .....	4-16
4-6 Three-Cycle Data Break Simplified Block Diagram .....	4-19
4-7 Three-Cycle Data Break Flow Chart .....	4-20
5-1 EAE Mode "A" Bit Assignments .....	5-5
5-2 EAE Mode "B" Bit Assignments .....	5-7
8-1 Table-Top System Space Requirements .....	8-2
8-2 Rack-Mounted System Space Requirements .....	8-3
8-3 PDP-8/F and PDP-8/M Space Requirements .....	8-4
8-4 Standard Basic Cabinet .....	8-5
8-5 Short Basic Cabinet .....	8-6
8-6 PDP-8/E Cabinet Configurations .....	8-7
8-7 Cabinet Floor Space Requirements .....	8-8
8-8 PDP-8/E Cable Layout .....	8-9
9-9 Recommended Operating Environment .....	8-10
8-10 Industrial Computer Enclosure .....	8-13
8-11 ICE Box Dimensional Data .....	8-14
8-12 Recommended Receptacle Wiring .....	8-19
9-1 PDP-8/E OMNIBUS .....	9-2
9-2 PDP-8/E Modules Mounted on the OMNIBUS .....	9-3
9-3 The System Expansion Capability .....	9-4
9-4 PDP-8/E OMNIBUS Configurator .....	9-5
9-5 OMNIBUS Pin Assignments .....	9-7
9-6 OMNIBUS Loads .....	9-8

9-7	NLT Flow Diagram .....	9-15
9-8	OMNIBUS/Basic System Interface .....	9-30
9-9	OMNIBUS/ I/O Signal Interface Block Diagram .....	9-31
9-10	Data Transfer Rates .....	9-32
9-11	Sample Programmed IO INTERFACE Control Logic .....	9-35
9-12	Parallel to Serial Output Buffer .....	9-37
9-13	Serial to Parallel Output Buffer .....	9-38
9-14	Timing for Sample Programmed I/O Interface Control .....	9-40
9-15	Basic Data Break Control Logic .....	9-41
9-16	Sample One-Cycle Data Break Interface .....	9-46
9-17	Data Break Control Timing Diagram .....	9-48
9-18	3-Cycle Data Break Implementation Flow Diagram .....	9-49
9-19	Typical Component Layout .....	9-53
9-20	Interface Cable Pin Assignment .....	9-55
9-21	2 Input NAND Gate Typical Characteristics Example .....	9-59
9-22	Typical D-Type Flip-Flop Timing Example .....	9-59
9-23	J-K Flip-Flop Timing Example .....	9-60
9-24	One-Shot Delay Timing and Logic Symbol Example .....	9-61
9-25	Delays Determining Total Operating Frequency Example .....	9-62
10-1	Parallel Connection of Peripherals .....	10-3
10-2	External Bus Signals and Related OMNIBUS Signals .....	10-4
10-3	Block Diagram, Programmed Data Transfer .....	10-9
10-4	Timing Diagram, Skip Bus Application .....	10-11
10-5	Timing Diagram, Programmed Data Transfer .....	10-12
10-6	Block Diagram, Data Break Transfer .....	10-14
10-7	Timing Diagram, Data Break Transfer .....	10-15
10-8	Connector Pin Identification .....	10-19
10-9	M103 Device Selector Module .....	10-21
10-10	M624 Bus Driver Module .....	10-22
10-11	Bus Driver Application .....	10-23
10-12	M101 Bus Data Interface .....	10-24
10-13	Bus Data Application .....	10-24
10-14	Interface Hardware .....	10-26
10-15	Mounting Panel Wiring .....	10-28
10-16	Interface Logic Diagrams .....	10-29
10-17	Maximum Bus Lengths .....	10-32
10-18	Typical System Interface Block Diagram .....	10-35
10-19	M783 Bus Drivers .....	10-35
10-20	M784 Bus Receivers .....	10-36
10-21	M785 Bus Transceiver .....	10-37

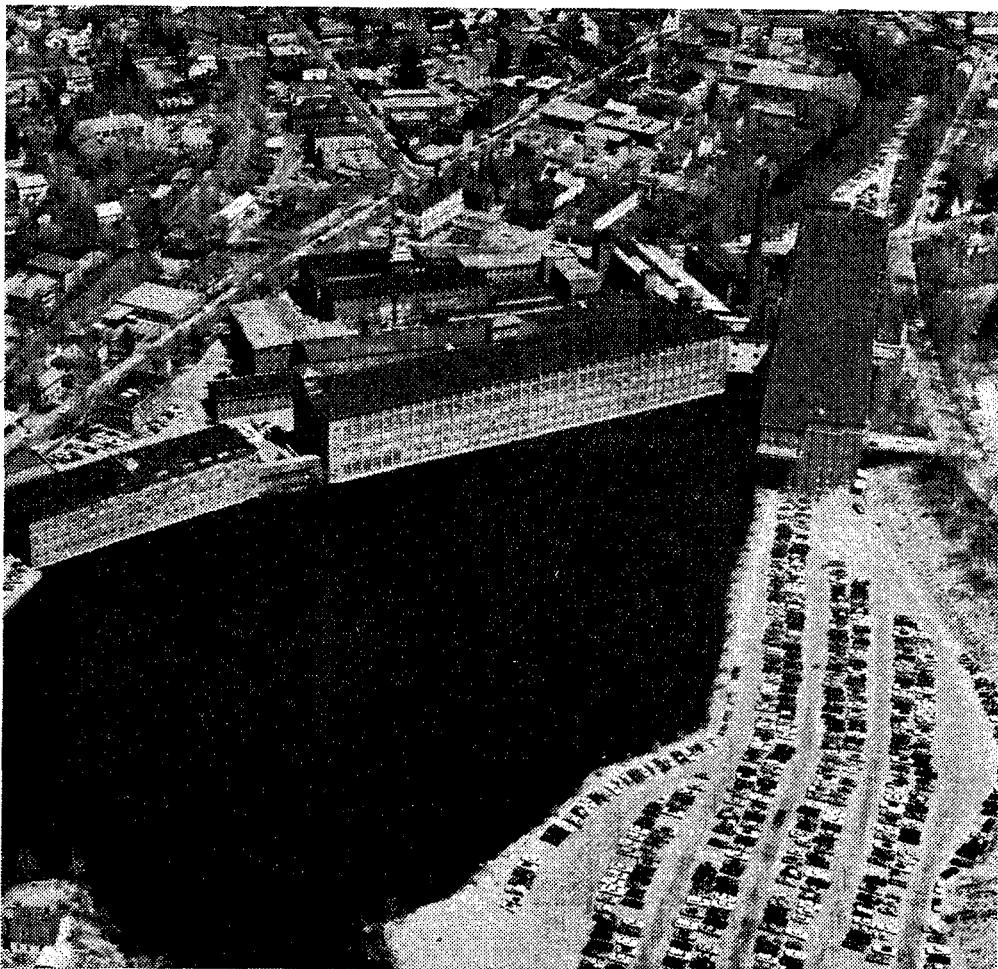
# list of tables

Table		Page
2-1	New PDP-8/E Instructions .....	2-10
2-2	PDP-8/M System Expandability .....	2-12
3-1	Programmer's Console Control and Indicator Functions ..	3-2
3-2	Memory Reference Instructions .....	3-9
3-3	The Housekeeping Instruction .....	3-10
3-4	Group 1 Operate Microinstructions .....	3-12
3-5	Microprogrammed Combinations of Group 1 Microinstructions .....	3-13
3-6	Group 2 Operate Microinstructions .....	3-14
3-7	Microprogrammed Combinations of Group 2 Microinstructions .....	3-15
3-8	Group 3 Operate Microinstructions .....	3-16
4-1	Interrupt IOT Instructions .....	4-7
5-1	EAE Mode Changing Instructions .....	5-4
5-2	KE8-E Mode A Instructions .....	5-6
5-3	KE8-E Mode B Instructions .....	5-8
5-4	Mode A and Mode B Instruction Differences .....	5-11
8-1	System Environmental Specifications .....	8-12
8-2	Installation Data .....	8-15
9-1	Programmed I/O OMNIBUS Signals .....	9-9
9-1a	Table of Transfer Control Signals .....	9-12
9-2	Additional Programmed I/O OMNIBUS Signals .....	9-13
9-3	Data Break OMNIBUS Signals .....	9-16
9-4	Basic System OMNIBUS Data and Control Signals .....	9-19
9-5	PDP-8/E Device Codes .....	9-33
9-6	Operations Decoder IOT's .....	9-34
9-7	Data Break Control Signals .....	9-42
9-8	One-Cycle Data Break Sequence of Events .....	9-43
10-1	Data Control Lines (Data Break Interface) .....	10-17
10-2	External Bus Connector Pin Assignments .....	10-19

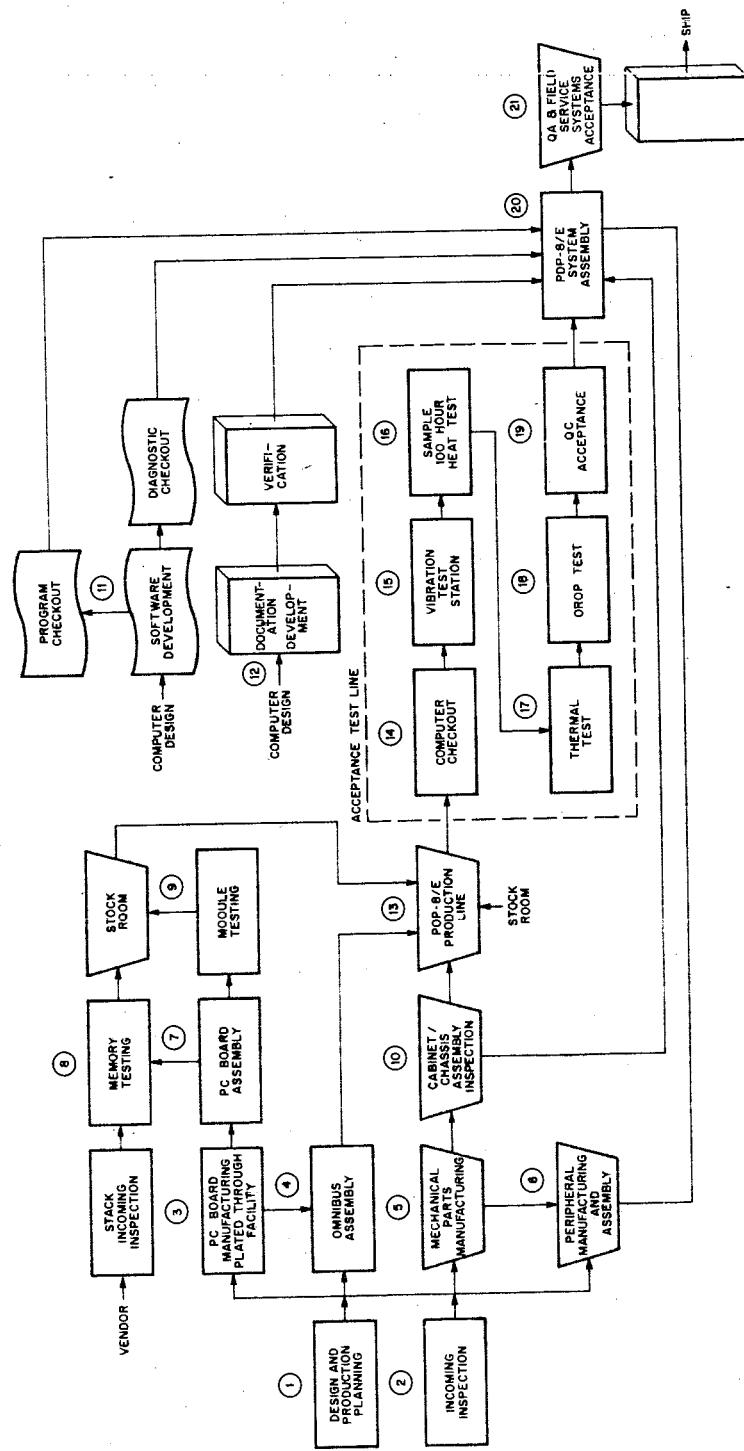


## pdp8/e story

The PDP-8/E story is a guided tour, using pictures and descriptions, of Digital Equipment Corporation. We want you to see the skilled people, the manufacturing processes, the scores of test stations, and the wide variety of DEC products—all of which contribute to produce the finest, most cost effective computers and related products on the market.



The home office and main manufacturing facilities for DEC, the third largest computer manufacturer in the world, are located in this mill complex in Maynard, Massachusetts. We have 1,000,000 square feet here, about 100 times more than when the company started producing digital modules 14 years ago.



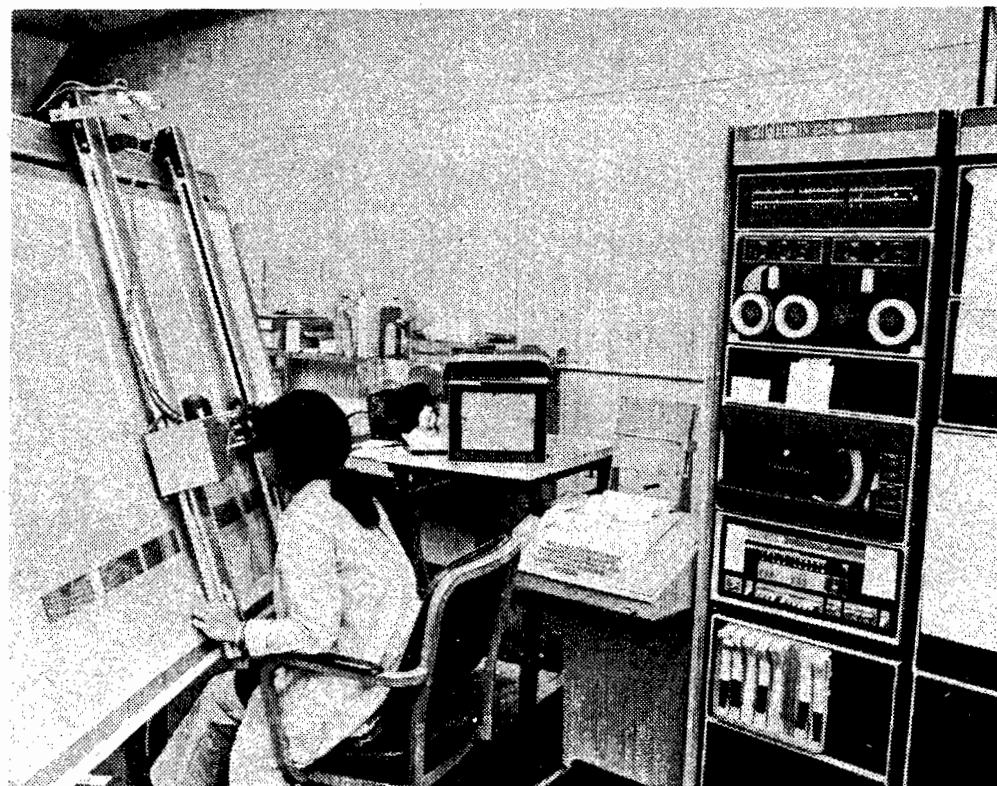
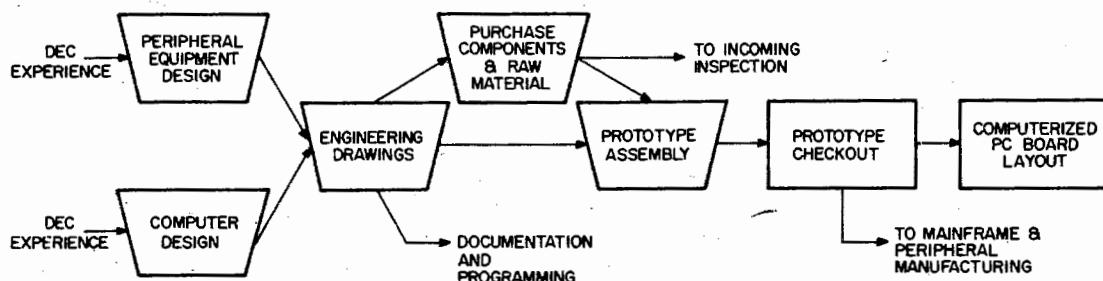
### THE STORY

A diagram is provided for you to relate the photo-story to the actual events that occur here every day at DEC. It begins with design, evolves to incoming inspection of components and raw materials, then progresses to a finished computer system ready for shipment to a customer. The numbers in each block refer to the part of the story about that particular process or test.

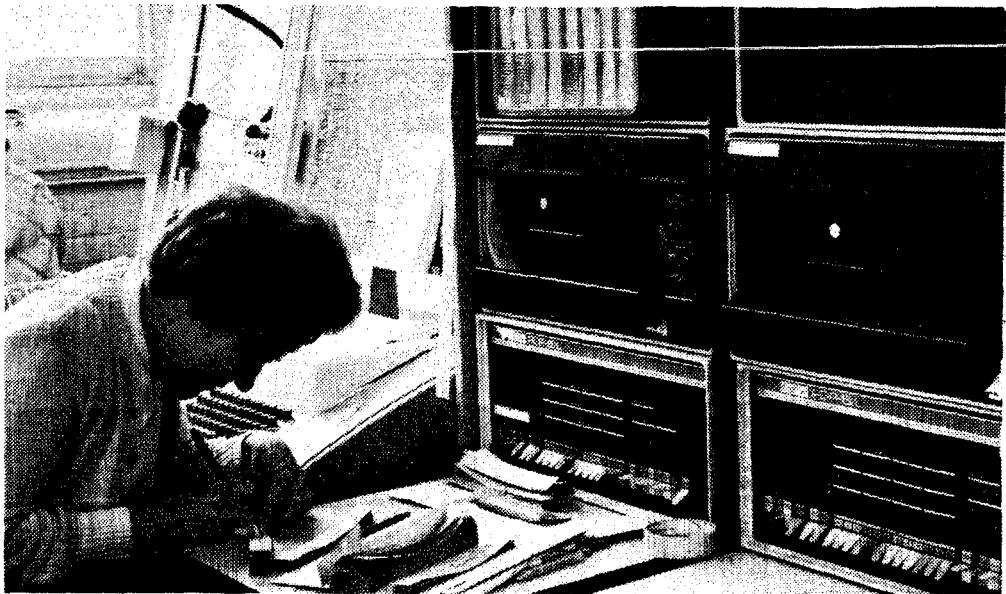
### (1) Design and Production Planning

The experience that DEC has acquired from many years of computer and peripheral manufacturing goes into DEC's newest computers and peripherals. No equipment is manufactured until the prototype has undergone full evaluation by engineering, quality assurance, and field service.

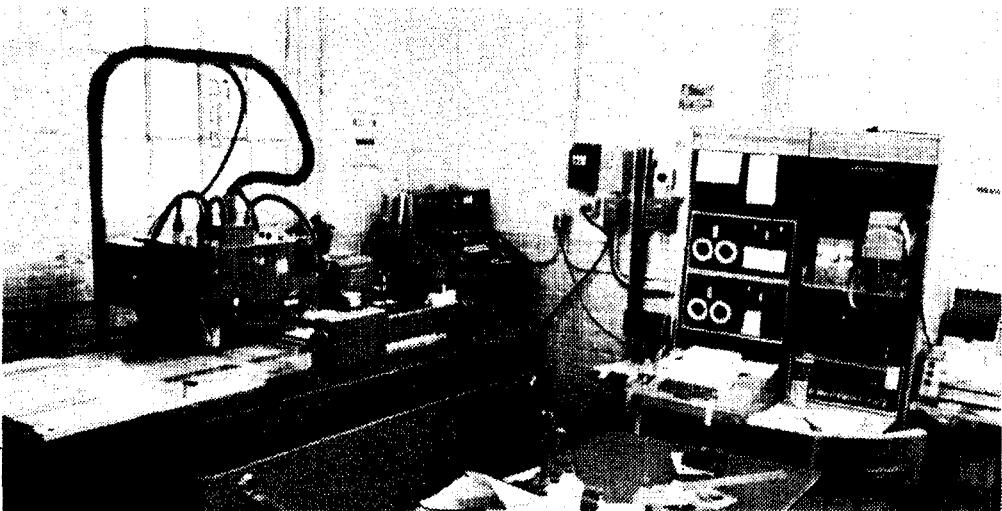
After evaluation, production planning begins. New test stations to accommodate high volume testing are designed and produced. DEC's programming department immediately goes to work on new programs for all computerized testing.



Circuit layouts are finalized by logging the XY coordinates of components on a GEMS digitizer interfaced to a PDP-8/E. Automated plotting affords greater precision than traditional hand-taping methods.



DEC uses a PDP-8 Computer with a Digitizer to prepare for highly accurate automated drilling operations on PDP-8 logic module boards. Drilling coordinates are retrieved from a layout of a module (shown on the drafting table). The information is stored in core memory, and the computer generates a paper tape that contains digitized information about the location of the holes to be drilled in the module boards.



The paper tape containing the digitized information is then taken to another PDP-8 computer for post processing to produce another paper tape with all of the various control signals to run the drilling machine. Thus, a PDP-8 Computer is actively involved in producing new PDP-8 Computers.

The X- and Y-coordinate information is first plotted out on an automatic plotter to check its accuracy and then post processed in the larger PDP-8 Computer. Next comes a test run on the drilling machine to see the results.



DEC uses computers to design more computers.

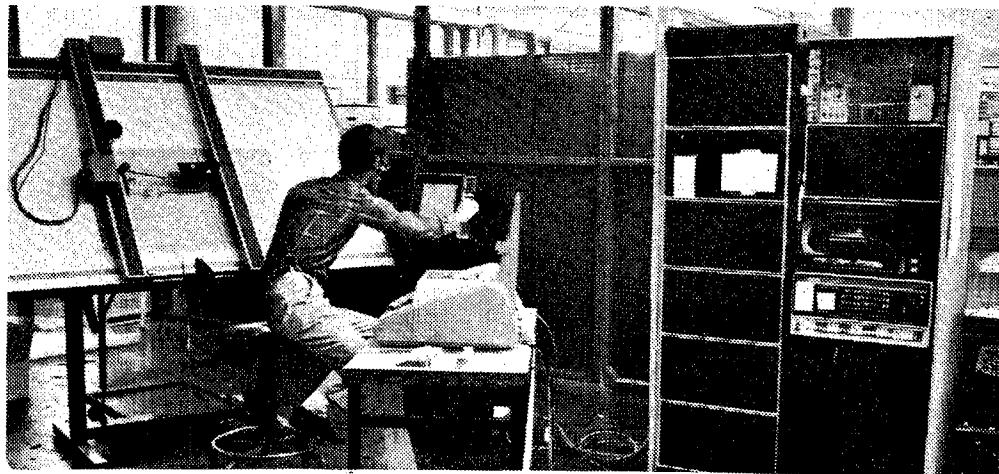
The PC board layout system (using a PDP-8 Computer, a KV Graphics System, and a Digitizer) is another example of computers being used to design more computers.

The computer is used to design and lay out each circuit board and obtain drilling coordinates.

The system provides the layout of a PC board from hand-drawn sketches by inputting X- and Y-coordinate information into the computer in digitized form. When the operator wants a connector to be placed at a particular location, he locates the digitizer cursor at the starting point and commands the computer via the Teletype®. The appropriate connector appears on the graphics display.

This information, in digitized form, is available to lay out the PC board, drill holes for the various components, operate the computerized component insertion machine, and other specialized functions. With this system, DEC is able to computerize a large part of the process of laying out and producing printed circuit boards.

® Teletype is a registered trademark of Teletype Corporation.



## QUALITY

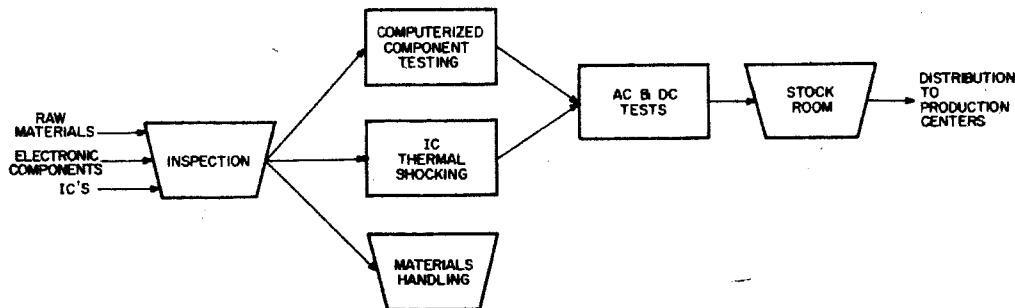
We have built more than 20,000 computers. Naturally, at DEC computerized testing techniques play a major role. Dynamic testing controlled by computers begins as each component is received at the plant and continues through most all production phases. As the major components of the computer progress through the assembly lines, the testing becomes more and more complex, and culminates with the final acceptance test of the finished system. Before a unit progresses to the next assembly or test station, it must meet the rigid standards imposed by DEC.

Computerized testing is ideal for quality control. Many similar tests are continually being run. By automating the tests, all results are calculated the same way and printed out in a standard format, thereby increasing test reliability and accuracy. The cost of quality control tests is drastically reduced by cutting manhours required for other test methods. The computer can control the tests, as well as acquire data and calculate results, and the system is flexible enough to make real-time "decisions" as the test progresses.

The advantages of using small computers during design, production, and testing are mainly economical. Small computers are inexpensive and can be located in the shop, right where the action is.

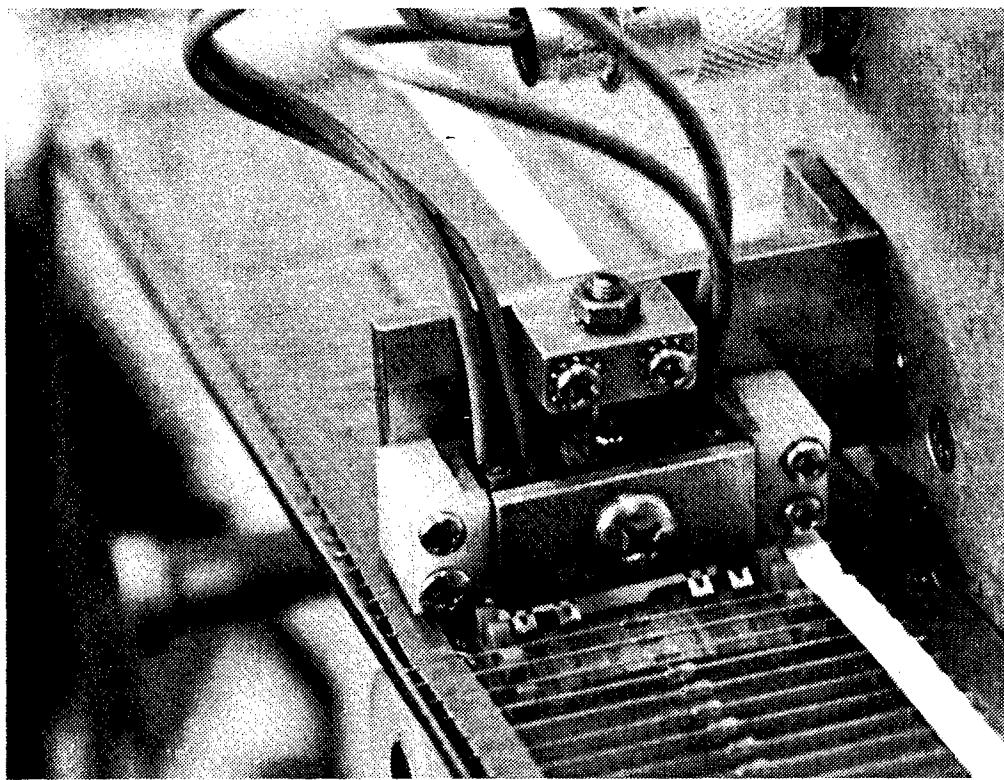
In manufacturing, computers provide on-the-spot testing. When repeatability of testing is important, computers make certain that all components meet the required standards. A computer can do the same task identically again and again; human variation in performing tests is virtually eliminated. The result is a test that is identical for each component being tested.

A written record of test results is often necessary. In computerized testing, the record is available the instant the test results are available. This is particularly important, especially on an assembly line where the unit must be qualified at one station before moving on to the next station. The test operator can press a button and instantly receive a printout of test results.

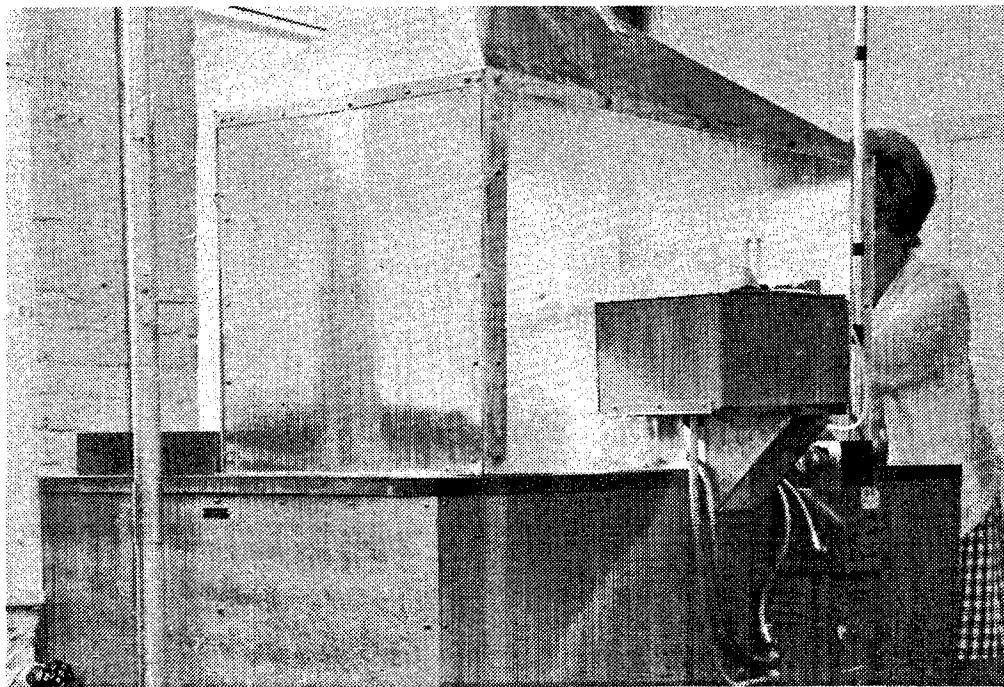


### (2) Incoming Inspection

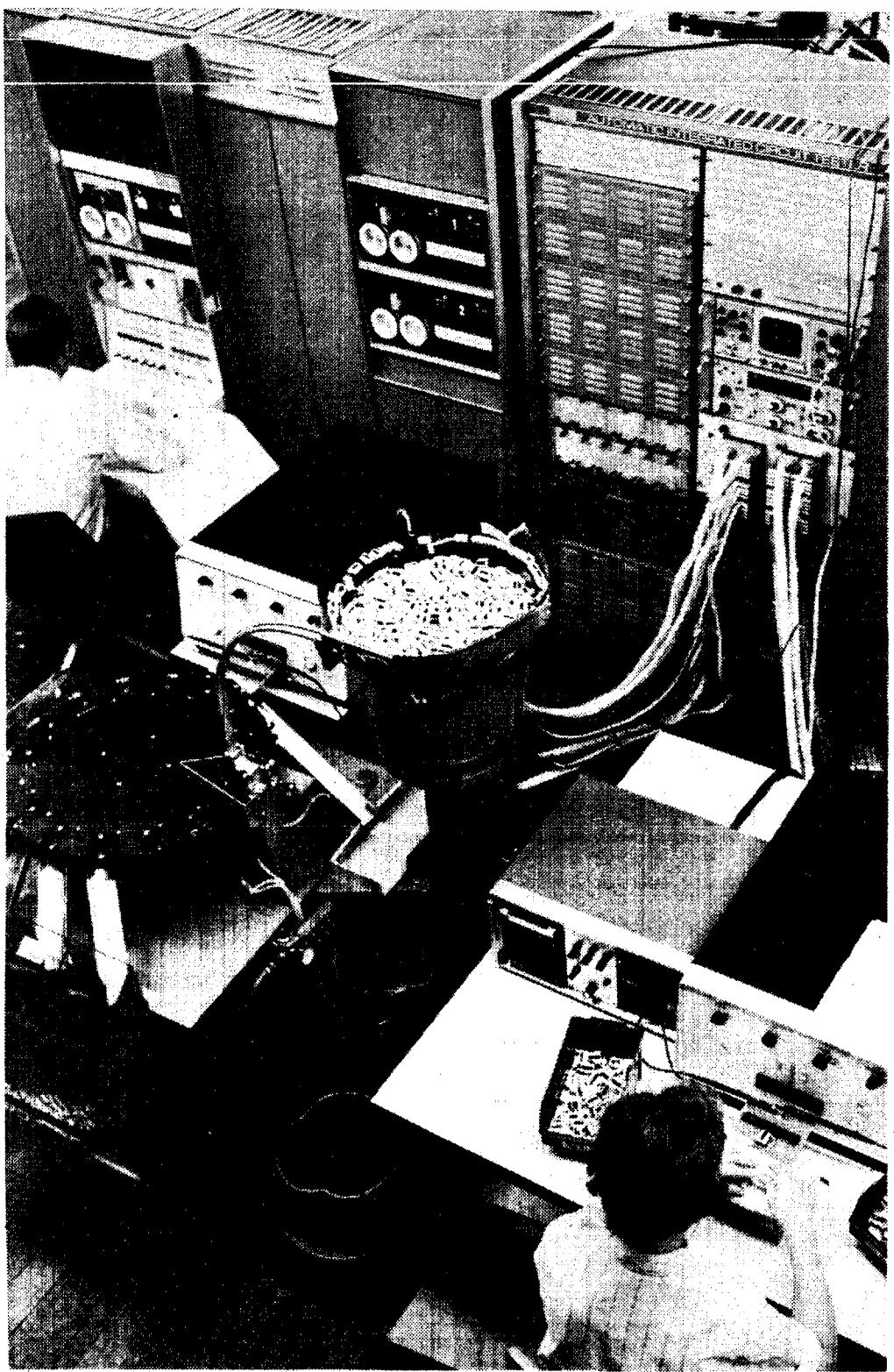
Inspection, testing and more testing, right from the beginning, is a major factor in the PDP-8 family success story. All material, components, and integrated circuits (ICs) must pass rigorous inspections before being placed in DEC's stock room.



All incoming components are 100% tested. Here, diodes are being tested automatically.



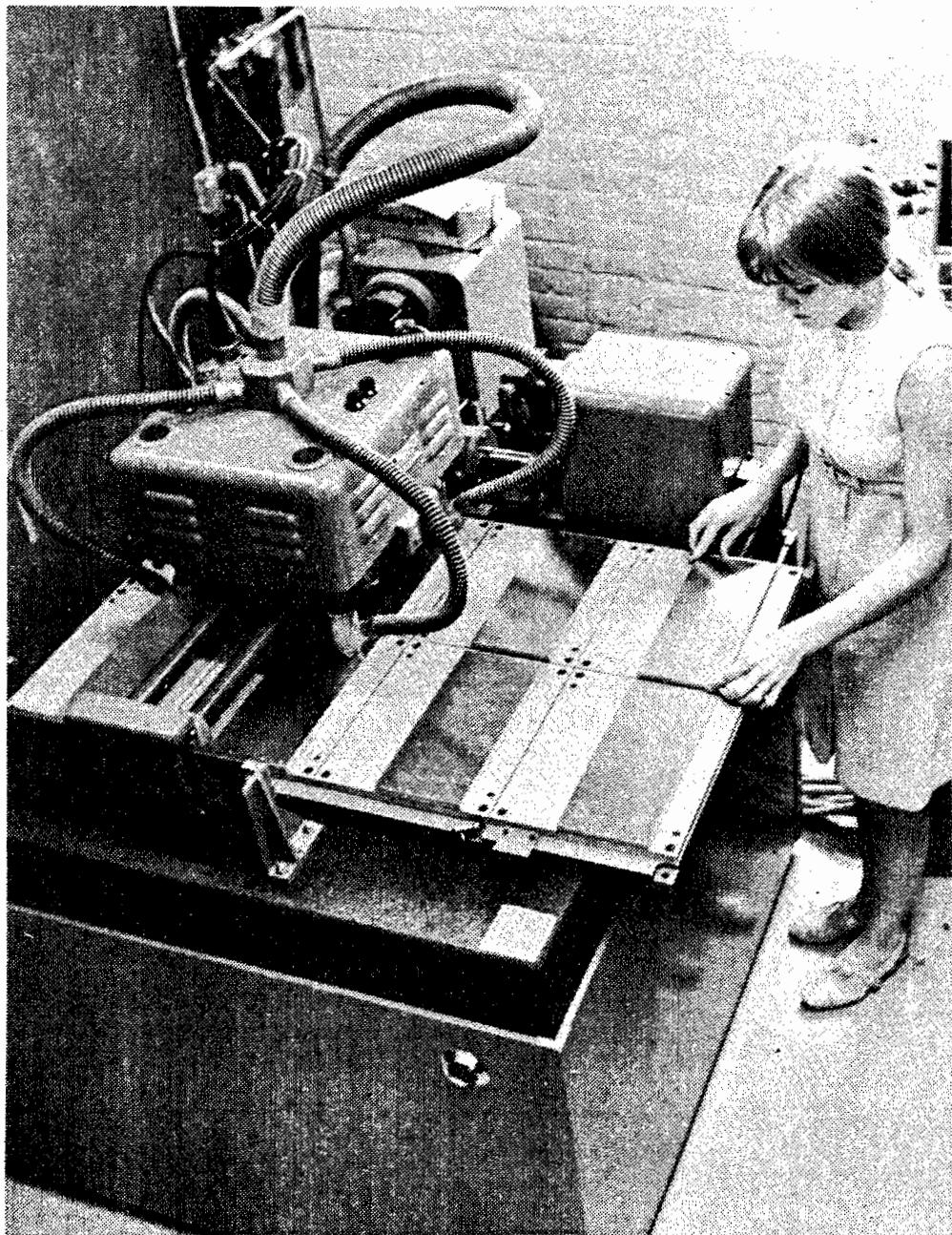
IC's are first given a cold test by placing the IC's into a bath at 32°F for 2 minutes. The IC's are then cycled into another chamber at a temperature of 212° F to force any possible fault to appear. Then testing for faults begins.



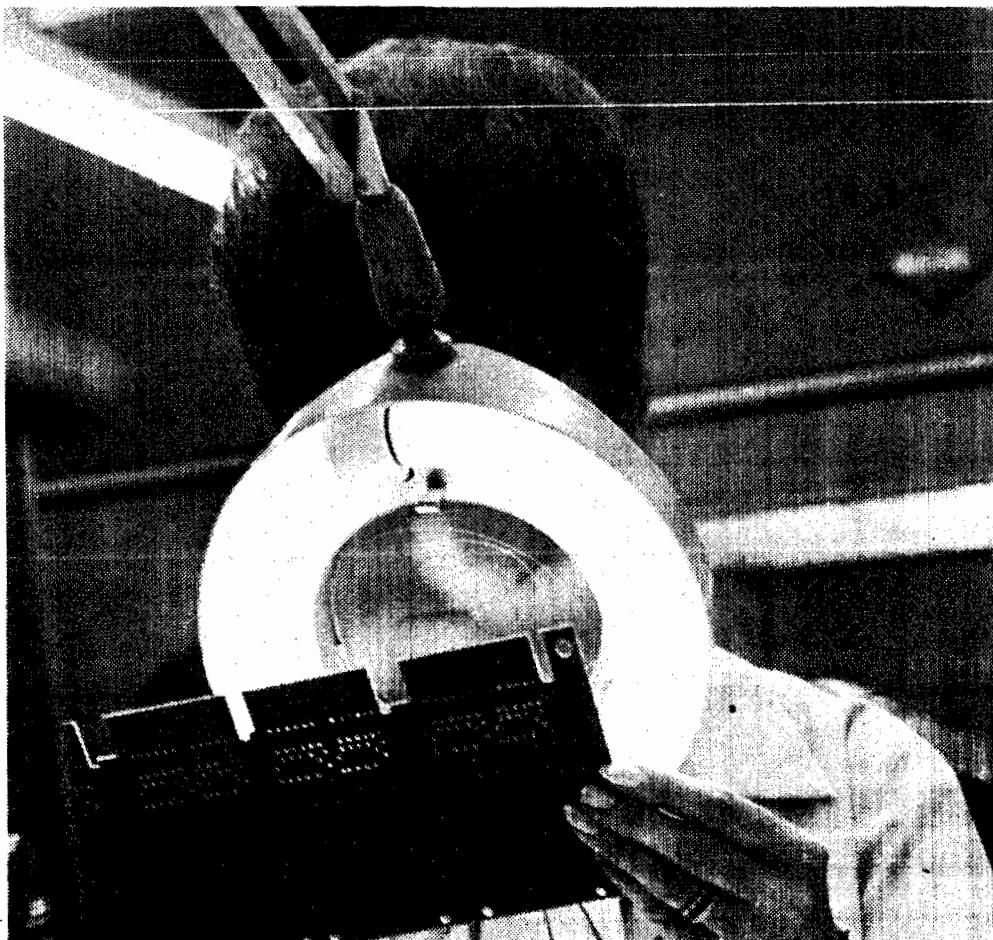
Incoming integrated circuits undergo computer controlled testing, with 40 dc and 16 ac tests performed in 1.1 seconds. This 100% inspection speeds production by minimizing the diagnosis of component failures in module test.

### **(3) PC Board Manufacturing Plated-Through Facility**

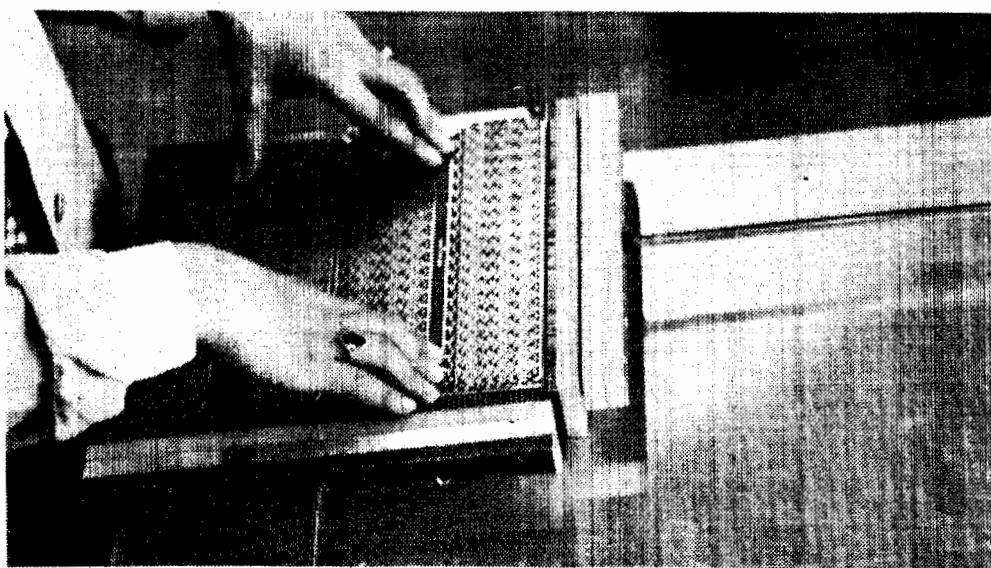
The manufacturing of printed circuit (PC) boards requires a facility that provides a controlled process and rigorous quality control. DEC is a world leader in the manufacturing of logic modules. We produce more than 3,000,000 modules per year and have been producing logic modules since 1957.



Twenty module boards are drilled simultaneously from a PDP-8 computer-generated coordinate tape. Other pantograph-controlled machines drill up to 200 boards simultaneously from a PDP-8 computer-generated template.

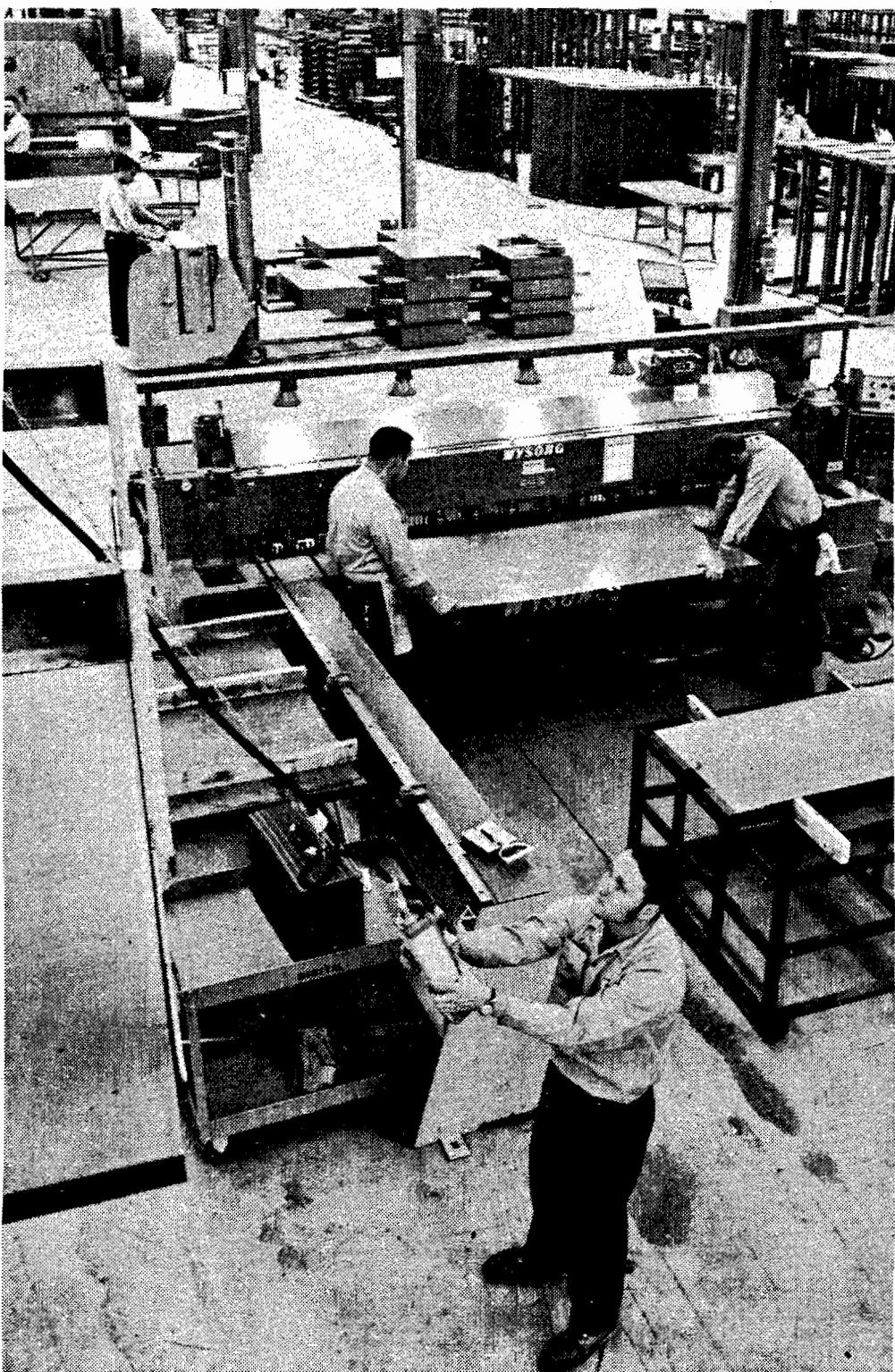


Quality of plated-thru holes is checked in our new electrochemical facility before boards go to the module assembly area.



**(4) OMNIBUS Assembly** A PC Board and Connector Block are assembled here.

**(5) Cabinet Assembly**



Cabinets for DEC systems are manufactured in this portion of DEC's Westfield, Massachusetts production facility.



#### (6) Peripheral Manufacturing

The blossoming of more peripheral assembly lines is a very real indication of DEC's continual expansion of products. At the DEC manufacturing plants shown above, just such an assembly line is producing the famous DECTape. Each component is given the usual controlled inspection procedure. Modules, which are used to control the operation of each DECTape, are produced in DEC's automated module assembly area. Quality control is the highest priority item. A series of severe tests and checks are run on all products.

### (7) PC Board Assembly

The PC Board Assembly includes inserting components, soldering component leads and gold plating all printed circuit connectors.

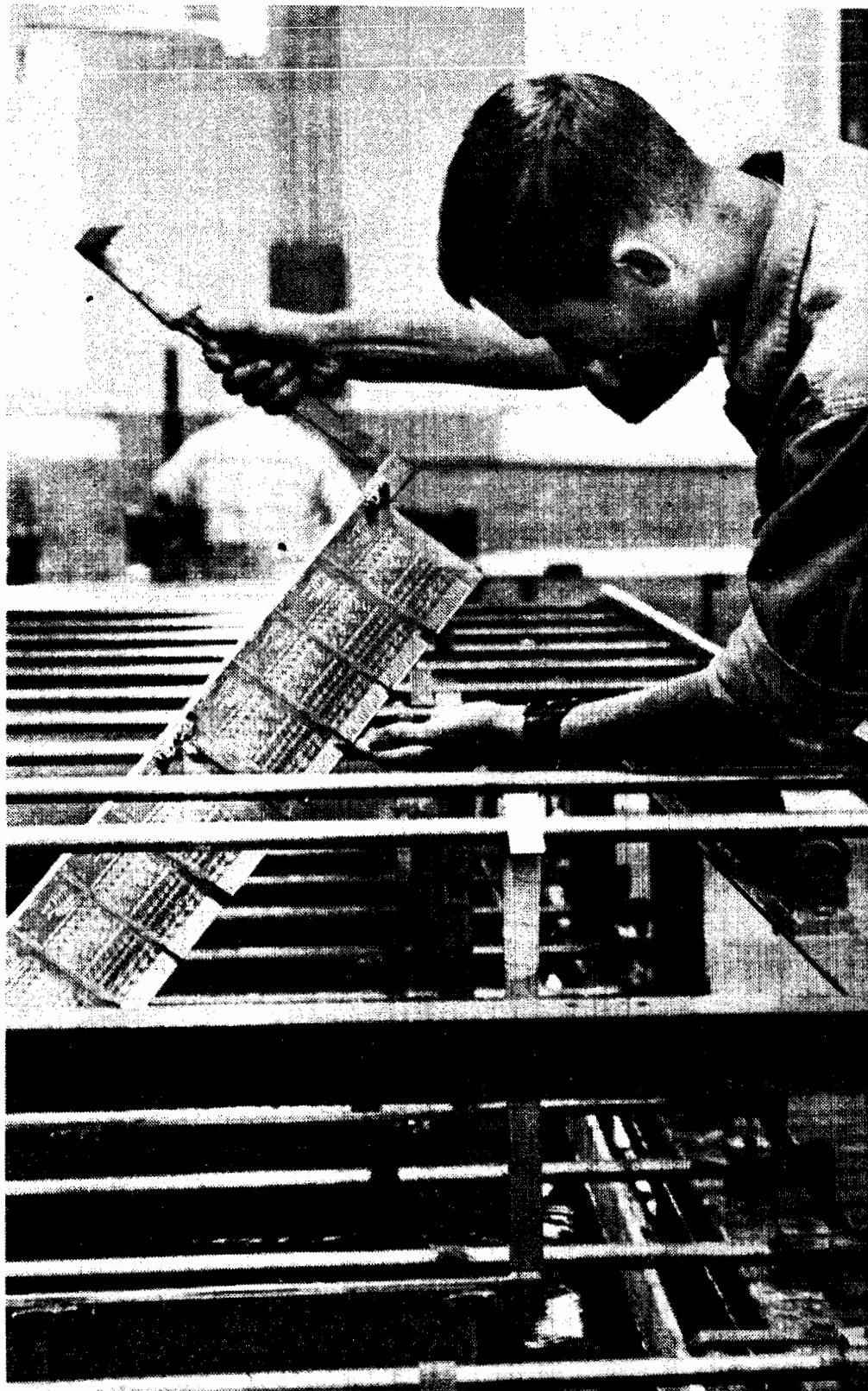


Component Insertion Machine

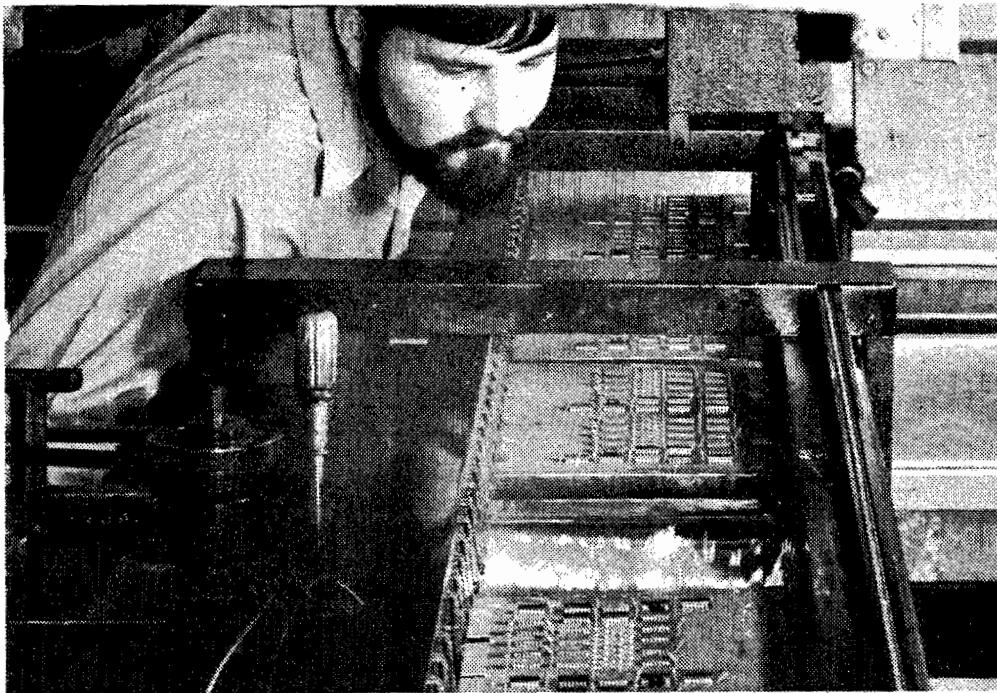
DEC designed and built a multistation component-insertion system to insert diodes, resistors, and capacitors into PC boards. Eight stations are controlled by one PDP-8 Computer. Each station contains a component-insertion machine with table driven stepping motors directly coupled to a rotary incremental-optical encoder.

An X-Y table holding a batch of printed circuit boards is stepped back and forth under a stapling mechanism that inserts electronic components into predrilled holes on the boards at high rates.

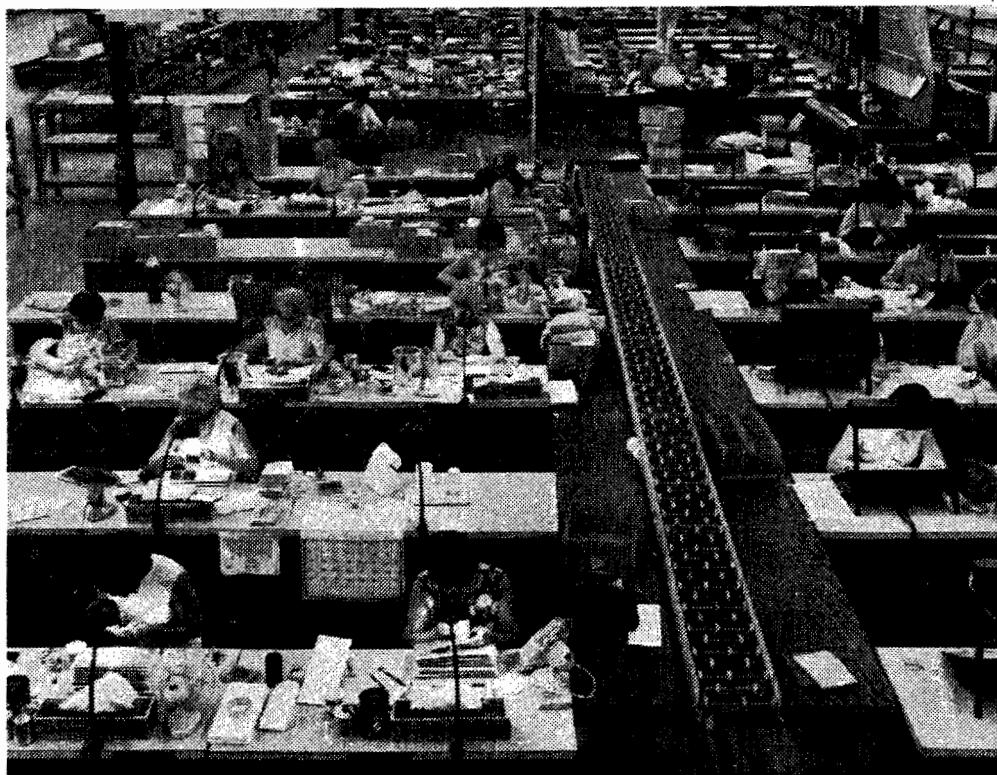
The PDP-8 System uses a magnetic tape deck containing a library for PC-board parts lists. Each station has a custom-built control panel that permits the operator to start, stop, back up, go forward, jog-in offsets, and select parts from lists. The electronic parts are loaded into the insertion machines in paper-taped belts on large reels.



**Checking the appearance of board contacts being gold-plated. Our 100 micro-inch plating is verified by periodic checking on a radiation gauge.**



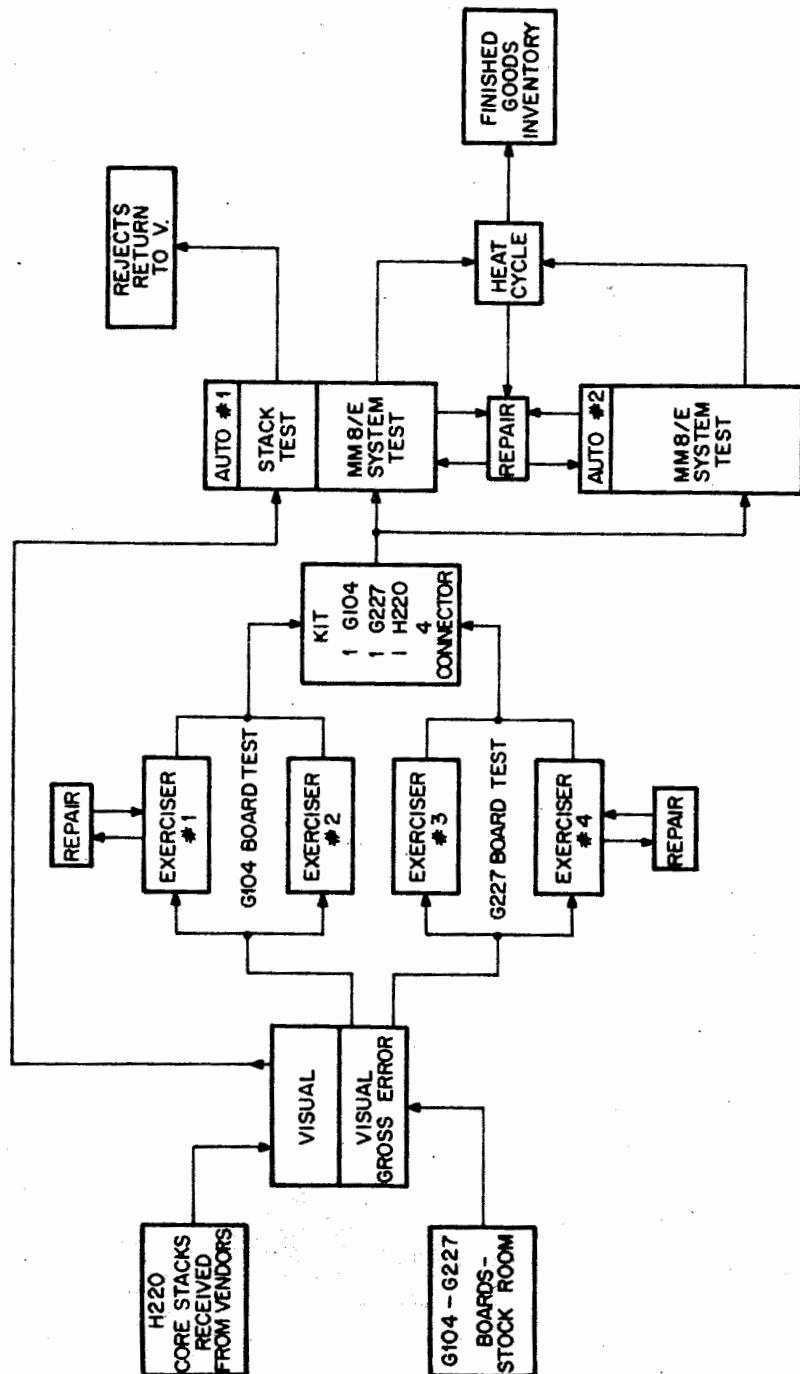
This flow-soldering machine solders all component leads to the board and makes all solder runs in one fast, exceedingly reliable, operation. More than 1000 modules are soldered on this assembly line each day.

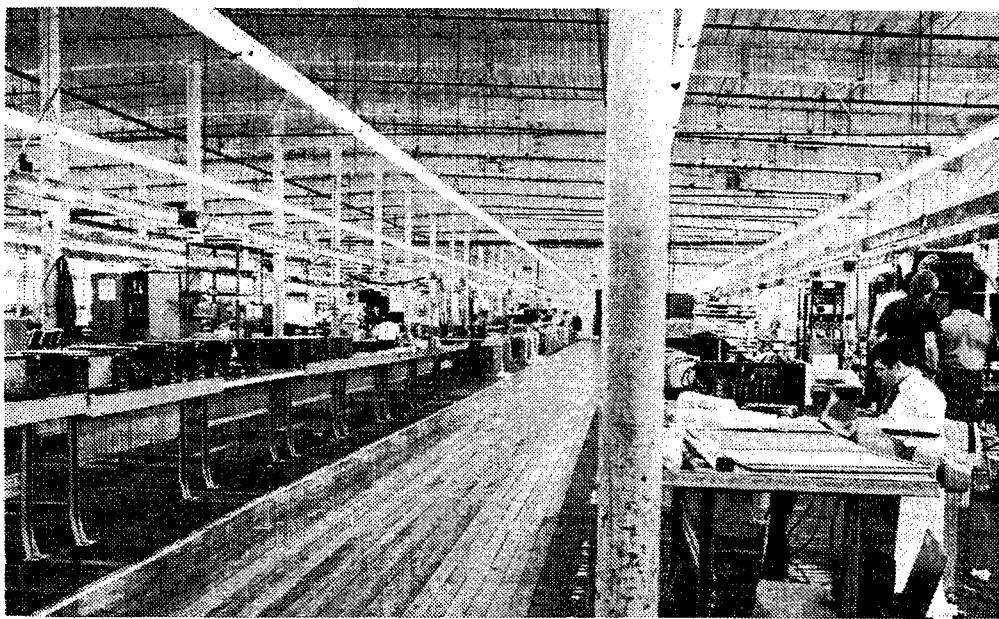


DEC has more than 2 million square feet of manufacturing space. This view shows a portion of a module assembly area.

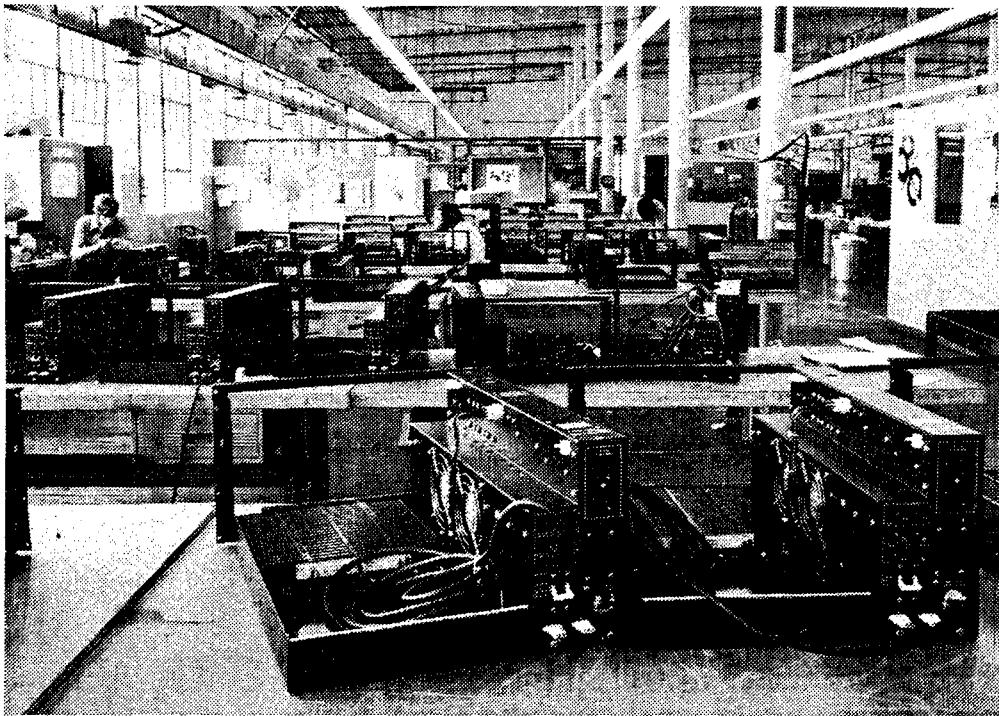
## (8) Computers Test Memory Systems

Computers perform three complex testing operations on each memory system before it is approved by quality control. Each module is visually inspected and taken to a manual memory exerciser, qualified, and placed with other modules where a memory system kit is assembled. A complete memory system test is performed, and the assembled memory system is qualified. A final test is performed with diagnostic programs, exercising the memory system at its highest specified temperature limits in a heat chamber. Refer to MM8-E flow of inspection and testing.



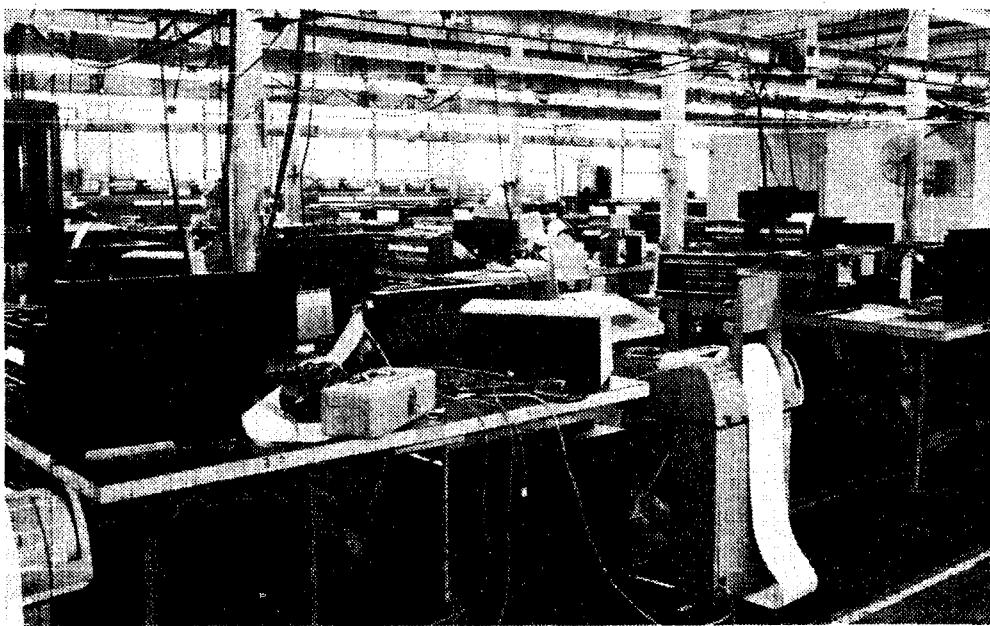


The PDP-8/E production line has the capability of manufacturing 1,000 PDP-8/E Computers per month.

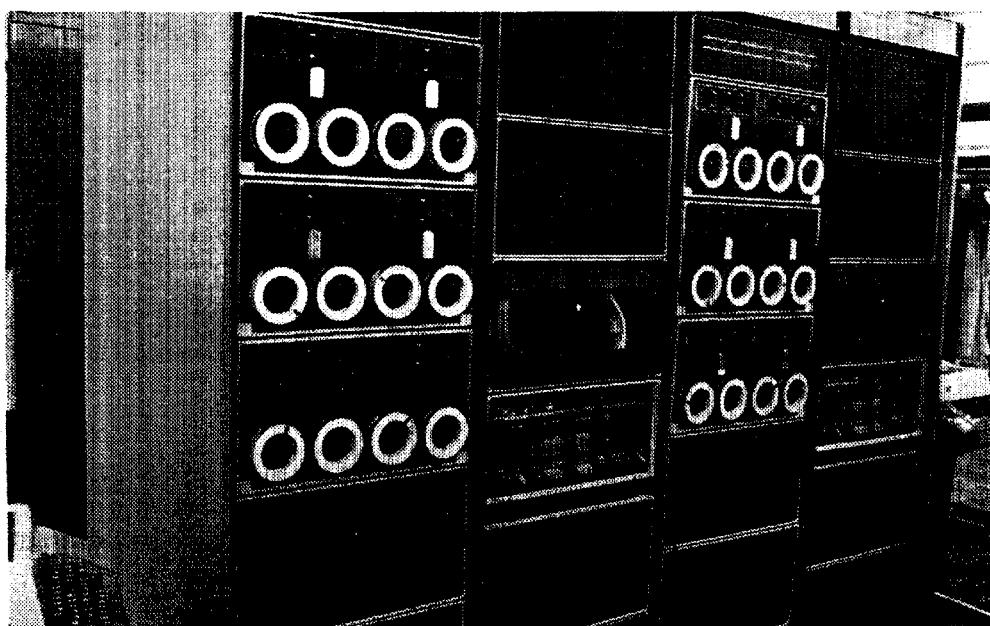


Final Assembly Area

Here, 40 PDP-8/E Computers are shown in various stages of assembly. After assembly is complete, each unit is moved to another area where power is applied and the assembled unit is tested.



Assembled computer testing is done in DEC's acceptance test line. Up to 64 test stations can be controlled by one of two PDP-8/E Computer Systems; there are 6 DECTapes on each system, containing an assortment of test programs and exercises for each test station. Thus, 64 computers can be tested simultaneously by a master controller. The PDP-8/E master controller loads diagnostic programs directly into memory of the new PDP-8/E computers under test, thereby checking out the new computers thoroughly and efficiently. The DECTapes contain all of the programs required to check out the various PDP-8/E's, as well as the operating programs to control the entire test line.

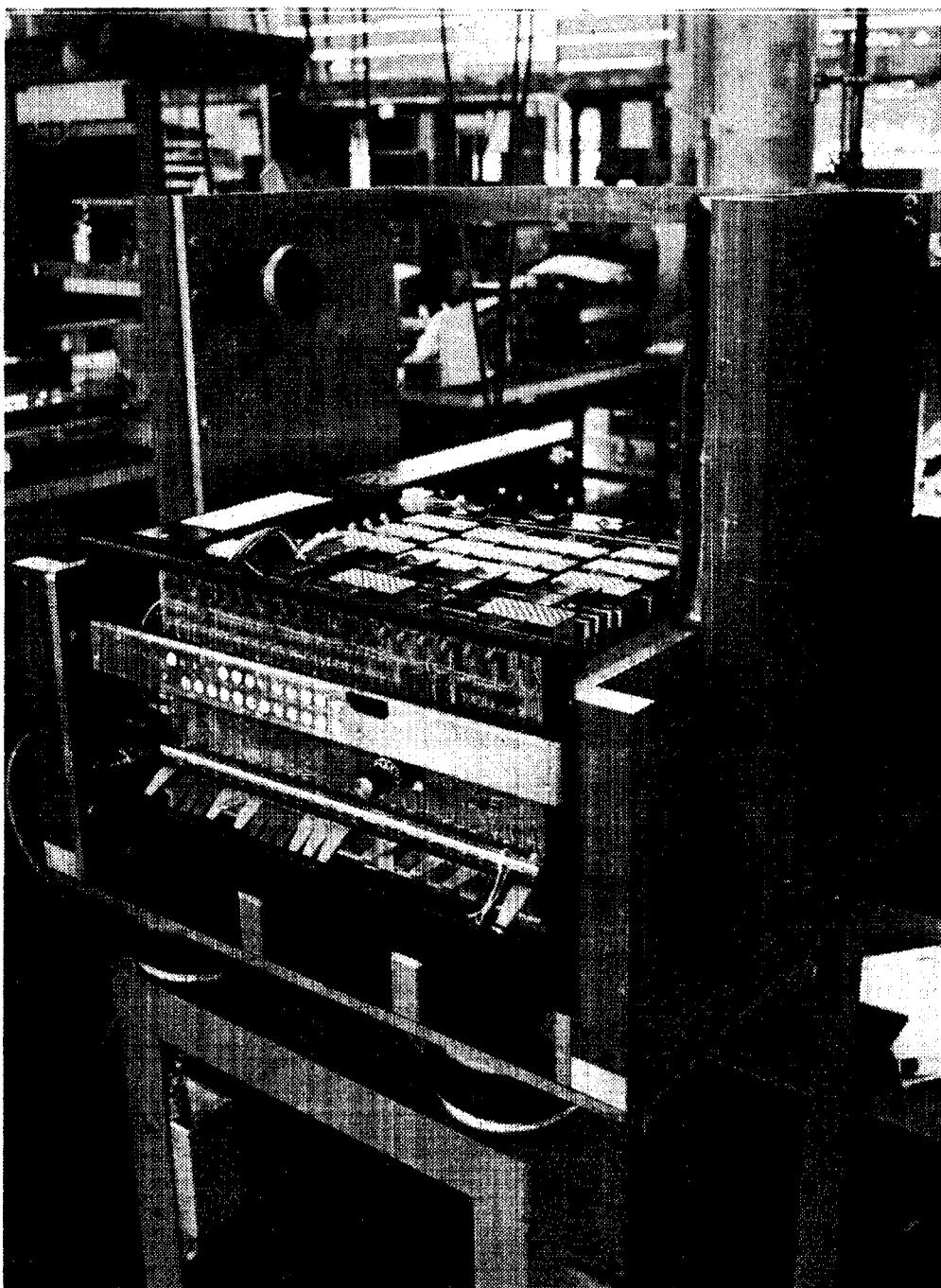


#### (14) Computer Checkout



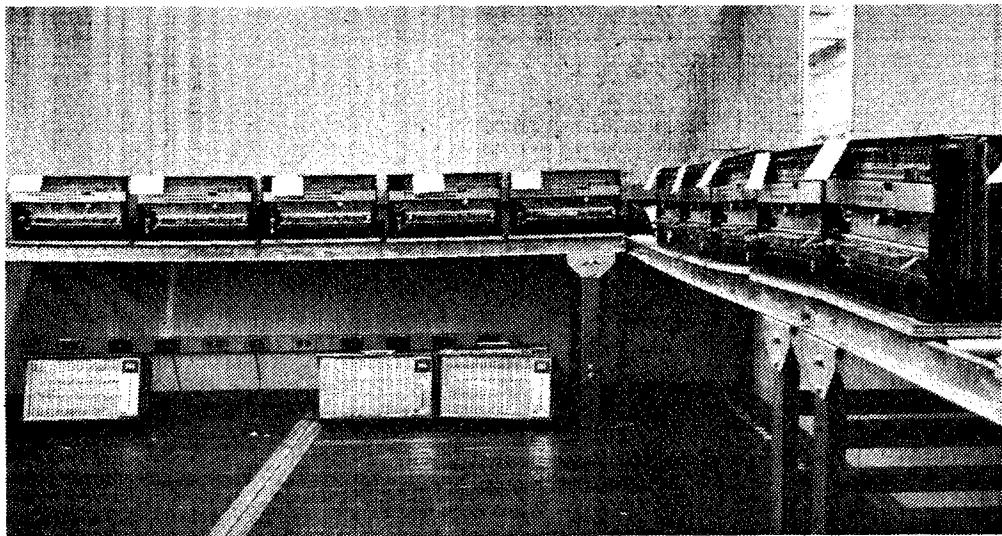
All assembled computers are tested at the 8/E Acceptance Test Station. By coding of the switches on the front of the computer, a technician can request certain diagnostic programs to be loaded into the PDP-8/E. Another switch enables Auto or Manual operation. The technician can either manually go through each test program while he is watching the results or place the switch in the Automatic Mode allowing the PDP-8/E Computer to continually cycle the various test programs through the unit without an operator. On the far left of the test panel is a switch labeled HEAT BOX. This switch activates the heater elements of another unit (not shown) and gives the computer a final heat test at this station.

**(15) Vibration Test Station**

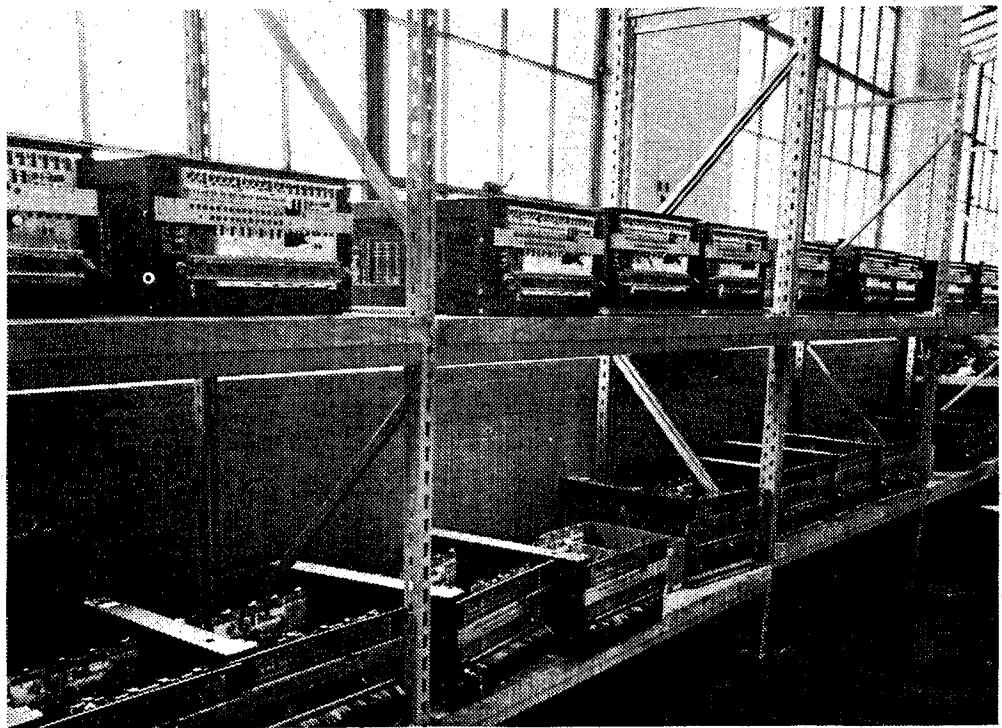


In this production phase of the computer testing, the 8/E is placed on a vibration table and vibrated for several minutes at 70 cycles per second. This test checks for any loose components, cold solder joints, and other malfunctions that can appear under severe vibration conditions. Following this test, the computer is rechecked with the various diagnostic programs. While the unit is undergoing the vibration tests, the memory checkerboard diagnostic is run.

## (16) Sample 100-Hour Heat Tests



DEC takes a random sample of working PDP-8/E's and runs them for a period of 100 hours at 131° F. This workout allows us to check for "early failing components or sub-assemblies." The information gained helps us to improve the long-term reliability of all the units. In another test, all 8/E's are placed into a cold chamber at 32° F or 0° C. This forces a computer through another thermal shocking process with a very rapid change in its temperature. Following this cycle, the machine is returned to the heat room at 131 degrees F. This two-stage cycle not only verifies operation at the specified limits, but also subjects the machine to much more stress than the environmental change in the field.





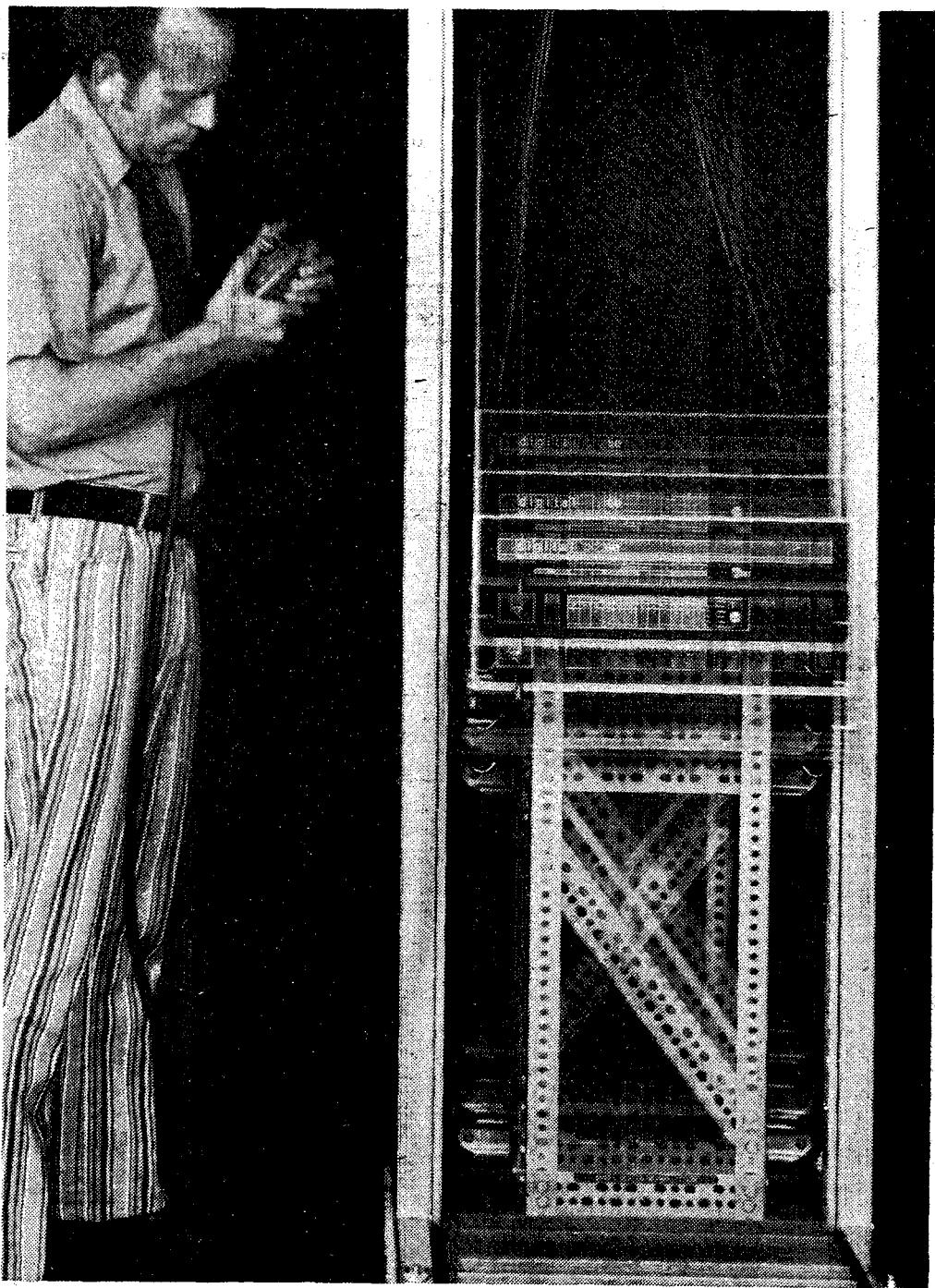
Cold Chamber

**(17) Thermal Shocking**

As a part of the testing and acceptance process, we place each computer in a cold chamber and a memory checkerboard program is run. The chamber temperature is reduced to the minimum specified temperature of the computer; then, the computer is placed in a heat chamber to operate at 131° F. The acceptance test station detects any faults while exercising the computer under test.



Heat Chamber



#### (18) The Drop Test

Some of the most frequent problems in initial installation of a computer are caused by the vibration and rough handling during shipment. To combat these problems, DEC has devised a test that is even rougher than your local transportation company. The 8/E is raised approximately 3 feet above the lower platform and then dropped hard. The test is calculated to place the various components in the 8/E under a 20G force. A second test is performed with the 8/E in a vertical position (panel up) with a 16G impact force.

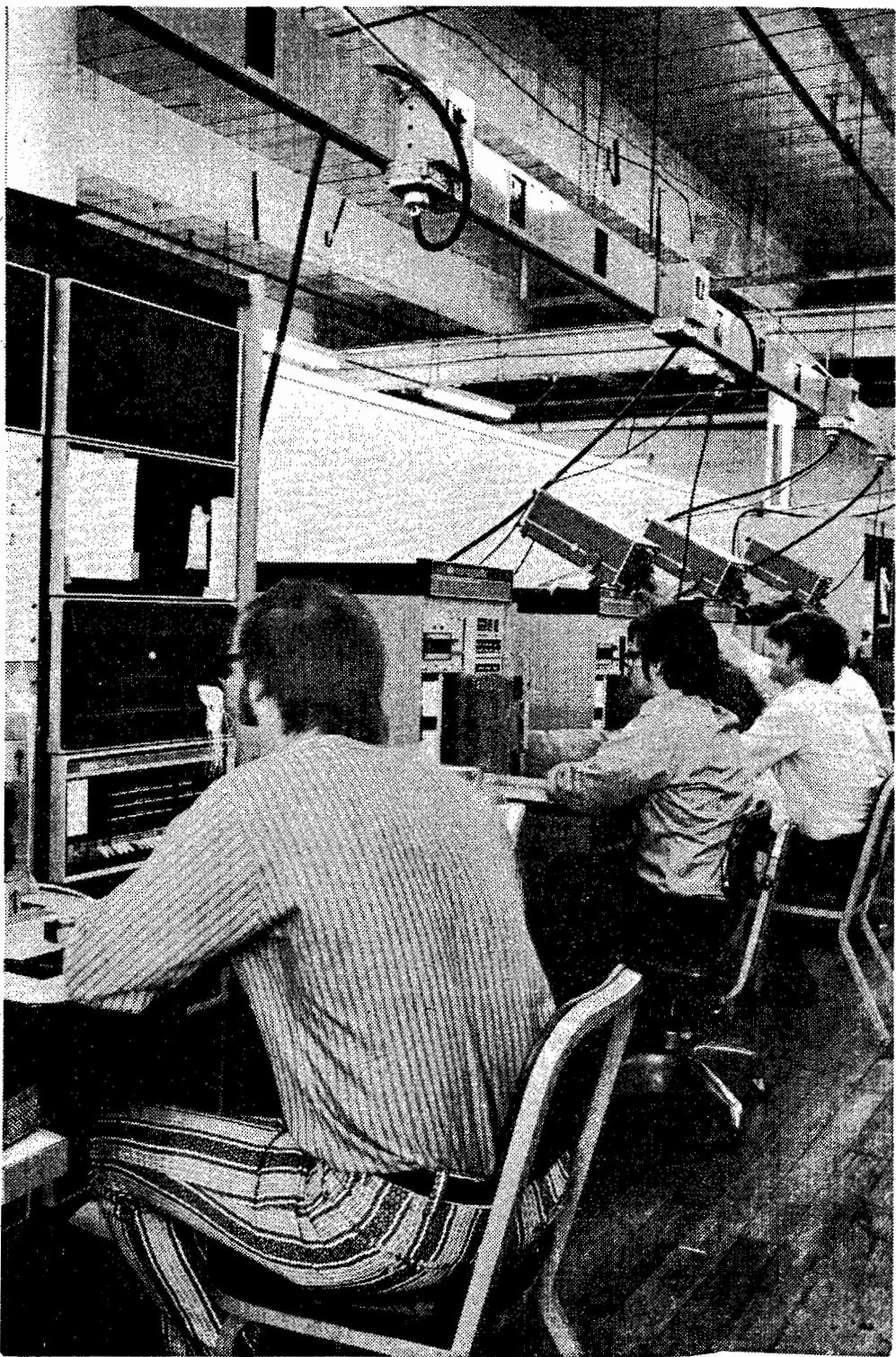
**(19) Quality Assurance and Field Service Acceptance**

At the end of the acceptance test line, the Quality Assurance and Field Service Acceptance groups (independent of the production test groups) run their own tests to verify the quality and performance of the units being shipped.



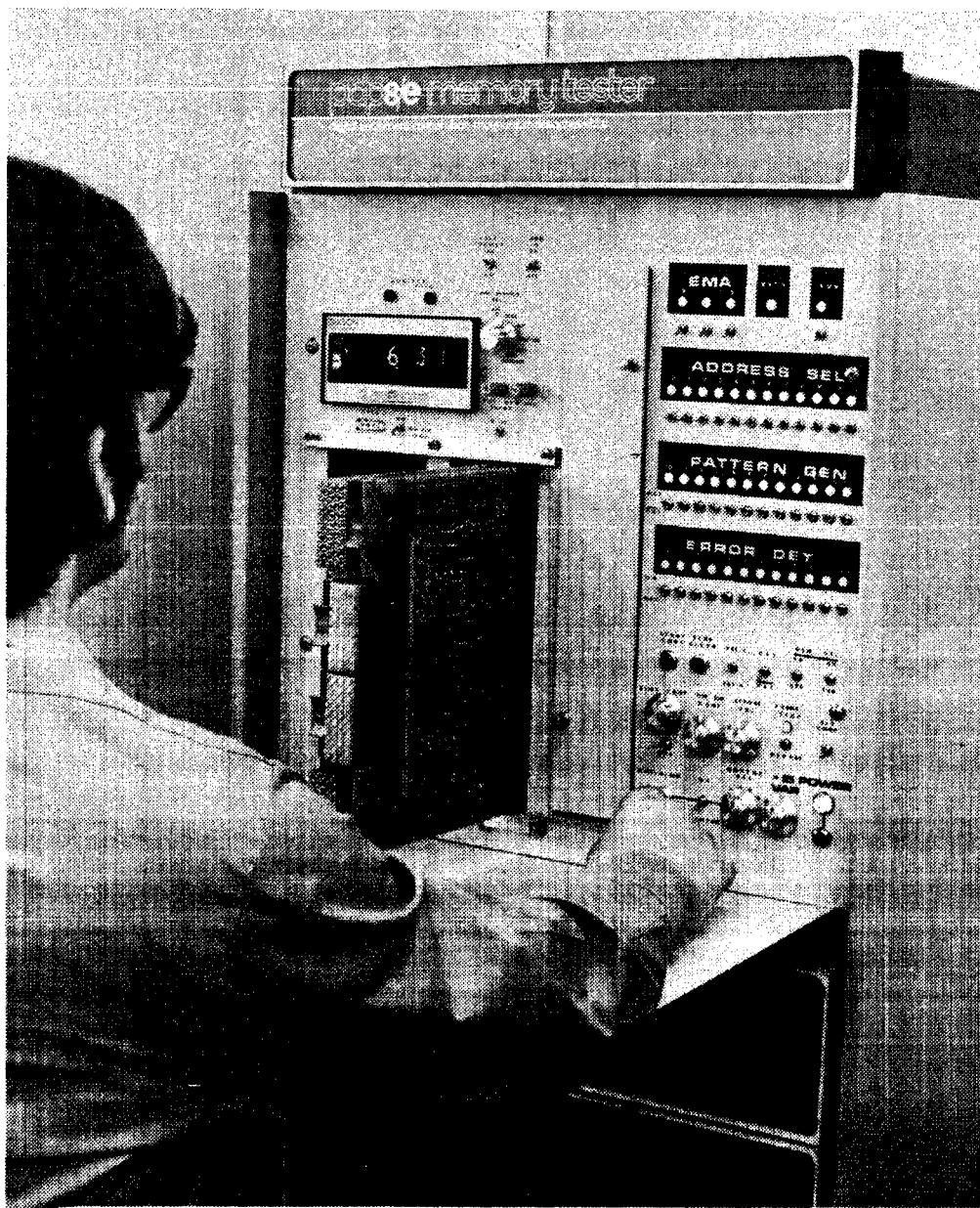
**(20) PDP-8/E System Assembly and Test**

After checkout of the basic computer and its internal options, the unit is moved to the system assembly area where it is installed in a cabinet containing peripheral equipment to form larger systems. In the system assembly and test area, all customer-ordered options are assembled and tested to make absolutely certain that the system is operating according to equipment and program specifications. This continuing testing process assures DEC's customers, all over the world, that each system delivered will go right to work for them and provide many years of reliable service thereafter.



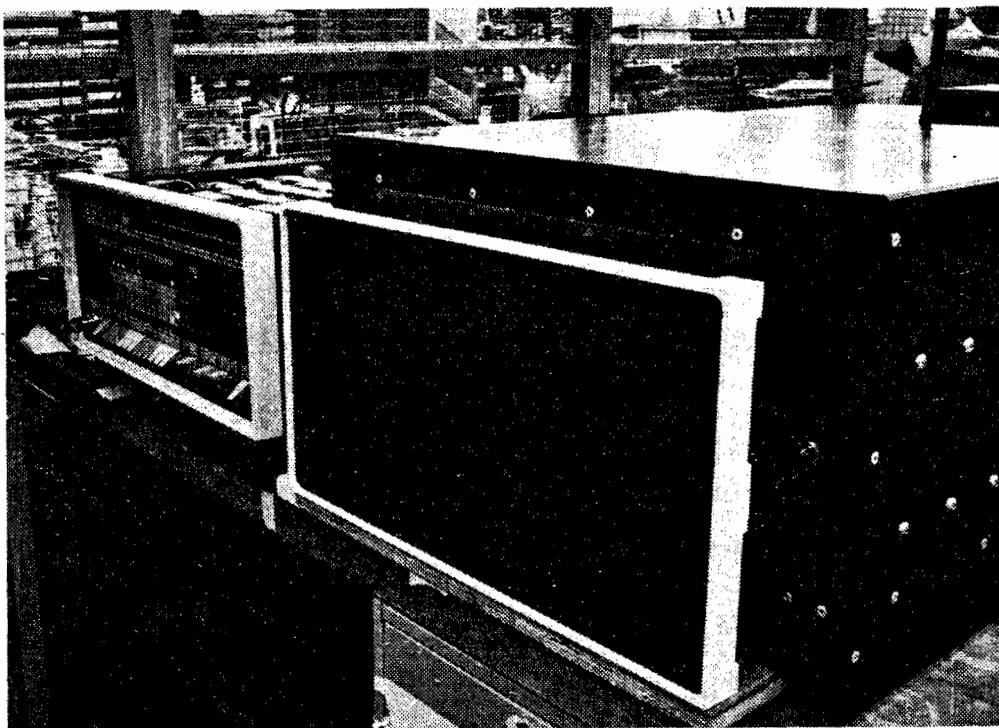
Memory Test Line

Qualifying PDP-8/E memory modules is accomplished by this test line. Every component in the memory modules are subject to thorough testing under a variety of conditions.



The PDP-8 Computer performs the dynamic testing of the memory units (MM8-E's—3-card ensemble). After each memory system kit has been assembled, the kit is tested at DEC's fully automatic station (AUTO #1 or AUTO #2) where typical operations of system characteristics are run to reflect normal operating frequency used by the computer. The tester varies the voltages and currents within the memory system upper and lower limits to ensure that the memory system meets the requirements of the specification. For each parameter tested, corresponding Schmoo-type curves are obtained. The total test time requires only 5 minutes for each memory system tested.

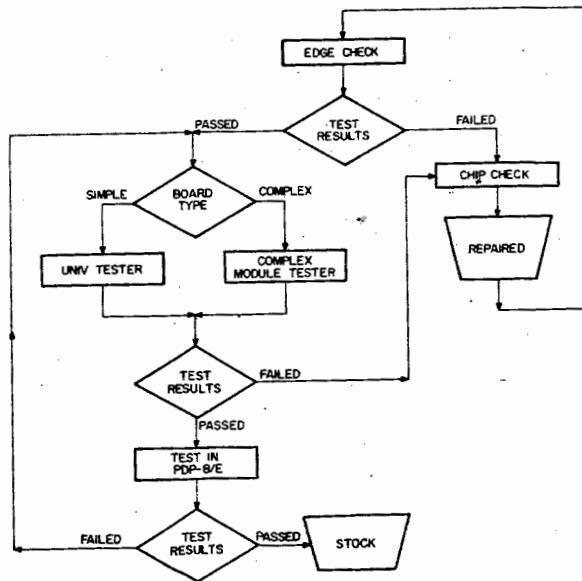
Again, a PDP-8 Computer is working to qualify new PDP-8 Computers. This automated testing technique allows no variation in quality; *no marginal units survive these tests.*



Memory System Heat Test

A final system test is performed by running memory diagnostic programs while the system is operating under maximum allowable temperature. The memory modules are installed in a heating chamber and connected to a PDP-8 Computer. If a fault occurs, a teleprinter connected to the computer prints out the type of fault; if the memory system performs flawlessly, the teleprinter prints a verification.

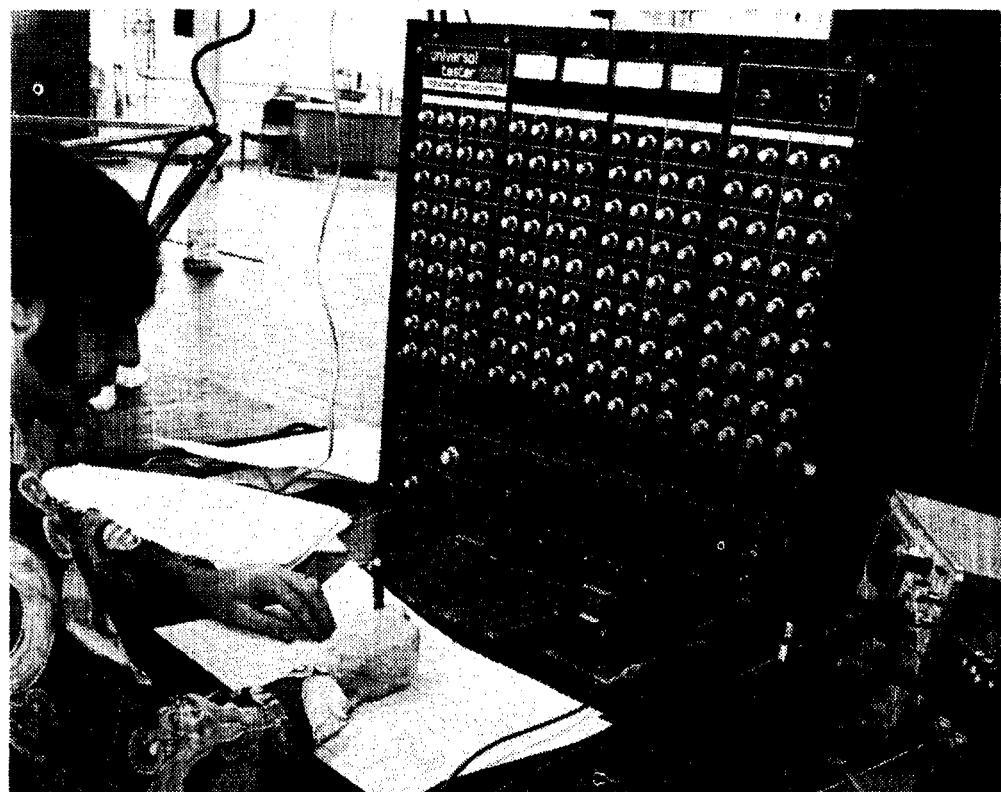
### (9) Computers Test Logic Modules



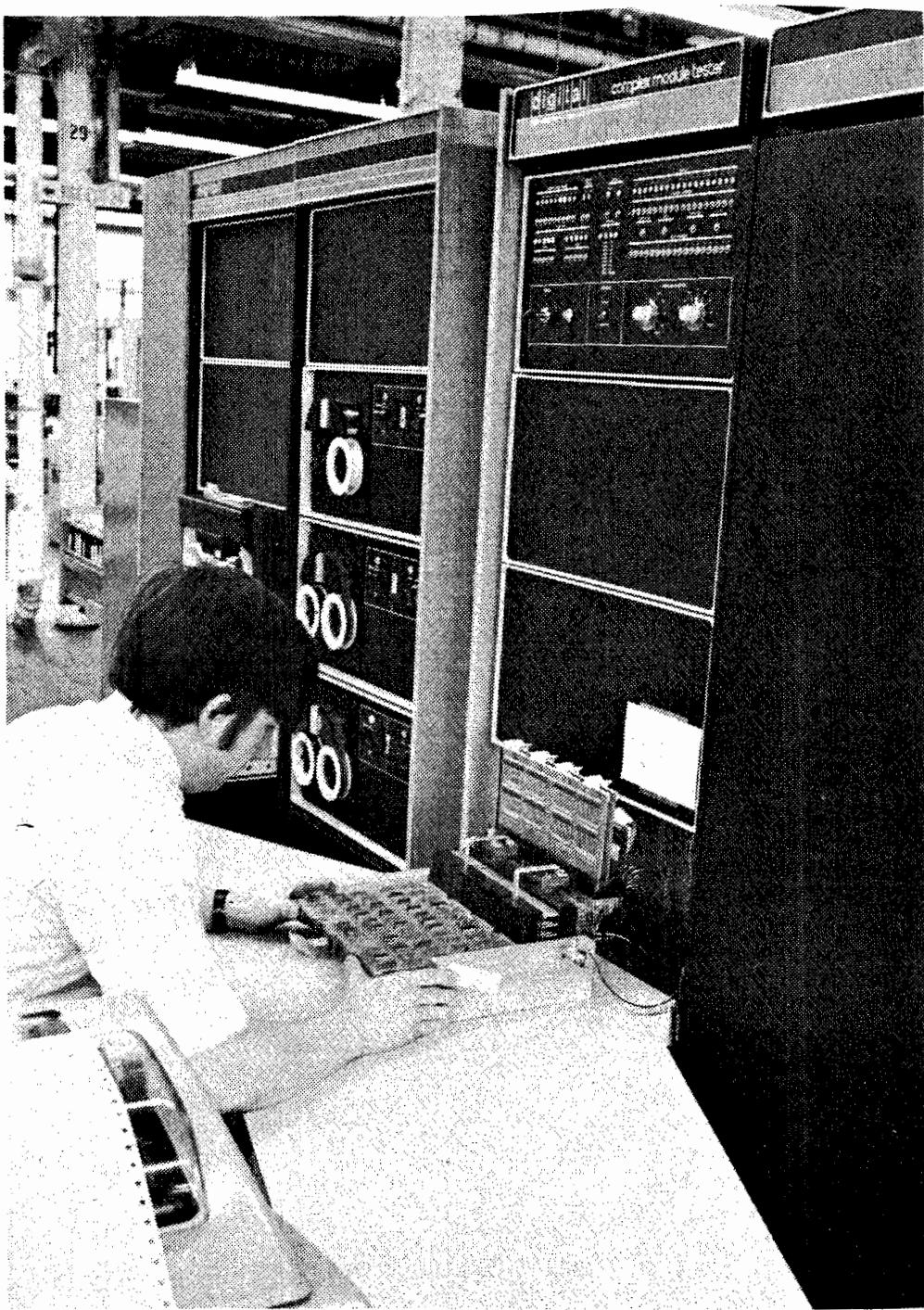
Module Computerized Tests Flow Diagram

A series of computerized tests is performed on all logic modules to maintain DEC's highest standards and ensure long life. Computers using diagnostic programs exercise and test every component on every module before a module is qualified for customer use. Hundreds of repetitive tests are performed in seconds as the computer evaluates every parameter, including maximum and minimum allowable current, frequency, and other important values. If a fault occurs, a teleprinter signals the operator; otherwise, the teleprinter verifies that the unit "passed the tests."

A detailed diagram of module testing is provided above, from the least complicated test to the most complicated test. Computerized testing begins with the edge check, which qualifies all of the circuit paths. If the module is simple, it is routed to the universal tester; otherwise, the module is qualified by the complex module tester. Any time a component failure is detected, a "chip" test is run to locate and replace the failed component. Each accepted module is then tested in a PDP-8/E System and qualified by a series of diagnostic programs that thoroughly exercise every component on the module.



**Universal Tester or Logic Analyzer**—This unit is the tester especially developed for PDP-8/E modules. Using this sensitive tester, a technician can isolate faults on any circuit card used in the PDP-8/E Computer. Through the various controls on the tester, the technician sets up all the various inputs that a circuit board uses. Then, with an oscilloscope he can monitor the output at various pins to verify the operation of circuit paths.



Module Computerized Tests

**Complex Module Tester**—The Complex Module Tester uses a Computer. The operator inserts the circuit board into a connector block and the computer applies the correct inputs and checks the correct outputs from that circuit board to verify its operation. If the circuit board is rejected here, it is passed to another test center where a technician uses the Universal Tester to further diagnose the fault.



**Chip-Checker**—The chip-checker tests individual IC's while mounted on a module board. This unit indexes in X and Y around a circuit board with a special probe that connects to and checks out each integrated circuit on the circuit board. The computer in the background stores the programs for both testing and indexing the tester. DEC tests the integrated circuits (IC's) before being assembled on a circuit board by the incoming inspection method and tests once again after the IC's are assembled on a board.

## (11) Software Development



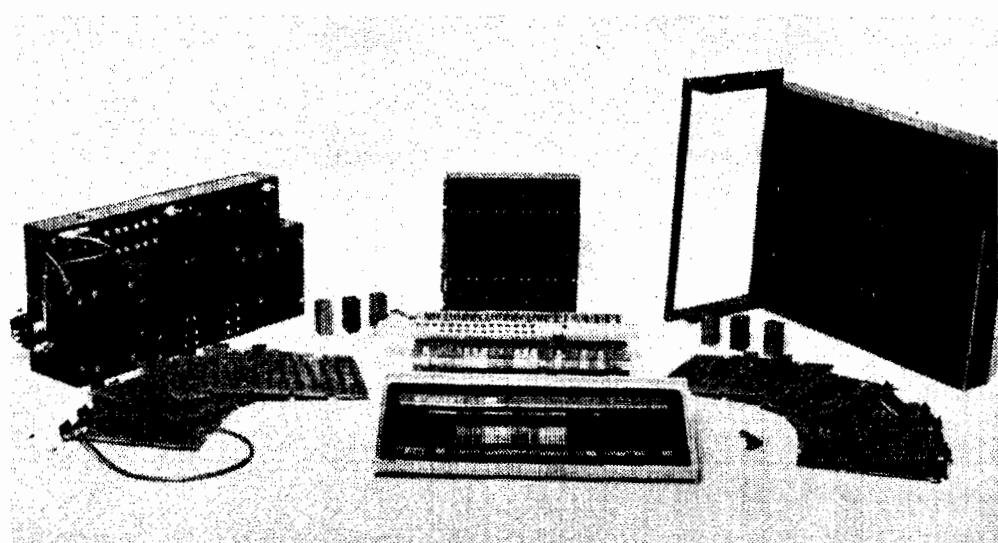
We develop new PDP-8/E software every day. Each new program is exhaustively tested on a PDP-8/E Computer before it is released for customer use. In addition to programs developed for customer use, DEC has developed a special series of diagnostic tests that are used by the various test stations.

#### **(12) Documentation Development**

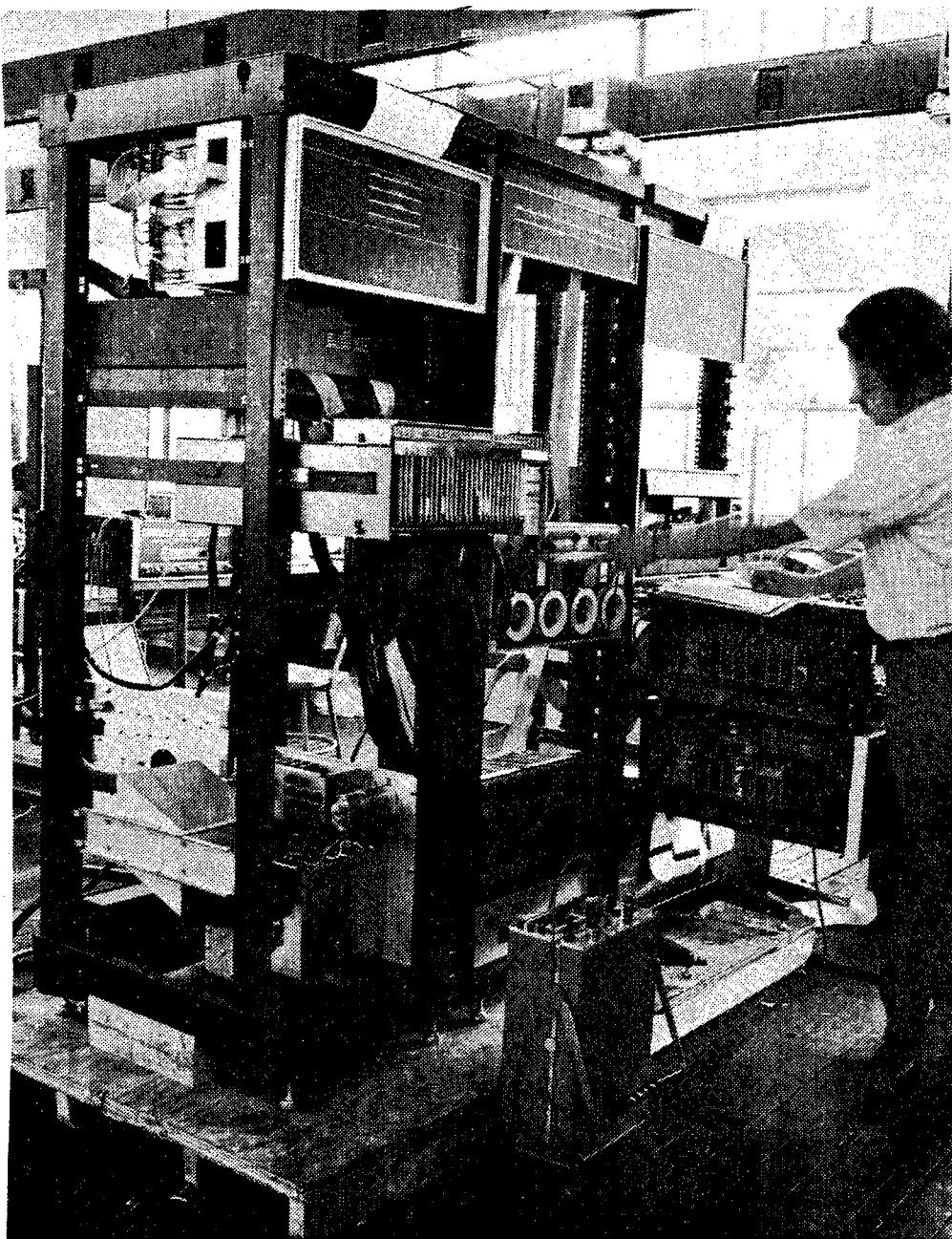
The explosion in computer technology demands the continual development of new computers and peripheral devices. In turn, continuing education for the people who use computers is absolutely necessary. DEC responds to this need for easily assimilated, accurate information by verifying PDP-8/E documentation with both engineering and programming. Our customers are equipped with up-to-date drawings, operating procedures, theory of operation, maintenance procedures, and programming instruction manuals.



### (13) The PDP-8/E Production Line

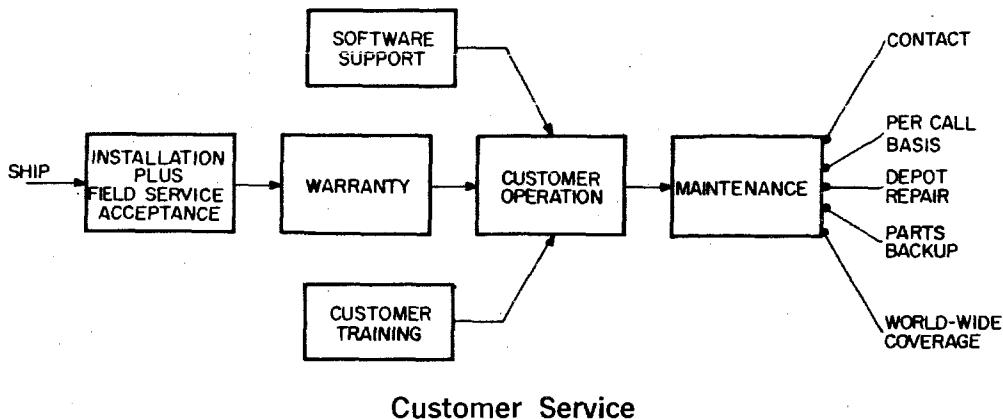


**PDP-8/E System Assembly**—After testing all the various components of a PDP-8/E Computer, the components are carefully assembled. This photo shows all of the components for a basic 4K Computer arranged to illustrate how modular the 8/E is and how spare parts can easily be the key to zero downtime.



### CUSTOMER SERVICE

With the PDP-8/E computer fully checked out and shipped to a user facility, the scene shifts from the factory to the customer. Each PDP-8/E computer or system is installed by DEC's Field Service engineers. Each installation includes system performance checkout using a series of diagnostic programs and other programs to establish successful operation. Each system (depending upon the purchase agreement) is fully backed by a warranty which assures the customer of complete DEC support at no cost for a period of 90 days.



To further support the customer, DEC provides a software support service that assures a complete trouble-free operating software package.

For OEM customers, DEC provides special documentation support on equipment produced by the OEM. DEC will provide a complete system package containing both theory of operation and maintenance.

How to use the PDP-8/E system and how to maintain it is another customer need that DEC satisfies by offering classroom and laboratory instruction designed to familiarize each customer with his system. Courses include programming, hardware familiarization and system familiarization that provides instruction on how to program a system, how to operate a system, how to maintain a system, and detailed knowledge of the system so that a customer may design and build interfaces to the system.

Each customer has the choice of maintaining his own system or employing DEC Field Service to support his system. His option does not stop there; he may elect to purchase a service contract or simply call his local DEC field service to obtain support on a per call basis. DEC support does not terminate; it continues throughout the life of the computer. The second PDP-1 computer system produced by DEC in 1959 has been supported by DEC Field Service for more than 12 years. This service will continue indefinitely.

#### **CUSTOMER TRAINING PROGRAMS**

Digital Equipment Corporation offers an extensive training program to every organization that purchases or presently owns a DEC computer. Our training objective is to familiarize the user with the hardware and software associated with his computer system, and with this in mind, we provide eleven courses for the PDP-8 Family Computers.

**Software:** Five courses ranging from a fundamental Introductory Programming Course to a sophisticated monitor system course. Designed to enable the user to: utilize the standard system software, write his own system programs, incorporate DEC programs as part of his system programs.

**(Programming)**

**Hardware:** Six courses ranging from hardware familiarization to system maintenance. Designed to enable the user to: isolate and evaluate problems if they occur, design interfaces for his system.

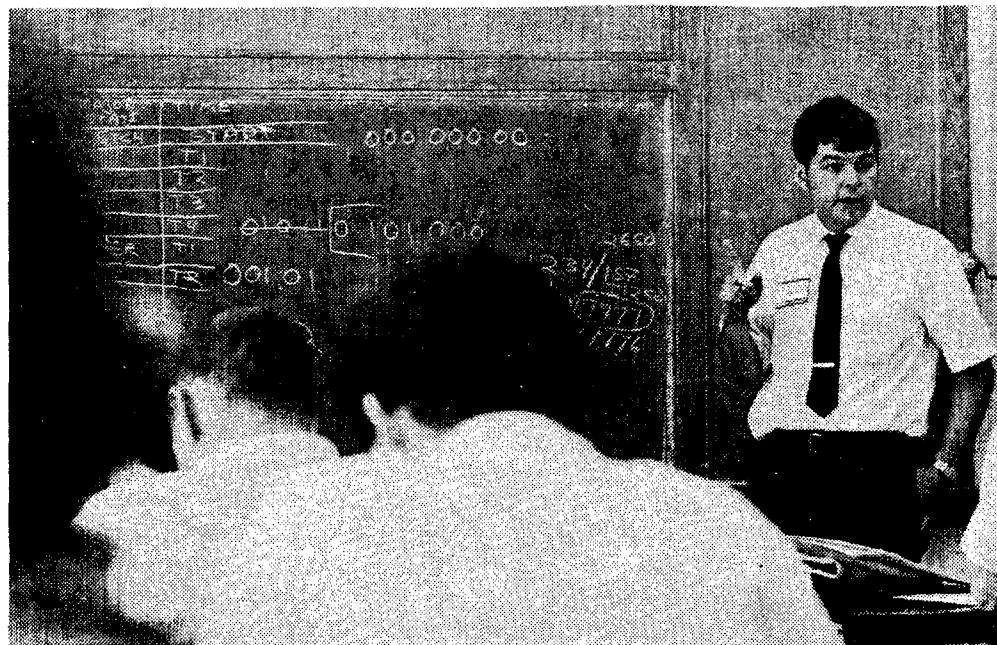
Digital offers training facilities in many countries in the world. We presently have training facilities in Maynard, Massachusetts; Palo Alto; California; Australia; England; France; Germany; and Scandinavia. Our training staff consists of full time professional instructors who continually re-evaluate our courses to ensure the content is current and that it meets the needs of our students. Special Arrangements can be made to conduct courses on-site.

The next few pages illustrate our training environment—from the formal classroom aspect to the lab sessions where the student reinforces his classroom learning with actual programming and debugging time on a computer system.

After completing their training, our students leave with a "can do" outlook. Come and find out for yourself.

For further information about our training program and the scheduling of our courses, check the appropriate block on the information request card in the back of the book.

Each Digital customer is provided the opportunity to familiarize himself with all aspects of our computers and peripheral equipment. Professional class rooms employing the latest techniques are used to train customers to maintain and program the PDP-8/E and peripheral equipment. Well equipped laboratories with a complete array of equipment are employed to assure a high level of confidence of each graduating student. Courses are offered from the beginner level to the more advanced level of instruction.



A hardware class goes through the logic with a timing breakdown.



One of the training laboratories—usually a very busy place.



Happiness is—an assembled, edited program that works.



A peripheral class investigates the inner workings of one of our disk pack units.

## **REPAIR SERVICE**

The key to maintaining your PDP-8 computer system is no further away than your telephone. Digital Equipment Corporation provides 113 service centers throughout the free world employing nearly 1000 trained engineers for repair and a complete range of technical assistance.



This field service engineer is not out to set the world's record on servicing a computer. However, like all field service engineers, he is fast, knowledgeable, professional, and courteous. It is men like him that give Digital Equipment Corporation "high marks" in field service.



#### For Depot Repair Service

Depot repair service save the customer money and time. If you operate on a tight budget . . . or if the DEC products you (or your customers) use are far from our service facilities—Digital's repair depots may be the most economical solution to your maintenance problems.

Depots provide cash-and-carry maintenance and repair service on Teletypes, computers, many standard options and peripherals. You save the cost of a service man's travel time and expense. DEC currently has depots in or near Boston, New York, Chicago, Houston, Los Angeles, San Francisco, Ottawa, Munich, and London. Other services provided at these depots include trading in your old equipment, converting your teletype or punch, etc.

## **MAITENANCE CONTRACTS**

The best method of assuring that your operating system is performing in peak condition all of the time is with a field service contract. With a DEC Field Service Contract, a highly trained engineer or technician will come in at regular intervals and perform carefully planned preventive maintenance to keep your PDP-8/E in top condition. Should your computer go down, you're sure to get prompt, expert service to set it right again. Everything you need to keep up your computer is yours for a fixed monthly charge, whether you need little more than a quick dusting of the keys or a complete overhaul. All contract customers are preferred on a service priority basis.



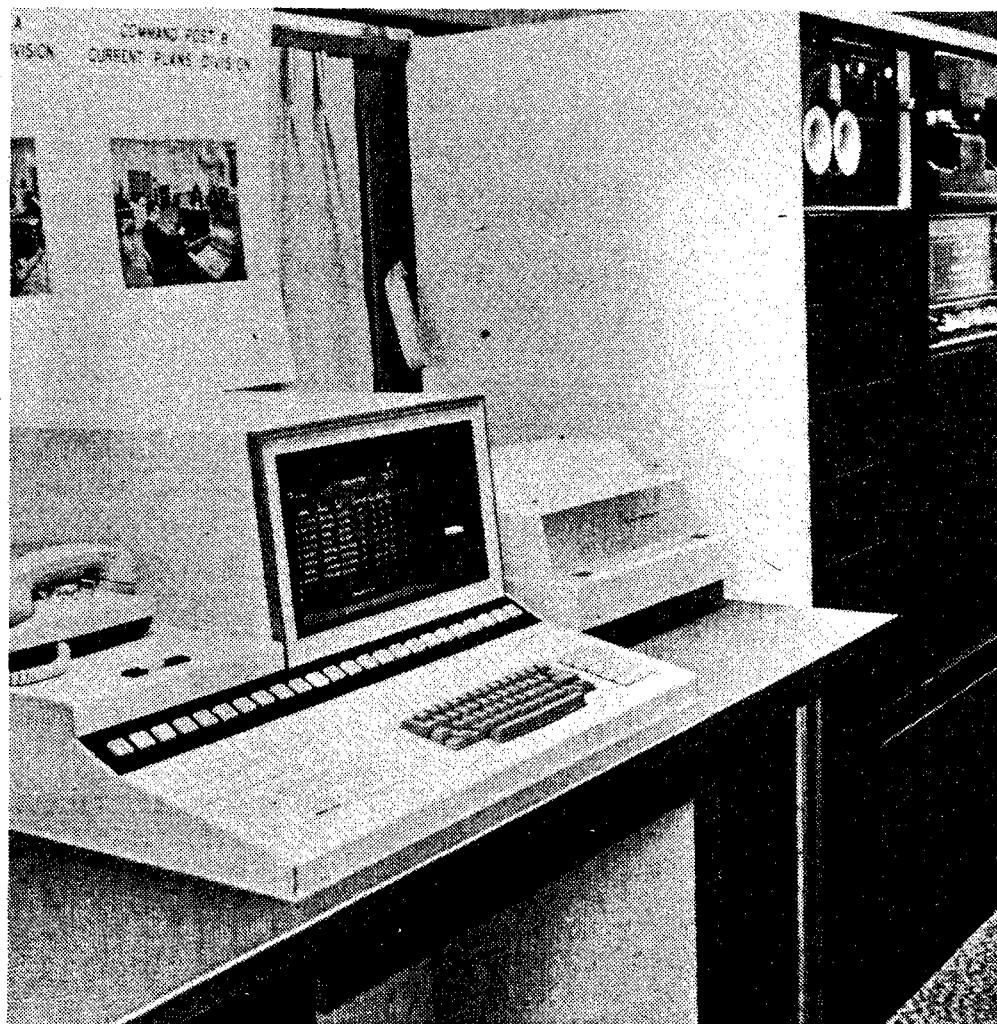
**Customer Service Contracts Guarantee Continuous Operation**

## USER APPLICATIONS

As a result of the interest in the PDP-8 family more than 11,000 PDP-8 computers are installed all over the world. Applications have increased to embrace almost every discipline known to man and new applications are finding their way to the PDP-8 family every day. Representative categories of these computer applications are given below.

### DATA COMMUNICATIONS

- Multiple Data Terminal Systems
- Inter-City Remote Data Systems
- Data Concentrators
- Communications Systems for News Services
- Multi-User Time-Shared Computation
- Message Switching Stations



### Communications

Interior of modified motor van shows equipment that makes up data automation and communications terminal. In the center of the photo is CRT or television-like CRT display and at the right is the small Digital Equipment Corporation PDP-8/I Computer.

## **BIO-MEDICAL**

Environmental Laboratory Response  
Nuclear Medicine  
General Research  
Clinical Laboratories  
Digital Image Processing  
Intensive Care  
Multiphasic Health Screening  
Radiation Treatment Planning  
Cancer Research  
Bone Densitometry



### **Helping Hospitals**

VT05 serves as an excellent information inputting device into an information retrieval system. It is perfect for a hospital environment because the VT05 is quieter than the standard electric typewriter.



### **Healing People**

The first U.S. installation of Digital Equipment Corporation's new Radiotherapy Planning System (RAD-8) is being used at Chicago Wesley Memorial Hospital.

The RAD-8 produces optimized irradiation plans in a fraction of the time required using existing manual techniques. It provides both computerized planning and teaching capability.

## **EDUCATION**

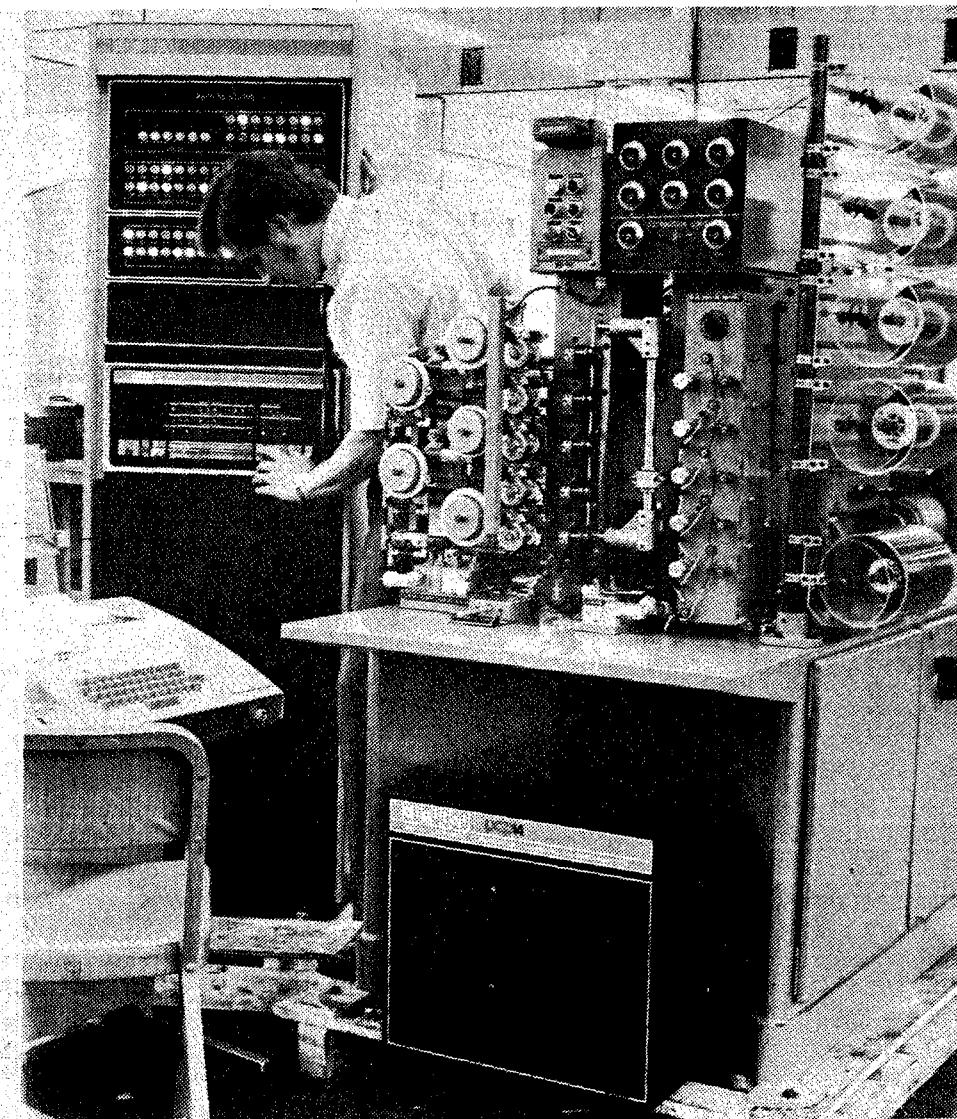
Computer Assisted General Instruction  
Computer Assisted Instruction of Programming Fundamentals  
Computer Controlled Audio-Visual Teaching Facilities  
University Computing Centers



At Centereach, Long Island, two junior high schools and one senior high school in central Long Island have initiated a programmed instruction and individualized progress program for mathematics students. The program has resulted in a significant improvement in academic performance with corresponding reductions in professional staffing. It is anticipated that reductions in staffing will save the school district \$50,000 per year.

## **INDUSTRIAL**

Automatic Drafting  
Automatic Digitizing  
High Speed Plotting, Editing, and Verification  
Automatic Stackers for Warehouses and Stockrooms  
Computer Controlled IC, LSI, and MSI Mask Making  
Concrete Batching  
Paper Mill Control  
Transfer and Assembly Line Control and Monitoring  
Data Acquisition  
Process Control  
Pipe-Line Control  
Engineering Simulations



DEC Control system used at Western Electric to reduce down-time and maintenance.

### **The PDP-14 as a PDP-8/E Peripheral**

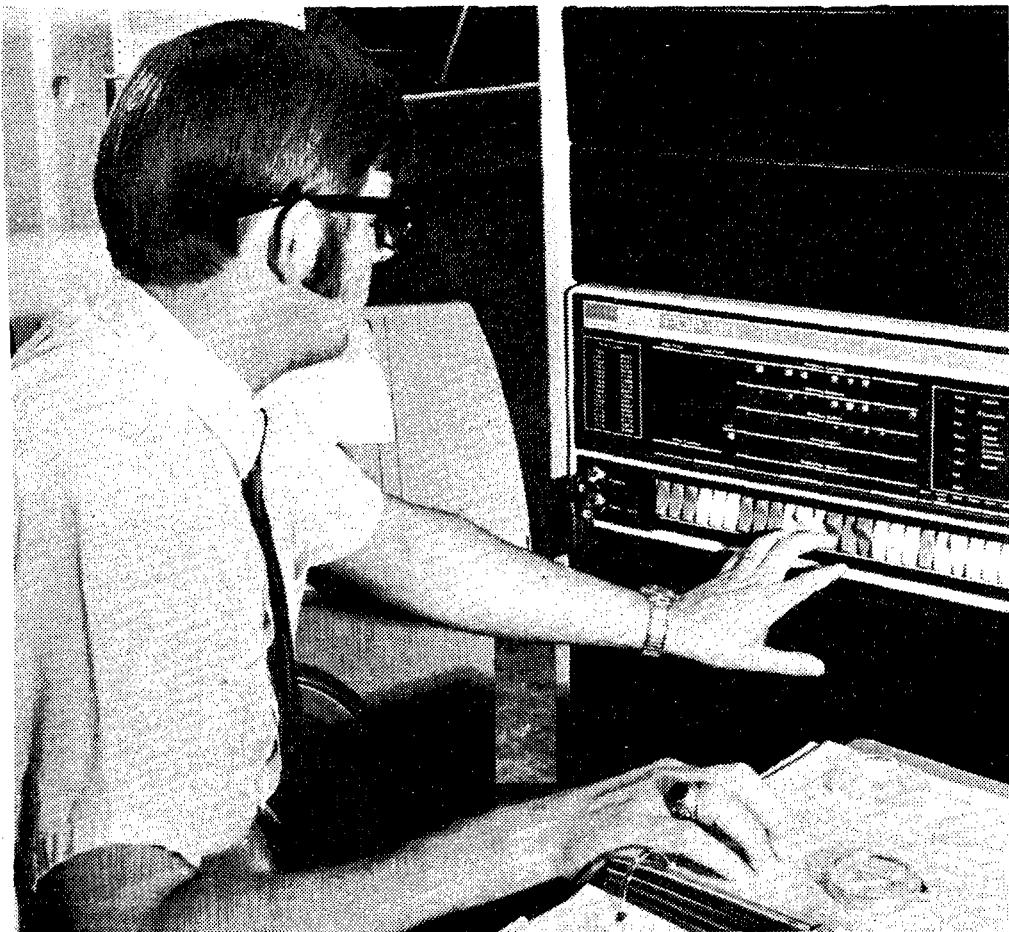
The PDP-14 programmable controller is often used as a peripheral to the PDP-8. In such configurations the PDP-14 provides control functions and/or remote digital inputs and outputs for the PDP-8. The standard PDP-8 to PDP-14 interfaces are described in detail in the PDP-14 Users' Manual order number DEC-14-GGZB-D. Outlined below are some applications for which the PDP-14 is used with a PDP-8 and the general approaches used.

The PDP-14 is a controller for machines or processes which sequences the machine system through its operation by solving control statements. In short, it is a solid state programmable replacement for relay controls. It uses 115 vac inputs (from limit switches, push buttons, etc.) and outputs (to motor starters, solenoids, etc.) as well as DC inputs and outputs if needed. Built into the controller however, is the ability to communicate with the PDP-8 Computer. The PDP-14 can provide the PDP-8 status information regarding the control such as the change state of an input or output. In addition, the PDP-8 can request the current state (ON or OFF) of any input or output connected to the PDP-14. Thus the PDP-14 can do more than just control; it can also function as a peripheral connecting digital I/O to the computer.

**Control and Monitoring**—The PDP-14 is widely used for machine tool control on transfer lines, metal cutting machines, etc. A PDP-8 is often used to monitor the machine's operation by timing the cycle, and recording the number of cycles, the number of parts and other information. The PDP-8 receives its "event signals" from one or more PDP-14's on a program interrupt basis. The PDP-8 can also collect and report management information on the production cycle. The PDP-14 performs the control while the PDP-8 does the monitoring.

**Interactive Control**—Often the PDP-14 is used to control the mechanical operations of a machine or process system but relies on a PDP-8 to make some control decisions, such as what step to do next. The PDP-8 memory holds the "variables" of the control sequence while the PDP-14 holds the constants or repetitive functions. Typical applications include materials handling, in which the PDP-8 instructs the PDP-14 where to move the material; the PDP-14 then moves the material according to other loads in the system, etc. Gauging and measuring equipment are other examples of PDP-8/PDP-14 interactive control: the PDP-14 controls the positioning of the part to be gauged while the PDP-8 performs the measurements and calculations using an analog-to-digital converter. Assembly machines are a further example in which the PDP-8 is used to track rejects through the machine which is controlled by a PDP-14.

Thus the PDP-14 and PDP-8 are well suited partners for sophisticated applications. For further information on PDP-14 and PDP-8 applications consult the PDP-14 Users' Manual or contact your local DEC Office.



### Eyeglasses

AT AMERICAN OPTICAL CORPORATION, Ophthalmic Lens Development Department, a Digital Equipment Corporation PDP-8 small computer is solving complex calculations so that prescription information from the ophthalmologist or optometrist can be translated into instructions used in the eyeglass lens fabrication process. These lenses are required by the Aniseikonic patient, who in many instances, cannot function normally in his study or work without special corrective lenses.

The DEC PDP-8 replaces a programmable desk-type calculator. It is used also as a training device for new engineers and as a "pilot plant" for evaluating new ideas.

Here are some thoughts about the PDP8 from people who see it:

"The PDP-8 permits a much higher degree of reliability than the calculator because fewer inputs and outputs are needed, thus fewer chances of error." "Using the DEC computer, with revised programs, there will be a reduction in the man-hours and machine time required in this special task. The DEC PDP-8 permits an engineer to spend most of his time analyzing intermediate results rather than preparing input and output for the smaller desktop machine which then requires 10 to 15 minutes of computing time. The PDP-8 Computer handles that part of the work in seconds."

## **AEROSPACE**

- Observatory Control
- Telemetry Data Logging
- Radio Astronomy
- Space Probe Instrumentation Calibration
- Lunar Sample Analysis
- Inertial Guidance Checkout

## **GEOPHYSICS/MARINE**

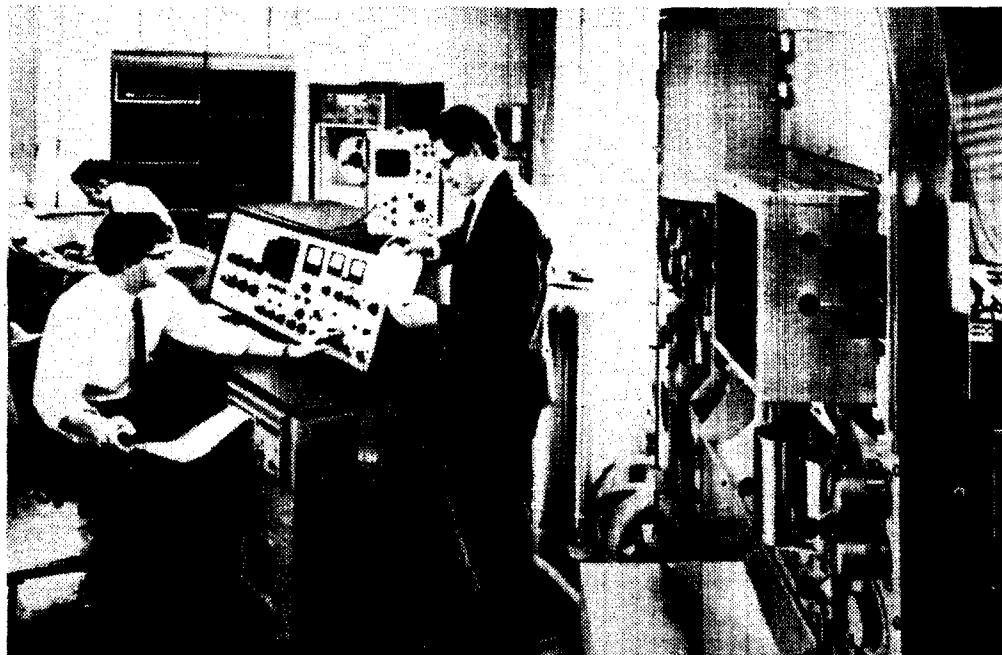
- Oceanographic Data Acquisition
- Shipboard Hydrographic Data Acquisition
- Seismic Monitoring
- Weather Control Research
- Shipboard Navigation Using Satellite Data

## **TRANSPORTATION**

- Automated Transit
- Traffic Control
- Toll Collection and Parking Lot Collection

## **PHYSICS AND CHEMISTRY**

- Gas Chromatography
- Vehicular Emissions Analysis
- Mass Spectrometry
- X-Ray Diffractometry
- Pulse Height Analysis
- Nuclear Magnetics Resonance
- Spectrophotometry



A greater understanding of the composition of natural perfume and flavor materials has been realized by a producer of aromatic chemicals since linking a DEC PDP-8 Computer to their mass spectrometer.

## **ENVIRONMENT**

**Pollution Monitoring  
Contamination Studies**

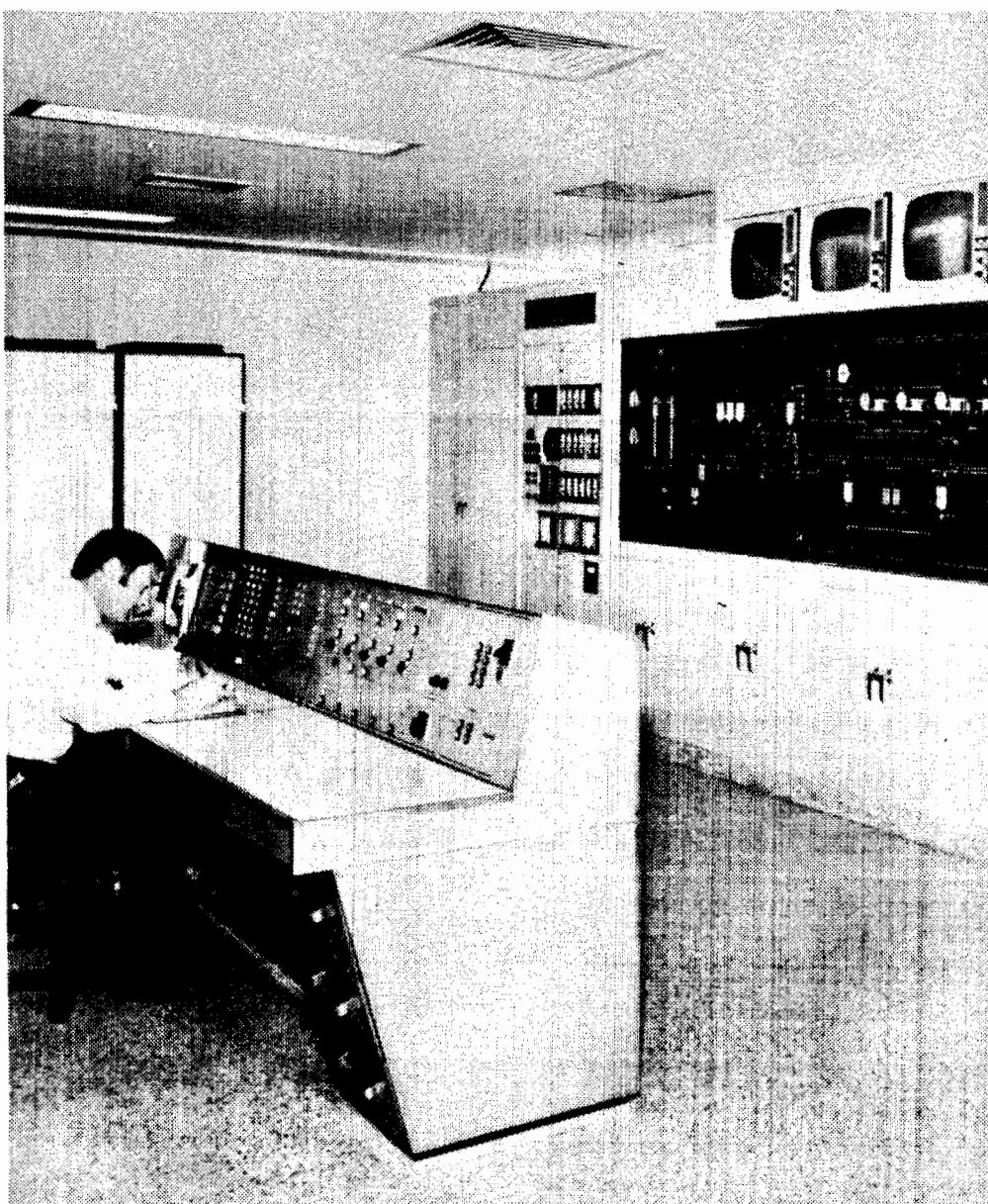


### **Pollution Testing**

Foreign automobiles exported to this country must conform with the 1970 air pollution standards established by the U.S. government. A Digital Equipment Corporation PDP-8 small computer is used to monitor exhaust emission test data. Without this computer, data would have to be recorded manually.

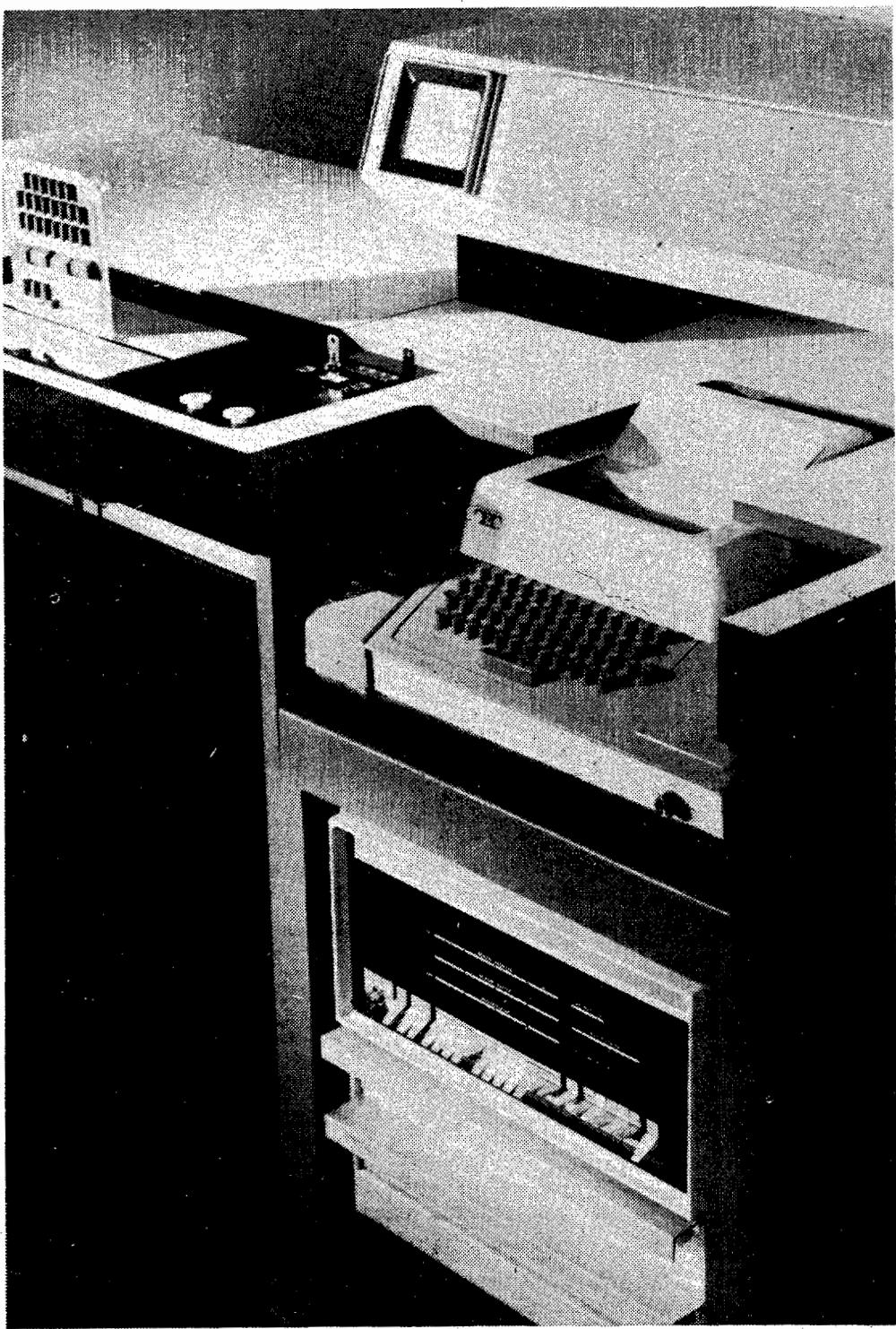
## **INDUSTRIAL QUALITY CONTROL AND TESTING**

Automatic Gauging and Measuring  
Electronic Component Testing  
Cable Capacitance Testing  
Gear Checking  
Noise Measurement



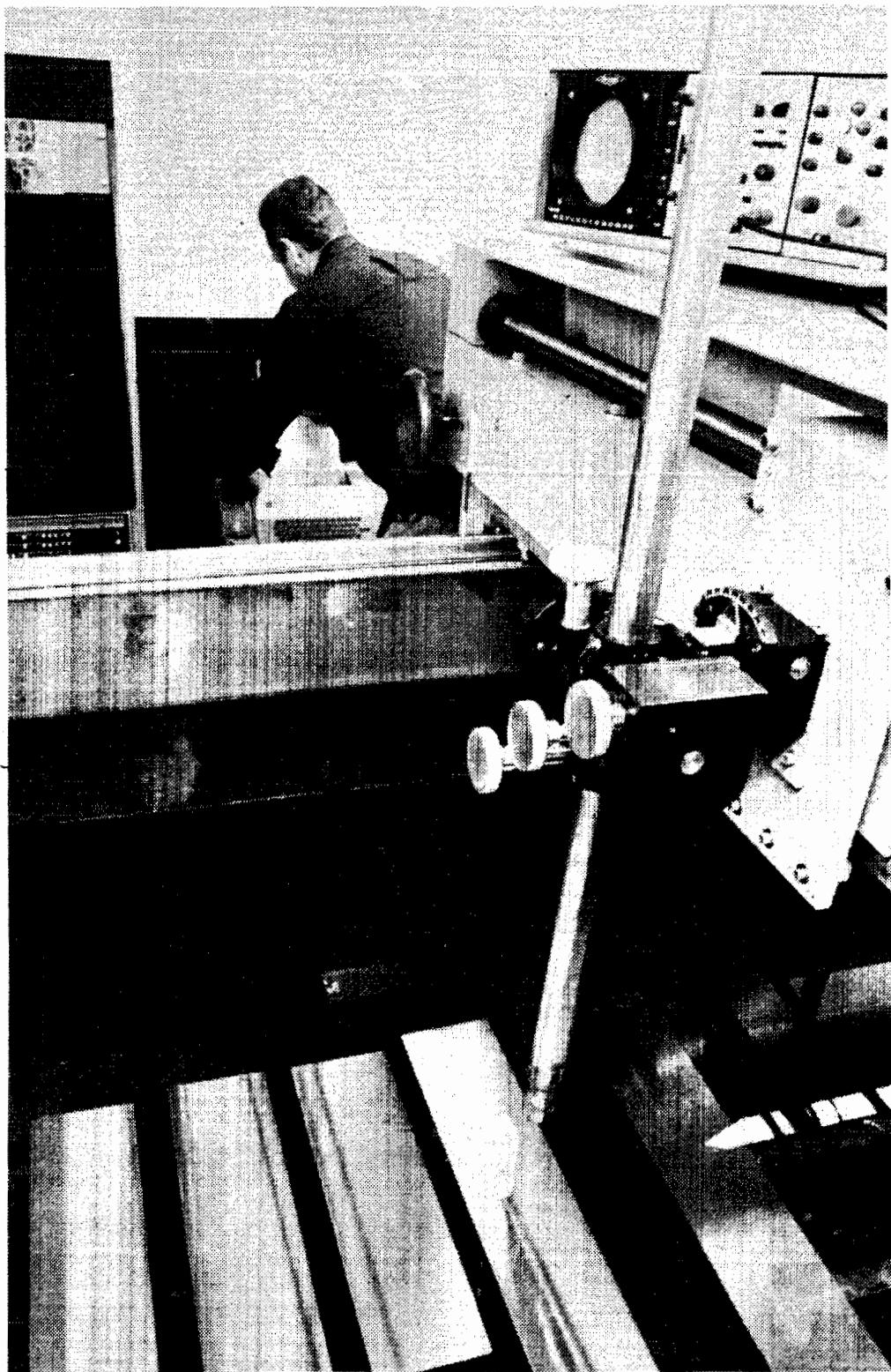
### **Down Under**

In SYDNEY, Australia—Tons of gelatine, for table desserts, photographic emulsions, and pharmaceutical tablets, are produced each day at Davis Gelatine's new extraction facility here. The entire process is monitored from this control room by a Digital Equipment Corporation PDP-8 Computer located in the left background.



**Logic Circuit Tester—**

An OEM Logic Circuit Analyzer, using a Digital Equipment Corporation PDP-8 Computer, tests and diagnoses complex logic circuits on large-scale arrays or printed circuit boards. The system is capable of performing up to 4,000 tests per second on devices with as many as 240 pins.



PDP-8 is used here as part of a quality inspection system for checking the composition of cast steel.

## **BUSINESS AND COMMERCE**

Accounting and Record Keeping  
Business Systems for Newspapers  
Mailing Systems  
Inventory Control  
Automatic Typing  
Management Information  
Information Systems for Brokers  
Reservation Systems  
Warehouse Automated Storage and Retrieval  
Commodity Market Information Systems  
Fast Food Industry



### **Buy or Sell?**

Message Concentrator in the Vancouver, B.C., Stock Exchange Instant Quotation Service is a PDP-8 small computer manufactured by Digital Equipment of Canada, Ltd. Service allows brokers to get information from the exchange, one of three in Canada, via the Telex equipment in their offices, rather than having to telephone the exchange or rely on the ticker tape.

## **GRAPHIC ARTS**

Newspaper Typesetting  
Commercial Typesetting

## **NUMERICAL CONTROL**

Direct Computer Control of Machine Tools  
NC Tape Preparation for Machine Tools  
Digitizing Systems for NC Machine Control

## **CRYPTOGRAPHY**

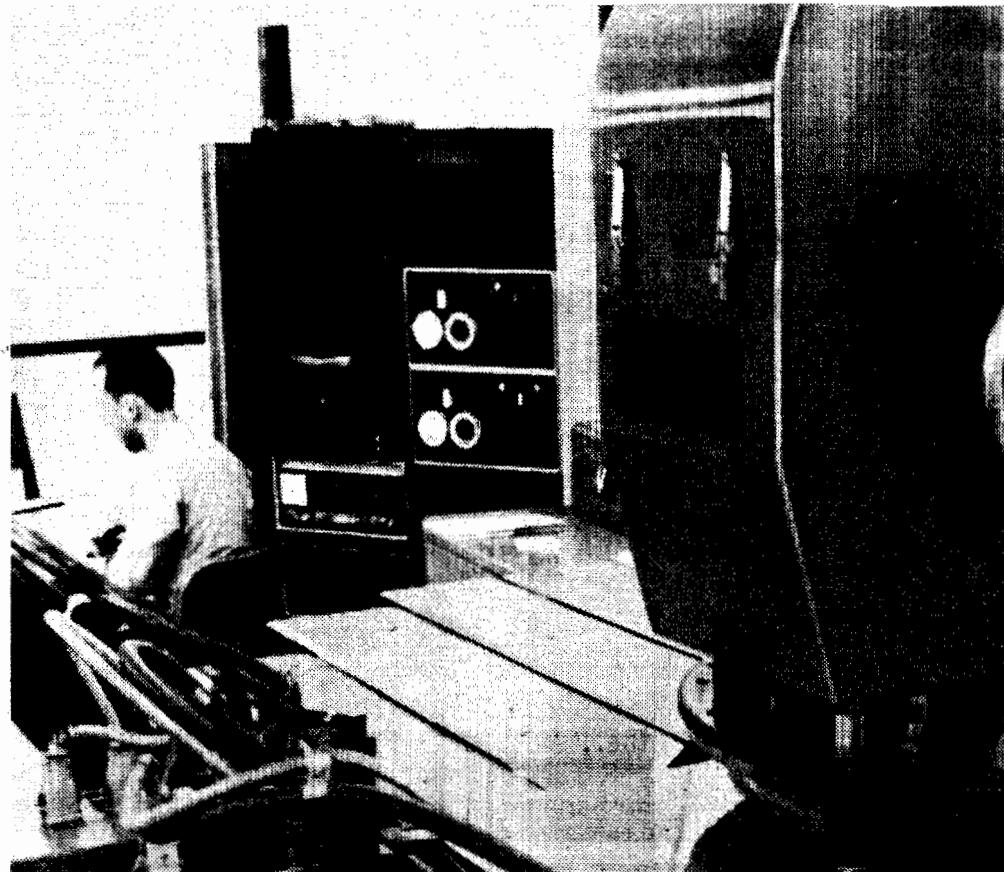
Cryptographic Analysis  
Code Generation

## **ENTERTAINMENT**

Computer Animated Motion Pictures  
Television Station Control  
Scoreboard and Toteboard Display  
Museum Exhibit Control  
Planetarium Control

## **ENGINEERING**

Engineering Calculation  
Ship Model Design  
Computer Aided Design  
Hybrid Simulation



PITTSBURGH, Pa.—At the new home of the Pirates baseball team, Three Rivers Stadium, a huge million-dollar information display scoreboard dominates centerfield. The scoreboard, towering 30 feet high, and 274 feet long, lets the baseball fans enjoy the game more and understand it better. The scoreboard is controlled by a Digital Equipment Corporation PDP-8 Computer.

The DEC PDP-8, located in the press box area, allows a non-technically oriented operator to manipulate a series of push buttons on a keyboard console. These buttons cause words to move up, down, or laterally; to expand or contract; and even rotate via a series of static and moving light displays.

The scoreboard—relying on techniques from the movie industry—also uses cartoon animation. Sequential “action” sketches of the cartoon figures are fed into a scanner, such as used in facsimile systems. The scanner translates the image into the PDP-8’s computer language, which then passes it through a decoding unit to be converted, so that the resulting signal is then transmitted directly to the display, or stored on magnetic disk memories for later playback.



DEC computer controls huge million-dollar scoreboard at Pittsburgh Pirates new Three Rivers baseball stadium

## PDP-8/E FUNCTIONAL CHARACTERISTICS

TYPE:	Single address, fixed word length, parallel transfer programmed data processor
WORD LENGTH:	12 bits
CYCLE TIME:	1.2 or 1.4 microseconds
MEMORY CAPACITY:	4096 or 8192 words, expandable to 32K 8 autoindex registers per 4K memory field
STORAGE MODE:	Two's complement numbers, 6-bit or ASCII characters
ADDRESSING CAPABILITY:	Typically, one instruction may address 256 locations directly or 4096 locations indirectly
INSTRUCTION SET:	6 memory reference instructions, 20 microprogrammable operate microinstructions, and 8 input/output transfer instructions for the CPU and each of up to 63 I/O devices
INSTRUCTION EXECUTION TIME:	Operate microinstruction: 1.2 microseconds Directly addressed MRI: 2.6 microseconds Indirectly addressed MRI: 3.8 microseconds Autoindexed MRI: 4.0 microseconds or less
INPUT/OUTPUT CAPABILITY:	Programmed data transfer, program interrupt system transfer, and 13 channels of internal and/or external direct memory access (data break)
SIZE AND WEIGHT:	Typically 19 x 10.5 x 24 inches (48 x 26 x 61 centimeters) at about 95 pounds (43 kilograms)
OPERATING ENVIRONMENT:	Ambient temperature 32° to 130° Fahrenheit 0° to 55° Centigrade Relative humidity 100% to 90% (noncondensing)
POWER REQUIREMENT:	Typically 150 Watts at 115 VAC, 60 Hz or 230 VAC, 50 Hz

# 2

## system introduction

### PDP-8/E BASIC SYSTEM

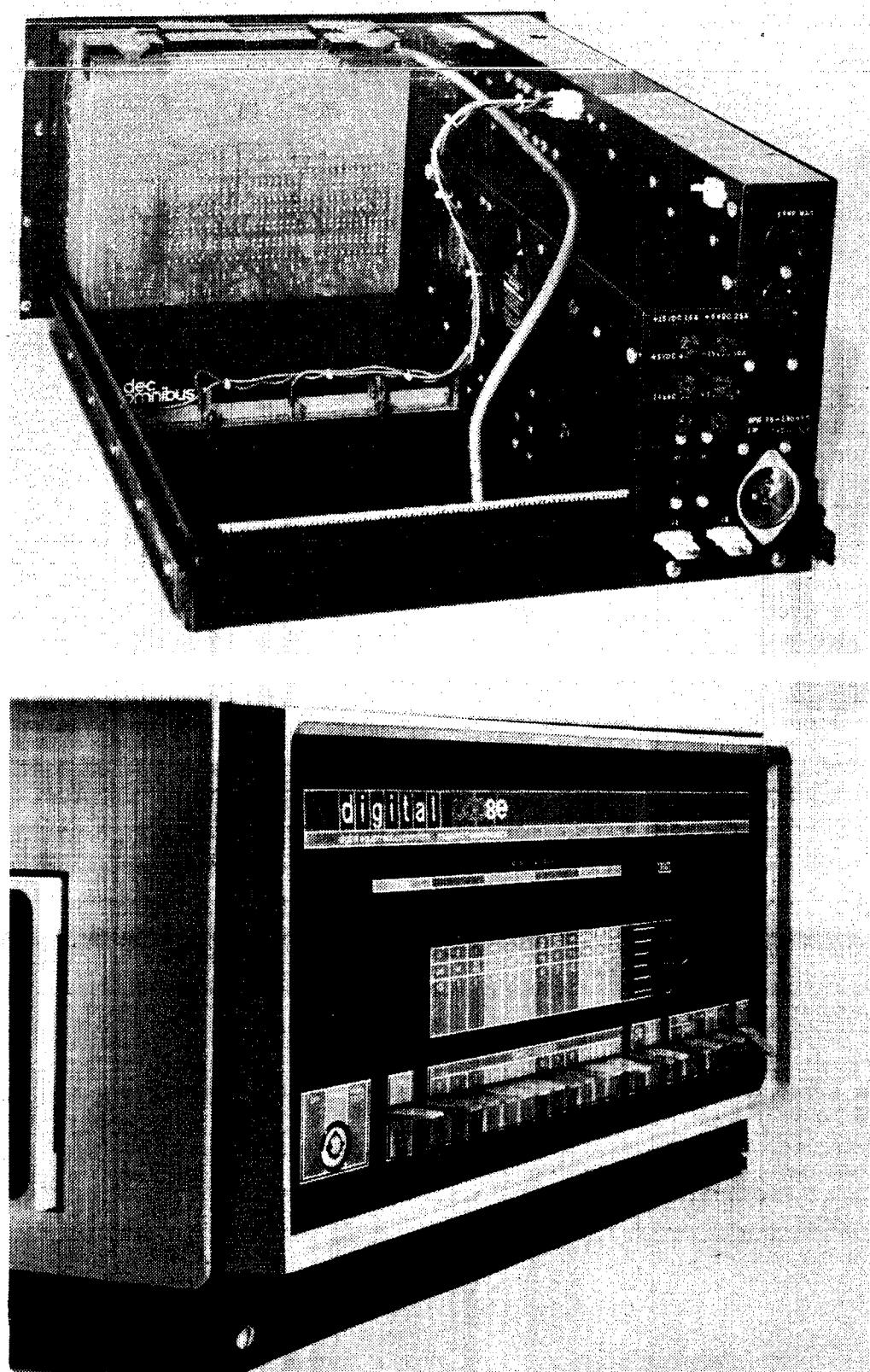
The PDP-8/E is specially designed as a general purpose computer. Its development is the successful culmination of many years of computer design research directed toward providing better computers at the lowest possible price. The PDP-8/E is designed to meet the needs of the average user, yet it is capable of modular expansion to accommodate almost any requirements for a user's specific application.

The PDP-8/E basic processor is a single-address, fixed word length, parallel transfer computer using 12-bit, two's complement arithmetic. The cycle time of the random access memory is 1.2 microseconds for fetch and defer cycles without autoindexing and 1.4 microseconds for all other cycles. Standard features include indirect addressing and facilities for instruction and skipping and program interrupts as a function of input/output device conditions.

Five 12-bit registers are used to control computer operations, address memory, perform arithmetic or logical operations and store data. A programmer's console provides switches and indicators that permit convenient monitoring and modification of machine states and major registers. The PDP-8/E may be programmed manually, using the programmer's console, or remotely, by means of a console terminal.

The 1.2/1.4 microsecond cycle time of the PDP-8/E provides a computation rate of 385,000 additions per second. Each addition requires 2.6 microseconds (with the addend in the accumulator), while subtraction requires 5.0 microseconds (with the subtrahend in the accumulator). Multiplication is performed in 256.5 microseconds or less by a subroutine that operates on two signed, 12-bit numbers to produce a 24-bit product, leaving the 12 most significant bits in the accumulator. Division of the two signed, 12-bit numbers is performed in 342.4 microseconds or less by a subroutine that produces a 12-bit quotient in the accumulator and a 12-bit remainder in memory. Similar signed multiplication and division operations are performed in approximately 40 microseconds utilizing the optional KE8-E Extended Arithmetic Element.

The flexible, high capacity input/output capabilities of the PDP-8/E allow it to operate a variety of peripheral devices. Besides a choice of console terminals, the PDP-8/E supports more than 60 input/output device options including high-speed paper tape equipment, card readers, line printers, disk and magnetic tape bulk storage devices and a wide range of data acquisition, transmission and display peripherals.



PDP-8/E Programmed Data Processor  
(Table-top Model)

Every PDP-8/E system is completely self-contained. A single source of 115 or 230 volt AC power is required; however, internal power supplies produce all necessary operating voltages for the system. Rack mounted computers are supplied with standard cabinets that are large enough to accommodate the PDP-8/E and several peripherals in less than 5 square feet (0.5 square meters) of floor space. The table top version is a convenient alternative for users who plan to install the computer in a confined area, such as an office. The table top PDP-8/E weighs only 100 pounds (45 kilograms) and displaces less than 3 cubic feet (0.8 cubic meters).

The basic PDP-8/E computer consists of a table top or rack mounted cabinet with an H274 (or H274-A) power supply, and an OMNIBUS on which the KK8-E Central Processor, memory system, programmer's console and console terminal control are mounted. In the PDP-8/E, a bus is defined as a group of 12 signal lines carrying related information, such as the 12 bits of an instruction or data word. The OMNIBUS may be considered as a wide bus containing several busses, along with many other signal lines. Each OMNIBUS contains 20 identical, non-dedicated module slots, and each slot will accept a 144-pin QUAD-size module. The OMNIBUS provides a two-way signal path between corresponding pins of the modules that are plugged into it.

The PDP-8E central processor consists of five QUAD modules that plug directly into the OMNIBUS. The memory system is contained on an additional three QUAD modules, while the programmer's console and console terminal control occupy one module each. Figure 2-1 is a block diagram of the basic PDP-8/E that illustrates the signal paths between the central processor, the memory system and the OMNIBUS. Signals that do not pass through the OMNIBUS are routed between adjacent modules by means of H851 Edge Connectors.

#### **KK8-E CENTRAL PROCESSING UNIT**

The KK8-E Central Processor consists of the major registers module, major registers control module, timing generator, bus loads module and RFI shield. These five functional units contain most of the timing and gating circuitry used to manipulate data and generate control signals.

#### **M3800 MAJOR REGISTERS MODULE**

The major registers module contains five special purpose registers that are used in almost every programming application, as well as additional gating circuits and a 12-bit parallel adder. These components are described separately in the following paragraphs.

##### **Accumulator (AC)**

The accumulator, or AC, is a 12-bit register in which arithmetic and logical operations are performed. The accumulator may be cleared, complemented or incremented under program control, and its contents may be rotated right or left. The content of the memory buffer register may be combined with the content of the AC by two's complement addition or by a bitwise logical AND operation. The content of the programmer's console switch register may be combined with the content of the AC by a bitwise logical OR operation. In every case, the result is left in the AC.

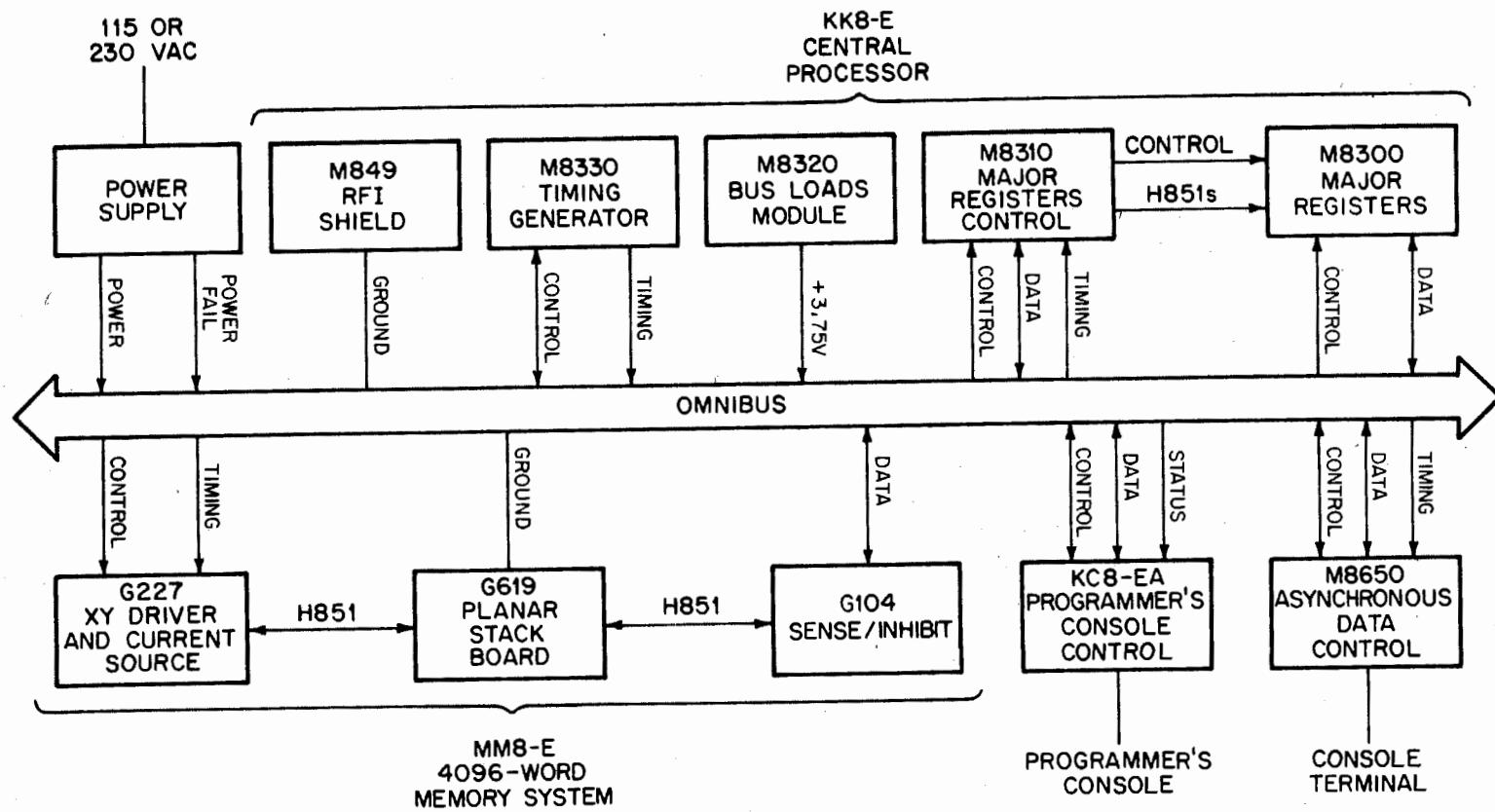


Figure 2-1 PDP-8/E Basic System Block Diagram

The AC may also serve as an input/output register. All programmed data transfers between memory and I/O devices pass through the AC to data lines located on the OMNIBUS. I/O transfers performed via data breaks, or direct memory access, do not pass through the AC, however.

#### **Multiplier Quotient Register (MQ)**

The multiplier quotient registers, or MQ, is a 12-bit bidirectional shift register that acts as an extension of the AC during extended arithmetic operations. When a KE8-E Extended Arithmetic Element is installed, the MQ contains the multiplier at the beginning of a multiplication and the least significant half of the product at the conclusion. It contains the least significant half of the dividend at the start of a division and the quotient at the conclusion, or the least significant part of a number during shift and normalize operations. The MQ is available as a temporary storage register, even if a KE8-E is not installed.

#### **Program Counter (PC)**

The program counter, or PC, is a 12-bit register that contains the address of the memory location from which the next instruction will be taken. The PC is automatically incremented by 1 after each instruction is read from memory. It may be incremented under program control, to conditionally skip the next sequential instruction, or loaded from the memory buffer register, to cause a programmed jump to a prescribed memory location.

#### **Central Processor Memory Address Register (CPMA)**

The CPMA is a 12-bit register that contains the memory address currently selected for reading or writing. This register is never cleared; information is always jam transferred in and the original content is lost. The CPMA may be loaded from the memory buffer register, the program counter or the programmer's console switch register. Extended memory and data break interfaces provide additional memory addressing capabilities.

#### **Memory Buffer Register (MB)**

The MB is a 12-bit register through which all information is transferred between the central processor registers and memory. Data may be read into the MB from any memory location in 0.6 microseconds and rewritten at the same location in another 0.6 microseconds. The content of any location may be read, incremented, tested and rewritten in the same location in a total of 3.8 microseconds or less. The MB may be loaded from either the AC, the PC or memory.

#### **Data Gates and Adders**

The major registers module also contains the gating circuitry necessary to move data from one register to another. At the heart of the data gating circuitry is a 12-bit parallel adder. Information from a register is gated to the adder inputs. The output of the adder is applied to a set of shift gates, and the output of the shift gates serves as data input to all of the major registers.

### **M8310 MAJOR REGISTERS CONTROL MODULE**

The major registers control module contains the link, the major register control circuits, the major state generator and the instruction register, as well as additional miscellaneous control circuits. This circuitry is responsible for the actual decoding and execution of most PDP-8/E instructions. Control signals are transmitted between the major registers control module and the major registers module by means of two H851 Edge Connectors. Important components of the major registers control module are described separately in the following paragraphs.

#### **Link (L)**

The link is a 1-bit register that serves as a high-order extension of the AC. It is used as a carry register for two's complement arithmetic. The link may be set, cleared or complemented under program control. At the same time, it may also be rotated left or right as part of the accumulator.

#### **Major Register Control Circuits**

The major register control circuits gate timing, data and control signals to enable the adder input and shift gates of the major registers module. They also gate timing pulses that regulate data transfers to and from the major registers.

