digital

PDP-11 Systems Handbook

PDP-11 Systems Handbook

Featuring: MicroPDP-11/83
MicroPDP-11/73
MicroPDP-11/53
PDP-11/84

digital

PDP-11 Systems Handbook

Featuring: MicroPDP-11/83
MicroPDP-11/73
MicroPDP-11/53
PDP-11/84

digital

Contents

Chapter 3 ▪ MicroPDP-11/73 System

Chapter 4 ▪ MicroPDP-11/53 and MicroPDP-11 /53 PLUS Systems

Chapter 5 ▪ Q-Bus Technical Description

Chapter 6 ▪ PDP-11/84

Chapter 7 ▪ UNIBUS Technical Description

Chapter 8 ▪ System Software and Layered Products

Chapter 9 ▪ System Options

Chapter 10 ▪ Networks

Chapter 11 ▪ Services and Documentation

Appendices

Preface

The PDP-11 Hardware Handbook is a reference for Digital's family of supermicrocomputer systems and describes the PDP-11/84, the MicroPDP-11/83, the MicroPDP-11/73, and the MicroPDP-11/53 as well as related options and software.

Chapter 1 provides an overview of the PDP-11 family and addresses the significant role that family systems continue to play in providing Digital computer solutions.

Chapter 2 discusses the MicroPDP-11/83 system in detail. Further, this chapter presents information common to all MicroPDP-11 systems.

Chapter 3 presents the MicroPDP-11/73 system and describes its features and benefits. Common technical issues with the MicroPDP-11/ 83 system are addressed in Chapter 2.

Chapter 4 presents the MicroPDP-11/53 system. Common technical issues with the MicroPDP-11/83 system are addressed in Chapter 2.

Chapter 5 is a technical summary of the Q-bus, in which important bus-related MicroPDP-11 system issues are discussed.

Chapter 6 provides a description of the PDP-11/84 system, the PDP-11 family's UNIBUS offering.

Chapter 7 is a technical summary of the UNIBUS, in which important bus-related PDP-11 system issues are discussed.

Chapter 8 describes the broad software resources available for use with the PDP-11 family of computers. Digital provides software solutions for virtually every business and scientific application.

Chapter 9 provides information regarding the full range of options that can enhance the performance and utility of PDP-11 computer systems.

Chapter 10 presents some fundamental concepts of computer networking in order to lend a fuller perspective on the capabilities of PDP-11 systems.

Chapter 11 explains about the services and documentation resources available to help you use your PDP-11 system to its maximum advantage.

Appendix A is a comparison chart of the PDP-11 family of computers.

Appendix B describes the PDP-11 Console Command set.

Appendix C describes PDP-11 instruction timing.

Appendix D provides information about the MicroPDP-11/23 system.

Appendix E provides information about the PDP-11/24 and PDP-11/44 systems.

Your comments and suggestions about this handbook would be greatly appreciated. Please complete and return the Reader Comment Sheet at the back of this handbook.

Chapter 1 ▪ Introduction to the PDP-11 Family

▪ The PDP-11 Tradition Continues

The PDP-11 family of computers represents Digital's ongoing commitment to high-quality, cost-effective computer solutions for today's most demanding business. PDP-11 systems continue to make important contributions in nearly every major industry, in tasks as diverse as data processing, office automation, and process control. The PDP-11 family comprises a broad range of systems from entry-level departmental systems to large-capacity corporate systems networked to today's best and most sophisticated technology.

The PDP-11 family of computer systems provides special benefits to users:

- **Appropriateness**—PDP-11s are optimal machines. They fit hundreds of applications—process control, scientific, communications, education, business, banking, and construction, just to name a few. Over 500,000 PDP-11s have been installed worldwide, running the gamut of computing needs from accounting to zoology.

- **Versatility**—PDP-11s offer one of the widest selections of business solutions in the industry. A broad range of application software includes a complete set of operating systems, data management tools, programming languages, and communications capabilities. All are proven software. Thousands of PDP-11 applications are available now—and the development continues to grow.

- **Cost-effectiveness**—When considering your cost per user, PDP-11s are the best bargain in multiuser, multitasking systems. Your applications are perfectly matched to a PDP-11 and you can be sure that you're not spending money on power you don't need.

- **Compatibility**—A rich, mature set of products and options across the PDP-11 family—that's what commonality within the PDP-11 architecture is all about. It's so easy to build precisely the system you need for your particular solution. Hardware and software compatibility makes this a lasting investment. No need to retrain programmers when you move to newer, more powerful PDP-11s. If you ever decide to move from PDP-11s to Digital's larger VAX systems, it is easy.

- **Investment Value**—Today's PDP-11s are the result of Digital's continuing research and development. No wonder the PDP-11s have become an industry standard. In fact, Digital is investing more on R&D in the 16-bit environment than ever before—twice the amount we were spending two years ago. So go ahead and plan, assured that your PDP-11 investment will be well-protected tomorrow and will continue to grow in the future.

▪ Powerful, High-performance Systems Offer Fast, Reliable Response Time

The development of the high-speed J-11 chipset—one of the finest examples of Digital's leading-edge technology—is the heart of our newest PDP-11 systems. The J-11 chipset provides top-notch performance and high user access in a very small package.

Today's PDP-11s—the entry-level MicroPDP-11/53 to the midrange MicroPDP-11/73 to the most powerful MicroPDP-11/83 and PDP-11/84—are the result of years of technological advancement. And each system is based on highly reliable, state-of-the-art CMOS technology.

The fast, dependable response time of the PDP-11 systems provides the standard for cost-sensitive realtime applications. PDP-11s have a reputation for predictability, flexibility, and quick I/O response. And Digital's library of PDP-11 realtime systems and products is the result of almost three decades as a leading computer supplier to the realtime market.

PDP-11 computers are embedded in and control a full range of realtime solutions worldwide. Digital's systems are an excellent choice for realtime distributed process control. PDP-11 systems are used in realtime applications that range from a single drilling machine to a complex robot arm.

For example, PDP-11s are used to control the temperatures of furnaces used in the manufacture of integrated circuits. One engineering firm uses MicroPDP-11s to control the temperature to within one-tenth of a degree at 1,000 degrees Fahrenheit, allowing engineers to efficiently regulate the baking process within the furnace and thereby increase their profits.

You can build exactly the system you need, whatever your purpose. The J-11 technology provides the base on which to build your total hardware and software solutions.

▪ PDP-11 Buses Built for Reliable Performance

All PDP-11 system products rely on either the Q-bus or UNIBUS—two powerful, yet simple buses. Chapters Five and Seven discuss the Q-bus and the UNIBUS respectively.

The first PDP-11 was designed around a unique communications pathway, the UNIBUS, that tied processor, memory, and other system elements together. This architecture was a radical departure from conventional minicomputer design. So effective was the design that subsequent machines, including today's PDP-11/84, use the UNIBUS.

The Q-bus was introduced later for low-cost PDP-11s. Through the years the Q-bus has been redesigned so that it is now the technical equal of the UNIBUS, while retaining the smaller size and lower cost that originally set it apart.

Both buses support rapid communications between the CPU, memory, and mass storage. Both handle a vast range of memories, processors, coprocessors, and communications and peripheral devices.

The simplicity and reliability of their design explain why the PDP-11 family continues to be an excellent long-term investment. The Q-bus and UNIBUS proven technology is hard to beat.

▪ PDP-11s—Your Best Business Choice

You now have the power to develop applications that address your most demanding needs—plus the personal productivity tools, such as word processing and spreadsheets, that you once associated with stand-alone systems. PDP-11s are real file management systems. They serve the same purpose as smaller systems, yet they offer the expandability of more powerful systems, as well as a wide variety of integrated software solutions.

As multiuser, multitasking systems, the MicroPDP-11 computers provide more cost-effective solutions than stand-alone systems. And they offer you compatible machines that are both modular and upgradable.

By design, PDP-11s complement Digital's larger more powerful VAX systems by offloading large systems and providing a network link with all your computing resources.

For many applications, it's not necessary to add more CPU power and increase capacity by spending additional money for a mainframe. With the huge array of PDP-11 software suited to your business needs, your PDP-11s integrate easily into an existing computing environment. And Digital's state-of-the-art networking architecture allows data to be communicated throughout your organization.

PDP-11 systems are perfect for running a departmental application. They provide your department with the control you want over your system. For example, a MicroPDP-11/83 can run your shop-floor application without backlog from a large batch job. Your system is dedicated to your department

needs, while providing easy access to data in your larger systems. And in dedicated applications, system overhead is low, making the PDP-11s your most cost-effective choice.

• Digital's Networking Strategy Enables You to Integrate All Your Computer Resources

By combining your computer resources with Digital's Network Architecture (DNA), you can integrate applications among departments, consolidate computing power, give employees timely access to data, and provide people with an efficient way to communicate. This, in turn, will help you save money, be more profitable, and respond more quickly to the demands of business.

Our networks connect computers of all sizes—even computers made by other manufacturers—whether in the same department, the same building, or scattered around the world.

You can incorporate DECnet for Digital communications, Ethernet local area networks (LANs), Internet emulators, SNA Gateways, and the Packetnet X.25 protocol. The DNA concept exemplifies Digital's commitment to integrated growth and development and provides the solution for LANs and wide area networks.

Digital's ThinWire Ethernet provides you with a set of low-cost, easy-to-install cables for connecting your PDP-11 family of systems—as well as personal computers, workstations, and network servers—into a local area network.

Only Digital provides you with such completely integrated networking capabilities.

• Software Compatibility to Keep Investment Costs Down

The wealth of existing software for PDP-11s makes these machines ideal for a variety of uses. Literally thousands of applications programs have been written for the PDP-11s. And the software development for PDP-11s continues to grow, while protecting your software investment.

Digital's wide choice of PDP-11 operating systems, designed and optimized for multiple areas, includes realtime, timesharing, commercial, and scientific applications. And the high degree of compatibility among PDP-11 languages, system programs, and information-management services makes it simple to interconnect your varied operations and guarantees that programs can easily move among systems.

Each of the PDP-11 operating systems comes in various forms, allowing you to individually create the system specifically for your needs. See the software chart at the end of this chapter. Digital's sales and support staff, as well as

our distributors and OEMS, will help you develop the best solution for the quickest return on your investment.

The A-to-Z Integrated System offers a single computing environment for your business and office needs—word processing, business graphics, database management, and much more. A-to-Z is layered on and bundled with our popular Micro/RSX operating system. You can easily install the software yourself, with no need for technical personnel to manage it.

Because A-to-Z is an integrated system, data can be shared between applications. For example, an order entry clerk can enter purchase orders into the system, while a secretary is typing letters at another terminal, and a sales manager is preparing a monthly sales report at a third terminal. At the same time, A-to-Z can accept and respond to a request for inventory status from a shipping clerk in the warehouse.

The networking capabilities of A-to-Z are impressive. Using A-to-Z Electronic Mail and DECnet, data is shared between users on a single system or multiple systems.

Should you need to grow into a larger VAX system, you will find A-to-Z available on the MicroVAX family of products. Once again, Digital has taken the extra effort to allow for your growth and investment protection.

The thousands of applications that have been written for the PDP-11s satisfy the diverse computing needs of multiple levels of users. And Digital's coordinated development efforts continue to provide layered software product compatibility for the 16-bit family, subset compatibility for the 32-bit family.

For example, you can port BASIC-PLUS-2/COBOL to VMS, linked with VAX BASIC, and executed without having to change the source code.

Because we work to develop products that can grow with your changing business needs and strive to keep your investment costs down, Digital continues to deliver the latest in 16-bit technology without jeopardizing your software investments.

▪ Maximum Packaging Flexibility Allows You to Configure with the Latest in Peripheral Technology

When the PDP-11s were designed, we wanted you to have the same flexibility in selecting enclosures as you have in specifying performance. And it's cost-effective to match the right performance with the right package.

Choose the systems that meet your needs today. Then add to them. Upgrade or change them. Link them throughout your organization as your requirements change.

More than enough options are available to meet a variety of space and environmental requirements. If needs change, compatibility within the Q-bus family and UNIBUS family lets you reconfigure memory, storage, and communication options without having to buy new peripherals.

Systems can be expanded by adding an extensive array of communications options—Digital and analog input and output options, IEEE interfaces, single- and multiple-line parallel interfaces, intelligent front-end communication processors, and a selection of multiplexers for your terminals and modems.

A carefully planned Digital Storage Architecture (DSA) framework of standardized interfaces permits the addition of the latest in storage products. So you can incorporate new technologies to host systems without having to develop additional specialized controllers or software drivers. Digital is committed to providing you with greater storage capacities in the future and the PDP-11 systems are designed with this in mind.

▪ Digital's Attention to Detail Saves You Time and Money

Time is money. You buy yourself extra time when you buy from Digital because we've taken the time to think through any problems you might otherwise encounter.

Once you decide on a PDP-11, you can put your system to work very quickly. The implementation period is fast because Digital has designed the product with care.

You buy quality and completeness when you buy a PDP-11 system. No time is wasted figuring why a system doesn't work. Digital makes sure it all works. From the cables to the terminals to the operating systems, PDP-11 systems and their options are simple and straightforward.

Expanding your system is even easier, without any downtime involved. Just plug in your new board, terminal, or disk. Conversions are a thing of the past.

▪ Digital's Quality and Service...An Ongoing Commitment

At Digital, quality isn't an afterthought. Ensuring that our products meet the highest standards is the essential guiding principle in every phase of product development and manufacturing.

With years of experience and thousands of systems in use worldwide, Digital's Field Service exemplifies our commitment to customer satisfaction. In fact, Digital's Field Service organization recently received the highest rating among major competitors.

That's because we provide the most inclusive range of service options in the industry:

- Optional onsite support programs including extended coverage plans.
- Remote hardware monitoring.
- Training.
- Customer-runnable diagnostics.
- Self-maintenance with DECmailer board replacement service.

Whatever your goal, Digital can help. We can work with you on short term and long-term planning, so that you buy only what you need. You can establish training programs that will help make your people productive more quickly and connect new systems to existing equipment. And help you choose the most effective maintenance option.

Digital brings a combination of planning, direct assistance, documentation, discussion, and hands-on experience to your start-up process. You avoid the confusion you may have thought inevitable with a new system, and get started much faster. At Digital, quality isn't just something we build into our products—it's an ongoing commitment to excellence and the success of our customers.

▪ PDP-11 Systems Grow with You

As your business expands, so will your computing needs. PDP-11 systems are designed with your future in mind.

MicroPDP-11/53

The MicroPDP-11/53 is an entry-level system with an attractive price and performance combination to meet the toughest of your computing requirements.

The heart of the MicroPDP-11/53 is a 15-megahertz, J-11 single-board computer with 0.5 Mbytes of onboard memory. This computing engine can easily devote itself to your dedicated process-control program or become a shared resource for your department's multiuser workload.

Packaged in a trim, pedestal enclosure that can fit underneath or beside your desk, the MicroPDP-11/53 provides ample space for memory and communications options. And there's room for connecting as many as 26 I/O devices. For cabinet integration, the MicroPDP-11/53 is also available in a rackmount model.

The Q-bus links the MicroPDP-11/53 with a compatible set of mass-storage devices. Two new half-height disk drives are available, allowing more storage devices to be housed directly in the system chassis. Now there's room for three integrated mass-storage devices, as opposed to two full-height devices.

For a price-competitive entry-level system, the MicroPDP-11/53 will answer your needs.

MicroPDP-11/73

The MicroPDP-11/73 computer provides one-third more compute power than the MicroPDP-11/53.

Its 15-megahertz J-11 chip drives the MicroPDP-11/73, supporting a wide range of realtime or multitasking applications for as many as 41 users, depending on the enclosure. Memory is expandable in 1- and 2-Mbyte increments and, with its memory-management unit, the MicroPDP-11/73 can address, protect, and segment up to 4 Mbytes of memory.

The choice of system enclosures includes a pedestal or rackmount model, a floorstand model with casters, or a 42-inch-high expander cabinet.

Storage choices vary with integral and external formatted Winchester disks, streaming cartridge tapes, and a dual-floppy-disk drive. High-capacity disk drives can be configured with an expander cabinet, allowing you greater than 1 Gbyte of storage space.

This midrange system is the perfect choice for many applications that require larger storage options than the entry-level system.

MicroPDP-11/83

The computing power of the MicroPDP-11/83 has twice the performance of the MicroPDP-11/73. For the most power and the greatest multiuser, multitasking capability in a 16-bit Q-bus system, the MicroPDP-11/83 will certainly fill your requirements.

The MicroPDP-11/83, by combining an 18-megahertz J-11 chip and a companion floating-point accelerator chip with a new private memory interconnect on one module, increases throughput and boosts realtime computing power to much higher performance levels.

Enclosures for the MicroPDP-11/83 include a floorstand model on casters and a 42-inch-high cabinet that accommodates two chassis for a total of 16 backplane slots.

Choose from a wide array of mass-storage devices and communications interfaces. The MicroPDP-11/83 supports Digital's high-capacity, high-performance disk subsystems—disks that were once available only on much larger systems, now allow you to expand to greater than 1 Gbyte of storage.

Imagine having all this power, more storage capacity, greater I/O capability, and room for more options in the same versatile MicroPDP-11 packaging. And it's all available to a greater number of concurrent users.

PDP-11/84

Your PDP-11 investment continues with the most powerful, yet most cost-effective UNIBUS PDP-11—the PDP-11/84. Its computing power is the same as the powerful MicroPDP-11/83, and like all UNIBUS PDP-11s it supports existing UNIBUS device interfaces, UNIBUS peripherals, and other UNIBUS options, such as battery backup.

Based on the high-speed, 18-megahertz, J-11 chipset, the PDP-11/84 can handle more users and more memory-resident programs than previous UNIBUS PDP-11s, while delivering maximum system throughput. You can easily expand main memory to 4 Mbytes.

Packaging for the PDP-11/84 includes the 10.5-inch rackmountable enclosure with expansion capabilities. The 42-inch-high cabinet features an open cardcage that offers 12 to 21 slots of which a maximum of 17 can be devoted to system option expansion.

The UNIBUS links the PDP-11/84 processor with the industry's most comprehensive set of mass-storage subsystems. Using the standard Digital Storage Architecture (DSA) and compatible controllers, the PDP-11/84 offers large configuration flexibility.

With the PDP-11/84 you can support more terminals, handle more users efficiently, and develop more sophisticated applications than you ever could with previous PDP-11s.

▪ PDP-11 System Comparison Chart

This chart compares our J-11-based systems with other members of the PDP-11 family.

	MicroPDP-11/23	*MicroPDP-11/53*	*MicroPDP-11/73*	*MicroPDP-11/83*	*PDP-11/24*	*PDP-11/44*	*PDP-11/84*
CPU Microprocessor	F-11 chip	J-11 chipset	J-11 chipset	J-11 chipset	F-11 chip	F-11 chip	J-11 chipset
Performance (PDP-11/70 = 1.0)	0.2	0.5	0.7	1.2	0.25	0.65	1.2
Microcycle Time (nanoseconds)	300	267	267	222	300	180/240	222
Minimum Memory	256 Kbytes	512 Kbytes onboard memory	1 Mbyte	2 Mbytes ECC PMI	1 Mbyte ECC	1 Mbyte ECC	1 Mbyte ECC PMI
Maximum Memory	4 Mbytes	4 Mbytes	4 Mbytes	4 Mbytes	4 Mbytes	4 Mbytes	4 Mbytes
Cache Memory (8 Kbyte)	no	no	yes	yes	no	yes	yes
Floating-point Microcode	optional	standard	standard	standard	optional	optional	standard
Floating-point Coprocessor	optional	no	no	yes	optional	optional	yes
Commercial Instruction Set	optional	no	no	no	standard	standard	standard

	MicroPDP-11/23	*MicroPDP-11/53*	*MicroPDP-11/73*	*MicroPDP-11/83*	*PDP-11/24*	*PDP-11/44*	*PDP-11/84*
Number of Onboard Serial Lines	2	2	1	1	2	2	1
Storage (Integrated by Digital)							
Flexible Disks	800 Kbyte	1.2 Mbyte	800 Kbyte	800 Kbyte	800 Kbyte	800 Kbyte	800 Kbyte
Fixed/Removable Disks	31 Mbytes 71 Mbytes (external)	42 Mbytes 71 Mbytes (external)	71 Mbytes 159 Mbytes	456 Mbytes 159 Mbytes 71 Mbytes	456 Mbytes 205 Mbytes	456 Mbytes 205 Mbytes	456 Mbytes 205 Mbytes
Cartridge Tapes	95 Mbytes (external)	95 Mbytes (external)	95 Mbytes	95 Mbytes	95 Mbytes	95 Mbytes	95 Mbytes
Streaming Tapes	40 Mbytes	40 Mbytes	40 Mbytes	40 Mbytes	145 Mbytes 40 Mbytes (TU81) 40 Mbytes (TSU05)	145 Mbytes 40 Mbytes (TU81) 40 Mbytes (TSU05)	145 Mbytes 40 Mbytes (TU81) 40 Mbytes (TSU05)
Enclosures							
Pedestal/Tabletop	yes	yes	yes	yes	no	no	no
Rackmount	yes	yes	yes	yes	yes	yes	yes
25 inch-high Floorstand	no	no	yes	yes	no	no	no

	MicroPDP-11/23	*MicroPDP-11/53*	*MicroPDP-11/73*	*MicroPDP-11/83*	*PDP-11/24*	*PDP-11/44*	*PDP-11/84*
42-inch high Cabinet	no	no	no	yes	no	no	yes
Maximum Number of Users	26 (with cable concentrator)	26 (with cable concentrator)	41 (with cable concentrator)	65 (with cable concentrator)	120	120	120

▪ PDP-11 Operating Systems

Operating System	Description	Major Applications	Compatible Bus	Languages Most Often Used	Human Interface	Important Attribute	Additional Layered Product
RSX Family RSX-11S RSX-11M RSX-11M-PLUS Micro/RSX	Multiuser Realtime General Purpose Multitasking Priority Scheduling	Laboratory Manufacturing Engineering Process Control Program Development	Q-bus UNIBUS DIBOL	MACRO-11 FORTRAN-77 BASIC-PLUS-2 PDP-11 PASCAL COBOL-81	DCL DECnet	Performance in technical environments	A-to-Z DATATRIEVE-11 DECdx DECmail-11 Development Kit FMS PDP-11 Symbolic Debugger RTEM-11 SORT/MERGE WPS PLUS
RSTS Family RSTS/E Micro/RSTS	Multiuser General Purpose Timesharing	Commercial Finance Education Program Development	Q-bus UNIBUS	BASIC-PLUS-2 BASIC-PLUS DIBOL MACRO-11 COBOL-81 FORTRAN-77	DCL	Ease of use Low cost per user	DATATRIEVE-11 DECdx DECmail-11 Development Kits DIBOL FMS PDP-11 Symbolic Debugger WPS PLUS

Operating System	Description	Major Applications	Compatible Bus	Languages Most Often Used	Human Interface	Important Attribute	Additional Layered Product
RT-11	Single user Realtime Process Control	Laboratory Scientific	Q-bus UNIBUS	MACRO-11 FORTRAN-77 PASCAL BASIC-PLUS	DCL	Simplicity Flexibility Realtime performance	
ULTRIX-11	Multiuser General Purpose Timesharing Native UNIX™ implementation System V, Version 7 and 2.9 BSD	Program Development Telecommunications Education Laboratory Scientific	Q-bus UNIBUS	C	C-Shell	Performance for program development Portability	
DSM-11	Multiuser Timesharing	Medical Hospital Commercial Text processing Data management	Q-bus UNIBUS	ANSI standard MUMPS	MUMPS	Performance Programmer productivity	

Operating System	Description	Major Applications	Compatible Bus	Languages Most Often Used	Human Interface	Important Attribute	Additional Layered Product
MicroPower/ Pascal Micro/RSX RSX-11M-PLUS RT-11	Host development for small realtime target systems Dedicated applications	Robotics Industrial automation Nuclear research Medical device research	Q-bus	PASCAL	Same as host (development) None on target system	Low cost, high speed applications with no operating systems on target	
CTS-300 (RT-11 based)	Multiuser General Purpose	Small Business	Q-bus UNIBUS	DIBOL (Digital's Business-oriented Language)		Performance for program development	
IAS	Multiuser Timesharing	Government Banks Manufacturing Hospitals Education	Q-bus UNIBUS	Fortran-77 PDP-11 COBOL BASIC-PLUS-2		Performance Cost effective	

The above information does not imply that each operating system is used only in the area(s) noted but is intended to provide assistance in selecting the operating system best suited to your needs.

™UNIX is a trademark of the American Telephone & Telegraph Company.

Chapter 2 ▪ MicroPDP-11/83 System

▪ Introduction

Digital's MicroPDP-11 supermicrosystem family of computers, including the MicroPDP-11/83, the MicroPDP-11/73, and the MicroPDP-11/53, provides users with low-cost, medium- to high-performance systems designed for real-time, timesharing, and batch applications.

The MicroPDP-11 family supports the 16-bit PDP-11 architecture and all PDP-11 operating systems. This architectural commonality enables MicroPDP-11 supermicrosystems to be compatible with one another and with larger Q-bus and UNIBUS systems and to run programs that have been developed on these systems.

This chapter describes the MicroPDP-11/83 system as well as characteristics and features common to the MicroPDP-11/83, the MicroPDP-11/73, and the MicroPDP-11/53.

▪ MicroPDP-11/83 Features

The MicroPDP-11/83 is a supermicrocomputer based on Digital's high-performance, 18-MHz, 16-bit processor module, the KDJ11-BF. The MicroPDP-11/83 Q-bus-based microcomputer extends the MicroPDP-11 family to a new level of performance and utility.

The premium-performance system for those applications that require the highest integer and floating-point speeds, the MicroPDP-11/83 can accommodate up to 65 users depending on the application. The MicroPDP-11/83 outperforms the PDP-11/70 in many applications.

The MicroPDP-11/83 serves users who require data processing for a variety of applications in the laboratory, factory, medical, educational, and engineering settings. It also serves the commercial customer who requires data processing for such applications as business, office, banking, insurance, and administration.

Because it is fully compatible with software developed for any PDP-11, the MicroPDP-11/83 offers MicroPDP-11 users easy growth and expansion capabilities. Each MicroPDP-11/83 system can be easily integrated into distributed processing environments with other Digital system products via DECnet or into local area networks with Ethernet. These compatibility and communications features place the MicroPDP-11/83 among the most practical and cost-effective computing solutions available today.

The KDJ11-BF CPU features the J-11 implementation of the full PDP-11 instruction set including EIS, floating-point accelerator, memory management, 4-Mbyte addressing capability, 8-Kbyte cache, 32-Kbyte bootstrap and diagnostic ROM, and a switch-selectable console serial-line unit.