#### **Major State Generator**

The major state generator determines which of three major states the central processor is about to enter. Each major state corresponds to a signal that is asserted to enable the appropriate register control circuitry. A fourth major state is entered when none of the signals produced by the major state generator are asserted. Specifying one of the four major states determines which data gating circuits will be enabled during a given memory cycle.

#### **Instruction Register (IR)**

The IR is a 3-bit register that contains the operation code of the instruction that is currently being executed. The three most significant bits of each instruction are loaded into the IR after the instruction is read from memory. This data is decoded and used to determine which major states will be entered during instruction execution.

### **M8320 BUS LOADS MODULE**

The bus loads module receives +5 and +15 volt inputs from the power supply and provides +3.75 volt (voltage level high) output to load the OMNIBUS signal lines. Most signal lines are considered to be inactive until the voltage level is pulled to ground by a component that is asserting the line.

### **M8330 TIMING GENERATOR MODULE**

The timing generator module contains the time pulse generator, interrupt control circuits, the processor input/output transfer instruction decoder and other miscellaneous control circuits.

The time pulse generator provides four time states, designated TS1 through TS4, and four time pulses, designated TP1 through TP4. Each time pulse overlaps the end of one time state and the beginning of the

following time state. The time states are used to initiate sequential, time-synchronized gating operations. The time pulses are used for memory timing and as gating pulses throughout the system. In addition, the power clear pulse generator produces pulses that reset registers and control circuits during power turn-on and turn-off. Several of these pulses are available for the control of peripheral devices.

The interrupt control circuits comprise the major portion of the interrupt system. This circuitry responds whenever an interrupt request signal is received from an interface controller module. Processor input/output transfer instructions are used to initialize and operate the interrupt system under program control.

#### **M849 RFI SHIELD MODULE**

The radio frequency interferences (RFI) shield module ensures that signals which are not synchronized with memory do not interfere with the memory circuits. Aside from a ground path, the RFI shield has no important connections to or from the OMNIBUS.

#### **KC8-EA PROGRAMMER'S CONSOLE**

The programmer's console module contains the circuitry required to operate the PDP-8/E programmer's console. This console consists of an array of controls and indicators that facilitate computer operation and maintenance. Twenty-two switches provide convenient control of the system by allowing the operator to start and stop program execution, examine and modify the content of memory, select various modes of operation, or load and execute short machine language programs.

A 6-position rotary switch selects one of six registers or groups of registers for display in 12 bits of the 28-lamp indicator panel. A lighted indicator lamp indicates the presence of a binary 1 in the specified bit position of a register or control flip-flop. The 15-bit address of the memory location being accessed and the state of the RUN flip-flop are displayed at all times.

A 3-position key operated switch permits the computer to be locked in a power off state, a power on state with all switches and indicators activated, or a power on state with only the SW switch and RUN indicator activated. This feature serves to protect a running program from inadvertent switch or control operation.

#### **M8650 ASYNCHRONOUS DATA CONTROL MODULE**

The KL8-E Asynchronous Data Control, consisting of one M8650 module, contains the receive, transmit and control circuitry needed to interface an LT33 or LT35 Teletype terminal, VT05 DECTerminal, or any similar asynchronous device with the central processor. This module serves as a serial-to-parallel converter for transmitting input signals, or a parallel-to-serial converter for transmitting output signals. It also performs certain control functions such as instruction skipping as a function of terminal condition and transfer of program control via program interrupt.

Eight models of the KL8-E provide a variety of transmit/receive rates ranging from 110 baud to 1200 baud. The 110 baud model is available with a choice of cabling for EIA/CCITT or 20 mA operation, while all

other models differ only in jumper connections, which may be changed in the field to vary the transmission rate, the receive rate, or both.

The asynchronous data control is also provided with split plugs that permit it to be assigned any two user-designated device codes. In this manner, up to 17 KL8-E Asynchronous Data Control interfaces may be installed on a single PDP-8/E to provide multiple terminal capability.

#### **MEMORY SYSTEM**

The basic PDP-8/E memory system is a 4096-word (4K) or 8192-word (8K), 12-bit random access memory that performs all normal functions of data storage and retrieval. Additional units of 4K memory or 8K memory, consisting of 3 quad modules each, may be used as extended memory to increase memory capacity up to 32K.

Programs that used the interrupt system must reserve memory locations 0000 and 0001 in the first 4K of memory, while locations 00010 through 00017 (octal) of every 4K memory unit are used as autoindex registers. All other memory locations are available to the programmer for storage of either instructions or data.

The MM8-E Core Memory System consists of a planar stack board, XY driver and current source module, and sense inhibit module. The three memory system modules contain special circuits such as read/write switches, address decoders, inhibit drivers and sense amplifiers. These circuits perform the operations necessary to transfer data into or out of the core array. They do not perform arithmetic or logical operations on the data.

#### **G227 XY Driver and Current Source Module**

This module contains the circuitry that decodes data from the memory address lines and drives the XY wires of the core array. The circuits include selection switches, XY current sources, address decoder, stack discharge switch and power on/off write protection. The XY currents are controlled remotely from the sense/inhibit board, so that the module does not require any adjustments. The XY driver and current source module is also used in the memory parity option.

#### **G619 Planar Stack Board**

The basic 4K memory stack consists of 4096 12-bit words of memory, along with the X-axis and Y-axis diode selection matrix. It also contains a thermistor that supplies temperature information to the XY current control circuit. The core array is a 3-dimensional, 3-wire memory with center tapped sense/inhibit wire. The same memory stack module is also used in the memory parity option. It has no important connections to or from the OMNIBUS.

#### **G104 Sense/Inhibit Module**

The sense/inhibit module contains 12 sense amplifiers, inhibit drivers and memory registers. It also includes the slice control, the -6 volt sense amplifier power supply, the current control for the XY current source, strobe and clear control logic, and the field selector, which is used in the sense/inhibit module as well as in the XY driver. The field selector samples 3 jumper connections to determine the memory field

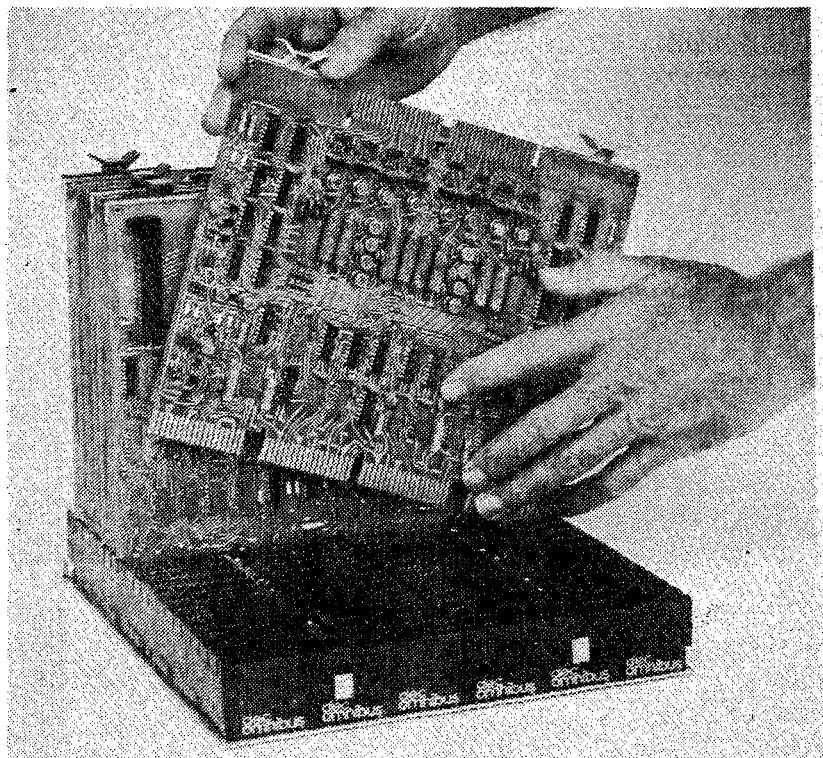
being accessed. Slice level, strobe delay and XY current may be selected within 4 discrete steps by means of additional jumper connections (two per axis). The proper combination is predetermined for any given stack, and appropriate jumper connections may be selected to eliminate the need for adjustments in the system.

### **INTERFACING**

The PDP-8/E OMNIBUS functions as to an internal input/output bus which was designed to eliminate wiring and provide convenient access to data and control signals.

The KA8-E Positive I/O Bus Interface provides an extension of the bus system that facilitates interfacing PDP-8 family positive bus equipment with the PDP-8/E. The positive I/O bus was specially designed for use with PDP-8/I and PDP-8/L compatible peripherals, but it may be employed with almost any positive bus equipment.

PDP-8/E systems provide three types of data transfer: programmed data transfers, program interrupt transfers and direct memory access transfers. Programmed data transfer is the easiest and most direct method of handling data I/O. Program interrupt transfers provide an extension of programmed I/O capabilities by allowing I/O operations involving two or more devices to be performed concurrently. The data break system uses direct memory access for applications involving extremely fast data transfer rates. All three I/O techniques are described in Chapter 4 of this handbook.



The OMNIBUS eliminates back plane wiring and provides access to 144 data and control signals. Interfacing is accomplished by inserting modules into the non-dedicated slots.



PDP-8/M Programmed Data Processor

#### **DIFFERENCES BETWEEN THE PDP-8/E AND ITS PREDECESSORS**

As new members of a computer family are developed, differences between them and their predecessors inevitably result. In most cases, these differences are either benign or beneficial. All major differences between the PDP-8/E and its predecessors are listed in Table 2-1, in order to give users of earlier machines a concise summary.

**Table 2-1 New PDP-8/E Instructions**

OCTAL CODE	NEW INSTRUCTION (MNEMONIC)	PREVIOUS FUNCTION
6000	SKON	NOP
6003	SRQ	ION
6004	GIF	ADC or NOP
6005	RTF	ION (ORed with ADC)
6006	SGT	IOF (ORed with ADC)
6007	CAF	ION (ORed with ADC)
7002	BSW	NOP
7014	Reserved	RAR RAL
7016	Reserved	RTR RTL
74X1	MQ Instructions	Only available with EAE.
7521	SWP	MQL MQA

Octal codes 7014 and 7016 produced predictable but undocumented results in the PDP-8/I and PDP-8/L. In the PDP-8/E these codes are specifically reserved for future expansion.

The PDP-8/E console terminal uses the same IOT instructions as earlier machines, but also recognizes additional IOTs. The skip IOTs may no longer be microprogrammed with other console terminal IOTs. This should impose no constraint on the user, since it is generally undesirable to combine skip IOTs with other IOTs. Reader Run is no longer set by INITIALIZE, so that any routine using the low-speed paper tape reader must begin with a KCC instruction.

In general, the signals and functions at the external I/O bus interface are the same as for previous machines. Users who constructed peripherals using previous editions of the *Small Computer Handbook* as a guide may expect their peripherals to work on the PDP-8/E. Note, however, that the PDP-8/E is equipped only with a positive bus, and a DW08-A bus converter is needed to interface with some older, negative bus equipment.

The BAC lines at the external I/O bus interface are merely the buffered DATA bits of the OMNIBUS. Since the DATA lines are used for bidirectional transfer, any input of data at the external I/O bus interface will cause an immediate change at the BAC outputs. Simultaneous input and output transfers in the same IOP should be checked; in such situations, the register in the peripheral must be edge-triggered.

At the conclusion of the IOP dialogue, the DATA bus is used for updating the PC, and then for determining break or API priority. Users may no longer rely on the BAC lines being available until the end of the current major state. However, IOP width and separation may be adjusted, if desired, to accommodate slow I/O devices. External IOIs are faster in all cases except for IOIs ending in 7.

The extended arithmetic element has been redesigned, and several powerful features have been added. Previous EAE users may expect their programs to run without modification on the new EAE, but it is wise to recode EAE programs to make use of the new SAM, DCM, DAD, DST, DPIC and DPSZ instructions.

The time required to access the data break system has been greatly reduced. Maximum benefit will be obtained on machines without an EAE and with only internal options or options that are not activated while data break is in use. An added feature, ADM, permits the user to add an input word to the content of a memory location. An internal multiplexing scheme allows the use of up to 12 external and/or internal direct memory access devices.

The programmer's console of the PDP-8/E differs from the control panels of its predecessors in the following respects:

1. Only the MA, EMA and RUN indicators are permanently displayed. All other register displays are selected by means of a 6-position rotary switch.
2. Machine stops occur after TP4. Thus, the MA lamps indicate the next memory address to be accessed, while the major state indicator lamps show the next major state to be executed.

3. Extended memory field information is entered manually by means of switch register bits 6-11 and the EXTD ADDR LOAD switch.
4. Operation of the ADDR LOAD switch places the central processor in the FETCH major state.
5. Program execution is started by operating and releasing the CLEAR switch, then operating and releasing the CONTINUE switch.
6. Turning the POWER switch to the PANEL LOCK position extinguishes all indicator lamps except for the run indicator.

Any attempt to deposit data in a non-existent memory field will not stop the machine. An attempt to read data from non-existent memory yields a zero operand. A jump to non-existent memory will hang up the program, since there is then no way to jump back to existing memory.

#### **PDP-8/F AND PDP-8/M**

The PDP-8/F and PDP-8/M computers contain the same central processor and memory system as the PDP-8/E, along with an equivalent power supply which is sufficient to drive one fully utilized OMNIBUS at the power levels indicated in Table 2-2. Although the PDP-8/F and PDP-8/M are supplied with only a single 20-slot OMNIBUS, the system may be expanded to contain up to 32K of memory and 26 peripheral device control modules.

**Table 2-2 PDP-8/M System Expandability**

Current required/ available for:	Voltage level			OMNIBUS slots
	+5	-15	+15	
PDP-8/M	6.6A	3.3A	0.6A	8
All options	10.4A	1.7A	0.4A	12
Total available:	17.0A	5.0A	1.0A	20

The PDP-8/M is an original equipment manufacturer's version of the PDP-8/F. It is supplied with one of two front panels, corresponding to the two models offered. The PDP-8/M-MC includes a KC8-M operator's panel containing a POWER switch, power ON indicator, SW switch (for MI8-E Bootstrap Loader) and RUN indicator. All of these indicators are solid state light emitting diodes.

The PDP-8/M-DC includes a KC8-ML programmer's console containing all the switches and indicators found on the standard KC8-EA programmer's console. The indicator lamps on the KC8-ML, however, are all solid state light emitting diodes. The PDP-8/F is supplied with a KC8-FL programmer's console, which is similar to the KC8-ML console.

Most options available for the PDP-8/E may be interfaced to a PDP-8/F or PDP-8/M. In particular, note that PDP-8/M-MC users will require either an MI8-E Bootstrap Loader, a KP8-EA Power Fail and Auto Restart option or a customer-designed loader in order to initialize the machine.

## PDP-8/E OPERATING SYSTEMS AND SOFTWARE

More than one thousand fully developed and documented programs are available to every PDP-8 user.

A functional data processing system requires more than hardware alone. In fact, one of the most common problems encountered during design and implementation of a computer system is that program development and software support often entail a much larger expense in terms of both time and money than the cost of the data processing equipment. This is not the case with PDP-8/E systems. The Digital Equipment Corporation Software Distribution Center maintains a library of more than 700 programs, ranging from sophisticated applications routines for limited user populations to complete operating systems. No matter how exotic or innovative a particular application may appear, the chances are that a significant portion of the requisite supporting software is already available, along with full documentation which generally includes binary and source language tapes, listings, instruction manuals, execution timing summaries and a frank appraisal of worst-case interactions.

PDP-8/E software is designed to provide maximum adaptability and reduce early obsolescence. Most programs and packages will run on a wide variety of hardware configurations, usually including non-standard peripheral devices and often without reassembly or re-compilation. The OS/8 Operating System, for example, may be used with virtually any peripheral device by simply coding a 1- or 2-page device handler and adding it to the library of standard device handlers supplied with the system. The COS-300 Commercial Operating System is typical of software designed to eliminate obsolescence; it includes a special utility routine for use when incorporating new features into the system as they become available.

Many widely-accepted programming languages are supported on the PDP-8/E, including full FORTRAN IV (three versions) and full standard Dartmouth College BASIC (two versions). The memory restrictions that have prevented very long programs from running on small computer systems in the past are largely overcome by a program chaining feature that facilitates running programs of virtually any length in as little as 8K of memory. Timesharing, resource sharing, full I/O device independence and real-time support are among the capabilities available to every PDP-8/E user.

Additional programs and applications packages may be obtained from DECUS, the Digital Equipment Computer Users Society. DECUS is a non-profit user's group (the second largest such group, worldwide) that sponsors technical symposia, publishes a periodic newsletter, and maintains a library of more than 1200 programs for the various DEC computers. Every customer who has purchased or ordered a computer manufactured by DEC is eligible for an installation membership in DECUS. Two classes of individual membership are also available. Membership in DECUS is strictly voluntary, and does not require payment of dues. Programs from the DECUS library are available to all members on a request basis. In some instances, a nominal charge may be associated with a particular program. A complete catalog of available programs may be obtained from the society.

The remainder of this chapter contains brief descriptions of a selection of PDP-8/E programs and software packages. This is not, by any means, an exhaustive summary of available software. It is intended to illustrate the powerful features and versatile applications of standard PDP-8/E programs and systems, with an emphasis on newly-released software products.

### **OS/8 Operating System**

OS/8 is a comprehensive library of system programs operating under the supervision of an integrated executive. The OS/8 operating system represents a major advance in small computer software development, with capabilities that were formerly available only on such powerful machines as the PDP-10. This breakthrough in software technology makes all the features of a sophisticated operating system available to PDP-8/E users. OS/8 was specifically designed to shorten the time required for program development, increase throughput at dedicated data processing installations, and facilitate system management.

Programmers can take advantage of OS/8 by storing data files or executable programs in a system library, where they may be accessed for loading, modification or execution by means of simple keyboard commands entered at the console terminal. OS/8 provides for convenient program chaining, so that a problem may be divided into a set of smaller programs, each written in the language which is best suited to it. In the same manner, very large programs may be coded in small segments that can be overlaid during execution, to conserve memory storage.

Programs written under OS/8 may be coded in a manner that allows complete I/O device independence. Program I/O is performed by standardized calls to system device handlers and a comprehensive I/O supervisor called the User Service Routine. This feature permits programs to be written without regard for the characteristics of a particular I/O device. When a device independent program is executed, the user enters a runtime I/O specification command selecting the I/O devices to be employed during program execution, thus tailoring the I/O to a specific application or system configuration. When a system is expanded, programs use the new I/O capabilities to full advantage immediately, with no rewriting or reassembly.

Logical names may be assigned to devices within the OS/8 system. This permits symbolic referencing of peripheral devices and makes certain classes of devices fully interchangeable from a programming standpoint. User programs retain full control over the length of I/O buffers, to ensure optimum use of available storage and fast, efficient block data transfers.

OS/8 consists of an executive and a library of system programs. The executive, which supervises the execution of system programs, is comprised of 4 major components: the Keyboard Monitor; Command Decoder; User Service Routine; and I/O device handlers. The Keyboard Monitor accepts commands from the console terminal to assign logical device names; load, run and save system or user programs; and execute the "invisible" debugging routine, which is so designated because it

appears to the user as though it does not occupy any memory. These features of the Keyboard Monitor provide full communication between the user and the OS/8 executive by means of only seven monitor level commands.

The Command Decoder is called during execution of a system program or a device independent user program. It accepts a command line from the console terminal and decodes the command to determine what combination of input files, output files and runtime options will be used during the current execution of the program. In this manner, I/O specification commands are standardized for most system programs, greatly reducing the time required to become familiar with the system command structure.

The User Service Routine, or USR, controls file directory operations under OS/8. Any system or user program may access the USR by executing a standard calling sequence. Functions performed by the USR include loading device handlers; searching file directories; creating, opening and closing files; calling the Command Decoder; and program chaining.

The resident portion of OS/8 is limited to only 256 locations, allowing maximum utilization of available storage for user programs or data. Non-resident portions of the system are swapped into memory from the system device automatically, as required. Although OS/8 runs in only 8K of memory, it expands to utilize up to 32K, if available. The system requires at least 64K words of disk storage or at least one DECTape unit; however, use of two or more DECTape drives is recommended for DECTape-based systems to decrease file access time and facilitate duplicating tapes.

Every OS/8 system is easily expanded to include virtually any peripheral devices. Fully supported I/O device options include high- or low-speed paper tape equipment, card readers, line printers, and a selection of hard-copy or CRT console terminals, as well as a wide variety of disk and magnetic tape bulk storage devices. Non-standard devices may be added to any system by coding a 1- or 2-page device handler and appending it to the standard device handlers supplied with the system. Full I/O device independence is maintained, even for non-standard devices.

#### **Timeshared-8/E**

Timeshared-8 (also known as TSS/8 or EduSystem 50) combines powerful multi-language timesharing software with the proven dependability and economy of the PDP-8/E computer. The system allows up to 17 users simultaneous access to a multi-language, interactive resource/timesharing system which includes an exceptionally powerful version of the BASIC language, as well as FOCAL, FORTRAN, ALGOL, PAL Assembly Language, QUICKPOINT, and a full complement of utility software for editing, debugging, and device handling.

A key advantage of Timeshared-8 over other timesharing systems is that each user has full access to all the system peripherals and devices. The ability of any user to access any system device is called resource sharing. With resource sharing, peripheral devices such as disk storage units, magnetic tape drives, high-speed paper tape equipment, card

readers and line printers are available to any user while the computer is servicing other users.

More than 100 Timeshared-8 systems are in use today. They have been installed in schools, for student problem-solving, simulation studies, data reduction, business education and administrative processing. Industrial firms and government agencies use Timeshared-8 for software development, computation, production of numerical control tapes, control of drafting machines, and a host of other applications. Timeshared-8 software includes compilers, interpreters and assemblers, described separately in the following paragraphs, as well as applications and utility packages. Additional programs are easily added to the system, and all programs are available to every user at all times.

**FORTRAN-D**, an acronym for FORmula TRANslator, Disk, is a Time-shared-8 language based on the FORTRAN scientific processing language; however, it has relaxed format specifications to make the language more amenable to interactive program development and problem solving. Error diagnostics aid the novice programmer during both compilation and execution. FORTRAN-D programs may be stored in source language, for subsequent editing and program refinement, or in compiled binary format, for optimum efficiency with frequently-run programs. Data files may be manipulated on the disk by means of READ and WRITE statements, and two data files may be open simultaneously. Because the console terminal is used so frequently for input and output under Timeshared-8, FORTRAN-D includes two special I/O instructions, ACCEPT and TYPE, which perform the same function as console READ and WRITE statements without the necessity for device codes. The RUBOUT function is implemented under the ACCEPT command.

**BASIC**, as implemented under Timeshared-8, is a full, standard version of the language developed at Dartmouth College. In addition, Time-shared-8 BASIC has capabilities for program storage, data file storage, character strings, additional functions, multiple statements on a single line, and interactive editing, plus an extensive set of more than 70 error diagnostics. Programs up to 350 lines in length are handled easily, and program chaining techniques may be used to run programs of virtually any length.

Timeshared-8 BASIC allows one program to create or access up to 100 data files on either disk or DECTape. Files may be as long as 370,000 characters in length. They are opened and named with the OPEN command, or closed with the CLOSE command. Information is stored using the PUT function and retrieved with a GET statement, while file structure may be specified by means of a RECORD command. Numeric BASIC files are fully compatible with OS/8 FORTRAN, permitting programs to be created in a timesharing environment for subsequent batch processing under OS/8.

**PAL-D**, an acronym for Program Assembly Language Disk, is a full, standard, PDP-8/E assembler implemented under Timeshared-8. With PAL-D, the timesharing terminal user has access to the equivalent of a complete 4K PDP-8/E disk monitor system plus 35 additional monitor level commands for powerful input, output and device control. PAL-D may be

used to program the Timeshared-8 computer just like any 4K PDP-8/E; all instructions function exactly as they would on a stand-alone computer, including input/output transfer (IOT) instructions. Timeshared-8 also places many IOT instructions that are not available in stand-alone operation at the user's disposal. Programs interact with multi-character buffers, rather than single-character registers. This permits I/O manipulation of strings of characters, rather than individual characters, so that a user may run programs of greater complexity in his 4K memory allocation than could be run on a 4K stand-alone machine.

**ALGOL**, an acronym for ALGOrithmic Language, is a popular, international, scientific language with highly sophisticated program blocking and logical operations. The version of ALGOL implemented under Timeshared-8 is IFIP Subset ALGOL 60, available from DECUS. It is a one-pass, compile-and-go language with convenient interactive editing and self-explanatory error diagnostics.

Logical (Boolean) operations are a powerful feature of Timeshared-8 ALGOL. Programs may use the following Boolean operators to form logical expressions: TRUE, FALSE, NOT, AND, OR, IMP, and EQU (equivalence). A Boolean expression may be used directly, in a conditional IF statement, or indirectly, to assign a logical value to a Boolean variable. Unlike the IF statements of FORTRAN and BASIC, the ALGOL conditional IF has several extended forms to provide for highly complex conditional branching. In addition, up to 18 conditional IF statements may be nested to implement incredibly powerful logical operations.

**FOCAL**, an interactive, algebraic language developed specifically for the PDP-8/E and described in detail later in this chapter, is also supported under Timeshared-8. In this application, FOCAL is particularly valuable for performing algebraic calculations without the necessity for writing a conventional stored program. FOCAL's desk calculator mode of operation makes the full computational power of the computer available for one-time only calculations performed in response to simple, sentence structured keyboard commands.

Timeshared-8 supports a wide variety of utility programs including EDIT, PIP, and ODT, described later in this chapter, as well as COPY, a DEC-tape file handling program, and SYSTAT, a system status report generator. Any language or program that can run on a 4K PDP-8/E may be added to the Timeshared-8 system, where it will be available to all users simultaneously. A few examples are QUICKPOINT for numeric control of machine tools, TRAC and SAIBOL for list processing, and FOCARL, an advanced version of FOCAL.

Since Timeshared-8 gives each user the equivalent of a dedicated 4K PDP-8/E, machine language programs or software modules can be developed very efficiently under the interactive timesharing monitor for later use on a stand-alone machine. One large corporation uses Timeshared-8 systems at three sites almost exclusively for program development and refining, at a time savings of nearly 40 to 1 versus development on a stand-alone computer. Cost reductions are equally impressive; Timeshared-8 is operated at a cost of less than \$1.00 per terminal hour.

Timeshared-8 systems are widely employed in an educational environment for Computer Aided Instruction (CAI), computer science applications, and general problem solving. They may also be used for computer simulation, the modelling of a real-world process by a digital computer. Aside from simple laboratory apparatus such as the inclined plane or equipment used in Young's double-slit experiment, the computer can simulate apparatus that is out of reach of most school laboratories, such as a linear accelerator or cryogenic chamber. In fact, virtually any process can be simulated under Timeshared-8.

### **CAPS-8 Cassette Programming System**

The CAPS-8 Cassette Programming System combines the convenience of a magnetic tape resident operating system with the economy and basic adaptability of a paper tape system. Versatile, low-cost bulk storage is provided by the new TA8-E/TU60 Magnetic Tape Cassette Drive and Controller, described in Chapter 7. The system also includes an 8K (or larger) PDP-8/E, an LA-30 DECwriter Terminal and an optional line printer.

The CAPS-8 Keyboard Monitor accepts typed commands to load and run library programs, list tape directories, and perform various functions associated with cassette file management. System command structure is very similar to that of OS/8, providing the same concise format and ease of use. The tape cassette serves as an unformatted, file-structured bulk-storage device which contains the system executive along with a library of system and user programs. Monitor level control is maintained by refreshing the memory resident portion of the system executive from the cassette whenever necessary.

System programs supplied with the CAPS-8 Programming System include a cassette-based version of the BASIC language; PALC, a PDP-8/E assembler; and COPY, a cassette file manipulation routine that makes multiple copies of part or all of a cassette. A re-designed symbolic editor similar to OS/8 EDIT is also included. Any program or routine that runs on the CAPS-8 system hardware configuration may be added to the system library.

### **DEC/X8 System Exerciser**

DEC/X8 is a powerful and adaptable modular software system for testing PDP-8/E hardware in a systems environment. It is designed to test worst-case interactions with all peripheral devices running and the processor heavily loaded. A unique feature of the exerciser is called "rotation," a process by which each job is run from every memory field and data transfers occur to every memory field. The modular structure of DEC/X8 enables the user to tailor the exerciser to his individual needs and his particular hardware configuration.

DEC/X8 consists of three major sections. The control section is the DEC/X8 monitor, the software mainframe and true "workhorse" of the exerciser. The monitor controls interrupt servicing, deferred service, queueing, and user-exerciser communications. It requires only 4K of memory, yet it is self-expanding to utilize up to 32K, if available.

A second section is the DEC/X8 builder. The builder is used only during the exerciser building phase, when it provides the means by which the

user "inserts" software modules into the DEC/X8 mainframe and saves the fully configured exerciser in a form suitable for future use. In 4K systems, the builder is necessarily restricted to paper tape input and output. In systems with 8K or more of memory, however, the builder becomes virtually device independent by interfacing directly with the OS/8 Operating System. Both the monitor and the builder reside in the same binary file.

The third section of DEC/X8 consists of all available DEC/X8 software modules. Each of these modules is designed to exercise a specific function and/or device associated with PDP-8/E hardware. Modules used to test I/O peripherals are interrupt driven, while modules that test the CPU functions execute as background routines. The TC01DT module, for example, exercises an interrupt driven TC01 or TC08 DECTape system with up to 8 transport units. The NOTFUN module, which is typical of the background exercisers, verifies that non-functional IOT instructions may be executed without affecting the system. Output from all tests is supplied in the form of printed reports, and a full set of error diagnostics is included.

### **COS-300 Commercial Operating System**

COS-300 is Digital Equipment Corporation's DATASYSTEM 300 Series Commercial Operating System, a comprehensive applications package for commercial EDP users. The COS-300 system software provides a powerful minicomputer base for the DIBOL language, a data entry package, the COS-300 System Monitor with device handlers, and a collection of utility programs. These software components, described separately in the following paragraphs, comprise a self-contained disk or tape resident operating system for implementing data management functions in small-to medium-size commercial applications.

The COS-300 Monitor provides program operation master control. To economize memory requirements, the monitor resides in two segments, only one of which is core resident. A comprehensive set of interactive monitor commands implements full job control, file manipulation and editing capabilities. One portion of the monitor contains an interactive line editor, used to make insertions, deletions or changes to any text file. The editor provides automatic generation, sequencing and resequencing of line numbers by means of simple commands, as well as versatile I/O capabilities. It also permits batch mode commands to be edited into a system file for later execution.

**CREF** is a utility routine for program development, which provides an alphabetic listing of all symbols used in a DIBOL program and associated line numbers indicating where each symbol is defined or referenced. The COS-300 version of CREF is similar to the OS/8 version, described later in this chapter.

**BUILD** is a utility routine used to create data files under COS-300. BUILD consists of a powerful key-word data-entry package which allows a control program to specify key words followed by ordered strings of formatted data. Features include hash totalling, range checking, check-digits computation, automatic field duplication, default and incremental fields, and many more. During execution BUILD scans each data input line and suppresses output when an erroneous record is encountered.

	Compiler/ Assembler Size	Run-Time Package Size	Minimum Storage Requirements	Maximum Storage Utilized	Compilation/ Assembly Speed	Execution Speed (relative)
PAL III Assembler DEC-08- LPALA-A	2K	—	4K	Assembler. 4K. Run-Time. 32K.	From high speed paper tape. 3-15 min. From low speed paper tape. 6-90 min.	Machine speed.
OS/8 PAL-8 Assembler DEC-S8- OPALA-A	3.5K	—	8K	Assembler. 12K. Run-Time. 32K.	From DECdisk or DECpack. 5 sec.-2 min. From DECTape. 20 sec.-5 min.	Machine speed.
8K SABR Assembler DEC-08-A2D2	5K	—	8K	Assembler. 8K. Run-Time. 32K.	From DECdisk. 5 sec.-2 min. From DECTape. 30 sec.-4 min.	Approx. 65% slower than machine speed.
OS/8 SABR Assembler DEC-S8- OSABA	5K	—	8K	Assembler. 8K. Run-Time. 32K.	From DECdisk or DECpack. 5-30 sec. From DECTape. 20 sec.-2 min.	Approx. 50% slower than machine speed.
OS/8 Batch Processor DEC-S8- OBOSA-A	Not Applicable	1K	12K	32K	Not Applicable	Not Applicable
Standard Floating Point Package (23-bit) DEC-08-NFPPA-A	Not Applicable	1.1K	4K	32K	Not Applicable	Medium/Fast FADD { 275 FSUB { sec FMPY / 1 FDIV / msec
EAE Floating Point Package (23-bit) DEC-8E- NEA-EA-A	Not Applicable	1K	4K	32K	Not Applicable	Very fast FADD { 170 FSUB { sec FMPY / 185 FDIV / sec
Special Floating Point Package (27-bit) DEC-08- NFPEA-A	Not Applicable	1.1K	4K	32K	Not Applicable	Medium.

**SORT** is a poly-phase sorting routine that will sort COS-300 file records into ascending or descending order. Up to 8 fields (with sub-fields) of a fixed length record may be specified as sort keys, and timing is relatively independent of the size of the sort keys. A SORT control program defines data files and sort keys, names the I/O files, and specifies 3 to 7 work files. SORT easily handles multi-reel files.

**UPDATE** is a master-file maintenance program used to change existing records on a COS-300 data file, insert new records, delete old records, and then print a report showing all changes, insertions, deletions, and the control program.

**PIP** (Peripheral Interchange Program) is a versatile COS-300 utility routine, similar to the OS/8 version described later in this chapter, that moves files from one I/O or mass storage device to another in any standard format.

**DAFT** (Dump and Fix Technique) is a COS-300 utility program, written in DIBOL, that facilitates dumping data files and entering minor changes

Number of Symbols Permitted (approx.)	Number of Program Statements Permitted (approx.)	Peripherals/ Options Required	Additional Peripherals/ Options Utilized	Features
600	Limited by number of symbols.	LT33-D	PC8-E	<ul style="list-style-type: none"> <li>• Largest possible symbol table on basic 4K machine.</li> <li>• Easy to use.</li> <li>• Includes all basic PDP-8 family symbol definitions.</li> </ul>
12K = 1800 8K = 800	No limit.	OS/8 system	All OS/8 Peripherals.	<ul style="list-style-type: none"> <li>• Literals.</li> <li>• Conditional assembly.</li> <li>• Cross-referenced listings.</li> <li>• Page headings with title, date, page number.</li> </ul>
275	2000 per program or subroutine.	PC8-E, LT33-D or LA30-P	DF32-D, RF08/RS08, TC08/TU56	<ul style="list-style-type: none"> <li>• Page independent relocatable code.</li> <li>• Automatic linkage to off-page symbols.</li> <li>• Local, external, and common assignment.</li> <li>• Conditional assembly.</li> </ul>
250	2000 per program or subroutine.	OS/8 system	All OS/8 peripherals.	<ul style="list-style-type: none"> <li>• Programs may be written without regard to page boundaries.</li> <li>• Generates relocatable binary code.</li> <li>• Automatic paging and link generation to off page symbols.</li> <li>• External symbols.</li> <li>• Text pseudop &amp; many others.</li> <li>• Ability to chain to or call FORTRAN programs &amp; subroutines.</li> </ul>
Not Applicable	Not Applicable	CR8-E or CR8-F or PC8-E OS/8 System	All OS/8 Peripherals.	<ul style="list-style-type: none"> <li>• Full power of OS/8 available from cards.</li> <li>• Maintains log of batch programs run.</li> <li>• Supports OS/8 FORTRAN, RTPS FORTRAN IV, OS/8 BASIC, and PAL-8 and SABR assemblers.</li> </ul>
Not Applicable	4K = 3,000 32K = 31,000	None	LT33-D or LA30-P	<ul style="list-style-type: none"> <li>• Good Execution speed.</li> <li>• Interpreter makes it easy to do floating point calculations in assembly language.</li> <li>• FJMP, FJMS, FISZ, FCDF, and conditional skips.</li> <li>• Auto-increment registers.</li> <li>• Can use direct subroutine calls.</li> </ul>
Not Applicable	4K = 3,000 32K = 31,000	PDP-8/E, -8/F, or -8/M KE8-E	LT33-D or LA30-P	<ul style="list-style-type: none"> <li>• Very fast execution speed.</li> <li>• Includes all the convenient features of the standard Floating Point Package.</li> </ul>
Not Applicable	4K = 3,000 32K = 31,000	None	LT33-D or LA30-P	<ul style="list-style-type: none"> <li>• Seven digit precision.</li> <li>• Uses the same interpreter as the standard Floating Point Package.</li> <li>• Uses the same data format as 8K FORTRAN, OS/8 FORTRAN and PDP-8 BASIC.</li> </ul>

to data files. DAFT can search a data file on a specified key, print records or parts of records on the line printer or console terminal, and search for a specified record. Record skip and backspace features are included to facilitate locating particular records for subsequent editing, and DAFT may also be used to place a specified number of copies of the current record onto an output file.

Also included with COS-300 are PATCH, a utility routine that enters modifications or enhancements to system software with automatic checksum calculation to validate all entries; BOOT, a system initializer that permits any operating system to be loaded onto any device in the fully initialized system, and CONVRT, a file conversion routine that translates various file formats.

### ASSEMBLERS

Use of an assembly program has become a standard practice in programming digital computers. This process allows the programmer to code machine language instructions in a symbolic language, which is much easier to work with than the 12-bit binary instruction codes that actually operate the computer. The PAL Program Assembly Language used with PDP-8/E computers provides significant advantages over machine language coding: it is more meaningful and convenient than

	Compiler/ Assembler Size	Run-Time Package Size	Minimum Storage Requirements	Maximum Storage Utilized	Compilation/ Assembly Speed	Execution Speed (relative)
Paper Tape Symbolic Editor DEC-08-ESAC	Not Applicable	1K	4K	4K	Not Applicable	Not Applicable
OS/8 Symbolic Editor DEC-S8- OEDTA-A	Not Applicable	2K	8K	8K	Not Applicable	Not Applicable
TECO (Text Editor and Corrector) DEC- S8-UTECA-A	Not Applicable	2.5K	8K	12K	Not Applicable	Not Applicable
FOCAL DEC-08-AJAE	Not Applicable	2.5K	4K	8K	Not Applicable (FOCAL is fully inter- pretive).	Slow.
8K FORTRAN DEC-08-A2B1	5K	3.5K	8K	Compiler, 8K. Run-Time, 32K.	From high speed paper tape, 15-25 min.	Medium.
OS/8 FORTRAN DEC-S8- LFORA-A	5K	3K	8K	Compiler, 8K. Run-Time, 32K.	From DECdisk or DECpack, 10-20 sec. From DECtape, 3-4 min.	20 to 30% faster than 8K FORTRAN
RTPS FORTRAN IV DEC-08- LHTPA-A	5K	3K	8K	32K	From DECdisk or DECpack, 10-20 sec. From DECtape, 4-6 min.	Very fast.

numeric code; instructions or data may be referenced by symbolic names without concern for, or even knowledge of, their actual addresses in memory; decimal, octal and alphanumeric data may be expressed in a form that is more convenient and familiar than binary numbers; and programs may be altered more efficiently, so that program debugging is considerably simplified.

Program Assembly Language provides optimum utilization of the PDP-8/E processor because there is a one-to-one correspondence between PAL mnemonics and PDP-8/E machine instructions. Maximum programming efficiency is maintained through full provision for microprogramming, memory allocation, direct or indirect addressing, and the like. Programs written in PAL may be supplied as input to any of a class of PDP-8/E assemblers which will translate the symbolic language program into its machine language equivalent.

**PAL III**, the basic PDP-8/E assembly program, is a two-pass assembler with optional third pass that uses either high- or low-speed paper tape and console terminal I/O. The PAL III assembler builds a table of user-defined symbols during its first pass. At the end of the first pass, the input tape is reloaded for a second pass, during which PAL III generates a binary format program tape. The optional third pass may be used to

	Number of Symbols Permitted (approx.)	Number of Program Statements Permitted (approx.)	Peripherals/ Options Required	Additional Peripherals/ Options Utilized	Features
	Not Applicable	Up to 3,200 characters	LT33-D or LA30-P	PC8-E	<ul style="list-style-type: none"> <li>• Versatile and easy to use.</li> <li>• 15 powerful editing commands.</li> <li>• Special character editing feature allows changing individual characters or words without retyping entire line.</li> <li>• Fully interactive—allows programmer to verify changes and recorrect if necessary.</li> </ul>
	Not Applicable	Up to 5,600 characters	OS/8 System	All OS/8 peripherals	<ul style="list-style-type: none"> <li>• Character string search.</li> <li>• Device independent input/output.</li> <li>• Easy to use—has all the features of paper tape version.</li> </ul>
	Not Applicable	Up to 4,000 characters plus an additional 2,000 characters in the Q-registers.	OS/8 System	All OS/8 peripherals; PDP-12 display.	<ul style="list-style-type: none"> <li>• An extremely powerful character editor for the advanced programmer.</li> <li>• Character string searches.</li> <li>• Programmed editing loops.</li> <li>• Editing macro instructions.</li> <li>• Conditional editing.</li> <li>• Q-registers may contain text or numeric data and may be used in arithmetic operations.</li> </ul>
	200	4K = 120 8K = 400	LT33-D	PC8-E	<ul style="list-style-type: none"> <li>• Easy to learn, easy to use.</li> <li>• Run large programs in 4K machine.</li> <li>• Immediate mode commands for fast problem solving and debugging.</li> <li>• Program may be stopped at any time—variables examined, etc. and then program execution resumed.</li> </ul>
	230 symbols, 1300 fl. pt. or 3900 fixed pt. variables per prog. or subroutine	8K = 200 32K = 1200	PC8-E, LT33-D or LA30-P	None.	<ul style="list-style-type: none"> <li>• Well known programming language.</li> <li>• Does not require mass storage.</li> <li>• E, F, I, A, X, H, format specs.</li> </ul>
	230 symbols, 1300 fl. pt. or 3900 fixed pt. variables per prog. or subroutine	8K = 250 32K = 1200	OS/8 system	All OS/8 Peripherals.	<ul style="list-style-type: none"> <li>• Easy to use—compile, load and go with two simple commands.</li> <li>• No paper tape required.</li> <li>• Implied DO-loops, Hollerith constants, program chaining, simple interface with assembly language.</li> <li>• Process data using disk and DECtape files.</li> </ul>
	150 symbols, 3000 variables per program or subroutine	8K = 350 32K = 2450	RTPS system	Most OS/8 Peripherals plus AD8-E, DK8-E, VC8-E/VR14.	<ul style="list-style-type: none"> <li>• Direct access I/O.</li> <li>• n-dimensional arrays, n ≤ 12.</li> <li>• Generalized array subscripting.</li> <li>• Mixed-mode arithmetic.</li> <li>• Boolean operations.</li> <li>• Logical IF.</li> <li>• Octal constants.</li> <li>• Error trace-back.</li> <li>• Multiple-level overlays.</li> </ul>

produce a full symbolic program and symbol table listing on the console terminal and/or paper tape, if desired. Like all PDP-8/E assemblers, PAL III performs extensive error diagnosis and identifies errors by location and type whenever they are encountered during an assembly. The PAL III assembler runs on any standard PDP-8/E.

**MACRO-8** is an advanced, two-pass assembler with optional third pass that has the same features as PAL III plus many additional capabilities such as user defined macros, double-precision integers, floating-point constants, arithmetic and Boolean operators, literals, text manipulation facilities and automatic off-page linkage generation. These features were incorporated by decreasing the size of the user symbol table, so that MACRO-8 requires only 4K of memory. The symbol table may be enlarged by deleting unused features, if necessary. Like PAL III, MACRO-8 uses console terminal and high- or low-speed paper tape I/O.

**PAL8** is an extended assembler that runs under the OS/8 operating system. It includes some of the best features of both PAL III and MACRO-8, plus such additional features as conditional assembly, expanded symbol table allowing up to 1800 entries on a 12K system, rapid binary symbol table search, extended pseudo-operations and paginated listings with page headings and numbered pages. A load and go option causes the input program to be assembled, listed, loaded and executed, all as the

	Compiler/ Assembler Size	Run-Time Package Size	Minimum Storage Requirements	Maximum Storage Utilized	Compilation/ Assembly Speed	Execution Speed (relative)
4K BASIC (EDUSystem-10) EDU8-B10	Not Applicable	3.5K	4K	4K	Not Applicable. EDUSystem-10 is fully interpretive.	Medium.
Multi-User BASIC (EDUSystem-20) EDU8-B20	Not Applicable	4.5K	8K	32K	Not Applicable. EDUSystem-20 is fully interpretive.	Slow/ Medium.
Multi-User BASIC (EDUSystem-25) EDU8-B25	Not Applicable	6.5K	12K	32K	Not Applicable. EDUSystem-25 is fully interpretive.	Slow/ Medium.
Batch BASIC (EDUSystem-30) EDU8-B30	3.5K	2K	4K	4K	From DECdisk, 5-10 sec. From DECtape, 30-60 sec	Medium.
8K BASIC (PDP-8 BASIC) DEC-08-LBASA-A	Not Applicable	4K	8K	8K	Not Applicable. PDP-8 BASIC is fully interpretive.	Medium.
REAL-TIME BASIC (LAB- 8/E BASIC) DEC-LB-U70P	Not Applicable	5K	8K	8K	Not Applicable. LAB-8/E BASIC is fully interpretive.	Medium.
OS/8 BASIC DEC-S8- LBASA-A	3K	4K	8K	32K	From DECdisk or DECpack. 5-10 sec. From DEC- tape 15-60 sec	With EAE option—Fast. Without EAE— Medium.

result of a single keyboard command. PAL8 communicates with the user via the device-independent OS/8 Command Decoder. Thus, it may operate with any I/O devices that are present in the system.

**SABR** (Symbolic Assembler for Binary Relocatable programs) is an advanced, one-pass assembler with an optional second pass. Since SABR generates off-page and off-field linkages automatically for either directly or indirectly addressed instructions, SABR programs may be written without regard for page length restrictions. In addition, the binary program output produced by SABR is fully page-relocatable; it may be loaded into any available pages of memory, in any memory field.

SABR supports an extensive list of pseudo-operations which provide for conditional assembly, external subroutine calls, argument passing to external subroutines, and many other facilities. Any SABR program may execute calls to a large library of SABR or FORTRAN functions and subroutines. These are loaded automatically, along with the relocatable SABR program output, by the Linking Loader. SABR also permits allocation of COMMON storage, which greatly facilitates passing large amounts of data from one routine to another.

Number of Symbols Permitted (approx.)	Number of Program Statements Permitted (approx.)	Peripherals/ Options Required	Additional Peripherals/ Options Utilized	Features
50	50	LT33-D	KP8-E MI8-E	<ul style="list-style-type: none"> <li>• Easy to learn and use.</li> <li>• Immediate mode for fast problem solving.</li> </ul>
4K = 200 32K = 700	4K = 200 32K = 320	LT33-D	Additional terminals, KP8-E, MI8-E.	<ul style="list-style-type: none"> <li>• Up to 8 users.</li> <li>• Simple to learn, easy to use.</li> <li>• Immediate mode for fast problem solving.</li> </ul>
400 typical. Variable storage depends on number of users and amount of core memory.	250 typical. Depends on number of users and amount of core memory.	LT33-D and TC08	Additional terminals KP8-E, MI8-E	<ul style="list-style-type: none"> <li>• Program and data file storage.</li> <li>• Character string variables.</li> <li>• Program chaining.</li> <li>• Plus all the features of EDUSystem-20.</li> </ul>
500 (plus data statements)	220	LT33-D, DF32 or RF08/RS08 or TC08/TU56	CR8-F or CM8-F, LE8, KP8-E, MI8-E	<ul style="list-style-type: none"> <li>• Batch processing capability.</li> <li>• Programs can be saved on disk or DECtape.</li> <li>• Very large programs can be run on 4K machine.</li> <li>• Program chaining for even larger programs.</li> </ul>
800	175	LT33-D or LA30-P & PC8-E	PC8-E, LE8	<ul style="list-style-type: none"> <li>• Simple to learn, easy to use.</li> </ul>
700	150	LT33-D or LA30-P & PC8-E	PC8-E, LE8, VC8-E/VR14, AD8-E, DK8-EP	<ul style="list-style-type: none"> <li>• Simple to learn, easy to use.</li> <li>• Lab interface.</li> <li>• Real-Time features.</li> </ul>
500 symbols. Up to 8,000 variables in 32K core.	400 typical	OS/8 System	All OS/8 peripherals; KE8-E	<ul style="list-style-type: none"> <li>• Easy to learn and use.</li> <li>• Numeric and alphanumeric data and program storage on disk and tape.</li> <li>• Character strings.</li> <li>• User function interface to machine language.</li> <li>• Program chaining and compile-only</li> </ul>

## LANGUAGE INTERPRETERS AND COMPILERS

**FORTRAN IV** is a widely-accepted programming language made up of formatted statements that are very similar to the language of conventional algebra and higher mathematics. The FORTRAN IV language implemented under the OS/8 Operating System is a superset of ANSI FORTRAN Standard X3.9-1966. It is available in three versions for use on a PDP-8/E alone, PDP-8/E with KE8-E Extended Arithmetic Element, or PDP-8/E with FPP-12 Floating-Point Processor. The FPP-12 is a 24-bit processor with a full complement of arithmetic and control instructions, which is described in detail in Chapter 7. With an FPP-12, the OS/8 FORTRAN IV package provides an extremely versatile Real Time Programming System that finds wide application in laboratory and educational environments, or wherever fast, powerful real-time capabilities are required.

**OS/8 FORTRAN IV** permits generalized array subscripting and one- to twelve-dimensional arrays. This makes bulk data easier to store and access than ever before, in any language; however, FORTRAN IV also offers direct access I/O. With this additional feature, the user may directly reference any record in a data file. In the traditional scientific area, direct access I/O provides the capability for virtual arrays. In the commercial area, it greatly simplifies and speeds up processing.

**Mixed mode arithmetic** is implemented under OS/8 FORTRAN IV, along with octal constants, logical IF statements, and general integer expressions in IF statements. In addition, OS/8 FORTRAN IV provides for initial values in specification statements and a full set of Boolean operators, including EQU and XOR.

**Text manipulation** is greatly facilitated by Hollerith field specifications for text, as well as literals and constants. DATA statements, BLANK statements and BLOCK COMMON are fully supported, and there is provision for arithmetic function definition. All features are highly optimized with respect to execution timing, so that the OS/8 FORTRAN IV package affords exceptional speed, executing ten 25X25 matrix inversions, for example, in slightly more than one minute on the FPP-12 configuration.

**Real-time device independent input/output** is provided by OS/8 FORTRAN IV with the FPP-12. In this version, the PDP-8/E (or any PDP-8 except the PDP-8/S) functions as a fully parallel I/O processor. While the PFP-12 is processing data, the PDP-8/E may be acquiring data, displaying it, reading or writing bulk storage files, driving a plotter, and so on, thus vastly increasing system throughput. The interrupt driven I/O system permits on-line I/O in this dual processor configuration;



while the FPP-12 is crunching numbers, the I/O processor is free to handle real-time data flow. The system is also fully device independent, of course, allowing run-time peripheral device specifications without re-compilation or reassembly.

**Error diagnostics** are an important aspect of the FORTRAN language. Like any good FORTRAN compiler, OS/8 FORTRAN IV detects, flags and explains many format errors, giving clear English comments. However, beyond that, OS/8 FORTRAN IV provides complete error traceback, showing the entire flow of program control that terminated in the erroneous statement.

**Overlays** are another key design feature of OS/8 FORTRAN IV, which provides a tree-structured dynamic overlay mechanism that automatically loads overlays on call, without the need for call overlay, call link or call chain statements. As many as seven independent overlay levels may be defined, with up to 16 overlays in each level and 63 subroutines in each overlay. The dynamic overlay structure combines with the OS/8 program chaining capability, permitting OS/8 FORTRAN IV to handle programs of virtually any length.

The **real-time power** of OS/8 FORTRAN IV admits to many applications in a laboratory environment. Scientists may use it to acquire, process and display data from experiments directly, without concern for the details of machine language, and experimental processes may be controlled right in the lab. In other applications, the high speed and full language capability of OS/8 FORTRAN IV are ideally suited to an educational environment, in which students are expected to perform computational tasks on large amounts of data, as well as learn the FORTRAN language. The exceptionally large set of self-explanatory diagnostics makes it easy for a student to locate and correct his errors.

**OS/8 BASIC** is implemented as a pure compiler and associated run-time system that is exceptionally fast and core efficient. It may be used for interactive programming, in much the same manner as the Timeshared-8 version of BASIC, and it also accepts terminal format ASCII input files which may be compiled, loaded and executed in response to a single monitor command. Alternately, program and data files may be prepared under the interactive monitor for subsequent stand-alone processing under BATCH. The run-time system permits dynamic file management and program chaining, while complete I/O device independence is maintained through the OS/8 Operating System. OS/8 BASIC permits convenient interfacing with functions or software modules written in other languages; assembly language functions are particularly easy to implement.

**FOCAL** (FOrmula CALculator) is a powerful interactive programming language designed for use by students, technicians and managers who require the full problem-solving capabilities of a general purpose digital computer. FOCAL is similar to BASIC and FORTRAN in many respects; however, it is considered to be easier to learn and much better suited to one-time calculations. FOCAL has relatively low memory requirements, yet it offers a full range of mathematical functions, extendable I/O, and versatile self-editing capabilities. The basic FOCAL command set contains only 12 powerful commands, which are all the programmer requires for even the most sophisticated applications.

FOCAL offers an immediate or "calculator mode" approach to one-time only problems, making the full calculating power of the computer available without the necessity for writing a conventional program. Subroutine handling under FOCAL allows program control to transfer to a group of steps (as in normal subroutine handling), or to a single step which may be located anywhere in the program. Subroutines may be terminated either conditionally or unconditionally, and both types of subroutine return control to the statement following the subroutine call.

Since FOCAL was designed specifically for use with PDP-8 computers, it offers a highly optimized set of sophisticated, powerful commands that are custom tailored to the capabilities of the PDP-8/E. FOCAL runs on any standard PDP-8/E. It is fully documented in *Programming Languages*, 1972.

**DIBOL** (Digital's Business Oriented Language) is a commercial EDP oriented language similar to COBOL, consisting of data definition and procedural statements. The data definition section defines the type and size of data elements for record overlays and automatic memory clear. The procedure section of the language consists of a select group of procedural verbs that permit data manipulation and comparison, arithmetic expressions, subscripting, sub-element manipulation, branching, tracing, line-printer overlap, program chaining, subroutines, rounding and cursor control. The verbs used with DIBOL include: ACCEPT, CALL CHAIN, DISPLAY, END, FINI, FORM, GOTO, IF, INCR, INIT, ON ERROR, READ, RETURN, STOP, TRACE/NOTRACE, TRAP, WRITE, XMIT, and data manipulation statements.

#### **UTILITY PROGRAMS**

**BATCH** provides PDP-8/E users with a batch processing monitor that is integrated into the OS/8 monitor structure. The system is organized in such a way that it may be used in either a keyboard input configuration or as a batch stream processor. BATCH may be run on any OS/8 system equipped with at least 12K of memory, and it will support up to 32K of memory along with any I/O devices that may be present in the system.

**OS/8 BATCH** processing is ideally suited to frequently run production jobs, large and long-running programs, and programs that require little or no interaction with the user. BATCH permits the user to prepare his job on punched cards, high-speed paper tape or the OS/8 system device and leave it for the computer operator to start and run. Output is returned to the user in the form of line printer and/or console terminal listings that include program output as well as a comprehensive summary of all action taken by the user program, the monitor system and the computer operator.

BATCH provides optional spooling of output files. This feature serves to increase throughput on any system, but it is particularly valuable when a line printer is not available. BATCH also performs extensive command analysis and error diagnosis, as well as detailed interaction with the user/operator to facilitate initializing the system and establishing system parameters.

Almost any program that runs under interactive OS/8 may also be run under BATCH. Since BATCH is called from the keyboard in the same

manner as any other system program, interactive users may use BATCH to execute multiprogram utility routines, even when continuous batch processing is not desired. With a few exceptions, BATCH uses the standard OS/8 command set.

**PIP** (Peripheral Interchange Program) is a versatile file manipulation routine that runs under the OS/8 Operating System. OS/8 PIP may be used to transfer ASCII, core image, or binary files from one device to another. PIP can also merge or delete files and list, zero, or compress file directories. All communication between PIP and the user is accomplished by means of standard OS/8 Command Decoder specification strings typed at the console terminal. PIP accepts a choice of 14 run-time options, which may be used to specify file formats, perform special character conversion and rudimentary editing, or allocate off-line storage. An extensive list of error diagnostics provides complete file security by making it nearly impossible to accidentally zero the system device directory, for example, or interrupt a file compression under circumstances that might cause a loss of data.

**CREF** (Cross Reference Utility Program) aids the development programmer in writing, debugging and maintaining assembly language programs by providing the ability to pinpoint all references to a particular symbol. Input is supplied to the OS/8 version of CREF in the form of an ASCII listing file produced by either PAL8 or the SABR assembler. CREF processes this file to produce a cross reference table containing every user-defined symbol and literal, sorted alphabetically, and a list of numbers specifying the lines on which each is referenced. CREF will also produce an optional, sequence-numbered listing of the input file.

**ODT** (Octal Debugging Technique) is an integral part of the OS/8 Operating System that facilitates running prototype programs under carefully controlled conditions, modifying programs during execution, or monitoring the state of mainframe memory and the major registers. The OS/8 version of ODT does not require any memory aside from certain areas of the 256-location resident monitor; it is swapped into memory from the system device whenever required, while overlaid portions of the running program are saved on the system device for later restoration.

The breakpoint feature of ODT permits program execution to be suspended whenever the program encounters a specified instruction or iterates past the instruction a given number of times. Other features provide for fast, selective examination or modification of memory; examination or modification of the AC, PC, link, memory field and data field registers; indirect address computation and binary memory search. All functions are executed in response to concise, easily learned keyboard commands typed at the console terminal.

ODT is also available as a stand-alone program that may be used on any PDP-8, with or without an operating system. The stand-alone version is core resident; however, it is also fully page relocatable so that it may be loaded into any block of memory that is not used by the object program. Both versions of ODT are documented in *Introduction to Programming*.

**DDT** (Dynamic Debugging Technique) is a standard program debugging routine that runs on any PDP-8/E. On-line debugging with DDT provides close control over program execution and automatic generation of dynamic, printed, program status information. The DDT command set and capabilities are very similar to those of ODT, described above; however, DDT permits the user to communicate by means of decimal notation, rather than octal notation, and symbolic tags, rather than absolute addresses. DDT includes a breakpoint feature that traps, or suspends, program execution at a predetermined point, and a variety of commands to facilitate monitoring or modifying the content of the major registers and memory during program execution.

**TECO** (Text Editor and Corrector) is an extremely powerful text editing and correcting program that runs under the OS/8 Operating System. TECO may be used to edit any form of ASCII text, including program files or listings, manuscripts, and data files. Since TECO is a character-oriented editor rather than a line editor, text edited with TECO does not have extraneous line numbers associated with it, nor is it necessary to replace an entire line of text in order to change one character.

A selected subset of TECO commands provides less than 20 easily learned mnemonics which afford full editing capabilities to the novice programmer after only a few hours of instruction. The basic commands are very similar to corresponding Symbolic Editor commands; however, the I/O device independence of OS/8 system programs permits TECO to create or modify ASCII files on any medium. Other TECO capabilities and features include character string search and replacement, 36 variable length temporary storage buffers with associated integer counters, match control characters, choice of decimal or octal radix, and a number of versatile I/O techniques.

TECO commands may be combined in sophisticated command strings which are essentially "editing programs." Once a command string has been written to perform a specified editing task, it may be saved on any convenient medium for subsequent execution whenever the same editing job is required. Advanced TECO commands provide extensive capabilities for conditional execution, branching, program control and multi-file processing. A macro programming feature is included, along with commands that facilitate the creation, maintenance and use of a TECO macro library. The full TECO command set is actually a highly sophisticated programming language which is well suited to such applications as generalized format conversion, text processing and file management.

**EDIT** (Symbolic Editor) is used to create and modify ASCII source files so that these files may be used as input to other programs, such as FORTRAN, BATCH, or SABR. The Symbolic Editor is fully supported under OS/8 and also as a stand-alone program. The stand-alone version of EDIT is very flexible: it may be used to create symbolic programs or data files at the console terminal; examine, edit, and correct the files; and then prepare an ASCII paper tape that is suitable input to a wide variety of processing routines. EDIT includes a search feature which allows the programmer to scan a line of text for the next occurrence of a specified character. Other commands permit blocks of text to be inserted, deleted, appended, listed or changed. Once the internal text

buffer contains a correct image of the file, EDIT may be instructed to punch a specified portion of the file onto paper tape for subsequent editing or processing.

The stand-alone version of EDIT runs on any standard PDP-8/E. It occupies about 1000 memory locations, allowing maximum space for text buffering, and provides 15 powerful editing commands. Unlike many line editors, EDIT permits individual characters to be changed without retyping the entire line. It is fully interactive, so that editing changes may be verified and recorrected if necessary.

The OS/8 version of EDIT incorporates all features of the stand-alone package and provides full I/O device independence under the OS/8 Operating System. Files may be read from or routed to any device in the system. A special output command permits large blocks of text to be examined at the line printer, rather than on the console terminal, and expanded search commands facilitate inter-buffer searches for either single characters or strings of characters. Both versions of the Symbolic Editor are documented in *Introduction to Programming*.

**SRCCOM** (Source Compare) is an OS/8 utility program that compares two source files line by line and creates a third file listing all differences between the two sources. The input files are usually two different versions of a single assembly language program. In this case, SRCCOM will note any editing changes which transpired between the two versions, making it a valuable debugging and maintenance tool. Four run-time options may be used to suppress comparison of comment fields, tabs, spaces or blank lines, if desired. SRCCOM runs on any OS/8 hardware configuration.

**BITMAP** is an OS/8 utility program used to construct a table, or map, showing the memory locations used by a given binary file. BITMAP will accept any absolute binary file as input and route its output map to any supported I/O device. A selection of 4 run-time options is provided, along with full error checking and diagnosis. BITMAP runs on any OS/8 hardware configuration.

#### FLOATING-POINT MATH PACKAGES

Use of a floating-point package permits the PDP-8/E to perform arithmetic operations that many other computers can only duplicate after the addition of costly optional hardware. The three floating-point packages available for the PDP-8/E represent three optimizations of the trade-offs between speed, accuracy and hardware configuration. Two 23-bit floating-point packages maintain 5 or 6 significant (decimal) digits of accuracy for all operations. One package, designed for use with the KE8-E Extended Arithmetic Element, is capable of adding any two numbers in the range  $-10^{65} < X < 10^{65}$  in less than 160 microseconds. Floating-point multiplication is accomplished in about 200 microseconds, while the cosine function, which is typical of routines having longer execution times, is implemented in less than 2.5 milliseconds. The other 23-bit floating-point package executes without an extended arithmetic element and requires about 300 microseconds for a typical floating-point addition, or 1 millisecond for a typical multiplication.

The 27-bit floating-point package is similar to the two 23-bit packages; however, it maintains greater accuracy for a smaller range of numbers. Without the use of an extended arithmetic element, the 27-bit package will operate on any two numbers in the range  $-10^{38} < X < 10^{38}$  in about the same amount of time required for the corresponding 23-bit operation.

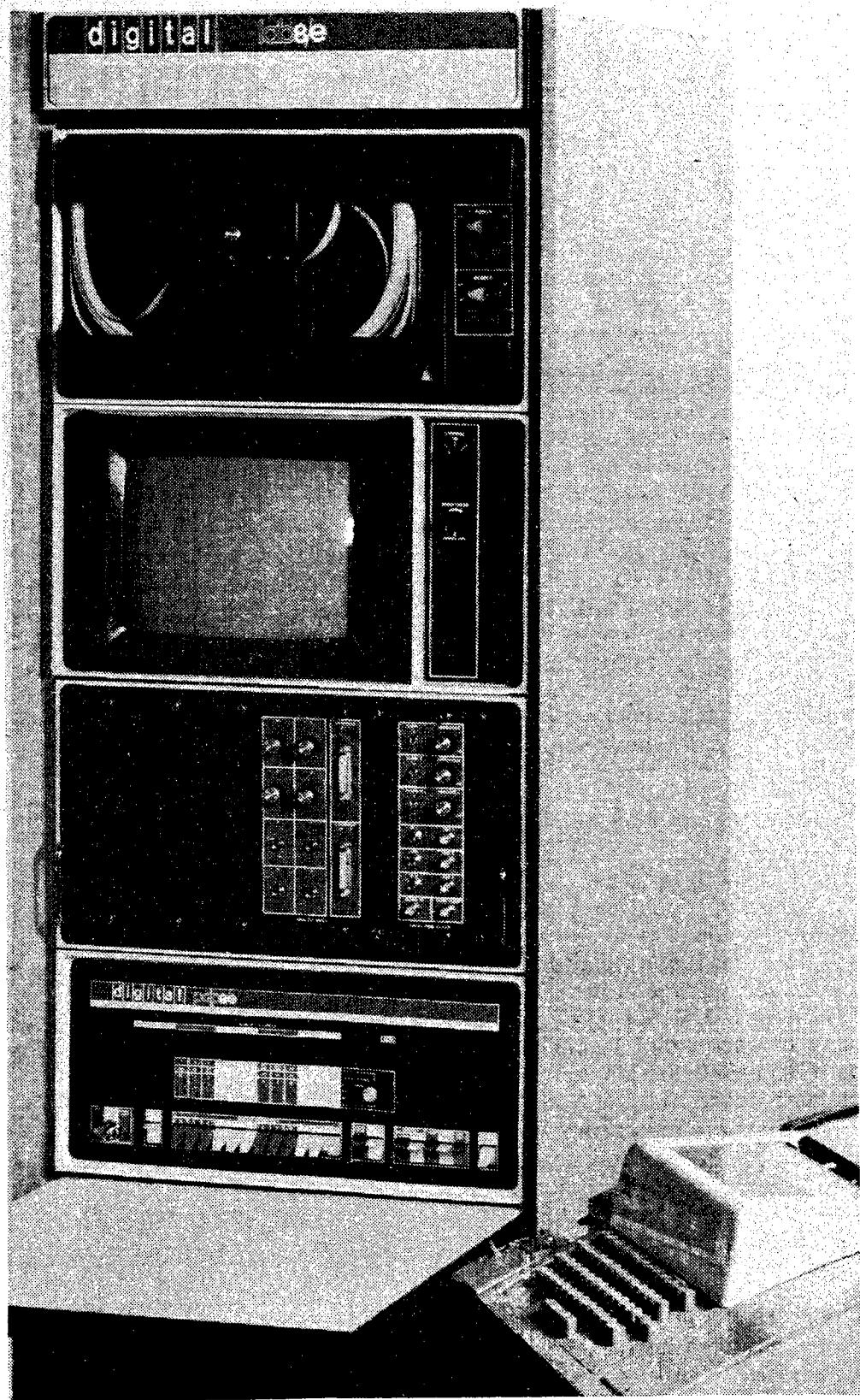
All three floating-point packages contain interpreters which will accept, decode and execute floating-point pseudo-instructions. Individual components of the packages may be called as subroutines, in single-instruction mode, to perform such operations as conventional arithmetic, trigonometric function evaluation, square root extraction, exponential function computation and calculation of natural logarithms. Alternately, any of the floating-point packages may be employed in interpretive mode, to operate on a string of pseudo-instructions. In this mode of operation, the packages function as a versatile software floating-point processor which performs all of the operations listed above, as well as floating skips, floating-point I/O, floating jumps to floating-point subroutines, and so on.

All three floating-point packages require only about 1200 storage locations in any memory field. They will accept floating-point data or pseudo-instructions from any field of memory, and all three packages perform I/O operations on floating-point numbers in either FORTRAN I, E, or F format.

#### APPLICATIONS PACKAGES

Integrated Applications Packages are special purpose software systems that provide a specific solution to one aspect of a general problem. By employing the proper applications package and, perhaps, additional hardware and software, a PDP-8/E computer system may be customized to fit almost any operational requirement. Numerous applications packages are afforded through the DECUS library, in addition to those available from the Software Distribution Center.

The LAB-8/E is a versatile, PDP-8/E based, applications system that provides powerful laboratory data acquisition, manipulation and display capabilities at a cost less than that of most special purpose laboratory instruments. Each LAB-8/E system includes a PDP-8/E computer with integrated analog-to-digital converter, real time clock, point plot display, three Schmitt triggers, and additional hardware selected to fit the intended application. The LAB-8/E is designed to be used as a total laboratory system, not simply as a computer with laboratory peripherals. The peripherals plug directly into an H945 laboratory cabinet, and they are interfaced in a manner that permits the Schmitt trigger to start the clock, the clock to gate the analog-to-digital converter, the A/D converter to increment the multiplexer, and so on. This kind of flexibility makes it possible to configure an interactive laboratory system around the PDP-8/E and expand the system, as needed, to fulfill virtually any particular requirements. The PDP-8/E software applications packages described in the following paragraphs are typical of the wide range of programs designed around the LAB-8/E applications system.



**Basic Signal Averager** software allows for an effective technique that improves the signal-to-noise ratio of an evoked response. Raw data acquired from an ADC is stored in a current buffer during each sweep, then added to the sum buffer after the sweep is completed. During a sweep, control may be exercised to display the sum, data channels, and number of points. Program start-up is conversational, allowing the number of channels, number of points, sweep rate, and delay from starting sync pulse to be selected for each run. Sampling rates range from 33 microseconds (25 with 1 channel) to 4 milliseconds for each of 1024 points (or more). Either the input signal or the averaged data may be displayed, plotted, or printed for a specified range of points. Accumulation of the average may be suspended at any time, and the average may be contracted or expanded.

**The Advanced Signal Averager** provides unique features not found in any other averager; specifically, the ability to back average, sort or edit averaged data contingent on an external event, sample at two different rates, and calculate the raw statistics needed to compile standard deviation, confidence limits, or trends. It provides statistical confidence limits around the average and standard deviation for each data point in the average, as well as first-order trend read-out. Up to 1024 points are handled in double precision at sampling rates from 175 microseconds to 2 seconds per point. Dual resolution sweeps occur simultaneously, with provision for positive or negative delays from the sync signal. Data is automatically normalized before output, and it may be displayed between movable cursors. An elaborate X-Y plotting routine provides grid lines and plotter calibration. Trend analysis is performed on each data point, with confidence limits or standard deviation being calculated for every point while the average is being taken.

**The NMR Signal Averager** is specifically designed to solve the problems of signal averaging encountered during NMR or ESR spectroscopy. It was written to provide an easily-used averager for situations in which the computer could control the sweep of the spectrometer. However, it actually represents a versatile averaging package that can be used in any situation requiring application of a -3 to 3 volt sweep voltage. When this sweep voltage is acceptable, the NMR averager offers three distinct advantages: The necessity for a sync pulse is eliminated; multiple sampling of each point is possible; and a sophisticated calibration routine permits determination of the exact frequency of any line in the accumulated spectrum. The NMR averager is loaded from one short tape, and all routines are co-resident with no need for overlays. Decimal and binary output of the accumulated signal is provided, along with plotting, curve smoothing and analog/digital integration capabilities. The NMR Averager samples 1024 points in double precision at sweep rates ranging from 1 to 4095 seconds. It provides an unlimited number of sweeps, which are interruptable at any time.

**The NMR Simulator** will calculate, display and plot theoretical NMR spectra of any system of spin-1/2 nuclei containing up to six spins. Its inputs, in addition to chemical shifts and coupling constants, are sweep offset, sweep width, and spectrometer frequency. It will punch the calculated transitions on paper tape for later analysis, if desired. While the principle use of this program is for NMR, it is not limited to proton

range values. Shifts, coupling constants, width and offset may fall anywhere in the range of  $10^{-616}$  to  $10^{616}$  Hertz. This program differs from DECUS versions in that it contains a first Lorentzian line shape routine and plot routines for both the stick figure and Lorentzian curve spectre. It also features more explicit terminal commands and input queries.

**Histogram Application Programs** are available for neurophysiologists studying the activity of single nerve cells. These programs aid in examining spontaneous and simulated activity by providing for the acquisition and analysis of spike train signals. The Time Interval Histogram Program yields a frequency distribution which is used to determine mean firing rates in spontaneously active nerve cells under different environmental conditions. The Post-Stimulus Histogram is a tool for monitoring and studying single cell response to a stimulus. It forms a histogram showing all post-stimulus activity by presenting the stimulus several times. The mean time of response can be derived directly from the histogram, and response characteristics of cells are represented graphically, for evaluation. The Latency Histogram Program gives a frequency distribution showing when the first activity occurs subsequent to a stimulus. To inform the user about any activity occurring outside the time frame of study, underflow and overflow channels record pulses which arrive before the set minimum time and after the set maximum time channels. Output data may be routed to a CRT display for smoothing, expansion, plotting or photographing. All three histogram programs provide variable resolution of the frequency distribution, user-specified minimum time (to eliminate stimulus artifacts without sacrificing memory), overflow and underflow channels, and sophisticated computer output.

**The Auto- and Cross-Correlation Application Package** is designed to correlate data at sampling rates ranging from 0.1 to 204.7 milliseconds, on-line, with the user controlling all parameters from the console terminal. It displays and scales data while computing, and provides output that can be post-processed with FOCAL or user programs.

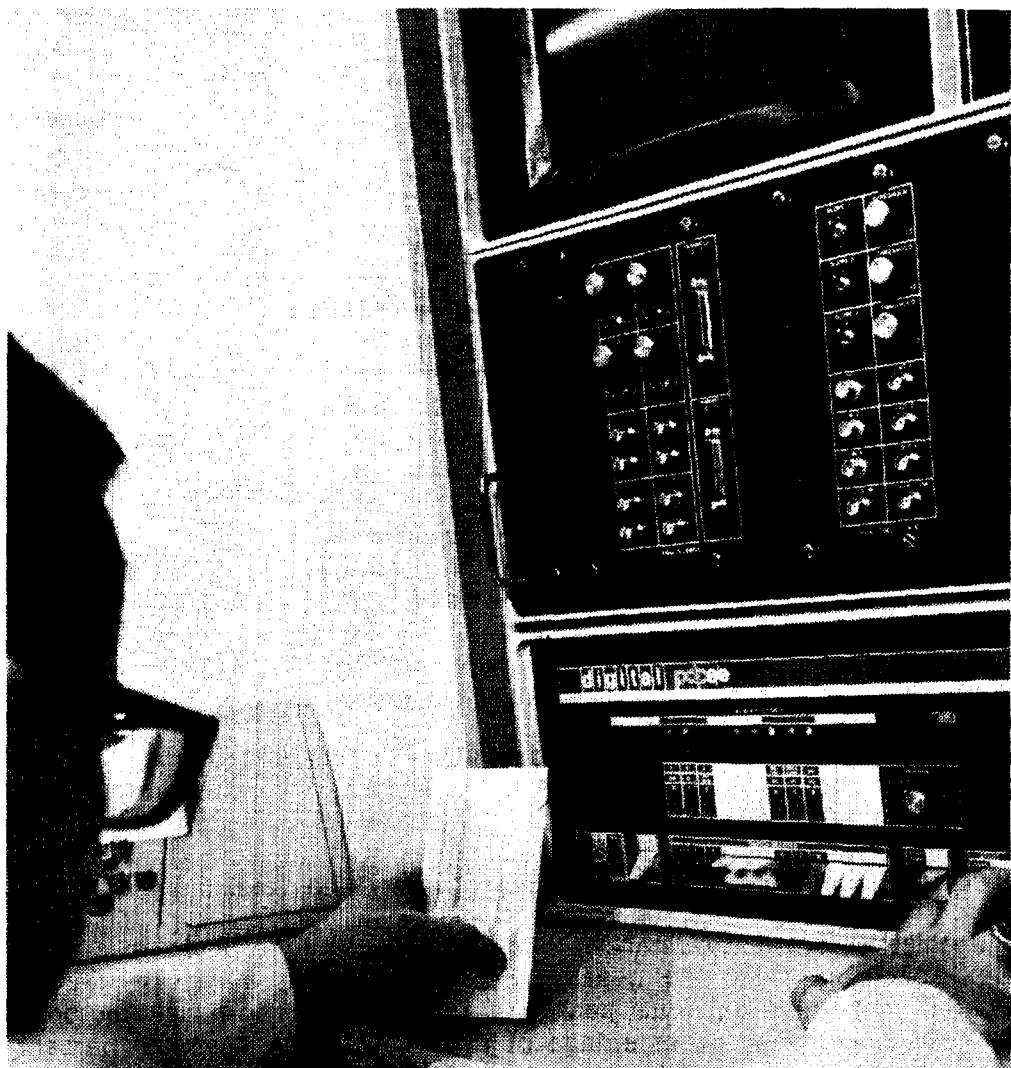
Correlation, as it applies to waveforms, may be used to detect periodic signals buried in noise or to provide a measure of similarity between two waveforms. Auto-correlation measures the similarity of a signal to a time-delayed version of itself, while cross-correlation measures the degree of similarity between one source input and a second source. No synchronizing event, such as the trigger required in signal averaging, need be available for the application of correlation techniques.

**DAQUAN** is a program for data acquisition in the time domain and/or general purpose data reduction. DAQUAN is used to acquire data by boxcar, multisweep signal averaging. After the data is acquired, a wide variety of subsequent processing techniques such as smoothing, differentiation, or integration may be used to reduce the data.

A special feature of DAQUAN is its ability to determine peaks in a complex spectrum. Once the peaks have been defined, a report may be printed containing individual peak information. This report includes peak minima, maxima as a percentage of the largest peak, and the percent area.

The command structure of DAQUAN is flexible and easy to use. For example, only a single command is required to perform such operations as aligning a sloping baseline, scaling a spectrum, performing many integrations or comparing two spectra simultaneously.

**BASIC/RT** is a laboratory version of the language developed at Dartmouth College. In addition to normal computational features, BASIC/RT incorporates new instructions for servicing the analog-to-digital converter, real-time clock and point-plotting CRT display in a real-time data acquisition environment. The language is implemented as an incremental compiler which converts BASIC statements into machine code as they are typed at the console terminal. This feature saves considerable debugging time, because a program is ready to run as soon as it has been typed at the console. Some of the special instructions used to perform real-time data acquisition and reduction tasks in the BASIC language are: ACCEPT, ADC (sample the A/D converter), REJECT, REAL TIME, ADB (retrieve data), DATA, SET CLOCK, SET RATE, WAIT, WAITC, USE, and PLOT X, Y.



# 3

## system operation

### PROGRAMMER'S CONSOLE OPERATION

The switches and indicators on the PDP-8/E programmer's console are designed to allow manual control over the detailed operation of the computer and present a convenient indication of program conditions within the machine. The PDP-8/E may be programmed manually, by means of switches on the programmer's console, and program execution may be started, stopped, monitored, or toggled between various modes of operation. The console switches also provide a convenient means of selecting a memory location for examination and selectively modifying the content of memory.

The indicator lights on the programmer's console provide a continuous display of the logical state of major registers, busses and control flip-flops inside the PDP-8/E, as well as several important registers contained in commonly used processor options, such as the extended arithmetic element. Figure 3-1 shows the KC8-EA Programmer's Console, which is typical of the models available. Table 3-1 describes the function of the various switches and indicators. This table is intended as a reference for the advanced programmer or system operator; most users will want to be thoroughly familiar with the remainder of this chapter before attempting to operate the programmer's console.

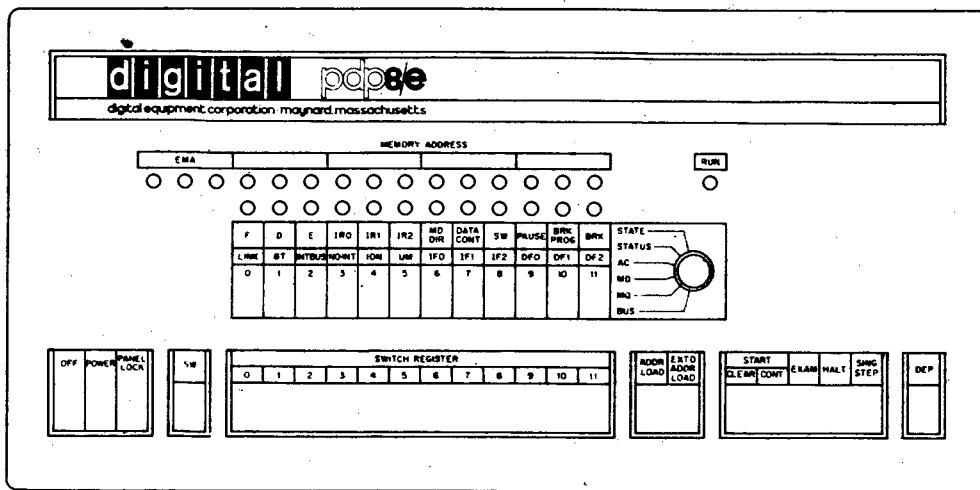


Figure 3-1. PDP-8/E Programmer's Console

**Table 3-1 Programmer's Console Control and Indicator Functions**

CONTROL OR INDICATOR	FUNCTION
<b>OFF/POWER/ PANEL LOCK</b>	In the counter-clockwise, or OFF position, this key operated switch disconnects all primary power to the computer. In the POWER, or vertical position, it applies power to the computer and all manual controls. In the PANEL LOCK, or clockwise position, it applies power to the computer, the switch register, the SW switch and the RUN indicator only. In this position, a running program is protected from inadvertent switch operation.
<b>SW</b>	When the SW switch is up, the OMNIBUS SW line is disabled (logical 1, or voltage level high). When it is down, the SW line is asserted. This switch is used by certain peripheral options such as the MI8-E Bootstrap Loader.
<b>ADDR LOAD</b>	Pressing the ADDRess LOAD switch loads the contents of the SR into the CPMA register and enables the FETCH major state for the next processor cycle (which will begin when the RUN indicator is lit).
<b>EXTD ADDR LOAD</b>	Pressing the EXTendeD ADDRess LOAD switch loads the content of SR bits 6-8 into the instruction field register and the content of SR bits 9-11 into the data field register. The instruction and data field registers are contained in the KM8-E Memory Extension and Time Share option.
<b>CLEAR</b>	Pressing the CLEAR switch generates an INITIALIZE pulse that loads a binary 0 into bits 0-11 of the AC, the link, all I/O device flag registers and all interrupt system flip-flops. This is equivalent to executing a CAF instruction.
<b>CONT</b>	Pressing the CONTinue switch sets the RUN flip-flop and issues a MEM START L signal to begin program execution at the memory location addressed by the current content of the CPMA register.
<b>EXAM</b>	Pressing the EXAMine switch loads the contents of the memory location addressed by the current content of the CPMA register into the MB register and then increments the CPMA and PC registers. Repeated operation of this switch permits the content of sequential memory locations to be examined.

<b>HALT</b>	Pressing the HALT switch clears the RUN flip-flop, causing the computer to stop at the beginning of the next FETCH cycle. Operating the computer with the HALT switch depressed causes one complete instruction to be executed whenever the CONTINUE switch is pressed.
<b>SING STEP</b>	Pressing SINGLE STEP clears the RUN flip-flop and causes the computer to stop at TS1 of the next machine cycle. Operating the computer with the SINGLE STEP switch depressed causes one machine cycle to be executed whenever the CONTINUE switch is pressed.
<b>DEP</b>	Lifting the spring-loaded DEPOSIT switch loads the content of the SR into the MB register and into memory at the address specified by the current content of the CPMA register, then increments the CPMA and PC registers. Use of the DEPOSIT switch facilitates manual storage of information in sequential memory locations.
<b>EMA</b>	The Extended Memory Address register displays the content of the 3-bit EMA bus (EMA0-2) contained on the OMNIBUS. EMA0-2 normally carries the memory field designation of the memory field being accessed.
<b>MEMORY ADDRESS</b>	The MEMORY ADDRESS register displays the content of the 12-bit MA bus (MA0-11) contained on the OMNIBUS. It combines with the EMA register to provide the 15-bit address of the next memory location to be accessed.
<b>RUN</b>	The RUN indicator lamp is lighted to show that the RUN flip-flop is set, and all machine circuits are activated and capable of executing instructions.
<b>Indicator Selector Switch</b>	This 6-position rotary switch designates which of six possible registers (or combinations of registers) is to be gated into the adjacent 12-bit display. Setting the Indicator Selector Switch to:
<b>BUS</b>	Displays the logical state of the 12-bit DATA bus (DATA0-11) contained on the OMNIBUS.
<b>MQ</b>	Displays the content of the Multiplier Quotient register.
<b>MD</b>	Displays the logical state of the 12-bit MEMORY DATA bus (MD0-11) contained on the OMNIBUS. This bus normally carries the content of the last memory location addressed by the EMA and MEMORY ADDRESS registers.

<b>AC</b>	Displays the content of the accumulator.
<b>STATUS</b>	Each display lamp is lighted to indicate the designated condition:
<b>INDICATOR LAMP/BIT POSITION</b>	<b>TURNED ON TO INDICATE:</b>
0	The link is set.
1	The Greater Than Flag (GTF) is set. The GTF is contained in the KE8-E Extended Arithmetic Element.
2	The OMNIBUS interrupt request line is asserted.
3	The interrupt inhibit flip-flop is set. The interrupt inhibit flip-flop is contained in the KM8-E Memory Extension and Time Share option.
4	The interrupt system is enabled.
5	The USER MODE line is asserted. Signal USER MODE L originates in the time share portion of the KM8-E Memory Extension and Time Share option to disable execution of all OSR, LAS, IOT and HLT instructions when the computer is operated in a timesharing environment.
6-8	Displays the content of the 3-bit instruction field register (IFO-2) contained in the KM8-E Memory Extension and Time Share option.
9-11	Displays the content of the 3-bit data field register (DFO-2) contained in the KM8-E Memory Extension and Time Share option.
<b>STATE</b>	Each display lamp is lighted to indicate the designated condition:
<b>INDICATOR LAMP/BIT POSITION</b>	<b>TURNED ON TO INDICATE:</b>
0	FETCH major state is enabled.
1	DEFER major state is enabled.
2	EXECUTE major state is enabled.
3-5	Displays the content of the 3-bit instruction register (IRO-2).

- 6 Displays the state of the MD DIR line on the OMNIBUS. Signal MD DIR is high (and the lamp is lighted) during operations that read data from memory. MD DIR is low (and the lamp is extinguished) during operations that write data into memory.
- 7 Displays the state of the BREAK DATA CONT line on the OMNIBUS. Signal BREAK DATA CONT is low (and the lamp is lighted) during an ADM operation.
- 8 The SW line on the OMNIBUS is asserted. This can only occur when the programmer's console SW switch is depressed.
- 9 The PAUSE line on the OMNIBUS is asserted. Signal I/O PAUSE L is generated during IOT instruction execution.
- 10 The BREAK IN PROG line on the OMNIBUS is asserted, indicating that one or more devices are requesting a data break. The highest priority device will begin a DMA operation at the beginning of the following cycle.
- 11 The BREAK CYCLE line on the OMNIBUS is asserted, indicating that the processor is currently performing a DMA operation under the control of a peripheral device.

---

#### MEMORY ORGANIZATION

PDP-8/E memory is divided into 4096-word blocks called memory fields. The memory fields are numbered sequentially from field 0, which is the first 4096 words of memory supplied with the basic system, up to field 7, if a full 32K of memory is installed. Within each memory field, the 4096 storage locations are numbered sequentially, in octal, from 0000 to 7777. This 4-digit octal number is called the 12-bit address of the memory location. In any given memory field, every storage location has a unique 12-bit address.

Each memory field is further subdivided into 32 pages of 128 words each. Memory pages are numbered sequentially, in octal, from page 0 (which contains addresses 0000 to 0177) to page 37 (addresses 7600 to 7777). Within each memory page, the 128 locations on the page are numbered sequentially, in octal, from 0 to 177. This number is called the page address of the memory location. Page addresses are not redundant; the page address of a memory location is simply the octal value of the low-order 7 bits of the 12-bit address.

The first five bits of a 12-bit memory address are called the page bits. The octal value of the page bits for any memory address is identical to the number of the memory page on which the address is located. The last seven bits of the 12-bit address are called the page address bits.

The octal value of the page address bits for any memory address is identical to the page address of the memory location. Thus, location 4716 is at page address 116 on page 23, while location 2257 is at page address 057 on page 11.

Unlike memory fields, which may be physically separated by being located on different modules plugged into the OMNIBUS, memory pages do not correspond to any physical separation within memory. The computer has no way of recognizing which page of memory it is executing

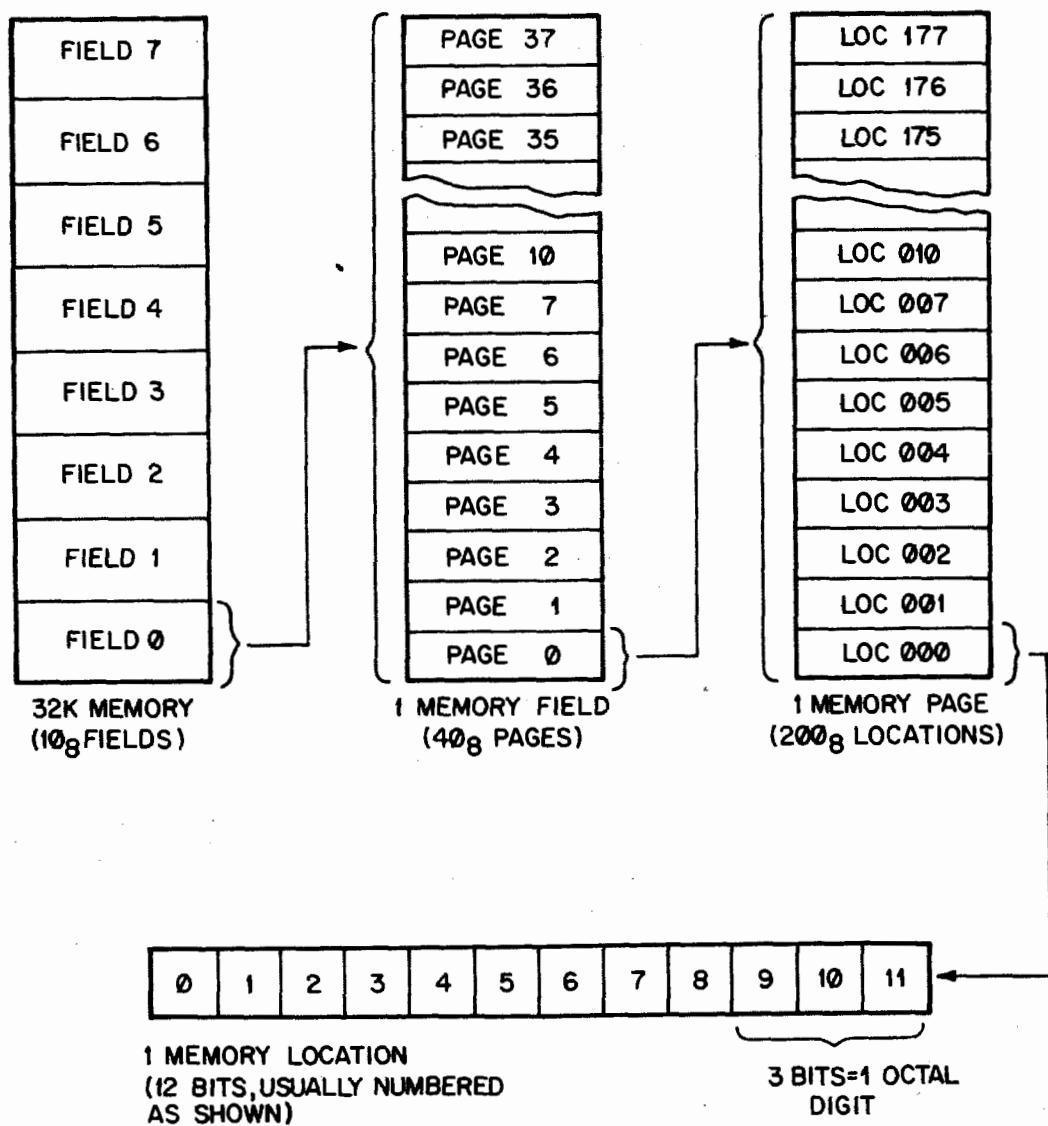


Figure 3-2. PDP-8/E Memory Organization

in, and it is not cognizant of executing across a page boundary. Memory pages represent a more or less artificial subdivision of memory that facilitates understanding the PDP-8/E memory reference instruction decoding process.

The individual bits of a PDP-8/E memory word are usually numbered, for reference purposes, as shown in Figure 3-2. The bits of major registers are numbered in the same manner, but the abbreviated register name is prefixed to the number for identification purposes. Thus, bit 0 is always the high-order bit of a memory word, while AC0 is the high-order bit of the accumulator and PC11 is the low-order bit of the program counter.

### **MEMORY AND PROCESSOR INSTRUCTIONS**

A PDP-8/E instruction is a single, 12-bit word, stored in memory, that tells the computer to perform a specific operation or sequence of operations. Like most stored program computers, the PDP-8/E makes no distinction between instructions and data; it will manipulate instructions as though they are stored variables or attempt to execute data as instructions if it is programmed to do so. The 12-bit value that tells the computer to execute a specified instruction is called the octal code for that instruction. In addition to its unique octal code, every instruction has an assigned mnemonic, which is a 3- or 4-character name that may be supplied to an assembler program to generate the corresponding octal code. There are three general classes of PDP-8/E instructions, each of which is handled somewhat differently by the central processor.

Memory reference instructions, or MRIs are instructions that cause the computer to operate on the content of a memory location, or to use the content of a memory location to operate on the accumulator. Every MRI specifies an operation, which is coded in the first 3 bits of the instruction, and the address of an operand, which is coded in the last 9 bits. There are five PDP-8/E memory reference instructions. Typical applications of MRIs include depositing the content of the AC at a specified address in memory, or jumping to a subroutine with a specified entry address.

Augmented instructions cause the computer to perform a logical (non-arithmetic) operation on the content of one of the major registers. Typical applications of augmented instructions include rotating the AC right or left, testing the content of the AC or link, loading an I/O device buffer from the AC and operating the I/O device, or initializing and operating the interrupt system. Since augmented instructions do not reference a memory address, all 12 bits of the instruction are available for coding the precise operation or sequence of operations to be performed.

There is one housekeeping instruction that comprises the third class of PDP-8/E instructions. This instruction is similar to the MRIs, in that it references a memory address, but similar to the augmented instructions in the manner in which it is executed. It is used to load the PC with a specified memory address, so that the instruction stored at this address will be the next instruction to be executed.

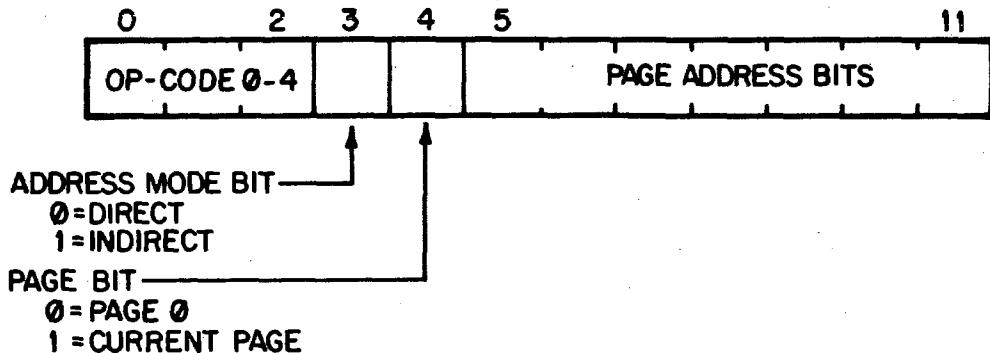


Figure 3-3. Memory Reference Instruction Format

#### **Memory Reference Instructions**

Every memory reference instruction contains an operation code, or OP-code, that occupies the first 3 bits of the instruction and an address code that occupies the last 9 bits. This format is illustrated in Figure 3-3. The OP code of an MRI is one of the digits 0 to 4, corresponding to one of five possible operations. The address code specifies the address of an operand, if the instruction is directly addressed, or the address of a pointer to the operand, if the instruction is indirectly addressed.

Bit 3 of an MRI is called the address mode bit. If this bit is set (contains a 1), the MRI is indirectly addressed. This means that the address code of the MRI specifies the page address of a memory location in which the 12-bit address of the operand is stored. If the address mode bit is not set, the instruction is directly addressed. In this case, the address code specifies the page address at which the operand itself is stored.

Bit 4 of an MRI is called the page bit, and bits 5-11 are the page address bits. If the page bit is set, the page address bits contain a page address in the memory page on which the MRI itself is stored (called the current page). If the page bit is not set, the page address bits contain a page address on page 0. In either case, the address specified by the page bit and the page address bits will be the address of the operand, if the MRI is directly addressed, or the address of a memory location that contains the 12-bit address of the operand, if the instruction is indirectly addressed.

In this manner, an MRI may address any one of 400 (octal) locations directly, unless it is stored on page zero. If the MRI is stored in one of the locations 0000-0177, the current page is page zero and the MRI may only address 200 (octal) locations directly. An MRI may address any of 7777 (octal) locations indirectly, however the pointer to the addressed location must reside on page 0 or the current page.

Table 3-2 lists the mnemonics for the five memory reference instructions, their octal codes, and the operations they perform. Only the first 3 bits of the octal codes are listed explicitly; the remaining 9 bits make up the address code, which depends upon where in memory the operand for the MRI is stored.

**Table 3-2 Memory Reference Instructions**

MNEMONIC	OCTAL	OPERATION
AND	0XXX	<b>Logical AND.</b> The content of the memory location specified by XXX is combined with the content of the AC by a bitwise logical AND operation. The result is left in the AC, the operand is restored to memory, and the original content of the AC is lost. This instruction, often called "extract" or "mask", may also be considered as a bit-by-bit binary multiplication.
TAD	1XXX	<b>Two's Complement Add.</b> The content of the memory location specified by XXX is combined with the content of the AC by two's complement addition. The result is left in the AC, the operand is restored to memory, and the original content of the AC is lost. If there is a high-order carry from AC0, the link is complemented.
ISZ	2XXX	<b>Increment and Skip if Zero.</b> The content of the memory location specified by XXX is incremented by 1 and restored to memory. If the content of the referenced location becomes zero, the PC is incremented by 1 to skip the next sequential instruction. If the content of the referenced location does not become zero, the next instruction is executed.
DCA	3XXX	<b>Deposit and Clear the Accumulator.</b> The content of the AC is stored in the memory location specified by XXX and the AC is set to zero. The original content of the referenced memory location is lost.
JMS	4XXX	<b>Jump to Subroutine.</b> The content of the PC is stored in the memory location specified by XXX. The PC is then loaded with 1 more than the address of this location (XXX+1), so that the instruction stored in the memory location following the referenced location is the next instruction to be executed. The content of the AC is not affected.

#### The Housekeeping Instruction

The only housekeeping instruction is the JMP instruction, with an OP-code of 5, whose format is illustrated in Figure 3-4. Table 3-3 lists the octal code for this instruction and describes the operation it performs.

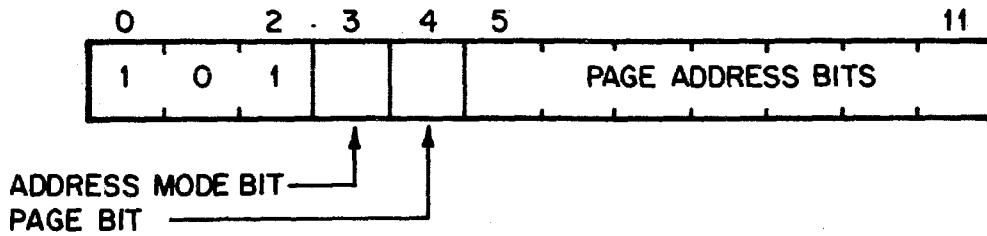


Figure 3-4. Jump Instruction Format

Table 3-3 The Housekeeping Instruction

MNEMONIC	OCTAL	OPERATION
JMP	5XXX	Jump. The 12-bit address of the memory location specified by XXX is loaded into the PC, so that the instruction stored at this address will be the next instruction to be executed. The original content of the PC is lost. The content of the AC is not affected.

#### AUGMENTED INSTRUCTIONS

The two augmented instructions are the input/output transfer instruction and the operate instruction. Input/output transfer instructions, which have an OP-code of 6, provide for communication between the central processor and all peripheral devices. They are also used to communicate with the interrupt system. Operate instructions, with an OP-code of 7, are used to perform logical operations on the content of the major registers.

#### The Input/Output Transfer Instruction

Input/output transfer (IOT) instructions are used to initiate the operation of peripheral devices and to transfer data between peripherals and the central processor. Figure 3-5 shows the format of an IOT instruction. Bits 0-2 contain the OP-code, which must be 6 to specify an IOT instruction. Bits 3-8 contain a device selection code that is transmitted to every peripheral device whenever the IOT instruction is executed. Device selectors within the peripheral devices monitor these device codes. When a peripheral device recognizes a device code as that peripheral's assigned code, the device accepts the last three bits of the IOT instruction.

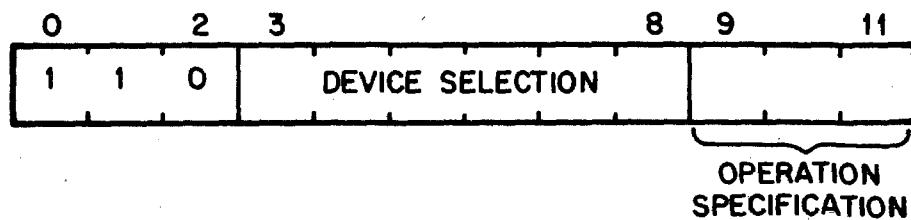


Figure 3-5. IOT Instruction Format

Bits 9-11 of an IOT instruction contain the operation specification code. These bits may be set to specify one of up to eight operations. If a peripheral device is capable of performing more than eight different operations, it is necessary to assign two or more device codes to the peripheral device.

### The Operate Instruction

The operate instruction consists of 3 groups of microinstructions. Group 1 microinstructions, which are identified by the presence of a 0 in bit 3, are used to perform logical operations on the content of the accumulator and link. Group 2 microinstructions, which are identified by the presence of a 1 in bit 3 and a 0 in bit 11, are used primarily to test the content of the accumulator and link, then conditionally skip the next sequential instruction. Group 3 microinstructions have a 1 in bit 3 and a 1 in bit 11. They are used to perform logical operations on the content of the accumulator and multiplier quotient registers.

Operate microinstruction from any group may be microprogrammed with most other operate microinstructions of the same group. The octal code for a microprogrammed combination of two (or more) microinstructions is the bitwise logical OR of the octal codes for the individual microinstructions. When more than one operation is microprogrammed into a single instruction, the operations are performed in a prescribed sequence, with logical sequence 1 microinstructions performed first, then logical sequence 2 microinstructions, and so on. Two operations with the same logical sequence number are performed simultaneously.

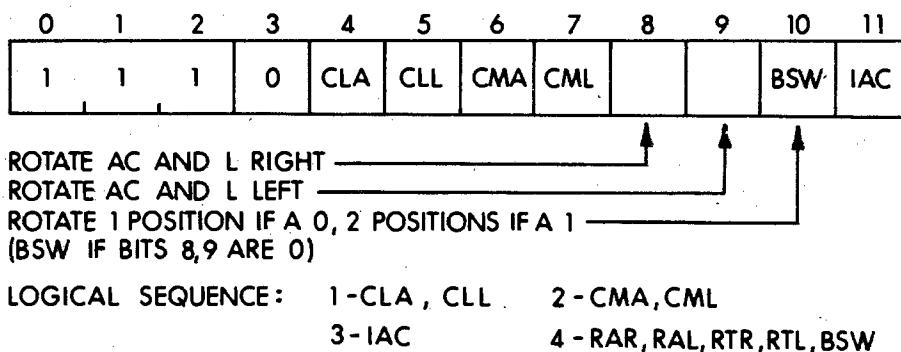


Figure 3-6. Group 1 Operate Microinstructions

### GROUP 1 MICROINSTRUCTIONS

Figure 3-6 shows the format of a group 1 microinstruction. The OP-code must be 7, to indicate an operate instruction, and bit 3 must contain a 0, to indicate a group 1 microinstruction. Any one of bits 4 to 11 may be set (loaded with a binary 1) to indicate a specific group 1 microinstruction. If more than one of these bits is set, the instruction is a microprogrammed combination of group 1 microinstructions, which will be executed according to the logical sequence shown in Figure 3-6.

Table 3-4 lists the group 1 microinstructions, their assigned mnemonics and the operations they perform. Two or more of these microinstructions may be microprogrammed into a single 12-bit instruction, as long as the instruction does not contain more than 1 of the logical sequence 4 microinstructions (RAR, RAL, RTR, RTL and BSW). This restriction should not impose any constraint on the user, since the five logical sequence 4 microinstructions perform mutually incompatible operations.

**Table 3-4 Group 1 Operate Microinstructions**

MNEMONIC	OCTAL	OPERATION
NOP	7000	<b>No Operation.</b> This instruction causes a 1-cycle delay in program execution, without affecting the state of the computer. It may be used for timing synchronization or as a convenient means of deleting another instruction from a program.
IAC	7001	<b>Increment Accumulator.</b> The content of the accumulator is incremented by 1.
BSW	7002	<b>Byte Swap.</b> The content of the six low-order bits of the AC is exchanged with the content of the six high-order bits. That is, AC0 is exchanged with AC6, AC1 is exchanged with AC7, etc. The content of the link is not affected.
RAL	7004	<b>Rotate Accumulator Left.</b> The content of AC1-11 is shifted into AC0-10. The content of AC0 is shifted into the link, and the content of the link is shifted into AC11.
RTL	7006	<b>Rotate Two Left.</b> Equivalent to two consecutive RAL operations.
RAR	7010	<b>Rotate Accumulator Right.</b> The content of AC0-10 is shifted into AC1-11. The content of the link is shifted into AC0, and the content of AC11 is shifted into the link.
RTR	7012	<b>Rotate Two Right.</b> Equivalent to two consecutive RAR operations.
CML	7020	<b>Complement Link.</b> The content of the link is complemented.
CMA	7040	<b>Complement Accumulator.</b> The content of each bit of the AC is complemented. This has the effect of replacing the content of the AC with its one's complement.
CLL	7100	<b>Clear Link.</b> The link is loaded with a binary 0.
CLA	7200	<b>Clear Accumulator.</b> Each bit of the AC is loaded with a binary 0.

Table 3-5 lists four microprogrammed combinations of group 1 microinstructions which are used so frequently that they have been assigned their own mnemonics. Note that the octal codes for a microprogrammed combination of operate microinstructions is the bitwise logical OR of the octal codes of the individual microinstructions. Other frequently used combinations of operate microinstructions are listed in the appendix of this handbook.

**Table 3-5 Microprogrammed Combinations of Group 1 Microinstructions**

MNEMONIC	OCTAL	OPERATION
CIA	7041	<b>Complement and Increment Accumulator.</b> The content of the AC is replaced with its two's complement. This is a microprogrammed combination of CMA and IAC.
STL	7120	<b>Set the Link.</b> The link is loaded with a binary 1. This is a microprogrammed combination of CLL and CML.
STA	7240	<b>Set the Accumulator.</b> Each bit of the AC is loaded with a binary 1. This is a microprogrammed combination of CLA and CMA.
GLK	7204	<b>Get the Link.</b> The AC is cleared and the content of the link is shifted into AC11 while a 0 is shifted into the link. This is a microprogrammed combination of CLA and RAL.

#### GROUP 2 MICROINSTRUCTIONS

Figure 3-7 shows the format of a group 2 microinstruction. The operation code must be 7, to indicate an operate instruction, and bit 3 must contain a 1 while bit 11 must contain a 0, to indicate a group 2 microinstruction. Bits 4-10 may be set to indicate a specific group 1 microinstruction. If more than one of bits 4-7 or 9-10 is set, the instruction is a microprogrammed combination of group 2 microinstructions, which will be executed according to the logical sequence shown in Figure 3-7. Table 3-6 lists the group 2 microinstructions, their mnemonics, and the operations they perform.

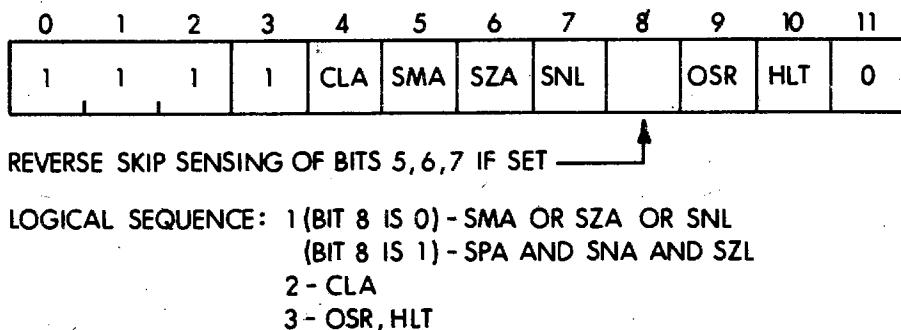


Figure 3-7. Group 2 Operate Microinstructions

**Table 3-6 Group 2 Microinstructions**

MNEMONIC	OCTAL	OPERATION
HLT	7402	Halt. Clears the run flip-flop so that program execution stops at the end of TP4 of the current machine cycle.
OSR	7404	Logical OR with Switch Register. The content of the programmer's console switch register (SR) is combined with the content of the AC by a bitwise logical OR operation. The result is left in the AC and the original content of the AC is lost. The content of the SR is not affected.
SKP	7410	Skip. The content of the PC is incremented by 1, to skip the next sequential instruction.
SNL	7420	Skip on Non-Zero Link. The content of the link is sampled. If the link contains a 1, the content of the PC is incremented to skip the next sequential instruction. If the link contains a 0, the next instruction is executed.
SZL	7430	Skip on Zero Link. The content of the link is sampled. If the link contains a 0, the content of the PC is incremented to skip the next sequential instruction. If the link contains a 1, the next instruction is executed.
SZA	7440	Skip on Zero Accumulator. The content of each bit of the AC is sampled. If every bit contains a 0, the content of the PC is incremented to skip the next sequential instruction. If any bit contains a 1, the next instruction is executed.
SNA	7450	Skip on Non-Zero Accumulator. The content of each bit of the AC is sampled. If any bit contains a 1, the content of the PC is incremented by 1 to skip the next sequential instruction. If every bit contains a 0, the next instruction is executed.
SMA	7500	Skip on Minus Accumulator. The content of ACO is sampled. If ACO contains a 1, indicating that the AC contains a negative two's complement number, the content of the PC is incremented to skip the next sequential instruction. If ACO contains a 0, the next instruction is executed.

**Table 3-6 Group 2 Microinstructions (Cont.)**

MNEMONIC	OCTAL	OPERATION
<b>SPA</b>	<b>7510</b>	Skip on Positive Accumulator. The content of AC0 is sampled. If AC0 contains a 0, indicating that the AC contains a positive two's complement number (or zero), the content of the PC is incremented to skip the next sequential instruction. If AC0 contains a 1, the next instruction is executed.
<b>CLA</b>	<b>7600</b>	Clear Accumulator. Each bit of the AC is loaded with a binary 0.

Skip microinstructions may be microprogrammed with CLA, OSR or HLT microinstructions, and also with other skip microinstructions that have the same value in bit 8. Skip microinstructions which have a 0 in bit 8 may not be microprogrammed with skip microinstructions which have a 1 in bit 8, however.

When two or more skip microinstructions are microprogrammed into a single instruction, the resulting condition on which the decision will be based is the logical OR of the individual conditions when bit 8 is 0, or the logical AND of the individual conditions when bit 8 is 1 (see Figure 3-7).

Table 3-7 lists every legal combination of skip microinstructions, along with the resulting condition upon which the decision to skip or execute the next sequential instruction is based. This table does not include microprogrammed combinations of skip microinstructions and the CLA, OSR or HLT microinstructions.

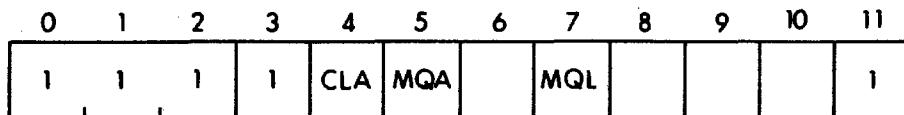
**Table 3-7 Microprogrammed Combinations of Group 2 Microinstructions**

MNEMONIC	OCTAL	OPERATION
<b>SZA    SNL</b>	<b>7460</b>	Skip if AC = 0 or L = 1 or both.
<b>SNA    SZL</b>	<b>7470</b>	Skip if AC = 0 and L = 0.
<b>SMA    SNL</b>	<b>7520</b>	Skip if AC < 0 or L = 1 or both.
<b>SPA    SZL</b>	<b>7530</b>	Skip if AC $\geq 0$ and L = 0.
<b>SMA    SZA</b>	<b>7540</b>	Skip if AC $\leq 0$ .
<b>SPA    SNA</b>	<b>7550</b>	Skip if AC > 0.
<b>SMA    SZA    SNL</b>	<b>7560</b>	Skip if AC $\leq 0$ or L = 1 or both.
<b>SPA    SNA    SZL</b>	<b>7570</b>	Skip if AC > 0 and L = 0.

### GROUP 3 MICROINSTRUCTIONS

Group 3 microinstructions are used to transfer data between the AC and multiplier quotient (MQ) registers. Although these microinstructions are intended primarily for use with the KE8-E Extended Arithmetic Element, they are also useful when the MQ is employed as a temporary storage register, even if an EAE is not installed.

Figure 3-8 shows the format of a group 3 microinstruction. The operation code must be 7, to indicate an operate instruction, while bits 3 and 11 must both contain a 1, to indicate a group 3 microinstruction. Any one of bits 4, 5 or 7 may be set to indicate a specific group 3 microinstruction. If more than one of these bits is set, the instruction is a microprogrammed combination of group 3 microinstructions.



LOGICAL SEQUENCE: 1 - CLA  
2 - MQA, MQL  
3 - ALL OTHERS

Figure 3-8. Group 3 Operate Microinstructions

Table 3-8 lists the three group 3 microinstructions, their assigned mnemonics, and the operations they perform. This table also lists two useful microprogrammed combinations of group 3 microinstructions.

Table 3-8 Group 3 Microinstructions

MNEMONIC	OCTAL	OPERATION
CLA	7601	Clear Accumulator. Each bit of the AC is loaded with a binary 0.
MQL	7421	Multiplier Quotient Load. The content of the AC is loaded into the MQ. The AC is cleared, and the original content of the MQ is lost.
MQA	7501	Multiplier Quotient into Accumulator. The content of the MQ is combined with the content of the AC by a bitwise logical OR operation, and the result is loaded into the AC. The original content of the AC is lost, but the original content of the MQ is not affected. Note that this instruction provides the programmer with a direct inclusive OR operation.

**Table 3-8 Group 3 Microinstructions (Cont.)**

MNEMONIC	OCTAL	OPERATION
<b>SWP</b>	<b>7521</b>	<b>Swap Accumulator and Multiplier Quotient.</b> The content of the AC and the content of the MQ are exchanged. This is a microprogrammed combination of MQA and MQL.
<b>CAM</b>	<b>7621</b>	<b>Clear Accumulator and Multiplier Quotient.</b> Each bit of both the AC and the MQ loaded with a binary 0. This is a microprogrammed combination of CLA and MQL.

#### **INSTRUCTION EXECUTION AND TIMING**

The major state generator provides control signals that may enable one of three major states during each memory cycle.

The FETCH major state is used to fetch an instruction from memory. FETCH is enabled whenever execution of an instruction was completed at the end of the last memory cycle. During a FETCH cycle, the processor reads an instruction from the memory location whose address is contained in the PC and decodes the first 3 bits of the instruction. If the instruction is an augmented instruction (OP-code 6 or 7) it is executed during the FETCH cycle. If the instruction is a JMP or memory reference instruction it is decoded further. Directly addressed JMP instructions will also be executed during the FETCH cycle; however, indirectly addressed JMP instructions and all MRIs require at least one additional cycle.

If an indirectly addressed JMP or MRI instruction was read from memory during the last FETCH cycle, the DEFER major state will be enabled during the following cycle. If a directly addressed MRI was read, the EXECUTE major state will be enabled next.

The DEFER major state is used to decode indirect memory references. During a DEFER cycle, the processor computes the 12-bit address of the memory location specified by bits 4-11 of the indirectly addressed JMP or MRI instruction and reads the address of an operand from this location. If the referenced location is an autoindex register, its content is incremented by 1 during the DEFER cycle, and the incremented value is taken as the operand address. Execution of an indirectly addressed JMP instruction will be completed during the DEFER cycle, but if the instruction is an indirectly addressed MRI, the EXECUTE major state must be enabled to complete execution during the following cycle.

Memory reference instruction execution is always completed during an EXECUTE cycle. The EXECUTE major state is entered from FETCH, when a directly addressed MRI is read from memory, or from DEFER, when the current instruction is an indirectly addressed MRI. In either case, instruction execution will be completed by the end of the EXECUTE cycle, and the FETCH major state will be enabled during the following cycle.

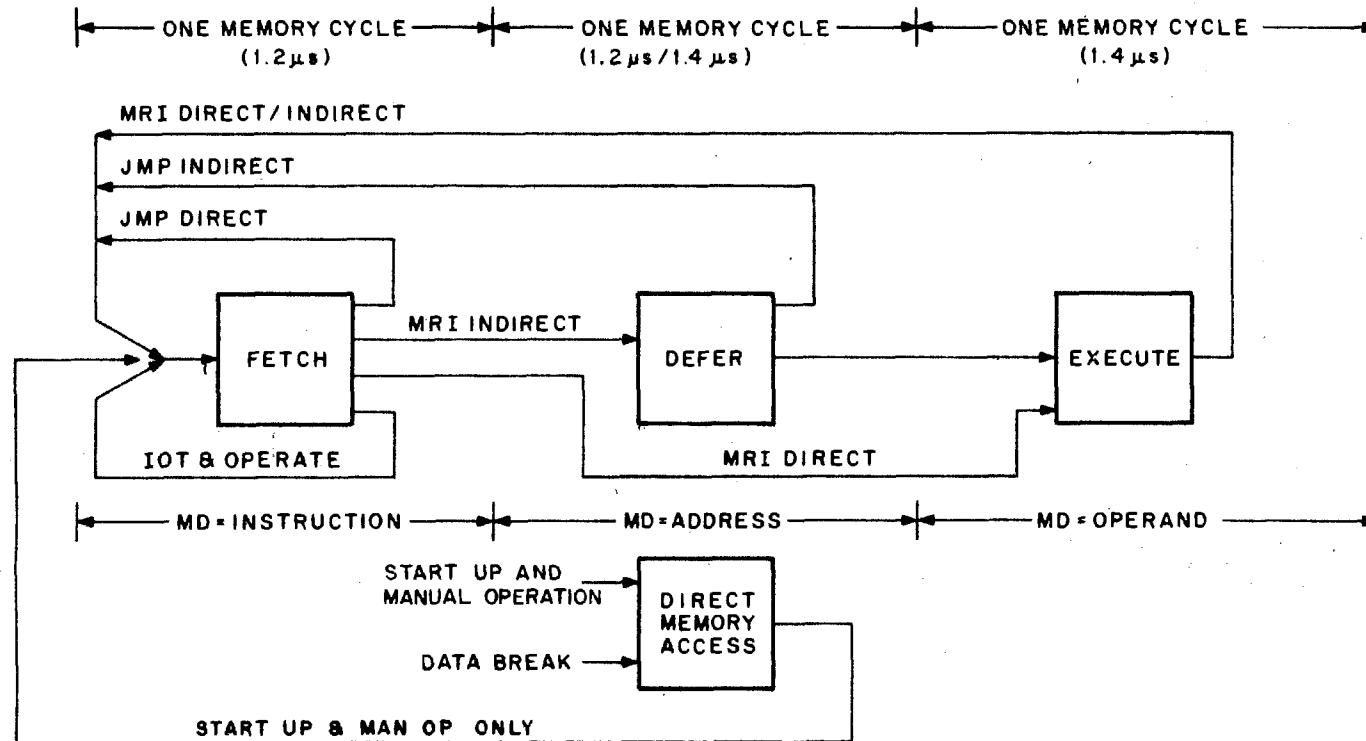


Figure 3-9. Major State Flow Diagram

A fourth major state, Direct Memory Access or DMA, is defined when neither FETCH, DEFER nor EXECUTE is enabled. The DMA state is entered during data break transfers and during manual operation of the switches on the programmer's console. Figure 3-9 is a diagram that illustrates major state flow as a function of instruction type. This diagram indicates which major states will be entered during execution of any given type of instruction.

#### Memory Timing

The timing generator produces four time state signals, designated TS1 through TS4, and four time pulse signals, designated TP1 through TP4. The timing diagram of Figure 3-10 illustrates the relationship between the time state and time pulse signals for a fast (1.2 microseconds) memory cycle. A slow (1.4 microseconds) memory cycle is produced when the EXECUTE major state is enabled or when the DEFER major state is enabled and the current instruction is an indirectly addressed MRI with autoindexing. Slow cycle timing is identical to fast cycle timing except that TS2 is extended for an additional 0.2 microseconds. All time state and time pulse signals are gated out onto the OMNIBUS where they are used as control signals throughout the system.

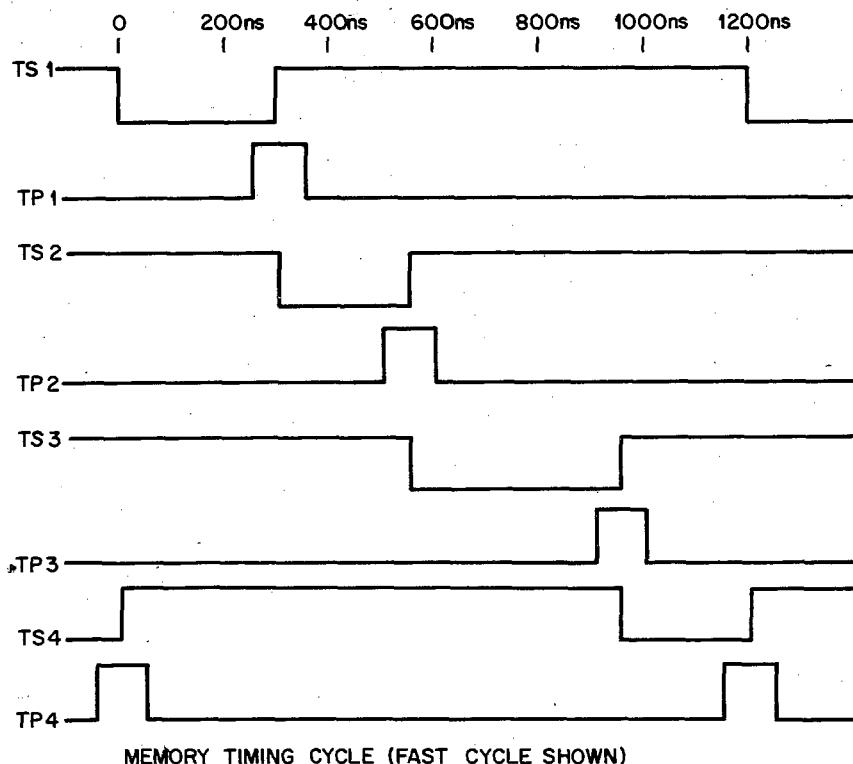


Figure 3-10. PDP-8/E Memory Timing Diagram  
(Fast Cycle)

#### FETCH Major State

Figure 3-11 is a simplified flow chart showing the general sequence of operations that occurs during every FETCH cycle. Notice that FETCH is always a fast cycle, and that the major state to be enabled during the next cycle depends on the type of instruction that is read from memory during the FETCH cycle.

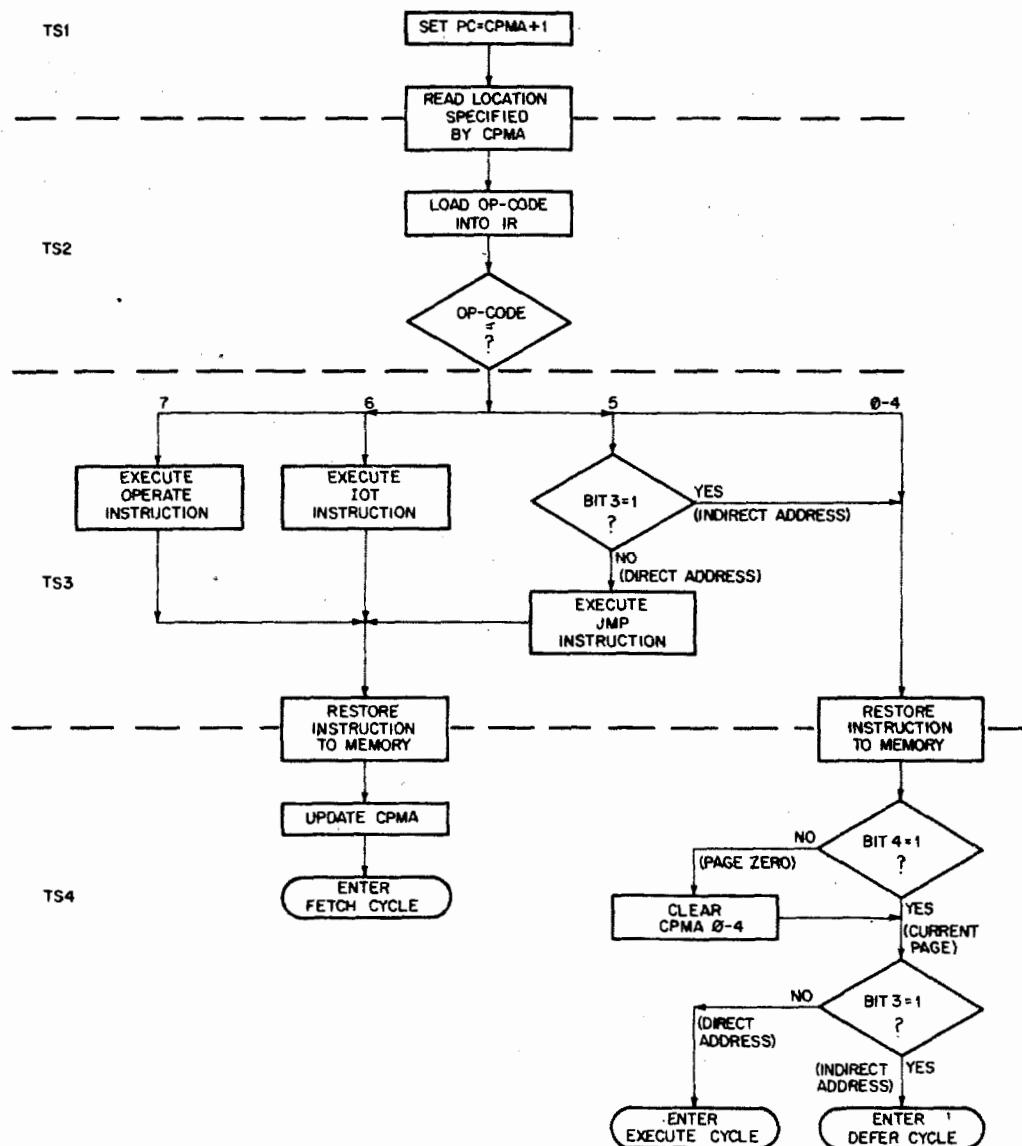


Figure 3-11. FETCH Major State

#### DEFER Major State

Figure 3-12 is a simplified flow chart showing the sequence of operations that occurs during a DEFER cycle. DEFER is enabled whenever the current instruction is an indirectly addressed JMP or memory reference instruction. It will be a slow cycle if the current instruction references one of the autoindex registers (locations 0010 to 0017) or a fast cycle in any other case. DEFER is always entered from the FETCH major state. A DEFER cycle will be followed by a FETCH cycle, if the current instruction is an indirect JMP, or by an EXECUTE cycle, if the current instruction is an indirectly addressed MRI.

### EXECUTE Major State

Figure 3-1 is a simplified flow chart showing the sequence of operations that occurs during an EXECUTE cycle. EXECUTE is entered from FETCH, if the current instruction is a directly addressed MRI, or from DEFER, if the instruction is an indirectly addressed MRI. An EXECUTE cycle is always followed by a new FETCH cycle.

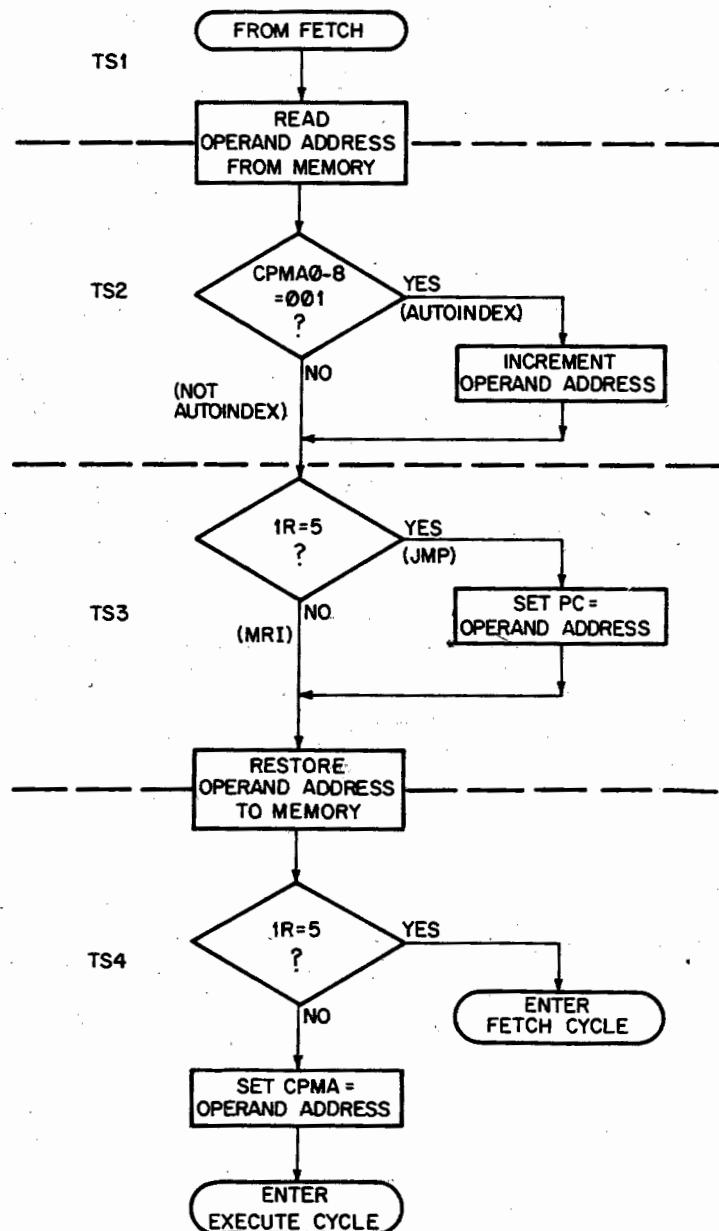


Figure 3-12. DEFER Major State

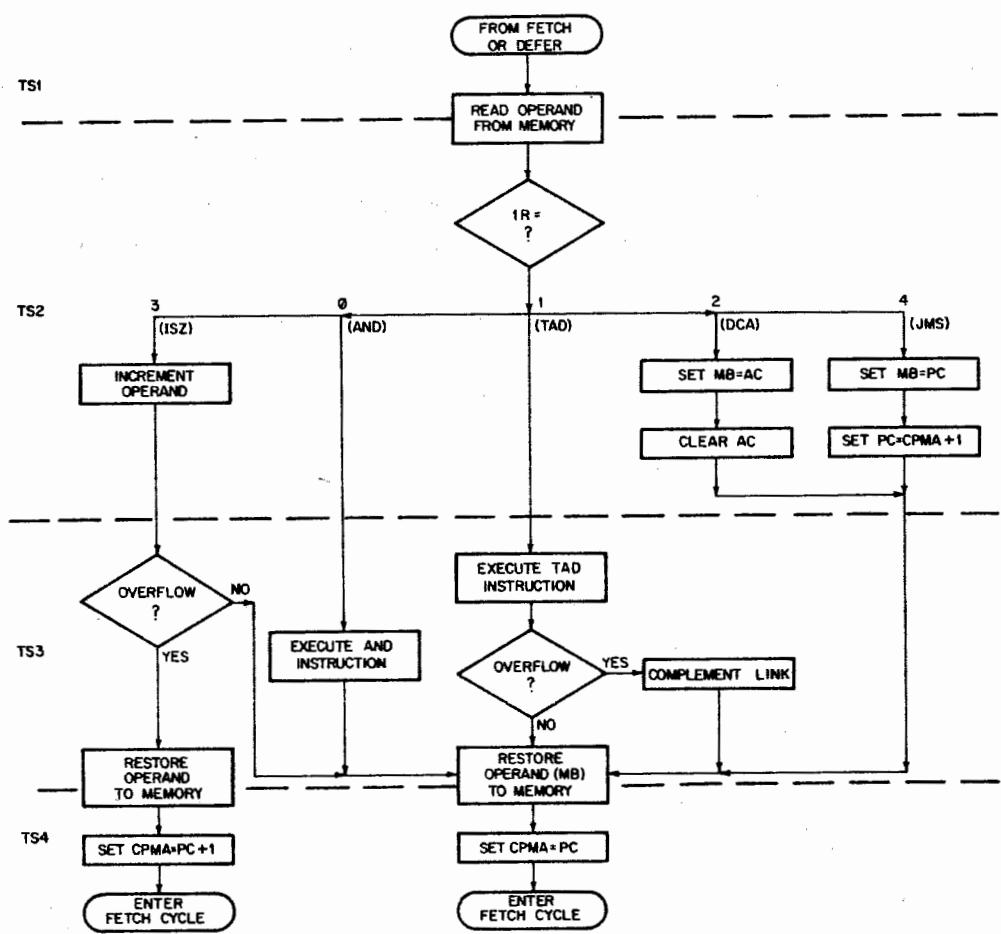


Figure 3-13. EXECUTE Major State

# 4

## input/output operations

Three types of data transfer may be used to receive or transmit information between the PDP-8/E and one or more peripheral I/O devices. Programmed data transfer provides a straight forward means of communicating with relatively slow I/O devices, such as the console terminal. Program interrupt transfers use the interrupt system to service several peripheral devices simultaneously, on an intermittent basis, permitting computational operations to be performed concurrently with the data I/O operations. Data break transfers rely on direct memory access to transfer variable-size blocks of data between high-speed peripherals and the processor with a minimum of program control. This choice of I/O techniques affords PDP-8/E users the means to execute data I/O operations at rates ranging from a few characters per second up to more than 10,000 characters per second, depending only upon the characteristics of a given peripheral device.

### **PROGRAMMED DATA TRANSFER**

Programmed data transfer is the easiest and most common means of performing data I/O. Every input/output transfer (IOT) instruction initiates one programmed transfer which may transmit data or status information either to or from a peripheral device. The amount of information that will be transferred by a single IOT instruction depends upon the particular operation that is coded into the instruction and the complexity of the I/O device interface. In general, programmed data transfers are limited to a maximum of 12 bits of data or 1 bit of status information per IOT instruction; however, there are many exceptions to this rule, and an I/O device may transfer any amount of data in response to a single IOT instruction if its interface circuitry is designed accordingly.

A programmed data transfer begins when the central processor reads an instruction from memory, loads the first three bits of the instruction into the instruction register, and recognizes that the current instruction is an IOT instruction, with an OP-code of 6. If the processor is operating in executive mode (i.e. not in a timesharing environment), it then concludes TS2 by transmitting a control signal to every peripheral device. This control signal instructs each peripheral to accept and decode bits 3-8 of the IOT instruction.

Bits 3:8 of every IOT instruction contain the device selection code that determines the specific I/O device for which the IOT instruction is intended. When an I/O device recognizes a device code as one of its assigned codes, it identifies itself to the processor as either an internal device whose interface control module plugs directly into the OMNIBUS or an external device that is connected to the positive I/O interface. The designated device then accepts and decodes bits 9-11 of the IOT instruction.

Bits 9-11 of every IOT instruction contain the operation specification code that determines the specific operation to be performed. The nature of this operation for any given IOT instruction depends entirely upon the circuitry designed into the I/O device interface; however, the usual practice is to use bit 11 (codes 0-1) to specify operations that test device status, bit 10 (codes 2-3) for operations that set or modify device status, and bit 9 (codes 4-7) for operations involving actual data transfer between the processor and the I/O device.

In order to simplify interface design for external (non-OMNIBUS) I/O devices, operation specification bits are transmitted to these devices as positive-going Input/Output Pulses (IOPs) generated by the KA8-E Positive I/O Interface. IOPs are simply the buffered, low-order 3 bits of the IOT instruction. If bit 11 of an IOT instruction is set, the external device receives an IOP1 pulse. Setting bit 10 generates an IOP2 pulse, while bit 9 generates an IOP4 pulse. If an IOT instruction ends in 0, none of the operation specification bits are set and no IOPs are generated. In any case, the device receives one additional pulse after the last IOP, as a signal to begin the specified I/O operation. IOPs are transmitted serially, so that edge-triggered logic may be used in the device interface. IOP spacing and duration are adjustable over a wide range; however, increasing the spacing or duration of the IOPs results in longer IOT execution time for all external devices.

#### **Peripheral Device Status**

Many I/O devices, including almost all serial devices such as the console terminal, maintain only one bit of status information. This is usually the state of a busy/done flip-flop which indicates whether the device is in the process of performing a data transfer or free to commence a new I/O operation. Thus, IOT instructions which set or modify device status often require that no information be transferred other than the operation specification bits of the IOT instruction.

If an information transfer is required, the I/O device must decode the operation specification bits to determine the exact nature of the transfer, gate the content of its device buffer onto the OMNIBUS if necessary, then generate up to three control signals which enable the adder circuits and shift gates of the central processor to perform one of six possible operations:

1. Data may be received from a device, ORed with the content of the AC, and the result loaded into the AC.
2. Data may be received from a device and added to the content of the PC.

3. Data may be received from a device and loaded into the PC.
4. The content of the AC may be sent to a device, and the AC may then be cleared.
5. Data may be received from a device and loaded into the AC.
6. The content of the AC may be sent to a device.

The six operations listed above are the only data transfer operations that may be performed during a programmed data transfer by any I/O device, but not all of these operations are performed by every device. Note that these operations are performed by circuitry in the central processor, controlled by signals generated at the I/O device interface. The maximum of three control signals an I/O device may generate for this purpose provides a total of eight data transfer operations, two of which are redundant.

All of the operations associated with a programmed date transfer must be completed by the end of TS3 of the FETCH cycle in which the IOT instruction was read from memory. If a peripheral device requires additional time to complete a data transfer, it may transmit a control signal that disables processor timing during some or all of the operations it is capable of performing. All peripheral devices connected to the positive I/O interface rely on this feature to extend IOT instruction timing for up to 3.4 microseconds, depending upon the nature of the operation being performed.

Since IOPS are transmitted serially, the time required to execute an external IOT instruction depends upon how many IOPs must be generated. This, in turn, depends upon the operation specification code of the IOT instruction. Thus, input/output transfer instructions directed to standard external I/O devices will be executed in 2.6 microseconds if the octal code for the IOT ends in 1, 2, or 4 (one operation specification bit set). The IOT instruction will require 3.6 microseconds if its octal code ends in 3, 5, or 6 (two operation specification bits set), 4.6 microseconds if the octal code ends in 7, or 1.2 microseconds if the octal code ends in 0.

#### **Programmed Data Transfer Timing Constraints**

Most I/O devices are capable of performing one 12-bit data transfer between the accumulator and the device buffer in 4.6 microseconds or less. Once this transfer is completed, however, the I/O device must dispose of the transferred data, if the operation was an output transfer, or load new data into its buffer, for the next input transfer. In the case of the Teletype terminal, for example, it is possible to read the content of the Teletype data buffer into the AC in 2.6 microseconds, which implies a maximum transfer rate of about 120,000 characters per second. The actual Teletype transfer rate of 10 characters per second reflects the fact that the mechanical mechanism within the keyboard/reader requires at least one tenth of a second to recognize which key on the console has been typed and load the device buffer with the corresponding ASCII code. If manual input is being generated at the keyboard, the maximum transfer rate may fall to 3 or 4 characters per second, corresponding to the highest typing speed of an average typist.

Similar problems arise to limit the maximum rate at which output transfers occur. The Teletype mechanism requires a minimum of about 100 milliseconds to read the device buffer, select a corresponding ASCII character, and print or punch the character. Even relatively high-speed devices such as line printers typically operate at rates of about 2000 characters per second or less, which is much slower than the maximum rate at which the processor may perform programmed data transfers.

This explains why programmed transfers often involve an exchange of status information. If the central processor is executing programmed transfers to transmit information to a device on a continual basis, it is essential that the processor check the status of the device busy/done flip-flop before executing a transfer IOT instruction, and postpone each transfer until after the device has finished all operations required to complete the previous data transfer. If data is to be read from an I/O device, the processor must still monitor the device status in this manner and postpone each transfer until the I/O device has finished assembling the data and loading this information into its device buffer. When programmed data transfer is employed, the processor uses IOT instructions that transfer device status information to monitor (or modify) the state of the device busy/done flip-flop, as well as any other status indicators which may be present in the device.

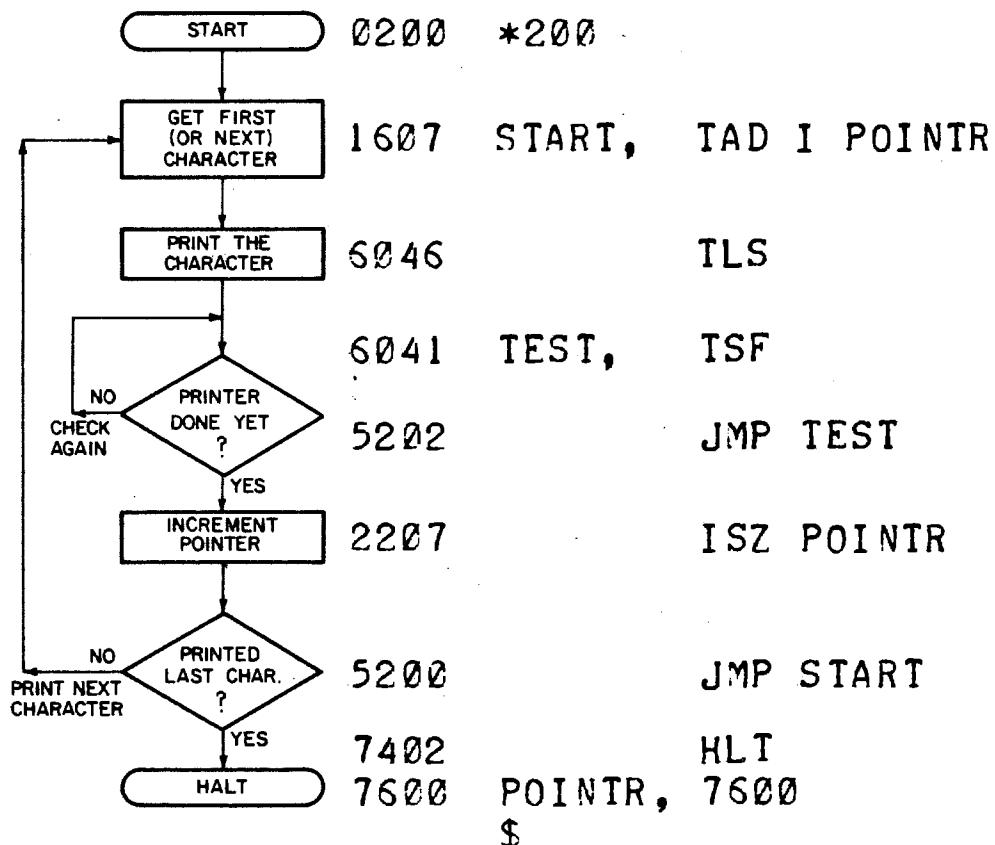


Figure 4-1 Program to Print ASCII Characters Using Programmed Data Transfer

Figure 4-1 shows a flowchart and the corresponding assembly language program which may be used to print 128 consecutive ASCII characters on the PDP-8/E console terminal. If the ASCII character codes are stored in page 37 of a 4K memory while the routine is loaded into the first 8 locations of page 2 and started at location 0200, this program will print all 128 characters in about 13 seconds. One character is transferred every 100 milliseconds, almost all of which represents time required for the Teletype terminal to complete the previous data transfer. Thus, the program executes a TLS instruction in 3.6 microseconds, then cycles through the delay loop about 250,000 times until the device status changes and it skips out of the loop to execute the next data transfer.

In summary, programmed data transfer provides an easy, convenient means of performing data I/O with a minimum of hardware and software support. The highest transfer rate that may be realized using programmed I/O is only slightly lower than the maximum rate at which the fastest available I/O device may operate. The major drawback associated with this technique is that the central processor must hang up in a waiting loop while the I/O device completes the last transfer and prepares for the next transfer. It is possible to use some of this waiting time to perform intermediate calculations, but it is rarely convenient to do so. On the other hand, programmed data transfer techniques permit easy hardware implementation and simple, economical interface design. For this reason, almost all I/O devices except bulk storage units rely heavily on programmed data transfer for routine data I/O.

### **PROGRAM INTERRUPT TRANSFERS**

The program interrupt system may be used to initiate programmed data transfers in such a way that the time spent waiting for device flags is greatly reduced or eliminated altogether. It also provides a means of performing concurrent programmed data transfers between the central processor and two or more peripheral devices. This is accomplished by isolating the I/O handling routines from the mainline program and using the interrupt system to ensure that these routines are entered only when an I/O device flag is set, indicating that the device is actually free to perform the next data transfer, or that it requires some sort of intervention from the running program.

#### **Interrupt System Operation**

All peripheral device status indicators are ORed onto a special OMNI-BUS signal line called the interrupt request line, which is asserted whenever one or more of the device flags is set. The processor interrogates the interrupt enable flip-flop and the interrupt request line during TS4 of every memory cycle in which execution of an instruction was completed. If the interrupt system is enabled and the interrupt request line is asserted, indicating that the interrupt system has been turned on and that one or more device flags were set while the current instruction was being executed, the processor executes a program interrupt.

A program interrupt is simply a conventional JMS instruction to memory location 0000 in field 0 (octal code 4000) which is built and executed by circuits in the processor. Three conditions must be satisfied before a program interrupt may occur: The interrupt system must be

enabled, the interrupt request line must be asserted by a peripheral whose device flag is set, and the processor must be in TS4 of the cycle in which execution of an instruction was completed, implying that the major state generator has just enabled the FETCH major state for the following cycle. If these conditions are met, the processor asserts an OMNIBUS signal called INT IN PROG, which forces a 4 (OP-code for a JMS instruction) into the instruction register, clears the CPMA register and enables the EXECUTE major state.

During the next machine cycle, the processor turns off the interrupt system so that no further interrupts may occur until the current interrupt has been serviced. It then executes the hardware-generated JMS to memory location 00000 which was built during TS4 of the previous cycle. Since the processor has no way of knowing that the JMS instruction is actually a program interrupt, it proceeds as though the EXECUTE cycle had been entered from FETCH and treats the hardware-generated JMS as a normal machine instruction. This causes the address of the instruction that was due to be executed when the program interrupt occurred to be stored in location 0000 of memory field 0. The PC is then loaded with 0001, so that the instruction stored in location 0001 of field 0 is the next instruction to be executed.

Figure 4-2 is a simplified flow chart that shows how the processor interrogates the interrupt system during TS4 of every memory cycle. If all conditions for a program interrupt are met, the processor concludes TS4 by initializing its control circuitry to execute a program interrupt during the following cycle. Once the interrupt has been executed, the previous content of the PC will be stored in location 0000 of field 0, the interrupt system will be turned off, and the PC will point to location 00001, which contains the next instruction to be executed.

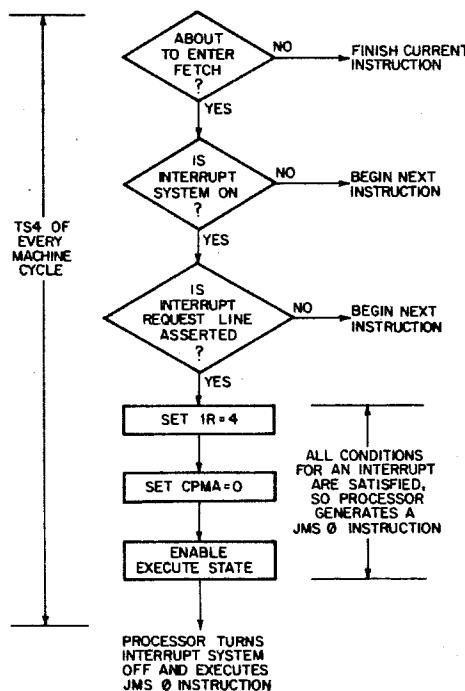


Figure 4-2 Program Interrupt Flow of Events

The interrupt system is a simple piece of hardware that has far-reaching program implications. When an interrupt occurs, mainline execution is suspended and the instruction stored in memory location 00001 is executed next. This instruction is usually a jump to the starting address of an interrupt service routine, which generally performs all of the following operations:

1. Save the content of any registers (AC, MQ, link, etc.) that will be used by the interrupt service routine.
2. Determine which peripheral device caused the program interrupt.
3. Determine why the device caused an interrupt.
4. Correct the condition that resulted in the interrupt.
5. Restore the content of registers that were used by the interrupt service routine.
6. Turn the interrupt system on.
7. Resume mainline execution, usually by executing a JMP I O instruction (octal code 5400).

A running program maintains control over the interrupt system by executing the processor IOT instructions listed in Table 4-1. Several of these interrupt IOT instructions are also used in conjunction with the KM8-E Extended Memory and Time Share option, to service program interrupts that originate in extended memory or in a timesharing environment. The user flag, save field register and interrupt inhibit flip-flop are hardware components of the extended memory and time share control, whose functions are described, briefly, in the following paragraphs. The greater than flag (GTF) is contained in the KE8-E Extended Arithmetic Element.

**Table 4-1 Interrupt IOT Instructions**

MNEMONIC	OCTAL	OPERATION
<b>SKON</b>	<b>6000</b>	<b>Skip if Interrupt System On.</b> The state of the interrupt enable flip-flop is tested. If this flip-flop is set, indicating that the interrupt system is enabled, the PC is incremented to skip the next sequential instruction and the interrupt system is turned off in the same manner as by an IOF instruction.
<b>ION</b>	<b>6001</b>	<b>Interrupt Turn On.</b> The next program instruction is executed, then the interrupt system is enabled. Delaying the interrupt enable in this manner gives the interrupt service routine time to resume background program execution (by means of a JMP I O instruction) before another program interrupt occurs.

**Table 4-1 Interrupt IOT Instructions (continued)**

MNEMONIC OCTAL	OPERATION
<b>IOF</b>	<b>6002</b> <b>Interrupt Turn Off.</b> The interrupt system is disabled during TS3, inhibiting further program interrupts, including any interrupt request that might have been flagged during execution of the IOF instruction.
<b>SRQ</b>	<b>6003</b> <b>Skip on Interrupt Request.</b> The state of the OMNI-BUS interrupt request line is tested. If this line is asserted, indicating that one or more devices are requesting a program interrupt, the PC is incremented to skip the next sequential instruction.
<b>GTF</b>	<b>6004</b> <b>Get Flags.</b> The following machine states are read into the indicated bits of the accumulator:  AC BIT LOADED WITH CONTENT OR STATE OF:  AC0 Link AC1 Greater than flag AC2 Interrupt request line AC3 Interrupt inhibit flip-flop AC4 Interrupt enable flip-flop AC5 User flag AC6-11 Save field register
<b>RTF</b>	<b>6005</b> <b>Restore Flags.</b> This instruction is the converse of GTF. AC0 is loaded into the link, AC1 is loaded into the Greater Than Flag, AC5 is loaded into the User Flag, AC6-11 are loaded into the Save Field Register, the interrupt system is enabled in the same manner as by an ION instruction, and the interrupt inhibit flip-flop is set.
<b>SGT</b>	<b>6006</b> <b>Skip if Greater Than.</b> If the Greater Than Flag is set, the PC is incremented to skip the next sequential instruction.
<b>CAF</b>	<b>6007</b> <b>Clear All Flags.</b> This instruction is logically equivalent to operating the CLEAR switch on the programmer's console. It generates an INITIALIZE pulse on the OMNIBUS and at the external I/O interface. The AC and Link are cleared. The action of INITIALIZE depends upon the design of each peripheral control, but it generally clears all I/O device flags and motion control flip-flops, and sets the interrupt enable flip-flop in each peripheral device. A CAF instruction should not be executed while a device is active. For example, a CAF instruction should not be executed within 100 milliseconds of a TLS instruction.

### **Interrupt System Hardware Components**

The only means of turning the interrupt system on is by executing an ION instruction, which unconditionally sets the interrupt enable flip-flop on the timing generator module. Once the interrupt enable flip-flop has been set, TP1 of the next FETCH cycle automatically sets the interrupt delay flip-flop, and fully enables the interrupt system. The 1-cycle delay provided by the interrupt delay flip-flop prevents a program interrupt from occurring during TS4 of the cycle in which the ION instruction was executed (or during any DMA cycles that might intervene following execution of the ION instruction). This gives the running program time to execute one additional instruction, usually a JMP I O (octal code 5400) to transfer program control to the background routine before another interrupt occurs.

The interrupt system monitors two OMNIBUS signals, F SET L, which is asserted whenever an instruction is in its concluding cycle, and MS IR DIS L, which is asserted during every DMA operation. If an interrupt request occurs during any cycle in which the interrupt delay flip-flop is set (implying that the interrupt enable flip-flop is also set), F SET L is asserted, and MS IR DIS L is not asserted, the interrupt system generates a pulse that is clocked into the interrupt sync flip-flop by OMNIBUS signal INT STROBE H. This sets the interrupt sync flip-flop, which generates signal INT IN PROG H and gates it onto the OMNIBUS.

Signal INT IN PROG H loads the IR register, zeroes the CPMA register and enables the EXECUTE major state during the following cycle. At TP1 of this cycle, the interrupt enable flip-flop is cleared to prevent further interrupts until after the current interrupt has been serviced. This flip-flop is also cleared by signal INITIALIZE H and by the IOF instruction. Clearing the interrupt enable flip-flop automatically clears the interrupt delay and interrupt sync flip-flops.

When extended memory is installed, it becomes necessary to inhibit program interrupts while the processor is loading the instruction field register and transferring control from one memory field to another. The extended memory and time share control contains an interrupt inhibit flip-flop which is set during TS3 of every instruction that modifies the content of the instruction field register (i.e. RTF, RMF and CIF). This flip-flop remains set until the next JMP or JMS instruction is executed. While the interrupt inhibit flip-flop is set it generates signal INT INHIBIT, which is displayed in bit 3 of the programmer's console STATUS indicator register (labelled NO INT), and grounds OMNIBUS signal INT IN PROG H, thereby preventing any device from entering a program interrupt request. This allows the processor ample time to complete any operations that may be required to initialize the machine registers before branching to a different memory field, and then execute the JMP or JMS into extended memory. The interrupt inhibit flip-flop is cleared during TS3 of every JMP or JMS instruction.

The extended memory and time share control also contains two interrupt buffers that preserve memory field and machine status information during an interrupt. When an interrupt occurs, the content of the 1-bit user flag and the 3-bit instruction field register are automatically loaded into bits 0 and 1-3, respectively, of interrupt buffer A, while the

content of the 3-bit data field register is loaded into interrupt buffer B. The two interrupt buffers are often considered as a single, 7-bit register called the save field register. The save field register is never cleared; information is always jam transferred in during execution of a program interrupt, and retained until the next program interrupt occurs or the next RTF instruction is executed. The content of the save field register is displayed in bits 5-11 of the programmer's console STATUS indicator register.

If it becomes necessary to service multiple or nested program interrupts, the GTF instruction may be used to read the content of the save field register into the AC so that this value may be stored in memory, along with the content of the AC, the MQ, location 00000, and other registers that might be modified by a program interrupt. It is then possible to re-enable the interrupt system while the current program interrupt is being serviced. When extended memory and timesharing status information is saved in this manner, it may be restored to the save field register by means of an RTF instruction. Further information on techniques for servicing multiple interrupts and program interrupts that originate in extended memory or a timesharing environment appears later in this handbook, as well as in Chapter 6 of *Introduction to Programming 1972*.

*0	PAL8-V7	PAGE 1
0000 *0		
00000 0201	ROTATE	/FIRST RETURN ADDRESS
00001 2022	ISZ POINTR	/INCREMENT POINTER
00002 7410	SKP	/PRINTED ALL CHARS?
00003 7402	HLT	/YES: HALT
00004 3017	DCA SAVEAC	/NO: SAVE AC
00005 7204	RAL	/GET THE LINK
00006 3020	DCA SAVEL	/AND SAVE IT
00007 1422	TAD I POINTR	/GET NEXT CHAR
00010 6046 PRINT, TLS		/PRINT IT
00011 7300	CLA CLL	/CLEAR AC AND LINK
00012 1020	TAD SAVEL	/GET STORED LINK
00013 7810	RAR	/ROTATE INTO PLACE
00014 1017	TAD SAVEAC	/GET STORED AC
00015 6201	ION	/ENABLE INTERRUPTS
00016 5400	JMP I 0	/RESUME MAINLINE
00017 0001	SAVEAC, I	
00020 0000	SAVEL, 0	
00021 2000	TIMER, 0	
00022 7600	POINIR, 7600	
0200 *200		/DUMMY MAINLINE FOLLOWS
02020 5010	START, JMP PRINT	/PRINT 1ST CHAR
02021 7604	ROTATE, RAL	/ROTATE AC BIT LEFT
02022 2021	DELAY, ISZ TIMER	/DELAY LOOP:
02023 5202	JMP DELAY	/STALLS ABOUT 20 MSEC
02024 5201	JMP ROTATE	
\$		

Figure 4-3 Simplified Interrupt Service Routine

The program example of Figure 4-3 illustrates a very simple interrupt service routine. This example performs the same operations as the example of Figure 4-1; however, the interrupt driven program executes a dummy background routine that rotates one bit endlessly through the accumulator while the stored characters are being printed on the Teletype. A delay loop is included in the background routine, so that the rotating display moves slowly enough to be visible at the programmer's console AC register. This example requires nearly the same amount of time as the previous example to print all 128 stored characters, even though it is executing another data processing operation concurrently with the data transfer.

Highly developed interrupt service routines written for the PDP-8/E permit the processor to exercise simultaneous control over many peripheral devices while executing a background program that may be completely unaware of the detailed operation of the I/O process. Devices can be serviced on a first come, first served or a round robin basis. Additional techniques permit interrupts from a high priority device to supersede low priority interrupts, so that the high priority device is always serviced in the shortest possible time. At large installations, a software priority interrupt system may be designed to service many different I/O devices on a priority basis. Whichever system is employed, use of the program interrupt system affords a significant increase in I/O processing capability by eliminating the processor waiting time that is often associated with programmed I/O.

### **DATA BREAK TRANSFERS**

Data break, sometimes called direct memory accessor DMA, is the preferred form of data transfer for use with high-speed storage devices such as magnetic disk or DECTape units. Direct memory access is indicated whenever it becomes necessary to transfer data contained in a block of consecutive memory locations out to a high-speed peripheral device, or to read sequential words of data from a high-speed device into a specified memory buffer area.

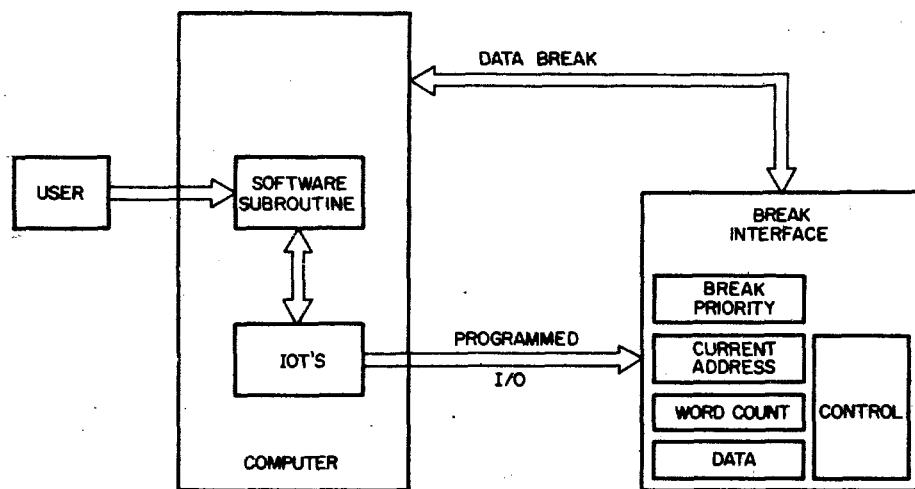
Data break peripherals are supplied with software subroutines featuring convenient, standardized calling sequences that perform all normal functions of data I/O. Thus, for most applications, users need not be concerned with the detailed operating characteristics of a particular DMA device. The remainder of this chapter describes the general operation of the data break system in terms that apply to all standard DMA devices.

#### **Current Address (CA) Register**

A 12-bit current address register is associated with every data break device. At the beginning of a data break transfer, the CA register contains the 12-bit address of the memory location in which the last data break transfer was performed. The content of the CA register is incremented by 1 during a data break transfer, and the incremented value is used as the address of the memory location with which the current transfer will be performed. In this manner, a single I/O operation may transfer up to 4096 words of data between a peripheral device and a series of sequential memory locations.

#### **Word Count (WC) Register**

A 12-bit word count register is also associated with every data break device. At the beginning of a data break transfer, the WC register contains the negative (two's complement) of the number of 12-bit words that remain to be transferred. The content of the WC register is incremented by 1 during every data break transfer. If this value becomes zero, word count overflow has occurred indicating that the word currently being transferred is the last word in the data block. Word count overflow generates a control signal which clears the I/O device enabling circuits and inhibits further data transfers.



General block diagram of basic data break interface

#### Data Break Priority

Up to 12 data break peripherals may be interfaced with a PDP-8/E. One of the 12 DATA lines on the OMNIBUS is assigned to each data break device for the purpose of determining break priority. The highest priority line, DATA0 is assigned to the fastest data break device, while the lowest priority line, DATA11, would be assigned to the slowest device if a full 12 data break devices are installed. When two or more devices request a data break simultaneously, the higher priority device makes the first DMA transfer. (A thirteenth device may be added without an assigned DATA bus priority bit, for most applications. This device will automatically assume lowest priority.)

#### Data Break Transfers

When an internal data break peripheral is ready to perform a data transfer, it gates its priority bit onto the DATA bus and determines whether a higher priority device is also requesting a data break. If there is no higher priority request by TS4 of the current machine cycle, the device asserts two signals which disable the processor major state generator, instruction register and CPMA register. At the same time, the DATA bus is gated to the processor adder inputs to provide a bidirectional data path between the peripheral and the memory system. At this point, the processor enters a DMA cycle and the peripheral assumes control over the processor logic circuits by asserting various signal lines on the OMNIBUS.

During a data break, the peripheral device may generate signals which gate the content of a specified memory location onto the DATA BUS and out to the I/O device register, or it may gate the content of the device data register to a specified memory location, overwriting the previous content of that location. The device may also gate the content of its data register and the content of a memory location to the processor adder inputs, add these values, and restore the two's complement sum to the designated memory location. This process, called Add Data to Memory or ADM, is commonly used to increment the content of a memory location.

Unlike program interrupts, which are recognized only during the cycle in which execution of an instruction was completed, a data break always occurs at the conclusion of the machine cycle in which it was requested. If two devices request a data break simultaneously, the higher priority device begins its data break during the next cycle and the low priority request is honored as soon as the high priority device relinquishes control of the processor. Once all break requests have been honored, instruction execution resumes at the point where it was discontinued. The processor major registers are never modified during a data break transfer, so that no restoration of machine status is necessary.

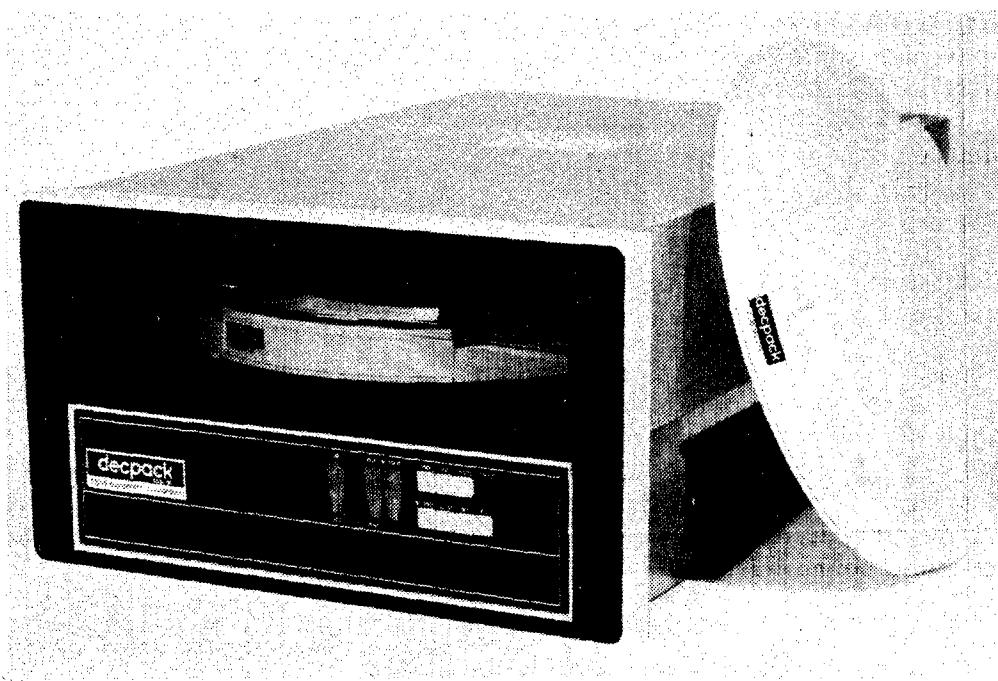
Standard PDP-8/E data break peripherals employ one of two types of data break. High-speed devices, or devices which retain information in their data registers for a relatively short period of time, usually operate on a single-cycle data break. Single-cycle data break is the fastest I/O transfer method, but it also requires more complex control circuitry. Low priority data break devices usually employ a three-cycle data break, which is somewhat slower but easier to implement.

Single-cycle data break devices have their own self-contained word count and current address registers. Once a single-cycle device has been initialized to transfer a block of data, it forces the processor into a DMA state for one machine cycle whenever it is ready to transfer a 12-bit data word. The WC and CA registers are incremented during this cycle, while the data transfer is being performed. As long as WC overflow does not occur, the device circuits remain enabled so that the device relinquishes control of the processor and begins to assemble the next word of data that will be transferred.

Three-cycle data break devices have the addresses of two field 0 memory locations hard-wired into their control modules. The device uses one of these memory locations as its WC register and the other as its CA register. When a three-cycle device is ready to transfer a word of data via data break, it forces the processor to enter the DMA state for three consecutive cycles. During the first cycle, the device uses the ADM feature to increment the memory location designated as its WC register and test for WC overflow. The CA register is incremented in the same manner during the second cycle, and the actual data transfer occurs during the third cycle.

Every standard three-cycle data break peripheral has an assigned WC register address and an assigned CA register address, both of which are unique to that peripheral. Device WC registers usually have even memory addresses; that is, addresses for which bit 11 is a 0. The memory address of the CA register for a given device is usually one more than the WC register address, so that the same circuitry may generate the eleven high-order bits of both register addresses.

A three-cycle data break may be interrupted by a data break request from a higher priority device. Three-cycle devices test the priority bits of the DATA bus at the end of their word count and current address cycles. If there is a higher priority request on the bus, the device relinquishes control of the processor until its priority is again highest, and then resumes the data transfer at the point where it was discontinued.



The RK05 DECPACK Drive, with a data transfer rate of nearly 1.5 million bits per second, is a typical single-cycle data break device.

#### **Single-cycle Data Break**

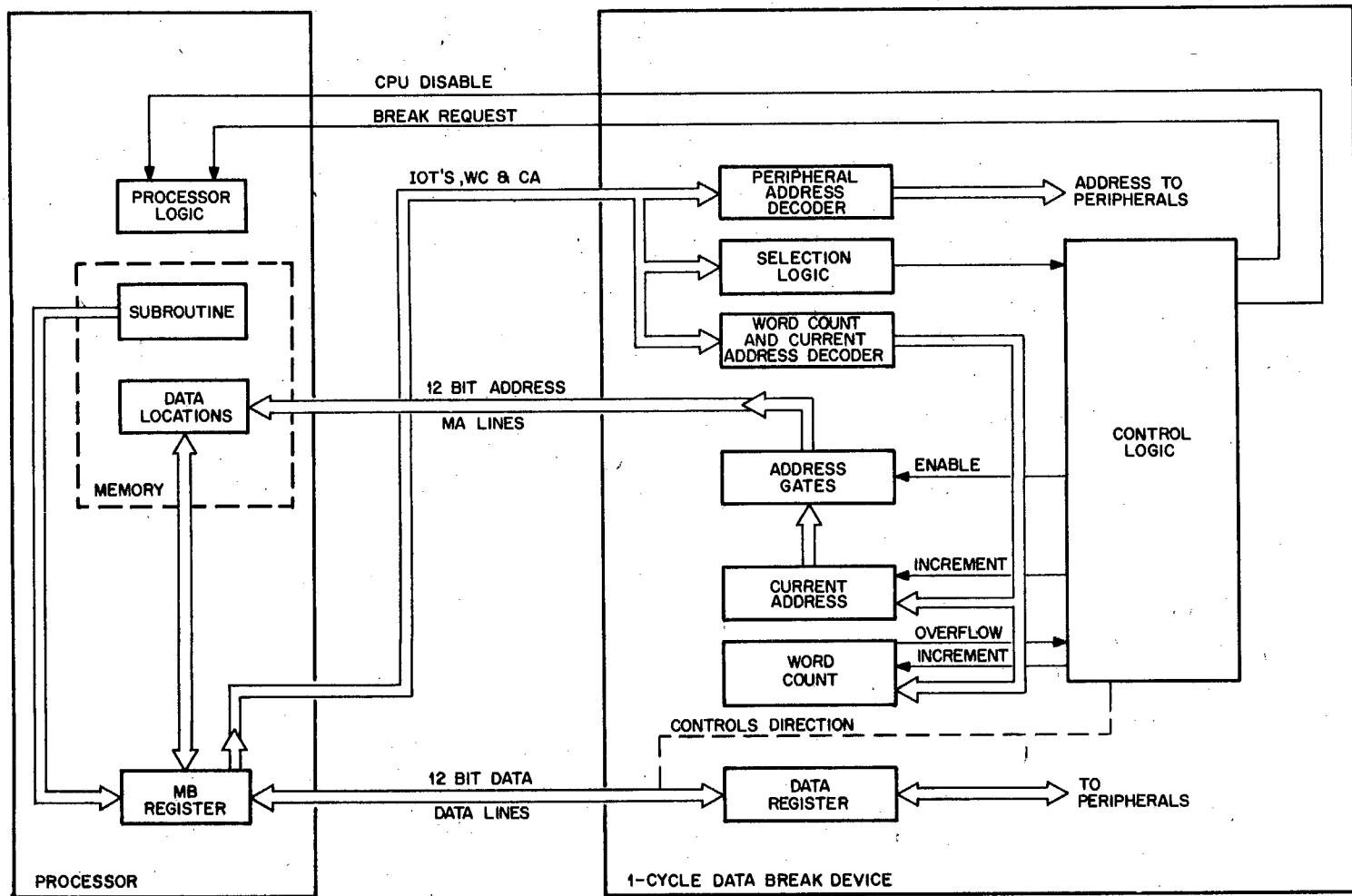
Figure 4-4 is a simplified diagram that illustrates the interaction between the processor and a single-cycle data break device. Because the device is a single-cycle device, the WC and CA registers are shown in the peripheral. The flow chart of Figure 4-5 shows the sequence of operations required to effect a block transfer via single-cycle data break. This flow of events is divided into three phases, designed as initial set-up, data transfer, and exit.

During initial set-up, the running program executes IOT instructions which load the device WC register with the two's complement of the number of words in the block to be transferred. The CA register is then loaded with one less than the 12-bit address of the first memory location with which a transfer will be performed. Additional IOT instructions may specify the direction of the transfer, the off-line location with which the first transfer will occur and the means of accessing this location, the memory field of the processor data buffer, or similar information, depending upon the precise nature of the peripheral device. The initial set-up routine usually concludes by executing an IOT instruction that enables the device control circuitry. The actual block data transfer begins at this point and continues without program control until WC overflow occurs.

The peripheral begins each block data transfer by accessing the off-line location with which the first word of the data block will be transferred. Even relatively high-speed devices such as magnetic disks may require many machine cycles to perform this operation; however, the processor may use the access time to execute program instructions as

4-15

Figure 4-4 Single-Cycle Data Break Simplified Block Diagram



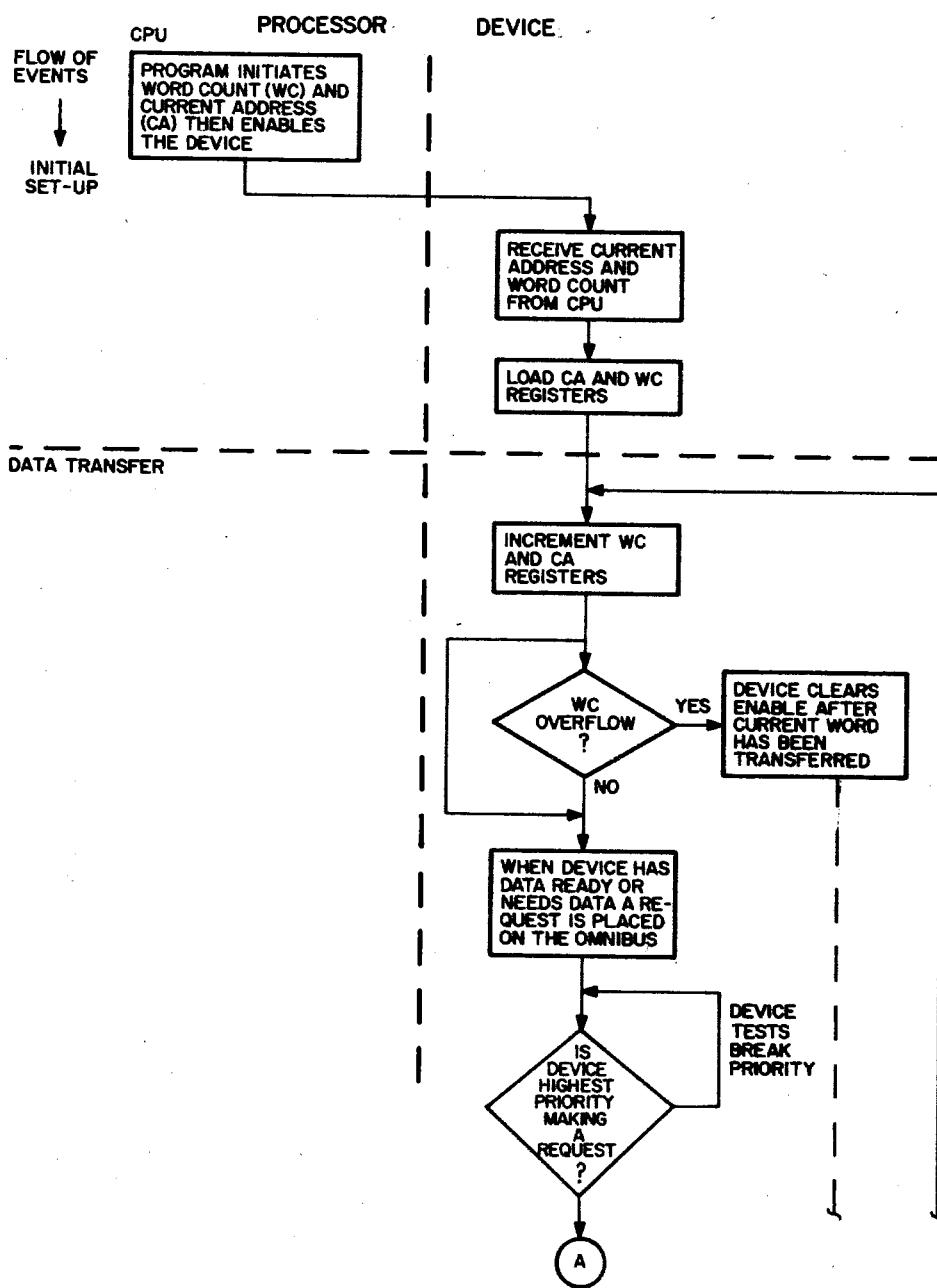
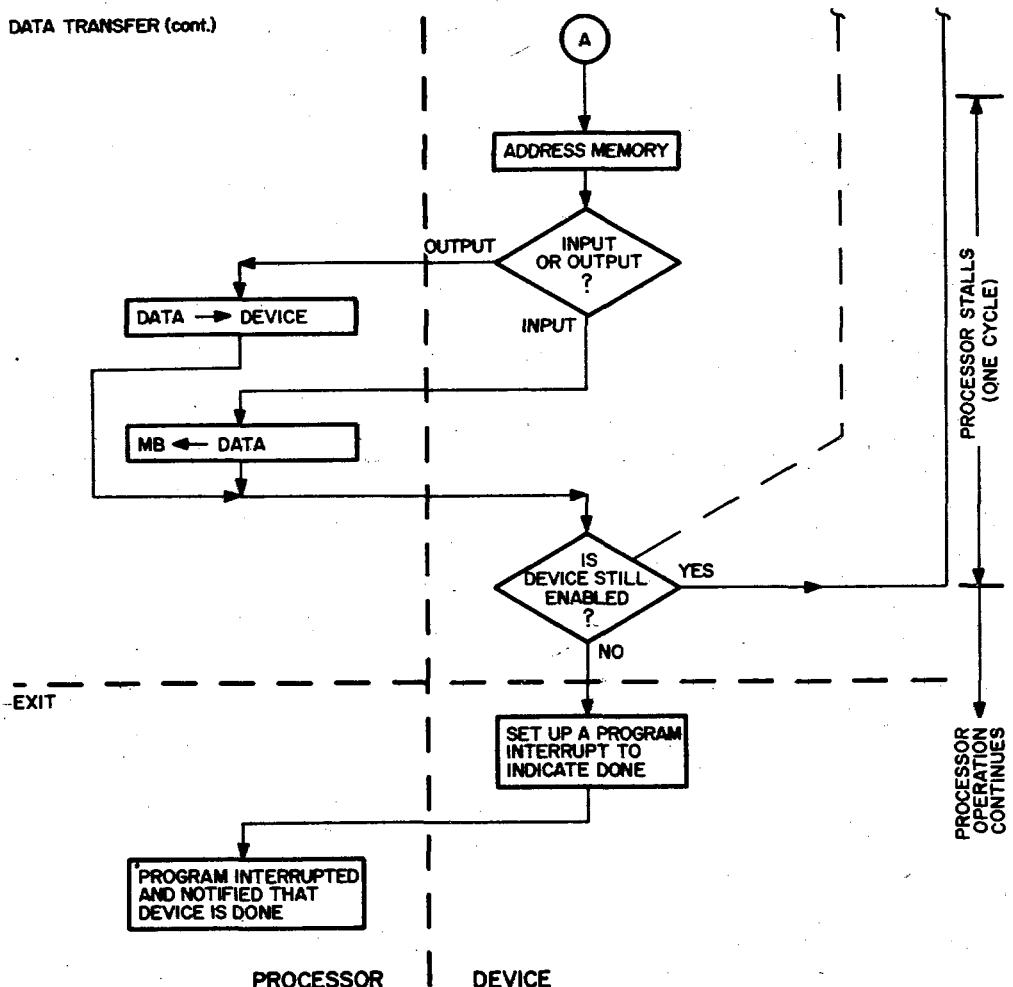


Figure 4-5 Single-Cycle Data Break Flow Chart



**Figure 4-5 Single-Cycle Data Break Flow Chart (continued)**

long as it does not attempt to operate on data which has not been transferred yet. As soon as the peripheral assembles a word of data in its data register, if the operation is an input transfer, or accesses the specified off-line storage location, in the case of an output transfer, it places a data break request on the OMNIBUS. The processor enters the DMA state during the following cycle, and one word of data is transferred between the peripheral and memory. The processor then resumes program execution until the device is ready to transfer another word of data.

A single-cycle device increments its WC and CA registers while each word of data is being transferred. If WC overflow occurs, the device completes the current transfer and begins the exit phase of the data break operation. In most cases, this simply involves clearing the enable circuitry and setting the device flag, which will cause a program interrupt provided that the interrupt system is enabled. The processor normally responds by executing IOT instructions to determine whether error conditions were flagged during the transfer, for example, or to turn off any device components that were not disabled automatically by WC overflow.

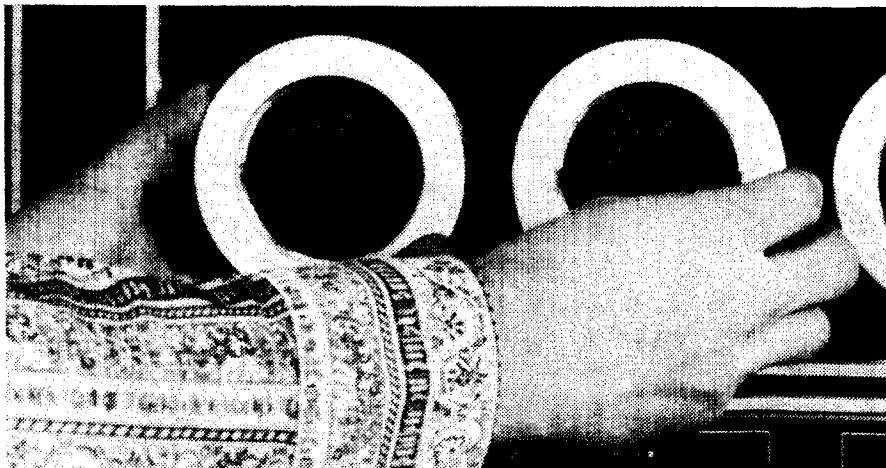
### **Three-cycle Data Break**

Figure 4-6 is a simplified diagram that illustrates the interaction between the processor and a three-cycle data break device. Because the device is a three-cycle device, the WC and CA registers are shown in computer memory. The flow chart of Figure 4-7 shows the sequence of operations required to effect a block data transfer via three-cycle data break. The initial set-up and exit phases include operations similar to those required for a single-cycle data break; however, the data transfer phase of a three-cycle break is divided into a WC cycle, a CA cycle and a data transfer cycle.

During initial set-up, the running program uses MRIs to load the memory locations designated as the device WC and CA registers, then execute IOT instructions that initialize the device and specify any necessary transfer parameters. Once the device control circuitry has been enabled, the processor is free to perform other tasks while the peripheral accesses storage locations and executes data break transfers.

Every three-cycle data break begins with a WC cycle during which the device gates the address of its WC register onto the OMNIBUS, fetches the register content into the MB, and gates a 1 to the processor adder inputs via the DATA bus, thereby incrementing the word count. The resulting addition is tested for overflow while the incremented word count is restored to memory. If overflow occurs, the peripheral clears its enabling circuits and sets its device flag as soon as the current transfer has been completed. In any event, the device concludes each WC cycle by testing the DATA bus to determine whether any higher priority device has entered a break request.

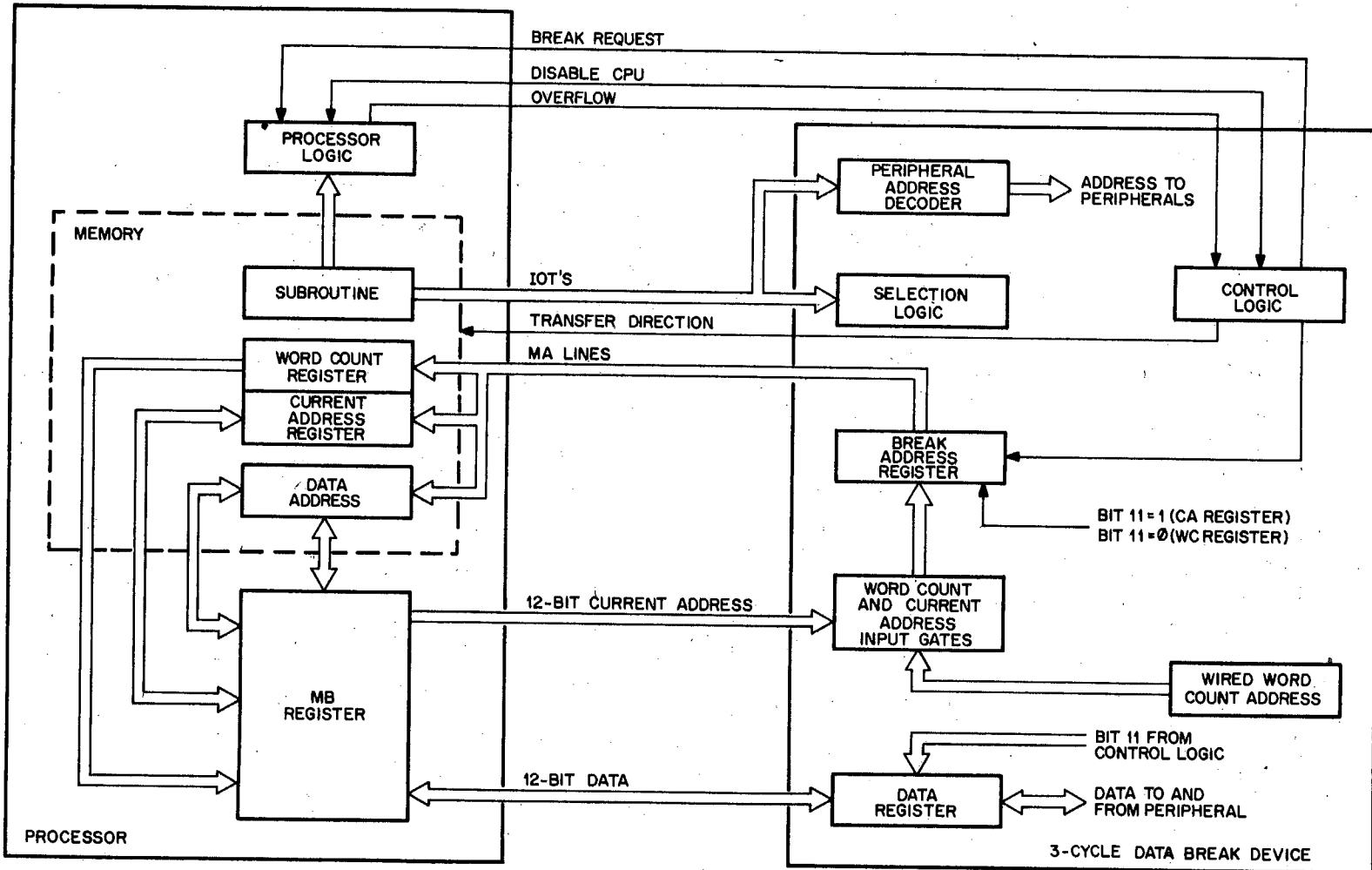
If there is no higher priority request on the DATA bus, the device concludes its WC cycle and immediately begins its CA cycle. The CA register is incremented in the manner just described, and the incremented value is restored to memory and also transferred to the device break address register. Break priority is tested again at the end of the CA cycle.



DECtape is a unique, virtually indestructible, mylar magnetic tape permitting random access in either direction using three-cycle data break or programmed data transfer.

4-19

Figure 4-6 Three-Cycle Data Break Simplified Block Diagram



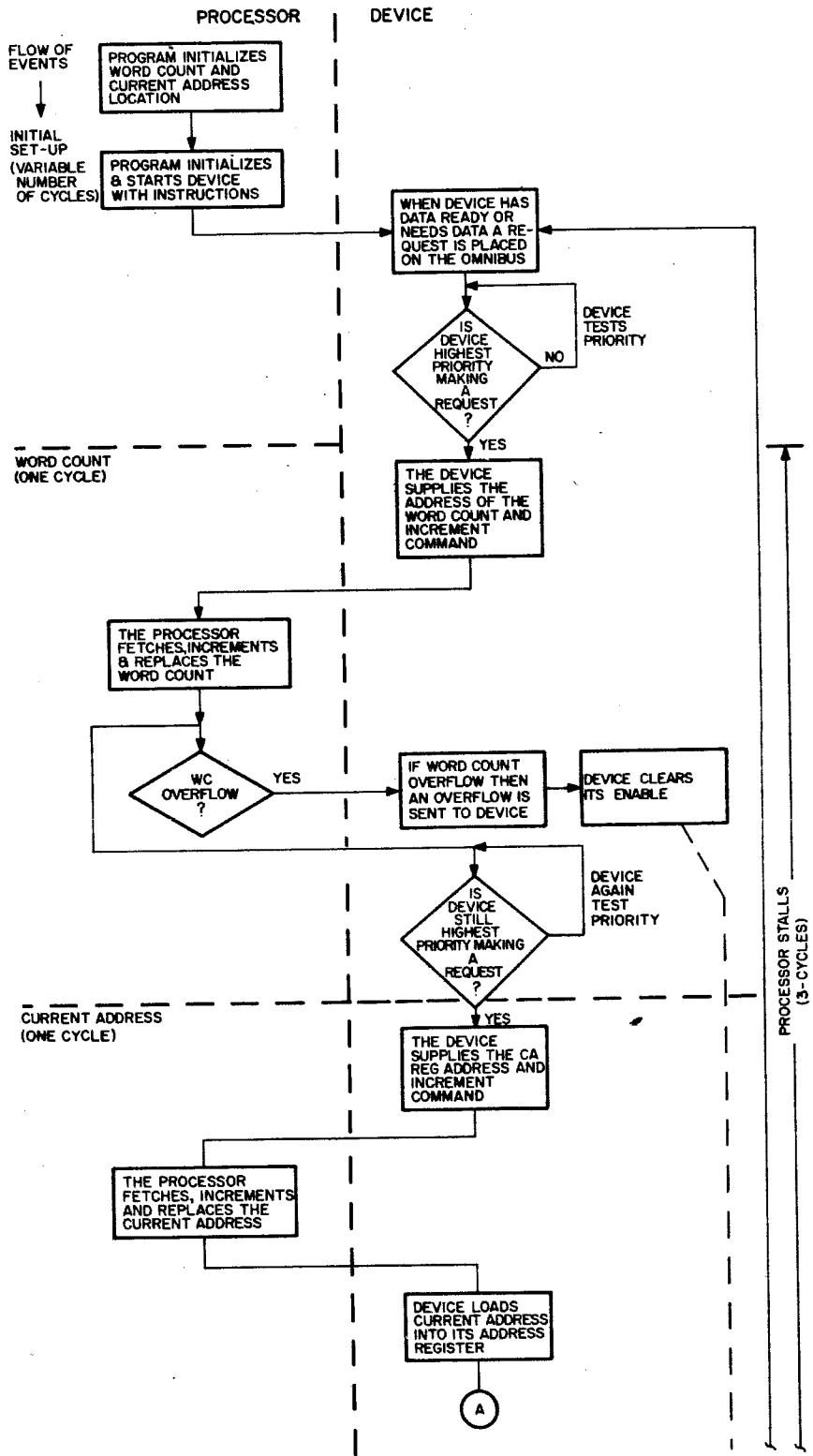


Figure 4-7 Three-Cycle Data Break Flow Chart

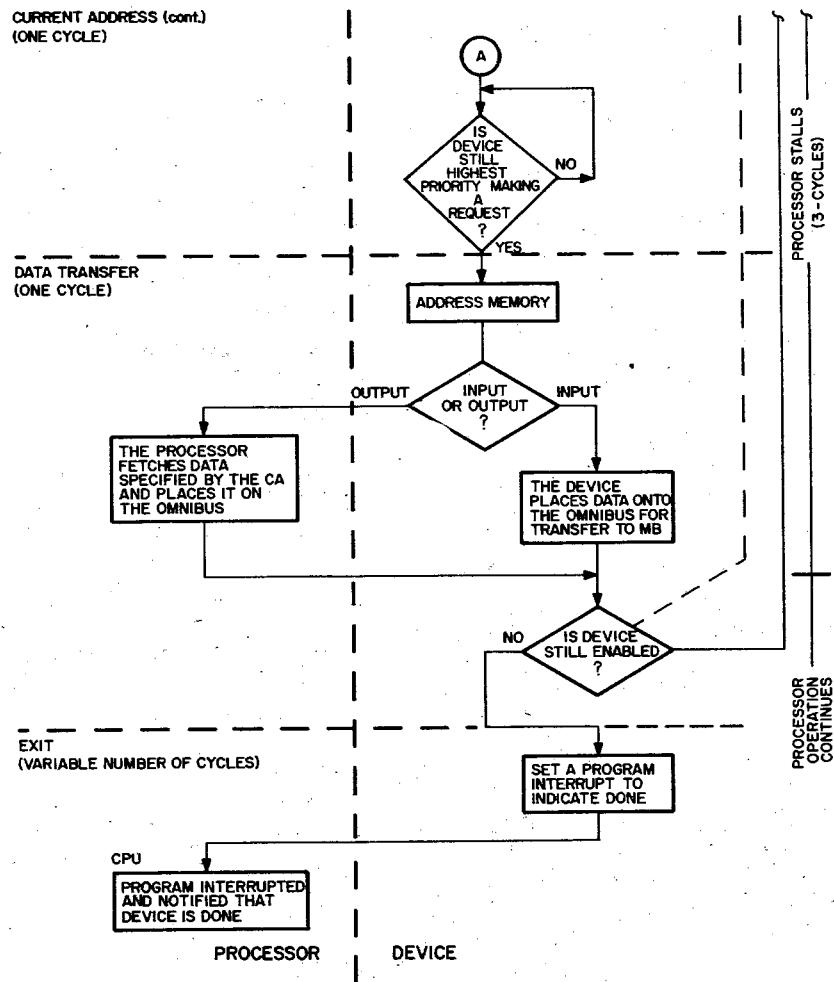


Figure 4-7 Three-Cycle Data Break Flow Chart (continued)

During the data transfer cycle, the peripheral generates a signal to specify the direction of the transfer, gates the content of its break address register onto the OMNIBUS, and either accepts or transmits one word of data. If WC overflow did not occur during the WC cycle, the device relinquishes control of the processor and begins to prepare for the next data transfer. When WC overflow does occur, the device flag is set at the end of the next data transfer cycle, and the running program executes IOT instructions to perform any operations that may be required to terminate the block I/O process.

Both types of data break transfer offer data transfer rates far higher than the fastest speed at which transfers may be executed using programmed I/O or the program interrupt system. Although it is possible to transfer a single word of data via data break, a DMA device may be initialized to transfer up to 4096 consecutive words of data just as easily, using the same general sequence of instructions. Once a device has been initialized, data break transfers are executed automatically, without program control, until the entire block of data has been transferred.



# 5

## processor options

### **MECHANICAL EXPANSION OPTIONS**

Mechanical expansion options include cabinets, expansion boxes and front panels that affect the external physical properties of the PDP-8/E computer. Further information regarding the selection and installation of mechanical expansion options appears in Chapter 8 of this handbook.

#### **System Expander Boxes**

The BA8-AA System Expander Box extends the system expansion capability of a rack mounted PDP-8/E computer. Each BA8-AA includes a power chassis assembly and an OMNIBUS assembly capable of accommodating up to 20 PDP-8/E modules. Rack-mountable chassis slides are also included, along with a BCO8H-3F Cable Set and a KC8-EB blank front panel. The BCO8H-3F Cable Set is three and one half feet in length.

The BA8-AB System Expander Box includes a power chassis assembly and OMNIBUS assembly capable of accommodating up to 20 PDP-8/E modules, along with a BCO8H-3F Cable Set, a KC8-EB blank front panel, and a table-top cover. It provides the same system expansion capability for the table-top PDP-8/E as the BA8-AA provides for the rack-mounted system.

The BE8-A OMNIBUS Expander consists of an additional OMNIBUS assembly, capable of accommodating up to 20 PDP-8/E modules, together with an M935 Bus Connector. Addition of a BE8-A OMNIBUS Expander permits either the BA8-AA System Expander Box or the BA8-AB System Expander Box to accommodate up to 38 PDP-8/E modules.

### **FRONT PANEL OPTIONS**

The KC8-EC Turnkey Front Panel contains a key-operated ON/OFF switch that controls application of primary power to the PDP-8/E system. This front panel is used as an alternate panel for the PDP-8/E in systems that contain a KP8-E Power Fail and Auto Restart Option.

The KC8-EB Blank Front Panel, which is supplied with both of the system expander boxes described above, may also be used in place of the KC8-EA panel or the KC8-EC panel on the PDP-8/E. The KC8-EB front panel is suitable for PDP-8/E systems that include a KP8-E Power Fail and Auto Restart option as well as some external means of switching primary power to the system.

## **KE8-E EXTENDED ARITHMETIC ELEMENT**

The KE8-E Extended Arithmetic Element (EAE) for the PDP-8/E enables the central processor to perform arithmetic operations at high speeds by incorporating the EAE components with the existing central processor logic circuitry so the two systems operate asynchronously. Most users employ the EAE in conjunction with the 23-bit Floating-Point Package, described in Chapter 2, to provide fast, convenient floating-point arithmetic, mathematical and trigonometric function evaluation, and formatted floating-point I/O. The option consists of two QUAD modules containing circuits that perform parallel arithmetic operations on positive binary numbers. It includes the registers and control logic circuits described in the following paragraphs.

**Step Counter.** The 5-bit step counter register is used to record the number of shifts performed during a logical or arithmetic shift operation and to stop the operation once the correct number of shifts has been executed. When an ASR, LSR, SCL, or SHL instruction is executed, the step counter is loaded with the complement of the step count contained in bits 7-11 of the memory location following the instruction. Bits 7-11 of the AC are loaded directly into the step counter during execution of an ACS instruction. The step counter is cleared for MUY, DVI and NMI instructions. The step counter is incremented as each shift is performed, and step counter overflow terminates the shift operation.

**EAE Instruction Register.** The EAE IR is a 12-bit register that is loaded during the FETCH cycle of EAE instruction execution. Bits 6 and 8-10 of the EAE IR are of particular interest, since these bits identify the particular EAE operation to be executed. (Bits 4, 5 and 7 are used by the group 3 operate instructions described in Chapter 3.)

**EAE Timing and Control Logic.** The EAE control logic is contained on modules which plug into the PDP-8/E OMNIBUS. These circuits are used in conjunction with the accumulator, link, multiplier quotient and memory buffer registers of the basic PDP-8/E to perform asynchronous arithmetic operations. The EAE control logic adds a larger class of arithmetic instructions to the group 3 operate instruction list.

**EAE Mode Flip-Flop.** The state of the EAE mode flip-flop determines which of two subsets of EAE instructions is currently implemented. The mode flip-flop is set to mode A when power is applied to the machine, when the CLEAR key on the programmer's console is operated, and when a CAF instruction is executed. It may be set to mode B or reset to mode A by certain EAE instructions.

**Greater Than Flag.** The greater than flag (GTF, not to be confused with GTF instruction) is a 1-bit register that is activated during execution of mode B EAE instructions. The GTF remains cleared during execution of all mode A instructions. When the GTF is activated, it receives the content of MQ11 during right shift operations. This facilitates subsequent round-off by indicating whether the content of the MQ should be rounded up (GTF set) or left alone (GTF cleared). The GTF is also set during execution of an SAM instruction, whenever the signed number in the MQ at the end of the operation is greater than or equal to the signed number that was in the AC at the beginning of the operation.

### **Programming the Extended Arithmetic Element**

Extended Arithmetic Element instructions are an extension of the group 3 microinstructions. Like the other group 3 microinstructions introduced in Chapter 3, they have an OP-code of 7, while bits 3 and 11 are both set to contain binary ones. Mode A instructions are wholly compatible with PDP-8/I extended arithmetic element instructions, so that programs written for the PDP-8/I extended arithmetic element may run on the KE8-E Extended Arithmetic Element without modification. Mode B provides a greatly expanded set of instructions that is available for new programming on the PDP-8/E. Several EAE operations may be executed in either mode. The common features of these operations are described below.

#### **Multiplication**

During a multiplication operation, the content of the 12-bit MQ register is multiplied by a 12-bit multiplier (whose location depends upon the instruction mode). At the conclusion of the multiplication, the 12 most significant bits of the product are in the accumulator while the 12 least significant bits are in the MQ register. The multiplication is an unsigned integer multiply. That is, multiplier and multiplicand are treated as 12-bit, positive binary numbers with the binary point positioned after the least significant bit of each. The binary point of the product is positioned after the least significant bit of the MQ register. If the accumulator is non-zero at the start of the multiplication, its content is added to the low-order half of the product (contained in the MQ register). The link is always cleared.

#### **Division**

During a division operation, the content of the AC and MQ registers is treated as a 24-bit dividend with the 12 high-order bits in the AC. This number is divided by a 12-bit divisor (whose location depends upon the instruction mode) and the quotient and remainder are left in the MQ and AC registers, respectively. The division is an unsigned integer divide. The link is cleared if the first subtraction produces a negative result, indicating that divide overflow has not taken place. If the first subtraction produces a positive result, the link is set to indicate that divided overflow has occurred, and the division operation is terminated immediately. The content of the AC and MQ registers is modified by divide overflow, even though the operation is terminated prematurely. Thus, the divide instruction is ordinarily followed by a test of the link to check for overflow before further computation occurs.

#### **Left Shift**

During a left shift operation the link, AC and MQ are treated as one long register, with a high-order bit in the link and low-order bits in the MQ. The previous content of the link is lost during each shift, while AC0 is shifted into the link, MQ0 is shifted into AC11, and a zero is shifted into MQ11. The number of shifts to be executed is determined by a shift count contained in bits 7-11 of the location following the left shift instruction. Program execution resumes at the location following the shift count.

### **Logical Right Shift**

During the logical right shift operation the link, AC and MQ are treated as one long register. MQ11 is either lost or shifted into the GTF, depending upon the mode of the instruction. AC11 is shifted into MQ0, while a zero is shifted into the link and into AC0. As in a left shift, the number of shifts to be executed is determined by a shift count contained in bits 7-11 of the location following the logical right shift instruction. Program execution resumes at the location following the shift count.

### **Arithmetic Right Shift**

The arithmetic right shift operation is identical to the logical right shift except that AC0 is shifted into itself and into the link.

### **Normalization**

Normalization is the process of converting a number with a known binary point into a fraction and an exponent. The step counter is initially cleared. The content of the link, AC and MQ are then shifted left, as described above, until AC0 and AC1 are different. The step counter is incremented once for each shift. (If AC2 through MQ11 are all zero, the number is already normalized and no shift occurs.) At the conclusion of a normalize operation, the step counter contains the binary number by which the AC and MQ were multiplied to accomplish normalization. Normalize instructions must not be microprogrammed with other instructions because the resulting octal codes are reserved to switch instruction modes.

The standard group 3 operate instructions introduced in Chapter 3 are implemented by logic circuitry in the PDP-8/E central processor. Table 3-8 lists these instructions, all of which are available even if an EAE is not installed. The additional EAE instructions described below may be considered as an extension of the group 3 operate instruction set. The extended instructions are implemented by circuitry contained in the KE8-E Extended Arithmetic Element.

KE8-E mode changing instructions are available in either mode of operation. Table 5-1 lists the KE8-E mode changing instructions, their mnemonics, and the operations they perform.

**Table 5-1 EAE Mode Changing Instructions**

MNEMONIC	OCTAL	OPERATION
<b>SWAB</b>	<b>7431</b>	<b>Switch from A to B.</b> If the mode flip-flop was set to A, it is set to B. If the mode flip-flop was already set to B, it remains in mode B. In either case, an MQL instruction is also executed.
<b>SWBA</b>	<b>7447</b>	<b>Switch from B to A.</b> If the mode flip-flop was set to B, it is set to A. If the mode flip-flop was already set to A, no operation occurs.

The following instruction sequence is used to test the EAE mode flip-flop and determine which mode is currently implemented:

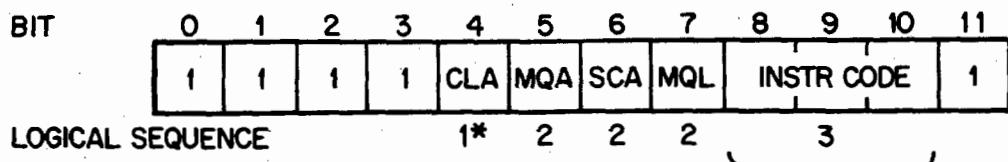
MNEMONIC	OCTAL
CAM	7621
DPSZ	7451

A skip will occur if the EAE is in mode B. If the EAE is in mode A, the skip will not occur and the SC will be loaded into the AC and normalized (a meaningless operation that modifies the content of the AC).

#### Mode A Instructions

Figure 5-1 shows the format of a KE8-E mode A instruction. The OP-code must be 7, while bits 3 and 11 are both 1. Bits 4, 5 and 7 are used by the group 3 operate microinstructions introduced in Chapter 3. Bit 6 is set to indicate an SCA instruction. Bits 8-10 are set to indicate one of the mode A instructions listed in Figure 5-1. These instructions may be microprogrammed with SCA and the group 3 microinstructions to form non-conflicting combined operations, except where indicated in Figure 5-1. The microprogrammed combination of two (or more) extended arithmetic element instructions is the bitwise logical OR or the octal codes for the individual instructions.

Most of the mode A EAE instructions require an operand, which is assumed to occupy the next word in memory, following the instruction. After execution of an EAE instruction that requires an operand, program execution resumes at the memory location following the operand. The greater than flag (GTF), explained in more detail in the next section, is always zero for mode A instructions. Table 5-2 lists the mode A instructions, their mnemonics and the operations they perform.



\* = EXCEPT FOR MQL

\*\* MUST NOT BE COMBINED  
WITH OTHER EAE  
OPERATIONS

- 0 = OPERATION
- 1 = SCL
- 2 = MUY
- 3 = DVI
- 4 = NMI \*\*
- 5 = SHL
- 6 = ASR
- 7 = LSR

Figure 5-1 EAE Mode "A" Bit Assignments

**Table 5-2 KE8-E Mode A Instructions**

MNEMONIC	OCTAL	OPERATION
SCA	7441	<b>Step Counter OR with AC.</b> The content of the step counter is combined with the content of the low-order 5 bits of the AC (AC7-11) by a bitwise logical OR operation, and the result is loaded into AC7-11. AC0-6 remain unchanged.
SCA CLA	7641	<b>Step Counter to AC.</b> The content of the step counter is loaded directly into AC7-11. AC0-6 are cleared. This instruction is a microprogrammed combination of SCA and CLA.
SCL	7403	<b>Step Counter Load from Memory.</b> The next word in memory is treated as an operand. The one's complement of the low-order 5 bits of this operand (bits 7-11) is loaded into the step counter, and program execution resumes at the location following the operand. The SCL instruction is most commonly used during interrupt servicing, to restore the content of the step counter.
MUY	7405	<b>Multiply.</b> The next word in memory is taken as a multiplier. Multiplication occurs as described above, and program execution resumes at the location following the multiplier.
DVI	7407	<b>Divide.</b> The next word in memory is taken as a divisor. Division occurs as described above, and program execution resumes at the location following the divisor. If divide overflow occurs, the link is set. If the division was legal, the link is cleared.
NMI	7411	<b>Normalize.</b> The content of the AC and MQ are normalized as described above. This instruction must not be microprogrammed with any other instruction.
SHL	7413	<b>Shift left.</b> The content of the AC and MQ is shifted left as described above. The number of shifts performed is equal to one more than the content of the 5 low-order bits (bits 7-11) of the next location in memory. Program execution resumes at the location following the shift count.
ASR	7415	<b>Arithmetic Shift Right.</b> The content of the link, AC and MQ are shifted right as described above. The number of shifts performed is equal to 1 more than the content of the 5 low-order

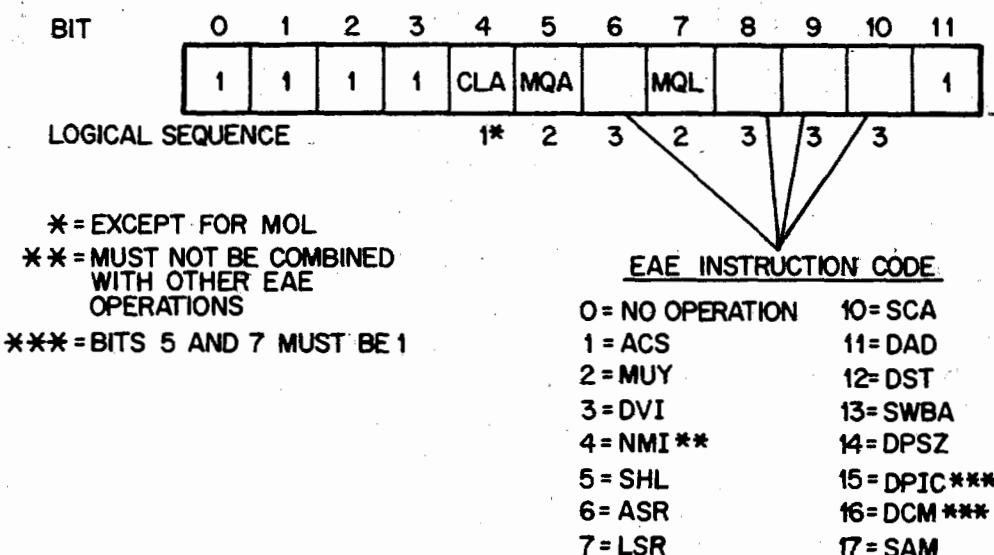
**Table 5-2 KE8-E Mode A Instruction (Cont.)**

MNEMONIC	OCTAL	OPERATION
		bits (bits 7-11) of the next location in memory. The previous content of MQ11 is lost as each shift is executed. Program execution resumes at the location following the shift count.
LSR	7417	<b>Logical Shift Right.</b> The content of the link, AC and MQ are shifted right as described above. The number of shifts performed is equal to 1 more than the content of the 5 low-order bits (bits 7-11) of the next location in memory. The previous content of MQ11 is lost as each shift is executed. Program execution resumes at the location following the shift count.

#### Mode B Instructions

Mode B instructions differ from mode A instructions in the use of bit 6 of the instruction word, the location of operands, and in greatly increased double-precision arithmetic capability. Figure 5-2 shows the format of a mode B instruction. As with mode A instructions, mode B instructions may be microprogrammed to combine non-conflicting logical operations.

Some mode B instructions require a double precision operand, which is simply two consecutive memory locations that are assumed to contain a 24-bit number with the 12 most significant bits in the location having the lower memory address. A double precision operand is addressed by specifying the 12-bit address of the high-order half of the operand.



**Figure 5-2 EAE Mode "B" Bit Assignments**

The Greater Than Flag (GTF) is activated during execution of mode B instructions. The GTF may be manipulated by means of processor IOT instructions described in Chapter 4. It is conditionally loaded by the SAM instruction, and it receives the content of MQ11 during right shift operations. Table 5-3 lists the mode B instructions, their mnemonics, and the operations they perform.

**Table 5-3 KE8-E Mode B Instructions**

MNEMONIC	OCTAL	OPERATION
ASC	7403	<b>Accumulator to Step Count.</b> The low-order 5 bits of the AC (AC7-11) are loaded into the step counter, and the AC is then cleared.
MUY	7405	<b>Multiply.</b> The next word in memory is taken as the address of a multiplier. If extended memory is installed, the multiplier is obtained from the current data field. Multiplication occurs as described above, and program execution resumes at the location following the address of the multiplier.
DVI	7407	<b>Divide.</b> The next word in memory is taken as the address of a divisor. If extended memory is installed, the divisor is obtained from the current data field. Division occurs as described above, and program execution resumes at the location following the address of the divisor. If divide overflow occurs, the link is set. If divide overflow does not occur, the link is cleared.
NMI	7411	<b>Normalize.</b> The content of the AC and MQ is normalized as described above. This command must not be microprogrammed with any other instruction.
SHL	7413	<b>Shift Left.</b> The content of the AC and MQ is shifted left as described above. The number of shifts performed is equal to the content of the 5 low-order bits (bits 7-11) of the next location in memory. A shift count of zero is legal, and leaves the link, AC, and MQ registers unchanged. Program execution resumes at the location following the shift count.
ASR	7415	<b>Arithmetic Shift Right.</b> The link is loaded from ACO and remains unaltered for the remainder of the operation. The content of the AC and MQ is then shifted right as described above. The number of shifts performed is equal to the content of the 5 low-order bits (bits 7-11) of the next location in memory. A shift count of zero is legal, and loads the link from ACO but leaves the AC and MQ registers unchanged.

**Table 5-3 KE8-E Mode B Instructions (Cont.)**

MNEMONIC	OCTAL	OPERATION
		Bits shifted out of MQ11 are shifted into the GTF, to facilitate round-off operations. Program execution resumes at the location following the shift count.
LSR	7417	<b>Logical Shift Right.</b> The link is cleared and remains unaltered for the remainder of the operation. The content of the AC and MQ is shifted right as described above. The number of shifts performed is equal to the content of the 5 low-order bits (bits 7-11) of the next instruction in memory. A shift count of zero is legal, and clears the link without changing the AC or MQ registers. Bits shifted out of MQ11 are shifted into the GTF to facilitate round-off operations. Program execution resumes at the location following the shift count.
SCA	7441	<b>Step Counter OR with AC.</b> The content of the step counter is combined with the content of the low-order 5 bits of the AC (AC7-11) by a bitwise logical OR operation, and the result is loaded into AC7-11. AC0-6 remain unchanged.
SCA CLA	7641	<b>Step Counter to AC.</b> The content of the step counter is loaded into AC7-11. AC0-6 are cleared. This instruction is a microprogrammed combination of SCA and CLA.
SAM	7457	<b>Subtract AC from MQ.</b> The content of the AC is subtracted from the content of the MQ in two's complement arithmetic. The result is loaded into the AC. The MQ remains unchanged. If a borrow is propagated from the most significant bit, the link is set. Otherwise, the link is cleared. Hence, the link is set if and only if the original content of the AC was less than or equal to the content of the MQ. The GTF is helpful when comparing signed numbers. It is set if the signed number in the MQ is greater than or equal to the original signed number in the AC, and cleared otherwise.
DAD	7443	<b>Double Precision Add.</b> The double precision word addressed by the next memory location is added to the previous content of the AC and MQ registers. If extended memory is installed, the double-precision word is obtained from the current data field. If there is a carry from the most significant bit, the link is set. If there is

**Table 5-3 KE8-E Mode B Instructions (Cont.)**

MNEMONIC	OCTAL	OPERATION
		no carry, the link is cleared. Program execution resumes at the memory location following the operand address. This instruction may be microprogrammed with the CAM instruction to produce a double precision load (DLD) instruction.
DST	7445	<b>Double Precision Store.</b> The content of the MQ and AC is stored at the double precision location addressed by the next memory location. If extended memory is installed, the storage location will be in the current data field. The AC, MQ and link remain unchanged. Program execution resumes at the location following the operand address. This instruction may be microprogrammed with the CAM instruction to produce a Double Precision Deposit Zero (DDZ) instruction.
DPIC	7573	<b>Double Precision Increment.</b> The double precision constant "one" is added to the double precision number in the AC and MQ by two's complement arithmetic. The high-order carry (or lack thereof) is propagated into the link. This instruction requires that the MQL and MQA bits be set to function as defined.
DCM	7575	<b>Double Precision Complement.</b> The content of the AC and MQ, considered as a 24-bit number, is complemented and incremented. This has the effect of replacing the content of the AC and MQ with its two's complement. The high-order carry (or lack thereof) is propagated into the link. This instruction requires that the MQL and MQA bits be set in order to function as defined.
DPSZ	7451	<b>Double Precision Skip if Zero.</b> The double precision number contained in the AC and MQ is tested. If all bits are zero, the PC is incremented to skip the next sequential instruction. If any bit is 1, the next instruction is executed.

Table 5-4 lists the major differences between mode A and mode B instructions.

**Table 5-4 Mode A and Mode B Instruction Differences**

INSTRUCTION	MODE A	MODE B
MUY	The next location holds the multiplier.	The next location holds the address of the multiplier.
DVI	The next location holds the divisor.	The next location holds the address of the divisor.
SHL, LSR, ASR	The next location holds one less than the number of shifts. On Right Shifts, MQ11 is lost.	The next location holds the number of shifts. (A shift of zero places is legal). On Right Shifts, MQ11 is shifted into the GT flag.

Figure 5-4 summarizes cycle times and indicates the longest practical machine cycle. Note that the longest cycle time plus 0.3  $\mu$ s. is the maximum time to enter a DMA cycle, provided the Break Device synchronizes at Int. Strobe time as recommended in Chapter 9. It is possible, by a small amount of programming, to reduce the longest cycle to 6.2  $\mu$ s. This programming consists of pretesting the AC on a normalize, and limiting long shifts to 15 places. Note, for example, that

MQL                    /AC MQ, O AC  
 LSR                    /Mode B Shift, 6 places  
 6

is equivalent to an 18-bit logical right shift and has a longest cycle of 3.5  $\mu$ s., rather than 7.1  $\mu$ s. Also, the total execution time for a straight 18-bit shift is 8.3  $\mu$ s., as opposed to 5.9  $\mu$ s. for the above sequence.

	MODE A			MODE B			NOTES
	MEM CYCLES	INSTR TIME	LONGEST CYCLE	MEM CYCLES	INSTR TIME	LONGEST CYCLE	
SWAB	1	1.2 $\mu$ s	1.2 $\mu$ s	1	1.2	1.2	
SWBA	1	1.2	1.2	1	1.2	1.2	
SCL	2	2.6	1.4	Not Available			
ACS	Not Available			1	1.2	1.2	
MUY	2	7.4	6.2	3	8.6	6.2	
DVI	2	7.4	6.2	3	8.6	6.2	No overflow
NMI	1	1.5+.3N	8.1	1	1.5+.3N	8.1	
SHL	2	2.6+.3N	8.9*	2	2.9+.3N	9.2**	25-place shift
ASR	2	2.6+.3N	8.9*	2	2.9+.3N	9.2**	25-place shift
LSR	2	2.6+.3N	8.9*	2	2.9+.3N	9.2**	25-place shift
SCA	1	1.2	1.2	1	1.2	1.2	
DAD	Not Available			4	5.2	1.4	
DST	Not Available			4	5.2	1.4	
DPSZ	Not Available			1	1.2	1.2	
DPIC	Not Available			1	1.6	1.6	
DCM	Not Available			1	1.6	1.6	
SAM	Not Available			1	1.2	1.2	

\*Computed from 1.4+.3N

\*\*Computed from 1.7+.3N

## **MEMORY EQUIPMENT OPTIONS**

The basic 4K or 8K memory supplied with the PDP-8/E may be expanded in increments of 4K or 8K. One memory extension and timeshare control is required whenever the system is expanded above 4K. Expansion in 4K increments is achieved by adding one MC8-E Core Memory with Memory Extension and Time Share Control to the 4K system, and then adding up to 6 units of the MM8-E 4K Core Memory. The 4K system is expanded in 8K increments with the addition of one MC8-EJ 8K Core Memory with Memory Extension and Timeshare Control, plus one or two units of MM8-EJ 8K Core Memory. The 8K memory extensions and 4K memory extensions may be mixed in one processor. In any event, either one MC8-E, one MC8-EJ or one KM8-E (described below) is required whenever the basic 4K memory is expanded. The memory extension and timeshare control portion of the MC8-E and MC8-EJ are programmed in the same manner as the KM8-E.

### **KM8-E Memory Extension and Time-Share Option**

This option provides the user with two primary capabilities. The memory extension portion extends the addressing capabilities of the machine from 4069 words up to 32,768 words. The time-share portion enables the computer to operate in either the normal manner (Executive Mode) or the User Mode. User Mode enables the machine to function in a time-sharing environment in which a user program is prevented from disturbing or interfering with another user program. The KM8-E option is packaged on one PDP-8/E module that plugs into the OMNIBUS. This option is required whenever memory capacity is extended beyond 4096 words.

The functional circuit elements which make up the memory extension control perform as follows:

**Instruction Field Register (IF)**—The IF is a three-bit register that serves as an extension of the PC. The contents of the IF determine the field from which all instructions are taken and the field from which operands are taken in directly-addressed AND, TAD, ISZ, or DCA instructions. Depressing the console EXTD ADDR LOAD switch transfers the instruction field in SWITCH REGISTER bits 6 through 8 into the IF register. During a JMP or JMS instruction, the IF is set by transfer of information from the instruction buffer register. When a program interrupt occurs, the contents of the IF are automatically stored in bits 0 through 2 of the save field register for restoration to the IF from the instruction buffer register at the conclusion of the program interrupt subroutine.

**Data Field Register (DF)**—This three-bit register determines the memory field from which operands are taken in indirectly-addressed AND, TAD, ISZ, or DCA instructions. Depressing the console EXTD ADDR LOAD switch transfers the SWITCH REGISTER bits 9 through 11 into the DF register. During a CDF instruction, the DF register is loaded from MD6-8 to establish a new data field. When a program interrupt occurs, the contents of the DF are automatically stored in bits 3-5 of the save field register. The DF is set by a transfer of information from save field register bits 3 through 5 by the RMF instruction. This action is required to restore the data field at the conclusion of the program interrupt subroutine.

**Instruction Buffer Register (IB)**—The IB serves as a three-bit input buffer for the instruction field register. All field number transfers into the instruction field register are made through the instruction buffer, except transfers from the operator's console switches. The IB is set by depressing of the console EXTD ADDR LOAD switch in the same manner as the instruction field register. A CIF microinstruction loads the IB with the programmed field on MD6-8. An RMF microinstruction transfers save field register bits 0 through 2 into the IB to restore the instruction field that existed before a program interrupt.

**Save Field Register (SF)**—When a program interrupt occurs, this seven-bit register is loaded from the user build flip-flop, and the IR and UF registers. The SF register is loaded during the cycle in which the program count is stored at address 0000 of the JMS instruction forced by a program interrupt request, then the instruction field and data field registers are cleared. An RMF instruction can be given immediately before exit from the program interrupt subroutine to restore the instruction field and data field by transferring the SF into the IB and the DF registers. (Also, see GTF and RTF instructions.)

**Extended Address Gating**—This logic consists of an output gating structure and control logic for gating the extended memory field address to core memory. The contents of the IF register are placed on the EMA0-2 lines unless an AND I, TAD I, ISZ I, or DCA I instruction is encountered. If such an instruction is encountered, the contents of the IF are placed on EMA 0-2 for the Fetch and Defer cycles, and the contents of the DF are placed on EMA0-2 for the Execute cycle. The extended memory field address is changed only at TP4 and remains available for the entire memory cycle.

**Data Transfer Gating**—This gating allows the contents of the save field register, instruction field register, or the data field register to be strobed into the accumulator via DATA lines 6-11. During an RIB or GTF instruction, bits 6 through 11 of the AC receive contents of the save field register. During an RIF instruction, bits 6 through 8 of the AC receive the contents of the instruction field register. During an RDF instruction, bits 6 through 8 of the AC receive the contents of the data field register.

**Device Selector and Instruction Decoding**—Bits 3 through 5 of the IOT instruction are decoded to produce the IOT command pulses for the memory extension control. Bits 6 through 8 of the instruction are not used for device selection since they specify a field number in some commands. Therefore, the select code for this device selector is designated as 2N. Bits 9 through 11 are also decoded to implement specific commands. The instruction decoding logic is common to the time-share portion of the KM8-E option.

#### **Programming**

Instructions associated with the extended memory portion KM8-E option are defined below:

### **Get Flags (GTF)**

Octal Code: 6004

Operation: Reads the contents of the interrupt inhibit flip-flop, and the SF register to AC3, AC5-11 respectively. The other AC bits are loaded with information from the CPU and the EAE; i.e., link, greater-than-flag, interrupt bus, interrupt on.

### **Restore Flags (RTF)**

Octal Code: 6005

Operation: Loads the user buffer flip-flop, the instruction buffer register, and the data field register with the contents of AC bits 5, 6-8, and 9-11 and inhibits processor interrupts until the next JMP or JMS instruction. At the conclusion of the JMP or JMS instruction, the contents of the user buffer flip-flop and the instruction buffer register are transferred into the user field flip-flop and the instruction field register, respectively. The contents of the other AC bits are loaded into the CPU and EAE to cause the converse of the GTF instruction. The Interrupt On flip-flop in the CPU is unconditionally set by this instruction.

### **Change to Data Field N (CDF)**

Octal Code: 62N1

Operation: Loads the data field register with the program-selected field number ( $N = 0$  to 7). All subsequent memory requests for operands are automatically switched to that data field, except for directly-addressed AND, TAD, ISZ, or DCA instructions.

### **Change to Instruction Field N (CIF)**

Octal Code: 62N2

Operation: Loads the instruction buffer register with the program-selected field number ( $N = 0$  to 7) and inhibits processor interrupts until the next JMP or JMS instruction. At the conclusion of either of these instructions, the contents of the instruction buffer register are transferred into the instruction field register.

### **Change Data Field, Change Instruction Field (CDF, CIF)**

Octal Code: 62N3

Operation: Performs the combination of CDF and CIF operations.

### **Read Data Field (RDF)**

Octal Code: 6214

Operation: ORs the contents of the data field register into bits 6-8 of the AC. All other bits of the AC are unaffected.

### **Read Instruction Field (RIF)**

Octal Code: 6224

Operation: ORs the contents of the instruction field register into bits 6-8 of the AC. All other bits of the AC are unaffected.

### **Read Interrupt Buffer (RIB)**

Octal Code: 6234

Operation: ORs the contents of the save field register (which is loaded from the instruction and data field during a program interrupt) into bits 6-8 and 9-11 of the AC, respectively. Thus, AC 6-11 contains the instruction and data fields that were in use before the last program interrupt. AC 5 is loaded by the time-share bit of the save field register. All other bits of the AC are unaffected.

### **Restore Memory Field (RMF)**

Octal Code: 6244

Operation: Restores the contents of the save field register (which is loaded from the instruction and data field during a program interrupt) into the instruction buffer, the data field register, and the user buffer (if time share option is enabled). This command is used upon exit from the program interrupt subroutine in another field.

Instructions and data are accessed from the currently assigned instruction and data fields, where instructions and data may be stored in the same or different memory fields. When indirect memory references are executed, the operand address refers first to the instruction field to obtain an effective address, which, in turn, refers to a location in the currently assigned data field. All instructions and operands are obtained from the field designated by the contents of the instruction field register, except for indirectly addressed operands, which are specified by the contents of the data field register. In other words, the DF is effective only in the execute cycle that is directly preceded by the defer cycle of a memory reference instruction, as follows:

Indirect (Bit 3)	Page or Z Bit (Bit 4)	Field In IF	Field In DF	Effective Address
0	0	m	n	The operand is in page 0 of field m at the page address specified by bits 5 through 11.
0	1	m	n	The operand is in the current page of field m at the page address specified by bits 5 through 11.
1	0	m	n	The absolute address of the operand in field n is taken from the contents of field m located in page 0 designated by bits 5 through 11.

1           1           m           n     The absolute address of the operand in field n is taken from the contents of field m located in the current page, designated by bits 5 through 11.

Each field of extended memory contains eight auto-index registers in addresses 10 through 17. For example, assume that a program in field 2 is running (IF = 2) and using operands in field 1 (DF = 1) when the instruction TAD I 10 is fetched. The defer cycle is entered (bit 3 = 1), and the contents of location 10 in field 2 are read, incremented, and re-written. If address 10 in field 2 originally contained 4321, it now contains 4322. In the execute cycle, the operand is fetched from location 4322 of field 1. Program control is transferred between memory fields by the CIF instruction. The instruction does not change the instruction field directly, as this would make it impossible to execute the next sequential instruction; instead, it loads the new instruction field in the IB for automatic transfer into the IF when either a JMP or JMS instruction is executed. The DF is unaffected by the JMP and JMS instructions.

The 12-bit program counter is set in the normal manner and, because the IF is an extension on the most significant end of the PC, the program sequence resumes in the new memory field following a JMP or JMS. Entry into a program interrupt is inhibited after the CIF instruction until a JMP or JMS is executed.

#### NOTE

The IF is not incremented if the PC goes from 7777 to 0000. This feature protects the user from accidentally entering a nonexistent field.

To call a subroutine that is out of the current field, the data field register is set to indicate the field of the calling JMS, which establishes the location of the operands as well as the identity of the return field. The instruction field is set to the field of the starting address of the subroutine. The following sequence returns program control to the main program from a subroutine that is out of the current field.

```
/PROGRAM OPERATIONS IN MEMORY FIELD 2  
/INSTRUCTION FIELD = 2; DATA FIELD = 2  
/CALL A SUBROUTINE IN MEMORY FIELD 1  
/INDICATE CALLING FIELD LOCATION BY THE CONTENTS OF THE DATA FIELD
```

CIF 10	/CHANGE TO INSTRUCTION
	/FIELD 1 = 6212
JMS I SUBRP	/SUBRP = ENTRY ADDRESS
CDF 20	/RESTORE DATA FIELD

SUBRP,    SUBR	/POINTER
	/CALLED SUBROUTINE, LOCATED IN
	/FIELD 1

SUBR,	0	/RETURN ADDRESS STORED HERE
	CLA	
	RDF	/READ DATA FIELD INTO AC
	TAD RETURN	/CONTENTS OF THE AC = 6202 + /DATA FIELD BITS
	DCA EXIT	/STORE INSTRUCTION SUBROUTINE /NOW CHANGE DATA FIELD IF DESIRED
EXIT,		
	JMP I SUBR	/A CIF INSTRUCTION
RETURN, CIF		/RETURN TO CALLING PROGRAM
		/USED TO CALCULATE EXIT
		/INSTRUCTION

When a program interrupt occurs, the current instruction and data field numbers are automatically stored in the 6-bit save field register, then the IF and DF are cleared. The 12-bit program count is stored in location 0000 of field 0 and program control advances to location 0001 of field 0. At the end of the program interrupt subroutine, the RMF instruction restores the IF and DF from the contents of the SF. Alternatively, the GTF and RTF instructions may be used to handle the Save Field and Link information. The following instruction sequence at the end of the program interrupt subroutine continues the interrupted program after the interrupt has been processed:

/RESTORE MQ IF REQUIRED

/RESTORE L IF REQUIRED

	CLA	
	TAD AC	/RESTORE AC
	RMF	/LOAD IB FROM SF
	ION	/TURN ON INTERRUPT SYSTEM
	JMP I 0	/RESTORE PC WITH CONTENTS OF /LOCATION 0 AND LOAD IF FROM IB
OR		
0,	0	/PC STORAGE
	DCA ACSV	/SAVE AC,
	MQA CLA	
	DCA MQSV	/MQ,
	GTF	
	DCA FLAGS	
	CLA	
	TAD MQSV	

MQL	/RESTORE MQ
TAD FLAGS	
RTF	/REPLACE FLAGS, ION
CLA	
TAD ACSV	/AC
JMP I O	/AND EXIT

### **Time-Share Description**

The Time-Share portion of the KM8-E operates in two modes as denoted by the user flag (UF) flip-flop. When the UF flip-flop is in the logic 1 state, operation is in the user mode and a user program is running in the central processor. When the UF flip-flop is in the logic 0 state, operation is in the executive mode and the time-sharing system's monitor is in control of the central processor. The four instructions (CINT, SINT, CUF, and SUF) are used by the time-sharing system's monitor in the executive mode and are never used by a user program. If a user program attempted to use one of these instructions, execution of the instruction would be blocked (see next paragraph). The KM8-E option adds the necessary hardware to the PDP-8/E to implement these instructions.

In executive mode, the computer operates normally. When the computer is operated in user mode, operation is normal except for IOT, HLT, LAS, and OSR instructions. When one of these instructions is encountered, the hardware inhibits the normal instruction sequence (other than rewriting the instruction in memory), and generates an interrupt at the end of the current memory cycle by setting the UINT flip-flop. The time-sharing system's monitor program then analyzes the source of interrupt, and takes appropriate action.

The time-share option requires at least 8K of core memory; thus, it is packaged with the memory extension option. A jumper on the KM8-E module is used to select the time-share function. The module is shipped with this jumper in place (time-share function disabled).

### **Programming**

Instructions associated with the time-share portion of the KM8-E are defined as follows:

#### **Clear User Interrupt (CINT)**

Octal Code: 6204  
 Operation: Clears the user interrupt flip-flop.

#### **Skip on User Interrupt (SINT)**

Octal Code: 6254  
 Operation: Increments the PC when the user interrupt flip-flop is set so the next sequential instruction is skipped.

#### **Clear User Flag (CUF)**

Octal Code: 6264  
 Operation: Clears the user buffer flip-flop.

#### NOTE

If the machine is stopped while in user mode, the user flag (UF) is cleared by operating the extended address load key (EXT ADDR LOAD).

Octal Code: 6274

Operation: Sets user buffer flip-flop and inhibits processor interrupts until the next JMP or JMS instruction. At the conclusion of either of these instructions, the content of the user buffer flip-flop is transferred into the user field flip-flop.

#### MP8-E Memory Parity

The memory parity option adds the circuits required to generate, store, and check the parity of memory words. This option replaces the 12-bit memory system with, effectively, a 13-bit system by adding the generating and storage capabilities for the parity bit. Odd parity (odd number of binary ones in the 13-bit word) is generated and stored for each word entered into memory. Parity is formed for each word retrieved from memory and this result is checked against its stored parity bit. If the two differ, a parity error flag is set to indicate that an error occurred. This flag is normally connected to the program interrupt system to cause the computer to enter a program interrupt subroutine for locating the interrupt source. Once the interrupting source is located, the computer enters an appropriate service routine to service the error condition. This routine can repeat the program step in which the error occurred to verify the error condition, can perform a simple read/write check for the error's address, or can determine machine status for the error detected and re-establish or print out these conditions, and then halt. The routine can also return the machine to the main program.

The MP8-E option consists of three PDP-8/E modules that plug into the OMNIBUS. Two of these modules (X-Y Driver and Current Source, and Core Stack) are identical to those of the MM8-E basic core memory and use the same addressing methods. However, only eight bits of the possible 12 bits are used. These eight-bit locations correspond to the eight possible memory fields and store up to 32,768 ( $8 \times 4096$ ) parity bits. The third module (Sense-Inhibit) contains device and operation decoding circuits, field decoding circuits, eight sense amplifiers, an eight-bit register, eight inhibit drivers and circuits for controlling the operations. This module also contains three control and status flip-flops that are controlled by IOT instructions. These flip-flops select odd or even parity generation and checking, enable or disable interrupts for parity errors, and store a parity error condition.

The following routine initializes the parity bits for a read-only or write-protected memory:

```
LOOP, TAD I 10          /INITIALIZE LOC 10 WITH STARTING ADD.  
ISZ COUNT             /TURN OFF PARITY INTERRUPT  
JMP LOOP              /SET COUNTER  
                      /READ DATA, REWRITE PARITY  
                      /CONTINUE UNTIL DONE  
                      /CLEAR PARITY ERROR FLAG  
                      /TURN ON PARITY INTERRUPT
```

## **Programming**

Instructions associated with the MP8-E option are:

### **Disable Memory Parity Error Interrupt (DPI)**

Octal Code: 6100

Operation: Disables the generation of interrupts for parity errors by clearing the interrupt enable flip-flop of the memory parity option.

### **Skip On No Memory Parity Error (SMP)**

Octal Code: 6101

Operation: Senses the memory parity error flag; if it contains a 0 (signifying no error has been detected), the PC is incremented so that the next instruction is skipped.

### **Enable Memory Parity Error Interrupt (EPI)**

Octal Code: 6103

Operation: Enables interrupts from the memory parity option. The memory parity interrupt is automatically enabled when power is turned on, by the CLEAR key on the front panel and by the CAF IOT instruction.

### **Clear Memory Parity Error Flag (CMP)**

Octal Code: 6104

Operation: Clears the memory parity error flag. The parity error flag is also cleared when power is turned on, by the CLEAR key on the front panel, and by the CAF IOT instruction.

### **Skip on No Memory Parity Error and Clear Memory Parity Error Flag (SMP, CMP)**

Octal Code: 6105

Operation: Senses the memory parity error flag; if it contains a 0, the next instruction is skipped. The memory parity error flag is then cleared.

### **Check For Even Parity (CEP)**

Octal Code: 6106

Operation: Causes parity to be checked for an even number of binary 1's in the entire word. This operation is effective only during the execute cycle immediately following this instruction.

### **Skip on Memory Parity Option (SPO)**

Octal Code: 6107

Operation: Increments the PC when the system includes a memory parity option so that the next sequential instruction is skipped.

Use of these instructions is discussed below:

- a. The DPI instruction is useful in certain diagnostic maintenance programs where it is desired to disable interrupts resulting from parity errors. This instruction also gives the user more flexibility for multiple program interrupt usage.
- b. The SMP instruction is used as a programmed check for memory parity errors. When used in a program interrupt subroutine, this instruction can be followed by a jump to a portion of the routine that services the memory parity option.
- c. The EPI instruction is used to return the memory parity option to normal operation after a DPI command.
- d. The CMP instruction initializes the memory parity option in preparation for normal programmed operation of the computer.
- e. The CEP instruction is useful in diagnostic maintenance programs. By altering the parity check from odd to even, parity errors can be forced, to permit checking for proper functions of the parity option.
- f. The SPO instruction permits the user to automatically check whether or not the system is equipped with a memory parity option.
- g. The SMP, CMP instruction is a combination of SMP and CMP instructions, and permits the operations performed by these instructions to be implemented by one instruction.

#### **MR8-EA 256-Word Read-Only-Memory**

The MR8-EA option provides the user with read-only-memory (ROM) capabilities such as might be used for hardwired controller, communications or process-control functions. This option is provided in 256-word increments package on one module. However, the module, because of its thickness, requires two module slots.

Information stored in the ROM is established by wiring the unit at the factory. The information content must be specified by the user at the time of purchase.

The number of ROM modules used is limited only by the amount of basic core or read/write capability required and the maximum address capabilities of the machine. A ROM can be assigned any memory field address, however, it must be assigned a block of 256 addresses beginning with an even-number memory page. Field and page addresses are selected by jumpers on its address decoding circuits. When used in other than field 0, the KM8-E option is required.

In situations where a small amount of ROM is desired, an MR8-E can be installed which uses locations already allotted to the 4K memory. The MR8-E automatically disables core memory using the same address. The core addresses can be re-enabled by removal of the MR8-E.

#### **MI8-E Bootstrap Loader**

This option uses a 32-word read-only-memory (ROM) with diodes that can be arranged to accommodate any program up to 32 words in length. This option is normally used as a hardware Read-In-Mode (RIM) paper tape loader for loading of programs from the PDP-8/E paper tape reader of the console teleprinter. However, it can be used for any user-desig-

nated programs of 32 words or less. The MI8-E option is contained on one PDP-8/E module that plugs into the OMNIBUS.

The MI8-E operates in a shadow address mode with core memory. That is, the addresses used for this device can overlap core memory addresses and can be used by core memory whenever the MI8-E option is not operating. The MI8-E can be used in any memory field; the field is selected by jumpers on the module. For a 32-word program, the MI8-E occupies the last 32 locations in the field (7740 (octal) through 7777 (octal)). The starting and ending addresses within this 32-address group are selected by jumpers on the module. Thus, programs requiring less than 32 locations can also be readily implemented.

The MI8-E option is selected, using the console SW control. However, this control has no effect unless the machine is stopped (RUN flop is reset). When this control is depressed, addresses 7740 (octal) through 7777 (octal) access the MI8-E hardware only. Core memory is prevented from responding to these addresses by outputs of the MI8-E control logic.

To operate the M18-E option, the SW key is depressed, loading the starting address and starting the computer. The M18-E then assumes control and provides instructions from its ROM to the MD lines during each FETCH major state. These instructions can load paper tape programs from the PDP-8/E paper tape reader or the console teleprinter, or perform user-designated functions. When the ending address is reached (as determined by module jumpers and MA inputs), the last instruction is executed and the Bootstrap Loader resets itself.

#### **REAL TIME CLOCK OPTIONS**

##### **Type DK8-EA Real Time Clock (Line Frequency)**

The DK8-EA is a fixed-interval line frequency clock option to the PDP-8/E that causes an interrupt 100 or 120 times per second, depending on line frequency. The clock and control are contained on one PDP-8/E module, which plugs into the OMNIBUS.

#### **Programming**

The following instructions control the DK8-EA line frequency clock:

##### **Enable Interrupt (CLEI)**

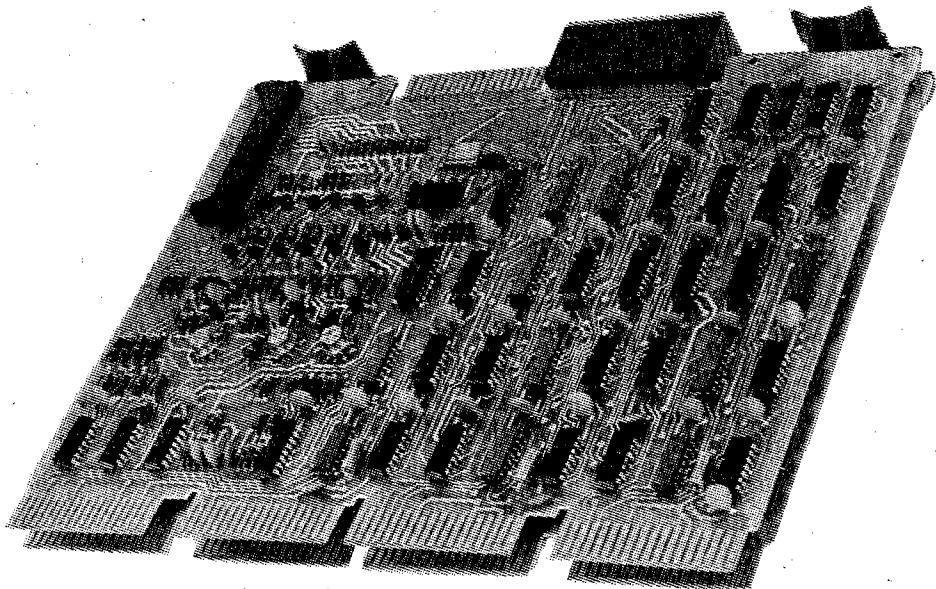
Octal Code: 6131

Operation: Enables the clock interrupt so that each clock pulse will cause a program interrupt request.

##### **Disable Clock Interrupt (CLDI)**

Octal Code: 6131

Operation: Disables the clock interrupt so that the clock cannot cause program interrupts.



**DK8-EP Real Time Clock**

**Skip on Clock Flag and Clear Flag (CLSK)**

Octal Code: 6133

Operation: Senses the clock flag, which is set with each clock pulse; if it is set, the next sequential instruction is skipped, and the clock flag is cleared.

**Type DK8-EC Real Time Clock (Crystal)**

The DK8-EC is a fixed-interval crystal-controlled clock option to the PDP-8/E that is used to cause an interrupt every 50, 500, or 5,000 times per second (jumper selectable). The clock frequency is derived from a 20-MHz crystal. The clock and control are contained on one PDP-8/E module, which plugs into the Omnibus.

**Programming**

The instructions which control the DK8-EC crystal clock are the same as those shown above for the DK-EC line frequency clock.

### Type DK8-EP Programmable Real Time Clock

The DK8-EP real time clock option offers the PDP-8/E user a method for accurately measuring and counting intervals or events in a number of ways.

The DK8-EP system consists of two PDP-8/E modules (M860 and M518) containing:

- a. A 12-bit binary counter using MSI integrated circuits with an overflow bit.
- b. A 12-bit buffer register.
- c. A 20-MHz crystal clock with frequency dividers.
- d. All associated control functions, IOT decoding and registers.
- e. Three Schmitt Trigger input event circuits (requires DK8-EF).

Logically, the DK8-EP contains the following features:

- a. **Clock Enable Register**  
This register controls the rate of the time base and the mode of counting, and selectively enables each of the three input channels and the interrupt line.
- b. **Clock Buffer**  
The Clock Buffer stores data being transferred from the AC to the clock counter, or from the clock counter to the AC. It also permits presetting of the clock counter.
- c. **Clock Counter**  
This register is a 12-bit binary counter that may load the clock buffer or to be loaded from it. When an overflow occurs and the clock enable mode is 01, the clock buffer is automatically loaded into the clock counter. The overflow is set by the most significant bit of the clock enable register going from 1 to 0.
- d. **Programmable Time Base**  
The Programmable Time Base provides count pulses to the clock counter according to the rate set by the clock enable register.
- e. **Crystal Clock**  
The clock is a simple crystal-controlled clock, which operates at 20 MHz + or - 0.1%. MSI integrated circuit decade counters divide the base clock frequency down to any of the following rates: 1 MHz, 100 kHz, 10 kHz, 1 kHz, or 100 Hz.

### Programming

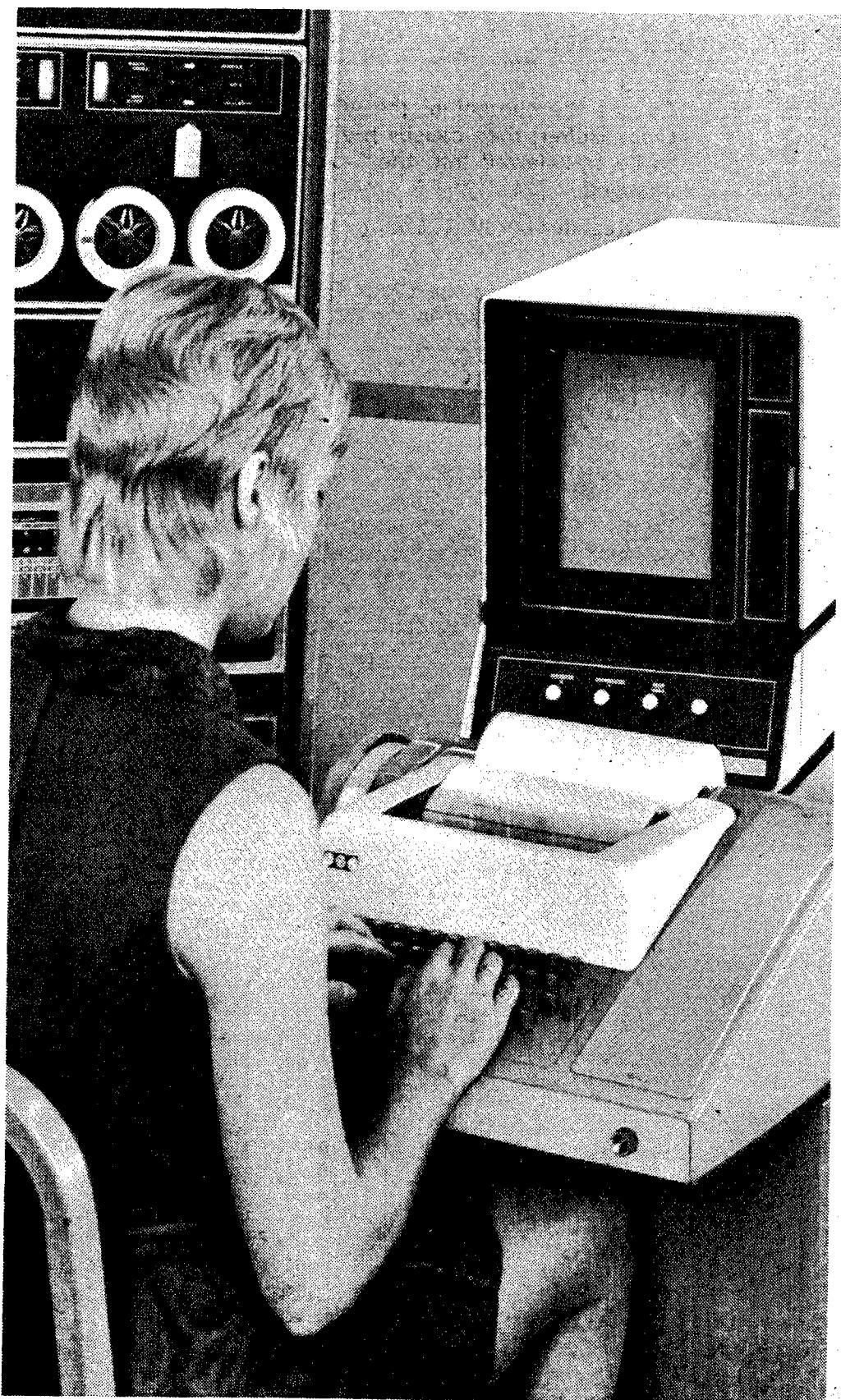
The following IOT instructions control the DK8-EP real time clock:

#### Skip on Clock Interrupt (CLSK)

Octal Code: 6131

Operation: Causes the content of the PC to be incremented by one if an interrupt condition exists, so that the next instruction is skipped. The interrupt conditions are as follows:

- \*a. Enable Event Interrupt 1 and Input 1
- \*b. Enable Event Interrupt 2 and Input 2
- \*c. Enable Event Interrupt 3 and Input 3
- \*d. Enable Overflow Interrupt and Overflow



### **AC to Clock Buffer (CLAB)**

Octal Code: 6133

Operation: Causes the content of the AC to be transferred into the Clock Buffer; then causes the content of the Clock Buffer to be transferred into the Clock Counter. The AC is not changed.

### **Clear Clock Enable Register per AC (CLZE)**

Octal Code: 6130

Operation: Clears the bits in the Clock Enable Register corresponding to those bits set in the AC. The AC is not changed.

### **Set Clock Enable Register per AC (CLDE)**

Octal Code: 6132

Operation: Sets the bits in the Clock Enable Register corresponding to those bits set in the AC. The AC is not changed.

### **Load Clock Enable Register (CLEN)**

Octal Code: 6134

Operation: Causes the content of the Clock Enable Register to be transferred into the AC.

### **Clock Enable Registers Functions**

AC BIT	FUNCTION
0	Enables clock overflow to cause an interrupt.
1 & 2	<p><b>Mode</b></p> <p>00 Counter runs at selected rate. Overflow occurs every 4096 counts. Flag remains set.</p> <p>01 Counter runs at selected rate. Overflow causes Clock Buffer to be transferred to the Clock Counter, which continues to run. Overflow remains set until cleared with IOT 6135.</p> <p>*10 Counter runs at selected rate. When an enabled event occurs, the Clock Counter is transferred to the Clock Buffer, and the Counter continues.</p> <p>*11 Counter runs at selected rate. When an enabled input occurs on any channel three, the Clock Counter is transferred to the Clock Buffer, and the Clock Counter continues to run from zero.</p>

### **Rate Selection**

Contents of Bits 3-5	Octal Value	Interval Between Pulses	Frequency
000	0	Stop	0
*001	1	-	External Input
010	2	10 <sup>-2</sup> sec	100 Hz
011	3	10 <sup>-3</sup> sec	1 KHz
100	4	10 <sup>-4</sup> sec	10 KHz
101	5	10 <sup>-5</sup> sec	100 KHz
110	6	10 <sup>-6</sup> sec	1 MHz
111	7	Stop	0

\* Available only as a LAB-8/E Option

- 6 Overflow starts ADC. (When the Clock Counter overflows, the analog-to-digital converter, type AD8-EA, is started.)
- 7 When set to 1, inhibits clock.
- \*8 Events in Channels 1, 2, or 3 cause an interrupt request and overflow.
- \*9, 10, & 11 Enable Events 1, 2, and 3
  - 9 —Event 3
  - 10—Event 2
  - 11—Event 1

#### **Clock Status to AC (CLSA)**

Octal Code: 6135  
 Operation: Interrogates the Clock Input and Overflow Status flip-flops. The clock status information is inclusively ORed into the AC, then the status bits corresponding to set AC bits are cleared. This ensures that only one occurrence of an Event will be transferred to the program. The status condition is established as follows:

AC Bit	Status Condition
0	Overflow
* 9	Event 3
*10	Event 2
*11	Event 1

#### **Clock Buffer to AC (CLBA)**

Octal Code: 6136  
 Operation: Clears the AC, then transfers the content of the Clock Buffer into the AC.

#### **Clock Counter to AC (CLCA)**

Octal Code: 6137  
 Operation: Clears the AC, transfers the content of the Clock Counter to the Clock Buffer, then transfers the content of the Clock Buffer into the AC.

#### **NOTE**

The clock counter may be read while it is counting. Gating in the clock control section prevents data from being strobed out of the counter before a specified time following a clock pulse. This time, approximately 300 ns, allows the data to settle in the counter.

This feature allows the counter to be read any number of times without introducing timing errors in counting the amount of time between intervals, and also eliminates false counts that are the result of reading the counter as one or more bits are in transition from one state to another.

\* Available only on LAB-8/E Option

### **Example Subroutine #1**

This example illustrates how the DK8-EP can be used as a double-precision (24-bit) free-running clock, using the clock counter as the low order 12 bits and a memory location as the high order 12 bits. Because all of the clock's registers have been set to zero initially by the clear key, the program needs only to zero the high order words, set the enable register, and turn on the interrupt. After 4096 counts, the clock counter overflows, signalling an interrupt. The service routine simply increments the high order word, then returns to the main program.

```
CLA  
DCA HIGH    /ZERO HIGH ORDER WORD  
TAD ENABLE  /OVER + MODE 00 + RATE  
CLOE        /SET ENABLE REGISTER  
ION          /INTERRUPT ON
```

ENABLE = OVERFL + MODE 00 + RATE

```
/SERVICE ROUTINE  
CLSK         /CLOCK SKIP?  
JMP OTHERS   /NOT A CLOCK FLAG  
CLSA         /READ STATUS, CLEAR FLAGS  
SPA CLA      /IGNORE OTHER CLOCK INTERRUPTS  
ISZ HIGH     /INCREMENT HIGH  
JMP RETURN   /RETURN TO MAIN PROGRAM
```

With this simple program, time can be kept during program execution. With the clock set to its fastest rate (1  $\mu$ s per tick), this double-precision counter could mark time for only just over 16 seconds; with the clock set to its slowest rate, it could mark time for over 100 days.

A simple routine could be written to interrogate elapsed time by using the CLCA (clock counter to AC) command.

### **Example Subroutine #2**

The DK8-EP can also easily be programmed to function as an alarm clock, counting off a period of time, giving an alarm, automatically resetting itself, and continuing. The alarm could be used to ring a bell, as indicated in the example; however, a more practical use would be to start an analog-to-digital conversion to take a number of samples from the outside world.

This example will ring the bell every second:

```
START,      CLA  
           TAD COUNTER      /SET COUNTER TO -1000  
           CLAB  
           CLA  
           TAD ENABLE       /SET ENABLE REGISTER  
           CLOE  
AGAIN,      CLSK          /CLOCK SKIP?  
           JMP .-1  
           CLSA          /YES, READ STATUS  
           CLA  
           TAD BELL        /RING BELL
```

TLS	
TSF	
JMP .—1	
JMP AGAIN	
COUNTER,	—1750
ENABLE,	MODE 01 + 1 MS
BELL,	207

This program could easily be modified to work in the interrupt mode by setting bit 0 of the enable register to a 1. An interrupt would then occur every second, and this could be used to ring the bell.

#### Type KP8-E Power Fail Detect

The KP8-E and its related shut-down and restart subroutines are designed to restore computer operation automatically following a failure of the computer's primary power source. This OMNIBUS option protects an operating program in the event of such a failure by causing a program interrupt, enabling continued operation for 1 ms; this allows the interrupt routine to detect the low power as initiator of the interrupt and to store both the contents of active registers (AC, L, MQ, etc.) and the program count in known core memory locations.

Variations of the AC line below the predetermined threshold level at a rate of one per second or less will also cause the shut-down circuits to be activated. When power is restored the power low flag clears, and a routine beginning in address 0000 starts automatically. This routine restores the contents of the active registers and program counter to the conditions that existed when the interrupt occurred, then continues the interrupted program.

The power failure option consists of three circuits, contained on a single PDP-8/E module.

- a. A power interrupt circuit, which monitors the status of the computer power supply and sets a power low flag when power is interrupted (due to a power failure or to the operation of the POWER switch on the operator's console). This flag causes a program interrupt when an interruption in computer power is detected.
- b. A shutdown sequence circuit, which ensures that, when a power interrupt occurs, the computer logic circuits will continue operation for 1 ms to allow a program subroutine to store the contents of the active registers. If, at the end of the 1 ms interval, computer operation still continues, it is halted. When power conditions are suitable for computer operation, a restart circuit clears the power low flag and restarts the program. A manual RESTART switch located on the right side of the power fail module enables or disables the automatic restart operation. With this switch in the ON (up) position, the option clears the MA and produces a MEMORY START pulse 1500ms after power conditions are satisfactory. The MA is cleared so that operation restarts by executing the instruction in address 0000. That instruction must be a JMP to the starting address of the subroutine that restores the contents of the active registers and

the program counter to the conditions existing prior to the power low interrupt. The 1500-ms delay ensures that slow mechanical devices, such as Teletype equipment, have completed any previous operation before the program is resumed. Simulation of the manual START function causes the processor to generate a power clear pulse to clear internal controls and I/O device registers. With the RESTART switch in the OFF (down) position, the power low flag is cleared, but the program must be started manually, possibly after resetting peripheral equipment or by starting the interrupted program from the beginning. The shut-down circuitry is unaffected by the switch.

- c. A skip circuit provides programmed sensing of the condition of the power low flag by adding the IOT SPL (6102) instruction to the computer repertoire.

### **Programming**

#### **Skip On Power Low (SPL)**

Octal Code: 6102

Operation: Senses the content of the power low flag. If the power low flag contains a 1 (indicating that a power failure has been detected), the contents of the PC are incremented by one, so that the next sequential instruction is skipped.

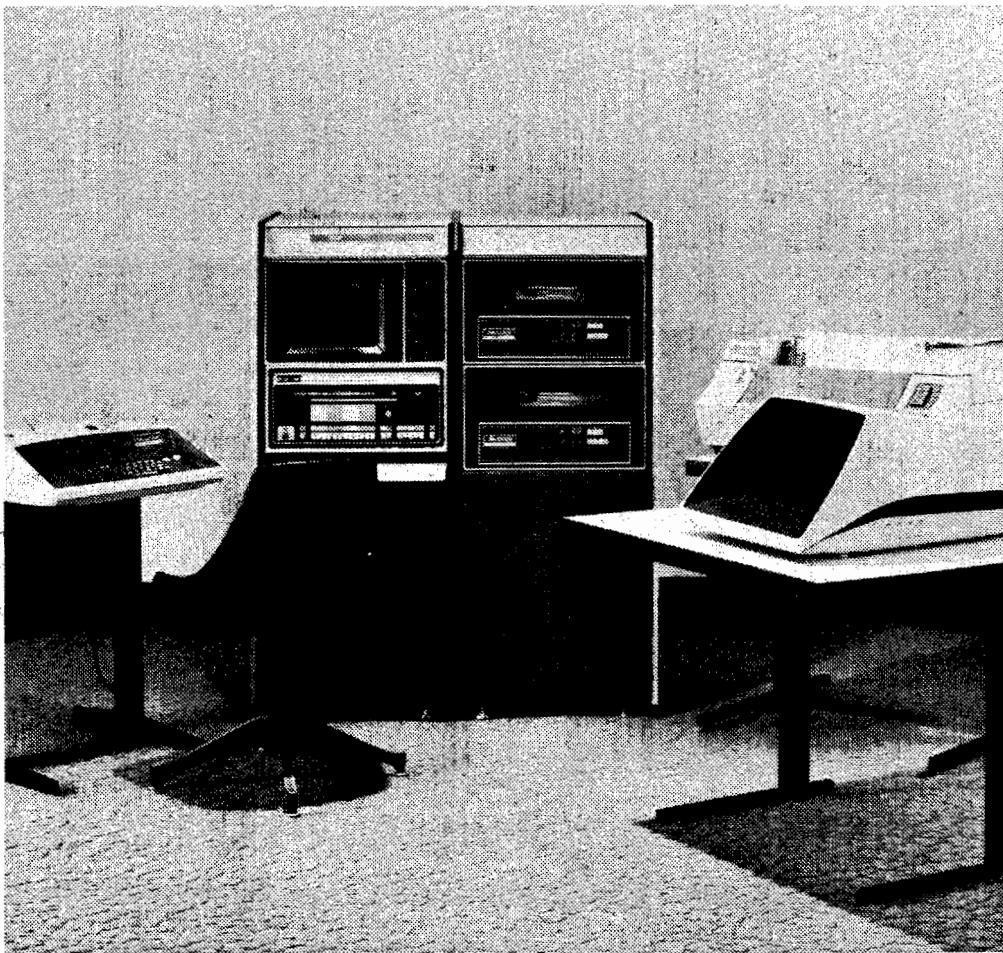
Because the time that computer operation can be extended after a power failure is limited to 1 ms, the condition of the power low flag should be the first status check made by the program interrupt subroutine. The interrupt subroutine, starting with the SPL microinstruction (and including the power fail program sequence), can be executed in less than 30  $\mu$ s. The power fail program sequence stores the contents of the active registers and program counter in designated core memory locations, then relocates the calling instruction of the power restore subroutine to address 0000, as follows:

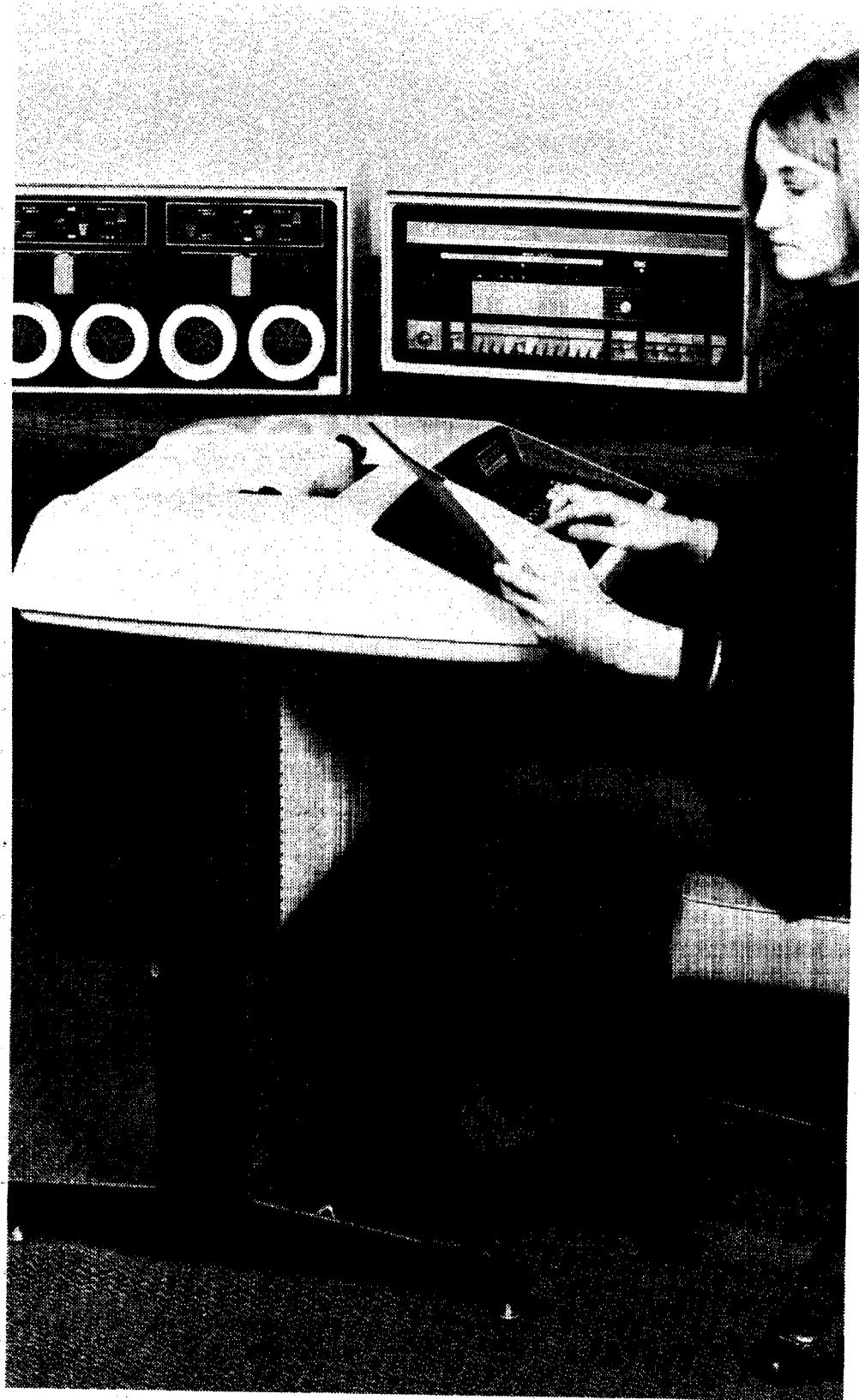
Address	Instruction	Remarks
0000	—	/STORAGE FOR PC AFTER PROGRAM INTERRUPT
0001	JMP FLAGS	/INSTRUCTION EXECUTED AFTER PROGRAM INTERRUPT
FLAGS,	SPL	/SKIP IF POWER LOW FLAG = 1
	JMP OTHER	/INTERRUPT NOT CAUSED BY POWER LOW, CHECK OTHER FLAGS
DCA AC		/INTERRUPT WAS CAUSED BY POWER LOW, SAVE AC
RAR		/GET LINK
DCA LINK		/SAVE LINK
MQA		/GET MQ
DCA MQ		/SAVE MQ
TAD 0000		/GET PC
DCA PC		/SAVE PC
TAD RESTRT		/GET RESTART INSTRUCTION

DCA 0000	/DEPOSIT RESTART INSTRUCTION IN 0000
HLT	
RESTRT, JMP ABCD	/ABCD IS LOCATION OF RESTART ROUTINE

Automatic program restart begins by executing the instruction stored in address 0000 by the power fail routine. The power restore subroutine restores the contents of the active registers, enables the program interrupt facility, and continues the interrupted program from the point at which it was interrupted, as follows:

Address	Instruction	Remarks
0000	JMP ABCD	
ABCD,	TAD MQ	/GET MQ
	MQL	/RESTORE MQ
	TAD LINK	/GET LINK
	CLL RAL	/RESTORE LINK
	TAD AC	/RESTORE AC
	ION	/TURN ON INTERRUPT
	JMP I PC	/RETURN TO INTERRUPTED PROGRAM





# 6

## internal i/o options

### CONSOLE TELEPRINTERS

#### DECwriter

The PDP-8/E DECwriter option comprises the LA30 DECwriter and LC8-E Control.

#### LC8-E DECwriter Control

The LC8-E DECwriter Control is an interface between the PDP-8/E processor and the parallel version of the LA30 DECwriter. The LC8-E Control is one PDP-8/E module which plugs into the OMNIBUS.

The LA30 DECwriter is available in EIA, 20mA and read only models. Device codes for the keyboard and printer are selectable by means of wired jumpers on the control module, allowing up to 17 LC8-E controllers to be installed on a single processor. Connections to the LA30 are made via a standard 25-foot (7.7-meter) cable which plugs directly into the LC8-E module.

In operation, the LA30 is considered as two devices, a keyboard and a printer. Therefore two device codes are assigned. If the LC8-E is used to replace the KL8-E console Teletype control, these device codes would be codes 03 for the keyboard and 04 for the printer. Other pairs of device codes can be assigned according to the normal sequence for additional Teletype controllers. The instruction list given assumes that device codes 03 and 04 have been selected. The control unit contains a programmable interrupt enable flip-flop which controls the generation of program interrupt requests from both the keyboard and printer. This flip-flop is set when power is turned on or when INITIALIZE is generated. It can also be set or cleared under program control (as specified by AC11) by the KIE instruction.

#### Specifications

Type of transmission	Parallel TTL levels
Type of reception	Parallel TTL levels
Number of data elements per character	Seven
Maximum input/output rate	30 characters per second*

\* See LA30 Specification

#### Keyboard

When a key is depressed on the LA30 keyboard, the seven bit ASCII representation of the character is established on the seven data input lines to the LC8-E control. Also generated is the signal Transmitter Stroke to transfer this character into the LC8-E input buffer and to set the keyboard (receiver) flag. This causes a program interrupt request if the interrupt enable flip-flop is set and can be tested by a skip IOT whether



DEC's new LA30 DECwriter is a dot matrix impact printer that operates at a speed of 30 characters per second, three times the speed of commonly used teleprinters. Its quiet operation and high reliability are the result of the systematic elimination of mechanical parts, substituting, where possible, solid state logic modules.

the interrupt is enabled or not enabled. A READ IOT transfers the buffer contents to AC5-11, sets AC4 and clears the keyboard (receiver) flag. Setting AC4 is to make the input character compatible with the Teletype, where the most significant bit is always set on keyboard input.

### **PROGRAMMING**

The following instructions assume device code 03 and that the LC8-E replaces the KL8-E. For any other device codes and in use as an additional keyboard, other mnemonics should be assigned.

#### **Clear Keyboard Flag (KCF)**

Octal Code: 6030

Operation: Clears the keyboard Flag.

#### **Skip on Keyboard Flag (KSF)**

Octal Code: 6031

Operation: Senses the keyboard flag and increments the PC if it is set, thereby skipping the next sequential instruction.

#### **Read Keyboard Buffer Static (KRS)**

Octal Code: 6034

Operation: Inclusively OR's the contents of the LC8-E input buffer with the AC and leaves the result in the AC Register.

#### **Set/Clear Interrupt Enable (KIE)**

Octal Code: 6035

Operation: Sets or clears the interrupt enable flip-flop as defined by AC11. Set if AC11(1); clear if AC11(0).

#### **Read Keyboard Buffer Dynamic (KRB)**

Octal Code: 6036

Operation: Performs the combined operations of KCC & KRS instructions. Clears the AC and the Keyboard Flag; loads AC5-11 from the LC8-E input buffer; sets AC4.

### **Printer**

An IOT instruction is used to load the LC8-E printer buffer from AC5-11 and clear the printer flag. The information in the buffer is transferred to the LA30 DECwriter on the seven data output lines from the LC8-E; when they have settled, the control line receive strobe is asserted to initiate a print operation. When the LA30 DECwriter has completed the print operation, it indicates that it is again ready to print by setting the printer flag in the LC8-E. This causes a Program Interrupt Request if Interrupt Enable is set; the flag can be tested by a SKIP IOT whether Interrupt is enabled or not enabled.

### **PROGRAMMING**

As with the Keyboard, it is assumed that the LC8-E replaces the KL8-E Console Teletype Control. The following instructions apply to the printer operation:

#### **Set Printer Flag (TFL)**

Octal Code: 6040

Operation: Sets the Printer Flag.

#### **Skip on Printer Flag (TSF)**

Octal Code: 6041

Operation: Senses the printer flag and increments the PC if it is set thereby skipping the next sequential instruction.

**Clear Printer Flag (TCF)**

Octal Code: 6042

Operation: Clears the Printer Flag.

**Load Printer Buffer and Print (TPC)**

Octal Code: 6044

Operation: Transfers AC5-11 to the LC8-E Printer Buffer and at TS1 of the next instruction asserts Receive Strobe to cause the character held in the buffer to be printed.

**Skip on Printer or Keyboard Interrupt (TSK)**

Octal Code: 6045

Operation: If either the printer flag or keyboard flag is set and the interrupt Enable flip-flop is set, increments the PC thereby skipping the next sequential instruction.

**Load Printer Sequence (TLS)**

Octal Code: 6046

Operation: This instruction combines TCF and TPC. It clears the printer flag and transfers the contents of AC5-11 to the LC8-E printer buffer. At TS1 of the following instruction, it asserts Receive Strobe to cause a character held in the buffer to be printed.

**LA30 Differences from Teletype**

From the above instruction lists it can be seen that the LC8-E is very similar to the KL8-E Console Teleprinter Control. There are differences mostly caused by the different characteristics of the LA30. These differences are summarized in the following:

1. There is no paper tape reader; hence no reader control,
2. There is no paper tape punch,
3. The maximum input/output rate is 30 characters/second,
4. Output to the printer section of the LA30 is only 7 bits (AC5-11), 8 bits can be sent but AC4 is ignored,
5. If a non-printing character is sent to the LA30, it does not go through a normal print cycle but indicates that it is ready to print again in approximately 1 to 2 $\mu$  seconds.
6. The LA30 has no hardware TAB, FORM FEED or VERTICAL TAB feature,
7. Carriage return takes several character times but the Printer Flag is set approx. 2 $\mu$ s after a CAR RET is sent. The Printer Flag is not set again until the CAR RET has finished and the next character has been printed,
8. It is possible, by changing the internal switch on the LA30, for the keyboard to generate lower case characters. Normally this is set so that upper case is generated whether the keyboard is in SHIFT or not. The Printer cannot print lower case; it interprets lower case codes as upper case,
9. There is no BELL, CNTRL G is treated as non-printing,
10. End of line (> 80 characters) is trapped and any subsequent characters sent before a CAR RET are not printed.

### **LA30 DECwriter**

DEC's new LA30 DECwriter is a dot matrix impact printer that operates at a speed of 30 characters per second, three times the speed of commonly used Teletypewriters. Its quiet operation and high reliability are the result of the systematic elimination of mechanical parts, substituting, where possible, solid state logic modules. This reduction in moving parts means, for instance, that when the DECwriter is idle no parts are moving; conventional Teletypewriters with their extensive mechanical linkages can wear out even while not being used.

In order to print a character on the DECwriter, a 7-dot matrix is moved along the 9 7/8" wide page by a stepping motor. Seven spring-loaded wires, driven by solenoids, are arranged vertically in the printing head. Characters are created while a solid state logic controlled motor advances the head along the line.

The DECwriter is a full-scale hard copy I/O terminal that uses a dot matrix to generate a character on ordinary paper; most others require special thermal or electrostatic paper. The DECwriter also uses the same widely available fan-folded paper as 80-column line printers, which allows a user to reduce costs by standardizing size and opening second sources for his paper supply. The terminal uses a standard, 1/2 in. wide, 40-yard nylon ribbon.

### **Specifications**

#### **Printer**

Printing Speed:	30 characters per second asynchronous; 250 ms carriage return (max.)
Line Length:	80 character positions
Character Spacing:	10 characters per inch
Line Spacing:	6 lines per inch
Paper:	9-7/8"-wide tractor-driven continuous form original plus one copy
Typeface:	5 x 7 dot matrix
Ribbon:	1/2-inch x 120 feet, nylon

#### **Data Entry**

Code:	USASCII-1968
	96 characters (128 optional)

#### **Interface:**

LC8-E

#### **Environmental/Physical**

Temperature:	50° F-100° F
Humidity:	5-90% (noncondensing)
Power:	Type LA30-PA: 115VAC, 60 Hz
Dimensions:	Type LA30-PD: 230VAC, 50 Hz 20-1/2 inches wide x 31 inches high x 24 inches deep

### **VT8-E Alphanumeric and Graphic Display Terminal**

The VT8-E is a low-cost, high-speed, alphanumeric and graphic CRT display with a teletype keyboard for data entry. It includes a separate 8-key cluster for cursor control, end of line, end of screen, home, and power on/off.

On input, data is transferred from the keyboard to the accumulator of the computer. A keyboard interrupt is generated (if enabled) by each key stroke. On output, the VT8-E operates with the computer using the single cycle data break, allowing data to be transferred at memory speed. Up to four VT8-Es may operate concurrently on one PDP-8/E, PDP-8/M, or PDP-8/F. Graphics and alphanumerics may be displayed at the same time by switching display modes at the refresh rate under program control.

In the graphic mode, the VT8-E will display 189 points per line x 200 lines, flicker-free. Each bit of a data word represents a dot or space on the screen. It takes 16 data words to display each line and 3200 words to display 200 lines.

In the alphanumeric mode, the VT8-E displays either 32 or 64 characters per line x 20 lines. Each character requires a single data word. A unique end-of-buffer character reduces buffer requirements for short display files.

The VT8-E employs a crystal-driven real-time clock which may be used separately by the computer system. A hard copy interface to an LA30A-P or LS01-ED line printer is included. Also included are built-in maintenance features which facilitate trouble shooting.

#### **Display**

#### **Display**

##### **alphanumeric mode**

**Viewing area:**  $8\frac{1}{4} \times 6\frac{1}{4}$  inches

32 characters/line

$8\frac{1}{4} \times 4\frac{1}{4}$  inches

64 characters/line

5 x 7 Dot matrix

20 lines of characters/screen

64 or 32 characters/line (jumper selectable)

64 ASCII character set

(upper case characters)

##### **graphic mode**

**Viewing area:**  $7 \times 6\frac{1}{4}$  inches

32 characters/line

$7 \times 4\frac{1}{4}$  inches

64 characters/line

Number of points/line—189

Number of lines —200

Maximum number of flicker-free points: 37,800

Character size: .09" width x .15" height (64 character)  
.185" width x .22" height (32 character)

Character spacing (horizontal): 40% of character width

Character spacing (vertical): 43% of character height

Refresh rate (screen): 50 or 60 frames/sec. to match local line frequency (jumper selectable)

Refresh method: raster scan

Standard features: selective blink, bright, blank, and cursor and programmable audible tone

**Keyboard:**

Capable of generating either upper case codes only or upper and lower case codes, selectable by switch on keyboard.

**Interface:** 3 quad modules which plug directly into the OMNIBUS™

**Power (display):** 115/230 Volts, 60/50 Hz

**Cable length:** 15 feet

**Real-time clock:** Generates flag or interrupt at refresh frequency (60 Hz/50 Hz); crystal controlled.

**VT8-E Instructions**

MNEMONIC	OCTAL CODE	FUNCTION
DPLA	6050	Load starting address of data buffer
DPGO	6051	Load starting extended address of data buffer and go—start display after next vertical retrace in one of the two modes. Enable or disable interrupts from keyboard and printer.
DPSM	6052	Stop the display. Inhibit video and further device-initiated breaks.
DPMB	6053	Maintenance instruction—perform a single one-cycle data break.
DPMD	6054	Maintenance instruction—read extended break, address or status registers.
DPCL	6056	Skip on real-time clock flag; clear the flag clear the flag if it is set.
DPBELL	6057	Generate a half-second audible tone.

**Keyboard Instructions**

DKCF	6030	Clear keyboard flag.
DKSK	6031	Skip on keyboard flag.
DKCC	6032	Clear keyboard flag, clear AC.
DKOB	6034	OR contents of keyboard buffer with AC, and deposit in AC.
DKIN	6035	Enable interrupt if AC 11 = 1. Disable interrupt if AC 11 = 0.
DKRB	6036	Read keyboard buffer—transfer contents of keyboard buffer to AC—clear keyboard flag.

**Printer Instructions**

PNSF	6040	Set printer flag.
PNSK	6041	Skip on printer flag.
PNCF	6042	Clear printer flag.
	6043	Not used.
PNLP	6044	Load printer buffer from AC 5-11—print.
PNSI	6045	Skip if about to interrupt.
PNPC	6046	Load printer buff—print—clear printer flag.

### Model ASR 33 Teletype

The standard Teletype Model ASR 33 (automatic send-receive) is used to type in or print out information at a rate of up to ten characters per second, or to read in or punch out perforated tape at ten characters per second. Signals transferred between the Model ASR 33 and the control logic are standard, serial, 11-unit code, Teletype signals. The signals consist of spaces and marks which correspond to open circuit and bias current in the Teletype, and to 0s and 1s in the Teletype control and computer. The start bit (space, 0, open circuit) and subsequent eight-character bits are one-unit-of-time duration and are followed by the stop bit, which occupies two units.

The eight bit code used by the Model ASR 33 Teletype unit is the American Standard code for information interchange (ASCII) modified. To convert the ASCII code to Teletype code, add 200 octal (ASCII + 200 (octal) = Teletype). This code is read in the normal octal form used in the computer. Bits are numbered from right to left, from 1 through 8, with bits 1 through 3 containing the least significant octal number. The first information bit transmitted is bit 1, which is read into AC11. Figure 7-5 illustrates the context and description of the ASCII Teletype code and its associated bit content in the AC.

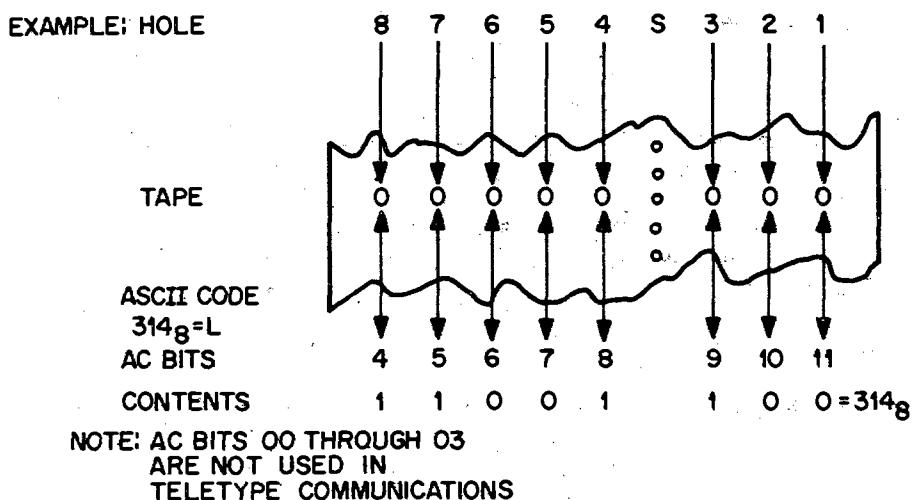


Figure 7-5 Relationship Between Teletype Tape AC Contents and Binary and Octal Number Being Transferred

The ASR 33 generates all assigned codes except 340 through 374 and 376, which are not assigned. Generally, codes 207, 212, 215, 240 through 337, and 377 are sufficient for Teletype operation. The ASR 33 detects all characters, but does not interpret all of the codes that it generates as commands.

The standard number of characters printed per line by the ASR 33 is 72. The sequence for proceeding to the next line is a carriage return followed by a line feed (as opposed to a line feed followed by a carriage return). Appendix C lists the Teletype character code. Punched tape format (for 264 (octal)) is as follows:

<b>Binary Code</b>			<b>Tape Channel</b>	
(Punch = 1)	8 7	6 5 4	S	3 2 1
Octal Code	1 0	1 1 0	(Sprocket)	1 0 0

### Teleprinter Control

Refer to Data Communications Equipment Options—KL8-E Asynchronous Data Control.

### PAPER TAPE READER AND PUNCH OPTIONS

The options available for paper tape facilities are listed below.

- PR8-E Reader (with Control Unit)
- PC8-E Reader/Punch (with Control Unit)

#### Type PR8-E Paper Tape Reader

The PR8-E is available in two versions: the rack mounted version (PR8-EA) and the Table Top version (PR8-EB).

The PR8-E reader senses eight-hole unoiled grey perforated paper tape photoelectrically at a maximum rate of 300 characters per second. The control unit of the PR8-E plugs into the OMNIBUS and controls the action of the reader from program instructions. All connections between the control unit and the reader are made using a BC08-K cable.

A read operation is initiated by an RFC instruction from the computer. The control unit, in turn, initiates tape movement and sensing of a character, transfers the character to its reader buffer (RB), and sets its device flag to indicate that a character is available for transfer to the computer. The computer senses the reader flag by issuing an RSF instruction, and transfers the character from the RB to AC04 through 11 by issuing an RRB instruction. The RRB instruction also clears the reader flag to ready the unit for another read operation.

The control unit also contains an interrupt enable flip-flop. This flip-flop, controlled by program instructions, determines whether the reader can generate an interrupt request to the program interrupt facility. When set by an RPE instruction or initialize input, this flip-flop enables generation of an interrupt request from the reader flag being set. When cleared by a PCE instruction, this flip-flop inhibits interrupt requests.

#### Programming

Instructions for operating the reader are as follows:

##### Set Reader/Punch Interrupt Enable (RPE)

Octal Code: 6010  
 Operation: Sets the reader/punch interrupt enable flip-flop so that an interrupt request can be generated when reader or punch flag is set.

##### Skip on Reader Flag (RSF)

Octal Code: 6011  
 Operation: Senses the reader flag; if it contains a binary one, increments the PC by one so that the next sequential instruction is skipped.

##### Read Reader Buffer (RRB)

Octal Code: 6012  
 Operation: ORs the content of the reader buffer into AC4-11 and clears the reader flag. This command does not clear the AC.

**Reader Fetch Character (RFC)**

Octal Code: 6014

Operation: Clears the reader flag, loads one character into the RB from the tape, and sets the reader flag when the RB is full.

**Read Buffer and Fetch New Character (RRB, RFC)**

Octal Code: 6016

Operation: Combines RRB and RFC. The contents of the reader buffer is ORed into the AC. The flag is immediately cleared, and a new character is read from tape into the reader buffer. The flag is then set.

**Clear Reader/Punch Interrupt Enable (PCE)**

Octal Code: 6020

Operation: Clears the reader/punch interrupt enable flip-flop so that interrupt requests cannot be generated.

A program sequence loop to read a character from perforated tape can be written as follows:

RFC	/FETCH CHARACTER FROM TAPE
LOOK,	RSF /SKIP IF READER FLAG = 1
	JMP LOOK /JUMP BACK & TEST FLAG AGAIN
CLA	/CLEAR AC
RRB	/LOAD AC FROM RB, CLEAR READER FLAG

**PC8-E Reader/Punch**

The PC8-E is available in two versions: the rack mountable version (PC8-EA) and the table top version (PC8-EB).

The PC8-E consists of a reader and punch mounted on the same chassis and a control unit which plugs into the OMNIBUS and controls the action of the reader/punch from program instructions. All connections between the control unit and reader/punch are made using two BC08-K cables.

**Specifications**

Tape Type	1-inch fan-folded unoiled grey paper
Channels	8 data channels plus feedhole
Read Character Rate (Continuous)	300 characters/second
Read Character Rate (Start-Stop Mode)	25 characters/second
Punch Character Rate	50 characters/second

The reader portion of the PC8-E operates in the same manner as the PR8-E. The punch portion executes the following additional instructions:

**Set Reader/Punch Interrupt Enable (RPE)**

Octal Code: 6010

Operation: Sets the reader/punch interrupt enable flip-flop so that an interrupt request can be generated when punch or reader flag is set.

**Clear Reader/Punch Interrupt Enable (PCE)**

Octal Code: 6020

Operation: Clears the reader/punch enable flip-flop so that interrupt requests cannot be generated.

**Skip on Punch Flag (PSF)**

Octal Code: 6021

Operation: Senses the punch flag; if it contains a binary one, increments the PC by one so that the next sequential instruction is skipped.

**Clear Punch Flag (PCF)**

Octal Code: 6022

Operation: Clears the punch flag in preparation for receiving a new character from the computer.

**Load Punch Buffer and Punch Character (PPC)**

Octal Code: 6024

Operation: Transfers the eight-bit character in AC4-11 into the PB, then punches that character. The instruction does not clear the punch flag or the PB.

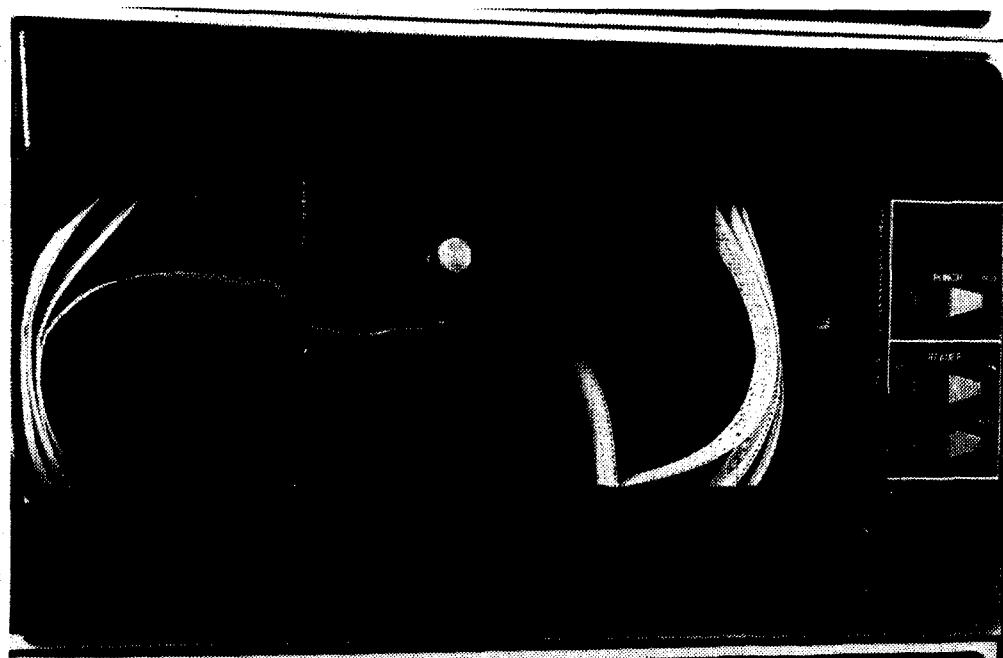
**Load Punch Buffer Sequence (PLS)**

Octal Code: 6026

Operation: Clears the punch flag, transfers the contents of AC4-11 into the punch buffer, punches the character in the PB on tape, and sets the punch flag when the operation is completed.

A program sequence loop to punch a character when the punch buffer is free can be written as follows:

```
FREE,PSF      /SKIP IF PUNCH FLAG = 1
JMP FREE     /JUMP BACK & TEST FLAG AGAIN
PLS          /CLEAR PUNCH FLAG & PB, LOAD PB
              /FROM AC, PUNCH CHARACTER, SET
              /PUNCH FLAG WHEN DONE
```

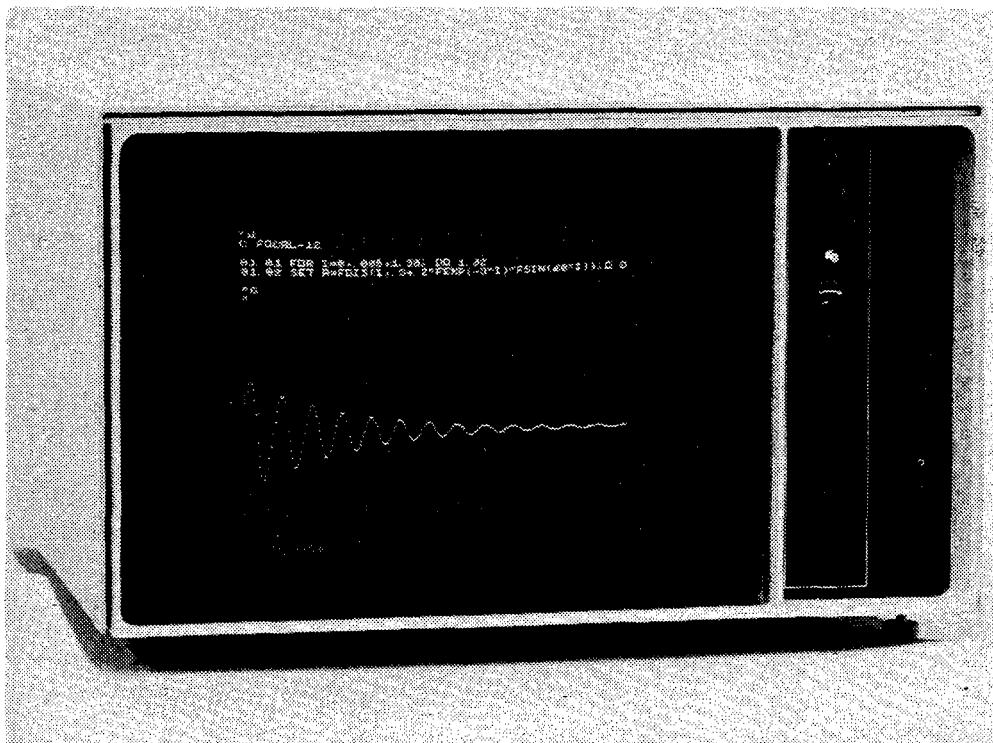


## CRT DISPLAYS

### Point Plot Display System

The VC8-E, when combined with a VR14 Oscilloscope, or a customer's scope, is capable of displaying data in the form of 1024<sub>10</sub> by 1024<sub>10</sub> dot array. Under programmed control, a bright spot may be momentarily produced at any selected point in this array. Thus a series of these intensified dots may be programmed to produce graphical output on a CRT.

Interfacing to the PDP-8/E Processor is accomplished with the VC8-E Control which plugs directly into the OMNIBUS. Information is applied from the processor's AC Register to the display by means of programmed IOT instructions. The displayed information can therefore be on line sampling or memory data or data from a mass storage device. The graphical presentation is limited only by the extent of programming the user desires to implement.



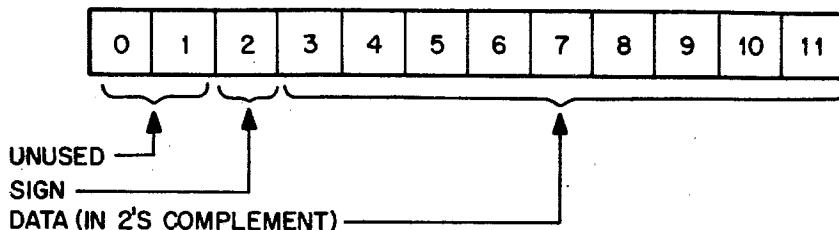
VR14 Display

### Type VR14 Oscilloscope Display

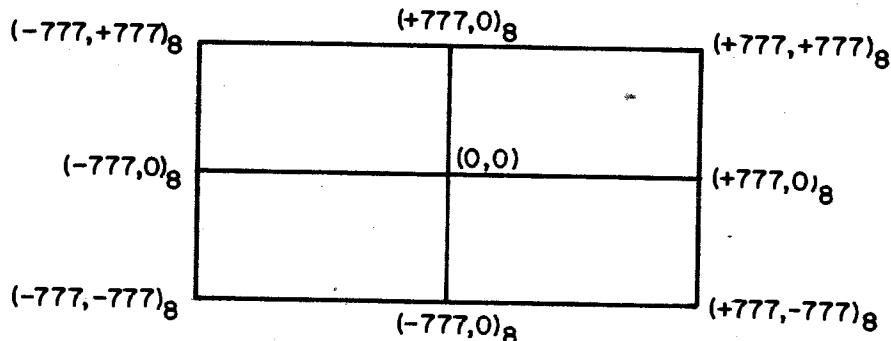
The VR14 is a compact solid-state CRT display with self-contained power supplies and a viewing area of 6 3/4 in. x 9 in. The VR14 can plot 1500 random points and up to 75 in. of vector with no flicker. X/Y deflection speed is 900 ns intensified, 700 ns non-intensified, and less than 20  $\mu$ s is required for a maximum deflection step in any direction. Interface with the VC8-E is by means of connector assembly BC01K-10 (10 feet), BC01K-25 (25 feet), or BC01K-50 (50 feet), with ten feet the standard length, included with VR14 ordered with VC-E.

### VC8-E Point Plot Display Control

The VC8-E is a two-axis (X and Y) digital-to-analog converter plus intensifying circuitry (Z axis) that provides deflection and intensity information to the display oscilloscope. Coordinate data is transferred to the X and Y axis from bits 2-11 of the PDP-8/E accumulator. This data must be the PDP-8/E accumulator. The VC8-E interfaces with Tektronix 602, 611, and 613 oscilloscopes or with the VR14. It provides programmable two-color displays and storage mode on the 611 and 613.



The position of the oscilloscope beam will be determined by the contents of the X and Y buffer registers. Coordinate (0,0) is located in the center of the screen.



The user is reminded of the relationship between the signed octal numbers used above and their corresponding 2's complement form.

Signed Values (used in example)	2's Complement (10 bit)	Position
+777	0777	Top or right
+1	0001	
0	0000	Center
-1	1777	
-777	1001	Bottom or left

## **Specifications**

The VC8-E consists of a two-axis digital-to-analog converter and intensifying circuit that provides deflection and intensity signals, which are then applied to the input amplifier circuitry of such display units as the Type VR14 oscilloscopes. The control circuit for the VC8-E is located on a PDP-8/E module (M869), which plugs into the OMNIBUS.

The basic system of the VC8-E consists of the following circuitry:

- a. OMNIBUS interface, IOT decoding, skip, clear AC, and interrupt control.
- b. X-axis buffer, D/A converter, filter and summing amplifier, and bipolar line driver.
- c. Y-axis buffer, D/A converter, filter and summing amplifier, and bipolar line driver.
- d. Z-axis control, which consists of provision for intensity signal necessary for most oscilloscopes, and intensity and channel select signals necessary for the VR14 oscilloscope.

## **NOTE on Display Times**

The display times of those instructions that include intensification depend upon the type of oscilloscope used, for example:

VR14                            21  $\mu$ s

Tektronix 602                6  $\mu$ s

A switch is provided to select the proper settling interval.

## **Programming**

The instructions for outputting data to the oscilloscope display are defined as follows:

### **Clear All Logic (DILC)**

Octal Code:      6050

Operation:       Clears enables, flags, and delays.

### **Clear Done Flag (DICD)**

Octal Code: 6051  
Operation: Clears Done Flag.

### **Skip On Done Flag (DISD)**

Octal Code: 6052  
Operation: Skip if Done Flag (1). Do not Clear Done Flag.

### **Load X Register (DILX)**

Octal Code: 6053  
Operation: Clear Done Flag; load X register, wait for settle.\* Set Done Flag. Do not clear AC.

### **Load Y Register (DILY)**

Octal Code: 6054  
Operation: Clear Done Flag; load Y register, wait for settle.\* Set Done Flag. Do not Clear AC.

### **Intensify (DIXY)**

Octal Code: 6055  
Operation: Clear Done Flag; intensify; Set Done Flag.

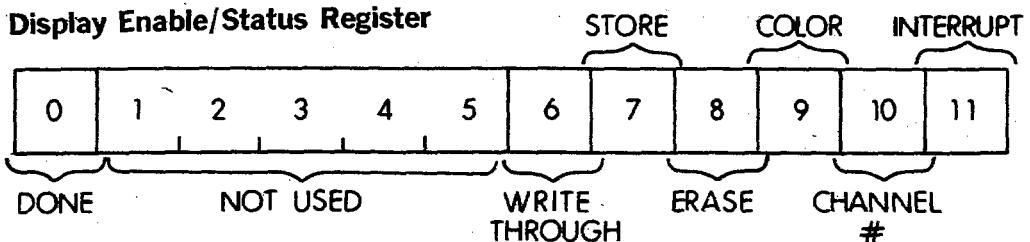
### **Load Enable (DILE)**

Octal Code: 6056  
Operation: Transfer contents of AC to Enable Register as defined below. Clears AC.

### **Read Enable/Status Register (DIRE)**

Octal Code: 6057  
Operation: Transfer the contents of the Display Enable/Status Register to the AC as defined below:

\* SEE NOTE ON DISPLAY TIMES



BITS	TITLE	DESCRIPTION
0	Done Flag	May be read using a DIRE (transfer enable to AC) command. It may not be set under program control using the DILE (load enable, clear AC) command.
6	Write Through	When set to a 1 and a 6055 intensity command is given, this will generate a small ellipsis on 611, 613 storage scope. Its purpose is to locate the writing beam on the screen in a store mode without storing the ellipsis.
7	Store	When set to a 1, this will cause the 611, 613 to go to a store mode. When set to a 0 the 611, 613 will go to a non-store mode.
8	Erase	When set to 1, this will generate an erase cycle in the 611, 613 storage scope. When doing an erase nothing can be displayed until the done flag is set. The erase cycle last 450 Ms $\pm$ 50 Ms. The erase bit is a write bit only and can not be read back using the DIRE command.
9	Color	When set to a 1 it will cause the VR20 to go to a red mode. When set to a 0 it will cause the VR20 to go to a green mode. The time required is 1600 $\mu$ s from red to green, 300 $\mu$ s green to red. The done flag is set after 1600 $\mu$ s or 300 $\mu$ s.
10	Channel Number	Channel number selects the VR14, VR20 display channel. When set to a 0 information is displayed on channel 1. When set to a 1 information is displayed on channel 2.
11	Interrupt Enable	When set to a 1 it will cause the processor to interrupt (JMS 0) on done = 1.

The Done Flag (bit 0) may be read using a DIRE (transfer enable to AC) command. It may not be set under program control using the DILE (load enable, clear AC) command.

Channel number selects the VR14 display channel. Bit 10 = 0, channel 0; Bit 10 = 1, channel 1.

Interrupt set to a one will cause the processor to interrupt (JMS 0) on done = 1.

Both channel number and interrupt may be loaded from the read into the AC using the DILE and DIRE commands respectively.

### **Programming Examples**

The VC8-E is a very fast display control. So fast, in fact, that most display oscilloscopes cannot position their beam before an intensify command is performed. For this reason a "DONE" Flag has been incorporated into the control and should be used whenever random points are plotted sequentially.

•  
•  
•

CLA	
TAD X	/GET X-COORDINATE
DILX	/LOAD X REGISTER
CLA	
TAD Y	/GET Y-COORDINATE
DILY	/LOAD Y REGISTER
DISD	/SKIP ON DISPLAY DONE FLAG
JMP .-1	
DIXY	/INTENSIFY POINT

The following example displays a dot on the screen whose coordinates are set by the position of the ADC's parameter knobs 0 and 1:

START,	CLA
JMS	SAMPLE /POSITION OF KNOB 0
DILX	/LOAD
CLA	IAC
JMS	SAMPLE /POSITION OF KNOB 1
DILY	/LOAD
DISD	
JMP .-1	
DIXY	/INTENSIFY
JMP	START
SAMPLE,	0
ADLM	
ADST	

```
JMP .-1  
ADRB  
JMP I SAMPLE
```

A fun program for the VC8-E is Kaleidoscope. Pictures on the screen are varied by manipulating the switch register bits 9, 10, and 11.

```
START, TAD Y  
JMS SCALE  
CMA  
TAD X  
DCA X  
TAD X  
DILX  
JMS SCALE  
TAD Y  
DILY  
DISD  
JMP .-1  
DIXY  
DCA Y  
JMP START
```

```
SCALE, 0  
DCA TEM  
OSR  
CIA  
DCA C  
TAD TEM  
CLL  
SPA  
CML  
RAR  
ISZ C  
JMP .-5  
JMP I SCALE
```

## **VT05 ALPHANUMERIC DISPLAY TERMINAL**

The VT05 is a flexible, high-performance alphanumeric display terminal with a video cathode ray tube display and communications equipment. It is capable of transmitting data over standard phone lines and data sets in half or full duplex modes at rates up to 300 Baud. For remote users, the VT05 serves as a non-mechanical terminal that handles data speeds many times faster than that of conventional teletypewriters. If desired, the alphanumerics can be superimposed on a background video image derived from a closed circuit TV camera or video tape player.

For user convenience, the VT05 display includes the following outstanding features:

- Completely interchangeable with Teletype (20 mil current loop)
- EIA RS-232C compatible communications interface
- Totally self-contained
- Direct cursor addressing
- Concurrent video-alphanumeric imaging
- Easy-to-read characters
- Solid-state circuitry
- Comprehensive 64/128 character set keyboard

The VT05 Alphanumeric Display Terminal can be controlled by the KL8-E, EA, EB, EC or the DC02-FB, DC02-G and BC01A-25. The same program used with the Teletype units is used with the VT05 display.

### **Specifications**

#### **DISPLAY**

Screen Size— $10\frac{1}{8}$ " x  $7\frac{5}{8}$ "

Character Display Area— $8\frac{3}{4}$ " x  $6\frac{5}{8}$ "

Characters/Line—72

Number of Lines—20

Number of Characters Displayable—1440

Contrast Ratio—12:1

Type of Phosphor—P4 (white)

Deflection Type—Magnetic

Deflection Method—Raster Scan

Character Generation Method—5 x 7 dot matrix

Character Generator—Read Only Memory (ROM)

Refresh Buffer—MOS Memory

Memory Size:

ROM—2240 bits

Refresh Buffer—9816 bits

Display Refresh Rate—60 times/sec or 50 times/sec synchronized to power line frequency

Character Set—Upper case ASCII

Character Size—.23" x .11"

Cursor—Non-destructive, blinking (underline)

#### **VIDEO**

Standard EIA-compatible signal

#### **KEYBOARD/CONTROL**

Type—Electronic (wafer switch)

Standard model Teletype layout

Character Set—Selectable (upper case, standard

ASCII; upper/lower case, full ASCII)

Controls:

<b>Cursor</b>	—Up, down, left, right, home up
	—Direct addressing, Tab
<b>Erase</b>	—To end of line, to end of frame
<b>Erase Lock</b>	—Prevents inadvertent erasure
<b>Power</b>	—On, off
<b>Mode</b>	—Remote, local
<b>Transmission</b>	—Full, half duplex

#### **MECHANICAL/ENVIRONMENTAL**

##### **Dimensions:**

**Width**—19"

**Height**—12"

**Depth**—30"

**Weight**—55 lbs.

**Heat Dissipation**—800 BTU/hr. maximum

**Operating Temperature**—40°—100°F, 4.4°—37.8°C

**Humidity**—10 to 95%

#### **POWER INPUT**

VT05-A: 95-130 VAC, 60 Hz ± 2 Hz, single phase

VT05-B: 190-260 VAC, 60 Hz ± 2 Hz

VT05-C: 95-130 VAC, 50 Hz ± 2 Hz

VT05-D 190-260 VAC, 50 Hz ± 2 Hz

**Power Consumption**—130 watts

#### **DATA TRANSMISSION**

**Type**—Crystal-controlled, selectable speed

#### **APPLICATIONS**

##### **General-Purpose Timesharing**

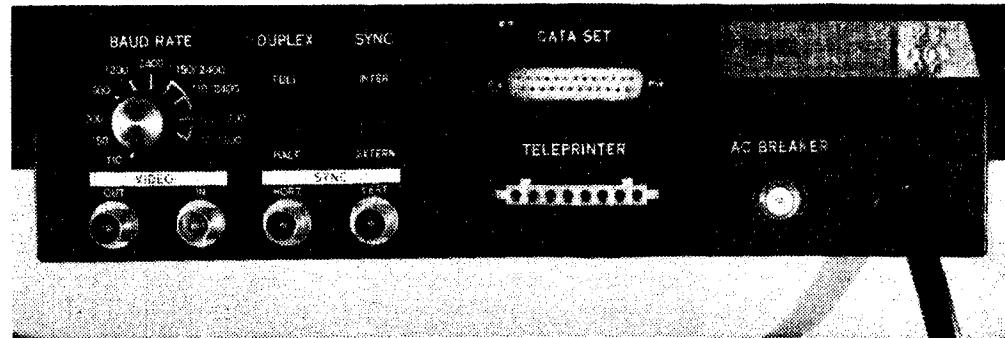
Timesharing systems are pioneering a new way of life in many scientific and technical disciplines. The time spent by professional workers at the terminal in dialog with a computer is critical productivity time. The obviously strong need for terminal equipment that increases this productivity is satisfied by the VT05 Alphanumeric Display Terminal. It is designed to make the professional's "on-line" time totally useful. Also, its selectable transmission speeds allow terminal users to utilize any available data communication system, including simple acoustical couplers and digital modems.

##### **Computer-Aided Instruction**

In the learning process, the VT05 terminal enables the simultaneous display of background video images and foreground alphanumeric information. At the elementary instruction level, foreground displays of words and numbers can be reinforced by static or dynamic pictures of the things themselves. The same technique is also appropriate for advanced levels of instruction such as medical school anatomy classes, repair mechanic training, and even photo intelligence evaluations. The background video image can be obtained directly from a TV camera or indirectly from a video tape player.

##### **Hospital Systems**

The VT05 fulfills all the necessary requirements for use in the hospital environment in multi-station paging, clinical and research applications. It is noiseless (no bothersome hum or clatter) and consequently eliminates intrusion upon the user, patients or subjects in the immediate vicinity. Also, it is extraordinarily simple to operate; no instruction manual is required, so anyone who can type can run it.



The VT05 utilizes solid-state elements, thereby guaranteeing high reliability, with correspondingly fewer maintenance problems. It is completely portable, weighing only 55 pounds, and is easily connected to a standard acoustical coupler or a data set even by an unskilled operator.

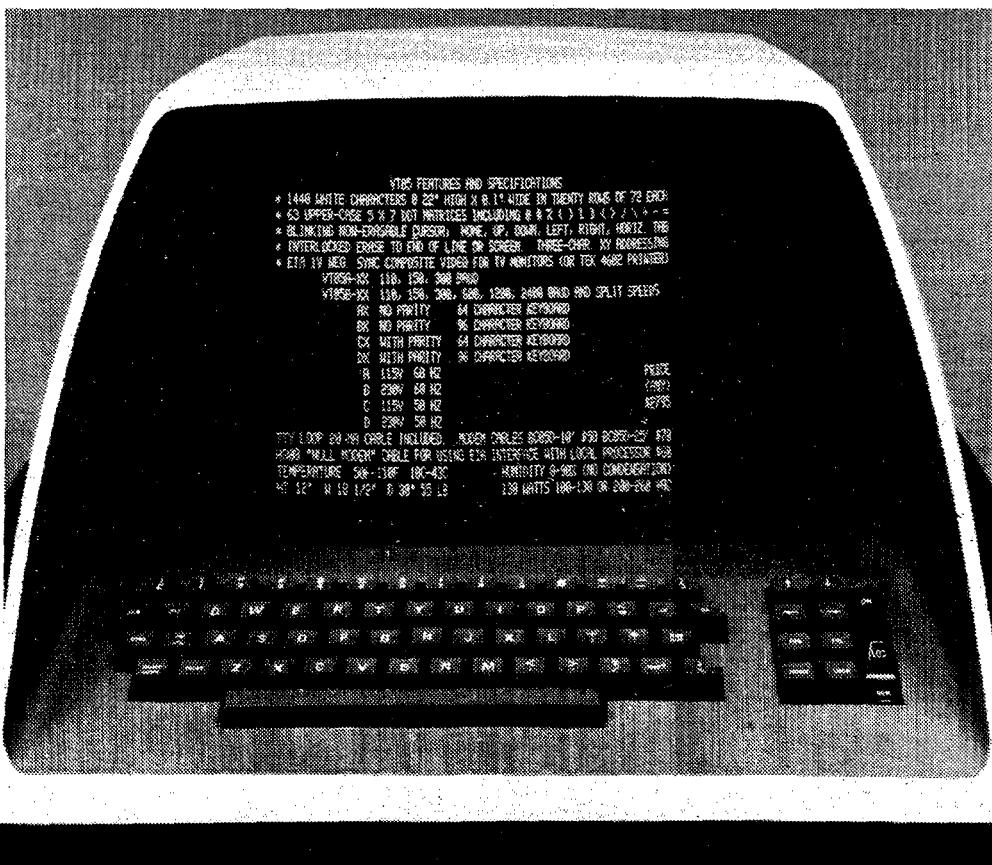
The CRT screen displays a total of 1440 characters. A keyboard-controlled cursor is operated under program control to help revise, correct or delete any character, any line or any combination. This control via the computer allows simple question-and-answer type data logging to be accomplished at remote stations by non-computer operators.

#### Industrial and Commercial

The VT05 is completely self-contained on one rugged, compact package. It includes the keyboard, CRT, refresh memory, communications interface, and power supply.

The characters displayed on the CRT are refreshed 60 (50) times per second which obviates any flicker. A tinted glass shield is provided to reduce glare and make the VT05 visually comfortable to use. The simple keyboard allows for rapid entry of data.

All of these features, plus its handsome modern design, make the VT05 an ideal clerical tool for office or laboratory. With its video capability, moreover, it can also serve as a remote monitor for hazardous experiments or production processes; e.g., working with radioactive materials, noxious fumes, or toxic substances.



## ASCII CODE ASSIGNMENTS

STANDARD TRANSMIT CODE ASSIGNMENTS

	7	0	0	0	1	1	1	1
Bit No.	6	0	0	1	1	0	0	1
	5							
4 3 2 1	0	1	0	1	0	1	0	1
0 0 0 0			SPACE	0	@	P	@	P
0 0 0 1			!	1	A	Q	A	Q
0 0 1 0			"	2	B	R	B	R
0 0 1 1			=	3	C	S	C	S
0 1 0 0			\$	4	D	T	D	T
0 1 0 1			%	5	E	U	E	U
0 1 1 0			&	6	F	V	F	V
0 1 1 1			*	7	G	W	G	W
1 0 0 0	C— (BS)	C→	(	8	H	X	H	X
1 0 0 1	HT		)	9	I	Y	I	Y
1 0 1 0	LF	C†	*	:	J	Z	J	Z
1 0 1 1	C:		—	:	K	{	K	{
1 1 0 0			.	<	L	\	L	\
1 1 0 1	CR	HOME	—	=	M	]	M	ALT.
1 1 1 0		ERASE LINE	.	>	N	^	N	^
1 1 1 1		ERASE SCREEN	/	?	O	—	O	DEL (RUB OUT)

FULL ASCII TRANSMIT CODE ASSIGNMENTS

	7	0	0	0	1	1	1	1
Bit No.	6	0	0	1	1	0	0	1
	5							
4 3 2 1	0	1	0	1	0	1	0	1
0 0 0 0			SPACE	0	@	P	@	P
0 0 0 1			!	1	A	Q	a	q
0 0 1 0			"	2	B	R	b	r
0 0 1 1			=	3	C	S	c	s
0 1 0 0			\$	4	D	T	d	t
0 1 0 1			%	5	E	U	e	u
0 1 1 0			&	6	F	V	f	v
0 1 1 1			*	7	G	W	g	w
1 0 0 0	C— (BS)	C→	(	8	H	X	h	x
1 0 0 1	HT		)	9	I	Y	i	y
1 0 1 0	LF	C†	*	:	J	Z	j	z
1 0 1 1	C:		—	:	K	{	k	{
1 1 0 0			.	<	L	\	l	\
1 1 0 1	CR	HOME	—	=	M	]	m	)
1 1 1 0		ERASE LINE	.	>	N	^	n	~
1 1 1 1		ERASE SCREEN	/	?	O	—	O	DEL (RUB OUT)

RECEIVE CODE ASSIGNMENTS

	7	0	0	0	1	1	1	1
Bit No.	6	0	0	1	1	0	0	1
	5							
4 3 2 1	0	1	0	1	0	1	0	1
0 0 0 0			SPACE	0	@	P	@	P
0 0 0 1			!	1	A	Q	A	Q
0 0 1 0			"	2	B	R	B	R
0 0 1 1			=	3	C	S	C	S
0 1 0 0			\$	4	D	T	D	T
0 1 0 1			%	5	E	U	E	U
0 1 1 0			&	6	F	V	F	V
0 1 1 1	BELL		*	7	G	W	G	W
1 0 0 0	C— (BS)	C→	(	8	H	X	H	X
1 0 0 1	HT		)	9	I	Y	I	Y
1 0 1 0	LF	C†	*	:	J	Z	J	Z
1 0 1 1	C:		—	:	K	{	K	{
1 1 0 0			.	<	L	/	L	/
1 1 0 1	CR	HOME	—	=	M	]	M	
1 1 1 0	CAD	ERASE LINE	.	>	N	^	N	^
1 1 1 1		ERASE SCREEN	/	?	O	—	O	

CURSOR ADDRESS CODE ASSIGNMENTS

	7	0	0	0	1	1	1	1
Bit No.	6	0	0	1	1	0	0	1
	5							
4 3 2 1	0	1	0	1	0	1	0	1
0 0 0 0			1	17	33	49	65	
0 0 0 1			2	18	34	50	66	
0 0 1 0			3	19	35	51	67	
0 0 1 1			4	20	36	52	68	
0 1 0 0			5	21	37	53	69	
0 1 0 1			6	22	38	54	70	
0 1 1 0			7	23	39	55	71	
0 1 1 1			8	24	40	56	72	
1 0 0 0			9	25	41	57		
1 0 0 1			10	26	42	58		
1 0 1 0			11	27	43	59		
1 0 1 1			12	28	44	60		
1 1 0 0			13	29	45	61		
1 1 0 1			14	30	46	62		
1 1 1 0			15	31	47	63		
1 1 1 1			16	32	48	64		

C = Cursor Function

## X/Y PLOTTER OPTIONS

### Type XY8/E Incremental Plotter Control

The XY8/E Incremental plotter control provides the control logic and interface necessary to operate an encoded or unencoded digital incremental plotter. It operates with a variety of plotters to display data graphically on paper or film.

Except for setting the coordinates at which plotting begins, all plotter operations are controlled by the plotter control logic and the processor. A series of functions controlled by IOT instructions initializes the plotter control logic, generates program interrupt requests to indicate change in status, and initiates a plotting operation.

The principles of operation are basically the same for all plotters. Drum plotter operations are described below:

Bidirectional rotary stepping motors are employed for both the X and Y axes. Recording is produced by movement of a pen in relation to the surface of graph paper, with each instruction resulting in an incremental step. X-axis deflection is derived from the motion of the drum; Y-axis deflection is derived from the motion of the pen carriage. Further instructions lower and raise the pen to and from the surface of the paper. Inputs to the plotter from the digital signal source consist of drum up, drum down, carriage right, carriage left, pen up, and pen down pulses. All recording (discrete points, continuous curves, or symbols) is accomplished by the incremental stepping action of the paper drum and pen carriage.

Controls on the plotters permit single-step or continuous-step manual operation of the drum and carriage, and manual control of the pen solenoid. The recorder and control are connected to the computer program interrupt and instruction skip facility.

The entire interface is contained on a PDP-8/E module which plugs into the OMNIBUS.

## Programming

The following IOT instructions are used to operate the digital incremental plotters:

### Clear Interrupt Enable (PLCE)

Octal Code: 6500

Operation: Clears the interrupt enable flip-flop.

### Skip Plotter Flag (PLSF)

Octal Code: 6501

Operation: Senses the Plotter Flag, and, if it is set, increments the contents of the PC by one so that the next sequential instruction is skipped.

### **Clear Plotter Flag (PLCF)**

Octal Code: 6502

Operation: Clears the Plotter Flag preparatory to issuing a plotter operation command.

### **Pen Up (PLPU)**

Octal Code: 6503

Operation: Raises the plotter pen from the surface of the graph paper

### **Load Direction Register, Set Flag (PLLR)**

Octal Code: 6504

Operation: Loads the direction register from AC6-11, which performs the following function:

### **Pen Down (PLPD)**

Octal Code: 6505

Operation: Lowers the pen to the surface of the graph paper

### **Clear Flag, Load Direction Register, Set Flag (PLCF, PLLR)**

Octal code: 6506

Operation: This microinstruction combines octal instructions 6502 and 6504. It clears the Plotter Flag, loads the direction register from AC6-11, then sets the flag.

### **Set Interrupt Enable (PLSE)**

Octal code: 6507

Operation: Sets the interrupt enable flip-flop.

Any sequence of programmed IOTs must assume that the pen location is known at the start of the routine, as there is no way to specify an absolute location in an incremental plotter except by the manual controls on the recorder. During a subroutine, the PDP-8/E can track the location of the pen on the paper by counting the instructions that increment the position of the pen and the drum.

#### **Type XY8-EA Digital Incremental Plotter**

The XY8-EA consists of the XY8-E interface described above and the Calcomp (California Computer Products) Model 565 Digital Incremental Plotter. The Model 565 is a high-speed, drum-type plotter, capable of performing up to 18,000 steps per minute. Each of these steps causes the drum or pen carriage to move a fixed increment in either a positive or negative direction. The size of this increment can be 0.01 inch, 0.003 inch, or 0.1 mm, depending on the gear ratios used for the drum and carriage drives.

A bidirectional roll paper feed and takeup mechanism accepts chart paper rolls 12 inches wide by 120 feet long. The paper is driven by pins on the drum which engage holes on both edges of the paper to maintain registration between the recording pen and the paper. If desired, single sheets of chart paper may be used for plotting instead of the roll paper.

#### **Type XY8-EB Digital Incremental Plotter**

The XY8-EB consists of the XY8-E interface together with the Calcomp Model 563 Digital Incremental Plotter. This is very similar to the Type XY8-EA, except that the Model 563 accepts a paper width of 30 inches.

#### **Type XY8-EH, EJ, EK Digital Incremental Flatbed Plotter**

The XY8-EH, EJ, EK plotters consists of the XY8-E interface plotter control described above and the Houston Instruments Model DP-10 Plotter.

The Digital Incremental plotter combines the low price and physical attributes of a high quality flatbed X-Y recorder, with the precision, accuracy, and stability obtainable only with incremental positioning. The X and Y pen positioning beams are driven by precision, bi-directional stepping motors which are geared to produce an increment of .005" for each input pulse. Plots up to 11" x 17" may be generated online, off-line, or remotely depending on the system configuration. Input to the plotter is standard 8 vector format, and is plug to plug compatible with existing incremental plotter controllers designed to drive continuous chart plotters such as the XY8-E.

### **SPECIFICATIONS**

#### **Input Requirements:**

- Positive or Negative going pulse greater than 10 volt amplitude, less than  $10\mu$  seconds rise time with greater than  $4\mu$  seconds duration. SK—19—32—SL connector.
- Maximum Pulse rate—200 increment commands/second (1/3.33 ms).
- Pen Up/Down stabilizing time—60 ms, Down; 10 ms, Up.

- Input Functions—(+X), (−X), (+Y), (−Y), PEN UP, PEN DOWN. Normally, eight plotting vectors are obtained by appropriate combination of the basic directions.
- 3 volt inputs available for compatibility with DTL or TTL logic.

#### **Step Size and Speed**

- .005" Increment
- 1.5 IPS (0°, 90°,  
180°, 270°)
- 2.12 IPS (45°, 135°,  
225°, 315°)

#### **Physical Dimensions and Mounting (Vertical Mount)**

- Depth—6½ inches
- Width—17-5/8 inches (19" with rack mounting)
- Height—15-¾ inches

#### **Pens and Chart Hold Down**

- Supplied with ball point and fibre tip disposable pens.
- Chart held down by vacuum system—capable of handling either 8½" x 11" or 11" x 17" charts.

#### **CONTROLS**

- "POWER" "ON/OFF"—Toggle Switch
- Manual Pen Position—Three-position Toggle switches with center "off" position. One switch positions pen in the (+) or (−) X direction; the other positions the pen in (+) or (−) Y. A single step in the respective direction will result if the switch is momentarily actuated. If the switch remains actuated for more than approximately 1 second, a continuous movement will occur at a nominal speed of 1 inch/second, until the switch is released. These positioning switches are not functional when the "Pen" is in the "Remote" position.
- "LOAD/PLOT" Toggle Switch—spring loaded to the "PLOT" mode. Whenever the "LOAD" position is momentarily selected, the pen will automatically be positioned to the lower left corner of the plotter for the dual purpose of (1) establishing an X and Y reference zero point, and (2) locating the pen beam so that a new chart may be loaded without interference.
- "PEN" "REMOTE"/"UP"/"DOWN"—3-position toggle switch—In "REMOTE" position, the pen will be under program control and will remain latched in whatever state that was last selected

by the program. "UP" and "DOWN" will allow the operator to raise or lower the pen manually, irrespective of the program selected state of the pen.

### **POWER REQUIREMENTS:**

Can be connected for either 115 or 230 VAC, 50/60 Hz. Connected for 115 VAC unless suffix "J."

Maximum Apparent Power is 120 Volt/Ampères with Pen Down and No Pen Motion. Line Voltage tolerance is plus or minus 10% from nominal input requirements.

### **CONFIGURATIONS**

Type	Type Number
115VAC input power	XY8-EH
230VAC input power	XY8-EJ
Table Top version, 115VAC	XY8-EK

### **LINE PRINTER OPTION**

#### **LE8 Line Printer**

The LE-8 line printer offers the user a low-cost, high-speed, flexible method of printing computer output at a rate dependent upon the option selected. It accepts ASCII characters from the AC.

Each character is selected from the set of 64 (or 96) available by means of six-bit or seven-bit binary code. (Appendix E lists the ASCII code for each character.) Each code is loaded separately from the computer into a 20-character (or, in the case of the 132-column model, 22-character) core storage Line Printer Buffer from AC 6-11 (or AC 5-11), with the least significant bit appearing in AC11. After each code is transferred into the Line Printer Buffer, the Line Printer Done Flag appears, indicating that the printer is ready to receive the next character. When the Line Printer Buffer is filled, or a control character has been received, the print cycle is initiated. Character codes not in the character set in Appendix E are printed as spaces. The line feed command and carriage return command are similar to the corresponding commands in the Teletype. The form feed command advances the paper to the top of the page. The Printer Done Flag is set after each of these operations.

During the print cycle, the paper and inked ribbon pass between a row of 80 hammers (132 in the 132-column model) and the continuously rotating drum that contains all of the available characters. Variable reluctance pickoffs scan the stored characters in synchronism with the rotating characters, and the control system actuates the appropriate hammer as the desired character approaches the print position. The full line is printed in 20-column segments, with one drum revolution required for each segment. After the last character of a line is printed, the Line Printer Buffer is cleared automatically.

There are no operator controls in the control module. The following controls are on the printer:

**TOP OF FORM**—Advances paper to top-of-form position; disabled in on-line mode

**PAPER STEP**—Advances paper one line; disabled in on-line mode

**ON LINE/OFF LINE**—Selects mode of operation for the printer

**MASTER CLEAR**—Initializes printer to ensure proper state of electronic elements

**PRINT INHIBIT**—Inhibits print hammers.

The line printer is available in any of the following combinations:

LE8-FA	80 columns	64 characters	60 Hz
LE8-FB	80 columns	64 characters	50 Hz
LE8-HA	80 columns	96 characters	60 Hz
LE8-HB	80 columns	96 characters	50 Hz
LE8-JA	132 columns	64 characters	60 Hz
LE8-JB	132 columns	64 characters	50 Hz
LE8-KA	132 columns	96 characters	60 Hz
LE8-KB	132 columns	96 characters	50 Hz

The interface is contained on one PDP-8/E module, which plugs into the OMNIBUS.

The specifications for the LE8 line printer are as follows:

**Printable characters**

character set	64 or 96
type	Open Gothic print
size	Typically 0.095 inches high and 0.065 inches wide
code format	ASCII
characters per line	80 or 132
drum speed	1760 rpm (64 character drum)

**Print rate**

80 column model	
64 character	356 Lines/minute, columns 1-80 460 Lines/minute, columns 1-60 650 Lines/minute, columns 1-40 1110 Lines/minute, columns 1-20
96 character	253 Lines/minute, columns 1-80 330 Lines/minute, columns 1-60 478 Lines/minute, columns 1-40 843 Lines/minute, columns 1-20

**132 column model**

64 character	245 Lines/minute, columns 1-132 290 Lines/minute, columns 1-110 356 Lines/minute, columns 1-88 460 Lines/minute, columns 1-66 650 Lines/minute, columns 1-44 1110 Lines/minute, columns 1-22
--------------	---

96 character	173 Lines/minute, columns 1-132 205 Lines/minute, columns 1-110 253 Lines/minute, columns 1-88 330 Lines/minute, columns 1-66 478 Lines/minute, columns 1-44 843 Lines/minute, columns 1-22
Format	Top-of-form control, single line advance and perforation step over.
Paper feed	One pair of pin-feed tractors for $\frac{1}{2}$ -inch hole center, edge-punched paper. Adjustable for any paper width from 4 inches to $9\frac{7}{8}$ inches on the 80-column model; or a maximum width of $14\frac{7}{8}$ inches for the 132 column model.
Paper slew speed	13 inches per second
Print area	8 or 13.2 inches wide, left justified
Character Spacing	10 characters per inch
Line spacing	6 lines per inch for 80-column, 6 or 8 lines for 132-column printer
Line advance time	20 milliseconds
Character synchronization	Variable reluctance pick-offs sense drum position
Printer Dimensions	
Height	80 column                          132 column 46 inches                            46 inches
Width	24 inches                            48 inches
Depth	22-inches                            25-inches
Weight	275 pounds                         420 pounds
Printer Power Requirements	
	115 vac + or - 10%, 60 Hz + or - 3 Hz, single phase, 300 watts or 240 vac + or - 10% 50 Hz + or - 3 Hz, single phase, 300 watts
Signal cable	25 foot interconnecting signal cable is supplied with system
Paper	
Type	standard fanfold, edge punched
Dimensions	4 inches to $9\frac{7}{8}$ inches wide (80 column)  4 inches to $14\frac{7}{8}$ inches wide (132 column) with 11 inches between folds

weight	
(single copy)	15 pound bond (minimum)
multi copy)	12 pound bond with single-shot carbon for up to six parts
Ribbon	
type	inked roll
width	9 inches (80 column); 14 inches (132 column)

### **Programming**

The IOT instructions which command the line printer are:

#### **Skip on Character Flag (PSKF)**

Octal Code: 6661

Operation: Senses the content of the line printer done flag; if it contains a binary 1, the contents of the PC are incremented by one so that the next sequential instruction is skipped.

#### **Clear the Character Flag (PCLF)**

Octal Code: 6662

Operation: Clears the Line Printer Done Flag.

#### **Skip on Error (PSKE)**

Octal Code: 6663

Operation: Senses the content of the Line Printer Error Flag; if it contains a binary 1, indicating that an error (drum gate open, out of paper, excessive temperature) has been detected, the contents of the PC are incremented by one so that the next sequential instruction is skipped.

#### **Load Printer Buffer, Print on Full Buffer or Control Character (PSTB)**

Octal Code: 6664

Operation: Loads the character into the print buffer, and prints if the buffer is full, or if the character was a control instruction. This instruction does not clear the AC.

#### **Set Program Interrupt Enable Flag (PSIE)**

Octal Code: 6665

Operation: Sets the interrupt enable (IE) flip-flop to a one, permitting the Printer Done Flag to request a program interrupt.

#### **Clear Line Printer Flag, Load Character, and Print (PCLF, PSTB)**

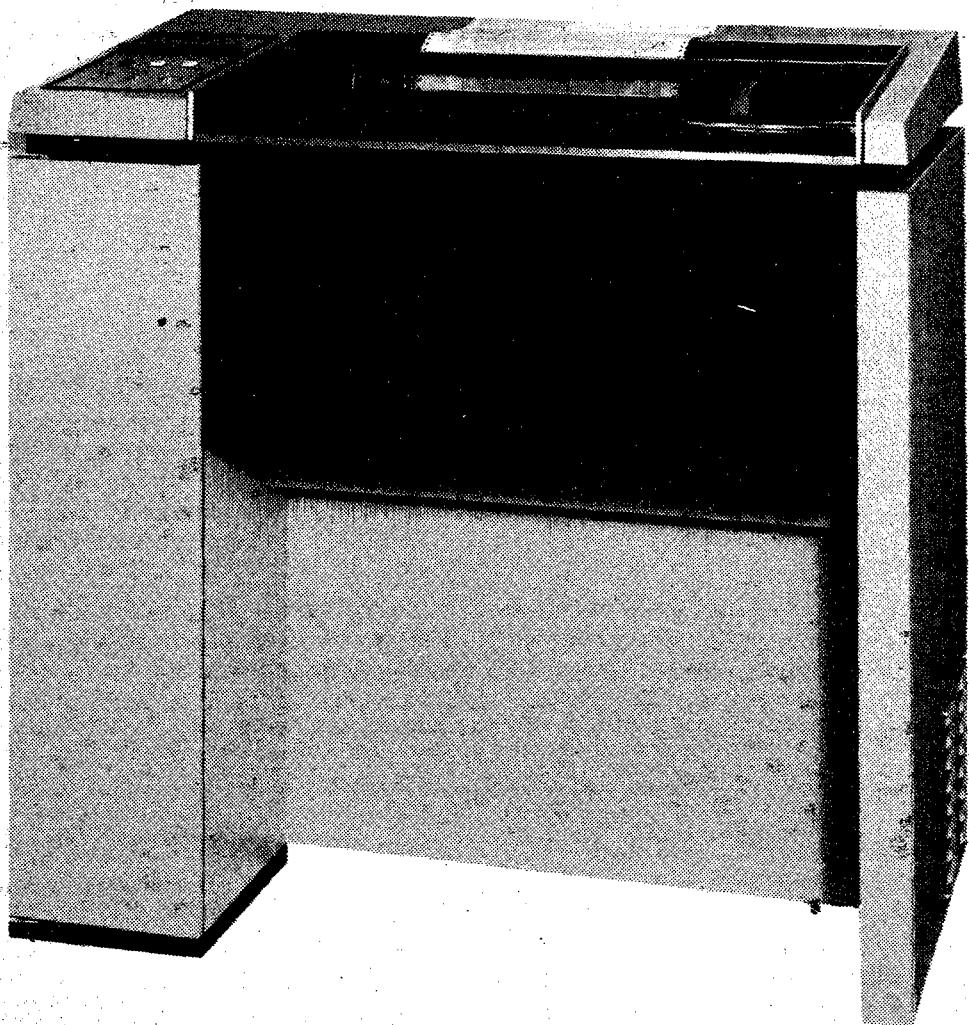
Octal Code: 6666

Operation: This is a microprogram combination of PCLF and PSTB.

#### **Clear Program Interrupt Flag (PCIE)**

Octal Code: 6667

Operation: Clears the interrupt enable flip-flop.



**LE-8 Line Printer 132—Column Model**

#### **DATA COMMUNICATIONS EQUIPMENT OPTIONS**

##### **DC08H Automatic Calling Unit Controller**

Digital Equipment Corporation offers several data communications products that have the ability to control Bell System Automatic Calling Units or equivalent. Among these are the DC08H, a general-purpose PDP-8 unit designed to be used with any of several data communications systems. It interfaces with up to ten Bell System 801A (Dial-Pulse) or 801C (Touch-Tone®) Automatic Calling Units. The interface conforms to the Electronic Industries Association (EIA) Standard RS-232-B.

The Bell System 801 Automatic Calling Units will dial, under the control of a DC08H or similar unit, a local or remote terminal (possibly another computer) for the purpose of establishing a data path between that terminal and the modem associated with the 801 Automatic Calling Unit (801 ACU).

The following leads go from the 801 ACU to the DC08H controller: Power Indicator (PWI), Data Line Occupied (DLO), Abandon Call and Retry (ACR), Data Set Status (DSS), and Present Next Digit (PND).

The PWI lead indicates that the 801 is receiving power. The DLO lead indicates that the line associated with the 801 is in use.

The ACR lead indicates that either too much time was taken to dial, or else a long time has passed and the distant terminal was not answered. In general, it is an indication that something has gone wrong and that the DC08H should tell the 801 to hang up and try again. Observe that this is an unusual lead in that the 801 is telling the DC08H that the DC08H should tell the 801 to hang up. The decision to hang up must be made by the DC08H. The PND lead is a request from the 801 to the DC08H to present the next digit to the 801 to be dialed. The data set status lead tells the DC08H that the modem associated with the 801 has been placed on the line and in data mode.

The DC08H provides the following leads to the 801: Digit Present (DPR), Call Request (CRQ), and four Digit-Leads labelled NB1 through NB4. The Digit Present lead tells the 801 that it can read the status of the leads NB1, NB2, NB3, and NB4 and obtain the binary representation of the number that should be dialed next. The Call Request Lead is a request from the DC08H to the 801 telling it to connect itself with the telephone line and go off hook. The four-digit leads NB1, NB2, NB3, and NB4 contain a binary representation of the digit to be dialed. These leads are controlled by the DC08H under program control. Lead NB1 has a weight of 1, lead NB2 has a weight of 2, lead NB3 has a weight of 4, and lead NB4 has a weight of 8.

If the PWI lead is ON and the DLO lead is OFF (i.e., the 801 is powered and the line is not in use) a READY signal is generated in the DC08H. This permits the issuance of an RTC instruction to assert the Call Request Lead to the 801. The 801 will seize the line and, upon receipt of dial tone, will assert the Present Next Digit lead, thus requesting the DC08H to tell it the first digit to be dialed. Since the line is now in use, the Data Line Occupied line will be asserted.

Under program control, the DC08H will present the next digit on the Digit Leads and will tell the 801 by asserting the Digit Present Lead. The 801, seeing Digit Present true, will read the Digit Leads NB1 through NB4, and will lower the Present Next Digit Lead. In response to this, the hardware lowers the Digit Present lead. When the 801 has finished dialing the first digit, it will again raise the PND lead, the DC08H will again present digit data on the digit leads, and assert Digit Present, etc. This sequence repeats for each digit dialed, including the last digit.

When the last digit has been dialed, there are two actions that may occur, depending upon the choice of a wired option in the 801. If the distant modem involved in the call is not a type that responds with an answer tone, such as is done by the Bell System 100 Series and similar modems, the End-of-Number code mode of operation must be used. In this case, a special code mode of operation must be used: a special code EON, (octal value 14) indicating End-of-Number, is sent after the last digit dialed. This causes the 801 to connect the modem to the line immediately, and the modem and its associated controller must determine when the called station has answered and is sending data. A second option, which should be ordered if the aforementioned option is ordered, is called Do Not Stop ACR When DSS Goes ON. This latter option provides a convenient interval timer, so that the program can check after a prescribed interval to determine whether or not data is being received by the associated modem.

If Bell System Series 100 modems, or similar modems using the exchange of "handshaking" signals at the beginning of a call are used, the Detect Answer options may be used instead of the End-of-Number option. In the Detect Answer mode, the 801 retains control of the line and looks for an answering tone from the called station. When that tone is detected, the modem is connected to the line, and Data Set Status is asserted. This will stop the Abandon Call and Retry timer. Observe that in the End-of-Number mode, the connection of the modem to the line is immediate, so Data Set Status comes on before any answer has been received from the called station. In this case it is desirable to leave the Abandon Call timer running, so the program can be flagged at some subsequent time to check for the receipt of an answer. In the Detect Answer mode, Data Set Status comes on only after an answer has been received, so it is generally desirable to stop the timer. Should an answer tone not be received because of an incorrect number, the calling of a modem whose Data Terminal Ready lead is not asserted, or the reaching of a busy signal, the Abandon Call and Retry timer will generate a signal to the DC08H which, in turn, will signal the program. The program should then initiate action to cause the DC08H to drop or retry the call.

The remaining option is whether the call should be terminated under modem control or under DC08H control. The first of these options is called Terminate Call After DSS Goes ON via Data Set. When this option is ordered, control of the call is transferred to the modem as soon as it is placed on the line and DSS has gone TRUE. It is necessary that this option be selected when a DC08H unit is used, because this option, by relieving the DC08H of responsibility for a call in progress, permits the DC08H to utilize its control facilities in the setting up of other calls. The other option, called Terminate Call After DSS Goes On, via CRQ, requires that the DC08H remain permanently associated with a call which is in progress, so that the CRQ lead is held true. Therefore, the termination of calls by modem control is a desirable option for the most efficient operation of the DC08H.

### **DP8-EA and DP8-EB Synchronous Modem Interface**

The DP8-EA and DP8-EB interface the PDP-8/E with a full-duplex or half-duplex synchronous modem for computer-to-terminal or intercomputer transfer of data. The PDP-8/E is capable of interfacing up to four communication links (channels), each having its own unique program instructions, modem interface, baud rate, priority assignment and access addresses. Data is exchanged between the PDP-8/E and the DP8-EA/EB in parallel form, using the data break facility of the computer. Data is exchanged between the DP8-EA/EB and a modem in serial form. Thus, the DP8-EA/EB performs parallel-to-serial conversion for transmit functions and serial-to-parallel conversion for receive functions. The interface also provides level conversion, character detection, modem control, and program-controlled interface with the computer.

Data exchange between the PDP-8/E and DP8-EA/EB interface is accomplished via the data break facility to or from any location within 32K of memory. Word count (WC), Current Address (CA) and character detection are performed using additional data break cycles to a specified set of locations in field zero. The DP8-E interface for one communication link consists primarily of MSI logic packaged on two PDP-8/E modules, which plug into the OMNIBUS. A cable provided with the modules mates with a connector on the modem. Two types of interface units (designated DP8-EA and DP8-EB) are available. A DP8-EA interface operates with a Bell System 200-Series Modem or equivalent, and DP8-EB operates with a Bell System 300-Series Modem or equivalent connector.

#### **Specifications**

##### **Data Transfers**

Transfers are maintained via three single cycle data breaks (1.4 micro seconds each cycle) for both transmit and receive. An additional cycle is required for each special character to be tested for (receive circuits only).

##### **Transfer Mode**

Modem—Full—or half-duplex (serial data)  
Computer—Multi-cycle data break (parallel data)

##### **Modem Interface**

Jumper-selectable for:  
1) Bipolar EIA/CCITT (RS232-C)  
2) Current Mode; where MARK = 5 ma or less and SPACE = 23 ma or greater  
3) TTL compatible

##### **Baud Rate**

71,000 bits/second (max)

##### **Character Length**

Jumper selectable for 6, 7, or 8 bits

##### **Response Time**

Break cycles: 1/Baud rate

Program Interrupts: 1/Baud rate \* bits/character (one character time)

##### **Character Recognition**

Detects four program selected characters.  
Flag bit (Bit 0) stored with character determines whether program is flagged or character is stripped.

##### **Cycle Time**

Single Cycle Data Break—1.4 Micro Seconds  
All Instructions—1.2 micro seconds.

<b>Carrier Detect</b>	Jumper selection detects carrier/AGC ON and/or OFF transitions.																				
<b>Control Transfers</b>	Control transfers are maintained via the Data Bus. The types of control available are: Idle, Terminal Ready, Enable, Transmit Request and Transfer Field.																				
<b>Synchronization Character</b>	Transmit: Non-hardware function; part of data for transmission. Receive: Receive sync code is jumper-selectable. Two or more consecutive sync characters must be detected before hardware is activated.																				
<b>Clock</b>	Modem timing or tabs for customer-supplied clock																				
<b>Modem Compatibility (Typical)</b>	<table border="1"> <thead> <tr> <th>Type</th><th>Speed (Baud)</th></tr> </thead> <tbody> <tr> <td>Bell 201A</td><td>2000</td></tr> <tr> <td>" 203A</td><td>2400</td></tr> <tr> <td>" 205B</td><td>600,1200 or 1800</td></tr> <tr> <td>" 301B</td><td>40,800</td></tr> <tr> <td>" 303B,C</td><td>19,000 to 50,000</td></tr> <tr> <td>Rixon FM-12</td><td>1200</td></tr> <tr> <td>" Sebit 48</td><td>4800</td></tr> <tr> <td>G.E. TDM Series</td><td>2400</td></tr> <tr> <td>Lenkurt 26C</td><td>120-2400</td></tr> </tbody> </table>	Type	Speed (Baud)	Bell 201A	2000	" 203A	2400	" 205B	600,1200 or 1800	" 301B	40,800	" 303B,C	19,000 to 50,000	Rixon FM-12	1200	" Sebit 48	4800	G.E. TDM Series	2400	Lenkurt 26C	120-2400
Type	Speed (Baud)																				
Bell 201A	2000																				
" 203A	2400																				
" 205B	600,1200 or 1800																				
" 301B	40,800																				
" 303B,C	19,000 to 50,000																				
Rixon FM-12	1200																				
" Sebit 48	4800																				
G.E. TDM Series	2400																				
Lenkurt 26C	120-2400																				
<b>Additional Features</b>																					
Break Priority	Jumper selectable for priorities 0 through 6.																				
Device Codes	Jumper selectable for using up to four DP8-E modules.																				
Access Address	Jumper selectable for up to six groups corresponding to DP8-E assignment (up to four active and two spares). Each group or interface module can have up to 16 access or file addresses.																				