The MSV11-J PMI memory includes a 1- or 2-Mbyte capacity quad-height board designed with 256-Kbyte RAM. A system can be configured with up to four Mbytes of PMI memory. It supports parity error-correction code, block-mode direct-memory access (DMA) transfers, and the PMI architecture that facilitates high-speed data transfers including double-word reads. Chapter 6 includes a detailed discussion of PMI technology.

The MicroPDP-11/83 is offered in the BA23 pedestal or rackmount enclosure. Also available is the BA123 floorstand enclosure, which includes a 12-slot backplane and space for one 5.25-inch cartridge-tape or floppy drive and up to three 5.25-inch mass-storage disk drives. There is ample space for memory and communications options and room for connecting up to 20 I/O devices.

Furthermore, the MicroPDP-11/83 is available in the H9642 42-inch-high cabinet that can incorporate up to two BA23 enclosures. The H9642 cabinet provides support of RA series fixed or removable-media disk mass storage, backplane expansion up to 16 usable slots, and additional I/O connectivity.

The MicroPDP-11/83 consists of a KDJ11-BF microprocessor module and an MSV11-J 1- or 2-Mbyte memory module. See Figure 2-1. These two modules communicate locally via the high-speed Private Memory Interconnect (PMI) bus using 22-bit address lines and 16-bit data lines.

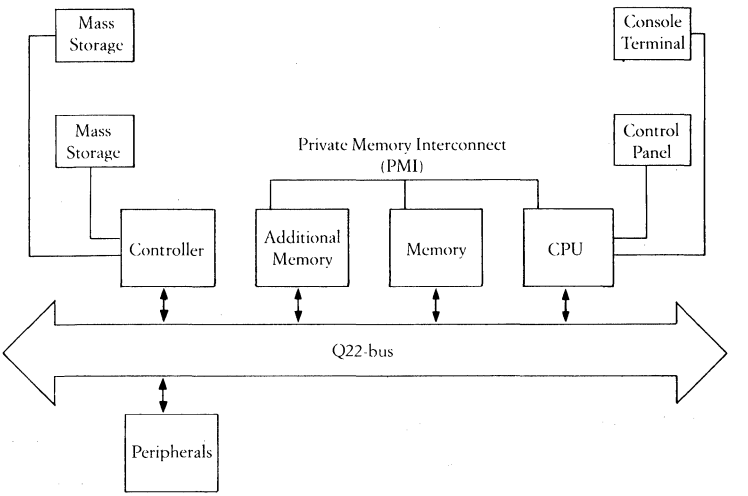


Figure 2-1 ▪ MicroPDP-11/83 System Block Diagram

KDJ11-BF CPU Module

The KDJ11-BF is a quad-height processor module designed for use in high-speed, realtime applications and for multiuser, multitasking environments.

Features of the KDJ11-BF CPU module include:

- J-11 (DCJ11) microprocessor chipset, including an 18-MHz clock, 16-bit I/O, 32-bit internal data path, and addressing capability up to 4 Mbytes.
- Floating-point accelerator chip.
- Complete PDP-11 instruction set including the Extended Instruction Set (EIS).
- 8-Kbyte direct-mapped cache memory.
- Q22-bus interface that supports block-mode DMA and up to four Mbytes of physical memory.
- Line-frequency clock.
- Four levels of interrupts.
- Powerfail/auto-restart.

- Console emulator in microcode.
- One console terminal serial-line unit.
- 32-Kbyte erasable read-only memory (ROM)

MSV11-J Memory Module

The MSV11-J is a quad-height memory module supporting the PMI protocol of the KDJ11-BF processor. Features of the MSV11-J PMI memory include:

- 1- or 2-Mbyte capacity.
- 256-Kbyte RAM chips.
- Parity and flag error detection.
- Error correction on single bit errors.
- Block-mode DMA protocol support.
- LEDs for parity detection and power status.
- Switch selection of starting (256-Kbyte boundaries) and CSR addresses.
- High-speed (3.8 Mbytes/second) Private Memory Interconnect.

▪ MicroPDP-11/83 Configurations

MicroPDP-11/83 systems are available in standard systems and in system building block configurations. Detailed configuration information can be found in the PDP-11 Systems and Options Catalog.

Standard System

The MicroPDP-11/83 is available in three different enclosures—BA23, BA123, and MicroSystem Cabinet. All systems have a KDJ11-BF processor with floating point accelerator, 2-Mbyte PMI, one TK50 tape subsystem and either one or two DHQ11 direct memory access multiplexers.

BA23	KDJ11-BF/Floating Point/BA23
	2 Mbytes PMI/RD54
	TK50/DHQ11
BA123	KDJ11-BF/Floating Point/BA123
	2 Mbytes PMI/RD54
	TK50/DHQ11 (2)
MicroSystem Cabinet	KDJ11-BF Floating Point
(H9642)	2 Mbytes PMI/H9642/BA23 (2)/ TK50/RA81
	DHQ11 (2)

System Building Blocks

The system building block is another method of selecting a microsystem. With system building blocks, you first select the CPU module and then the enclosure. Each of the remaining components is then chosen from its own specific menus. The menu categories are mass-storage device, load device, communications device, power cord, documentation, diagnostics, operating-system license, console terminal, terminals, printers, and cables. BA23, BA123, and H9642 cabinet enclosures are available for system building blocks.

Consult the current *PDP Systems and Options Catalog* for further details on system building blocks.

Enclosures

The MicroPDP-11/83 is available in four packaging variations:

- BA23A-AR rackmount enclosure.
- BA23A-AF pedestal enclosure.
- BA123-A floorstand enclosure with casters.
- H9642 cabinet with dual BA23A.

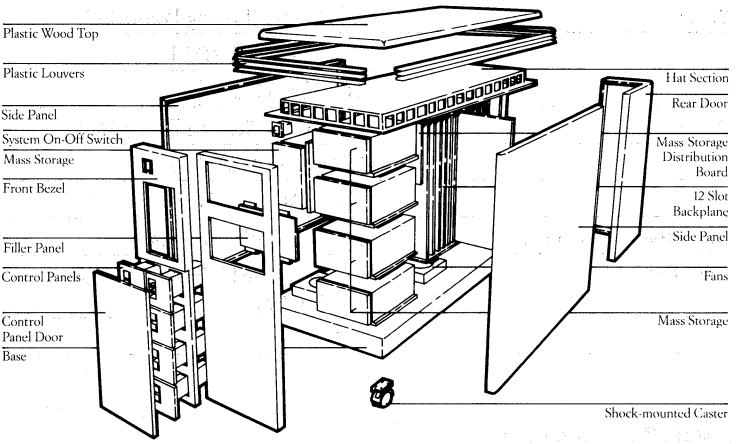


Figure 2-2 ▪ Pedestal/Tabletop/Rackmount Enclosure Chassis

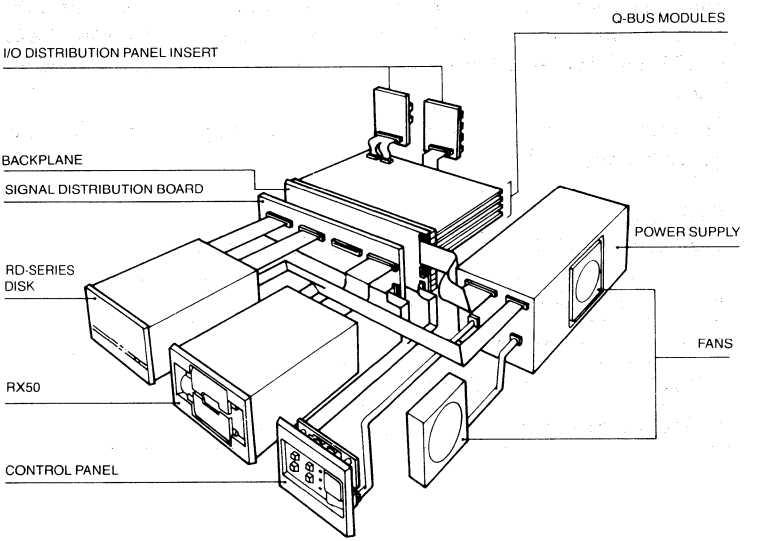


Figure 2-3 ▪ Cabinet Enclosure Chassis

Mass Storage

Digital's Mass Storage Control Protocol (MSCP), the design basis for all of Digital's new storage systems, manages disk accesses, data transfers, error recovery, and performance optimizations. Because the operating system deals with MSCP as opposed to specific disk drives, you need not modify it to accept new kinds of drives.

Mass storage for MicroPDP-11/83 systems includes the RA81 456-Mbyte disk drive and TK50 95-Mbyte tape drive as well as the RA60 205-Mbyte removable-disk drive, TSV05 40-Mbyte industry standard 1600 b/in streaming tape drive, RD54 159-Mbyte fixed-disk drive, RD53 71-Mbyte fixed-disk drive, RD32 42-Mbyte half-height disk drive, the RX50 800-Kbyte diskette drive, and the RX33 1.2-Mbyte half-height diskette drive.

▪ MicroPDP-11 Specifications

The specifications provided below describe the MicroPDP-11 supermicrosystems enclosures, the BA23, BA123, and the H9642. Refer to the *PDP Systems and Options Catalog* for further details on system specifications.

Backplane Assemblies

The backplane for all MicroPDP-11 sy stems in the BA23 enclosure has an 8-slot by 4-row backplane for mounting dual- or quad-height Q-bus modules. The Q-bus is implemented in all of the eight slots.

In the BA123 enclosure, the backplane has 13 slots and 4 rows. No additional backplanes can be connected to it. The first 12 slots provide space for dual- or quad-height Q-bus modules. The thirteenth slot provides space for disk cable management. The Q-bus is not implemented in this slot.

Customers may mount up to two BA23 enclosures in an H9642 cabinet, which will allow for 16 available slots of which 2 are reserved for cable management.

Power Supplies

In all MicroPDP-11 systems, both BA23 and H9642 enclosures have a 230-watt power supply that drives power to all eight slots of each BA23 backplane, mass-storage devices, control-panel switches and indicators, and to the two fans.

The BA123 enclosure has 460-watt power supply consisting of two regulators. Each regulator supplies 230 watts of power to one-half of the slots in the backplane, to the mass-storage devices inside the system, to the control-panel switches and indicators, and to the three fans.

Power Supply for the BA23 and H9642 Systems

120 Vac tolerance	88–128 VRMS
240 Vac tolerance	176–256 VRMS
120 Vac frequency	47–63 Hz
240 Vac frequency	47–63 Hz
Power Supply for the BA123 System	
120 Vac tolerance	88–128 VRMS
240 Vac tolerance	176–256 VRMS
120 Vac frequency	47–63 Hz
240 Vac frequency	47–63 Hz

Control Panels

The MicroPDP-11 BA23, BA123, and H9642 control panels contain the switches and indicators for user control of the system. There is an on/off switch, and Halt, Restart, Write-Protect, and Ready buttons. These controls allow the user to apply and remove ac power, to stop and start the current program operation, and to protect the data stored on the 5.25-inch disk drives.

I/O Distribution Panels

The MicroPDP-11 I/O distribution panel, located at the rear of the system, is used to connect the cables from the console terminal, printing terminal, and other external devices that operate with the system. The BA23 I/O distribution panel has room for six panel inserts of which two are Type A and four are Type B. The BA123 I/O distribution panel has room for ten panel inserts of which four are Type A and six are Type B. The H3490 I/O distribution panel used in the H9642 cabinet has room for six Type A inserts and 11 Type B inserts. This allows for the connection of two BA23-A boxes in an FCC-compliant manner and allows for connection of up to 49 serial lines.

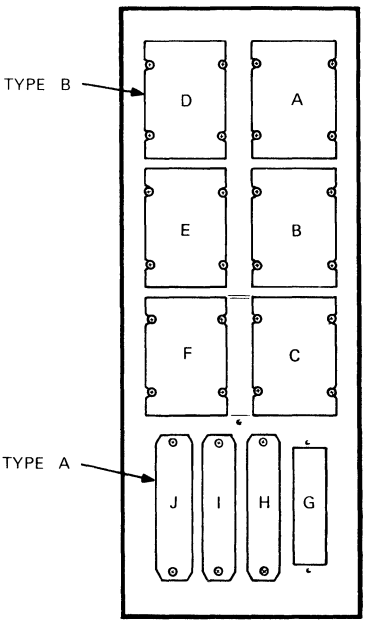


Figure 2-4 ▪ Floorstand Enclosure I/O Distribution Panel

Fans

In MicroPDP-11 systems, the BA23 and H9642 contain two brushless fans within the system unit that provide a flow of air from left to right (side to side) to cool the internal assemblies and modules.

The BA123 enclosure contains three brushless, variable-speed fans within the system unit that provide a flow of air from top to bottom to cool the internal assemblies and modules.

Size and Weight	
BA23 (pedestal) system	
Height	62.2 cm (24.5 in)
Width	25.4 cm (10.0 in)
Depth	72.4 cm (28.5 in)
Weight	32 kg (70 lb)
BA23 (rackmount) system	
Height	13.3 cm (5.25 in)
Width	48.3 cm (19.0 in)
Depth	64.8 cm (25.5 in)
Weight	25 kg (55 lb)
BA23 (floorstand with casters) system	
Height	62.2 cm (24.5 in)
Width	33.0 cm (13.0 in)
Depth	69.8 cm (27.5 in)
Weight	60 kg (133 lb)
H9642 (two BA23s in cabinet) system	
Height	106.0 cm (41.7 in)
Width	53.9 cm (21.2 in)
Depth	80.0 cm (31.4 in)
Weight	150.4 kg (332 lb)

Power Requirements	
BA23 (pedestal/tabletop/rackmount) system	
120 Vac (nominal)	4.4 A
240 Vac (nominal)	2.2 A
Input power	345 W
BA123 (floorstand with casters) system	
120 Vac (nominal)	6.8 A
240 Vac (nominal)	4.4 A
Input power	690 W

H9642 (two BA23s in cabinet) system	
120 Vac (nominal)	4.4 A
240 Vac (nominal)	2.2 A
Input power	Depends on mass storage

Operating Environment	
Temperature range	15°C–32°C(59°F–90°F)
Relative humidity	20%–80% (noncondensing)
Maximum altitude	8000 feet above sea level

▪ Additional Information

Additional information concerning the central processor including general registers and related material, PMI, memory system, console functions, line-time clock, and console is available in Chapter 6.

▪ MicroPDP-11 Architecture Summary

This section discusses the computer architecture characteristics of the MicroPDP-11 computer family including that of the MicroPDP-11/83, MicroPDP-11/73, and MicroPDP-11/53.

Computer architecture is the characteristics of the computer as observed by the operator and programmer at the assembly-language level. These characteristics include instructions sets, data types, addressing modes, registers, address space, and memory management. This section discusses these architectural characteristics.

Multiple system implementations of common computer architectures have allowed Digital's customers to continue to upgrade and expand, at the lowest possible cost, as their needs have changed. Within the PDP-11 family, customers can move to other computers without re-investing in software, peripherals, communications devices, or training. Common computer architectures ensure computer compatibility. For a detailed description of the PDP-11 architecture, refer to the *PDP-11 Architecture Handbook*.

The MicroPDP-11 architecture includes the following characteristics:

- 64-Kbyte virtual-address space.
- 4-Mbyte physical-address space.
- 16-bit word size.
- Memory management.
- Two sets of eight 16-bit general registers.
- Multiple addressing modes.
- Eight processor-priority levels.
- Variable instruction size.
- Full set of PDP-11 privileged registers.
- Full set of PDP-11 data types.
- Full set of PDP-11 instructions.

The MicroPDP-11 instruction set uses 16-bit addressing that provides a directly addressable virtual-address space of 65,536 (64K) bytes. Actual memory capacity in the MicroPDP-11 is 4,096,000 (4M) bytes. Memory management translates the 16-bit virtual addresses into the full 22-bit physical addresses needed to address 4 Mbytes. The processor's memory-management hardware includes mapping registers used by the operating system that provide page protection by operating mode.

The MicroPDP-11 systems have two sets of eight 16-bit general registers that can be used for high-speed, temporary storage or as accumulators, index registers, or base registers. Two registers with special purpose are the program counter and the stack pointer.

The MicroPDP-11 processors offer a variety of addressing modes, including an indexed-addressing mode, that uses the general registers to identify instruction operand locations.

Floating-point instructions are standard on the MicroPDP-11/83 in hardware and in microcode, and on the MicroPDP-11/73 and the MicroPDP-11/53 in microcode only.

▪ Address Space and Memory

The MicroPDP-11 uses the 8-bit byte for addressing. PDP-11 instructions use a 16-bit virtual address to identify a byte location. Memory management translates 16-bit virtual addresses into the 22-bit physical addresses needed to address 4 Mbytes of memory.

MicroPDP-11 instructions can address memory using either direct addressing or indirect (deferred) addressing. With direct addressing, the address in one of the general registers points directly to the data in memory. With indirect addressing, the address in a general register points to an address stored in memory, which in turn points to the data.

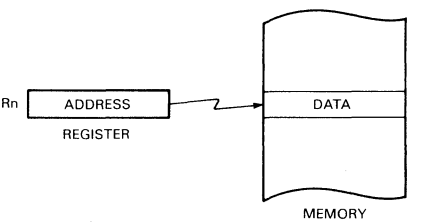


Figure 2-5 ▪ Direct Addressing

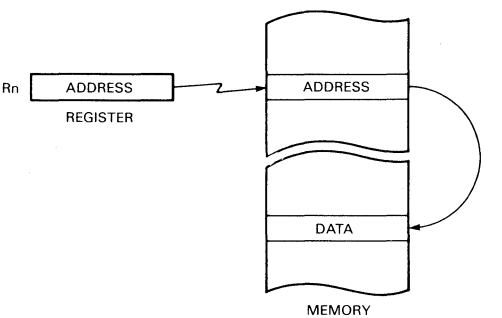


Figure 2-6 ▪ Indirect or Deferred Addressing

In MicroPDP-11 memory locations and peripheral-device (I/O-device) registers, the last 8-Kbytes of physical-address space are reserved for I/O-device addressing. Other physical-memory locations have been reserved for interrupt and trap handling.

Physical-address Space

Physical-address space is a contiguous series of word-addressable hardware locations used to define memory and I/O-device registers. The MicroPDP-11 architecture specifies that physical addresses may be up to 22 bits long and provides a physical-address space of 4 Mbytes. The last 8-Kbytes are used for I/O-device addressing.

Virtual-address Space

Through MicroPDP-11 memory-management hardware, the operating system provides an execution environment in which users can write programs without having to know where the programs are loaded in physical memory. In this environment, users can write programs that are too large to fit into the allocated physical memory. This environment is called virtual-address space.

A virtual address is a 16-bit integer that a program uses to identify a storage location in virtual memory. Virtual memory may be the set of all physical-memory locations in the system plus the set of disk blocks that the operating system designates as extensions of physical memory.

A program written for a MicroPDP-11 processor sees a 16-bit address space that references up to 64 Kbytes of memory. This 64-Kbyte window is known as the program's virtual-address space. Each MicroPDP-11 program's virtual-address space begins with address 0 and can extend upward to a maximum of 64 Kbytes.

Memory Management

Memory management enables the operating system to map virtual addresses into physical addresses. This physical address is then used to specify a location in the storage device.

The MicroPDP-11 system has to convert a fairly small virtual address to a large physical address. This allows the 16-bit MicroPDP-11 processor to access a very large physical memory a little at a time. Figure 2-7 shows the MicroPDP-11 translating the 16-bit virtual addresses into the 22-bit physical addresses.

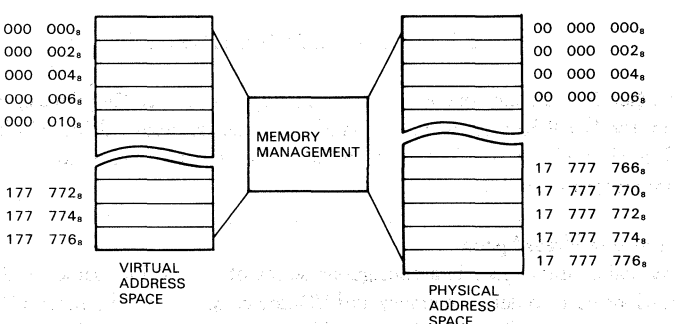


Figure 2-7 ▪ MicroPDP-11 Memory Management

Memory Protection

MicroPDP-11 memory management provides a second important feature beyond managing where the code and data should go. Memory management also controls who may have access to the code and data. This is important if a multiuser system is to protect the operating-system software from the users as well as protect individual programs from one another.

The view that each running program has of physical memory is completely controlled by the operating system. To the program, sections of physical memory can be labeled read/write, read-only, or invisible. For example:

- The program code should be marked read-only if the programmer has written pure code that does not modify itself.
- Fixed program data can also be marked read-only so that it cannot be damaged.
- Variable program data is marked read/write.
- Sections of memory the program has no need to know about are made invisible.

The MicroPDP-11 processors assign these protection attributes on a per-page basis.

Registers and Stacks

A register is a location within the processor that can be used for high-speed, temporary data storage and addressing, or as an accumulator during computation.

The MicroPDP-11/83, MicroPDP-11/73, and MicroPDP-11/53 use two sets of eight 16-bit general registers. Each set has the six general purpose registers, a program counter, and a stack pointer.

A stack is an array of consecutively addressed data items that are referenced on a last-in, first-out basis using a register. Data items are added to and removed from the low address end of the stack. A stack grows toward lower addresses as items are added and shrinks toward higher addresses as items are removed.

A stack can be created anywhere in the user's program address space. Any register can be used to point to the current item on the stack. The operating system, however, automatically reserves portions of each process address space for stack data structures. User software references its stack data structure, called the user stack, through a general register designated as the stack pointer. When the user runs a program image, the operating system automatically provides the address of the area designated for the user stack.

MicroPDP-11 Registers

The MicroPDP-11 registers shown in Figure 2-8 can be used as operands for arithmetic and logical operations or for addressing in memory. Register operations are internal to the processor and do not require bus cycles (except for instruction fetch). All memory and peripheral device data transfers require bus cycles and longer execution time. Thus general purpose registers used for processor operations result in faster execution times.

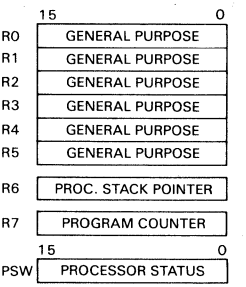


Figure 2-8 ▪ MicroPDP-11 Registers

The program counter (PC or R7) contains the address of the next instruction to be executed. Normally, the PC is used only for addressing and not for arithmetic or logical operations.

When an interrupt or trap occurs, the processor status word (PSW) and the program counter are saved on the processor stack. The stack pointer (SP or R6) contains the address of the top of this stack in memory. The PSW and PC contain all the information needed for the processor to resume execution where it left off. The last-in, first-out stack allows orderly processing of interrupts and traps even when the processor is already processing.

Processor Status Word

The processor status word (PSW) is a special processor register found in the MicroPDP-11 that is used to check a program's status and to control synchronous error conditions. The processor status word, shown in Figure 2-9, contains two sets of bit fields—condition codes and trap enable flags.

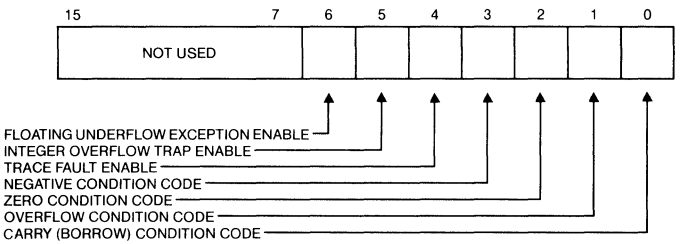


Figure 2-9 ▪ Processor Status Word

The condition codes indicate the outcome of a particular logical or arithmetic operation. The branch-on-condition instructions can be used to transfer control to a code sequence that handles the condition.

There are two kinds of exceptions that concern the user process—trace faults and arithmetic exceptions. The trace fault is used to debug programs or evaluate performance. Arithmetic exceptions include:

- Integer or floating-point overflow, in which the result was too large to be stored in the given format.
- Integer or floating-point divide-by-zero, in which the divisor supplied was zero.
- Floating-point underflow, in which the result was too small to be expressed in the given format.

When an exception occurs, the processor immediately saves the current state of execution and traps to the operating system. The operating system automatically searches for a procedure that wants to handle the exception.

▪ Addressing Modes

There are eight basic PDP-11 addressing modes that use the general registers to identify operand location. The processor's addressing modes allow almost any operand to be stored in a register or in memory, or as an immediate constant.

MicroPDP-11 addressing modes include register, register-deferred, autoincrement, autoincrement-deferred, autodecrement, autodecrement-deferred, index, and index-deferred.

- *Register mode* (mode 0)—contains the operand in the register.
- *Register-deferred mode* (mode 1)—contains the address of the operand in the register.
- *Autoincrement mode* (mode 2)—interprets the contents of the register as the address of the operand in memory and, in addition, increments the contents of the register by 2 (word instructions) or by 1 (byte instructions) after the operand is accessed in memory. This leaves the register pointing to the next consecutive word or byte and makes stepping through a list of operands easier. Because both R6 (stack pointer) and R7 (program counter) normally contain addresses, they are always autoincremented by 2.
- *Autoincrement-deferred mode* (mode 3)—uses the contents of the register as a pointer to the address of the operand. The pointer in the register is then incremented by 2 after the address is located. Where mode 2 steps through a list of sequential operands, mode 3 steps through a list of sequential addresses that in turn point to operands stored anywhere in memory.
- *Autodecrement mode* (mode 4)—decrements the contents of the register by 2 (word instructions) or by 1 (byte instruction) before using the register as the address of an operand in memory. Where mode 2 steps through a list of operands at ascending addresses, mode 4 does the same by descending addresses.
- *Autodecrement-deferred mode* (mode 5)—interprets the contents of the register as the address of a word in memory, which in turn points to the operand. The register is decremented by 2 before accessing the address in memory. Where mode 3 steps through a list of addresses in ascending memory order, mode 5 steps through the list in descending memory order.
- *Index mode* (mode 6)—adds the contents of the word immediately following the instruction to the contents of the register, and uses the resulting sum as the address of an operand in memory. This allows you to specify the starting address of a list independently of the offset of an entry in the list. By changing the starting address but not the index, you can move the fifteenth entry in list A to the fifteenth entry in list B. By changing the index but not the starting address, you can move from the fifteenth entry in list A to the twentieth entry in list A. The starting address can be specified in the register and the offset in the word following the instruction, or vice versa.

- *Index-deferred mode* (mode 7)—adds the word following the instruction to the register in the same way as mode 6, but the resulting sum is used as the address of a word in memory that in turn points to the operand. Where mode 6 accesses operands stored in a list or table, mode 7 uses addresses stored in a list or table to access operands stored anywhere in memory.

Modes 2, 3, 6, and 7 can be particularly useful in conjunction with the program counter. The resulting addressing will be independent of where in memory the instruction is executed. Each time the processor implicitly uses the program counter to fetch a word from memory, the program counter is automatically incremented by 2 after the fetch is completed.

PC immediate mode is a special case of mode 2, using the program counter (R7) as the register. It accesses the word immediately following the instruction and is a fast way to read a constant operand.

PC absolute mode is a special case of mode 3, where an absolute address (that is, constant regardless of where in memory the instruction is executed) is stored in the word immediately following the instruction. This absolute address is used as a pointer to the operand.

PC relative mode is a special case of mode 6, using the program counter (R7) as the register. The updated contents of the program counter (instruction address + 4) are added to the contents of the word immediately following the instruction (the offset) and the sum is used as a pointer to the operand in memory. The offset and the position of the operand relative to the instruction are independent of where they are located in memory, so PC relative mode is helpful in writing position-independent code.

PC relative-deferred mode is a special case of mode 7, using the program counter as the register. The word following the instruction (the offset) is added to the updated program counter (instruction address + 4), and the resulting sum is used as a pointer to a location which in turn contains the address of the operand.

▪ Exceptions and Interrupts

While running one process, the processor executes instructions and controls data flow to and from peripherals and main memory. To share processor, memory, and peripheral resources among many processes, the processor provides two arbitration mechanisms called exceptions and interrupts. Exceptions are events that occur synchronously with respect to instruction execution; interrupts are external events that occur asynchronously.

The flow of execution can change at any time. The processor distinguishes between changes in flow that are local to a process and those that are of systemwide context and independent of any particular process. Process-local changes occur as the result of a user software error or when user software calls operating-system services. Process-local changes in program flow are handled through the processor's exception-detecting mechanism and the operating system's exception dispatcher.

Systemwide changes in flow generally occur as the result of interrupts from devices or interrupts generated by the operating-system software. Interrupts are handled by the processor's interrupt-detection mechanism and the operating system's interrupt-service routines (systemwide changes in flow may also occur as the result of severe hardware errors; these are handled either as special exceptions or high-priority interrupts).

Systemwide changes in flow generally take priority over process-local changes in flow. The processor uses a priority system for servicing interrupts. To arbitrate between all possible interrupts, each kind of interrupt is assigned a priority, and the processor responds to the highest-priority pending interrupt. For example, interrupts from realtime I/O devices would take precedence over interrupts from mass-storage devices, terminals, printers, and other less time-critical devices.

The processor services interrupts between instructions or at well-defined points during the execution of long iterative instructions. When the processor acknowledges an interrupt, it switches rapidly to a special systemwide context so that the operating system can service the interrupt. Systemwide changes in the flow of execution are handled in a way that makes them totally transparent to individual processes.

Exception and Interrupt Vectors

The processor can automatically initiate changes in the normal flow of program execution. The processor recognizes two kinds of events that cause it to invoke conditional software—exceptions and interrupts. Some exceptions, such as arithmetic traps, affect an individual process only. Others affect the system as a whole, such as a machine check. Interrupts include both device interrupts, such as those signaling I/O completion, and software-requested interrupts, such as those signaling the need for a context-switch operation.

The processor knows which software to invoke when an exception or interrupt occurs because it references specific locations, called vectors, to obtain the starting address of the exception or interrupt dispatcher. Each vector tells the processor how to service the event.

Processor-priority Levels

The MicroPDP-11 processors recognize eight processor-priority levels. The highest four processor-priority levels are reserved for interrupts generated by hardware, and the lowest four levels are reserved for interrupts requested by software.

Context Switching

In the multiprogramming environment, several individual streams of code can be ready to execute at the same time. Instead of allowing each stream to execute one at a time, the operating system can intervene and switch between the streams of code that are ready to execute. The stream of code the processor is executing at any one time is determined by its hardware context.

The hardware context contains the information that is loaded in the processor's registers that identify where the stream of instructions and data are located, which instruction to execute next, and what the processor is doing during execution.

The process is the stream of instructions and data defined by the hardware context. Each process has a unique identification in the system. The operating system switches between processes by requesting the processor to save one process hardware context and load another.

▪ MicroPDP-11 Processor Operating Modes

In a high-performance, multiprogramming system, the processor must provide the basis for protection and sharing among the processes competing for the system's resources. The basis for protection in the system is the processor's operating mode. The operating mode in which the processor executes determines the instruction-execution and memory access privileges. Instruction-execution privileges determines which instructions the processor will execute. Memory-access privileges determine which locations in memory the current instruction can access.

At any one time, the processor is either executing code in the context of a particular process, or it is executing code in the systemwide interrupt-service context. In the MicroPDP-11 context of a process, the processor recognizes three access modes—kernel, user, and supervisor.

- *Kernel mode* allows execution of all instructions. In a multiprogramming environment, the most privileged functions of the operating system—physical I/O operations, resource management, and job scheduling—are implemented in code that runs in kernel mode. The access-control provisions of memory management protect these elements from tampering by programs running in less privileged modes.
- *User mode* prohibits the execution of instructions, such as halt and reset, that would allow one program in a multiprogramming environment to harm the system as a whole. Each user's virtual-address space permits writing only into its own areas in memory.
- *Supervisor mode* has the same level of privilege as user mode and can be useful for programs being shared among users but still requiring protection.

▪ Data Types

The data type of an instruction operand determines the number of bits of storage to be treated as a unit and what the interpretation of that unit is. Each MicroPDP-11 instruction set operates on integer, floating-point, and character-string data types. For each of these data types, the selection of an operation immediately tells the processor the size of the data and its interpretation.

There are several variations of these primary data types. Table 2-1 provides a summary of the seven data types.

Table 2-1 ▪ Data Type Representation

Data Type	Size	Range in Decimal Notation	
Integer:		Signed	Unsigned
Byte	8 bits	– 128 to + 127	0 to 255
Word	16 bits	– 32768 to + 32767	0 to 65535
Floating Point:			
D_floating	8 bytes	16 decimal digits of precision	
F_floating	4 bytes	7 decimal digits of precision	
String:			
Character	0 to 65535 bytes	One character per byte	
Decimal	0 to 31 bytes	One digit per byte	

Integer data is stored as binary values in either byte or word formats on a MicroPDP-11. A byte is 8 bits; a word is 2 bytes. The MicroPDP-11 interprets an integer as either a signed value or an unsigned value. The sign in signed values is determined by the high-order bit.

Floating-point values are stored using a signed exponent and a binary, normalized fraction. Floating-point data types can represent positive and negative numbers with a much greater absolute value than integer data (as large as $1.7 \times 1{,}038$), or with a fractional value (as small as 0.29×10^{-38}). PDP-11-based systems use F__floating-point and D__floating-point data types. MicroPDP-11/83 processors support F__ and D__ floating-point in hardware and in microcode. The MicroPDP-11/73 and MicroPDP-11/53 processors support F__ and D__ floating-point in microcode only.

Single-precision F__floating-point data is 4 bytes long with an 8-bit excess 128 exponent. The effective 24-bit fraction yields approximately seven decimal digits of precision.

Double-precision D__floating-point data is 8 bytes long with an 8-bit excess 128 exponent. The effective 56-bit fraction yields approximately 16 decimal digits of precision.

Floating-point instructions are standard in the hardware of the MicroPDP-11/83 processor. They are standard in the microcode on the MicroPDP-11/73 and MicroPDP-11/53 processors. These units implement 46 microcoded instructions that perform arithmetic, logical, and conversion operations, and operate six to ten times faster than equivalent software routines.

Character data are strings of bytes containing any binary data such as ASCII codes. Various standards, the most common of which is ASCII, assign an interpretation to some or all of the 256 different codes that can be represented by this data type.

The first character in the string is stored in the first byte, the second character is stored in the second byte, and so on in ascending order. An 8-bit character is stored at any addressable byte in memory or in the low-order byte of a general register.

A character-string is a sequence of up to 65,535 bytes in memory which can be located in two consecutive registers or two consecutive words in memory. The first word is the length of the character string (unsigned integer format), and the second word is the address of the most significant character (MSC). Subsequent characters through the least significant character (LSC) are stored in ascending memory locations.

Several kinds of decimal-string data formats are used in business applications where their correspondence to COBOL data types, keypunch codes, or printable characters is used. All represent numbers consisting of 0 to 31 decimal digits, with an implied decimal point to the right of the least significant digit (LSD). All are stored in memory as contiguous bytes.

There are separate instructions for packed-decimal string operations and zoned-numeric string operations, and for the decimal conversions between the two.

The address of any data item is the address of the first byte in which the item resides. All integer, floating-point, and character-string data can be stored starting at any address in memory. A bit field, however, does not necessarily start at a byte boundary in memory. A bit field is simply a set of contiguous bits between 0 and 32 bits in length. The starting bit location is identified relative to a given byte address or register. The instruction set can interpret a bit field as a signed or unsigned integer.

▪ Instruction Set

An instruction consists of an operation code (opcode) and zero or more operands that are described by a data type and addressing mode. The MicroPDP-11 instruction set is based on over 100 different kinds of operations, each addressable in several ways.

To choose the appropriate instruction, it is necessary only to become familiar with the operations, data types, and addressing modes. For example, the ADD operation can be applied to any of several sizes of integer or floating-point operands, and each operand can be addressed directly in a register, directly in memory, or indirectly through pointers stored in registers or memory locations.

The MicroPDP-11 executes a large set of variable-length instructions, recognizes a variety of data types, and uses eight 16-bit registers. The following are the groupings of instructions:

- Load, store, and move instructions copy data between registers, memory, and I/O devices.
- Arithmetic instructions perform operations that interpret the numeric data, such as add, multiply, and negate.
- Shift and rotate instructions manipulate data within its original location.
- Data conversion instructions translate one data type to another.
- Logical instructions perform operations that manipulate bits and compare operands.
- Program control instructions redirect the flow of execution.
- Miscellaneous instructions include all of the remaining instructions.

Most instructions, other than floating-point and string-data instructions, use one of three basic formats—single operand, double operand, and branch. The single operand for instructions such as CLR (clear) and NEG (negate) is specified by the destination-address field, and the result is left in the same location. Instructions such as ADD and SUBtract use two operands specified by the source address and the destination address as input. These instructions leave the result at the destination address. See Table 2-2 for a summary of the Instruction Set.

A user program can test the outcome of an arithmetic or logical operation. The processor provides a set of condition codes and branch instructions for this purpose. The condition codes indicate whether the previous arithmetic or logical operation produced a negative or zero result or whether there was a carry, borrow, or overflow. There is a variety of branch-on-condition instructions—those for overflow and carry or borrow, and those for signed and unsigned relational tests.

Table 2-2 ▪ MicroPDP-11 Instruction Set Summary

Operator	Function
ABSD	Take absolute value of D_floating
ABSF	Take absolute value of F_floating
ADC	Add carry bit to word
ADCB	Add carry bit to byte
ADD	Add
ADDD	Add D_floating
ADDF	Add F_floating
ADDN(I)	Add numeric decimal strings
ADDP(I)	Add packed decimal strings
ASH	Arithmetic shift register
ASHC	Arithmetic shift two combined registers
ASHN(I)	Arithmetic shift numeric decimal string
ASHP(I)	Arithmetic shift packed decimal string
ASL	Arithmetic shift word left
ASLB	Arithmetic shift byte left
ASR	Arithmetic shift word right
ASRB	Arithmetic shift byte right
BCC	Branch if carry bit is clear
BCS	Branch if carry bit is set
BEQ	Branch if equal to zero
BGE	Branch if greater than or equal to zero
BGT	Branch if greater than zero
BHI	Branch if higher
BHIS	Branch if higher or same
BIC	Bit clear word
BICB	Bit clear byte
BIS	Bit set word
BISB	Bit set byte
BIT	Bit test word
BITB	Bit test byte
BLE	Branch if less than or equal to zero

Table 2-2 ▪ MicroPDP-11 Instruction Set Summary (Cont.)

Operator	Function
BLO	Branch if lower
BLOS	Branch if lower or same
BLT	Branch if less than zero
BMI	Branch if minus
BNE	Branch if not equal to zero
BPL	Branch if plus
BPT	Breakpoint trap
BR	Branch (unconditional)
BVC	Branch if overflow bit clear
BVS	Branch if overflow bit set
CCC	Clear all condition
CFCC	Copy floating condition codes
CLC	Clear carry condition code
CLN	Clear negative condition code
CLR	Clear word
CLRB	Clear byte
CLRD	Clear D__floating
CLRF	Clear F__floating
CLV	Clear overflow condition code
CLZ	Clear zero condition code
CMP	Compare word
CMPB	Compare byte
CMPC(I)	Compare character strings
CMPD	Compare D__floating
CMPF	Compare F__floating
CMPN(I)	Compare numeric decimal strings
CMPP(I)	Compare packed decimal strings
COM	Take one's complement of word
COMB	Take one's complement of byte
CSM	Call supervisor mode
CVTLN	Convert long integer to numeric decimal

Table 2-2 ▪ MicroPDP-11 Instruction Set Summary (Cont.)

Operator	Function
CVTLP	Long integer to packed decimal
CVTNL	Convert numeric decimal to long integer
CVTNP	Convert numeric decimal to packed decimal
CVTPL	Convert packed decimal to long integer
CVTPN	Convert packed decimal to numeric decimal
DEC	Decrement word
DECB	Decrement byte
DIV	Divide
DIVD	Divide D__floating
DIVF	Divide F__floating
DIVP(I)	Divide packed decimal strings
EMT	Emulator trap
HALT	Halt
INC	Increment word
INCB	Increment byte
IOT	Input/output trap
JMP	Jump
JSR	Jump to subroutine
LDCDF	Load and convert D__floating to F__floating
LDCFD	Load and convert F__floating to D__floating
LDCID	Load and convert integer to D__floating
LDCIF	Load and convert integer to F__floating
LDCLD	Load and convert long integer to D__floating
LDCLF	Load and convert long integer to F__floating
LDD	Load D__floating
LDEXP	Load exponent
LDF	Load F__floating

Table 2-2 ▪ MicroPDP-11 Instruction Set Summary (Cont.)

Operator	Function
LOCC(I)	Locate character
L2D	Load two string descriptors
L3D	Load three string descriptors
MARK	Facilitates stack cleanup
MATC(I)	Match character string
MFPD	Move from previous data space
MFPI	Move from previous instruction space
MFPS	Move from processor status word
MFPT	Move from processor type
MODD	Multiply and separate integer and D__floating
MODF	Multiply and separate integer and F__floating
MOV	Move word
MOVB	Move byte
MOVRC(I)	Move reverse-justified character string
MOVTC(I)	Move translated character string
MTPD	Move to previous data space
MTPI	Move to previous instruction space
MUL	Multiply
MULD	Multiply D__floating
MULF	Multiply F__floating
MULP(I)	Multiply packed decimal strings
NEG	Negate (take two's complement) of word
NEGB	Negate (take two's complement) of byte
NEGD	Negate D__floating
NEGF	Negate F__floating
NOP	No operation
RESET	Reset bus
ROL	Rotate register word left
ROLB	Rotate register byte left

Table 2-2 ▪ MicroPDP-11 Instruction Set Summary (Cont.)

Operator	Function
ROR	Rotate register word right
RORB	Rotate register word right
RTI	Return from interrupts
RTS	Return from subroutine
RTT	Return from interrupts
SBC	Subtract carry bit from word
SBCB	Subtract carry bit from byte
SCANC(I)	Scan character string
SCC	Set all condition code bits
SEC	Set carry condition code
SEN	Set negative condition code
SETD	Set D__floating mode
SETF	Set D__floating mode
SETI	Set floating for integer mode
SETL	Set floating for long integer mode
SEV	Set overflow condition code
SEZ	Set zero condition code
SKPC(I)	Skip character string
SPANC(I)	Span character string
STCDF	Store and convert D__floating to F__floating
STCDI	Store and convert D__floating to integer
STCDL	Store and convert D__floating to long integer
STCFD	Store and convert F__floating to D__floating
STCFI	Store and convert F__floating to integer
STCFL	Store and convert F__floating to long integer
STD	Store D__floating
STEXP	Store exponent
STF	Store F__floating
STFPS	Store floating-point program status word
STST	Store floating-point status
SUB	Subtract word
SUBD	Subtract D__floating

Table 2-2 ▪ MicroPDP-11 Instruction Set Summary (Cont.)

Operator	Function
SUBF	Subtract F_floating
SUBN(I)	Subtract numeric decimal strings
SUBP(I)	Subtract packed decimal string
SWAB	Swap bytes in word
SXT	Sign extend
TRAP	Trap
TST	Test word
TSTB	Test byte
TSTD	Test D_floating
TSTF	Test F_floating
TSTSET	Test word, set low bit
WAIT	Wait for interrupt
WRTLCK	Read/lock destination, write/unlock RO
XOR	Exclusive OR word

▪ Additional Documentation

- *KDJ11-B CPU Module User's Guide*
- *MicroPDP-11 Owner's Manual BA123 Floorstand*
- *MicroPDP-11 Owner's Manual BA23 Pedestal/Rackmount*
- *MicroPDP-11 Owner's Manual H9642 Cabinet*
- *MicroPDP-11 System Technical Manual Cabinet*
- *MicroPDP-11 System Technical Manual Floorstand*
- *MicroPDP-11 System Technical Manual Pedestal/Rackmount*
- *PDP-11 Architecture Handbook*

Chapter 3 ▪ MicroPDP-11/73 System

▪ Introduction

The MicroPDP-11/73 computer, featuring the J-11 CPU chipset, is a powerful and compact midrange system that provides performance that approaches that of the MicroPDP-11/83.

NOTE

The MicroPDP-11 series of processors have common characteristics. Those characteristics common to the MicroPDP-11/73 and other MicroPDP-11 processors are not repeated in this chapter. For information on registers, console functions, and line clock, refer to Chapters 2 and 6.

MicroPDP-11/73 provides a wide range of realtime or multiuser applications. The system is designed to sit on or under your desk or in the computer room. And systems can be configured to address large scale tasks that are usually beyond the range of similarly priced computers from other manufacturers.

As a member of Digital's Q-bus family of 16-bit supermicrosystems, the MicroPDP-11/73 can use a wide variety of software already written for PDP-11s, including operating systems, popular languages, application software, and communications packages—software developed expressly for the MicroPDP-11 family, such as A-to-Z, Digital's flexible, integrated system for small business.

The MicroPDP-11/73 is supported by Ethernet local area networks. Ethernet provides low-cost, high-speed local area communications among the various departments of an organization, whether they are located in one building or throughout a complex of buildings.

▪ MicroPDP-11/73 Features

The MicroPDP-11/73 is based on the KDJ11-BB, Digital's 16-bit Q-bus processor. As a true PDP-11, the MicroPDP-11/73 shows all the characteristics of the PDP-11 family and architecture. With these PDP-11 family characteristics and with thousands of applications already available for the PDP-11 family, the MicroPDP-11/73 is a quality choice for any business, technical, or commercial setting.

The MicroPDP-11/73 is a multiuser computer that can serve as a host to personal computers, or as a front end to larger hosts such as VAX systems. It is designed to fit comfortably in most environments—on a floorstand, on a tabletop, or as a rack-mountable system.

The MicroPDP-11/73 is especially well-suited to technical users and is an ideal choice for large, multitasking, multiuser environments requiring high-performance workstations, as well as for low-cost, timesharing commercial applications where excellent terminal response time is important.

Users can migrate easily to other MicroPDP-11 family member systems, to larger PDP-11 systems, or to large VAX systems. Digital provides a number of upgrade kits specifically designed to facilitate transition to high-level systems.

The MicroPDP-11 family brings together state-of-the-art hardware technology, proven PDP-11 computer architecture, and a large selection of working software all in one compact, inexpensive package. The increased CPU speed—three times that of the MicroPDP-11/23—will now accommodate 41 terminals (twelve active users), allowing users to share common data and information.

The use of EEROM on the CPU allows reconfiguration of the system through the keyboard without manual adjustments to the hardware.

Easily expanded storage, memory, communications, or other options are available as your applications grow. If your system configuration must be changed, the MicroPDP-11/73 makes it easy through MicroPDP-11 system compatibility.

The MicroPDP-11/73 features include:

- Low cost per user for high performance and minicomputer capability.
- Small packaging size available in a BA23 (floorstand, tabletop, rack-mount version) or BA123 enclosure.
- The ability to operate in any normal business environment with no special air conditioning or power requirements.
- Simple plug-in connectors for easy installation.
- Established PDP-11 architecture.
- Common hardware with other PDP-11 systems.
- Low maintenance costs and power consumption.

KDJ11-BB CPU Module

The powerful MicroPDP-11/73, designed and developed by Digital, can deliver fast processing and high user access capacity because of the J-11 chipset.

The KDJ11-BB CPU module includes the J-11 chipset, cache memory, a 32-Kbyte bootstrap/diagnostic ROM, a console serial line unit, and a line frequency clock. Memory is expandable in 1- and 2-Mbyte increments and, with its memory management unit, the MicroPDP-11/73 can address, protect, and segment up to 4 Mbytes of memory.

The floating-point microcode, a standard feature of the J-11 chipset, adds 46 instructions for floating-point math to enhance FORTRAN performance.

MSV11-Q Memory Module

The MSV11-Q is a 1-Mbyte, quad-height, 64-Kbyte, MOS memory module that is standard with MicroPDP-11/73 systems.

▪ MicroPDP-11/73 Configurations

The MicroPDP-11/73 is offered as a packaged system (BA23) or as a System Building Block in the BA23 or BA123 for maximum flexibility.

MicroPDP-11/73 hardware includes:

- Integrated dual-diskette drive (RX50), or single doubled-sided diskette drive (RX33).
- Optional, external 71- or 159-Mbyte fixed disk.
- Optional, external, 800-Kbyte dual-diskette drive.
- Disk/diskette controller using Mass Storage Control Protocol.
- Eight-slot or twelve-slot extended Q-bus backplane.
- Asynchronous and synchronous communications options.
- Complete PDP-11 instruction set including EIS.
- Floating-point standard in microcode.
- Full 4-Mbyte addressing capability.
- Line frequency clock.
- Four levels of supported interrupts.

Standard System

The MicroPDP-11/73 is available in a BA23-based standard system. The standard system includes the MicroPDP-11/73 CPU, 1-Mbyte of memory, RD53 71-Mbyte disk drive, RQDX3 disk controller, TK50 95-Mbyte tape drive and controller, in the BA23 pedestal/tabletop enclosure.

Diagnostics and documentation are included in some standard systems, but can be ordered separately as well.

System Building Blocks

The MicroPDP-11/73 is available as a system building block in a 13-inch wide floorstand enclosure (BA123), with shock-mounted casters. This features a 12-slot backplane and ample space for memory and communications options and room for up to 40 additional serial lines. The backplane of the pedestal enclosure (BA23) has 12 quad-height slots for easy expansion or integration with other systems. Enclosures can accommodate dual-height and quad-height modules.

With the floorstand enclosure you have the space for a 5.25-inch cartridge-tape subsystem and up to three additional 5.25-inch mass-storage disk subsystems.

Enclosures

The MicroPDP-11/73 is available in the BA23 or BA123 enclosure.

The BA23 features an 8-slot backplane and two full-height 5.25-inch mass storage cavities. The BA23 pedestal/rackmount enclosure can contain up to two RX33 diskette drives or one TK50, and up to two RD31/RD32 fixed-disk drives or one RD53.

The BA123, available in the MicroPDP-11/73 system building blocks, provides a floorstand enclosure option for MicroPDP-11/73.

Mass Storage

Digital's Mass Storage Control Protocol (MSCP), the design basis for all of Digital's new storage systems, takes care of all disk accesses, data transfers, error recovery, and performance optimizations. Because the operating system deals with MSCP as opposed to specific disk drives, you don't need to modify it to accept new kinds of drives.

Integral storage can include a 1.2-Mbyte floppy, a 42-, 71-, or 159-Mbyte formatted Winchester disk with a dual 800-Kbyte diskette subsystem, or the 95-Mbyte streaming cartridge tape. With BA23-based systems, storage can be expanded with full-height external devices.

▪ Additional Information

For information regarding MicroPDP-11/73 specifications such as backplane assemblies, power supplies, control panels, I/O distribution panels, fans, size and weight, power requirements, and operating environment, refer to Chapter 2.

Chapter 4 ▪ MicroPDP-11/53 and MicroPDP-11/53 PLUS Systems

▪ Introduction

The MicroPDP-11/53 and MicroPDP-11/53 PLUS supermicrosystems are Digital's entry-level Q-bus PDP-11 systems. The MicroPDP-11/53 and MicroPDP-11/53 PLUS systems provide powerful and compact midrange computing solutions to a wide range of needs.

NOTE

The MicroPDP-11 series of processors have common characteristics. Those characteristics common to the MicroPDP-11/53 and the MicroPDP-11/53 PLUS and other MicroPDP-11 processors are not repeated in this chapter. For information on registers, console functions, and line clock, refer to Chapters 2 and 6.

MicroPDP-11/53 systems feature the J-11-based (16-bit) KDJ11-DA CPU Module, which includes a 15-MHz clock and 512 Kbytes of onboard memory for improved price/performance over earlier MicroPDP-11 systems. MicroPDP-11/53 systems offer half-height storage devices for additional cost savings and high reliability.

The MicroPDP-11/53 PLUS supermicrosystem is based on the KDJ11-DB CPU Module, which is identical in functionality to the KDJ11-DA, but which has an additional 1 Mbyte of memory for a total of 1.5 Mbytes of onboard memory. The additional memory provides an increase in performance of up to 35 percent over the MicroPDP-11/53 in applications that frequently access between 0.5 Mbytes and 1.5 Mbytes of memory. MicroPDP-11/53 PLUS systems support high-performance full-height 5.25-inch storage technologies.

Offering twice the system level performance of the MicroPDP-11/23 at a comparable price, the MicroPDP-11/53 and the MicroPDP-11/53 PLUS systems provide excellent, cost-efficient solutions to low-end PDP-11 computing needs.