#### **Current Mode Electrical Specifications (Applicable to the Bell 300 Series Modem or equivalent)**

**Receiver Input Current/Voltage levels with 100 ohms Termination**

Mark—5 ma ( $-0.7 < E_o < .1$ )

Driver Output Impedance with Power Off: Not Specified

Driver Output Short Circuit Current: Not Specified

Driver Slew Rate between the 7 ma and the 21 ma levels

Typical 14 ma/100 ns  
Max. 14 ma/ 50 ns  
Min. 14 ma/200 ns

**Receiver Input Impedance**

$120 > Z_{in} > 90$

**Receiver Output with Open Circuit Input**

Logic one—Mark—off

**Receiver Output with Input  $> 23$  ma**

Logic Zero—Space—On

**Receiver Output with Input  $< 5$  ma**

Logic ONE—Mark-off

## Driver Distortion Limits

Mark to Space or Space to Mark must be achieved within 25% of bit interval.

## Receiver Open Circuit Voltage

-0.8V to -1.3V

## RS-232-C Electrical Specifications

Driver output logic levels with 3K to 7K load

15 volts >  $V_{oh}$  > 5V

-5 volts >  $V_{ol}$  > -15V

Driver output voltage with open circuit

$V_o$  < 25 volts

Driver output impedance with power off

$Z_o$  > 300 ohms

Output short circuit current

$I_o$  < 5 amps

Driver slew rate

$\frac{dv}{dt}$  < 30 volts/usec.

Receiver input impedance

7K ohms >  $R_{in}$  > 3K ohms

Receiver input voltage

±15V compatible w/driver

Receiver output with open circuit input

Mark

Receiver output with 300 ohms to ground on input

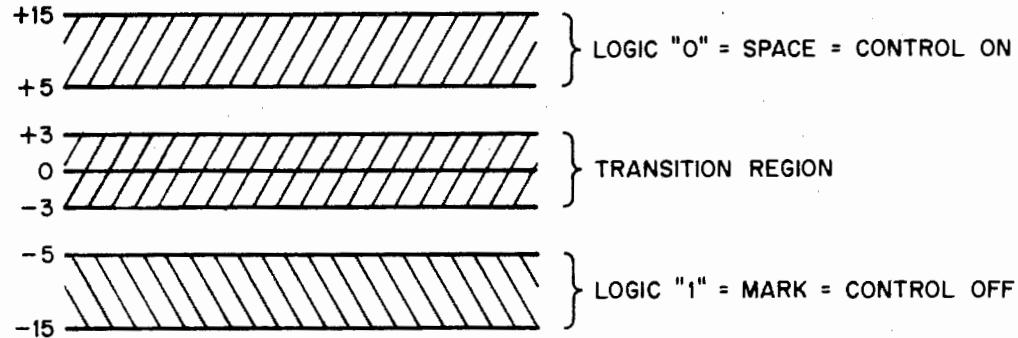
Mark

Receiver output with +3 volt input

Space

Receiver output with -3 volt input

Mark



## **Programming**

The IOT instructions which follow control the DP8-EA/EB. For multiple-channel interfacing, the octal codes listed are used for Channel 1; IOTs containing device codes 42 and 43 are as used for Channel 2, etc., as follows:

Channel	Access Addresses (9 per channel)	IOT Device Codes
1	7720-7730	640x/641x
2	7700-7710	642x/643x
3	7660-7670	644x/645x
4	7640-7650	646x/647x
	*7620-7630	
	*7600-7610	

\* These spare access addresses may be used in case of conflict with existing programs.

Access address assignments are determined by low order bits (8-11) as follows:

0000	
0001	
0010	
0011	
0100	Test Characters
0101	Receive Word Count (WC) [2's Complement of Number of Words to be received]
0101	Receive Current Address (CA)
0111	Transmit Word Count (WC)
1000	Transmit Current Address (CA) [2's Complement of Number of Words to be transmitted]
1001	*
0110	

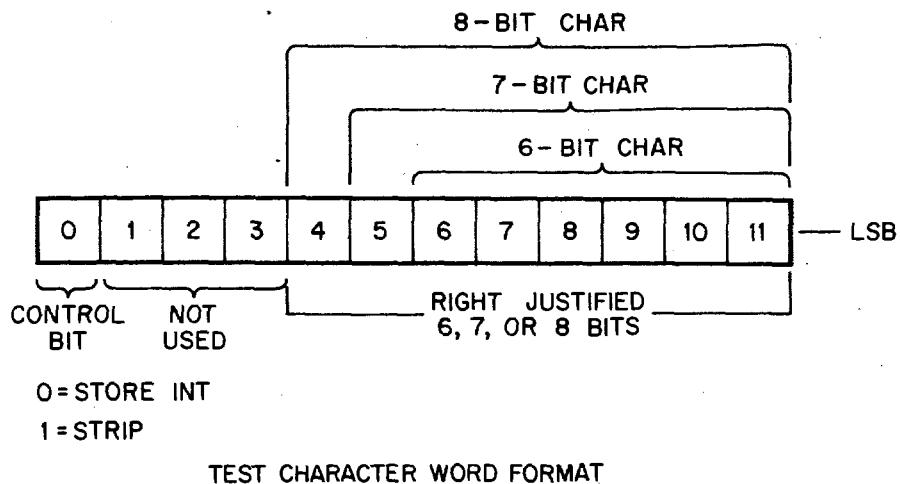
\* Access address counter increments to these locations prior to character transfer with the PDP. When the counter is set at 0110, a Received Character has been transferred to a location specified by the Receive Current Address. When the counter is set at 1001, a character for transmit has been transferred to the DP8 from the location specified by the Transmit Current Address.

### **Test Characters**

A Test Character is a vehicle by which the programmer is provided greater control and flexibility over the input/output of data. Four test character locations are available as indicated in the access address assignments.

Test Characters allow the user to identify interesting characters by causing the DP8-E character detected flag to set and thereby cause an interrupt. This can be used with the SRCD instruction which asserts the SKIP line when the Character Detected Flag is set.

The format of the test character is given below. Bit 0 is the control bit that determines if the test character is to be stored or stripped. The least significant bit is bit 11 and the most significant bit is bit 4.



**Word Count and Current Address Bits**—Word Count and current address are 12 bits wide.

**Transmit and Receive Data**—the 6, 7, or 8 bit character is right justified. When 6 or 7 bit characters are used, the remaining bits up to 8 should be negated.

**Control Word**—The AC bits vs. Control are as follows:

AC00 Terminal Rdy

AC01 Idle (1)

AC02 Enable (2)

AC03 Send Rqst

AC04 For customer use (Write only TTL output)

AC05 For customer use (Write only TTL output)

1. If word count goes to zero while in IDLE mode, the Transmit Current address and Word Count will no longer be incremented and access to the last address will continue until the instruction SSTO (Skip on transmit word count o'flow) is assigned or the Idle Bit is negated.
2. If Enable is negated, Interrupt Request, Break in progress and Break Priority Gates are inhibited and the Break Request Flip Flop is latched in the ZERO state.

**Character Recognition**—Character recognition (detection) is accomplished for 6, 7, or 8 bit characters. The characters must be stored right justified. When 6 or 7 bit characters are used, the remaining bits, up to 8, should be negated.

The stripping or flag generation upon character detection is dependent upon MD00. If MD00 is set to a ONE and the stored character is found to compare with the received character, further memory cycles will be terminated (i.e. the word count and current address will not be incremented and there will be no stored character). If MD00 is a ZERO and there is a character comparison, the character detected flag will be raised, the number of the recognized character will be stored for one character time in a two bit register, and the received character will be transferred to the current address.

**Field Selection**—The selected field (increments of 4K of core up to 32K) combined with the current address forms a 15 bit address for transfers to and from core.

The field for character transfer is specified by program instruction (SLFL) and the contents of the AC. The field vs. AC assignments are as follows:

AC00	}	
AC01	}	Transmit field (octal 0-7)
AC02	}	
AC03	}	
AC04	}	Receive field (octal 0-7)
AC05		

**Character Detected (Reading of)**—When the instruction "Read Character Detected (SRCD)" is used to determine what character was detected, two bits, corresponding to the two low-order bits of the Access Address are transferred to AC10 and AC11 as follows:

Access Address		
AC10	AC11	(Base 2)
0	0	0000
0	1	0001
1	0	0010
1	1	0011

### Instructions

All instructions are fully decoded and two device codes are required for an instruction set. Up to four sets of instructions are available and are paired as follows:

640X/641X, 642X/643X, 644X/645X, 646X/647X.

### Transmit Go (SGTT)

Octal Code: 6405/6425  
6445/6465

### Operation:

SGTT sets the Transmit Go Flip Flop. This instruction implies that the program is ready to transmit data (i.e., the Current Address (CA) and Word Count (WC)), have been updated. Upon receipt of this instruction, the hardware will assert the modem Request to Send (RS) lead. When the modem responds with Clear to Send (CS), memory references will begin. Memory references will cease only when WC decrements to Zero ( $WC \rightarrow 0$ ). In this event if SGTT is not issued in less than one character time, the transmit line will be maintained at Mark Hold. Transmit Request should be asserted SGTT instruction and should not be cleared until two bit times after the last bit has been transmitted.

**Receive Go (SGRR)**

Octal Code: 6404/6424  
6444/6465

Operation: SGRR sets the Receive Go Flip Flop. This instruction implies that the program is ready to receive data from the communications line, (i.e. the Current Address (CA) and Word Count (WC),) have been updated. The hardware, upon receipt of this instruction, will begin memory references if two consecutive synchronizing characters have been recognized by the hardware on the incoming serial data line. Memory references will cease only when WC decrements to Zero (WC → 0) and SGRR is not issued in less than one character time.

**Skip if Character Detected (SSCD)**

Octal Code: 6400/6420  
6440/6460

Operation: The SSCD Instruction causes the program to skip the next instruction if the character detect flag is a ONE. The character detect flag is a ONE if an assembled character is found to compare one of the stored characters in one of the first four locations of the Access Address. Additionally, the SSCD Instruction clears the character detected flag. If the program is required to identify which of the four stored characters compared to the contents of the Receive Buffer, then a Read Character detected (SRCD Instruction should be utilized. See the SRCD instruction for details).

**Clear Sync Detect (SCSD)**

Octal Code: 6406/6426  
6446/6466

Operation: Clears the "Sync Character Detection" Flip Flops. This instruction enables the programmer to initialize the sync detection circuits and clear the receive registers without initializing the modem interface.

**Skip if Receive Word Count Overflow (SSRO)**

Octal Code: 6402/6424  
6444/6464

Operation: Skips the next instruction and clears the flag if the Receive O'Flow Flag is a ONE. The receive O'Flow Flag is a ONE if during the Receive Data break sequence the Word Count (in core) overflowed.

**Skip if Transmit Word Count Overflow (SSTO)**

Octal Code: 6403/6423  
6443/6463

Operation: Skips the next instruction and clears the flag if the Transmit O'Flow Flag is a ONE. The Transmit O'Flow Flag is a ONE if during the Transmit Data Break sequence the Word Count (in core) overflowed.

**Clear Synchronous Interface (SCSI)**

Octal Code: 6401/6421  
6441/6461

Operation: Initializes all active functions in the synchronous interface.

**Read Transfer Address Register (SRTA)**

Octal Code: 6407/6427  
6447/6467

Operation: Transfers the contents of the transfer address register to AC00-AC11. In use, the Transfer latches the Current Address (CA) prior to incrementing and returning it to core. During Data transfers (transmit or receive) this register then becomes the 8/E's memory address (MA). This instruction is primarily for diagnostic and/or program debug.

**Load Control (SLCC)**

Octal Code: 6412/6432  
6452/6472

Operation: Transfers the contents of AC00-AC05 for selecting Terminal Ready, Idle Mode and Synchronous Interface Enable respectively.

(AC00) *Terminal Ready* permits the modem to enter into the data mode.

(AC01) *Idle Mode* allows a continuous transmission from the same location in core without program intervention. The hardware will enter the Idle Mode when the Word Count goes to ZERO. Further, the transmit current address and Word Count will no longer be incremented and access to the last address will continue until the SGTT Instruction is issued or the Idle Bit is negated.

(AC02) *Interface Enable* allows program interrupts and data break cycles.

(AC03) *Transmit Request* activates the Request to Send line. See SGTT instruction.

(AC04, AC05) are for customer use. When modem timing signals are used one EIA (or current mode) transmitter is available to be used with AC04 or AC05.

**Skip if Ring Flag (SSRG)**

Octal Code: 6410/6430  
6450/6470

Operation: Skips the next instruction and clears the Ring Flag if the Ring Flag is a ONE.

**Skip if Carrier/AGC Flag (SSCA)**

Octal Code: 6411/6431  
6451/6471

Operation: Skips the next instruction and clears the Carrier/AGC Flag if the Flag is in the ONE state. The Carrier/AGC Flag is in the ONE state if the Carrier/AGC line has made an ON and/or OFF transition. The detected transitions are jumper selectable.

**Read Status 2 (SRS2)**

Octal Code: 6414/6434  
6454/6474

Operation: Transfers status to AC00-AC07. This instruction is primarily for diagnostic and/or program debug. The AC vs. Status is as follows:

AC00	Carrier/AGC
AC01	Request to Send
AC02	Terminal Ready
AC03	Clear to Send
AC04	TEMA 0
AC05	TEMA 1
AC06	TEMA 2
AC07	Receive Data (inv.)

Field Select Register

#### **Read Status 1 (SRS1)**

Octal Code: 6415/6435  
6455/6475

Operation: Transfers status to AC00-AC07. This instruction is primarily for diagnostic and/or program debug. The AC vs. Status is as follows:

AC00	R-RQST	Receive and Transmit
AC01	T-RQST	Break Requests
AC02	Sync 2	Received "Sync"
AC03	Sync 1	Characters
AC04	REMA 0	
AC05	REMA 1	
AC06	REMA 2	
AC07	Modem Ready	

Field Select Register

#### **Load Field (SLFL)**

Octal Code: 6413/6433  
6453/6473

Operation: Transfers the contents of AC00-AC05 to the field select registers. AC00-AC02 selects the transmit field while AC03-AC05 selects the Receive Field. The selected field (increments of 4K of core—up to 32K) combined with the current address forms a 15 bit address for data transfers to and from core.

#### **Skip on Bus Error (SSBE)**

Octal Code: 6416/6436  
6456/6476

Operation: Skips the next instruction and clears the Bus Error Flag if the flag was in the ONE state. The Bus Error Flag will be in the ONE state if a Transmit or Receive Break Request has not been serviced in less than 1/BAUD time. This flag implies that the Break bus is either overloaded or is inoperative.

#### **Read Character Detected (SRCD)**

Octal Code: 6417/6437  
6457/6477

Operation: The contents of a two bit register which contains the address of the detected character is transferred to AC10 and AC11. The two bits correspond to the two low order bits of the access address where the characters for detection are stored.

### Maintenance Instruction

The SRCD instruction issued when AC00 is set to a ONE causes a single clock pulse on the *External Clock* or secondary Transmit data line (circuit SBA) Jumper selectable line to the modem.

### Summary of Instructions

CODE	MNEMONIC	INSTRUCTION
6400/6420/6440/6460	SSCD	Skip if character detected
6401/6421/6441/6461	SCSI	Clear Synchronous Interface
6402/6422/6442/6462	SSRO	Skip if Receive Word Count O'Flow
6403/6423/6443/6463	SSTO	Skip if Transmit Word Count O'Flow
6404/6424/6444/6464	SGRR	Receive Go
6405/6425/6445/6465	SGTT	Transmit Go
6406/6426/6446/6466	SCSD	Clear Sync Detect
6407/6427/6447/6467	SRTA	Read Transfer Address Register
6410/6430/6450/6470	SSRG	Skip if Ring Flag
6411/6431/6451/6471	SSCA	Skip if Carrier/AGC Flag
6412/6432/6452/6472	SLCC	Load Control
6413/6433/6453/6473	SLFL	Load Field
6414/6434/6454/6474	SRS2	Read Status 2

AC00	Carrier/AGC
AC01	Request to Send
AC02	Terminal Ready
AC03	Clear to Send
AC04	TEMA 0
AC05	TEMA 1
AC06	TEMA 2
AC07	Receive Data (Inv.)

6415/6435/6455/6475	SRS1	Read Status 1
		AC00 R-RQST
		AC01 T-RQST
		AC02 SYNC 2
		AC03 SYNC 1
		AC04 REMA 0
		AC05 REMA 1
		AC06 REMA 2
		AC07 Modem Ready

6416/6436/1656/1676	SSBE	Skip on Bus Error
6417/6437/6457/6477	SRCD	Read Character Detected Low Order Bits (Access Address) AC10 and AC11
6417/6437/6457/6477 with AC00—ONE	—	Test Clock

**Interfacing**

**DP8-E Terminated Modem Leads**—The following chart shows the modem control leads for models 201, 301 and 303 as used in the DP8-E. Unless otherwise specified the 201 levels are Bi-polar levels while the 301 and 303 are current mode.

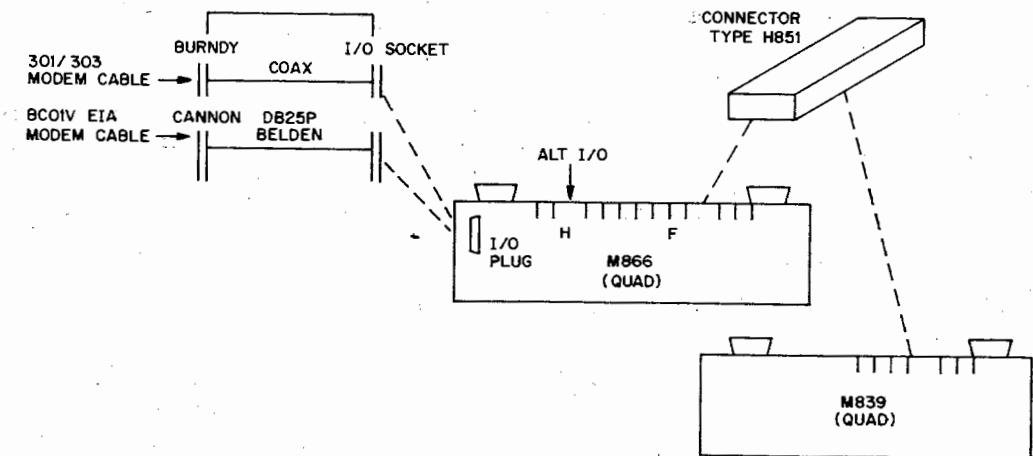
Logic Print	Model 301 (EB)	Model 303 (EB)	Model 201 (EA)
Send Data	Send Data	Send Data	Send Data
Received Data	Received Data	Receive Data	Receive Data
Clear to Send	Clear to Send	Clear to Send	Clear to Send
Interlock/Data Set Ready	Interlock	Data Set Ready	Interlock
Carrier/AGC	Carrier On-Off	AGC Lock	Carrier on-off
Serial Clock Transmit	Serial Clock Transmit	Serial Clock Transmit	Serial Clock Transmit
Serial Clock Receive	Serial Clock Receive	Serial Clock Receive	Serial Clock Receive
Terminal Ready		Data Terminal Rdy*	Remote Control
Ring		Ring Indicator*	Ring Indicator 1
External Timing	Serial Clock Transmit (External)	Serial Clock Transmit (External)	External Timing

\* Bi-polar

**Interface Connections**—The DP8-E is interfaced to a modem according to the following:

DP8-EA → EIA  
DP8-EB → 301/303

(Assembly → M839 + M866 + BC01V-25  
(Assembly → M839 + M866 + BC01W-25



**Top Connector and I/O Pin Assignments**  
**M839 - M866 "F" Connector**

NAME	SOURCE	Pin
	M839	M866
SRCD (Low)		*
This Code (Inverted)	*	FA1
INT RQST	*	FBI
REC DATA (Mark +3)		FC1
REMA (Inverted)	*	FD1
TEMA (Inverted)	*	FE1
SCR-P (Inverted)		FF1
BREAK GRANT		FH1
INITIALIZE (Inverted)	*	FI1
GTP4		FL1
BREAK RQST		FM1
SSBE (Inverted)	*	FN1
SCRT (Inverted)		FP1
CS	*	FR1
SD	*	FS1
IDLE		FT1
RQST	*	FU1
SRS1	*	FV1
Ground		FA2-V2
Spare		None

**The BC01 V (Edge to EIA) and BC01 w (Edge to 301/303) adapter cables perform the following:**

**I/O CONNECTORS**

NAME	Edge	EIA	301	303 (note 1)
Signal Ground	VV	7	All	
Frame Ground	B	1	Shields	
Clear to Send	T	5	C	C
Receive Data	J	3	K	K
Interlock/Data Set Ready	Z	6	F	F
Serial Clock XMIT	N	15	J	J
Serial Clock Receive	R	17	L	L
Carrier/AGC	BB	8	M	M
Ring	X	22	—	F(outer)
Send Data	F	2	E	E
Terminal Ready	DD	20	—	M(outer)
Send Request	V	4	D	D
External Timing	L	24	H	H
-6 Volts	—	—	—	—
+6.4 Volts	—	—	—	—
External Clock	L	—	—	—
SEC Transmit Data	FF	—	—	—
SEC Receive Data	JJ	—	—	—

NOTE: 303 Modem connectors F + M shields (outer connector) carry EIA signals as indicated.

**Jumper Selection**

Unless otherwise specified, the following selections will be provided:

a. DP8-E A, B	ACCESS ADDRESS	IOT CODES	BREAK PRIORITY
1st	7720-7730	640X/641X	5
2nd	7700-7710	642X/643X	4
3rd	7660-7670	644X/645X	3
4th	7640-7650	646X/647X	2

- b. 8 Bits/Character
- c. Normal Clock Phase
- d. Level conversion for DP8-EA will be EIA
- e. Level conversion for DP8-EB will be current mode.
- f. Sync code will be 226 (Octal)
- g. Carrier on/off transister
- h. Full Duplex

**To alter selection:**

M839 Jumpers	<u>Select</u>	<u>Remove Jumper</u>
Bits/Character	6	B8, B7, B78, C7, C8
	7	B8, B6, C8
	8	B7, B6, C7
	1	P2, P3, P4, P5, P6, P7
	2	P1, P3, P4, P5, P6, P7
	3	P1, P2, P4, P5, P6, P7
	4	P1, P2, P3, P5, P6, P7
	5	P1, P2, P3, P4, P6, P7
	6	P1, P2, P3, P4, P5, P7
	7	P1, P2, P3, P4, P5, P6
Break Priority (Generate)	7720(1)	A6
	7700(2)	A6, A7
	7660(3)	A5
	7640(4)	A5, A7
	7620(5)	A5, A6
	7600(6)	A5, A6, A7
Access Address	640X/641X(1)	6, 7
	642X/643X(2)	6, N7
	644X/645X(3)	N6, 7
	646X/647X(4)	N6, N7
Device Code	226	S5, S6, S8, S11
Sync Code (Remove jumper S4-S7 to Select Zero)		

NOTES: (1) 1st DP8      (2) 2nd DP8      (3) 3rd DP8  
           (4) 4th DP8      (5) Spare

**M866 Jumpers\***

\*Jumpers are production inserted with the exception  
  of: C and  $\Delta$

	<u>Select</u>	<u>Remove Jumper</u>	<u>Add Jumper</u>
CO/AGC Transition	ON OFF ON & OFF	OFF ON OFF & ON	
Clock Phase	Inverted	N(TWO)	$\Delta$ (TWO)
Level Conversion	EIA Current TTL	T, CT T, E E, CE	C(Six)
Break Priority (Detect)	1 2 3 4 5 6 7	P1—P6 P2—P6 P3—P6 P4—P6 P5—P6 P6	
Full Duplex		HD	

### EIA RS-232-C Interface Pin Assignments

Pin Number	Circuit	Description
1	AA	Protective Ground
2	BA	Transmited Data
3	BB	Received Data
4	CA	Request to Send
5	CB	Clear to Send
6	CC	Data Set Ready
7	AB	Signal Ground (Common Return)
8	CF	Received Line Signal Detector
9	—	(Reserved for Data Set Testing)
10	—	(Reserved for Data Set Testing)
11	—	Unassigned
12	SCF	Sec. Rec'd Line Sig. Detector
13	SCB	Sec. Clear to Send
14	SBA	Secondary Transmitted Data
15	DB	Transm. Signal Element Timing (DCE Source)
16	SBB	Secondary Received Data
17	DD	Received Signal Element Timing (DCE Source)
18	—	Unassigned
19	SCA	Secondary Request to Send
20	CD	Data Terminal Ready
21	CG	Signal Quality Detector
22	CE	Ring Indicator
23	CH/CI	Data Signal Rate Selector (DTE/DCE Source)
24	DA	Transmit Signal Element Timing (DTE/DCE Source)
25	—	Unassigned

## EIA (RS-232-C) to Equivalent CCITT

Inter-change Circuit	CCITT Equivalent	Description
AA	101	Protective Ground
AB	102	Signal Ground/Common Return
BA	103	Transmitted Data
BB	104	Received Data
CA	105	Request to Send
CB	106	Clear to Send
CC	107	Data Set Ready
CD	108.2	Data Terminal Ready
CE	125	Ring Indicator
CF	109	Received Line Signal Detector
CG	110	Signal Quality Detector
CH	111	Data Signal Rate Selector (DTE)
CI	112	Data Signal Rate Selector (DCE)
DA	113	Transmitter Signal Element Timing (TDE)
DB	114	Transmitter Signal Element Timing (DCE)
DD	115	Receiver Signal Element Timing (DCE)
SBA	118	Secondary Transmitted Data
SBB	119	Secondary Received Data
SCA	120	Secondary Request to Send
SCB	121	Secondary Clear to Send
SCF	122	Sec. Rec'd Line Signal Detector

### Redundancy Check Option

The KG8-EA redundancy check option is designed to complement the DP8-EA and KG8-EA synchronous interface by providing parity generation and checking facilities. Vertical redundancy checks (VRC), longitudinal redundancy checks (LRC), and cyclic redundancy checks (CRC) can be performed by this option. The cyclic redundancy check is industry compatible CRC-12 and CRC-16.

The KG8-EA operates directly with the PDP-8/E from program-controlled instructions. Thus, when not used with the communications equipment, it can be used with other devices. The KG8-EA consists primarily of MSI logic packaged on a single PDP-8/E module, which plugs into the OMNIBUS. All control functions and character options are programmable. The primary purpose of the KG8-EA parity option is to reduce processor overhead for data communications applications where character parity (VRC) and/or Block Check Character (BCC) Accumulation (LRC or CRC) are required for error detection. The types of parity generation or checks that the KG8-EA can perform are defined below:

- Vertical Redundancy—Parity is on a character basis where one bit slot of each character is reserved for the parity bit. Odd parity (odd number of binary ones) is generated by this option; however, capabilities are provided for checking odd or even parity.
- Longitudinal Redundancy—This type is a BCC accumulation over a block of message characters; that is, the LRC is an accumulated EXCLUSIVE OR of all character bits (including parity).

bits) in a message. This method is more reliable than the VRC in detecting errors. A system may use both the LRC and VRC to increase the probability of detecting errors. Both the transmitting and receiving station must compute the BCC; at the end of the message block, the BCC Accumulations are compared at the receiving stations. If they are equal, the message is assumed to be without error.

- c. Cyclic Redundancy—As implemented in this option, is industry compatible for CRC BCC accumulation (CRC16/12).

The CRC check sum is the remainder derived from dividing the numerical value of the message by a constant. The division is performed serially, the quotient is discarded, and the remainder is stored. Both the transmitting and receiving stations must compute the BCC accumulation. At the end of each message block, the BCC accumulation is sent to the receiving station for comparison with the receive station check sum. If the two are equal, the message is assumed to be without error. CRC and VRC operations can be combined to increase the probability of error detection.

### Specifications

Vertical Redundancy Check (VRC)

Tests or computes odd parity for up to eight-bit characters. Parity bit is either right-justified (AC11) or left-justified (AC04).

Longitudinal Redundancy

Computes or compares BCC accumulation for 6, 7, 8, 12 or 16-bit characters. Two bytes required for LRC 16.

Cyclic Redundancy Check (CRC)

Industry compatible for CRC-12 and CRC-16. Division constants used are:  
 $X^{12} + X^{11} + X^3 + X^2 + X^1$  for CRC 12,  
and  $X^{16} + X^{15} + X^2 + 1$  for CRC-16 where X is modulo 2.

Cycle Times

VRC (compute): 1.4  $\mu$ s  
(test) : 1.2  $\mu$ s  
CRC or LRC : 1.2  $\mu$ s  
CRC and VRC (compute): 1.4  $\mu$ s  
CRC and VRC (test) : 1.2  $\mu$ s  
LRC and VRC (compute): 1.4  $\mu$ s  
LRC and VRC (test) : 1.2  $\mu$ s

### Programming

The instructions associated with the DP8-EP option are as follows:

#### Compute VRC (RCCV)

Octal Code: 6XX3

Operation: Transfers character in AC4-11 to DP8-EP parity register, Clears AC and generates odd vertical parity. Result is then jam-transferred to AC with parity bit in AC04 or AC11 as defined by the program.

**Test VRC and Skip (RCTV)**

Octal Code: 6XX0

Operation: Checks parity of character in AC4-11. For odd parity, the next instruction is skipped if parity of character is odd.

**Generate BCC (RCGB)**

Octal Code: 6XX4

Operation: Generates an LRC or CRC Block Check Character (BCC). The LRC can be generated from 6, 7, 8, 12, or 16 (two six-bit bytes) bit characters, while CRC 12/16 can be computed from 6 to 8-bit characters, respectively. BCC verification: The transmitted BCC is compared to the Receive BCC by treating the BCC as part of the overall accumulation. In doing so the receive BCC generator will go to zero if there were no errors in transmission. This instruction also provides the functions defined for RCCV and RCTV if the appropriate control bits are included. (see RCLC instruction).

**READ BCC LOW (RLRL)**

Octal Code: 6XX2

Operation: Jam-transfers the 6, 7, 8, or 12 LSBs of BCC accumulation to the AC (right-justified). The quantity of bits transferred to the AC is dependent on the BCC length selected with the RCLC instruction. The LSB of each byte is also right-justified.

**Read BCC High (RCRH)**

Octal Code: 6XX1

Operation: Jam-transfers the 8 MSBs of BCC accumulation to AC (right-justified). The instruction is used for the 16-bit BCC. The LSB of each byte is also right-justified.

**Clear BCC Accumulation (RCCB)**

Octal Code: 6XX6

Operation: Clears the 16-bit BCC register.

**Load Control (RCLC)**

Octal Code: 6XX5

Operation: Jam-transfers content of AC to redundancy control register to define the operation as follows:

AC05 = 1: CRC BCC  
0: LRC BCC

AC 6 7 8

0 0 0 = 16-bit BCC  
0 0 1 = 12-bit BCC  
0 1 0 = 8-bit BCC  
0 1 1 = 7-bit BCC  
1 0 0 = 6-bit BCCAC 9 = 0: Generated parity to AC4  
= 1: Generated parity to AC11

AC10 = 1: An RCGB instruction also causes a RCCV instruction sequence to accrue. The BCC accumulation will be computed with the corrected character parity.

AC11 = 1: An RCGB instruction also causes a RCTV instruction sequence to accrue.

#### Maintenance Test Clock (RCTC)

Octal Code: 6XX7

Operation: This instruction can only be implemented by grounding test point DA1 on the module. RCTC causes a single clock pulse to the registers, permitting single step testing of LRC and CRC operations.

#### Interface to Bell 201 Modems

The DP8-EA Synchronous Data Interface module is connected to a Bell Model 201 modem (or equivalent) by a 25-ft cable terminated at the modem end with a 25-pin male connector. Standard interface signals are bipolar (EIA/CCITT); however, current mode or TTL compatible signals can be selected using jumper options on the DP8-EA. Interface signals versus connector pin assignments are provided in Table 7-1. In addition, signal or protective ground is provided on pin 1. Signal ground is provided on pin 7.

**Table 7-1**  
**Connector Pin Assignments for Bell Series 201**

Pin	Signal
2	Send Data
3	Receive Data
4	Send Request
5	Clear to send
6	Interlock
8	Carrier on-off
15	Serial Clock transmit
17	Serial clock Receive
20	Remote control
22	Ring indicator 1
24	External timing

### **KL8-E Asynchronous Data Control**

The KL8-E control unit is a PDP-8/E module which plugs into the OMNI-BUS and controls the operation of a Teletype or other similar asynchronous devices from the programmed instructions. This module contains the shift clock, the control logic for IOT decoding, parallel-to-serial and serial-to-parallel converters, and program-control of interrupt and flag facilities. Serial information read or written by the Teletype unit is assembled or disassembled by the KL8-E control for transfer between the Teletype and the AC.

For program operation, the Teletype unit and control are considered as a Teletype in (TTI) for input intelligence from the keyboard or the perforated-tape reader, and as a Teletype Out (TTO) for computer output information to be printed and/or punched on tape. Therefore, two device select codes are used; select code 03 initiates operations associated with the keyboard/reader (TTI), and select code 04 performs operations associated with the teleprinter/punch (TTO). The control unit contains a programmable interrupt enable flip-flop that is common to both the keyboard/reader and teleprinter/punch. This flip-flop is set when power is turned on or INITIALIZE is generated, and can be set or cleared (as specified by AC11) using the KIE instruction. If AC11 is a 1 when the KIE instruction is issued, the interrupt enable flip-flop is set to permit the generation of interrupt requests whenever the keyboard/reader flag or teleprinter/punch flag is set. In contrast, if AC11 is a 0 when KIE is issued, no interrupt can be generated by this control unit. Functions performed by the keyboard/reader and teleprinter/punch are described in subsequent paragraphs.

### **Specifications**

(Also see KL8-EA—KL8-EG)

Interface	20 ma current loop operation.
Character Parameters	Reader controlled by reader enable leads
Baud Rate (Standard)	8 data bits and 1 or 2 (standard) stop units 110 (Other rates available upon special order)
Binary Input/Output	8-bit parallel
Other features	<ol style="list-style-type: none"><li>1) Programmable interrupt enable</li><li>2) Program control for clearing keyboard flag without setting reader run flip-flop.</li><li>3) Program control for setting teleprinter flag.</li><li>4) Jumper-selectable device codes</li><li>5) Input to 17 KL8-E's per PDP-8/E</li></ol>

### **Keyboard/Reader**

The keyboard and tape reader control contains an eight-bit shift register (TTI) which assembles and holds the code for the last character struck on the keyboard or read from the tape. Teletype characters from the keyboard/reader are received serially by the TTI register. The Teletype character code is loaded into the TTI so that spaces (the absence of holes) correspond with binary 0s and holes (marks) correspond to binary 1s. Upon program command, the contents of the TTI are transferred in parallel to the AC.

When a Teletype character is to be read from the paper tape reader, the control de-energizes a relay in the Teletype unit to release the tape feed latch. When released, the latch mechanism stops tape motion only when a complete character has been sensed, and before sensing of the next character is started.

When an eight-bit character has been assembled in the TTI, the keyboard flag is set to cause a program interrupt if the interrupt enable flip-flop has been set. When the program services the interrupt, it senses the flag with a KSF instruction and, with the flag set, issues a KRB instruction which clears the AC, clears the keyboard flag, transfers the contents of the TTI into the AC, and enables advance of the tape feed mechanism.

### **Programming**

The following instructions are used for supplying data to the computer from the keyboard/reader:

#### **Clear Keyboard Flag (KCF)**

Octal Code: 6030

Operation: Clears the keyboard flag without setting the reader run flip-flop. The AC is not cleared by this instruction.

#### **Skip on Keyboard Flag (KSF)**

Octal Code: 6031

Operation: Increments the contents of the PC if the keyboard flag is set, so that the next instruction is skipped.

#### **Clear Keyboard Flag (KCC)**

Octal Code: 6032

Operation: Clears the keyboard flag and AC and sets the reader run flip-flop. This action allows the hardware to begin assembling the next input character in the TTI register. If the reader is activated and there is tape in the reader, a serial character is read from tape and is assembled in the TTI register. The keyboard can also load characters into the TTI register provided the reader is deactivated. In either case, the keyboard flag is set when the character is assembled on the TTI.

#### **Read Keyboard Buffer Static (KRS)**

Octal Code: 6034

Operation: ORs the contents of TTI register with AC4 through 11, and leaves the result in AC4-11. This is termed a static command because neither the AC nor keyboard flag is cleared.

#### **Set/Clear Interrupt Enable (KIE)**

Octal Code: 6035

Operation: Sets or clears the interrupt enable flip-flop as defined by AC11. If AC11 is asserted, generates an interrupt request for a keyboard or teleprinter flag. If AC11 is negated, interrupt requests cannot be generated.

### **Read Keyboard Buffer Dynamic (KRB)**

Octal Code: 6036

Operation: Performs the combined operations of the KCC and KRS instructions. Clears the AC and keyboard flag and transfers the contents of the TTI register to AC4 through AC11. This instruction also sets the reader run flip-flop to begin assembly of another character in the TTI register. When this operation is complete, the keyboard flag is set to indicate another character is available.

A typical TTI instruction sequence for keyboard (manual) input is:

LOOK, KSF	/SKIP IF KEYBOARD FLAGS
JMP LOOK	/JUMP BACK & TEST FLAG AGAIN
KRB	/TRANSFER TTI CONTENTS INTO AC

This sequence waits for the TTI to set its flag, indicating that it has a character ready to be transferred. It then skips to the KRB command which causes the character to be transferred from the TTI to the AC.

The computer clears all flags which are on the clear flag bus (including both the keyboard flag and the reader run enable) when the console CLEAR pushbutton is depressed. This means that the user program must set the reader run enable to obtain data from the reader. The instruction sequence given below is a typical TTI instruction sequence for both keyboard and reader input.

If this sequence of instructions is made a subroutine of the main program, it can be accessed each time an input character is desired. Consequently,

KCC /CLEAR TTI FLAG, SET READER RUN CLEAR/AC

READ, 0 /STORE PC HERE FOR RETURN ADDRESS  
KSF /SKIP IF FLAG = 1  
JMP --1 /TEST FLAG AGAIN  
KRB /READ CHAR INTO AC  
JMP 1 READ /EXIT TO MAIN PROGRAM

### **Teleprinter/Punch**

On program command a character is transferred from the AC to the output shift register (TTO) for transmission to the teleprinter/punch unit. The Teletype control generates the start space, shifts the eight character bits serially into the printer selector magnet of the Teletype unit, and then generates the stop marks. Bit transfer from the TTO to the teleprinter punch unit is at the normal Teletype rate. A character transfer requires 100 ms for completion at 110 baud. The teleprinter flag is set when the last bit of the character code is sent to the teleprinter/punch, indicating that the TTO is ready to receive a new character from the AC.

The flag activates the program interrupt synchronization element and the instruction skip element. When using instruction skip, the program checks the flag by means of the TSF instruction. If the flag is set, the program issues the TLS instruction, which clears the flag and sends a new character from the AC to the TTO. AC-to-TTO transfer time is short compared to the print/punch time, so the program must wait for the flag to set before issuing another TLS.

### **Programming**

Instructions for use in outputting teletype data are as follows:

#### **Set Teleprinter Flag (TFL)**

Octal Code: 6040

Operation: Sets the teleprinter flag to ready the logic for another character.

#### **Skip on Teleprinter Flag (TSF)**

Octal Code: 6041

Operation: If the teleprinter flag is set, increments the contents of the PC by one so that the next sequential instruction is skipped.

#### **Clear Teleprinter Flag (TCF)**

Octal Code: 6042

Operation: Clears the teleprinter flag. This instruction can be micro-programmed with TPC.

#### **Load Teleprinter and Print (TCP)**

Octal Code: 6044

Operation: Transfers AC bits 4-11 to the TTO and starts shifting the character out to the printer/punch unit. This instruction does not clear the teleprinter flag. This instruction can be micro-programmed with TCF to produce TLS.

#### **Skip On Printer or Keyboard Flag (TSK)**

Octal Code: 6045

Operation: Skips the next instruction if the printer or keyboard flag is set and the interrupt enable flip-flop is set.

#### **Load Teleprinter Sequence (TLS)**

Octal Code: 6046

Operation: This instruction combines TCF and TPC. The teleprinter flag is cleared and the contents of AC bits 4-11 are transferred to the TTO, where the hardware shifts the character out to the printer/punch unit. When the TTO has finished outputting the character and is ready for another character, the teleprinter flag is set. The whole operation, from the time at which the TLS has cleared the flag and TTO starts character transfer, until the time the hardware finishes with the character and again sets the flag, requires 100 ms at 110 baud.

A typical TTO instruction sequence is:

```
        CLA
        TAD X      /PUT CHARACTER CODE INTO AC FROM
                    /LOCATION X
FREE,      TLS      /LOAD TTO FROM AC & PRINT/PUNCH
                TSF      /TEST FLAG TO SEE IF DONE PRINTING,
                    /SKIP IF = 1
                JMP FREE /TEST FLAG AGAIN
                CLA      /CLEAR CHARACTER CODE FROM AC
        .
        .
        .
```

This sequence sends one character to the TTO and waits for printing/punching before sending another character. It does not require that the flag be set to output the character. By making the instruction sequence a subroutine of a larger program, it can be accessed by a JMS each time a character is to be output. Assume that the subroutine is entered with the character code in the AC:

```
TYPE,    0
        TLS      /LOAD TTO FROM AC AND PRINT/PUNCH
        TSF      /TEST FLAG, SKIP IF = 1
        JMP .-1
```

By rearranging this subroutine, the 100 ms (at 110 baud) spent waiting for the character to be output and the flag to be set is used to continue the main program, making more efficient use of program time.

```
TYPE,    0      /TEST FLAG TO SEE IF TELEPRINTER FREE,
        TSF      /SKIP IF YES OR ...
        JMP .-1   /WAIT TILL IT IS BY TESTING AGAIN AND
                    /AGAIN
        TLS      /OUTPUT CHARACTER
        CLA      /CLEAR CHARACTER FROM AC
        JMP I TYPE /EXIT TO CONTINUE PROGRAM
```

This subroutine tests the flag first, and waits only if a previous character is still being outputted. It clears the AC, exits immediately after sending the character to the TTO, and continues to run the user's program, instead of waiting while the Teletype (a much slower I/O device) is typing/punching the preceding character.

The computer clears all flags which are on the clear flag bus (including teleprinter flags) when the console CLEAR pushbutton is depressed. This means that the user program must account for setting the teleprinter flag initially and after each TCF (if any), or the program hangs up in the wait loop of the print routine. The only way to set the flag

is by initializing it. This instruction should appear among the first few executed, and must appear before any attempt to output a character. The following example initializes the flag as the first instruction of the program and makes optimum use of the punch/print time.

BEGIN, TFL

```
    .  
    .  
TYPE, 0  
    TSF      /SKIP IF FLAG = 1 OR ...  
    JMP .-1   /WAIT UNTIL IT IS LOAD TTO &  
    TLS      /TYPE CHARACTER  
    CLA      /CLEAR CHARACTER FROM AC  
    JMP I TYPE /EXIT CHARACTER FROM AC  
    .          /EXIT & CONTINUE PROGRAM WHILE  
    .          /TELEPRINTER IS FINISHING CHARACTER  
    .
```

#### Asynchronous Data Controls KL8-EA through KL8-EG

In addition to the KL8/E Asynchronous Data Control described above, the following options are available:

- KL8-EA 110 baud, EIA data lead interface
- KL8-EB 150 baud, EIA data lead interface
- KL8-EC 300 baud, EIA data lead interface
- KL8-ED 600 baud, EIA data lead interface
- KL8-EE 1200 baud, EIA data lead interface
- KL8-EF 1200 baud transmit, 150 baud receive,  
            EIA data lead interface
- KL8-EG 2400 baud transmit, 150 baud receive,  
            EIA data lead interface

These options are programmed identically to the KL8-E previously described, and like the KL8-E have jumper selectable device codes so that a number of asynchronous data terminals (not to exceed 17) may be connected to a PDP-8/E merely by adding the necessary KL8-units. Each unit requires two device codes.

The EIA data lead interface conditions the transmitted and received data leads to the requirements of EIA specification RS-232C and CCITT Recommendation V24. These leads, along with Data Terminal Ready and Request to Send (both of which are held in the asserted state), are brought out in a standard 25 pin male connector suitable for direct connection to a modem. The modem used should be a full duplex private (non-switched) line modem such as the Bell System 103F or a switched network modem used in manual mode such as the Bell System 103A without automatic answering. Since the KL8 Asynchronous Data Controls do not provide program control of the modem interface leads, use of these controls in automatic originating or automatic answering applications is not recommended.

## CARD READER OPTIONS

### Type CR8-F Card Reader and Control

The CR8-F Card Reader option equips the PDP-8/E computer to accept input from EIA standard data cards. It reads 12-row, 80-column punched cards at a nominal rate of 200 cards per minute photoelectrically. The control circuit for this device is located on a single PDP-8/E module, which plugs into the OMNIBUS. The card reader has an internal power supply and can be tested off-line. For table space requirements, please refer to the specification section which follows.

A select instruction starts a card moving through the read station, where all 80 columns are read on a column-by-column basis, beginning with column one. Card data may be read in any one of three modes. In the binary reading mode, the data is transferred directly from the rows of the card to bits in the AC. The top row of the card (row 12) goes into AC0 and the bottom row (row 9) goes into AC11. In the alphanumeric reading mode, the data is automatically decoded into a six-bit BCD representation and transferred into the least significant six bits of the accumulator. Use of the six-bit decoding minimizes the size of translation tables and is fully compatible with the Hollerith code as used at this time. A proposed expansion of the Hollerith code would require use of the compressed reading mode. In this mode, rows 9, 12, 11, 0, and 8 are transferred directly to AC4, AC5, AC6, AC7, and AC8, respectively, while rows 1 through 7 are decoded into three-bit BCD representation in AC9, AC10, and AC11. This decoding is based on the lack of double punches in rows 1 through 7, both in the present Hollerith and the proposed extension of Hollerith. If such a double punch is read in the compressed reading mode, the CR8-F validity checking circuitry will assert a one in AC0 (the sign bit). Regardless of the reading mode being used, a punched hole is interpreted as a binary one and the absence of a hole is binary zero.

Four program flags indicate card reader conditions to the computer. (The status of these flags may be examined by means of the RCNI instruction.) The Data Ready Flag sets, requesting a program interrupt, when a column of information is ready to be transferred into the AC. A read instruction (alphanumeric, binary, or compressed) must be issued within 1.0 ms after the Data Ready Flag sets in order to avoid data loss. The Card Done Flag sets, requesting a program interrupt, when the card leaves the read station. A new select instruction must be issued immediately after the Card Done Flag sets to keep the reader operating at rated speed.

The Ready True Transition Flag sets, requesting a program interrupt, whenever the ready lead from the card reader to the control goes true, indicating that the card reader is ready. This feature permits the computer program to perform other tasks while awaiting manual intervention to clear a card reader problem such as lack of cards. The interrupt will notify the computer when the card reader is ready to resume reading cards. The fourth flag, the Trouble Transition Flag, sets, requesting a program interrupt, whenever the ready lead from the card reader to the control goes false, indicating an error condition in the card reader. (Error condition when used here refers to a transport error, not a data error

such as an improper validity check.) This flag is cleared by initialize or by means of the Clear Transition Flags instruction.

### Specifications

Size:	18 in. high; 14 in. wide; and 18 in. deep.
Weight:	52 lb.
Card Rate:	200 per minute
Input Power:	115 Vac + or - 10 Vac, 60 Hz + or - 5 Hz, single phase, 300W maximum (50 Hz unit available)
Card Specification:	The card reader is designed to read $7\frac{3}{8}$ in. x $3\frac{1}{4}$ in. cards conforming to the material and size requirements of EIA Standard RS-292 Media 1.
Card Capacity:	Both input hopper and output stacker hold 400 cards. Cards may be added or removed during reader operation.

### Programming

The following instructions are used with the CR8-F option:

#### Skip on Data Ready (RCSF)

Octal Code:	6631
Operation:	Senses the status of the data ready flag; if it is set (indicating that information for one card column is ready to be read), the contents of the PC are incremented by one, so that the next sequential instruction is skipped.

#### Read Alphanumeric (RCRA)

Octal Code:	6632
Operation:	Transfers the six-bit Hollerith code for the 12 bits of a card column into AC6-11, and clears the Data Ready Flag. This instruction does not detect illegal characters.

#### Read Binary (RCRB)

Octal Code:	6634
Operation:	Transfers the 12-bit binary code for a card column directly into the AC, and clears the Data Ready Flag. Information from the card column is transferred into the AC so that card rows 12, 11, and 0 enter AC0-2 and card rows 1 through 9 enter AC3-11, respectively.

#### Read Conditions Out to Card Reader (RCNO)

Octal Code:	6635
Operation:	Reads AC10 into a Ready True Transition/Trouble Transition interrupt enable flip-flop. If AC10 is a 1, this flip-flop is set, enabling the generation of an interrupt whenever the reader goes from not ready to ready or from ready to not ready. This flip-flop is cleared when the PDP-8/E is initialized. The RCNO instruction also reads AC11 into a flip-flop which, when set by AC11 being a 1, enables the generation of an interrupt whenever the card done or data

ready flags are raised. For program compatibility with other family-of-eight computers, initializing the PDP-8/E sets the card done/data ready interrupt enable.

#### **Read Compressed (RCRC)**

Octal Code: 6636

Operation: Transfers an eight-bit compressed code for the 12 bits of a card column into AC4-11, and clears the data ready flag. Data from row 9 goes to AC4, zones 12, 11, and 10 to AC5, 6, and 7 respectively, and data from row 8 goes to AC8. Data from rows 1 through 7 is compressed into a BCD representation in AC9, 10, and 11. Should there be more than one bit of data in rows 1 through 7 (an invalid condition), hardware validity check circuitry will read a 1 into AC0 (sign bit).

#### **Read Conditions int from Card Reader (RCNI)**

Octal Code: 6637

Operation: Status of Ready True Transition Flag, Trouble Transition Flag, Card Done Flag, and Data Ready Flag is read into AC3, AC2, AC1, and AC0 respectively.

#### **Skip on Card Done Flag (RCSD)**

Octal Code: 6671

Operation: Senses the status of the card done flag; if it is set (indicating that the card has passed the read station), the contents of the PC are incremented, skipping the next instruction.

#### **Select Card Reader and Skip if Ready (RCSE)**

Octal Code: 6672

Operation: Senses the status of the card reader; if it is ready, the contents of the PC are incremented, skipping the next sequential instruction, a card is started toward the read station from the input hopper, and the Card Done Flag is cleared.

#### **Clear Card Done Flag (RCRD)**

Octal Code: 6674

Operation: Clears the Card Done Flag. This instruction allows a program to stop reading at any point in the card deck.

#### **Skip if Interrupt Being Generated (RCSI)**

Octal Code: 6675

Operation: Senses the status of all flags. If a flag is raised and the generation of interrupts by that flag is enabled, the next sequential instruction is skipped.

### **Clear Transition Flags (RCTF)**

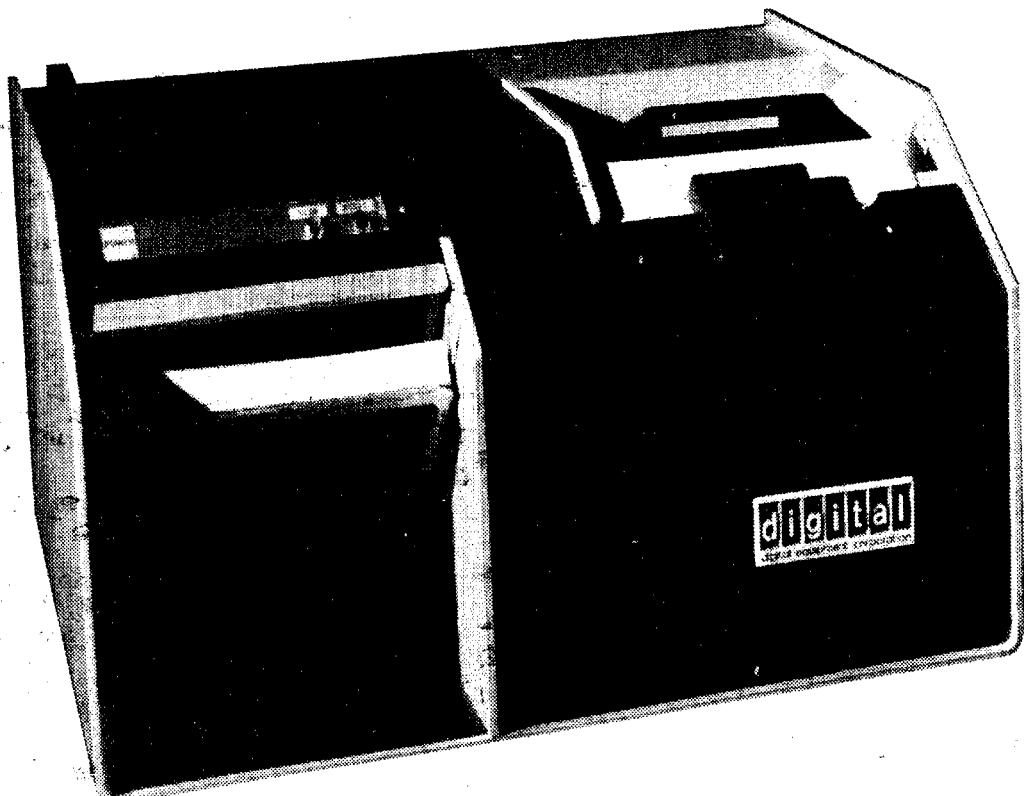
Octal Code: 6677

Operation: Clears the Trouble Transition Flag and the Ready True Transition Flag.

### **Example Subroutine**

A logical instruction sequence to read cards is the following:

START,	RCSE	/START CARD MOTION AND SKIP IF /READY
	JMP NOTRDY	/JUMP TO SUBROUTINE THAT TYPES /OUT "CARD READER MANUAL INTER- /VENTION REQUIRED" OR HALTS
NEXT,	RCSF	/DATA READY?
	JMP DONE	/NO, CHECK FOR END OF CARD
	RCRA or RCRB	/YES, READ ONE CHARACTER OR ONE /COLUMN AND CLEAR DATA READY FLAG
DONE,	DCA I STR	/STORE DATA
	RCSD	/END OF CARD?
	JMP NEXT	/NO, READ NEXT COLUMN
	JMP OUT	/YES, JUMP TO SUBROUTINE THAT /CHECKS CARD COUNT OR REPEATS AT /START FOR NEXT CARD



DEC offers the CR8-F Card Reader and Control Option and the CM8-F Optical Mark Card Reader and Control Option.

The CR8-F performs validity checking only when using the RCRC instruction. A programmed validity check can also be performed by reading each card column in both the alphanumeric and binary modes (within the 1.0 ms time limitation), and then making a comparison check.

### **Switches and Indicators**

**POWER** (toggle circuit breaker and indicator). All power to the card reader is controlled by this switch.

**STOP** (momentary-action pushbutton/indicator switch). Actuation of the STOP switch immediately overrides the PICK COMMAND and lowers the READY line. The card reader will stop operation after the card currently in the track is read completely. Power is not removed from the reader by this action. The red STOP indicator is illuminated as soon as the switch is depressed.

**RESET** (momentary-action pushbutton/indicator switch). Actuation of the RESET switch clears all error flip-flops and initializes all counters.

**READ CHECK** (indicator). The READ CHECK alarm indicator denotes that the card just read may be torn on the leading or trailing edges or have punches in the 0 or 81st columns. A READ CHECK error will cause the reader to stop.

**STACK CHECK** (indicator). The STACK CHECK alarm indicator denotes that a card has failed to reach the read station after a PICK COMMAND has been received.

**HOPPER CHECK** (indicator). The HOPPER CHECK alarm indicator denotes that either the input hopper is empty or the stacker is full.

**LAMP TEST** (pushbutton switch). The LAMP TEST switch illuminates all front-panel indicators to identify an inoperative indicator lamp.

**MODE** (toggle switch). When placed in the LOCAL position, this switch disables the PICK COMMAND input to the card reader, allowing the operator to run the reader off-line by depressing the RESET switch on the front panel. When placed in the REMOTE position, this switch enables the PICK COMMAND input to the card reader, placing the reader on-line for normal remote control operation. Data and other output signals are present at all times.

**SHUTDOWN** (toggle switch). When placed in the MANUAL position this switch energizes the blower for continuous operation whether or not cards are in the input hopper. When placed in the AUTO position, this switch provides an automatic shutdown of the blower when the input hopper is emptied. The blower will automatically restart when cards are placed in the hopper and the RESET switch is depressed.

### Type CM8-F Optical Mark Card Reader and Control

The CM8-F Optical Mark Card Reader option permits the PDP-8/E computer to accept information from marked or punched data cards with timing marks at a nominal rate of 200 cards per minute. It reads 12-row, 40-column mark sense cards and 12-row, 40-column punched cards. The control circuit is located on a single PDP-8/E module that plugs into the OMNIBUS.

A select instruction starts a card moving through the read station, where all 40 columns are read on a column-by-column basis, beginning with column one. Card data may be read in any one of three modes. In the binary mode, the data is transferred directly from the rows of the card to bits in the AC. The top row of the card (row 12) goes into AC0 and the bottom row (row 9) goes into AC11. In the alphanumeric mode, the data is automatically decoded into a six-bit BCD representation and transferred into the least significant six bits of the accumulator. Use of the six-bit decoding minimizes the size of translation tables and is fully compatible with the Hollerith code as used at this time. A proposed expansion of the Hollerith code would require use of the compressed reading mode. In this mode, rows 9, 12, 11, 0, and 8 are transferred directly to AC4, AC5, AC6, AC7, and AC8, respectively, while rows 1 through 7 are decoded into three-bit BCD representation in AC9, AC10, and AC11. This decoding is based on the lack of double punches in rows 1 through 7, both in the present Hollerith and the proposed extension of Hollerith. If such a double punch is read in the compressed reading mode, the CR8-F validity checking circuitry will assert a one in AC0 (the sign bit). Regardless of the reading mode being used, a punched hole or a non-reflective mark is interpreted as a binary one and the absence of such a hole or mark is interpreted as a binary zero.

The flag and interrupt facilities are identical to those of the CR8-F.

### Specifications

Size:	13 in. high, 20 in. wide, 15 in. deep.
Weight:	70 lb.
Card rate:	300 per minute.
Hopper capacity:	550 cards.
Operating temperature:	15°-40°C (50°-105° F)
Power consumption:	400 watts.

### Programming

The instruction set and example subroutine set forth for the CR8-F also applies to the CM8-F.

### **RK8-E Disk Cartridge System**

The RK8-E consists of a single-cycle data break control and one RK05 DECpack drive. The control is made up of three quad modules which plug directly into the PDP-8 OMNIBUS. The modular design permits computing power to be easily expanded since one RK8-E control can support four RK05 DECDrives. Each RK05 contains its own head position control and power supply, so that expansion doesn't require the purchase of this additional equipment.

The RK05 DECpack Drive is a random-access, high density, rotating memory with movable read/write heads and a removable disk cartridge. Each RK05 provides 1.6 million words of storage, giving a fully expanded RK8-E disk system over 6 million words of information and offering the PDP-8 user an economical alternative to magnetic tape.

The disk is driven by an induction motor and rotates smoothly and quietly with little electrical or mechanical noise. The drive comes on-line less than ten seconds after power is applied.

Indicator lights on each drive show the following conditions: power on, system ready, cylinder on, fault, write protect, load, write, and read.

To assure accurate storage and transfer, DECpack systems employ a write-check function, maintenance features, and hardware features which verify the correct track selection and provide a check-sum.

Voice coil positioning and an optical position transducer provide fast access times and accuracy to within 100 millionths of an inch. The unique head carriage design requires no lubrication and supports the head at its center of gravity for accurate, repeatable tracking. By eliminating mechanical braking, the design eliminates a major source of wear and critical adjustment. To protect the disk from damage, an emergency retract power supply automatically forces the heads to their home position if line power fails.

Cartridges are easy to load and unload since each disk is permanently mounted inside a protective case that automatically opens when inserted in the disk drive. As the cartridge is pushed into place, the access door is tightly sealed, protecting the disk from dust and dirt.

For high reliability, the electronics of the DECpack system are cooled with forced air whenever power is on, even when the disk is stopped. Clean cabinet air is introduced into the rear of the drive through a foam pre-filter. The air passes over the electronics module through the blower to a high-efficiency filter which removes 99.97 percent of the ambient particles greater than 0.3 micron in size. The resulting ultra-clean air is fed at a rate of more than 30 cfm to the power supply, positioner, and disk cartridge.

The DECpack power supply is self-contained in each drive to simplify system installation and ensure compatibility. Due to drive efficiency, only 160 watts per drive are required during operation.

The RK8-E's storage capacity opens a broad range of applications formerly restricted by the high cost of fast access bulk storage. In such applications as inventory control, an RK8-E cartridge disk system provides an inexpensive storage device for information on thousands of items, with the option of expansion as the need arises.

## **SPECIFICATIONS**

### **Capacities**

Disks/drive	1
Surfaces/disk	2
Heads/drive	2
Recording density	2200 bpi maximum
Disk capacity (words-formatted)	1,662,976
Disk capacity (bits-unformatted)	24,400,000
Drives/controller	1 to 4
Total system capacity (words)	6,665,904
Tracks	400 + 6 spare
Cylinders	200 + 3 spare
Sectors per track	16
Words/sector	256
Words/track	4096

### **Transfers**

Minimum transfer	128 words
Data transfer rate	8.32 $\mu$ sec/word
Bit transfer rate	1.44 million bits per second
Bit transfer code	Double frequency, non-return-to-zero

### **Access Times**

Disk rotational speed	1500 RPM
Average latency	20 ms (half revolution)
Typical access times (including head settling)	10 ms (track-to-track) 50 ms (average random move) 85 ms (200 track movement)

### **Physical Control**

3 Quad plug into  
OMNIBUS (3.1 Amps @ +5V). One cable from control to first drive.

### **AC Drive Power**

Operating Power	160 W at 2.1 Amps (110V)/1.1 Amps (230V)
Idle Power	80 W at 0.74 Amps (110V)/0.37 Amps (230V)
Starting Current (power only)	1.8 Amps (115V)/ 0.9 Amps (230V)
Starting Current (to start spindle)	10 Amps for 2 sec. (115V) 5 Amps for 2 sec. (230V)

### **Environment**

Ambient temperature	60 to 110° F
Relative humidity	8 to 80 per cent (no condensation)

**Dimensions**

Drive dimensions	19-inch wide 26½-inches deep 10½-inches high (Rack mounted, slides incl.)
Weight of drive	110 pounds

**Ordering Information**

Controller/Drive Model No.	RK8-EA control plus one drive 60 Hz RK8-EB control plus one drive 50 Hz
----------------------------	--

**Additional Drives**

RK05-AA	95-130 VAC, 60 Hz
RK05-BB	190-260 VAC, 50 Hz

**Cartridge Model No.**

RK03-KB

**Disk Format**

The header word of every sector contains the cylinder address of that sector. During a normal read or write cycle, the RK8E will seek a cylinder and then read the header word of the first sector it finds after the seek is complete. This header word is compared with the cylinder address register to ensure the correct cylinder has been found. The control then searches for the sector previously specified by the program and carries out the read or write as contained in the command register.

The disk format consists of a 140μS preamble of zero bits (timing only), a synch bit (data = 1), a 16-bit header word containing the cylinder address, 256 data words of 12-bits each, a 16-bit cyclic redundancy check sum, and a 25 microsecond postamble/erase delay zone. The header word format is 16-bits consisting of 5 zero bits, an 8-bit cylinder address followed by 3 zero bits.

**CRC Character**

Data is recorded on the RK8E Disk in blocks of 256 twelve-bit words. Each block, therefore, contains 3072 bits of data, recorded as a single string of data bits, with no separations to indicate word boundaries or to include error checking (parity) bits.

A long string of data bits is susceptible to single-bit errors caused by drop-ins and drop-outs and burst errors which are caused by unwanted physical motion of the read/write heads.

The RK8E Control checks for this type of error by using a Cyclic Redundancy Check (CRC) that generates a block check word while writing or reading the data. The block check word that is calculated during writing, is written immediately following the data and becomes part of the data string. The process used to calculate the block check word is such that, if no errors occur, the block check word calculated during reading is identical to that calculated during writing.

### **Write Protect**

Any attempt to write on a write protected drive will be inhibited and produce an error condition. Write protect can only be turned off by a manual control on each drive. It can be turned on again by a Command Register bit loaded under program control.

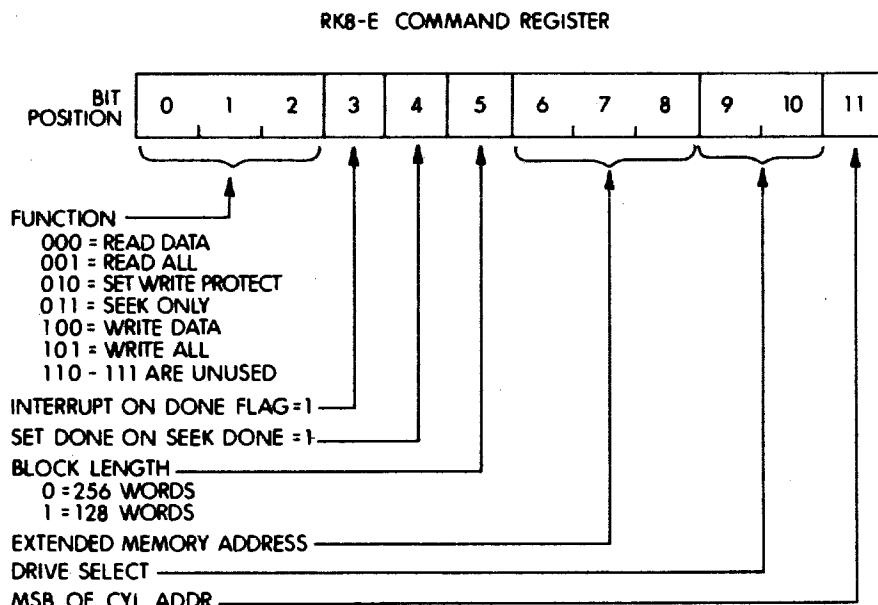
### **MAJOR REGISTERS**

The major registers of the RK8E control are in two broad categories; those that are loaded or read by user software and those that are transparent to the user software. The first category includes the Command Register, Current Address Register, Disk Address Register, and Status Register. The second category includes the four word Data Buffer Registers, the CRC Register, the Major State Register, the modulo 12 and modulo 16 bit counters and the modulo 128 or 256 Word Register.

#### **Command Register**

The Command Register is loaded by IOT 67X6 from AC bits 0-11. This IOT also clears the accumulator and the Status Register.

The Command Register bit functions are:



### **Current Address Register**

This 12-bit register is loaded from the accumulator with IOT 67X4 and the accumulator is cleared. It specifies, with bits 6-8 of the Command Register, the absolute memory address that data is to be transferred to or from memory. The current address register is also incremented after each memory transfer. A current Address Register overflow does not increment the Extended Memory Address contained in the Command Register.

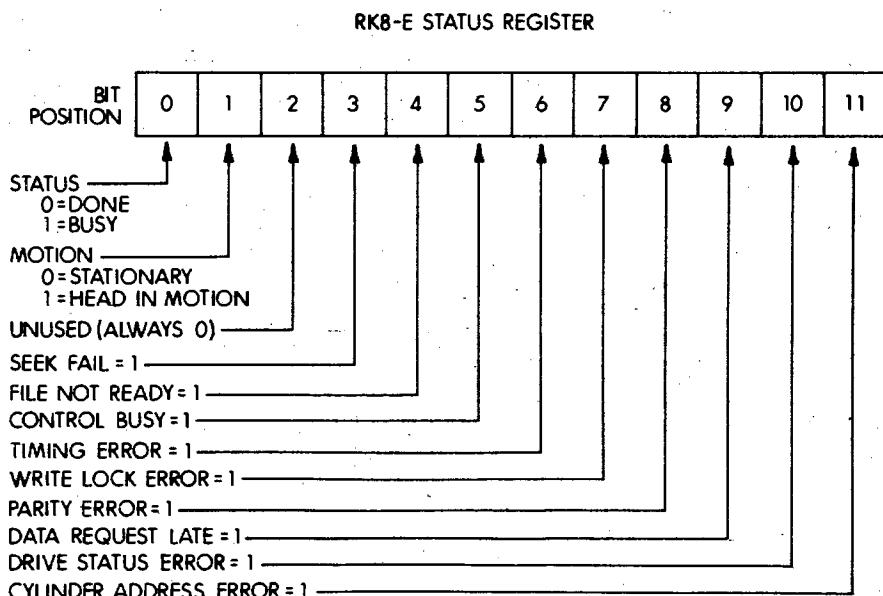
### Disk Address Register

The disk address register is loaded with IOT 67X3. This IOT also clears the accumulator and executes the instruction in the Command Register.

The disk address register specifies the sector address where data is to be transferred. To address an individual sector, one of 203 cylinders, one of two surfaces and one of 16 sectors must be addressed. The addressing scheme is binary, with sectors being the least significant bits, followed by the surface bit, followed by the 8 cylinder bits, with the most significant cylinder bit (Bit 11) in the command register. The largest valid octal address is 14537 or in decimal: cylinder address 202, Head Select 1 sector 15.

### Status Register

The status register contains all the information a program requires to determine if the command to the RK8E is complete and whether or not it was successful:



## PROGRAMMING

### General

The following describes basic program sequences required to operate the RK8E.

### Format A Disk Cartridge

The RK8E control contains all the logic required to format a disk cartridge. All the program must do is address every sector on the disk and write on every sector using write all mode. The track address (cylinder and surface when DLAG was issued) will automatically be written on the particular sector header word selected by the control. The data written is not important except that the data should somewhere contain addressing information so a check can be made that the RK8E and drive found the correct sector on the correct surface and cylinder.

## RK8-E IOT Instructions

MNEMONIC	CODE	OPERATION
<b>DSKP</b>	<b>67X1</b>	<b>Disk Skip on Flag.</b> If the Transfer Done Flag or the Error Flag is set, the PC is incremented to skip the next sequential instruction.
<b>DCLR</b>	<b>67X2</b>	<b>Disk Clear.</b> Function regulated by AC bits 10 and 11:  AC10    AC11  0        0        (DCLS) 0        1        (DCLC)  Clear AC and Status Register. Clear AC and Control, Major registers are cleared. This instruction will stop the control even if it is rewriting the header.
		1        0        (DCLD)  Clear AC, Recalibrate Selected Drive to track 000 and clear Status Register.
<b>DLAG</b>	<b>67X3</b>	<b>Load Address and Go.</b> The disk cylinder, surface and sector bits are loaded from ACO-6, AC7 and AC8-11 respectively, and the function indicated by the current content of the command register is executed.
<b>DLCA</b>	<b>67X4</b>	<b>Load Current Address.</b> The content of the AC is loaded into the disk current address register, and the AC is cleared.
<b>DRST</b>	<b>67X5</b>	<b>Read Status.</b> The AC is cleared, and the content of the disk status register is transferred into the AC.
<b>DLDC</b>	<b>67X6</b>	<b>Load Command.</b> The content of the AC is loaded into the disk command register. The AC and status register are both cleared.
<b>DMAN</b>	<b>67X7</b>	<b>Maintenance Instruction:</b>

To write an unformatted disk, the command register function bits (bit 0, 1, 2) should contain octal (Write All). To read a newly formatted disk, the function bits should contain octal 1 (Read All). Using these function bits prevents the RK8E from reading header words and reporting header errors that will certainly exist on an unformatted disk.

### **Sequence of instructions to Format a disk**

- A) Set up current address.
- B) Set up command register 5000 for Write All, 1000 for Read All.
- C) Set up first disk address and Go.
- D) Wait for Transfer Done Flag, check for errors.
- E) Set up current address again—make changes.
- F) Set up second disk address and Go.

### **Normal Read/Write**

The programming sequence for a Write Data or Read Data Mode is very similar to the sequence to format a disk. The sequence is:

- A) Set up current address.
- B) Set up command register 0000 for Read Data, 1000 for Write Data.
- C) Set up required disk address and Go.
- D) Wait for Transfer Done Flag, check for errors.

### **Seek Only**

The sequence for Seek Only (command register function bits equal three) is different from Write All, Read All, Write Data or Read Data. It is necessary to put two skip or interrupt sequences in the program.

The programming sequence is:

- A) Set up command register, 3000 for seek only and desired drive number.
- B) Load disk address and Go.
- C) Wait for Transfer Done Flag.
- D) Clear status register "Done Flag."
- E) Drive is now seeking but if the Transfer Done Flag is to be set when the seek is complete, bit 4 of the command register must be a 1, issue a load command with selected drive number in AC 10-11 and AC Bit 4=1.
- F) Wait for Transfer Done.

The alternative to E) and F) above is to check the condition of Status Bit 1 which is zero when the seek is complete.

### **Overlapped Seek**

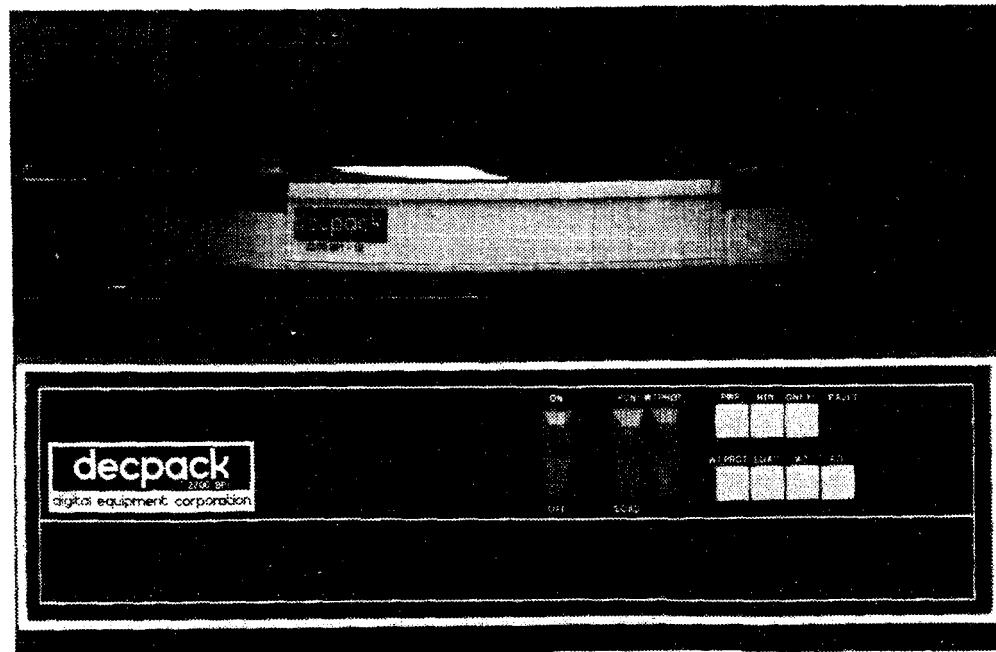
Overlapped seeks make use of the seek only feature. The Program starts multiple drives seeking and then periodically selects a different drive to determine if it has completed seeking. A different drive may be selected after 3.4 step c but in general, programs operating with other program interrupt devices, could be confused as to which drive actually completed a seek. For example, if bit 4 of the command register is a one (allowing seek complete to set the Transfer Done Flag) and multiple drives are seeking, it is possible to select a different drive just as the one previously selected completes its seek. The Transfer Done Flag will be set but the unsophisticated program will think the newly selected drive has completed a seek. A program has two methods of getting around this problem.

- A) Before selecting a new drive, change the command register to make bit 4 a zero without changing the drive number. Then change the drive number with bit 4 equal to a 1. If the new drive has completed a seek, there will be no confusion as to which drive set the Transfer Done Flag.
- B) Leave bit 4 of the command register set and check bit 1 of the Status Register to determine if the drive selected or the drive previously selected set the Transfer Done Flag. Bit 1 of the Status Register is zero if the selected drive has completed the seek.

#### **Data Transfers on Consecutive Sectors**

Octal 1 or octal 4 (Read All or Write All) in the function portion of the Command Register allows a program to format a virgin disk cartridge by disabling the checking of headers. This feature is also used to transfer data on consecutive sectors. Assume a program requires to transfer 512 words of data on two consecutive sectors. The program should use Read Data and Write Data (octal 0 and 4 in the function portion of the command register) for the transfer of the first 256 words of data. When the first 256 words have been transferred and Transfer Done Flag is set, there is a minimum of 100  $\mu$ s for the program to set up the control for transfer to the next consecutive sector. The function portion of the command register should be changed to octal 1 or octal 5 (Read All or Write All), the status register cleared and the next sector specified along with surface and cylinder when issuing DLAG. The control will not check the header of the next sector but will transfer 256 words of data.

It is not necessary to change the Current Address Register, if a 512 word buffer is available in memory.



## OMNIBUS MAGNETIC TAPE OPTIONS

### TA8-E DECcassette

The TA8-E DECcassette is a minimagtape system designed and manufactured by DIGITAL as an alternate choice to paper tape. It joins an extensive mix of peripherals that give the PDP-8 user a wide range of linear bulk-storage devices: high-speed paper tape, DECcassette, DEC-tape, and DECmagtape.

Two major components comprise the TA8 DECcassette system; the TU60 dual cassette drive and the quad board interface card.

The TU60 consists of several elements. The first are two separate cassette drives, each having a single solenoid and a pair of DC motors to provide direct reel-to-reel motion without a capstan. The second element consists of two large printed-circuit boards securely mounted on a moveable frame such that the boards swing up for servicing. These boards are the electronics of the TU60 and perform the data formatting, error checking (CRC generation and testing), and bit-to-byte conversion. The third element is the self-contained DC power supply. All these elements are housed in a compact, completely enclosed chassis that mounts in a standard 19" Digital cabinet, or can be used in a table-top configuration.

The TA8-E interface is contained on a single quad board that uses one Omnibus™ slot. It is connected to the TU60 via two flat ribbon cables in the rack mounted version, or by a single, round cable in the table-top configuration. In both cases the cables are standard lengths, up to a maximum of 25 feet.

### Cassette and Tape

The physical cassette is a digital-grade "Philips" type, with the hub modified to optimize data capacity.

The tape is a heavy "sandwich" type having a 1 mil base, hard oxide, and the same proprietary coating used on Decatape. The length is 150 feet. By using this stronger, heavier tape, the useful life of the cassette is vastly extended. In actual "real world" life tests, performed under normal operating conditions, the cassette tape has withstood thousands of full passes with error rates well within the specified error rates. The performance of the cassette is so reliable that Digital guarantees a minimum of 1000 passes!

Two cassette drives are contained in one chassis and may be ordered as either a rack mount or table top version. The cassette system, TA8-E, has the following variations:

TA8-AA—M8331 module & TU60-AA, Rack Mount, 115VAC  
TA8-AB—M8331 module & TU60-AB, Rack Mount, 230VAC  
TA8-BA—M8331 module & TU60-BA, Table Top, 115VAC  
TA8-BB—M8331 module & TU60-BB, Table Top, 230VAC

Jumpers on the interface card can be arranged to select IOT codes 70 through 77; code 70 for the first TU60, and code 77 for the eighth, or last TU60. The entire interface is contained on one 8-1/2 quad module.

**SPECIFICATIONS:**

Medium:	0.150" wide computer-grade, 100% certified, 150 ft., 1 mil mylar substrate, proprietary DEC "Philips-Type" cassette.
Type of Recording:	Phase encoded, blocked.
Number of Tracks:	One (full width).
Cassette Capacity:	Full tape 93,000 bytes minimum; with 256 byte blocks, 87,000 bytes minimum.
Transfer Rate:	560 bytes/sec. (peak); with 256 byte block, 487 bytes/sec. (average).
Number of Drives:	Two per control electronics.
Functions:	Read forward one block, write forward one block, back up one block, space forward to file gap, space back to file gap, write gap, rewind, manual rewind.
Data Format:	Varied block length, hardware formatted, with length software controlled.
Block Length:	1 byte minimum, no maximum.
Error Control:	16-bit cyclic redundancy check (CRC), hardware generated and appended to data at time of writing. Tested during read by hardware via program command.
Error Rate:	1 in $10^8$ write errors 1 in $10^8$ unrecoverable read errors 1 in $10^7$ recoverable read (3 retry)
Drive: Reel-to-reel.	Average speed, read/write 9"/sec. Average speed, search 21"/sec.
Start/Stop Time:	< 20 ms., linear ramp controlled.
Density of Recording:	350-700 bits/inch.
Rewind Time:	< 30 sec., typical 20 sec. Speed, servo controlled.
Data Interface:	8-bit parallel, synchronous program transfers via processor interface.
Read Electronics:	Peak detection/phase lock loop (low threshold read).
Power Requirements:	90V to 132V, 180V to 265V, 48 to 63 Hz (self-contained DC supply). 120 watts maximum.

**Beginning/End Tape  
Detector:** Clear leader.

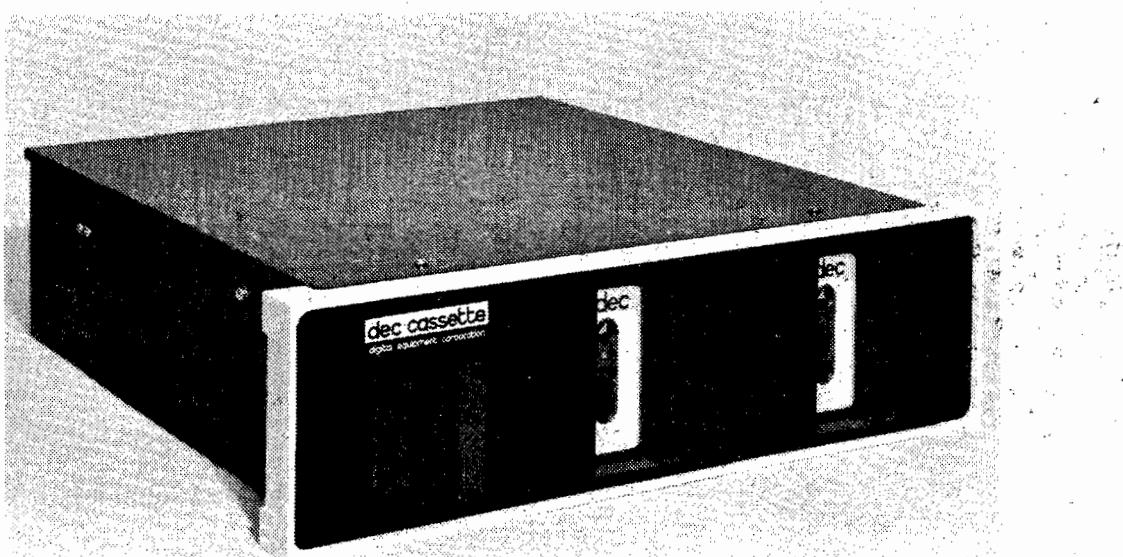
**Environmental:** Operating temp. range: 50°F. to 105°F., storage temp. range (cassette): 40°F. to 122°F., operating humidity range: 20% to 80% non-condensing. Max. wet bulb 85°F. Altitude 0 to 10,000 feet.

**Indicators and Controls:** Power on (each drive), manual rewind, write protect (on cassette).

**Size:** 5-1/4" x 19" x 18-1/2"

**Weight:** 32 lbs.

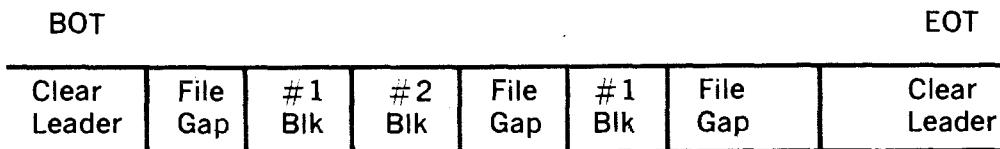
**TA8 Variations:**  
**TA8-AA** Rack Mount 115 VAC  
**TA8-AB** Rack Mount 230 VAC



### TA8-E DECcassette IOT Instructions

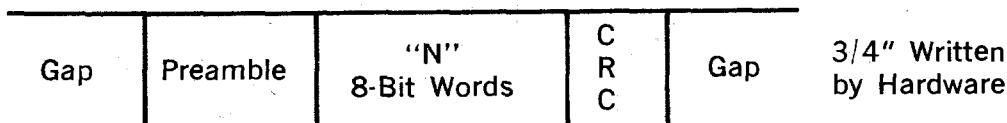
MNEMONIC	OCTAL CODE	FUNCTION
KCLR	67x0	Clear all. Clears Status A and B registers. See note below.
KSDR	67x1	Skip on Data flag, for either a read or a write.
KSEN	67x2	Skip on, EOT/BOT or EOF or/Drive Empty or Timing Error, Block Error or Write Lock and "Write" True.
KSBF	67x3	Skip on Ready Flag.
KLSA	67x4	Load Status A from AC 4-11, clear AC, load complement Status A back into AC.
KSAF	67x5	Skip on any flag or error condition.
KGOA	67x6	Assert the contents of Status A, transfer data into the AC for a read, out of the AC into the Read/Write buffer for a write.
KRSB	67x7	Read Status B into AC 4-11.

Typical tape format is shown in the diagram below:



This diagram illustrates a 2 block file and a one block file written on tape.

Block format is shown in the diagram below:



The first word written into a block should be the byte count for that particular block.

When the software determines that it has written the last data byte into that block, the Read/Write CRC command must be given to cause the hardware calculated CRC character to be written into the last two bytes in the block. Conversely, when reading, when the last character has been read the Read/Write CRC command must again be given, after N+1 data flags, to let the hardware check the validity of the block read.

After the transport is told to go, the following flags will appear in approximately the times noted:

Data Flag:	One per word per 1.78 ms.
File Flag:	One per file, time depends on number of blocks per file.
Error Flags:	These will appear at the end of the block, if true, but timing error flag will be raised when it occurs.
Drive Empty:	The bit in Status B will be a one if there is no cassette in the drive. The skip and interrupt tests will occur if the drive is empty.
EOT/BOT:	This flag will be true whenever the optical sensors have detected clear leader/trailer.
Rewind:	Will remain true as long as drive is rewinding and is selected.
Write Lock Out:	Bit is true in Status B if tab is missing on cassette or the Drive is empty or Drive is rewinding. Bit will not interrupt or skip.
Ready Flag:	This bit is asserted in Status "B" when a drive is selected and is able to accept commands from the interface. This bit is false when commands are being executed.

### Cassette Format

A cassette is formatted so that it consists of a sequence of one or more files. Each file is preceded and followed by a file gap. (A gap in this sense is a set length of specially coded tape.)

- A file consists of a sequence of one or more records, separated from each other by a record gap. The first record of a file is called the file header record and contains information concerning the name of the file, its type, length, and so on. A record generally contains 128 (decimal) characters of information and there are approximately 600 records per cassette tape.
- Records consist of a sequence of one or more cassette bytes; a byte in turn consists of eight bits each. Characters and numbers are stored in bytes using the standard ASCII character codes and binary notation.
- The last file on a cassette tape is called the sentinel file. This file consists of only a file header record and represents the figurative end-of-tape. A zeroed or blank cassette tape is one consisting of only the sentinel file and an initial file gap.

The OMNIBUS Magnetic Tape Options include:

- a. The TD8-EM Dual DECTape Transport Control and TU56 Dual DECTape Transport
- b. The TM8-E DECmagtape Transport Control and TU10 DECmagtape transport

#### DECTapes

The DECTape unit can interface directly with the OMNIBUS via the TD8-E or to the External Bus via the TC08. The configurations are defined in the following table. For information on the TC08 Controller, refer to section 4 of this chapter.

Four basic DECTape configurations are identified in the following table.

SYSTEM DESIGNATION	DECTape	CONTROL	PREREQUISITE	REMARKS
None	TU56 (Dual Drive)	TC08	KA8-E* KD8-E PDP-8/E	Up to 4 Dual TU56's per control. (8 drive units)
None	TU56H (Single Drive)	TC08	KA8-E* KD8-E PDP-8/E	Up to 4 single DECTape drive units.
TD8-EM	TU56-M (Dual Drive)	TD8-E	PDP-8/E	Control plugs into OMNIBUS.
TD8-EH	TU56-MH (Single Drive)	TD8-E	PDP-8/E	Control plugs into OMNIBUS.

#### TD8-E DECTape Option

The DECTape system is a standard option for the PDP-8/E that serves as an auxiliary magnetic tape data storage facility. The DECTape system stores information at fixed positions on magnetic tape, as in magnetic disk or drum storage devices, rather than at unknown or variable positions, as in conventional magnetic tape systems. This feature allows replacement of blocks of data on tape in a random fashion without disturbing other previously recorded information. In particular, during the writing of information on tape, the system reads format (mark) and timing information from the tape and uses this information to determine the exact position at which to record the information to be written. Similarly, in reading, the same mark and timing information has a number of features to improve its reliability and make it exceptionally useful for program updating and program editing applications. These features are: phase or polarity sensed recording on redundant tracks, bidirectional reading and writing, and a simple mechanical mechanism util-

\* Magnetic tape options operated on the external bus of the PDP-8/E require the use of the KA8-E Positive I/O Bus Interface module and the KD8-E Data Break Interface module as prerequisites.

izing hydrodynamically lubricated tape guiding (the tape floats on air over the tape guides while in motion).

### Specifications

Tape Characteristics	Capacity—260 feet of $\frac{3}{4}$ inch, 1 mil Mylar sandwich tape, coated both sides. Reel diameter—3.9 inches Tape Handling—direct drive hubs and specially designed guides float the tape over the head. No capstans or pinch rollers are used. Speed— $97 \pm 14$ ips Density— $350 \pm 55$ bpi Information capacity—2702 <sub>s</sub> Blocks with 201 <sub>s</sub> 12-bit words per block (188,672 12-bit words) Tape Motion—bi-directional
Word Transfer Rate	33,300 3-bit characters per second
Addressing	Mark and timing tracks allow searching for a particular block by number in a forward or backward direction.
Tape Motion Timing	Start Time—150 msec $\pm$ 15 msec Stop time—100 msec $\pm$ 10 msec Turn around time—200 msec $\pm$ 20 msec
Mounting	TU56 Drive mounts in a standard 19 inch equipment rack
Size	10 $\frac{1}{2}$ inches high 19 inches wide $9 \frac{3}{4}$ inches deep } TU56 Drive 1 Quad Module } TD8-E Control plugs into OMNIBUS
Cooling	Internally mounted fan provided for TU56
Environmental Conditions	Temperature—40°F to 90°F Note: The magnetic tape manufacturer recommends 40-60% relative humidity and 60° to 80° as an acceptable operating environment for DEC-tape.

### Tape Compatibility

Tapes may be certified, programmed, read, modified, and rewritten interchangeably on either the larger automatic DECTape units (TC08/TC01) or on the TD8-E. DEC provides all the necessary subroutines and MAINDECs for the TD8-E; for example:

- Read/Write Subroutines
- Tape Certification Routine
- MAINDEC Maintenance Programs

- PS/8 Programming System (12K Minimum Configuration)
- A new 4K Keyboard Operating System with Program Directory, Line Editor, and PAL III\* Assembler.
- A DECTape Copy Program

\* (A Paper Tape Device is required; either ASR-33 or PC8-E, for input and output with PAL-III.)

### **TD8-E DECTAPE CONTROL**

The TD8-E is a low cost interface for the TU56 DECTape units. A TD8-EM consists of a TD8-E and one TU56-M Dual DECTape drive. The TD8-EH consists of a TD8-E and one TU56-MH Single DECTape drive.

The TD8-E is contained on a single quad Flip-Chip module which plugs directly into the OMNIBUS of the PDP-8/E. It is connected to the TU56 by a special interface cable (P.N. 7008447). It uses a standard TU56 with no modifications. The Read/Write Amplifiers (G888) must be plugged into the TU56 drives.

When reading, writing, or searching, the PDP-8/E acts as a controller for the DECTape. That is, all data transfers to and from the 8/E are through the AC in non-interrupt, non-data break mode. The PDP-8/E is completely committed to the tape operation and cannot perform any other functions until the tape operations have been completed.

Up to four TD8-E interfaces can be used with a PDP-8/E. Each TD8-E can drive either a single or dual transport. It is therefore possible to have eight DECTape drives connected to the PDP-8/E through four TD8-E's. When a dual transport is used on the TD8-E's, the first TD8-E will control units 0 and 1; the second TD8-E will control units 2 and 3; the third, units 4 and 5; and the fourth, units 6 and 7.

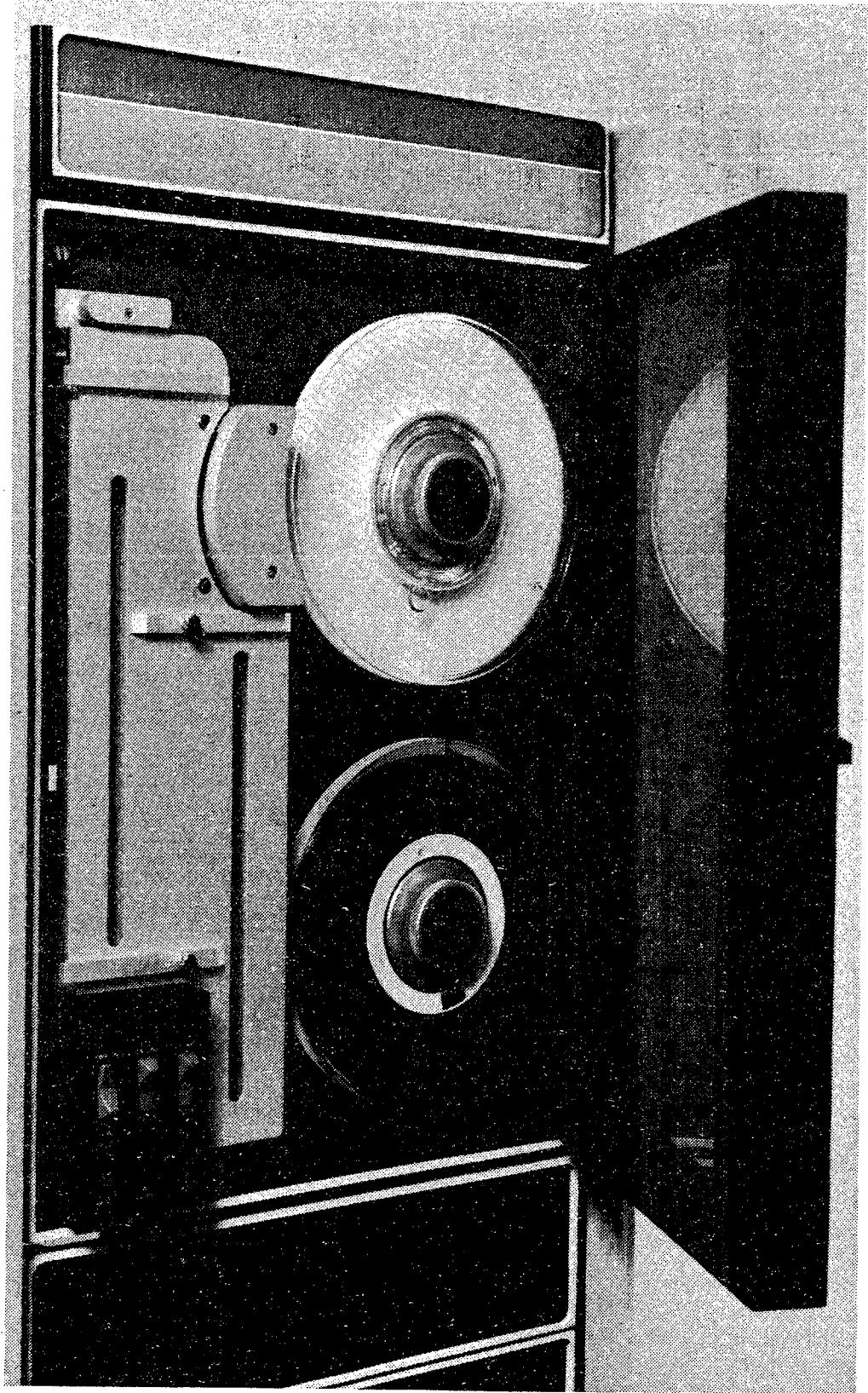
A comprehensive set of diagnostic routines is included with the TD8-E which checks all of its functions. The TD8-E is also supplied with subroutine software which search, read, and write PDP-8 compatible DECTapes. DECTapes written with the TC01 or TC08 control can be read with the TD8-E using this software and vice versa. Because of the close dependency of the hardware with the software, Digital Equipment Corporation will not guarantee operation of the TD8-E with any software other than that which is supplied by Digital Equipment Corporation.

The TD8-E was designed as a low-cost DECTape interface with limited functions. It is not a replacement for the TC08 which makes transfers of data direct to memory concurrent with CP operations. Its primary use is for library storage of programs and blocks of data. The TD8-E will, however, like the TC08, certify DECTapes by writing and verifying the mark and time tracks and block numbers.

Refer to Section 4 of this chapter for a detailed discussion of TU56.

### **TU10 DECmagnetic Tapes**

The DECmagtape can interface directly with the OMNIBUS via the TM8-E, or to the EXTERNAL BUS via the TC58. The configurations of both categories are defined in the following table. For information on the TC58 controller, refer to section 4.



TU10 DECmagtape

**DECmagtape Configurations**

SYSTEM OPTION	EQUIPMENT	NO. OF CHANNELS	DENSITIES (BPI)	TAPE SPEED (IPS)	OTHER INFORMATION
TM8-EA	TM8-E Control & TU10-EA(master)	9	800	45	Control plugs into OMNIBUS. TU10-EA contains a master and one slave. Up to 7 additional TU10 slaves may be added. 7 and 9 track TU10's can be mixed on the same system. For example, a 7 track master can be operated with a 9 track slave etc. The master consists of logic modules which plug into the TU-10 electronics.
TM8-FA	TM8-E Control & TU10-FA(master)	7	800/556/200	45	Same as above.
TC58 *	TC58 Control(master) & TU10-EE(slave)	9	800	45	DW08A I/O conversion panel, KA8-E Positive I/O Bus and KD8-E Data Break Interface are prerequisites. The master is contained with the TC58 controller. Up to 7 additional TU10 slaves may be added. 7 and 9 track TU10's can be mixed on the same system.
TC58 *	TC58 Control(master) & TU10-FE(slave)	7	800/556/200	45	Same as above..

\* Refer to Section 4 for TC58 Description

## **OMNIBUS DECmagtape Unit and Control Type TM8-E/F**

### **NOTE**

The following information is preliminary and is subject to change without notice. The reader should consult with the local DEC sales office.

The TM8E control provides the interface between the PDP-8/E and the TU10 master-slave magnetic tape transport system. The TU10 master can control 7 slaves; therefore the TM8-E is capable of controlling 8 transports.

The data transfer is via single cycle data break with a transfer rate of 36 KHZ. The transport operates at 45 ips and uses 7 channel formats at 200, 556 or 800 bpi or 9 channel format at 800 bpi.

The TM8-E contains six registers which are used to control the transports and report the status of the transports to the computer. The registers are loaded and read using IOT instructions which require no data break.

### **PROGRAMMING**

The following Instructions are used to program the TM8-E:

#### **Load Word Count Register (LWCR)**

Octal Code: 6701

Operation: Load Word Count Register and Clear the AC  
AC → WC, 0 → AC

#### **Clear Word Count Register (CWCR)**

Octal Code: 6702

Operation: Clear Word Count Register

#### **Load Current Address Register (LCAR)**

Octal Code: 6703

Operation: Load Current Address Register and Clear the AC  
AC → CA, 0 → AC

#### **Clear Current Address (CCAR)**

Octal Code: 6704

Operation: Clear Current Address

#### **Load Command Register (LCMR)**

Octal Code: 6705

Operation: Load Command Register and Clear the AC  
AC → CM, 0 → AC

#### **Load Function Register (LFGR)**

Octal Code: 6706

Operation: Load Function Register (GO bit) and Clear AC  
AC → Function Register 0 → AC

**Load Data Buffer Register (LDBR)**

Octal Code: 6707  
Operation: Load Data Buffer Register and Clear AC  
 $AC \rightarrow DB$ ,  $0 \rightarrow AC$

**Read Word Count Register (RWCR)**

Octal Code: 6711  
Operation: Clear AC and Read Word Count Register  
 $0 \rightarrow AC$ ,  $WC \rightarrow AC$

**Clear Transport (CLT)**

Octal Code: 6712  
Operation: Clear Transport

**Read Current Address Register (RCAR)**

Octal Code: 6713  
Operation: Clear AC and Read Current Address Register  
 $0 \rightarrow AC$ , then  $CA \rightarrow AC$

**Read Main Status Register (RMSR)**

Octal Code: 6714  
Operation: Clear AC and Read Main Status Register  
 $0 \rightarrow AC$ , then  $MS \rightarrow AC$

**Read Command Register (RCMR)**

Octal Code: 6715  
Operation: Clear AC and Read Command Register  
 $0 \rightarrow AC$ , then  $CM \rightarrow AC$

**Read Function Register & Status (RFSR)**

Octal Code: 6716  
Operation: Clear AC Read Function Register and Status 1  
 $0 \rightarrow AC$ , then Function and Status 1  $\rightarrow AC$

**Read Data Buffer (RDBR)**

Octal Code: 6717  
Operation: Clear AC and Read Data Buffer  
 $0 \rightarrow AC$ , then  $DB \rightarrow AC$

**Skip if Error Flag (SKEF)**

Octal Code: 6721  
Operation: Skip if error flag is set.

**Skip if Control not Busy (SKCB)**

Octal Code: 6722  
Operation: Skip if the control is not busy. The TM8-E becomes busy when a go is given to the transport and becomes not busy at MTTF.

### **Skip When Job Done (SKJD)**

Octal Code: 6723

Operation: Skip if the job is done (MTTF is set). The job done flag (MTTF) sets at the end (LRCS) of a Read, Read/Compare, Write File Mark, or Write operation, and at the end of a record, if an EOT, EOF or BOT was encountered or the WC overflowed during a space operation.

MTTF sets when a transport begins to do a rewind and a new transport may be selected, when a transport goes off-line following an off-line operation, and when a rewinding transport has reached BOT and is ready.

### **Skip When Tape Ready (SKTR)**

Octal Code: 6724

Operation: Skip if tape unit is ready (TUR is true).

### **Clear Controller and Master (CLF)**

Octal Code: 6725

Operation: Clear the Controller and Transport Master if TUR, if not clear MTTF, EF and Status Registers.

0 → Control Registers

Octal Code: 6726

Reserved for Maintenance

Octal Code: 6727

Reserved for Maintenance

### **Description of Registers**

6701 LWCR The 12 bit Word Count Register may be loaded from AC 0—11 any time the control is not busy. If the register is loaded during Control Busy, data reliability and tape compatibility cannot be assured. The Word Count must be loaded to the 2's complement of the number of words to be transferred or blocks to be spaced. The Word Count is incremented at TPI of the break cycle during Data Transfers and at LPCS during a space forward, and at the first word of a block during a Space Reverse.

Recommended block length is per USA Standards, Document USAS X3.22-1967. Recorded Magnetic Tape for information interchange (800 cpi, NRZ1).

6702 CWCR This IOT clears the Word Count Register and is essentially for maintenance use. It should not be used during Control Busy.

6703 LCAR The 12 bit Current Address Register may be loaded from AC 0—11 any time the control is not busy. It must be loaded to one less the Memory Address where the first data is taken or placed. If the Register is loaded during Control Busy, the following occurs:

- 1). In wrap around mode, function bit 6 = 0, location of the data transfer can not be assured within the selected memory field.
- 2). In EMA INC Enable mode, function bit 6 = 1, location of the data transfer can not be assured within the memory.

The Current Address Register is incremented at Break Request prior to the break cycle.

6704	CCAR	This IOT clears the Current Address Register and is essentially for maintenance use. It should not be used during Control Busy.
6705	LCMR	The Command Register can only be loaded from AC 0—11 during Control Not Busy. If the IOT is issued during Control Busy, an illegal function will be indicated and the current operation aborted. The transport may have to be rewound.

Bits

0, 1, 2      Unit selection: These determine which of the eight transports will be used.

0 0 0      Transport 0

0 0 1      Transport 1

0 1 0      Transport 2

0 1 1      Transport 3

1 0 0      Transport 4

1 0 1      Transport 5

1 1 0      Transport 6

1 1 1      Transport 7

Bit 3      Parity: 0 = Even  
                1 = Odd

Bit 4      Enable Interrupt on Error Flag

Bit 5      Enable Interrupt on MTTF (job done flag)

Bits 6, 7, 8      Extended Memory Address: These three bits determine which memory field the controller uses. The manner in which these bits are loaded depends upon the setting of the EMA Enable bit, Function Register bit 6.

Bits			
6 7 8			
0 0 0	Field 0		
0 0 1	Field 1		
0 1 0	Field 2		
0 1 1	Field 3		
1 0 0	Field 4		
1 0 1	Field 5		
1 1 0	Field 6		
1 1 1	Field 7		
Bit 9	Reserved for Future Use		
Bits 10, 11	Density: These bits select the density for the transports operation.		
10 11			
0 0	200 bpi	7 channel	
0 1	556 bpi	7 channel	
1 0	800 bpi	7 channel	

This also serves as a core dump mode. When issued to a 9 channel transport, data is written in 7 channel format and zero's are written in channels 0 and 1 on the tape.

1 1 800 bpi 9 channel

**6706 LFGR** The function register must be the last register to be loaded, since this register contains the GO bit.

Bit 0 Bit 1 Bit 2

0 0 0 Off Line: The selected transport is taken off-line and rewound to BOT. The MTTF is set when the transport responds to the function, the controller may then select and use another transport. The transport must be manually reset to the on-line state. The Word Count and Current Address Registers need not be loaded.

0 0 1 Rewind: The transport rewinds at high speed (150 ips) to BOT and stops. The MTTF is set when the transport responds to the function. The controller may then select and use another transport. Should the rewinding

transport be reselected, another MTTF will occur when the tape has stopped at BOT. The word count and Current Address Registers need not be loaded.

- |       |  |   |  |
|-------|--|---|--|
| 0     | 1  | 0 | Read: Data may be transferred from the tape to memory in the forward direction only. All registers must be loaded.   |
| 0     | 1  | 1 | Read/Compare: Tape data is compared to data in core memory. All registers must be loaded. If there is a comparison error, CA incrementation ceases, and the R/C error bit is set. Tape motion continues to the end of the record. The CA register contains the address of the word which failed. |
| 1     | 0  | 0 | Write: Data may be written on the tape in the forward direction only. All registers must be loaded. When the proper number of words have been written the transport writes the appropriate check characters to end the block.  |
| 1     | 0  | 1 | Write End of File (File Mark): The transport writes the file mark which consists of a one word block. The CA and WC registers need not be loaded.  |
| 1     | 1  | 0 | Space Forward: The transport moves forward at 45 ips the number of records specified by the WC register, or until a File Mark is read. If End of Tape is read space forward will stop at the first inter-record gap. The CA register need not be loaded.   |
| 1     | 1  | 1 | Space Reverse: The transport moves in the reverse direction at 45 ips the number of blocks specified by the WC or until a file mark or BOT marker is read. The CA register need not be read.   |
| Bit 3 | Extend Gap: This bit causes the transport to write with a minimum 3 inch gap between blocks.   |   |  |
| Bit 4 | Enable Check Characters: When this bit is set, it will allow the check characters to be read into the computer during a read function. When the word count overflows, this bit will allow at least one break during 7-track operation for the LPC or two breaks during 9-track operation for the CRC and LPCC. If a record length incorrect error occurs, the check character is considered bad and can not be used. This feature will be used primarily for 9-track error correction. |   |  |

	Bit 5	GO: This bit causes the controller to issue a GO command to the transport when the transport is capable of accepting it. The GO will not be issued if the specified function is illegal.
	Bit 6	<p>EMA INC Enable: If this bit is not set, the TM8-E will treat the extended memory the same way any other PDP-8 Family data break option would, i.e., each 4K block is used in a wrap around mode.</p> <p>If this bit is set, the extended memory will be treated as a continuous memory rather than as 4K blocks. When the last location in a field is reached, the EMA bits are incremented and the transfer continues in the next field. I.e.: If a word is placed in field 2, location 7777, the following word will be placed in field 3, location 0000 if the EMA increment bit is set. If it is not, the word will be placed in field 2, location 0000.</p> <p>In both modes of operation, the Current Address must be set to one less than the first location to be accessed. The 12 bit CA register and the 3 EMA bits are treated as one 15 bit register with the EMA bits most significant. I.e.: to access field 2, location 20, load EMA = 2, CA = 0017; to access field 2, location 0, load EMA = 1, CA = 7777 if in EMA increment mode; to access field 2, location 0, load EMA = 2, CA = 7777 if not in EMA increment mode.</p> <p>If memory field 7 is selected, the EMA cannot increment, but will wrap around in field 7 and an EMA 7 increment error will occur.</p>
6707	LDBR	Load Data Buffer Register and Clear the AC: This is primarily used for maintenance.
6711	RWCR	Clear the AC and Read The Word Count Register into the AC: This is primarily used for maintenance but also may be useful during Error Check routines.
6712	CLT	Clear Transport: This will clear the transport's master registers.
6713	RCAR	Clear the AC and Read Current Address Register: This is primarily used for maintenance but may also help during error check routines.
6714	RMSR	The 12 bit main status register is used to report the most important status of the transport and control to the computer. It may be read into AC 0-11 at any time.

Bit	
0	Error Flag: The Error Flag will interrupt the processor if the interrupt enable bit (CM04) is set. An illegal function or select error will set the Error flag immediately, halting data breaks and ending a Write operation. BOT, EOT, Read/Compare Error, Bad Tape, Lateral or Longitudinal parity errors, Record length incorrect, data late, or EMA 7 increment error will set the Error Flag after MTTF is set.
1	Rewind Status: The selected transport is rewinding.
2	Beginning of Tape (BOT): The BOT reflective strip is being sensed by the selected transport.
3	Select Remote: The selected transport is not on-line.
4	Parity Error: A longitudinal or lateral parity error has been detected.
5	File Mark (EOF): The selected transport detected a file mark during a space, read, or Read/Compare operation.
6	Record length incorrect: During a read or READ/Compare operation, the record length was different from the contents of the WC. The Word Count may be read to determine whether the record was long or short.
7	Data Request Late: The computer failed to service the break request before the next data transfer to or from the transport.
8	End of Tape (EOT): The EOT reflective strip has been sensed by the selected transport.
9	File Protect: The selected transport has a write lockout ring. No write functions will be accepted.
10	Read/Compare Error: A comparison failure occurred during the Read/Compare function. The CA contains the address of the bad word.
11	Illegal Function: <ol style="list-style-type: none"> <li>1. Issuance of LCMR, LFGR, or LDBR while the control is busy.</li> <li>2. Specifying any density but 800 bpi for a 9-channel transport.</li> <li>3. A space reverse function when the transport is at BOT.</li> </ol>

		4. Read, Read/Compare or Space Forward after a Write or WEOF command. 5. Changing to transports which is not ready. (TUR is false)
6716	RFSR	Clears the AC and Read the Function and 2nd status register.
	Bits 0-5	Function Register.
	6	Transport channel: The selected transport is 7-channel if the bit is 0, and 9-channel if it is 1.
	7	Bad Tape: Bad tape error indicates two or more consecutive characters missing, followed by data within the time of settling down. The CRC and LPCC will not cause bad tape errors.
	8	EMA 7 INC Error: This occurs if an attempt is made to increment the EMA from field 7 to field 0. The data will wrap around in field 7.
	9	Lateral Parity Error: A lateral parity error was detected.
	10	Reserved for Future Use.
	11	Longitudinal Parity Error: A longitudinal parity error was detected.
6717	RDBR	Clear the AC and Read data buffer into the AC. This is primarily used for maintenance.

### TU10 MASTER

The TU10 Master controls the function timing, write pulses, generation of all the check characters and checking of parity, and is capable of controlling 8 slaves on a common bus. The TU10 Master unit includes 1 TU10 slave.

### TU10 SLAVE

The TU10 DECMagtape Transport is a solid-state, magnetic tape handling device that controls tape motion and reads or records digital information on magnetic tape in industry-compatible formats.

The TU10 uses vacuum columns and a servo-controlled single capstan to control tape motion. The only contact with the oxide surface is the magnetic head and a rolling contact on one low-friction, low-inertia bearing. Dancer arms and pinch rollers, which shorten tape life and can cause errors, are not used in the TU10.

Tape transport commands can be issued manually from the TU10 control panel or remotely from the processor by means of the Controller. Indicators on both the transport and the controller indicate transport status.

Each tape transport consists of the TU10 cabinet, reel and reel motor control, capstan drive, and read/write components. The circuitry which controls the motion of the transport, generates the write pulses, timing gaps, parity, and check characters, and checks the parity is located in the Controller. These logic circuits may be shared by up to 8 TU10's.

## SPECIFICATION

### Power and Cabling

TU10 Power:                   tape transport power (reel motors and fans) provided by internal power supply in each transport

#### Cabling:

- a) 2 BC08P-15 to connect TC58  
    1 BC08N-15 to TU10
- b) 3 BC08N-15 to Bus TU10's together
- c) 2 BC08L-15 to connect TM8-E to TU10

### TU10 DECmagtape Transport

Mounting:                   mounts in standard H960-CA cabinet

Size:                       26 inches high, 19 inches wide, 25 inches deep

Cooling:                   internally mounted fans

Controls:                   front panel mounted

### Environmental Conditions

Temperature:               40°F to 100°F for system  
                              60°F to 80°F for magnetic tape

Humidity:                   20% to 95% (non-condensation) for system  
                              40% to 60% (non-condensation) for tape

### Power Input Requirements

TU10-EE, FE               115 Vac, 60 Hz at 14A

TU10-EH, FH               115 Vac, 50 Hz at 14A

TU10-EF, FF               230 Vac, 60 Hz at 7A

TU10-EJ, FJ               230 Vac, 50 Hz at 7A

### Local Transport Controls

PWR ON/PWR OFF           power control switch

ON-LINE/OFF-LINE          local or programmed operation

START/STOP                tape motion control

LOAD/BR REL               releases brake for loading

UNIT SELECT                selects unit for program control

FWD/REW/REV               tape direction control

### Tape Characteristics

Capacity: 2400 feet of  $\frac{1}{2}$ -inch, industry standard, 1-mil Mylar tape.

Reel Diameter: 10- $\frac{1}{2}$  inch standard reels

Tape Handling: direct-drive reel motors; servo-controlled single capstan; vacuum tape buffer chambers with constant tape winding tension

Tape Speed: 45 inches per second, reading and writing

Rewind Speed: 150 inches per second (approximately 3-minute rewind time for 2400-foot reel)

Packing Density: 7-channel—200, 556, and 800 BPI, selectable under program control  
9-channel—800 BPI

### Data Recording and Transfer

Recording Mode: NRZI, industry compatible

Magnetic Head: Dual gap, read-after-write

Data Transfers: Direct memory access (non-processor request)

Transfer Rate: 36,000 characters per second, maximum

BOT, EOT Detection: photoelectric sensing of reflective strip, industry compatible

Write Protection: write protect ring sensing

Data Checking: read-after-write parity checking; longitudinal redundancy check; cyclic redundancy check (9-channel only)

Interrecord Gap: reads tape with gap of 0.48 inches or more; writes tape with gap of 0.52 inches or more (compatible with industry standard)

### TU10 Models

No. of Channels	Type of Unit	115 VAC		230 VAC	
		60 Hz	50 Hz	60 Hz	50 Hz
9-channel	Master	TU10-EA	TU10-EC	TU10-EB	TU10-ED
	Slave	TU10-EE	TU10-EH	TU10-EF	TU10-EJ
7-channel	Master	TU10-FA	TU10-FC	TU10-FB	TU10-FD
	Slave	TU10-FE	TU10-FH	TU10-FF	TU10-FJ

NOTE: DECmagtape units TU20 and TU30 are also compatible with the TM8-E and TC58 controllers.

## LABORATORY PERIPHERALS

### AD8-EA Analog-to-Digital Converter

The AD8-EA converter is a 10-bit successive-approximation type with sample and hold circuits, conversion circuits, an input buffer, and control logic contained on two PDP-8/E modules. The converter can be used singularly with one channel input having an input range from -5 to +5 volts or can be used with an AM8-EA multiplexer to perform conversions for up to 16 channels having full-scale inputs from +1 to -1 volts. Analog inputs are connected to the module by H851 connectors from the multiplexer or by a shielded twisted pair from an external device. A 7008533 analog input cable is supplied with these options to facilitate input connection.

Operation of the AD8-EA converter is controlled by IOT instructions. A conversion is initiated by an ADST instruction, or from the Real Time Clock DK8-EP. An input starts the conversion and clears the A/D Done Flag. When the conversion is complete, the converter sets its A/D Done Flag. This flag is sensed by an ADSK instruction. If it set, the next instruction is skipped so that the 10-bit digital word can be transferred to AC2-11 by an ADRB instruction. Since the 10-bit word is in two's complement form, AC00 and AC01 copy AC02 (sign-extended format). The converter contains an interrupt enable flip-flop that is controlled by program instructions. When enabled, this flip-flop permits the converter to generate interrupt requests to the program interrupt facility upon completion of conversion.

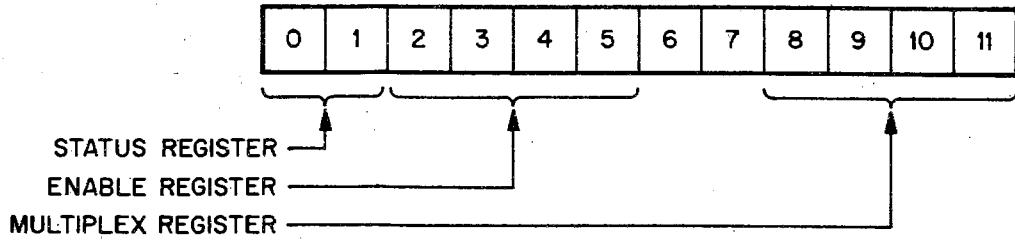
The converter also contains circuits for detection and sensing of a timing error. A timing error is defined as the receipt of a conversion request while a conversion is in progress. If this condition occurs, a Timing Error Flag is set. The Timing Error Flag is sensed by an ADSE instruction. An ADST or ADCL instruction clears this flag. This instruction also clears the A/D Done Flag so that another conversion can be implemented.

### Specifications

Input Voltage Range:	-5.0 to +5.0 Volts
Input Impedance:	Signal Return Input $\geq 10\text{ k}\Omega$ Signal Input $> 1\text{ M}\Omega$
Output Format:	Parallel: 10 bits right-justified and sign extended, two's complement
Resolution:	$\pm \frac{1}{2}$ LSB at 20 degrees C
Conversion Time:	20 $\mu\text{sec}$ nominal to complete conversion and set done flag.
Sample Acquisition Time:	Approx 3 $\mu\text{s}$

## PROGRAMMING

The following instructions are used to program the operations involving the AD8-EA and AM8-EA. Each instruction is completed in 1.2  $\mu\text{sec}$ .



Status Register	0—A/D Done Flag Status (Done = 1) 1—Timing Error Flag Status (Error = 1)
Enable Register	2—Enable Interrupt on A/D Done = 1 3—Enable Interrupt on Timing Error = 1 (note 1) 4—Enable External (e.g. clock) A/D start 5—Auto-Increment Mode (note 2)
Multiplex Register	8-11—Indicates current channel (0 — 17 <sub>8</sub> ) to be sampled by A/D

Note 1: The Timing Error Flag indicates that either an ADRB, and ADLM, ADST or external A/D start was attempted while a conversion was in progress. ADLM will be honored while an external A/D start or ADST will be ignored under this condition. ADLM or ADRB will cause an erroneous result to appear in A/D Buffer.

Note 2: When this bit is set, the occurrence of A/D Done = 1 will increment the Multiplex Register by 1. Incrementing past channel 17<sub>8</sub> will cause the MUX register to reset to channel 0.

Eight instructions are used to program the A/D Converter and Multiplexer. Each instruction is completed in 1.2μs and is defined as follows:

#### **Clear All (ADCL)**

Octal Code: 6530

Operation: Clears the A/D Done Flag and Timing Error Flags to ready the converter for another conversion. This instruction also clears the MUX and Status Register.

#### **Load Multiplexer (ADLM)**

Octal Code: 6531

Operation: Load Multiplexer register from AC8-11 and Clear AC.

#### **Start Conversion (ADST)**

Octal Code: 6532

Operation: Clear A/D Done and Timing Error Flags and Start A/D Converter. Channel to be converted is determined by MUX register.

**Read A/D Buffer (ADRB)**

Octal Code: 6533

Operation: Clear A/D Done Flag and load the contents of the A/D Buffer into ACO-11.

**Skip On A/D Done (ADSK)**

Octal Code: 6534

Operation: Skip the next instruction if A/D Done = 1. Do not clear flag.

**Skip On Timing Error (ADSE)**

Octal Code: 6535

Operation: Skip the next instruction if Timing Error Flag = 1. Do not clear flag.

**Load Enable Register (ADLE)**

Octal Code: 6536

Operation: Load Enable Register from AC2-5 and clear AC Register.

**Read Status Register (ADRS)**

Octal Code: 6537

Operation: Read A/D Status, Enable Register, and MUX into ACO-11.

**PROGRAMMING EXAMPLES****Normal Mode—**

The simplest method of programming the analog-to-digital converter is to have the program issue a start command, loop on the done flag until the conversion process is complete and the done flag is set to a "one", then the value of the converter's buffer is read into the PDP-8/E Accumulator. The program looks like this:

ADST	/Clear the ADC done flag and start conversion
ADSK	/Skip the next instruction when done
JMP -1	/Jump back one instruction
ADRB	/Read ADC buffer into AC

If the Analog-to-Digital Converter had been enabled to accept start pulses from an external device, such as a clock, then a timing error could occur. To check for this the following code could be added after the ADST command:

•	
•	
•	
ADSE	/Skip the next instruction on error
SKP	/unconditional skip
JMS*ERROR	/Go to error routine
•	
•	
•	

When the ADC is equipped with the multiplexer option, the channel to be sampled is selected prior to starting the conversion process. This is done using the ADLM command. For example a simple program to continuously "read" the value of one of the parameter knobs and display the digital value in the PDP-8/e accumulator look like this:

```
START, CLA           /clear the PDP-8/E accumulator
                      TAD CHN   /get the channel # (0-3 for knobs)
AGAIN,  ADLM          /load multiplexer from AC
                      ADST      /start
                      ADSK      /skip when finished
                      JMP .-1    /
                      ADRB      /Read ADC value
                      JMP AGAIN  /repeat process
```

#### Clock Mode—

In this special mode, an external event, usually the clock overflow starts the conversion process. This mode sample is taken at regular intervals as defined by the clock rate. The following example takes  $1000_{10}$  samples at the specified clock rate and stores them in memory.

```
•
•
•
INITIALIZE CLOCK AND ADC ENABLE REGISTER
•
•
•
START, CLA
TAD NUMBER
DCA COUNTER
TAD ADDRESS
DCA POINTER      /POINTER IS AN AUTO-INDEX REGISTER

ADLOOP ADSK
JMP .-1
ADRB
DCA I POINTER
ISZ COUNTER
JMP ADLOOP
•
•
•
NUMBER ,—1750s      /# of samples (100010 in this case)
COUNTER , 0
ADDRESS , n-1
POINTER , 0      /Beginning of table
```

### **Fast Sample Mode—**

In fast sample mode the PDP-8/E processor is allowed to proceed while the called for conversion is still in process. The conversion still requires its full time to complete but since the order of events has changed, the sample may be taken at the full speed of the Analog-to-Digital Converter. The following example demonstrates this:

```
ADLOOP , ADSK
        JMP .-1
ADSTART , ADRB
        ADST
        DCA I POINTER
        ISZ COUNTER
        JMP ADLOOP
```

To make use of this program it is necessary to enter the program at ADSTART.

### **AM8-EA 8-Channel Analog Multiplexer**

The AM8-EA is an 8-channel analog multiplexer designed for use with the AD8-EA A/D Converter. The multiplexer accepts bipolar analog input having a full-scale range of  $\pm 1V$  and converts these inputs into a full-scale  $\pm 5.0$  volt output supplied to the AD8-EA Converter.

The AM8-EA consists of multiplexer switches and scaling amplifiers for 8 analog channels. The AD8-EA can be expanded to 16 channels in eight-channel groups by adding an AM8-EA 8-channel Multiplexer Module. Multiplexer operation is controlled by the AD8-EA IOT instructions. These instructions and the associated multiplexer control provide the capabilities for random or sequential selection of channels, combining the operation of the A/D converter with that of the multiplexer. Two programmable address modes are provided: autoincrement or non-autoincrement. The AM8-EA is set to nonautoincrement mode when INITIALIZE is generated.

In the autoincrement mode, channel addresses are incremented automatically at the completion of a conversion by an A/D Done Flag from the converter. The program specifies the first address of interest by issuing an ADLM instruction and then can issue an instruction to start an A/D conversion. Upon completion, the A/D Flag increments the multiplexer channel address for the next sample. This process can continue until the AUTO MODE flip-flop is reset.

### **Specifications**

<b>Input Voltage</b>	Bipolar, $\pm 1V$
<b>Input Impedance</b>	70K ohms $\pm 2\%$ , shunted by 300 pf
<b>Output</b>	Bipolar, $\pm 5V$ full scale
<b>Common Mode Rejection</b>	Greater than 25 dB, 35dB typical

Overload Protection	$\pm 67V$ from fault line (indefinitely)
Overload Recovery Time	8 $\mu s$
Frequency Response	Flat from 0 to 30 KHz, —3dB 60 KHz
Leakage Current	Negligible at 70K ohms impedance
Long Term Stability (1 hour)	1% for $\pm 30^\circ C$

#### **DR8-EA 12-Channel Buffered Digital I/O**

The DR8-EA Digital I/O can be used to control 12 discrete digital switching circuits located externally, and can be used to accept 12 discrete digital inputs from external sources. The unit consists of IOT control logic; a 12-bit input buffer, a 12-bit output buffer, and 3 multiplexer ICs that control the flow of data for input and output operations. All circuits are TTL logic and are mounted on a single PDP-8/E module which plugs into the OMNIBUS. The standard TTL outputs are connected to the external load via two H854 connectors on the module. Inputs from external sources are also connected to the DR8-EA using H854 connectors.

Data outputs are updated under program control. Standard output drivers have a TTL 30-unit load capability. For an output function the computer issues a DBRO, DBSO, or DBCO instruction. For DBSO instructions, only logical ones in the AC are loaded into the output register; AC bits containing logical ZEROS do not affect output register bits. For DBCO instructions, logical ONE's in AC result in logical ZEROS in corresponding bits of the output register. For DBRO instructions, the contents of the output register is transferred into the AC register.

Data inputs must be TTL compatible, have negative transition to .8V or less for a logical ONE, and have a pulse duration of greater than 50ns. Pulse rise and fall time should be less than 150ns for maximum noise immunity. In one mode of operation, the input register bits, once set by the data inputs, remain set until read by a DBRI instruction. In the second mode of operation, the input can be placed directly through gating on the bus, and will remain as long as the input remains. The DBRI instruction is also used to read the input data. When this IOT is issued, the content of the input register is gated to the AC via the OMNIBUS. A DBCI instruction, used with DBRI instruction, enables inputs that occurred too late to be read by the next DBRI instruction. Correct usage of this feature results in "zero dead time" for events. Any of the input lines can cause an interrupt if the proper jumpers are selected. The interrupt facility can be enabled by instruction DBEI and disabled by instruction DBDI.

A maximum of 8 DR8-EA options can be used. Each device selector code is determined by the user by means of jumpers. Device codes 50 to 57 are legal; however, the DR8-EA normally comes with device code 50 installed.

The DR8-EA is contained entirely on one PDP-8/E module.

## Specifications

<b>Input Format</b>	Parallel, 12 bits.
<b>Input Levels</b>	Compatible TTL levels. Input circuitry switches at 0.8 to 2.4 Volts, and is protected to allow input swings as positive as +20 Volts and as negative as -15 Volts.
<b>Input Connections and Pulse Width</b>	Inputs to inverter buffers are normally held high by resistors. A negative transition to 0.8V or less will cause the input to become a logical ONE. Optional inputs bypass the flip-flop for direct interrogation of input line status.
<b>Output Format</b>	Parallel, 12 bits.
<b>Output Levels</b>	TTL-compatible levels capable of driving 30 unit loads. Output lines are protected from short circuits to ground.
<b>Environmental:</b>	0°C to 55°C 10% to 90% relative humidity (non-condensing)
<b>Power Requirements:</b>	+5.0 volts, 2.25 amps (worst case)

## Programming

The following instructions are used for DR8-EA operation. The X refers to a jumper selectable code. However, the DR8-EA normally comes with code 50 installed.

### Disable Interrupt (DBDI)

Octal Code: 65X0  
Operation: Disable all interrupts that are caused by a logical ONE on the input.

### Enable Interrupts (DBEI)

Octal Code: 65X1  
Operation: Set Interrupt Enable Flip-Flop. This tests the IN FLAG and causes an Interrupt Request if IN FLAG equals ONE.

### Skip on Flag (DBSK)

Octal Code: 65X2  
Operation: Tests the IN FLAG. If the Flag is a ONE, the next sequential memory location is skipped.

### Clear Selective Input Register (DBCI)

Octal Code: 65X3  
Operation: ONE's in the AC clear respective bits in the Input Register.

### Transfer Input to the AC(DBRI)

Octal Code: 65X4  
Operation: Transfers the complete 12-bit Input Register to the AC.

**Clear Selective Output Register (DBCO)**

Octal Code: 65X5

Operation: ONE's in the AC clear the respective bits in the Output Register.

**Set Selective Output Register (DBSO)**

Octal Code: 65X6

Operation: ONE's in the AC set the respective bits in the Output Register.

**Transfer Output to AC (DBRO)**

Octal Code: 65X7

Operation: Transfer the complete 12-bit Output Register to the AC.

**Programming examples**

To clear all registers

CLA CMA	/Set AC to 7777
DBCI	/set all input bits to zero
DBCO	/set all output bits to zero
DBDI	/disable interrupts

To service the occurrence of events

START,DBSK	/has event happened
JMP-1	/no, check again
DBRI	/yes, read register
DBCI	/clear way for reoccurrence
SPA	/was it event 0
JMS SUB0	/yes, go service 0
RAL	/no, shift left
SPA	/was it event 1
JMS SUB1	/yes, go service 1
RAL	/no, shift left
...	
RAL	/no, shift left
SPA	/was it event 11
JMS SUB11	/yes, go service 11
JMP START	/no, go wait for another event
SUB0, 0	/return location
DCA SAVE	/save for further checking

(Service event)

CLA	
TAD SAVE	/get for further checking
JMP I SO	/go check further

S1,

### **Interface**

The DR8-EA interfaces to the PDP-8/E OMNIBUS by plugging directly into the bus.

Interface to the outside world is by two (2) edge connectors on the M863 module. Signals leaving the board (12 bits parallel) are high (+3 volts) for a logical false and ground (0 volts) for a logical true. Each output line has approximately 20 milliamperes of drive (high level) and 20 milliamperes of sink (low level). Output levels remain fixed except when changed by the processor.

Signals entering from the "outside world" must be TTL in nature. The input represents approximately two (2) unit loads. When jumpered for "edge detection" a negative going edge (3 volts to 0 volts) is sensed. The signal must remain low (0 volts) for at least 50 NS. When sensing for an external level (jumpered so as to bypass the "flop") ground (0 volts) represents a logical true and a high (+3 volts) represents a logical false. With all bits jumpered this way the option represents a 12-bit parallel input register rather than an event detector.

An optional means of interfacing to the DR8-EA is available by using two (2) BC08J-X cables. Each cable (ribbon type) is terminated by a Berg type connector on one end (for interfacing to the DR8-EA module) and a standard DEC flip-chip on the other. One cable is used for the input and the other for output. Two BC08J-10 10-foot cables are supplied as part of the DR8-EA.

### **Cable Descriptions.**

The BC08S-1 cable is used to jumper the input to the output for diagnostic purposes. It is part of the DR8-EA option. If the user desires interface cables, the following can be purchased:

The BC08J cable consisting of the 1210091 connector, cable and the M953 module, and is available in several standard lengths. A laboratory mounting panel is available for this option.

### **Jumper Descriptions**

The chart defined below will enable the user to change the IOT device code by changing the jumper across the specified split lug.

device selector	jumper		
(normal conf) 50	6H	7H	8H
51	6H	7H	8L
52	6H	7L	8H
53	6H	7L	8L
54	6L	7H	8H
55	6L	7H	8L
56	6L	7L	8H
57	6L	7L	8L

The normal configuration will be factory installed with device selector code 50.

The input jumpers will be factory installed with A jumper (edge triggered flip-flop). To change to level enables, use jumper B. The A,B, lugs are on all 12 bits.

Jumpers will also be provided to insulate the inputs from the interrupt and skip circuitry.

J2 — Input		J1 — Output	
D	— Bit 0	D	— Bit 0
F	— Bit 1	F	— Bit 1
J	— Bit 2	J	— Bit 2
L	— Bit 3	L	— Bit 3
N	— Bit 4	N	— Bit 4
R	— Bit 5	R	— Bit 5
T	— Bit 6	T	— Bit 6
V	— Bit 7	V	— Bit 7
X	— Bit 8	X	— Bit 8
Z	— Bit 9	Z	— Bit 9
BB	— Bit 10	BB	— Bit 10
DD	— Bit 11	DD	— Bit 11

#### Pin Connections

The output and input pins corresponding the AC bit enabled on the DR8-EA are as follows:

#### Input and Output End Pins (BC08J)

Bit 0	— B1	Gnds	A1, C1, F1, K1,
Bit 1	— D2		N1, R1, T1, C2,
Bit 2	— D1		F2, J2, L2, N2,
Bit 3	— E2		R2, U2
Bit 4	— E1		
Bit 5	— H2		
Bit 6	— H1		
Bit 7	— K2		
Bit 8	— J1		
Bit 9	— M2		
Bit 10	— L1		
Bit 11	— P2		

#### LABORATORY MOUNTING PANEL

The laboratory peripheral panel is designed for compact yet versatile packaging of modular accessory equipment for laboratory environments. The panel is a 19-inch rack-mounted unit with H945 panel mounting frame and housing that accepts plug-in type modules or module panels. Modules can be single-width, double-width, or other multiples of single-width, and may contain a printed circuit card mounted on the vertical dimension. Controls and input/output connectors for peripheral equipment are mounted on the module front panel. Modules or module panels are attached to the panel frame using one fastener at the top and bottom of the module panel. Mating connectors are supplied with all panels.

#### The following options are available:

- H945      Housing (Rack Mountable Chassis) for mounting laboratory peripherals including space for mounting 11 panel units; 5 single panel units; 3 double panel units, and a single 1½ panel unit filler panels.

H945-AA	Table-top version.
H945-AB	Rack-mounted version.
DK8-EF	Optional panel for type DK8-EP Real Time Clock. Contains input/output jacks for the 3 Schmitt trigger inputs (event 1, 2 and 3) to the DK8-EP. Also includes potentiometer controls for Schmitt trigger threshold adjustment and jacks for both external frequency input and clock overflow output.
DK8-ES	DK8-EP with DK8-EF.
DR8-EB	Panel for DR8-EA 12-Channel Digital Buffered I/O containing one connector for digital input and one for digital output.
DR8-EC	DR8-EA with DR8-EB.
AM8-EC	Analog input panel—16-channel A/D panel used for AM8-EA multiplexer inputs. Panel contains four 3-conductor phone jacks and four 10-turn vernier controls and 2 connectors. Panel requires 3 single-panel-unit widths.
AM8-ED	Simple analog input panel 16-channel A/D panel used for AM8-EA multiplexer inputs. Panel contains two connectors and requires a single-panel-unit width.
AM8-ES	AM8-ED with AD8-EA.
VM03	Model 602 Tektronix Oscilloscope Mounting Hardware.

## **DB8-E INTERPROCESSOR BUFFER**

The DB8-E interprocessor buffer allows two PDP-8/E's to transfer data between themselves or it may be used single ended as a data path between a PDP-8/E and user designed logic.

Device codes are jumper selectable between 50 and 57 allowing up to 8 DB8-E's to be connected to one PDP-8/E. The PDP-8/E's may be interconnected at distances up to 100 feet apart by means of two (2) BC08-R type cables.

All logic is mounted on a single QUAD size board which plugs directly into the OMNIBUS. Two (2) 40 pin connectors type H854 mounted on the module receive cable type BC08-R or BC08-J. On the terminal end of the cable, connector type H856 is provided.

### **SPECIFICATIONS**

Maximum Transfer Rate	One 12-bit word at a maximum rate of approximately 5K Hz.
Physical Characteristics	The entire option is contained on one 8½" PDP-8/E QUAD module.
Temperature Operating Range	32°F to 131°F (0°C to 55°C)
Power Requirements	+5 volts at 600ma.
Data Format	12 parallel bits in and 12 parallel bits out.

### **PROGRAMMING**

The following instructions are used for the DB8-E operation:

#### **Skip on Receive Flag (DBRF)**

Octal Code: 65X1  
Operation: Skip if the Receive Flag equals one.

#### **Read Incoming Data (DBRD)**

Octal Code: 65X2  
Operation: Read the Incoming Data into the AC and clear the Receive Flag.

#### **Skip on Transmit Flag (DBTF)**

Octal Code: 65X3  
Operation: Skip if the Transmit Flag equals one.

#### **Transmit Data (DBTD)**

Octal Code: 65X4  
Operation: Transfer the contents of the AC Register to the Transmit Buffer. Transmit Data and set the Transmit Flag.

#### **Enable Interrupt (DBEI)**

Octal Code: 65X5  
Operation: Enable the Interrupt Request Line.

#### **Disable Interrupt (DBDI)**

Octal Code: 65X6  
Operation: Disable the Interrupt Request Line.



# 7.

## external i/o options

The external bus interface options enable the PDP-8/E user to interface PDP-8/I and PDP-8/L type peripherals (such as mass storage devices, data acquisition, and control equipment) with the PDP-8/E. It also permits user-designed equipment to be interfaced with the 8/E external bus through the use of a general-purpose interface unit. A type KA8-E Positive I/O Bus Interface unit is required for any type of peripheral connected to the external bus. A type KD8-E Data Break Interface unit is required for each external peripheral that uses the data break facilities of the computer. These interface units and the BB08-P General Purpose Interface are described below. The detailed relationships of programmed I/O transfers and data break transfers are described in Chapter 10.

### **KA8-E Positive I/O Bus Interface**

The KA8-E option enables the PDP-8/E user to interface PDP-8/I and PDP-8/L type peripherals with a PDP-8/E. This option converts OMNIBUS signals into positive programmed I/O bus signals used by PDP-8/I and PDP-8/L type peripherals. For example, 8/I and 8/L type peripherals require IOP pulses to perform their operations. The OMNIBUS does not generate internal IOP pulses, but does provide signals (MD bits 9-11) that can be converted to IOP pulses. Other signals normally required for programmed I/O transfers are also available on the OMNIBUS. The KA8-E merely buffers these signals and makes them available to the external bus at the correct time. Similarly, the KA8-E buffers peripheral inputs and makes them available to the OMNIBUS. A detailed description of the external bus interface, including signals, levels, timing relationships, and other interface data, is provided in Chapter 10.

Only one KA8-E can be used per machine. This module is required both for programmed I/O transfers and for external bus data break transfers. The KA8-E is also required whenever the BB08-P General Purpose Interface option is used, and when user-designed or user-installed logic is to be connected to the external bus. The KA8-E Positive I/O Bus Interface is contained on one PDP-8/E module that plugs into the OMNIBUS.

See Chapter 10 for details of interfacing with the external bus.

### **BB08-P General Purpose Interface Unit**

The BB08-P General-Purpose Bus Interface provides the PDP-8/E user with the capability of interfacing user-designed or user-installed logic with the PDP-8/E external bus. (The KA8-E Positive I/O Bus Interface module is a prerequisite for using the BB08-P.)

The BB08-P can interface one receive (input) and one transmit (output) device, or two receive, or two transmit devices, and control related

transfers from program instructions. In addition, the unit can supply operating power for the user's device.

The BB08-P logic is housed in one prewired DEC type H943 Mounting Panel with a self-contained Type H716 Power Supply. There are 34 module sockets not used by option modules; thus, these sockets are available for user logic modules. The spare sockets, located in two adjacent rows, can accommodate 34 single-height modules, or 17 double height modules or combinations.

The basic data format for transfers is 12-bit parallel. The organization of fields within this format is at the user's discretion; however, user logic must operate according to the following rules:

- a. Data user logic to computer, via BB08, must take inverted positive-bus form:  
OV (L) = logic true (1);  
+3V (H) = logic false (0).
- b. Data from computer to user logic, via BB08, must be accepted in true positive-bus form:  
+3V (H) = logic true (1);  
OV (L) = logic false (0).
- c. User logic must provide pulses to the BB08 to set the Transmit and Receive flags as required. These pulses must take the form of OV to +3V transitions of not less than 100 ns duration. Rise and fall times of these pulses should be 150 ns or less.

The user may, at his discretion, use any or all of the following spare logic gates on modules of the BB08 option:

- a. Four C/D flip-flops on the M216 or M206 Module at panel location A06.
- b. Two TTL logic inverters on the M111 Module at panel location A07.
- c. Eight open-collector bus drivers on the M623 Module at panel location 805.
- d. Eight TLL two-input NAND gates on the M113 Module at panel location B06.

The BB08-P receive section consists of 12 level-converter gates, a device selector, and Receive Flag circuits. For transfers to the computer, the user device sets the Receive Flag to initiate a program interrupt for servicing the device. The computer then interrogates the skip chain by issuing Skip-on-External-Flag instructions. When a skip instruction with 37 (octal) is detected by the BB08, this device returns a skip pulse that causes a conditional jump in the computer. The program then clears the Receive Flag by issuing an IOT 6372 (octal) and transfers the input word to the computer with a 6374 (octal) instruction.

The BB08-P transmit section consists of 12 level-converter gates, a device selector, and Transmit Flag circuits. The most important difference between transmit and receive logic is that transfer from the buffered accumulator bus to the user's device is enabled whenever the BB08 device selector decodes 636X (octal).

## Specifications

Data Format	12-bit parallel. Can be discrete bits or any organization of fields.
Receive/Input	TTL compatible of the inverted positive bus form: 0V (L) = logic 1 +3V (H) = logic 0
Transmit/Output	TTL compatible of the true positive bus form: +3V (H) = logic 1 0V (L) = logic 0
Power Supply Outputs (Available for User Logic)	3A at +5V 1.3A at + 15V
Power	115VAC, 60 Hz, 1A

## Programming

The following instructions are used for BB08-P operation:

### Skip On Transmit Flag (GTSF)

Octal Code: 6361  
Execution Time: 2.6  $\mu$ s  
Operation: Skips the next instruction if the Transmit Flag is set.

### Clear Transmit Flag (GCTF)

Octal Code: 6362  
Execution Time: 2.6  $\mu$ s  
Operation: Resets the Transmit Flag.

(User Designated)

Octal Code: 6364  
Execution Time: 2.6  $\mu$ s  
Operation: This instruction is not used by BB08-P; however, the BB08-P decodes this IOT to make IOP4 available to user. User can use the IOP4 pulse to strobe data into his device.

### Skip On Receive Flag (GRSF)

Octal Code: 6371  
Execution Time: 2.6  $\mu$ s  
Operation: Skips the next instruction if Receive Flag is set.

### Clear Receive Flag (GCRF)

Octal Code: 6372  
Execution Time: 2.6  $\mu$ s  
Operation: Resets the Receive Flag.

### **Read Device Buffer (GRDB)**

Octal Code:

6374

Execution Time:

2.6  $\mu$ s

Operation:

Transfers data from receive device to AC0-11.

### **KD8-E Data Break Interface**

The KD8-E Data Break Interface option provides the PDP-8/E user with the one- and three-cycle data break facilities of the computer. Each KD8-E implements one of the 12 available data break channels of the PDP-8/E.

Each KD8-E contains the hardware to implement one standard data break channel and logic for establishing multiplexing priority between break devices. The KD8-E option is contained on one PDP-8/E module that plugs into the OMNIBUS.

Data break operations and the relationships of the KD8-E for these operations are described in detail in Chapter 10. Transfer time is 1.4 microseconds (715 kHz) for the single-cycle data break devices and 4.2 microseconds for 3-cycle data break devices.



### **DF32-D DEC Disk File & Control & DS32-D DEC Disk File Expander**

The DF32-D Disk File is a fast, low-cost, random-access, bulk-storage device and control for use with the PDP-8/E computer. [When the DF32-D is used with the PDP-8/E, the KD8-E Data Break interface and the KA8-E Positive I/O Bus interface are also required.] Operating through the three-cycle Data Break Facility, the DF32-D provides 32,768 13-bit words (12 bits plus parity) of storage, and is economically expandable to 131,072 words when using the DS32-D Expander Disk.

Transfer rate of the DF32-D is 32 or 64  $\mu$ s per word (optional when timing track is written); average access time is 16.67 ms for 60 Hz power (20 ms with 50 Hz power).

Two basic assemblies make up the DF32-D; the storage unit with read/write electronics and computer interface logic. The storage unit contains a nickel-cobalt-plated disk, driven by a hysteresis synchronous motor. Data is recorded on a single disk surface by 16 fixed-position read/write heads.

Disk motor and shaft, read/write data heads, and timing and address heads are mounted on a 19-inch relay rack assembly, which permits easy access to the unit by sliding the unit in and out of a standard Digital Equipment Corporation cabinet.

The DS32-D Extender Disk File is also a slide-mounted assembly with a storage element and read/write electronics. Information transfers are made via the DF32-D logic, and are controlled by the DF32-D.

### **Specifications**

Storage Capacity	32,768 13-bit words; expandable to 131,072 words in increments of 32,768 words, using DS32-D.	
Data transfer rate	60 Hz power	50 Hz power
Average access time	32 (64) $\mu$ s per word	39 (78) $\mu$ s per word
Write lock switches	16.67 ms                    20.0 ms Inhibit writing on lower and/or upper 16K or any 32K disk surface; may be used to inhibit one or more 32K disks in an expanded configuration.	
Addressing Scheme	Random or absolute addressing from 0 to 32K words with variable block sizes from 1 word to 4096 words.	
Data assembly	Read/write on disk is serial, with external transfer parallel by word.	
Data Availability	16 $\mu$ s (48 $\mu$ s with alternate timing track) from the time word is assembled until new word starts to shift into assembly register. (A similar timing condition exists during the write operation.)	
Data tracks	16 per disk, 2048 words per track	
Recording method	NRZI	
Density (max)	1100 BPI	
Timing tracks	2 plus 2 spare	

Size	10-1/2 in. high and 23-5/8 in. deep in a standard 10-in. rack.		
Heat Dissipation	1700 Btu/hr.		
Data Transfer Path	3-cycle Break	Address Locations 7750 Word Count 7751 Memory Address	
Program Interrupt	Data Transfer-completion flag and/or non-existent disk.		
Write lock Switches	Inhibit write only on lower or upper 16K or both on one or more discs.		
Select Switches	Rotary Switches to select disk unit number.		

### Programming

The following instructions operate the disk system:

#### Clear Disk Memory Address Register (DCMA)

Octal Code: 6601  
 Execution Time: 2.6  $\mu$ s  
 Operation: Clears disk memory address register, parity error, and completion flags. This instruction also clears the disk memory request flag and interrupt flags.

#### Load Disk Memory Address Register and Read (DMAR)

Octal Code: 6603  
 Execution Time: 3.6  $\mu$ s  
 Operation: Loads the content of the AC into the disk memory address register and clears the AC. This IOT initiates readings of information from the disk into the specified core location. Clears parity error and completion flags.

#### Load Disk Memory Address Register and Write (DMAW)

Octal Code: 6605  
 Execution Time: 3.6  $\mu$ s  
 Operation: Loads the content of the AC into the disk memory address register and clears the AC. This disk then begins to write information into the disk from the specified core location. Clears parity error and completion flags. Data break must be allowed to occur within 33  $\mu$ s (66  $\mu$ s) after issuing this instruction

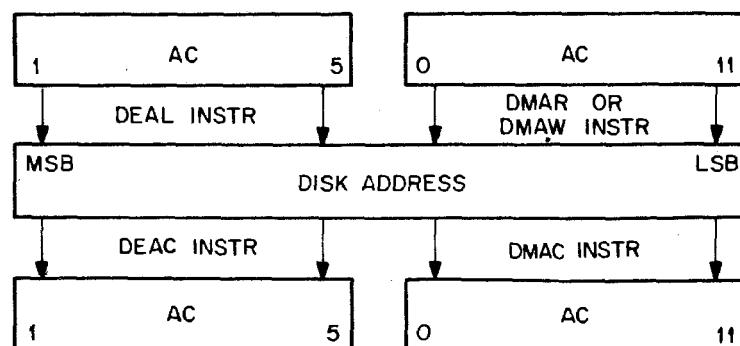


Figure 7-10 Address Words

### **Clear Disk Extended Address Register (DCEA)**

Octal Code: 6611  
Execution Time: 2.6  $\mu$ s  
Operation: Clears the disk extended address and memory address extension register.

### **Skip on Address Confirmed Flag (DSAC)**

Octal Code: 6612  
Execution Time: 2.6  $\mu$ s  
Operation: Skips next instruction if address confirmed flag is a one. Flag is set for 16  $\mu$ s whenever the address on the disk equals the contents of the disk address registers. Clears the AC.

### **Load Disk Extended Address (DEAL)**

Octal Code: 6615  
Execution Time: 3.6  $\mu$ s  
Operation: Clears the disk extended address and memory address extension registers and loads them with the track address data held in the AC. ORs the contents of these registers, plus the photocell mark and three error flags, into the AC. (See DEAC instruction.)

### **Read Disk Extended Address Register (DEAC)**

Octal Code: 6616  
Execution Time: 3.6  $\mu$ s  
Operation: Clears the AC, then loads the contents of the disk extended address register into the AC to allow program evaluation. Skips the next instruction if address confirmed flag is a one.

#### **NOTE**

Write lock switch status is true only when disk unit contains a write command. The nonexistent disk condition will appear following the completion of a data transfer during read, where the address acknowledged was the last address of a disk and the next word to be addressed falls within a nonexistent disk. The completion flag for this data transfer is set by the nonexistent disk condition 16  $\mu$ s after the data transfer.

### **Skip On Zero Error Flag (DFSE)**

Octal Code: 6621  
Execution Time: 2.6  $\mu$ s  
Operation: Skips the next instruction if parity error, data request late, and write lock switch flag are all zero. Indicates no errors.

### Skip on Data Completion Flag (DFSC)

**Octal Code:**

6622

### **Execution Time:**

2.6  $\mu$ s

**Operation:** Skips the next instruction if the completion flag is a one, indicating data transfer is complete.

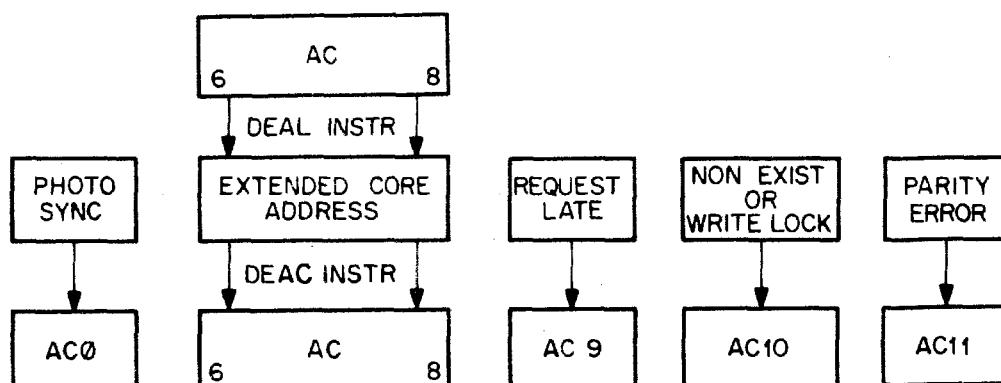
#### **Read Disk Memory Address Register (DMAC)**

**Octal Code:**

6626

### **Execution Time:**

**3.6  $\mu$ s**  
Clears the AC, then loads the contents of the disk memory address dress register into the AC to allow program evaluation. During read, the final address will be the last one transferred.



**Figure 7-11**

Three-cycle data break locations: Work Count address is 7750 (field 0). Current Address is 7751 (field 0).

Three maintenance IOTs are also used by the DF32-D. These IOTs are used to simulate certain pulses within the disk control for static logic tests. Since they all use device code 63, this code should not be used by other peripheral devices when a DF32-D is part of the system.

## **NOTE**

For the DEAL and DEAC instructions, refer to the diagrams shown below:

Field Bits 6-8 + Cell 7751 → Current Address  
 (DEAL, DEAC Inst.) (Current Address) (Memory) Address  
 (15 Bit)

The computer can handle 12 bits; therefore, the high order bits for disk and memory address are manipulated by the DEAL and DEAC instructions. Low order bits are manipulated in the AC.

## **TYPE RF08 DISK FILE AND CONTROL AND TYPE RS08 EXPANDER DISK FILE**

The RF08 control and the RS08 disk combine to provide fast, low-cost, random access, bulk storage for the computer. One RF08/RS08 provides 262,144 13-bit words of storage. Up to four RS08 disks can be added to the RF08 control for a total of 1,048,576 words of storage. Data is recorded on a single disk surface by 128 fixed read/write heads.

Data transfer is accomplished through the three-cycle break system of the computer and its associated required options, which are the same as for the DF32/DS32 system. Fast track-switching time permits spiral read or write. Data may be read or written in blocks of from 1 to 4096 words. Transfers across disks are handled automatically by the control unit.

### **RF08/RS08 Specifications**

<b>Disks</b>	Four RS08s may be controlled by one RF08 for 1,048,576 words.	
<b>Storage Capacity</b>	Each RS08 stores 262,144 13-bit words (12 plus one even parity bit)	
<b>Data Transfer Path</b>	<b>3-Cycle Break</b>	<b>Address Locations</b> 7750 Word Count 7751 Current Address
<b>Data Transfer Rate</b>	60 Hz Power 16.0 $\mu$ s per word	50 Hz Power 19.2 $\mu$ s per word
<b>Minimum Access Time</b>	258 $\mu$ s	320 $\mu$ s
<b>Average Access Time</b>	16.9 ms	20.3 ms
<b>Maximum Access Time</b>	33.6 ms	40.3 ms
<b>Program Interrupt</b>	33 ms Clock Flag Data Transmission Complete Flag Error Flag	
<b>Write Lock Switches</b>	Eight switches per disk capable of locking out any combination of eight 16,384 word blocks in addresses 0 to 131,071.	
<b>Data Tracks</b>	128	
<b>Words Per Track</b>	2048	
<b>Recording Method</b>	NRZ1	
<b>Density</b>	1100 bpi Maximum	
<b>Timing Tracks</b>	3 plus 3 spare (spares can be used to recover data on disk)	

### **RF08/RS08 Specifications (Cont)**

<b>Operating Environment</b>	Recommended temperature 65° to 90°F.
<b>Vibration/Shock</b>	Good isolation is provided. To prevent data errors, extreme vibrations should be avoided while the RS08 is transferring information.
<b>Heat Dissipation</b>	RF08: 150W RS08: 300W
<b>AC Power Requirements</b>	115/230 ± 10% Vac, single phase, 50 ± 2 or 60 ± 2 Hz, 5A (maximum) for logic power. (Logic power for one RF08 and up to four RS08s is provided by one DEC Type 705B Power Supply) Additional line current is required for RS08 disk motor as shown below.
<b>RS08 Motor Power Requirements</b>	Motor start, 5.5A for 20 ± 3s. Motor run, 4.0A continuous @ 115 Vac. (A stepdown autotransformer is provided for 230 Vac operation).
<b>Line Frequency Stability</b>	Maximum line frequency drift 0.1 Hz/s. A constant frequency motor-generator set or static ac/ac inverter should be provided for installation with unstable power sources.
<b>Motor Bearing Life</b>	Expected operating life of at least 20,000 hours, under standard computer operating environment.
<b>Reliability</b>	Six recoverable errors and one nonrecoverable error in $2 \times 10^9$ bits transferred. A recoverable error is defined as an error that occurs only once in four successive reads. All other errors are nonrecoverable. On-off cycling of the RS08 is not recommended. For this reason, the RS08 motor control operates independently of the computer power control.
<b>Cabinet</b>	A dedicated cabinet is designed to accommodate one RF08, up to two RS08s and power supply. Two additional RS08s can be mounted in a second cabinet. Other equipment should not be mounted in disk cabinets.
<b>Shipping Information</b>	<b>Weight of RF08, one RS08, power supply and cabinet:</b> 590 lb (crated) 500 lb (uncrated) <b>Weight of RF08, two RS08, power supply and cabinet:</b>

## **Programming Instructions**

The programming instructions for the RF08/RS08 differ slightly from those provided in the DF32/DS32 description. The extended address capability and associated instructions (DCEA, DEAL, and DEAC) are replaced, in sequence, by interrupt enable and memory address extension register instructions (DCIM, DIML, and DIMA).

### **Clear Disk Interrupt Enable and Core Memory Address Extension Register (DCIM)**

Octal Code: 6611

Event Time: 1

Indicators: Iot, Fetch, Pause [IR = 6, F]

Execution Time: 4.25  $\mu$ s

Operation: Clear the disk interrupt enable (DIE) and core memory address extension (MAE) registers.

Symbol: 0  $\rightarrow$  DIE, 0  $\rightarrow$  MAE

### **Load Interrupt Enable and Memory Address Extension Register (DIML)**

Octal Code: 6615

Event Time: 1, 3

Indicators: Iot, Fetch, Pause [IR = 6, F]

Execution Time: 4.25  $\mu$ s

Operation: Clear the interrupt enable (IE) and MAE, then load the interrupt enable and memory address extension registers with data held in the AC. Then clear AC.

#### **NOTE**

Transfers cannot occur across memory fields. Attempts to do so will cause the transfer to "wrap around" within the specified memory field.

Symbol: 0  $\rightarrow$  IE, 0  $\rightarrow$  MAE  
AC 3-15  $\rightarrow$  IE, AC 6-8  $\rightarrow$  MAE  
0  $\rightarrow$  AC

### **AC TO DISK STATUS REGISTER**

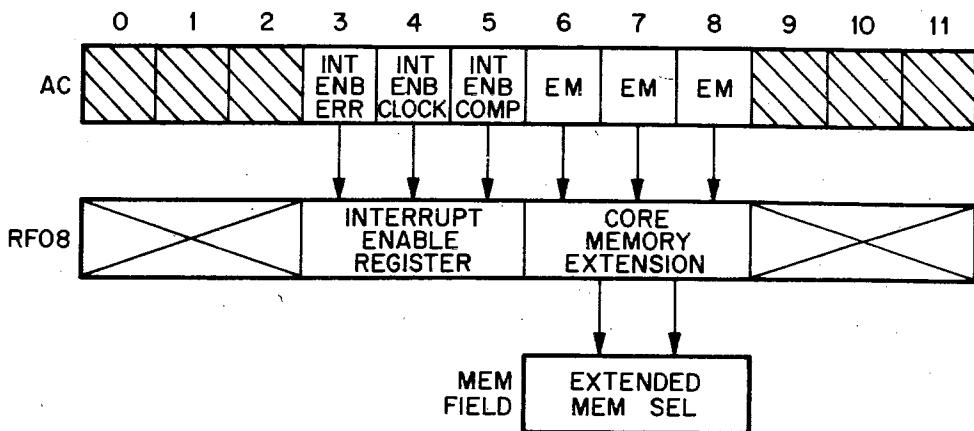


Figure 7-12 AC TO DISK STATUS REGISTER

**Load Interrupt and Extended Memory Address (DIMA)**

Octal Code: 6616

Event Time: 2, 3

Indicators: lot, Fetch, Pause [IR = 6, F]

Execution Time: 4.25  $\mu$ s

Operation: Clear the AC. Then load the contents of the status register (STR), into the AC to allow program evaluation.

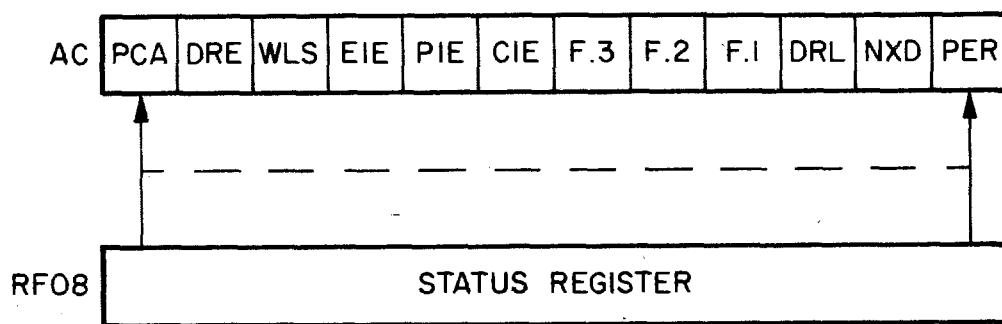


Figure 7-13 DISK TO AC STATUS REGISTER

AC Bit	Abbr.	Description
0	PCA	Photocell Sync Mark (available 100 $\mu$ s status)
1	DRE	Data Request Enable (maintenance only status)
2	WLS	Write Lock Status
3	EIE	Error Interrupt Enable
4	PIE	Photocell Interrupt Enable
5	CIE	Completion Interrupt Enable
6-8	F	(FIELD) Core Memory Extension Fields
9	DRL	Data Request Late
10	NXD	Nonexistent Disk
11	PER	Parity Error

Symbol: 0 → AC  
STR → AC

In addition to these changes in instructions, the RF08/RS08 utilizes six additional instructions: DFSE, DISK, DCXA, DXAL, DXAC, and DMMT.

**Skip on Disk Error (DFSE)**

Octal Code: 6621

Event Time: 1

Indicators: Iot, Fetch, Pause [IR = 6, F]

Execution Time: 4.25  $\mu$ s

Operation: Skip next instruction if there is parity error, data request late, write lock status, or nonexistent disk flag set.

Symbol: Parity error, data request late, write lock status, or nonexistent disk flags are set, PC + 1  $\rightarrow$  PC.**Skip Error or Completion Flag (DISK)**

Octal Code: 6623

Event Time: 2

Indicators: Iot, Fetch, Pause [IR = 6, F]

Execution Time: 4.25  $\mu$ s

Operation: If either the error or data completion flag (or both) is set, the next instruction is skipped.

Symbol: If PER or Data Complete, PC + 1  $\rightarrow$  PC.**Clear High Order Address Register (DCXA)**

Octal Code: 6641

Event Time: 1

Indicators: Iot, Fetch, Pause [IR = 6, F]

Execution Time: 4.25  $\mu$ s

Operation: Clear the high order 8-bit disk address register (DAR).

Symbol: 0  $\rightarrow$  DAR**Clear and Load High Order Address Register (DXAL)**

Octal Code: 6643

Event Time: 1, 2

Indicators: Iot, Fetch, Pause [IR = 6, F]

Execution Time: 4.25  $\mu$ s

Operation: Clear the high order 8 bits of the DAR. Then load the DAR from data stored in the AC. Then clear AC.

Symbol: 0  $\rightarrow$  DAR high order 8 bits,AC  $\rightarrow$  DAR,0  $\rightarrow$  AC

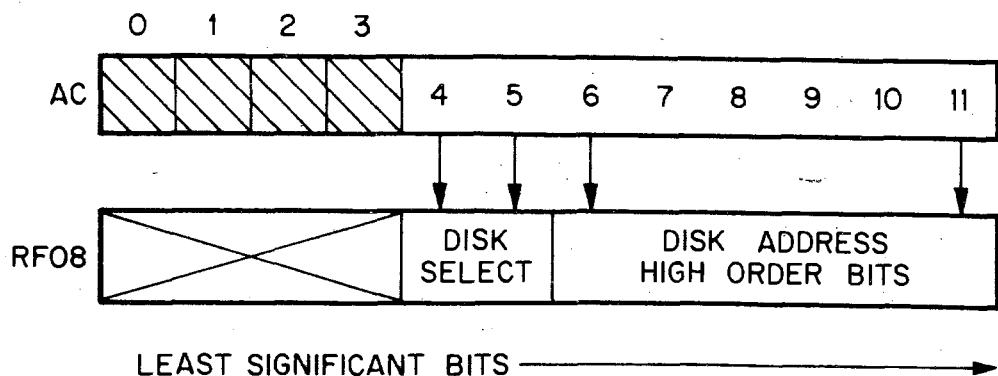


Figure 7-14 Higher-Order Address Word Transfer

**Clear Accumulator and Load DAR Into AC (DXAC)**

Octal Code: 6645

Event Time: 1, 3

Indicators: Iot, Fetch, Pause [IR = 6, F]

Execution Time: 4.25  $\mu$ s

Operation: Clear the AC; then load the contents of the high order 8-bit DAR into the AC.

Symbol:  $0 \rightarrow AC$ ,  
DAR high order 8 bits  $\rightarrow AC$

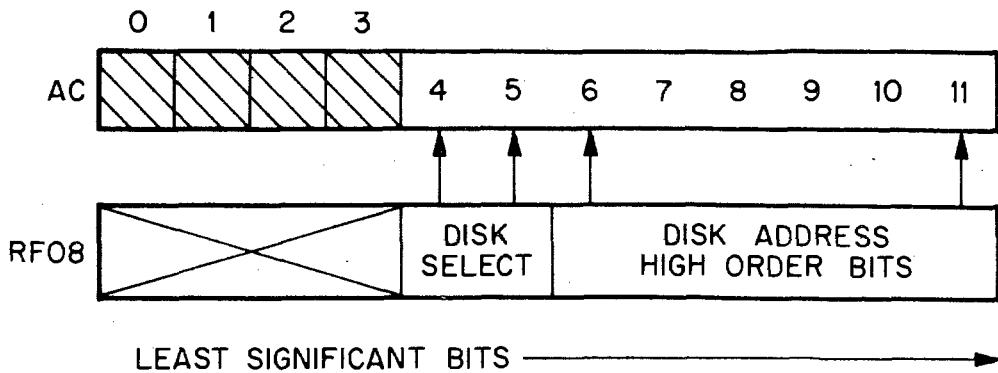


Figure 7-15 Disk Address Transfer to AC

### **Initiate Maintenance Register (maintenance purposes only) (DMMT)**

Octal Code: 6646

Event Time: 2, 3

Indicators: Iot, Fetch, Pause [IR = 6, F]

Execution Time: 4.25  $\mu$ s

Operation: For maintenance purposes only with the appropriate maintenance cable connections and the disk disconnected from the RS08 logic, the following standard signals may be generated by IOT 66-46 and associated AC bits. AC is cleared and the maintenance register (MAIR) is initiated by issuing an IOT 6601 command.

- AC (1) Track A Pulse
- AC (1) Track B Pulse
- AC (1) Track C Pulse
- AC (1) DATA PULSE (DATA HEAD #0)
- AC (1) Photocell
- AC (1) DBR

Setting DBR to a 1 causes data break request in computer.

Symbol: AC → MAIR

Three-cycle data break locations: word count address is 7550 (field 0), current address is 7751 (field 0).

### **DF32 Programming Compatibility**

The IOT instructions 660X and 6622 are identical in every respect to the DF32 instruction; i.e., the same operations are performed. The 661X and 662X instructions differ only in the following:

- a. IOT 6615 does not transmit the extended disk address bits for addressing over 32K; instead, AC 3-5 are assigned to enable or disable conditions on the program interrupt line. The AC is cleared upon execution of this instruction.
- b. IOT 6616 no longer reads back the extended address bits by 1 through 5 into the AC. These bits are assigned to examine the status of interrupt enable. In addition, AC2 indicates the status of write lock and AC10 shows only nonexistent disk conditions. AC1 shows the condition of data request enable used for maintenance purposes.
- c. IOT 6621 has been changed to skip on error rather than no-error. Non-existent disk has been included as an error skip condition.
- d. IOT 6623 (DISK) is a new skip instruction that will skip on either error or completion flags or both.

The DF32 maintenance instruction IOT 663X is not assigned to the RF08 system.

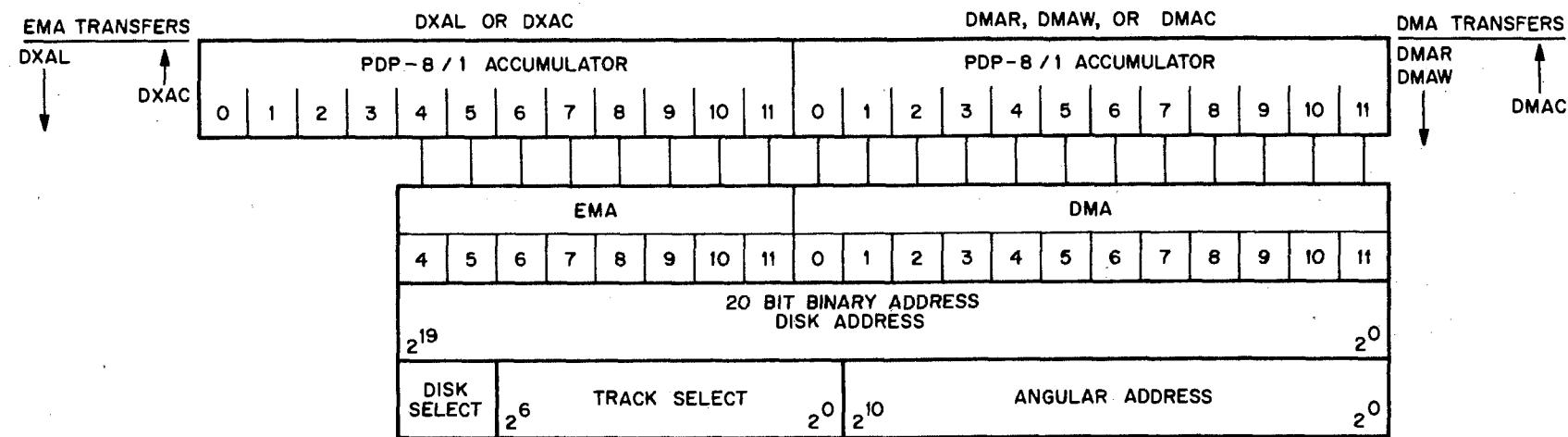


Figure 7-16 RF08 Addressing Format

## Programming Example

### Software

A sample of a typical I/O routine for the RF08/RS08 is as follows:

```

0200 4777      JMS I  (DISKIO)
0201 0000  FUNCT, 0          /X0=READ, X1=WRITE (X=0-7 MEMORY FIELD)
0202 0000  WDCT, 0          /+ WORD COUNT
0203 0000  CORE, 0          /CORE LOCATION
0204 0000  DSKHI, 0         /HIGH ORDER 8 BITS
0205 0000  DSKLOW, 0        /LOW ORDER 12 BITS
0206 5020      JMP ERROR   /ERROR RETURN (AC=ERROR CONDITION)
                           /NORMAL RETURN (AC=0)

0207 0000  DISKIO, 0
0210 7300      CLL CLA
0211 1607      TAD I DISKIO
0212 6615      DIML           /LOAD EXTENDED MEMORY BITS
0213 1607      TAD I DISKIO
0214 0376      AND (7
0215 7640      SZA CLA

0216 7126      STL RTL    /+2
0217 1375      TAD (3
0220 1374      TAD (6600
0221 3236      DCA RORW   /6603=READ, 6605=WRITE
0222 2207      ISZ DISKIO
0223 1607      TAD I DISKIO
0224 7041      CIA
0225 3773      DCA I (7750 /STORE=WORD COUNT
0226 2207      ISZ DISKIO
0227 1607      TAD I DISKIO
0230 3772      DCA I (7751 /LOAD CORE ADDRESS
0231 2207      ISZ DISKIO
0232 1607      TAD I DISKIO
0233 6643      DXAL           /LOAD HIGH ORDER 9
                           /BITS OF DISK ADDRESS,
                           /BITS OF DISK ADDRESS,

0234 1607      TAD I DISKIO
0235 2207      ISZ DISKIO
0236 0000  RORW, . 0          /READ OR WRITE
0237 6623      DISK           /DONE?
0240 5237      JMP -1        /NO
0241 6621      DFSE           /YES, ERROR?
0242 2207      ISZ DISKIO   /SKIP TO NORMAL RETURN
0243 5607      JMP I DISKIO  /RETURN

6615  DIML = 6615
6623  DISK = 6623
6643  DXAL = 6643
6621  DFSE = 6621
0020  ERROR = 20

0372 7751
0373 7750
0374 6600
0375 0003
0376 0007
0377 0207

CORE    0203
DFSE   6621
DIML   6615
DISK   6623
DISKIO  0207
DSKHI  0204
DSKLOW 0205
DXAL   6643
ERROR  0020
FUNCT  0201
RORW   0236
WDCT   0202

```

## MAGNETIC TAPE OPTIONS

The External Bus Magnetic Tape Options include:

- a. The TU56 Dual DECTape Transport and TC08 DECTape control,
- b. The TU10 DECMAGtape Transport and TC58 Automatic Magnetic Tape Control.

### DECTape

The DECTape system is a standard option for the PDP-8/E that serves as an auxiliary magnetic tape data storage facility. The DECTape system stores information at fixed positions on magnetic tape, as in magnetic disk or drum storage devices, rather than at unknown or variable positions, as in conventional magnetic tape systems. This feature allows replacement of blocks of data on tape in a random fashion without disturbing other previously recorded information. In particular, during the writing of information on tape, the system reads format (mark) and timing information from the tape and uses this information to determine the exact position at which to record the information to be written. Similarly, in reading, the same mark and timing information has a number of features to improve its reliability and make it exceptionally useful for program updating and program editing applications. These features are: phase or polarity sensed recording on redundant tracks, bidirectional reading and writing, and a simple mechanical mechanism utilizing hydrodynamically lubricated tape guiding (the tape floats on air over the tape guides while in motion).

Four basic DECTape configurations are identified in the following table.

SYSTEM DESIG- NATION	DECTape	CON- TROL	PREREQ- UISITE	REMARKS
None	TU56 (Dual Drive)	TC08	KA8-E KD8-E PDP-8/E	Up to 4 Dual TU56's per control. (8 drive units)
None	TU56 (Single Drive)	TC08	KA8-E KD8-E PDP-8/E	Up to 4 single DECTape drive units.
TD8-EM	TU56 (Dual Drive)	TD8-E	PDP-8/E	Up to 4 Dual Drive TU56's per control. (8 drive units) Control plugs into OMNIBUS.
TD8-EA	TU56H (Single Drive)	TD8-E	PDP-8/E	Up to 4 single drive units. Con- trol plugs into OMNIBUS.

Magnetic tape options operated on the external bus of the PDP-8/E require the use of the KA8-E Positive I/O Bus Interface module and the KD8-E Data Break Interface module as prerequisites.

### **DECtape Format**

DECtape utilizes a 10-track read/write head. Tracks are arranged in five nonadjacent redundant channels: a timing channel, a mark channel, and three information channels. Redundant recording of each character bit on nonadjacent tracks materially reduces bit dropouts and minimizes the effect of skew. The series-connection of corresponding track heads within a channel and the use of Manchester phase recording techniques, rather than amplitude sensing techniques, virtually eliminate dropouts.

The timing and mark channels control the timing of operations within the control unit and establish the format of data contained on the information channels. The timing and mark channels are recorded prior to all normal data reading and writing on the information channels. The timing of operations performed by the tape drive and some control functions are determined by the information on the timing channel. Therefore, wide variations in the speed of tape motion do not affect system performance. Information read from the mark channel is used during reading and writing data to indicate the beginning and end of data blocks and to determine the functions performed by the system in each control mode. During normal data reading, the control assembles 12-bit computer-length words from four successive lines read from the information channels of the tape. During normal data writing, the control disassembles 12-bit words and distributes the bits so they are recorded on four successive lines on the information channels. A mark-channel error-check circuit ensures that one of the permissible marks is read in every six lines on the tape. This 6-line mark-channel sensing requires that data be recorded in 12-line segments (12 being the lowest common multiple of 6-line marks and 4-line data words) which correspond to three 12-bit words.

A tape contains a series of data blocks that can be of any length which is a multiple of three 12-bit words. Block length is determined by information on the mark channel. A uniform block length is usually established over the entire length of a reel of tape by a program that writes mark and timing information at specific locations. The ability to write variable-length blocks is useful for certain data formats. For example, small blocks containing index or tag information can be alternated with large blocks of data. (Software supplied with DECtape allows writing for fixed block lengths only.)

Between the blocks of data are areas called interblock zones. The interblock zones consist of 30 lines on tape before and after a block of data. Each of these 30 lines is divided into five 6-line control words. These 6-line control words allow compatibility between DECtape written on any of DEC's 12-, 18-, or 36-bit computers. As used on the PDP-8/E, only the last four lines of each control word are used.

Block numbers normally occur in sequence from 1 to n. There is one block numbered 0 and one block n + 1. Programs are entered with a statement of the first block number to be used and the total number of blocks to be read or written. The total length of the tape is equivalent to 849,036 lines, which can be divided into any number of blocks up to 4096 by prerecording of the mark track. The maximum number of blocks is determined by the following equation in which n(b) equals number of

blocks and  $n(w)$  equals number of words per block ( $n(w)$  must be divisible by 3).

$$n(b) = \frac{212112}{n(w) + 15} - 2$$

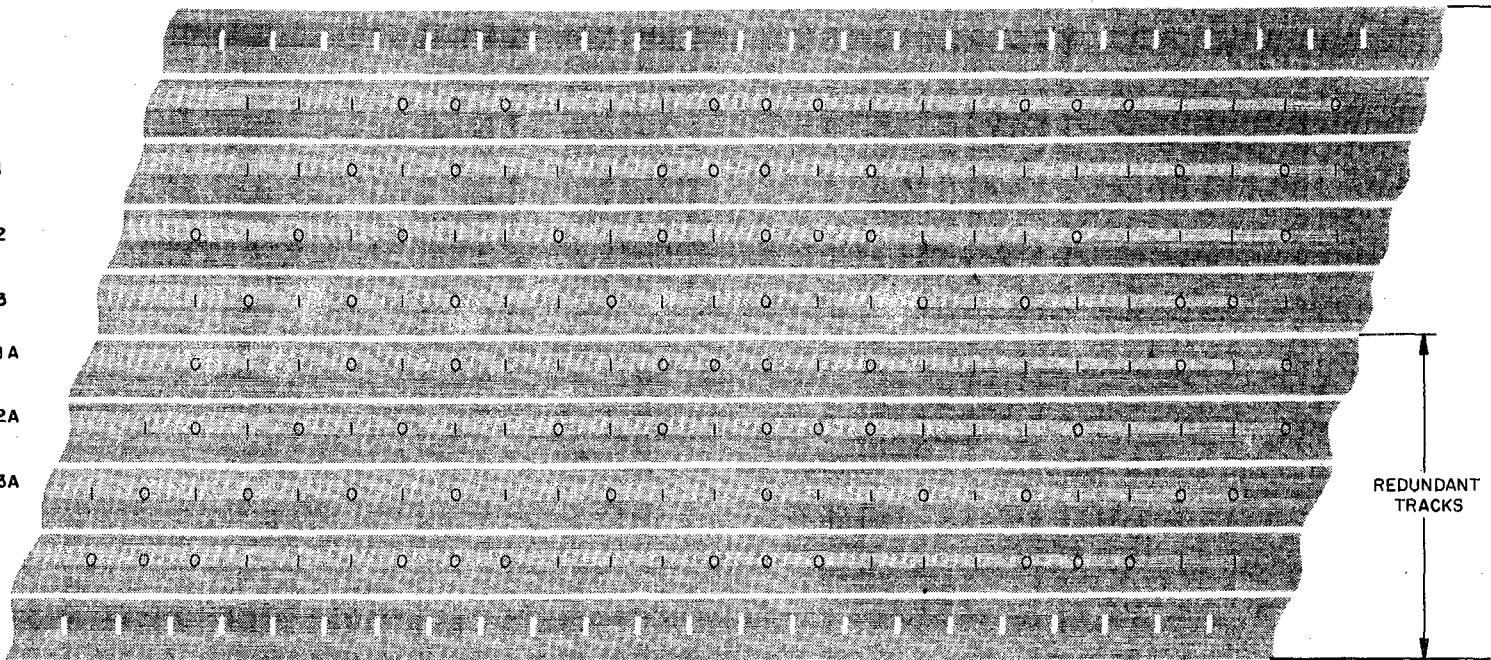
DECtape format is illustrated in Figures 7-17 through 7-20.



7.21

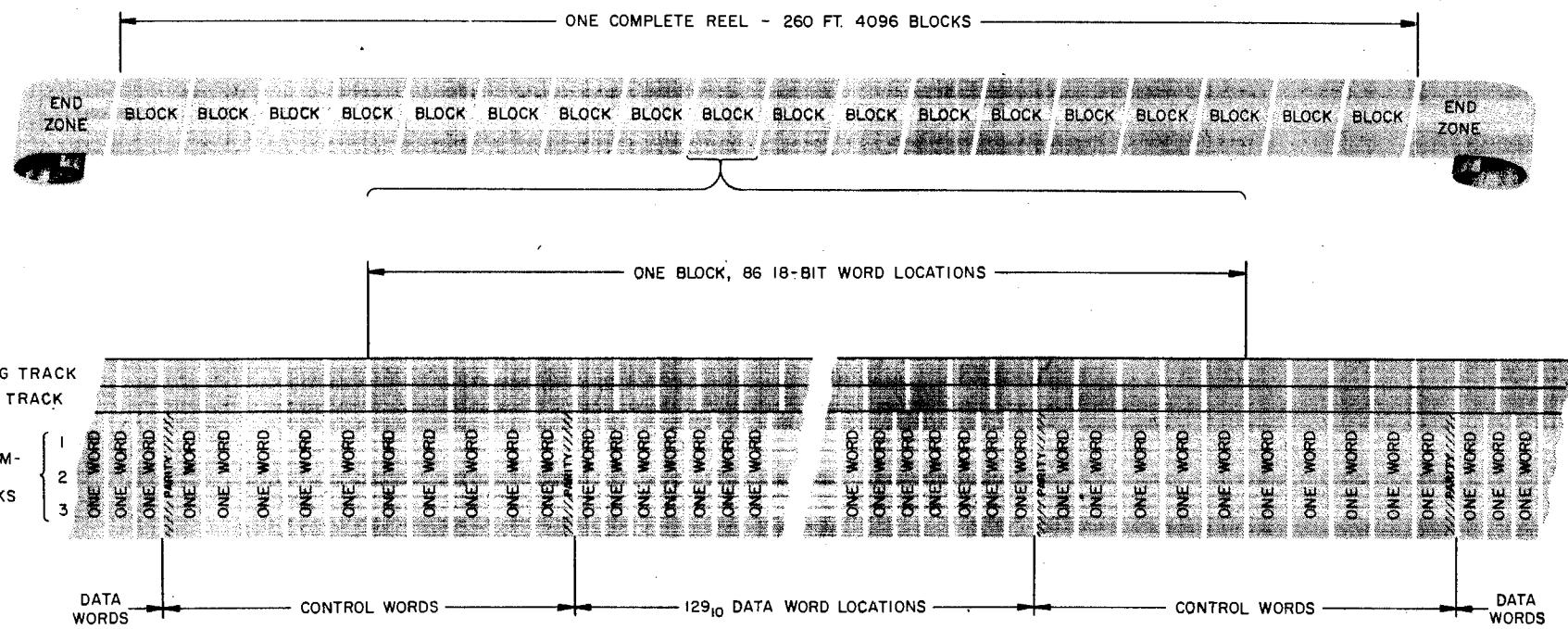
Figure 7-17 DECTape Track Allocations

TIMING TRACK 1  
MARK TRACK 1  
INFORMATION TRACK 1  
INFORMATION TRACK 2  
INFORMATION TRACK 3  
INFORMATION TRACK 1A  
(Same as IT 1)  
INFORMATION TRACK 2A  
(Same as IT 2)  
INFORMATION TRACK 3A  
(Same as IT 3)  
MARK TRACK 1A  
(Same as MT 1)  
TIMING TRACK 1A  
(Same as TT 1)



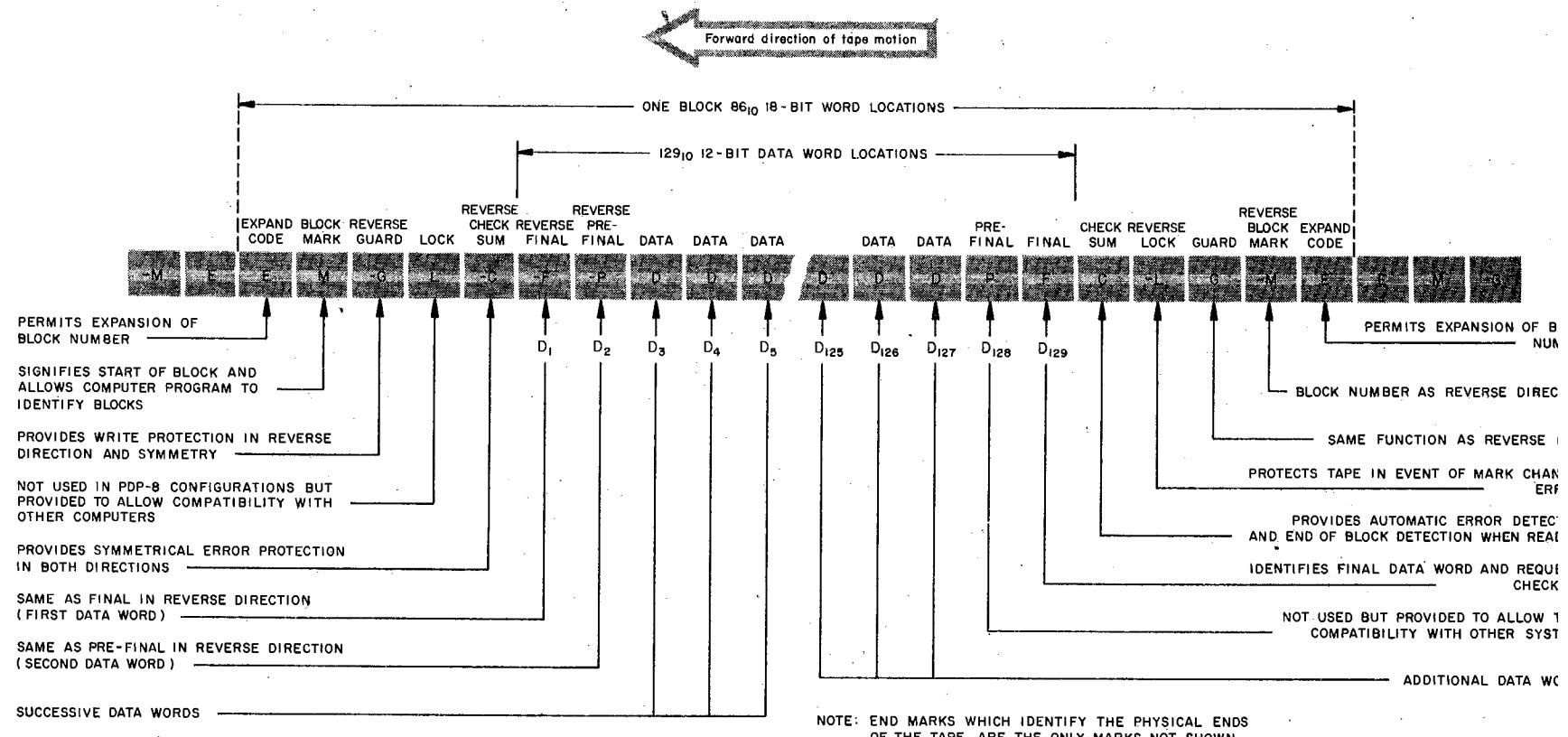
7-22

Figure 7-18 DECtape Mark Channel Format



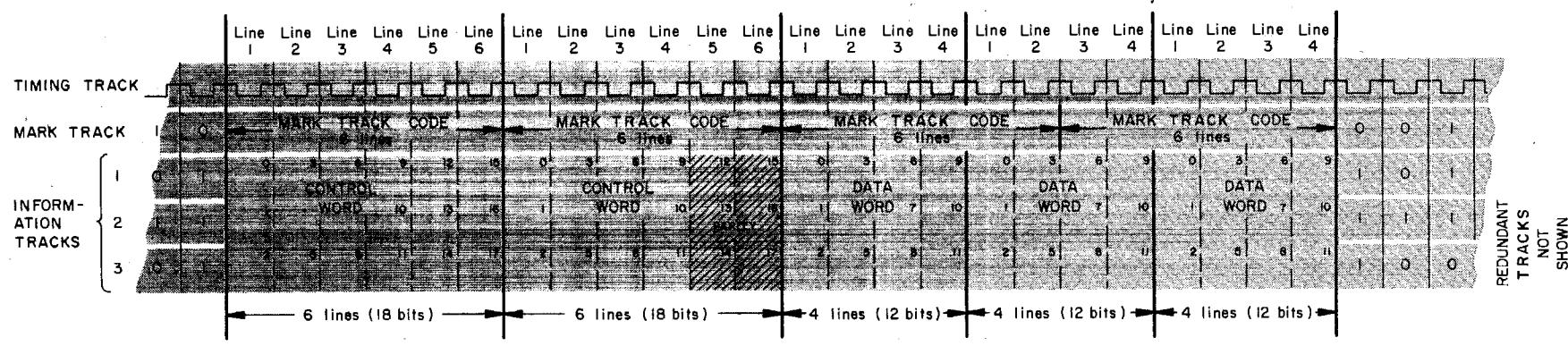
7-23

Figure 7-19 DECTape Control Word and Data Word Assignments



**Figure 7-20** DECtape Format Details

7-24



### **TU56 Dual DECTape Transport and TC08-P DECTape Control**

A DECTape system on the external bus can contain up to four TU56 Dual DECTape transports (the equivalent of up to eight single tape transports) controlled from one TC08-P unit. Data transfers between the computer and tape are implemented using the three-cycle data break facilities of the computer (refer to Chapter 10 for three-cycle data break description). Thus, the KA8-E Positive I/O Bus Interface and KD8-E Data Break Interface units are prerequisites.

Data is stored on tape in the form of three-bit words (refer to tape format) and is transferred between the tape and computer in the form of 12-bit words. A 12-bit read/write buffer in the TC08 assembles and disassembles the information for transfer. For transfers to the computer, data is read from four consecutive lines of tape and assembled into a 12-bit word. When transferred to the computer, the 12-bit word is supplied via external bus lines DATA00-11 to the KD8-E Data Break Interface. This unit, in turn, provides the word to OMNIBUS lines DATA0-11 under data break control. For transfers to tape, the KA8-E unit buffers the 12-bit words and provides them to the TC08 via external bus lines BMB00-11. The TC08 disassembles these words and supplies them to the tape transport for the writing of four tape lines. Transfer of command and control signals is effected by IOT instructions. These instructions are provided to the TC08 via the BMB00-11 external bus lines.

The TC08 contains registers and control flip-flops that form two status registers (designated A and B) for transfer of information to and from the computer accumulator.

### **TU56 Dual DECTape Transport**

The TU56 provides the PDP-8/E user with a compact, high-reliability dual-reel tape transport in just 10½ inches of rack space. When used with the TC08-P control, the TU56 provides two fixed address, magnetic tape facilities for high-speed loading, readout, and updating of programs and data. The TU56 transport contains the tape read/write heads, drive mechanisms, and switching circuits for tape drive and direction. All transport operations (except local) are controlled by the TC08 from program instructions. The TC08 selects the transport, controls tape motion and direction, selects a read or write operation and buffers data transferred. Information is stored in the form of three-bit words on a one-mil Mylar tape with ten tracks. This tape, ¾ inches in width and 260 feet in length, is contained on a reel that is less than four inches in diameter. Information can be recorded or read for either direction of tape motion.

Redundant recording (each bit of data and timing is recorded on two tracks) ensures high reliability and eliminates the need for parity checking. Data words are recorded on six of the ten tracks and four tracks are allotted for mark and timing channels. Other features include TTL logic, dynamic braking for shorter turnaround time, and DC motor drive to eliminate line frequency dependency. Connections from the read/write head are made directly to the external control, which contains the read and write amplifiers.

The logic circuits of the TU56 transport control tape movement in either direction over the read/write head. Tape drive motor control is com-

pletely through the use of solid-state switching circuits to provide fast, reliable operation. These circuits control the torque of the two motors that transport the tape across the head according to the established function of the device: i.e., go, stop, forward, or reverse. In normal tape movement, full torque is applied to the forward or leading motor and a reduced torque is applied to the reverse or trailing motor to keep proper tension on the tape. Since tape motion is bidirectional, each motor serves as either the leading or trailing drive for the tape, depending upon the forward or reverse control status of the TU56.

Tape movement can be controlled by commands originating in the computer or by manual operation of switches on the front panel of the transport. Manual control is used to mount new reels of tape on the transport, or as a quick maintenance check for proper operation of the control logic in moving the tape.

Since DECTape is a fixed address system, the programmer need not know accurately where the tape has stopped. To locate a specific point on tape he must only start the tape motion in the search mode. The address of the block currently passing over the head will be automatically transferred to core where it can be compared with the desired block address and tape motion continued or reversed accordingly. TU56 typical time characteristics are provided below, but are not accurately controlled.

Start Time	150 ms*
Stop Time	100 ms*
Turnaround Time	200 ms*

\*Also, see control specifications. These times are frequently lengthened by the particular control.

#### Specifications

Transfer rate	33,300 three-bit characters per second
Information capacity	2.7 million bits per reel
Density	350 + or - 55 bits per inch
Tape speed	93 + or - 12 inches per second
Tape motion	Bidirectional
Start time	150 + or - 15 ms
Stop time	100 + or - 10 ms
Turn around time	200 + or - 50 ms
Reel capacity	250 ft. of $\frac{3}{4}$ inch, 1 mil Mylar tape
Reel size	3.9 inches in diameter
Mounting	Mounts in a standard 19-inch equipment rack
Size	10 $\frac{1}{2}$ in. high, 19 in. wide, 9 $\frac{3}{4}$ in. deep
Cooling	Internally mounted fans provided
Power requirements	<ol style="list-style-type: none"> <li>+ 10V @ 0.53 amps or + 5V @ 0.55 amps</li> <li>- 15V @ 0.45 amps</li> <li>115/220 VAC + or - 10% @ 2.85/1.43 amps 47-63 Hz</li> </ol>
Environmental	Temperature: 40 degrees F to 90 degrees F Humidity: 15% to 80% Relative Humidity Internal Temp Rise: 10% F above ambient
Reliability	Recoverable Error Rate-less than 1 part in 2.5 x 10 ↑ 10 transfers

### **TC08 DECTape Control**

The TC08 control buffers and controls information transfers between one to eight TU56 transports (one to four TU56 Dual DECTape transports) interfacing with the external bus of the PDP-8/E. Transfers are implemented using the three-cycle data break facilities of the computer; thus, the KA8-E Positive I/O Bus Interface and the KD8-E Data Break Interface modules are prerequisites.

During both input and output operations, the TC08 receives data and control information from the processor and generates the appropriate signals to the selected transport to execute the programmed commands. Binary information is transferred between the tape transport and the computer as one 12-bit computer word every  $133\frac{1}{3} \mu s$ . When writing, the TC08-P disassembles the 12-bit word into four successive three-bit words to be written on tape. During read operations, the TC08 assembles the four successive three-bit words into one 12-bit word for transfer to the computer. Transfers between the computer and the control always occur in parallel for a 12-bit word. Data transfers use the three-cycle data-break (high speed channel) facility of the computer. (Refer to Chapter 10 for details of 3-cycle data-break transfers.)

The TC08 contains the following primary control and data processing circuits:

- a. Device selector and IOT decoding logic to command a transport from program instructions.
- b. A 12-bit buffer register for assembling tape inputs and disassembling computer data.
- c. A command and status register (designated Status Register A) for defining: (1) the active transport, (2) direction of tape, (3) tape motion, (4) operating mode, (5) function (read/write, search, etc.), (6) interrupt enable, and (7) clearing of flags.
- d. A status register (designated Status Register B) for indicating error status and other status.
- e. Flag circuits that provide the program with conditional indications and requests.
- f. Tape motion and direction control circuits.
- g. Mark track generation and detection circuits with error detectors.
- h. Longitudinal parity generation and checking circuits.
- i. Data break request circuits.

Programmed IOT instructions are generated to clear, read, or load Status Register A and to read or load Status Register B. An IOT skip instruction is also provided to test the status of flag circuits. These instructions are provided to the TC08-P control via the KA8-E Positive I/O Bus Interface and external bus lines BMB-00-11.

A control and indicator panel is also provided with the TC08. A single control, NORMAL/WRTM, places the TC08 in the write timing and mark track mode (WRTM), or else in the NORMAL mode. The indicators denote the current status of the control including the tape transport selected, motion, function, interrupt status, error flags, and other status indications.

Three program flags in the TC08-P control serve as condition indicators and request originators.

- a. DECTape Flag (DT): This flag indicates the active/done status of the current function.
- b. Data Flag (DF): This flag requests a data break to transfer a block number into the computer during a search function, or when a data word transfer is required during a read or write function.
- c. Error Flag (EF): Detection of any nonoperative condition by the control sets this flag in status register B and stops (except for parity errors) the selected transport. The error conditions indicated by this flag are:
  - (1) Mark Track Error: This error occurs any time the information read from the mark channel is erroneously decoded.
  - (2) End of Tape: The end zone on either end of the tape is over the read head.
  - (3) Select Error: This error occurs 5  $\mu$ s after loading status register A to indicate any one of the following conditions:
    - (a) Specifying a unit select code which does not correspond to any transport select number, or which is set to multiple transports.
    - (b) Specifying a write function with the WRITE ENABLED/ WRITE LOCK switch in the WRITE LOCK position on the selected transport.
    - (c) Specifying an unused function code (i.e., AC6-8 = 111).
    - (d) Specifying any function except write timing and mark track with the NORMAL/WRTM switch in the WRTM position.
    - (e) Specifying the write timing and mark track function with the NORMAL/WRTM switch in the NORMAL position.
  - (4) Parity Error: This error occurs during a read data function if the longitudinal parity or check sum over the entire data word, the reverse check character, and the check character is not equal to 1.
  - (5) Timing Error: This error indicates a program fault caused by one of the following conditions:
    - (a) A data break did not occur within 17  $\mu$ s (+ or - 30%) of the data break request.
    - (b) The DT flag was not cleared by the program before the control attempt to set it.
    - (c) The read data or write data function was specified while a data block was passing the read head.

Three-cycle data break locations: The TC08-P uses location 7754 of field 0 for word count and 7755 of field 0 for current address.

**Control Modes**—The DECTape system operates in either the normal or continuous mode, as determined by bit 5 of status register A during a DTXA command. Operation in each mode is as follows:

- a. Normal (NM): Data transfers and flag settings are controlled by the format of information on the tape.

- b. Continuous (CM): Data transfers and flag settings are controlled by a word count read from core memory during the first cycle of each three-cycle data break, and by tape format.

**Functions**—The DECTape system performs one of seven functions, as determined by the octal digit loaded into status register A during a DTXA command. These functions are:

- a. Move: Initiates movement of the selected transport tape in either direction. Mark channel decoding is inhibited in this mode except for end of tape.
- b. Search: As the tape is moved in either direction, sensing of a block mark causes a data transfer of the block number. If the word count overflows in either NM or CM, the DT flag is set and causes a program interrupt. After finding the first block number, the CM can be used to avoid all intermediate interrupts between the current and the desired block number. This makes a virtually automatic search possible.
- c. Read Data: This function is used to transfer blocks of data into core memory with the transfer controlled by the tape format. In NM, the DT flag is set at the end of a block and causes a program interrupt. In CM, transfers stop when the word count overflows, the remainder of the block is read for parity checking, and then the DT flag is set.
- d. Read All: Read all is used to read tape in an unusual format, since it causes all lines to be read. In NM, the DT flag is set at each data transfer. In CM, the DT flag is set when WCO occurs. In either case, the DT flag causes a program interrupt.
- e. Write Data: This function is used to write blocks of data with the transfer controlled by the standard tape format. After word count overflow occurs, zeros are written in all lines of the tape to the end of the current block. Then the parity checksum for the block is written. The DT flag rises as in the read function.
- f. Write All: The write all function is used to write an unusual tape format (e.g., block numbers). The DT flag assertions are similar to the read all function.
- g. Write Timing and Mark Track: This function is used to write on the timing and mark tracks, permitting blocks to be established or block lengths to be changed. The DT flag assertions are also similar to the read all function. This function is illegal unless a manual switch in the control is positioned to WRTM.

**Programmed Operation**—Prerecording of a reel of DECTape, prior to its use for data storage, is accomplished in two passes. During the first pass, the timing and mark channels are placed on the tape. During the second pass, forward and reverse block mark numbers, the standard data pattern, and the automatic parity checks are written. These functions are performed by the DECTOG program. Prerecording utilizes the write timing and mark channel function, and a manual switch on the control, which permits writing on the timing and mark channels, activates a clock, which produces the timing channel recording pattern and enables flags for program control. Unless this control function and switch are used simultaneously, it is physically impossible to write on the

mark or timing channels. An indicator lamp on the control panel lights when the manual NORMAL/WRTM switch is in the WRTM position. Under these conditions only, the write register and write amplifier, used to write on information channel 1 (bits 0, 3, 6, and 9), are used to write on the mark channel. This prerecording operation need only be performed once for each reel of DECTape.

There are two registers in the TC08 DECTape Control that govern tape operation and provide status information to the operating program. Status register A contains three unit selection bits, two motion bits, the continuous mode/normal mode bit, three function bits, and three bits that control the flags. Status register B contains the three memory field bits and the error status bits. PDP-8/E IOT microinstructions are used to clear, read, and load these registers. In addition, there is an IOT skip instruction to test control status.

Since all data transfers between DECTape and the computer memory are controlled by the data break facility, the program must set the WC and CA registers (locations 7754 and 7755, respectively) before a data break. After initiating a DECTape operation, the program should always check for error conditions (a program interrupt would be initiated if the error flag is enabled and if the program interrupt system is enabled). The DECTape system should be started in the search function to locate the block number selected for transfer; when the correct block is found, the transfer is accomplished by programmed setting of the WC, CA, and status register A.

When searching, the DECTape control reads block numbers only. These are used by the operating program to locate the correct block number. In NM, the DECTape flag is raised at each block number. In CM, the DECTape flag is raised only after the word count reaches zero. The current address is not incremented during searching and the block number is placed in core memory at the location specified by the content of the CA. Data is transferred to or from the computer core memory from locations specified by the CA register which is incremented by one before each transfer.

Each time the DECTape system is ready to transfer a 12-bit word, and when the start of the data position of the block is detected, the data flag is raised to initiate a data break request to the data break facility. Therefore, the main computer program continues running, but is interrupted approximately every  $133\frac{1}{3}\ \mu s$  for a data break to transfer a word. Transfers occur between DECTape and successive core memory locations specified by the CA. The initial transfer address minus one is stored in the CA by an initializing routine. The number of words transferred is determined by the tape format in NM, or by tape format and the word count in CM. At the conclusion of the data transfer, the DT flag is raised and a program interrupt occurs. The interrupt subroutine checks the DECTape error bits to determine the validity of the transfer, and either initiates a search for the next information to be transferred or returns to the main program.

During all normal writing transfers, a check character (the six-bit logical equivalent of the words in the data block) is computed automatically by

the control and is recorded automatically as one of the control words immediately following the data portion of the block. This same character is used during reading to determine that the data playback and recognition take place without error.

Any one of the eight tape transports may be selected for use by the program. After using a particular transport, the program can stop the transport currently being used and select another transport, or can select another transport while permitting the original selection to continue running. This is a particularly useful feature when rapid searching is desired, since several transports may be used simultaneously. Caution must be exercised, however; although the original transport continues to run, no tape-end detection or other sensing takes place. Automatic tape-end sensing that stops tape motion occurs in all functions, but only in the selected tape transport.

The following is a list of timing considerations for programmed operations. (These times are based on 129 12-bit data words per block.)

$n(s)$  = the number of block numbers to be read in the search function and CM, counting through the one causing the word count overflow. Only the block number causing the word count overflow requests a program interrupt.

$n(d)$  = number of words transferred divided by the number of words per block. If the remainder does not equal 0, use the next larger whole number.

$n(A)$  = number of words transferred.

#### OPERATION

Answer a data break request  
Word transfer rate

Block transfer rate

Change function from search to  
read data for the current block  
after DT flag from block number

Change function from search to  
write data for current block after  
DT flag from block number

Change function from read data to  
search for the next block after  
DT flag from transfer completion

Change function from write data  
to search for next block after  
DT flag from transfer completion

DECtape flag rises in continuous  
mode

Move function

Search function

Read data function

Read all function

#### TIMING

Up to 17  $\mu$ s + or - 30%  
One 12-bit word every 133  $\mu$ s +  
or - 30%

One 129-word block every 18.2 ms  
+ or - 30%  
400  $\mu$ s + or - 30%

400  $\mu$ s + or - 30%

1000  $\mu$ s + or - 30%

1000  $\mu$ s + or - 30%

Never

$(n(s)) \times (18.2 \text{ ms} + \text{ or } - 30\%)$   
 $(n(D)) \times (18.2 \text{ ms} + \text{ or } - 30\%)$   
 $(n(A)) \times (133 \mu\text{s} + \text{ or } - 30\%)$

OPERATION	TIMING
Write data function	(n(D)) x (18.2 ms + or - 30%)
Write all function	(n(A)) x (133 $\mu$ s + or - 30%)
Write T & M function	(n(A)) x (133 $\mu$ s + or - 30%)
In normal mode	
Move function	Never
Search function	Every 18.2 ms + or - 30%
Read data function	Every 18.2 ms + or - 30%
Read all function	Every 133 $\mu$ s + or - 30%
Write data function	Every 18.2 ms + or - 30%
Write all function	Every 133 $\mu$ s + or - 30%
Write T & M function	Every 133 $\mu$ s + or - 30%

### Programming

The following instructions are associated with TC08-P operation:

#### Read Status Register A (DTRA)

Octal Code: 6761

Execution Time: 2.6  $\mu$ s

Operation: Transfers content of Status Register A to the AC.; ORs AC0-9 with Status Register with the result appearing in AC. The AC is not cleared before the transfer. AC bit assignments are defined in Figure 7-21.

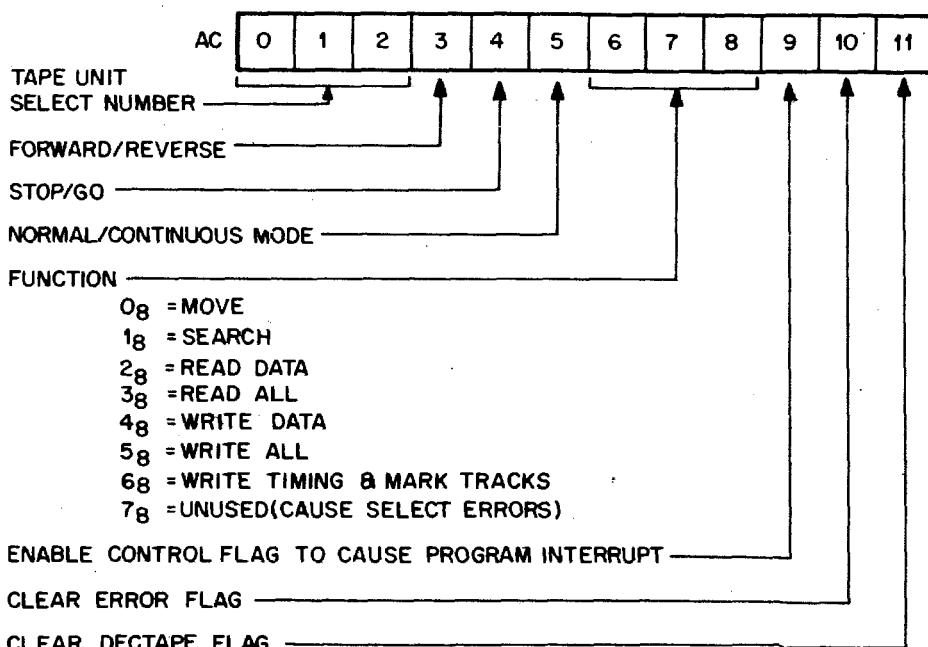


Figure 7-21 Status Register A Bit Assignments

#### Clear Status Register A (DTCA)

Octal Code: 6762

Execution Time: 2.6  $\mu$ s

Operation: Clears Status Register A; DECTape and Error flag are undisturbed.

### **Clear and Load Status Register A (DTLA)**

Octal Code: 6766  
Execution Time: 3.6  $\mu$ s  
Operation: Clears Status Register A, then EXCLUSIVE ORs content of AC0-9 into Status Register A. Samples AC10 and 11 to control clearing of DECTape and error flags, then clears AC.

### **Load Status Register A (DTXA)**

Octal Code: 6764  
Execution Time: 2.6  $\mu$ s  
Operation: EXCLUSIVE ORs content of AC0-9 into Status Register A. Samples AC bits 10 and 11 to control clearing of Error and DECTape flags, then clears the AC.

### **Skip On Flag (DTSF)**

Octal Code: 6771  
Execution Time: 2.6  $\mu$ s  
Operation: If either DECTape or Error flags is set, skips the next instruction.

### **Read Status Register B**

Octal Code: 6772  
Execution Time: 2.6  $\mu$ s  
Operation: ORs content of Status Register B into AC. The AC is not cleared before transfer; AC bit assignments are defined in Figure 7-22.

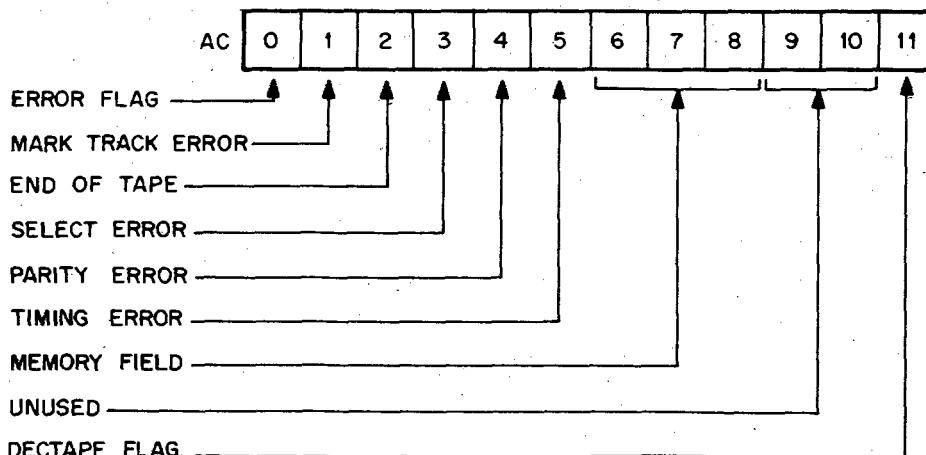


Figure 7-22 Status Register B Bit Assignments

### **Load Status Register B**

Octal Code: 6774  
Execution Time: 2.6  $\mu$ s  
Operation: Loads memory field portion of Status Register B with content of AC6-8, then clears the AC.

An elementary subroutine for reading or writing DECTape is given below. This routine does not use the interrupt, and exits with the DECTape drive halted.

The format for calling the subroutine is:

JMS (IDTAPE)	Effective JMS to IDTAPE, i.e., indirect JMS if IDTAPE is not on same page as calling sequence.		
WORD 1,	Bits 0-2, unit number Bit 3, start search (0=forward 1=reverse) Bits 6-8, memory field for transfer Bit 10, error return (0=JMP WORD 5) (1=JMP I WORD 5)		
	Bit 11, function (0=READ 1=WRITE)		
WORD 2,	Block number for start of transfer		
WORD 3,	2's complement of the number of words to transfer		
WORD 4,	Memory address of first transfer minus 1		
WORD 5,	Error return or address for error return (to correspond to Bit 10 of Word 1)		
RETURN,	Transfer completed, return with AC cleared		
ID7400, 7400	/AND MASK (MUST BE FIRST CELL IN /PAGE)		
IDTAPE, 0	/ENTRY TO SUBROUTINE		
CLA			
TAD I IDTAPE	/SAVE WORD 1		
DCA IDCODE			
ISZ IDTAPE	/ADVANCE TO BLOCK NUMBER (WORD /2)		
ID0200, TAD . IDCODE			
AND ID7400	/UNIT NUMBER AND DIRECTION BIT		
TAD ID0010	/PUT INTO SEARCH MODE		
DTCA DTXA			
DTLB	/CLEAR FIELD BITS		
TAD IDWC			
DCA I IDCA	/SET UP CURRENT ADDRESS (7755)		
/ERROR WHILE SEARCHING . . . NORMALLY ENTERED WITH B STATUS REGISTER IN THE AC, PERFORMS TURN AROUND IF END /ZONE ERROR, AND FORCES THE STOP-GO BIT TO GO			
IDSERR, RTL			
RAL	/MOVE END ZONE FLAG TO LINK		
CLA CML			
TAD ID0200	/GET DECTAPE GO FLAG		
/CHANGE DIRECTION IF AND ONLY IF THE LINK IS ZERO,			
IDCONT, SNL	/CHECK DIRECTION AND SIGN		
TAD ID0400	/REVERSE DIRECTION		
DTXA	/ENTER AND GO IN SEARCH MODE		
DTSF DTRB	/IDLE . . AND LOAD ERROR FLAG		
JMP .-1	/WAIT UNTIL FLAG COMES UP		
SPA	/TEST ERROR FLAG		
JMP IDSERR			

DTRA		/GET DIRECTION BIT
RTL		
RTL		/DIRECTION BIT GOES TO LINK
SZL	CLA	
TAD	ID0002	/REVERSE . . . GET "BLOCK TO FIND" /-2
TAD	I IDWC	/ADD IN LAST BLOCK SEEN
CMA		/COMPLEMENT
TAD	I IDTAPE	/ADD IN "BLOCK TO FIND"
CMA		
SZA	CLA	/BLOCK NUMBERS MATCH?
JMP	IDCONT	/REENTER SEARCH LOOP
SZL		/CHECK DIRECTION BIT
JMP	IDCONT+1	/TURN AROUND IF REVERSE

/END OF SEARCH LOOP, TAPE IS NOW AT DESIRED BLOCK  
/TRAVELING IN A FORWARD DIRECTION,

ISZ	IDTAPE	
TAD	I IDTAPE	/GET WORD COUNT
DCA	I IDWC	
ISZ	IDTAPE	
TAD	I IDTAPE	/GET TRANSFER ADDRESS
DCA	I IDCA	
TAD	IDCODE	
DTLB		/LOAD FIELD BITS
IAC		/GET READ-WRITE FLAG
AND	IDCODE	
RTL	CLL	/MULTIPLY BY 20 (OCTAL)
RTL		
TAD	ID0130	/BUILD INSTRUCTION
DTXA		/START UP READ OR WRITE
DTSF	DTRB	/WAIT . . AND LOAD ERROR FLAG
JMP	-1	
ISZ	IDTAPE	/ADVANCE TO WORD 5
SMA		/SKIP IF ERROR FLAG SET
ISZ	IDTAPE	/ADVANCE TO WORD 6 . . . NORMAL
		/EXIT
SPA	CLA	/SKIP FOR NORMAL EXIT
TAD	IDCODE	/GET INDIRECT RETURN BIT
RTR		/MOVE TO LINK
SNL	CLA	/SKIP IF JMP I <WORD 5>
JMP	+3	
TAD	I IDTAPE	/MAKE DOUBLE INDIRECT RETURN
DCA	IDTAPE	
DTRA		
AND	ID0200	/GET STOP-GO BIT
TAD	ID0002	/PRESERVE DECTAPE ERROR FLAGS
DTXA		/STOP TAPE
JMP	I IDTAPE	/EXIT
IDWC, IDCA, ID0010,	7754 7755 10	/WORD COUNT FOR DATA BREAK /CURRENT ADDRESS FOR DATA BREAK /SEARCH FUNCTION BIT

ID0400, 400	/ FORWARD-REVERSE BIT
ID0130, 130	/ USED TO BUILD READ AND WRITE
	/ CODE
ID0002, 2	
IDCODE, 0	

### **Software**

Four types of programs have been developed as DECTape software for the PDP-8/E:

- a. Subroutines which the programmer may easily incorporate into a program for data storage, logging, data acquisition, data buffering (queuing), etc.
- b. A library calling system for storing named programs on DECTape and a means of calling them with a minimal size loader.
- c. System software which provides for storing, assembling, and editing of programs on DECTape, thereby greatly increasing the versatility and flexibility of the PDP-8/E.
- d. Programs for preformatting tapes controlled by the content of the switch register to write the timing and mark channels, to write block formats, to exercise the tape and check for errors, and to provide each of maintenance.

Program development has resulted in a series of subroutines which read or write any number of DECTape blocks, read any number of 129-word blocks as 128 words (one memory page), or search for any block (used by read and write, or to position the tape). These programs are assembled with the user's program and are called by a JMS instruction. The program interrupt is used to detect the setting of the DECTape flag, thus allowing the main program to proceed while the DECTape operation is being completed. A program flag is set when the operation has been completed. Thus, the program effectively allows concurrent operation of several input/output devices along with operation of the DECTape system. These programs occupy two memory pages (400 (octal) = 256 (decimal) words).

The library system has the following features: First, the computer state remains unchanged when it exits. Second, the library calls programs by name from the keyboard and allows for expansion of the program file stored on the tape. Finally, the library conforms to existing system conventions, namely, that all of memory except for the last memory page (7600 (octal)—7777 (octal)) is available to the programmer. The PDP-8/E DECTape library system is loaded by a 17 (decimal)—instruction bootstrap routine that starts at address 7600 (octal). This loader calls a larger program into the last memory page, whose function is to preserve on the tape the content of memory from 6000 (octal) through 7577 (octal), and then load the INDEX program and the directory into those same locations. Since the information in this area of memory has been preserved, it can be restored when operations have been completed. The basic system tape contains the following programs:

- a. INDEX: Typing this word causes the names of all programs currently on file to be typed out.

- b. UPDATE: Allows the user to add a new program to the files. UPDATE queries the operator about the program's name, its starting address, and its location in core memory.
- c. GETSYS: Generates a skeleton library tape on a specified DECTape unit.
- d. DELETE: Causes a named file to be deleted from the tape.

Starting with the basic library tape, the user can build a complete file of his active programs and continuously update it. One of the uses of the library tape may be illustrated as follows:

The programmer may call the PDP-8/E FORTRAN compiler from the library tape and with it compile the program, obtaining the object program. The FORTRAN operating system may then be called from the library tape and used to load the object program. At this time the library program UPDATE is called, the operator defines a new program file (consisting of the FORTRAN operating system and the object program), and adds it to the library tape. As a result, the entire operating program and the object program are now available on the DECTape library tape.

The DECTape system software is permanently stored on DECTape, from which it can be rapidly loaded. Any systems programs such as the assemblers (XPAL and XMACRO), the Symbolic Editor (XEDIT), or the Binary Loader (XLOAD), can be loaded in less than one minute.

The system software uses a standard DECTape format. There are 128 (200 octal) words per block and 1464 (2701 octal) blocks, so the user has the remaining 1336 blocks for rapid access storage of his own programs.

The primary advantage for users are:

- a. Efficient use of high-speed transfer rates between DECTape and core memory.
- b. Symbolic programs may now be stored, edited, and assembled on DECTape, greatly increasing the versatility and flexibility of the PDP-8/E.
- c. The computational workload can be more than doubled, compared to high-speed paper tape systems.

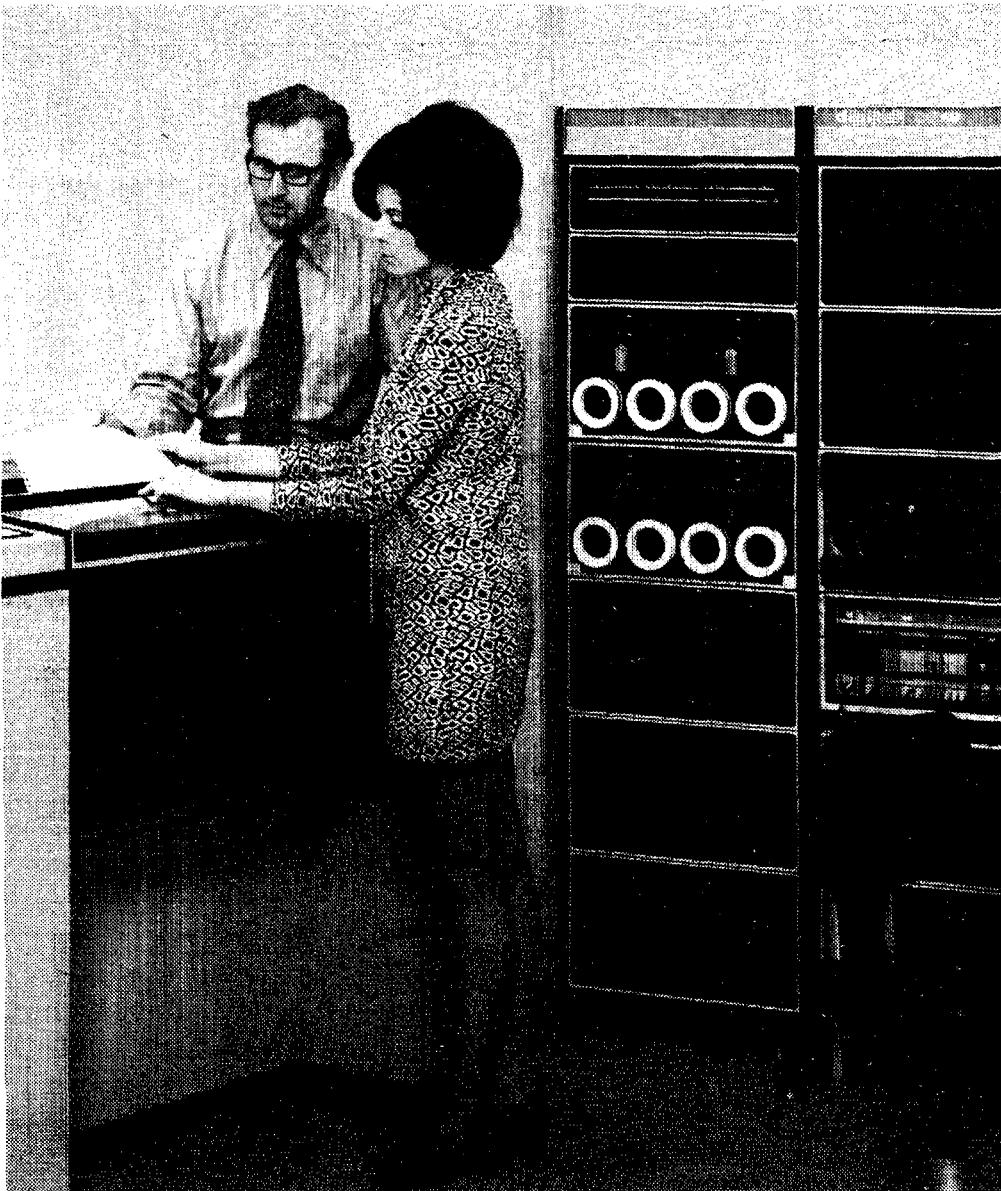
User's programs are written exactly as before for assembly by the PAL or MACRO-8 Assemblers. Using the Symbolic Editor, source programs are typed directly onto DECTape. After assembly, fast symbolic debugging can be done with DDT-8—after loading the program symbol table into DDT with the symbolic loader, XSYM.

The Binary Loader (XLOAD) can load the assembled binary program directly from the DECTape for program execution. Source files, symbol table, and program listings can be stored on DECTape and listed later, if desired. A duplicating program, XDUP, is available for copying programs.

This DECTape system also includes system calls to load any program from DECTape, to update or delete source files, and to restore the system for use by another programmer.

Although the system operates with one DECTape, a two DECTape configuration is strongly recommended as it will permit duplication of programs and saving of back-up master tapes. In a single DECTape system, if the system library is accidentally destroyed, the stored data cannot be replaced immediately because there is no means of recovery.

The last group of programs, called DECTOG, is a collection of short routines controlled by the content of the switch register. It provides for the recording of timing and mark channels and permits block formats to be recorded for any block length. Patterns may be written in these blocks and then read and checked. Writing and reading is done in both directions and checked. Specified areas of tape may be "rocked" for specified periods of time. A given reel of tape may thus be thoroughly checked before it is used for data storage. These programs may also be used for maintenance and checkout purposes.



## **DATA ACQUISITION PERIPHERALS**

Digital Equipment Corporation manufactures a number of data acquisition and control subsystems designed for use with the PDP-8/E. In subsystem can be purchased initially and expanded as required. Three major analog multiplexer subsystems are available, covering analog input ranges from 10 mv to 100 V and having channel capabilities from 4 to 1024 channels. In addition, two types of Digital-to-Analog Converter subsystems are provided as options. A Universal Digital Controller subsystem is also offered as an option. This subsystem provides capabilities for controlling or interrogating up to 3072 discrete digital loads or sources.

### **AD01-A 10 (or 11)-Bit Analog-to-Digital Converter**

The AD01-A is a flexible, low cost, multichannel analog acquisition option for the PDP-8/E computer. The standard AD01-A consists of an expandable solid-state multiplexer, channel selection circuits for up to 32 channels, a programmable input range selector, a 10-bit A/D converter, control, and interface, and power supply. Module additions can be used to expand the multiplexer in four-channel groups up to 32 channels. (Optional 32 channel expander available).

The AD01-A interfaces with the external bus of the PDP-8/E computer to provide ten-bit digitization of unipolar analog signals having a full-scale range of 0V to +1.25V, +2.5V, +5V or +10V. An optional sign-bit addition permits eleven-bit bipolar operation. A programmable input range selection extends the dynamic range of the AD01-A (at moderate sampling rates) to an equivalent of 13 bits for unipolar inputs or to 14 bits for bipolar inputs. An optional sample and hold amplifier is also available to reduce the conversion aperture to 100 ns. Each multiplexer channel switch utilizes an enhancement mode MOSFET that is normally open when unselected, or when power is off. These switches provide overload protection for up to + or - 20V, and signal protection against short circuits.

Operation of the AD01-A is controlled by program instructions from the computer. An ADSC instruction selects the multiplexer channel, system gain, and the interrupt or noninterrupt mode as defined by an AC word.

The selected channel input is connected to the programmable-gain amplifier, which scales the analog input for a 0 to +10V output that is provided to the A/D converter summation junction, either directly or via the sample-and-hold. The ADSC instruction also clears the A/D Done Flag and initiates a conversion cycle. The conversion is performed by successive approximation. For standard unipolar AD01-A, the analog input results in a 10-bit binary output code. The sign-bit option permits conversion of bipolar inputs (0 to + or - 1.25V, + or - 2.5V, + or - 5V, or + or - 10V) to an eleven-bit, two's complement code with an extended sign format. For this format, AC bits 0 and 1 are connected to the same source and denote polarity of the analog input (1 = -V, 0 = +V).

When the conversion is complete, the A/D converter sets its A/D Done Flag. This flag is sensed by an ADSF instruction and, if set, causes the next program instruction to be skipped so that the converter word can be transferred to the AC, using an ADRB instruction. The ADRB instruction also clears the A/D Done flag to ready the AD01-A for another cycle.

The above operations can also be implemented using microprogrammed IOT instructions for "best sampling" operation.

### Specifications

<b>Resolution</b>	Unipolar 10 bits, or 1 part in 1024
<b>Converter Accuracy</b>	Bipolar (option) sign + 10 bits $\pm 0.1\%$ of 10V full-scale input $\pm 0.125\%$ of full-scale with sample & hold $\pm \frac{1}{2}$ least significant bit
<b>Quantizing Error</b>	22 $\mu$ s including channel & gain selection unipolar; 29 $\mu$ s bipolar.
<b>Thrput Rate</b>	100 ns
<b>S &amp; H Aperture</b>	0.1 $\mu$ s with sample and hold
<b>Sample &amp; Hold Acquisition</b>	5 $\mu$ s to .01% of full-scale step change
<b>No. of Analog Inputs</b>	4 minimum, expandable to 32 in groups of 4 Channels (64 channels available)
<b>Input Impedance</b>	1000 megohms in parallel with 20 pf.
<b>Input Isolation</b>	Enhancement mode MOSFET switches, "off" when unselected or power off.
<b>Channel Selection (program selectable)</b>	5-bit address
<b>Input Voltage Range (program selectable)</b>	Unipolar: 0 to + 1.25, + 2.5, + 5.0, + 10.0V full-scale Bipolar (option): 0 to $\pm 1.25$ , $\pm 2.5V$ , $\pm 5.0$ , $\pm 10.0V$ CRC and VRC (compute): 1.4 $\mu$ s full-scale
<b>Overload Capability</b>	$\pm 20V$ on all ranges without damage
<b>Cross Channel Attenuation</b>	78 db, DC-80Hz for 20 volts p-p signals, 100 ohm source impedance
<b>Input Gain</b>	Program selectable

### Accuracy Specifications

Program Selected					Comments
Gain	$\times 1$ +10V	$\times 2$ +5.0V	$\times 4$ +2.5V	$\times 8$ +1.25V	
Full Scale Input	0.05%	0.05%	0.05%	0.05%	Selectable Unipolar
Gain Accuracy					% of full-scale for 30 days
Resolution	9.8mv 550	4.9mv 300	2.45mv 175	1.22mv 110	per bit
Zero Drift					uv/degreeC RTI

<b>Zero drift w/sample &amp; hold</b>	750 2mv	400 1mv	225 1mv	140 1mv	uv/degreeC RTI 0-0 RTI
<b>Noise</b>					
<b>w/sample &amp; hold Repeatability</b>	+-.05% 2.4mv	1.2mv +-.05%	1.2mv +-.05%	1.2mv +-.1%	0-0 25 degree C and 3 sigma confidence
<b>Sign Bit</b>	Adds one bit				
<b>Word Length</b>	Control word 12 bits				
<b>Modes of Operation</b>	Output word 12 bits				
	Interrupting/noninterrupting (program-selectable)				
<b>Environmental</b>	0 to + 55 degrees C operating				
	-25 degrees to + 85 degrees C storage to				
	90% without condensation				
<b>Power</b>	115/230V + or - 10%				
	47 to 470 Hz, single phase less than 50W				

### **Programming**

The following instructions are associated with AD01A operation:

#### **Skip on A/D Done Flag (ADSF)**

Octal Code: 6531  
 Execution Time: 2.6  $\mu$ s  
 Operation: Skips the next program instruction if the A/D Done Flag is set.

#### **Read A/D Buffer (ADRB)**

Octal Code: 6532  
 Execution Time: 2.6  $\mu$ s  
 Operation: Transfers the content of the A/D buffer to AC0 through AC11 and clears the A/D Done Flag.

#### **Convert Analog Input (ADCV)**

Octal Code: 6534  
 Execution Time: 2.6  $\mu$ s  
 Operation: Clears the A/D Done Flag and initiates a conversion operation.

#### **Select Multiplexer Channel and Gain (ADSC)**

Octal Code: 6535  
 Execution Time: 3.6  $\mu$ s  
 Operation: Loads the content of AC into multiplexer input register and implements channel and gain selection. Also clears the A/D Done Flag and initiates a conversion.

**Read A/D Buffer, Clear Flag and Start Conversion (ADRC)**

Octal Code: 6536

Execution Time: 3.6 $\mu$ s

Operation: Jam transfers content of A/D buffer to AC0-AC11, clears flag, and starts a new conversion.

**Select Channel and Gain and Read A/D Buffer (ADSR)**

Octal Code: 6537

Execution Time: 4.6 $\mu$ s

Operation: Transfers the content of the AC to the multiplexer input register, clears the AC, and transfers the content of the A/D buffer to the AC. The A/D Done Flag is then cleared, and another conversion is initiated.

**NOTE**

The ACRC (6536) and ADSR (6537) instructions are used for "burst sampling" conversions in the noninterrupt mode.

A program service routine for the AD01-A can be written as follows:

ADSC	/SELECT CHANNEL, GAIN AND MODE AND START /CONVERSION
ADSF	/SKIP ON A/D DONE FLAG
JMP.—1	/JUMP BACK AND TEST FLAG AGAIN
EXIT	/READ A/D BUFFER

**AFC8 Low-Level Analog Input Subsystem**

The AFC8 is a computer-based unit that multiplexes up to 1024 analog inputs, selects gain, and performs a 12-bit analog-to-digital conversion. Analog inputs can be provided by thermocouples or strain-gain sources having a source impedance of 0 to 500  $\Omega$ . Both channel selection and gain of the multiplexer are under control of the computer; thus, the multiplexer can handle combinations of low-level and high-level analog inputs having a range from 10 mV dc full scale to 100V full scale. Channel sampling is performed at a maximum rate of 200 channels per second. Field wiring terminates on screw terminal connectors, and requires only simple 2-wire twisted pair inputs.

Analog inputs are converted to 12-bit digital words (11 bits plus sign) for transfer to the PDP-8/E computer. Sampling of analog inputs is initiated by the computer issuing IOT instructions. Two IOT instructions are normally required for each channel sample. The first instruction defines the multiplexer gain for the channel and loads a three-bit gain

word from the AC into a multiplexer register. The second instruction defines one of 1024 possible channels for the sample and loads an 11-bit address from the AC into a register. This instruction also starts a multiplexer timing cycle, in which system gain is established, the channel is selected, and the analog output is made available to the A/D converter. When the A/D converter completes the conversion, it loads the 12-bit word into a buffer register, terminates multiplexer sampling, and sets its device flag. The computer senses the state of the device flag by issuing ADSF instructions. When the flag is set, the A/D converter returns a skip request and the computer issues an ADRB instruction to transfer the digital word to AC00 through AC11. This instruction also clears the A/D flag to ready the device for another conversion.

The subsystem is housed in 19-inch industrial type (H964) cabinets that have their own cooling and low-voltage power supplied. Four of these cabinets are required for the maximum channel capability of 1024 channels. The multiplexer is connected to the PDP-8/E computer using a Positive I/O Bus Interface module.

### Specifications

Analog Input Voltage Ranges	$\pm 10$ mV full scale $\pm 50$ mV full scale $\pm 100$ mV full scale $\pm 200$ mV full scale —200 mV to + 500 mV full scale —200 mV to + 1 V full scale —200 mV to + 5 V full scale —200 mV to + 10 V full scale —200 mV to + 100 V full scale
Analog Input Current Ranges	$\pm 1$ mA full scale $\pm 5$ mA full scale $\pm 10$ mA full scale $\pm 50$ mA full scale
A/D Converter Output Word	Parallel, 12 bits (11 bits plus sign)
Resolution	5 $\mu$ V
Accuracy	$\pm 25\mu$ V or $\pm 0.05\%$ of f.s., whichever is greater
Common Mode Rejection	120 dB or greater above 60 Hz
Common Mode Voltage	200V
Normal Mode Rejection	50 dB or greater at 60 Hz
System Sampling Rate	200 channels per second (max.)
Single Channel Sampling Rate	20 samples per second max. at stated accuracy
Expansion Capabilities	System can be expanded in groups of 8 channels up to a maximum of 1024 channels.

Internal A/D Conversion 50  $\mu$ s  
Time  
Resolution Sign + 11 bits

**Programming**

Instructions for multiplexer operation are listed below.

**Set Multiplexer Gain (ADSG)**

Octal Code: 6542  
Execution Time: 2.6  $\mu$ s  
Operation: Loads a three-bit gain word from AC09 through 11 into a multiplexer register for selection of system gain for channel.

**Set Multiplexer Address (ADSA)**

Octal Code: 6544  
Execution Time: 2.6  $\mu$ s  
Operation: Loads an eleven-bit address from AC01 through 11 into a multiplexer register for selection of a channel. Also starts multiplexer timing to select gain and channel and provide analog sample to A/D converter.

**Skip on A/D Flag (ADSF)**

Octal Code: 6531  
Execution Time: 2.6  $\mu$ s  
Operation: Senses A/D converter flag. If flag is a one, increments the PC, and skips the next sequential instruction so the A/D converter can be serviced.

**Read A/D Converter Buffer (ADRB)**

Octal Code: 6534  
Execution Time: 2.6  $\mu$ s  
Operation: Transfers twelve-bit word in A/D buffer to AC00 through AC11, and resets A/D converter flag.

A program for selecting multiplexer gain and channel and transfer of digital word to the computer can be written as follows:

```
TAD XX
ADSG    /LOAD MULTIPLEXER GAIN WORD
CLA
TAD YY
ADSC    /LOAD MULTIPLEXER ADDRESS WORD
ADSF    /SKIP IF A/D CONVERTER FLAG IS 1
JMP .-1  /JUMP BACK AND SENSE FLAG AGAIN
```

ADRB /READ A/D BUFFER AND TRANSFER CONTENTS TO  
/AC00-11. CLEAR A/D CONVERTER FLAG.

**Type AF04A Guarded Scanning Integrating Digital Voltmeter**

The Type AF04A is a Guarded Scanning Integrating Digital Voltmeter system, with wide dynamic range and high common-mode rejection, and is capable of expansion to 1000 channels. The AF04A is used with the PDP-8/E to multiplex up to 1000 three-wire analog channels into a six-decimal digit integrating digital voltmeter (IDVM). Each digit is BCD-coded for input and display by the IDVM. Full scale ranges are from + or - 10 mV to + or - 300V, with automatic ranging, 300 percent overranging, and a usable 5  $\mu$ s resolution. Guarded input construction and active integration assist in attaining an effective common-mode rejection of greater than 140 dB at all frequencies. (Normal mode rejection is infinite at multiples of power line-frequency.)

This system is ideally suited for data acquisition of process monitoring where a wide range of signals requires large dynamic range. The 10mV range has 0.001 percent resolution, and, coupled with a common-mode noise rejection greater than 140 dB at all frequencies, allows accurate direct measurement of thermocouples, strain gauges, load cells, and other low-level transducers without additional amplification.

The AF04A IDVM, operated under program control, is capable of either random channel selection or sequential channel selection. The computer selects either program-controlled ranging (for fastest speed) or auto-ranging, as well as the integration time of the integrating digital voltmeter.

The digitized data, as well as the current channel address, is read by the computer in either two or three bytes.

A decimal display of the digitized value, including sign and decimal location, is continuously displayed on the front panel. The current channel number is also displayed. Front-panel controls on the IDVM allow for manual setting of all the programmed functions. A front-panel control allows continuous display of the internal secondary standard, which can be prewired to a particular channel for reference checking during normal operation. The AF04A may be manually controlled, completely independent of the computer.

**Specifications**

Full scale + or -	10mV, 100mV, 1V, 10V, 100V, 300V and automatic ranging
Over-ranging	300% on all but highest range
Maximum Input Voltage	300V
Resolution	5 $\mu$ V (usable), 0.1 $\mu$ V (LSB)
Accuracy (overall worst case with daily calibration)	+ or - 0.004% of reading or + or - 0.01% of full scale
Temperature Stability (RMS full scale and zero drift)	+ or - 0.006%/day

Temperature coefficient	+ or - 0.003% of reading/degrees C
Full scale	+ or - 0.002% of full scale/degrees C
Zero	(+ or - 0.006% of full scale/degrees C on 10mV and 1V range)
Line voltage stability	+ or - 0.005%/10% change
Maximum common-mode voltage	+ or - 300V from power line ground
Common-mode rejection (166.6ms integration period and 1000-ohm source unbalance)	>140 dB at all frequencies
Normal-mode rejection	Infinite at multiples of line frequency
Input impedance	
10, 100, 1000 mV ranges	1000 megohms/V
10, 100, 300V ranges	10 megohms
Internal secondary standard	
Value	+ or - 1.000V
Accuracy	+ or - 0.002% traceable to National Bureau of Standards
Stability	+ or - 0.005%/month
Temperature coefficient	negligible

#### Selected Resolution

DC		0.001%		0.01%		0.1%	
Voltage	Maximum Reading	Resolu-	Maximum Reading	Resolu-	Maximum Reading	Resolu-	
Range		tion		tion		tion	
10mV	30.0000mV	0.1 $\mu$ V	030.00mV	1 $\mu$ V	0030.00mV	10 $\mu$ V	
100mV	300.000mV	1 $\mu$ V	0300.00mV	10 $\mu$ V	00300.0mV	100 $\mu$ V	
1000mV	3000.00mV	10 $\mu$ V	03000.0mV	100 $\mu$ V	003000.mV	1mV	
10V	30.0000V	100 $\mu$ V	030.000V	1mV	0030.00V	10mV	
100V	0300.000V	1mV	0300.00V	10mV	00300.0V	100mV	
1000V*	0300.00V	10mV	00300.0V	100mV	000300.V	1V	

\*1000V range is scanner-limited to 300V peak maximum.

#### Scanning Speed (Programmed Range)

Resolution	Integration Time	Total Time	Speed Scanning
0.1%	1.6 ms	20 ms	50 ch/s
0.01%	16.6 ms	40 ms	25 ch/s
0.001%	166.6 ms	188 ms	5 ch/s

Scanning Speed (Auto-Range)—Add 6-36 ms, depending on per-channel voltage span.

#### Programming

The IOT instructions associated with the scanning IDVM are designed to minimize the computer overhead associated with this option, while retaining maximum program controlled flexibility. The IOT instructions are:

### Select Range and Gate (VSEL)

Octal Code: 6542

Execution Time: 2.6  $\mu$ s

Operation: Transfers the contents of the accumulator to the AF04A control register. Control Word 1 is used only if a range change is required (see Figure 7-17).

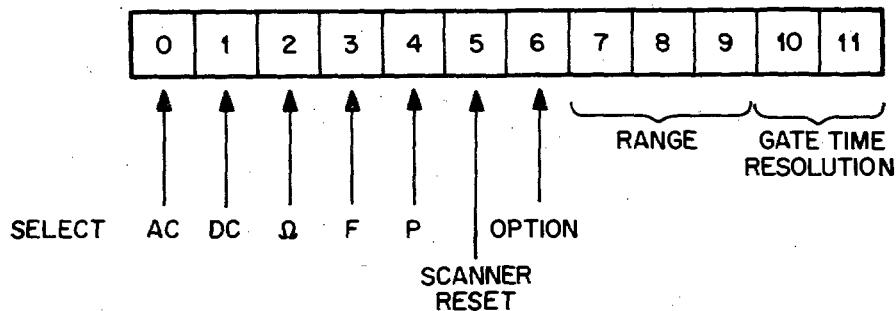


Figure 7-24 Control Word 1 (from Computer)

### Select Channel and Convert (VCNV)

Octal Code: 6541

Execution Time: 2.6  $\mu$ s

Operation: Transfers the contents of the accumulator to the AF04A channel address register. Automatically digitizes the analog signal on the selected channel (see Figure 7-18).

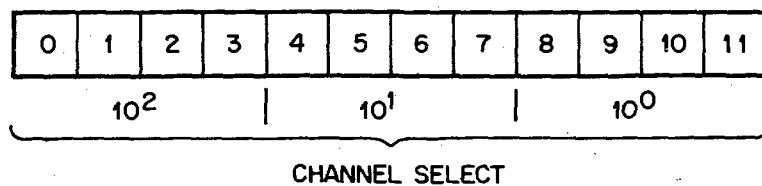


Figure 7-25 Control Word 2 (from Computer)

### Index Channel and Convert (VINX)

Octal Code: 6544

Execution Time: 2.6  $\mu$ s

Operation: Increments the last channel address by one and automatically digitizes the analog signal on the selected channel. The contents of the control register are unchanged.

### **Byte Advance (VBA)**

Octal Code: 6564

Execution Time: 2.6  $\mu$ s

Operation: The total data word from the AF04A is 36 bits long. The first data word after the flag is set is always the 12 most significant bits. The BYTE ADVANCE command requests the next 12 most significant bits. When the data is available, the data ready flag is set again. To select the 12 least significant bits, a second BYTE ADVANCE command is required. When the data is available, the data ready flag is set again.

### **Sample Current Channel (VSCC)**

Octal Code: 6571

Execution Time: 2.6  $\mu$ s

Operation: Digitizes the analog signal on the current channel. This command is not required except when multiple samples are required on any channel. (Using this command on a preselected channel saves up to 10 ms per sample.)

### **Frequency and Period Measurement Options for AF04A**

A separate input permits the IDVM to be used as a frequency counter capable of counting to 2 mHz with selectable gate times of 1, 10, and 100 ms, providing measurement resolution of 10 Hz. Increased accuracy at low frequencies (to 10 kHz with automatic 250% overranging) is accomplished with the period-measurement mode. This mode counts an internal frequency source for 1, 10 or 100 periods of the frequency being measured, thereby providing increased full-scale accuracy. Period readout is in milliseconds. Frequency and voltage measurements may be made within one scanning cycle by grouping all frequency inputs in one master or slave scanner and all voltage inputs in another master or slave scanner. The output of one scanner may then be connected to the frequency-input connector of the IDVM, and the output of the other scanner to the voltage input. One of the optional control word bits is used to program the IDMV for frequency or period measurements.

### **Specifications (See Figure 7-29)**

#### **Frequency Measurements**

Range: 10 Hz to 2 mHz

Sensitivity: 100 mV rms or -1V pulses, at least 0.3  $\mu$ s wide at 50% points. 100V rms maximum working voltage.

Input Impedance: 22K shunted by less than 1000 pF, including internal cabling.

Time Base: 100 kHz crystal oscillator with initial accuracy of + or - 0.0005%, long-term stability + or - 0.001%/wk; temp. coefficient + or - 0.0002%/degrees C.

#### **Period Measurements**

Range: 1, 10, and 100 period average. Input frequency from 10 Hz to 25 kHz sine wave or 0.1 pps to 25,000 pps.

**Sensitivity:** 100 mV rms or -IV pulses, at least  $0.3\mu s$  wide at 50% points. 100V rms maximum working voltage.

**Input Impedance:** 22K shunted by less than 1000 pF, including internal cabling.

**Accuracy:**  $\pm 1$  count + time base accuracy + trigger error. Trigger error  $< \pm 0.03\%$  for 100 mV rms sine wave with 40 dB signal-to-noise ratio.

**Time Base:** 100 kHz crystal oscillator with initial accuracy of  $\pm 0.0005\%$ , long-term stability  $\pm 0.0001T/wk$ ; temp. coefficient  $\pm 0.0002\%/\text{degrees C}$ .

Selected Resolution

Selected Resolution	0.001%		0.01%		0.1%	
Function	Maximum Reading	Resolution	Maximum Reading	Resolution	Maximum Reading	Resolution
Frequency	2000.00kHz	10Hz	02000.0kHz	100Hz	002000kHz	1kHz
Period	99.9999msec	0.1 $\mu$ s	999.999msec	1.0 $\mu$ s	9999.99msec	10 $\mu$ s

Figure 7-29

### **Additional AF04-A Options**

A type AF04-X expansion Mounting Panel is available which provides an additional 200 channels. For each 10 channels implemented, the Type AF04-S 10-Channel Guarded Reed Relay Multiplexer Switch is required.

Thermocouple reference junctions

Extended scanner for more than 1000 channels

### **AA50-A Digital-To-Analog Conversion Subsystem**

The AA50-A DAC is a general-purpose, program-controlled DAC subsystem that converts 12-bit (11 bits plus sign) words into analog outputs having a continuously adjustable full-scale range of 0 to  $\pm 10V$  at 10 mA.

The AA50-A is housed in a H911 type mounting panel and is furnished complete with power supply, I/O cables, control and interface logic, and up to six DAC modules, each providing one analog output. The unit interfaces with the external bus of the PDP-8/E. All operations are controlled by IOT instructions, including the selection of the DAC module to receive the 12-bit output word. Each DAC module contains a buffer register and a scaling amplifier with reference mounted on the same module.

For an output function, the computer issues an IOT instruction that specifies the DAC module to receive the 12-bit word. The control logic of the AA50-A decodes the IOT, performs input gating for the 12-bit word from AC0-11, and loads the words into output buffer of the designated DAC module. The word remains in the output buffer until the buffer is updated by another input; thus, the resulting analog output is available until updating occurs.

### **Specifications**

Digital Input	Parallel, 11 bits plus sign in two's complement form
Coding	3777 (octal) = + 10V 0000 (octal) = 0V 4000 (octal) = - 10V
Standard Analog Output	0 to $\pm 10V$ @ 10 ma (adjustable)
Settling Time	20 $\mu$ s to $\frac{1}{2}$ LSB (measured at output connector with no capacitive loading)
Accuracy	0.05% of full scale
Linearity	$\pm \frac{1}{2}$ LSB ( $\pm 2.44mV$ for $\pm 10V$ DAC output)
Capacitive Loading	0.1 $\mu$ f at output connector will not cause instability

## **Programming**

The following instructions are associated with AA50-A operation:

### **Select DAC 0 (DACS0)**

Octal Code: 6551  
Execution Time: 2.6  $\mu$ s  
Operation: Transfers content of AC to DAC module 1 and converts it to analog output.

### **Select DAC 1 (DACS1)**

Octal Code: 6552  
Execution Time: 2.6  $\mu$ s  
Operation: Transfers content of AC to DAC module 2 and converts it to analog output.

### **Select DAC 2 (DACS2)**

Octal Code: 6553  
Execution Time: 3.6  $\mu$ s  
Operation: Transfers content of AC to DAC module 3 and converts it to analog output.

### **Select DAC 3 (DACS3)**

Octal Code: 6554  
Execution Time: 2.6  $\mu$ s  
Operation: Transfers content of AC to DAC module 4 and converts it to analog output.

### **Select DAC 4 (DACS4)**

Octal Code: 6555  
Execution Time: 3.6  $\mu$ s  
Operation: Transfers content of AC to DAC module 5 and converts it to analog output.

### **Select DAC 5 (DACS5)**

Octal Code: 6556  
Execution Time: 3.6  $\mu$ s  
Operation: Transfers content of AC to DAC module 6 and converts it to analog output.

Device codes 56 and 57 are used when additional (up to three total) AA50's are required.

## **AA05-A/AA07 Digital-to-Analog Converter and Control**

The AA05 Digital-to-Analog Converter (DAC) provides housing power and control for up to 24 10-bit DAC modules. The AA07 Expansion Unit extends the capacity of the system to 64 channels of DAC.

Each conversion channel may use any of four printed circuit card DAC modules. These modules include two single-buffered units, Types A608 and A609, and two double-buffered units, Types A610 and A611. A608 is a single-buffered, 10-bit DAC, with unipolar output (0V to + 10V). Type A609 is a single-buffered, 10-bit DAC with bipolar output and variable offset. A610 and A611 are similar to A608 and A609, respectively, except that the former are double-buffered units.

The principal power supply furnishes all power for up to 64 DAC modules, with the exception of the -10V reference power. Reference power is furnished by the Type H706 Reference Power Supply, which is optional to the AA05/AA07 unit. A maximum of five H706 supplies can be allocated to the various DAC channels, two of which are in the AA05 and three of which are in the AA07.

Each DAC in the AA05/AA07 DAC and expansion unit are used with the PDP-8/E computer to control up to 64 DAC channels. Both the DAC address and the digital word to be converted are program-controlled as two I/O data words for 12-bit computers. The DAC address is stored in the AA05 and remains there until changed by the program for fast updating of any channel.

Six indicators on the front panel of this device indicate the binary address of the DAC channel currently being addressed. All data bits and I/O transfer commands are buffered to present a minimum load to the computer bus even with 64 DACs in use. The AA07 expansion assembly allows expansion to 64 single- or double-buffered DACs.

The AA05/AA07 consists of a 10-bit buffer register, level converters, a precision divider network, and a current-summing amplifier capable of driving large external loads. Provisions are made for double-buffering and bipolar output voltage where required. A precision reference voltage, supplied externally by the H706 power supply, ensures greater efficiency and optimum scale-factor matching in multiple-channel systems. The AA05/AA07 DAC utilizes four separate instructions. These instructions clear the DAC address register, transfer the contents of AC(0-9) to the input register of the selected DAC, and update all double-buffered channels (if applicable).

### Specifications

Standard Output	Unipolar, 0V to + 10V at 10 ma
Optional Output	Bipolar, + or - 5V or + or - 10V
Output Impedance	Less than 1 ohm
Temperature Coefficient	0.1mV/degrees C plus temperature coefficient of reference supply (worst-case for DEC reference supply $\pm 0.6\text{mV}/\text{degrees C}$ )
Resolution Accuracy	0.1% of full-scale + or - 5mV
Settling Time (Full-scale)	5 $\mu\text{s}$ for 1 DAC module. Less than 100 $\mu\text{s}$ for up to 12 DAC modules
Environmental Power	0 degrees to 50 degrees C 7A (max) at 115V, 60Hz

## **Programming**

The following instructions are associated with the AA05A DAC:

### **Clear DAC Address (DACL)**

Octal Code: 6551  
Execution Time: 2.6  $\mu$ s  
Operation: Clears DAC address register.

### **Load DAC Address (DALD)**

Octal Code: 6552  
Execution Time: 2.6  $\mu$ s  
Operation: Loads content of AC in DAC address register.

### **Load DAC Input Register (DALI)**

Octal Code: 6562  
Execution Time: 2.6  $\mu$ s  
Operation: Loads content of AC in DAC input register specified by DAC address register.

### **Update All Channels (DAUP)**

Octal Code: 6564  
Execution Time: 2.6  $\mu$ s  
Operation: Updates all double-buffered channels to provide DAC outputs to loads.

## **Universal Digital Controller (UDC)**

The UDC is a digital input/output system with a controller having 256 12-bit addressable channels. Each channel can be used as an input or output path. When used for output functions, a channel can control 12 discrete off/on devices such as relays, flip-flops, etc. When used for input functions, a channel can be used to interrogate the status of 12 discrete off/on sources such as switches, relays, and flip-flops. Thus, the UDC provides the capability for accessing a total of 3072 discrete digital points either for input (status) or output (control) functions in 12-bit combinations.

All input/output data is handled in the form of 12-bit words. The data is unstructured except for the generic module type and address word read to the computer after an interrupt. Accumulator bits 0 through 3 receive a four-bit code denoting the generic type or function performed by the module specified and by the eight-bit address in AC04 through AC11.

Any UDC channel or word can be input or output. When dedicated for an input function, the type of interrupt desired must be specified by the program. The type of interrupt is defined by AC10 and AC11 as follows:

AC10	AC11	TYPE OF INTERRUPT
0	0	None
0	1	Deferred processing
1	0	Immediate processing
1	1	Both

Once an interrupt type is selected and an interrupt occurs, the UDC locates the interrupting address, using an address scan cycle. This cycle requires to 20  $\mu$ s. Once the interrupting address has been located, the address and module generic type are made available to the computer.

The functional capability to EXCLUSIVE OR an AC word with an I/O word is provided by Change-Of-State (COS) gating. The AC bits are loaded into the COS register with the Load Previous Status (octal 6357) IOT. Data from the word of I/O presently addressed is at the gates and the EXCLUSIVE-OR function is performed.

Two bits, pulse open and pulse close, are hard-wired at the I/O word in question; their purpose is to mask out data changes that are not pertinent. The EXCLUSIVE-OR function is defined by the following:

DATA BIT	AC BIT	PULSE OPEN	PULSE CLOSE	COS OUT
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

The UDC is housed in an H964 industrial-type cabinet(s) complete with cooling fans and low-voltage supplies. The smallest configuration of the UDC consists of one file in a single cabinet. This file contains the IOT and interface control logic; the address scanner logic, and capabilities for handling up to four I/O channels or words. This basic system can be expanded from four channels to 256 channels in four-channel groups by adding system units, file and cabinets.

### Specifications

Operating Modes	Digital Output Digital Input Interrupt or Noninterrupt
Data Format	Parallel, 12-bit unstructured
Addressing Capability	256 12-bit channel or words or 3072 discrete digital points

Input/Output Module Selection	Directly addressable and location independent
Interrupt Module Identification	4-bit Generic code type
Interrupt Structure	Immediate or deferred by module assignment and program
Interrupt Scan or Address Location Time	5 $\mu$ s typical
I/O Data Rate	105 12-bit word per second
System Clock Rates	3 clock rates available to each I/O channel or word (1) 60 Hz, 6.3 VAC (line power) (2) 175 Hz, to 1.75 kHz (3) 1.75 Hz to 17.5 kHz
Standard Output Drive Capabilities	250 ma at up to +55V (suitable for relay drivers)
Standard Inputs	2 amps, 500V, 100VA, (Mercury-wetted relays)
Functional Modules Available	15 ma at +6V. Contact Sense Contact Interrupt Flip-Flop Relay Single-Shot Relay Flip-Flop Driver Single Shot Driver Latching Relay Input/Output Counters Digital to Analog Converters

### Programming

The following instructions are associated with UDC operation:

#### Skip on Scan Not Busy (UDSS)

Octal Code: 6351  
 Execution Time: 2.6  $\mu$ s

Operation: Skips the next instruction if Scan Not Busy flag is a one, denoting that the address scanner has located the interrupt channel, so that UDC can be serviced.

#### Start Interrupt Scan (UDSC)

Octal Code: 6353  
 Execution Time: 3.6  $\mu$ s

Operation: Enables address scan function if interrupt flag is set and interrupt type (immediate or deferred) is present.

### **Read Address and Generic Type (UDRA)**

Octal Code: 6356  
Execution Time: 3.6  $\mu$ s  
Operation: Transfers the generic type and address to the AC after interrupting address has been located.

### **Load Previous Status (UDLS)**

Octal Code: 6357  
Execution Time: 4.6  $\mu$ s  
Operation: Loads content of AC into COS register and reads the result of the EXCLUSIVE OR function of the COS logic to AC.

### **Skip On UDC Flag and Clear Flag (UDSF)**

Octal Code: 6361  
Execution Time: 2.6  $\mu$ s  
Operation: Skips the next instruction and clears the UDC flag if UDC interrupt Flag is set.

### **Load Address (UDLA)**

Octal Code: 6363  
Execution Time: 3.6  $\mu$ s  
Operation: Loads 8-bit address from AC into address register scanner.

### **Enable UDC Interrupt Flag (UDEI)**

Octal Code: 6364  
Execution Time: 2.6  $\mu$ s  
Operation: Sets the interrupt enable flip-flop so that UDC can generate interrupt requests.

### **Disable UDC Interrupt Flag (UDDI)**

Octal Code: 6365  
Execution Time: 3.6  $\mu$ s  
Operation: Clears interrupt enable flip-flop so that UDC cannot generate interrupt requests.

### **Clear AC and Read Data (UDRD)**

Octal Code: 6366  
Execution Time: 3.6  $\mu$ s  
Operation: Clears the AC and transfers data specified by address of address scanner register to AC.

### **Load Data and Clear AC (UDLD)**

Octal Code: 6367  
Execution Time: 4.6  $\mu$ s  
Operation: Transfers content of AC to address specified by address scanner register, then clears AC.

## VW01 WRITING TABLET

### TYPE VW01 Writing Tablet

The VW01 Writing Tablet converts graphical information, in the form of X and Y coordinates, to digital data that can be input to a digital computer. The major components of the VW01 are the writing tablet, spark pen, component box, and computer interface logic.

The user places a sheet of paper on the writing tablet and draws sketches, schematics and hand-written symbols or characters using the special ball-point pen. The sound of the spark emitted by the pen is picked up by microphones located along the X- and Y-axes of the writing tablet. The time lapse, from spark emission until sound is picked up by each bank of microphones, is accurately measured to provide a digital record of the X and Y coordinates of the spark pen location on the paper.

The digitized graphic data is input to a digital computer for immediate or delayed processing.

The VW01 provides an efficient man/machine interface with digital systems that allows the user complete freedom of expression.

The VW01 consists of the writing tablet, VW01-AP interface and BC08B I/O cable. The KA8-E positive I/O Bus is required.

### VW01-MX Multiplex Option

The VW01-MX Multiplex option allows up to four VW01 Writing Tablets to be used with a single VW01 computer interface. This option consists of the VW01-MX Multiplexer and up to four VW01-MA Writing Tablet assemblies. If the VW01-MX Multiplex option is included as part of the system, additional cabling and interface requirements must be considered.

### Modes of Operation

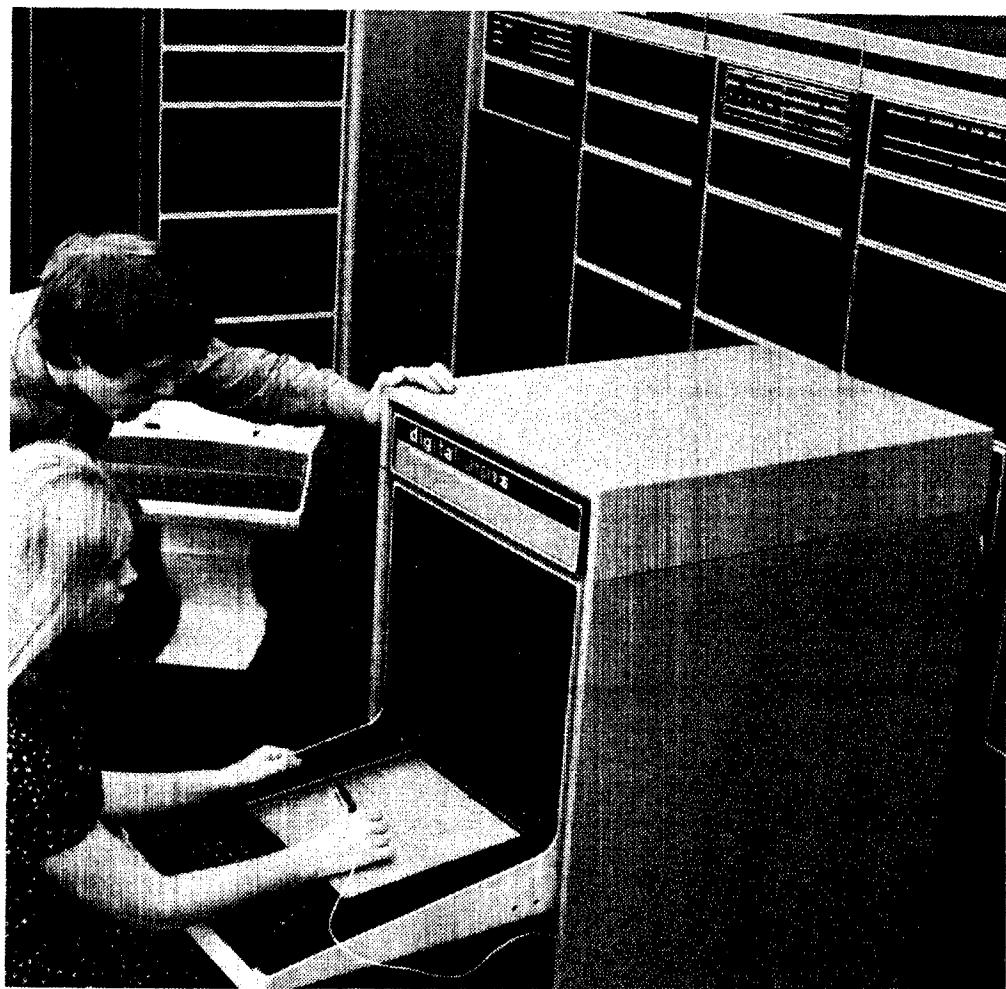
The VW01 operates in either of two modes: Single Point or Data Input.

**Single Point Mode**—In the Single Point mode of operation, a single spark is generated each time the spark pen is pressed against the writing surface. The spark is initiated by the closure of a microswitch within the spark pen. The Single Point mode is used if the operator desires to plot points. For example, to plot points at four different locations, he positions the pen point at each location. Then, by pressing and releasing the pen at each position, the corresponding X-Y coordinate pairs are sensed and digitized.

**Data Input Mode**—In the Data Input mode, a continuous series of sparks are generated at a constant rate, under control of clock pulses. The X-Y coordinate pairs are continuously generated and input to the computer. This mode allows the user to draw continuous lines, circles, curves, etc. that can be displayed on a CRT.

The normal data rate at which X-Y coordinate pairs can be generated is 200 Hz.

The Data Input mode can also be used for tracking applications. Tracking is a technique that is used to move a cursor, or other type of position indicator to a specific X-Y coordinate location on the display. With appropriate programming, the cursor will follow the spark pen movement. The spark pen is then pressed on the writing tablet at a specific X-Y coordinate location to draw on the display.



VW01 Writing Tablet

### **Digitizing the Graphic Data**

At the time a spark is generated, X and Y clock pulses are initiated and the X and Y registers are incremented until the sound of the spark is received by the X and Y microphones. As soon as a microphone detects the sound, the associated X or Y clock pulses are inhibited and the register stops incrementing. The binary numbers contained in the X and Y registers will be directly proportional to the X and Y coordinates of the position at which the spark was emitted.

For example, if a spark is generated at a point on the writing tablet that is four inches from the X-axis microphone and another spark is generated at a point on the writing tablet that is eight inches from the X-axis microphone, the time for the sound wave to travel from the point of generation to the X-axis microphone would be twice as long as the time required for the first spark. Thus the binary number contained in the X register for the second spark generated would be approximately double the value of the binary number for the first spark generated. When the spark pen is moved, a different set of binary numbers, proportional to the new spark pen position are entered into the X and Y registers.

### **Computer Input**

When the binary numbers that represent a pair of X-Y coordinates are settled in the 10-bit X and Y registers, the VW01 computer interface logic requests a program interrupt. When the computer services the interrupt request, the 10-bit digital words specifying each coordinate are successively read into the computer AC by IOT instructions.

### **Specifications**

Component	Dimensions (in.)		
	Height	Width	Depth
Writing tablet	13	13	1.5
Interface logic rack	10.44	19	12
Component box	3.18	19	5

The spark pen is 5.5 inches long and 0.25 inch in diameter

Digital Resolution      10-bit resolution in both X and Y axes.

Graphic Resolution      1000 x 1000 line pairs; 90 lines per inch

Reproducibility      One (least significant) bit in 1000, in both X and Y axes.

### **Drift**

Constant Temperature      With the spark pen stationary, the X and Y registers will not vary more than  $\pm$  one bit in 1024.

4.4° to 32°C      With the spark pen stationary, the X and Y registers will vary  $\pm$  two bits per thousand per degree change Centigrade.

+ 40° to 90°F      With the spark pen stationary, the X and Y register will vary  $\pm$  1.4 bits per thousand pen degree change Fahrenheit.

**Data rate**

FAST SCAN	200 X-Y coordinate pairs per second. The data rate can be decreased to 1 X-Y coordinate pair per second.
SCAN	100 X-Y coordinate pairs per second per tablet; used only with VW01-MX Multiplex option.
Single Point	Determined by user's manual activation of the spark pen microswitch.
Multiplex latency	With the VW01-MX Multiplex option, the interval from each writing tablet DATA READY flag to the time the next writing tablet is enabled is 1.4 msec.

**Spark pen longevity\* (typical)**

Spark gap	50 x 10 <sup>7</sup> discharges, minimum
Ink Cartridge	5000 ft. of inked lines
Writing tablet surface	11 x 11 inches
Input power requirements	115V, 50/60-Hz + 2%, single phase, 17-30A, or 230V, 50 Hz + 2%, single phase, 8-15A.
Operating temperature range	+40 to +90°F (4.4 to 32°C)
Operating humidity range	20 — 55% relative humidity, without condensation.

**Programming**

The following instructions are used to program the VW01.

**Set Tablet Controls (WTSC)**

Octal Code: 6054

Operation: The following functions are cleared by I/O Buffered Power Clear. The Set Tablet Controls IOT, with the appropriate bit set, sets or clears the following functions, depending upon the bit selected.

**ACCUMULATOR BITS**

0	1	2	3	4	5	6	7	8	9	10	11
		Single Point CLR		Pen Data Intr En CLR		Data Ready Intr En CLR		FAST SCAN SET	SCAN	Writing Tablet En CLR	SET

\* The spark gap length of service is extended by using the ON/OFF switch located on the writing tablet.

**Writing Tablet EN—SET**

AC bit 11 = 1

The writing tablet is initially enabled for operation in FAST SCAN. To change to Single Point, SCAN Multiplex, or Single Point, the appropriate function must be selected.

**Writing Tablet EN—CLR**

AC bit 10 = 1

The writing tablet is disabled from performing any control functions.

**SCAN**

AC bit 09 = 1

The writing tablets are enabled to operate in the multiplex mode. Up to four writing tablets can be multiplexed. Each tablet will have a data rate of 100 Hz and the tablets will operate in sequential order.

**FAST SCAN**

AC bit 08 = 1

Enables the logic for the operation of one tablet at a data rate of 200 Hz. Using the Select Tablet IOT with the appropriate bit set, a single writing tablet can be selected for Data Input operation. I/O Buffered Power Clear always selects writing tablets 01, and FAST SCAN.

**Data Ready Intr EN—SET**

AC bit 07 = 1

The DATA READY flag is enabled onto the I/O interrupt bus.

**Data Ready Intr EN—CLR**

AC bit 06 = 1

The DATA READY flag is disabled from the I/O interrupt bus.

**Pen Data Intr EN—SET**

AC bit 05 = 1

The PEN DATA flag is enabled onto the I/O interrupt bus.

**Pen Data Intr EN—CLR**

AC bit 04 = 1

The PEN DATA flag is disabled from the I/O interrupt bus.

**Single Point—SET**

AC bit 03 = 1

The writing tablet is enabled for a single pair of X-Y coordinates. The microswitch in the pen must be activated. An X-Y coordinate pair is present when the DATA READY flag is set.

Single Point can be selected for the multiplex of the writing tablets. When the microswitch in any of up to four pens is activated, the associated tablet takes control of the I/O bus and an X-Y coordinate pair is ready when the DATA READY flag is set. The tablet that set the DATA READY flag will then have to be cleared by using the Select Tablet IOT with the appropriate bit set. All tablets should be cleared before initiating Single Point (Multiplex) operation.

**Right/Left**

This bit indicates the current setting of the RIGHT/LEFT switch. A logical 1 indicates RIGHT and a logical 0 indicates LEFT.

**Single Point**

A logical 1 indicates Single Point mode of operation.

**TAB 01 through TAB 04**

Tablet 01 indicates the writing tablet 01 ON/OFF switch is in the ON position and writing tablet 01 is selected. With the multiplex option, tablet 01 is set by Buffered Power Clear or the Clear All Flags IOT, and writing tablets 02, 03, and 04 are cleared. TAB 02 through TAB 04 are logical 1 only when the associated ON/OFF switch is ON and that writing tablet is selected.

**Pen Data Intr EN**

Indicates the status of the pen data interrupt enable.

**Data Ready Intr EN**

Indicates the status of the DATA ready interrupt enable.

**SCAN**

Indicates whether SCAN or FAST SCAN has been selected.

**Writ Tab EN**

Indicates the status of the writing tablet enable.

**Clear Data Ready Flag (WTCD)**

Octal code: 6061

Operation: This IOT is issued to clear the DATA READY flag.

**Single Point—CLR**

AC bit 02 = 1

The Single Point mode operation will be disabled.

**Read X (WTRX)**

Octal code : 6052

Execution

Time: 2.6  $\mu$ s

Operation: The Read X IOT "OR"s 10 bits from the X register into the processor accumulator.

**ACCUMULATOR BITS**

0	1	2	3	4	5	6	7	8	9	10	11
		X0	X1	X2	X3	X4	X5	X6	X7	X8	X9

Read X Word Format

**Read Y (WTRY)**

Octal code: 6062

Execution

Time: 2.6  $\mu$ s

Operation: The Read Y IOT "OR"s the 10-bit Y register into the processor accumulator. The Y coordinate bits are read into the same accumulator bit positions as indicated for the X coordinate bits.

**Read Status (WTRS)**

Octal code: 6072

Execution

Time: 2.6  $\mu$ s

Operation: The Read Status IOT reads the flag and status indicator bits into the processor accumulator as follows:

**ACCUMULATOR BITS**

0	1	2	3	4	5	6	7	8	9	10	11
DATA RDY FLAG	PEN DATA FLAG	RIGHT = 1 LEFT = 0	Single Point	TAB 01	PEN DATA INTR EN	TAB 02	DATA RDY INTR EN	TAB 03	FAST SCAN = 1 SCAN = 0	TAB 04	WRIT TAB EN

**Status Word Format**

Logical 1 bit indicates condition selected.

**DATA READY Flag**

The DATA READY flag is set when an X-Y coordinate pair is updated to the current position of the spark pen on the writing tablet surface.

**PEN DATA Flag**

The PEN DATA flag is set when an X-Y coordinate pair is updated to the current position of the spark pen and the spark pen microswitch is activated.

**Clear Pen Data Flag (WTCP)**

Octal code: 6051

Execution

Time: 2.6  $\mu$ s

Operation: This IOT is issued to clear the PEN DATA flag.

**Writing Tablet Skip (WTSK)**

Octal code: 6071

Execution

Time: 2.6  $\mu$ s

Operation: The writing Tablet Skip IOT can only be used to perform a computer program skip on a writing tablet I/O interrupt. The two writing tablet flags that can provide an I/O interrupt are the DATA READY flag and the PEN DATA flag. The appropriate flag has to be enabled onto the I/O interrupt bus using the data ready interrupt enable or the pen data interrupt enable.

**Select Tablet (WTSE)**

Octal code: 6074

Execution

Time: 2.6  $\mu$ s

Operation: The Select Tablet IOT is used when the VW01-MX multiplex option is implemented, in conjunction with the TAB 01 through TAB 04 control bits. When FAST SCAN is selected, only one writing tablet can be active, and this tablet can be selected by setting the appropriate tablet control bit in the accumulator.

### ACCUMULATOR BITS

0	4	5	6	7	8	9	10	11
	SET DATA	SET PEN	CLR SET	TAB 01	TAB 02	TAB 03	TAB 04	
	READY	DATA	XY					

### Select Tablet Word Format

#### **Clear Set XY (WTMN)**

Octal code: 6064

##### Execution

Time: 2.6  $\mu$ s

Operation: The Clear Set XY IOT is used only for maintenance purposes. When the CLR SET XY bit position in the accumulator is cleared and the Clear Set XY IOT is issued, the X and Y registers will be cleared. When the CLR SET XY bit position is set and the IOT is issued, the X and Y registers will be set.

## FLOATING POINT PROCESSOR TYPE FPP-12

DEC's new floating point processor gives the PDP-8/E computer a dual processor capability. It also does calculations as much as 39 times faster than before, while maintaining seven-digit accuracy.

The unit (FPP-12) is designed for all types of floating point arithmetic. The computational speed of the PDP-8/E is dramatically increased because the floating point calculations are done by hardware rather than by software, which is usually the case. Typically, a three-word, 36-point floating point multiply took 1,100 microseconds when done by software, and 500 microseconds when done by software and an Extended Arithmetic Element. An FPP-12 equipped PDP-8/E can do the same calculation in 28 microseconds.

Adding the FPP-12 as a parallel processor decreases the time needed to run a specific program.

When a calculation has to be done, it is transferred from the central processor to the floating point processor, while the central processor continues with its program. Without the FPP-12, the calculation has to be done by the central processor unit, which interrupts the program until the calculation is done. Also, the FPP-12 simplifies programming by giving a programmer direct access to 32,768 words or core memory and by eliminating the paging steps usually required. Eliminating paging can also lead to further reductions in the time required to execute a program. These time-saving features, when combined with the time saved by using hardware to do floating point calculations, allow an FPP-12 equipped PDP-8/E to execute application programs as much as 100 times faster than they could be done by software alone.

### Floating Point Number System

The term, floating point, implies a movable binary point in a similar manner to the movable decimal point in scientific notation. An exponent is used to keep track of the number of spaces the binary or decimal point is moved.

Examples of scientific notation:

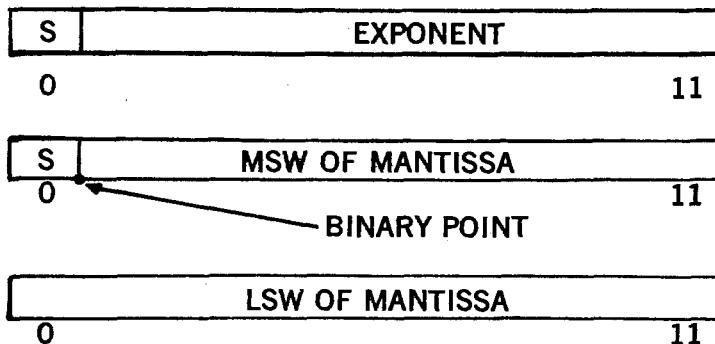
$$234 = 23.4 \times 10^1 = 2.34 \times 10^2$$

Examples of binary floating-point notation:

$$\begin{aligned} (1011) &= (101.1) \times 2^1 = (10.11) \times 2^2 = (1.011) \times 2^3 \\ &= 0.1011 \times 2^4 = 0.01011 \times 2^5 \end{aligned}$$

Note that in all cases of binary floating-point notation given above, there are four significant bits. However, in the last example the mantissa which multiplies the exponent contains six bits. Given a fixed number of bits, it is desirable to adjust the exponent and the binary point to eliminate leading zeroes to retain the maximum significance for a given format length. The FPP12 normalizes or removes leading zeroes as the last step in every floating-point arithmetic operation.

The floating-point data format used by the FPP12 is identical to the format used by the PDP-8 floating-point system (DEC-08-YQYB-D). As shown below, there is a 12-bit signed 2's complement exponent and a 24-bit signed 2's complement mantissa.

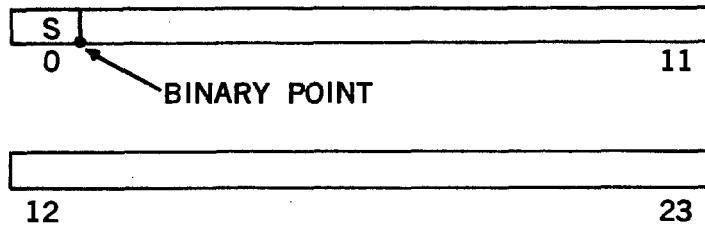


The FPP-12 carries all calculations to 28 bits of precision then rounds to 24 bits after normalization. After rounding, the results are rechecked for proper normalization prior to completion of the instruction.

In fixed point arithmetic, a calculation which results in a number whose magnitude cannot be expressed in 12 or 24 bits is an error. With the FPP-12, the number range is  $2^{+2047}$  to  $2^{-2048}$ . Exceeding the upper limit,  $2^{+2047}$ , causes the FPP-12 to interrupt the CPU and set its exponent overflow status bits. A calculation resulting in an exponent smaller than  $2^{-2048}$  is an exponent underflow which normally causes a program interrupt. The programmer has the option at initialization to request that the underflow trap be ignored, in which case, the result of a calculation in which underflow occurred is set to 0.

#### Double Precision

For those calculations where full 24-bit precision is not necessary and where core space is of a premium, the FPP-12 is used in fixed point double precision mode. Each operand consists of a 24-bit signed 2's complement fraction as shown below. As with the floating-point mode, each calculation is carried to 28 bits of precision and rounded to 24 bits. In this instance, normalization is not performed allowing the occurrence of leading zeroes which reduces the precision of subsequent calculations. The largest numbers that may be represented in double precision format are  $+2^{23}-1$  and  $-2^{23}$ . Calculations producing numbers that exceed this range cause the floating point processor to initiate a program interrupt with the fraction overflow status bit set to a one.



#### Operation

The FPP-12 is initialized and interrogated as to its status through PDP-8/E IOT's. Once initialized, the FPP-12 operates much like a central processor fetching instructions and operands and storing results in memory. Data breaks are generally requested as needed. However, the usual number of breaks requested by the FPP-12 is two per instruction performed by the processor. This means that while the FPP-12 is "stealing cycles," programs can be run simultaneously at slightly reduced speed.

### Active Parameter Table Format

#### **Location**

P	Field Bits of Operand Address	Field Bits of Base Reg.	Field Bits of Index Register Location	Field Bits of FPC
P+1	Lower 12 bits of FPC			
P+2	Lower 12 bits of index register location			
P+3	Lower 12 bits of Base Reg			
P+4	Lower 12 bits of operand address			
P+5	Exponent of FAC			
P+6	MSW of FAC			
P+7	LSW of FAC			

**NOTE:** APT address points to location P.

It should be noted that once initialized the FPP-12 will execute programmed instructions until

1. an error condition occurs,
2. an exit instruction is reached,
3. an exit IOT is issued,
4. an I/O preset is issued by the PDP-CPU\*,
5. the PDP-CPU encounters any type of halt.

#### **Initialization**

In order to execute the first instruction of any program the FPP-12 must have the following information:

1. The address of the first instruction (FPC)
2. The initial contents of the floating AC (FAC)
3. The core address of index register 0. (Index registers 1 through 8 are stored in the next 7 sequential 12 bit words.) (X0)
4. The base register which contains the core address of the first location in the data block. (The data block consists of 128 thirty-six bit words.)

To simplify initialization, the four parameters listed above are placed in core in an active parameter table (APT), shown above, by the CPU. Two initializing IOT's are then issued to the FPP-12. FPCOM (6553) loads a command register and the most significant 3 bits of the APT pointer. FPST (6555) loads the remaining 12 bits of the APT pointer and starts the floating-point processor. Whenever the floating-point processor performs an exit, the current values of the FPC, FAC, X0, base reg., and operand address are deposited in the APT to be used either for restarting the FPP-12 or for debugging.

#### **IOT List**

**FPINT** 6551 Skip on FPP "interrupt request" flag.

**FPHLT** 6554 Halt the processor at the end of the current instruction. Store active registers in core, set a status register bit, and the "interrupt request" flag.

\*This operation while the FPP-12 is running might necessitate a program reload.

- FPCOM** 6553 If the FPP is not running and the FPP "interrupt request flag" has been reset, set the command register to the contents of the AC. The three least significant bits of the AC set the field bits of the "Active Parameter Table" address. If the FPP is running or the interrupt request flag is set, the instruction is ignored.
- FPICL** 6552 Clear the FPP "interrupt request" flag.
- FPST** 6555 If the FPP is not running and the FPP "interrupt request flag" is reset, set the location of the "Active Parameter Table" to the contents of the AC, initiate the FPP and skip the next instruction. If the FPP is not running or the FPP "interrupt request flag" has not been reset, the instruction is ignored.
- FPRST** 6556 Read the FPP status register into the AC.
- FPIST** 6557 Skip on FPP "interrupt request" flag. If the skip is granted, clear the flag and read the FPP status request into the AC.

#### CPU AC After Read Status Instruction

AC0	Double Precision Mode
AC1	Instruction Trap
AC2	C.P.U. Force Trap
AC3	Divide by Zero
AC4	Fraction Overflow (double precision mode only)
AC5	Exponent Overflow
AC6	Exponent Underflow
AC7	
AC8	
	Unused
AC9	
AC10	
AC11	Run

The following data are transferred to the FPP by issuing the FPCOM (load command register instruction 6553):

AC0	Select double precision mode
AC1	Exit of exponent underflow error
AC2	Enable memory protection
AC3	Enable interrupt
AC4	Do not store op address on exits
AC5	Do not store address of index registers on exits
AC6	Do not store address of indirect pointer list on exits
AC7	Do not store FAC of exists
AC8	Unused
AC9	
AC10	
AC11	Data field of "Active Parameter Table"

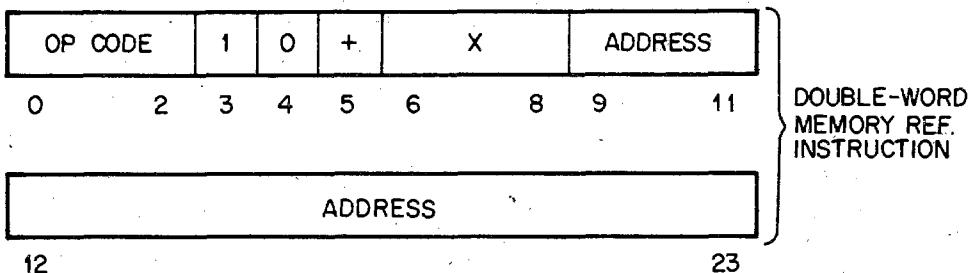
## **Instruction Set and Detailed Programming Spec**

### **Methods for Memory Reference Instructions**

The FPP-12 is capable of three modes of addressing for memory referencing instructions:

1. Double-word direct addressing
  2. Single-word direct addressing
  3. Single-word indirect addressing

A full indexing capability is available for both methods 1 and 3. The determined address for memory referencing instructions indicates the exponent in floating-point mode and generally directs to the most significant word in double precision mode. The format for double-word addressing is shown below:



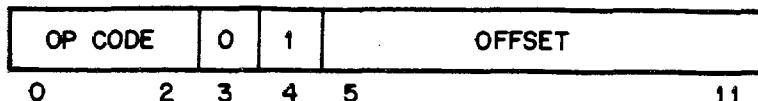
### Example 1

If bit 4 is a 0, a double-word instruction is indicated. Setting bit 3 of double-word instruction to a 1 indicates a memory referencing instruction. A non-zero quantity in bits 6-8 causes the address contained in bits 9-23 to be modified by a specified index register. Setting bit 5 to a one causes the specified index register to be incremented prior to use in modification of the address. It should be noted that index register zero can be incremented and tested but is not used for address modification.

## **Single-Word Addressing Formats**

The two single-word address formats utilize a data block that is specified by a base address which is contained in the base register. The data block contains 128 3-word locations. In double-precision mode, the exponent of locations is ignored on the data block.

Single word formats are distinguished by bit 4 being a one. Bit 3 is the indirect indicator in a similar manner to PDP-8 code. The single-word direct address format example shown below the core address is equal to the sum of the 7-bit offset times 3 plus the contents of the base register.



### Example 2

If bit 3 is a one, the following indirect format is specified:

OP CODE	0	1	OFFSET		
0	2	3	4	5	11

### Example 3

The effective address for Example 3 is given by the following equation:

$$\text{address} = C ((\text{offset} * 3) + \text{base address}) + \underline{[C(X+X_0) + \text{bit } 5 * 1]} [2 \text{ or } 3]$$

This term = 0 if  $X = 0$

### Index Registers

Any core location may be used as an index register. Index register 0 is determined by the 15-bit  $X_0$  address. The  $X_0$  address is initially set from the active-parameter table, but may be altered by the MVX instruction. Index register X is in core location  $X_0 + X$  where  $X = 0, \dots, 7$ .

Accessing successive data points in floating-point mode requires incrementing the operand address by (3)<sub>8</sub> for each new data point. In double-precision mode, the proper increment is (2)<sub>8</sub> for each new data point. To account for the difference between the two modes, the selected index register is multiplied by 3 in floating-point mode or 2 in double-precision mode before it is used as an address modifier.

### Instruction Set

OP CODE	MNEMONIC	MEMORY REFERENCE INSTRUCTIONS
0	FLDA	Load the FAC from the effective address.
1	FADD	Add the operand to the contents of FAC and store the result in the FAC.
5	FADDM	Add the operand to the contents of the FAC and store the results in the operand.
2	FSUB	Subtract the operand from the contents of the FAC and store the result in the FAC.
3	FDIV	Divide the operand into the contents of the FAC and store the results in the FAC.
4	FMUL	Multiply the contents of the FAC by the operand and store the result in the FAC.
7	FMULM	Multiply the contents of the FAC by the operand and store the results in memory.
6	FSTA	Replace the operand with the contents of the FAC.

## Program Examples

LOCATION	MNEMONIC	OCTAL CODE	
X	FSUB A	2401 5432	/subtract 1 from the /FAC
15432	A,	0001 2000 0000	
X	FSUB B	2205	/subtract 1 from the /FAC
Base Register + 5	B,	0001 2000 0000	
X	FSUB C, 2	2421 5432	/subtract 1 from the /FAC
15432 + 3 (Index Reg 2)	C',	0001 2000 0000	
X	FSUB I D	2603	/subtract 1 from the /AC
Base Register + 3	D,	5412 0132 6724	
26724		0001 2000 0000	

## Special Instructions

The FPP-12 special instructions are similar in nature to the nonmemory referencing instructions for the PDP-8. The set of special instructions includes conditional jumps, two types of subroutine calls, two types of unconditional jumps, several index register operations, a number of accumulator controls, two mode control instructions, and several operational instructions. Altogether, the special group has 26 defined instructions, five trapped instructions, and 14 undefined codes which do not perform any operation. Special instructions which may consist of 1 or 2 12-bit words are denoted by zeroes in bits 3 and 4 as shown below:

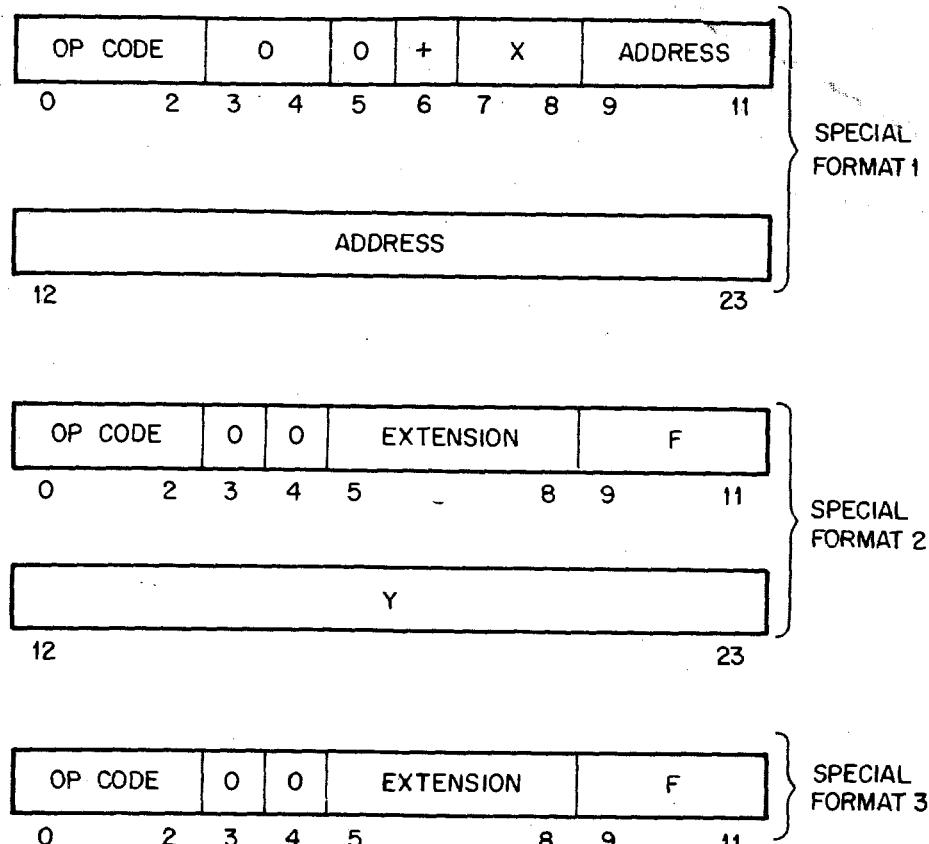
### SPECIAL FORMAT 1

OP CODE	MNEMONIC
---------	----------

2 JNX

The index register X is incremented if bit 5 = 1 and a jump is executed to the address contained in bits 9-23 if index register X is non-zero.

The JNX instruction is similar to the following sequence of PDP-8 instructions.



ISZ  
JMP TAG

- |                       |   |
|-----------------------|---|
| 3<br>4<br>5<br>6<br>7 | <b>Trapped Instructions</b><br>The "instruction trap" status bit is set and the FPP-12 exits causing a PDP interrupt.<br>The unindexed operand address is dumped into the active parameter table. |
|-----------------------|---|

**SPECIAL FORMAT 2**

SPECIAL FORMATTING			
OP CODE	EXTENSION	MNEMONIC	
0	10	FSTAX	The contents of the index register specified by bits 9-11 are replaced by the contents of bits 12-23.
0	11	ADDX	The contents of bits 12-23 are added to the index register specified by bits 9-11.
0	12-17	NOP	These codes are undefined single-word instructions and perform no operation.

**Conditional Jumps**—Jumps, if performed, are to the location specified by bits 9-23 of the instruction.

1	0	JEQ	Jump if the FAC = 0
1	1	JGE	Jump if the FAC >= 0
1	2	JLE	Jump if the FAC <= 0
1	3	JA	Jump always
1	4	JNE	Jump if the FAC $\neq$ 0
1	5	JLT	Jump if the FAC < 0
1	6	JGT	Jump if the FAC > 0
1	7	JAL	Jump if impossible to fix the floating-point number contained in the FAC; i.e., if the exponent is greater than (23) <sub>10</sub> .

#### POINTER MOVES

1	10	SETX	Set X0 the location of index register zero to the address contained in bits 9-23 of the instruction.
1	11	SETB	Set the base register to the address contained in bits 9-23.

#### SUBROUTINE CALLS

##### OP CODE EXTENSION MNEMONIC

1	13	JSR	Jump and save return. The jump is to the location specified in bits 9-23 and the return is saved on the 1st location of the data block.
---	----	-----	---

The JSR is used in writing re-entrant code as the return address is stored in the user's data block. A possible return from a re-entrant subroutine is via the two instruction sequences as follows:

	LDA 0	0200	/Load AL with contents /of 1st location of the data /block
	JAC	0007	/Jump to the location /specified by the /least significant 15 bits /of the AC mantissa /JAC is a special /Format 3 instruction
1	12	JSA	An unconditional jump is deposited in the address and address + 1 where address is specified by bits 9-23. The FPC is set to address + 2.
1	14-17	NOP	These codes are single-word NOP's.

**SPECIAL FORMAT 3  
INSTRUCTIONS**

1	1	ALN	The mantissa of the FAC is shifted until the FAC exponent equals the contents of the index register specified by bits 9-11. If bits 9-11 are zero, the FAC is aligned such that the exponent = $23_{10}$ . <sup>1</sup> In fixed-point mode an arithmetic shift is performed on the FAC fraction. The number of shifts is equal to the absolute value of the contents of the specified index register. If the contents of the index register is positive, shifting is towards the least significant bit; otherwise shifting is towards the most significant bit. In fixed-point mode the FAC exponent is not altered.
---	---	-----	---

<sup>1</sup>Setting the exponent =  $(23)_{10}$  intergerizes or fixes the floating point number. The JAL instruction tests to see if fixing is possible.

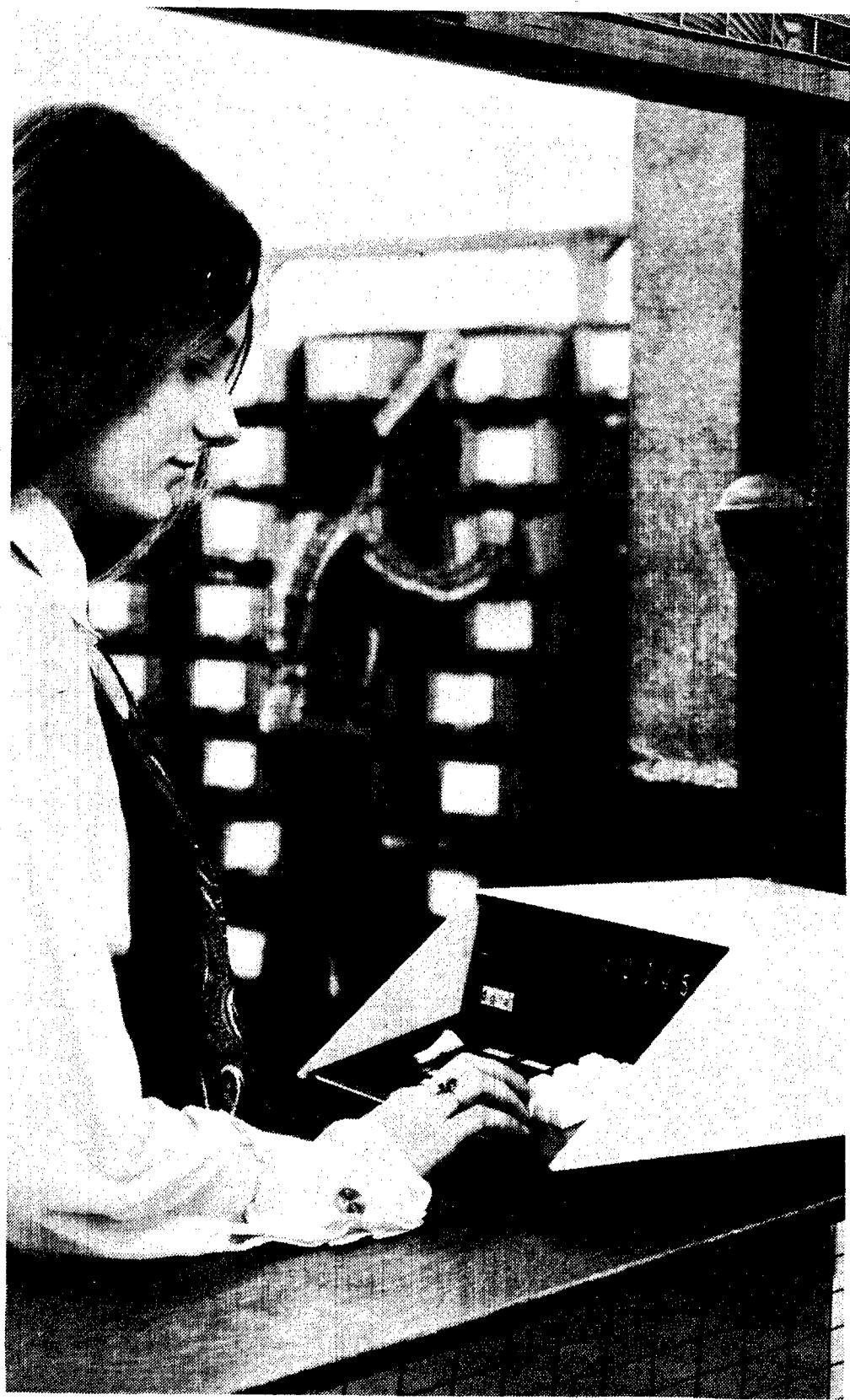
OP CODE	EXTENSION	MNEMONIC	
0	2	FLATX	The FAC is fixed and the least significant 12 bits of the mantissa are loaded into the index register specified by bits 9-11. In fixed-point mode the least significant 12 bits of the FAC is loaded into the specified index register. The FAC is not altered by the FLATX instruction.
0	3	FLDAX	The contents of the index register specified by bits 9-11 are loaded right justified into the FAC mantissa. The FAC exponent is loaded with $(23)_{10}$ and then the FAC is normalized. This operation is typically termed floating a 12-bit number. In fixed-point mode the FAC is not normalized.
0	4-7	NOP	These single-word instructions perform no operation.

**OPERATE GROUP**

<b>OP</b>		<b>BITS</b>		
<b>CODE</b>	<b>EXTENSION</b>	<b>9-1</b>	<b>MNEMONIC</b>	
0	0	0	FEXIT	Dump active registers into the active parameter table, reset the FPP-12 run flip flop to the 0 state, and interrupt the PDP-8 processor.
0	0	1	FPAUSE	Wait for external synchronizing signal. This instruction is designed to cooperate with the AIP-12 option.
0	0	2	FCLA	Zero the FAC mantissa and exponent.
0	0	3	FNEG	Form the two's complement of the FAC mantissa.
0	0	4	FNORM	Normalize the FAC. In fixed-point mode FNORM is a NOP.

**OP**

<b>CODE</b>	<b>EXTENSION</b>	<b>BITS</b>		
		<b>9-1</b>	<b>MNEMONIC</b>	
0	0	5	START F	Start floating-point mode.
0	0	6	START D	Start double-precision mode.
0	0	7	JAC	Jump to the location specified by the least significant 15 bits of the FAC mantissa.



RT01 DEC-link® Data Entry Terminal

### **RTO1 DEC-link® Data Entry Terminal**

DEC-link is a low-cost, self-contained data entry device which is remotely locatable. It features teletype and EIA serial line compatibility.

DEC-link offers 16 unique characters which a monitoring computer may use for either numeric data or control functions. It can display up to 12 digits of decimal data (plus decimal point) as well as status indicators.

Data is entered via an integral 16 character keyboard; numeric data is displayed on "Nixie"® tubes.

The status indicators are used to indicate non-numeric information such as "repeat transmission," "computer ready," etc. Four programmable status indicators are standard on DEC-link.

Interface to a computer is easily accomplished via any fully duplex, 4-wire data communications teletype interface.

Modem interface signals, corresponding to EIA RS-232C specifications, are also provided.

### **APPLICATIONS**

DEC-link provides easy and economical access to numeric information in a computer. It lends itself to such applications as:

- Stockroom Inventory Control
- Data Logging
- Information Retrieval
- Production Line Monitoring
- Quality Control Monitoring
- Work Flow Monitoring
- Security Systems
- Machine Efficiency Reporting
- Management Information Systems

DEC-link fills the gap in price, performance, and usage between full-scale, video displays and electro-mechanical, hard-copy devices.

### **SPECIFICATIONS**

#### **General**

Line Voltage: 115 VAC, 230 VAC 47-62 Hz.

Power: 30W

Size: 15" W x 12" D x 6" H

Weight: 12 lbs.

Aux. Switches: on-off

#### **Display Options**

Lamps: 4 Status Indicators (programmed control)

Digits: 4, 8, or 12 Nixie tubes

Decimal Point: Programmable over 12 digits

#### **Control Functions**

Clear Display: Code (100)

Load Status Indicators: Code (129), to (137), "P" through "<"

**Data Input**

**Input Levels:**

20 MA TTY Isolated Current Loop

EIA RS232C

**Receive Rate:** 110 or 300 Baud

**Character Format:**

8 level asynchronous serial ASCII

1 or 2 stop bits

**Data Output**

**Output Levels:** Isolated Transistor switch capable of passing 20MA

**EIA RS232 Levels:**

Data Terminal Ready

Transmitted Data

Received Data

Protective Ground

Signal Ground

**Transmission Rate:** 110 or 300 Baud

**Character Format:** 8 level asynchronous serial ASCII

**Character Rate:**

10 Characters/Second (110 Baud)

30 Characters/Second (300 Baud)

**Output Connectors:**

4 lug Jones Strip (TTY)

Cinch DB 25P (EIA)

**Character Set**

**Number of Characters:** 16

**Code:** ASCII 8 Level

**Character Codes:**

ASCII 0 through 9

A through F

**RT02 Data Entry Terminal**

The RT02 Alphanumeric Display is a low-cost data entry terminal offering both local and remote operation and featuring Teletype and EIA serial line compatibility. It can receive, store and display 32 alphanumeric characters on a single-line, gas-discharge type readout panel. Character repertoire is a modified 64-character ASCII set.

Data is entered via a 16-pad keyboard which includes a shift key to enable entry of a full 30 characters that the monitoring computer may interpret as either numeric data or control functions. Interfacing to a computer is accomplished via a standard full duplex 4-wire data communications Teletype interface such as those available for the PDP-8/E. Modem interface signals corresponding to EIA RS-232-C specifications are also provided.

Terminal usage may be defined entirely by software; that is, the application determines the significance of each character. For example, a key representing "account number" in a banking application could represent "cost per part" in a manufacturing monitoring application. The conversational response and simple 16-pad keyboard contribute to ease of operation by eliminating the need for the operator to interpret complex code numbers and symbols.

The RT02 is compact and lightweight so that it can be moved easily from one location to another. An all solid-state design guarantees the user a terminal with long, trouble-free operation. The bright red, 0.2 inch high, single-line readout is easily visible and offers greater readability while minimizing the possibility of reading errors. The need for an operator to manipulate a cursor or other such line-determining device is eliminated. Maximum security is assured because only one line of information is displayed at any instant. The elimination of electro-mechanical parts, paper and ribbon replacements makes routine maintenance unnecessary.

## SPECIFICATIONS

### General

Line Voltage:	115 Vac, 230 Vac; 47-63 Hz
Power:	50 W maximum
Size:	13½" w x 6½" h x 16" d
Weight:	14 lbs.

### Alphanumeric Display

#### Number of Character

Positions:	32
Character Set:	64-character modified ASCII
Character Height:	0.2"
Character Aspect Ratio:	5 x 7
Color:	Red
Viewing Angle:	120°

### Display Control Functions

Clear Display

Blank Display

Unblank Display

### Keyboard

Number of Keys:	16
Character Set:	ASCII
Number of Characters:	30 (with shift key)
Keyboard Control:	N-Key Rollover
Key Construction:	Environmentally sealed

### Data Input/Output

Levels: TTY 20mA Isolated Current Loop  
EIA RS-232-C

Transmit/Receive Rates: 110/110 baud  
150/150 baud  
300/300 baud  
1200/1200 baud  
110/1200 baud  
150/1200 baud

Character Format: 8 Level Asynchronous Serial ASCII  
2 Stop Bits (110 baud)  
1 Stop Bit (150, 300, 1200 baud)  
Even Parity

Input/Output Connectors: 8-Pin MATE-N-LOK (TTY)  
CINCH DB 25P (EIA)

### DW08-A I/O conversion panel

Digital's DW08-A Conversion panel enables any PDP-8/E computer to economically communicate with I/O devices of opposite logic levels. The DW08 contains its own integral power supply and takes up only 5-1/4 inches in height in a standard 19 inch rack.

The DW08 Positive-to-Negative Bus Converter accepts the positive I/O bus of a PDP-8/E and KA8/E, KD8/E option. Outputs consist of a Negative Bus, as well as a continuation of the Positive Bus. Positive Bus signal levels are defined (see figure 7-32) as high (+3 volts) and low (0 volts); Negative Bus signals are defined as high (0 volts) and low (-3 volts). The name bus denotes a combination of input (received by the computer) and output (sent by the computer) signals.

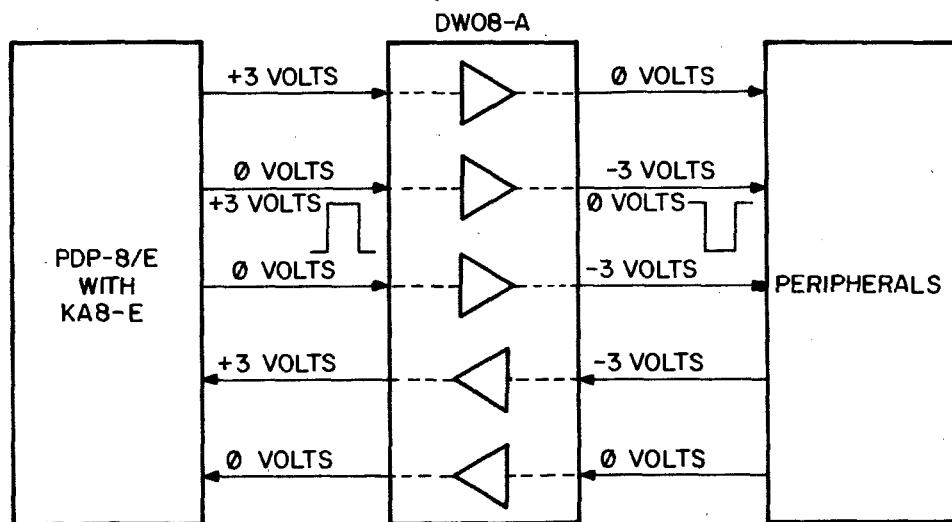


Figure 7-32 DW08-A conversion levels

The Positive Bus (with TTL logic levels of 0 volts and +3 volts) inputs on five M904 cable connectors at locations A01 to A05. The continued positive bus outputs are obtained on five M903 or M904 cable connections at locations B01 to B05. Input level conversion produces a high level out (0 volts) for a high level in (+3 volts); and a low level out (-3 volts) for a low level in (0 volts). Certain timing signals used on PDP-8/E computers (BIOP1, BIOP2, BIOP4, BTS3 (1), BTS 1 (1), and B INITIALIZE are clamped and inverted before level conversion.

The Negative Bus inputs and outputs are obtained on eleven WO11 or WO31 cable connections at location A13 to A23. Input signals to the computer on the Negative Bus are level converted to produce a low level out (0 volts) on the Positive Bus for a high level in (0 volts) and a high level out (+3 volts) for a low level in (-3 volts).

#### Cable Lengths

Delays occur within the DW08A unit due to level conversion, which effects the maximum length of the I/O bus of the PDP-8/E. The effective I/O

cable length consumed by the DW08A is 10 feet, which must be subtracted from the maximum permissible bus length from the PDP-8/E to the farthest device on the converted bus.

### Specifications

Dimensions:	5 1/4" high for 19" mounting. A cabinet depth of 15" is required because of the power supply.
AC Input:	120 VAC or 240 VAC $\pm 10\%$ , 50 Hz to 60 Hz
Power Consumption:	115 watts
Heat Dissipation:	300 BTU/hour
Weight:	20 lbs.





# 8

## installation planning and delivery

Digital Equipment Corporation has more than a decade of experience installing and maintaining over 20,000 computers in a wide variety of operating environments all around the world. Highly qualified Digital engineers are available to either perform or assist in the performance of every phase of computer delivery and installation, including the initial site survey, site preparation, delivery and unpacking, equipment installation and final acceptance testing. Adequate planning and initial site preparation are particularly important because they can simplify the installation process and result in an efficient, more reliable data processing system. The suggestions contained in this chapter are intended to simplify and enumerate some of the factors that should be considered during planning and preparation for delivery of a computer system.

The *Computer Site Preparation Handbook*, published by Digital Equipment Corporation, provides an invaluable reference that covers almost every aspect of pre-delivery planning and system installation. This handbook contains suggestions that will help to optimize the performance of any data processing installation, and it is particularly pertinent to installations that include magnetic disk or magnetic tape I/O equipment. A companion document, the *PDP-8 Site Preparation Worksheet*, contains additional data and a convenient checklist for evaluating system requirements.

### **PRE-DELIVERY PLANNING AND SITE PREPARATION**

The size of the system to be installed is the prime consideration in determining the degree of planning and site preparation required. A small table-top system may have requirements similar to those of household appliances, needing only desk space and a connection to existing primary power. A medium-size system often requires the more controlled environment of a typical modern office. Larger systems, and especially systems containing magnetic disk and tape equipment, may require a well controlled environment that maintains acceptable temperature, humidity, cleanliness and the like. Regardless of the size of the proposed system, primary planning considerations include provision for adequate space, safety and fire precautions, adequate electrical power, and any environmental conditioning that may be required.

Space and layout requirements will differ at various installations, depending upon the intended application. The floor area required for a particular system may be determined by considering the size of the specific components to be installed, length-to-width ratio of the room, location of columns or obstructions, and provision for future expansion. Adequate space must be provided to allow unrestricted access to all equipment doors and panels for maintenance. Space must also be allocated for printer forms stands, storage cabinets, card and paper tape files, work tables, communications equipment, and related items. Personnel should have easy access to peripheral devices that require manual operation or adjustment, and control panels should not be located directly on main aisles or traffic centers. Best results are achieved by preparing scale layouts of proposed system configurations in the room under consideration.

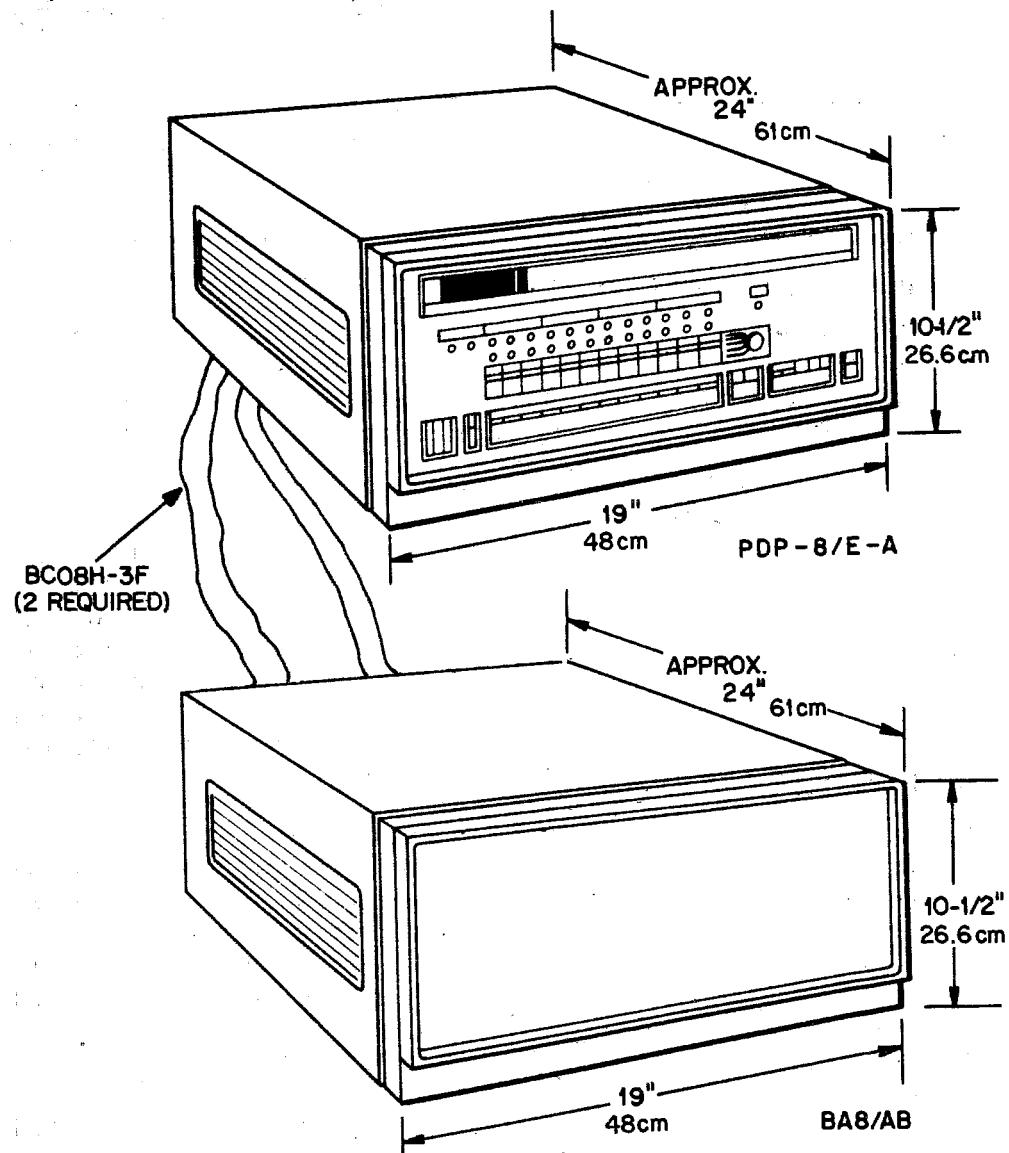


Figure 8-1. Table-top System Space Requirements

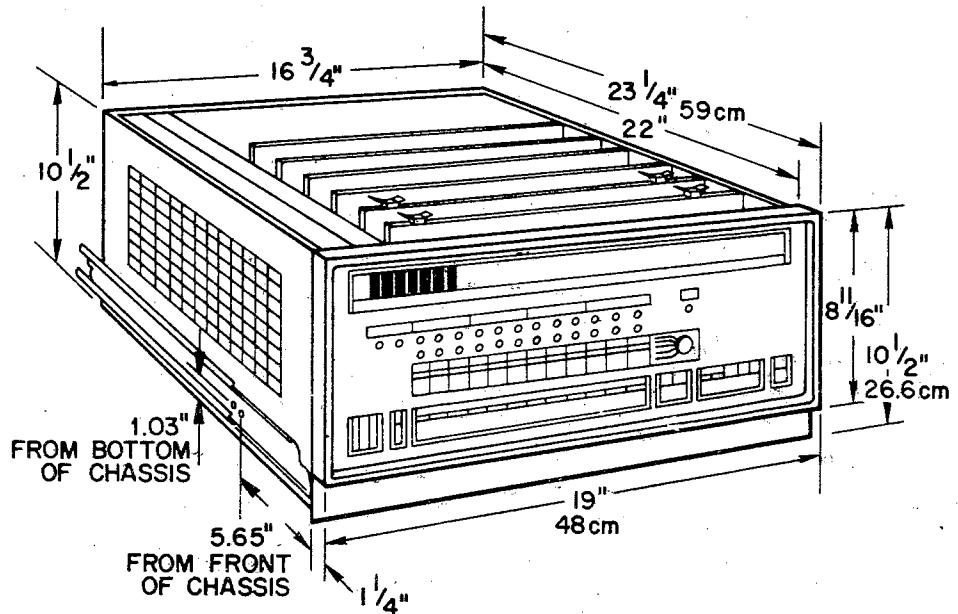


Figure 8-2. Rack-Mounted System Space Requirements

#### **Processor Options**

The PDP-8/E is available in either a table-top or a rack mounted configuration, each of which offers advantages that should be considered when planning the type of system desired. The table-top PDP-8/E shown in Figure 8-1 requires much less space than the rack-mounted version; it may be placed on one corner of a desk, and operated directly from the user's office. No special cooling is needed, and a standard 3-prong electrical outlet provides adequate power and grounding in most cases.

The table-top PDP-8/E is supplied with a single OMNIBUS assembly, chassis and table-top cover; power supply, and a 15-foot (4.6-meter) power cord. The 20 non-dedicated OMNIBUS slots provide ample room within the chassis for many optional devices which may be included with the system or ordered separately at a later date.

A table-top system may be further expanded by the addition of a BA8-A System Expander Box, also illustrated in Figure 8-1, which includes a KC8-EB Front Panel, one OMNIBUS assembly, a chassis and cover, power chassis assembly, and a 3.5 foot (1.0 meter) BC08H Cable Set. The basic system expander box accepts up to 20 QUAD-size modules. An additional OMNIBUS assembly may be installed inside the expander box to provide space for 18 more modules.

Rack-mounted PDP-8/E computers can be installed in a standard Digital Equipment Corporation cabinet or a customer-designed cabinet. The rack-mounted PDP-8/E, illustrated in Figure 8-2, comes equipped with one 20-slot OMNIBUS assembly, an H724 (or H724A) power supply, a 15-foot (4.6-meter) power cord and two chassis slides. System expandability is identical to that of the table-top model; installation of a rack-mounted BA8-B System Expander Box provides a total of up to 76 non-dedicated module slots.

Figure 8-3 gives dimensional information and illustrates the chassis layout of the PDP-8/F and PDP-8/M computers. The space requirements for the PDP-8/F and PDP-8/M are essentially identical to that of the PDP-8/E. Unlike the PDP-8/E, however, the PDP-8/F and PDP-8/M have room for only a single 20-slot OMNIBUS assembly inside the chassis. The PDP-8/M and PDP-8/F computers may be expanded to provide space for up to 58 modules by the addition of a system expander box. Most of the following information regarding PDP-8/E systems applies equally well to the PDP-8/F and PDP-8/M.

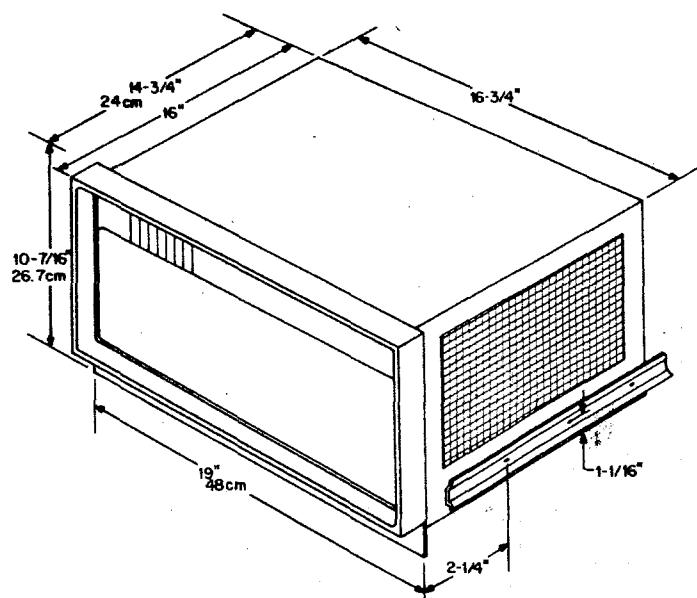
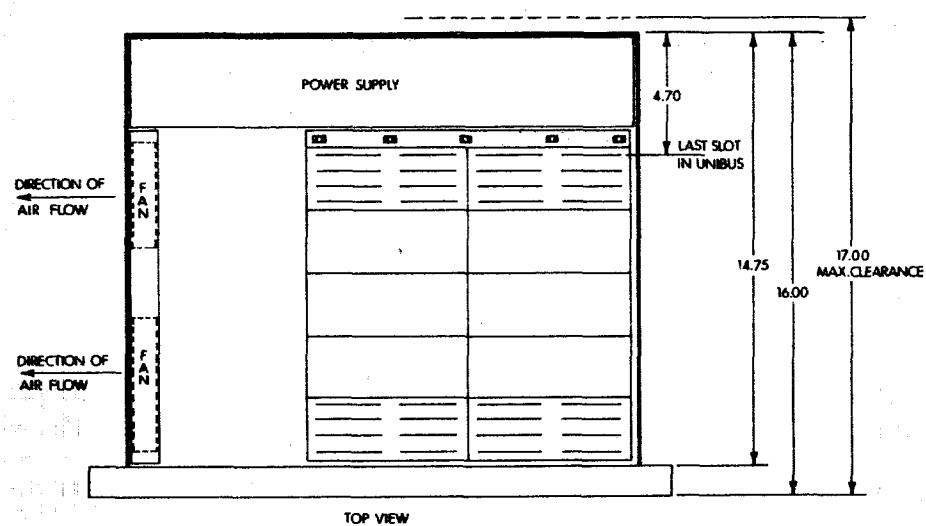


Figure 8-3. PDP-8/F and PDP-8/M Space Requirements

### Cabinet Options

Standard Digital Equipment Corporation cabinets allow the PDP-8/E to share floor space with an assortment of peripheral devices. This provides easy access to peripherals that require operator initialization or adjustment, such as high-speed paper tape units, along with minimal system space requirements. Use of standard cabinets can also simplify electrical wiring and eliminate most external cabling, contributing to a safer and more convenient data processing area.

The PDP-8/E may be installed in an H960-BC standard basic cabinet, illustrated in Figure 8-4, or an H967-BA short basic cabinet, illustrated in Figure 8-5. Either cabinet provides space for a selection of peripheral device options, which may be included with the basic system or ordered separately at a later date. Additional peripherals may be mounted in one

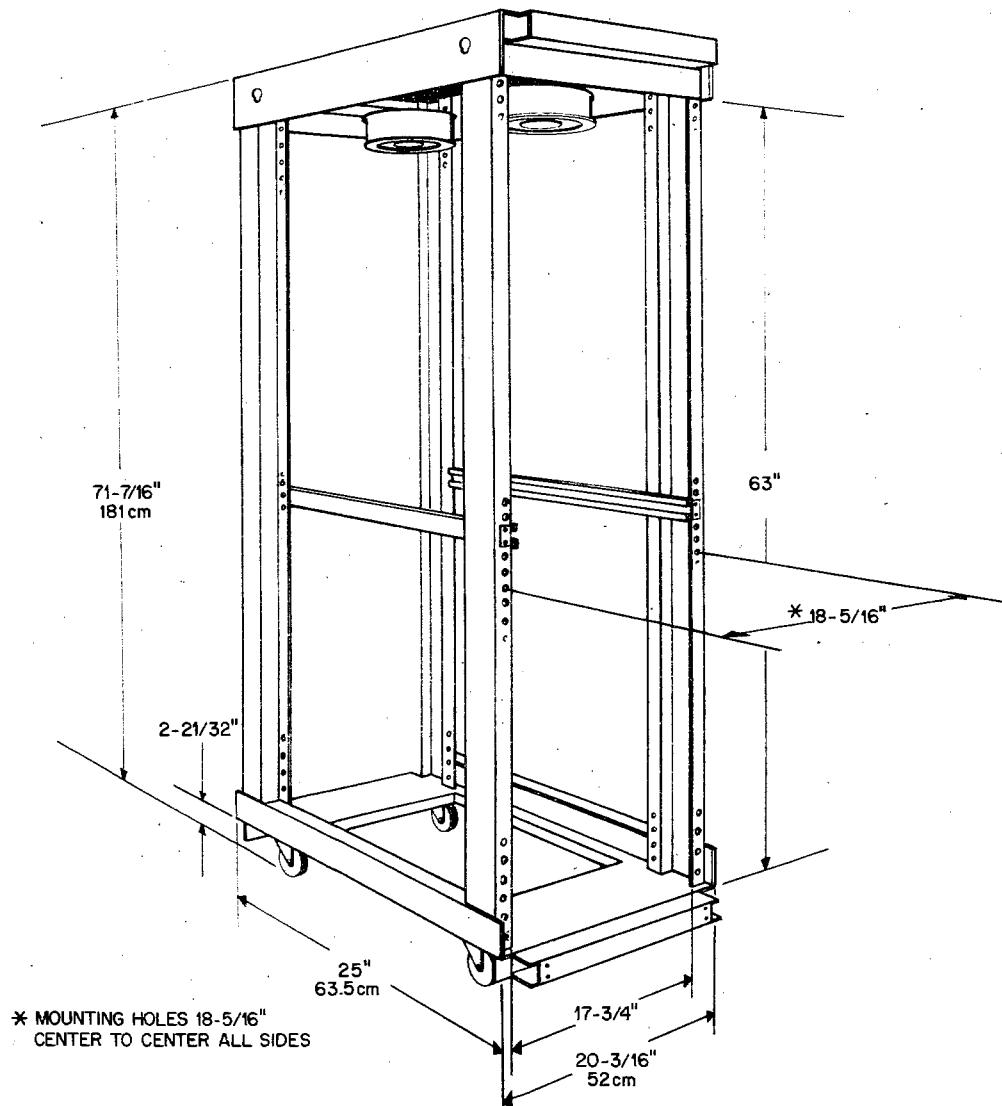
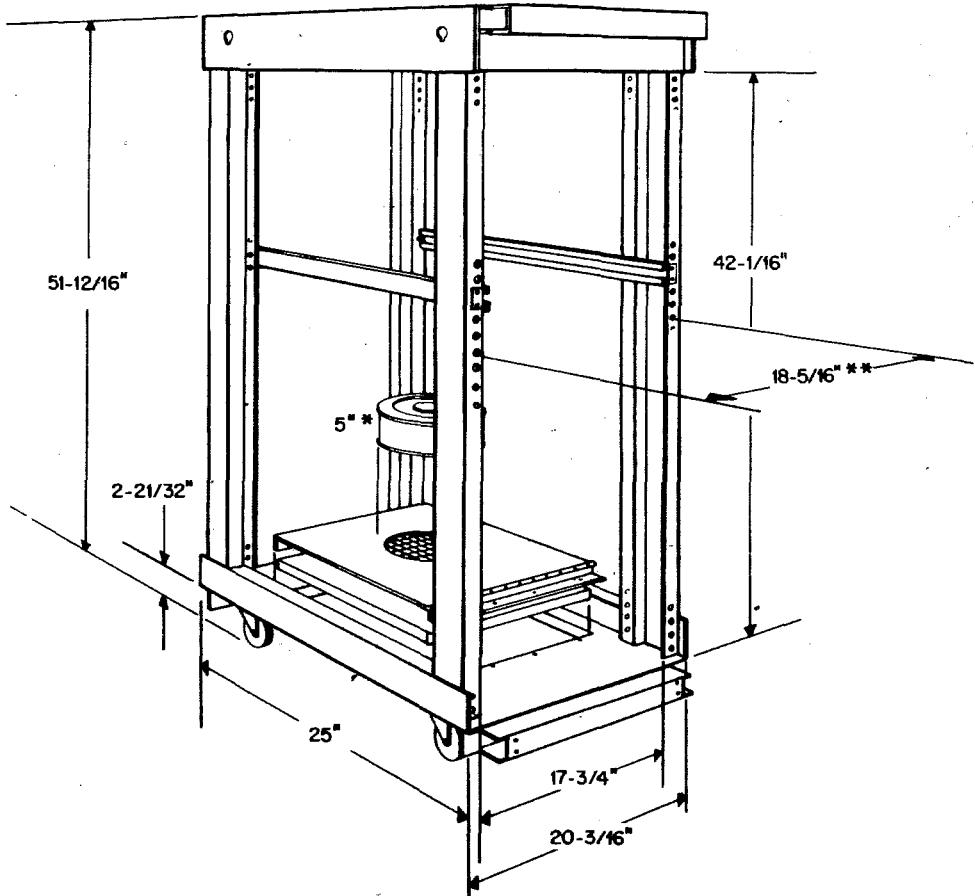


Figure 8-4. Standard Basic Cabinet



\* EFFECTIVE MOUNTING SPACE IS  
37-1/16" DUE TO 5" CLEARANCE  
NEEDED FOR FAN ASSEMBLY

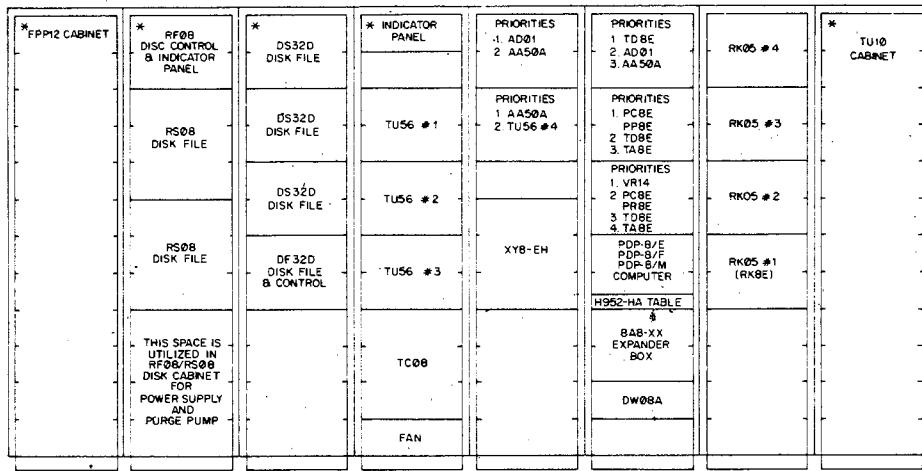
\*\* MOUNTING HOLES 18-5/16"  
CENTER TO CENTER ALL SIDES

Figure 8-5. Short Basic Cabinet

or more H961-A standard option cabinets, which are similar to the H960-BC standard basic cabinet, or H967-AA short option cabinets, which are similar to the H967-BA short basic cabinet. Both types of option cabinet are designed to be positioned adjacent to the H960-BC or H967-BA basic cabinet that houses the PDP-8/E computer, as illustrated in Figure 8-6. Although cabinet space is generally non-dedicated, adhering to the placement of options shown in Figure 8-6 affords optimum system performance for all configurations. Most peripheral devices that require a dedicated cabinet are supplied with the cabinet.

Figure 8-7 is a top view of a standard cabinet or short cabinet that shows the floor space required for either type of installation. It is very important to allow enough space around the cabinet so that unimpeded access to the equipment is maintained, even with the doors fully open. Cabinets housing the PDP-8/E and certain peripherals will not have front swinging doors; in these cases, the floor space indicated in Figure 8-7 is required so that the equipment may slide freely out of the cabinet for maintenance. Additional space should be provided in front of cabinets housing equipment that requires frequent adjustment by operating personnel.

### 8 Family — Standard Cab Configuration Guide



### 8 Family — Short Cab Configuration Guide

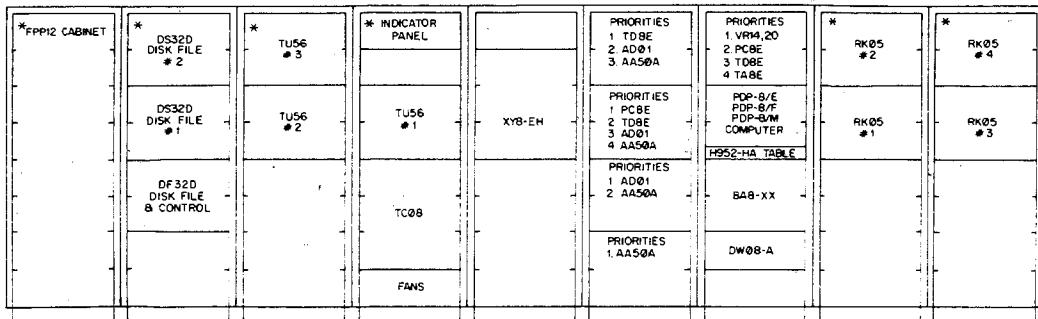


Figure 8-6. PDP-8/E Cabinet Configurations

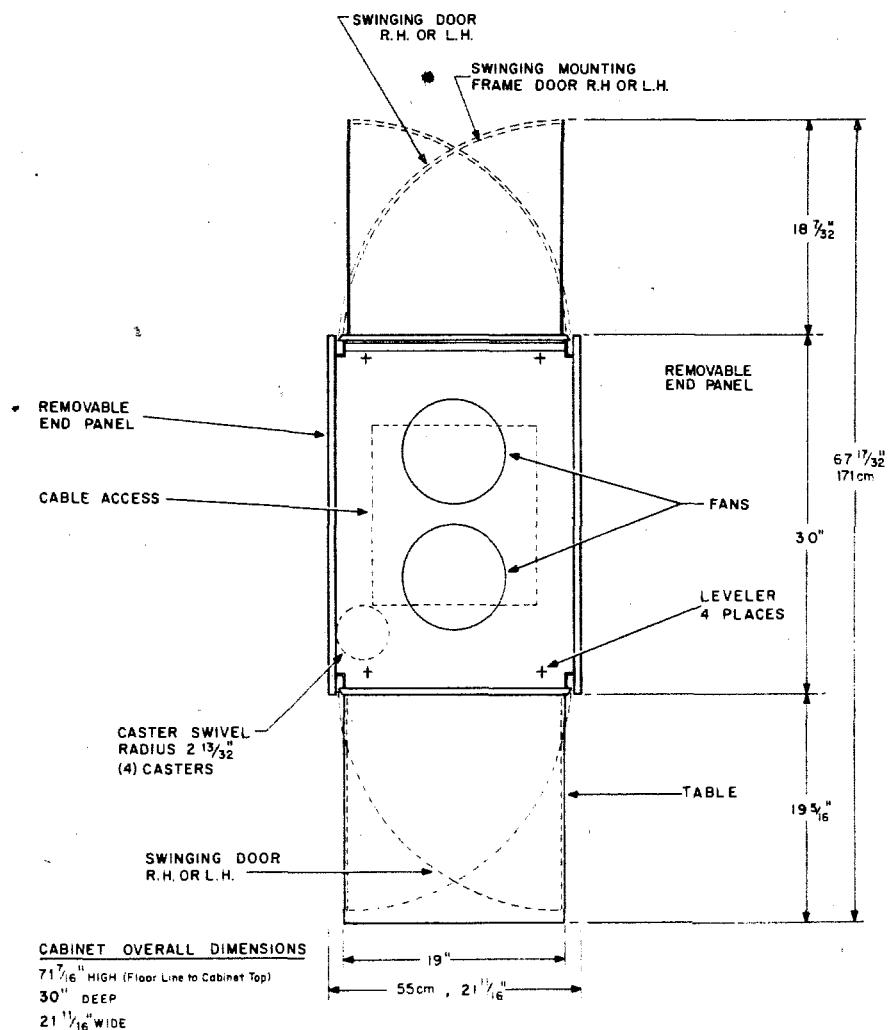


Figure 8-7. Cabinet Floor Space Requirements

### I/O Cabling Requirements

The cabling for table-top and rack-mounted computers differs slightly. On table-top models, cabling is routed from the lower rear side, through the strain relief on the processor, as shown in Figure 8-8. This figure also indicates how cables serving rack-mounted equipment may be routed into each cabinet through a panel located at the bottom of the cabinet. The casters provide ample clearance for cables that pass underneath the cabinet.

All Digital Equipment Corporation interconnecting cables are standard lengths and factory installed. If the cabinets must be shipped separately because of shipping or receiving restrictions, the cabinet interconnecting cables are reconnected at the installation site. Cable lengths should be as short as possible and under no circumstances should any of the maximum lengths be exceeded. External interconnecting cables may be protected from damage by installing a protective cover over the cables in a manner that will not prevent a safety hazard to operating personnel.

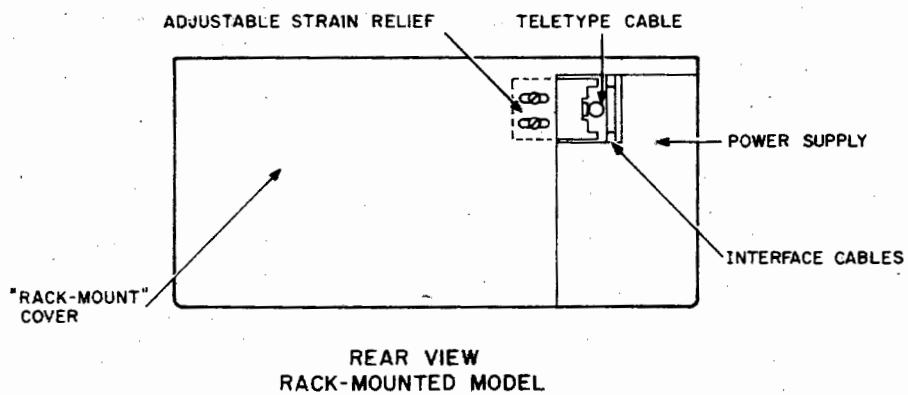
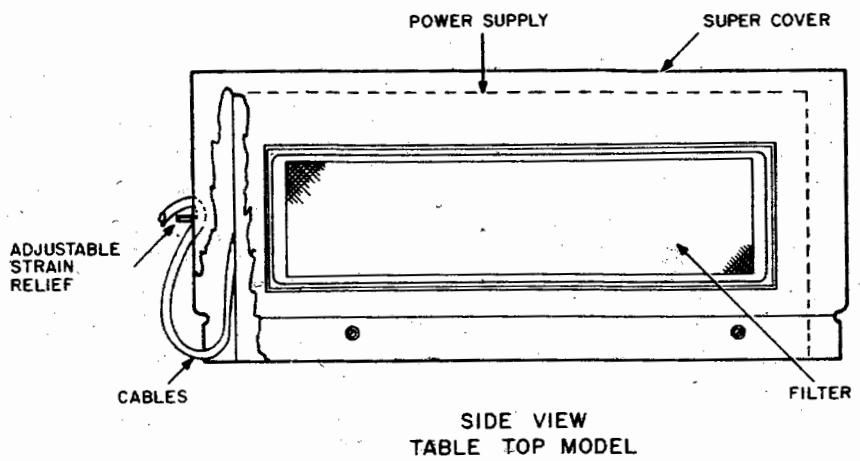


Figure 8-8. PDP-8/E Cable Layout

#### **TEMPERATURE AND HUMIDITY**

Recommended operating conditions for a typical computer system provide an ambient temperature of  $21^{\circ}\text{C} \pm 1^{\circ}\text{C}$  ( $70^{\circ}\text{F} \pm 2^{\circ}\text{F}$ ) with a non-condensating relative humidity of  $45\% \pm 5\%$ . As indicated in Table 8-1, however, PDP-8/E systems are designed to permit operation, storage and shipment under highly adverse conditions, when necessary. Figure 8-9 provides psychrometric charts which give recommended operating range as a function of temperature, relative humidity and absolute humidity. Certain peripheral devices have somewhat more stringent temperature and humidity requirements than the system as a whole.

## PDP-8/E EXTERNAL ENVIRONMENTAL REQUIREMENTS

	RECOMMENDED OPERATION	WORST CASE OPERATION	PACKED FOR SHIPMENT OR STORAGE
TEMPERATURE	°F      68 TO 72	32 TO 100	4 TO 150
	°C      20 TO 22	0 TO 38	-20 TO 65
RELATIVE HUMIDITY (NO CONDENSATION)	40% TO 50%	10% TO 90%	10% TO 90%

## SYSTEM INTERNAL OPERATING REQUIREMENTS

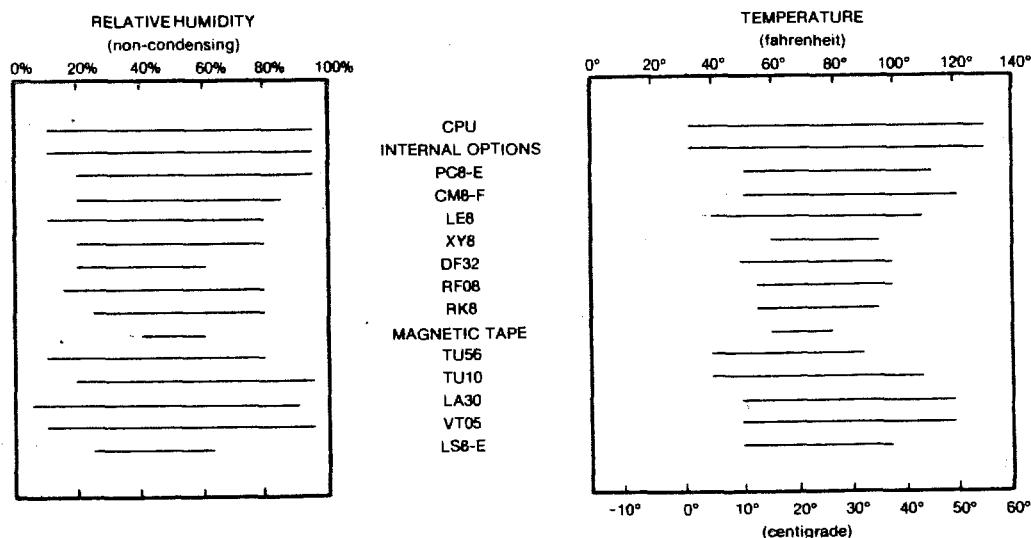


Figure 8-9. Recommended Operating Environment

PDP-8/E computer systems are air cooled, with cooling air circulated internally by fans in each cabinet. The air flow pattern varies slightly from one unit to the next; however, in general, air enters standard cabinets through the top filter and exits at the bottom. On short cabinets, air enters through a filter at the bottom of the unit and exits at the top rear. A minimum clearance of 30 inches (75 centimeters) above each cabinet is recommended to permit free circulation of cooling air. If this requirement cannot be met, some other means of allowing free air flow above and around the equipment should be devised.

#### **FACTORS AFFECTING SYSTEM RELIABILITY**

Although Digital Equipment Corporation computer systems are designed to be substantially more reliable in adverse environments than other systems of similar size, system environment has an unavoidable effect on system reliability. Reasonable control of the system environment will usually minimize maintenance requirements and provide more reliable system operation. Undesirable factors which may affect system reliability in some cases include the following:

**Extreme of temperature and humidity.** High temperature increases the rate of deterioration of virtually every material. In addition, thermal gradients induce temporary and permanent changes in most materials. High temperature and high humidity can combine to cause moisture absorption, resulting in dimensional and handling changes in paper and plastic media. Low humidity permits the build-up of static electricity, which can be annoying to personnel and may affect the system in extreme cases.

**Mechanical vibration.** Shock and vibration can cause slow degradation of mechanical parts and, when severe, may cause data errors on magnetic disks.

**Radiated emissions.** Sources of radiation, such as FM or RADAR transmitters operated in very close proximity to the computer, may affect the operation of the processor and certain peripheral devices.