The MicroPDP-11/53 and MicroPDP-11/53 PLUS systems include the MicroPDP-11/73 instruction set including floating-point instructions, the Extended Instruction Set (EIS), memory management, 4-Mbyte addressing capability, 32-Kbyte bootstrap and diagnostic ROM, and two switch-selectable console serial-line units. Additional memory can be used to expand the

current 0.5 Mbytes of parity memory included on the KDJ11-DA (1.5 Mbytes in the MicroPDP-11/53 PLUS KDJ11-DB). A system can be configured with up to 4 Mbytes of memory. These memory expansions support parity generation and parity error-detection, and block-mode direct-memory access (DMA) transfers.

NOTE

The MicroPDP-11/53 and MicroPDP-11/53 PLUS processors have only one hardware interrupt level while the MicroPDP-11/73 and MicroPDP-11/83 processors have four levels. This affects only those few applications that use multilevel interrupt Q-bus options. (All Digital Q-bus options are single-level interrupt devices.)

The MicroPDP-11/53 and MicroPDP-11/53 PLUS processors offer full hardware compatibility with existing Q-bus-equipped PDP-11 processors (MicroPDP-11/73, MicroPDP-11/83) and with most PDP-11 software. The processors also support the entire range of Q-bus communications and mass-storage options.

▪ MicroPDP-11/53 and MicroPDP-11/53 PLUS Features

The MicroPDP-11/53 and MicroPDP-11/53 PLUS are based on the KDJ11-DA and KDJ11-DB 16-bit Q-bus processors. The MicroPDP-11/53 and MicroPDP-11/53 PLUS systems show all the characteristics of the PDP-11 family and its architecture.

The MicroPDP-11/53 and MicroPDP-11/53 PLUS systems are multiuser computers that can serve as hosts to personal computers or as front ends to larger hosts such as VAX systems. The MicroPDP-11/53 and MicroPDP-11/53 PLUS systems are especially wellsuited to technical users and are excellent choices for multitasking, multiuser environments as well as for low-cost, timesharing commercial applications.

Users can migrate easily to other MicroPDP-11 family member systems, to larger PDP-11 systems, or to large VAX systems. Digital provides a number of upgrade kits specifically designed to facilitate transition to higher level systems.

The MicroPDP-11 family brings together state-of-the-art hardware technology, proven PDP-11 computer architecture, and a large selection of software all in one compact, inexpensive package.

MicroPDP-11/53 and MicroPDP-11/53 PLUS features include:

- Low cost per user for high performance.
- Compact packaging in a variety of forms.

- The ability to operate in any normal business environment without special air conditioning or power requirements.
- Simple plug-in connectors for easy installation.
- Established PDP-11 architecture.
- Common hardware with other PDP-11 systems.
- Low maintenance and power costs.

KDJ11-DA and KDJ11-DB CPU Modules

The MicroPDP-11/53 and MicroPDP-11/53 PLUS achieve high processor speed and user access through the use of the J-11 chipset, an integral part of the KDJ11-DA and KDJ11-DB CPU modules.

These modules includes the J-11 chipset, a 32-Kbyte bootstrap/diagnostic ROM, a console serial line unit, a line frequency clock, and 512 Kbytes or 1.5 Mbytes of onboard memory. Memory is expandable in 1- and 2-Mbyte increments. The MicroPDP-11/53 and MicroPDP-11/53 PLUS can address, protect, and segment up to 4 Mbytes of memory.

The floating-point microcode, standard with the J-11 chipset, adds 46 instructions for floating-point math to enhance FORTRAN performance.

The following data shows the CPU-level performance of the MicroPDP-11/53 and the MicroPDP-11/53 PLUS relative to that of the MicroPDP-11/73. Note that the MicroPDP-11/53 PLUS will not increase performance in all cases. Specifically, applications that do not frequently access the additional Megabyte of memory available in the MicroPDP-11/53 PLUS will not show an increase in performance over the current MicroPDP- 11/53 systems.

Processor Performance Comparison

MicroPDP-11/53	MicroPDP-11/53 PLUS	MicroPDP-11/73
0.70	0.70–0.90	1.0

Specific features of the KDJ11-DA and KDJ11-DB system modules include:

- J-11 (DCJ11) microprocessor chipset.
- 15-MHz clock.
- Addressing capability up to 4 Mbytes
- Onboard parity memory.
- Complete MicroPDP-11/73 instruction set, including floating-point instructions and the Extended Instruction Set (EIS).

- Q22-bus interface that supports block-mode DMA and up to four Mbytes of physical memory.
- Line-frequency clock.
- One level of hardware interrupt.
- Console emulator in microcode.
- 32-Kbyte read-only memory (ROM) for bootstraps and diagnostics.

MSV11-Q Memory Module

The MSV11-Q memory is a 1-Mbyte quad-height 64-Kbyte MOS memory module that is standard with the MicroPDP-11/53 and MicroPDP-11/53 PLUS systems.

▪ MicroPDP-11/53 and MicroPDP-11/53 PLUS Configurations

The low-cost MicroPDP-11/53 and MicroPDP-11/53 PLUS are available in the economical BA23 pedestal, tabletop, and rackmount enclosures.

Standard Systems

For ordering convenience, both MicroPDP-11/53 and MicroPDP-11/53 PLUS systems are available in two BA23-based standard systems. Each standard system is offered in both the BA23 pedestal and rackmount variations.

Diagnostics and documentation are included in some standard systems, but can be ordered separately as well.

Enclosures

The BA23A enclosure for the MicroPDP-11/53 and MicroPDP-11/53 PLUS systems is offered in either a pedestal/tabletop model or a rackmount model. The enclosure features an 8-slot backplane with two full-height 5.25-inch mass-storage cavities. Therefore, the BA23-based MicroPDP-11/23 and MicroPDP-11/23 PLUS systems can support up to 26 I/O devices using a cable concentrator.

The BA23 pedestal/rackmount enclosure can contain up to two RX33 diskette drives or one TK50, and up to two RD31/RD32 fixed-disk drives or one RD53.

Mass Storage

MicroPDP-11/53 and MicroPDP-11/53 PLUS standard systems include the following mass storage options:

- One integrated RX33 1.2-Mbyte half-height diskette drive that reads and writes both RX50 and industry standard media (MicroPDP-11/53 only).
- Optional, external 42-, 71- (MicroPDP-11/53 PLUS), or 159-Mbyte fixed disk.
- TK50 95-Mbyte tape drive (MicroPDP-11/53 PLUS only).
- Communications options.
- Disk/diskette controller using Mass Storage Control Protocol.

Additional Information

For information regarding MicroPDP-11/53 and MicroPDP-11/53 PLUS specifications such as backplane assemblies, power supplies, control panels, I/O distribution panels, fans, size and weight, power requirements, and operating environment, refer to Chapter 2.

Chapter 5 ▪ Q-bus

▪ Introduction

The Q-bus is the common communications path for the data, address, and control information that is transferred between the CPU, memory, and device interfaces. Each of the supermicrocomputers use only the Q-bus for these communications.

The 22-bit Q-bus consists of 42 bidirectional and two unidirectional signal lines that are built into the backplane assembly. Logic modules are installed in the backplane and connected to these signal lines with backplane connectors. The signal lines are defined as follows:

- Sixteen multiplexed data/address lines (BDAL<15:00>).
- Two multiplexed address/parity lines (BDAL<17:16>).
- Four nonmultiplexed extended address lines (BDAL<21:18>).
- Six data transfer control lines (BBS7, BDIN, BDOUT, BRPLY, BSYNC, BWTBT).
- Six system control lines (BHALT, BREF, BEVNT, BINIT, BDCOK, BPOK).
- Ten interrupt control and direct memory access control lines (BIAKO, BIAKI, BIRQ4, BIRQ5, BIRQ6, BIRQ7, BDMGO, BDMR, BSACK, BDMGI).

In addition to the data, address, and control signal lines, a number of power, ground, and spare lines have been defined for the 22-bit Q-bus (hereafter referred to as the Q22 bus). For a detailed description of these lines, refer to Table 5-1.

▪ Bus Communications

All communications on the bus are performed asynchronously to allow some devices to transfer at data rates greater than those of other devices. The bus operates with a *master and slave relationship* . When more than one device requests the use of the bus, the device with the highest priority gains access. It becomes the bus master and controls the data transfers until it releases the bus. In performing the transfers, it addresses another device that is designated as a slave.

The current data cycle is overlapped with the arbitration for the next cycle, enhancing the system performance. The upper eight Kbytes of address space are reserved for I/O devices. Some of the addresses are fixed within this space, and others are allowed to float, depending on the system configuration.

The bus transactions consist of initialization, arbitration, data transmission, and miscellaneous:

- The *initialization* lines of the bus provide the information required to start the processor after powering up and cause an orderly shutdown of the processor during power failures. In addition, they allow the processor to reset the I/O subsystem.
- The *arbitration* lines control access to the data transmission portion of the bus.
- The *data transmission* lines allow words or bytes to be moved about on the bus. Transmission of data is always accomplished with one device acting as master and the other acting as slave. The master controls the direction and length of transmission.
- The *miscellaneous* lines provide other functions, including processor control and memory refresh.

Master/Slave Relationship

A master/slave relationship exists throughout each bus transaction. At any time, one device has control of the bus and is termed the bus master, and the other device is termed the slave. The bus master, which is typically the processor or a direct-memory access (DMA) device, initiates a bus transaction. The slave device responds by acknowledging the transaction in progress and by receiving data from, or transmitting data to, the bus master. The bus control signals transmitted or received by the bus master or the bus slave device must complete the sequence according to bus protocol.

The processor controls bus arbitration to determine which device becomes bus master at any given time. A typical example of this relationship is the processor, as master, fetching an instruction from memory, which is always a slave. Another example is a disk-drive device as master transferring data to memory as slave. Communications on the bus are interlocked so that, for certain control signals issued by the master device, there must be a response from the slave in order to complete the transfer. It is the master/slave signal protocol that precludes the need for synchronizing clock pulses.

Because bus-cycle completion by the bus master requires response from the slave device, each bus master includes a time-out error circuit that will abort the bus cycle if the slave device does not respond to the bus transaction within 10 microseconds.

Bus Signals and Pin Assignments

The connectors on the backplane assembly and the logic modules have corresponding pin and signal assignments. Table 5-1 lists the signal and pin designations for the data/address, control, power/ground, and spare lines. All Q22-bus signals are asserted low and negated high, except BPOK and BDCOK, which are asserted high. Most signals are bidirectional, and transactions on the bus are performed asynchronously so that devices can exchange data at their own rates and the same interfaces can be used for different devices. The data and address lines are time-multiplexed.

Table 5-1 • Bus Signals and Pin Assignments

Data and Address			
Signal	**Pin**	**Signal**	**Pin**
BDAL0	AU2	BDAL11	BR2
BDAL1	AV2	BDAL12	BS2
BDAL2	BE2	BDAL13	BT2
BDAL3	BF2	BDAL14	BU2
BDAL4	BH2	BDAL15	BV2
BDAL5	BJ2	BDAL16	AC1
BDAL6	BL2	BDAL17	AD1
BDAL7	BL2	BDAL18	BC1
BDAL8	BM2	BDAL19	BD1
BDAL9	BN2	BDAL20	BE1
BDAL10	BP2	BDAL21	BF1
Data Transfer Control			
Signal	**Pin**	**Signal**	**Pin**
BDOUT	AE2	BSYNC	AJ2
BRPLY	AF2	BWTBT	AK2
BDIN	AH2	BBS7	AP2

Table 5-1 ▪ Bus Signals and Pin Assignments

Interrupt Control			
Signal	**Pin**	**Signal**	**Pin**
BIRQ7	BP1	BIRQ4	AL2
BIRQ6	AB1	BIAK0	AN2
BIRQ5	AA1	BIAKI	AM2
Direct Memory Access Control			
Signal	**Pin**	**Signal**	**Pin**
BDMR	AN1	BDGMO	AS2
BSACK	BN1	BMDGI	AR2
System Control			
Signal	**Pin**	**Signal**	**Pin**
BHALT	AP1	BINIT	AT2
BREF	AR1	BDCOK	BA1
BEVNT	BR1	BPOK	BB1
Power and Ground			
Voltage	**Pin**	**Voltage**	**Pin**
+ 5B (battery)	AS1	- 12	BB2
+ 12B (battery)	AS1 (not supplied by Digital)	GND	AC2
+ 12B	BS1	GND	AJ1
+ 5B	AV1	GND	AM1
+ 5	AA2	GND	AT1
+ 5	BA2	GND	BC2
+ 5	BV1	GND	BJ1
+ 12	AD2	GND	BM1
+ 12	BD2	GND	BT1

Table 5-1 ▪ Bus Signals and Pin Assignments (Cont.)

Spares			
Signal	**Pin**	**Signal**	**Pin**
SSpare1	AE1	MSpareB	AL1
SSpare3	AH1	MSpareB	BL1
SSpare8	BH1	PSpare1	AU1
SSpare2	AF1	ASpare2	BU1
MSpareA	AK1		

▪ Data Transfer Bus Cycles

Seven types of data transfer operations can occur and are listed in Table 5-2. During the bus cycle, executed by the bus master, 8-bit bytes or 16-bit words can be written or read. In block mode, multiple words can be transferred to or from sequential word addresses, starting from a single bus address. The bus signals used for the data transfer operations are listed in Table 5-3.

Table 5-2 ▪ Data Transfer Operations

Mnemonic	Description	Bus Master Function
DATI	Data word input	Read
DATO	Data word output	Write
DATOB	Data byte output	Write byte
DATIO	Data word input/output	Read-modify-write
DATIOB	Data word input/byte output	Read-modify-write byte
DATBI	Data block input	Read block

Table 5-3 ▪ Bus Signals for Data Transfers

Mnemonic	Description	Function
BDAL<21:00> L	22 data/address lines	BDAL<15:00> L are used for word and byte transfers. BDAL<17:16> L are used for extended addressing, memory parity error (16), and memory parity error enable (17) functions. BDAL<21:18> L are used for extended addressing beyond 256 Kbytes.
BSYNC L	Bus-cycle control	Indicates bus transaction in progress.
BDIN L	Data-input indicator	Indicates bus transaction in progress.
BDIN L	Data-input indicator	Strobe signal
BRPLY L	Slave's acknowledgment of bus cycle	Strobe signal
BWTBT L	Write/byte control	Control signal
BBS7 L	I/O device select	Indicates address is in the I/O page.

Data transfer bus cycles can be reduced to five basic types—DATI, DATO(B), DATIO(B), DATBI, and DATBO. These transactions occur between the bus master and one slave device selected during the addressing portion of the bus cycle.

Bus-cycle Protocol

Before initiating a bus cycle, the previous bus transaction must have been completed (BSYNC L negated) and the device must become bus master. The bus cycle can be divided into an addressing portion and a data transfer portion. During the addressing portion, the bus master outputs the address for the desired slave device, memory location, or device register. The selected slave device responds by latching onto the address bits and holding this condition for the duration of the bus cycle until BSYNC L becomes negated. During the data transfer portion, the actual data transfer occurs.

Device Addressing

The device-addressing portion of a data transfer bus cycle comprises an address setup and deskew time and an address hold and deskew time. During the address setup and deskew time, the bus master performs the following:

- Asserts BDAL<21:00> L with the desired slave-device address bits.
- Asserts BBS7 L if a device in the I/O page is being addressed.
- Asserts BWTBT L if the cycle is a DATO(B) or DATBO.

During this time, the received address BBS7 L and BWTBT L signals are asserted at the slave-bus receiver for at least 75 nanoseconds before BSYNC goes active. Devices in the I/O page ignore the nine high-order address bits BDAL<21:13> and instead decode BBS7 L along with the 13 low-order address bits. An active BWTBT L signal during address setup time indicates that a DATO(B) or DATBO operation will follow, while an inactive BWTBT L indicates a DATI, DATBI, or DATIO(B) operation will follow.

The address hold and deskew time begins after BSYNC L is asserted. The slave device uses the active BSYNC L bus receiver output to clock BDAL address bits, BBS7 L, and BWTBT L into its internal logic. BDAL<21:00> L, BBS7 L, and BWTBT L will remain active for 25 nanoseconds minimum after BSYNC L bus receiver goes active. BSYNC L remains active for the duration of the bus cycle. Memory and peripheral devices are addressed similarly except for the way the slave device responds to BBS7 L. Addressed peripheral devices do not decode address bits on BDAL<21:13> L. Addressed peripheral devices may respond to a bus cycle when BBS7 L is asserted (low) during the addressing portion of the cycle. When asserted, BBS7 L indicates that the device address resides in the I/O page (the upper 4,000 bytes of address space). Memory devices generally do not respond to addresses in the I/O page; however, some system applications may permit memory to reside in the I/O page for use as DMA buffers, read-only memory bootstraps, or diagnostics.

DATI Bus Cycle

The DATI bus cycle, shown in Figure 5-1, is a read operation. During DATI, data is input to the bus master. Data consists of 16-bit word transfers over the bus. During the data transfer portion of the DATI bus cycle, the bus master asserts BDIN L for 100 nanoseconds minimum after BSYNC L is asserted. Figure 5-2 shows the DATI bus cycle timing. The slave device responds to BDIN L active as follows:

- Asserts BRPLY L for 0 nanoseconds minimum (8 microseconds maximum to avoid bus timeout) after receiving BDIN L for 125 nanoseconds maximum before BDAL bus driver data bits are valid.
- Asserts BDAL<17:00> L on the MicroVAX and BDAL<21:00> L on the MicroPDP-11 with the addressed data and error information for 0 nanoseconds minimum after receiving BDIN and 125 nanoseconds maximum after assertion of BRPLY.

When the bus master receives BRPLY L, it does the following:

- Waits at least 200 nanoseconds deskew time and then accepts input data at BDAL<17:00> L bus receivers. BDAL<17:16> L are used for transmitting parity errors to the master.
- Negates BDIN L for 200 nanoseconds minimum to 2 microseconds maximum after BRPLY L goes active.

The slave device responds to BDIN L negation by negating BRPLY L and removing read data from BDAL bus drivers. BRPLY L must be negated 100 nanoseconds maximum prior to removal of read data. The bus master responds to the negated BRPLY L by negating BSYNC L.

Conditions for the next BSYNC L assertion are as follows:

- BSYNC L must remain negated for 200 nanoseconds minimum.
- BSYNC L must not become asserted within 300 nanoseconds of previous BRPLY L negation.

NOTE

Continuous assertion of BSYNC L retains control of the bus by the bus master, and the previously addressed slave device remains selected. This is done for DATIO(B) bus cycles where DATO or DATOB follows a DATI without BSYNC L negation and a second device addressing operation. Also, a slow slave device can hold off data transfers to itself by keeping BRPLY L asserted, which will cause the master to keep BSYNC L asserted.

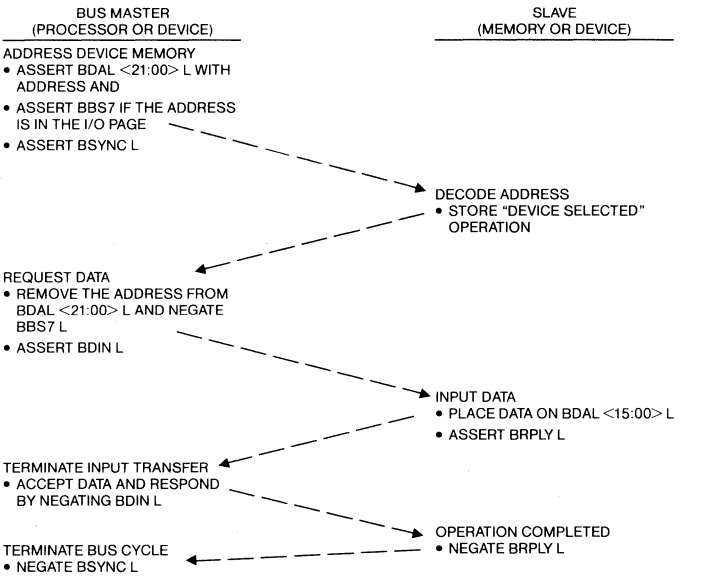
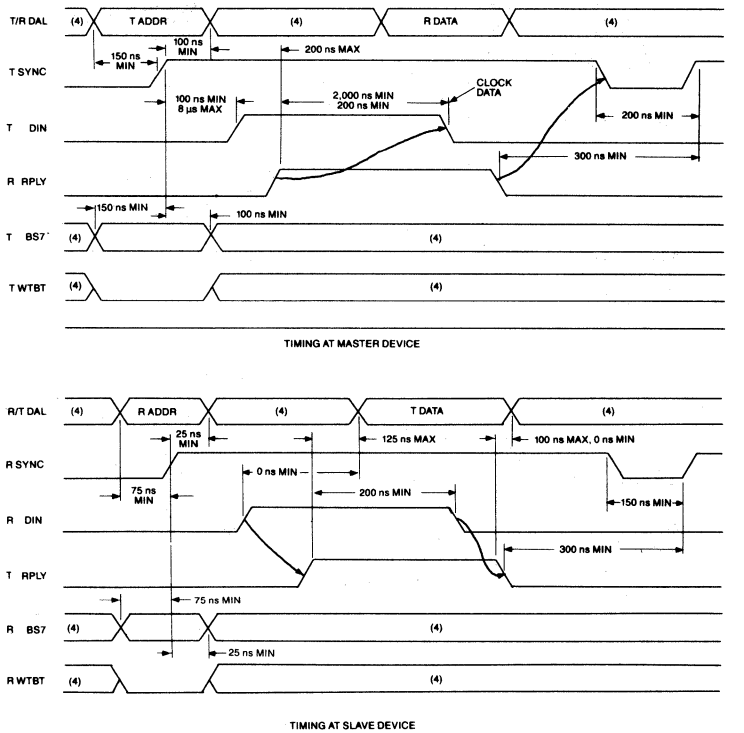


Figure 5-1 ▪ DATI Bus Cycle



NOTES

1 Timing shown at Master and Slave Device Bus Driver inputs and Bus Receiver Outputs
2 Signal nameprefixes are defined below
T = Bus Driver input
R = Bus Receiver Output
3 Bus Driver Output and Bus Receiver input signal names include a "B" prefix
4 Don't care condition

Figure 5-2 • DATI Bus-cycle Timing

DATO(B) Bus Cycle

The DATO(B) bus cycle, shown in Figure 5-3, is a write operation. Data is transferred in 16-bit words (DATO) or 8-bit bytes (DATOB) from the bus master to the slave device. The data transfer output can occur after the addressing portion of a bus cycle when BWTBT L has been asserted by the bus master, or immediately following an input transfer part of a DATIO(B) bus cycle.

The data transfer portion of a DATO(B) bus cycle comprises a data setup and deskew time and a data hold and deskew time.

During the data setup and deskew time, the bus master outputs the data on BDAL<15:00> L at least 100 nanoseconds after BSYNC L is asserted. If the transfer is a word transfer, the bus master negates BWTBT L at least 100 nanoseconds after BSYNC L assertion. BWTBT L remains negated for the length of the bus cycle. If the transfer is a byte transfer, BWTBT L remains asserted. If the transfer is the output of a DATIOB, BTWBT L becomes asserted and lasts the duration of the bus cycle.

During a byte transfer, BDAL<00> L selects the high or low byte. This occurs while in the addressing portion of the cycle. If asserted, the high byte BDAL<15:08> L is selected; otherwise, the low byte BDAL<07:00> L is selected. An asserted BDAL16 L at this time will force a parity error to be written into memory if the memory is a parity-type memory. BDAL17 L is not used for write operations. The bus master asserts BDOUT L at least 100 nanoseconds after BDAL and BWTBT L bus drivers are stable. The slave device responds by asserting BRPLY L within 10 microseconds to avoid bus timeout. This completes the data setup and deskew time.

During the data hold and deskew time the bus master receives BRPLY L and negates BDOUT L. BDOUT L must remain asserted for at least 150 nanoseconds from the receipt of BRPLY L before being negated by the bus master. BDAL<17:00> L bus drivers remain asserted for at least 100 nanoseconds after BDOUT L negation. The bus master then negates BDAL inputs.

During this time, the slave device senses BDOUT L negation. The data are accepted, and the slave device negates BRPLY L. The bus master responds by negating BSYNC L. However, the processor will not negate BSYNC L for at least 175 nanoseconds after negating BDOUT L. This completes the DATO(B) bus cycle. Before the next cycle, BSYNC L must remain unasserted for at least 200 nanoseconds. Figure 5-4 shows DATO(B) bus-cycle timing.

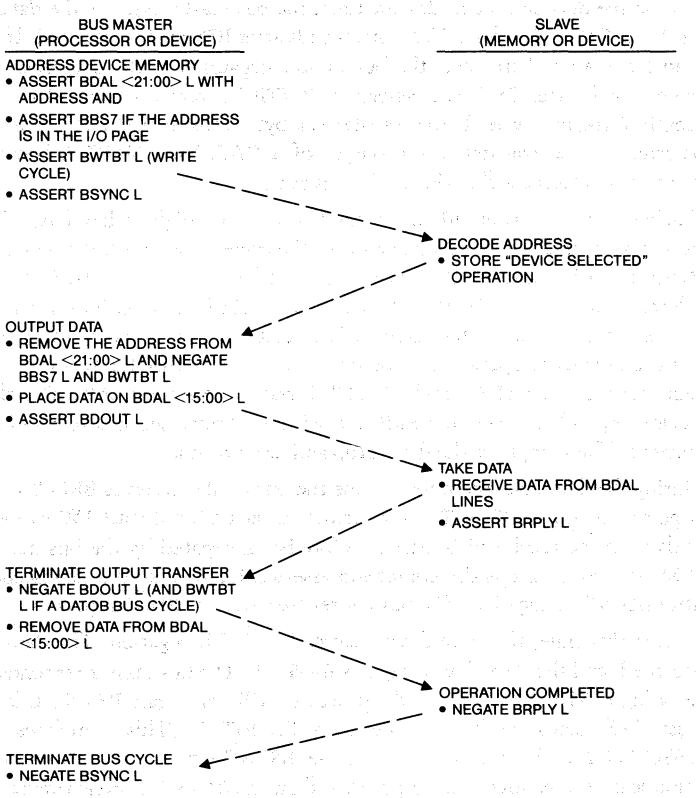
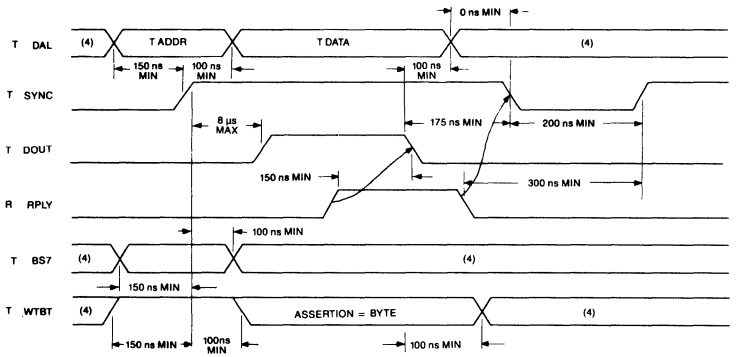
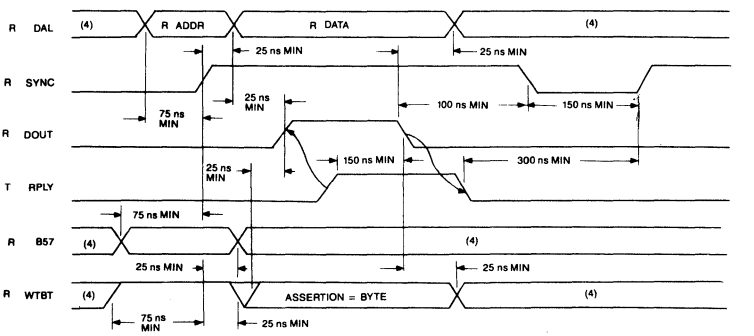


Figure 5-3 ▪ DATO or DATOB Bus Cycle



TIMING AT MASTER DEVICE



TIMING AT SLAVE DEVICE

NOTES

1 Timing shown at Master and Slave Device Bus Driver inputs and Bus Receiver Outputs
2 Signal nameprefixes are defined below
T = Bus Driver input
R = Bus Receiver Output
3 Bus Driver Output and Bus Receiver input signal names include a "B" prefix
4 Don't care condition

Figure 5-4 • DATO or DATOB Bus-cycle Timing

DATIO(B) Bus Cycle

The protocol for a DATIO(B) bus cycle is identical to the addressing and data transfer portions of the DATI and DATO(B) bus cycles; it is illustrated in Figure 5-5. After the bus addresses the device, a DATI cycle is performed; however, BSYNC L is not negated. BSYNC L remains active for an output word or byte transfer DATO(B). The bus master maintains at least 200 nanoseconds between BRPLY L negation during the DATI cycle and BDOUT L assertion. The cycle is terminated when the bus master negates BSYNC L, as described for DATO(B). Figure 5-6 shows DATIO(B) bus-cycle timing.

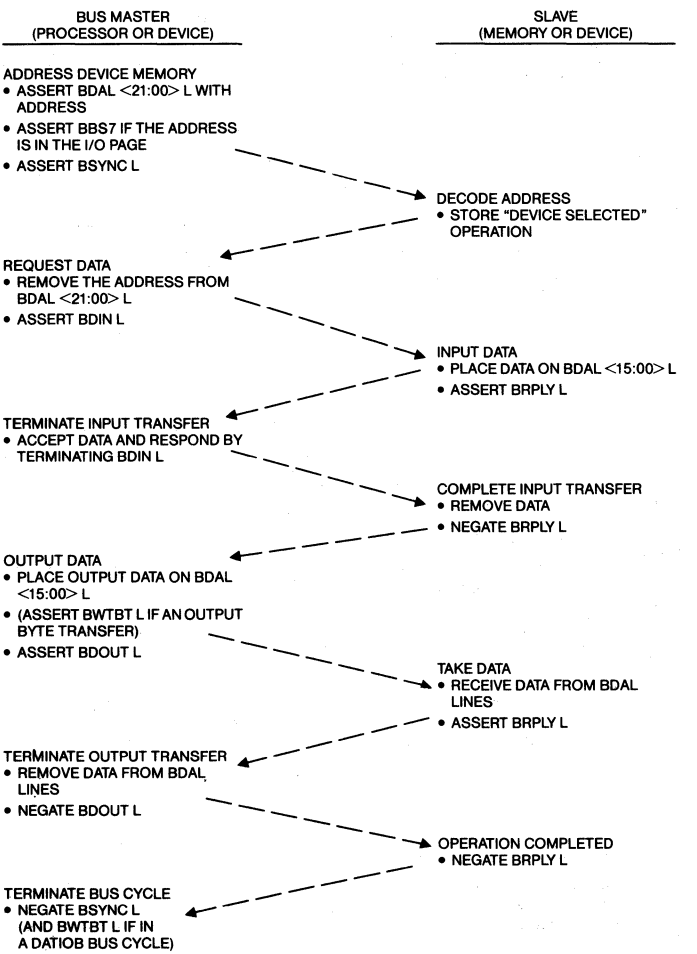
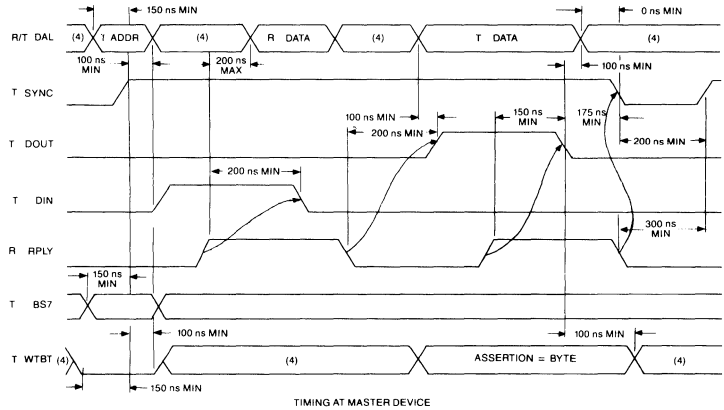
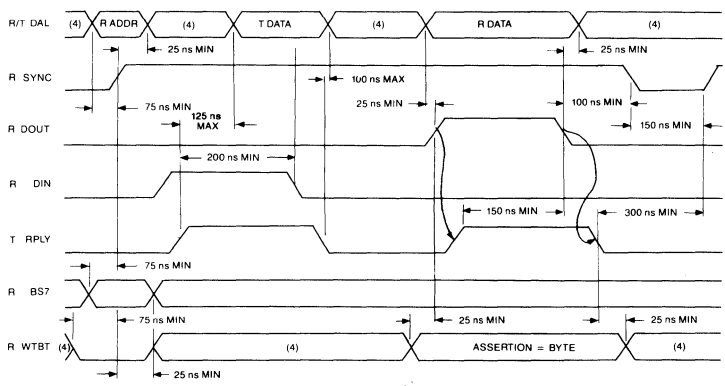


Figure 5-5 ▪ DATIO or DATIOB Bus Cycle



TIMING AT MASTER DEVICE



TIMING AT SLAVE DEVICE

NOTES

1. Timing shown at Requesting Device Bus Driver inputs and Bus Receiver Outputs.
2. Signal name prefixes are defined below:
   T = Bus Driver Input
   R = bus Receiver Output
3. Bus Driver Output and Bus Receiver input signal names include a "B" prefix
4. Don't care condition

Figure 5-6 ▪ DATIO or DATIOB Bus-cycle Timing

▪ Direct Memory Access

The direct memory access (DMA) capability allows direct data transfer between I/O devices and memory. This is useful when using mass storage devices such as disk drives that move large blocks of data to and from memory. A DMA device needs only the starting address in memory, the starting address in mass storage, the length of the transfer, and whether the operation is read or write. When this information is available, the DMA device can transfer data directly to or from memory. Because most DMA devices must perform data transfers in rapid succession or else lose data, they are provided the highest priority.

DMA transfer is accomplished after the processor (normally bus master) has passed bus mastership to the highest-priority DMA device that is requesting the bus. The processor arbitrates all requests and grants the bus to the DMA device located electrically closest to it. A DMA device remains bus master indefinitely until it relinquishes its mastership.

The following control signals are used during bus arbitration:

BDMGI L	DMA grant input
BDGMO L	DMA grant output
BDMR L	DMA
BSACK	Bus grant acknowledge

DMA Protocol

A DMA transaction can be divided into three phases:

- *Bus mastership acquisition* phase.
- *Data transfer* phase.
- *Bus mastership relinquish* phase.

During the *bus mastership acquisition* phase, a DMA device requests the bus by asserting BDMR L. The processor arbitrates the request and initiates the transfer of bus mastership by asserting BDMGO L. The maximum time between BDMR L and BDMGO L assertion is DMA latency. This time is processor-dependent. BDMGO L/BDMGI L is one signal that is daisychained through each module in the backplane. It is driven out of the processor on the BDMGO L pin, enters each module on the BDMGI L pin, and exits on the BDMGO L pin. This signal passes through the modules in descending order of priority until it is stopped by the requesting device. The requesting device blocks the output of BDMGO L and asserts BSACK L. If BDMR L is continuously asserted, the bus will be jammed.

During the *data transfer* phase, the DMA device continues asserting BSACK L. The actual data transfer is performed as described earlier.

The DMA device can assert BSYNC L for a data transfer 250 nanoseconds minimum after it receives BDMGI L and its BSYNC L bus receiver becomes negated.

During the *bus mastership relinquish* phase, the DMA device relinquishes the bus by negating BSACK L. This occurs after completing (or aborting) the last data transfer cycle (BRPLY L negated). BSACK L can be negated up to a maximum of 300 nanoseconds before negating BSYNC L. Figure 5-7 shows the DMA protocol and Figure 5-8 shows DMA request/grant timing.

NOTE

If multiple data transfers are performed during this phase, considerations must be given to the use of the bus for other system functions, such as memory refresh (if required).

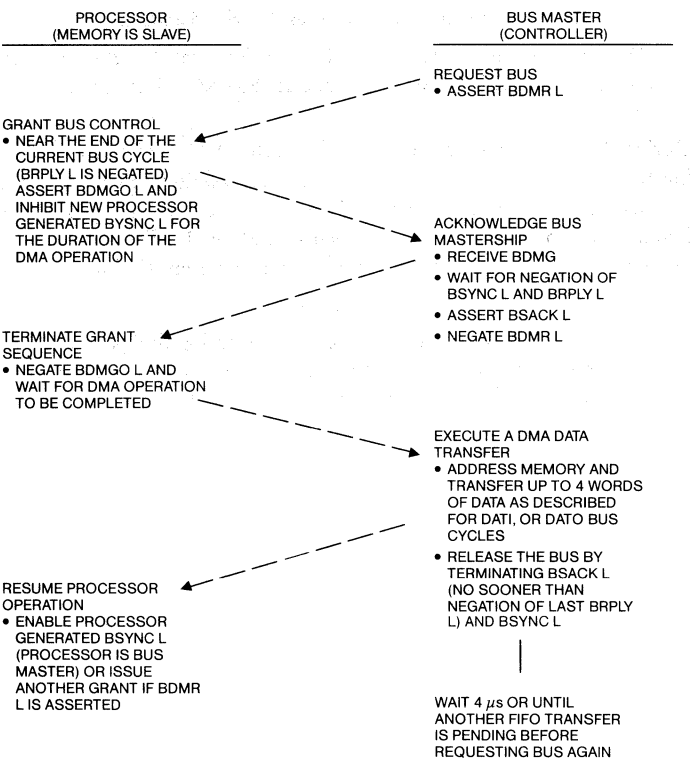
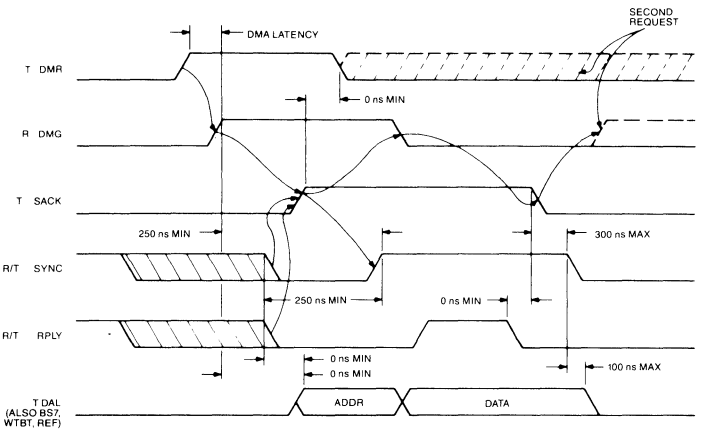


Figure 5-7 ▪ DMA Protocol



NOTES

1. Timing shown at Requesting Device Bus Driver inputs and Bus Receiver Outputs.
2. Signal name prefixes are defined below:
   T = Bus Driver Input
   R = bus Receiver Output
3. Bus Driver Output and Bus Receiver input signal names include a "B" prefix
4. Don't care condition

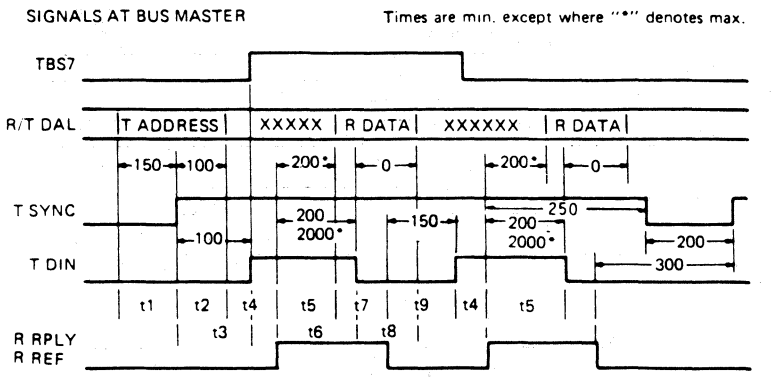
Figure 5-8 ▪ DMA Request/Grant Timing

Block Mode

For increased throughput, block mode can be implemented on a device. In a block-mode transaction, the starting memory address is asserted, followed by data for that address and data for consecutive addresses. By eliminating the assertion of the address for each data word, the transfer rate is almost doubled.

DATBI Bus Cycle

The device-addressing portion of the DATBI bus cycle is the same as previously described for other bus cycles. The bus master gates BDAL <21:00>, BBS7, and the negation of BWTBT onto the bus. Figure 5-9 shows the DATBI bus-cycle timing.



t1	=	address to T SYNC 150ns MIN.	
t2	=	address hold	100nx min
t3	=	T SYNC to T DLN	100ns min
t4	=	T DIN to R RPLY T (drive + T (prop) + T (receive) + T (delay) + T (drive) + T (receive)	
t5	=	R RPLY to data	200nx max
t6	=	R RPLY to T DIN	200nx min
t7	= =	T DIN to R RPLY T (drive) + (prop) + T (receive) + T (delay) + T (drive) + T (prop) + T (receive)	
t8	=	R RPL to data	0ns min
t9	=	R RPLY to T DIN	150ns min
T cell	=	t4 + t6 + t7 + t9	-since t must be > t5 for master to have valid data and t9 > t8

Figure 5-9 ▪ DATBI Bus-cycle Timing

The master asserts the first BDIN 100 nanoseconds after BSYNC and asserts BBS7 a maximum of 50 nanoseconds after asserting BDIN for the first time. BBS7 is a request to the slave for a block-mode transfer. BBS7 remains asserted until a maximum of 50 nanoseconds after the assertion of BDIN for the last time. BBS7 can be gated as soon as the conditions for asserting BDIN are met.

The slave asserts BRPLY a minimum of 0 nanoseconds (8 milliseconds maximum to avoid bus timeout) after receiving BDIN. It asserts BREF concurrently with BRPLY if it is a block-mode device capable of supporting another BDIN after the current one. The slave gates BDAL <15:00> onto the bus a minimum of 0 nanoseconds after the assertion of BDIN and 125 nanoseconds maximum after the assertion of BRPLY.

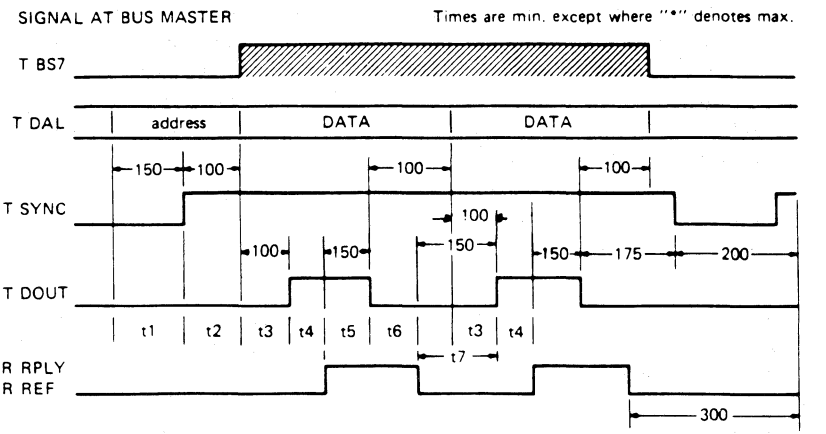
The master receives the stable data from 200 nanoseconds maximum after the assertion of BRPLY until 20 nanoseconds minimum after the negation of BDIN. It negates BDIN a minimum of 200 nanoseconds after the assertion of BRPLY.

The slave negates BRPLY a minimum of 0 nanoseconds after the negation of BDIN. If BBS7 and BREF are both asserted when BRPLY is negated, the slave prepares for another BDIN cycle. BBS7 is stable from 125 nanoseconds after BDIN is asserted until 150 nanoseconds after BRPLY is negated. The master asserts BDIN a minimum of 150 nanoseconds after BRPLY is negated, and the cycle is continued as before (BBS7 remains asserted, and the slave responds to BDIN with BRPLY and BREF). BREF is stable from 75 nanoseconds after BRPLY is asserted until a minimum of 20 nanoseconds after BDIN is negated.

If BBS7 and BREF are not both asserted when BRPLY is negated, the slave removes the data from the bus 0 nanoseconds minimum and 100 nanoseconds maximum after negating BRPLY. The master negates BSYNC a minimum of 250 nanoseconds after the assertion of the last BRPLY and a minimum of 0 nanoseconds after the negation of that BRPLY.

DATBO Bus Cycle

The device-addressing portion of the DATBO bus cycle is the same as described earlier. The bus master gates BDAL<21:00>, BBS7, and the assertion of BWTBT onto the bus. Figure 5-10 shows the DATBO bus-cycle timing.



t1	=	address to T SYNC	150nx min
t2	=	address hold	100ns min
t3	=	data to T DOUT	100ns min
t4	=	T DOUT to R RPLY	
	=	T (drive) + T (prop) + T (receive) + T (delay) + T (drive) + T (prop) + T (receive)	
t5	=	R RPLY to T DOUT	150ns min
t6	=	T DOUT to R RPLY	
	=	T (drive) + T (prop) + T (receive) + T (delay) + T (drive) + T (prop) + T (receive)	
t7	=	R RPLY to T DOUT	150ns min
T cell	=	t3 + t4 + t5 + t6 + t7	-since t3 < t7

Figure 5-10 ▪ DATBO Bus-cycle Timing

A minimum of 100 nanoseconds after BSYNC is asserted, data on BDAL<15:00> and the negated BWTBT are put onto the bus. The master then asserts BDOUT a minimum of 100 nanoseconds after gating the data.

The slave receives stable data and BWTBT from a minimum of 25 nanoseconds before the assertion of BDOUT to a minimum of 25 nanoseconds after the negation of BDOUT. The slave asserts BRPLY a minimum of 0 nanoseconds after receiving BDOUT. It also asserts BREF concurrently with BRPLY if it is a block-mode device capable of supporting another BDOUT after the current one.

The master negates BDOUT 150 nanoseconds minimum after the assertion of BRPLY. If BREF was asserted when BDOUT was negated, and if the master wants to transmit more data in this block-mode cycle, then the new data

is gated onto the bus 100 nanoseconds minimum after BDOUT is negated. BREF is stable from 75 nanoseconds maximum after BRPLY is asserted until 20 nanoseconds minimum after BDOUT is negated. The master asserts BDOUT for 100 nanoseconds minimum after gating new data onto the bus and 150 nanoseconds minimum after BRPLY negates. The cycle continues as before.

If BREF was not asserted when BDOUT was negated, or if the bus master does not want to transmit more data in this cycle, then the master removes data from the bus a minimum of 100 nanoseconds after negating BDOUT. The slave negates BRPLY a minimum of 0 nanoseconds after negating BDOUT. The bus master negates BSYNC a minimum of 175 nanoseconds after negating BDOUT and a minimum of 0 nanoseconds after the negation of BRPLY.

DMA Guidelines

- Bus masters that do not use block mode are limited to four DATI, four DATO, or two DATIO transfers per acquisition.
- Block-mode bus masters that do not monitor BDMR are limited to eight transfers per acquisition.
- If BDMR is not asserted after the seventh transfer, block-mode bus masters that do monitor BDMR can continue making transfers until the bus slave fails to assert BREF or until they reach the total maximum of 16 transfers. Otherwise, they stop after eight transfers.

Interrupts

The interrupt capability of the Q22 bus allows any I/O device to temporarily suspend (interrupt) current program execution and divert processor operation to service the requesting device. The processor inputs a vector address from the device to start the service routine (handler). Like the device-register address, the device vector is set by the hardware at locations within a designated range below location 001000 (octal).

The interrupt vector in the MicroVAX is determined by adding 200 (hex) to the vector supplied by the device and using this as a longword offset into the system-control block (SCB). This process yields the starting physical address of the Q22-bus device interrupt routine.

The interrupt vector in the MicroPDP-11 indicates the first of a pair of addresses. The content of the first address is read by the processor and is the starting address of the interrupt handler. The content of the second address is a new processor status word. The new processor status word can raise the interrupt priority level, thereby preventing lower-level interrupts from breaking into the current interrupt service routine. Control is returned to the interrupted program when the interrupt handler is ended. The original interrupted program's address and its associated processor status word are stored on a stack. The original program address and processor status word are restored by a return-from-interrupt instruction at the end of the handler. The use of the stack and the Q-bus interrupt scheme can allow interrupts to occur within nested interrupts, depending on the processor status word.

Interrupts can be caused by Q22-bus options or by the CPU. Interrupts that originate from within the processor are called *traps*. Traps are caused by programming errors, hardware errors, special instructions, and maintenance features.

The Q22-bus signals used in interrupt transactions are

BIRQ4	Interrupt request priority level 4
BIRQ5	Interrupt request priority level 5
BIRQ6 L	Interrupt request priority level 6
BIRQ7 L	Interrupt request priority level 7
BIAKI L	Interrupt acknowledge output
BIAK0 L	Interrupt acknowledge output
BDAL<21:00> L	Data/address lines
BDIN L	Data input strobe
BRPLY L	Reply

Device Priority

The MicroPDP-11 on the Q22 bus supports the following two methods of device priority. MicroVAX systems, however, use distributed arbitration only.

- Distributed Arbitration—priority levels are implemented on the hardware. When devices of equal priority level request an interrupt, priority is given to the device electrically closest to the processor.
- Position-Defined Arbitration—priority is determined solely by electrical position on the bus. The closer a device is to the processor, the higher its priority is.

Interrupt Protocol

The interrupt protocol has three phases:

- The *interrupt request* phase.
- The *interrupt acknowledge and priority arbitration* phase.
- The *interrupt vector transfer* phase.

Figure 5-11 shows the interrupt request/acknowledge sequence and Figure 5-12 shows the interrupt protocol timing.

The *interrupt request* phase begins when a device meets its specific conditions for interrupt requests, such as when the device is ready, done, or an error has occurred. The interrupt enable bit in a device status register must be set. The device then initiates the interrupt by asserting the interrupt request line(s). BIRQ4 L is the lowest hardware priority level and is asserted for all interrupt requests. The level at which a device is configured must also be asserted. A special case exists for level 7 devices which must also assert level 6. This is described in the following four-level interrupt discussion:

Interrupt Level	Lines Asserted by Device
4	BIRQ4 L
5	BIRQ4 L, BIRQ5 L
6	BIRQ4 L, BIRQ6 L
7	BIRQ4 L, BIRQ6 L, BIRQ7 L

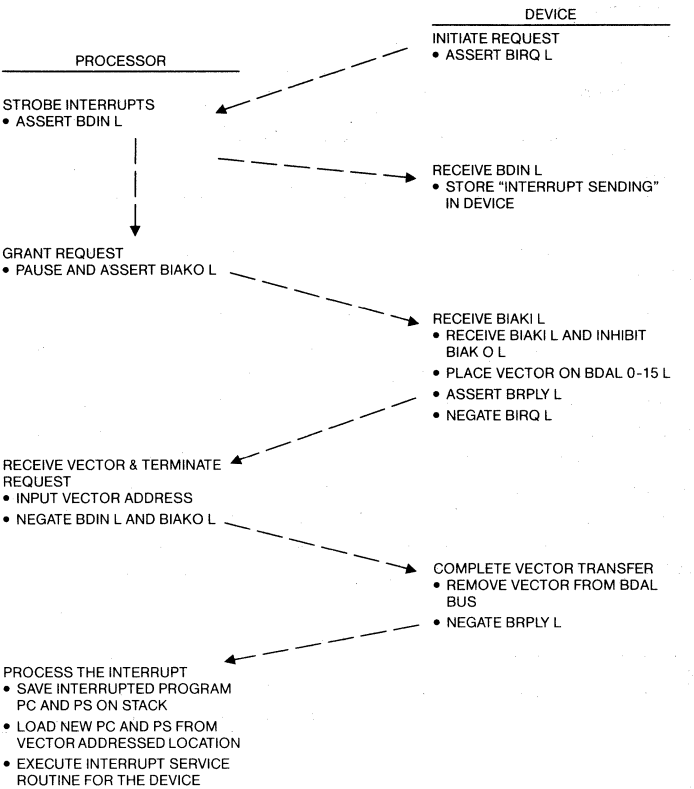
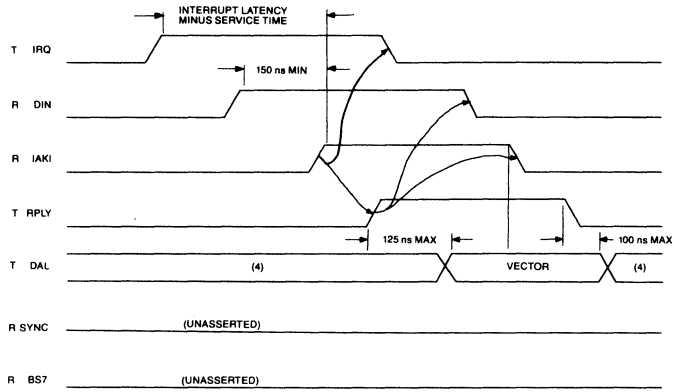


Figure 5-11 ▪ Interrupt Request/Acknowledge Sequence



NOTES
1. Timing shown at Requesting Device Bus Driver inputs and Bus Receiver Outputs.
2. Signal name prefixes are defined below:
   T = Bus Driver Input
   R = bus Receiver Output
3. Bus Driver Output and Bus Receiver input signal names include a "B" prefix
4. Don't care condition

Figure 5-12 ▪ Interrupt Protocol Timing

The interrupt request line remains asserted until the request is acknowledged.