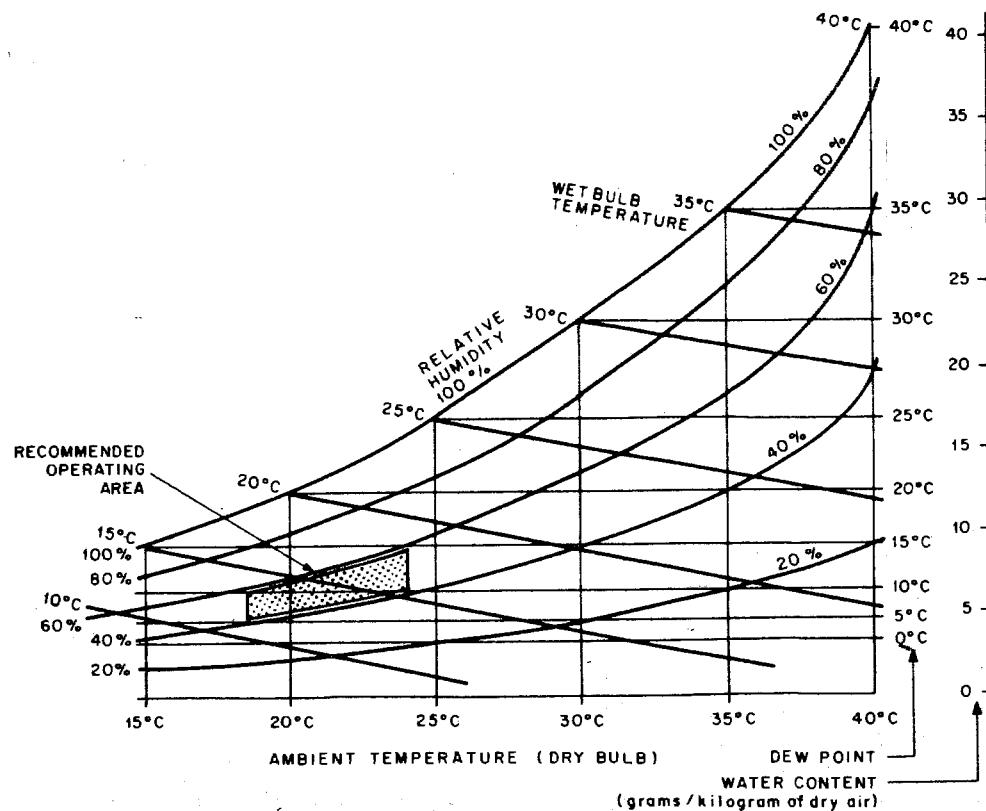
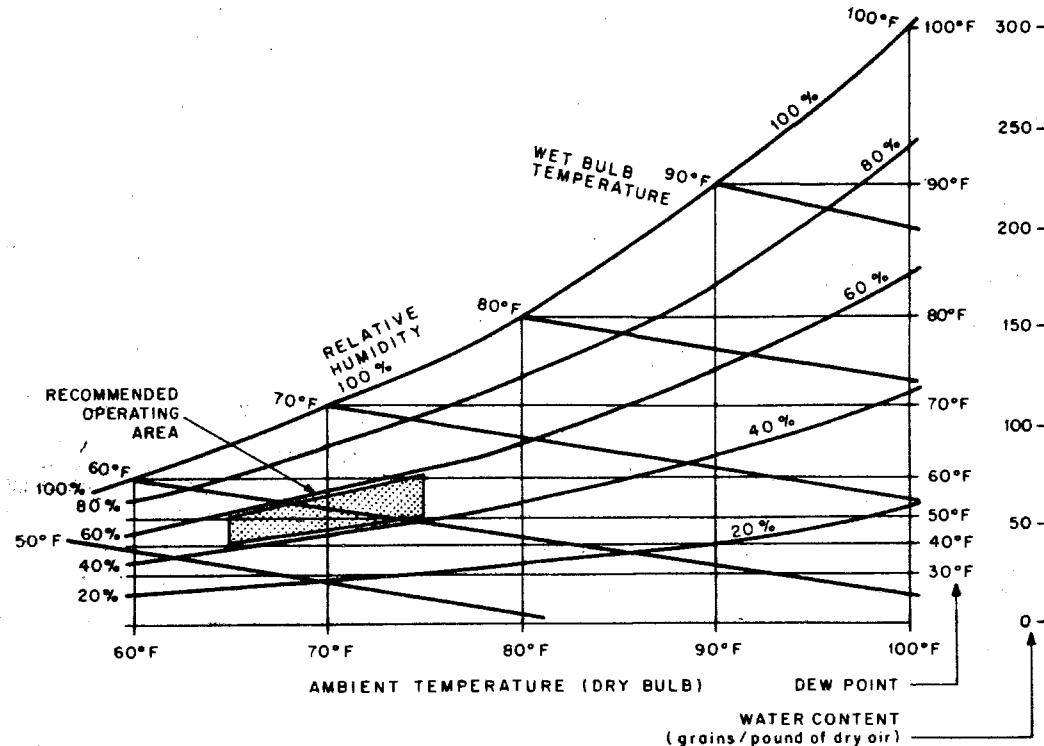
**High altitude.** Airborne systems, or systems operated at altitudes above 7000 feet (2000 meters) occasionally require additional blowers for adequate cooling. Disk subsystems have a maximum altitude specification of 12,000 feet (3500 meters).

The Computer Site Preparation Handbook contains suggestions for isolating and eliminating all of the conditions listed above, as well as other factors which may affect the operation of a computer system in extreme cases. Digital Equipment Corporation should be notified if any of these factors may be present at the proposed installation site.

#### **EXTENDED OPERATION UNDER EXTREME CONDITIONS**

For systems that must operate in a highly adverse environment, Digital Equipment Corporation provides the Industrial Computer Enclosure (ICE box), a standard, environmentally conditioned cabinet designed to house and protect rack-mounted equipment operating under hostile conditions. An ICE box provides safe, compact packaging for the PDP-8/E and such peripheral devices as a console terminal, high-speed paper tape reader and punch, DECTape, disk units, A/D or D/A converters, and the like.

**Table 8-1. System Environmental Specifications**



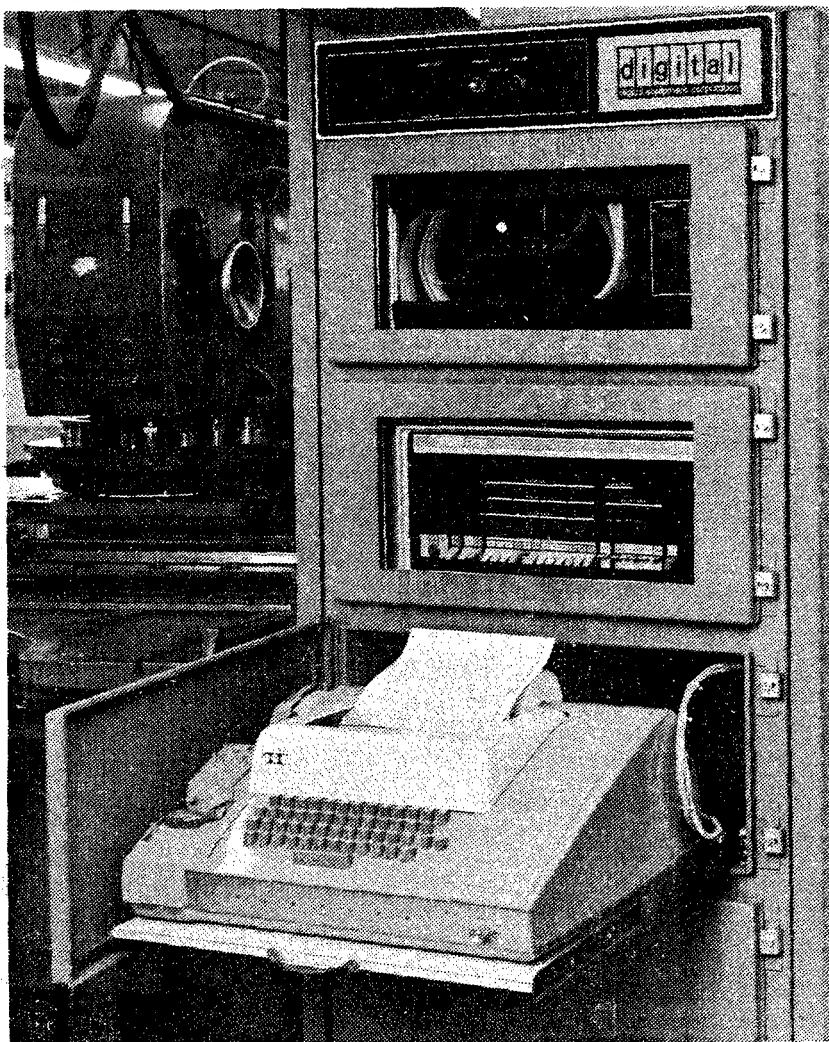


Figure 8-10. Industrial Computer Enclosure

The ICE box is constructed to NEMA-12 and JIC specifications, and is offered in 3 models. Each model comes equipped with either an air conditioner or an air-to-air heat exchanger, depending upon the ambient temperature of the particular industrial environment. Model H992, shown in Figures 8-10 and 8-11, is typical of the configurations available. The H992-AA ICE box includes an air conditioner for use in ambient temperatures up to 130°F (55°C). The H992-AB model contains an air-to-air heat exchanger for use in ambient temperatures up to 100°F (38°C).

#### ELECTRICAL CONSIDERATIONS

Computers and related equipment require a reliable power source with minimal voltage and frequency fluctuations. The exact power requirements for a given installation will depend upon the intended application. In general, line voltage disturbances with a magnitude greater than  $\pm 10\%$  of nominal voltage and a duration greater than 5 milliseconds are undesirable.

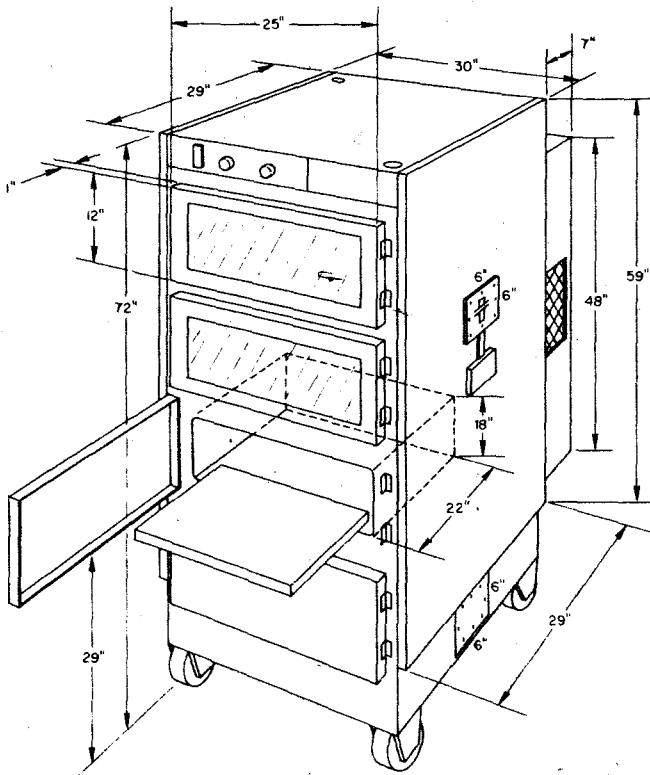


Figure 8-11. ICE Box Dimensional Data

Local disturbances in the power supply may be caused by overloading of transformers and feeders or switching of large loads such as elevators, air conditioners and lighting. Other local disturbances result from devices such as arc welding and X-ray equipment injecting RF components into the power line. In some areas, power source disturbances can result from voltage fluctuations or power factor corrections at the public utility. The effects of these conditions may be minimized if they are identified prior to equipment installation by monitoring the electrical service for a period of time consistent with expected system operating time throughout a typical week. Test equipment used to monitor the service should have sufficient response to detect objectionable disturbances of short duration.

#### **General Power Requirements**

The total current required for a computer system may be determined from the data in Table 8-2 by adding the requirements for every cabinet and every free-standing peripheral device. Even though an installation may not use all of the options in a particular cabinet, it is advisable to provide adequate power for the entire cabinet. Once the total current drawn by the installation is known, the type and quantity of AC connectors may be determined. Connectors must conform to statutory requirements of the locality in which the equipment is installed. In the U.S., power lines must terminate in NEMA receptacles in order to be compatible with the NEMA plugs supplied with the equipment.



PDP-8/F Processor



PDP-8/M and PDP-8/E Comparison

## Installation Data

Unit	Weight (lbs)	Dimensions (in.)			Heat Dissipation BTU/Hr.	115V AC Current Amps		Power Consumption (KW)	Mounting Panels	Remarks
		Height	Width	Depth		Nom.	Surge			
PDP-8/E Rack-mounted	90	10 1/2	19	23 1/4	1,700	5.6 (Max.)	—	0.50	2	
PDP-8/E Table-Top	100	10 1/2	19	24	1,700	5.6 (Max.)	—	0.50	—	
Standard Cabinet H960B (Empty)	120	71 1/4	21 1/8	25	—	—	—	—	12 (available)	
System Expander Box BA8/AA (rack-mounted)	90	10 1/2	19	23 1/4	1,700	5.6 (Max.)	—	0.50	2	
System Expander Box BA8/AB (Table Top)	100	10 1/2	19	24	1,700	5.6 (Max.)	—	0.50	—	
Teletype ASR-33	70	45	23	19	375	2.0	8.0	0.11	—	
Paper Tape Reader PR8/E	32	10 1/2	19	16	510	1.0	—	0.15	2	
Joy Stick Cursor Controller	3	5 3/4	5	8 5/8	—	—	—	—	—	Table space required
Card Readers CR8-F or CM8-F	70	13	20	15	1200	3.2	—	0.40	—	Table space required
Magtape Drive Industry Compatible TU10	143	26 1/4	19	26	3,750	11	—	1.10	Separate Cabinet	TU10C = 9 track TU10D = 7 track
DECtape Control TC08	30	10 1/2	19	—	1,000	3.0	—	0.30	4	Controls up to 4 TU56 Transports
Dual DECTape Transport TU56	80	10 3/8	19	15 1/4	1,740	115V=2.85A +10V=800MA $\pm$ 10% -15V=550MA $\pm$ 10%	—	0.51	2	Optional +5V or +10V supply may be used
DECCassette Tape System TA8-E	32	5	19	19	300	1.0	—	0.12	1	

Installation Data (cont)

Unit	Weight (lbs)	Dimensions (in.)			Heat Dissipation BTU/Hr.	115V AC Current Amps		Power Consumption (KW)	Mounting Panels	Remarks
		Height	Width	Depth		Nom.	Surge			
Oscilloscope Display: VR03A	17½	6	8 ½	17 ¾	170	0.5	—	0.05	2	Tektronix Model RM602
Oscilloscope Display: VR12	68	10 ½	19	17	680	1.75	—	0.20	2	
Disk File and Control DF32D/DS32D	75/60	10 ½	19	23 ½	1,700	3	3	0.50	2	
Disk File and Control RF08/RS08	depending on size of system	71 ¾	21 ¼	30	510/620 (Plus RS08 Motors)	1.3/2.6 Plus 4.0A per RS08 motor	—	0.15/0.30 (Plus RS08 Motors)	Separate Cabinet	Dimensions are for subsystem mounted in 2 standard cabinets
Disk Cartridge System RK8-E	110	10 ½	19	27	800 (Logic) 2400 (Per Drive)	2.0 (motors & blowers)	10.0	0.20		
A/D Converter AF01A	55	8 ½	19	19 ½	200	0.5	—	0.06	2	
Line Printer and Control LE8/FA (80 col., 64 ch.)	280	46	24	22 ¼	1,125	30	—	0.33	—	Paper: standard fanfold Edge punched 4" to 9 ½" wide Paper guide: 5" paper catcher 11 ½"
Line Printer & Control: LE8/JA (132 col., 64 ch.)	300	46	48	25	1,700	50	—	0.50	—	Paper: 14" to 14 ½" wide Paper guide: 5"
30" Incremental Plotter & Control XY8/EA	31	10	39 ½	14 ¾	600	1.5	—	0.175	—	Table space required for Plotter
12" Incremental Plotter & Control XY8/EB	18	9 ¾	18	14 ¾	600	1.5	—	0.175	—	Table space required for Plotter
IDVM AF04A	300	71 ¾	21 ¼	30	2,350	6.0	13.2	0.69	Separate Cabinet	

8-17

### **Power Failure**

In case of a power failure, the system will shut down automatically with no loss of data or damage to system hardware. Source power supply failures are of two types: power outages and line voltage irregularities. Power outages may include short duration dips in voltages as well as prolonged failures. If the frequency of power outages is unacceptably high, they may be prevented by the installation of static inverters, motor generator sets, or a combination of both types of line buffering equipment.

Voltage irregularities may result from transient electrical noise or inductive spikes superimposed on the line voltage. This problem can be caused by a wide variety of industrial, medical and communications equipment operated in the vicinity of the power distribution system. The effects of line disturbances that persist even when transient-producing devices are disconnected from the riser and power distribution panel may be minimized by installation of an isolation transformer or RF filter. Installing power service and distribution in accordance with Digital Equipment Corporation specifications, as outlined in the Computer Site Preparation Handbook, will reduce the extent to which such measures are necessary.

### **Ground Requirements**

A system involving a digital/analog interface usually requires that the digital system ground be tied to the analog system ground at a single point, usually at the analog/digital interface. A low-resistance ground connection is required in these cases. In systems where no analog interface is involved, the grounding provided by a large electrical conduit is usually adequate.

The 3-wire plug supplied with table-top PDP-8/E provides the only ground connection required by this processor; however, rack-mounted systems and, especially, systems that include a variety of peripherals require additional grounding. The grounding schemes described in the Computer Site Preparation Handbook are recommended as effective means of keeping electrical noise and differential potentials under control in large systems. Whatever grounding system is used, it should provide less than 10 ohms impedance to moist earth from DC to 10 megahertz. It should also be insulated from sources of electrical noise, to prevent noise from entering the system via ground.

## **AC POWER FACILITY INSTALLATION AND TESTING**

The AC power requirements of the PDP-8/E computer are consistent with good electrical practice. The importance of correct electrical connections cannot be overstressed. Significant operational difficulties are likely in the event of either a poor neutral or a poor ground circuit. Voltage readings must be made at the power receptacle before the computer is plugged in. It is an extremely wise precaution to take a voltage reading from the frame of the computer to an established ground point before touching the cabinet.

Figure 8-12 illustrates the recommended wiring for the wall receptacle servicing a small computer system. It is advisable to provide a separate central load breaker panel for the computer system, with a circuit breaker for the computer and one for each cabinet or free-standing peripheral receptacle; however, this is not an operational requirement. Following installation of the power facilities, voltage readings should be taken at each receptacle in the system. A checkout procedure for testing the electrical system is provided by Digital Equipment Corporation on request. In any event, the tests described in the following section should be run as soon as all equipment has been uncrated and installed.

Digital Equipment Corporation engineers are available for assistance and consultation during installation and testing. Further technical assistance in the field is provided by home office engineers, branch office applications engineers and field service engineers.

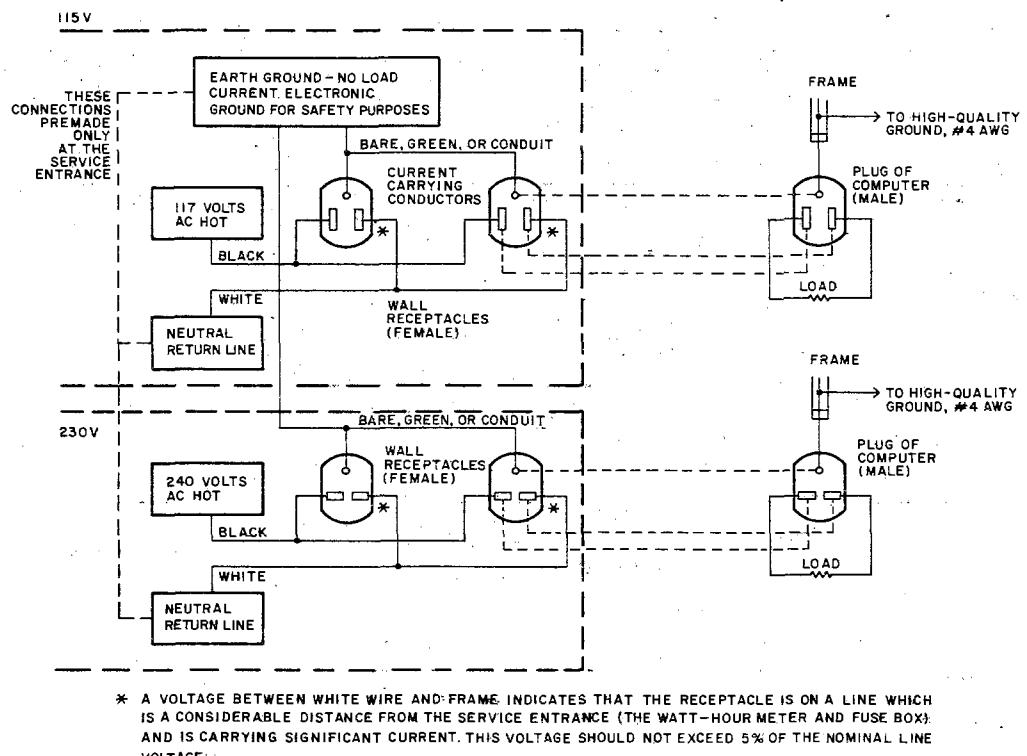


Figure 8-12. Recommended Receptacle Wiring

## GROUNDING AND FACILITY POWER TESTS

**CAUTION: THIS PROCEDURE INVOLVES MEASUREMENT OF DANGEROUS VOLTAGES**

If these tests indicate that faulty wiring exists, a qualified, licensed electrician should be consulted.

These tests should be performed in sequence; do not proceed beyond an abnormal indication until the fault has been corrected.

The computer should not be plugged in until the completion of tests 1 through 6; these tests are to be made at the computer receptacle with no peripheral equipment connected to AC power.

TEST STEP	AC VOLTMETER FROM	TO	MAXIMUM VOLTAGE IF MISWIRED	EXPECTED METER READING	INDICATES
1.	HOT	NEUTRAL	240 VAC	117 VAC	OK
2.	HOT	GROUND	240 VAC	117 VAC	OK
3.	NEUTRAL	GROUND	240 VAC	0 VAC	OK, UNLOADED CIRCUIT
	NEUTRAL	GROUND	240 VAC	UP TO 10 VAC	SEE* NOTE
4.	GROUND	CONDUIT OR BOX	240 VAC	0 VAC	OK
5.	NEUTRAL	CONDUIT OR BOX	240 VAC	UP TO 10 VAC	SEE *NOTE
6.	If 0 VAC was found in test 3, tests 4 and 5 should be made with the computer, or some other significant load, on the line. Test 4 should again be 0 VAC and test 5 should show some voltage up to 10 VAC. A reversal of these readings indicates a reversal of the ground and neutral lines.				
7.	Test steps 1 through 6 should next be made at each appliance receptacle which is to be used to supply power to a peripheral device. Test steps 9 through 11 must be made before any peripheral device is plugged in to AC power.				
8.	The previous tests have assured a properly wired AC supply for the computer. The following tests should now be made with the computer in operation. They are to be made between each appliance receptacle and the plug prongs of the peripheral device which is about to be plugged in. An open circuit in the AC wiring of the device will negate the validity of these tests; power switches must be on and all cables to the computer must be plugged into their correct slots.				

TEST STEP	FROM RECEPTACLE	TO PLUG	MAXIMUM VOLTAGE IF MISWIRED	EXPECTED METER READING	INDICATES
9.	HOT	HOT	117 VAC	117VAC ***	OK
10.	NEUTRAL	NEUTRAL	UP TO 10 VAC**	0 VAC	OK
11.	GROUND	GROUND	UP TO 10 VAC**	0 VAC	OK

\*\* In addition to a reversal of neutral and ground in the peripheral device, a reading here indicates also that the computer and the receptacle being tested are not on the same circuit.

\*\*\* There is an AC path through the line filter to ground.

#### INSTALLATION PROCEDURE FOR PDP-8/M AND TELETYPE

The remainder of this chapter provides detailed instructions for receiving, unpacking and installing the PDP-8/M computer with ASR-33 Teletype terminal. This is typical of procedures employed with all small computer systems, and illustrative of the time and labor required. Note that systems covered by a warranty or service contract must be installed by or in the presence of a DEC representative. If this is not done, the warranty will be voided.

Read each step carefully and completely, then perform it before continuing to the next step. The following tools will be required:

Knife   
Common Screwdriver   
Phillips Screwdriver   
Wire Cutters

1. Teletype: All instructions are referenced from the front of the unit.

1.1 Open the Teletype carton and remove:

- A. Packing material.
- B. Stand.
- C. Box containing the copy holder, power supply, and chad box.
- D. Typing unit.

**Do not tip the typing unit during or after the following step, as damage may occur.**

1.2 Remove the seven mounting screws holding the typing unit to the shipping pallet.

1.3

- A. Remove the tape from across the top of the cover.
- B. Unwrap the cables.
- C. Inspect the typing unit for external damage and loose screws.

1.4 Lift the clear cover to expose the print head.

- A. Remove the twist tie on the left side which holds the print head.
- B. Remove the cloth bag containing the platen (roller) knob and the ON/OFF knob.

1.5

- A. Remove the back cover from the stand.
- B. Remove the plastic parts bag inside.

1.6 Remove the power supply from the box described in step 1.1.C.

- A. Mount the power supply under the lip on the stand's front panel, about 1" from the right side panel. There are two clips attached to the power supply which slip over the lip.
- B. Attached the green wire from the power supply to one of the three tabs on the right side panel.

**The following step places the typing unit in an over-balanced condition. Use care to ensure that it does not fall to the floor. The help of an assistant is recommended.**

**1.7**

A. Place the typing unit on the stand with the back and right sides of each flush with the other.

B. Secure the typing unit to the stand using the four large screws and washers found in the plastic parts bag.

**1.8** There should be 4 cables coming out of the back of the typing unit. Connect them as follows:

A. Connect the short green wire to one of the three tabs on the right side-panel.

B. Connect the short 6-conductor cable with a plastic female connector to the matching male plug on the power supply.

C. The signal wire is the long wire with a flat plastic connector. Pass it and the 3 prong power cable through the square opening in the lower right side of the rear panel of the stand.

D. Secure the signal wire to the lower opening with the plastic clamp and screw provided.

**1.9** Install the copy holder by tilting it to the rear and inserting it in the 2 holes below the paper mounting recess, then pivoting it forward and pushing it down into the slots in front of the recess.

**1.10** Install the chad box under the punch pan by inserting the back of the flange surface between the stand and the sub-base, then pushing towards the rear.

**1.11** Install the roller knob, aligning the flat surfaces of the knob and the roller.

**1.12** Install the ON/OFF knob.

Included with the Digital Equipment Corporation manuals are three TTY manuals concerning the Model ASR-33. Volume 1 Chapter 2 contains instructions for installing and preparing it for operation. Sections pertaining to answer-back and wire changes should be disregarded.

**1.13** Replace the back cover on the stand using care to ensure that no wires are pinched.

**1.14** Remove the metal plugs from the ends of a roll of Teletype paper and insert the paper-holder. Load the paper, using the roller knob. Paper may be aligned by releasing roller pressure (lever at right hand side of roller).

**1.15** To load the paper tape punch unit:

A. Turn the Teletype Off. Plug the Teletype into a properly wired AC outlet.

B. Turn the Teletype switch to LOCAL. Depress the ON button on the paper tape unit.

- C. Thread tape into the back of the punch until it touches the pinch rollers. Lift the punch lid to view.
  - D. Depress the BREAK key on the keyboard. The pinch rollers should roll the tape through.
  - E. Repeat step D until the tape loads. If the pinch rollers do not rotate or if the Teletype chatters constantly, try again from step 1.15 and then call a DEC representative.
2. Computer:
- 2.1 Inspect the computer carton for damage and save it for future use.
  - 2.2 Remove the PDP-8/M from the box and place it on a sturdy table. Remove all wrappings. Turn the box upside down if difficulty is encountered.
  - 2.3
    - A. Remove the 5 screws which hold the top cover on the computer (2 on each side, 1 at the back). Lift the cover off.
    - B. Ensure that all modules and edge connectors are firmly seated.
    - C. Ensure that the flat black connector is firmly plugged into the matching connector on the M8650 module.
    - D. Run this cable out through the vertical slot at the back of the computer.
    - E. Replace the cover on the computer, using care not to pinch the cable.
  - 2.4 Be sure the Teletype switch is in the OFF position.
  - 2.5 Be sure the PDP-8/M OFF/POWER/PANEL LOCK switch is in the OFF position—key turned counter clockwise.
  - 2.6 Connect the Teletype cable from the PDP-8M to the cable on the Teletype. This cable is keyed for proper mating.

**Good grounding is essential for safety and proper operation of the system. Both the Teletype and the PDP-8/M should receive power from the same AC source. If a 3 prong outlet is not available use an adapter, and be sure to connect the ground wires. Check the AC facility for proper power and grounding after plugging in the equipment but before touching the chassis.**
  - 2.7 Plug the PDP-8/M and Teletype into the AC outlet.
  - 2.8 Turn the PDP-8/M key-operated switch clockwise until it is straight up and down (POWER position).
  - 2.9 Turn the Teletype switch to LINE. The Teletype should not chatter rapidly. If it does, call your DEC representative.

- 2.10 Set all the PDP-8/M switches 0-11 down except for switch 4 (octal 200).
- 2.11 Press ADDR LOAD, then release.
- 2.12 Press EXTD ADDR LOAD, then release.
- 2.13 Press CLEAR, then release.
- 2.14 Ensure that the HALT and SING STEP switches are up.
- 2.15 Press CONT, then release.
- 2.16 The machine should now be running a diagnostic test program. If any of the following indication are missing abnormal, proceed to step 2.17.
- A. The Teletype bell should ring every five seconds.
  - B. The RUN light should be on.
  - C. All three EMA lights should be off.
  - D. Turn the rotary switch to the BUS (counterclockwise) position (steps D to I refer to the bottom row of 12 lights). These lights should be very dim.
  - E. Turn the rotary switch clockwise to the MQ position. During this test the MQ lights may be either on or off, but not changing.
  - F. Turn the rotary switch clockwise to the MD position. All lights should be on, but some will flash when the bell rings.
  - G. Turn the rotary switch clockwise to the AC position. All lights should be on, but some will flash when the bell rings.
  - H. Turn the rotary switch clockwise to the STATUS position. The LINK indicator will be on. INT BUS will be on, but will flash when the bell rings. Other lights should seem to glow dimly.
  - I. Turn the rotary switch to the full clockwise STATE position F, D, E, IR0, IR1, IR2, MD, DIR will be on, SW will be on only if the SW switch is down.
  - J. The MEMORY ADDRESS lights will all be on.
  - K. If the machine is operating as described above, go to step 2.18.
- 2.17 If the machine does not seem to be running as described above:
- A. Push the HALT switch down. Set the SWITCH REGISTER switches 0-11 down. Press EXTD ADDR LOAD and release.
  - B. Perform the following set of switch manipulation. In each step, there are 12 figures which correspond to the 12 switches which are labeled the SWITCH REG-

ISTER (SR) on the front of the computer. The black circle with up-arrow symbol indicates that the corresponding switch should be set to its "up" position. The white circle with down-arrow symbol means that the corresponding switch should be set in its "down" position.

Set SR to:					then press LOAD ADDR 7756	
Set SR to:					then lift DEP	6032
Set SR to:					then lift DEP	6031
Set SR to:					then lift DEP	5357
Set SR to:					then lift DEP	6036
Set SR to:					then lift DEP	7106
Set SR to:					then lift DEP	7006
Set SR to:					then lift DEP	7510
Set SR to:					then lift DEP	5357
Set SR to:					then lift DEP	7006
Set SR to:					then lift DEP	6031
Set SR to:					then lift DEP	5367
Set SR to:					then lift DEP	6034
Set SR to:					then lift DEP	7420
Set SR to:					then lift DEP	3776
Set SR to:					then lift DEP	3376
Set SR to:					then lift DEP and again lift DEP	5356

C. Open the basic software kit. Place the tape labelled Binary Loader (DEC-08-LBAA-PM) in the terminal's paper tape reader. At the front of this tape is about 8" of leader tape which has just a single row of data holes punched. Position the tape in the reader at this point.



D. Push HALT and SINGLE STEP switches up.

E. Set the SR switches 0-11 all up except SW7 and SW11, then press ADDR LOAD. Then release, press the CLEAR switch, and release. Press the CONT switch, then release. Push the paper tape reader switch to the START position. The tape will read in.



- If it stops before the end of the tape, an error has occurred. Try again from step 2.17A, and, if another error occurs, call a DEC Representative.
- F. When the reader stops, push the HALT switch down. Remove the tape from the reader.
- G. Place the tape labelled INSTRUCTION TEST 2 (MAIN-DEC-8E-DOBB-PB) in the terminal's paper tape reader. At the front of this tape is about 8" of leader tape which has just a single row of large holes punched. Put the tape in the reader at this point.
- H. Push the HALT switch up.
- I. Set the SR switches 0-11 all up.
- J. Press ADDR LOAD, then release. Press CONT, then release.
- K. Push the terminal paper tape reader switch to START. The tape will read in. If it stops before the single row of large holes at the end of the tape, an error has occurred. Try again from step 2.17A, and, if another failure occurs, call a DEC representative.
- L. If the reader stops on the single row of large holes at the end of the tape, turn the PDP-8/M rotary switch to the AC position. All lights in the bottom row should be out. If they are not call a DEC representative.
- M. Remove the tape from the reader.
- N. Place the reader switch in the FREE position, then repeat steps 2.10 through 2.16. If the machine still does not operate properly, turn the Teletype and PDP-8M off, unplug them, and call a DEC representative.
- 2.18 Leave the diagnostic test program running while you continue. The Teletype may be turned off, if desired, to stop the bell from ringing.
- 2.19 Open the software boxes, and check their contents against the packing slip on the outside of the box. Some O.E.M. customers may not receive the basic or extended software kits.
- 2.20 From the tape tray in the box labelled "Extended Software", remove the tape labelled:  
DEC-08-AJAE-PB  
FOCAL 1969 + INIT
- 2.21 A. Push the HALT switch down.   
B. If you turned the TTY off, turn it back on.
- 2.22 Place the FOCAL tape in the reader, positioning the leader holes over the read station.

- 2.23 Push the HALT switch up.
- 2.24 Set all the switches 0-11 down.
- 2.25
  - A. Press ADDR LOAD then release.
  - B. Turn the reader switch to the START position.
- 2.26 Press the CONT switch, then release. The tape will read in. If it does not, perform steps 2.17 A, B, C, D, E, F and continue from 2.22.
- 2.27 Turn the rotary switch to the AC position.
- 2.28 When the tape stops, all AC lights should be out.
- 2.29 Press the CONT switch, then release. The tape will start reading again.
- 2.30 When the tape stops, all AC lights should be out. Remove the tape from the reader.
- 2.31 Set all the switches 0-11 down except for number 4.
- 2.32 Press ADDR LOAD, then release.
- 2.33 Press CLEAR, then release.
- 2.34 Press CONT, then release.
- 2.35 The computer will type,  
“CONGRATULATIONS!!  
YOU HAVE SUCCESSFULLY LOADED ‘FOCAL, 1969’ ON  
A PDP-8 COMPUTER.”  
“SHALL I RETAIN LOG, EXP, ATN ?:”
- 2.36 For more information about FOCAL, refer to the book titled “PROGRAMMING LANGUAGES” in the Extended Software box.

#### **EQUIPMENT SUPPORT SERVICES**

Digital Equipment Corporation offers many services to ensure that DEC equipment is properly operated, properly maintained, and that the down time caused by any equipment failures is minimal. Maintenance options include depot and field maintenance service, training, and information services.

#### **Maintenance and Service Options**

Digital Equipment Corporation’s Field Service Organization offers a wide range of services to accommodate the needs of DEC equipment users. Customers may choose from a selection of service contract options, per call service or depot repair maintenance, any of which will ensure a high degree of operating efficiency for DEC equipment.

#### **Service Agreement**

DEC offers a total equipment maintenance program to all customers through the Service Agreement. Behind this agreement are 15 years of

solid experience in computer manufacture, installation and servicing for virtually every kind of user. In addition to supplying all parts, labor, travel expenses and test equipment required for remedial maintenance, the service agreement provides the key to sustained reliability through regular preventive maintenance. This carefully planned program of diagnosis and testing helps identify weak component or modules which are then replaced, thereby reducing the probability of future failures.

Engineering changes are automatically incorporated in covered systems at no cost to contract customers. These changes reflect the latest advances in computer technology to improve maintainability and reliability. As the computer ages, it will continue to measure up to the current state of the art in system performance.

Simplified planning and budgeting are added features of the service agreement. Since spare parts and their related logistics networks are supplied by DEC, inventory investments are minimized. Precise financial planning is possible, with the customer satisfying his total maintenance requirement through fixed monthly payments.

#### **Service Contract Coverage**

On-call contract coverage provides remedial maintenance when DEC is notified of a system malfunction. Preventive maintenance is scheduled during the period of coverage selected by the customer, which can be as little as 8 hours a day, Monday through Friday, and as much as 24 hours a day, 7 days a week. The contract coverage begins with the principal Period of Coverage, which consists of 8 consecutive hours of on-call coverage from 8:00 AM to 5:00 PM, Monday through Friday. This coverage may then be extended in 4-hour increments.

In lieu of on-call contract coverage, an on-site resident engineer may be required due to the critical nature of an application or the size and complexity of a particular installation. The services of a resident engineer consist of 40 hours of coverage during the normal work week. In addition, all necessary spare parts, materials, and test equipment are physically stationed at the user's site to further ensure prompt, efficient remedial and preventive maintenance.

#### **Eligibility for Service Contract Coverage**

A pre-service contract inspection will be required for installations which were not under the maintenance agreement immediately prior to the requested commencement date of the Service Agreement. All charges associated with repairs, including travel, labor and parts, will be billed to the user at the prevailing standard DEC rates. No pre-service contract inspection is required for service agreements scheduled to continue immediately after the expiration of a standard DEC warranty or service contract.

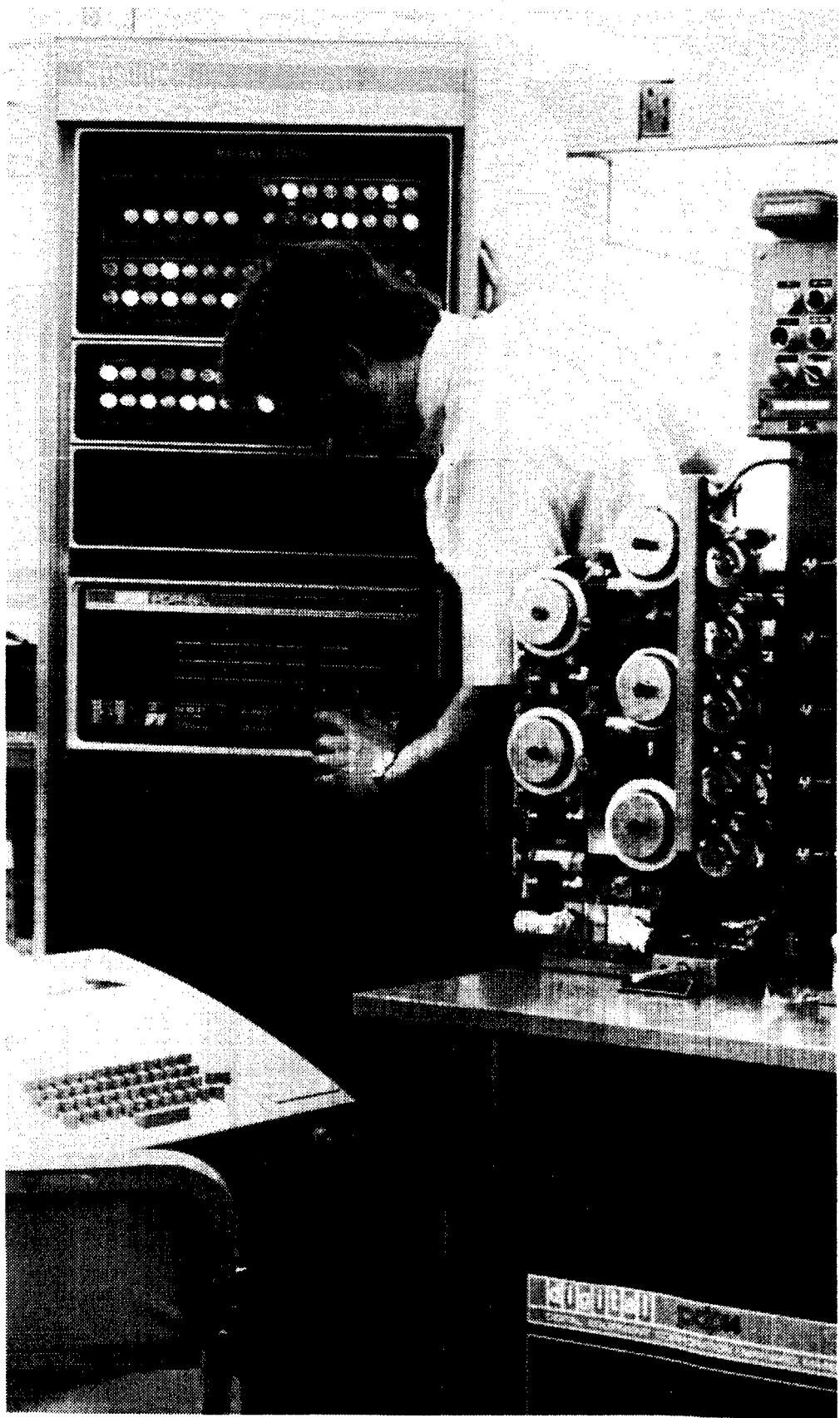
#### **Depot Repair**

Users who operate on a tight budget but do not want to compromise on service for Teletypewriters, small systems, and the like can profit from Digital Equipment Corporation's Field Service Depot maintenance program.

Field Service Depots provide cash-and-carry maintenance and repair service for many standard processors, console terminals and I/O devices. Depot maintenance services eliminate the cost of a service engineer's travel time and expenses while affording a lower hourly labor rate. When equipment requires maintenance or repairs, the user may simply contact the nearest depot and then send or bring in the equipment for servicing.

Depot repair services are strategically located throughout the world. At any of these depots, customers benefit from the same experienced personnel providing the same services they would receive if they requested a service call, but at a considerably lower cost.





# omnibus interfacing

Interfacing to the PDP-8/E OMNIBUS is accomplished with both hardware and software. For standard interfaces, DEC supplies necessary option modules and the equivalent programs (subroutines and MAIN-DECs) necessary to perform those DEC defined functions. This software is quite adequate to satisfy the operational requirements of each standard option as defined in Chapter 7. However, in the event that the user desires program functions different from those given in the software packages, he must change the software using the standard input/output devices such as a teleprinter, a card reader, or the console switches. Before undertaking this, however, the user should acquaint himself with the software routines (good documentation of the software is provided with each program tape) and read INTRODUCTION TO PROGRAMMING, a volume in the DIGITAL small computer handbook series.

This chapter provides the necessary information for users desiring to build a special interface. It deals primarily with the hardware consideration and it is understood that the user must create his own program for his own defined functions. If the user lacks sufficient experience to design his interface, he can contact his local DEC Sales Office for special assistance.

The means of transferring commands and signals from module to module is accomplished on what is called the "OMNIBUS." All PDP-8/E modules, including options, plug into the OMNIBUS in a significantly accessible manner.

The OMNIBUS is an etched board with rows of connectors soldered to the board. The pin assignment is the same on all connectors. Thus, the OMNIBUS accommodates 96 signals, which feed to 96 pins on the connectors. The user is generally only concerned with those signals that control data transfers, address memory, or contain the data to be transferred. However, the additional signals, such as timing, are readily available on the OMNIBUS to accommodate any tailor-made requirement in the event that the user should design and build his own interface module.

Many advantages are derived from the OMNIBUS approach. Because all connectors on the OMNIBUS contain the same signals, a module can be placed anywhere on the bus at the convenience of the user. All random wiring is eliminated with this type of arrangement. This feature provides greater performance, and reliability. Considerable space is conserved; thus providing a unique packaging capability that allows a high density of electronic circuitry in a small area.

When interfacing to the PDP-8/E, the designer may consider the OMNIBUS as his interface. If he follows the rules specified in section 1 of this chapter, he is more than half way toward designing his own interface. The nature of the OMNIBUS and all 96 signals are defined in a manner that makes interfacing relatively easy to accomplish.

Section 2 identifies the Data Transfer types and some guidelines to help the designer choose the transfer techniques for his needs; section 3 provides a general guideline for the designer building a Programmed I/O Interface Control Module; section 4 provides a general guideline for the designer building either a single-cycle or a three-cycle Data Break Interface; section 5 provides general design and construction guidelines. Section 6 includes some PDP-8/E interface hardware.

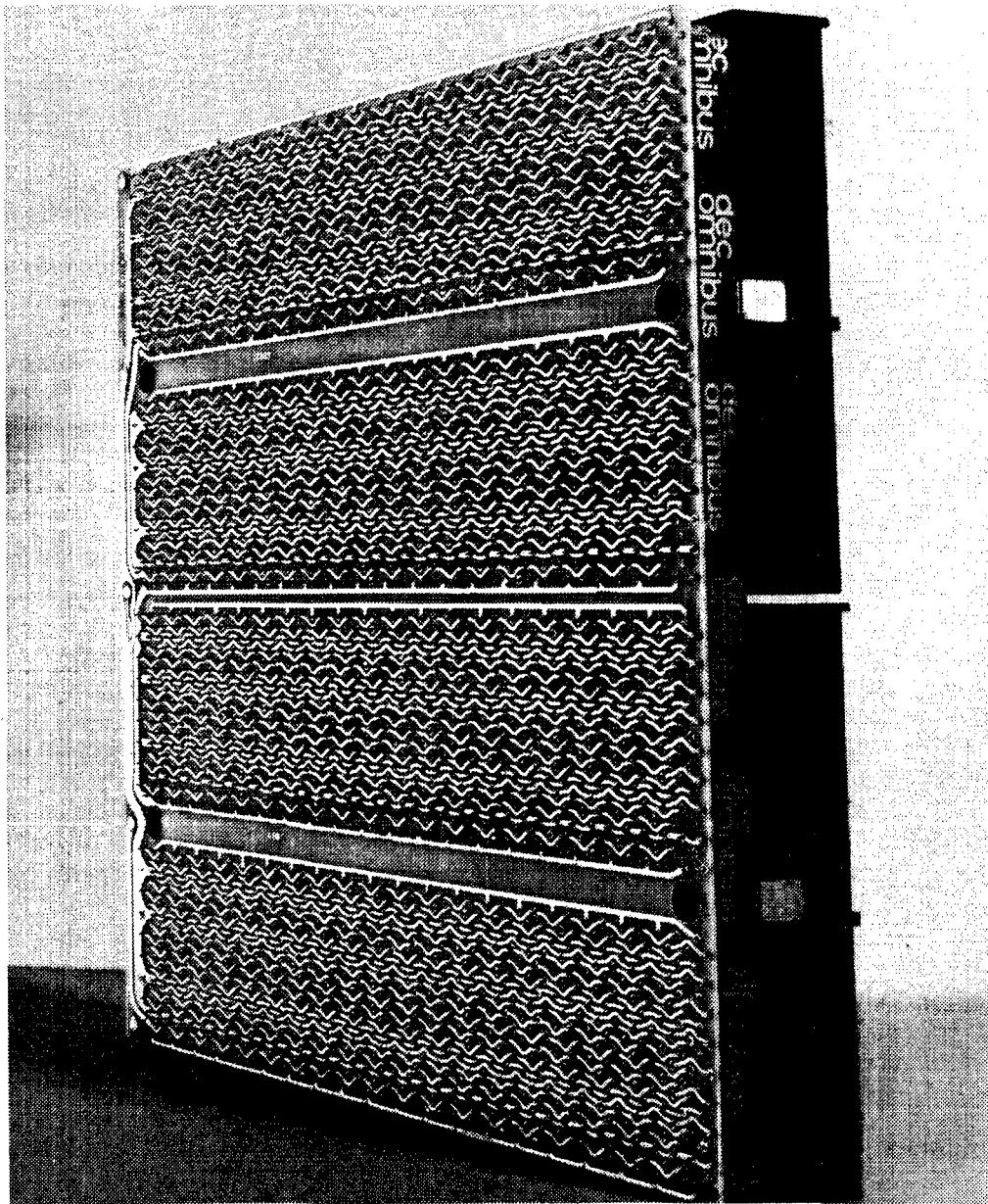


Figure 9-1 PDP-8/E OMNIBUS

## SECTION 1 OMNIBUS DESCRIPTION

### BUS STRUCTURE

The OMNIBUS (H919 OMNIBUS Assembly) is a back plane etched circuit board with ten H803 connectors mounted onto the board and wave soldered. The OMNIBUS is 10 $\frac{1}{2}$  inches by 10 $\frac{1}{2}$  inches with a 1 $\frac{5}{8}$  inch thickness. The OMNIBUS is attached to the bottom of the PDP-8/E mounting box and is the means by which all modules are connected. Figure 9-1 shows the OMNIBUS with all connectors mounted. A single assembly accommodates 20 PDP-8/E modules.

The OMNIBUS is designed so that all back plane wiring is eliminated and so that every pin in a given connector slot is defined. All modules plugging into the bus are PDP-8/E modules. If a functional unit on the bus requires more than one module, Type H851 edge type connectors on the top of the board connect multiple boards together. For cables to the "outside world," connectors on the side of the module connect to a shielded coaxial or flat ribbon cable. In this arrangement, up to 2 connectors for each module may be used.

Figure 9-2 shows the OMNIBUS with modules plugged into it. Each module functional unit can be placed anywhere on the bus or removed from the bus without affecting the operation or performance of the rest of the system not requiring that module.

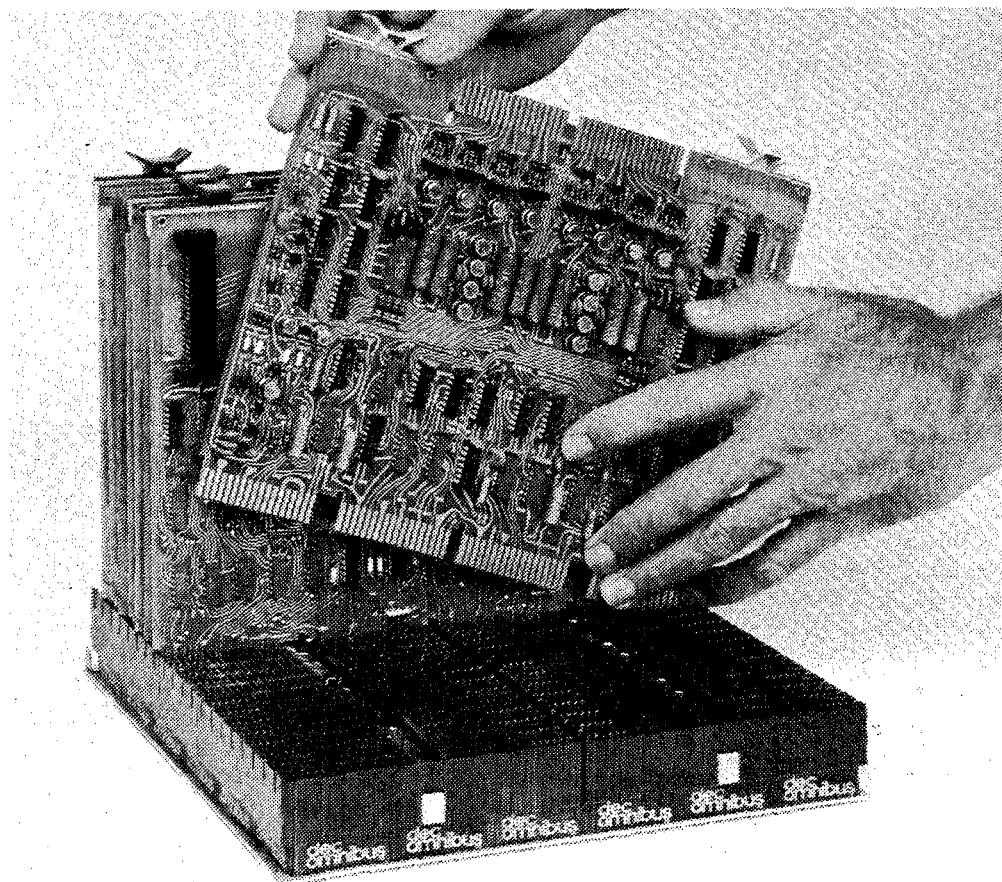


Figure 9-2 PDP-8/E Modules mounted on the OMNIBUS

Each OMNIBUS contains 20 slots; two assemblies with a M935 bus connector provides 40 slots, two of which are used to interconnect the assemblies. Thus, 38 slots are available when the OMNIBUS Expansion unit is used. However, the OMNIBUS can be expanded to accommodate an additional 37 modules. This is illustrated in Figure 9-3, which shows the basic OMNIBUS connected to the OMNIBUS expansion unit.

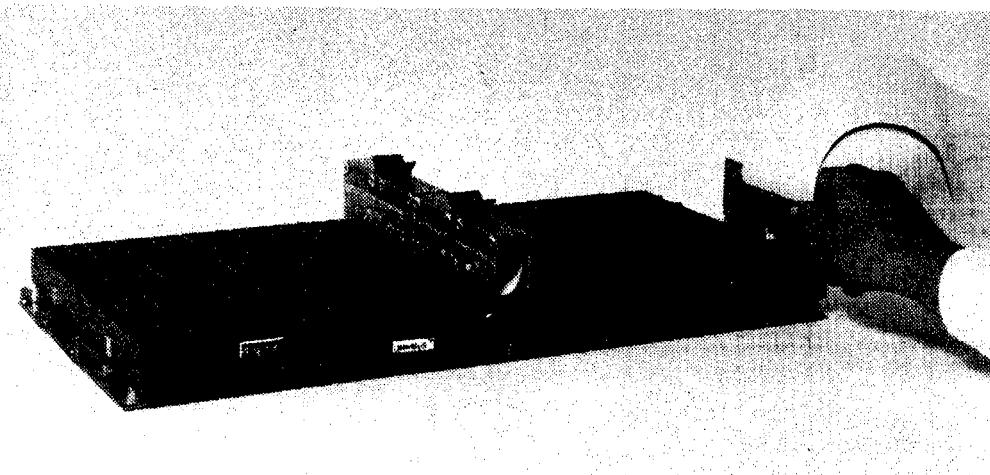


Figure 9-3 The System Expansion Capability.

#### BUS SPECIFICATIONS

##### Logic Levels

Logic 1—Max Voltage:	0.4V
Min Voltage:	-0.5V
Logic 0—Max Voltage:	5.0V
Min Voltage:	3.0V

#### SYSTEMS CONFIGURATION

The PDP-8/E with all primary options is identified in Figure 9-4. The basic system contains the central processor (4 modules) the programmer's console (1 module), 4K memory (3 modules), a shield (1 module) and a console Teletype control (1 module).

System expansion is easily accomplished simply by adding "off-the-shelf" control units necessary to accommodate the corresponding peripheral equipment, if additional machine capability is desired. For example, if it were desired to add additional memory capability to a basic PDP-8/E, a Memory Extension Control and Time Share Option, type KM8-E, could be added. Then 4K memory units could be added, up to a maximum 32K capability. For those customers who wish to use PDP-8/I or PDP-8/L compatible peripherals, an external bus option such as the Positive I/O Bus Interface Module, type KA8-E, and the Data Break Interface Module, type KD8-E, connects to the OMNIBUS to provide interfacing capabilities.

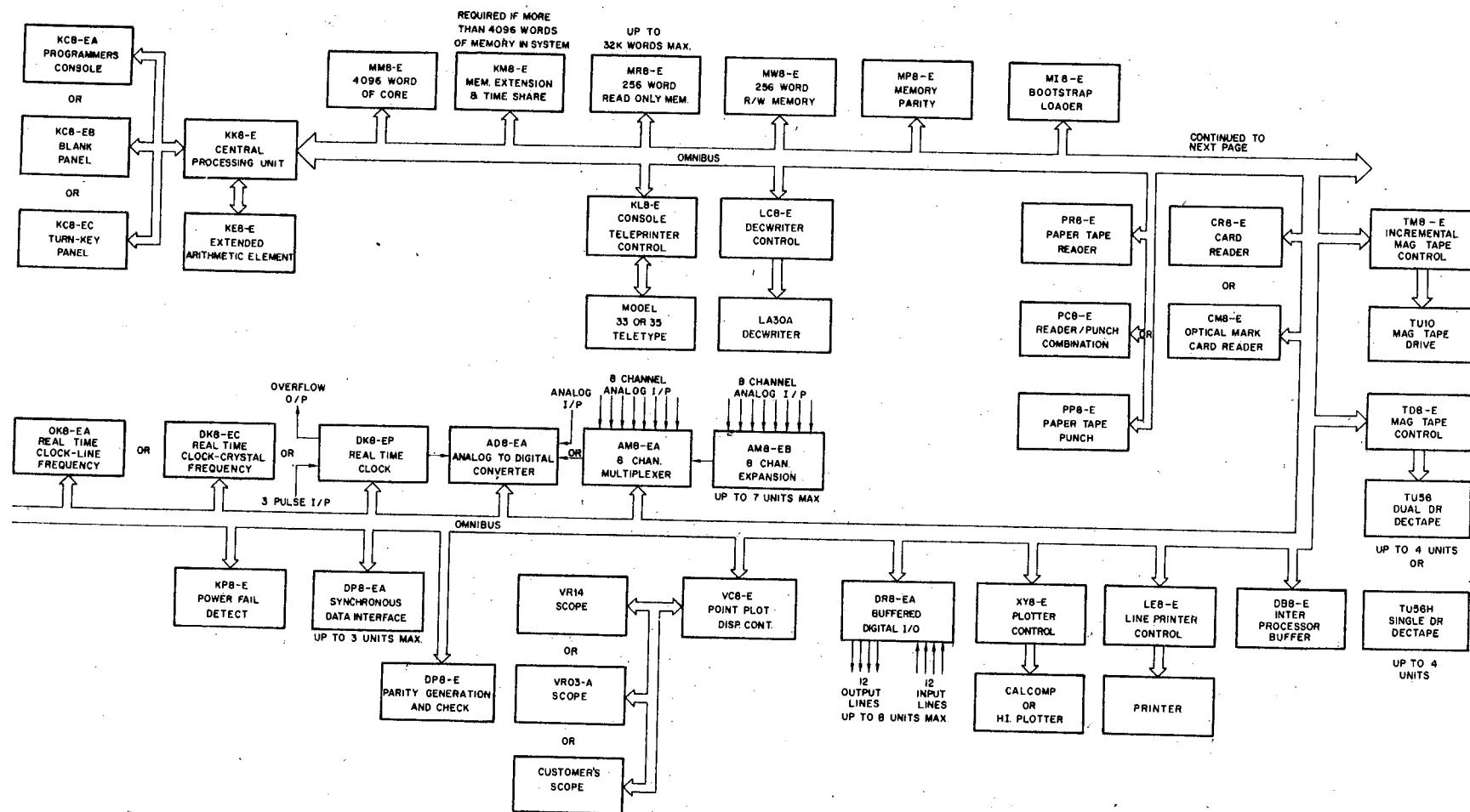


Figure 9-4 PDP-8/E—OMNIBUS Configurator

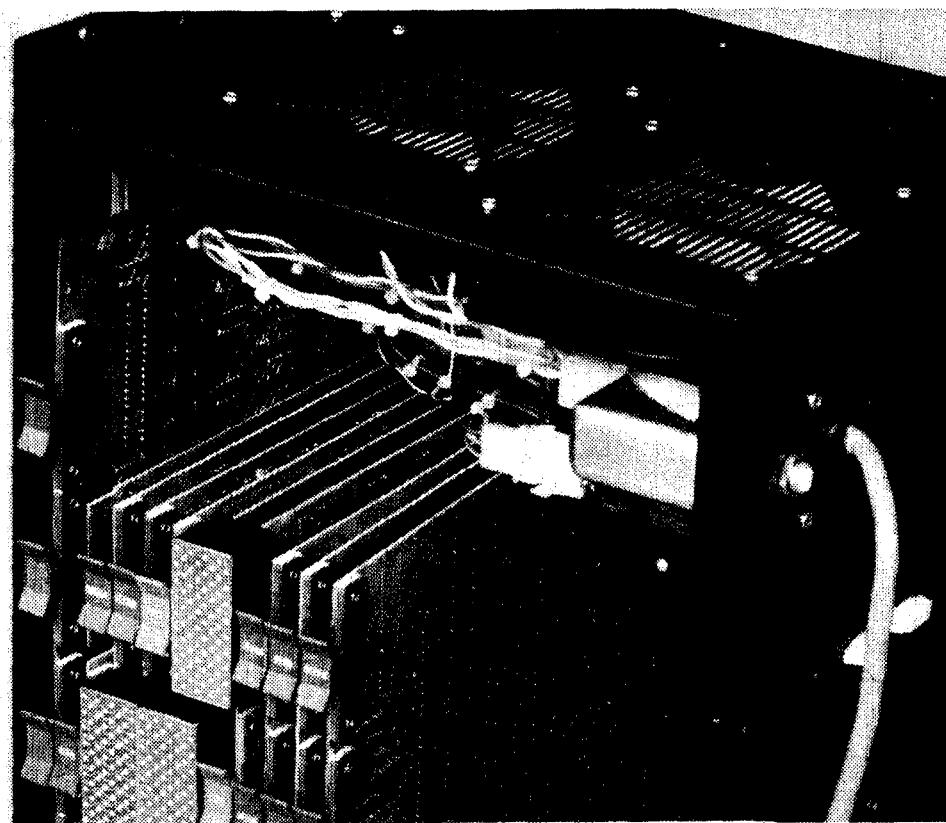
Almost all types of peripherals are included as an option to allow the user to expand his system to his own requirements. However, in the event that the user has a unique requirement such as a special control system, he may build his own control module by following the rules specified in this chapter. Refer to Chapter 11 for planning and installation.

### **RELATIONSHIP OF THE EXTERNAL BUS TO THE OMNIBUS**

The External Bus, which is mechanically and electrically organized the same as the I/O bus on the PDP-8/L or the PDP-8/I with KA8-I, plugs into the OMNIBUS by way of the Positive I/O Bus interface and the Data Break interface. Each of these modules receives the same signal on the same pins as any other module plugged into the OMNIBUS. The interfacing details to the External Bus are given in Chapter 10 of this handbook.

### **OMNIBUS SIGNALS**

The signals and pin assignments of the OMNIBUS are given in Figure 9-5. The L and H after the signal name identifies the most common assertion level. Bus Loads are provided in Figure 9-6. Each load corresponds to a description of each signal that is provided in Tables 9-1 through 9-4. The tables also identify the specific circuit by type 1 through 10 under the column heading "TYPE LOAD", "TYPE DRIVER". The corresponding circuit type is illustrated in Figure 9-6. The loading rules presented later in this chapter provide information on the electrical properties of these lines.



9-7

PIN	D1	D2	C1	C2	B1	B2	A1	A2
A	TP	+15V	TP	+5V	TP	+5V	TP	+5V
B	TP	-15V	TP	-15	TP	-15V	TP	-15V
C	GND	GND	GND	GND	GND	GND	SP GND *	GND
D	MA8L	IRØ L	I/O PAUSE L	TP1 H	MA4 L	INT STROBE H	MAØ L	EMAØ L
E	MA9 L	IR1 L	CØ L	TP2 H	MA5 L	BRK IN PROG L	MA1 L	EMA1 L
F	GND	GND	GND	GND	GND	GND	GND	GND
H	MA10 L	IR2 L	C1 L	TP3 H	MA6 L	MA,MS LOAD CONT L	MA2 L	EMA2 L
J	MA11 L	FL	C2 L	TP4 H	MA7 L	OVERFLOW L	MA3 L	MEM START L
K	MD8 L	DL	BUS STROBE L	TS1 L	MD4 L	BREAK DATA CONT L	MDØ L	MD DIR L
L	MD9 L	EL	INTERNAL I/O L	TS2 L	MD5 L	BREAK CYCLE L	MD1 L	SOURCE H
M	MD10 L	USER MODE H	NOT LAST XFER L	TS3 L	MD6 L	LA ENABLE L	MD2 L	STROBE H
N	GND	GND	GND	GND	GND	GND	GND	GND
P	MD11 L	F SET L	INT ROST L	TS4 L	MD7 L	INT IN PROG H	MD3 L	INHIBIT H
R	DATA 8 L	PULSE LA H	INITIALIZE H	LINK DATA L	DATA 4 L	RES 1 H	DATA Ø L	RETURN H
S	DATA 9 L	STOP L	SKIP L	LINK LOAD L	DATA 5 L	RES 2 H	DATA 1 L	WRITE H
T	GND	GND	GND	GND	GND	GND	GND	GND
U	DATA 10 L	KEY CONTROL L	CPMA DISABLE L	IND1 L	DATA 6 L	RUN L	DATA 2 L	ROM ADDRESS L
V	DATA 11 L	SW	MS,IR DISABLE L	IND2 L	DATA 7 L	POWER OK H	DATA 3 L	LINK L

\* THIS PIN IS CONNECTED TO GROUND ON THE BUS, BUT SERVES AS A LOGIC SIGNAL WITHIN MODULES TO FACILITATE TESTING.

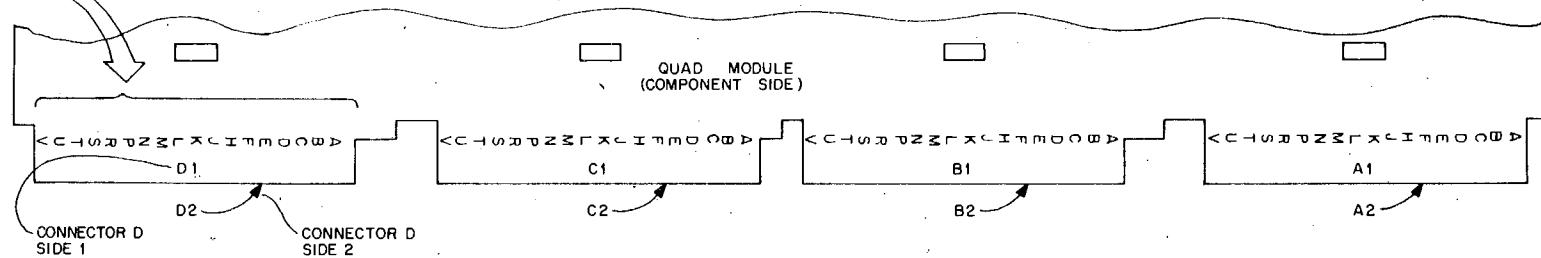


Figure 9-5 OMNIBUS Pin Assignment

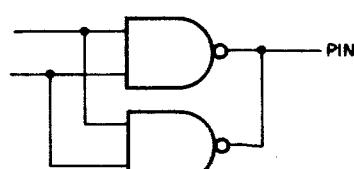
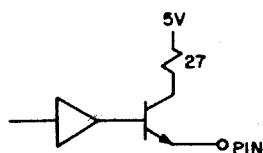
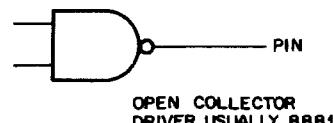
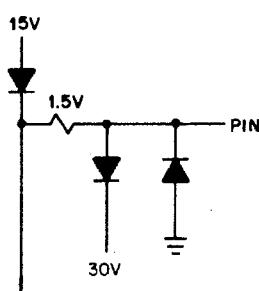
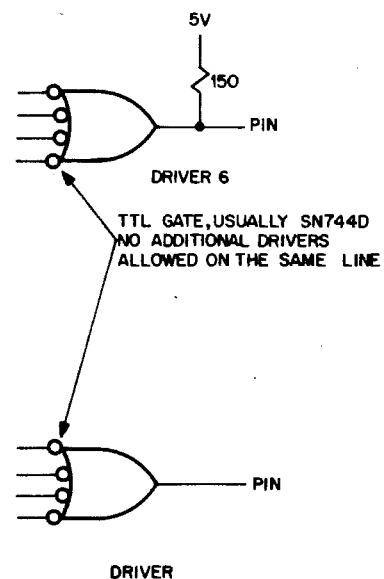
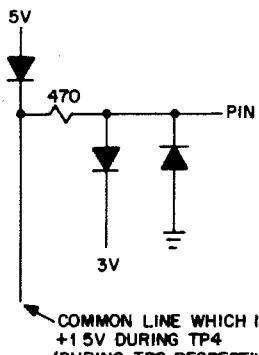
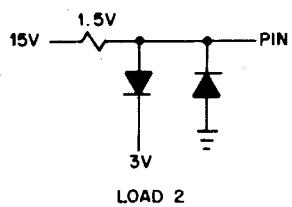
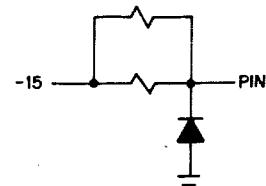
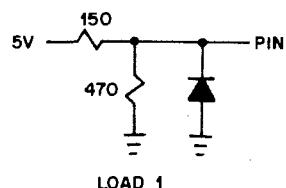


Figure 9-6 OMNIBUS Loads

**Table 9-1 Programmed I/O OMNIBUS Signals**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
MDO-11	Memory	Provides IOT instruction as follows: 6 <sub>8</sub> (used by processor) Device operation code 0 1 2 3 4 5 6 7 8 9 10 11 Device Select Code LOGIC STATES: Ground = 1 +3V = 0	2	8
I/O PAUSE L	Processor	Used to gate the device select and device operation codes into the programmed I/O interface decoders and generate BUS STROBE at TP3 and NOT LAST XFER H. I/O PAUSE is grounded when MDO-2 equals 6 (octal) during FETCH and not USER MODE. PAUSE begins 150 ns after the start of TP1 and continues until 150 ns after the start of TP3 if INT STROBE is high.	1	6
TP 3 H	Processor	TP3H is used to clear the flag and clock the output buffer of a Programmed I/O interface. It is generated in the timing generator as a positive-going 100 ns pulse. (See timing pulses in Table 9-1c)	1	6
INTERNAL I/O L	Interface	Signal INTERNAL I/O is grounded by the device selector decoder. The Positive I/O Bus Interface cannot generate IOP's when this line is grounded. This inhibits decoding any Internal OMNIBUS IOT instructions. Failure to ground this line will result in long IOT timing.	2	8
DATA0-11	Processor and Interface	The 12 DATA lines called DATA BUS serves as a bidirectional bus for both input and output data, between the AC register in the processor and the interface buffer register. The proces-	4	8

**Table 9-1 Programmed I/O OMNIBUS Signals (Continued)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
		<p>Processor internal gating and loading is controlled by C0, C1, and C2 signals. During TS3 of an IOT instruction, the contents of the DATA BUS is applied to the processor's major register gating in accordance with the C lines. For output transfers, information must be taken from the DATA BUS by edge triggering only, using the leading edge of TP3.</p>		
C lines C0, C1, C2	Interface	<p>Signals C0, C1, C2 determine the type of transfer between a device and the processor. These lines control the data path within the processor and determine if data is to be placed onto the DATA BUS or received from the DATA BUS. They are also used to develop the necessary load control signals required to load either the AC register or the PC register. When it is time for a device to make either an input or output transfer, the device will ground the appropriate combination of C control lines so that Major Register gating and Register loading is made possible. Refer to the Table below for Control line combinations and type of transfer. When the C Control lines are grounded at the Interface, the time required for the bus lines to settle must be considered.</p> <p>If, for example, data is to be transferred from a device to the PC Register, data must be transferred from the DATA BUS (see Table 9-1a) to the adders. From the adders, data is loaded into the PC with a PC load signal. PC load is developed from</p>	2	8

**Table 9-1 Programmed I/O OMNIBUS Signals (Continued)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
		C2L•BUS STROBE. Since BUS STROBE is generated by the processor during a normal IOT, C2L should be grounded not less than 280 ns before BUS STROBE. If the PC register is to be modified (both the PC and DATA applied to the adders), C2 L should be grounded not less than 400 ns before Bus STROBE L is generated.		
SKIP L	Interface	An IOT checks the flag for a ONE state and causes the device logic to ground the SKIP line if the flag is set. The result (PC + 1) is loaded into the CPMA. The SKIP line is sampled by the processor at TP3, and must be grounded 50 ns before TP3 in order for the skip to occur.	2	8
INT RQST L	Interface	Signal INT RQST is part of the Interrupt System. It is the method by which the device signals the processor that it has data to be serviced. When the device flag is set, signal INT RQST is immediately grounded. The processor samples the INT RQST line at INT STROBE time. If all the conditions for an interrupt are met, the processor then asserts signal INT IN PROG H.	2	8

**Table 9-1a Table of Transfer Control Signals**

Type of Transfer	Transfer Control Lines			Information Gated onto the Data Bus	Bus Set-up Time with respect to BUS STROBE	Action Required by Peripheral at Interface	Action by Processor**	Contents of Data Bus During Transfer
	C0	C1	C2					
Output AC→Data BUS AC unchanged	H	H	H	AC Reg.	280 ns	Load data bus into buffer.	Transfers AC to Data Bus. AC remains unchanged.	AC register only. User modification of this type of transfer may bring undesirable results.
Output AC→DATA Bus	L	H	H	AC Reg.	280 ns	Ground C0.	Transfers AC to Data Bus and clears AC.	AC Register.
AC Cleared						Load data bus into buffer.		
Input AC V Peripheral Data	H	L	H	Peripheral Data & Contents of AC reg.	280 ns	Gate peripheral data to data bus. Ground C1.	Transfers contents of AC to the data bus. The ORed result loaded into the AC.	AC ORed with Peripheral Data.
Input Jam- Data Bus → AC	L	L	H	Peripheral data	280 ns	Gate peripheral data to data bus. Ground C0 & C1.	Transfer data bus to AC register.	Peripheral Data
Relative Jump Data Plus PC→PC	*	H	L	Peripheral data	400 ns	Gate peripheral data to data bus. Ground C2.	Transfer contents of PC and Data Bus to adders. Load the added result into the PC.	Peripheral Data
Absolute Jump Data Bus → PC	*	L	L	Peripheral data	280 ns	Gate peripheral data to data bus. Ground C1 and C2.	Transfer contents of data bus to PC.	Peripheral Data

\* Don't Care

\*\* Bus Strobe loads AC or PC.

**Table 9-2 Additional Programmed I/O OMNIBUS Signals for the Sophisticated User**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
BUS STROBE L	Interface during an extended I/O	<p>Signal BUS STROBE is used to load the AC and PC registers. It is normally grounded by the processor during an I/O PAUSE and NOT LAST XFER H at TP3 time. Consequently, unless special I/O operations are being performed, the designer of an interface need not concern himself with BUS STROBE. BUS STROBE is a 100 ns negative-going pulse.</p> <p>For input transfers to the AC, or Absolute Jumps, data must be placed on the DATA BUS a minimum of 280 ns prior to BUS STROBE.</p> <p>For Relative Jumps, data must be placed on the DATA BUS a minimum of 400 ns prior to BUS STROBE.</p> <p>Input transfers to the AC and Absolute Jumps can take place within a normal IOT. However, Relative Jumps, which require 400 ns, present a timing problem. As with any operation requiring more than 280 ns, the problem is dealt with by stopping machine timing and grounding BUS STROBE at the interface. Allow 400 ns after data is applied to the DATA BUS before grounding BUS STROBE for 100 ns. Ground NOT LAST XFER at least 50 ns before TP3. This stops the processor timing at TP3 until NOT LAST XFER is again high thereby extending the length of TS3. Timing will continue only if NOT LAST XFER is high and BUS STROBE is generated.</p>	1 at both ends of bus.	9

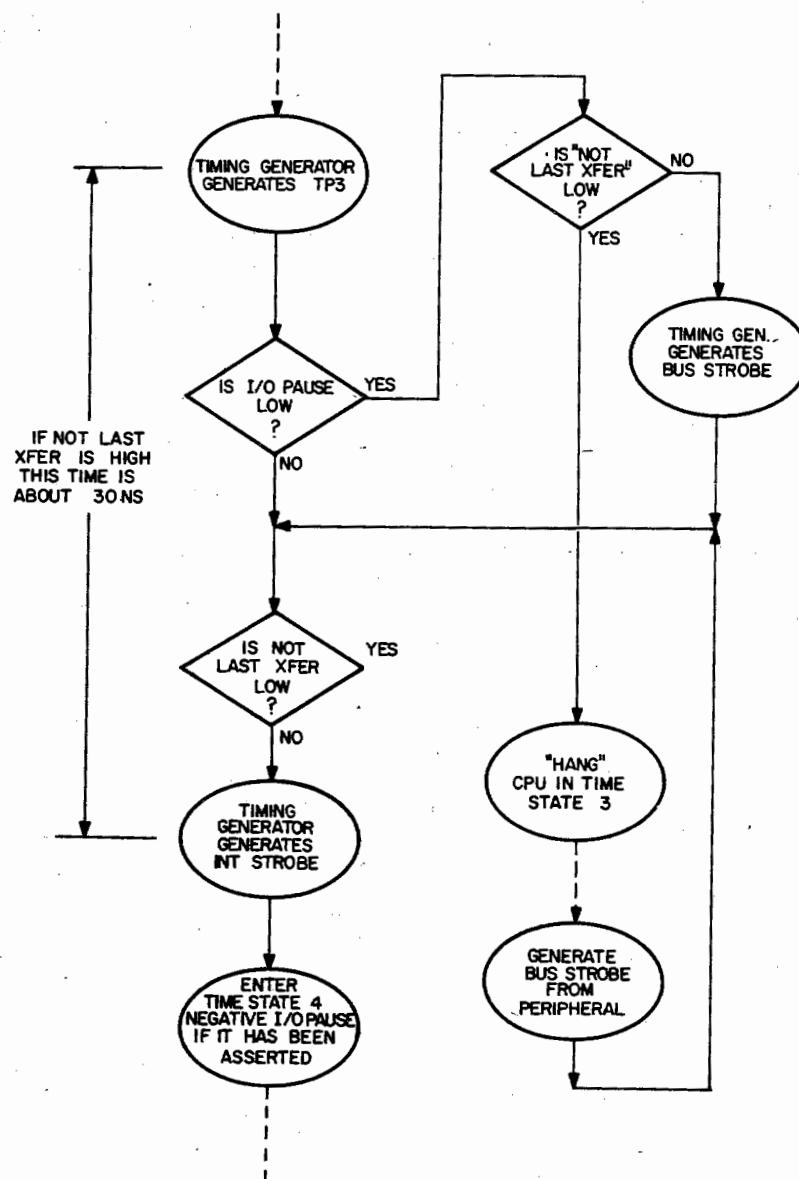
**Table 9-2 Additional Programmed I/O OMNIBUS Signals for the Sophisticated User (Continued)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
NOT LAST XFER L	I/O Interface	<p>A ground level on this line indicates to the processor that the next BUS STROBE does not terminate the I/O transaction. Since most internal I/O devices use only one transaction per IOT, this signal is normally not grounded by the internal I/O devices. Thus, the internal devices usually only asserts its "C" lines and INTERNAL I/O. However, if the transfer is such that more than 280 ns are required between the time the device data is applied to the DATA BUS and signal BUS STROBE is grounded, or if multiple transfers are being made in a single IOT, the processor timing may be stalled long enough to complete the transfer. If for example the contents of the PC is to be added to the contents of the device data, additional time beyond the 280 ns is required to allow the ripple action of the adders to be completed. In this case, 120 ns more are needed. The device must ground NOT LAST XFER at least 50 ns before TP3. At TP3, the processor timing stalls. When device data is applied to the DATA BUS, the device must wait 400 ns and then ground BUS STROBE for 100 ns. Signal NOT LAST XFER should be brought high before the time when BUS STROBE is generated. This will restart timing with TS4 and negate signal I/O PAUSE L.</p> <p>As indicated in the following flow diagram (Figure 9-7), NOT LAST XFER accomplishes three basic tasks:</p>	2	8

**Table 9-2 Additional Programmed I/O OMNIBUS Signals for the Sophisticated User (Continued)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
--------	--------	----------	-----------	-------------

- a) Determines if the timing is to stop,
- b) Determines when BUS STROBE is generated, (if extended I/O),
- c) Determines when timing is to resume.



**Figure 9-7 NLT Flow Diagram**

**Table 9-3 Data Break OMNIBUS Signals**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
MDO-11	Memory and Processor	In addition to its function under I/O, MD provides a one-way, 12-line data path between memory and the Data Break Interface. LOGIC STATES: high = 0 ground = 1	2	8
DATA0-11	Data Break Interface	The 12 DATA lines called DATA BUS are used to determine Break Priority, and to carry input data during a Data Break Cycle. LOGIC STATES: high = 0 ground = 1		
INT STROBE H	Processor	This 100 ns positive-going pulse occurs at TP3 (except for extended I/O and long EAE cycles) and is a necessary input to the timing chain to continue timing into TS4. For extended I/O and long EAE cycles, INT STROBE H is generated by BUS STROBE with NOT LAST XFER H. The leading edge of INT STROBE is the latest time in the machine cycle at which a break request can be accepted.	1	6
BREAK IN PROG L	Data Break Interface	This line is grounded at INT STROBE time if a break request is being made. This signal causes the BRK PROG lamp on the front panel to be lit during the next TS1 to indicate that a data break device has an active break request.	2	8
CPMA DISABLE L	Data Break Interface	This line is grounded by the Data Break Device at INT STROBE time if a break request is detected. CPMA BUS L causes the CP's Memory Address register to be disconnected from the MA lines at the next TP4. At the same time, the BKMA register of the highest priority device must be gated onto the MA BUS within 50 ns of the leading edge of TP4.	2	8

**Table 9-3 Data Break OMNIBUS Signals (Continued)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
OVER-FLOW L	Processor	This line is driven by a flip-flop that senses the carry from the adders at TP2. The flip-flop is set each time there is a carry or borrow out of the MB and is "ANDED" with TS3 before going to the OMNIBUS. This line is used, for example, during 3-cycle Breaks to indicate that the word count overflow has occurred.	2	8
BREAK DATA CONT L	Data Break Interface	BREAK DATA CONT is grounded when the contents of the MD are to be placed into the adders during a break cycle. This signal is used when a ONE is to be added to memory via the DATA BUS to increment either the Word Count or Current Address memory location. It is also used to perform an Add to Memory (ADM) type of Break. This line must not be changed during TS2 and is usually changed at TP4. Because MD DIR controls the transfer direction of memory data, the following truth table relates MD DIR and BREAK DATA CONT to the type of data break transfer.	2	8

#### BREAK DATA CONT USAGE

Type of Transfer	MD DIR	BREAK DATA CONT	INFO ON DATA BUS
Device→Memory	H	H	DEVICE INFO
Memory→Device	L*	X	X
Memory PLUS	H	L	0
Device→Memory	H	L	DEVICE INFO

\* Preferred Method

**Table 9-3 Data Break OMNIBUS Signals (Continued)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
MS, IR DIS L	Data Break Interface	This line is grounded at TP4 by the Break Device having the highest priority. This signal is used to disconnect the outputs of the MAJOR STATE and IR register outputs from the OMNI-BUS and from all circuitry within the CP. The processing is terminated at the end of the current instruction cycle and resumes when MS, IR DIS is again high. The start of MS, IR DIS L is the start of the DMA state. In addition, MS, IR DIS L also enables a data path from the DATA BUS to the adders to provide DATA to the MB or DATA + MD to the MB.	2	8
BREAK CYCLE L	Data Break Interface	BREAK CYCLE is grounded at TP4 by the break device having the highest priority. This signal causes the BRK lamp on the front panel to be lit during the next TS1 to indicate that the Break Cycle has started.	2	8
MA,MS LOAD CONT L	Data Break Interface	This line is grounded at TP1 by the device having the highest priority and remains grounded during Break until the TP4 following the last Break Cycle. MA,MS LOAD CONT L prevents the CPMA and MS registers from being loaded at TP4.	2	8
MD DIR	Processor or Data Break Interface	Refer to table 9-4	3	8

**Table 9-4 Basic System OMNIBUS Data and Control Signals**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
MD0-11	Processor and memory	<p>The 12 Memory Data lines carry information to and from memory of the currently addressed location. The contents of the addressed memory location are applied to the Memory Data lines beginning in the last half of TS1. If the major state is FETCH or DEFER (non-auto index), the contents of the MD lines will not change for the remainder of the cycle. The MD lines serve as the input into memory during every write operation (which occurs during TS3 and TS4). If the Major State is DEFER (auto index) or EXECUTE, the contents of the MD can change at TP2. This change is controlled by signal MD DIR which allows data to be applied to the MD lines from the memory register when MD DIR is grounded and inhibits the transfer of MB data to memory. The MB register provides the only external means of putting into memory and can do so only when MD DIR is high.</p> <p>For normal machine operation, the MD lines provide instructions, addresses, operands and data.</p> <p>For I/O devices, the MD lines provide the device select and operation codes.</p> <p>For data break devices, the MD lines carry data into the device.  <b>LOGIC STATES:</b> high = 0  grounded = 1</p>	2	8

**Table 9-4 Basic System OMNIBUS Data and Control Signals (Cont.)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
MD DIR	Processor or Programmer's Console or Data Break Device	<p>The control of external data to memory and data received from memory is provided by signal MD DIR. When high, MD DIR gates the contents of the MB register onto the MD lines and is thereby applied to memory during memory WRITE time. When grounded, MD DIR gates the contents of the memory Sense Amps out to the MD lines during the memory READ time. Thus, MD DIR can control only the place at which data is applied to the MD lines. When data is applied to the MD lines from the MB Register, data cannot be applied to the MD lines from the Sense Amps.</p> <p>During FETCH and DEFER (non-autoindex), the contents of the MD lines cannot change. The instruction or address read from memory is written back into the same memory addressed location. MD DIR L assures that these lines will not change. During EXECUTE and DEFER (autoindex), the contents of the addressed memory location are applied to the MD lines until TP2. At this time, MD DIR is brought high so that the contents of the MB Register can be applied to the MD lines and subsequently written into the same memory location during the WRITE portion of the memory cycle.</p> <p>During the manual operation of the processor from the Programmer's Console, MD DIR can be changed without considering the time states.</p>	3	8

**Table 9-4 Basic System OMNIBUS Data and Control Signals (Cont.)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
MA 0-11	Processor or Data Break Interface	<p>During a Data Break Operation, MD DIR is controlled by the Data Break device depending upon the type of transfer (input or output). MD DIR should be changed at TP1 time.</p> <p>Used to address memory to any one of 4096 possible locations. This address is changed only at TP4 time. The address is normally developed in the processor. However, during a data break, the MA lines can be used for break addresses, which originate in the data break module. When the processor is executing an instruction, the address always originates in the processor.</p> <p>LOGICAL STATES: 1 = low 0 = high</p>	3	8
EMA 0-2	Processor or Data Break Interface	<p>Used only when the extended memory is provided. These 3 bits are combined with the 12 bit Memory address to form a 15-bit memory address. This is necessary to specify one location out of 32,768 possible locations. The extended address bits specify the memory field in use. MA11 is the least significant bit and EMA0 is the most significant bit. All 12 or 15 lines are high for a zero and low for a one. Thus, if the machine does not contain a Memory Extension Control, the EMA bits are automatically zero (high), selecting the lowest 4K of memory.</p> <p>LOGICAL STATES: 1 = low 0 = high</p>	3	8

**Table 9-4 Basic System OMNIBUS Data and Control Signals (Cont.)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
DATA0-11	Nearly all portions of the machine except memory	The 12 DATA lines called DATA BUS serve as a multipurpose bidirectional bus. Generally, the DATA BUS is the in/out path between the peripheral and the AC register. However, the DATA BUS is also between the AC Register/MQ Register and the processor adders; and therefore capable of applying peripheral data or AC/MQ data to the adders.  The DATA BUS usage with respect to processor timing is illustrated in the following table.	4	8

**DATA BUS USAGE**

Machine Timing Within Major States	Major States			
	F	D	E DMA	
TS1	Indicators			
TS2	CPU	NOT USED	CPU	DATA → MB
TS3	I/O DIALOGUE (Only if an IOT Otherwise CPU)		CPU	NOT USED
TS4	Priority Determination			

MEM START	Programmer's Console or Power Fail Option	This line is grounded for a minimum of 100 ns to initiate a memory cycle. It must not be grounded after TP2. Memory cycles continue automatically until STOP is grounded.	2
ROM ADDRESS	Rom	When this line is high, the Read/Write memory runs normally. When this line is low, the Read/Write memory does not function, despite memory timing signals on the bus. This line is used when a small ROM is used	3 8

**Table 9-4 Basic System OMNIBUS Data and Control Signals (Cont.)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
		with addresses that overlap the main memory. If the ROM sees its address on the MA lines, this line is immediately grounded. If ROM ADDRESS is grounded during the Execute portion of a JMS instruction, the next instruction will be taken from the addressed location, thus saving a ROM location.		
MEMORY WRITE, SOURCE, STROBE, INHIBIT, RETURN	Timing Generator	These five signals control the memory, and may vary if different memory and timing modules are used. MEMORY WRITE is high during the write portion of the memory cycle. SOURCE is used to turn on memory current. It is high when read or write current is to be turned on. RETURN and SOURCE turn on (go high) at the same time, but RETURN turns off 50 ns later to insure that the stack does not remain capacitively charged. INHIBIT turns on the inhibit drivers when positive. STROBE provides a time reference from which the output of the sense amplifiers are sampled. STROBE goes positive before data is actually ready in the sense amplifiers. Each memory then delays this leading edge by the optimum amount. This precaution allows for stack variation. If data is read from memory, the negative going edge of STROBE indicates the data on the MD lines is valid.	3	8
IRO, 1 & 2	Processor	These 3 lines indicate the effective instruction being processed. They usually, (but not always), indicate the contents of the IR. Lines are low for a 1 and high		

**Table 9-4 Basic System OMNIBUS Data and Control Signals (Cont.)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
		for a 0. All instruction processing and major state gating is derived from these lines.		
F, D, E	Processor	The 3 major states lines, like the IR lines, indicate the effective major state. The appropriate line is low to indicate its major state. Only one of these lines should be grounded at any time.	3	8
		A fourth major state called Direct Memory Access (DMA) is activated when MS, IR DISABLE is low and F,D or E are not grounded by an external device.		
F SET		F SET is similar to major states described above. This line must not be grounded by modules external to the CP. F SET indicates the next machine cycle is a fetch unless disabled. Note that DMA State causes F SET.	2	8
LINK LOAD LINK DATA	Timing Generator, Peripherals	These two lines may be used to jam one bit of information into the LINK. LINK DATA is low for a 1 and high for a zero. LINK LOAD is normally high and is brought to ground for 100 ns (minimum) to cause loading.	LINK LOAD 1 on both ends of bus LINK DATA: 2	9 8
LINK	Processor	This line indicates the state of the link bit in the processor. It is high if the link is 0, and low if the link is a 1.	2	
LD ADD EN	Programmer's Console	When the ADDR LOAD key is pressed this signal is asserted (grounded). A data path is enabled to allow the DATA BUS to be transferred to the major register bus (see Pulse LA).	2	8

**Table 9-4 Basic System OMNIBUS Data and Control Signals (Cont.)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
KEY CONTROL	Programmer's Console	<p>This line is grounded by the Programmer's Console when the operator depresses the EXTD ADDR LOAD, EXAM, or DEP key. When EXTD ADDR LOAD key is depressed, CPMA LOAD is inhibited preventing an address, intended for the Memory Extension Control, from being loaded into the CPMA. If the DEP or EXAM key is used, CPMA LOAD is again inhibited so that data will not be loaded into the CPMA. KEY CONTROL L also generates STOP which resets RUN so that the timing will stop at the next TS1 and causes MA + 1 to go to the PC register. KEY CONTROL L is negated at TP4. KEY CONTROL L also prevents interrupts from occurring.</p> <p>When ADDR LOAD is depressed, KEY CONTROL H remains high so that CPMA LOAD may be developed.</p>	2	8
STOP	Timing Generator or Programmer's Console	STOP is asserted (grounded) by the STOP key and F SET, by the SINGLE STEP key, by KEY CONTROL (low), or by the HLT instruction. It is sampled at TP3. If this line is low, processing is stopped at the end of the current memory cycle.	2	8
PULSE LA	Programmer's Console	When ADDR LOAD or EXTD ADDR LOAD key is pressed, this positive pulse either causes the contents of the DATA BUS to be loaded into the CPMA (if KEY CONTROL is high), or causes DATA 6 thru 11 to go to the Extended Memory IB, IF, DF (if Key Control is low). Note that PULSE LA does not initiate a memory cycle (see LD ADDR EN).	5	10

**Table 9-4 Basic System OMNIBUS Data and Control Signals (Cont.)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
IND 1 & IND 2	Programmer's Console	These 2 lines control the data placed on the DATA lines at TS1 time.	2	8
		IND 1 IND 2 Effect		
		High High Status word goes to DATA BUS at TS1		
		High Low C (MQ) goes to Data Bus at TS1		
		Low High Data Bus		
		Low Low C (AC) goes to Data lines at TS1		
		Status word format:		
		Bit Function	Front Panel Abbr.	
		0 Link	L	
		1 "Greater than" flip-flop	GT	
		2 Interrupt Bus	INT BUS	
		3 No Int. Allowed	NO INT.	
		4 Interrupt On	ION	
		5 User Mode	UM	
		6 Instruction Field 0	IF 0	
		7 Instruction Field 1	IF 1	
		8 Instruction Field 2	IF 2	
		9 Data Field 0	DF 0	
		10 Data Field 1	DF 1	
		11 Data Field 2	DF 2	
SW	Programmer's Console	SW is a line controlled by front panel switch SW. When the switch lever is up, the line is low. When the lever is down, the line is high.	2	8
INT IN PROG H	Timing Generator	INT IN PROG signifies that the CP is in the process of honoring an interrupt request. This line is asserted (brought to +3V) at INT STROBE time if all	4	8

**Table 9-4 Basic System OMNIBUS Data and Control Signals (Cont.)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
		conditions for granting an interrupt request are present. It is negated at TP1 of the next cycle. Grounding this line prevents honoring an interrupt, even though all other conditions are met.		
OVER-FLOW L	Processor or Data Break Interface	This line is driven from a flip-flop that senses the carry from the adders at TP2. The flip-flop is set each time there is a carry or borrow out of the MB and is "ANDED" with TS3 before going to the bus. For the basic system, the flip-flop is set by a carry from an ISZ instruction. Refer to Table 9-1b for use in Data Breaks.	2	8
RUN L	Timing Generator	When low, RUN indicates that the machine is executing instructions. The Run Flip-Flop is set by Mem Start and cleared at TP3 if STOP is asserted.	NO LOAD	7
TS1 L, TS2 L, TS3 L, TS4 L	Timing Generator	These time state lines are high if negated, and low if asserted. Each time state precedes its corresponding time pulse. Time states are always 200 ns or more in duration, and change 50 ns after the leading edge of the time pulse. The machine is in TS1 when stopped.	1	6
TP1 H, TP2 H, TP3 H, TP4 H	Timing Generator	These 100 ns positive-going pulses originate in the timing module. The exact spacing of the timing pulses is a function of fast or slow cycle. The source of all timing is a 20 MHz crystal clock and a frequency divider. The timing generator	1	6

**Table 9-4 Basic System OMNIBUS Data and Control Signals (Cont.)**

SIGNAL	ORIGIN	FUNCTION	TYPE LOAD	TYPE DRIVER
		starts running any time Mem Start is issued, and continues to run until TP4 occurs. At that time, if STOP was negated (high) at TP3, the timing generator continues to run. Each Time Pulse except TP3 overlaps 2 Time States. For example, TP1 begins 50 ns before the end of TS1 and ends 50 ns after TS2 has started.		
POWER OK H	Power Supply	POWER OK, when high, indicates that the dc voltage from the power supply is adequate to allow proper functioning of the machine. If this line becomes negated, no new memory cycles will be started. Also, after a delay long enough to complete the current cycle, the memory drive current is inhibited. When this line is negated, INITIALIZE is generated.	5	10
INITIALIZE H	Timing Generator	INITIALIZE is a positive-going pulse of at least 600 ns duration. This pulse is used to clear AC and LINK, and to clear all flags in peripherals. It is generated if POWER OK is negated, by the Clear Key on the front panel, and by an IOT (6007).	5	10
USER MODE L	Extended Memory Control	This signal originates in the Time Share portion of the Extended Memory Control. When asserted (ground), it disables OSR, LAS, IOT, and HLT instructions. OSR and LAS are disabled at the panel by inhibiting the placing of SR on the DATA lines. The IOT and HLT instructions are disabled in the Central Processor.	2	8

## **SECTION 2 HOW TO CHOOSE THE TYPE OF I/O TRANSFER**

The type of I/O transfer must be first considered before beginning the task of designing the I/O interface.

The basic types of peripherals are used with the PDP-8/E: one that is designed to transmit or receive one character (12-bit word) per service routine by the processor; and one that is designed to transmit or receive a block of characters (a series of 12-bit words) per service routine by the processor.

### **DATA TRANSFER TYPES**

Data transfer can occur in any one of three data transfer facilities. These are: Programmed I/O Transfers, Interrupt facility, and Data Break facility.

**PROGRAMMED I/O TRANSFER**—The simplest method of accomplishing an input/output transfer is the Programmed I/O Transfer. This method relies upon the processor to occasionally check the Status Flag and service the flag with a subroutine.

**INTERRUPT FACILITY**—A more efficient method of input/output transfers is to employ the Interrupt System. This method includes all of the elements in the Programmed I/O transfer except the time of transfer. The device decides when to transfer by grounding an INTERRUPT REQUEST line. The processor responds at the end of the current instruction.

**DATA BREAK TRANSFER**—A still more efficient method of transfer is to the Data Break System. Whenever the data break device decides that it is time to transfer, it generates MS, IR DIS to force the processor into a Direct Memory Access State and CPMA DIS to disable the CPMA register. This leaves the data break device free to supply its own address and to manipulate the Major Registers Control logic so that it can input and output data at will. The processor responds to a break at the end of the current cycle. Note that, in general, data break requires more hardware than Programmed I/O. Additional logic is necessary to handle addressing, etc., and some programmed I/O is necessary to initialize and check status of the device.

### **INTERFACING TO THE PROCESSOR**

Two sides of the interfacing must be considered: the processor side and the interface control side. It is necessary to understand that both the processor and the interface control share common lines on the OMNI-BUS. Furthermore, although the interface control may place information on these common lines, only when certain control lines are asserted is information loaded into the processor. This requirement is necessary for both Data Break and Programmed I/O Transfers.

Because each line on the OMNIBUS is shared by a number of devices as well as elements of the processor, it is most important that each line be used by only one device at any given time except where specified in this handbook.

The OMNIBUS/Basic System Interface is illustrated in figure 9-8. For most I/O operations, data is received and outputted by the Major Registers module. Data is received from the OMNIBUS 12 data lines called DATA BUS and applied to the input gates. By asserting the appropriate control lines, data can be loaded into the MA, MB, MQ, AC or PC register; or be added to existing data in one of the registers and correspondingly become new data for the DATA BUS or Memory Data or a new memory address. Thus for I/O operations, the user's path into the basic system is via the DATA BUS. Data transmitted from the processor is also via the DATA BUS except when using a Data Break Device. Notice that the MD lines on the OMNIBUS are connected to memory via a two-way path. When a Data Break Controller is connected to the MD lines, information may be transmitted from memory without having to go through any Major Register. However, from the Data Break device, data is always received on the DATA BUS and applied to memory via the Major Registers.

A more comprehensive picture of both Data Break Control and I/O Interface Control operating with the basic system is provided in figure 9-9. The logical sections of both I/O interface and Data Break interface are identified. Each of these sections consists of simple logic and are expanded upon in sections 3 and 4 of this chapter.

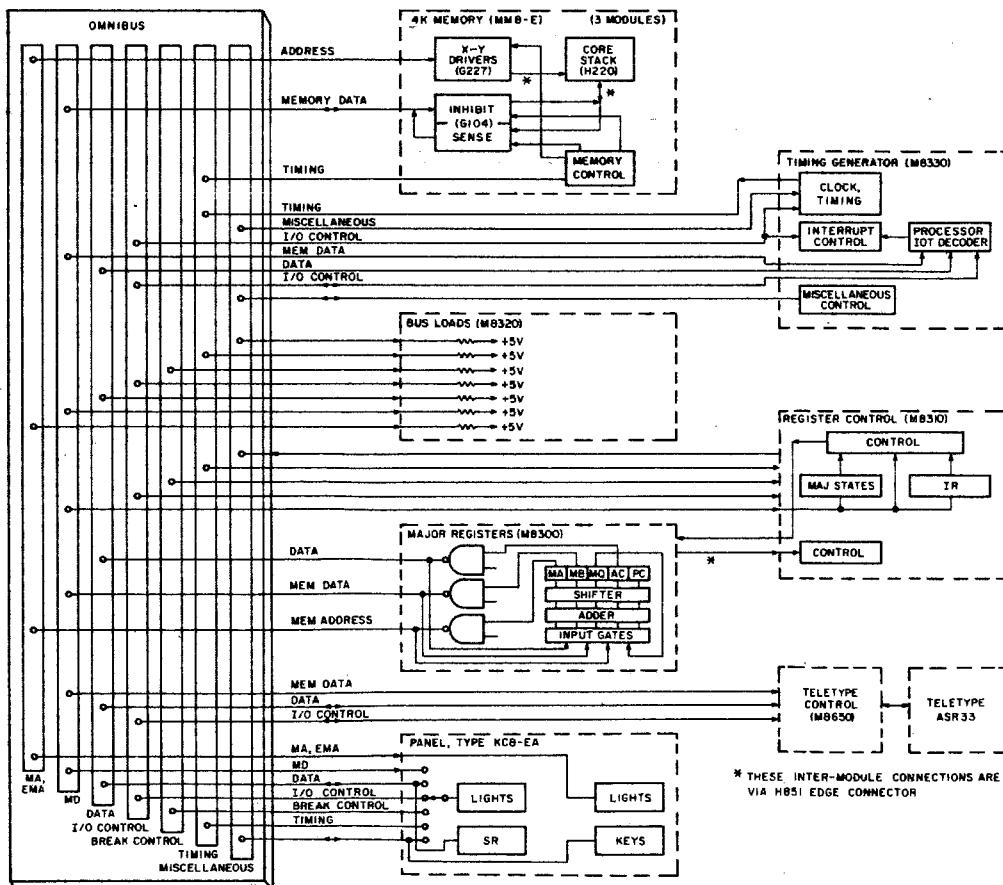
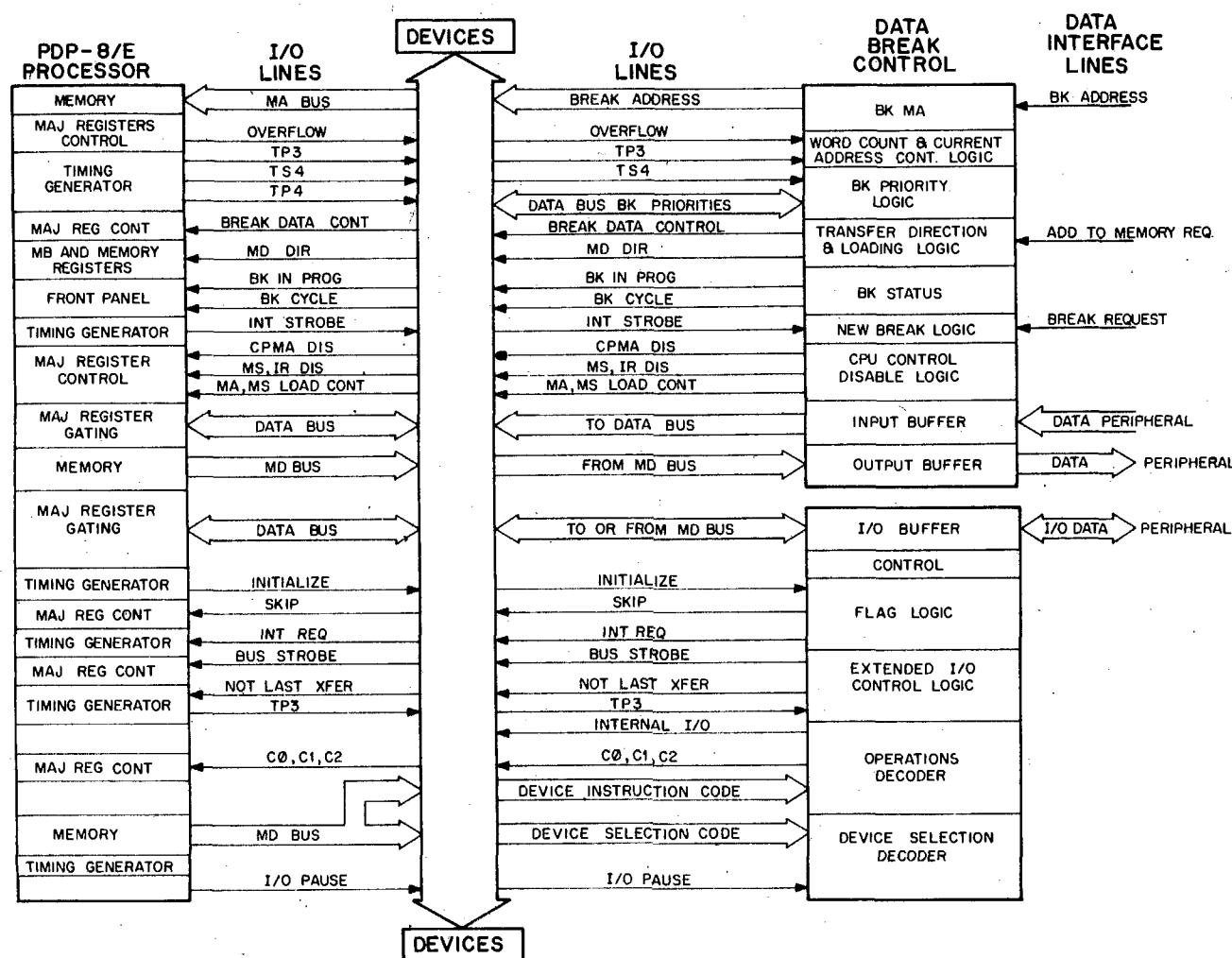


Figure 9-8 OMNIBUS/Basic System Interface

Figure 9-9 OMNIBUS/ I/O Signal Interface Block Diagram



### **DATA TRANSFER RATES**

One means of determining the type of data transfer is by examining your projected data transfer rates. Figure 9-10 illustrates a general guideline based upon transfer rates. However, it does not consider the number of I/O devices or the amount of calculations expected of the processor.

Any device with transfer rates up to approximately 60 characters per second can be easily serviced by a simple programmed I/O. However, adding more interfaces requires a closer examination. The Programmed Interrupt System should be employed when using transfer rates above 60 characters per second up to 5K characters per second. For data rates above 5K characters per second, the Data Break Facility should be employed.

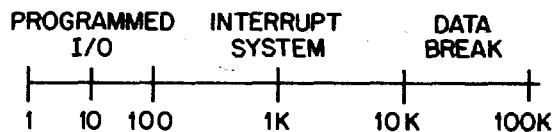


Figure 9-10 Data Transfer Rates (Characters Per Second)

## **DEVICE CODES**

The device codes for selecting each device are given in Table 9-5.

**Table 9-5 PDP-8/E Device Codes**

00	Central Processor, Type KK8-E
01, 02	High Speed Tape Reader/Punch & Control, Type PC8-E, PP8-E, PR8-E
03, 04	Console Teletypewriter Control, Type KL8-E
05, 07	Oscilloscope Display Control, Type VC8-E
10	Power Fail Detect and Auto Restart, Type KP8-E
10	Memory Parity, Type MP8-E
11	Redundancy Check Control, Type DP8-EP
12	Reserved for Card Punch and Control
13	Real Time Clock, Type DK8-EA, DK8-EC, DK8-EP
14-17	Reserved for Special Systems and Customer's use
20-27	Memory Extension and Time Share Control, Type KM8-E
30-37	
36, 37	General Purpose Interface, Type BB08
40-47	Synchronous Data Interface, Type DP8-EA, DP8-EB
50-51	
52	Analog Multiplexer, Type AM8-EA
53	A/D Converter, Type AD8-EA
50-57	Buffered Digital I/O, Type DR8-EA
60-62	Disk and Control, Type DF32-D
60-62, 64	Disk Control, Type RS08
63, 67	Card Reader and Control, Type CR8-E, CM8-E
65	Plotter Control, Type XY8-E
66	Line Printer and Control, Type LE8
70-72	Industry Standard Magnetic Tape and Control, Type TM8-E
70-77	DECtape Control, Type TD8-E
73-75	Disk File and Control, Type RK8
76-77	DECtape Control, Type TC08

### SECTION 3 DESIGNING BASIC PROGRAMMED I/O INTERFACE CONTROL CIRCUITS

The basic interface control circuits to either transfer data in or transfer data out consists of: 1) a device selection circuit, 2) a device operations decoder, 3) I/O control logic, and 4) input/output buffers. An example of a basic programmed I/O interface control is illustrated in figure 9-11 followed by the related timing.

A general rule to follow for interfacing with the OMNIBUS is given as follows:

**SIGNALS TAKEN OFF THE OMNIBUS:** Use DEC380, DEC314, or DEC384 ICs (or equivalent). This is recommended to minimize bus loading.

**SIGNALS PLACED ONTO THE OMNIBUS:** Gate signals onto the OMNIBUS with DEC8881 ICs (or equivalent). This is recommended because of the low leakage current characteristic that allows a greater number of devices to be "wired ORed" together.

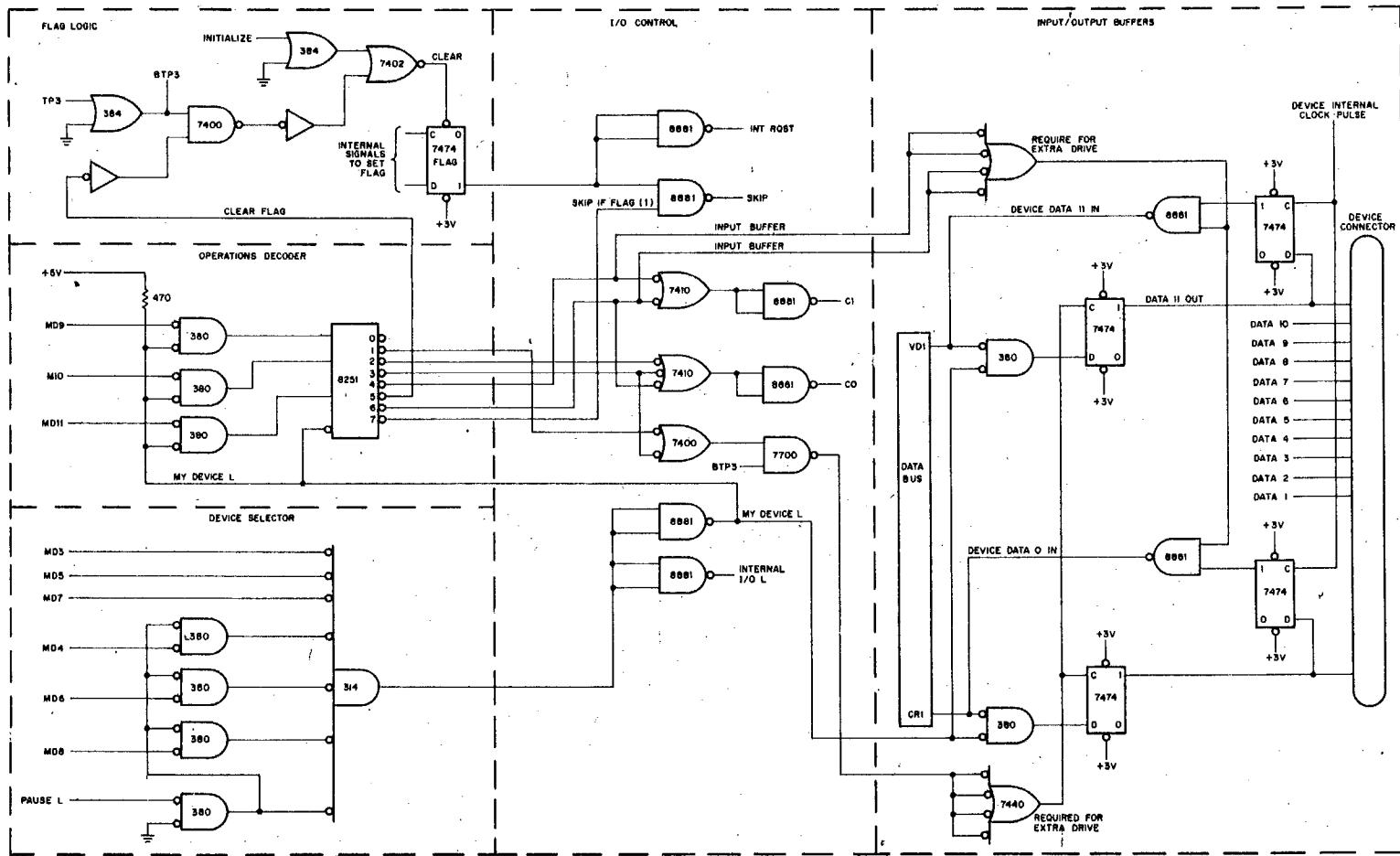
**DEVICE SELECTION CIRCUIT**—MD3-8 bits are used to carry the device select information. The example given in figure 9-11 shows the DEC380 and DEC314 being used as a simple decoder. When octal 52 is received and signal PAUSE is grounded by the processor, gate 314 is qualified. The output is used to assert signals INTERNAL I/O L and MY DEVICE L. No operation can occur unless signal MY DEVICE is grounded by the device selection decoder.

**OPERATIONS DECODER**—MD9-11 bits determine the type of operation to be performed. Three DEC380's are shown (see figure 9-11) receiving these bits. The outputs of these gates are in turn presented to a binary-to-octal decoder type 8251 and the decoded results control the interface in the manner shown in table 9-6.

Table 9-6 Operations Decoder IOT's

IOT	ACTION REQUIRED BY INTERFACE	RESULT
6521	CLOCK OUTPUT BUFFER	AC→DATA BUS→DEVICE AC UNCHANGED
6522	GROUND C0	CLEAR AC
6523	CLOCK OUTPUT BUFFER & GROUND C0	DATA BUS→DEVICE; CLEAR AC
6524	GATE DEVICE DATA ONTO DATA BUS AND GROUND C1	DEVICE DATA ORed with AC→AC
6525	CLEAR FLAG	CLEAR FLAG
6526	GATE DEVICE DATA ONTO DATA BUS AND GROUND C0 and C1	JAM INPUT OF DEVICE DATA→AC
6527	GROUND SKIP line if flag(1)	PC + 1→CPMA

Figure 9-11. Sample Programmed I/O INTERFACE Control Logic



**FLAG LOGIC**—The flag is represented as a 7474 D-type flip-flop. For an input transfer, the flag may be clocked by the device internal clock pulse with a DATA IN signal used as the data input. If the transfer is to be an output transfer, the clock input could be a timing pulse and the data input could be the output of the operations decoder. The flag represented in the example is used for an input transfer. For both input and output transfers, two flags are required.

**INTERRUPT REQUEST**—The processor responds to the INT RQST line by completing the current instruction and then executing a JMS to location 0. Simultaneously, the interrupt system is turned off. The execution of the JMS instruction saves the current program count in location 0. It is up to the program to identify the interrupting device by polling sequentially (testing) device flags. After the device has been serviced, the interrupt service routine returns to the main program with a JMP I 0 instruction.

**OUTPUT BUFFER**—The output buffer serves to receive processor data during an IOT instruction and outputs data to a device at the device timing. Two types of output transfers can be made depending upon the device. A parallel to parallel transfer will transfer the parallel data from the DATA BUS to an equal number of parallel lines to the device. A parallel to serial transfer will load the parallel data from the DATA BUS into the buffer and shift the data out to a single output line to the device. Figure 9-12 illustrates a parallel to serial output buffer. Signal LOAD BUFFER gates the contents of the DATA BUS onto the set side of the register flip/flops. This illustration shows 8 data bit flip/flops in between an ENABLE and LINE flip/flops. A shift control circuit must also be added to provide the necessary buffer control.

A more simple version is the parallel to parallel transfer. The illustration in Figure 9-12 could be slightly modified to include only the input gates and the flip/flops; data can be applied to the output circuit from the one side of each flip/flop.

In the sample Programmed I/O Interface Control (figure 9-11), each input flip-flop is a type 7474 and is represented on the illustration as a 12-bit buffer register. The data OUTPUT is clocked by IOT 6521 or 6523 and TP3. However, for a parallel to serial conversion (figure 9-12), the LOAD IOT loads the buffer with TP3H and shifts the data with each internal clock & shift pulse.

**INPUT BUFFER**—The input buffer serves to receive device data at the device timing and applies the data to the DATA BUS during an IOT instruction. Figure 9-13 illustrates a serial to parallel input buffer representing 8 bits applied to DATA 4 through 11 of the DATA BUS. A slight modification eliminating the shift function and having each data input applied to the corresponding flip/flop makes the serial input into a parallel input type buffer.

In the sample Programmed I/O Interface Control (figure 9-11), each buffer register is a type 7474 and is represented on the illustration as a 12-bit buffer register. The data input is clocked in by the device's internal timing and gated out to the DATA BUS by IOT 6524.

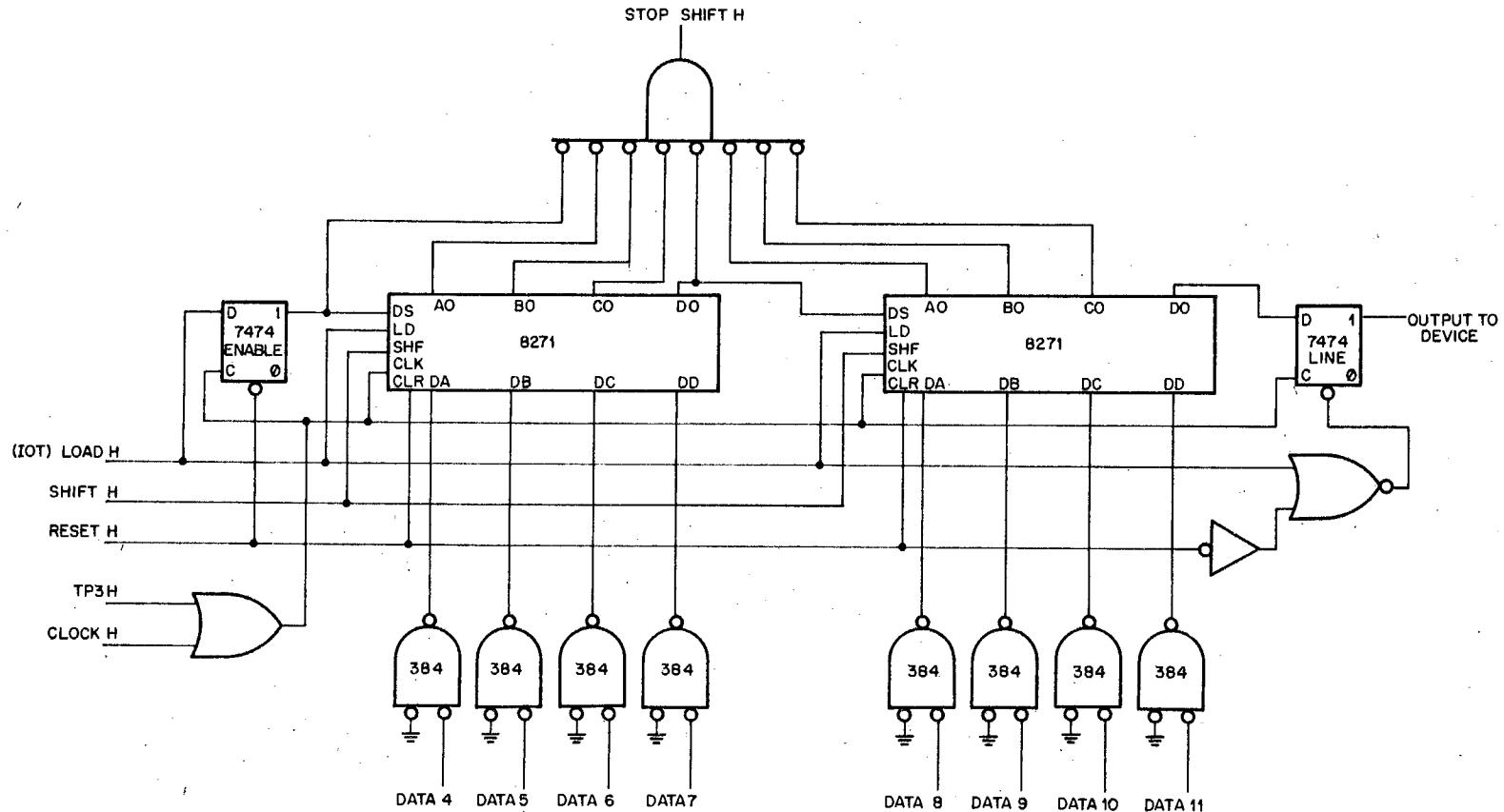


Figure 9-12 Parallel to Serial Output Buffer

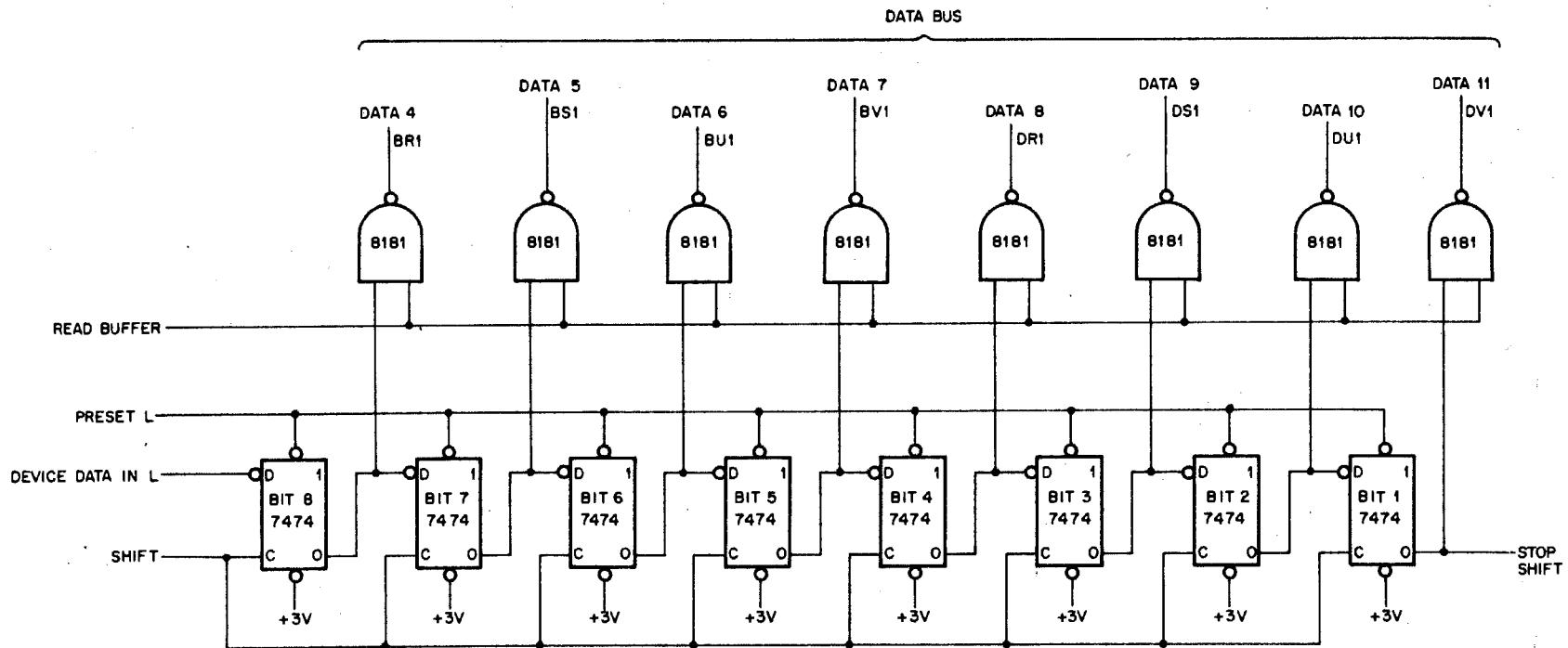


Figure 9-13 Serial to Parallel Input Buffer

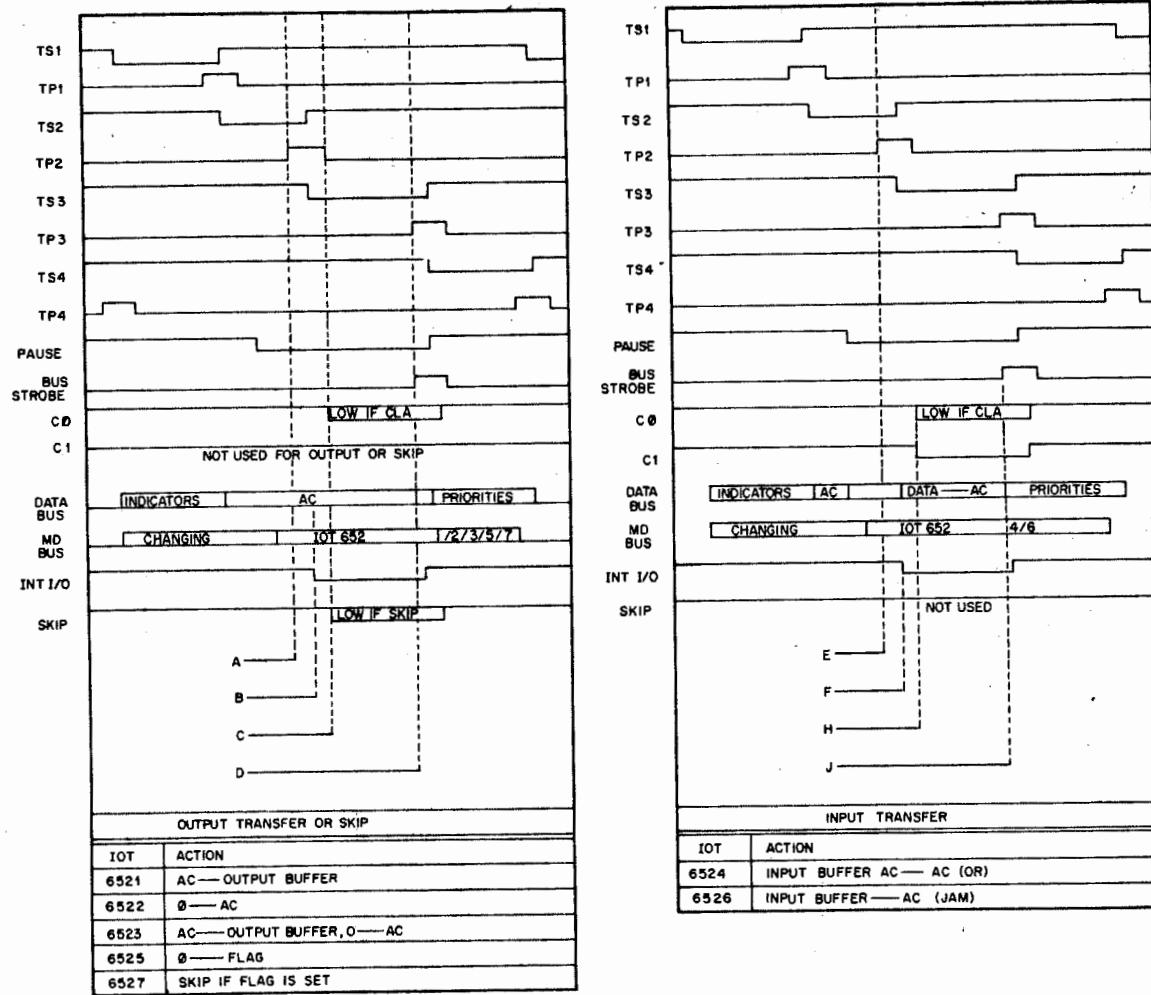
**I/O CONTROL**—The I/O Control includes INT RQST which immediately responds when the flag is set; SKIP which is grounded when IOT 6527 is decoded and the flag is set; C0 and C1 which may be grounded by the operations decoder during various conditions of data transfer and input/output enabling logic which responds to the operations decoder and controls the I/O buffers.

**INPUT/OUTPUT TIMING FOR PROGRAMMED I/O INTERFACES**—A timing diagram corresponding to the Programmed I/O interface example is illustrated in figure 9-14. An explanation of the time periods from A to J is given in the following.

PERIOD	TIME	FUNCTION
A—D & E—J	350ns	Time required to perform the transfer (PAUSE)
A—B & E—F	$\leq 70\text{ns}$	Time required to decode the device selection and assert INTERNAL I/O.
A—C & E—H	$\leq 100\text{ns}$	Time required to decode the IOT and assert the necessary "C" lines or SKIP and supply data if needed.
D & J		The time when the transfer takes place. Note that the DATA BUS will change at this time. This is the reason that edge triggering must be used.

**EXTENDED I/O**—Only if the data input time is a problem should the extended I/O functions be employed. The two control signals that allow extended I/O are NOT LAST XFER and BUS STROBE. When NOT LAST XFER is grounded, the processor timing is stalled at TP3. BUS STROBE, which is normally asserted by the processor, must be grounded when the data is ready to be loaded into one of the major registers. When BUS STROBE is asserted and NOT LAST XFER is not asserted, the processor timing resumes. The net result is the extension of Time State 3.

**Figure 9-14 Timing for Sample Programmed I/O Interface Control**



## SECTION 4 DESIGNING A BASIC DATA BREAK INTERFACE

### GENERAL

The data break control (refer to figure 9-15) consists of logic mounted on an OMNIBUS compatible QUAD type module. The communications link to the processor is via the OMNIBUS into which the module plugs. The communication to the controlled peripheral is via a connector (mounted on the data break interface module) and corresponding cable to the peripheral.

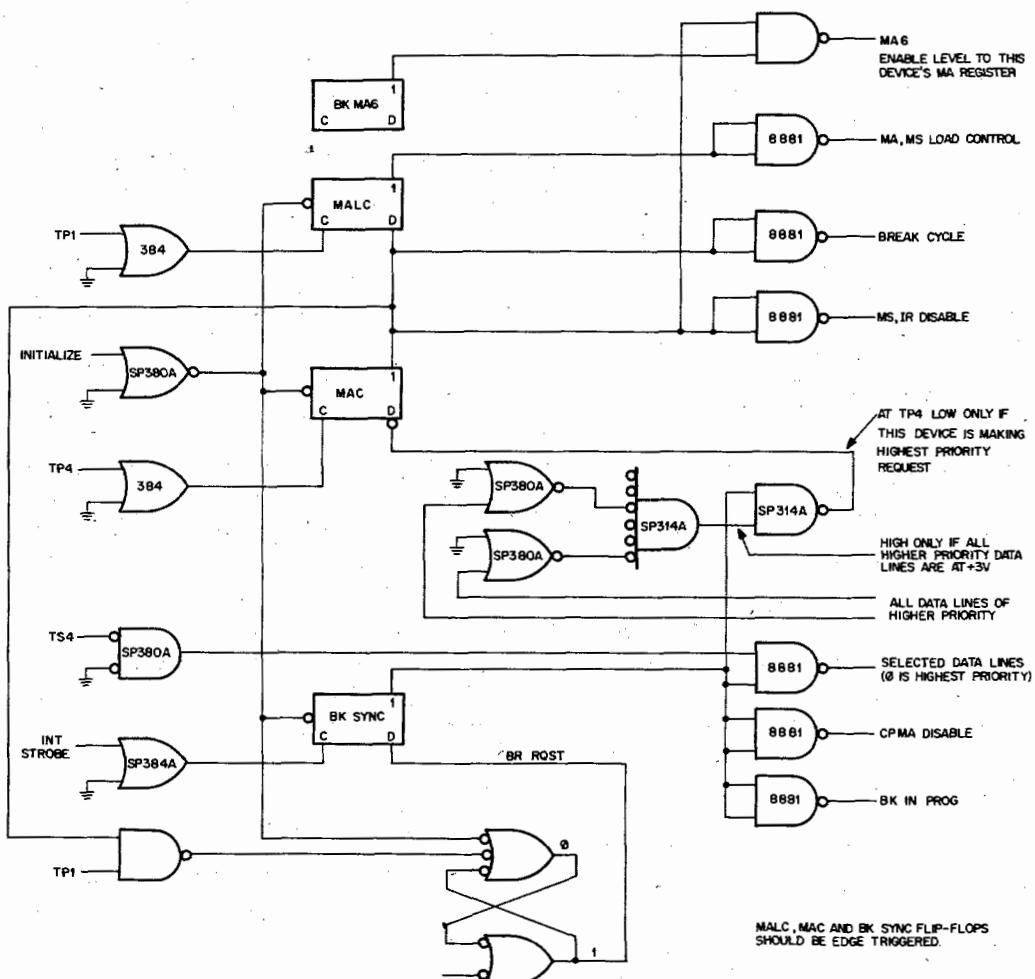


Figure 9-15 Basic Data Break Control Logic

The data break control breaks into the processor sequence of events at the end of a processor cycle whenever a data break peripheral has data to be transferred into or out of the processor. When a new break request is received by the data break control, the data break control waits for INT STROBE and then initiates disabling of the CPMA register. It waits for MY PRIORITY to be established and then disables the Major State and Instruction registers. This places the processor in the DMA state by disqualifying the output gates and in the same manner inhibits any outputs of the Instruction Register.

**BREAK ADDRESS**—The data break control contains a 15-bit Break Memory Address (BKMA) Register (12 for 4K and 3 for extended memory). When applied to the MA lines, the BKMA can directly address any memory location in any one of 8 memory fields. If the device is a 3-cycle break device, the control must be concerned with 3 addresses.

**DATA PATHS**—For all data break output transfers, the data path is via the MD lines. For all input transfers, the data path is via the DATA BUS.

**STATUS REGISTERS**—The status registers of a data break device may be read into the AC register by means of an IOT instruction. To accomplish this, a separate interface control, meeting the requirements of a programmed I/O operation is required.

**BREAK PRIORITIES**—Twelve break priorities are available via the DATA BUS during TS4. The highest priority is DATA 0. A break priority decoding network in each data break device checks all higher-order bits to make sure they are all at +3v and grounds the DATA line of its priority to inhibit any lower order devices. The device doing the decoding executes its break cycle. PRIORITY MUST BE TESTED PRIOR TO EVERY DMA CYCLE.

**TRANSFER DIRECTION AND LOADING LOGIC**—A method of controlling the type of transfer (input, output, or add to memory) must be provided on the data break control interface. To transfer data into memory via the DATA BUS, the device data must be applied to the memory buffers. This is accomplished by leaving signals BREAK DATA CONT H and MD DIR H so that the device data can be applied to memory. When it is necessary to add the device data to memory data, MD DIR is left high, and the BREAK DATA CONT line is grounded. For output transfers, MD DIR must be grounded. The MB register is automatically loaded every TP2. A summary of the transfer types and the signals required by the data break control are summarized in table 9-7.

Table 9-7 Data Break Control Signals

TYPE OF XFER	MD DIR	BREAK DATA CONT	INFO ON DATA BUS
DEVICE→MEMORY	H	H	DEVICE INFO
MEMORY→DEVICE*	L*	X	X
	H	L	0
MEMORY PLUS	H	L	DEVICE INFO
DEVICE→MEMORY			

\* PREFERRED METHOD.

**DATA BREAK INTERNAL LOGIC AND TIMING**—Refer to the example provided in this section for the internal logic and break timing.

### BASIC ONE-CYCLE DATA BREAK INTERFACE

The basic one-cycle break interface required to transfer data consists of a Break Memory Address Register (BKMA) to address memory independently of the processor; a Break Priority Network to assure the activation of the device with the highest priority; Input/Output buffers and Break Control Logic. A sample data break interface is illustrated in figure 9-16. The data break sequence of events are described in terms of the primary data break control signals and the processor timing is given in table 9-8

Table 9-8 One-cycle Data Break Sequence of Events

DATA BREAK EVENT	PROCESSOR TIMING	DESCRIPTION
BREAK REQUEST	Any time. Sampled by the leading edge of INT STROBE	Signal BREAK REQUEST is developed by the device any time a input or output transfer is to be made. It is loaded into a New Break (NBR) flip-flop by INTERRUPT STROBE, sets the flip-flop, and causes the start of a series of events leading to data break transfers.
ADD TO MEMORY	Any time. Must be asserted not later than TP1.	Signal ADD TO MEMORY is generated at the same time as BREAK REQUEST whenever data is to be transferred into memory. It is loaded into the ADM flip-flop by TP1.
DATA IN	TP3 Any time. Must be asserted not later than TP1.	Signal DATA IN is enabled only when the data transfer is to memory and at the same time as BREAK REQUEST. It is loaded into a flip-flop by TP1.
INT STROBE L		The following signals are generated as the result of INT STROBE loading the NBR: a) BK IN PROG L (IF BREAK REQUEST) b) CPMA DIS L (IF BREAK REQUEST) c) DEVICE PRIORITY L (IF BREAK REQUEST)

**Table 9-8 One-cycle Data Break Sequence of Events (Cont.)**

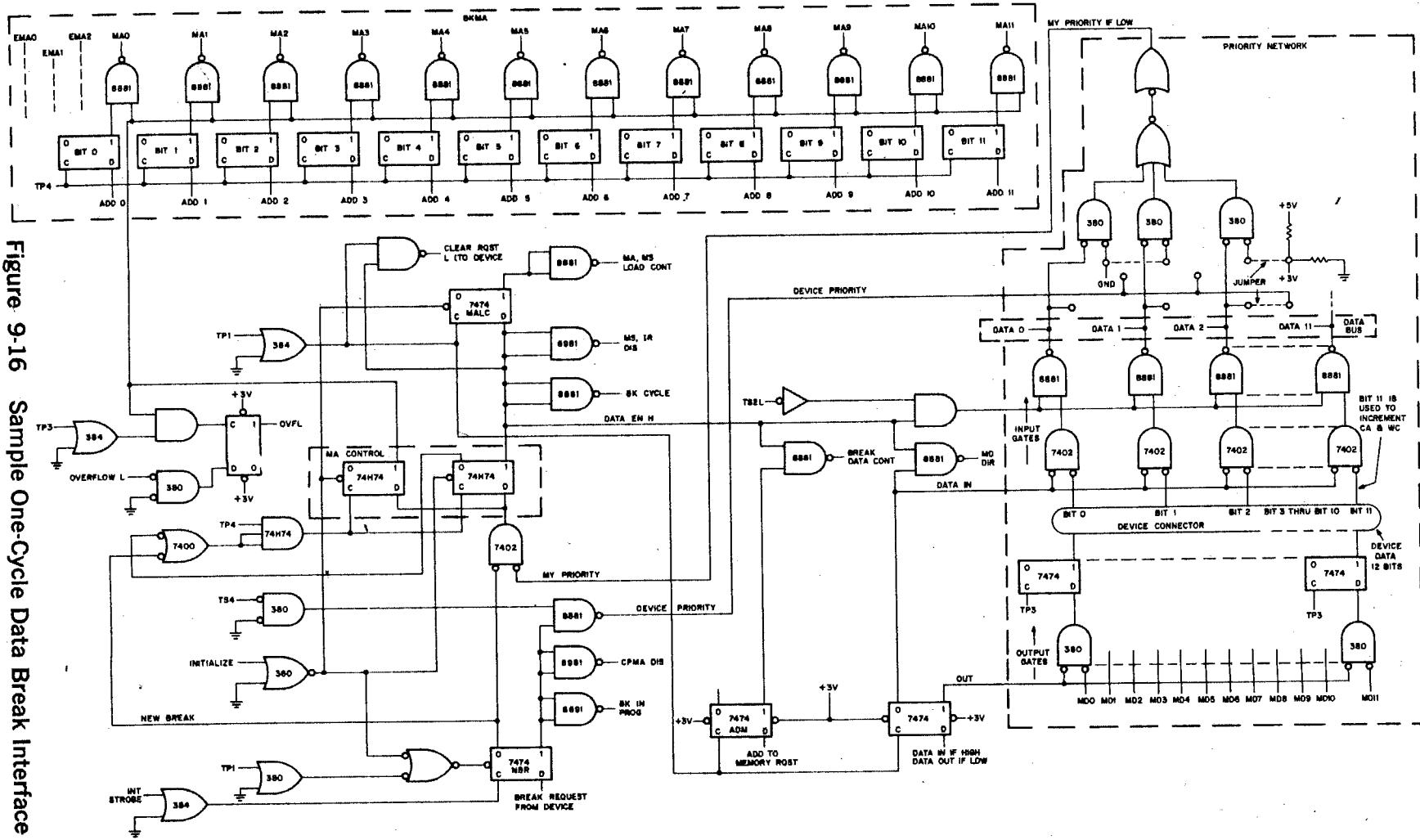
DATA BREAK EVENT	PROCESSOR TIMING	DESCRIPTION
BREAK PRIORITY	TS4	Since each device priority was applied to the DATA BUS at TP3, all priorities are tested during TS4. With the sample data break interface having a 3rd priority, signal MY PRIORITY is developed if DATA 0 and DATA 1 are high. The condition of MY PRIORITY L and NBR (0) L will cause the MA CONTROL flip-flop to set at TP4.
BREAK ADDRESS	TP4	The break address is supplied by the data break device. The contents of the Address lines are loaded into the BKMA by TP4 and the 1 output of the MA CONT is used to gate the Break Address onto the MA lines.
PROCESSOR DMA STATE	TP4	The designer should watch the propagation delays of circuits so that not more than 50ns elapses between the start of TP4 and the arrival of MAC(1) to the BKMA output gates. When the MA CONT flip-flop is set, signal MS, IR DIS is grounded. This disconnects the outputs of the processor's Major State and Instruction registers and thereby causes the processor to enter into the DMA state. Signal BK CYCLE is also grounded. If the transfer direction is from memory to the device, MD DIR is grounded at TP1. If the transfer direction is from the device to memory, BREAK DATA CONT is grounded at TP1.

**Table 9-8 One-cycle Data Break Sequence of Events (cont)**

<b>DATA BREAK EVENT</b>	<b>PROCESSOR TIMING</b>	<b>DESCRIPTION</b>
INHIBIT MS & MA register loading	TP1	When MA CONT is set, the MALC is loaded at TP1. This grounds the MA, MS LOAD CONT line which prevents the MS and the MA registers from being loaded. The processor is now conditioned so that data break transfers will in no way affect the previous or the next processor instruction. At TP1 the ADM and/or OUT flip-flops are loaded to control the type of data transfer. The break request may be cleared by TP1. This allows the MA control flip-flop to be set at TP4.
INPUT TRANSFER	TS2	Device data is gated in by DATA IN (0) L and applied to the DATA BUS by DATA EN H and TS2 L.
OUTPUT TRANSFER	TP3	Memory data is gated into the data break interface when a DATA IN L signal is present and loaded into the input buffer by TP3.
NEXT WORD	TP3	At TP3 of the Data Break Cycle, signal INT STROBE L is again generated in the processor. If signal BREAK REQUEST is asserted at this time indicating that another data word is to be transferred, the break priorities will again be tested during TS4 and a new break address will be applied to the MA lines at TP4. Otherwise, those signals that disabled the processor during the last break cycle will be negated and the processor continues with the current instruction.

**Figure 9-16** Sample One-Cycle Data Break Interface

946



**Timing for Sample Data Break Interface**—The data break control timing with respect to the processor timing is illustrated in figure 9-17. The diagram illustrates 2 complete processor cycles and a portion of a third cycle. For the first cycle, only that portion beginning with TP3 is of interest. This is the time required by the data break device to assert the control signals necessary to halt the processor, address memory, and be ready for input or output transfer. If there are to be no additional transfers, the break control signals are negated at the end of the break cycle as shown on the diagram. Otherwise, the break control signals will continue into the next cycle.

### THREE-CYCLE DATA BREAKS

All of the previous information has dealt with one-cycle breaks. Three-cycle breaks consist of three break cycles in succession, the first two of which are used to control word count and current address. See Chapter 6 for a detailed discussion of three-cycle data break theory.

The data break hardware for a three-cycle break is more complicated than that for a one-cycle data break. In addition to the normal data break equipment, the three-cycle control requires internal major state control to accommodate word count, current address and data transfer cycles. A means for loading the BKMA register from the MD lines during the current address cycle must also be provided. Priority must be checked three times: once each before WC, CA and B cycles in order to allow higher priority devices access to memory in the minimum amount of time.

The hardware implementation for a three-cycle break can be accomplished using the previous one-cycle break example and the flow diagram for a three-cycle break illustrated in figure 9-18 as a guide.

948

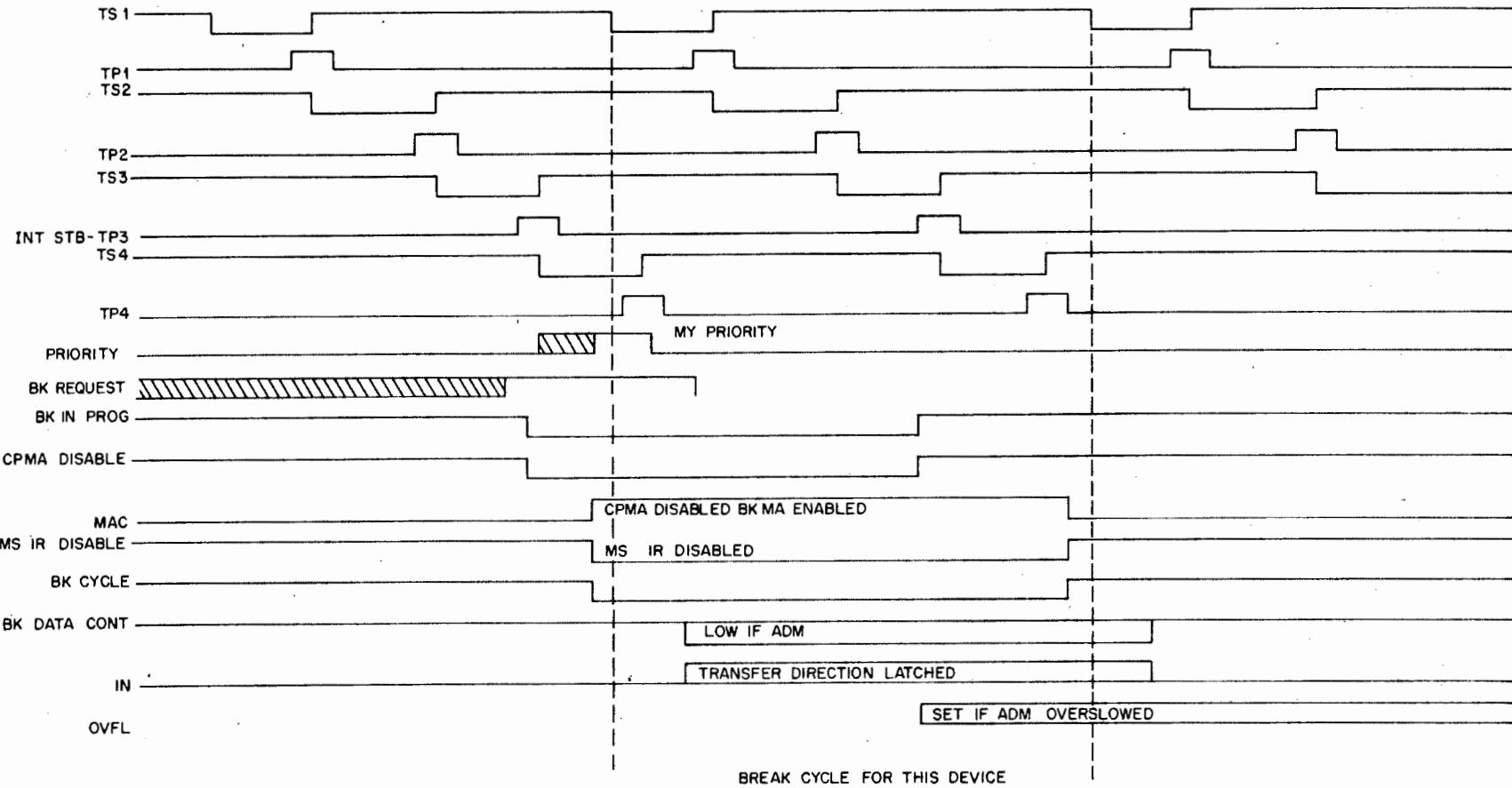


Figure 9-17 Data Break Control Timing Diagram

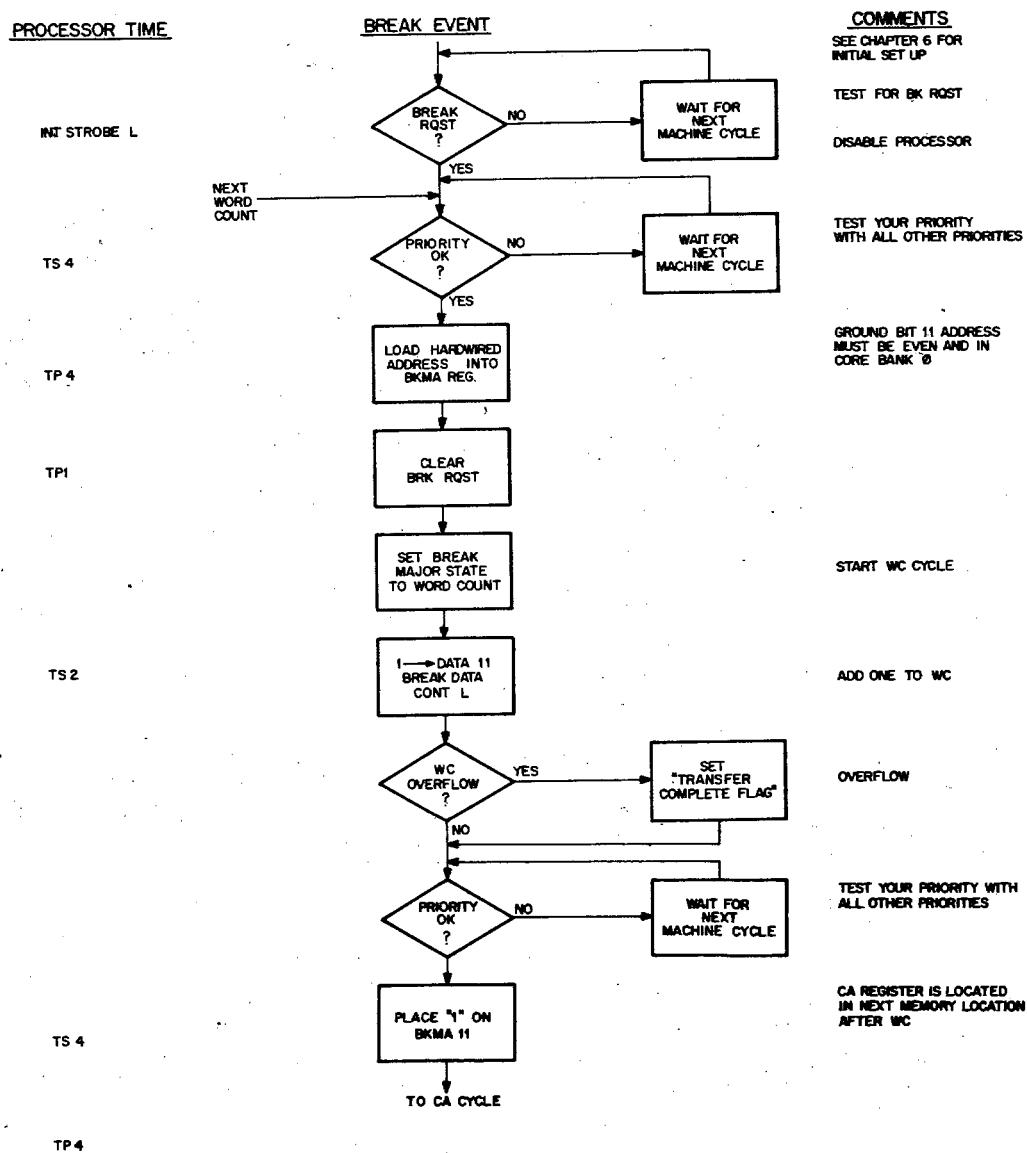


Figure 9-18 3-Cycle Data Break Implementation Flow Diagram

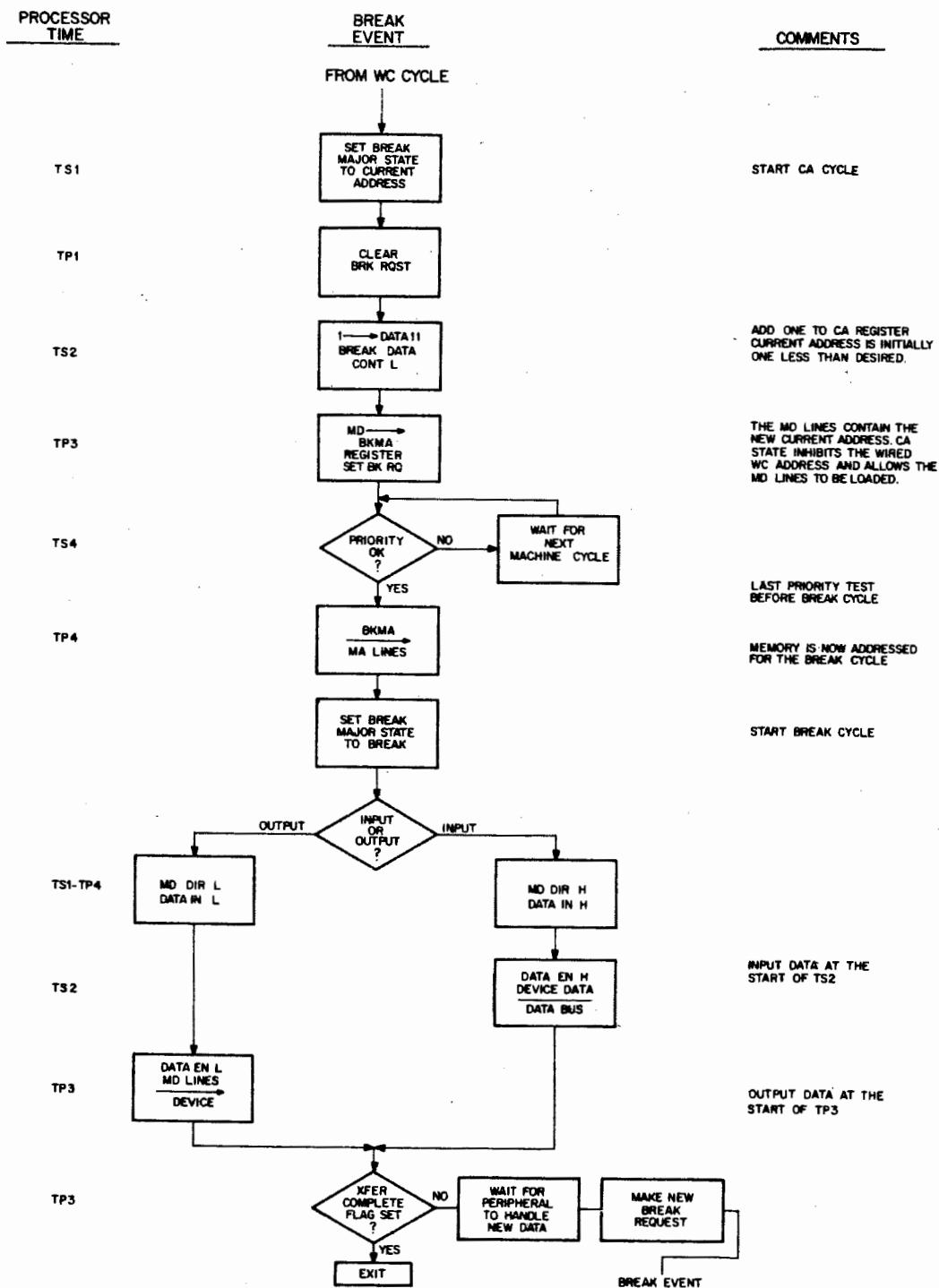


Figure 9-18 3-Cycle Data Break Implementation Flow Diagram (continued)

## **DESIGN CHECK LIST FOR SINGLE CYCLE DATA BREAK INTERFACE**

The following information summarizes the important design considerations of a single cycle Data Break Interface.

### **PDP-8/E CONTROL LINES**

- a. To ground BK IN PROG and CPMA DIS, use the leading edge of INT STROBE H.
- b. Ground the bit on the DATA BUS corresponding to its priority while examining all other priority bits only during the TS4 Time Period.
- c. At the leading edge of TP4, provided all higher-order PRIORITY bits on the DATA BUS are high:
  - 1. Place the break address on the MA lines within 50ns after the leading edge of TP4,
  - 2. Ground BREAK CYCLE,
  - 3. Ground MS, IR DIS,
- d. At the leading edge of TP1; if all conditions within C above were met: 
  - 1. Ground MA, MS LOAD CONTROL,
  - 2. Set the ADM flip-flop and ground BK DATA CONT and/or MD DIR to establish the data-transfer path,

#### **NOTE**

Control lines are generally negated in the same order in which they were asserted.

### **DATA TRANSFER PATHS**

- a. For data input transfers or add to memory:  
Place input or modifying data on the 12-bit DATA BUS for the duration of TS2.
- b. For Data output transfers:  
Use a Time Pulse to gate data from the output buffer to the device. Data is available on the MD lines at TP2, TP3, or TP4 time.
- c. Overflow:  
If a modification of memory was made, the OVERFLOW line will be low during TS3. Sample this line with TP3 if you wish.

## SECTION 5 GENERAL DESIGN & CONSTRUCTION GUIDELINES

### INTERFACE DESIGN OPTIONS.

Basically, the user has two options in designing his interface. One option is to build an interface module to the external bus, and the other is to build an interface module to the OMNIBUS.

External bus interfacing allows the designer to deal with the wire-wrap system, which by definition is easier for him to alter. The user then does not have to be concerned with the rigid pin assignment imposed upon the OMNIBUS. Chapter 10 explains how to do such interfacing.

Interfacing to the OMNIBUS is simple and direct, providing that the designer conforms to the bus pin assignment. He has several options in selecting the type of module that he wishes to place on the bus. He may, for instance, construct a wire-wrap assembly and place it at the far end of the OMNIBUS. This assembly can be connected directly to the OMNIBUS. There is enough room to place a fairly complicated controller with wire-wrap pins. The restriction is, of course, that the pin assignment must conform to the OMNIBUS. The user may purchase a single quad board containing wire-wrap pins and IC sockets from DEC. Upon this board he may easily construct any type of interface to his specifications.

Another method that provides the highest density of components is the use of an etched board. The user can build blank boards and purchase from DEC most of the necessary IC's, connectors and cables. The advantages of this approach are fully realized when large numbers of duplicate interfaces are to be made. The cost of building an etched board is, of all methods, the lowest—provided that enough interfaces are to be constructed so the designer can recoup his rather high initial engineering costs.

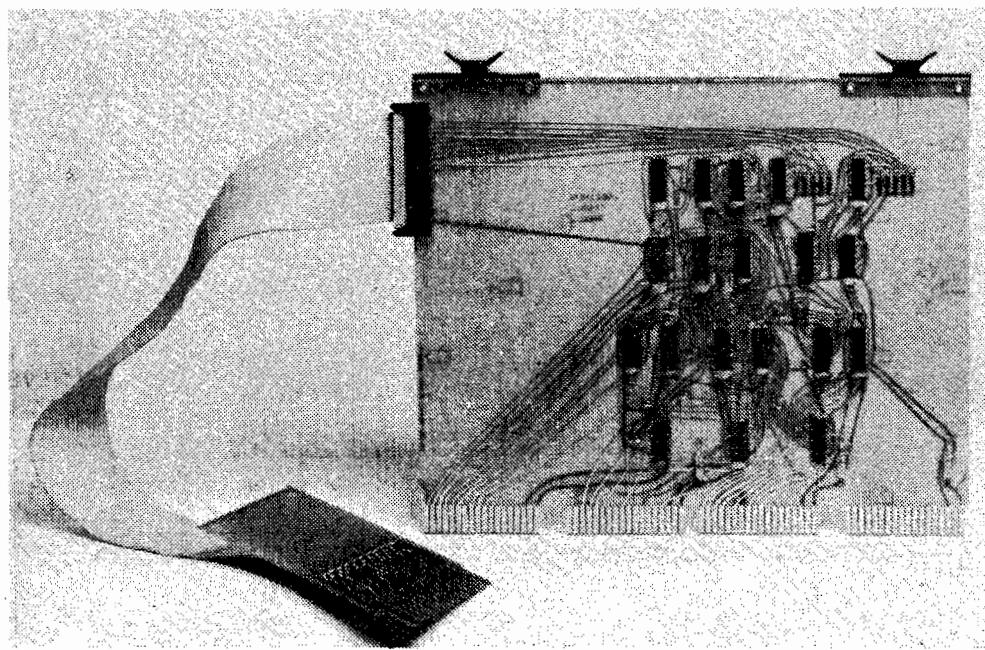
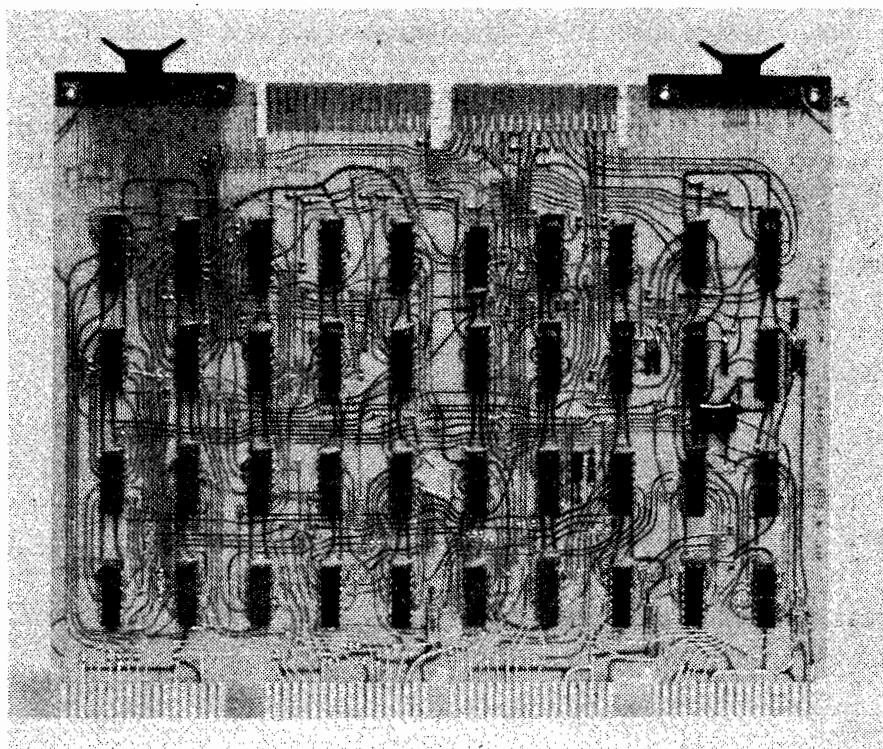
### Board Layout

When connecting the +5V supply, the designer should try to split up the runs into three separate parts, and connect each run to a different pin. Then, if there is a short somewhere from +5V to ground, it is three times as easy to find. A good rule to follow is to limit the number of IC's to be mounted on any one board to 50. If the designer has a requirement for more than 50 IC's, he should consider using a second board. A typical layout of components is shown in Figure 9-19.

Uninsulated components should not project more than  $\frac{3}{8}$  in. above the board, insulated components should not project more than  $\frac{1}{32}$  in. above the board. The grounding scheme should be carefully planned. Pins C, F, N, and T (except AC1) provide the ground lines needed to operate the board. It is good practice to use as many ground lines as possible; connect all ground pins together, and make runs as short as possible to the ground pins of the ICs. Poor grounding can cause occasional annoying malfunctions.

### Etched Circuit Layout and Construction Rules

The following layout and construction rules should be used as a guide to assure optimum performance of the interface module:



**Figure 9-19 Typical Component Layout**

### **General Cable Rules and Suggestions**

Cabling is an important consideration when designing and constructing an interface control module that plugs into the OMNIBUS. The designer's only concern is that his cabling is adequate between his peripheral and the interface control module that he is designing to plug into the OMNI-BUS. No additional cabling is required. The interface cable connects to a 40-pin connector type H854 shown mounted on the upper left side of the module. The peripheral connector is a 40-pin type connector. A channel "cut-away" along the length of the power supply allows bundling of many cables and allows them to be clamped onto the side of the channel.

### **DEC Supplied Interface Cables**

The user may purchase from DEC the necessary interface cables. Two standard lengths are provided—a six ft. cable (part number BC08J-6) and a ten ft. cable (part number BC08J-10). Each cable contains a 40 pin connector type H856, which connects to the interface module and a 36-pin module type M953 which in turn connects to the users peripheral.

Each cable provides 18 signal lines and 22 ground lines. A ground line follows each signal line with several ground lines on each end of the cable. This arrangement is illustrated in Figure 9-20.

The design is intended to provide an electrical shield between lines, and to provide adequate grounding between the two ends of the cable. The use of ribbon cable is convenient, saving space and eliminating bundle problems. However, round coaxial cabling can be used. The coaxial cable then has to be larger because it consists of several layers of conductor and insulator material. Where the environment is considered hostile (such as may be the case in a factory), coaxial cabling is recommended. However, in most cases, the flat cable will prove adequate.

### **Cabling Rules**

There are cabling precautions that the designer of any interface control module should follow. These include:

- a. Do not run the AC line immediately adjacent to a low level signal.
- b. Do not attempt to drive a line directly with the output of a flip-flop. The flip-flop may be triggered by noise being sent back along the line.
- c. Tie all grounds together at the board and at the far end of the cable.
- d. If there are two or more cables running parallel, there must be an intervening electrostatic shield.
- e. Use as low an impedance as possible, but not lower than 100 ohms on the lines, and terminate the lines at the far end to eliminate ring; or drive them with a higher impedance and wait for them to settle down.

Module M953 Pins		Module H856 Pins	
Signal	Ground	Signal	Ground
B1	A1	D	A, B, C, E
D1	C1	J	P, S
E1		N	
H1	F1	T	U
J1		X	
L1	K1	BB	Y, AA
M1		FF	
P1	N1	LL	CC
S1	R1	RR	HH, KK, MM
*U1	T1		
*V1			
*A2			
*B2			
D2	C2	F	SS
E2		L	
H2	F2	R	H, K
K2	J2	V	M
M2	L2	Z	W
P2	N2	DD	EE
S2	R2	JJ	PP
T2		NN	
V2	U2	TT	UU, VV
* not used			

Figure 9-20 Interface Cable Pin Assignment

If the user requires a more complicated interface controller that requires two boards, a connector type H851 is used to interconnect the signals from one board to the next. This is shown in Figure 9-2. The connector receives 36 etched finger type pins from both modules and slides onto both modules. Pin A1 connects to pin A1, etc.

### **INTERFACE TIMING CRITERIA**

In nearly all instances, the user need not concern himself with the details of timing. This section is included to assist the person with an exceptional case.

There are basically three types of timing with which the user may be concerned. First, there is the data exchange time between the computer itself and the peripheral (a function of the I/O structure of the machine). Secondly, there is data break timing, which is a function of the break priority system and the data break peripherals. The third consideration is Interrupt timing.

#### **General Timing Rules**

General timing rules are as follows:

- a. If maximum machine speed is necessary, do not use the positive I/O Bus interface, type KA8-E. Instead, design all peripherals so that they plug directly into the OMNIBUS.
- b. When the above is not feasible, do not microcode IOTs. For example, replace the KRB instruction with its equivalent KCC and KRS instruction. The result is a slightly longer time for the overall IOT; however, the processor is stopped for two shorter periods of time, rather than one long period of time. Hence, the data break and interrupt systems have access to the processor and memory sooner.

#### **Interrupt Timing**

Interrupt timing is concerned with the interval from the time that a flag is seen until the time it can be serviced. Although this time is fairly easily calculated, the exact details depend upon the number of flags the processor checks before it finds a flag that is set.

#### **Timing Example:**

An interrupt request is asserted by one of the control modules. The processor must finish the current instruction upon which it is operating, possibly a TAD indirect through an auto index register. TAD indirect

through an auto index register takes 1.2 microseconds plus 2 times 1.4 microseconds, or 4.0 microseconds total. In addition, the processor might not have seen the flag as much as 300 ns before that time, so it is a maximum of 4.3 microseconds from the time that the flag got set until the time the processor can start executing the JMS to location zero. This assumes no EAE option and that no device is using the break facility at that time. Thus, up to this time, it has taken the processor 4.3 microseconds to recognize that there is a flag to be serviced. This example assumes that the KM8-E Memory Extension control is installed.

### TIMING EXAMPLE

TAD I	4.0 $\mu$ s	Through an Auto Index register
(JMS)	0.3 $\mu$ s	Worst case time for processor to see flag
DCA AC	1.4 $\mu$ s	Jump to location 0, store PC
GTF	2.6 $\mu$ s	Store AC and Link
	1.2 $\mu$ s	Pick up flags in memory extension control unit
DCA FLAGS	2.6 $\mu$ s	Store flags
MQA	1.2 $\mu$ s	Load AC with MQ
DCA MQ	2.6 $\mu$ s	Store MQ
JMP I .+1	2.2 $\mu$ s	Jump somewhere to Interrupt flag scan
IOT FLAG SKIP	1.2 $\mu$ s	Skip on flag—Start of Flag Scan Routine
IOT	1.2 $\mu$ s	Perform I/O Transfer
Total	20.5 $\mu$ s	Total time required to get to IOT and perform instruction.
DCA I 10	4.0 $\mu$ s	DCA indirect through some auto index register
CLR FLAG	1.2 $\mu$ s	
ISZ	2.6 $\mu$ s	Bookkeeping—To see if last transfer
JMP	1.2 $\mu$ s	Dismiss
TAD MQ	2.6 $\mu$ s	Restore MQ
MQL	1.2 $\mu$ s	Transfer the contents of AC into MQ
TAD FLAGS	2.6 $\mu$ s	Check status of flags
RTF	1.2 $\mu$ s	Restore flags
TAD AC	2.6 $\mu$ s	Restore AC
JMP I O	2.4 $\mu$ s	Go back to location 0
Total	21.6 $\mu$ s	Total time required to get back to main program again

Thus  $20.5 \mu$ s +  $21.6 \mu$ s =  $42.1 \mu$ s total for this example.

The interrupt timing requirements are a function of the amount of coding that it takes to determine what has to be done and how much time is available to do it. There are two times that must always be considered. The first is the length of time from the time of the interrupt request until the source of the interrupt request has been recognized and the data retrieved. The second time is the length of time it takes to finish processing the data, including all of the housekeeping routines and the time it takes to restore the major registers to their original state.

A special instruction, SRQ (octal 6003), allows the programmer to test for possible interrupts before actually restoring the machine. One may save considerable time entering and exiting the interrupt program by merely returning to the flag scan routine, if the SRQ instruction indicates the presence of a second interrupt.

The program interrupt system is satisfactory for data rates less than 10 KHz (one word every 100 microseconds). Above this rate, the user should examine each individual case for its merits and decide whether or not he should use the data break facility or possibly tie the machine up with high data rates.

### **Timing Requirements for Data Break Facilities**

The important timing consideration in data break as in program interrupt, is whether sufficient time is available from the time the flag (in this case, the break request) is set to the time all the data is moved in or out of memory. The first item in question is the period of time before operation actually starts on the break request. The break request timing for the PDP-8/E has improved considerably in that a break request can be honored between major states of an instruction, whereas in the older model machines, the break system had to wait until an instruction was completely processed. The break system is synchronized 300 nanoseconds before the end of every memory cycle. At the same time the processor tests for the possibility of interrupt, it tests for the possibility of break. Unlike the case of an interrupt, the break system need not wait until the processor is ready to go into a FETCH. If the processor is programmed to do only machine instructions (assuming no EAE or external I/O), it would take the processor no more than 1.7 microseconds to begin servicing the break request. For the external bus, this time would be 4.9 microseconds, maximum. The choice of one-cycle or three-cycle breaks is an important timing consideration. Basically, the internal operation of the PDP-8/E is such that it readily adapts to one cycle break. Users who are planning on constructing their own break device should think in terms of one cycle break.

**Timing and Break Priorities**—Timing problems result when the break priority system has not been carefully planned. Suppose, for example, the highest speed break device is a complex parallel disk that serves up a word once every 5 microseconds. Suppose two break devices request a break simultaneously (e.g., the disk and a slower device such as the DECTape that has 50 microseconds for which the word is available). If the priorities of these two devices are not correct (i.e., DECTape assigned first priority), timing problems are inevitable. If the DECTape and disk simultaneously request a break, both devices must wait 1.7 microseconds for the current memory cycle to finish. The DECTape then makes a 3-cycle break request since it (incorrectly) has the highest priority.