During the *interrupt acknowledge and priority arbitration* phase, the processor will acknowledge interrupts under the following conditions:

- The device interrupt priority is higher than the current interrupt priority level.
- The processor has completed instruction execution and no additional bus cycles are pending.

The processor acknowledges the interrupt request by asserting BDIN L and, 150 nanoseconds minimum later, asserting BIAKO L. The device electrically closest to the processor receives the acknowledgment on its BIAKI L bus receiver.

The two types of arbitration are discussed separately. If the device that receives the acknowledgment uses the four-level interrupt scheme, the following occurs:

- If not requesting an interrupt, the device asserts BIAKO L, and the acknowledgment propagates to the next device on the bus.
- If the device is requesting an interrupt, it checks that no higher-level device is currently requesting an interrupt. This is done by monitoring higher-level request lines. The following table lists the lines that are monitored by devices at each priority level.

Device Priority Level	Line(s) Monitored
4	BIRQ5, BIRQ6
5	BIRQ6
6	BIRQ7
7	–

In addition to asserting levels 7 and 4, level 7 devices must assert level 6. This is done to simplify the monitoring and arbitration by level 4 and 5 devices. In this protocol, level 4 and 5 devices need not monitor level 7 since level 7 devices assert level 6. Level 4 and 5 devices will become aware of a level 7 request because they monitor the level 6 request. This protocol has been optimized for level 4, 5, and 6 devices, because level 7 devices seldom are used.

- If no higher-level device is requesting an interrupt, the acknowledgment is blocked by the device (BIAKO L is not asserted). Arbitration logic within the device uses the leading edge of BDIN L to clock a flip-flop that blocks BIAKO L.
- If a higher-level request line is active, the device disqualifies itself and asserts BIAKO L to propagate the acknowledgment to the next device along the bus.

Signal timing must be considered when implementing four-level interrupts. Note Figure 5-12.

If a single-level interrupt device receives the acknowledgment, it reacts as follows:

- If not requesting an interrupt, the acknowledgment is blocked using the leading edge of BDIN L, and arbitration is won. The interrupt vector transfer phase begins.
- If the device was requesting an interrupt, the acknowledgment is blocked using the leading edge of BDIN L, and arbitration is won. The interrupt vector transfer phase begins.

The *interrupt vector transfer* phase is enabled by BDIN L and BIAKI L. The device responds by asserting BRPLY L and its BDAL<15:00> L bus driver inputs with the vector address bits. The BDAL bus-driver inputs must be stable within 125 nanoseconds maximum after BRPLY L is asserted. The processor then inputs the vector address and negates BDIN L and BIAKO L. The device then negates BRPLY L and, 100 nanoseconds maximum later, removes the vector address bits. The processor then enters the device's service routine.

NOTE

Propagation delay from BIAKI L to BIAKO L must not be greater than 500 nanoseconds per Q-bus slot. The device must assert BRPLY L within 10 microseconds maximum after the processor asserts BIAKI L.

Four-level Interrupt Configurations

High-speed peripheral devices can attain better software performance using the four-level interrupt scheme. Both position-independent and position-dependent configurations can be used.

The position-independent configuration is shown in Figure 5-13. This allows peripheral devices that use the four-level interrupt scheme to be placed in the backplane in any order. These devices must send out interrupt requests and monitor higher-level request lines, as described. The level 4 request is always asserted by a requesting device regardless of priority. If two or more devices of equally high priority request an interrupt, the device physically closest to the processor will win arbitration. To function properly, devices that use the single-level interrupt scheme must be modified or placed at the end of the bus for arbitration.

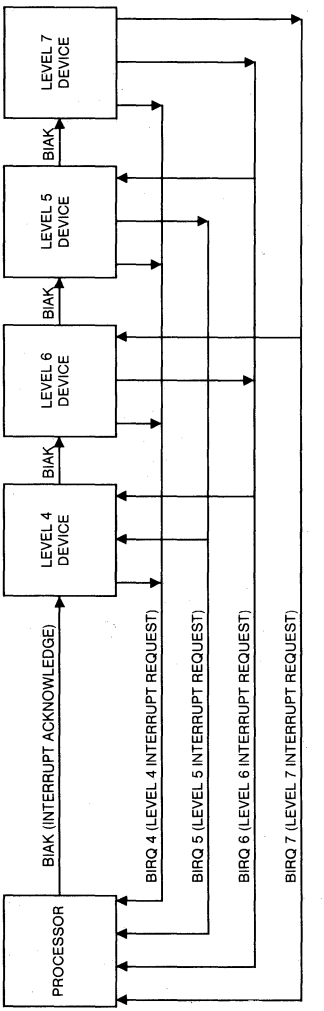


Figure 5-13 ▪ Position-independent Configuration

The position-dependent configuration, shown in Figure 5-14, is simpler to implement. A constraint is that peripheral devices must be inserted with the highest-priority device located closest to the processor and the remaining devices placed in the backplane in decreasing order of priority, with the lowest-priority devices farthest from the processor. With this configuration, each device has to assert only its own level and level 4. Monitoring higher-level request lines is unnecessary. Arbitration is achieved through the physical positioning of each device on the bus. Single-level interrupt devices on level 4 should be positioned last on the bus.

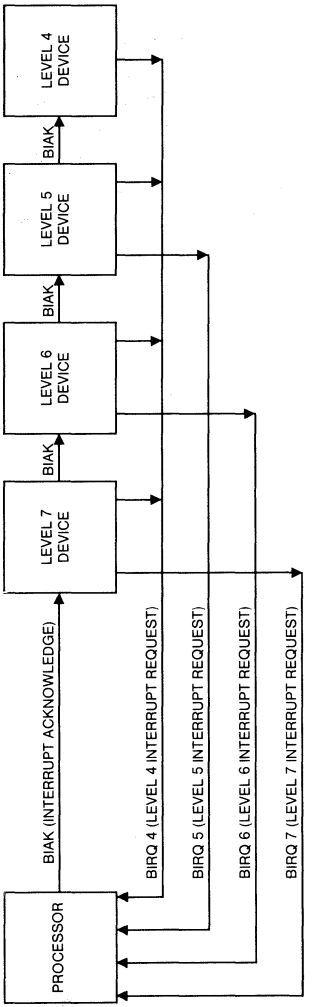


Figure 5-14 ▪ Position-dependent Configuration

▪ Control Functions

The following Q22-bus signals provide control functions:

BREF L	Memory refresh also (also block mode)
BHALT L	Processor halt
BINIT L	Initialize
BPOK H	Power OK
BDCOK H	dc power OK

BREF

The BREF signal is used as a control signal issued by memory devices for block mode transfers.

Halt

Assertion of BHALT L for at least 25 microseconds interrupts the processor, which stops program execution and forces the processor unconditionally into console mode.

Initialization

Devices along the bus are initialized when BINIT L is asserted. The processor asserts BINIT L as a result of executing a powerup or powerdown sequence or after detection of a G character in console mode (MicroPDP-11) or an S character (MicroVAX).

Power Status

Power-status protocol is controlled by two signals, BDCOK H and BPOK H. These signals are driven by some external device (usually the power supply).

- BDCOK H—When asserted, this signal indicates that dc power has been stable for at least 3 milliseconds. Once asserted, this line remains asserted until the power fails. It indicates that only 5 microseconds of dc power reserve remains.

- BPOK H—When asserted, this signal indicates that there is at least an 8-millisecond reserve of dc power and that BDCOK H has been asserted for at least 70 milliseconds. Once BPOK H has been asserted, it must remain asserted for at least 3 milliseconds. The negation of this line, the first event in the powerfail sequence, indicates that power is failing and that only 4 milliseconds of dc power reserve remain.

Powerup/Powerdown Protocol

Powerup protocol begins when the power supply applies power with BDCOK H negated. This forces the processor to assert BINIT L. When the dc voltages are stable, the power supply or other external device asserts BDCOK H. BINIT L is negated as a result of BDCOK H being asserted. The processor continues to test for BPOK H until it is asserted. The power supply asserts BPOK H 70 milliseconds minimum after BDCOK H is asserted. The processor then performs its powerup sequence. Normal power must be maintained at least 3 milliseconds before a powerdown sequence can begin.

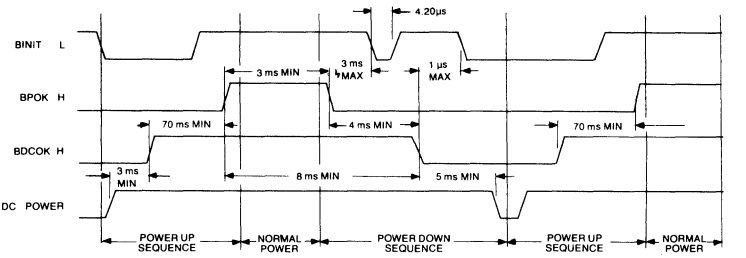
A powerdown sequence begins on MicroVAX systems when the power supply negates BPOK H. A powerfail interrupt is initiated if the current interrupt priority level is less than 1E (hex).

No later than 3 milliseconds after BPOK H negates, the processor asserts BINIT L for 8 to 20 microseconds. The power supply must negate BDCOK H no sooner than 4 milliseconds after the negation of BPOK H. This allows mass-storage devices to protect themselves against erasures and erroneous writes during a power failure.

The processor asserts BINIT L again no later than 1 microsecond after the negation of BDCOK H. The dc power must remain stable for a minimum of 5 microseconds after BDCOK H negates. BDCOK H must remain negated for a minimum of 3 milliseconds.

A powerdown sequence begins on the MicroPDP-11 when the power supply negates BPOK H. When the current instruction is completed, the processor traps to a powerdown routine at location 24_8. The end of the routine is terminated with a HALT instruction to avoid any possible memory corruption as the dc voltages decay.

When the processor executes the HALT instruction, it tests the BPOK H signal. If BPOK H is negated, the processor enters the powerup sequence. It clears internal registers, generates BINIT L, and continues to check for the assertion of BPOK H. If it is asserted and dc voltages are still stable, the processor will perform the rest of the powerup sequence. Figure 5-15 illustrates powerup/powerdown timing.



NOTE
Once a power down sequence is started, it must be completed before a power-up sequence is started.

Figure 5-15 ▪ Powerup/Powerdown Timing

▪ Bus Electrical Characteristics

The electrical specifications for the signals on the Q22 bus are as follows:

Input Logic Levels	
TTL logical low:	0.8 Vdc maximum
TTL logical high:	2.0 Vdc minimum
Output Logic Levels	
TTL logical low:	0.4 Vdc maximum
TTL logical high:	2.4 Vdc minimum

Load Definition

The ac loads are the maximum capacitance allowed per signal line to ground. A unit load is defined as 9.35 pF (picofarads) of capacitance. The dc loads are defined as maximum current allowed with a signal-line driver asserted or unasserted. A unit load is defined as 210 microamperes in the unasserted state.

120-Ohm Bus

The electrical conductors interconnecting the bus-device slots are treated as transmission lines. A uniform transmission line, terminated in its characteristic impedance, will propagate an electrical signal without reflections. Since bus drivers, receivers, and wiring connected to the bus have finite resistance and nonzero reactance, the transmission line impedance is not uniform and therefore introduces distortions into pulses propagated along it. Passive components of the bus, such as wiring, cabling, and etched signal conductors, are designed to have a nominal characteristic impedance of 120 Ω.

The maximum length of interconnecting cable excluding wiring within the backplane is limited to 4.88 meters (16 feet).

Bus Drivers

Devices driving the 120-Ω bus must have open collector outputs and meet the following specifications:

- *dc Specifications*

  - Output low voltage when sinking 70 milliamps of current: 0.7 V maximum.
  - Output high leakage current when connected to 3.8 Vdc: 25 microamperes (even if no power is applied, except for BDCOK H and BPOK H)

These specifications must be met at worst-case supply voltage, temperature, and input-signal levels.

- *ac Specifications*

  - Bus-driver output-pin capacitive load: Not to exceed 10 picofarads.
  - Propagation delay: Not to exceed 35 nanoseconds.
  - Skew (difference in propagation time between slowest and fastest gate): Not to exceed 25 nanoseconds.
  - Rise/Fall Times: Transition time (from 10–90 percent for positive transition, and from 90–10 percent for negative transition) must be no faster than 10 nanoseconds.

Bus Receivers

Devices that receive signals from the 120-Ω Q22 bus must meet the following requirements:

- *dc Specifications*

  - Input low voltage (maximum): 1.3 V.
  - Input high voltage (minimum): 1.7 V.
  - Maximum input current when connected to 3.8 Vdc: 80 microamperes (even if no power is supplied).

These specifications must be met at worst-case supply voltage, temperature, and output-signal conditions.

- *ac Specifications*

▪ Bus-receiver input-pin capacitance load: Not to exceed 10 picofarads.
▪ Propagation delay: Not to exceed 35 nanoseconds.
▪ Skew (difference in propagation time between slowest and fastest gate): Not to exceed 25 nanoseconds.

Bus Termination

The 120-Ω bus must be terminated at each end by an appropriate terminator, as shown in Figure 5-16. This is a voltage divider with its Thevenin equivalent equal to 120 Ω and 3.4 Vdc nominal. The MicroVAX and MicroPDP-11 processor terminations are provided by the processor and backplane.

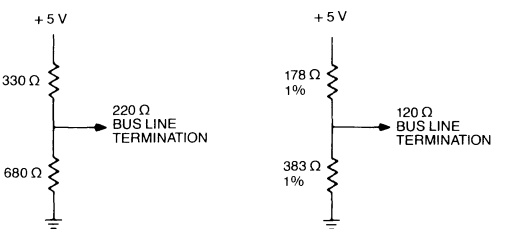


Figure 5-16 ▪ Bus Terminations

Each of the bus signals starting with the letter B must see an equivalent network with the following characteristics at each end of the bus:

Input impedance (with respect to ground)	120 ± 0 + 5%, -15%
Open circuit voltage	3.4 Vdc + 5%
Capacitance load	Not to exceed 30 pF

NOTE

The resistive termination can be provided by the combination of two modules (that is, the processor module supplies 220 Ω to ground. This, in parallel with another 220-Ω module, provides 120 Ω). Both of these terminators must be physically resident within the backplane.

▪ Bus Interconnect Wiring

The electrical characteristics of the wiring that connects the slots on the Q22-bus backplane must conform to certain requirements. These requirements ensure the proper operation of the installed modules.

Backplane Wiring

The wiring that connects all device interface slots on the backplane must meet the following specifications:

- The conductors must be arranged so that each line exhibits a characteristic impedance of 120 Ω, measured with respect to the bus common return.
- Crosstalk between any two lines must be no greater than 5 percent. Note that worst-case crosstalk is manifested by simultaneously driving all but one signal line and measuring the effect on the undriven line.
- The dc resistance of the signal path, as measured between the near-end terminator module and the far-end terminator module (including all intervening connectors, cables, backplane wiring, or connector module) must not exceed 2 Ω.
- The dc resistance of the common return path, as measured between the near-end terminator module and the far-end terminator module (including all intervening connectors, cables, backplane wiring, or connector module) must not exceed an equivalent of 2 Ω per signal path. Thus the dc resistance of the composite-signal return path must not exceed 2 Ω divided by 40 bus lines (or 50 milliohms). Although this common return path is nominally at ground potential, the conductance must be part of the bus wiring. The specified low-impedance return path must be provided by the bus wiring as distinguished from the common system or power ground path.

Intrabackplane Wiring

The wiring that connects the bus connector slots within one contiguous backplane is part of the overall bus transmission line. Because of implementation constraints, the nominal characteristic impedance of 120 Ω may not be achievable. Distributed wiring capacitance in excess of the amount required to achieve the nominal 120-Ω impedance may not exceed 60 picofarads per signal line.

Power and Ground

Each bus interface slot has connector pins assigned for the following dc voltages. The maximum allowable current per pin is 1.5 amperes. + 5 Vdc must

be regulated to ± 5 percent with a maximum ripple of 100 mV pp. + 12 Vdc must be regulated to ± 3 percent with a maximum ripple of 200 mV pp.

- + 5 Vdc—Three pins (4.5 amperes maximum per bus device slot).
- + 12 Vdc—Two pins (3.0 amperes maximum per bus device slot).
- Ground—Eight pins (shared by power return and signal return).

Bus-system Configurations

Before configuring any system, three characteristics for each module in the system must be known. These characteristics are:

- Power consumption— + 5 Vdc and + 12 Vdc current requirements.
- ac bus loading—the amount of capacitance a module presents to a bus signal line. ac loading is expressed in terms of ac loads; one ac load equals 9.35 picofarads of capacitance.
- dc bus loading—the amount of dc leakage current a module presents to a bus signal when the line is high (undriven). dc loading is expressed in terms of dc loads; one dc load equals 210 microamperes (nominal).

Power-consumption, ac-loading, and dc-loading specifications for each module are included in the *VAX Systems and Options Catalog* and the *PDP-11 Systems and Options Catalog.*

NOTE

The ac and dc loads and the power consumption of the processor module and backplane must be included in determining the total loading of a backplane.

Single-backplane Configurations

The following rules apply to the use of single-backplane systems:

- When using the MicroVAX II with 240-Ω processor termination, or the MicroVAX I with 220-Ω processor termination, the bus can accommodate modules that have up to 20 ac loads total before additional termination is required. If more than 20 ac loads are included, the other end of the bus must be terminated with 120 Ω, and then up to 35 ac loads may be present.

- When using the MicroPDP-11 systems with 120-Ω processor termination, up to 35 ac loads can be used without additional termination. If 120-Ω bus termination is added, up to 45 ac loads can be configured in the backplane.
- The bus can accommodate modules for up to 20 total dc loads.
- The bus signal lines on the backplane can be up to 35.6 centimeters (14 inches) long.

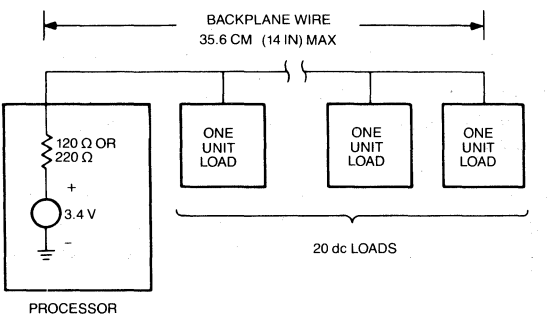


Figure 5-17 ▪ Single-backplane Configuration

Dual-backplane Configurations

The following rules apply to the use of dual backplanes in MicroVAX and MicroPDP-11 cabinet configurations:

- As illustrated in Figure 5-18, two backplanes may make up the system.
- The signal lines on each backplane can be up to 25.4 centimeters (10 inches) long.
- Each backplane can accommodate modules that have up to a total of 22 ac loads. Unused ac loads from one backplane may not be added to the other backplane if the second backplane loading will exceed 22 ac loads. It is desirable to load backplanes equally, or with the highest ac load in the first backplane.
- dc loading of all modules in both backplanes cannot exceed 20 loads total.
- Both ends of the bus must be terminated with 240 Ω on the MicroVAX II, 220 Ω on the MicroVAX I, and with 120 Ω on the MicroPDP-11.

- The cable connecting the two backplanes must be 61 centimeters (2 feet) or greater in length.
- The cable used must have a characteristic impedance of 120 Ω.

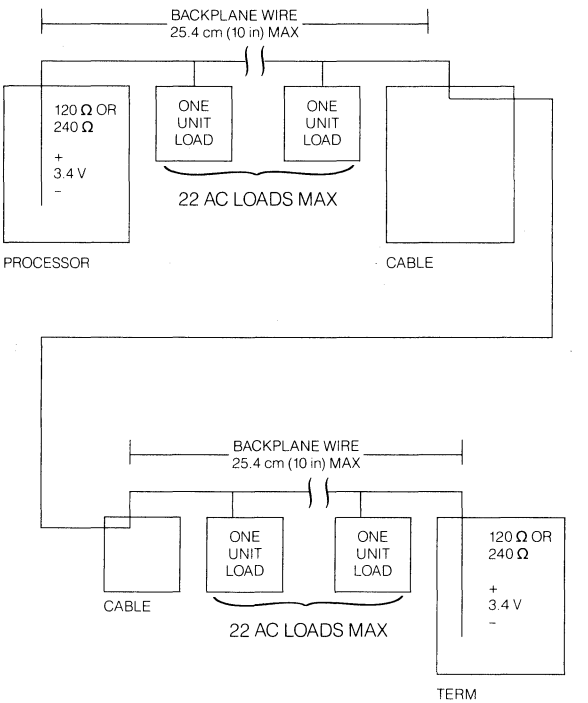


Figure 5-18 • Dual-backplane Configuration

Power Supply Loading

Total power requirements for each backplane can be determined by obtaining the total power requirements for each module in the backplane. Obtain separate totals for + 5 V and + 12 V power. Power requirements for each module are specified in the *VAX Systems and Options Catalog* and *PDP-11 Systems and Options Catalog.*

Module Connector Pin Identification

The supermicrosystems use both dual-height and quad-height modules. The convention used to identify the pin connectors on the modules and backplane is shown in Figures 5-19 and 5-20. A dual-height module has two rows of connector pins, A and B, as shown in Figure 5-19. A quad-height module has four rows labeled A, B, C, and D, which are shown in Figure 5-20.

Each row has two sides. The side of the module where the components are located is designated as Side 1. The opposite, or soldered side, is designated as Side 2. Each row on each side consists of 18 contacts, for a total of 36 contacts per row. The contact designations are A through V, excluding G, I, O, and Q. The positioning notch between the two rows of pins mates with a protrusion on the connector block for correct module positioning. Examples of typical pin identifications on a module are shown in Figures 5-19 and 5-20.

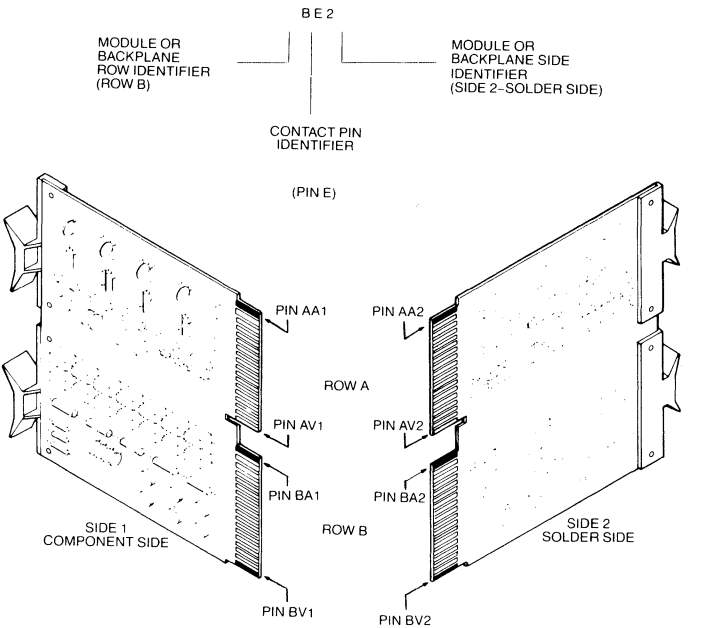


Figure 5-19 ▪ Dual-height Module Contact Finger Identification

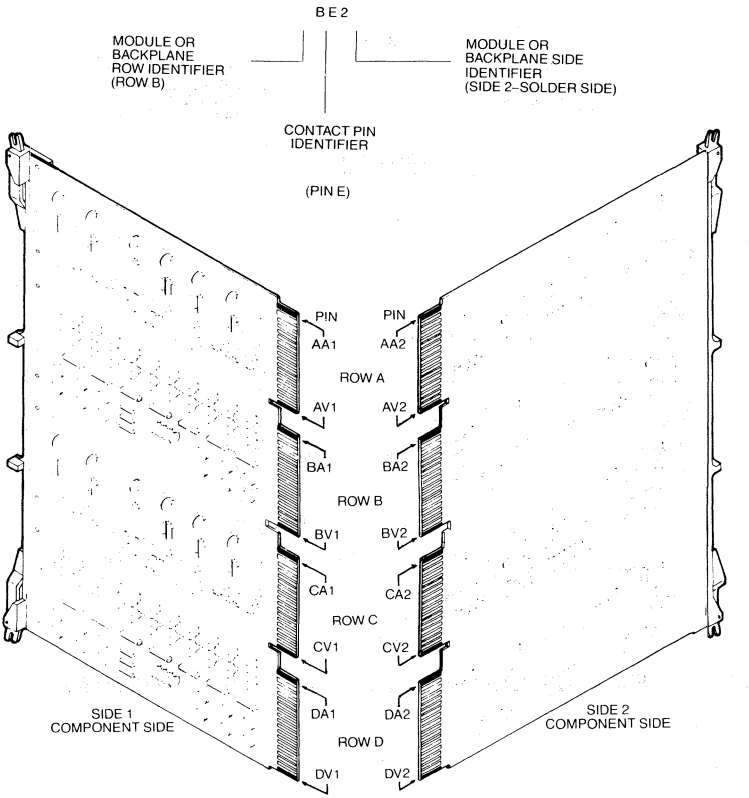


Figure 5-20 ▪ Quad-height Module Contact Finger Identification

Table 5-4 ▪ Bus-pin Identifiers

Bus Pin	Mnemonics	Description
AA1	BIRQ5 L	Interrupt Request Priority Level 5
AB1	BIRQ6 L	Interrupt Request Priority Level 6
AC1	BDAL16 L	Extended-address bit during addressing protocol; memory-error data line during data transfer protocol.
AD1	BDAL17 L	Extended-address bit during addressing protocol; memory-error logic enable during data transfer protocol.
AE1	SSPARE1 (Alternate + 5B)	Special Spare—Not assigned or bused in Digital's cable or backplane assemblies; available for user connection. Optionally, this pin can be used for + 5 V battery (+ 5B) backup power to keep critical circuits alive during power failures. A jumper is required on some bus options to disconnect the + 5B circuit in systems that use this line as SSPARE1.
AF1	SSPARE2	Special Spare—Not assigned or bused in Digital's cable or backplane assemblies; available for user interconnection. In the highest-priority device slot, the processor can use this pin for a signal to indicate its RUN state.
AH1	SSPARE3 (SRUN simultaneously)	Special Spare—Not assigned or bused in Digital's cable or backplane assemblies; available for user interconnection. An alternate SRUN signal may be connected in the highest-priority set.
AJ1	GND	Ground—System signal ground and dc return.
AK1	MSPAREA	Maintenance Spare—Normally connected together on the backplane at each option location (not a bused connection).

Table 5-4 ▪ Bus-pin Identifiers (Cont.)

Bus Pin	Mnemonics	Description
AL1	MSPAREB	Maintenance Spare—Normally connected together on the backplane at each option location (not a bused connection).
AM1	GND	Ground—System signal ground and dc return.
AN1	BDMR L	Direct Memory Access (DMA) Request—A device asserts this signal to request bus mastership. The processor arbitrates bus mastership between itself and all DMA devices on the bus. If the processor is not bus master (it has completed a bus cycle and BSYNC L is not being asserted by the processor), it grants bus mastership to the requesting device by asserting BDMGO L. The device responds by negating BDMR L and asserting BSACK L.
AP1	BHALT L	Processor Halt—When BHALT L is asserted for at least 25 microseconds, the processor responds by halting normal program execution and entering console mode.
AR1	BREF L	Memory Refresh—Asserted by a DMA device. This signal forces all dynamic MOS memory units requiring bus refresh signals to be activated for each BSYNC L/BDIN L bus transaction. It is also used as a control signal for block mode. CAUTION The user must avoid multiple DMA data transers (burst or "hog" mode) that could delay refresh operation if using DMA refresh. Complete refresh cycles must occur once every 1.6 milliseconds if required.

Table 5-4 ▪ Bus-pin Identifiers (Cont.)

Bus Pin	Mnemonics	Description
AS1	+ 12B or + 5B	+ 12 V dc or + 5 V battery backup power to keep critical circuits alive during power failures. This signal is not bused to BS1 in all of Digital's backplanes. A jumper is required on all bus options to disconnect the backup circuit from the bus in systems that use this line at the alternate voltage.
AT1	GND	Ground—System signal ground and dc return.
AU1	PSPARE1	Spare (not assigned; customer usage not recommended). Prevents damage when modules are inserted upside down.
AV1	+ 5B	+ 5 V Battery Power—Secondary + 5 V power connection. Battery power can be used with certain devices.
BA1	BDCOK H	dc Power OK—Power supply-generated signal that is asserted when there is sufficient dc voltage available to sustain reliable system operation.
BB1	BPOK H	Power OK—Asserted by the power supply 70 milliseconds after BDCOK is negated (when ac power drops below the value required to sustain power; approximately 75 percent of nominal). When negated during processor operation, a powerfail trap sequence is initiated.
BC1	SSPARE4 BDAL 18L (22-bit only)	Special Spare—Not assigned. Bused in cable and backplane assemblies; available for user interconnection.
BD1	SSPARE5 BDAL 19L (22-bit only)	Special Spare—Caution—these pins can be used as test points in some options.

Table 5-4 ▪ Bus-pin Identifiers (Cont.)

Bus Pin	Mnemonics	Description
BE1	SSPARE6 BDAL 20L	These bused address lines are address lines <21:18> and are used only during the address portion of the bus operation.
BF1	SSPARE7 BDAL 21L	These bused address lines are address lines <21:18> and are only used during the address portion of the bus operation.
BH1	SSPARE8	Special Spare—Not assigned or bused in Digital's cable and backplane assemblies; available for user interconnection.
BJ1	GND	Ground—System signal ground and dc return.
BK1 BL1	MSPAREB MSPAREB	Maintenance Spare—Normally connected together on the backplane at each option location (not a bused connection).
BM1	GND	Ground—System signal ground and dc return.
BN1	BSACK L	This signal is asserted by a DMA device in response to the processor's BDMGO L signal, indicating that the DMA device is bus master.
BP1	BIRQ7 L	Interrupt Request Priority Level 7
BR1	BEVNT L	External Event Interrupt Request—When asserted, the processor responds by entering a service routine via vector address 1008. A typical use of this signal is a line time-clock interrupt.
BS1	+ 12B	+ 12 V dc battery-backup power (not bused to AS1 in all of Digital's backplanes). Not supplied by Digital.
BT1	GND	Ground—System signal ground and dc return.

Table 5-4 ▪ Bus-pin Identifiers (Cont.)

Bus Pin	Mnemonics	Description
BU1	PSPARE2	Power Spare 2—Not assigned a function. Not recommended for use. If a module is using – 12 V (on pin AB2), and if the module is accidentally inserted upside down in the backplane, – 12 Vdc appears on pin BU1.
BV1	+ 5	+ 5 V Power—Normal + 5 Vdc system power.
AA2	+ 5	+ 5 V Power—Normal + 5 Vdc system power.
AB2	– 12	– 12 V Power— – 12 Vdc (optional) power for devices requiring this voltage. NOTE Modules that require negative voltages contain an inverter circuit on each module that generates the required voltages(s). The – 12 V power is not required with Digital-supplied options.
AC2	GND	Ground—System signal ground and dc return.
AD2	+ 12	+ 12 V Power— + 12 Vdc system power.
AE2	BDOUT L	Data Output—BDOUT, when asserted, implies that valid data is available on BDAL<0:15> L and that an output transfer, with respect to the bus-master device, is taking place. BDOUT L is deskewed with respect to data on the bus. The slave device responding to the BDOUT L signal must assert BRPLY L to complete the transfer.

Table 5-4 ▪ Bus-pin Identifiers (Cont.)

Bus Pin	Mnemonics	Description
AF2	BRPLY L	Reply—BRPLY L is asserted in response to BDIN L OR BDOUT L and during IAK transactions. It is generated by a slave device to indicate that it has placed its data on the BDAL bus or that it has accepted output data from the bus.
AF2	BDIN L	Data Input—BDIN L is used for two types of bus operations: When asserted during BSYNC L time, BDIN L implies an input transfer with respect to the current bus master, and requires a response (BRPLY L). BDIN L is asserted when the master device is ready to accept data from a slave device.
AH2	BDIN L	Data Input—BDIN L is used for two types of bus operations: When asserted during BSYNC L time, BDIN L implies an input transfer with respect to the current bus master and requires a response (BRPLY L). BDIN L is asserted when the master device is ready to accept data from a slave device. When asserted without BSYNC L, it indicates that an interrupt operation is occurring. The master device must deskew input data from BRPLY L.
AJ2	BSYNC L	Synchronize—BSYNC L is asserted by the bus-master device to indicate that it has placed an address on BDAL<0:17> L. The transfer is in process until BSYNC L is negated.

Table 5-4 • Bus-pin Identifiers (Cont.)

Bus Pin	Mnemonics	Description
AK2	BWTBT L	Write/Byte—BWTBT L is used in two ways to control a bus cycle: It is asserted at the leading edge of BSYNC L to indicate that an output sequence is to follow (DATO or DATOB), rather than an input sequence. It is asserted during BDOUT L, in a DATOB bus cycle, for byte addressing.
AL2	BIRQ4 L	Interrupt Request Priority Level 4
AM2 AN2	BIAKI L BIAK0 L	Interrupt Acknowledge—In accordance with interrupt protocol, the processor asserts BIAK0 L to acknowledge receipt of an interrupt. The bus transmits this to BIAKI L of the device electrically closest to the processor. This device accepts the interrupt acknowledgment under two conditions: The device requested the bus by asserting BIRQXL. The device has the highest-priority interrupt request on the bus at that time. If these conditions are not met, the device asserts BIAK0 L to the next device on the bus. This process continues in a daisychain fashion until the device with the highest interrupt priority receives the interrupt acknowledge signal.

Table 5-4 ▪ Bus-pin Identifiers (Cont.)

Bus Pin	Mnemonics	Description
AP2	BBS7 L	Bank 7 Select—The bus master asserts this signal to reference the I/O page (including that portion of the I/O page reserved for nonexistent memory). The address in BDAL<0:12> L when BBS7 L is asserted is the address within the I/O page.
AR2 AS2	BDMGI L BDMGO L	Direct Memory Access Grant—The bus arbitrator asserts this signal to grant bus mastership to a requesting device, according to bus mastership protocol. The signal is passed in a daisychain from the arbitrator (as BDMGO L) through the bus to BDMGI L of the next priority device (electrically closest device on the bus). This device accepts the grant only if it requested to be bus master (by a BDMR L). If not, the device passes the grant (asserts BDMGO L) to the next device on the bus. This process continues until the requesting device acknowledges the grant. CAUTION DMA device transfers must not interfere with the memory-refresh cycle.
AT2	BINIT L	Initialize—This signal is used for system reset. All devices on the bus are to return to a known, initial state. That is, registers are reset to 0, and logic is reset to state 0. Exceptions should be completely documented in programming and engineering specifications for the device.

Table 5-4 ▪ Bus-pin Identifiers (Cont.)

Bus Pin	Mnemonics	Description
AU2 AV2	BDAL0 L BDAL1 L	Data/address Lines—These two lines are part of the 16-line data/address bus over which address and data information is communicated. Address information is first placed on the bus by the bus master device. The same device then either receives input data from, or outputs data to, the addressed slave device or memory over the same bus lines.
BA2	+ 5	+ 5 V Power—Normal + 5 Vdc system power.
BB2	− 12	− 12 V Power— − 12 Vdc (optional) power for devices requiring this voltage. Voltages normally not supplied by Digital.
BC2	GND	Ground—System signal ground and dc return.
BD2	+ 12	+ 12 V Power— + 12 V system power.
BE2 BF2 BH2 BJ2 BK2 BL2 BM2 BN2 BP2 BR2 BS2 BT2 BU2 BV2	BDAL2 L BDAL3 L BDAL4 L BDAL5 L BDAL6 L BDAL7 L BDAL8 L BDAL9 L BDAL10 L BDAL11 L BDAL12 L BDAL13 L BDAL14 L BDAL15 L	Data/address Lines—These 14 lines are part of the 16-line data/address bus previously described.

▪ Additional Documentation

VAX Systems and Options Catalog

PDP-11 Systems and Options Catalog

Chapter 6 ▪ PDP-11/84

▪ Introduction

The PDP-11/84, the UNIBUS member of the PDP-11 family, offers the highest levels of functionality and performance for a machine in its price range. The UNIBUS processor, based on the J-11 chipset and an advanced floating-point processor, surpasses the performance of the earlier PDP-11/70 at about one-third the price. Many new features such as a UNIBUS adapter module (UBA) with direct memory access (DMA) cache and a high-performance private memory interconnect (PMI) that accelerates communications between the CPU and memory are all standard on the PDP-11/84. The 8192-byte high-speed cache memory provides fast access of instructions and data and greatly enhances program execution speeds. The PDP-11/84 handles more users and more memory-resident programs while delivering maximum system throughput. Main memory can be easily expanded to 4 Mbytes. These and other features combine to make the PDP-11/84 the most powerful, yet cost-effective, UNIBUS PDP-11 system now offered.

Integral to the PDP-11/84 central processor unit are hardware features and expansion capabilities that are common to the PDP-11/44, PDP-11/24, PDP-11/70, and PDP-11/34A.

▪ PDP-11/84 Hardware Features

- Powerful CPU features Digital's high-performance J-11 chipset.
- PMI enhances data transfer between memory, the CPU, and the UNIBUS adapter module.
- High-speed DMA cache provides caching of data for DMA devices.
- 8-Kbyte CPU cache memory speeds program execution.
- Concurrent processing allows simultaneous execution of CPU instructions and UNIBUS I/O data transfers.
- Programmable bus mastership gives the CPU unconditional PMI bus access regardless of pending DMA I/O requests.
- System supports up to 4 Mbytes of error correcting code (ECC) MOS memory.

- Integral floating-point processor operates with 32-bit and 64-bit numbers for faster FORTRAN or BASIC execution.
- Memory management unit provides dual-register set, 22-bit addressing, separate instruction (I) and data (D) space, and three operating modes (kernel, supervisor, user).
- Console serial-line unit with eight baud rates (switch-selectable from the rear panel).
- Source control of the line-frequency clock.
- Optional battery-backup unit provides power to the memory and air-moving devices in the event of a power failure.
- Standard 32-Kbyte bootstrap and diagnostic ROM resident on the CPU module.
- Mounting space for four standard M9312-type UNIBUS bootstrap PROMs to support Digital or customer devices.
- EEROM to set system characteristics and assist user in writing bootstrap programs.

▪ System Architecture

The PDP-11/84 integrates the CPU, the UBA, and the ECC MOS memory into a general purpose system. PMI offers a high-performance communications path between these three modules. Figure 6-1 shows a system-level block diagram.

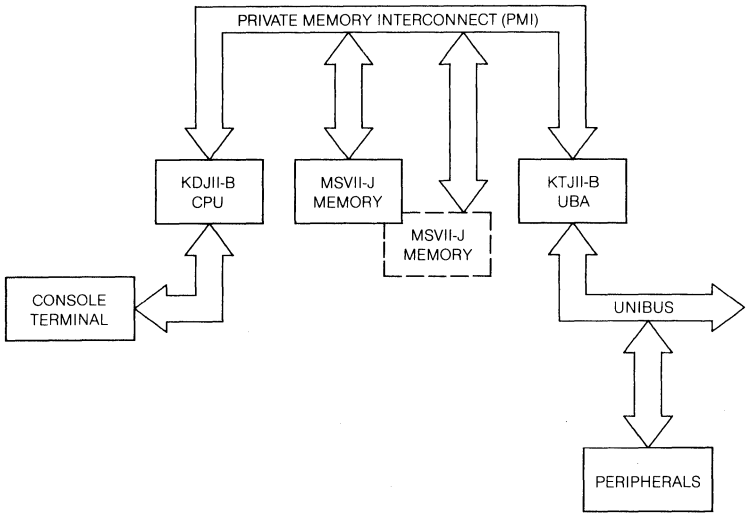


Figure 6-1 ▪ PDP-11/84 System-level Block Diagram

The CPU module consists of the J-11 chipset, floating-point processor, cache memory, a source selectable line-frequency clock, a boot and diagnostic facility, a console serial-line unit, and electrically erasable programmable ROMs (EEPROM). The J-11 microprocessor implements the PDP-11/70 instruction set, including floating point instructions and memory management. The cache is composed of 8 Kbytes of fast, static MOS memory that buffers the processor data from main memory. Dual tag cache enables the PDP-11/84 system to perform cache accesses while concurrently monitoring the cache for DMA hits. Thus, while DMA activity is transferring data on the UNIBUS, the CPU can still access cache memory without being delayed by the DMA. This simultaneous execution makes efficient use of the PMI.

The UBA contains the UNIBUS map for converting between 18-bit UNIBUS addresses and 22-bit PMI addresses, the DMA cache, four sockets for M9312-compatible boot ROMs, and the UNIBUS-to-high-speed-PMI adapter logic, which interfaces the CPU and PMI memory to the UNIBUS. A block diagram of the CPU and the UBA is shown in Figure 6-2.

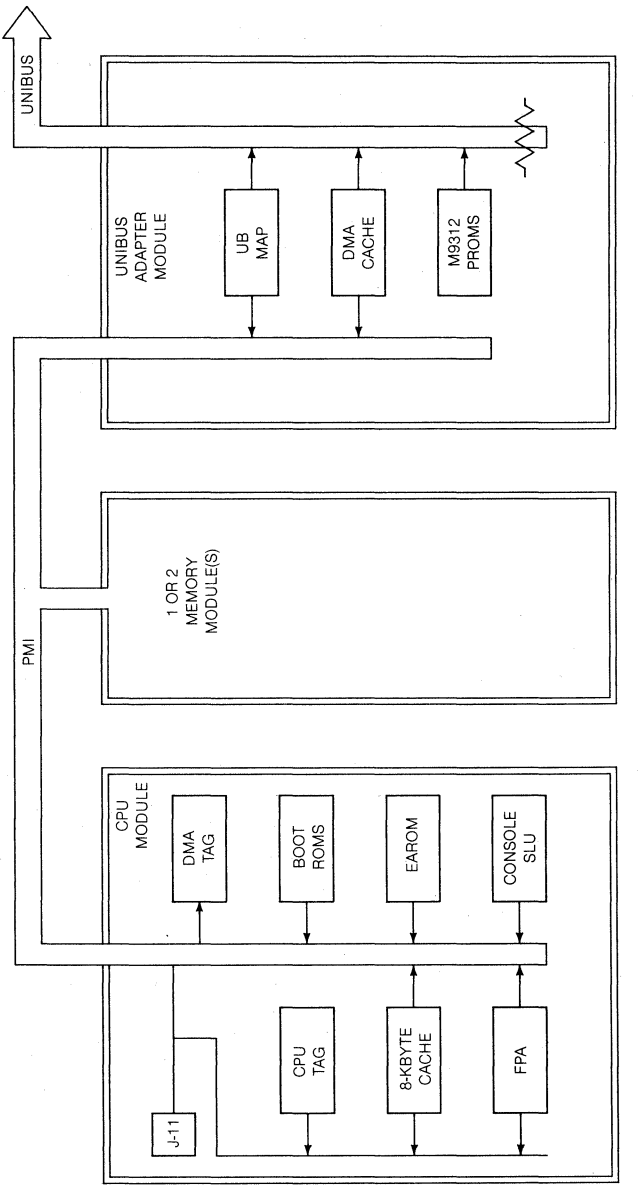


Figure 6-2 ▪ PDP-11/84 CPU Block Diagram

The UNIBUS remains the primary I/O control path in the PDP-11/84 system. It is conceptually identical to the UNIBUS of previous PDP-11 systems; the memory in the system still appears to be on the UNIBUS to all UNIBUS devices, through the UNIBUS map. There is the additional advantage of the PMI's high speed communication between the CPU and PMI memory, allowing high-speed data transfers including double-word reads. The PDP-11/84 system also features programmable bus management, where the CPU can gain bus mastership and compete with DMA devices. The PDP-11/84 can be programmed to give the CPU occasional unconditional bus priority regardless of the number of pending DMA requests.

▪ Central Processor

The PDP-11/84 CPU module includes the powerful J-11 microprocessor chipset, a floating-point processor, cache memory, a 32-Kbyte bootstrap/diagnostic ROM, a console serial-line unit, and a line-frequency clock. Standard features of the J-11 chipset include the floating-point instructions, the PDP-11 instruction set, general-purpose registers, the processor status word, traps and interrupts, memory-system registers, and the DMA mechanism.

The J-11 chipset consists of a data chip and a control chip. The data chip performs all arithmetic and logic functions, handles all data and address transfers, and generates most of the signals used for system timing. In addition to the primary execution data path, the data chip contains memory management logic, an I/O state sequencer, and floating-point and cache control registers. The control chip directs the operation of the data chip with microinstructions. The major components of the control chip are the microprogram control store and the microprogram sequencing logic.

The machine operates in three modes: kernel, supervisor, and user. When the machine is in kernel mode, a program has complete control of the machine. When the machine is in any other mode, the processor is inhibited from executing certain instructions and can deny direct access to the peripherals on the system. This hardware feature can be used to provide complete executive protection in a multiprogramming environment.

The CPU contains 12 general registers that can be used as accumulators, index registers, or stack pointers. Stacks are extremely useful for nesting programs, creating reentrant coding, and as temporary storage where a last-in/first-out structure is desirable. One additional register is used as the PDP-11/84's program counter. Three other registers are used as processor stack pointers, one for each operational mode.

The CPU performs all of the computer's computation and logic operations in a parallel binary mode through step-by-step execution of individual instructions.

A standard feature of the PDP-11/84 CPU module is the floating-point processor. This coprocessor is a 40-pin chip that significantly improves the computation speed for arithmetic applications of the CPU module.

General Registers

The general registers can be used for many purposes. Usage varies with requirements. The general registers can be used as accumulators, index registers, autoincrement registers, autodecrement registers, or as stack pointers for temporary storage of data. The *PDP-11 Architecture Handbook* describes the uses of general registers in more detail. Arithmetic operations can be performed between one general register and another, from one memory or device register to another, or between a memory or a device register and a general register. The general registers, shown in Figure 6-3, include a dual set of six registers (R0 through R5), three stack pointers corresponding to the three processor modes (R6), and a single program counter (R7).

The machine's program counter (PC) contains the address of the next instruction to be executed and thereby controls the order of execution of instructions. The PC is a general register in the sense that it is directly used by all single-operand and double-operand instructions. Much of the power of the PDP-11/84 instruction set is achieved by utilizing the PC in conjunction with various addressing modes. It is a general register normally used only for addressing purposes and not as an accumulator for arithmetic operations.

Register R6 is normally used as the processor stack pointer (SP) indicating the last entry on the current mode's hardware stack, a common temporary storage area with last-in/first-out characteristics. The three stacks are called the kernel stack, the supervisor stack, and the user stack. When the central processor is operating in kernel mode, it uses the kernel stack; in supervisor mode, the supervisor stack; and in user mode, the user stack. When an interrupt or trap occurs, the PDP-11/84 automatically switches to the mode specified by the service routine and saves its current status on that mode's processor stack. This stack-based architecture facilitates reentrant programming.

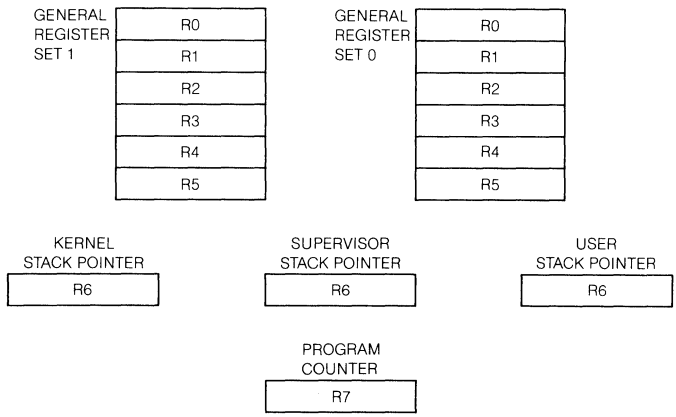


Figure 6-3 ▪ PDP-11/84 General Registers

The remaining twelve registers are divided into two sets (set 1 and set 0) of unrestricted registers. At any given time, either register set 1 or set 0 is used. The two sets cannot be used simultaneously. The current register set in operation is determined by the processor status word.

The two sets of registers can be used to increase the speed of realtime-data handling or facilitate multiprogramming. Each of the six registers in general register set 0 could each be used as an accumulator and/or index register for a realtime task or device, or as general registers for a kernel- or supervisor-mode program. The six registers in general register set 1 could be used by the remaining programs or user-mode program. The supervisor or kernel can therefore protect its general registers and stacks from user programs, or from other parts of the supervisor or kernel program.

Processor Status Word (PSW)

The processor status word contains information on the current status of the PDP-11/84. This information includes current and previous operational modes; current register set selection; current processor priority; an indicator for detecting the execution of an instruction to be trapped during program debugging; and the condition codes describing the results of the last instruction. Bit 8 is reserved for future Digital use. Bits 9 and 10 are unused and always read as zeros.

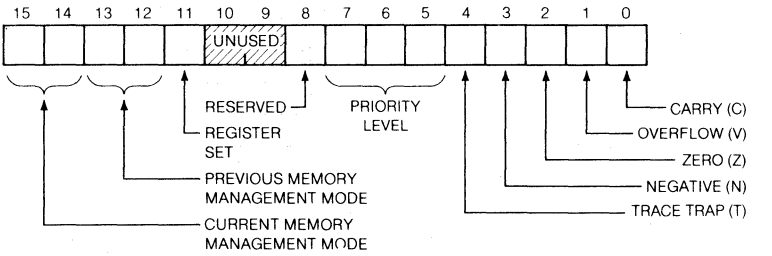


Figure 6-4 ▪ 17 777 776 Processor Status Word

▪ *Processor Memory Management Mode Fields*

Mode information includes the current mode, either kernel, supervisor, or user (bits 15, 14), and the mode the machine was in before the last interrupt or trap (bits 13, 12). The three modes permit a fully protected environment for a multiprogramming system by providing the user with three distinct sets of processor stacks and memory management registers for memory mapping.

In addition, certain PDP-11 instructions are privileged in that their operation is inhibited in supervisor and user modes. For example, in supervisor or user mode, the processor will ignore the reset and set priority level (SPL) instructions and the HALT instruction will cause a trap through the vector at virtual address 4 in kernel data space. In kernel mode, the processor will execute all instructions. A summary of the effects of processor modes on various instruction types is provided in Table 6-1.

Table 6-1 ▪ Instructions Influenced by Processor Modes

	Operation in Kernel Mode	**Operation in Supervisor/User Mode**
HALT	Depends on halt option selected	Traps through a vector at location 4 in kernel data space
WAIT, RESET, SPL	Executes as specified	Executes as a NOP
RTI, RTT, MPTS	Can alter PSW 7–5	Cannot alter PSW 7–5
Stack Reference	Checked for stack overflow	Not checked for stack overflow

- *Register Selection Field*
  This one-bit field selects which of the two general-purpose register (GPR) sets will be used. When the bit is clear (0), GPR set 0 will be used. When the bit is set, GPR set 1 will be used.

- *Processor Priority Field*
  The CPU operates at any of eight levels of priority, 0-7. When the CPU is operating at level 7, an external device cannot interrupt it with a request for service. The central processor must be operating at a lower priority than the priority of the external device's request in order for the interruption to take effect. The current priority is maintained in processor status word (bits 5-7) and is set by the software and used by the hardware to determine which interrupts will be processed. The eight processor levels provide an effective interrupt mask, which can be dynamically altered by the kernel-mode program through use of the SPL instruction. This instruction allows a kernel-mode program to alter the central processor's priority without affecting the rest of the processor status word.

- *Trace Trap Field*
  The debugging trace trap is enabled by setting bit 4 (the T bit) of the processor status word and can be set and cleared under program control. When set, a processor trap will occur through location 14 on completion of instruction execution, and a new processor status word and program counter will be loaded. This bit is especially useful for debugging programs because it provides an efficient method of single-stepping the program and is transparent to the general programmer.

  Interrupt and trap instructions both automatically cause the previous processor status word and program counter to be saved and replaced by the new values corresponding to those required by the routine servicing of the interrupt or trap. The user can thus cause the central processor to automatically switch modes (context switching), alter the CPU's priority, or disable the T bit whenever a trap or interrupt occurs.

  NOTE

  The T bit (bit 4) of the processor status word can be set indirectly by executing an RTI (return from interrupt) or RTT (return from interrupt, delaying T-bit trap) instruction with the desired processor status word on the stack. The traced instruction is the instruction after the one that set the T bit.

The following are special cases of the T bit:

- An instruction that clears the T bit—After fetching the traced instruction, an internal flag (trace flag) is set. The trap occurs at the end of this instruction's execution. The status word on the stack, however, will have a clear T bit.
- An instruction that sets the T bit—Because the T bit is already set, setting it again has no effect. The trap will occur.
- An instruction that causes an instruction trap—The instruction trap is performed and the entire routine for the service trap is executed. If the service routine exits with an RTI, or in any other way restores the stacked status word, the T bit is set again, the instruction following the traced instruction is executed, and unless it is one of the special cases noted previously, a trace trap occurs.
- An instruction that causes a stack overflow—The instruction completes execution as usual. The stack overflow does not cause a trap. The trace trap vector is loaded into the program counter and the processor status word; the old program counter and processor status word are pushed onto the stack. Stack overflow occurs again, and this time the trap is made.
- An interrupt between setting the T bit and fetching the traced instruction—The entire interrupt service routine is executed and then the T bit is set again by the exiting RTI. The traced instruction is executed (if there are no other interrupts) and, unless it is a special case noted above, a trace trap occurs.