Therefore, the waiting time of the disk is 3 times 1.4 microseconds or 4.2 microseconds plus 1.7 microseconds (a total of 5.9 microseconds), to service the DECTape. The result is that the DECTape was serviced and the disc has lost its data because 5 microseconds have expired. This is an obvious situation where the disk must have a higher priority. Other situations may not be obvious without examining the timing requirements before assigning a priority to each device. As a general rule, the user should set up his priority based upon the device that has data available for the shortest amount of time.

### **GENERAL PROPAGATION DELAY GUIDELINES**

When designing an interface module, the designer should consider the individual propagation delays of such logic elements as NAND gates, flip-flops, J-K flip-flops, one shot delays, etc. He should add each delay in a logic chain to determine the overall delay of his module.

## 2 Input NAND Gate Delay

Typical characteristics of a NAND gate used with the PDP-8/E logic are illustrated in Figure 9-21. Where high fan out is required, a SN7440 type gate is preferred.

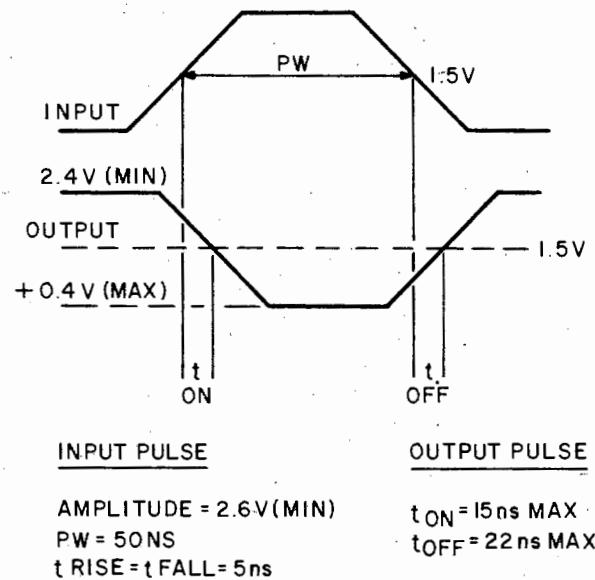


Figure 9-21 2 Input NAND Gate Typical Characteristics Example

## Flip-Flop Propagation Delays

Typical D-type flip-flops trigger on the leading or rising edge of a positive clock pulse; the propagation delay is measured from the threshold point of this edge. The set-up time of the flop is also measured from this threshold point. Data on the input must be settled at least 20 nano-seconds prior to the clock transition.

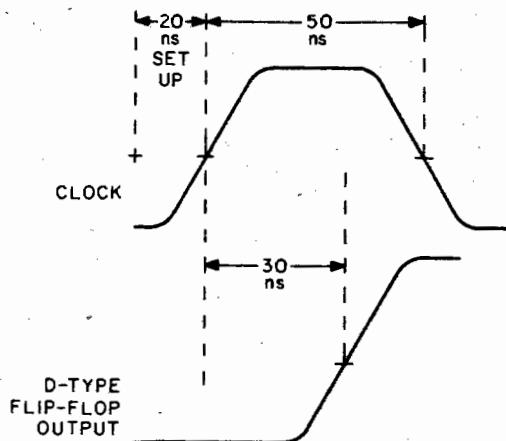


Figure 9-22 Typical D-Type Flip-Flop Timing Example

### J-K flip-flops

J-K type flip-flops are, in effect, trailing edge triggering devices as explained previously. The only restriction on the J and K inputs is that they must be settled by the time that the rising edge occurs. Timing is shown in Figure 9-23.

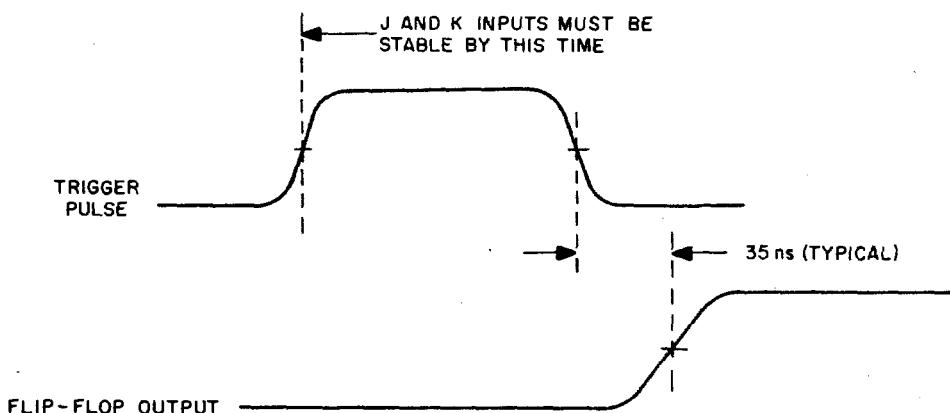


Figure 9-23. J-K Flip-Flop Timing Example

When using the dc Set or Reset inputs of either flip-flop type, propagation delays are referenced to the falling edge of the pulse. This is due to the inverted sense of these inputs. When resetting ripple type counters (where the output of one flip-flop is used as the trigger input to the next stage), the Reset pulse must be longer than the maximum propagation delay of a single stage. This ensures that a slow flip-flop does not introduce a false transition, which could ripple through and result in an erroneous count.

**One-Shot Delay**—Calibrated time delays of adjustable duration are generated by the Delay Multivibrator such as the M302. When triggered by a level change from a logical one to a logical zero, this module produces a positive output pulse that is adjustable in duration from 50 to 750 ns with no added capacitance. Delays up to 7.5 milliseconds are possible without external capacitance. Basic timing and the logic symbol are shown in Figure 9-24. The 100 picofarad internal capacitance produces a recovery time of 30 ns. Recovery time with additional capacitance can be calculated using the formula:

$$t(r) \text{ Nanoseconds} = 100.3 C \text{ Total (Picofarands)}$$

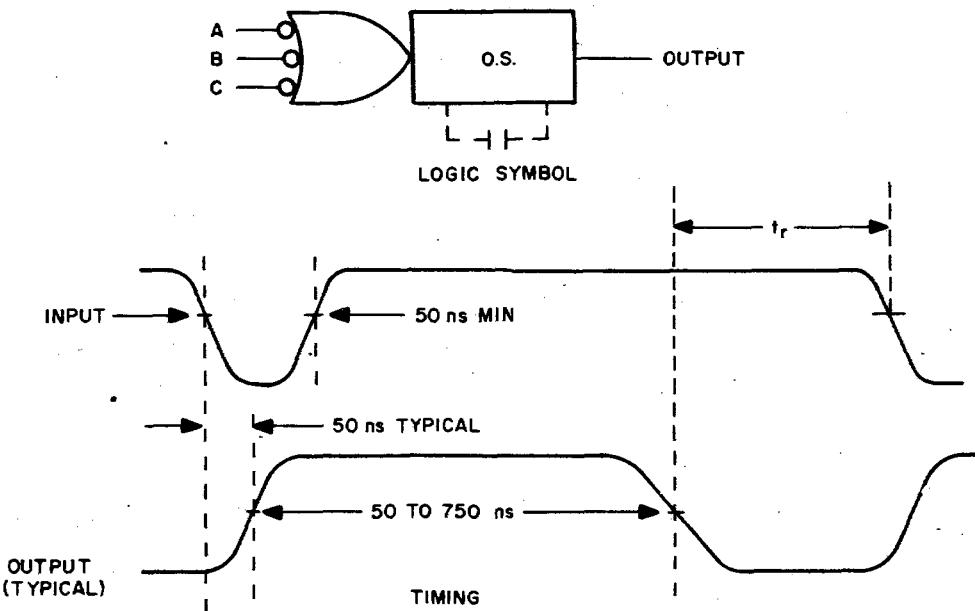


Figure 9-24 One-Shot Delay Timing and Logic Symbol Example

#### Maximum Operating Frequency

Once the designer has determined the individual propagation delays in each logic element, he must then add these delays corresponding to a simple logic chain. He then compares the results with the system frequency to assure that his logic circuit can meet the requirements imposed by the system frequency. Figure 9-25 illustrates a situation in which various logic components in a given chain are examined and all delays are added. The following assumptions are made:

- A standard clock pulse width of 50 nanoseconds is assumed. This period is measured from the threshold point of the leading edge to the threshold point of the trailing edge.
- One flip-flop propagation delay of 35 nanoseconds from the trailing edge of the clock pulse to the threshold point of the final state of the flip-flop is allowed.
- Two gate-pair delays of 30 ns each are assumed. (A gate-pair consists of two inverting gates in series.) Two gate-pair delays are usually required to perform a significant logic function with a minimum of parallel operations. The two gate-pair delays total 60 ns.

The time necessary to perform these operations before the next occurrence of the clock pulse is the sum of the delays;  $50 + 35 + 60$ , or 145 nanoseconds. Allowing 20 ns for variations within the system, the resulting period is 165 ns, corresponding to a 6 MHz clock rate.

Note that the D-type flip-flop triggers on the leading edge of the clock pulse and the J-K flip-flop triggers on the trailing edge. When calculating system timing using flip-flops, remember that the flip-flop inputs must be settled at least 20 nanoseconds prior to the occurrence of the clock pulse.

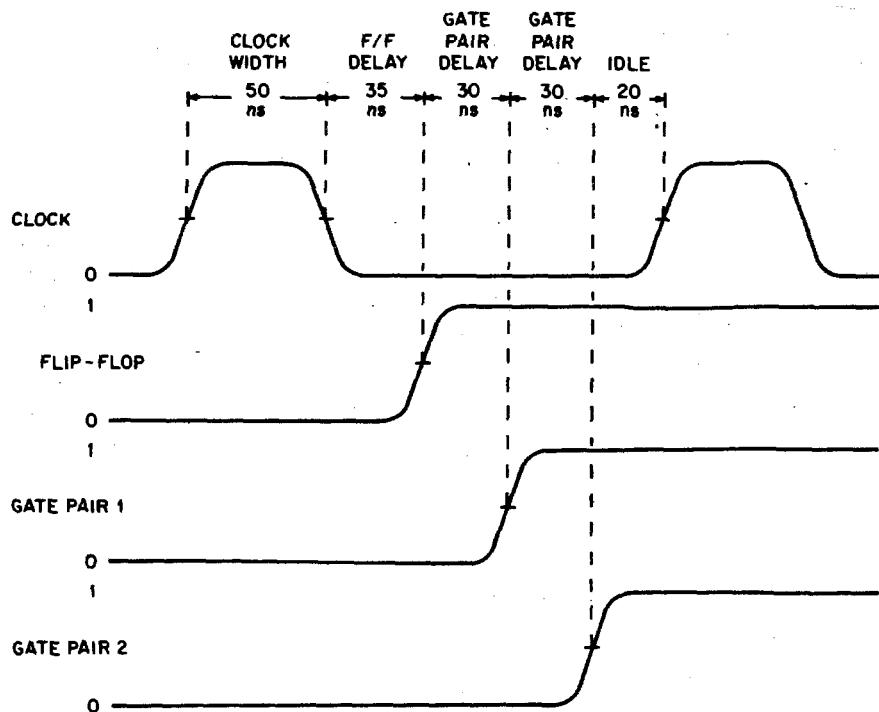


Figure 9-25 Delays Determining Total Operating Frequency Example

The preparation of a timing diagram that considers delays introduced by all logic elements aids the designer in achieving predictable system performance. Don't forget that minimum (including zero) delays are possible, and that no good design should rely on a finite gate delay. Using a similar approach, the designer must ensure that his selection and operation decoding scheme works within the allotted time, and that data can be made available in time for TP3 to either strobe the data in or strobe the data out, depending upon the type of transfer. I/O PAUSE is used to gate the device and operation decoders and TP3 strobes the data either into the OMNIBUS or onto the data lines to the peripheral. A total of 280 nanoseconds is allowed to bring the contents of either the input or output data line into the interface controller. All input signals to the processor must be established at the processor at least 130 nanoseconds before TP3. This discussion assumes either an I/O transfer or a programmed interrupt. With data break devices, the preceding information does not apply.

#### LOADING RULES

Almost all signal lines on the OMNIBUS are driven positive by a load resistor (refer to Figure 9-6) and pulled to ground by open collector ICs. The designer should use a gate such as the DEC380, DEC314, DEC348 ICs or their equivalent as an input device. These gates have a fairly high input impedance and, therefore, do not load down the bus. Wherever possible, one of the inputs of a gate should be held positive by I/O PAUSE or some other logic signal in order to further reduce the input load. Conversely, to drive output lines (such as the bus lines) the user should consider DEC8881 or its equivalent (paying careful attention to the leakage current).

### **Device Selection Inputs**

When designing the device selection decoder, a NAND gate such as the DEC380, which is a two-input gate, should be strobed by I/O PAUSE. This helps to remove the load from the MD lines.

### **Skip and Interrupt Request Lines**

As a general rule, the DEC8881 gate outputs should be limited to one per device code for skip line and one per device code for the Interrupt Request Line. A potential problem exists when too many gates are tied into these lines.

### **Electrical Considerations of Driving a Line**

Most signal lines on the OMNIBUS are tied through a load register to +5V. Users who want to look at any one line must do so with the DEC380 gate. Users desiring to drive any bus line to ground must do so with the output of the DEC8881 gate. The limitation of the amount this gate can carry must be considered. A major factor is the output leakage, as no switch is a perfect open circuit nor a perfect short circuit. This is a fundamental limitation. Tolerance for ground level should be considered up to .4V, as defined by the TTL logic. The high signal must be a minimum of 2.6V; however, 3.0V is recommended. An additional consideration is that the DEC380 requires some current at its input.

### **GROUNDING**

Pins C,F,N, and T on the OMNIBUS are used for ground signals. The user who is making modules that are designed to plug into the OMNIBUS should utilize all of the ground pins and tie all of the ground pins together. He should make the connecting lines as short as possible. The user should also attempt to keep the leads from the ground pins to the ground terminals on the ICs as short as possible. The shorter the ground runs from the integrated circuits on the module to the ground pins and the more duplication there is (parallel paths), the quieter the ground system is within the module. The designer should pay careful attention to the recommendations of the ICs to ensure that good construction practices are followed. (Do not overlook the local bypass capacitors required at an IC.) The designer should use as much as possible a .01 microfarad ceramic disk capacitor across Vcc to ground for every IC used.

### **TESTING TECHNIQUES**

When the interface module has been completely assembled and checked, the designer should perform an initial checkout and then proceed to test his interface in the system. This should be followed by a complete peripheral system integration.

#### **Initial Checkout**

The tester should remove the power source connections prior to performing his initial test. He should then make an electrical test with an ohmmeter from +5V to ground, from -15V to ground, and from +15V to ground. He must ensure that there are no power shorts prior to connecting power. A short can damage the etched circuits.

#### **System Test**

The next test step is to plug the module into the OMNIBUS and generate an IOT to check out the device selection capability, logic levels, opera-

tion of the flag, and capability of the interface controller to receive and transfer data. A combination of 1's and 0's may be placed in the AC and transferred to the interface. With an oscilloscope or voltmeter, the tester can check each of the connector terminals corresponding to each of the data bits to ensure that the right information is being transferred. Another useful test is to generate a count pattern in the AC and observe that bit 11 is moving twice as fast as bit 10, which is moving twice as fast as bit 9, and so on. These waveforms can be examined for each line. This test indicates shorts existing between data lines.

### Final Testing

Before actual operation, the final test includes connecting the peripheral to the interface and transferring data into and out of the peripheral. For example, the 1's and 0's can be checked at the peripheral end for input to the peripheral and checked at the AC register for data transferred from the peripheral to the processor. Whenever possible, the programs used for testing should be collected into a Diagnostic Program for the device. A properly designed diagnostic, since it tests only one peripheral, is a powerful tool for finding system failures.

### PROGRAMMING RULES

The most successful method of programming is to begin a program as simply as possible, test it, and then add to the program until it performs the required job. Before beginning the programming, the programmer should become familiar with the programs that he will be using. Refer to Chapter 4 for a description of the standard programs and refer to Appendix A for a complete list of the PDP-8/E compatible programs. For best results, the programmer should avoid the use of the following device codes:

1. Devide code 0 (reserved for processor)
2. Device code 3
3. Device code 4
4. Avoid all codes in the 20 through 27 series (reserved for the extended memory control)
5. Avoid the Disk and DECTape device code series

Device codes 14-17 have been made available for the programmer's special use.

### DESIGN CHECK LIST

When interfacing to the Omnibus, certain things must be done and others should be done. The following is a check list to summarize the requirements.

#### a. Omnibus Compatibility

1. In looking at the bus, do you use only 380, 384, 314 type IC's?
2. Are DEC8881's or 7438's selected for 25  $\mu$ A leakage used to gate onto the EMA, MA, RUM ADDRESS and MS IR DISABLE lines?
3. Are two DEC8881's in parallel used to gate onto Link Load and Bus Strobe?

- 4. Are N8881, 8235 or 97401's used to gate all other signals onto the bus?
  - 5. Are all flip-flops receiving information from the bus leading edge triggered?
  - 6. When using direct clear or presets, do you gate a level (IOT) against TP3?
  - 7. Never gate TP3 and a data bit into a direct clear or preset.
  - 8. Never gate information onto the MD lines unless you're a memory.
  - 9. Does the sequence of modules on the OMNIBUS conform to the "module priorities" drawing in the print set?
- b. Timing
- 1. Is the path that pulls internal I/O less than 70 nsec. from pause?
  - 2. Are the paths that assert the "C" lines, data and skip, less than 100 nsec. from pause?
  - 3. Are the data, skip and "C" lines asserted by levels (IOT's), not pulses?
  - 4. If using long cycles is LAST XFER asserted at least 100 nsec. before TP3?
  - 5. If using long cycles, check the timing requirements in this chapter.
- c. Loading
- 1. Are MD 3-8 gated against pause?
  - 2. Are MD 9-10 gated against option select? (The decoded device code.)
  - 3. Are the Data Lines loaded only during an IOT?  
One input to the receiver should be option select.  
Only one receiver per device code is allowed per bit.
  - 4. Do you drive any 380, 384, 314 IC's with TTL? If so, do you have the proper pull up?
  - 5. Did you check the loading of long runs, such as Data Enables?
- d. Noise and Interference
- 1. Are all unused direct clear and presets tied off?
  - 2. Are all grounds used and tied together at both the front and back of the board?
  - 3. Are all signals to the OMNIBUS and control flip flops disabled by INITIALIZE? In some cases, such as magnetic recording, PWR OK may be preferred.
  - 4. Is there an .01  $\mu$ f. capacitor across pwr at each IC?
  - 5. Are there 6.8  $\mu$ f. capacitors between each +5V and ground?
  - 6. Does power and ground go to the correct pins on each IC? For instance, 380 power is pin 8, ground pin 1.
  - 7. Check Chapter 9 for lines which should not be used.
- e. For Convenience
- 1. Keep +5V runs separate (three runs). This makes finding shorts easier.
  - 2. Label all jumpers in etch.

## **SECTION 6 PDP-8/E INTERFACE HARDWARE**

The following Interface accessories are available to make interfacing to the OMNIBUS a simple task.

**H9190 M935 Kit**—contains the H9190 assembly with M Series connector blocks for standard M Series modules, power wiring harness, and power bus board. It includes M Series power bussing for all but the four slots in the first column. Also included are two M935 bus connectors. Four mounting spacers allow the H9190 to be easily mounted in the second half of an 8/E chassis.

**H803 Connector Block**—a high density, 8-slot connector block with wire wrap pins. This connector is designed to be used with M Series modules.

**M935 Bus Connector**—used to interconnect 8/E assemblies. The H9190 may be connected to the 8/E OMNIBUS using two M935's.

**H9190 Mounting Panel**—contains M Series connector blocks with 8/E-type packaging for standard M Series modules. Also included are the 8/E power wiring harness and power bus board. There is M Series power bussing for all but the four slots in the first column. Four mounting spacers allow the H9190 to be easily mounted in the second half of an 8/E chassis.

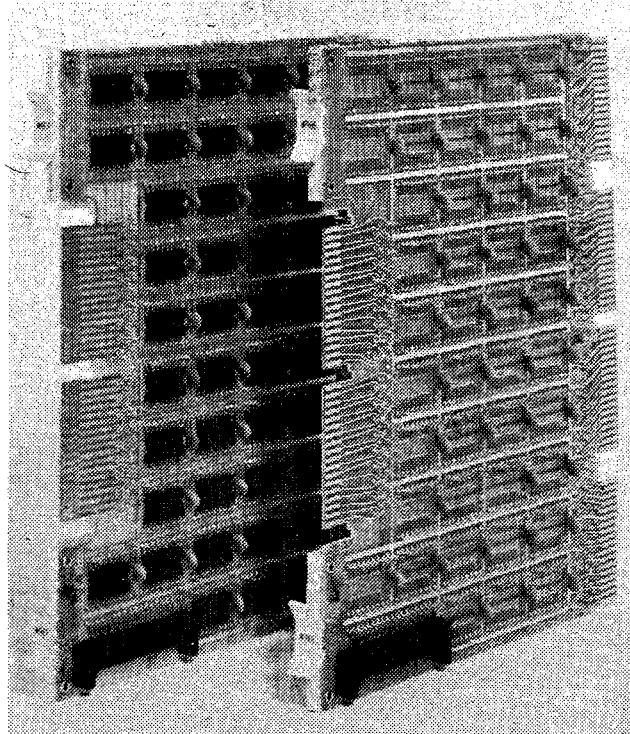
**H019 Mounting Bar**—an aluminum casting with the power bus board and power wiring harness. It also includes four mounting spacers for mounting in an 8/E chassis. Up to ten connector blocks of any type may be accommodated by this frame.



**H811-A Hand Wire Wrapping Tool (pencil type, 30-gauge)**



**H812-A Hand Unwrapping Tool (pencil type, 30-gauge)**



w966

W967

The W966 is the 8/E collage mounting board. It is double sided, extended length, and quad height with wire wrappable pins. It will accommodate 14- and/or 16- pin dual in-line IC's with or without 16-pin sockets. Two separate leads may be wire wrapped to each pin. Up to 42 IC's can be mounted on the W966. Discrete components may be directly soldered onto the board. The top center of the W966 board has 72 terminal fingers with terminating wire wrap pins. An I/O connector (male) terminating in wire wrap pins is mounted on the left side of the W966 board to provide access to the "outside world" when using BC08J-XX cable with a double sided connector board or a BC08K-XX single sided connector board. Both connector boards have 18 conductor lines.

All power and ground lines are common to the 8/E OMNIBUS.

AA2, BA2, CA2 +5

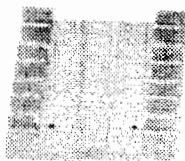
AC1, AC2, AF1, AF2, AN1, AN2, AT1, AT2  
BC1, BC2, BF1, BF2, BN1, BN2, BT1, BT2 GND  
CC1, CC2, CF1, CF2, CN1, CN2, CT1, CT2  
DC1, DC2, DF1, DF2, DNI, DN2, DT1, DT2

The W967 is similar in all details to the W966 except that the W967 is supplied with 42 low profile IC sockets.

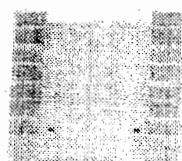


#### **H851 Edge Connector**

The H851 edge connector is used to bus signals from the top center terminal fingers to an adjacent quad board with similar terminals.



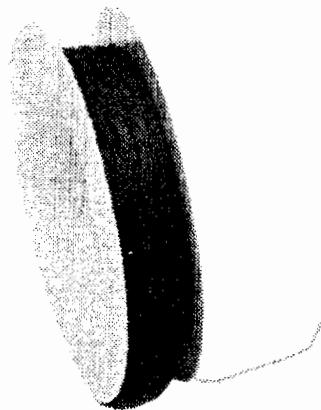
H852



H853

#### **H852, H853 Module Holders**

When using two or more W940, W941, W942, W943, W950 or W951 boards in parallel in logic connector blocks, rigidity of the boards is maintained by using the H852 rib type holder between board handles 1 and 2, 3 and 4, and using the H853 non-rib type holder between board handles 2 and 3.



**935—1000 foot roll, 30-gauge insulated wire.**

For additional information consult the latest edition of  
Digital's LOGIC HANDBOOK.

# 10

## i/o expansion techniques

The degree of versatility of a computer is determined by the type and number of peripheral devices that can be interfaced with it. The PDP-8/E was designed with this idea in mind. Consequently, it can be easily interfaced with a variety of peripherals in a variety of methods.

DEC is now offering two additional basic techniques of receiving and sending data to the OMNIBUS. These are:

1. Using the Positive I/O Bus Interface option,
2. Using standard M series Bus Receivers and Bus Transmitters.

Section 1 deals with the Positive I/O Bus technique of interfacing to the OMNIBUS and Section 2 describes the method of interfacing using standard M Series Modules and companion hardware.

### **SECTION 1 POSITIVE I/O BUS INTERFACING TECHNIQUES**

Previous discussions have brought out the fact that peripherals can be interfaced with either the OMNIBUS or the external bus. This means that a PDP-8/E user can utilize not only devices designed exclusively for the PDP-8/E, but also devices originally designed for use with the PDP-8/I and PDP-8/L computers, and even devices of his own manufacture. This also means that the entire catalog of M and K Series modules may be applied to satisfy any high speed and control application.

This section deals with the external bus, its applications, and the technique of interfacing peripherals to the bus. The user may wish to interface a DF32-D Disk File and Control unit with the PDP-8/E, for instance. This equipment was designed for the PDP-8/I and PDP-8/L; but, by interfacing to the external bus, it can also be used with the PDP-8/E. The user may want to transfer data between the PDP-8/E and a remote location. The external bus, which is designed to drive long interconnecting lines, is ideally suited for this application.

The first part of the chapter answers general questions about the external bus—What is it? How does it differ from the OMNIBUS? How does one use it? The remainder of the chapter guides the user through the initial uncertainties of interfacing by covering such topics as: types of connectors and cables to use with the external bus; timing criteria, loading rules, and voltage levels; and hardware and wiring techniques. DEC hope that this information will be helpful in designing and implementing any interface that the user might require.

Should questions arise regarding computer interface characteristics, the design of interfaces using DEC modules, or installation planning, customers are invited to telephone any of the DEC sales offices or the main plant in Maynard, Massachusetts. Digital Equipment Corporation makes no representation that the interconnection of its circuit modules in the manner described herein will not infringe on existing or future patent rights. Nor do the descriptions contained herein imply the granting of licenses to use, manufacture, or sell equipment constructed in accordance therewith.

### **THE NATURE OF THE EXTERNAL BUS**

What is the external bus? It is simply a number of signal lines (88, excluding grounds) that enable data transfers between the CP and peripherals. These lines carry data and control signals between the peripheral and two interface boards—the Positive I/O Bus interface (KA8-A) and the Data Break interface (KD8-E)—that plug into the PDP-8/E OMNIBUS. These two boards convert the internal bus signals into PDP-8/I and PDP-8/L-type bus signals. For instance, PDP-8/I peripherals need IOP pulses to perform instructions. The PDP-8/E does not generate internal IOP pulses, but it does provide signals (MD bits 09, 10, and 11) that can be converted into IOP pulses by the Positive I/O Bus interface. Other signals normally required by these peripherals are, in essence, available on the OMNIBUS. For example, BAC (buffered accumulator) bits must be supplied for the PDP-8/I peripherals. The PDP-8/E Data lines carry the necessary accumulator information. The Positive I/O Bus interface merely buffers the DATA bits and, thus, provides the external bus BAC signals.

Although the external bus consists of signal lines from both the positive I/O Bus interface and the Data Break interface, it is not always necessary to use both boards. When only programmed I/O transfer peripherals are used, the Positive I/O Bus interface provides all the necessary signals. However, if data break peripherals are to be connected, both interfaces must be used. Because each data break peripheral requires its own data break interface board, the number of signal lines comprising the bus may vary. There may be as many as 12 of these data break peripherals connected in the system, each contributing 36 signal lines to the external bus. Figure 10-1 illustrates the bus and its use when applied to a series of peripherals.

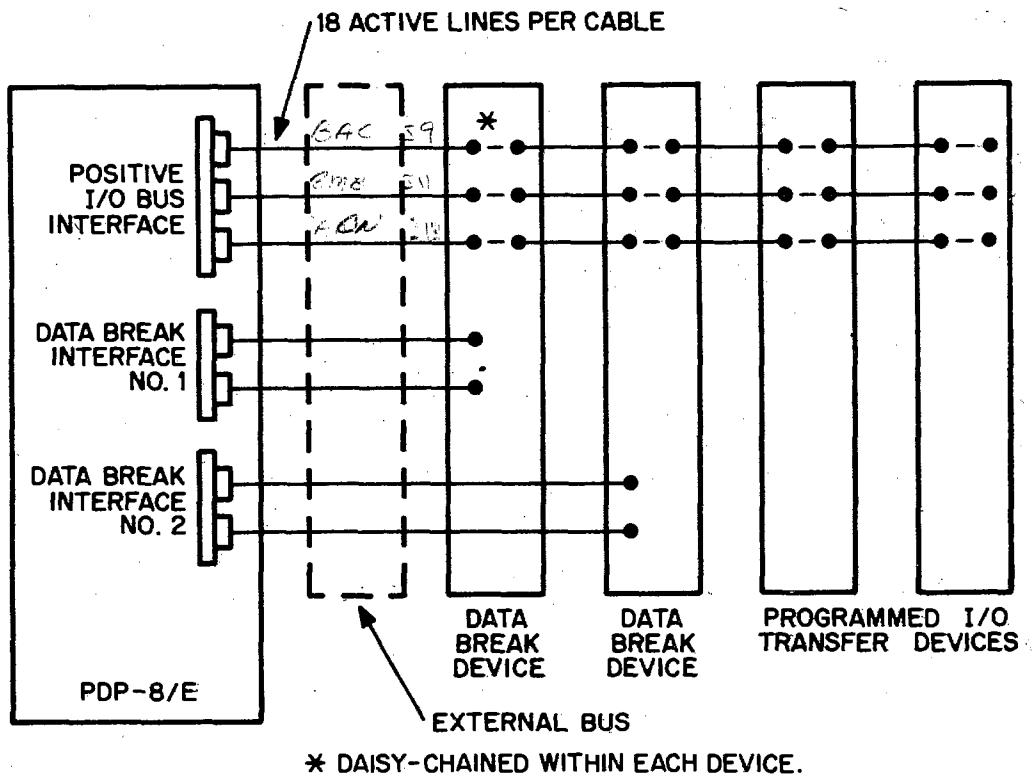


Figure 10-1 Parallel Connection of Peripherals

#### EXTERNAL BUS SIGNALS

Figure 10-2 shows not only the external bus signals, but also those OMNIBUS signals that are used by the two interfaces. Signal directions are shown for both buses. Some of the OMNIBUS signals—DATA 0-11, for instance—are common to both interfaces, but for clarity this commonality has been disregarded. The external bus signals are grouped according to the interface connector where they originate (Table 10-1 in paragraph 10.5 lists the bus signals and the connector and pin where each may be found). When similar signal lines are represented by one line of the drawing, as BAC 00-11, the actual number of lines is indicated in parentheses. The external bus signals are discussed in detail in the following section with emphasis on the relationship between these signals and the OMNIBUS signals.

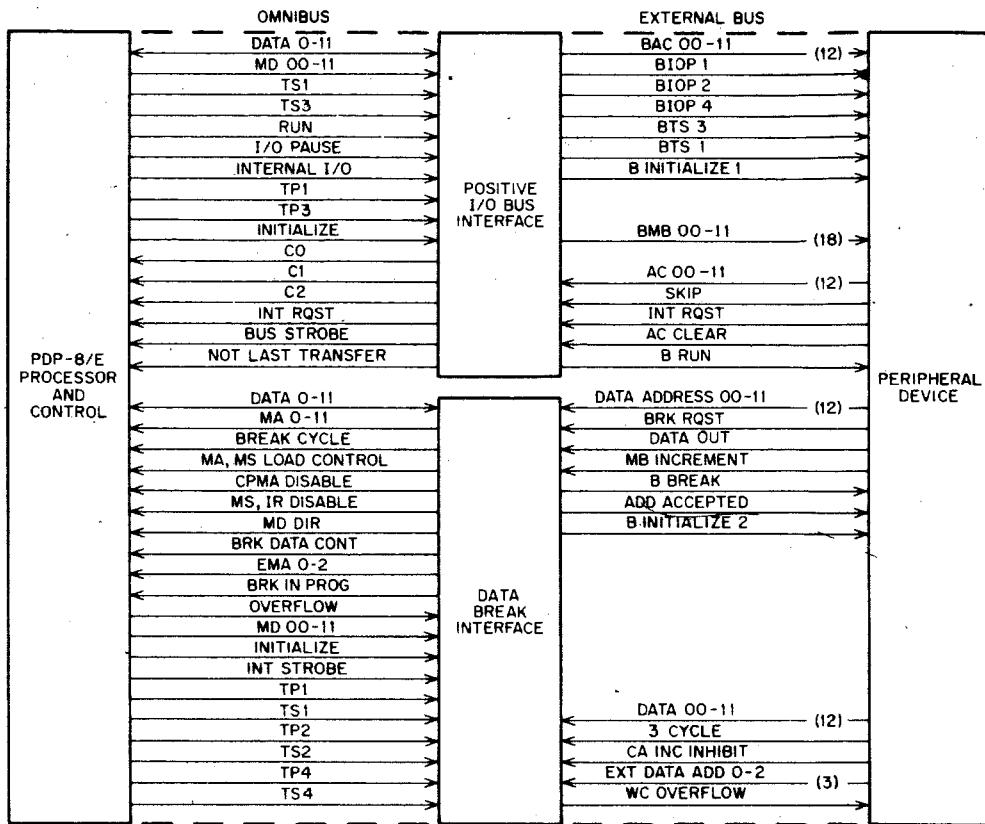


Figure 10-2 External Bus Signals and Related OMNIBUS Signals

SIGNAL NAME	DESCRIPTION
BAC 00-11	These signals represent the content of the PDP-8/E Accumulator register (AC). Information in the AC is transferred on the OMNIBUS DATA lines to the Positive I/O Bus interface. The interface buffers the signal and provides the BAC output. The BAC bits are strobed into registers in the peripheral when an IOT instruction is generated. 1 = +3V.
AC 00-11	The signals on these lines represent the contents of a register in the peripheral. This information is transferred to the Positive I/O Bus interface where it is put on the data lines for transfer to the PDP-8/E AC. 1 = GND.
BMB 00-11	The signals on these lines represent the content of the Memory Sense registers. This information is transferred from memory on the Memory Data (MD) lines. The MD lines are monitored by the Positive I/O Bus interface and the signals are converted to the BMB bits. These bits are used during IOT instructions; BMB03-08 carry the device selection code, while BMB09-11 are converted to BIOP pulses. 1 = +3V.

SIGNAL NAME	DESCRIPTION
BIOP 1, 2, & 4	These pulses are generated in response to the voltage levels on MD09-11 (BIOP4-1, respectively). These pulses generate IOT pulses within the peripheral, causing it to perform a certain operation. The width of the BIOP pulses and the interval between pulses are variable and can be adjusted on the Positive I/O Bus interface. Pulse = +3V.
BTS1, BTS3	These signals represent the TS1 and TS3 signals of the OMNIBUS. They synchronize operations in the peripheral with those in the computer and perform functions peculiar to the peripheral. They are primarily used in data break timing. Pulse = +3V.
B RUN	If this signal is GND, the computer is executing instructions.
AC CLEAR	When this signal is asserted (brought to GND) along with the AC bits, the result is a jam transfer of data to the AC. The signal may also be asserted by a separate IOP, clearing the AC.
SKIP	SKIP is asserted (grounded) by an IOT instruction. It causes the next sequential instruction to be skipped. If the SKIP bus is asserted during more than one IOT of an I/O instruction, the program skips a corresponding number of instructions. No more than three skips can be made by a single instruction.
B INITIALIZE 1	This 600 nanosecond-duration positive pulse is used to clear AC and link and to clear all flags in peripherals. It is generated at power turn on, and by the Clear All Flags (CAF) IOT, 6007.
DATA 00-11*	These lines transfer data from a data break peripheral to the data break interface. The peripheral transfers the information when it receives the B BREAK signal from the interface, indicating the start of the true break cycle. At TS2 of this break cycle, the data

SIGNAL NAME	DESCRIPTION
	break interface transfers the data to the OMNIBUS DATA 0-11 lines, which carry the data to the CP's memory buffer. 1 = GND.
B BREAK*	This signal is generated in the data break interface and transferred to the peripheral, where it enables a parallel loading of data, either into or out of the peripheral. The data break interface, in addition to generating B BREAK, asserts the OMNIBUS BREAK CYCLE line, notifying the computer that the break cycle has begun. 1 = GND.
DATA OUT*	This signal is produced by the peripheral and sampled by the data break interface. When DATA OUT is asserted (grounded) during the break cycle, data is transferred from the computer's memory to the peripheral.
DATA ADD 00-11*	These lines transfer address information from the peripheral to the OMNIBUS MA lines. If the peripheral is a 3-cycle break device, the address represents the memory location of the word count. Since this location is always the same for a 3-cycle device, the DATA ADDRESS lines are hard-wired in the peripheral. This address must be even (ending in 0, 2, 4, or 6) for word count. The data stored in this location represents the 2's complement of the number of data words to be transferred. The next sequential location is read from memory as the Current Address register. The data stored in this location represents the memory address of the data to be transferred. If the peripheral is a 1-cycle break device, the address on the DATA ADDRESS lines is provided by a register in the peripheral and represents the memory address of the data to be transferred. The address on the DATA ADDRESS lines is sampled by the TP4 pulse. The OMNIBUS CPMA DISABLE line is asserted by the data break interface at TP4 to enable the DATA ADDRESS information to be placed on the MA lines. 1 = GND.
BRK RQST*	This signal is asserted (brought to ground) by the peripheral when it is ready for a word transfer. When BRK RQST is present at INT STROBE time, the data break operation is entered. The OMNIBUS INT IN PROG line is asserted, and a load enable signal is provided for the data break interface break memory address (BKMA) register.

\*Pertains to Data Break interface only.

SIGNAL NAME	DESCRIPTION		
	DATA OUT	Operation Performed	Descriptive Term Used for Operation
ADD ACCEPTED*		This signal is generated by the data break interface when a BRK RQST signal has initiated the data break operation. ADD ACCEPTED is used in the peripheral to clear the BRK RQST flip-flop. Pulse = GND.	
MB INCREMENT*		When this signal is at ground level during the true break cycle, the contents of the memory location are acted upon as outlined in the following table.	
MB INCREMENT	DATA OUT	Operation Performed	Descriptive Term Used for Operation
Low	Low	Contents of the memory location are incremented.	MB Inc
Low	High	Data on the DATA 00-11 lines is added to the contents of the memory location.	Add to Memory (ADM)
CA INCREMENT INH*		When this signal is asserted (grounded) during the CA cycle of a 3-cycle data break, the CA is not incremented.	
3-CYCLE*		This signal is transferred from the peripheral to the data break interface to notify the interface logic to set either the WC flip-flop (3-cycle transfer) (ground input) or the B flip-flop (1-cycle transfer).	
WC OVERFLOW*		The interface transfers this signal to the peripheral to notify it that the word count location in memory has become zero and that the data transfer should end. The signal is also present when overflow occurs during MB increment or ADM. Pulse = GND.	
EXT DATA ADD 0-2*		These three lines are used when a KM8-E Memory Extension and Time Share interface is included in the basic PDP-8/E. The peripheral uses the lines to indicate the particular memory field involved in the transfer.	
		During a 3-cycle data break, WC and CA cycles always occur in field 0, while only the B cycle occurs in the field specified by the extended data address. 1 = GND.	
B INITIALIZE 2*		This positive signal clears all flags in the peripheral and is essentially the INITIALIZE signal of the OMNI-BUS. It is used by the break device in lieu of B INITIALIZE 1 so as to reduce loading on the latter.	

\* Pertains to Data Break interface only.

## APPLICATION

The nature of the external bus and its relationship to the OMNIBUS have been presented. Now, the use of the bus must be fully explored. First of all, the user wants to transfer data between his peripheral and the computer's memory. He can do this in any one of three ways—programmed I/O transfers, program interrupt transfers, or data break transfers. The basic ideas behind all three methods of data transfer have been discussed in previous chapters, and the user should be familiar with these before proceeding any further in this chapter.

### Programmed I/O Transfers

Figure 10-3 is a logic block diagram that shows the more important signals involved in a programmed I/O transfer. The transfer process is, of course, similar to that which takes place between the computer and a peripheral interfaced to the OMNIBUS. Each peripheral has a flag flip-flop that is set when the peripheral is ready to receive or send information. A programmed IOT instruction is used to check this flip-flop. The program enters a waiting loop until the peripheral is ready. When the flag flip-flop is set, IOT XXA (illustrated on the block diagram) asserts the SKIP bus. The program then skips to an instruction that transfers program control to a servicing subroutine. The subroutine carries on the IOT dialogue between peripheral and processor. Although the general process is similar, both the method of peripheral selection and the use of the SKIP function differ for external bus peripherals.

The method of peripheral selection will be examined first. As shown on the block diagram, the peripheral contains a device selector and an IOT generator. The device selector monitors the BMB03 through BMB08 lines. These lines are merely the buffered OMNIBUS MD03 through MD08 lines. Thus, when an IOT instruction is issued, BMB03 through BMB08 carry the code for a particular peripheral. The device selector responds to the code by producing a DEVICE SELECTED signal. This signal is applied to the IOT generator. Unlike the OMNIBUS peripherals, external bus peripherals require BIOP pulses to carry out the operations specified by the IOT instructions. These pulses are generated in the Positive I/O Bus interface and reflect the information present on the OMNIBUS MD09, 10, and 11 lines. BIOP pulses are applied to the IOT generator where they are regenerated as IOT pulses that initiate operations within the peripheral. Figure 10-3 shows that the BIOP pulses (1, 2, and 4) are generated only when the following conditions are met: the OMNIBUS I/O PAUSE line is asserted, indicating that an I/O transfer is to take place; the INTERNAL I/O line is negated, indicating that the I/O transfer is to or from an external bus peripheral; the MD09, or MD10, or MD11 line is asserted. Thus, if the first two conditions are met, and MD09 is asserted, BIOP4 is generated. Similarly, BIOP1 is generated when MD11 is asserted. Each BIOP pulse is regenerated as an IOT pulse within the peripheral. Thus, BIOP1 becomes IOT XX1; BIOP4 becomes IOTXX4. (In the preceding notation, XX represents the particular device selection code).

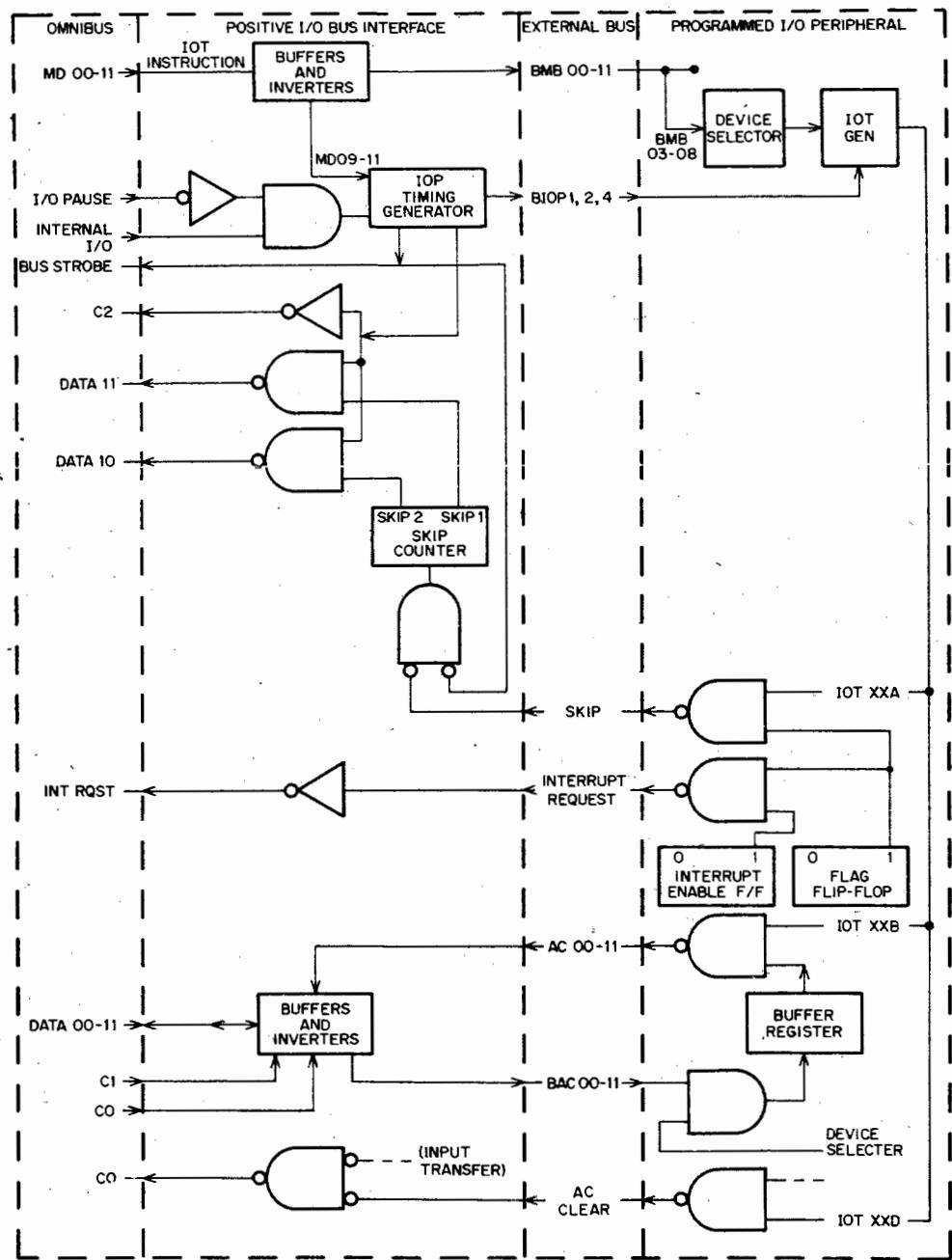


Figure 10-3 Block Diagram, Programmed Data Transfer

Specific IOT pulses perform specific operations in the peripheral. The primary use of specific IOT pulses, as well as the relationship between IOT, BIOP, and MD bits, is as follows:

<u>MD Bit</u>	<u>BIOP</u>	<u>IOT</u>	<u>Primary Use</u>
09	BIOP4	IOT XX4	Reading, loading, and clearing buffers.
10	BIOP2	IOT XX2	Clearing flags; clearing AC.
11	BIOP1	IOT XX1	Sampling flags; skipping.

These relationships are standard within DEC and are certainly not mandatory for the user. He may decide to use IOT XX1 to clear a flag, or he might wish to use IOT XX4 to sample a flag. He may even use combinations such as IOT XX5 by generating both BIOP1 and BIOP4 with the same IOT instruction. In any event, it is wise to develop some standard such as that expressed above.

The discussion so far has shown how the peripheral is selected and how operations are initiated by the IOT instruction. The use of the SKIP function must now be explained. One should recall that a flag flip-flop in the peripheral is sampled by an IOT pulse and a skip is effected. IOT XXA checks the status of this flip-flop (XXA is used here, rather than XX1, to emphasize the fact that "A" may be 1, 2, 5, or whatever the user wishes). When the peripheral is ready, the SKIP bus is asserted. A strobe signal from the IOP timing generator clocks the skip counter, a two-stage binary counter. This strobe signal is generated near the end of the BIOP pulse. If one BIOP pulse is generated by the IOT instruction, the skip counter is clocked only once and the SKIP 1 line is asserted. A control signal, produced at the end of the BIOP pulse, then asserts both the C2 line and the DATA 11 line of the OMNIBUS. When the C2 line is asserted, with C1 negated, DATA + PC goes to PC. In other words, the PC is incremented by one and the program skips one instruction. Figure 10-4 is a timing diagram of the SKIP function and is helpful in visualizing the process. Note that two strobes are generated, each performing the function shown on the diagram.

It should also be noted on the timing diagram that the CP operation is halted from TP3 to the beginning of the second BUS STROBE. This BUS STROBE, which occurs after NOT LAST TRANSFER has been negated, generates INT STROBE, which restarts the CP. This is the disadvantage inherent in external bus interfacing—the CP must remain inactive while BIOP pulses are generated and control lines are activated. Thus, while OMNIBUS I/O transfers are accomplished in 1.2 microseconds, the minimum time required for external bus I/O transfers is 2.6 microseconds when only one BIOP pulse is generated by the IOT instruction. Suppose two BIOP pulses are generated. The user might want to skip two instructions in the program, for example. In this case, the IOP timing generator produces three BUS STROBES. Each of the first two BUS STROBES clocks the skip counter. Thus, the SKIP 2 line is asserted. Again, C2 is asserted by a control signal, but now DATA 10 rather than DATA 11 is brought low. BUS STROBE three then enables DATA + PC to the PC, and the program skips the next two instructions. This may be advantageous for a specific application, but note the increased transfer processing time—it is now 3.6 microseconds. If three BIOP pulses are used, the CP is halted for 4.6 microseconds. It must be remembered that these are minimum values. Figure 10-4 shows that the time from

TP3 to the beginning of BIOP1 is variable, as is the width and the pulse separation when more than one BIOP is issued by the IOT instruction.

The user may want to increase IOP width and/or separation for specific applications. He should bear in mind, however, that this affects processing time and may slow the computer appreciably. The width and separation are controlled by separate potentiometers on the interface.

The minimum allowable values are 800 nanoseconds for pulse width and 200 nanoseconds for pulse separation. The resistor values in the timing circuits allow the user to increase these values to five times the minimum.

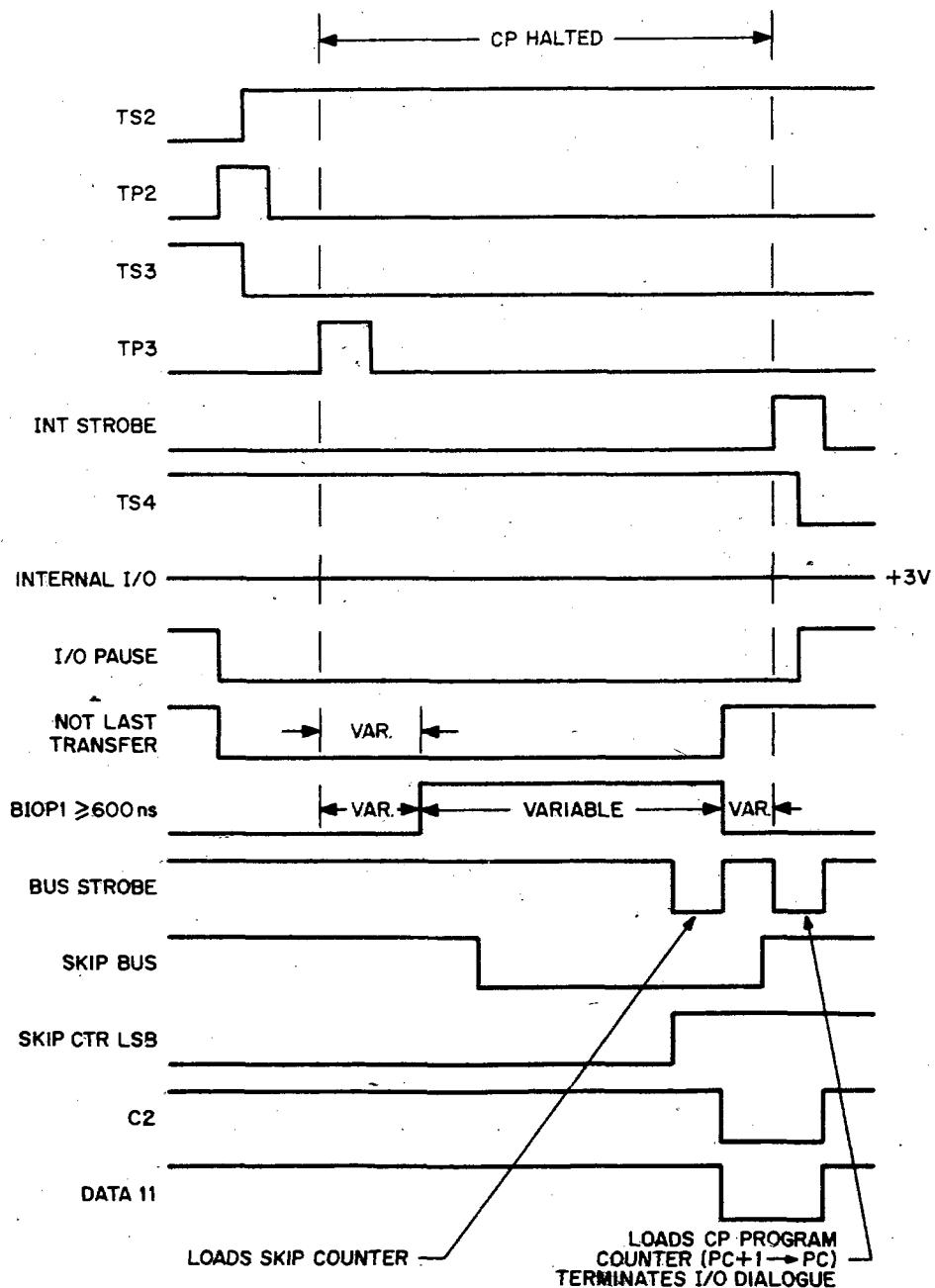


Figure 10-4 Timing Diagram, Skip Bus Application

Figure 10-5 shows another timing diagram, this one illustrating a data word transfer. Conventionally, BIOP4 is being used to accomplish the transfer. Note that the BIOP pulses do not occupy specific time slots. Thus, if BIOP1 and BIOP2 are not required, as in this example, the IOP timing generator produces BIOP4 without any delay. Again two BUS STROBES are generated, each performing the indicated function. The C1 control line must be asserted to indicate an input data transfer. This is done if data is placed on any external bus AC line. The data is buffered and inverted, and placed on the OMNIBUS DATA 0-11 lines (one of these lines, DATA XX, is illustrated). With only C1 asserted, a 1's transfer to the AC is carried out. If a jam transfer to the AC is desired, the C0 control line must be asserted along with C1. This can be done only if the external bus AC CLEAR line is brought to ground, which can be accomplished either by gating IOT4 in the peripheral or by other means that the user might select. Do not permanently ground AC CLEAR.

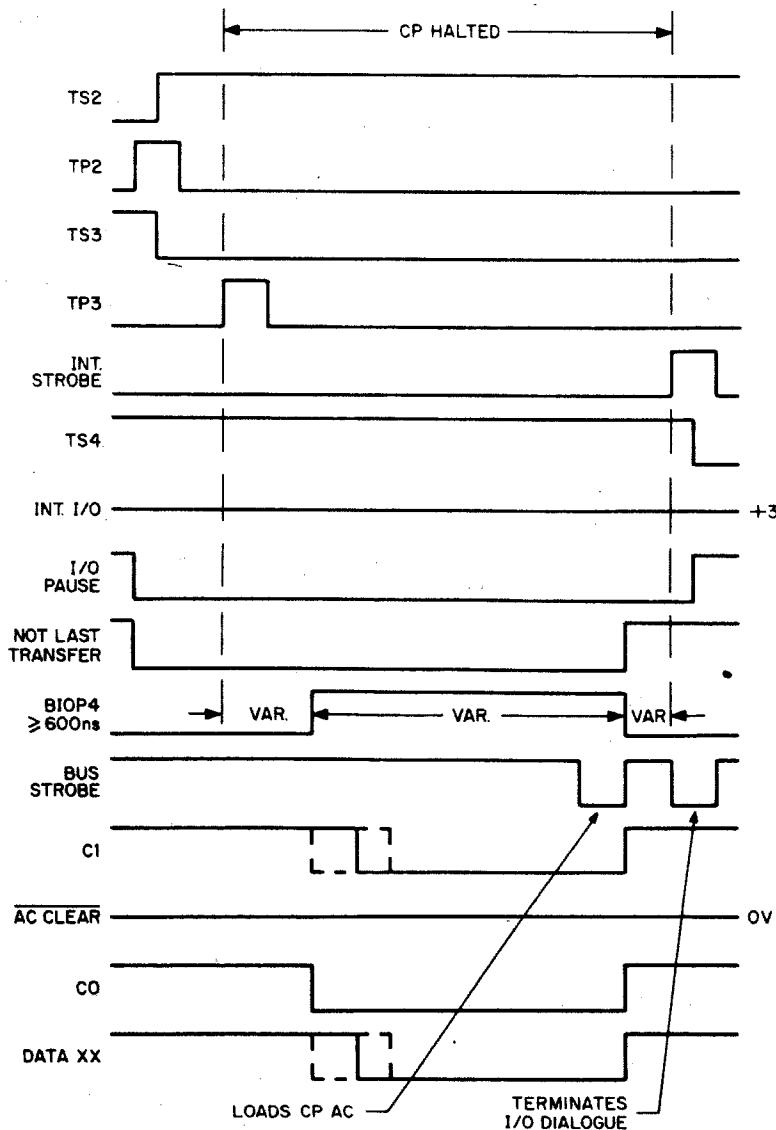


Figure 10-5 Timing Diagram, Programmed Data Transfer

### **Program Interrupt Transfers**

Program interrupt devices are connected to the external bus exactly as are programmed data transfer devices, and use the very same signals. The entire program interrupt process proceeds just as it does for OMNIBUS peripherals. The peripheral requests an interrupt by setting its flag flip-flop. The processor honors the request and services the peripheral in a program subroutine. The data transfer is accomplished by programmed transfers, as explained in the preceding section. The user must provide means of asserting (grounding) the external bus INTERRUPT REQUEST line when the peripheral is ready for a transfer. Figure 10-3 shows the peripherals flag flip-flop controlling the INTERRUPT REQUEST line. The SKIP line must also be used when more than one such interrupt device is connected onto the external bus and is utilized as detailed in the preceding section.

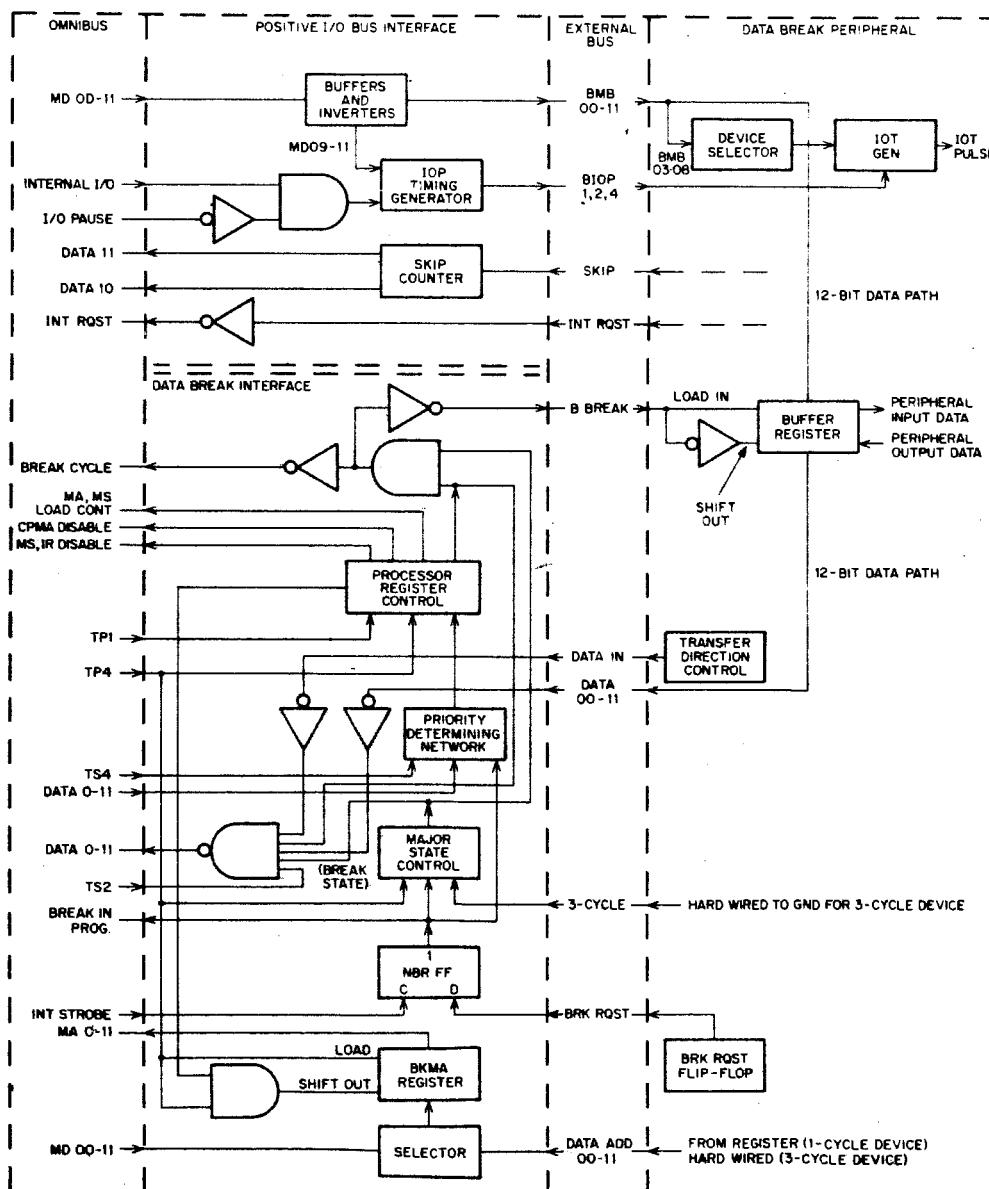
### **Data Break Transfers**

Data break transfers involving peripherals on the external bus are accomplished in much the same way as transfers involving OMNIBUS peripherals. Refer to figure 10-6, which is a logic block diagram that shows the more important functional blocks and signals involved in a data break transfer. Figure 10-7 shows the timing relationship of the major signals and, along with the block diagram, should be referred to throughout this discussion.

An important feature that is illustrated on the block diagram is the circular flow of data between OMNIBUS and peripheral. The peripheral receives data from the computer by way of the OMNIBUS MD lines and the external bus BMB lines. It sends data to the computer by way of the external bus DATA 00-11 lines and the OMNIBUS DATA 0-11 lines. Thus, there is a need for both interface boards when data break peripherals are used. In addition to providing a data path, the Positive I/O Bus interface provides BIOP pulses. These pulses, as in programmed data transfers, are used in conjunction with a device selector to produce IOT pulses in the peripheral. However, data break IOT pulses are used only to initiate the data break operation and, once the operation has started, program control of the peripheral ceases. Therefore, the program provides IOT instructions that tell the peripheral, via the IOT pulses, to perform any preliminary operations that may be required. When the peripheral has followed these instructions and is ready to either send or receive data, it requests a Data Break. The data break interface logic then assumes control over the operation.

When the Data Break interface receives the BRK RQST signal from the peripheral, it uses the next INT STROBE from the OMNIBUS to assert the OMNIBUS BRK IN PROG line. At TS4 the interface checks the priority network to make sure no higher priority device is present in the system. If none is present, the interface asserts two OMNIBUS processor-control lines at TP4. Assertion of these lines—CPMA DISABLE and MS, IR DISABLE—causes the processor to load its CPMA register and suspend operation, while the peripheral interface BKMA register assumes control of the OMNIBUS MA lines (as with OMNIBUS data break peripherals, the break can occur at the end of any processor major state). At the same time—TP4—the interface major state control logic enters either

the word count cycle or the break cycle, depending on the state of the 3 CYCLE signal from the peripheral. If this peripheral is a single-cycle break device, the interface logic enters the B (break) cycle. The B BREAK signal is generated and sent to the peripheral where it either loads the output buffer register with data or places data from this register on the lines to the interface logic. If the transfer direction is from the peripheral to memory, data from the buffer register is placed on the DATA 00-11 lines. At TS2 this data is placed onto the OMNIBUS DATA 0-11 lines. The next TP2 loads the data into the MB, providing the BREAK DATA CONT has not been asserted, and the MB contents are then written in memory.



**Figure 10-6 Block Diagram, Data Break Transfer**

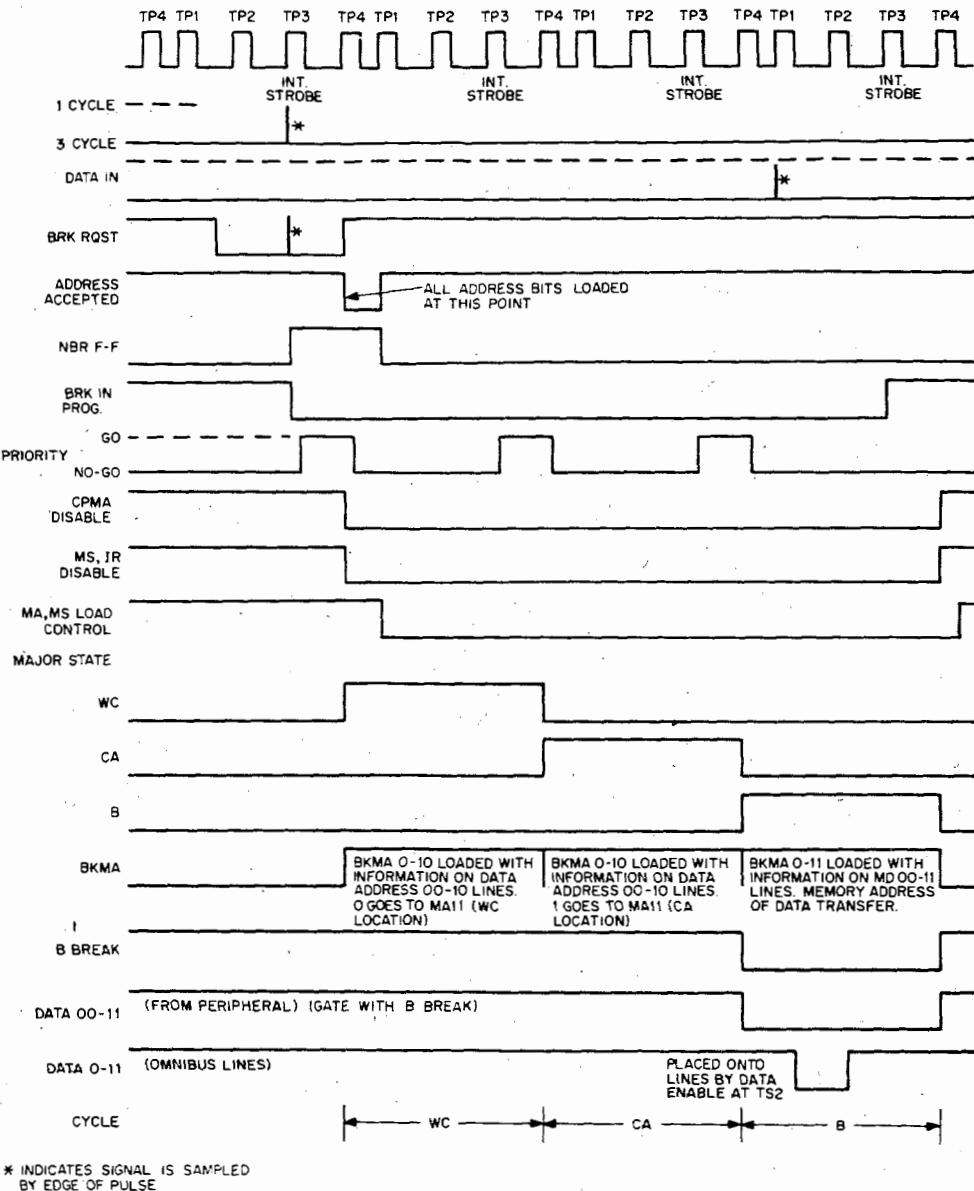


Figure 10-7 Timing Diagram, Data Break Transfer

Figure 10-7 indicates a peripheral-to-computer data transfer. It also shows the timing for a 3-cycle, rather than a single-cycle, break device, and indicates that the WC and CA cycles must be completed before B BREAK is generated at the beginning of the B cycle. In addition, the timing illustrates a very important and advantageous feature of the interface logic—priority is checked not only before entering the WC cycle, but also before entering the CA and B cycles. This means that high priority devices can override lower priority devices even in the middle of a break operation. As many as 12 Data Break peripheral interface boards can be inserted into the OMNIBUS. Each peripheral must first be assigned a priority level. Each interface contains an identical priority network that is wired in a unique fashion, so as to reflect the priority

level assigned to the peripheral. During any TS4 of a data break cycle, the highest priority interface, of those interfaces whose peripherals have made break requests, receives a 'go' signal from its priority network. All other interfaces whose peripherals have made requests receive a 'no-go' signal, and must wait until the next TS4 for another priority check.

This unique priority system allows the user to utilize his data transfer system in a most efficient manner. However, he must pay careful attention to the manner in which he assigns priorities. An essential rule should be: assign priorities in relation to the length of time data is available at the peripheral. For example: peripheral A and peripheral B both assert their BRK RQST lines at some time between TP1 and TP3 of a particular processor cycle. Peripheral A has been assigned a higher priority than peripheral B; and, thus, takes control of the break operation. If 'A' is a 3-cycle device, assuming no other higher priority devices are present, it does not relinquish control until it has transferred the data word during the B-cycle. During TS4 of this cycle, 'B' receives a priority 'go' signal and takes control of the operation at TP4. If 'B' is also a 3-cycle break device it must now go through WC and CA before the actual transfer can begin.

Seven to eight microseconds has elapsed from the moment that 'B' requested a break until the moment when its interface generates B BREAK. If 'B' is a very high speed device, the data that was present in its output register at the time of the break request may no longer be present when B BREAK is finally generated. The solution in this example is obvious (if 'A' is a slower device than 'B', that is): assign 'B' a higher priority than 'A'. OMNIBUS data break peripherals are also equipped with priority networks, although these generally differ from the network of the Data Break interface. Nevertheless, all types work together, and the only limitation is on the total number of priority networks, 12.

While OMNIBUS data break peripherals are generally single-cycle devices, those data break peripherals interfaced to the external bus are, for the most part, 3-cycle devices. Because the WC and CA registers of a 3-cycle device are located within core memory, the Data Break interface BKMA register is used differently from its counterpart in an OMNIBUS peripheral. Figure 10-7 indicates the use of the BKMA register during each cycle of the 3-cycle break operation. During WC, the hard-wired memory address of the word count register is loaded into the BKMA. The only restrictions on this address are that bit 11 be a '0' and that the memory location be in memory field 0.

The word count in memory is brought out, incremented, and deposited back in memory. During CA, the same hard-wired address is loaded into BKMA 0-10, and BKMA11 is asserted. The result is the memory address of the current address register. The current address is brought out of memory, incremented, and placed on the MD lines. Not only is this address deposited back in the CA register, but it is also loaded into the BKMA register at the beginning of the B cycle. Thus, the current address specifies the memory address to which, or from which, the data is to be transferred.

In the preceding explanation, data has been presented as a 12-bit data word that is transferred unaltered to or from a specified memory location. However, the interface logic provides the user with two features that somewhat alter this idea. The first, MB INCREMENT, allows the user to transfer a single bit of information via the OMNIBUS DATA 11 line. The result is an incrementation of the data contained in the specified memory location. The peripheral must cause the external bus MB INCREMENT line to be asserted with the DATA OUT line asserted. Each time MB INCREMENT is asserted, the interface logic asserts the OMNIBUS DATA 11 line and the data in the memory location is incremented.

The second feature, Add to Memory (ADM), allows the user to alter the 12-bit data word in memory. Specifically, the incoming data word and the data contained in the addressed memory location are added in the processor memory buffer. The result is then returned to the addressed location. As with MB INCREMENT, the peripheral must cause the MB INCREMENT line to be asserted; but now the DATA OUT line must be negated. The result is ADM. Table 10-1 lists the possible states of the DATA IN line and the MB INCREMENT line, and the resulting data transfer. L indicates that the line is asserted (grounded), while H indicates that the line is negated.

**Table 10-1. Data Control Lines (Data Break Interface)**

DATA OUT	MB INCREMENT	TYPE AND DISPOSITION OF DATA
L	H	12-bit data word transferred from the addressed memory location to the peripheral buffer via the BMB lines.
H	H	12-bit data word transferred from the peripheral buffer to the addressed memory location via the DATA00-11 lines.
L	L	MB INCREMENT. The contents of the addressed memory location are incremented.
H	L	Add to Memory (ADM). The twelve-bit data word on the DATA 00-11 lines is added to the contents of the addressed memory location, and the result stored in the addressed memory location.

#### **INTERFACING TECHNIQUES**

At this point the user should have a good understanding of what the external bus is and the way that he wants to use it. The remainder of this chapter is devoted to the technique of interfacing the bus to the user's peripheral.

When the user purchases a PDP-8/E, he selects the options he requires. Assume that he wants both interfaces. Digital Equipment Corporation then supplies the boards, the interconnecting cables, and the connectors on either end of each cable. The customer specifies whether the cable should be shielded flat cable or round or flat coaxial cable. In addition, he selects the cable length from a variety of standard lengths made available by DEC. The connectors on either end of the cable are standard DEC connectors and those on the free end will connect into any peripheral control manufactured by DEC.

#### **PDP-8/I and 8L-type Peripherals**

If the user has a PDP-8/I or PDP-8/L-type peripheral control, the PDP-8/E can be easily interfaced to it. If the control is a data break peripheral control, the user first plugs the two interface boards (with the cables already connected to the board) into the OMNIBUS, in any available slots. He then inserts the peripheral control connectors into the appropriate slots in the peripheral control connector block. In general, the appropriate slot is determined as follows:

<u>Cable Connector</u>	<u>Peripheral Module Slot</u>	<u>Signals</u>
I/O Bus Interface	Data Break Interface	
Cable 1	A1	BAC
Cable 2	A2	BMB
Cable 3	A3	AC INPUT
Cable 4	A4	DATA ADDRESS
Cable 5	A5	DATA BITS

(The above information is generally true; however, because peripheral control module arrangements sometimes vary, the user should check the peripheral control interconnection information to see that the stated correspondence is correct.)

If the peripheral is a programmed I/O device, only the Positive I/O Bus interface must be inserted into the OMNIBUS. In this instance, cables 1, 2, and 3 of the board connect as indicated above. If several peripherals are being connected to the bus, they are connected as indicated in figure 10-1. Table 10-2 lists the external bus signals and provides the connector pin assignment for each signal line. The location of a particular pin can be determined as follows: if the connector board is held so that the connector pins are to the left and the keying cutouts appear as shown in Figure 10-8 (the longer cutout on the bottom), pin A is the topmost pin, pin V is the bottommost, and side 2 is the nearer side. Pins are lettered A through V (excluding G, I, O, and Q). This convention of pin identification also applies to the M-series and G-series connectors and modules.

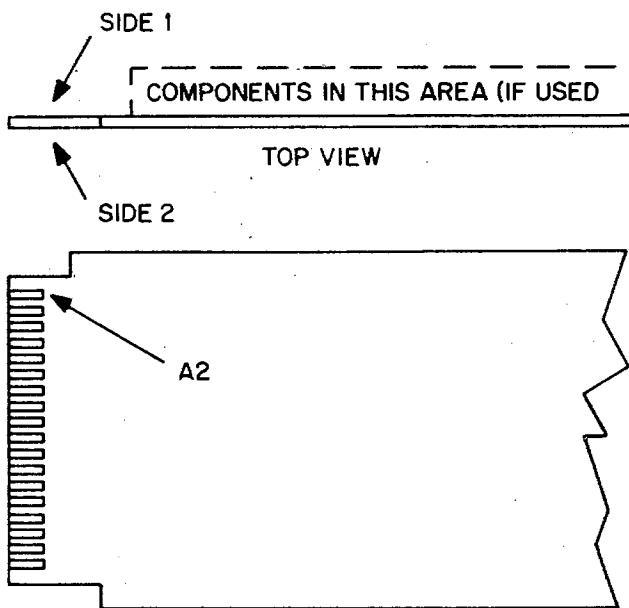


Figure 10-8 Connector Pin Identification

Table 10-2 External Bus Connector Pin Assignments

POSITIVE I/O BUS INTERFACE

<u>Cable 1</u>	<u>Signal Name</u>	<u>Connector Pin</u>
<u>Cable 2</u>	<u>Cable 3</u>	
BAC 00	BMB00(1)	AC 00*
BAC 01	BMB01(1)	AC 01*
BAC 02	BMB02(1)	AC 02*
BAC 03	BMB03(0)	AC 03*
BAC 04	BMB03(1)	AC 04*
BAC 05	BMB04(0)	AC 05*
BAC 06	BMB04(1)	AC 06*
BAC 07	BMB05(0)	AC 07*
BAC 08	BMB05(1)	AC 08*
BAC 09	BMB06(0)	AC 09 *
BAC 10	BMB06(1)	AC 10*
BAC 11	BMB07(0)	AC 11*
BIOP1	BMB07(1)	SKIP BUS*
BIOP2	BMB08(0)	INT RQST BUS*
BIOP4	BMB08(1)	AC CLEAR BUS*
BTS3	BMB09(1)	B RUN (0)
BTS1	BMB10(1)	T2
B INITIALIZE 1	BMB11(1)	V2

## DATA BREAK INTERFACE

<u>Signal Name</u>	<u>Connector Pin</u>
<u>Cable 1</u>	<u>Cable 2</u>
DATA ADD 00*	DATA 00*
DATA ADD 01*	DATA 01*
DATA ADD 02*	DATA 02*
DATA ADD 03*	DATA 03*
DATA ADD 04*	DATA 04*
DATA ADD 05*	DATA 05*
DATA ADD 06*	DATA 06*
DATA ADD 07*	DATA 07*
DATA ADD 08*	DATA 08*
DATA ADD 09*	DATA 09*
DATA ADD 10*	DATA 10*
DATA ADD 11*	DATA 11*
BRK RQST*	3 CYCLE*
DATA OUT*	CA INCREMENT INH*
B BREAK (0)	BWC OVERFLOW
ADD ACCEPTED	EXT DATA ADD 02*
MB INCREMENT*	EXT DATA ADD 01*
B INITIALIZE 2	EXT DATA ADD 00*

Signals marked \* are input signals, and are asserted at ground. Unmarked signals are output signals, and are asserted at +3V.

Note: Be sure to ground all other pins except U1, V1, A2, and B2.

### **Customer Peripherals**

The interfacing technique becomes more complicated when the user desires to connect his own peripheral. The user must decide if he wants to use the cables and connectors that are supplied with the interface board. If he elects to do so, he must use DEC connector blocks that accommodate the BC08J interface cables. He must then provide a mechanical interface between the DEC connector block and the peripheral input connectors, and, in addition, must provide an electrical interface between the external bus and the peripheral. The designer of the interface has to consider questions of voltage levels, loading criteria, mechanical compatibility, etc. The task of interfacing is greatly simplified if the customer makes use of DEC's line of M-series modules and compatible H-series connector blocks and mounting panels.

### **DEC Logic Module Interfacing**

As an example of the application of DEC modules to the user's interfacing problem, consider the following: the peripheral, whether a data break device or a programmed transfer device, must be able to recognize its device code and respond in some way to that code. The device code is contained in the BMB03-08 bits, which are provided by cable 2 from the Positive I/O Bus interface. The user can design his own integrated circuit or discrete component interface to decode the BMB03-08 bits and provide a Device Selected signal. He can then breadboard the circuit, test it, build it, and install it in some kind of connector,

which he then has to connect to the interface cable. Alternately, he could use a DEC M103 Device Selector module, which, in addition to providing the correct Device Selected signal, accomplishes most of the interfacing, provides diode clamp protection, and even provides extra logic gates for the customers use.

The M103 module is shown in Figure 10-9. Assume that the user assigns device code 14 to his peripheral. The selected BMB lines provide an enabling signal for gate 2 whenever device 14 is selected. The module also provides the peripheral with the necessary IOT pulses, as is indicated in Figure 10-9. The output signals are available for immediate use.

The M-series modules use TTL logic, and the input loading/output drive requirements are given in number of unit loads. A unit load is defined as follows: in the logic 0 state, the driver must be able to sink 1.6 milliamps (maximum) from the unit load's input circuit, while maintaining an output voltage equal to or less than 0.4 volts; in the logic 1 state, the driver must maintain an output voltage equal to or greater than 2.4 volts, while supplying the unit load with a leakage current of no more than 40 microamps. The M103 is capable of driving 37 TTL unit loads at the IOT outputs, while the DEVICE SELECTED output can drive 16 TTL unit loads.

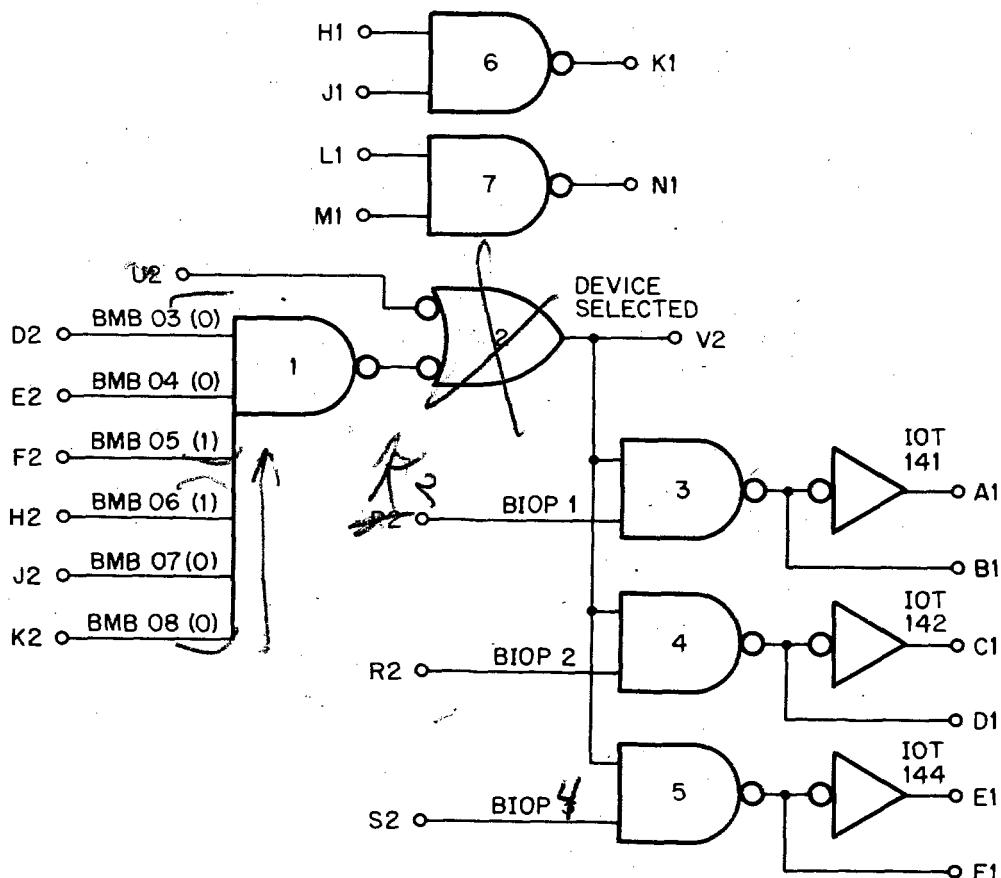


Figure 10-9 M103 Device Selector Module

Besides getting the peripheral selected, it is also necessary to transfer data. If, for example, data is to be sent to the PDP-8/E accumulator, it must be fed onto the external bus ACOO-11 lines. The M624 Bus Driver module can provide this capability. This module, shown in Figure 10-10, contains 15 bus drivers. Twelve have a common gate line and can be used as shown in Figure 10-11. An output register in the peripheral is loaded with the data that must be transferred. The IOT pulse can then clock the data onto the AC lines. The user provides the proper input drive for the bus drivers. The M624 presents the following input TTL unit loads; 12 loads at the clock input (IOT 14X line); 1 load at the data inputs.

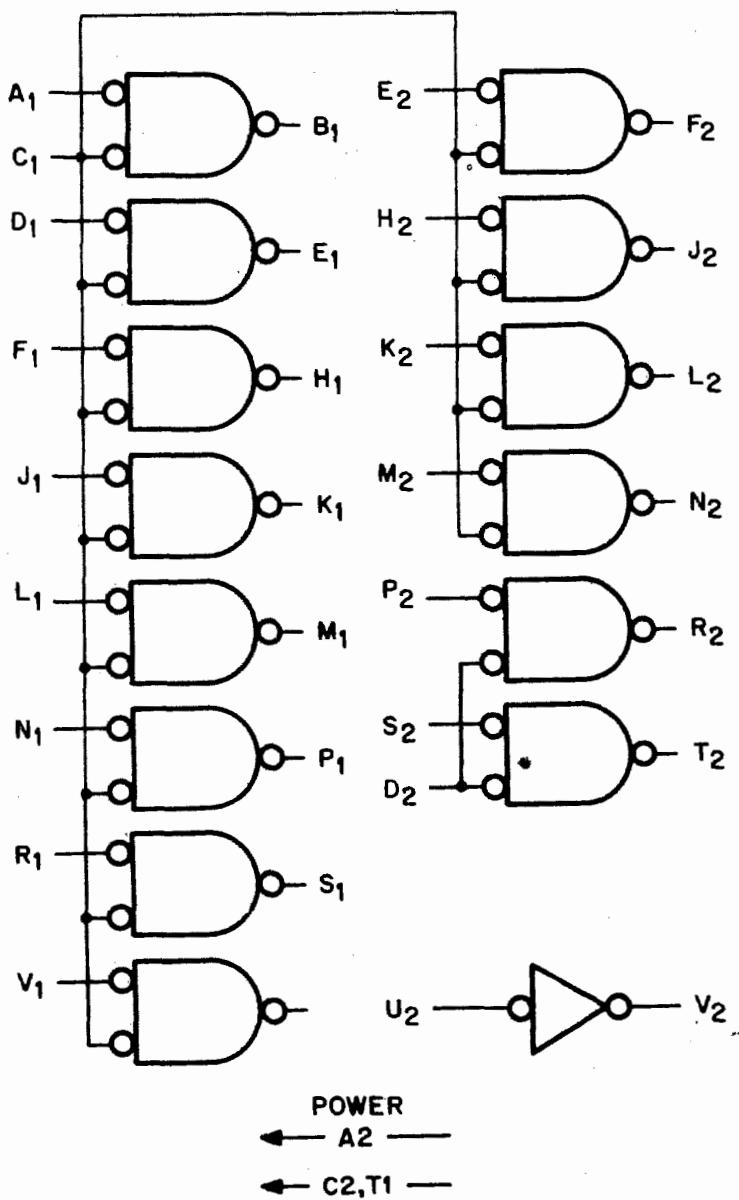


Figure 10-10 M624 Bus Driver Module

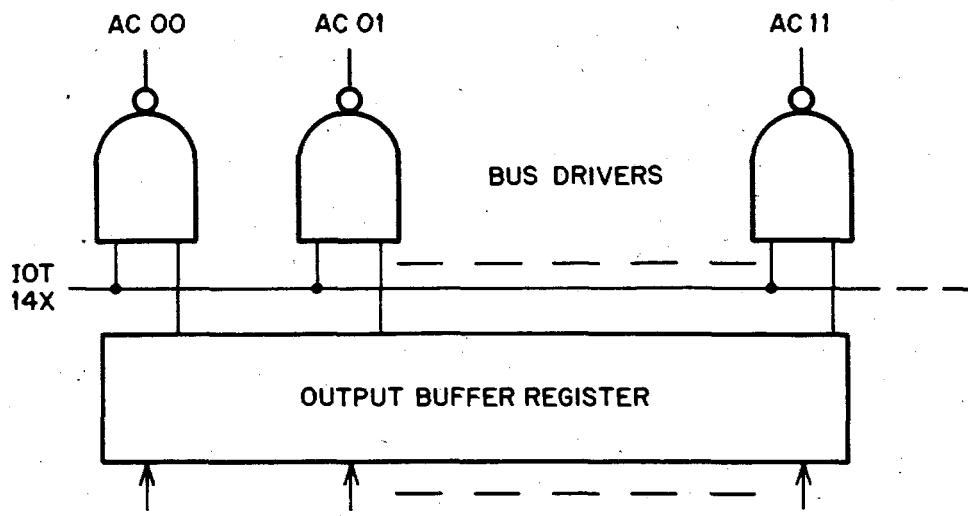


Figure 10-11 Bus Driver Application

The user also has to transfer data from the accumulator. Therefore, he must get the data on the BAC 00-11 lines into the peripheral buffer register. While doing this, he must take care that he does not place an excessive load on any BAC line. If he uses an M101 Bus Data interface, shown in figure 10-12, he can be sure of maintaining proper loading of these lines. Figure 10-13 shows how this interface must be used. Twelve of the gates are connected to the 'device select' line. When the device (XX) is selected, the BAC 00-11 data is applied to the input lines of the buffer register. IOT XXY then loads the register with the data word. The M101 presents the following input TTL unit loads: 15 loads at the clock input (device select line); 1 load at the data inputs.

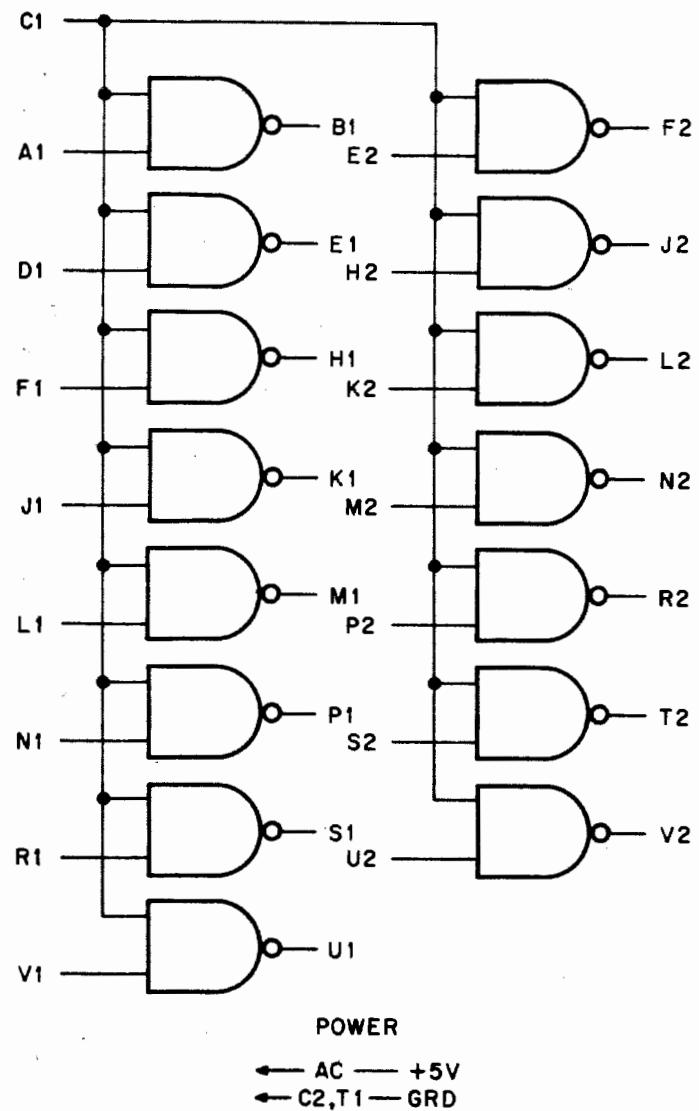


Figure 10-12 M101 Bus Data Interface

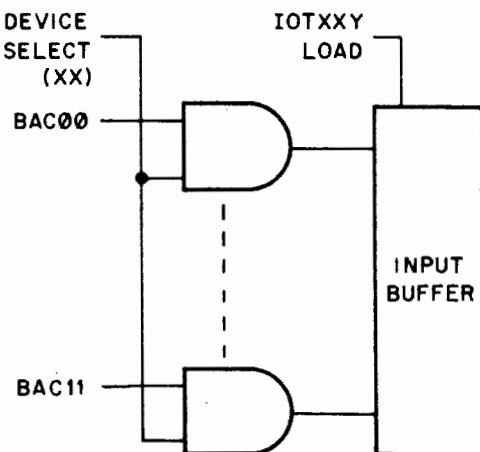


Figure 10-13 Bus Data Application

A variety of M-series modules are available from which the user can select to meet his needs. A list of these modules is given at the end of this chapter. This list should prove helpful in the design of interfaces. Attention is also directed to DEC's DIGITAL LOGIC HANDBOOK available from all sales offices.

M-series modules greatly simplify the task of electrical interfacing. However, this is only part of the problem. The external bus and the peripheral must also be mechanically interfaced. DEC's H-series of connector blocks and mounting panels are extremely valuable in this application. The following example takes the user through a mechanical interfacing procedure to acquaint him with the technique.

The user wants to interface a programmed transfer type of peripheral to the external bus. He wants to use the cables provided with the Positive I/O Bus interface, and he intends to solve part of the interfacing problem by using M-series modules. Not only does he need a connector block into which he can insert either an M-series module, or the connector on cables 1, 2, and 3, but he also needs something on which to mount each connector block. DEC's H803 connector block, providing slots for eight modules, accepts the connectors, the M-series modules, and even DEC's A-, K-, and W-series of modules. To mount these connector blocks, H911 mounting panels can be used. This mounting panel contains eight H803 connector blocks and can be mounted in a standard 19-in. equipment rack. Figure 10-14 shows an H911 mounting panel containing two H803 connector blocks. The wire-wrap pins of the connector blocks face the front of the mounting panel. Cable 1 from the Positive I/O Bus interface is pictured as connecting into slot A1. This is in accordance with the convention presented previously. Cable 2 and cable 3 (omitted for clarity) would be inserted into slots A2 and A3, respectively.

If a data break peripheral were being used, cable 1 from the KD8-E interface would be inserted into slot A4, while data break cable 2 would be inserted into slot A5. The M-series module is shown as being inserted into slot A6. It may be inserted into any available slot, except B1 through B5. Finally, the peripheral I/O cable is shown as being inserted into slot A7 (again just an arbitrary assignment). The peripheral I/O cable must be equipped with a connector that is compatible with the H803 connector block. Either an M903 connector (for use with shielded Mylar) or an M904 connector (for use with coaxial cable) is recommended.

Slots B1, B2, and B3 should be wired in parallel with A1, A2, and A3 so that the I/O bus can be continued to additional peripheral controls. In general, module layout is primarily a matter of common sense. The convention just given is a standard with DEC and may or may not meet with the user's approval. Customers should, nevertheless, always attempt to make parallel connections of the Positive I/O bus interface cables to facilitate possible future expansion.

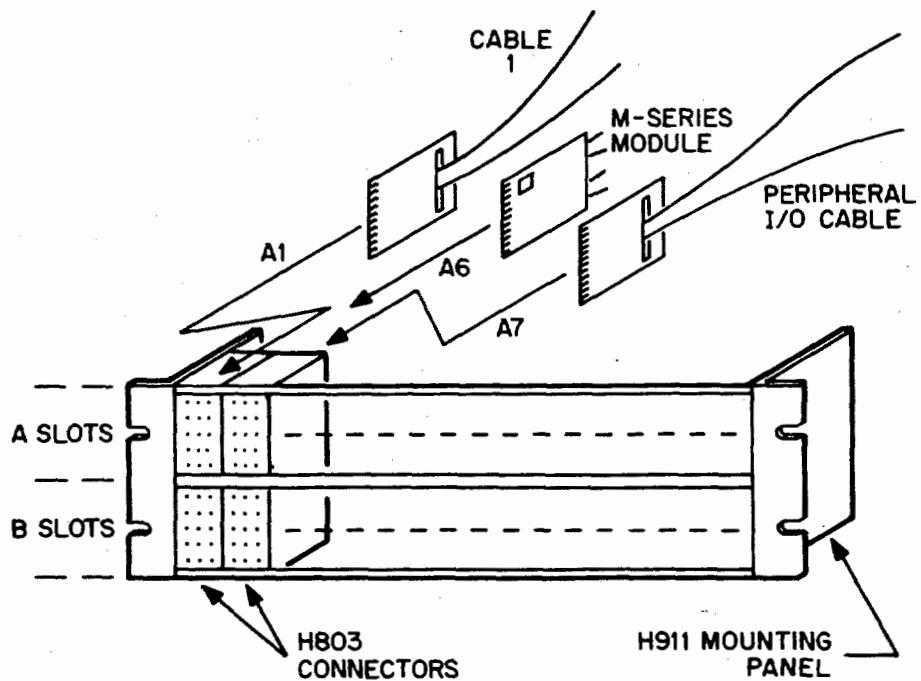


Figure 10-14 Interface Hardware

Connections between the various cable connectors and the M-series modules are made by selective wiring of the H803 connector blocks. The following suggestions and requirements are provided to help reduce the mounting panel wiring time. The connectors should be wired in the order given in steps 1 through 4.

1. The H803 mounting blocks have wire-wrap pins, thereby eliminating the problems associated with soldering, while at the same time providing highly reliable, long lasting connections. DEC recommends #30 AWG, Teflon-coated solid wire for connector block wiring. Smaller wire may be used if many connections are to be made to a single lug. A wire-wrapping tool must be used to wire the connector pins. DEC type H810 pistol grip hand wire-wrapping tool is designed for wrapping #24 or #30 solid wire on Digital-type connector pins (check the DIGITAL LOGIC HANDBOOK for further information about this tool and any other hardware mentioned in this chapter).

#### **CAUTION**

Whenever a wire-wrapping tool is used on a mounting panel containing modules, steps must be taken to avoid voltage transients that can burn out components. A battery-powered or air-operated tool is preferred, but even with these tools static charge can build up and burn out semiconductors. If the modules remain in the connector panel during wiring, ensure that the wire wrap tool is electrically grounded. Whenever soldering is done on a mounting panel containing modules, a 6V soldering iron should be used.

2. Certain connector pins on cable connectors and modules are reserved for specific functions. Cable connector pins are reserved as follows.

Signals: B1, D1, E1, H1, J1, L1, M1, P1, S1,  
D2, E2, H2, K2, M2, P2, S2, T2, V2.

Grounds: A1, C1, F1, K1, N1, R1, T1,  
C2, F2, J2, L2, N2, R2, U2.

Not Used: U1, V1, A2, B2.

Module pins are reserved as follows:

Positive dc voltage: A2 (usually +5V)

Negative dc voltage: B2 (usually -15V)

Ground: C2, T1.

Some form of bus strip (such as DEC 933 Horizontal Bussing Strips) should be used to make all connector power connections and all horizontally bussed signal connections. Negative DC voltage should be wired to pin B2 of module connectors only if the voltage is required by the module.

3. Adequate grounding must be provided. The user should not be concerned with the question of ground loops. At the frequencies dealt with in digital logic, many parallel paths are of utmost importance. There must be ground continuity between cabinets, and between the logic assembly and any equipment with which the logic connects. Continuity between mounting panels and between ground pins on the various connector blocks is achieved when the following instructions have been carried out. These instructions are illustrated in Figure 10-15.

a. Vertical grounding wires must interconnect each chassis ground lug with pin C2 and pin T1 grounds. Start these wires at the uppermost mounting panel and continue to the bottom panel. Begin by connecting C2 pins, then T1 pins, alternating thereafter. Space the wires about two inches apart, so that each of the chassis-ground lugs is in line with one of the wires. Each vertical wire should make three connections at each mounting panel.

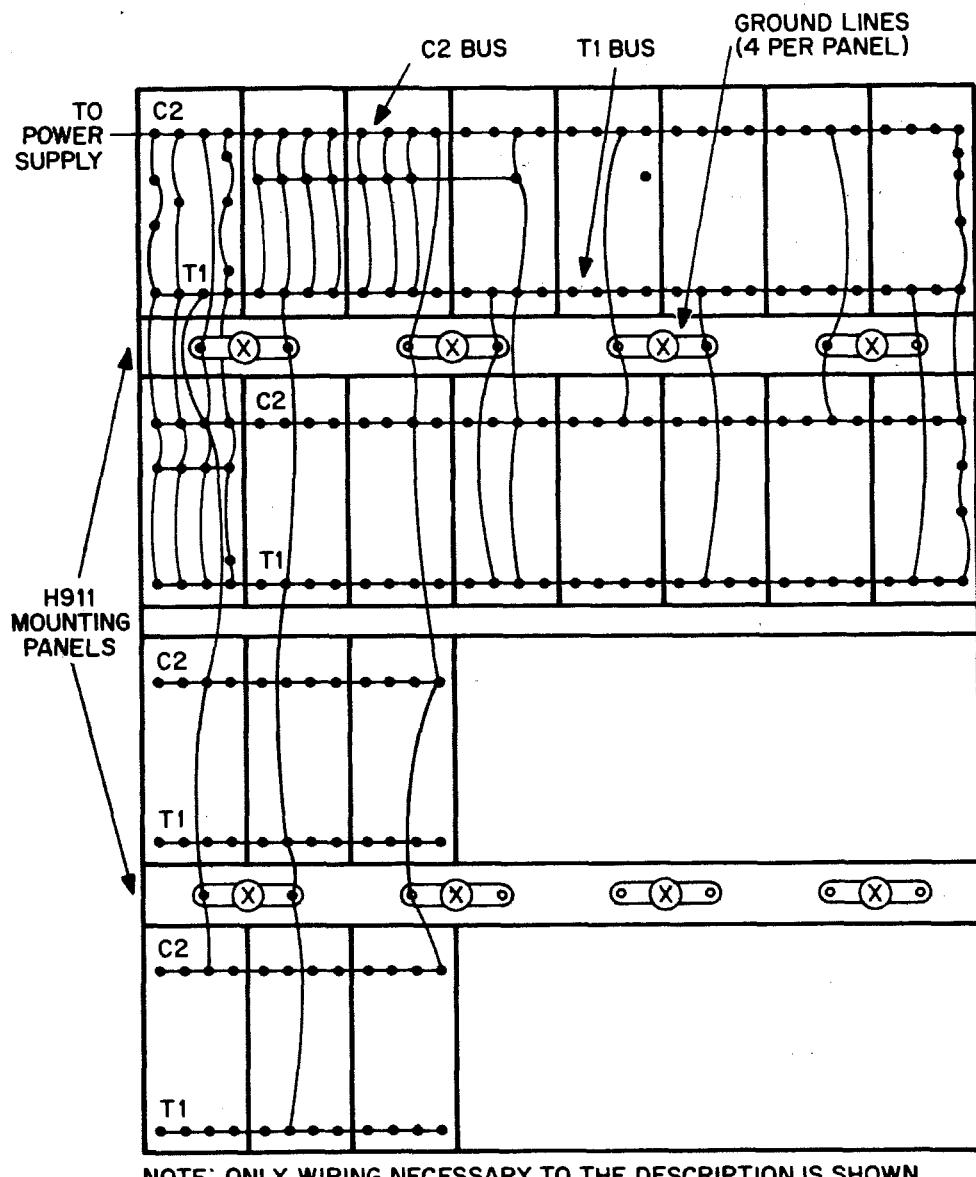
b. Connect pin C2 of each module to T1 of the same module, making connections to all other pins to be grounded along the way. Connect T1 of each module to C2 of the module beneath. Ensure that connections are made to the ground pins on the signal connectors.

c. Bus ground pins horizontally wherever possible.

4. After ground connections have been made, connect all signal wires in any convenient order. Point-to-point wiring produces short wire lengths, installs quickly, is easy to trace and change, and generally results in better appearance and performance than cabled wiring.

Certain restrictions and criteria must be observed when interfacing to the external bus. These are encountered when interfacing both with DEC modules and connector blocks, and with cus-

customer-designed circuits. These requirements are contained in the section dealing with restrictions and criteria. Users are strongly advised to consult this section prior to completing their interface package.



NOTE: ONLY WIRING NECESSARY TO THE DESCRIPTION IS SHOWN.

Figure 10-15 Mounting Panel Wiring

#### **Customer Designed Interfaces**

The customer who elects to design his own interface package must provide both electrical and mechanical interfaces. In addition to the requirements of the following section, the user should keep in mind the following definitions:

1. External bus signals are positive pulses or positive levels, allowing direct TTL-logic interfacing with appropriate diode clamp protection.

2. These positive pulses and levels change from ground (0V to 0.4V) to a positive voltage between 2.4V and 3.6V.
3. All signal lines are loaded within the two external bus interfaces in the PDP-8/E. Signals coming from the peripheral are inactive when a voltage potential is applied to them; conversely, signals going to the peripheral are inactive when no voltage potential is applied to them.

Figure 10-16 provides logic diagrams of the circuits to which the user must interface. The details of the method used are left to the customer.

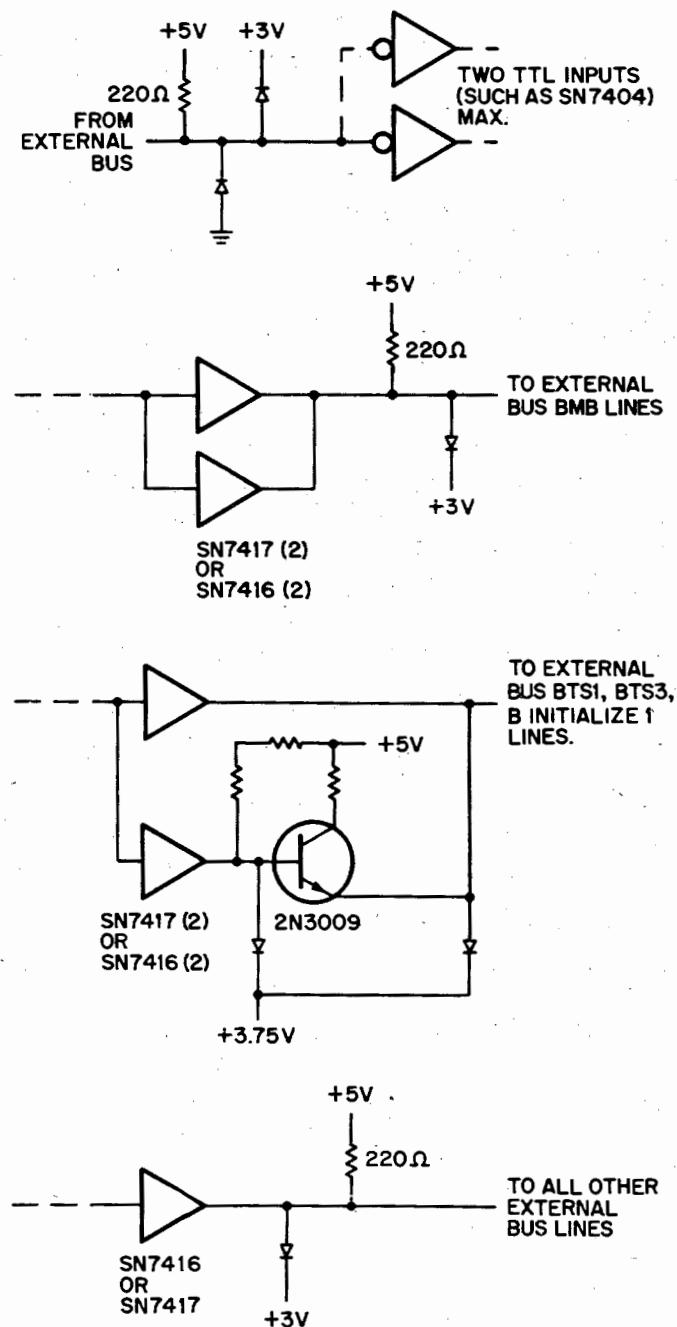


Figure 10-16 Interface Logic Diagrams

## **RESTRICTIONS AND CRITERIA**

### **Cooling**

The low power consumption of M-series modules results in a total dissipation of about 15W in a typical H911 mounting panel containing 64 modules. Convection cooling is sufficient for a few mounting panels, but forced air cooling should be used when a very large system is built.

### **Signal Terminating**

Termination is required on Positive I/O Bus interface cables longer than 20 feet, and may be desirable on shorter cables. The following signals should be shunted to ground by a 100 ohm resistor: IOP1; IOP2; IOP4; BTS1; BTS3; B INITIALIZE 1. If a series of peripherals is being used, the termination is inserted in the last peripheral. A DEC G717 resistor terminator module is available for this purpose.

### **Timing Criteria**

Timing criteria must be considered only when peripherals having high operating speeds (over 5kHz) are being interfaced. The following information concerning interrupt processing must be understood.

- a. The interrupt feature must be turned on via the ION instruction in order for the device to be allowed to interrupt the processor.
- b. In order to honor the interrupt, the central processor must have completed the instruction it is presently doing.
- c. When an interrupt request is honored, the hardware of the machine executes an effective JMS to location 0 in memory field 0, and also disables the interrupt system.
- d. An interrupt servicing routine must be resident in memory and the starting address of this routine must be defined in the memory location immediately following location 0.

The longest time required to honor an interrupt request is approximately the time duration of the slowest instruction. Thus, for a PDP-8/E without the EAE option, this time would be 4.3 microseconds (the time required to complete a 3-cycle instruction). For the PDP-8/E with the EAE option, the time would be 9.8 microseconds. These times assume an interrupt request just after the processor enters the FETCH state.

The following examples illustrate the use of this timing criterion.

#### **EXAMPLE 1—PDP-8/E without EAE option.**

Time between interrupts	:	50.0 $\mu$ s
Maximum processor time before interrupt	:	-4.3 $\mu$ s
Time for hardware JMS to location 0	:	-1.4 $\mu$ s
Maximum time allowed for servicing before possible error arises	:	44.3 $\mu$ s

#### **EXAMPLE 2—PDP-8/E with EAE option.**

Time between interrupts	:	50.0 $\mu$ s
Maximum processor time before interrupt (with EAE option installed—24-bit long shift)	:	-9.8 $\mu$ s

Time for hardware JMS to location 0	: -4.3 $\mu$ s
Maximum time allowed for servicing before possible error arises	: 35.9 $\mu$ s

Another timing criterion is concerned with peripheral gating time from IOP to SKIP, from IOP to AC input signals, and from IOP to AC CLEAR. To avoid time delay problems, these gating times must be limited to 100 nanoseconds.

The third timing criterion is concerned with the delays inherent in interconnecting cabling. DEC logic generates waveforms with rising edges containing frequencies of over 100 MHz. At these frequencies the inductance, mutual inductance, capacitance, and transmission line properties of the external bus cabling become significant. To avoid problems, consider the following when interfacing.

- a. The propagation delay of typical wiring (1.5 nanoseconds/ft) is often significant when overshoot and reflections are considered.
- b. The current carrying capacity of a wire is only  $V/Z(0)$  until the wave has propagated along the wire three times. Typical wiring has a characteristic impedance of approximately 150 ohms, so that the current available at the end of the wire for rising waveforms is only 20 millamps until reflections have propagated, regardless of the source current available.
- c. The inductance and capacitance of wiring combine to produce high frequency ringing on the transitions of waveforms. This ringing can be controlled by resistively terminating the line with approximately 100 ohms.
- d. The mutual inductance and capacitance of the wiring causes high-frequency crosstalk which may produce false operation of the logic. This crosstalk can be reduced in one of the following ways; minimizing the number of high frequency signal components by clipping or clamping high-frequency ringing with a level terminator circuit, or wiring with short wires and/or twisted pairs, thereby reducing coupling. Clamping can also be used to prevent the excursion of the output or input voltages beyond certain predetermined limits. This is sometimes necessary to prevent false triggering or electrical damage to gates.

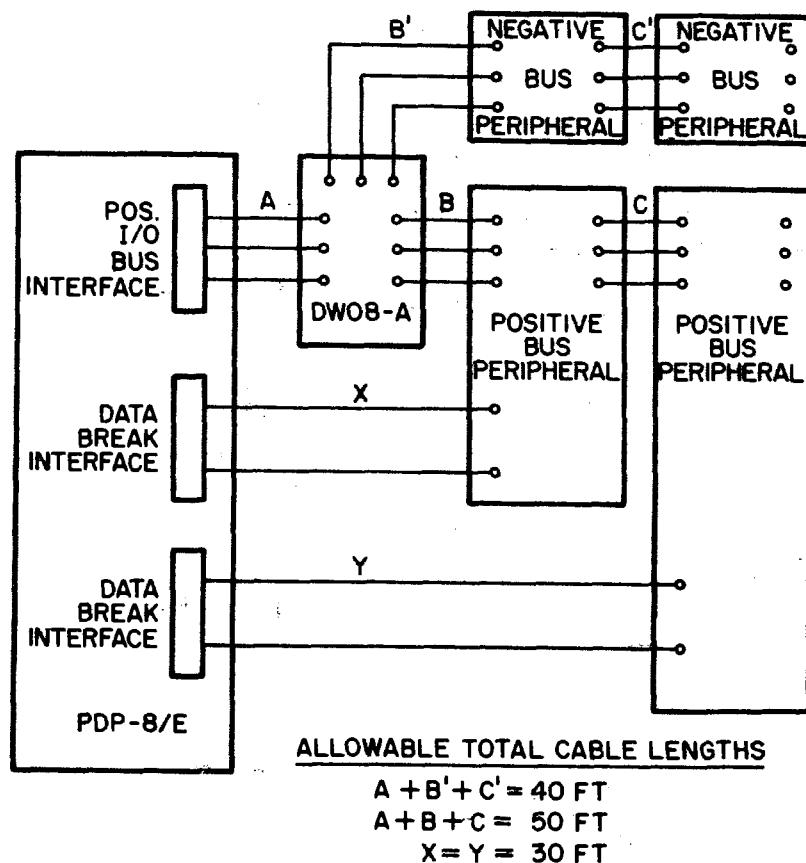
#### CABLING RULES AND SUGGESTIONS

1. Round and flat coaxial cable are electrically interchangeable and may be intermixed in a system. If cables will be subjected to extraordinary abuse, round coaxial cable is preferable when connecting free-standing cabinets.
2. Indiscriminate mixing of shielded flat cable and coaxial cable is not advised. DEC recommends that all cables be shielded flat cables, except when the user is trying to gain maximum length or is connecting free-standing cabinets. Not more than one change from BC08-J to coaxial, or vice-versa, should be made over the length of a bus.
3. The following cable length restrictions should be observed:

CABLE	TYPE	MAXIMUM LENGTH
1, 2, and 3 from the Positive I/O Bus interface	Coaxial	50 ft

If a DW08-A I/O converter panel is connected onto cables 1, 2, and 3, the system can accommodate negative bus peripherals. The normal positive bus maximum cable lengths remain as indicated; the maximum cable lengths for the converted bus (negative) are 10 feet shorter than that indicated. See Figure 10-17.

1 and 2 from the Data break interface	Coaxial	30 ft
---------------------------------------	---------	-------



NOTE: LENGTHS GIVEN FOR COAXIAL CABLE; EACH IS A MAXIMUM ALLOWABLE LENGTH.

Figure 10-17 Maximum Bus Lengths

4. DEC cable has the nominal characteristics listed below. The user should ensure that whatever cable he uses exhibits approximately the same characteristics.

$Z = 95 + or - 5 \text{ ohms}$   
 $C = 13.75 \text{ pF/ft (unterminated)}$   
 $L = 124 \text{ nH/ft}$   
 $R = 0.095 \text{ ohm/ft}$   
 $V(p) = 1.5 \text{ ns/ft}$

5. The cables supplied with the Positive I/O Bus interface and the Data Break interface can be obtained in standard length, as outlined in table 10-3.

**Table 10-3 Table Of Standard Cable Lengths**

CABLE	LENGTH
BC08J-6	6 feet
BC08J-10	10 feet

6. Cabling arrangements which cause the bus not to be a single transmission line driven at one end by the Positive I/O Bus Interface and Data Break Interface are forbidden. "T" connections to the bus for driving peripherals should be no longer than 1 foot, including the backplane wiring to the final module in the interface.

## **SECTION 2 OMNIBUS INTERFACING USING "OFF THE SHELF" MODULES**

Interfacing to the PDP-8/E OMNIBUS can be accomplished conveniently by utilizing M Series driver and receiver modules. The low-leakage current requirements for devices "wire ORed" to the OMNIBUS signals are met completely when the M783, M784, and M785 modules are used.

This approach takes advantage of Digital's broad line of proven interface modules. Customers who are designing new interfaces or who have existing interfaces using TTL Logic levels may tie them directly to the OMNIBUS via this procedure. The attractiveness of this procedure lies in the availability of fully engineered and warranted modules.

### **OMNIBUS SIGNAL SUMMARY:**

Most OMNIBUS lines are considered by the system to be inactive (voltage level high) until the line level is pulled to ground. Logic levels on these lines are defined as:

Logic 1—Max. Voltage: 0.4 V  
Min. Voltage: —0.5 V  
Logic 0—Max. Voltage: 5.0 V  
Min. Voltage: 3.0 V

Signal levels on the OMNIBUS may be converted to or from TTL levels with the following modules:

Bus Receiver —M784  
Bus Driver —M783  
Bus Transceiver —M785

### **THE "BUILDING BLOCK" APPROACH**

A block diagram illustrating a system relationship is shown in figure 10-18. The M783 and M784 Driver and Receiver or the counterpart—the M785 Transceiver are shown as the necessary prerequisites to establish OMNIBUS compatibility. The connecting link between the OMNIBUS and the interfacing M modules is with two M935 Bus connectors which interconnect the last slot of the OMNIBUS to the first slot of the H9190 assembly. The user interfacing options are immediately expanded by interconnecting the slots on the H9190 assembly to Bus Drivers and Bus Receivers and interconnecting the Bus Drivers and Bus Receiver modules to other M or K series modules. There are more than fifty M series modules to choose from and scores of K series modules.

### **M783—BUS DRIVERS**

The M783 module (represented in figure 10-19) consists of 12 two-input NAND gates with open-collector outputs. The gates are grouped into a set of 8 with a common enable line and 4 individual gates. Each output is capable of sinking 50 mA while maintaining a collector voltage of  $\leq 0.8V$ . The output leakage current is  $< 25 \mu A$ . All gate inputs are TTL compatible.

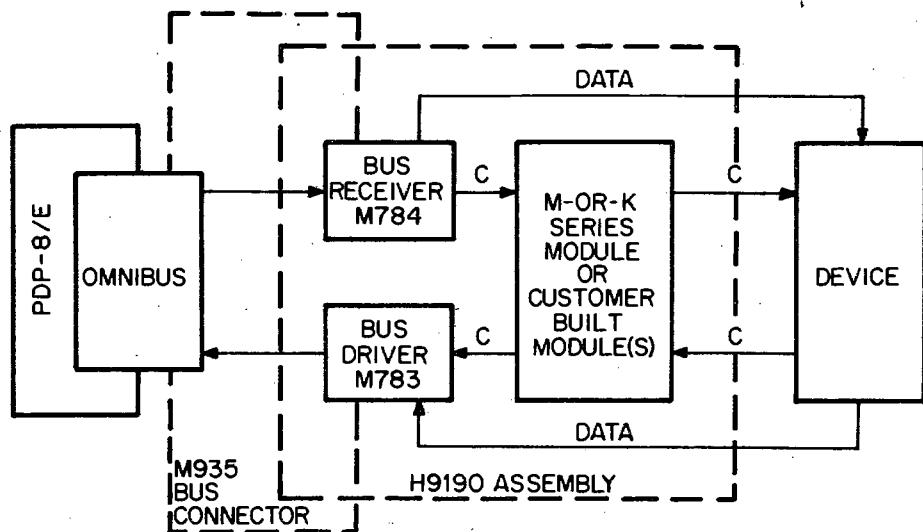


Figure 10-18 Typical System Interface Block Diagram

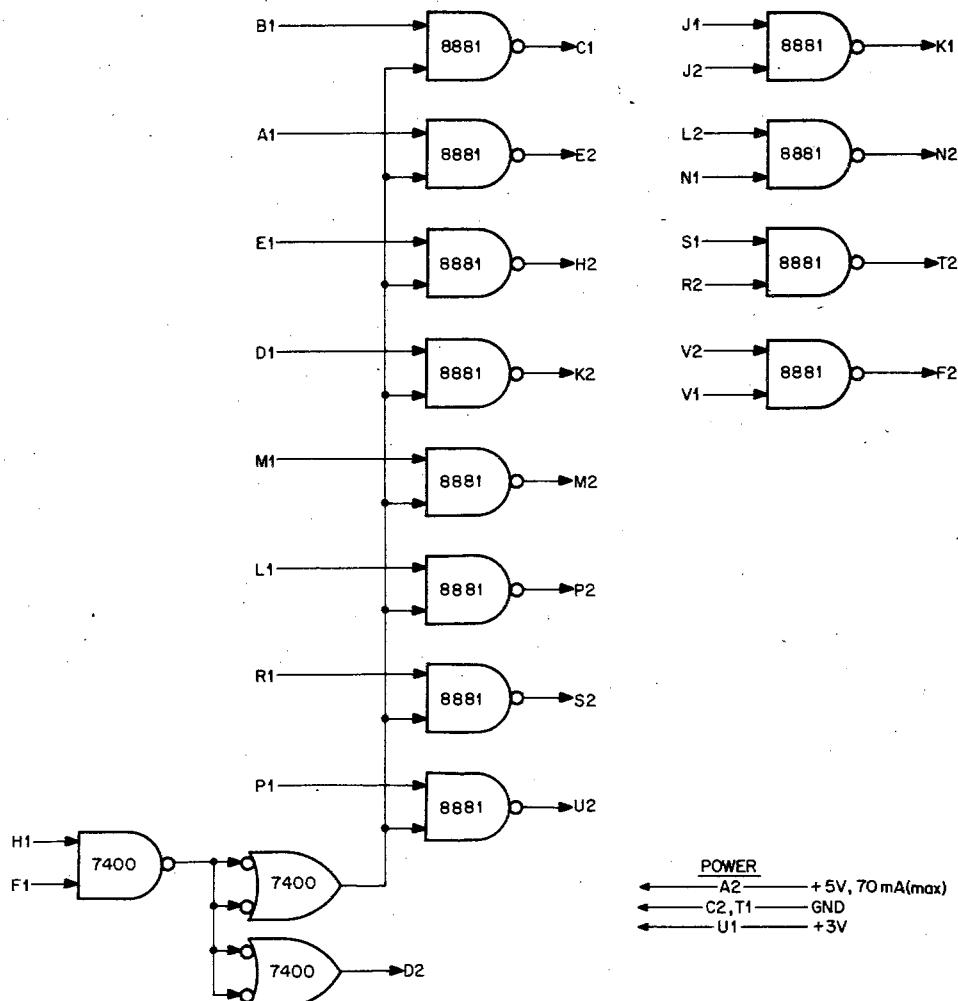


Figure 10-19 M783 Bus Drivers

## M784—BUS RECEIVERS

The M784 module (represented in figure 10-20) has 16 inverting receiver circuits constructed of two-input NOR gates with the common enable line grounded. Inputs are characterized as:

Low Level:  $< 1.4 \text{ V}$  at  $25 \mu\text{A}$  (max)

High Level:  $> 2.5 \text{ V}$  at  $160 \mu\text{A}$  (max)

All gate outputs are TTL compatible with a fan-out from each of 7 TTL loads.\*

\*One unit load is defined as:

Logic 0—sink  $1.6\text{mA}$  with  $V_{out} \leq 0.4 \text{ V}$

Logic 1—supply  $40.0 \mu\text{A}$  with  $V_{out} \geq 2.4 \text{ V}$

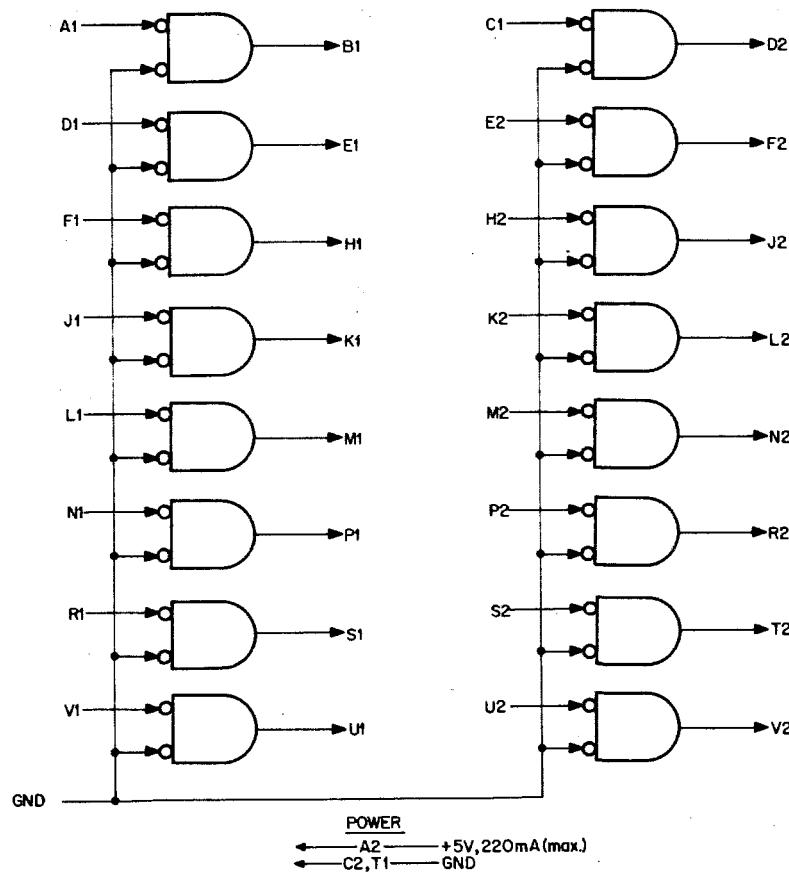


Figure 10-20 M784 Bus Receivers

## M785—BUS TRANSCEIVER

This composite module consists of 8 drivers and 8 receivers. Each set of 8 gates has a common enable line, convenient for strobing data to and from the OMNIBUS. The loading characteristics of the devices are identical to their M783 and M784 counterparts.

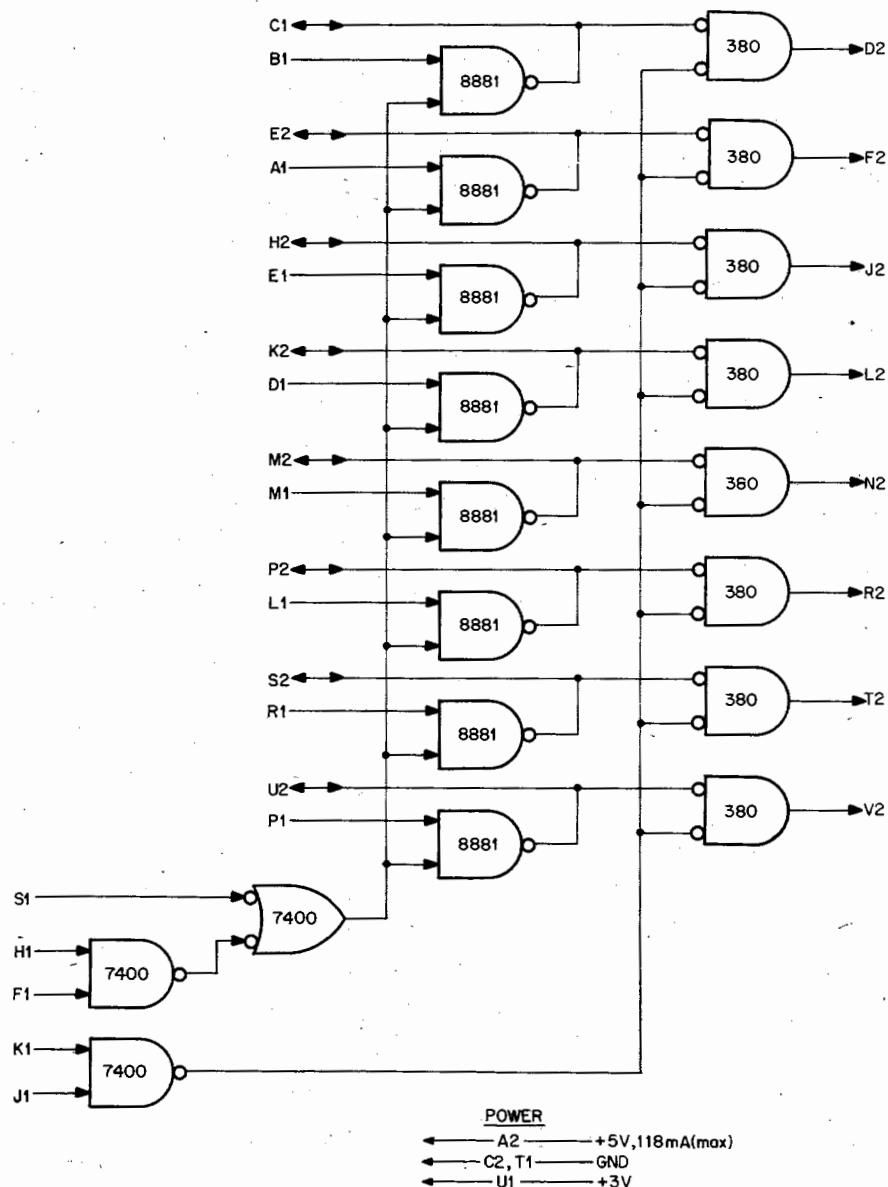
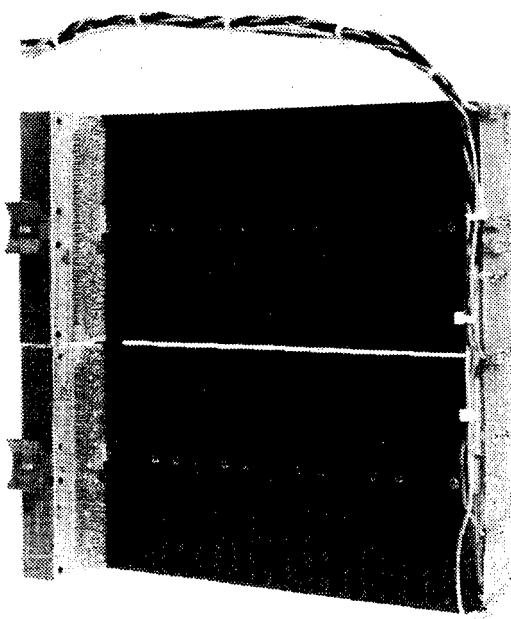
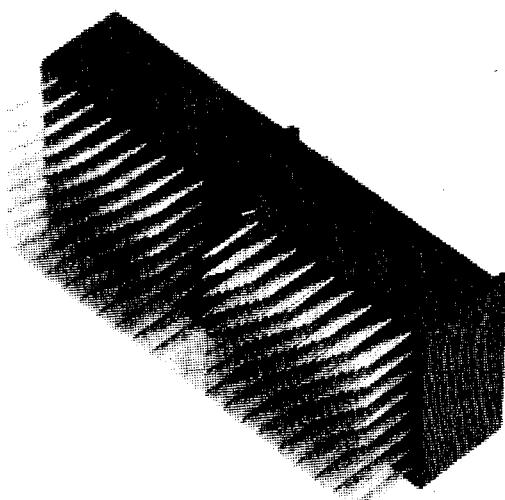


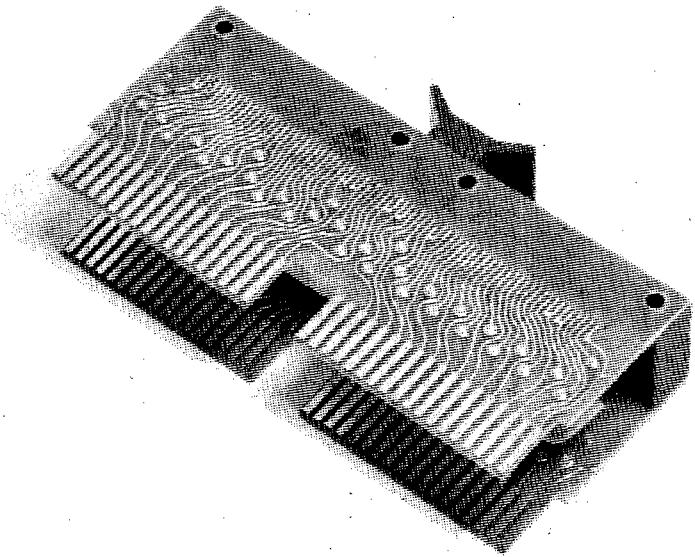
Figure 10-21 M785 Bus Transceiver



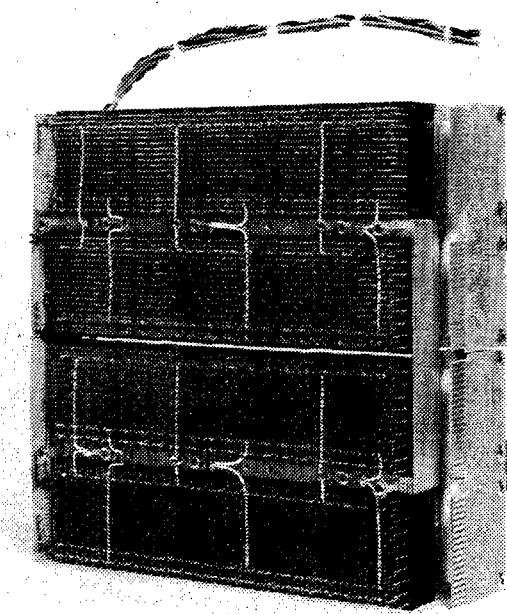
**H9190 M935 Kit**—contains the H9190 assembly with M Series connector blocks for standard M Series modules, power wiring harness, and power bus board. It includes M Series power bussing for all but the four lots in the first column. Also included are two M935 bus connectors. Four mounting spacers allow the H9190 to be easily mounted in the second half of an 8/E chassis.



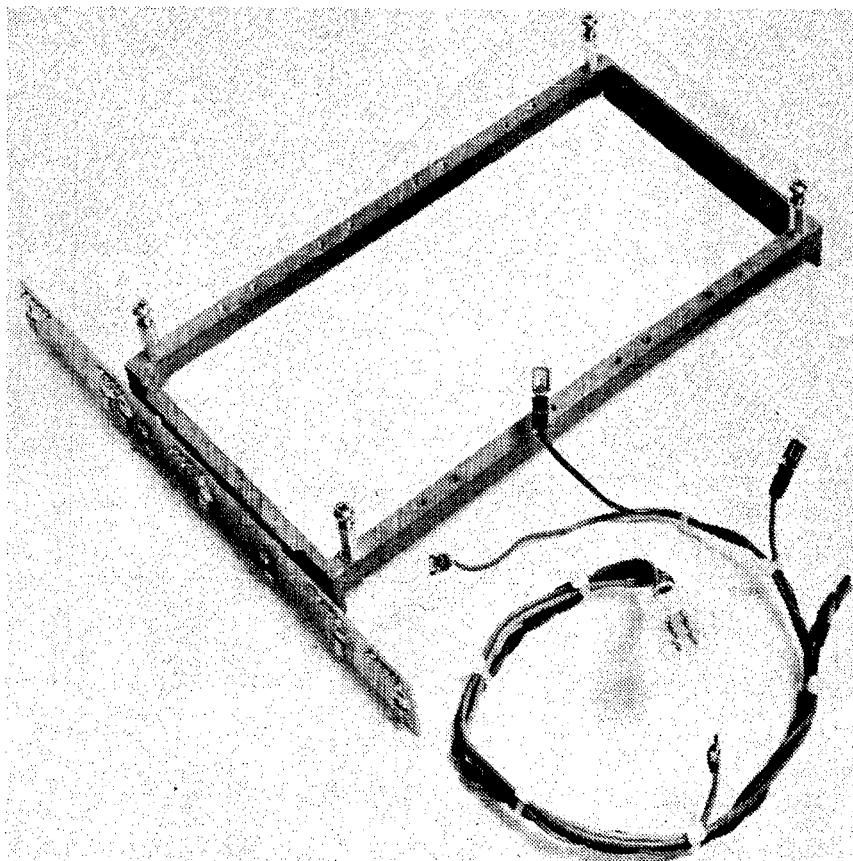
**H803 Connector Block**—a high density, 8-slot connector block with wire wrap pins. This connector is designed to be used with M Series modules.



**M935 Bus Connector**—used to interconnect 8/E assemblies. The H9190 may be connected to the 8/E OMNIBUS using two M935's.



**H9190 Mounting Panel**—contains M Series connector blocks with 8/E type packaging for standard M Series modules. Also included are the 8/E power wiring harness and power bus board. There is M Series power bussing for all but the four slots in the first column. Four mounting spacers allow the H9190 to be easily mounted in the second half of an 8/E chassis.



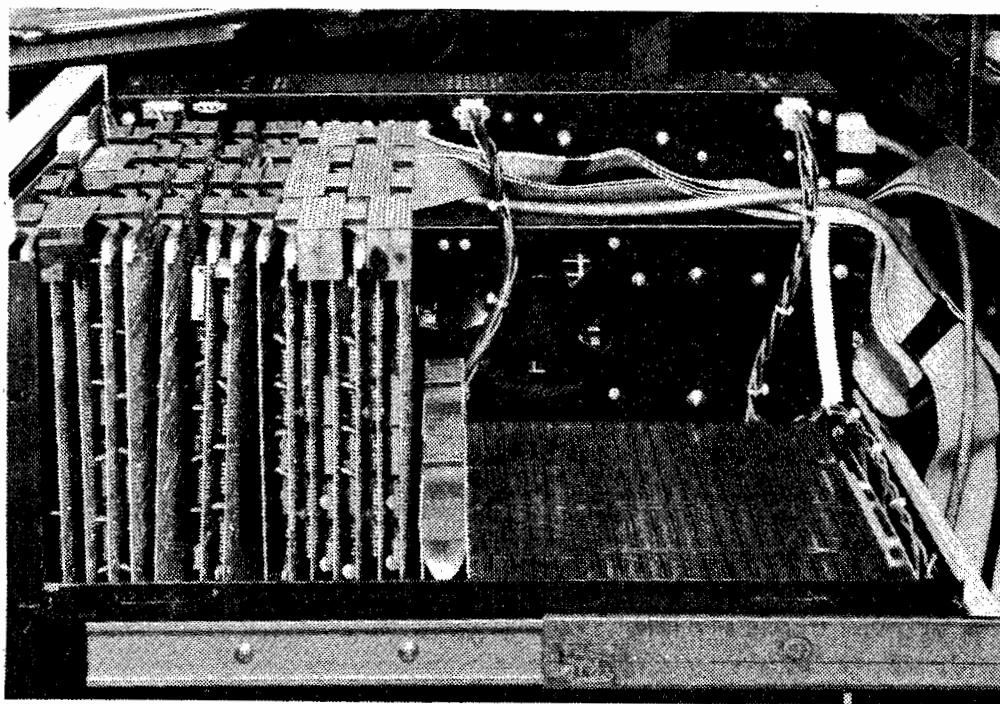
**H019 Mounting Bar**—an aluminum casting with the power bus board and power wiring harness. It also includes four mounting spacers for mounting in an 8/E chassis. Up to ten connector blocks of any type may be accommodated by this frame.

#### PHYSICAL PLACEMENT OF INTERFACE MODULES

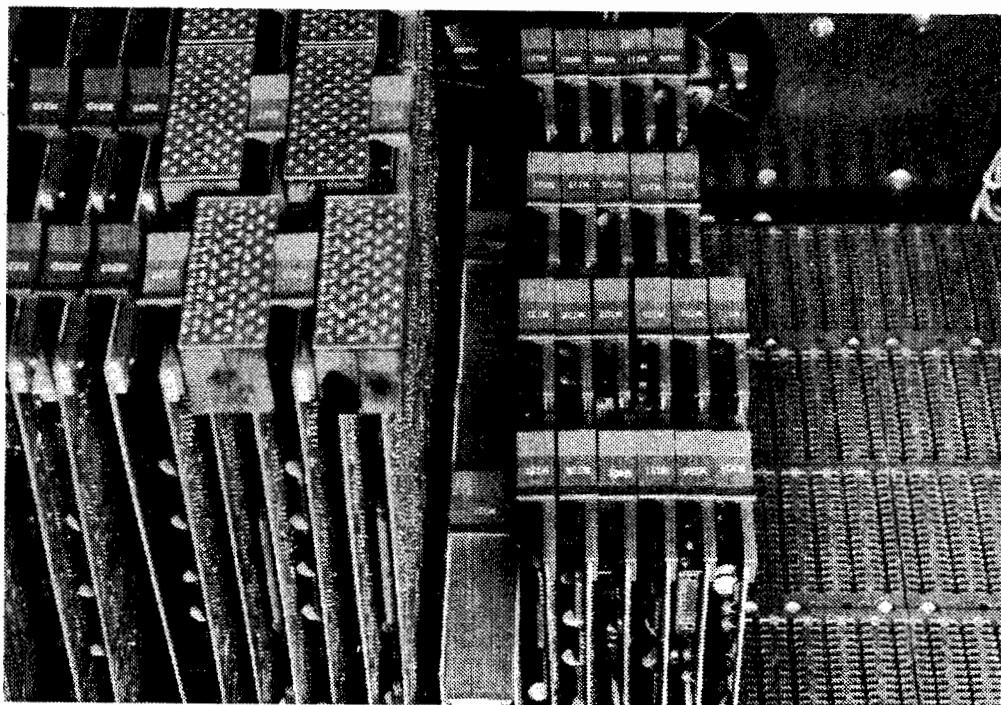
All pins of the OMNIBUS back-panel are dedicated to specific signal lines. For this reason any interface modules that are not pin-compatible cannot be inserted directly into the OMNIBUS. These modules should be plugged into type H803 connector blocks mounted externally to the OMNIBUS. This does not imply mounting externally to the PDP-8/E box, however. There is normally adequate mounting space available within the box itself for medium sized interfaces.

There are two general methods of interconnection to the OMNIBUS from the external logic. One method is to use a H9190 Mounting Panel connected to the OMNIBUS via 2 M935 bus connectors. This panel contains 10 H803 connectors and is physically similar to the BE8-A OMNIBUS Expander assembly, the difference being that the back-panel is wire wrappable rather than being bussed. Mechanical mounting is to existing BE8-A supports within the PDP-8/e box. A power cable is provided for connection to the PDP-8/E power supply. This supply was designed to support internal interfaces. It can supply up to 13 amps at +5 Vdc, 3.5 amps at -15 Vdc and 0.2 amps at + 15 Vdc.

The H019 Mounting Bar will mount up to 10 H803 connector blocks. It mounts in the same manner as the H9190 and includes the power cable.



The H9190 Assembly shown connected to the OMNIBUS via the  
M935 Bus connector



The H9190 Assembly shown with 21 M series modules and many  
unused slots.

A second method of connection is to mount the H9190 in a BA8-AB expander box. Connection to the OMNIBUS in this case is via type BC08H-3F flexprint cables. Use of this mounting method is necessary only with large systems that mount the BE8-A OMNIBUS Expander in the 8/E box.

# **INTERFACE EXAMPLE PAPER TAPE READER**

The objective is to allow a PDP-8/E to read 8-bit code from a paper tape reader. The design utilizes the computer's Interrupt and Skip facility in order to minimize the time required to service the reader. All logic functions are performed using "off the shelf" M Series modules.

## **Input/Output Transfer (IOT) Instruction Usage**

- IOT 6641—Skip if RDR FLAG set by DATA STROBE
  - IOT 6642—Load Data onto DATA and C0-C1 lines and reset RDR FLAG: (e.g. load data into AC and clear interrupt request)
  - IOT 6643—Reset RDR FLAG (e.g. clear interrupt request)
  - IOT 6644—Unused
  - IOT 6645—Unused
  - IOT 6646—Set RDR RUN flip-flop (e.g. initiate read char.)

## **System Operation**

Initially the RDR FLAG is reset by IOT 6643 to clear the interrupt line. IOT 6646 initiates a chain of operations that issues the reader motor drive signals and times the DATA STROBE to load hole sense data into the interface register. The DATA STROBE signal also sets the RDR FLAG which generates an interrupt to the 8/E. The condition of RDR FLAG can be monitored by using IOT 6641 as a Skip IOT. Once the RDR FLAG is set, IOT 6642 will load the register data into the accumulator of the 8/E and clear the interrupt line. The above procedure is repeated for reading of each character from the tape.

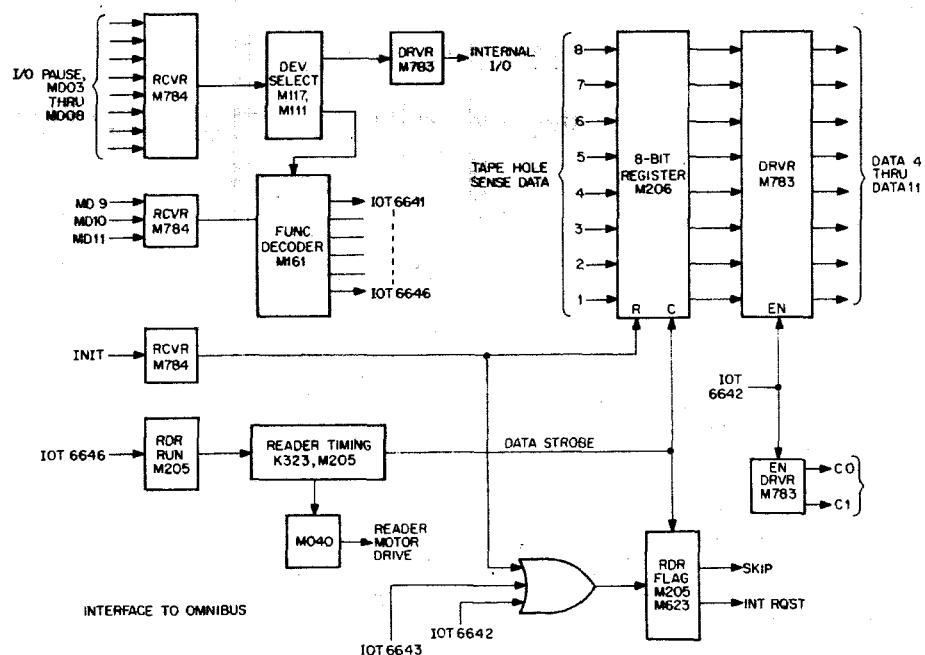


Figure 10-21 Block Diagram Paper Tape Reader Interface to OMNIBUS

## M-SERIES MODULE SUMMARY

The M Series TTL integrated circuit modules consists of more than 95 modules ranging from basic and functional logic modules to self-contained computer interfacing modules for applications such as instrumentation, computer interfacing, data gathering, control, etc.

DEC is also offering a new M Series Logic Lab for use in breadboarding M Series logic designs. This new M Series Logic Lab is used in education as a training device which offers the user an easy step-by-step method to gain an understanding of various logic functions such as AND, OR, NAND, NOR, etc. The breadboard and testing capability of the Logic Lab is an effective tool for bridging the gap between paper design and a fully tested, marketable product. For detailed information, the reader should acquire a free copy of DEC's 300 page Logic Handbook. Please write to Direct Mail, Digital Equipment Corporation, 146 Main Street, Maynard, Massachusetts 01754 and ask for a copy of the Logic Handbook if you do not already have one.

The following is a partial list of M Series modules available from Digital Equipment Corporation that can be used in designing special interfaces and special devices. The majority of these modules are described in DEC's DIGITAL LOGIC HANDBOOK. For modules that cannot be found in the handbook, contact the nearest Digital representative for information.

TYPE	FUNCTION	DESCRIPTION
M002	15 Loads	Fifteen +3V sources each capable of driving 10 unit loads. Can be used for tying off unused inputs.
M040	Solenoid Driver	Output ratings of -70V and 0.6A allow these 2 drivers to be used with a variety of medium current loads.
M050	50 ma Indicator and Driver	Output ratings of -20V and 50 millamps allow any of the 12 circuits on this module to drive a variety of incandescent lamps. These drivers can also be used as slow speed open collector PNP level shifters to -3V system.
M101	Bus Data Interface	Fifteen two-input NAND gates with one input of each gate tied to a common line. For use in strobing data from the PDP-8/I, PDP-8/L, or PDP-8/E I/O bus. Pins are compatible with the M111 module.
M103	Device Selector	Diode gate, buffering and clamping circuits necessary to decode IOT's from the PDP-8 I, PDP-8/L, or PDP 8/E positive bus. Output pulses are not regenerated, only buffered.
M111	Inverter	Sixteen inverter circuits with a fan-in of 1 unit load and fan-out of 10 unit loads.

TYPE	FUNCTION	DESCRIPTION
M112	NOR Gate	Ten positive NOR gates with a fan-in of 1 unit load and fan-out of 10 unit loads.
M113	Ten 2-Input NAND Gates	Ten 2-input positive NAND gates with a fan-in of 1 unit load and fan-out of 10 unit loads.
M115	Eight 3-input NAND Gates	Eight 3-input positive NAND gates with a fan-in of 1 unit load and a fan-out of 10 unit loads.
M117	Six 4-Input NAND Gates	Six 4-input positive NAND gates with a fan-in of 1 unit load and a fan-out of 10 unit loads.
M119	Three 8-Input NAND Gates	Three 8-input positive NAND gates with a fan-in of 1 unit load and a fan-out of 10 unit loads.
M121	AND/NOR Gates	Six gates that perform the positive logic function $AB + CD$ . Fan-in on each input is 1 unit load and gate fan-out is 10 unit loads.
M141	NAND/OR Gates	Twelve 2-input positive NAND gates that can be used in a wired OR manner. Gates are grouped in a 4-4-3-1 configuration with a fan-in of 1 unit load and a fan-out that depends on the number of gates ORed together.
M160	Gate Module	Three general purpose multi-input gates that can be used for system input selection. Fan-in is 1 unit load and fan-out is 10 unit loads.
M161	Binary to Octal/Decimal Decoder	A binary to 8-line or BCD to 10-line decoder. Gating is provided so that up to 6 binary bits can be decoded using only M161s. Accepts a variety of BCD codes.
M162	Parity Circuit	Two circuits, each of which can be used to generate even or odd parity signals for four bits of binary input.
M169	Gating Module	Four circuits that can be used for input selection. Each circuit is of an AND/OR configuration with four 2-input AND gates.
M202	Triple J-K Flip-flop	Three J-K flip-flops with multiple input AND gates on J and K. Versatile units for many control or counter purposes. All direct set and clear inputs are available on module pins.

TYPE	FUNCTION	DESCRIPTION
M203	Set-Reset Flip-flops	Eight single input set-reset flip-flops for use as buffer storage. Each circuit has a fan-in of 1 unit load and a fan-out of 10 unit loads.
M204	Counter-Buffer	Four J-K flip-flops that can be interconnected as a ripple or synchronous counter or used as general control elements.
M206	Six Flip-flops	6 D-type flip-flops that can be used in shift registers, counters, buffer registers, and general purpose control functions.
M207	Flip-flops	Six single-input J-K flip-flops for use as shift registers, ripple counters, and general purpose control functions.
M208	Buffer Shift	An internally connected 8-bit buffer or shift register. Provisions are made for gated single-ended parallel load, bipolar parallel output, and serial input.
M211	Binary Up/Down Counter	A 6-bit binary up/down ripple counter with control gates for direction changes via a single control line.
M212	6-Bit Shift Register	An internally connected left-right shift register. Provisions are made for gated single-ended parallel load, bipolar parallel output, and serial input.
M213	BCD Up/Down Counter	One decade of 8421 up or down counting is possible with this module. Provisions are made for parallel loading, bipolar output, and carry features.
M230	Binary to BCD Shift Register Converter	One decade of a modified shift register that allows high speed conversion (100 nanoseconds per binary bit) of binary data to 8421 BCD code. System use of this module requires additional modules.
M302	One Shot Delay	Two pulse-or-level-triggered one-shot delays with output delay adjustable from 50 nanoseconds to 7.5 milliseconds. Fan-in is 2 unit loads and fan-out is 25 unit loads.
M310	Delay Line	Fixed tapped delay line with delay adjustable in 50 nanoseconds increments from 50 nanoseconds to 500 nanoseconds. Two digital output amplifiers and one driver are included.

TYPE	FUNCTION	DESCRIPTION
M360	Variable Delay	Continuously variable delay line with a range of 50 nanoseconds to 500 nanoseconds. Module includes delay line drivers and digital output amplifiers.
M401	Clock	A gateable RC clock with both positive and negative pulse outputs. The output frequency is adjustable from 10 MHz to below 100 Hz.
M405	Crystal Clock	Stable system clock frequencies from 1 kHz to 10 MHz are available with this module. Frequency drift at either the positive or negative pulse output is less than 0.01 percent of the specified frequency.
M410	Reed Clock	A stable low frequency reed clock similar to the M452. Stability in the range 10 degrees C to 70 degrees C is better than 0.15 percent. For use with communications systems and available with only standard teletype and data set frequencies.
M452	Variable Clock	Provides 880Hz, 440Hz, and 220Hz square waves necessary for clocking and for the M706 and M707 modules in a 110-baud teletype system.
M501	Schmitt Trigger	Provides regenerative characteristics necessary for switch filtering, pulse shaping, and contact closure sensing. This circuit can be AND/OR expanded.
M502	Negative Input Converter	Pulses as short as 35 nanoseconds can be level shifted from -3V systems to standard M-Series levels by the two circuits in this converter. This module can also drive low impedance terminated cables.
M506	Negative Input Converter	This converter levels shift pulses as short as 100 nanoseconds from -3V systems to M-Series levels. Each of the 6 circuits on this module has a 10 milliamp load resistor on the negative input.
M507	Bus Converter	Six inverting level shifters that accept -3V and GND as inputs and have an open collector NPN transistor at the output. Output rise is delayed by 100 nanoseconds for pulse spreading.

TYPE	FUNCTION	DESCRIPTION
M516	Positive Bus	Six 4-input NOR gates with overshoot and undershoot clamps on one input of each gate. In addition, one input of each gate is tied to +3V with the lead brought out to a connector pin.
M602	Pulse Generator	The two pulse amplifiers in this module provide standard 50 nanoseconds or 110 nanoseconds pulses for M-Series systems.
M617	Six 4 Input NOR Buffers	Six 4-input positive NOR gates with a fan-in of 1 unit load and a fan-out of 30 unit loads.
M627	Power Amplifier	Six 4-input high speed positive NAND gates with a fan-in of 2.5 unit loads and a fan-out of 40 unit loads.
M650	Negative Output	The three non-inverting level shifters on this module can be used to interface the positive levels or pulses (duration greater than 100 nanoseconds) of K- and M-Series to -3V logic systems.
M652	Negative Output Converter	These two circuits provide high-speed, non-inverting level shifting for pulses as short as 35 nanoseconds or levels from M-Series to -3V systems. The output can drive low impedance terminated cables.
M660	Positive Level Driver	Three circuits provide low impedance, 100-ohm, terminated cable driving capability, using M-Series levels or pulses of duration greater than 100 nanoseconds. Output drive capability is 50 milliamps at +3V or ground.
M661	Positive Level Driver	Three circuits provide low-impedance unterminated cable driving. Characteristics are similar to M660 with the exception that +3V drive is 5 milliamps.
M730	8 Bus Positive Output Interface	General-purpose positive bus output module for use in interfacing many positive level (0 to +20V) systems to the PDP-8/I, PDP-8/L, or PDP-8/E. Module includes device selector, 12 bit parallel output buffer, and adjustable timing pulses.
M731	8 Bus Negative Output Interface	Identical to M730, except that outputs are level shifted for 0 to -20V systems.

TYPE	FUNCTION	DESCRIPTION
M732	8 Bus Positive Input Interface	General purpose positive bus input module for interfacing many positive level (0 to +20V) systems to the PDP-8/I, PDP-8/L, or PDP-8/E. Module includes device selector, 12 bit parallel input buffer, and adjustable timing pulses.
M733	8 Bus Negative Input Interfacer	Identical to M732 except that inputs are level shifted from negative voltage systems.
M734	I/O Bus Input Multiplexer	The M734 is a double height, single width module and is a three-word multiplexer used for strobing twelve-bit words on the positive voltage input bus; usually the input of the PDP8/I or the PDP8/L. Device selector gating is provided. The data outputs of the M734 Multiplexer consist of open collector npn transistors which allow these outputs to be directly connected to the bus. All inputs present one TTL unit load and function as follows:
M735	I/O Bus Transfer Register	The M735 provides one 12-bit input bus driver and one 12-bit output buffer register for input and output data transfers on the positive I/O bus of either a PDP8/I or a PDP8/L. Device selector gating plus additional signal lines provide the flexibility necessary for a complete interface with the exception of flag sense signals. Use of the M735 is not restricted to a computer, as it can be used in many systems to provide reception and transmission of data over cables.
M737	12-Bit Bus Receiver Interface	The M737 was designed primarily to receive and store in a buffer register twelve parallel data bits from the positive bus of the PDP-8/I or PDP-8/L. The M737 is pin compatible with the M738 Counter-Buffer Interface, the M107 Device Selector, the M108 Flag Module, and the 12-Bit Bus Paneloid E100. The 12-bit Bus Receiver Interface, M737, consists of three basic sections: device selector, flag, and buffer register section.
M738	Counter-Buffer Interface	The M738 was designed primarily to strobe twelve parallel bits onto the positive bus of the PDP-8/I or PDP-8/L. This module consists of three basic sections:

TYPE	FUNCTION	DESCRIPTION
		1) A twelve-bit bus driver, 2) A twelve-bit Up Counter which is presetable by jam transfer, and 3) A clock input gate circuit twelve-bit bus driver.
M783	Bus Drivers	The M783 consists of 12 two-input NAND gates with open-collector outputs. The gates are grouped into a set of 8 with a common enable line and 4 individual gates. Each output is capable of sinking 50 mA while maintaining a collector voltage of $\leq 0.8V$ . The output leakage current is $< 25 \mu A$ . All gate inputs are TTL compatible.
M784	Bus Receivers	This module has 16 inverting receiver circuits constructed of two-input NOR gates with the common enable line grounded. Inputs are characterized as:  Low Level: $< 1.4 V$ at $25 \mu A$ (max) High Level: $> 2.5 V$ at $160 \mu A$ (max)  All gate outputs are TTL compatible with a fan-out from each of 7 TTL loads.*
M785	Bus Transceiver	This composite module consists of 8 drivers and 8 receivers. Each set of 8 gates has a common enable line, convenient for strobing data to and from the OMNIBUS. The loading characteristics of the devices are identical to their M783 and M784 counterparts.
M901	Flexprint Cable Connector	Double-sided 36-pin shielded mylar cable connector. All pins are available for signals or grounds. Pins A2, B2, U1, and V1 have 10 ohm resistors in series.
M902	Resistor Terminator	Double-sided 36-pin terminator module with 100 ohm terminations on signal leads. Alternate grounds are provided as in the M903 and M904.
M903	Connector	Double-sided 36-pin shielded mylar cable connector with alternate grounds for I/O bus cables.
M906	Cable	Eighteen load resistors clamped to prevent excursions beyond $+3V$ and ground. This terminator can be used in conjunction with the M623 to provide cable driving ability.

## **K SERIES MODULES**

Another very important variety of "off-the-shelf" modules is the K series module. These are used in, but not limited to control applications. The number of applications using these modules runs into the hundreds. Representative applications include:

- Computer Based Data Aquisition
- Computer Based Control Systems
- Multiprocessor Systems
- Industrial Data Acquisition and Control
- Analog-to-Digital Conversion and Multiplexer Subsystems
- Digital Input and Output Subsystems
- Gas Chromatography Systems
- N/C Tape Preparation Systems

The combination of the M and K series modules using the "building block" approach with "off-the-shelf" modules is an ideal method of interfacing to the PDP-8/E processor for control applications. For more information and detailed examples, the reader should acquire a free copy of DEC's Control Handbook containing more than 200 pages of instructive material in the field of industrial control.

# appendix

## SPARE PARTS LIST

### PDP8E CUSTOMER RECOMMENDED SPARES

#### Option SP8-EA Kit (First Level) (4K)

DEC PART NO.	DESCRIPTION
M8300	Major Register
M8310	Register Control
M8330	Timing Module
G104	Sense/Inhibit Module
G227	X/Y Driver Module
12-05941	Slide Switch
12-05375	Slide Switch, Momentary
12-5849-13	Handle, Terra Cotta
12-5849-12	Handle, Amber
12-9219	Indicator Bulb
70-06994	Key Switch Assy.
54-9264	Power Supply Module
54-9262	Power Supply Module

#### SP8-EA KIT

#### Option SP8-EC Kit (First Level) (8K)

DEC PART NO.	DESCRIPTION
M8300	Major Register
M8310	Register Control
M8330	Timing Module
G111	Sense/Inhibit Module
G233	X/Y Driver Module
12-05941	Slide Switch
12-05375	Slide Switch, Momentary
12-5849-13	Handle, Terra Cotta
12-5849-12	Handle, Amber
12-9219	Indicator Bulb
70-06994	Key Switch Assy.
54-9264	Power Supply Module
54-9262	Power Supply Module

#### SP8-EC KIT

#### Option SP8-EB (Second Level) (4K)

10-00004	Capacitor
10-00016	Capacitor
10-00024	Capacitor
10-00067	Capacitor
10-00076	Capacitor
10-01610	Capacitor
10-03053	Capacitor
10-05306	Capacitor
10-09678	Capacitor
11-00113	Diode D662
11-00114	Diode IN644
11-05314	1N645
11-09977	1N749A
11-09979	1N1185A
11-10006	1N1201A
11-10181-0	Thyrauctor
11-10182-0	1N4721
11-10183-0	SCR (h724)
11-10625	Diode Light
12-05375	Switch Rocker
12-09403-01	Fan
12-10072	Terminal
12-10073	Connector Socket
12-10198-0	Relay KVP
12-10199-0	Thermal Relay
12-10626	Switch Slide
12-10627	Switch
13-00204	Resistor
13-00229	Resistor
13-00293	Resistor
13-00317	Resistor
13-00391	Resistor
13-00432	Resistor
13-00439	Resistor

#### Option SP8-ED (Second Level) (8K)

10-00004	Capacitor
10-00016	Capacitor
10-00024	Capacitor
10-00067	Capacitor
10-00076	Capacitor
10-01610	Capacitor
10-03053	Capacitor
10-05306	Capacitor
10-09678	Capacitor
11-00113	Diode D662
11-00114	Diode IN644
11-05314	1N645
11-09977	1N749A
11-09979	1N1185A
11-10006	1N1201A
11-10181-0	Thyrauctor
11-10182-0	1N4721
11-10183-0	SCR (h724)
11-10625	Diode Light
12-05375	Switch Rocker
12-09403-01	Fan
12-10072	Terminal
12-10073	Connector Socket
12-10198-0	Relay KVP
12-10199-0	Thermal Relay
12-10626	Switch Slide
12-10627	Switch
13-00204	Resistor
13-00229	Resistor
13-00293	Resistor
13-00317	Resistor
13-00391	Resistor
13-00432	Resistor
13-00439	Resistor

PDP8-E OPTION SP8-EB, SP8-ED (SECOND LEVEL) CONT.

DEC PART NO.	DESCRIPTION	DEC PART NO.	DESCRIPTION
13-00481	Resistor	13-00481	Resistor
13-00496	Resistor	13-00496	Resistor
13-01420	Resistor	13-01420	Resistor
13-02871	Resistor	13-02871	Resistor
13-02941	Resistor	13-02941	Resistor
13-02955	Resistor	13-02955	Resistor
13-02956	Resistor	13-02956	Resistor
13-04833	Resistor	13-04833	Resistor
13-04855	Resistor	13-04855	Resistor
13-04868	Resistor	13-04868	Resistor
13-05128	Resistor	13-05128	Resistor
13-05252	Resistor	13-05252	Resistor
13-09143-6	Potentiometer	13-09143-6	Potentiometer
13-09143-8	Potentiometer	13-09143-8	Potentiometer
13-10032	Resistor	13-10032	Resistor
13-10071	Resistor	12-10071	Resistor
13-10170	Thermister	13-10170	Thermister
15-01742	Transistor	15-01742	Transistor
15-03100	Transistor	15-03100	Transistor
15-03409-01	MPS6534B	15-03409-01	MPS6534B
15-05321	2N4258	15-05321	2N4258
15-05819	2N3055 Case	15-05819	2N3055 Case
15-09338	2N1613	15-09338	2N1613
15-09632	DEC 2007	15-09632	DEC 2007
15-09649	2N3762	15-09649	2N3762
15-10015	DEC 4008	15-11102	DEC 4011
15-01051	2N3054	15-10151	2N3054
16-09478	Transformer	16-09478	Transformer
16-09651	Transformer	16-09651	Transformer
16-09996	Transformer	16-09996	Transformer
18-09880	Crystal	18-09880	Crystal
18-09880-01	Crystal	18-09880-01	Crystal
19-05521	DEC 1540	19-05521	DEC 1540
19-05547	DEC 7474	19-05547	DEC 7474
19-05575-00	DEC 7400	19-05575-00	DEC 7400
19-05579	DEC 7440	19-05579	DEC 7440
19-05586	DEC 74H40	19-05586	DEC 74H40
19-09004	DEC 7402	19-09004	DEC 7402
19-09055	DEC 7495	19-09055	DEC 7495
19-09056	DEC 74H00	19-09056	DEC 74H00
19-09057	DEC 74H10	19-09057	DEC 74H10
19-09267	DEC 74H11	19-09267	DEC 74H11
19-09373	DEC ML-9601	19-09373	DEC ML-9601
19-09594	DEC 82513-930	19-09594	DEC 82513-930
19-09667	DEC 74H74	19-09667	DEC 74H74
19-09686	DEC 7404	19-09686	DEC 7404
19-09854	DEC 8251-1	19-10046	DEC 7442-1
19-09867	DEC 4007	19-09867	DEC 4007
19-09927	DEC 74H87	19-09927	DEC 74H87
19-09928	DEC 7416	19-09928	DEC 7416
19-09929	DEC 7417	19-09929	DEC 7417
19-09930	DEC 7405	19-09930	DEC 7405
19-09931	DEC 74H04	19-09931	DEC 74H04
19-09932	DEC 7483	19-09932	DEC 7483
19-09934	DEC 8266	19-09934	DEC 8266
19-09935	DEC 8235	19-09935	DEC 8235
19-09936	DEC 74151	19-09936	DEC 74151
19-09937	DEC 74153	19-09937	DEC 74153
19-09955	DEC 7412	19-09955	DEC 7412

**PDP8E CUSTOMER RECOMMENDED SPARES**  
**OPTION SP8-EB, SP8-ED (SECOND LEVEL). CONT.**

DEC PART NO.	DESCRIPTION	DEC PART NO.	DESCRIPTION
19-09971	DEC 6380A	19-09971	DEC 6380A
19-09972	DEC 6314	19-09972	DEC 6314
19-09973	DEC 97401	19-09973	DEC 97401
19-09981	UA723C Volt Reg.	19-09981	UA723C Volt Reg.
19-10010	DEC FSA2501	19-10010	DEC FSA2501
19-10011	DEC 7486	19-10011	DEC 7486
19-10391	DEC 5314	19-10391	DEC 5314
19-10392	DEC 5380	19-01392	DEC 5380
19-10394	DEC 5384	19-01394	DEC 5384
19-10624	DEC 74L54	19-10624	DEC 74L54
90-07208	Fuse	19-10973	R-C Network
90-08386-0	Fuse	90-07208	Fuse
90-08387-0	Fuse	90-08386-0	Fuse
90-08388-0	Fuse	90-08387-0	Fuse
90-08389-0	Fuse	90-08388-0	Fuse
90-08390-0	Fuse	90-08389-0	Fuse
12-10089	Socket	90-08390-0	Fuse
12-10090	Housing	12-10089	Socket
	SP8-ED KIT	12-10090	Housing
			SP8-EB KIT

**PDP8M CUSTOMER RECOMMENDED SPARES**

OPTION SP8-MA KIT (FIRST LEVEL) (4K)		OPTION SP8-MC KIT (FIRST LEVEL) (8K)	
M8300	Major Registers Module	M8300	Major Registers Module
M8310	Registers Control Module	M8310	Registers Control Module
M8330	Timing Module	M8330	Timing Module
G104	Sense/Inhibit Module	G111	Sense/Inhibit Module
G227	X/Y Drive Module	G233	X/Y Drive Module
11-10625	Light Emitting Diode	11-10625	Light Emitting Diode
12-10626	Slide Switch	12-10626	Slide Switch
12-05375	Slide Switch, Momentary	12-05375	Slide Switch, Momentary
12-05849-06	Handle, Russett Orange	12-05849-06	Handle, Russett Orange
12-05849-13	Handle, Terra Cotta	12-05849-13	Handle, Terra Cotta
54-09728	Regulator Board Assy.	54-09728	Regulator Board Assy.
	SP8-MA KIT		SP8-MC KIT

**PDP8F CUSTOMER RECOMMENDED SPARES**

OPTION SP8-FA KIT (FIRST LEVEL) (4K)		OPTION SP8-FC KIT (FIRST LEVEL) (8K)	
M8300	Major Registers Module	M8300	Major Registers Module
M8310	Registers Control Module	M8310	Registers Control Module
M8330	Timing Module	M8330	Timing Module
G104	Sense/Inhibit Module	G111	Sense/Inhibit Module
G227	X/Y Drive Module	G233	X/Y Drive Module
11-10625	Light Emitting Diode	11-10625	Light Emitting Diode
12-10626	Slide Switch	12-10626	Slide Switch
12-05375	Slide Switch, Momentary	12-05375	Slide Switch, Momentary
12-5849-12	Handle, Amber	12-5849-12	Handle, Amber
12-5849-13	Handle, Terra Cotta	12-5849-13	Handle, Terra Cotta
54-09728	Regulator Board Assy.	54-09728	Regulator Board Assy.
	SP8-FA KIT		SP8-FC KIT

**PDP8F—PDP8M CUSTOMER RECOMMENDED SPARES**  
**OPTION SP8-MB, SP8-MD (SECOND LEVEL)**  
**OPTION SP8-FB, SP8-FD (SECOND LEVEL)**

DEC PART NO.	DESCRIPTION	DEC PART NO.	DESCRIPTION
10-00004	Capacitor	15-10196	2N5302
10-00016	Capacitor	15-10706	GPS—A55 or MPS—A55
10-03053	Capacitor	15-10765	TRIAC MAC 11-3
10-05306	Capacitor	15-11102	DEC 4011
10-09678	Capacitor	16-09478	Transformer
11-10324	Solid State Lamp	16-09651	Transformer
11-10714	12A Diode Bridge NSS3514	16-09996	Transformer
12-09355	Switch, Micro	18-09880	Crystal
12-05033	Fan Boxer	18-09880-01	Crystal
12-10043	Switch, Miniature Rotary	19-05521	DEC 1540
12-10073	Connector, 40 Terminal	19-05547	DEC 7474
12-10627	Rotary Switch	19-05586	DEC 74H40
12-10790	Switch, DPST N.O.	19-09004	DEC 7402
12-10824	Thermostat	19-09055	DEC 7495
12-10830-5	Circuit breaker, 5 AMP	19-09056	DEC 74H00
12-10830-7	Circuit Breaker, 7 AMP	19-09057	DEC 74H10
13-00229	Resistor	19-09267	DEC 74H11
13-00317	Resistor	19-09373	DEC ML-9601
13-00439	Resistor	19-09594	DEC 82513-930
13-01420	Resistor	19-09667	DEC 74H74
13-02871	Resistor	19-09686	DEC 7404
13-02941	Resistor	19-09705	DEC 8881
13-02955	Resistor	19-09867	DEC 4007
13-02956	Resistor	19-09927	DEC 74H87
13-03156	Resistor	19-09928	DEC 7416
13-04833	Resistor	19-09929	DEC 7417
13-04855	Resistor	19-09930	DEC 7405
13-04868	Resistor	19-09931	DEC 74H04
13-05128	Resistor	19-09932	DEC 7483
13-05252	Resistor	19-09934	DEC 8266
13-05872	Resistor	19-09935	DEC 8235
13-10032	Resistor	19-09936	DEC 74151
13-10071	Resistor	19-09937	DEC 74153
13-10709	Resistor	19-09955	DEC 7412
15-03409-01	MPS6534B or 2N3133	19-09971	DEC 6380A
15-05321	2N4258	19-09972	DEC 6314A
15-09338	MPS6531 or 2N1613	19-09973	DEC 97401
15-09632	DEC 2007	19-10010	DEC FSA2501
15-09649	2N3762	19-10011	DEC 7486
19-09594	DEC 8251	19-10046	DEC 7442-1
15-10015	DEC 4008	19-10973	R-C Network
15-10151	RCA 40372 (2N3054)	90-7221	Fuse
		90-07226	Fuse
		90-08389	Fuse

SP8-MB KIT, SP8-FB KIT

SP8-MD, SP8-FD KIT

**CUSTOMER RECOMMENDED ASR-33 SPARES****OPTION LT33-B TELETYPE**

<b>DEC PART NO.</b>	<b>DESCRIPTION</b>
29-11424	Circuit Board
29-11495	Tape Sprocket
29-11443	Lever Universal
29-11144	Fuse
29-11367	Brush (Distrib.)
29-11412	Drive Gear
29-11417	Driven Gear
29-11411	Belt
29-11376	Shaft
29-11375	Bearings
29-11374	Bearings
29-17473	Ribbon
	LT33-B KIT

**CR8-E/CM8-E RECOMMENDED SPARE PARTS****(FIRST LEVEL)**

<b>DEC PART NUMBER</b>	<b>DESCRIPTION</b>
19-9971	IC DEC 6380
19-9705	IC DEC 8881
19-9704	IC DEC 314
19-9373	IC DEC 9601
19-9686	IC DEC 7404
19-9594	IC DEC 8251
19-9050	IC DEC 7475
19-9004	IC DEC 7402
19-5580	IC DEC 7450
19-5579	IC DEC 7440
19-5577	IC DEC 7420
19-5576	IC DEC 7410
19-5575	IC DEC 7400
19-5547	IC DEC 7474
	SP8-CR KIT

**CR8-E/CM8-E RECOMMENDED SPARE PARTS  
(SECOND LEVEL)**

<b>DEC PART NO.</b>	<b>DESCRIPTION</b>
16-5528	Delay line, 30 ns
10-0025	Capacitor, 560 pF, 100V, 5% DM
10-0016	Capacitor, 100 pF, 100V, 5% DM
10-0067	Capacitor, 6.8 μF, 35V, 20% S. TANT.
10-1610	Capacitor, 0.01 μF, 100V, 20% DISK.
70-7252	Cable, Card Reader Interface
	SP8-CS KIT

**OPTION LT33-ST TOOL KIT**

<b>DEC PART NO.</b>	<b>DESCRIPTION</b>
29-12521	8 oz. Scale
29-12522	32 oz. Scale
29-12602	64 oz. Scale
29-12520	Set of gauges
29-12523	Offset screwdriver
29-12524	Offset screwdriver
29-12525	8 Crochet hook
29-12526	12 Crochet hook
29-12527	Spring hook push
29-12528	Spring hook pull
29-12529	Screw holder
29-11418	Handwheel adap
29-12540	Handwheel
29-12553	Contact adjustment
29-12554	Gauge
29-12555	Gauge
29-12556	Bending tool
29-11498	Gauge
29-12558	Extractor
29-12559	Tweezer
29-12560	Tommy Wrench
29-12561	Tommy Wrench
29-12562	Key lever remover
	LT33-ST KIT

**DB8-E RECOMMENDED SPARE PARTS**

<b>DEC PART NO.</b>	<b>DESCRIPTION</b>
19-05547	IC DEC 7474
19-05575	IC DEC 7400
19-05579	IC DEC 7440
19-09486	IC DEC 384
19-09594	IC DEC 8251
19-09686	IC DEC 7404
19-09973	IC DEC 97401
19-09971	IC DEC 6380
19-09972	IC DEC 6314

SP8-DB KIT

### SPARE MODULES

<b>Module Part No.</b>	<b>Used In Option Number</b>	<b>Description</b>
A231	AD8-E	A/D Converter
A841	AD8-E	A/D Converter
A232	AM8-E	8-Channel Mux
M843	CR8-F (CM8-E)	Card Reader Optical Mark Card Reader
M8326	DB8-E	Interprocessor Buffer
M882	DK8-EA	Real-Time Clock (Line)
M883	DK8-EC	Real-Time Clock (Crystal)
M860	DK8-EP	Real-Time Clock (Programmable)
M518	DK8-EP	Real-Time Clock (Programmable)
M7104	RK8-E	DECpack Control
M7105	RK8-E	DECpack Control
M7106	RK8-E	DECpack Control
M839	DP8-E	Synchronous Modem
M866	DP8-E	Synchronous Modem
M863	DR8-E	12-Channel Buffered I/O
M8350	KA8-E	Positive I/O Bus
M8360	KD8-E	Data Break Interface
M884	KG8-E	Redundancy Check
M8340	KE8-E	Extended Arithmetic Element
M8341	KE8-E	Extended Arithmetic Element
M849	KK8-E	RFI Shield
M8300	KK8-E	Major Registers
M8310	KK8-E	Major Registers Control
M8320	KK8-E	Bus Loads
M8330	KK8-E	Timing Generator
M8335	VT8-E	DEC Display Control
M8336	VT8-E	DEC Display Control
M8337	VT8-E	DEC Display Control
M8342	LS8-E	Line Printer
M8650	KL8-E	Teletype Control
M8650 YA	KL8-EX	Asynchronous Interface (EIA Levels)
M8652 YC	KL8-F	Asynchronous Interface (Double Buffered)
M8653	KL8-M	Modem Control
M848	KP8-E	Power Fail & Auto Restart
M8329	LC8-E	DECwriter Control
M841	LE8-E	Line Printer Control
M847	MI8-E	Bootstrap Loader
G104	MM8-E	Sense Inhibit
G111	MM8-EJ	Sense Inhibit 8K
G227	MM8-E	X-Y Drivers 4K
G233	MM8-EJ	X-Y Drivers 8K
H220	MM8-E	Memory Stack (4K)
H212	MM8-EJ	Memory Stack (8K)
G105	MP8-E	Sense Inhibit
M880	MR8-EA	256 ROM
M840	PC8-E	Reader, Punch Control
M868	TD8-EH (TD8-EM)	OMNIBUS DECTape Control
M8321	TM8-E	OMNIBUS DECTape Control
M8322	TM8-E	MAGtape Control
M8323	TM8-E	MAGtape Control
M8327	TM8-E	MAGtape Control
M869	VC8-E	Display Generator
M885	VC8-E	Display Generator
H724(a)	P.S.	Power Supply (PDP8E)
M935	OMNIBUS Connector	Connector Module
M842	XY8-E	XY Plotter Control

**KG8-E RECOMMENDED SPARE PARTS**

DEC PART NO.	DESCRIPTION
19-09704	IC DEC 314
19-09485	IC DEC 380
19-05575	IC DEC 7400
19-09004	IC DEC 7402
19-09686	IC DEC 7404
19-05576	IC DEC 7410
19-05580	IC DEC 7450
19-05547	IC DEC 7474
19-10011	IC DEC 7486
19-09594	IC DEC 8251
19-09615	IC DEC 8271
19-09705	IC DEC 8881
19-10035	IC DEC 74197
10-00016	Capacitor, 100 pF, 100V, 5%
10-01610	Capacitor, 0.01 μF, 100V, 20% DISC
10-00027	Capacitor, 820 pF, 100V, 5%
13-00293	Resistor, 330Ω, 1/4W, 10%
13-01401	Resistor, 750Ω, 1/4W, 5%
13-00271	Resistor, 220Ω, 1/4W, 5%
	SP8-KG KIT

**KM8-E RECOMMENDED SPARE PARTS**

DEC PART NO.	DESCRIPTION
19-05575	IC DEC 7400
19-09705	IC DEC 8881
19-09615	IC DEC 8271
19-09935	IC DEC 8235
19-09934	IC DEC 8266
19-09594	IC DEC 8251
19-09667	IC DEC 74H74
19-05547	IC DEC 7474
19-05577	IC DEC 7420
19-05576	IC DEC 7410
19-09686	IC DEC 7404
19-09056	IC DEC 74H00
19-09486	IC DEC 384A
19-09972	IC DEC 6314A
13-00365	Resistor 1K, 1/4W, 5%
10-01610	Capacitor 0.01 MF DISK, 20%
10-05306	Capacitor 6.8 μF, 35V, 10%
	SP8-KM KIT

**KP8-E RECOMMENDED SPARE PARTS**

DEC PART NO.	DESCRIPTION
11-09991	Diode, AZ1-1/4M, 6.8V
11-00114	Diode, D664
11-05275	Diode, D672
15-03100	Transistor, 3009B
15-03409-01	Transistor, DEC 6534B
19-05547	IC DEC 7474
19-05575	IC DEC 7400
19-05576	IC DEC 7410
19-09004	IC DEC 7402
19-09050	IC DEC 7475
19-09373	IC DEC 9601
19-09486	IC DEC 384
19-09705	IC DEC 8881
19-09971	IC DEC 6380
19-09972	IC DEC 6314
	SP8-KD KIT

**LC8-E RECOMMENDED SPARE PARTS**

DEC PART NO.	DESCRIPTION
19-10394	IC DEC 5384
19-10392	IC DEC 5380
19-10391	IC DEC 5314
19-10046	IC DEC 7442
19-9929	IC DEC 7417
19-9973	IC DEC 97401
19-9686	IC DEC 7404
19-9056	IC DEC 74H00
19-9004	IC DEC 7402
19-5580	IC DEC 7450
19-5579	IC DEC 7440
19-5575	IC DEC 7400
19-5547	IC DEC 7474
10-1610	Capacitor, 0.01 μF, 100V, 20% Disk
10-0067	Capacitor, 6.8 μF, 35V, 20%
10-0024	Tant
70-8417	Capacitor, 47 pF, 100V, 5%DM Signal Cable
	SP8-LC KIT

**M18-E (M847) RECOMMENDED SPARE PARTS**

DEC PART NO.	DESCRIPTION
19-10436	IC DEC 74123
19-09004	IC DEC 7402
19-05547	IC DEC 7474
19-09935	IC DEC 8235
19-10041	IC DEC 74164
19-05575	IC DEC 7400
19-09705	IC DEC 8881
19-09686	IC DEC 7404
19-09485	IC DEC 380
19-09486	IC DEC 384
15-03100	Transistor DEC 3009B
11-00114	Diode D664
13-01423	Resistor 6.8K, 1/4W, 5%
10-00006	Cap., 0.01 μF, 100V, 20%
	SP8-MI KIT

**DK8-EA/DK8-EC—(M882/M883) RECOMMENDED SPARE PARTS(FIRST LEVEL)**

DEC PART NO.	DESCRIPTION
19-9705	DEC 8881
19-9704	DEC 314
19-9485	DEC 380
19-9051	DEC 7490
19-9050	DEC 7475
19-9004	DEC 7402
19-5589	DEC 7470
19-5576	DEC 7410
19-5575	DEC 7400
19-5547	DEC 7474
19-9486	DEC 384
	SP8-DK KIT

**DK8-EA/DK8-EC—(M882/M883) RECOMMENDED SPARE PARTS (SECOND LEVEL)**

DEC PART NO.	DESCRIPTION
18-9880	Crystal (M883 only)
16-9651	Pulse Transformer (M883 only)
10-9678	Capacitor 0.047 $\mu$ F, 16-15 20%
10-1610	Capacitor 0.01 $\mu$ F, 100V, 20%
10-0016	Capacitor 100 pF, 100V, 5%
10-0014	Capacitor 68 pF, 100V, 5%
10-0011	Capacitor 47 pF, 100V, 5%
10-0006	Capacitor 10 pF, 100V, 5%
10-1765	Capacitor 0.005 $\mu$ F,
	SP8-DL KIT

**KA8-E RECOMMENDED SPARE PARTS**

DEC PART NO.	DESCRIPTION
15-03100	Transistor, DEC 3009B
19-09705	IC DEC 8881
19-10010	IC DEC 2501
19-09971	IC DEC 6380
19-09921	IC DEC 7417
19-09928	IC DEC 7416
19-09686	IC DEC 7404
19-09373	IC DEC 9601 (M835 only)
19-09486	IC DEC 384
19-09004	IC DEC 7402
19-05578	IC DEC 7430
19-05577	IC DEC 7420
10-05576	IC DEC 7410
19-05575	IC DEC 7400
19-05547	IC DEC 7474
11-00114	Diode D664
11-00113	Diode D662
BC08J-10	Cable, 10 ft.
19-10436	IC DEC 74123 (M8350 only)
	SP8-KA KIT

**KD8-E RECOMMENDED SPARE PARTS**

DEC PART NO.	DESCRIPTION
10-01610	Capacitor 0.01 $\mu$ F, 100V, 20%
11-00113	Diode D662
19-05575	IC DEC 7400
19-05579	IC DEC 7440
19-09004	IC DEC 7402
19-09057	IC DEC 74H10
19-09267	IC DEC 74H11
19-09971	IC DEC 6380
19-09486	IC DEC 384
19-09615	IC DEC 8217
19-09667	IC DEC 74H74
19-09686	IC DEC 7404
19-09972	IC DEC 6314
19-09973	IC DEC 97401
19-09928	IC DEC 7416
19-09934	IC DEC 8266
19-09955	IC DEC 7412
19-10010	IC DEC 2501
	SP8-KD KIT

**KE8-E RECOMMENDED SPARE PARTS**

DEC PART NO.	DESCRIPTION
19-05585	IC DEC 7476
19-05576	IC DEC 7410
19-09955	IC DEC 7412
19-10018	IC DEC 74193
19-09934	IC DEC 8266
19-09267	IC DEC 74H11
19-05635	IC DEC 74H20
19-05586	IC DEC 74H40
19-09486	IC DEC 384
19-09004	IC DEC 7402
19-09667	IC DEC 74H74
19-09059	IC DEC 74H30
19-09973	IC DEC 97401
19-09485	IC DEC 380
23-001A1	IC Encoded ROM (Drives ROM 11-18)
23-002A1	IC Encoded ROM (Drives ROM 21-28)
19-09930	IC DEC 7405
19-09705	IC DEC 8881
19-05575	IC DEC 7400
19-09686	IC DEC 7404
19-09062	IC DEC 74H53
19-10011	IC DEC 7486
19-09935	IC DEC 8235
13-00295	Resistor 330 $\Omega$ /4W, 5%
13-00365	Resistor 1K, 1/4W, 5%
13-00317	Resistor 470 $\Omega$ , 1/4W, 10%
10-000067	Capacitor 6.8 $\mu$ F, 5V, 20%
	Solid Tantalum
	Capacitor 0.01 $\mu$ F, 100V, 20%
	Ceramic Disk
10-01610	SP8-KE KIT

**RECOMMENDED MR8-E SPARE PARTS**

DEC PART NO.	DESCRIPTION
15-05321	Transistor DEC 4258
15-03100	Transistor DEC 3009B
11-00114	Diode D664
19-10047	IC DEC 74145
19-10046	IC DEC 7442
19-09705	IC DEC 8881
19-09688	IC DEC 846
19-09667	IC DEC 74H74
19-09486	IC DEC 384
19-09971	IC DEC 6380
19-09267	IC DEC 74H11
19-09056	IC DEC 7402
19-05576	IC DEC 7410
19-05575	IC DEC 7400
	SP8-MR KIT

**RECOMMENDED MP8-E SPARE PARTS**

DEC PART NO.	DESCRIPTION
15-02155	Transistor DEC 1008
15-01881	Transistor DEC 2219
15-03100	Transistor DEC 3009B
15-10062	Transistor DEC 3734
15-09649	Transistor DEC 3762
15-10015	Transistor DEC 4008
15-0532	Transistor DEC 4258
15-03409-01	Transistor DEC 6534B
19-05575	IC DEC 7400
19-05590	IC DEC 7401
19-09004	IC DEC 7402
19-09686	IC DEC 7404
19-05580	IC DEC 7450
19-05547	IC DEC 7474
19-10724	IC DEC 74180
19-09056	IC DEC 74H00
19-09057	IC DEC 74H10
19-09267	IC DEC 74H11
19-05586	IC DEC 74H40
19-09967	IC DEC 74H74
19-09704	IC DEC 314
19-09485	IC DEC 380
19-09486	IC DEC 384
19-09594	IC DEC 8251
19-09705	IC DEC 8881
	SP8-MQ

**TD8-E RECOMMENDED SPARE PARTS  
(SECOND LEVEL)**

DEC PART NO.	DESCRIPTION
19-09004	IC DEC 7402
19-05590	IC DEC 7401
19-05578	IC DEC 7430
19-05576	IC DEC 7410
19-05575	IC DEC 7400
19-05547	IC DEC 7474
15-09338	Transistor DEC 6351
11-00114	Diode, D664
11-00113	Diode, D662
	SP8-TE KIT

**RECOMMENDED MP8-E SPARE PARTS (SECOND LEVEL)**

DEC PART NO.	DESCRIPTION
12-10043	Rotary Switch
16-09651	Transformer 8010
16-09996	Transformer 6501
16-09478	Transformer 1775
16-09559	Delay Line, 100ns
13-10032	Resistor, 16.9 ohm, 6W, 1%
13-02858	Resistor, 100 ohm, 1/8W, 1%
13-02956	Resistor, 196 ohm, 1/8W, 1%
13-05114	Resistor, 348 ohm, 1/8W, 1%
13-02955	Resistor, 750 ohm, 1/8W, 1%
13-03114	Resistor, 1K ohm, 1/8W, 1%
13-02871	Resistor, 1.21K ohm, 1/8W, 1%
13-04833	Resistor, 1.96K ohm, 1/8W, 1%
13-04856	Resistor, 4.64K ohm, 1/8W, 1%
13-04885	Resistor, 9.09K ohm, 1/8W, 1%
13-02941	Resistor, 14.7K ohm, 1/8W, 1%
13-03156	Resistor, 34.8K ohm, 1/8W, 1%
13-05128	Resistor, 56.2K ohm, 1/8W, 1%
13-05252	Resistor, 68.1K ohm 1/8W, 1%
13-10071	Thermistor, 1K, 1%
11-05275	Diode D672
11-00114	Diode D664
11-09991	Zener Diode 1/4M6, 8AZ1
19-10010	Diode Pack, DEC 2501
	SP8-MP KIT

**TD8-E RECOMMENDED SPARE PARTS**

DEC PART NO.	DESCRIPTION
19-10436	IC 74123
19-09935	IC 8235
19-09931	IC DEC 74H04
19-09929	IC DEC 7417
19-09712	IC DEC 8242
19-09705	IC DEC 8881
19-10391	IC DEC 5314
19-09686	IC DEC 7404
19-09615	IC DEC 8271
19-09594	IC DEC 8251
19-10394	IC DEC 5384
19-10392	IC DEC 5380
19-09054	IC DEC 7493
19-09050	IC DEC 7475
	SP8-TD KIT

**XY8-E RECOMMENDED SPARE PARTS**

DEC PART NO.	DESCRIPTION
15-03409-1	Transistor DEC 6534B
15-03100	Transistor DEC 3009B
19-09705	IC DEC 8881
19-09704	IC DEC 314
19-09686	IC DEC 7404
19-09594	IC DEC 8251
19-09373	IC DEC 9601
19-09485	IC DEC 380
19-05577	IC DEC 7420
19-05576	IC DEC 7410
19-05575	IC DEC 7400
19-05547	IC DEC 7474
13-000391	Resistor 1.5K, 1/4W, 5%
	SP8-XY KIT

# LOGIC CIRCUITS

## INTRODUCTION

The digital logic circuits in this chapter are used to interface I/O devices to the computer using Digital Equipment Corporation FLIP CHIP Modules. Logic handbooks published by DEC describe hundreds of FLIP CHIP Modules with their component circuits, associated accessories, hardware, power supplies, and mounting panels. The designer should study the logic handbooks carefully before beginning on interface design for a special I/O device.

The basic logic circuits used for interfacing to the computer are: AND, OR, NAND, NOR, Flip-Flop, Single-Shot, Schmitt Trigger, Inverter, Amplifier, and Bus Driver. A brief discussion of these circuits and their logic symbology follows.

The symbology employed with the PDP-8 family of computers and M-series modules is similar to MIL-STD-806B. This chapter describes DEC symbology with definitions of logic functions, graphic representations of the functions, and examples of their application. A Table of Combinations is also shown.

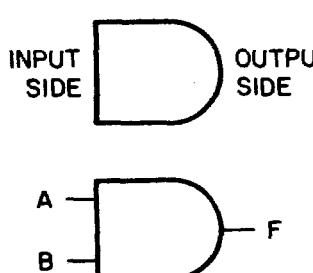
## LOGIC SYMBOLS

The following description of logic symbols contains truth tables that show graph representations of the logic functions. In the truth tables, the letter H stands for HIGH (+ 3V), and the letter L stands for LOW (0V). Examples of DEC symbology are shown along with figures and truth tables.

### State Indicator

The presence of the small circular symbol at the input(s) of a function indicates that an L input signal activates the function. The absence of this small circle indicates that an H input signal activates the function. Similarly, a small circle at the output of a function indicates that the output terminal of the activated function is relatively low, and the absence of the circle indicates that the output is relatively high.

### STATE INDICATOR ABSENT



INPUT		OUTPUT
A	B	F
L	L	L
L	H	L
H	L	L
H	H	H

Symbol, AND Function



INPUT		OUTPUT
A	B	F
L	L	L
L	H	H
H	L	H
H	H	H

Symbol, OR Function

### STATE INDICATOR PRESENT



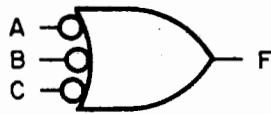
INPUT			OUTPUT
A	B	C	F
L	L	L	H
L	L	H	H
L	H	L	H
L	H	H	H
H	L	L	H
H	L	H	H
H	H	L	H
H	H	H	L

Symbol, NAND Function



INPUT			OUTPUT
A	B	C	F
L	L	L	H
L	L	H	L
L	H	L	L
L	H	H	L
H	L	L	L
H	L	H	L
H	H	L	L
H	H	H	L

Symbol, NOR Function

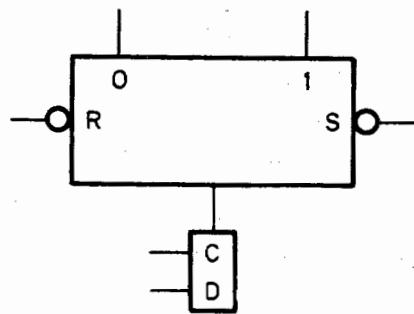


INPUT			OUTPUT
A	B	C	F
L	L	L	H
L	L	H	H
L	H	L	H
L	H	H	H
H	L	L	H
H	L	H	H
H	H	L	H
H	H	H	L

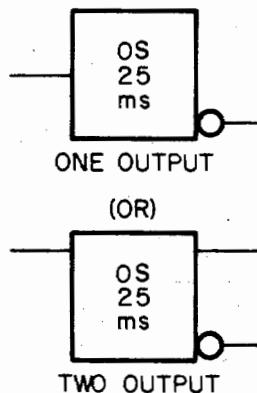
Symbol, NOR Function

Table of Combinations

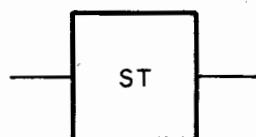
AND	OR	A    B    F
		H    H    H H    L    L L    H    L L    L    L
		H    H    L H    L    L L    H    L L    L    H
		H    H    H H    L    H L    H    H L    L    L
		H    H    L H    L    H L    H    H L    L    H



Symbol, Flip-Flop Function



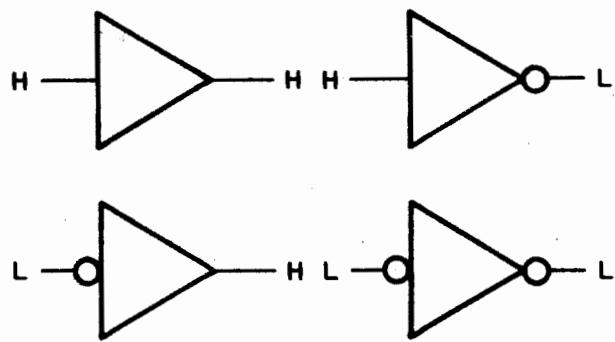
Symbol, One-Shot Function



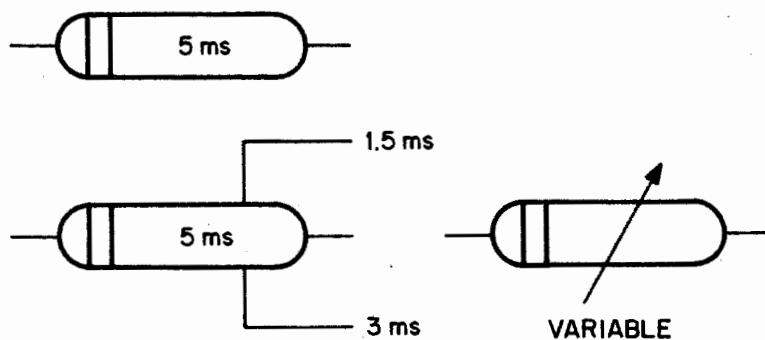
Symbol, Schmitt Trigger Function



Symbol, General Logic



Symbol, Amplifier



Symbol, Time Delay Function

# THE DEC TRAINING PROGRAM

Digital Equipment Corporation offers an extensive training program to every organization that purchases or presently owns a DEC computer. Our training objective is to familiarize the user with the hardware and software associated with his computer system.

Professional full-time instructors regularly conduct classes at Digital's main training facility in Maynard, Massachusetts; Palo Alto, California; Reading, England; Paris, France; Munich, Germany; Scandinavia and Australia.

Early application for enrollment in the desired course is suggested, as class sizes are limited. DEC provides all training materials necessary for each class.

**Software Courses** — The software courses are familiarization courses, and as a general rule will give participants a working knowledge of the appropriate:

Machine Language Instruction Set	Assembler Programs
Programming Techniques	Editor Programs
Input/Output Programming	Debugging Routines

**Hardware Courses** — The hardware courses are familiarization courses, with the exception of Systems Maintenance Courses, and will give participants a working knowledge of the appropriate:

Machine Language Instruction Set	
Logic Symbology	
Theory of Logic Operation of: Memory Unit	
	Central Processor Unit
	I/O Section

**Systems Maintenance Courses** — These courses are specially designed for the user who will maintain his own system or be involved with extensive interfacing, and who requires a good working knowledge of his system. The courses are patterned after those that DEC uses to train its own Field Service men, and will consist of lab time for on-the-job training.

The scheduled dates for DEC Training Courses at Maynard and Palo Alto will be listed in a separate document entitled "DIGITAL CUSTOMER TRAINING SCHEDULE" available from your local sales office, or the Training Departments.

To enroll in a scheduled course, use the enclosed registration form and mail it to the appropriate training facility, at Maynard, Massachusetts, or Palo Alto, California. A letter of confirmation will be sent to each registered student. In case of course cancellation, registered students will be notified during the week prior to the course's scheduled date. Digital Equipment Corporation, Training Department, 146 Main Street, Maynard, Massachusetts 01754. Telephone (617) 897-5111, Extension 2564, TWX 710-347-0212. Digital Equipment Corporation, Training Department, 560 San Antonio Road, Palo Alto, California 94306. Telephone (415) 326-5640, TWX 910-373-1266. Due to lack of adequate public transportation, a rental car is necessary when attending courses at either facility.

Digital can offer special training courses that will help solve the majority of your training problems. These courses can be tailored to your needs and your time schedule, and contain the information you desire your students to learn.

The cost to conduct a Special Course at your facility is the same as the on-site pricing for Standard Courses, plus any additional expenses necessary to prepare a course that we haven't taught before. If the course is conducted at one of our training facilities the cost is \$300 per student per week plus

Digital provides two types of training:

- Software Familiarization
- Hardware Familiarization

Both types of training assume that the individual has either a background in software or hardware fundamentals. For the individual desiring to get the utmost from his available training, or for the individual without the prerequisite background, the Introductory Programming Course will help provide the necessary foundation.

Introductory Programming Course — This course is designed to give the non-computer oriented person, or the individual with no machine language programming background, the fundamentals of computer arithmetic and machine language programming.

## **SOFTWARE COURSE DESCRIPTIONS**

### **Introductory Programming Course**

Course Length: one week

Course Cost: one training credit or \$300

Prerequisites: None

Description: The course gives non-computer oriented personnel the required programming background necessary to enter any of DEC's small computer software familiarization courses. The course consists of a description of: basic computer concepts, binary and octal numbering systems, computer arithmetic, problem solving, flow charting and programming techniques. Basic computer operation includes a description of a representative modified machine language instruction set, applicable programming techniques, use of computer operator console, and I/O programming fundamentals. Course consists of approximately 20% lab time.

## **PDP-8**

### **Family Software Course (Paper Tape)**

Course Length: one week

Course Cost: one training credit or \$300

Prerequisites: Introductory Programming Course or equivalent background

Description: Course covers general operation of the PDP-8 Family computers paper tape system software to include PAL III Assembler, Symbolic Tape Editor, On-line Debugging Technique (ODT), and Floating Point package.

## **PDP-8**

### **Family Software Course (PS-8)**

Course Length: one week

Course Cost: one training credit or \$300

Prerequisites: PDP-8 Family Software Course (Paper tape) or knowledge of the PDP-8 Family Paper Tape Software.

Description: This course covers the operation and familiarization of the 8K mass storage system including the operation of PS-8 editor, PAL 8 assembler, PS-8 Octal Debugging Technique (ODT), the SABR assembler, and user programming.

## **PDP-8**

**Family Software  
Course (4K Monitor)**

Course Cost: one training credit or \$300

Prerequisites: PDP-8 Family Software Course (Paper Tape) or knowledge of the PDP-8 Family Paper Tape Software.

Description: Course covers general operation of 4K mass storage software including monitor, PAL-D assembler Disk/DECtape editor, Disk/DECtape On-line Debugging Program (DDT-D), and Peripheral Interchange Program (PIP). Students will develop programs in the following areas: Disk/DECtape, extended memory, and monitor input/output.

## **INDAC-8**

Course Length: two weeks

Course Cost: two training credits or \$500

Prerequisites: Familiarity with FORTRAN, BASIC or similar procedural language helpful.

Description: This course is designed to discuss:

PDP-8 Family Computers – operation and programming to include the PDP-8 instruction set, loader programs (RIM, BIN, HELP), assembler (PAL-3), and Symbolic Editor.

4K Disk Monitor System – theory and operation to include the building of the monitor program.

## **PDP-8/I-8/L or 8/e**

**Hardware Familiarization  
Courses**

Course Length: one week

Course Cost: one training credit or \$300

Prerequisites: Background in basic electronics, computer technology, and machine language programming.

Description: The course covers the instruction set, central processor including instruction and data flow, memory operation, instruction logic, program interrupt, data break (one and three cycle), and I/O hardware.

## **PDP-8/I-8/L**

**Systems Maintenance  
Course**

Course Length: Three weeks

Course Cost: \$650. (training credits not applicable)

Prerequisites: Prior computer maintenance experience.

Description: The course covers systems description to include instruction set, logic operation of the central processor, power fail, extended memory, DF32-D, PC8/I high speed reader/punch unit, and DMO1. Maintenance is covered from theory of operation to actual troubleshooting on equipment. The course will utilize lab time for hands-on experience.

## **LAB 8/e**

**Hardware Course**

Course Length: two weeks

Course Cost: two training credits or \$500

Prerequisites: PDP-8/e hardware course or equivalent experience.

Description: This course is designed to train the PDP-8/e oriented person the theory and operation and maintenance of the following: extended memory (MM8/E), high speed reader/punch (PC8/E), A-D/D-A concepts and logic (AD8-EA), multiplexor and preamp (AM8-EA-EC), display control (UC8/E), Display (UR-14), digital I/O (DR8-EA), and clock (DK8-EP). The course will utilize lab time for hands-on experience.

## **PDP-8/e**

**Systems Maintenance  
Course**

Course Length: three weeks

Course Cost: \$650. (training credits not applicable)

Prerequisites: Prior computer maintenance experience.

Description: This course covers systems description to include instruction set, logic operation of the central processor, power fail, extended memory, DF32-D, PC8-E high speed reader/punch unit, and MI8-E bootstrap loader. Maintenance is covered from theory of operation to actual troubleshooting on equipment. The course will utilize lab time for hands-on experience.

## **TC08-TU56**

**Hardware Course**

Course Length: one week

Course Cost: \$300. (training credits not applicable)

Prerequisites: PDP-8/I, 8/L, or 8/e Hardware course, Systems Maintenance Course or equivalent experience.

Description: The course covers systems description to include tape format and programming considerations, applicable IOT instructions, operation of the tape transport, logic operation of the control unit including read and write operations.

## ASCII CHARACTER SETS

### OCTAL

		LAST DIGIT							
		0	1	2	3	4	5	6	7
FIRST TWO DIGITS	00	NULL	↑A	↑B	↑C	↑D	↑E	↑F	↑G
	01	↑H	TAB	LINE FEED	VERT. TAB.	FORM FEED	CAR. RET.	↑N	↑O
02	↑P	↑Q	↑R	↑S	↑T	↑U	↑V	↑W	
03	↑X	↑Y	↑Z	ALT MODE	CTRL SHIFT L	CTRL SHIFT M	CTRL SHIFT N	CTRL SHIFT O	
04	SPACE	!	"	*	\$	%	&	'	
05	(	)	*	+	,	-	.	/	
06	0	1	2	3	4	5	6	7	
07	8	9	:	;	<	=	>	?	
10	@	A	B	C	D	E	F	G	
11	H	I	J	K	L	M	N	O	
12	P	Q	R	S	T	U	V	W	
13	X	Y	Z	\	]	]	↑	←	
14	~	a	b	c	d	e	f	g	
15	h	i	j	k	l	m	n	o	
16	p	q	r	s	t	u	v	w	
17	x	y	z	{	}	ESCAPE	DELETE		

### DECIMAL

		LAST DIGIT									
		0	1	2	3	4	5	6	7	8	9
FIRST TWO DIGITS	00	NULL	↑A	↑B	↑C	↑D	↑E	↑F	↑G	↑H	TAB
	01	LINE FEED	VERT. TAB.	FORM FEED	CAR. RET.	↑N	↑O	↑P	↑Q	↑R	↑S
02	↑T	↑U	↑V	↑W	↑X	↑Y	↑Z	ALT MODE	CTRL SHIFT L	CTRL SHIFT M	
03	CTRL SHIFT N	CTRL SHIFT O	SPACE	!	"	*	\$	%	&	'	
04	(	)	*	+	.	-	.	/	0	1	
05	2	3	4	5	6	7	8	9	:	;	
06	<	=	>	?	@	A	B	C	D	E	
07	F	G	H	I	J	K	L	M	N	O	
08	P	Q	R	S	T	U	V	W	X	Y	
09	Z	t	\	]	↑	←	~	o	b	c	
10	d	e	f	g	h	i	j	k	l	m	
11	n	o	p	q	r	s	t	u	v	w	
12	x	y	z	{	}	ESCAPE	DELETE				

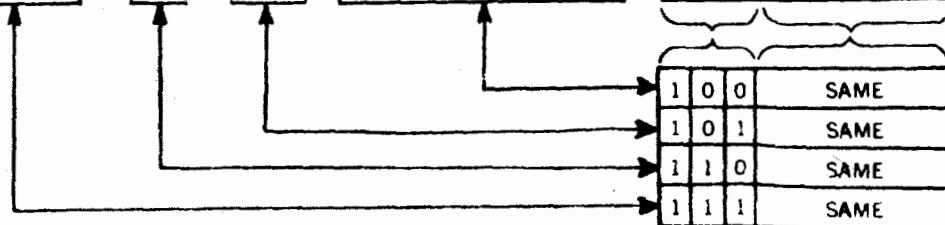
### Model 33 ASR/KSR Teletype Code (ASCII) in Binary Form

1 = HOLE PUNCHED = MARK  
0 = NO HOLE PUNCHED = SPACE

MOST SIGNIFICANT BIT  
LEAST SIGNIFICANT BIT

	8	7	6	5	4	S	3	2	1
	0	0	0	0	0	0	0	0	0
	0	0	0	0	1				
	0	0	0	1	0				
	0	0	0	1	1				
	0	0	1	0	0				
	0	0	1	0	1				
	0	0	1	1	0				
	0	0	1	1	1				
	0	1	0	0	0				
	0	1	0	0	1				
	0	1	0	1	0				
	0	1	1	0	0				
	0	1	1	1	0				
	0	1	1	1	1				
	1	0	0	0	0				
	1	0	0	0	1				
	1	0	0	1	0				
	1	0	0	1	1				
	1	0	1	0	0				
	1	0	1	0	1				
	1	0	1	1	0				
	1	0	1	1	1				
	1	1	0	0	0				
	1	1	0	0	1				
	1	1	0	1	0				
	1	1	1	0	0				
	1	1	1	0	1				
	1	1	1	1	0				
	1	1	1	1	1				

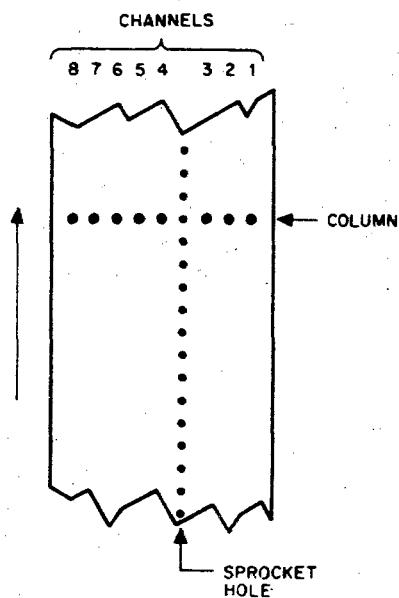
	@	SPACE	NULL/IDLE	START OF MESSAGE	END OF ADDRESS	END OF MESSAGE	END OF TRANSMISSION	WHO ARE YOU	ARE YOU
	A	!	"						
	B	#	\$						
	C	%	&						
	D	'							
	E	(	)						
	F	*							
	G								
	H								
	I								
	J								
	K								
	L								
	M								
	N								
	O								
	P								
	Q								
	R								
	S								
	T								
	U								
	V								
	W								
	X								
	Y								
	Z								
	[								
	-								
	Alt-mode								
	RUB OUT								



### Paper Tape Formats

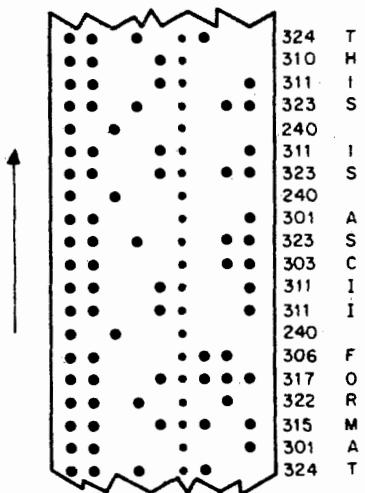
Manual use of the toggle switches on the operator console is a tedious and inefficient means of loading a program. This procedure is necessary in some instances, however, because the PDP-8/E computer must be programmed before any form of input to the memory unit is possible. For example, before any paper tape can be used to input information into the computer, the memory unit must have a stored program which will interpret the paper tape format for the computer. This loader program must be stored in memory with the console switches. A loader program consists of input instructions to accept information from the Teletype paper tape reader and instructions to store the incoming data in the proper memory locations.

Before the loader program can be written to accept information, the format in which the data is represented on the paper tape must be established. There are three basic paper tape formats commonly used in conjunction with PDP-8/E computer. The following paragraphs describe and illustrate these formats.



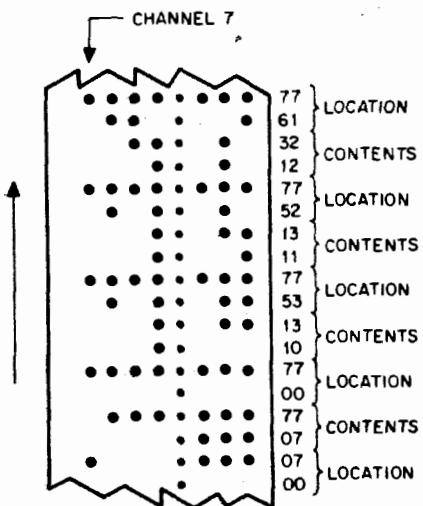
Data is recorded (punched) on paper tape by groups of holes arranged in a definite format along the length of the tape. The tape is divided into channels, which run the length of the tape, and into columns, which extend across the width of the tape, as shown in the adjacent diagram. The paper tape readers and punches used with PDP-8 family computers accept eight-channel paper tape.

\* Channel 8 is normally designated for parity check. The Teletype units used with the PDP-8/E computer do not generate parity, and Channel 8 is always punched.



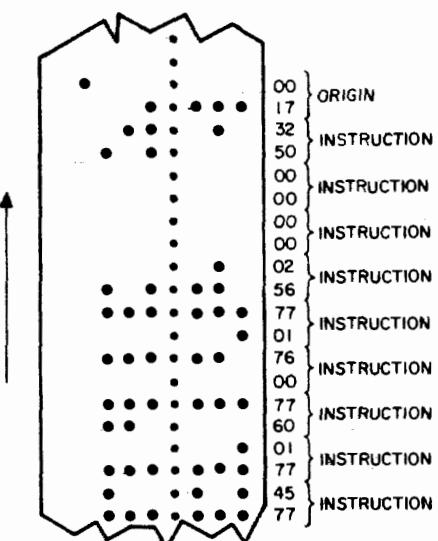
#### ASCII FORMAT

The USA Standard Code for Information Interchange (ASCII) format uses all eight channels\* of the paper tape to represent a single character (letter, number, or symbol) as shown in the diagram at left. The complete code is given in Appendix C.



#### RIM (READ IN MODE) FORMAT

RIM format tape uses adjacent columns to represent 12-bit binary information directly. Channels 1 through 6 are used to represent either address or information to be stored. A channel 7 punch indicates that the adjacent column and the following column are to be interpreted as an address specifying the location in which the information of the following two columns is to be stored. The tape leader and trailer for RIM format tape must be punched in channel 8 only (octal 200).



#### BIN (BINARY FORMAT)

BIN format tape is similar to RIM format except that only the first address of consecutive locations is specified. An address is designated by a channel 7 punch and information following an address is stored in sequential locations after the designated address, until another location is specified as an origin. The tape leader/trailer for BIN format tape must be punched in channel 8 (octal 200) only.

# PERFORATED-TAPE LOADER SEQUENCES

## READIN MODE LOADER

The readin mode (RIM) loader is a minimum length, basic perforated-tape reader program for the ASR33, it is initially stored in memory by manual use of the operator console keys and switches. The loader is permanently stored in 18 locations of page 37.

A perforated tape to be read by the RIM loader must be in RIM format:

Tape Channel <u>8 7 6 5 4 S 3 2 1</u>	<u>Format</u>
1 0 0 0 0 . 0 0 0	Leader-trailer code
0 1 A1 . A2	Absolute address to contain next 4 digits
0 0 A3 . A4	
0 0 X1 . X2	Content of previous 4-digit address
0 0 X3 . X4	
0 1 A1 . A2	Address
0 0 A3 . A4	
0 0 X1 . X2	Content
0 0 X3 . X4	
(Etc.)	(Etc.)
1 0 0 0 0 . 0 0 0	Leader-trailer code

The RIM loader can only be used in conjunction with the ASR33 reader (not the high-speed perforated-tape reader). Because a tape in RIM format is, in effect, twice as long as it need be, it is suggested that the RIM loader be used only to read the binary loader when using the ASR33. (Note that PDP-8 diagnostic program tapes are in RIM format.)

The complete PDP-8/I RIM loader (SA = 7756) is as follows:

Absolute Address	Octal Content	Tag	Instruction I Z	Comments
7756,	6032	BEG,	KCC	/CLEAR AC AND FLAG
7757,	6031		KSF	/SKIP IF FLAG = 1
7760,	5357		JMP .-1	/LOOKING FOR CHARACTER
7761,	6036		KRB	/READ BUFFER
7762,	7106		CLL RTL	
7763,	7006		RTL	/CHANNEL 8 IN ACO
7764,	7510		SPA	/CHECKING FOR LEADER
7765,	5357		JMP BEG+1	/FOUND LEADER
7766,	7006		RTL	/OK, CHANNEL 7 IN LINK
7767,	6031		KSF	
7770,	5367		JMP .-1	
7771,	6034		KRS	/READ, DO NOT CLEAR
7772,	7420		SNL	/CHECKING FOR ADDRESS
7773,	3776		DCA I TEMP	/STORE CONTENT
7774,	3376		DCA TEMP	/STORE ADDRESS

7775,	5356	JMP BEG	/NEXT WORD
7776,	0	TEMP,	/TEMP STORAGE

Placing the RIM loader in core memory by way of the operator console keys and switches is accomplished as follows:

1. Set the starting address 7756 in the switch register (SR).
2. Press LOAD ADDRESS key.
3. Set the first instruction (6032) in the SR.
4. Press the DEPOSIT key.
5. Set the next instruction (6031) in the SR.
6. Press DEPOSIT key.
7. Repeat steps 5 and 6 until all 16 instructions have been deposited.

To load a tape in RIM format, place the tape in the reader, set the SR to the starting address 7756 of the RIM loader (not of the program being read), press the LOAD ADDRESS key, press the START key, and start the Teletype reader.

Refer to Digital Program Library document DEC-08-LRAA-D for additional information on the Readin Mode Loader program.

### BINARY LOADER

The binary loader (BIN) is used to read machine language tapes (in binary format) produced by the program assembly language (PAL). A tape in binary format is about one-half the length of the comparable RIM format tape. It can, therefore, be read about twice as fast as a RIM tape and is, for this reason, the more desirable format to use with the 10 cps ASR33 reader or the Type PR8/I High-Speed Perforated-Tape Reader.

The format of a binary tape is as follows:

LEADER: about 2 feet of leader-trailer codes.

BODY: characters representing the absolute, machine language program in easy-to-read binary (or octal) form. The section of tape may contain characters representing instructions (channel 8 and 7 not punched) or origin resettings (channel 8 not punched, channel 7 punched) and is concluded by 2 characters (channel 8 and 7 not punched) that represent a check sum for the entire section.

TRAILER: same as leader.

Operation of the BIN loader in no way depends upon or uses the RIM loader. To load a tape in BIN format place the tape in the reader, set the SR to 7777 (the starting address of the BIN loader), press the LOAD ADDRESS key, set SR switch 0 up for loading via the Teletype unit or down for loading via the high speed reader, then press the START key, and start the tape reader.

Refer to Digital Program Library document Digital-8-2-U [DEC-08-LBAA-D] for additional information on the Binary Loader program.

Example of the format of a binary tape:

Tape Channel 8 7 6 5 4 S 3 2 1	Memory Location	Content	Comments
1 0 0 0 0 . 0 0 0			leader-trailer code
0 1 0 0 0 . 0 1 0			
0 0 0 0 0 . 0 0 0		0200	
0 0 1 1 1 . 0 1 0			
0 0 0 0 0 . 0 0 0	0200	CLA	origin-setting
0 0 0 0 1 . 0 1 0			
0 0 1 1 1 . 1 1 1	0201	TAD 277	
0 0 0 1 1 . 0 1 0			
0 0 1 1 1 . 1 1 0	0202	DCA 276	
0 0 1 1 1 . 1 0 0			
0 0 0 0 0 . 0 1 0	0203	HLT	
0 1 0 0 0 . 0 1 0			
0 0 1 1 1 . 1 1 1		0277	origin-setting
0 0 0 0 0 . 0 0 0			
0 0 1 0 1 . 0 1 1	0277	0053	
0 0 0 0 1 . 0 0 0			
0 0 0 0 0 . 1 1 1		1007	sum check
1 0 0 0 0 . 0 0 0			leader-trailer code

After a BIN tape has been read in, one of the two following conditions exists:

- a. No checksum error: halt with AC = 0
- b. Checksum error: halt with AC = (completed checksum) — (tape checksum)

## CHARACTER CODES

8-bit ASCII Code	6-bit Code	DEC 029 Card Code	DEC 026 Card Code	Character Representation	Remarks
240	40	blank	blank		space (non-printing)
241	41	11-8-2	12-8-7	!	exclamation point
242	42	8-7	0-8-5	"	quotation marks
243	43	8-3	0-8-6	#	number sign <sup>(10)</sup>
244	44	11-8-3	11-8-3	\$	dollar sign
245	45	0-8-4	0-8-7	%	percent
246	46	12	11-8-7	&	ampersand
247	47	8-5	8-6	'	apostrophe or acute accent
250	50	12-8-5	0-8-4	(	opening parenthesis
251	51	11-8-5	12-8-4 <sup>(1)</sup>	)	closing parenthesis
252	52	11-8-4	11-8-4	*	asterisk
253	53	12-8-6	12	+	plus
254	54	0-8-3	0-8-3	,	comma
255	55	11	11	-	minus sign or hyphen
256	56	12-8-3	12-8-3	.	period or decimal point
257	57	0-1	0-1	/	slash
260	60	0	0	0	
261	61	1	1	1	
262	62	2	2	2	
263	63	3	3	3	
264	64	4	4	4	
265	65	5	5	5	
266	66	6	6	6	
267	67	7	7	7	
270	70	8	8	8	
271	71	9	9	9	
272	72	8-2	11-8-2	:	colon
273	73	11-8-6	0-8-2	:	semicolon
274	74	12-8-4	12-8-6	<	less than
275	75	8-6	8-3	=	equals
276	76	0-8-6	11-8-6	>	greater than
277	77	0-8-7	12-8-2	?	question mark
300	00	8-4	8-4	@	at sign
301	01	12-1	12-1	A	
302	02	12-2	12-2	B	
303	03	12-3	12-3	C	
304	04	12-4	12-4	D	
305	05	12-5	12-5	E	
306	06	12-6	12-6	F	
307	07	12-7	12-7	G	

8-bit ASCII Code	6-bit Code	DEC 029 Card Code	DEC 026 Card Code	Character Representation	Remarks
310	10	12-8	12-8	H	
311	11	12-9	12-9	I	
312	12	11-1	11-1	J	
313	13	11-2	11-2	K	
314	14	11-3	11-3	L	
315	15	11-4	11-4	M	
316	16	11-5	11-5	N	
317	17	11-6	11-6	O	
320	20	11-7	11-7	P	
321	21	11-8	11-8	Q	
322	22	11-9	11-9	R	
323	23	0-2	0-2	S	
324	24	0-3	0-3	T	
325	25	0-4	0-4	U	
326	26	0-5	0-5	V	
327	27	0-6	0-6	W	
330	30	0-7	0-7	X	
331	31	0-8	0-8	Y	
332	32	0-9	0-9	Z	
333	33	12-8-2 <sup>(5)</sup>	11-8-5	[	opening bracket, SHIFT/K
334	34	11-8-7 <sup>(6)</sup>	8-7	\	backslash, SHIFT/L <sup>(8)</sup>
335	35	0-8-2	12-8-5	]	closing bracket, SHIFT/M
336	36	12-8-7 <sup>(7)</sup>	8-5	^	circumflex <sup>(2)</sup>
337	37	0-8-5 <sup>(3)</sup>	8-2 <sup>(3)</sup>	—	underline <sup>(4,9)</sup>

Footnotes:

- (1) On some DEC 026 Keyboards this character is graphically represented as □.
- (2) On most DEC Teletypes circumflex is replaced by up-arrow (^).
- (3) A card containing this code in column 1 with all remaining columns blank is an end-of-file card.
- (4) On most DEC Teletypes underline is replaced by backarrow (<).
- (5) On some 029 keyboards this character is graphically represented as a cent sign (¢).
- (6) On some 029 keyboards this character is graphically represented as logical NOT (¬).
- (7) On some 029 keyboards this character is graphically represented as vertical bar (|).
- (8) On some LP8 line printers, the character diamond (◊) is printed instead of backslash.
- (9) On some LP8 line printers, the character heart (♥) is printed instead of underline.
- (10) The number sign on some terminals is replaced by pound sign (£).

# LINE PRINTER CHARACTER CODES

LE8-E LINE PRINTER CODE

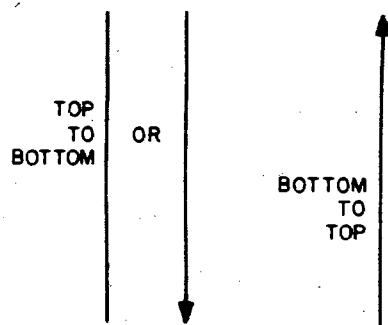
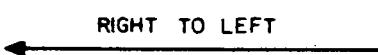
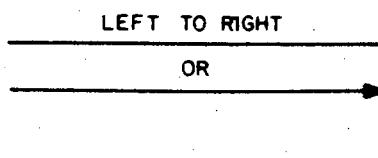
Octal digits in AC 5-8	Octal digit in AC 9-11							
	0 ,	1	2	3	4	5	6	7
00								
01			LF		FF	CR		
02								
03								
04	SP	!	"	#	\$	%	&	'
05	(	)	*	+	-	.	6	/
06	O	1	2	3	4	5	7	
07	8	9	:	;	<	=	>	?
10	@	A	B	C	D	E	F	G
11	H	I	J	K	L	M	N	O
12	P	Q	R	S	T	U	V	W
13	X	Y	Z	[	\	]	^	—
14	'	a	b	c	d	e	f	g
15	h	i	j	k	l	m	n	o
16	p	q	r	s	t	u	v	w
17	x	y	z	{	}		~	DEL

- Notes:
1. LF = Line Feed  
 FF = Form Feed  
 CR = Carriage Return  
 SP = Space
  2. Characters below the heavy line are available only on 96-character printers.
  3. On some early models of the LE8-E Line Printer, the \ (Code 134) is replaced by a ♦; and the \_\_\_ (Code 137) is replaced by a ♥.

# Flowchart guide

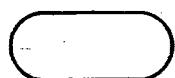
The following is a partial list of flowchart symbols which can be used to diagram the logical flow of a program. The symbols may be made sufficiently large to include the pertinent information.

## REPRESENTATION OF FLOW



The direction of flow in a program is represented by lines drawn between symbols. These lines indicate the order in which the operations are to be performed. Normal direction of flow is from left to right and top to bottom. When the flow direction is not from left to right or top to bottom, arrowheads are placed on the reverse direction flowlines. Arrowheads may also be used on normal flow lines for increased clarity.

## TERMINAL

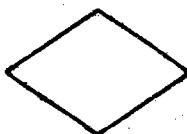


The oval symbol represents a terminal point in a program. It can be used to indicate a start, stop, or interrupt of program flow. The appropriate word is included within the symbol.

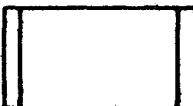
## PROCESSING



The rectangular symbol represents a processing function. The process which the symbol is used to represent could be an instruction or a group of instructions to carry out a given task. A brief description of the task to be performed is included within the symbol.

**DECISION**

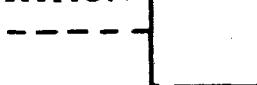
A diamond is used to indicate a point in a program where a choice must be made to determine the flow of the program from that point. A test condition is included within the symbol and the possible results of the test are used to label the respective flows from the symbol.

**PREDEFINED PROCESS**

This symbol is used to represent an operation or group of operations not detailed in the flowchart. It is usually detailed in another flowchart. A subroutine is often represented in this manner.

**CONNECTOR**

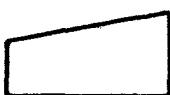
The circular symbol shown below represents an entry from or an exit to another part of the program flowchart. A number or a letter is enclosed to label the corresponding exits and entries. This symbol does not represent a program operation.

**ANNOTATION**

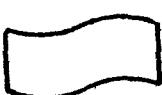
An addition of descriptive comments or explanatory notes for clarification is included within this symbol.

**INPUT/OUTPUT**

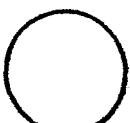
This symbol is used in a flowchart to represent the input or output of information. This symbol may be used for all input/output functions, or symbols for specific types of input or output (such as those which follow) may be used.

**MANUAL INPUT**

This symbol may be used to represent the manual input of information by means of on-line keyboards, switch settings, etc.

**PUNCHED TAPE**

The input or output of information in which the medium is punched tape may be represented by this symbol.

**MAGNETIC TAPE**

This symbol is used in a flowchart to represent magnetic tape input or output.

## PAPER TAPE LOADERS

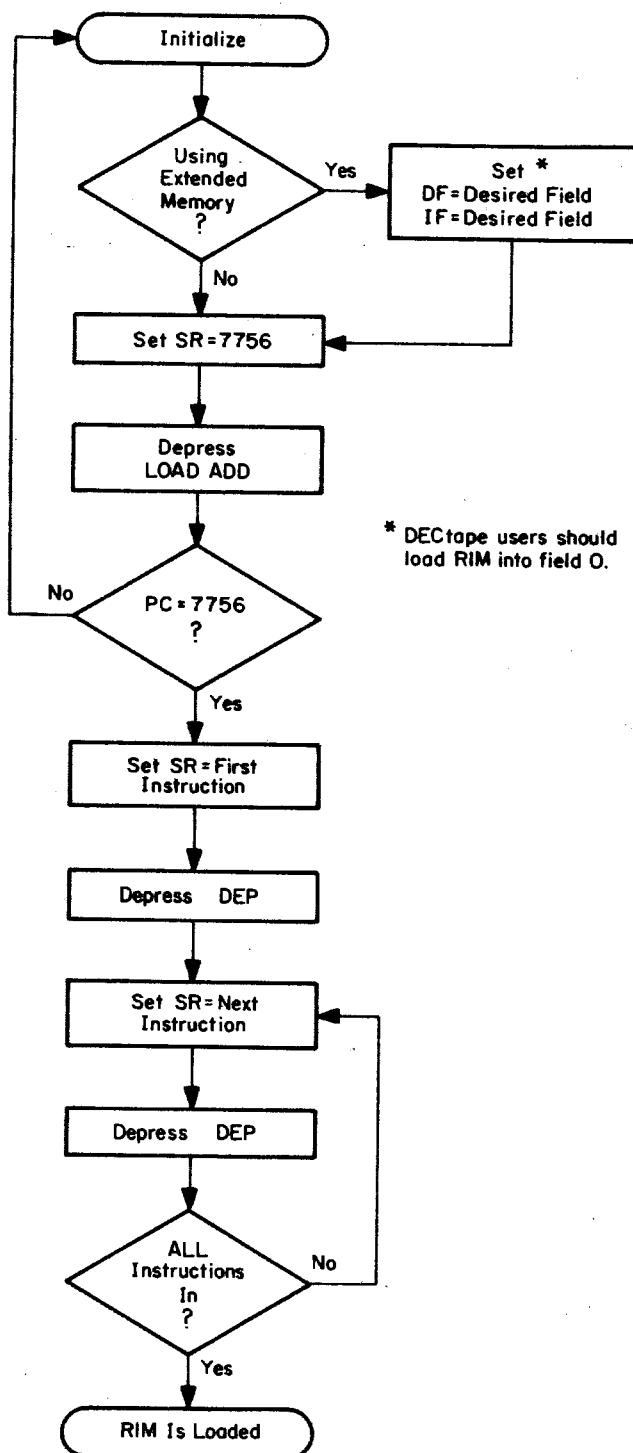
### READ-IN MODE (RIM) LOADER

The RIM Loader is used to load programs punched on RIM format paper tape into core memory. It is stored in core memory locations 7756-7776 (21<sub>8</sub> locations), and started at location 7756. There are two versions of the RIM Loader, permitting either the high- or the low-speed reader to be used as an input device. The locations and corresponding instructions for both versions are listed below.

Table E1-1 RIM Loader Programs

Location	INSTRUCTION	
	Low-Speed Reader	High-Speed Reader
7756	6032	6014
7757	6031	6011
7760	5357	5357
7761	6036	6016
7762	7106	7106
7763	7006	7006
7764	7510	7510
7765	5357	5374
7766	7006	7006
7767	6031	6011
7770	5367	5367
7771	6034	6016
7772	7420	7420
7773	3776	3776
7774	3376	3376
7775	5356	5357

Note: Location 7776 is used for temporary storage.

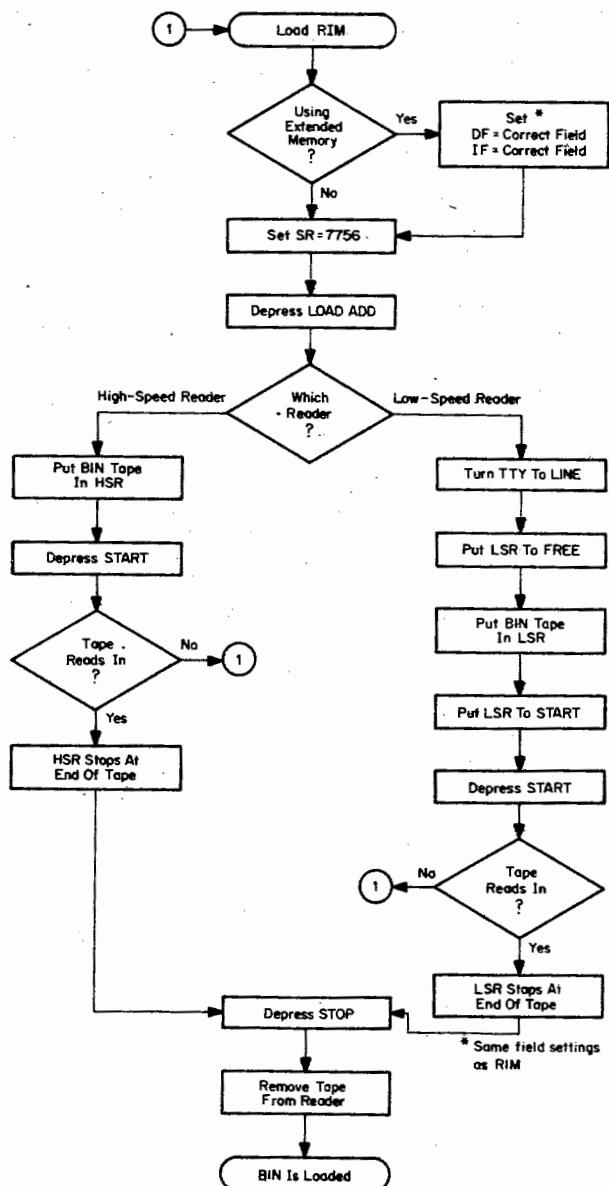


Loading the RIM Loader

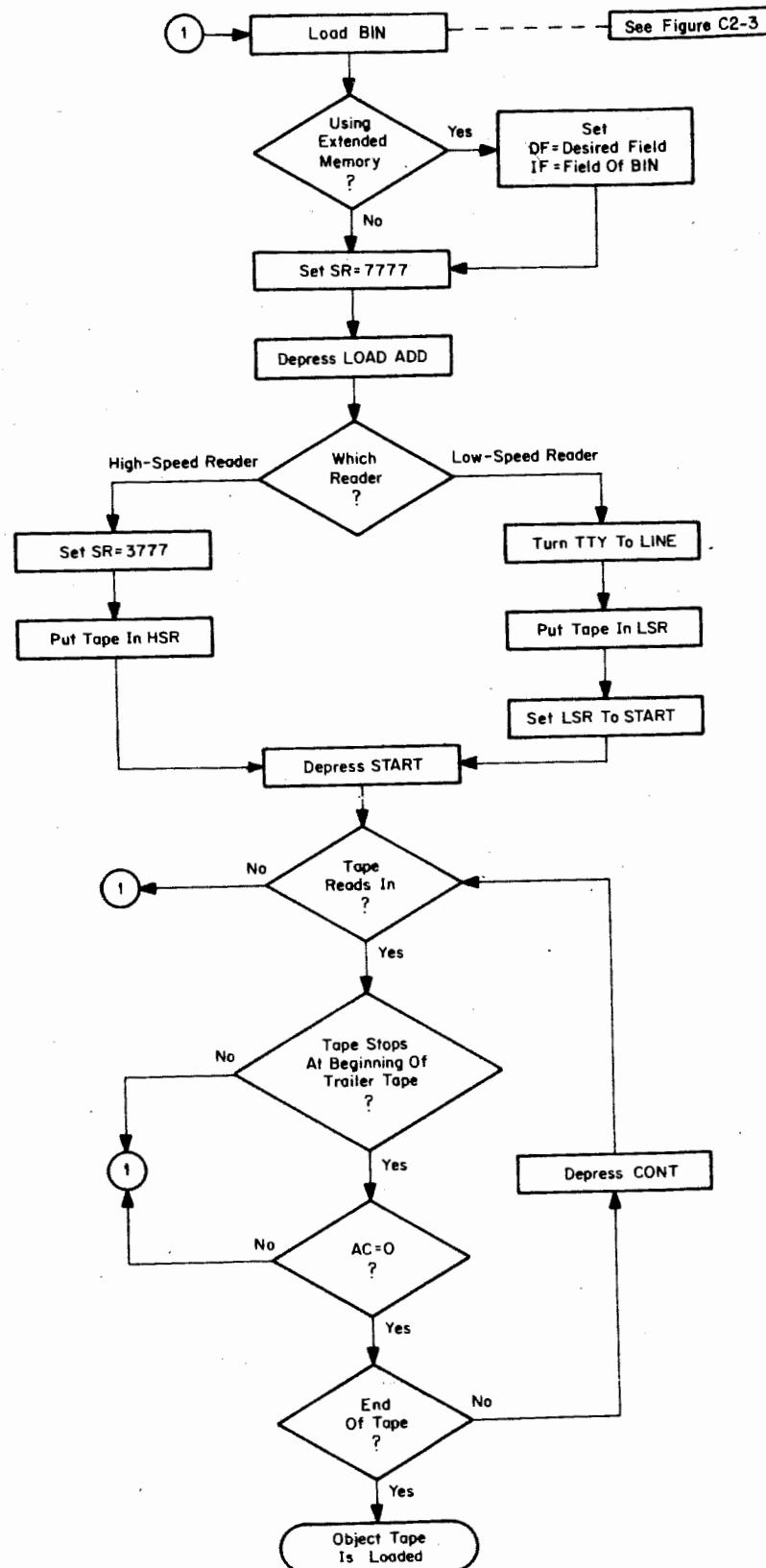
## BINARY (BIN) LOADER

The BIN Loader is used to load programs punched on BIN format paper tape into core memory. It is stored in core memory locations 7625-7752 and 7777 (127<sub>8</sub> locations), and started at location 7777. The RIM Loader is usually used to load a RIM format tape of the BIN Loader.

When the BIN Loader is used to load a binary tape, caution must be exercised to ensure that the tape is started with binary leader code (code 200) under the read station. If the tape is started before this code, the contents of core memory may be lost.



Loading the BIN Loader



Loading A Binary Tape Using BIN

# MATHEMATICAL TABLES

## POWERS OF TWO

	$2^n$	n	$2^{-n}$
	1	0	1.0
	2	1	0.5
	4	2	0.25
	8	3	0.125
	16	4	0.062 5
	32	5	0.031 25
	64	6	0.015 625
	128	7	0.007 812 5
	256	8	0.003 906 25
	512	9	0.001 953 125
	1 024	10	0.000 976 562 5
	2 048	11	0.000 488 281 25
	4 096	12	0.000 244 140 625
	8 192	13	0.000 122 070 312 5
	16 384	14	0.000 061 035 156 25
	32 768	15	0.000 030 517 578 125
	65 536	16	0.000 015 258 789 062 5
	131 072	17	0.000 007 629 394 531 25
	262 144	18	0.000 003 814 697 265 625
	524 288	19	0.000 001 907 348 632 812 5
	1 048 576	20	0.000 000 953 674 316 406 25
	2 097 152	21	0.000 000 476 837 158 203 125
	4 194 304	22	0.000 000 238 418 579 101 562 5
	8 388 608	23	0.000 000 119 209 289 550 781 25
	16 777 216	24	0.000 000 059 604 644 775 390 625
	33 554 432	25	0.000 000 029 802 322 387 695 312 5
	67 108 864	26	0.000 000 014 901 161 193 847 656 25
	134 217 728	27	0.000 000 007 450 580 596 923 808 125
	268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
	536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
	1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
	2 147 483 848	31	0.000 000 000 465 661 287 307 739 257 812 5
	4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 908 25
	8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
	17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
	34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 081 25
	68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
	137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
	274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 154 25
	549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
POWERS OF TWO	1 099. 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
	2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
	4 398 046 511 104	42	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
	8 796 093 022 208	43	0.000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
	17 592 184 044 416	44	0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
	35 184 372 088 832	45	0.000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
	70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
	140 737 488 355 328	47	0.000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
	281 474 976 710 656	48	0.000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625
	562 949 953 421 312	49	0.000 000 000 000 001 776 356 839 400 250 464 677 810 668 945 312 5
	1 125 899 906 842 634	50	0.000 000 000 000 000 888 178 419 700 125 232 338 905 334 472 656 25
	2 251 799 813 985 248	51	0.000 000 000 000 000 444 089 209 850 062 616 169 452 667 234 328 125
	4 503 599 627 370 496	52	0.000 000 000 000 000 222 044 604 925 031 308 084 726 333 668 164 062 5
	9 007 199 254 740 992	53	0.000 000 000 000 000 111 022 302 462 515 654 042 363 166 834 582 031 25
	18 014 398 509 481 984	54	0.000 000 000 000 000 055 511 151 231 257 827 021 171 513 417 041 015 625
	36 028 297 018 963 968	55	0.000 000 000 000 000 027 755 575 615 628 913 510 590 791 708 520 507 812 5
	72 057 594 037 927 936	56	0.000 000 000 000 000 013 877 787 807 814 456 755 215 395 854 260 253 906 25
	144 115 188 075 855 872	57	0.000 000 000 000 000 006 938 893 903 907 228 377 647 697 927 130 126 953 125
	288 230 376 151 711 744	58	0.000 000 000 000 000 003 469 446 951 953 614 188 823 848 963 565 063 476 562 5
	576 460 752 303 423 488	59	0.000 000 000 000 000 001 734 723 475 976 807 094 411 924 481 782 531 738 281 25
	1 152 921 504 606 846 976	60	0.000 000 000 000 000 000 867 361 737 988 403 547 205 962 240 891 265 869 140 625

## **SCALES OF NOTATION**

## **2<sup>x</sup> IN DECIMAL**

$x$	$2^x$	$x$	$2^x$	$x$	$2^x$						
0.001	1.00069	33874	62581	0.01	1.00695	55500	56719	0.1	1.07177	34625	36293
0.002	1.00138	72557	11335	0.02	1.01395	94797	90029	0.2	1.14869	83549	97035
0.003	1.00208	16050	79633	0.03	1.02101	21257	07193	0.3	1.23114	44133	44916
0.004	1.00277	64359	01078	0.04	1.02811	38266	56067	0.4	1.31950	79107	72894
0.005	1.00347	17485	09503	0.05	1.03526	49238	41377	0.5	1.41421	35623	73095
0.006	1.00416	75432	38973	0.06	1.04246	57608	41121	0.6	1.51571	65565	10398
0.007	1.00486	38204	23785	0.07	1.04971	66836	23067	0.7	1.62450	47927	12471
0.008	1.00554	05803	98468	0.08	1.05701	80405	61380	0.8	1.74110	11265	92248
0.009	1.00623	78234	97762	0.09	1.06431	01824	53360	0.9	1.86600	59630	73615

## **10<sup>±n</sup> IN OCTAL**

$10^6$	n	$10^{-n}$	$10^6$	n	$10^{-n}$	
1	0	1.000 000 000 000 000 000 00	112 402 762 000	10	0.000 000 000 006 676 337 66	
12	1	0.063 146 314 631 463 146 31	1 351 035 564 000	11	0.000 000 000 005 375 657 57	
144	2	0.005 075 341 217 270 243 66	16 432 451 210 000	12	0.000 000 000 000 043 136 32	
1 750	3	0.000 406 111 564 570 651 77	221 411 634 520 000	13	0.000 000 000 000 003 411 55	
23 420	4	0.000 032 155 613 530 704 15	2 657 142 036 440 000	14	0.000 000 000 000 000 264 11	
3 303	240	5	0.000 002 476 132 610 706 64	34 327 724 461 500 000	15	0.000 000 000 000 000 022 01
3 641	100	6	0.000 000 206 157 364 055 37	434 157 115 760 200 000	16	0.000 000 000 000 000 001 63
46 113	200	7	0.000 000 015 327 745 152 75	5 432 127 413 542 400 000	17	0.000 000 000 000 000 000 14
575 360	400	8	0.000 000 001 257 143 561 06	67 405 553 164 731 000 000	18	0.000 000 000 000 000 000 01
346 545	000	9	0.000 000 000 104 560 276 41			

## **n log<sub>10</sub> 2, n log<sub>2</sub> 10 IN DECIMAL**

$n$	$n \log_{10} 2$	$n \log_2 10$	$n$	$n \log_{10} 2$	$n \log_2 10$	
1	0.30102	99957	3.32192	80949	18.0617	99740
2	0.60205	99913	6.64385	61898	2.10720	99966
3	0.90308	99870	9.96578	42847	8.240823	99853
4	1.20411	99827	13.28771	23795	2.70926	99610
5	1.50514	99783	16.60964	04744	9.31029	99566
6					19.93156	85693
7					23.25349	66642
8					25.67542	47591
9					29.89735	28540
10					33.21920	09489

# **ADDITION AND MULTIPLICATION TABLES**

## Addition                          Multiplication

## **Binary Scale**

$$0 + 1 = \begin{array}{r} 0 \\ 1 \\ \hline 1 \end{array} + \begin{array}{r} 0 \\ 0 \\ \hline 0 \end{array} = \begin{array}{r} 0 \\ 1 \\ \hline 10 \end{array}$$

$$\begin{array}{r} 0 \times 0 = 0 \\ 1 \times 0 = 0 \\ 1 \times 1 = 1 \end{array}$$

## Octal Scale

	01	02	03	04	05	06	07	
1	02	03	04	05	06	07	10	
2	03	04	05	06	07	10	11	
3	04	05	06	07	10	11	12	
4	05	06	07	10	11	12	13	
5	06	07	10	11	12	13	14	
6	07	10	11	12	13	14	15	
7	10	11	12	13	14	15	16	
	02	03	04	05	06	07		
2	04	06	10	12	14	16		
3	06	11	14	17	22	25		
4	10	14	20	24	30	34		
5	12	17	24	31	36	43		
6	14	22	30	36	44	52		
7	16	25	34	43	52	61		

## MATHEMATICAL CONSTANTS IN OCTAL SCALE

$$\pi = 3.11037 \quad 552421, \quad e = 2.55760 \quad 521305, \quad \gamma = 0.44742 \quad 147707,$$

$$r^{-1} = 0.24276 \quad 01556, \quad e^{-t} = 0.27426 \quad 530661, \quad \ln \gamma = -0.43127 \quad 233602,$$

$$\sqrt{\pi} = 1.61337 \ 611067, \quad \sqrt{e} = 1.51411 \ 230704, \quad \log \gamma = -0.62573 \ 030645,$$

$\log \pi = 1.51544$     $163223$ ,    $\log e = 1.34252$     $166245$ ,    $\ln 2 = 0.54271$     $027760$ ,

$$\sqrt{10} = 3.12305 \quad 407267, \quad \log_{10} 10 = 3.24464 \quad 741136, \quad \ln 10 = 2.23273 \quad 067355,$$

## Octal-Decimal Conversion

The following table gives the multiples of the powers of 8. To convert a number from octal to decimal using the table, add the decimal number opposite the digit value for each digit position. To convert  $40277_8$  to decimal, the following numbers are obtained from the table and added.

Position	Digit	Table entry
5	4	16384
4	0	0
3	2	128
2	7	56
1	7	7
		$16575_{10} = 40277_8$

This process is reversed to convert a number from decimal to octal. Subtract out the largest table entry which allows a positive remainder, then take the column number (position coefficient) of the table entry as the Nth digit of the result, where N is the row number (digit position) of the table entry. Continue this process, operating on the remainder from each step in the next step, until all digits of the result have been found. For example, to convert  $23365_{10}$  to an equivalent octal number:

$$\begin{array}{r}
 23365 \\
 -20480 = 5 \times 8^4 \\
 \hline
 2885 \\
 -2560 = 5 \times 8^3 \\
 \hline
 325 \\
 -320 = 5 \times 8^2 \\
 \hline
 5 \\
 -0 = 0 \times 8^1 \\
 \hline
 5 \\
 -5 = 5 \times 8^0 \\
 \hline
 0 \qquad \qquad \qquad 55505_8 = 23365_{10}
 \end{array}$$

Octal Digit Position/ $8^n$	Position Coefficients (Multipliers)							
	0	1	2	3	4	5	6	7
1st ( $8^0$ )	0	1	2	3	4	5	6	7
2nd ( $8^1$ )	0	8	16	24	32	40	48	56
3rd ( $8^2$ )	0	64	128	192	256	320	384	448
4th ( $8^3$ )	0	512	1,024	1,536	2,048	2,560	3,072	3,584
5th ( $8^4$ )	0	4,096	8,192	12,288	16,384	20,480	24,576	28,672
6th ( $8^5$ )	0	32,768	65,536	98,304	131,072	163,840	196,608	229,376

### Octal-Deciml Integer Conversion Table

		Octal								Decimal									
		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7		
0000	0000	0000	0001	0002	0003	0004	0005	0006	0007	0000	0256	0257	0258	0259	0260	0261	0262	0263	
to	to	0010	0008	0009	0010	0011	0012	0013	0014	0015	0410	0264	0265	0266	0267	0268	0269	0270	0271
0777	0511	0020	0016	0017	0018	0019	0020	0021	0022	0023	0420	0272	0273	0274	0275	0276	0277	0278	0279
(Octal)	(Decimal)	0030	0024	0025	0026	0027	0028	0029	0030	0031	0430	0280	0281	0282	0283	0284	0285	0286	0287
		0040	0032	0033	0034	0035	0036	0037	0038	0039	0440	0288	0289	0290	0291	0292	0293	0294	0295
		0050	0040	0041	0042	0043	0044	0045	0046	0047	0450	0296	0297	0298	0299	0300	0301	0302	0303
		0060	0048	0049	0050	0051	0052	0053	0054	0055	0460	0304	0305	0306	0307	0308	0309	0310	0311
		0070	0056	0057	0058	0059	0060	0061	0062	0063	0470	0312	0313	0314	0315	0316	0317	0318	0319
Octal Decimal																			
10000	4096	0100	0064	0065	0066	0067	0068	0069	0070	0071	0500	0320	0321	0322	0323	0324	0325	0326	0327
20000	8192	0110	0072	0073	0074	0075	0076	0077	0078	0079	0510	0328	0329	0330	0331	0332	0333	0334	0335
30000	12288	0120	0080	0081	0082	0083	0084	0085	0086	0087	0520	0336	0337	0338	0339	0340	0341	0342	0343
40000	16384	0130	0088	0089	0090	0091	0092	0093	0094	0095	0530	0344	0345	0346	0347	0348	0349	0350	0351
50000	20480	0140	0096	0097	0098	0099	0100	0101	0102	0103	0540	0352	0353	0354	0355	0356	0357	0358	0359
60000	24576	0150	0104	0105	0106	0107	0108	0109	0110	0111	0550	0360	0361	0362	0363	0364	0365	0366	0367
70000	28672	0160	0112	0113	0114	0115	0116	0117	0118	0119	0560	0368	0369	0370	0371	0372	0373	0374	0375
		0170	0120	0121	0122	0123	0124	0125	0126	0127	0570	0376	0377	0378	0379	0380	0381	0382	0383
0200	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0600	0384	0385	0386	0387	0388	0389	0390	0391
0210	0136	0137	0138	0139	0140	0141	0142	0143	0144	0145	0610	0392	0393	0394	0395	0396	0397	0398	0399
0220	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0620	0400	0401	0402	0403	0404	0405	0406	0407
0230	0152	0153	0154	0155	0156	0157	0158	0159	0160	0161	0630	0408	0409	0410	0411	0412	0413	0414	0415
0240	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0640	0416	0417	0418	0419	0420	0421	0422	0423
0250	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0650	0424	0425	0426	0427	0428	0429	0430	0431
0260	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0660	0432	0433	0434	0435	0436	0437	0438	0439
0270	0184	0185	0186	0187	0188	0189	0190	0191	0192	0193	0670	0440	0441	0442	0443	0444	0445	0446	0447
0300	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0700	0448	0449	0450	0451	0452	0453	0454	0455
0310	0200	0201	0202	0203	0204	0205	0206	0207	0208	0209	0710	0456	0457	0458	0459	0460	0461	0462	0463
0320	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0720	0464	0465	0466	0467	0468	0469	0470	0471
0330	0216	0217	0218	0219	0220	0221	0222	0223	0224	0225	0730	0472	0473	0474	0475	0476	0477	0478	0479
0340	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0740	0480	0481	0482	0483	0484	0485	0486	0487
0350	0232	0233	0234	0235	0236	0237	0238	0239	0240	0241	0750	0488	0489	0490	0491	0492	0493	0494	0495
0360	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0760	0496	0497	0498	0499	0500	0501	0502	0503
0370	0248	0249	0250	0251	0252	0253	0254	0255	0256	0257	0770	0504	0505	0506	0507	0508	0509	0510	0511
1000	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	1400	0768	0769	0770	0771	0772	0773	0774	0775
to	to	0522	0523	0524	0525	0526	0527	0528	0529	0530	1410	0776	0777	0778	0779	0780	0781	0782	0783
1777	1023	0531	0532	0533	0534	0535	0536	0537	0538	0539	1420	0784	0785	0786	0787	0788	0789	0790	0791
(Octal)	(Decimal)	0540	0544	0545	0546	0547	0548	0549	0550	0551	1430	0792	0793	0794	0795	0796	0797	0798	0799
		0552	0553	0554	0555	0556	0557	0558	0559	0560	1440	0800	0801	0802	0803	0804	0805	0806	0807
		0561	0562	0563	0564	0565	0566	0567	0568	0569	1450	0808	0809	0810	0811	0812	0813	0814	0815
		0568	0569	0570	0571	0572	0573	0574	0575	0576	1460	0816	0817	0818	0819	0820	0821	0822	0823
		0576	0577	0578	0579	0580	0581	0582	0583	0584	1500	0832	0833	0834	0835	0836	0837	0838	0839
		0584	0585	0586	0587	0588	0589	0590	0591	0592	1510	0840	0841	0842	0843	0844	0845	0846	0847
		0592	0593	0594	0595	0596	0597	0598	0599	0600	1520	0848	0849	0850	0851	0852	0853	0854	0855
		0600	0601	0602	0603	0604	0605	0606	0607	0608	1530	0856	0857	0858	0859	0860	0861	0862	0863
		0608	0609	0610	0611	0612	0613	0614	0615	0616	1540	0864	0865	0866	0867	0868	0869	0870	0871
		0616	0617	0618	0619	0620	0621	0622	0623	0624	1550	0872	0873	0874	0875	0876	0877	0878	0879
		0624	0625	0626	0627	0628	0629	0630	0631	0632	1560	0880	0881	0882	0883	0884	0885	0886	0887
		0632	0633	0634	0635	0636	0637	0638	0639	0640	1570	0888	0889	0890	0891	0892	0893	0894	0895
		0640	0641	0642	0643	0644	0645	0646	0647	0648	1600	0896	0897	0898	0899	0900	0901	0902	0903
		0648	0649	0650	0651	0652	0653	0654	0655	0656	1610	0904	0905	0906	0907	0908	0909	0910	0911
		0656	0657	0658	0659	0660	0661	0662	0663	0664	1620	0912	0913	0914	0915	0916	0917	0918	0919
		0664	0665	0666	0667	0668	0669	0670	0671	0672	1630	0920	0921	0922	0923	0924	0925	0926	0927
		0672	0673	0674	0675	0676	0677	0678	0679	0680	1640	0928	0929	0930	0931	0932	0933	0934	0935
		0680	0681	0682	0683	0684	0685	0686	0687	0688	1650	0936	0937	0938	0939	0940	0941	0942	0943
		0688	0689	0690	0691	0692	0693	0694	0695	0696	1660	0944	0945	0946	0947	0948	0949	0950	0951

### Octal-Decimal Integer Conversion Table (continued)

		0 1 2 3 4 5 6 7							0 1 2 3 4 5 6 7						
10 17 al)	1024 to 1535 (Decimal)	2000 1024 1025 1026 1027 1028 1029 1030 1031 2010 1032 1033 1034 1035 1036 1037 1038 1039 2020 1040 1041 1042 1043 1044 1045 1046 1047 2030 1048 1049 1050 1051 1052 1053 1054 1055 2040 1056 1057 1058 1059 1060 1061 1062 1063 2050 1064 1065 1066 1067 1068 1069 1070 1071 2060 1072 1073 1074 1075 1076 1077 1078 1079 2070 1080 1081 1082 1083 1084 1085 1086 1087	2400 1280 1281 1282 1283 1284 1285 1286 1287 2410 1288 1289 1290 1291 1292 1293 1294 1295 2420 1296 1297 1298 1299 1300 1301 1302 1303 2430 1304 1305 1306 1307 1308 1309 1310 1311 2440 1312 1313 1314 1315 1316 1317 1318 1319 2450 1320 1321 1322 1323 1324 1325 1326 1327 2460 1328 1329 1330 1331 1332 1333 1334 1335 2470 1336 1337 1338 1339 1340 1341 1342 1343												
1st Decima 100 - 4096 100 - 8192 100 - 12288 100 - 16384 100 - 20480 100 - 24576 100 - 28672		2100 1088 1089 1090 1091 1092 1093 1094 1095 2110 1096 1097 1098 1099 1100 1101 1102 1103 2120 1104 1105 1106 1107 1108 1109 1110 1111 2130 1112 1113 1114 1115 1116 1117 1118 1119 2140 1120 1121 1122 1123 1124 1125 1126 1127 2150 1128 1129 1130 1131 1132 1133 1134 1135 2160 1136 1137 1138 1139 1140 1141 1142 1143 2170 1144 1145 1146 1147 1148 1149 1150 1151	2500 1344 1345 1346 1347 1348 1349 1350 1351 2510 1352 1353 1354 1355 1356 1357 1358 1359 2520 1360 1361 1362 1363 1364 1365 1366 1367 2530 1368 1369 1370 1371 1372 1373 1374 1375 2540 1376 1377 1378 1379 1380 1381 1382 1383 2550 1384 1385 1386 1387 1388 1389 1390 1391 2560 1392 1393 1394 1395 1396 1397 1398 1399 2570 1400 1401 1402 1403 1404 1405 1406 1407												
		2200 1152 1153 1154 1155 1156 1157 1158 1159 2210 1160 1161 1162 1163 1164 1165 1166 1167 2220 1168 1169 1170 1171 1172 1173 1174 1175 2230 1176 1177 1178 1179 1180 1181 1182 1183 2240 1184 1185 1186 1187 1188 1189 1190 1191 2250 1192 1193 1194 1195 1196 1197 1198 1199 2260 1200 1201 1202 1203 1204 1205 1206 1207 2270 1208 1209 1210 1211 1212 1213 1214 1215	2600 1408 1409 1410 1411 1412 1413 1414 1415 2610 1416 1417 1418 1419 1420 1421 1422 1423 2620 1424 1425 1426 1427 1428 1429 1430 1431 2630 1432 1433 1434 1435 1436 1437 1438 1439 2640 1440 1441 1442 1443 1444 1445 1446 1447 2650 1448 1449 1450 1451 1452 1453 1454 1455 2660 1456 1457 1458 1459 1460 1461 1462 1463 2670 1464 1465 1466 1467 1468 1469 1470 1471												
		2300 1216 1217 1218 1219 1220 1221 1222 1223 2310 1224 1225 1226 1227 1228 1229 1230 1231 2320 1232 1233 1234 1235 1236 1237 1238 1239 2330 1240 1241 1242 1243 1244 1245 1246 1247 2340 1248 1249 1250 1251 1252 1253 1254 1255 2350 1256 1257 1258 1259 1260 1261 1262 1263 2360 1264 1265 1266 1267 1268 1269 1270 1271 2370 1272 1273 1274 1275 1276 1277 1278 1279	2700 1472 1473 1474 1475 1476 1477 1478 1479 2710 1480 1481 1482 1483 1484 1485 1486 1487 2720 1488 1489 1490 1491 1492 1493 1494 1495 2730 1496 1497 1498 1499 1500 1501 1502 1503 2740 1504 1505 1506 1507 1508 1509 1510 1511 2750 1512 1513 1514 1515 1516 1517 1518 1519 2760 1520 1521 1522 1523 1524 1525 1526 1527 2770 1528 1529 1530 1531 1532 1533 1534 1535												
1000 777 ctd)	1536 to 2047 (Decimal)	3000 1536 1537 1538 1539 1540 1541 1542 1543 3010 1544 1545 1546 1547 1548 1549 1550 1551 3020 1552 1553 1554 1555 1556 1557 1558 1559 3030 1560 1561 1562 1563 1564 1565 1566 1567 3040 1568 1569 1570 1571 1572 1573 1574 1575 3050 1576 1577 1578 1579 1580 1581 1582 1583 3060 1584 1585 1586 1587 1588 1589 1590 1591 3070 1592 1593 1594 1595 1596 1597 1598 1599	3400 1792 1793 1794 1795 1796 1797 1798 1799 3410 1800 1801 1802 1803 1804 1805 1806 1807 3420 1808 1809 1810 1811 1812 1813 1814 1815 3430 1816 1817 1818 1819 1820 1821 1822 1823 3440 1824 1825 1826 1827 1828 1829 1830 1831 3450 1832 1833 1834 1835 1836 1837 1838 1839 3460 1840 1841 1842 1843 1844 1845 1846 1847 3470 1848 1849 1850 1851 1852 1853 1854 1855												
		3100 1600 1601 1602 1603 1604 1605 1606 1607 3110 1608 1609 1610 1611 1612 1613 1614 1615 3120 1616 1617 1618 1619 1620 1621 1622 1623 3130 1624 1625 1626 1627 1628 1629 1630 1631 3140 1632 1633 1634 1635 1636 1637 1638 1639 3150 1640 1641 1642 1643 1644 1645 1646 1647 3160 1648 1649 1650 1651 1652 1653 1654 1655 3170 1656 1657 1658 1659 1660 1661 1662 1663	3500 1856 1857 1858 1859 1860 1861 1862 1863 3510 1864 1865 1866 1867 1868 1869 1870 1871 3520 1872 1873 1874 1875 1876 1877 1878 1879 3530 1880 1881 1882 1883 1884 1885 1886 1887 3540 1888 1889 1890 1891 1892 1893 1894 1895 3550 1896 1897 1898 1899 1900 1901 1902 1903 3560 1904 1905 1906 1907 1908 1909 1910 1911 3570 1912 1913 1914 1915 1916 1917 1918 1919												
		3200 1664 1665 1666 1667 1668 1669 1670 1671 3210 1672 1673 1674 1675 1676 1677 1678 1679 3220 1680 1681 1682 1683 1684 1685 1686 1687 3230 1688 1689 1690 1691 1692 1693 1694 1695 3240 1696 1697 1698 1699 1700 1701 1702 1703 3250 1704 1705 1706 1707 1708 1709 1710 1711 3260 1712 1713 1714 1715 1716 1717 1718 1719 3270 1720 1721 1722 1723 1724 1725 1726 1727	3600 1920 1921 1922 1923 1924 1925 1926 1927 3610 1928 1929 1930 1931 1932 1933 1934 1935 3620 1936 1937 1938 1939 1940 1941 1942 1943 3630 1944 1945 1946 1947 1948 1949 1950 1951 3640 1952 1953 1954 1955 1956 1957 1958 1959 3650 1960 1961 1962 1963 1964 1965 1966 1967 3660 1968 1969 1970 1971 1972 1973 1974 1975 3670 1976 1977 1978 1979 1980 1981 1982 1983												
		3300 1728 1729 1730 1731 1732 1733 1734 1735 3310 1736 1737 1738 1739 1740 1741 1742 1743 3320 1744 1745 1746 1747 1748 1749 1750 1751 3330 1752 1753 1754 1755 1756 1757 1758 1759 3340 1760 1761 1762 1763 1764 1765 1766 1767 3350 1768 1769 1770 1771 1772 1773 1774 1775 3360 1776 1777 1778 1779 1780 1781 1782 1783 3370 1784 1785 1786 1787 1788 1789 1790 1791	3700 1984 1985 1986 1987 1988 1989 1990 1991 3710 1992 1993 1994 1995 1996 1997 1998 1999 3720 2000 2001 2002 2003 2004 2005 2006 2007 3730 2008 2009 2010 2011 2012 2013 2014 2015 3740 2016 2017 2018 2019 2020 2021 2022 2023 3750 2024 2025 2026 2027 2028 2029 2030 2031 3760 2032 2033 2034 2035 2036 2037 2038 2039 3770 2040 2041 2042 2043 2044 2045 2046 2047												

## **Octal-Decimal Integer Conversion Table (continued)**

		0	1	2	3	4	5	6	7		0	1	2	3	4	5	6	7	8
4000	2048	4000	2048	2049	2050	2051	2052	2053	2054	2055	4400	2304	2305	2306	2307	2308	2309	2310	2311
to	to	4010	2056	2057	2056	2059	2060	2061	2062	2063	4410	2312	2313	2314	2315	2316	2317	2318	2319
4777	2559	4020	2064	2065	2066	2067	2068	2069	2070	2071	4420	2320	2321	2322	2323	2324	2325	2326	2327
(Octal)	(Decimal)	4030	2072	2073	2074	2075	2076	2077	2078	2079	4430	2328	2329	2330	2331	2332	2333	2334	2335
Octal	Decimal	4040	2080	2081	2082	2083	2084	2085	2086	2087	4440	2336	2337	2338	2339	2340	2341	2342	2343
10000	4096	4050	2088	2089	2090	2091	2092	2093	2094	2095	4450	2344	2345	2346	2347	2348	2349	2350	2351
20000	8192	4060	2096	2097	2098	2099	2100	2101	2102	2103	4460	2352	2353	2354	2355	2356	2357	2358	2359
30000	12288	4070	2104	2105	2106	2107	2108	2109	2110	2111	4470	2360	2361	2362	2363	2364	2365	2366	2367
40000	16384	4100	2112	2113	2114	2115	2116	2117	2118	2119	4500	2368	2369	2370	2371	2372	2373	2374	2375
50000	20480	4110	2120	2121	2122	2123	2124	2125	2126	2127	4510	2376	2377	2378	2379	2380	2381	2382	2383
60000	24576	4120	2128	2129	2130	2131	2132	2133	2134	2135	4520	2384	2385	2386	2387	2388	2389	2390	2391
70000	28672	4130	2136	2137	2138	2139	2140	2141	2142	2143	4530	2392	2393	2394	2395	2396	2397	2398	2399
		4140	2144	2145	2146	2147	2148	2149	2150	2151	4540	2400	2401	2402	2403	2404	2405	2406	2407
		4150	2152	2153	2154	2155	2156	2157	2158	2159	4550	2408	2409	2410	2411	2412	2413	2414	2415
		4160	2160	2161	2162	2163	2164	2165	2166	2167	4560	2416	2417	2418	2419	2420	2421	2422	2423
		4170	2168	2169	2170	2171	2172	2173	2174	2175	4570	2424	2425	2426	2427	2428	2429	2430	2431
		4200	2176	2177	2178	2179	2180	2181	2182	2183	4600	2432	2433	2434	2435	2436	2437	2438	2439
		4210	2184	2185	2186	2187	2188	2189	2190	2191	4610	2440	2441	2442	2443	2444	2445	2446	2447
		4220	2192	2193	2194	2195	2196	2197	2198	2199	4620	2448	2449	2450	2451	2452	2453	2454	2455
		4230	2200	2201	2202	2203	2204	2205	2206	2207	4630	2456	2457	2458	2459	2460	2461	2462	2463
		4240	2208	2209	2210	2211	2212	2213	2214	2215	4640	2464	2465	2466	2467	2468	2469	2470	2471
		4250	2216	2217	2218	2219	2220	2221	2222	2223	4650	2472	2473	2474	2475	2476	2477	2478	2479
		4260	2224	2225	2226	2227	2228	2229	2230	2231	4660	2480	2481	2482	2483	2484	2485	2486	2487
		4270	2232	2233	2234	2235	2236	2237	2238	2239	4670	2488	2489	2490	2491	2492	2493	2494	2495
		4300	2240	2241	2242	2243	2244	2245	2246	2247	4700	2496	2497	2498	2499	2500	2501	2502	2503
		4310	2248	2249	2250	2251	2252	2253	2254	2255	4710	2504	2505	2506	2507	2508	2509	2510	2511
		4320	2256	2257	2258	2259	2260	2261	2262	2263	4720	2512	2513	2514	2515	2516	2517	2518	2519
		4330	2264	2265	2266	2267	2268	2269	2270	2271	4730	2520	2521	2522	2523	2524	2525	2526	2527
		4340	2272	2273	2274	2275	2276	2277	2278	2279	4740	2528	2529	2530	2531	2532	2533	2534	2535
		4350	2280	2281	2282	2283	2284	2285	2286	2287	4750	2536	2537	2538	2539	2540	2541	2542	2543
		4360	2288	2289	2290	2291	2292	2293	2294	2295	4760	2544	2545	2546	2547	2548	2549	2550	2551
		4370	2296	2297	2298	2299	2300	2301	2302	2303	4770	2552	2553	2554	2555	2556	2557	2558	2559
5000	2560	5000	2561	2562	2563	2564	2565	2566	2567	2568	5400	2816	2817	2818	2819	2820	2821	2822	2823
to	to	5010	2568	2569	2570	2571	2572	2573	2574	2575	5410	2824	2825	2826	2827	2828	2829	2830	2831
5777	3071	5020	2576	2577	2578	2579	2580	2581	2582	2583	5420	2832	2833	2834	2835	2836	2837	2838	2839
(Octal)	(Decimal)	5030	2584	2585	2586	2587	2588	2589	2590	2591	5430	2840	2841	2842	2843	2844	2845	2846	2847
5040	2592	2593	2594	2595	2596	2597	2598	2599	2599	5440	2848	2849	2850	2851	2852	2853	2854	2855	
5050	2600	2601	2602	2603	2604	2605	2606	2607	2607	5450	2856	2857	2858	2859	2860	2861	2862	2863	
5060	2608	2609	2610	2611	2612	2613	2614	2615	2615	5460	2864	2865	2866	2867	2868	2869	2870	2871	
5070	2616	2617	2618	2619	2620	2621	2622	2623	2623	5470	2872	2873	2874	2875	2876	2877	2878	2879	
		5100	2624	2625	2626	2627	2628	2629	2630	2631	5500	2880	2881	2882	2883	2884	2885	2886	2887
		5110	2632	2633	2634	2635	2636	2637	2638	2639	5510	2888	2889	2890	2891	2892	2893	2894	2895
		5120	2640	2641	2642	2643	2644	2645	2646	2647	5520	2896	2897	2898	2899	2900	2901	2902	2903
		5130	2648	2649	2650	2651	2652	2653	2654	2655	5530	2904	2905	2906	2907	2908	2909	2910	2911
		5140	2656	2657	2658	2659	2660	2661	2662	2663	5540	2912	2913	2914	2915	2916	2917	2918	2919
		5150	2664	2665	2666	2667	2668	2669	2670	2671	5550	2920	2921	2922	2923	2924	2925	2926	2927
		5160	2672	2673	2674	2675	2676	2677	2678	2679	5560	2928	2929	2930	2931	2932	2933	2934	2935
		5170	2680	2681	2682	2683	2684	2685	2686	2687	5570	2936	2937	2938	2939	2940	2941	2942	2943
		5200	2688	2689	2690	2691	2692	2693	2694	2695	5600	2944	2945	2946	2947	2948	2949	2950	2951
		5210	2696	2697	2698	2699	2700	2701	2702	2703	5610	2952	2953	2954	2955	2956	2957	2958	2959
		5220	2704	2705	2706	2707	2708	2709	2710	2711	5620	2960	2961	2962	2963	2964	2965	2966	2967
		5230	2712	2713	2714	2715	2716	2717	2718	2719	5630	2968	2969	2970	2971	2972	2973	2974	2975
		5240	2720	2721	2722	2723	2724	2725	2726	2727	5640	2976	2977	2978	2979	2980	2981	2982	2983
		5250	2728	2729	2730	2731	2732	2733	2734	2735	5650	2984	2985	2986	2987	2988	2989	2990	2991
		5260	2736	2737	2738	2739	2740	2741	2742	2743	5660	2992	2993	2994	2995	2996	2997	2998	2999
		5270	2744	2745	2746	2747	2748	2749	2750	2751	5670	3000	3001	3002	3003	3004	3005	3006	3007
		5300	2752	2753	2754	2755	2756	2757	2758	2759	5700	3008	3009	3010	3011	3012	3013	3014	3015
		5310	2760	2761	2762	2763	2764	2765	2766	2767	5710	3016	3017	3018	3019	3020	3021	3022	3023
		5320	2768	2769	2770	2771	2772	2773	2774	2775	5720	3024	3025	3026	3027	3028	3029	3030	3031
		5330	2776	2777	2778	2779	2780	2781	2782	2783	5730	3032	3033	3034	3035	3036	3037	3038	3039
		5340	2784	2785	2786	2787	2788	2789	2790	2791	5740	3040	3041	3042	3043	3044	3045	3046	3047
		5350	2792	2793	2794	2795	2796	2797	2798	2799	5750	3048	3049	3050	3051	3052	3053	3054	3055
		5360	2800	2801	2802	2803	2804	2805	2806	2807	5760	3056	3057	3058	3059	3060	3061	3062	3063
		5370	2808	2809	2810	2811	2812	2813	2814	2815	5770	3064	3065	3066	3067	3068	3069	3070	3071

Octal-Decimal Integer Conversion Table (Continued)							
Octal		Decimal					
Octal	Octal	0	1	2	3	4	5
0000	3072	0	1	2	3	4	5
0010	3080	3081	3082	3083	3084	3085	3086
0020	3090	3091	3092	3093	3094	3095	3096
0030	3096	3097	3098	3099	3100	3101	3102
0040	3104	3105	3106	3107	3108	3109	3110
0050	3112	3113	3114	3115	3116	3117	3118
0060	3120	3121	3122	3123	3124	3125	3126
0070	3128	3129	3130	3131	3132	3133	3134
0080	3136	3137	3138	3139	3140	3141	3142
0090	3144	3145	3146	3147	3148	3149	3150
0100	3152	3153	3154	3155	3156	3157	3158
0110	3160	3161	3162	3163	3164	3165	3166
0120	3168	3169	3170	3171	3172	3173	3174
0130	3186	3187	3188	3189	3190	3191	3192
0140	3194	3195	3196	3197	3198	3199	3200
0150	3202	3203	3204	3205	3206	3207	3208
0160	3210	3211	3212	3213	3214	3215	3216
0170	3218	3219	3220	3221	3222	3223	3224
0180	3226	3227	3228	3229	3230	3231	3232
0190	3234	3235	3236	3237	3238	3239	3240
0200	3242	3243	3244	3245	3246	3247	3248
0210	3250	3251	3252	3253	3254	3255	3256
0220	3258	3259	3260	3261	3262	3263	3264
0230	3276	3277	3278	3279	3280	3281	3282
0240	3284	3285	3286	3287	3288	3289	3290
0250	3292	3293	3294	3295	3296	3297	3298
0260	3300	3301	3302	3303	3304	3305	3306
0270	3308	3309	3310	3311	3312	3313	3314
0280	3316	3317	3318	3319	3320	3321	3322
0290	3324	3325	3326	3327	3328	3329	3330
0300	3332	3333	3334	3335	3336	3337	3338
0310	3340	3341	3342	3343	3344	3345	3346
0320	3348	3349	3350	3351	3352	3353	3354
0330	3366	3367	3368	3369	3370	3371	3372
0340	3374	3375	3376	3377	3378	3379	3380
0350	3382	3383	3384	3385	3386	3387	3388
0360	3390	3391	3392	3393	3394	3395	3396
0370	3398	3399	3400	3401	3402	3403	3404
0380	3416	3417	3418	3419	3420	3421	3422
0390	3424	3425	3426	3427	3428	3429	3430
0400	3432	3433	3434	3435	3436	3437	3438
0410	3440	3441	3442	3443	3444	3445	3446
0420	3448	3449	3450	3451	3452	3453	3454
0430	3456	3457	3458	3459	3460	3461	3462
0440	3464	3465	3466	3467	3468	3469	3470
0450	3472	3473	3474	3475	3476	3477	3478
0460	3480	3481	3482	3483	3484	3485	3486
0470	3488	3489	3490	3491	3492	3493	3494
0480	3496	3497	3498	3499	3500	3501	3502
0490	3508	3509	3510	3511	3512	3513	3514
0500	3516	3517	3518	3519	3520	3521	3522
0510	3524	3525	3526	3527	3528	3529	3530
0520	3532	3533	3534	3535	3536	3537	3538
0530	3540	3541	3542	3543	3544	3545	3546
0540	3548	3549	3550	3551	3552	3553	3554
0550	3558	3559	3560	3561	3562	3563	3564
0560	3576	3577	3578	3579	3580	3581	3582
0570	3588	3589	3590	3591	3592	3593	3594
0580	3596	3597	3598	3599	3600	3601	3602
0590	3608	3609	3610	3611	3612	3613	3614
0600	3620	3621	3622	3623	3624	3625	3626
0610	3632	3633	3634	3635	3636	3637	3638
0620	3644	3645	3646	3647	3648	3649	3650
0630	3656	3657	3658	3659	3660	3661	3662
0640	3668	3669	3670	3671	3672	3673	3674
0650	3680	3681	3682	3683	3684	3685	3686
0660	3692	3693	3694	3695	3696	3697	3698
0670	3704	3705	3706	3707	3708	3709	3710
0680	3716	3717	3718	3719	3720	3721	3722
0690	3728	3729	3730	3731	3732	3733	3734
0700	3740	3741	3742	3743	3744	3745	3746
0710	3752	3753	3754	3755	3756	3757	3758
0720	3764	3765	3766	3767	3768	3769	3770
0730	3776	3777	3778	3779	3780	3781	3782
0740	3788	3789	3790	3791	3792	3793	3794
0750	3800	3801	3802	3803	3804	3805	3806
0760	3812	3813	3814	3815	3816	3817	3818
0770	3824	3825	3826	3827	3828	3829	3830
0780	3836	3837	3838	3839	3840	3841	3842
0790	3848	3849	3850	3851	3852	3853	3854
0800	3860	3861	3862	3863	3864	3865	3866
0810	3872	3873	3874	3875	3876	3877	3878
0820	3884	3885	3886	3887	3888	3889	3890
0830	3896	3897	3898	3899	3900	3901	3902
0840	3908	3909	3910	3911	3912	3913	3914
0850	3920	3921	3922	3923	3924	3925	3926
0860	3932	3933	3934	3935	3936	3937	3938
0870	3944	3945	3946	3947	3948	3949	3950
0880	3956	3957	3958	3959	3960	3961	3962
0890	3968	3969	3970	3971	3972	3973	3974
0900	3980	3981	3982	3983	3984	3985	3986
0910	3992	3993	3994	3995	3996	3997	3998
0920	4004	4005	4006	4007	4008	4009	4010
0930	4016	4017	4018	4019	4020	4021	4022
0940	4028	4029	4030	4031	4032	4033	4034
0950	4040	4041	4042	4043	4044	4045	4046
0960	4052	4053	4054	4055	4056	4057	4058
0970	4064	4065	4066	4067	4068	4069	4070
0980	4076	4077	4078	4079	4080	4081	4082
0990	4092	4093	4094	4095	4096	4097	4098
1000	4104	4105	4106	4107	4108	4109	4110
1010	4122	4123	4124	4125	4126	4127	4128
1020	4140	4141	4142	4143	4144	4145	4146
1030	4158	4159	4160	4161	4162	4163	4164
1040	4176	4177	4178	4179	4180	4181	4182
1050	4194	4195	4196	4197	4198	4199	4200
1060	4212	4213	4214	4215	4216	4217	4218
1070	4230	4231	4232	4233	4234	4235	4236
1080	4248	4249	4250	4251	4252	4253	4254
1090	4266	4267	4268	4269	4270	4271	4272
1100	4284	4285	4286	4287	4288	4289	4290
1110	4302	4303	4304	4305	4306	4307	4308
1120	4320	4321	4322	4323	4324	4325	4326
1130	4338	4339	4340	4341	4342	4343	4344
1140	4356	4357	4358	4359	4360	4361	4362
1150	4374	4375	4376	4377	4378	4379	4380
1160	4392	4393	4394	4395	4396	4397	4398
1170	4410	4411	4412	4413	4414	4415	4416
1180	4428	4429	4430	4431	4432	4433	4434
1190	4446	4447	4448	4449	4450	4451	4452
1200	4464	4465	4466	4467	4468	4469	4470
1210	4482	4483	4484	4485	4486	4487	4488
1220	4500	4501	4502	4503	4504	4505	4506
1230	4518	4519	4520	4521	4522	4523	4524
1240	4536	4537	4538	4539	4540	4541	4542
1250	4554	4555	4556	4557	4558	4559	4560
1260	4572	4573	4574	4575	4576	4577	4578
1270	4590	4591	4592	4593	4594	4595	4596
1280	4608	4609	4610	4611	4612	4613	4614
1290	4626	4627	4628	4629	4630	4631	4632
1300	4644	4645	4646	4647	4648	4649	4650
1310	4662	4663	4664	4665	4666	4667	4668
1320	4680	4681	4682	4683	4684	4685	4686
1330	4708	4709	4710	4711	4712	4713	4714
1340	4726	4727	4728	4729	4730	4731	4732
1350	4744	4745	4746	4747	4748	4749	4750
1360							

### Octal-Decimal Fraction Conversion Table

Octal	Decimal	Octal	Decimal	Octal	Decimal	Octal	Decimal
.000	.000000	.100	.125000	.200	.250000	.300	.37500
.001	.001953	.101	.126953	.201	.251953	.301	.37695
.002	.003906	.102	.128906	.202	.253906	.302	.37890
.003	.005859	.103	.130859	.203	.255859	.303	.38085
.004	.007812	.104	.132812	.204	.257812	.304	.38281
.005	.009765	.105	.134765	.205	.259765	.305	.38476
.006	.011718	.106	.136718	.206	.261718	.306	.38671
.007	.013671	.107	.138671	.207	.263671	.307	.38867
.010	.015625	.110	.140625	.210	.265625	.310	.390625
.011	.017578	.111	.142578	.211	.267578	.311	.392578
.012	.019531	.112	.144531	.212	.269531	.312	.394531
.013	.021484	.113	.146484	.213	.271484	.313	.396484
.014	.023437	.114	.148437	.214	.273437	.314	.398437
.015	.025390	.115	.150390	.215	.275390	.315	.400390
.016	.027343	.116	.152343	.216	.277343	.316	.402343
.017	.029296	.117	.154296	.217	.279296	.317	.404296
.020	.031250	.120	.156250	.220	.281250	.320	.406250
.021	.033203	.121	.158203	.221	.283203	.321	.408203
.022	.035156	.122	.160156	.222	.285156	.322	.410156
.023	.037109	.123	.162109	.223	.287109	.323	.412109
.024	.039062	.124	.164062	.224	.289062	.324	.414062
.025	.041015	.125	.166015	.225	.291015	.325	.416015
.026	.042968	.126	.167968	.226	.292968	.326	.417968
.027	.044921	.127	.169921	.227	.294921	.327	.419921
.030	.046875	.130	.171875	.230	.296875	.330	.421875
.031	.048828	.131	.173828	.231	.298828	.331	.423828
.032	.050781	.132	.175781	.232	.300781	.332	.425781
.033	.052734	.133	.177734	.233	.302734	.333	.427734
.034	.054687	.134	.179687	.234	.304687	.334	.429687
.035	.056640	.135	.181640	.235	.306640	.335	.431640
.036	.058593	.136	.183593	.236	.308593	.336	.433593
.037	.060546	.137	.185546	.237	.310546	.337	.435546
.040	.062500	.140	.187500	.240	.312500	.340	.437500
.041	.064453	.141	.189453	.241	.314453	.341	.439453
.042	.066406	.142	.191406	.242	.316406	.342	.441406
.043	.068359	.143	.193359	.243	.318359	.343	.443359
.044	.070312	.144	.195312	.244	.320312	.344	.445312
.045	.072265	.145	.197265	.245	.322265	.345	.447265
.046	.074218	.146	.199218	.246	.324218	.346	.449218
.047	.076171	.147	.201171	.247	.326171	.347	.451171
.050	.078125	.150	.203125	.250	.328125	.350	.453125
.051	.080078	.151	.205078	.251	.330078	.351	.455078
.052	.082031	.152	.207031	.252	.332031	.352	.457031
.053	.083984	.153	.208984	.253	.333984	.353	.458984
.054	.085937	.154	.210937	.254	.335937	.354	.460937
.055	.087890	.155	.212890	.255	.337890	.355	.462890
.056	.089843	.156	.214843	.256	.339843	.356	.464843
.057	.091796	.157	.216796	.257	.341796	.357	.466796
.060	.093750	.160	.218750	.260	.343750	.360	.468750
.061	.095703	.161	.220703	.261	.345703	.361	.470703
.062	.097656	.162	.222656	.262	.347656	.362	.472656
.063	.099609	.163	.224609	.263	.349609	.363	.474609
.064	.101562	.164	.226562	.264	.351562	.364	.476562
.065	.103515	.165	.228515	.265	.353515	.365	.478515
.066	.105468	.166	.230468	.266	.355468	.366	.480468
.067	.107421	.167	.232421	.267	.357421	.367	.482421
.070	.109375	.170	.234375	.270	.359375	.370	.484375
.071	.111328	.171	.236328	.271	.361328	.371	.486328
.072	.113281	.172	.238281	.272	.363281	.372	.488281
.073	.115234	.173	.240234	.273	.365234	.373	.490234
.074	.117187	.174	.242187	.274	.367187	.374	.492187
.075	.119140	.175	.244140	.275	.369140	.375	.494140
.076	.121093	.176	.246093	.276	.371093	.376	.496093
.077	.123046	.177	.248046	.277	.373046	.377	.498046

**Octal-Decimal Fraction Conversion Table (continued)**

Octal	Decimal	Octal	Decimal	Octal	Decimal	Octal	Decimal
.000000	.000000	.000100	.000244	.000200	.000488	.000300	.000732
.000001	.000003	.000101	.000247	.000201	.000492	.000301	.000736
.000002	.000007	.000102	.000251	.000202	.000495	.000302	.000740
.000003	.000011	.000103	.000255	.000203	.000499	.000303	.000743
.000004	.000015	.000104	.000259	.000204	.000503	.000304	.000747
.000005	.000019	.000105	.000263	.000205	.000507	.000305	.000751
.000006	.000022	.000106	.000267	.000206	.000511	.000306	.000755
.000007	.000026	.000107	.000270	.000207	.000514	.000307	.000759
.000010	.000030	.000110	.000274	.000210	.000518	.000310	.000762
.000011	.000034	.000111	.000278	.000211	.000522	.000311	.000766
.000012	.000038	.000112	.000282	.000212	.000526	.000312	.000770
.000013	.000041	.000113	.000286	.000213	.000530	.000313	.000774
.000014	.000045	.000114	.000289	.000214	.000534	.000314	.000778
.000015	.000049	.000115	.000293	.000215	.000537	.000315	.000782
.000016	.000053	.000116	.000297	.000216	.000541	.000316	.000785
.000017	.000057	.000117	.000301	.000217	.000545	.000317	.000789
.000020	.000061	.000120	.000305	.000220	.000549	.000320	.000793
.000021	.000064	.000121	.000308	.000221	.000553	.000321	.000797
.000022	.000068	.000122	.000312	.000222	.000558	.000322	.000801
.000023	.000072	.000123	.000316	.000223	.000560	.000323	.000805
.000024	.000076	.000124	.000320	.000224	.000564	.000324	.000808
.000025	.000080	.000125	.000324	.000225	.000568	.000325	.000812
.000026	.000083	.000126	.000328	.000226	.000572	.000326	.000816
.000027	.000087	.000127	.000331	.000227	.000576	.000327	.000820
.000030	.000091	.000130	.000335	.000230	.000579	.000330	.000823
.000031	.000095	.000131	.000339	.000231	.000583	.000331	.000827
.000032	.000099	.000132	.000343	.000232	.000587	.000332	.000831
.000033	.000102	.000133	.000347	.000233	.000591	.000333	.000835
.000034	.000106	.000134	.000350	.000234	.000595	.000334	.000839
.000035	.000110	.000135	.000354	.000235	.000598	.000335	.000843
.000036	.000114	.000136	.000358	.000236	.000602	.000336	.000846
.000037	.000118	.000137	.000362	.000237	.000606	.000337	.000850
.000040	.000122	.000140	.000366	.000240	.000610	.000340	.000854
.000041	.000125	.000141	.000370	.000241	.000614	.000341	.000858
.000042	.000129	.000142	.000373	.000242	.000617	.000342	.000862
.000043	.000133	.000143	.000377	.000243	.000621	.000343	.000865
.000044	.000137	.000144	.000381	.000244	.000625	.000344	.000869
.000045	.000141	.000145	.000385	.000245	.000629	.000345	.000873
.000046	.000144	.000146	.000389	.000246	.000633	.000346	.000877
.000047	.000148	.000147	.000392	.000247	.000637	.000347	.000881
.000050	.000152	.000150	.000396	.000250	.000640	.000350	.000885
.000051	.000156	.000151	.000400	.000251	.000644	.000351	.000888
.000052	.000160	.000152	.000404	.000252	.000648	.000352	.000892
.000053	.000164	.000153	.000408	.000253	.000652	.000353	.000896
.000054	.000167	.000154	.000411	.000254	.000656	.000354	.000900
.000055	.000171	.000155	.000415	.000255	.000659	.000355	.000904
.000056	.000175	.000156	.000419	.000256	.000663	.000356	.000907
.000057	.000179	.000157	.000423	.000257	.000667	.000357	.000911
.000060	.000183	.000160	.000427	.000260	.000671	.000360	.000915
.000061	.000186	.000161	.000431	.000261	.000675	.000361	.000919
.000062	.000190	.000162	.000434	.000262	.000679	.000362	.000923
.000063	.000194	.000163	.000438	.000263	.000682	.000363	.000926
.000064	.000198	.000164	.000442	.000264	.000686	.000364	.000930
.000065	.000202	.000165	.000446	.000265	.000690	.000365	.000934
.000066	.000205	.000166	.000450	.000266	.000694	.000366	.000938
.000067	.000209	.000167	.000453	.000267	.000698	.000367	.000942
.000070	.000213	.000170	.000457	.000270	.000701	.000370	.000946
.000071	.000217	.000171	.000461	.000271	.000705	.000371	.000949
.000072	.000221	.000172	.000465	.000272	.000709	.000372	.000953
.000073	.000225	.000173	.000469	.000273	.000713	.000373	.000957
.000074	.000228	.000174	.000473	.000274	.000717	.000374	.000961
.000075	.000232	.000175	.000476	.000275	.000720	.000375	.000965
.000076	.000236	.000176	.000480	.000276	.000724	.000376	.000968
.000077	.000240	.000177	.000484	.000277	.000728	.000377	.000972

### Octal-Decimal Fraction Conversion Table (continued)

Octal	Decimal	Octal	Decimal	Octal	Decimal	Octal	Decimal
.000400	.000976	.000500	.001220	.000600	.001464	.000700	.001700
.000401	.000980	.000501	.001224	.000601	.001468	.000701	.001704
.000402	.000984	.000502	.001228	.000602	.001472	.000702	.001716
.000403	.000988	.000503	.001232	.000603	.001476	.000703	.001720
.000404	.000991	.000504	.001235	.000604	.001480	.000704	.001724
.000405	.000995	.000505	.001239	.000605	.001483	.000705	.001728
.000406	.000999	.000506	.001243	.000606	.001487	.000706	.001731
.000407	.001003	.000507	.001247	.000607	.001491	.000707	.001735
.000410	.001007	.000510	.001251	.000610	.001495	.000710	.001739
.000411	.00101C	.000511	.001255	.000611	.001499	.000711	.001743
.000412	.001014	.000512	.001258	.000612	.001502	.000712	.001747
.000413	.001018	.000513	.001262	.000613	.001506	.000713	.001750
.000414	.001022	.000514	.001266	.000614	.001510	.000714	.001754
.000415	.001026	.000515	.001270	.000615	.001514	.000715	.001758
.000416	.001029	.000516	.001274	.000616	.001518	.000716	.001762
.000417	.001033	.000517	.001277	.000617	.001522	.000717	.001766
.000420	.001037	.000520	.001281	.000620	.001525	.000720	.001770
.000421	.001041	.000521	.001285	.000621	.001529	.000721	.001773
.000422	.001045	.000522	.001289	.000622	.001533	.000722	.001777
.000423	.001049	.000523	.001293	.000623	.001537	.000723	.001781
.000424	.001052	.000524	.001296	.000624	.001541	.000724	.001785
.000425	.001056	.000525	.001300	.000625	.001544	.000725	.001789
.000426	.001060	.000526	.001304	.000626	.001548	.000726	.001792
.000427	.001064	.000527	.001308	.000627	.001552	.000727	.001796
.000430	.001068	.000530	.001312	.000630	.001556	.000730	.001800
.000431	.001071	.000531	.001316	.000631	.001560	.000731	.001804
.000432	.001075	.000532	.001319	.000632	.001564	.000732	.001808
.000433	.001079	.000533	.001323	.000633	.001567	.000733	.001811
.000434	.001083	.000534	.001327	.000634	.001571	.000734	.001815
.000435	.001087	.000535	.001331	.000635	.001575	.000735	.001819
.000436	.001091	.000536	.001335	.000636	.001579	.000736	.001823
.000437	.001094	.000537	.001338	.000637	.001583	.000737	.001827
.000440	.001098	.000540	.001342	.000640	.001586	.000740	.001831
.000441	.001102	.000541	.001346	.000641	.001590	.000741	.001834
.000442	.001106	.000542	.001350	.000642	.001594	.000742	.001838
.000443	.001110	.000543	.001354	.000643	.001598	.000743	.001842
.000444	.001113	.000544	.001358	.000644	.001602	.000744	.001846
.000445	.001117	.000545	.001361	.000645	.001605	.000745	.001850
.000446	.001121	.000546	.001365	.000646	.001609	.000746	.001853
.000447	.001125	.000547	.001369	.000647	.001613	.000747	.001857
.000450	.001129	.000550	.001373	.000650	.001617	.000750	.001861
.000451	.001132	.000551	.001377	.000651	.001621	.000751	.001865
.000452	.001136	.000552	.001380	.000652	.001625	.000752	.001869
.000453	.001140	.000553	.001384	.000653	.001628	.000753	.001873
.000454	.001144	.000554	.001388	.000654	.001632	.000754	.001876
.000455	.001148	.000555	.001392	.000655	.001636	.000755	.001880
.000456	.001152	.000556	.001396	.000656	.001640	.000756	.001884
.000457	.001155	.000557	.001399	.000657	.001644	.000757	.001888
.000460	.001159	.000560	.001403	.000660	.001647	.000760	.001892
.000461	.001163	.000561	.001407	.000661	.001651	.000761	.001895
.000462	.001167	.000562	.001411	.000662	.001655	.000762	.001899
.000463	.001171	.000563	.001415	.000663	.001659	.000763	.001903
.000464	.001174	.000564	.001419	.000664	.001663	.000764	.001907
.000465	.001178	.000565	.001422	.000665	.001667	.000765	.001911
.000466	.001182	.000566	.001426	.000666	.001670	.000766	.001914
.000467	.001186	.000567	.001430	.000667	.001674	.000767	.001918
.000470	.001190	.000570	.001434	.000670	.001678	.000770	.001922
.000471	.001194	.000571	.001438	.000671	.001682	.000771	.001926
.000472	.001197	.000572	.001441	.000672	.001686	.000772	.001930
.000473	.001201	.000573	.001445	.000673	.001689	.000773	.001934
.000474	.001205	.000574	.001449	.000674	.001693	.000774	.001937
.000475	.001209	.000575	.001453	.000675	.001697	.000775	.001941
.000476	.001213	.000576	.001457	.000676	.001701	.000776	.001945
.000477	.001216	.000577	.001461	.000677	.001705	.000777	.001949

**Table 1-1 PDP-8/E Memory Reference Instructions  
(Refer to Chapter 3)**

Mnemonic Symbol	Octal Code	Indicators	Execution Times			Operation
			Direct Address	Indirect Address	Auto- Indexed	
AND Y	0	IR = 0,F,E	2.6	3.8	4.0	Logical AND between Y and AC
TAD Y	1	IR = 1,F,E	2.6	3.8	4.0	Two's complement Add Y to AC
ISZ Y	2	IR = 2,F,E	2.6	3.8	4.0	Increment Y and skip if zero
DCA Y	3	IR = 3,F,E	2.6	3.8	4.0	Deposit at Y and clear AC
JMS Y	4	IR = 4,F,E	2.6	3.8	4.0	Jump to subroutine at Y
JMP Y	5	IR = 5,F	1.2	2.4	2.6	Jump to Y

**Table 1-2 Loading Constants Into The Accumulator**

Mnemonic	Decimal Constant	Octal Code	Instructions Combined
NL0000 =	0	7300	CLA CLL
NL0001 =	1	7301	CLA CLL IAC
NL0002 =	2	7305 (or)	CLA CLL IAC RAL
NL0002 =	2	7326	CLA CLL CML RTL
NL0003 =	3	7325	CLA CLL CML IAC RAL
NL0004 =	4	7307	CLA CLL IAC RTL
NL0006 =	6	7327	CLA CLL CML IAC RTL
NL0100 =	64	7203	CLA IAC BSW
NL2000 =	1024	7332	CLA CLL CML RTR
NL3777 =	2047	7350	CLA CLL CMA RAR
NL4000 =	-0	7330	CLA CLL CML RAR
NL5777 =	-1025	7352	CLA CLL CMA RTR
NL6000 =	-1024	7333	CLA CLL CML IAC RTL
NL7775 =	-3	7346	CLA CLL CMA RTL
NL7776 =	-2	7344	CLA CLL CMA RAL
NL7777 =	-1	7340	CLA CLL CMA

**Table 1-3 Group 1 Operate Microinstructions**

Mnemonic Symbol	Octal Code	Sequence	Operation
NOP	7000	—	No operation. Causes a 1.2 $\mu$ s program delay.
IAC	7001	3	Increment AC. The content of the AC is incremented by one in two's complement arithmetic.
RAL	7004	4	Rotate AC and L left. The content of the AC and the L are rotated left one place.
RTL	7006	4	Rotate two places to the left. Equivalent to two successive RAL operations.
RAR	7010	4	Rotate AC and L right. The content of the AC and L are rotated right one place.
RTR	7012	4	Rotate two places to the right. Equivalent to two successive RAR operations.
BSW	7002	4	Byte swap.
CML	7020	2	Complement L.
CMA	7040	2	Complement AC. The content of the AC is set to the one's complement of its current content.
CIA	7041	2, 3	Complement and increment accumulator. Used to form two's complement.
CLL	7100	1	Clear L.
CLL RAL	7104	1, 4	Shift positive number one left.
CLL RTL	7106	1, 4	Clear link, rotate two left.
CLL RAR	7110	1, 4	Shift positive number one right.
CLL RTR	7112	1, 4	Clear link, rotate two right.
STL	7120	1, 2	Set link. The L is set to contain a binary 1.
CLA	7200	1	Clear AC. To be used alone or in OPR 1 combinations.
CLA IAC	7201	1, 3	Set AC = 1.
GLK	7204	1, 4	Get link. Transfer L into AC11.
CLA CLL	7300	1	Clear AC and L.
STA	7240	2	Set AC = 1. Each bit of the AC is set to contain a 1.

**Table 1-4 Group 2 Operate Microinstructions**

Mnemonic Symbol	Octal Code	Sequence	Operation
HLT	7402	3	Halt. Stops the program after completion of the cycle in process. If this instruction is combined with others in the OPR 2 group the other operations are completed before the end of the cycle.
OSR	7404	3	OR with switch register. The OR function is performed between the content of the SR and the content of the AC, with the result left in the AC.
SKP	7410	1	Skip, unconditional. The next instruction is skipped.
SNL	7420	1	Skip if $L \neq 0$ .
SZL	7430	1	Skip if $L = 0$ .
SZA	7440	1	Skip if $AC = 0$ .
SNA	7450	1	Skip if $AC \neq 0$ .
SZA SNL	7460	1	Skip if $AC = 0$ , or $L \neq 1$ , or both.
SNA SZL	7470	1	Skip if $AC \neq 0$ and $L = 0$ .
SMA	7500	1	Skip on minus AC. If the content of the AC is a negative number, the next instruction is skipped.
SPA	7510	1	Skip on positive AC. If the content of the AC is a positive number, including zero, the next instruction is skipped.
SMA SNL	7520	1	Skip if $AC < 0$ , or $L = 1$ , or both.
SPA SZL	7530	1	Skip if $AC \geq 0$ and if $L = 0$ .
SMA SZA	7540	1	Skip if $AC \leq 0$ .
SPA SNA	7550	1	Skip if $AC > 0$ .
CLA	7600	2	Clear AC. To be used alone or in OPR 2 combinations.
LAS	7604	1, 3	Load AC with SR.
SZA CLA	7640	1, 2	Skip if $AC = 0$ , then clear AC.
SNA CLA	7650	1, 2	Skip if $AC \neq 0$ , then clear AC.
SMA CLA	7700	1, 2	Skip if $AC < 0$ , then clear AC.
SPA CLA	7710	1, 2	Skip if $AC \geq 0$ , then clear AC.

**Table 1-5 Group 3 Operate Microinstructions**

Mnemonic Symbol	Octal Code	Operation
NOP	7401	No Operation
MQL	7421	Load Multiplier Quotient
MQA	7501	Multiplier Quotient OR into Accumulator
SWP	7521	Swap Accumulator and Multiplier Quotient
CLA	7601	Clear Accumulator
CAM	7621	Clear Accumulator and Multiplier Quotient (CLA MQL)
ACL	7701	Clear Accumulator, Load Multiplier Quotient into Accumulator (CLA MQA)
CLA SWP	7721	Load Multiplier Quotient into Accumulator, Clear Multiplier Quotient

**Table 1-6 Programmed Data Transfer Instructions**

Mnemonic Symbol	Octal Code	Operation
ION	6001	Interrupt Turn On
IOF	6002	Interrupt Turn Off
SKON	6000	Skip if Interrupt On, IOF
SRQ	6003	Skip if Interrupt Request
GTF	6004	Get Flags
RTF	6005	Restore Flag, ION
SGT	6006	Skip if "Greater Than" Flag is Set
CAF	6007	Clear All Flags

**Table 1-7 KM8-E Memory Extension**

Mnemonic Symbol	Octal Code	Operation
GTF	6004	Get Flags
RFT	6005	Restore Flags, ION
CDF	62N1	Change to Data Field N (N=0 to 7)
CIF	62N2	Change to Instruction Field N (N=0 to 7)
CDI	62N3	Change Data Field, Change Instruction Field (CDF CIF)
RDF	6214	Read Data Field
RIF	6224	Read Instruction Field
RIB	6234	Read Interrupt Buffer
RMF	6244	Restore Memory Field

**Table 1-8 KE8-E Extended Arithmetic Element**

Mnemonic Symbol	Octal Code	Operation
MODE CHANGING INSTRUCTIONS		
SWAB	7431	Switch from Mode A to B
SWBA	7447	Switch from Mode B to A
SKB	7471	Skip if Mode B
STANDARD INSTRUCTIONS		
CAM	7621	0→ AC, 0→ MQ
MQA	7501	MQ “OR”ed with AC→ AC
ACL	7701	MQ→ AC (MQA CLA)
MQL	7421	AC→ MQ, 0→ AC
SWP	7521	AC→ MQ, MQ→ AC
MODE A INSTRUCTIONS		
SCA	7441	Step Counter “OR” with AC
SCA CLA	7641	Step Counter to AC
SCL	7403	Step Counter Load from Memory
MUY	7405	Multiply
DVI	7407	Divide
NMI	7411	Normalize
SHL	7413	Shift Left
ASR	7415	Arithmetic Shift Right
LSR	7417	Logical Shift Right
MODE B INSTRUCTIONS		
ACS	7403	AC to Step Count
MUY	7405	Multiply
DVI	7407	Divide
NMI	7411	Normalize
SHL	7413	Shift Left
ASR	7415	Arithmetic Shift Right
LSR	7417	Logical Shift Right
DOUBLE PRECISION INSTRUCTIONS		
DAD	7443	Double Precision Add
DST	7445	Double Precision Store
DPIC	7573	Double Precision Increment
DCM	7575	Double Precision Complement
DPSZ	7451	Double Precision Skip if Zero

**Table 1-9 Teletype Keyboard/Reader**

Mnemonic Symbol	Octal Code	Operation
KCF	6030	Clear Keyboard Flag
KSF	6031	Skip on Keyboard Flag
KCC	6032	Clear Keyboard Flag, and AC, Advance Reader
KRS	6034	Read Keyboard Buffer Static
KIE	6035	Set/Clear Interrupt Enable
KRB	6036	Read Keyboard Buffer, Clear Flag

**Table 1-10 Teletype Teleprinter/Punch**

Mnemonic Symbol	Octal Code	Operation
TFL	6040	Set Teleprinter Flag
TSF	6041	Skip on Teleprinter Flag
TCF	6042	Clear Teleprinter Flag
TPC	6044	Load Teleprinter and Print
TSK	6045	Skip on Printer or Keyboard Flag
TLS	6046	Load Teleprinter Sequence

**Table 1-11 PR8-E Paper Tape Readers**

Mnemonic Symbol	Octal Code	Operation
RPE	6010	Set Reader/Punch Interrupt Enable
RSF	6011	Skip on Reader Flag
RRB	6012	Read Reader Buffer
RFC	6014	Reader Fetch Character
RCC	6016	Read Buffer and Fetch New Character (RRB, RFC)
PCE	6020	Clear Reader/Punch Interrupt Enable

**Table 1-12 PP8-E Paper Tape Punch**

Mnemonic Symbol	Octal Code	Operation
RPE	6010	Set Reader/Punch Interrupt Enable
PCE	6020	Clear Reader/Punch Interrupt Enable
PSF	6021	Skip on Punch Flag
RCF	6022	Clear Punch Flag
PPC	6024	Load Punch Buffer and Punch Character
PLS	6026	Load Punch Buffer Sequence

**Table 1-13 PC8-E Reader/Punch**

Mnemonic Symbol	Octal Code	Operation
RPE	6010	Set Reader/Punch Interrupt Enable
RSF	6011	Skip on Reader Flag
RRB	6012	Read Reader Buffer
RFC	6014	Reader Fetch Character
RFC, RRB	6016	Read Buffer and Fetch New Character
PCE	6020	Clear Reader/Punch Interrupt Enable
PSF	6021	Skip on Punch Flag
PCF	6022	Clear Punch Flag
PPC	6024	Load Punch Buffer and Punch Character
PLS	6026	Load Punch Buffer Sequence

**Table 1-14 TC08-P DECtape Control**

Mnemonic Symbol	Octal Code	Operation	Time ( $\mu$ s)
DTRA	6761	Read Status Register A	2.6
DTCA	6762	Clear Status Register A	2.6
DTXA	6764	Load Status Register A	2.6
DTLA	6766	Clear and Load Status Register A	3.6
DTSF	6771	Skip on Flag	2.6
DTRB	6772	Read Status Register B	2.6
DTXB	6774	Load Status Register B	2.6

Address Locations:

7754 = Word Count

7755 = Current Address

**Table 1-15 TC58 DECmagtape System**

Mnemonic Symbol	Octal Code	Operation
MTSF	6701	Skip on Error Flag or Magnetic Tape Flag
MTCR	6711	Skip on Tape Control Ready
MTTR	6721	Skip on Tape Transport Ready
MTAF	6712	Clear Registers, Error Flag and Magnetic Tape Flag
MTRC	6724	Inclusive OR Contents of Command Register
MTCM	6714	Inclusive OR Contents of AC
MTLC	6716	Load Command Register
none	6704	Inclusive OR Contents of Status Register
MTRS	6706	Read Status Register
MTGO	6722	Mag Tape "GO"
none	6702	Clear AC

**Table 1-16 RK08-P Control and RK01 Disk Drive and Control**

Mnemonic Symbol	Octal Code	Operation	Time ( $\mu$ s)
DLDA	6731	Load Disk Address (Maintenance Only)	2.6
DLDC	6732	Load Command Register	2.6
DLLR	6733	Load Disk Address and Read	2.6
DRDA	6734	Read Disk Address	2.6
DLDW	6735	Load Disk Address and Write	2.6
DRDC	6736	Read Disk Command Register	3.6
DCHP	6737	Load Disk Address and Check Parity	4.6
DRDS	6741	Read Disk Status Register	2.6
DCLS	6742	Clear Status Register	2.6
DMNT	6743	Load Maintenance Register	3.6
DSKD	6745	Skip on Disk Done	3.6
DSKE	6747	Skip on Disk Error	4.6
DCLA	6751	Clear All	2.6
DRWC	6752	Read Word Count Register	3.6
DLWC	6753	Load Word Count Register	3.6
DLCA	6755	Load Current Address Register	3.6
DRCA	6757	Read Current Address Register	4.6

**Table 1-17 DF32-D Disk File and Control**

Mnemonic Symbol	Octal Code	Operation	Time ( $\mu$ s)
DCMA	6601	Clear Disk Address Register	2.6
DMAR	6603	Load Disk Address Register and Read	3.6
DMAW	6605	Load Disk Address Register and Write	3.6
DCEA	6611	Clear Disk Extended Address	2.6
DSAC	6612	Skip on Address Confirmed Flag	2.6
DEAL	6615	Load Disk Extended Address	3.6
DEAC	6616	Read Disk Extended Address	3.6
DFSE	6621	Skip on Zero Error Flag	2.6
DFSC	6622	Skip on Data Completion Flag	2.6
DMAC	6626	Read Disk Memory Address Register	3.6

Address Locations:

7750 = Word Count  
7751 = Memory Address

**Table 1-18 RF08 Disk File**

Mnemonic Symbol	Octal Code	Operation
DCIM	6611	Clear Disk Interrupt Enable and Core Memory Address Extension Register
DIML	6615	Load Interrupt Enable and Memory Address Extension Register
DIMA	6616	Load Interrupt and Extended Memory Address
DFSE	6621	Skip on Disc Error
DISK	6623	Skip Error or Completion Flag
DCXA	6641	Clear High Order Address Register
DXAL	6643	Clear and Load High Order Address Register
DXAC	6645	Clear AC & Load DAR into AC
DMMT	6646	Initiate Maintenance Register

**Table 1-19 TM8-E/F Control**

Mnemonic Symbol	Octal Code	Operation
LWCR	6701	Load Word Count Register
CWCR	6702	Clear Word Count Register
LCAR	6703	Load Current Address Register
CCAR	6704	Clear Current Address Register
LCMR	6705	Load Command Register
LFGR	6706	Load Function Register
LDBR	6707	Load Data Buffer Register
RWCR	6711	Read Word Count Register
CLT	6712	Clear Transport
RCAR	6713	Read Current Address Register
RMSR	6714	Read Main Status Register
RCMR	6715	Read Command Register
RFSR	6716	Read Function Register & Status
RDBR	6717	Read Data Buffer
SKEF	6721	Skip if Error Flag
SKCB	6722	Skip if Not Busing
SKJD	6723	Skip if Job Done
SKTR	6724	Skip if Tape Ready
CLF	6725	Clear Controller and Master

**Table 1-20 LE-8 Line Printer**

Mnemonic Symbol	Octal Code	Operation
PSKF	6661	Skip on Character Flag
PCLF	6662	Clear the Character Flag
PSKE	6663	Skip on Error
PSTB	6664	Load Printer Buffer, Print on Full Buffer or Control Character
PSIE	6665	Set Program Interrupt Flag
PCLF, PSTB	6666	Clear Line Printer Flag, Load Character, and Print
PCIE	6667	Clear Program Interrupt Flag

**Table 1-21 CR8-E Card Reader and Control or CM8-E Optical Mark Card Reader and Control**

Mnemonic Symbol	Octal Code	Operation
RCSF	6631	Skip on Data Ready
RCRA	6632	Read Alphanumeric
RCRB	6634	Read Binary
RCNO	6635	Read Conditions Out to Card Reader
RCRC	6636	Read Compressed
RCNI	6637	Read Condition In From Card Reader
RCSD	6671	Skip on Card Done Flag
RCSE	6672	Select Card Reader and Skip if Ready
RCRD	6674	Clear Card Done Flag
RCSI	6675	Skip If Interrupt Being Generated
RCTF	6677	Clear Transition Flags

**Table 1-22 XY8-E Incremental Plotter Control**

Mnemonic Symbol	Octal Code	Operation
PLCE	6500	Clear Interrupt Enable
PLSF	6501	Skip on Plotter Flag
PLCF	6502	Clear Plotter Flag
PLPU	6503	Pen Up
PLL R	6504	Load Direction Register, Set Flag
PLPD	6505	Pen Down
PLCF, PLL R	6506	Clear Flag, Load Direction Register, Set Flag
PLSE	6507	Set Interrupt Enable

**Table 1-23 VC8-E CRT Display Control**

Mnemonic Symbol	Octal Code	Operation
DILC	6050	Clears Enables, Flags and Delays
DICD	6051	Clears Done Flag
DISD	6052	Skip on Done Flag
DILX	6053	Load X Register
DILY	6054	Load Y Register
DIXY	6055	Clear Done Flag; Intensify; Set Done Flag
DILE	6056	Transfers AC to Enable Register
DIRE	6057	Transfers Display Enable/Status Register to AC

**Table 1-24 VW01 Writing Tablet**

Mnemonic Symbol	Octal Code	Operation
WTSC	6054	Set Tablet Controls
WTRX	6052	Read X
WTRS	6072	Read Status
WTSE	6074	Select Tablet
WTMN	6064	Clear Set XY

**Table 1-25 DC02-F 8-Channel Multiple Teletype Control**

Mnemonic Symbol	Octal Code	Operation
MTPF	6113	Read Transmitter Flag
MINT	6115	Set Interrupt Flip-Flop
MTON	6117	Select Specified Station
MTKF	6123	Read Receiver Flag Status
MINS	6125	Skip on Interrupt Request
MTRS	6127	Read Station Status
MKSF	6111	Skip on Key Board Flag
MKCC	6112	Clear Receive Flag
MKRS	6114	Receive Operation
NONE	6116	Combined MKRS & MICCC
MTSF	6121	Skip on Transmitter Flag
MTCF	6122	Clear Transmitter Flag
MTPC	6124	Transmit Operation
NONE	6126	Combined MTCF & MTPC

**Table 1-26 BB08-P General Purpose Interface Unit**

Mnemonic Symbol	Octal Code	Operation	Time ( $\mu$ s)
GTSF	6361	Skip on Transmit Flag	2.6
GCTF	6362	Clear Transmit Flag	2.6
	6564	(User-Assigned)	2.6
GRSF	6371	Skip on Receive Flag	2.6
GCRF	6372	Clear Receive Flag	2.6
GRDB	6374	Read Device Buffer	2.6

**Table 1-27 Universal Digit Controller (UDC)**

Mnemonic Symbol	Octal Code	Operation	Time ( $\mu$ s)
UDSS	6351	Skip on Scan Not Busy	2.6
UDSC	6353	Start Interrupt Scan	3.6
UDRA	6356	Read Address and Generic Type	3.6
UDLS	6357	Load Previous Status	4.6
UDSF	6361	Skip on UDC Flag and Clear Flag	2.6
UDLA	6363	Load Address	3.6
UDEI	6364	Enable UDC Interrupt Flag	2.6
UDDI	6365	Disable UDC Interrupt Flag	3.6
UDRD	6366	Clear AC and Read Data	3.6
UDLD	6367	Load Data and Clear AC	4.6

**Table 1-28 DR8-EA 12-Channel Buffered Digital I/O**

Mnemonic Symbol	Octal Code	Operation
DBDI	65x0	Disable Interrupt
DBEI	65x1	Enable Interrupt
DBSK	65x2	Skip on Done Flag
DBCI	65x3	Clear Selective Input Register
DBRI	65x4	Transfer Input to AC
DBCO	65x5	Clear Selective Output Register
DBSO	65x6	Set Selective Output Register
DBRO	65x7	Transfer Output to AC

**Table 1-29 MP8E-Memory Parity**

Mnemonic Symbol	Octal Code	Operation
DPI	6100	Disable Memory Parity Error Interrupt
SMP	6101	Skip on No Memory Parity Error
EPI	6103	Enable Memory Parity Error Interrupt
CMP	6104	Clear Memory Parity Error Flag
SMP, CMP	6105	Skip on No Memory Parity Error, Clear Memory Parity Error Flag
CEP	6106	Check for Even Parity
SPO	6107	Skip on Memory Parity Option

**Table 1-30 Synchronous Modem Interface**

Mnemonic Symbol	Octal Code	Operation
SGTT	6405	Transmit Go
SGRR	6404	Receive Go
SSCD	6400	Skip if Character Detected
SCSD	6406	Clear Sync Detect
SSRO	6402	Skip if Receive Word Count Overflow
SCSI	6401	Clear Synchronous Interface
SRTA	6407	Read Transfer Address Register
SLCC	6412	Load Control
SSRG	6410	Skip if Ring Flag
SSCA	6411	Skip if Carrier/AGC Flag
SRS2	6414	Read Status 2
SRS1	6415	Read Status 1
SLFL	6413	Load Field
SSBE	6416	Skip on Bus Error
SRCD	6417	Read Character Detected (if AC0=0) Maintenance Instruction (if AC0=1)
SSTO	6403	Skip if Transmit Word Count Overflows

Break Address Locations:

For additional interfaces:

	Device Codes	Break Locations
7720		
7721 Test Characters	42, 43	7700-7710
7722	44, 45	7660-7670
7723	46, 47	7640-7650
7724 Receive Word Count		
7725 Receive Current Address		
7726 Not Used		
7727 Transmit Word Count		
7730 Transmit Current Address		

**Table 1-31 Multicycle Data Break Locations**

Assigned Locations	Date Break Device	Channel
7640-7650	DP8-EA/ EB	4
7660-7670	DP8-EA/ EB	3
7700-7710	DP8-EA/ EB	2
7720-7730	DP8-EA/ EB	1
7750,7751	DF32-D	
7752,7753	(Reserved for Industry Standard Magnetic Tape)	
7754,7755	TC08-P	

**Table 1-32 KM8-E Time-Share**

Mnemonic Symbol	Octal Code	Operation
CINT	6204	Clear User Interrupt
SINT	6254	Skip on User Interrupt
CUF	6264	Clear User Flag
SUF	6274	Set User Flag

**Table 1-33 DK8-EP Programmable Real Time Clock**

Mnemonic Symbol	Octal Code	Operation
CLZE	6130	Clear Clock Enable Register per AC
CLSK	6131	Skip on Clock Interrupt
CLOE	6132	Set Clock Enable Register per AC
CLAB	6133	AC to Clock Buffer
CLEN	6134	Load Clock Enable Register
CLSA	6135	Clock Status to AC
CLBA	6136	Clock Buffer to AC
CLCA	6137	Clock Counter to AC

**Table 1-34 DK8-EA Line Frequency Clock**

Mnemonic Symbol	Octal Code	Operation
CLEI	6131	Enable Interrupt
CLDI	6132	Disable Interrupt
CLSK	6133	Skip on Clock Flag and Clear Flag

**Table 1-35 DK8-EC Crystal Clock**

Mnemonic Symbol	Octal Code	Operation
CLEI	6131	Enable Interrupt
CLDI	6132	Disable Interrupt
CLSK	6133	Skip on Clock Flag and Clear Flag

**Table 1-36 KP8-E Power Fail Detect**

Mnemonic Symbol	Octal Code	Operation
SPL	6102	Skip on Power Low

**Table 1-37 DP8-EP Redundancy Check Option**

Mnemonic Symbol	Octal Code	Operation
RCTV	6110	Test VRC and Skip
RCRL	6111	Read BCC Low
RCRH	6112	Read BCC High
RCCV	6113	Compute VRC
RCGB	6114	Generate BCC
RCLC	6115	Load Control
RCCB	6116	Clear BCC Accumulation

**Table 1-38 DR8-E Interprocessor Buffer**

Mnemonic Symbol	Octal Code	Operation
DBRF	65x1	Skip if the receive set to a 1
DBRD	65x2	Read incoming data into the AC, clear receive flag
DBTF	65x3	Skip if the transmit flag is set to a 1
DBTD	65x4	Load the AC into the transmit buffer, transmit and set the transmit flag
DBEI	65x5	Enable the Interrupt Request line
DBDI	65x6	Disable the Interrupt Request Line
DBCD	65x7	Clear done flag



# index

## A

AA05-A D/A Converter, 7-51  
AA50-A D/A Conversion Subsystem, 7-50  
AC indicator register, 3-3  
AC power, 8-18  
Access time, 4-14  
Accumulator (AC), 2-3  
A/D interface, 8-18  
Add Data to Memory, 2-11, 4-12  
Adder, 2-5  
ADDR LOAD switch, 3-2  
Address code, 3-8  
Address mode bit, 3-8  
Address, 3-5  
Address, ROM, 9-22  
ADM, 2-11, 4-12  
Advanced Signal Averager, 2-34  
AF04A Digital Voltmeter, 7-45  
AFC8 Analog Input Subsystem, 7-42  
ALGOL, 2-17  
Alphanumeric display terminals, 6-6, 6-20  
Altitude, 8-11  
AM8-EA 8-Channel Analog Multiplexer, 6-100  
AND instruction, 3-9  
Applications software, 2-32  
Arithmetic right shift, 5-4  
ASR33 Teletype, 6-9  
Assemblers, 2-21  
Asynchronous Data control, 2-7, 6-55  
Augmented instruction, 3-7, 3-10  
Auto-correlation applications packages, 2-35  
Autoindex registers, 2-8  
Automatic calling unit, 6-34

## B

BA8-AA System Expander Box, 5-1  
BA8-AB System Expander Box, 5-1  
BAC lines, 2-11  
Basic Signal Averager, 2-34  
BASIC, 2-16, 2-26  
BASIC/RT, 2-36  
PATCH, 2-27  
BB08-P Interface, 7-1  
BE8-A OMNIBUS Expander, 5-1

BIOP, 10-5  
BITMAP, 2-31  
Block data transfer, 4-14  
BMB instruction, 10-4  
Break cycle, 3-5  
BREAK DATA CONT, 3-4  
BREAK IN PROG, 3-5  
BSW instruction, 3-15  
BUILD, 2-19  
Building block approach, 10-34  
Bus connector, 10-39  
Bus driver, 10-34  
Bus indicator register, 3-2  
Bus loads module, 2-6  
Bus receiver, 10-36  
Bus specifications, 9-4  
BUS STROBE, 9-13  
Bus transceiver, 10-37  
Busy/done flip flop, 4-4

## C

CA lines, 9-10  
CA increment, 10-7  
CA register, 4-11, 4-13, 4-14  
Cabinet options, 2-3, 8-5  
Cabinets, 5-1  
Cables, interface, 9-54  
Cabling, 8-8  
Cabling, external bus, 10-31  
Cabling rules, 9-54  
CAF instruction, 4-8  
CAM instruction, 3-17  
CAPS-8 Cassette Programming System, 2-18  
Card readers, 6-61  
Cassette programming system, 2-18  
Cassette tape system, 6-75  
Central Processor Memory Address register (CPMA), 2-5  
Central processor, 2-1, 2-3  
Checklist, OMNIBUS, 9-64  
CIA instruction, 3-13  
CLA instruction, 3-12, 3-15, 3-16  
CLEAR switch, 3-2  
CLL instruction, 3-12  
CM8-F Card Reader, 6-65  
CMA instruction, 3-12  
CML instruction, 3-12  
COBOL, 2-27  
Command decoder, 2-14, 2-15

- Commercial Operating System, 2-13  
 Communications equipment options, 6-34  
 Computation rate, 2-1  
 Computer aided instruction, 2-18, 6-21  
 Computer simulation, 2-18  
 Connector block, 10-26, 10-38  
 Console, programmer's, 3-1  
 Console printer, 6-1  
 Console status word, 9-26  
 Console teleprinters, 6-1  
 Console terminal, 2-11  
 CONT switch, 3-2  
 Control signals, 9-19  
 Cooling, 8-3  
 Correlation, 2-35  
 COS-500, 2-13, 2-19  
 CPMA register, 2-5  
 CPU, 2-1  
 CR8-F Card Reader, 6-61  
 CREF, 2-19, 2-29  
 Cross-Correlation Applications Packages, 2-35  
 CRT Displays, 6-13  
 Current Address (CA) register, 4-11, 4-13, 4-14  
 Current page, 3-8  
 Cycle time, 2-1
- D**
- D/A interface, 8-18  
 DAFT, 2-20  
 DAQUAN, 2-35  
 Data acquisition peripherals, 7-39  
 Data break,  
     Address, 9-42  
     Data paths, 9-42  
     Direction, 9-42  
     Interface, 7-4, 9-41  
     Loading, 9-42  
     OMNIBUS signals, 9-16  
     Priority, 4-12, 4-13, 9-42  
     Request, 4-17  
     Subroutines, 4-11  
     System, 2-11  
     Timing, 9-47, 9-58  
     Transfer, external, 10-13  
     Transfers, internal, 4-11, 4-12  
 Data bus, 4-18, 9-16, 9-22  
 Data communications equipment options, 6-33  
 Data field register, 3-4, 4-11  
 Data gates, 2-5  
 DATA lines, 2-11  
 Data signals, 9-19  
 Data transfer, 2-9, 4-1, 9-29  
 DB8-E Interprocessor Buffer, 6-107
- DC08H Automatic Calling Unit Controller, 6-33  
 DCA instruction, 3-9  
 DDT, 2-30  
 DEC/X8 System Exerciser, 2-18  
 DECtape, 4-18, 6-80, 7-18  
 DECtape software, 7-36  
 DECUS, 2-13  
 Delivery, 8-1  
 DEP switch, 3-2  
 Device code assignment, 9-33  
 Device codes, 9-33  
 Device flag, 4-4, 4-5  
 Device independence, 2-14  
 Device names, 2-14  
 Device selection, 9-34, 9-63  
 Device selection code, 3-20  
 Device selector, 3-20  
 Device status, 4-2  
 DF32-D Disk File, 7-5  
 DIBOL, 2-27  
 Differential potentials, 8-18  
 Digital Equipment Computer User's Society, 2-13  
 Digital Voltmeter, 7-45  
 Dimensional information, 8-4  
 Direct memory access, 3-19, 4-11  
 Disk cartridge system, 6-67  
 Disk file and expander, 7-5, 7-9  
 Displays, CRT, 6-13  
 Division, 5-3  
 DMA state, 3-19  
 Documentation, 2-13  
 Double precision, 5-10  
 DP8-E Synchronous Modem Interface, 6-36  
 DR8-EA 12-channel I/O, 6-101  
 Driver, bus, 10-34  
 Driving a line, 9-63  
 DS32-D Disk File Expander, 7-5  
 Dump and Fix Technique, 2-20  
 DW08-A Bus Converter, 2-11, 7-80  
 Dynamic Debugging Technique, 2-30
- E**
- EAE, 5-2  
 Edge connector, 2-3  
 EDIT, 2-30  
 Editor, 2-30  
 Electrical noise, 8-18  
 Electrical power, 8-1  
 EMA 0-2, 9-21  
 EMA indicator, 3-2  
 Emissions, 8-11  
 Enclosure, Industrial Computer, 8-11  
 Environment, 8-1, 8-9  
 Error diagnostics, 2-26  
 EXAM switch, 3-2

**EXECUTE** major state, 3-4, 3-17, 3-21  
Execution time, 4-3  
Expandability, 2-12  
Expansion boxes, 5-1  
Expansion options, 5-1  
Expansion techniques, 10-1  
**EXTD ADDR LOAD** switch, 3-2  
Extended Arithmetic Element, 2-11, 5-1  
Extended I/O, 9-39  
External bus,  
    Applications, 10-8  
    Cabling, 10-31  
    Cooling, 10-30  
    Design features, 10-2  
    Interrupt transfers, 10-13  
    Peripheral selection, 10-8  
    Relation to OMNIBUS, 9-6  
    Restrictions, 10-29  
    Signals, 10-3  
    Timing, 10-30  
External I/O Bus Interface, 2-11  
External I/O device, 4-2, 7-1  
Extreme operating conditions, 8-11

#### F

**F SET L**, 4-9, 9-24  
Fabrication testing, 9-63  
Fast cycle, 3-19  
Features, standard, 2-1  
**FETCH** major state, 3-4, 3-17, 3-19  
Field, 3-6  
Filter, 8-11  
Fire precautions, 8-1  
Flags, 9-36  
Floating-point packages, 2-31  
Floating-point processor, 7-65  
Floor space, 8-6  
**FOCAL**, 2-17, 2-26  
**FORTRAN IV**, 2-25  
**FORTRAN-D**, 2-16  
**FPP-12 Floating-Point Processor**, 7-65  
Front panel options, 5-1

#### G

**G104 Sense/Inhibit module**, 2-8  
**G227 XY Driver and Current Source Module**, 2-8  
**G619 Planar Stack Board**, 2-8  
General purpose interface unit, 7-1  
**GLK** instruction, 3-13  
Greater Than Flag (GTF), 3-4, 4-7, 5-2  
Ground requirements, 8-18  
Grounding, 8-3, 9-63  
Group 1 Operate Microinstructions, 3-11

Group 2 Operate Microinstructions, 3-13  
Group 3 Operate Microinstructions, 3-16, 5-3  
**GTF** instruction, 4-8, 4-10  
**GTF**, 3-4, 4-7, 5-2

#### H

**H019 Mounting Bar**, 10-40  
**H803 Connector Block**, 10-38  
**H851 Edge Connector**, 2-3  
**H893 Connector Block**, 10-38  
**H911 Mounting Panel**, 10-26  
**H9190 Kit**, 10-38  
**H9190 Mounting Panel**, 10-39  
**H960 Standard Basic Cabinet**, 8-5  
**H961-A Short Basic Cabinet**, 8-6  
**H967-AA Short Option Cabinet**, 8-6  
**H967-BA Short Basic Cabinet**, 8-5  
**H992 ICE Box**, 8-13  
**HALT** switch, 3-3  
Hardware, OMNIBUS, 9-5  
High Altitude, 8-11  
Histogram Applications Packages, 2-35  
**HLT** instruction, 3-14  
Housekeeping instruction, 3-7, 3-10, 9-39  
Humidity, 8-9

#### I

Cabling, 8-8  
Capabilities, 2-1  
Control logic, 9-39  
Conversion, 7-80  
Device flags, 4-4, 4-5  
Device handlers, 2-14  
Device independence, 2-14  
Techniques, 2-9  
Timing constraints, 4-3  
Timing, general, 9-39  
**I/O PAUSE**, 3-5, 9-9  
**IAC** instruction, 3-12  
**ICE Box**, 8-11  
**IND 1 and 2**, 9-26  
Indicator lights, 3-1  
Inductive spikes, 8-18  
Industrial Computer Enclosure, 8-11  
**INHIBIT** signal, 9-23  
**INITIALIZE H**, 4-9  
Initialize pulse, 4-8, 9-28  
Input buffer logic, 9-36  
Input/output, see I/O  
Input/output pulse, 4-2  
Input/output transfer instruction, 3-10, 4-1  
Installation planning, 8-1  
Instruction field register, 3-4, 4-9  
Instruction register (IR), 2-6

Instruction timing, 3-17  
Instructions, 3-6  
INT IN PROG H, 4-9, 9-26  
INT INHIBIT, 4-9  
INT STROBE H, 4-9, 9-16  
Interface, M-series example, 10-42  
Interfacing, 2-9  
Interfacing logic modules, 10-20  
Interfacing peripherals, 10-20  
Internal I/O, 9-9  
Internal I/O device, 4-2  
Internal I/O options, 6-1  
Interpreter, 2-32  
Interprocessor buffer, 6-107  
Interrupt, 4-5  
    Buffer A and B, 4-9, 4-11  
    Control circuits, 2-7  
    Delay flip-flop, 4-9  
    Enable flip-flop, 3-4, 4-5, 4-9  
    Inhibit flip-flop, 3-4, 4-7, 4-9  
    Request line, 3-4, 4-5, 9-63  
    Request line loading, 9-63  
    Request logic, 9-36  
    Request timing, 9-11  
    Service routine, 4-7, 4-10  
    Sync flip-flop, 4-9  
    System, 4-5, 4-7  
    System hardware components, 4-9  
    System operation, 4-5  
    Timing, 9-56  
    Transfers, external bus, 10-13  
IOF instruction, 4-8  
ION instruction, 4-7  
IOP, 4-2  
IOP dialogue, 2-11  
IOT execution time, 4-3  
IOT instruction, 3-10, 4-1  
IR, 2-6, 4-9  
IR, 0-2, 9-23  
ISZ instruction, 3-9

#### J

JMP instruction, 3-10  
JMS instruction, 3-9

#### K

KA8-E Positive I/O Interface, 4-2  
KC8-EA Programmer's Console, 2-7, 3-1  
KC8-EB Front Panel, 5-1  
KC8-EC Turnkey Front Panel, 5-1  
KC8-MC Programmer's Console, 2-12  
KC8-ML Programmer's Console, 2-12  
KD8-E Data Break Interface, 7-4  
KE8-E Extended Arithmetic Element, 5-2  
Key control, 9-25

Keyboard monitor, 2-14  
KG8-E Redundancy Check Option, 6-51  
KK8-E Central Processor, 2-3  
KL8-E Asynchronous Data Control, 2-7, 6-55

#### L

LA30 DECwriter, 6-1  
LAB-8/E, 2-32  
Laboratory mounting panel, 6-105  
Laboratory peripherals, 6-96  
LC8-E DECwriter Control, 6-1  
LD ADD EN signal, 9-24  
LE8-E Line Printer, 6-29  
Left shift, 5-3  
Line driving, 9-63  
Line printer options, 6-29  
Line voltage, 8-13  
Link (L), 2-5, 3-4, 9-24  
Loads, OMNIBUS, 9-8  
Logic flags, 9-36  
Logic modules, 10-20  
Logical device names, 2-14  
Logical right shift, 5-4  
Logical sequence numbers, 3-11

#### M

M series interface, 10-42  
M series module placement, 10-40  
M series module summary, 10-43  
M3800 Major Registers Module, 2-3  
M783 Bus Driver, 10-34  
M784 Bus Receiver, 10-36  
M785 Bus Transceiver, 10-37  
M8320 Bus Loads Module, 2-6  
M8330 Timing Generator Module, 2-6  
M849 RFI Shield, 2-7  
M8650 Asynchronous Data Control Module, 2-7  
M935 Bus Connector, 10-39  
M935 Kit, 10-38  
MA, MS LOAD CONT signal, 9-18  
MA, 0-11, 9-21  
MACRO-8, 2-23  
Magnetic tape options, 6-75, 7-18  
Major register control circuits, 2-6  
Major registers module, 2-3  
Major state, 3-17, 9-24  
Major state generator, 2-6  
Manual programming, 3-1  
Manuals, 2-13  
Mask, 3-9  
MB increment, 10-7  
MB register, 2-5  
MD DIR signal, 3-5, 9-18, 9-20  
MD indicator register, 3-2  
Mechanical expansion options, 5-1

- Mechanical vibration, 8-11  
 MEM START signal, 9-22  
 Memory Address indicator, 3-2  
 Memory Buffer register (MB), 2-5  
 Memory cycle, 3-19  
 Memory field, 3-5  
 Memory instructions, 3-6  
 Memory locations, 3-5  
 Memory organization, 3-5  
 Memory page, 3-5  
 Memory reference instruction (MRI), 3-7  
 Memory system, 2-8  
 Memory timing, 3-19  
 Microinstructions, 3-11  
 Microprogramming, 3-11  
 Mixed mode arithmetic, 2-26  
 MM8-E Core Memory System, 2-8  
 Mnemonic, 3-6  
 Mode A instructions (EAE), 5-5  
 Mode B instructions, 5-7  
 Mode changing instructions, 5-4  
 Module layout, 9-52  
 Module mounting hardware, 10-39  
 Module Placement, 10-40  
 Mounting bar, 10-40  
 Mounting hardware, 10-39  
 Mounting panel, 10-26, 10-39  
 MQ indicator register, 3-2  
 MQ register, 3-16  
 MQA instruction, 3-16  
 MQL instruction, 3-16  
 MRI, 3-7  
 MS, IR DIS L, 4-9  
 Multiple interrupts, 4-10  
 Multiplication, 5-3  
 Multiplier Quotient Register (MQ), 2-5, 3-16
- N
- Nested interrupts, 4-10  
 New instructions, 2-10  
 NMR Signal Averager, 2-34  
 Noise, 8-18  
 NOP instruction, 3-12  
 Normalization, 5-4  
 NOT LST TRANSFER signal, 9-14  
 Numerical control, 2-16
- O
- Octal Debugging Technique, 2-29  
 ODT, 2-29  
 OFF/POWER/PANEL LOCK switch, 3-2  
 OMNIBUS, 2-3  
   DECtape, 7-80  
   Description, 9-3  
   Design checklist, 9-64  
   Hardware, 9-5
- Interface, 9-30, 10-34  
 Loads, 9-8  
 Magnetic tape options, 6-75  
 Programming, 9-64  
 Rules, external bus, 10-28  
 Signals, 9-6, 9-16, 9-19  
 Techniques, positive I/O, 10-1  
 OP-code, 3-8  
 Operate instructions, 3-11  
 Operating conditions, 8-9  
 Operating range, 8-9  
 Operating systems, 2-13  
 Operation code, 3-8  
 Operation specification code, 3-11, 4-2  
 Operations decoder, 9-34  
 Optical mark card reader, 6-66  
 OS/8, 2-14  
 Oscilloscope display, 6-14  
 OSR instruction, 3-14  
 OVERFLOW signal, 9-17, 9-27  
 Overlay, 2-14, 2-26
- P
- Page, 3-5  
 Page address bit, 3-8  
 Page addressing, 3-8  
 Page bit, 3-8  
 PAL III, 2-22  
 PAL Language, 2-22  
 PAL-D, 2-16  
 PAL8, 2-23  
 PALC, 2-18  
 Paper tape reader, 6-10  
 Paper tape reader/punch, 6-13  
 PAUSE I/O signal, 9-9  
 PAUSE line, 3-5  
 PC8-E Paper Tape Reader/Punch, 6-11  
 Peripheral device flag, 4-4  
 Peripheral device status, 4-2  
 Peripheral Interchange Program, 2-20  
 Peripheral interfacing, 10-20  
 PIP, 2-20, 2-29  
 Planar stack board, 2-8  
 Planning, 8-1  
 Plotter options, 6-25  
 Point Plot Display System, 6-13  
 Positive I/O interface, 4-2  
 Power clear pulse generator, 2-7  
 Power failure, 8-18  
 Power receptacles, 8-14  
 Power requirements, 8-14  
 PR8-E Paper Tape Reader, 6-10  
 Pre-delivery planning, 8-1  
 Preparation for delivery, 8-1  
 Printers, 6-1  
 Processor IOT instructions, 4-7

Processor options, 8-3  
Processor timing, 9-22  
Program Assembly Language, 2-22  
Program Counter (PC), 2-5  
Program interrupt transfer, 4-1, 4-5  
Programmed data transfer, 4-1  
Programmed I/O signals, 9-9, 9-13  
Programmed I/O timing, 9-39  
Programmer's console, 2-7, 2-11, 3-1  
Propagation delay, 9-58  
Psychrometric charts, 8-9  
PULSE LA, 9-25

## R

Radiated Emissions, 8-11  
RAL instruction, 3-12  
RAR instruction, 3-12  
Real time processing, 2-26  
Receiver, bus, 10-36  
Redundancy check option, 6-51  
Resource sharing, 2-15  
RETURN signal, 9-23  
RF08 Disk File, 7-9  
RFI shield module, 2-7  
RK05 DECpack, 4-14  
RK8-E DECdisk, 6-67  
ROM address signals, 9-22  
RS08 Disk File expander, 7-9  
RT01 Data Entry Terminal, 7-77  
RT02 Data Entry Terminal, 7-78  
RTF instruction, 4-8  
RTL instruction, 3-12  
RTR instruction, 3-12  
RUN indicator, 3-2  
RUN signal, 9-27

## S

SABR, 2-24  
Safety, 8-1  
Save field register, 4-7, 4-10  
Sense/inhibit module, 2-8  
SGT instruction, 4-8  
Shift, 5-3  
Short basic cabinet, 8-6  
Short cabinet, 8-5  
Short option cabinet, 8-5, 8-6  
Signal Averager, 2-34  
Signals, control and data, 9-19  
SING STEP switch, 3-2  
Single-cycle data break, 4-13  
Site preparation worksheet, 8-1  
Site survey, 8-1  
SKIP line, 9-63  
Skip microinstructions, 3-15  
SKIP signal, 9-11  
SKON instruction, 4-7  
SKP instruction, 3-14  
Slow cycle, 3-19

SMA instruction, 3-14  
SNA instruction, 3-14  
SNL instruction, 3-14  
Software, 2-13  
Software Distribution Center, 2-13  
SORT, 2-20  
Source Compare Program, 2-31  
SPA instruction, 3-15  
Space requirements, 8-2  
Special cooling, 8-3  
SRCCOM, 2-13  
SRQ instruction, 4-8  
STA instruction, 3-13  
Standard cabinet, 8-5  
Standard features, 2-1  
STATE indicator register, 3-4  
STATUS indicator register, 3-3  
Status word, 9-26  
Step counter, 5-2  
STL instruction, 3-13  
STOP signal, 9-25  
STROBE signal, 9-23  
STROBE, BUS, 9-13  
Subroutine, 3-9  
SW line, 3-5  
SW signal, 9-26  
SW switch, 3-5  
SWP instruction, 3-17  
Symbolic Editor, 2-30  
Synchronous modem interface, 6-34  
System configuration, 9-4  
System expander box, 5-1, 8-3  
System installation, 8-1  
System lay-out, 8-2  
SZA instruction, 3-14  
SZL instruction, 3-14

## T

Table-top system, 8-1  
TA8-E DECcassette, 6-75  
TAD instruction, 3-9  
TC08 DECTape Control, 7-27  
TD8-E DECTape, 6-80  
TECO, 2-30  
Teleprinters, 6-1  
Teletype, 6-9  
Temperature, 8-9  
Temperature, extreme, 8-11  
Terminal, 6-20  
Terminal, VT8-E, 6-6  
Testing, 9-63  
Text Editor and Corrector, 2-30  
Text manipulation, 2-26  
Three-cycle data break, 4-13, 4-18  
Time pulse, 3-19, 9-27  
Time pulse generator, 2-6  
Time state, 3-29, 9-27  
Time state generator, 2-6  
Timeshared-8/E, 2-15

Timesharing software, 2-15  
Timing, 3-17, 3-19  
Timing constraints, 4-3  
Timing generator, 2-6, 3-19  
Timing, interrupt, 9-56  
Timing, processor, 9-22  
Timing, programmed I/O, 9-39  
TK8-E Cassette Tape System, 6-74  
TM8-E Magtape System, 6-85  
Transceiver, bus, 10-37  
Transfer rates, 9-32  
TS1 to TS4, 9-27  
TU10 DECmagtape, 6-82  
TU56 DECTape transport, 7-25

**U**

Universal Digital Controller, 7-53  
Unpacking, 8-1  
User flag, 4-7  
User mode, 3-4  
USER MODE signal, 9-28  
User service routine, 2-14, 2-15

**V**

VC8-E Display System, 6-13  
Vibration, 8-11  
Voltage fluctuations, 8-13  
VR14 Oscilloscope Display, 6-14  
VT05 Alphanumeric Terminal, 6-20  
VT8-E Display Terminal, 6-6  
VW01 Writing Tablet, 7-57

**W**

WC overflow, 4-13, 4-21  
WC register, 4-11, 4-13  
Word count register, 4-11, 4-13  
WRITE signal, 9-23  
Writing tablet, 7-57

**X**

XY driver and current source  
module, 2-8  
XY plotter options, 6-25  
XY8-E Incremental Plotter Con-  
trol, 6-25

## **NOTES**