- *Condition Codes Field*

The condition codes contain information on the result of the last CPU operation. They include a negative bit (N), set if the result of the previous operation was negative; a zero bit (Z), set if the result of the previous operation was zero; an overflow bit (V), set if the result of the previous operation resulted in an arithmetic overflow; and a carry bit ©, set if the result of the previous operation caused a carry out of the most-significant bit.

Program Interrupt Request Register (PIRQ)

The PIRQ provides seven levels of software interrupt capability (see Figure 6-5). An interrupt request is queued by setting one of bits 15-9, which corresponds to interrupt priority levels 7 through 1, respectively. Bits 7-5 and 3-1 are set by the J-11 microprocessor to the encoded value of the highest pending request. When the program interrupt request is granted, the processor traps through the vector at virtual location 240. It is the responsibility of the interrupt service routine to clear the appropriate bit in the PIRQ before the interrupt is dismissed.

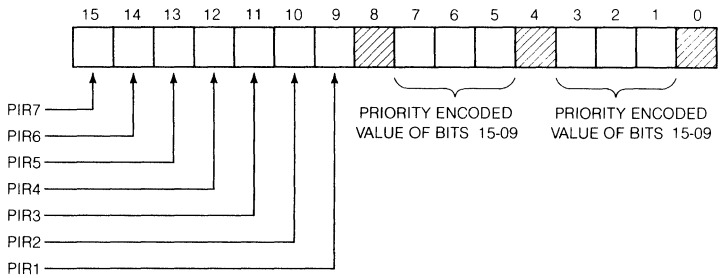


Figure 6-5 ▪ 17 777 772 Program Interrupt Request Register

Because the PIRQ interrupt can be caused by any priority between 7 and 1, the PIRQ service vector usually causes the processor to service the PIRQ interrupt at priority level 7 (this locks out further PIRQ interrupts). Bits 7-5 provide a convenient way of lowering the processor's priority to the level that actually caused the PIRQ interrupt. For example, if a level 5 PIRQ interrupt occurs, the CPU hardware would set bits 7-5 = 240. Putting this value in the PSW would place the CPU at priority level 5, blocking further level 5 interrupts.

Bits 3-1 once again provide the actual priority of the PIRQ interrupt. This time the bit field is positioned to allow its use as a word-offset (Bits 3-1 could be used with an indexed-JMP instruction to dispatch each PIRQ interrupt level to a unique address).

Bits 15-9 are read or write. Bits 7-5 and 3-1 are read-only. The remaining bits are always read as zeros. PIRQ is cleared by a console start, by a RESET instruction, or at powerup time.

Pipeline Processing

The J-11 chipset gets much of its performance from its prefetch and predecode mechanisms. The primary benefit of prefetch and predecode is that memory references are overlapped with internal operations, and the need for explicit instruction fetch and decode cycles is minimized. The prefetch and predecode operations are performed automatically by the microprocessor and cannot be altered by the user.

A primary function of the prefetch mechanism is to fill four registers with information and replenish the registers as required. These four registers, the virtual program counter (VPC), the physical program counter (PPC), the prefetch buffer (PB), and the instruction register (IR) are collectively referred to as the prefetch pipeline. The contents of registers in the beginning of the pipeline are used to determine the contents of registers farther down the pipeline. This four-stage pipeline processing enables the processor

to overlap execution when decoding, addressing, and fetching instructions. When the pipeline is filled, the prefetch mechanism is said to be in steady state.

Once the prefetch mechanism is in steady state, a stream of macroinstructions that operate only on registers may be executed at the rate of one per microcycle. While one instruction is being executed, the next one is being decoded, and the following one is being prefetched into the PB. This maintains the steady state, allowing the next macroinstruction to be executed in the next microcycle. The J-11 microprocessor bus is kept busy 100 percent of the time.

The instructions that operate on immediate data and a register also make maximum use of the prefetch mechanism. At steady state, a stream of the macroinstructions execute in two microcycles. At the same time the operation is being performed, the data in the PB is being moved to a scratch register. In both cycles, the steady state of the prefetch mechanism is maintained by prefetching the next instruction stream word.

The prefetch pipeline is refilled after a powerup sequence or if a prefetch fault occurs. Prefetch faults occur when the PSW, cache control register, PC or any other memory management registers are written. A prefetch fault invalidates only the PB. This means that the pipeline remains synchronized and can be refilled in two microcycles.

CPU Error Register

The CPU error register (Figure 6-6) assists the operating system by identifying the source of the abort that caused a trap through the vector at location 4.

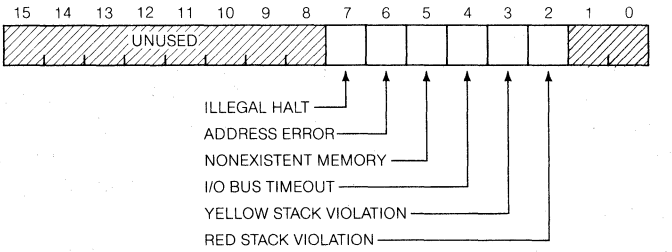


Figure 6-6 ▪ 17 777 766 CPU Error Register

Bits: 15-18

Name: Unused

Function: These bits are unused and are always read as zeros.

Bit: 7

Name: Illegal HALT (Read Only)

Function: Set when execution of a HALT instruction is attempted in user or supervisor mode, or in kernel mode when the halt-trap option is enabled.

Bit: 6

Name: Address Error (Read Only)

Function: Set when a word access is made to an odd byte address, or when an instruction fetch from an internal register is attempted.

Bit: 5

Name: Nonexistent Memory (Read Only)

Function: Set when reference is made to a nonexistent memory address.

Bit: 4

Name: I/O Bus Timeout (Read Only)

Function: Set when reference is made to a nonexistent I/O page address.

Bit: 3

Name: Yellow Stack Trap (Read Only)

Function: Set when a yellow zone stack overflow trap occurs.

Bit: 2

Name: Red Stack Trap (Read Only)

Function: Set when a red stack trap occurs.

Bits: 1-0

Name: Unused

Function: These bits are unused and are always read as zeros.

The CPU error register is cleared by any write reference to it, by a powerup, or by a console start. The RESET instruction has no effect on this register.

Stack Limit Protection

The J-11 microprocessor provides hardware protection for the kernel stack. The supervisor and user stacks are not protected by hardware, but may be checked by memory management and appropriate software.

Stack protection in kernel mode is provided by defining yellow and red stack traps. Kernel stack references are checked against a fixed limit of 400 (octal). If the virtual address of the stack reference is less than 400 (octal), a yellow stack trap occurs at the end of the current instruction. The PDP-11/84 treatment of yellow stack trap is identical to that of the PDP-11/44 treatment of yellow stack trap.

The J-11 chipset also checks for kernel stack aborts during interrupt, trap, or abort sequences. If an abort is caused by a kernel stack push during an interrupt, a trap, or an abort sequence, the J-11 initiates a red zone stack trap by setting CPU error register bit 2, loading virtual address 4 into the kernel stack pointer (R6) and trapping through location 4 in kernel data space. The J-11 microprocessor's definition of a red stack trap is unique.

Kernel Protection

To protect the kernel operating system against interference, the J-11 microprocessor incorporates a number of protection mechanisms. These include:

- Limiting execution of certain instructions.
- Limiting user-program access to data.
- Limiting user-program modification of data.
- Limiting user-program access to I/O.

- *Instruction Limitation*

As mentioned previously, certain instructions can be executed only in kernel mode. These instructions are:

- SPL—Set processor priority level.
- RESET—Initialize the I/O system.
- HALT—Halt the entire PDP-11/84.

If a user- or supervisor-mode program attempts to execute SPL or RESET, nothing happens. If a user- or supervisor-mode program attempts to execute a HALT instruction, a trap is taken. This allows the operating system to simulate halts while in a multiuser environment.

In addition, certain other instructions (such as MFPD—Move from previous data space) execute differently while in kernel mode.

- *Addressing Limitations*

The operating system can limit the amount of physical memory visible to any of the three modes of operation. Typically, only the kernel-mode program has access to the I/O page. Because the registers that control access to memory are themselves located in the I/O page, this means that only the kernel-mode program can change the access rights of other modes.

Memory is allocated by pages to each of the operating modes. Each operating mode allocates 8 pages for codes and 8 pages for data. Each page may be from zero to 8,192 bytes long, in increments of 64 bytes. Each page may be marked as read/write or read-only.

Taken together, these capabilities provide a robust protection scheme that ensures the integrity of a multiprogramming system. The hardware features are used extensively by the RSX, RSTS/E, IAS, and ULTRIX multiprogramming operating systems.

Figure 6-7 illustrates how the user's "view" of the entire system can be limited (much different from the kernel's "view"). The user has no ability to access anything outside of the user's area of memory. This gives complete protection.

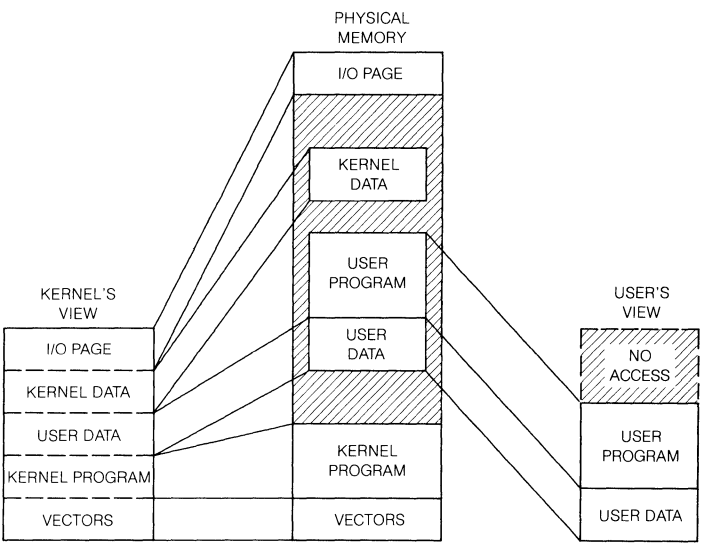


Figure 6-7 ▪ Kernel's and User's View of Physical Memory

Trap and Interrupt Service Priorities

Interrupts and traps are requests that cause the PDP-11/84 to suspend the execution of the current program and provide service for the device or condition that caused the interrupt or trap. Interrupts differ from traps in that interrupts are initiated asynchronously by some external event, while traps are caused by synchronous or asynchronous conditions internal to the PDP-11/84.

When an interrupt or trap occurs, the current PSW and PC are preserved in order to allow a return to the interrupted program. The new contents of the PC and the PSW are fetched from two consecutive memory words called a vector. The first word of the vector contains the interrupt or trap service routine starting address (the new PC), and the second word contains the new PSW. Some vectors are predefined by the PDP-11/84 while others are user defined.

The priority order for traps and interrupts is as follows (from highest to lowest priority).

1. Red stack trap
2. Address error
3. Memory management violation
4. Timeout/nonexistent memory
5. Parity error
6. Trace (T-bit) trap
7. Yellow stack trap
8. Powerfail
9. Floating-point trap
10. PIRQ 7
11. Interrupt level 7
12. Line-time clock
13. PIRQ 6
14. Interrupt level 6
15. PIRQ 5
16. Interrupt level 5
17. PIRQ 4
18. Interrupt level 4
19. PIRQ 3
20. PIRQ 2
21. PIRQ 1
22. Halt line*

*The halt line is given highest priority when the processor hangs up.

Hardware Detection Errors

The PDP-11/84 detects certain error conditions during program execution. These conditions, and the resultant actions, are described in the following sections.

- *Bus Timeout Error*

  A bus timeout error can occur during PDP-11/84 memory or UNIBUS cycles, when a slave device fails to respond. This generally occurs when the addressed memory and I/O device do not exist in the particular system. The processor recognizes the error condition and immediately traps through virtual address 4 in the kernel data space. The UNIBUS adapter module asserts the PMI timeout signal causing the CPU to abort.

- *Addressing Error*

  An addressing error occurs when an odd byte address is used with a word reference (odd addressing error), or an instruction stream fetch attempts to access an internal processor register. The internal processor registers include PARs, PDRs, CPU error, processor status word, program interrupt request, MMR0-MMR3, hit/miss, and cache control. When an addressing error happens, it sets bit 6 of the CPU error register and traps through virtual address 4 of the kernel data space.

- *Red Stack Aborts*

  A red stack abort occurs if, during the servicing of an abort, interrupt, or trap routine, an abort occurs while pushing the processor status word or program counter onto the kernel stack. This type of abort sets bit 2 of the CPU error register, loads the kernel stack pointer (R6) with virtual address 4, and then traps through location 4 in the kernel data space. The last PC and PSW are saved in locations 0 and 2 of the kernel data space.

▪ Private Memory Interconnect (PMI)

The UNIBUS is a general-purpose bus designed to handle numerous interface modules and a variety of peripheral devices. The UNIBUS protocols were designed for optimum data transfer on a 50-foot long bus. This length and the delay characteristic of the associated bus drivers and receivers result in a relatively low data transfer rate. To increase data transfer rate on the PDP-11/84 system, Digital's engineers had to extend and upgrade the existing bus structure so that it could function concurrently as two separate buses. The result was the development of the private memory interconnect (PMI), which shares most of the existing bus hardware, but at the same time has its own bus architecture and data transfer protocols.

The PMI's principal contribution is to double the speed of DMA and CPU/memory transfers. The basis for this process is serial double-word transfers, often called double-pumping. Double-word transfers involve moving two words in sequence with only one address. Performance improvement was achieved, first, with a private memory interconnect having faster drivers and receivers and accommodating the reduced transfer delay demands on a shorter bus. The interconnects transfer protocols then take advantage of the high-speed capability.

The PMI consists of a multiplexed data/address communication path between the CPU, PMI memory, and the UBA. Direct access to the PMI is reserved exclusively for these three modules. All communications between these modules and the UNIBUS devices are controlled and directed through the UBA. The CPU and UBA modules, but not the memory module, can become PMI master. All three can respond as PMI slaves. The CPU is always the PMI default master and the UBA is the UNIBUS's default master. The PMI arbitration logic is located on the CPU and arbitrates both PMI and UNIBUS interrupt/data transfers.

Once a bus structure is established, its effective speed in memory transfers depends on the bus protocols and control signals that make it work. In the PMI, the control signals manage PMI operations to reduce total transfer time. Twenty control signals are used in the PMI protocol.

The PMI supports standard UNIBUS interrupt transfers through the UBA. During a vectored interrupt transfer, the UNIBUS device initiates the interrupt request line that is then received directly by the processor logic. After receiving the interrupt request, the processor arbitration logic arbitrates priorities and grants PMI mastership to the UBA interrupting device. After acquiring bus mastership, the UNIBUS device can control data or vector interrupt transfers via the UBA.

▪ Memory System

Memory Management

Memory management hardware is standard on the PDP-11/84 system, providing a dual-register set, 22-bit addressing, separate instruction and data space, and three operating modes. The J-11 chipset contains the memory management unit (MMU) that provides the user with the hardware necessary to effect complete memory management and protection. The MMU is designed to provide access to all of physical memory and is an important part of multiuser, multiprogramming systems on which memory protection and relocation facilities are necessary.

The basic characteristics of PDP-11/84 memory management unit are:

- 16 kernel-mode memory pages.
- 16 supervisor-mode memory pages.
- 16 user-mode memory pages.
- 8 pages in each mode for instructions.
- 8 pages in each mode for data.
- Page lengths from 64 to 8,192 bytes, in increments of 64 bytes.
- Full protection and relocation for each page.
- Transparent operation.
- Memory access to 4 Mbytes.

The PDP-11/84 implements PDP-11/44-compatible memory management. The visible memory management state consists of 48 page address registers (PARs), 48 page descriptor registers (PDRs), and four memory management registers (MMRO, MMR1, MMR2, MMR3). Details of these registers may be found in both the *KDJ11-B CPU Module User's Guide* and the *PDP-11/84 Technical Manual.*

The PDP-11/84 processor can perform 16-bit, 18-bit, or 22-bit address mapping. This mapping operation provides compatibility with the earlier PDP-11 UNIBUS computers. This means that software written and developed for any PDP-11 computer can run on the PDP-11/84 without modification.

Error Correction Code (ECC)

Error correction code is a technique used to check the contents of memory to detect errors and correct them before sending the data to the processor. The process of checking is accomplished by combining the bits in a number of unique ways so that parity, or syndrome, bits are generated for each unique combination and stored along with the data bits in the same word. The memory word length is extended to store these unique bits. When memory is read, the data word is checked against the syndrome bits stored with the word. If they match, the word is sent on to the processor. If they do not match, an error exists and the mismatch of the syndrome bits determines which data bit is in error. The bit in error is then corrected, and this corrected data is then sent on to the processor. The ECC that is employed in MOS memory will detect and correct single-bit errors in a word, and detect double-bit errors in a word. Where a double-bit error is detected the data cannot be corrected so the processor is notified (as happens with a parity error in parity memory).

ECC provides maximum system benefits when used in a storage system that fails in a random single-bit mode rather than in blocks or large segments. Single-bit errors (or failures) are the predominant failure mode for modern MOS memories.

ECC memory provides fault tolerance with the result that multiple hard or soft single-bit failures can be present in a memory system without causing measurable degradation in either performance or reliability.

Battery-Backup Unit (BBU)

MOS memory is volatile. It depends on electricity to store information. Because a power loss or shutdown causes data loss, battery-backup units are designed to temporarily preserve data in memory. These units are available as options on the PDP-11/84 system.

Generally, the incidence of ac-line power loss varies inversely with the severity of loss. That is, there is an extremely small number of long-term failures of ac power, and a relatively larger number of short-term failures or drops in voltage. Battery backup units are not intended to preserve data overnight or over weekends, but rather to prevent data loss during infrequent, short-term failures of ac power.

UNIBUS Adapter (UBA)

The UNIBUS adapter is essentially a special purpose DMA device that contains the UNIBUS-to-PMI adapter logic, UNIBUS mapping, and four boot ROM sockets. The multifunction UNIBUS adapter links the PMI and UNIBUS, to which is connected one or more UNIBUS-compatible peripheral devices. The UBA communicates with the CPU and memory via the PMI protocol and communicates with peripheral devices on the UNIBUS via the UNIBUS protocol. Among its many functions, the UBA is responsible for translating the 18-bit UNIBUS DMA addresses to 22-bit PMI addresses and vice-versa exchanging information between UNIBUS and PMI protocols.

- *DMA Cache*

  Performance of UNIBUS DMA is improved by adding a 32-word cache to the UBA module (see Figure 6-8). When a DMA read from memory occurs and the address is an even 8-word boundary, the first word of data is transferred to the I/O device and the next 7 words are cached. When the data from the next 7 addresses is required, this data is taken directly from the cache. The DMA cache is a four-set associative case. DMA cache contains 32 16-bit data registers, arranged in four sets of eight data registers each. The data registers are located in RAM memory. The tag registers and valid bits are located in the UBA gate array. The operation of DMA cache is transparent to the software.

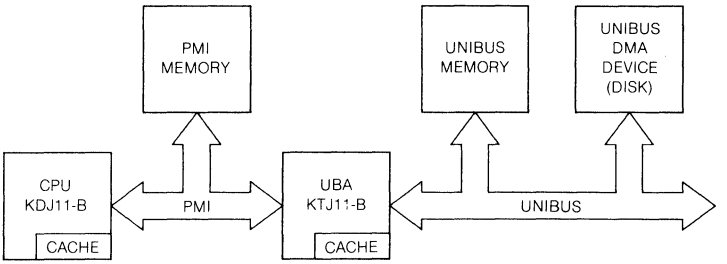


Figure 6-8 ▪ PDP-11/84 Cache Diagram

The DMA cache allows the UBA to group UNIBUS word read-operations into PMI octal words. It also provides a speed-matching buffer to level out variations in PMI or UNIBUS traffic. Because the cache is organized as four groups of eight words each, the UBA can very efficiently manage up to four concurrent UNIBUS DMA devices without thrashing.

▪ *UNIBUS Mapping*

The UNIBUS map is the interface between the UNIBUS and PMI memory. It responds as a slave to UNIBUS signals and is used to convert 18-bit UNIBUS addresses to 22-bit memory addresses. The 22-bit memory address is accompanied by an additional signal line, BBS7 L. The assertion of BBS7 L disables the PMI address decoding and selects the I/O page.

The UNIBUS address space is 256 Kbytes, of which the top 8,192 (8-Kbyte) addresses always refer to the I/O page. The lower 248 Kbytes of UNIBUS address space can be used by the UNIBUS map to refer to physical memory. The UNIBUS map can be programmed, via memory management register three (MMR3) to run with relocation enabled or disabled. UNIBUS mapping and the memory management registers are explained in detail in the *PDP-11 Architecture Handbook*.

The ability to install UNIBUS memory instead of, or in addition to, PMI memory has been preserved with the PDP-11/84. UNIBUS address space is assigned to UNIBUS memory in 8-Kbyte segments, starting with the segment below the I/O page and proceeding downward. The UNIBUS address segments assigned to UNIBUS memory cannot be used to access PMI memory via the I/O map. Whenever the CPU accesses UNIBUS memory, the UBA will disable the PMI memory response. The CPU module does not cache UNIBUS memory.

The memory configuration register (KMCR), one of three registers on the UBA, reflects the placement of UNIBUS memory by allowing the CPU boot and diagnostic programs to configure the UBA for the distribution of UNIBUS and main memory within the system. Additional KMCR bits allow the DMA cache to be enabled and disabled, provide diagnostic status of the read buffer, and supply information on the reboot status of the system. For a more detailed description of the operational UNIBUS memory and the registers on the UBA, refer to the *PDP-11/84 Technical Manual.*

- *Boot ROM Facility (M9312-Compatible)*
  The UBA boot ROM facility allows the user to install M9312-compatible boot programs written for UNIBUS devices that are not directly supported by the CPU boot programs. ROM programs that run on the M9312 also work on the UBA. The M9312-compatible boot programs are implemented with 1-4 512x4-bit ROMs. Each ROM contains 64 16-bit words of accessible code located in the first half of the ROM. The last 256 4-bit ROM locations are not used.

  The CPU module can be configured to boot the system from:

  - One of its self-contained boot programs.
  - EEROM, user-written in EEROM.
  - The UBA boot ROM facility.
  - An M9312-type ROM option that resides on the UNIBUS.

For details of the location of each ROM socket, along with its address range, see the *PDP-11/84 Technical Manual.*

Cache Memory

The PDP-11/84 has a large 8-Kbyte cache located on the CPU module that is used to decrease the system access time of instructions and data (see Figure 6-8). CPU cache operations occur only for memory on the PMI bus. Memory on the UNIBUS, if present, is not cached.

The 8-Kbyte write-through direct-map CPU cache has dual-tag stores that allow concurrent operations of the CPU and DMA. The cache is transparent to all programs and acts as a high-speed buffer between the processor and PMI memory. The data stored in cache represents the most active portion of the PMI memory being used. The processor accesses main memory only when data is not available in the cache.

Cache memory is a high-speed memory that buffers data between the CPU and main memory. When a memory access occurs, the system looks first for data in cache memory. If found (a hit), the data is read from cache and execution proceeds at the fastest rate. If not found (a miss), the data must be read from memory and written to cache.

In a write-through cache system, such as the earlier PDP-11/44, a CPU request to write data into memory causes data to be written to both the cache and to main memory. This ensures that both stores are always updated immediately.

Typical hit/miss ratios in a write-through cache system are summarized in Table 6-2. In a typical program, WRITEs occur only 10-15 percent of the time and READs occur 85-90 percent of the time. Thus, READ misses cause the cache to be updated.

Table 6-2 ▪ Typical Hit/Miss Operations

	CACHE	MAIN MEMORY
READ		
hit	no change	no change
miss	updated	no change
WRITE		
hit	updated	updated
miss	no change	updated

The top 8 Kbytes of physical memory (the I/O page) are not cached. This is because the I/O page contains device-status registers that, when read, must always convey the latest information.

When a DMA device writes to a cached location, the cache entry must be invalidated. The cache system monitors DMA transactions to determine if this action is needed. Because the PDP-11/84 contains two identical tag stores, this DMA monitoring can overlap CPU cache accesses. Only if a DMA write-hit occurs must the CPU be stopped to invalidate the cached location.

The following matrix (Table 6-3) shows the cache response for both DMA and CPU data transfers from and to the PMI memory space. Two cache tag memories are referenced in the matrix. The DMA tag matrix heading refers to DMA activity. The CPU tag matrix heading refers to PMI activity involving the CPU.

Table 6-3 ▪ PDP-11/84 Cache Response Matrix

▪ DMA TAG

	Hit	Miss
Read	Read memory	Read memory
Write word	Invalidate cache Write memory	Write memory
Write byte	Invalidate cache Write memory	Write memory
Read bypass	—	—
Write bypass	—	—
Read force miss	Read memory	Read memory
Write force miss	Write memory Invalidate cache	Write memory

▪ CPU TAG

	Hit	Miss
Read	Read cached data	Read memory and allocate cache
Write word	Write both cache and memory	Write memory No cache change
Write byte	Write both cache and memory	Write memory No cache change
Read bypass	Invalidate cache and read memory	Read memory No cache change
Write bypass	Invalidate cache and write memory	Write memory No cache change
Read force miss	Read memory No cache change	Read memory No cache change
Write force miss	Write memory No cache change	Write memory No cache change

Cache parity errors affect the cache response matrix in the following ways:

- During DMA write cycles, a DMA tag parity error forces a cache hit response, and the cache location is invalidated.
- During CPU read cycles (nonbypass), a CPU tag or data parity error forces a cache-miss response.
- During CPU write-byte cycles (nonbypass; nonforce miss), a CPU tag parity error forces a cache-hit response, but the data is loaded with bad parity.
- During CPU read-bypass or write-bypass cycles, a CPU tag or data parity error forces a cache-hit response. The cache location is invalidated.
- For all force-miss cycles, and for the CPU write-word (nonbypass) cycle, cache parity is ignored.

- *Cache Registers*

The PDP-11/84 contains hardware that allows the user to control the cache memory. This hardware consists of the cache control register, the memory system error register, and the hit/miss register. These registers allow for a broad spectrum of cache implementations and considerable flexibility in designing a cache memory scheme to fit a specific application.

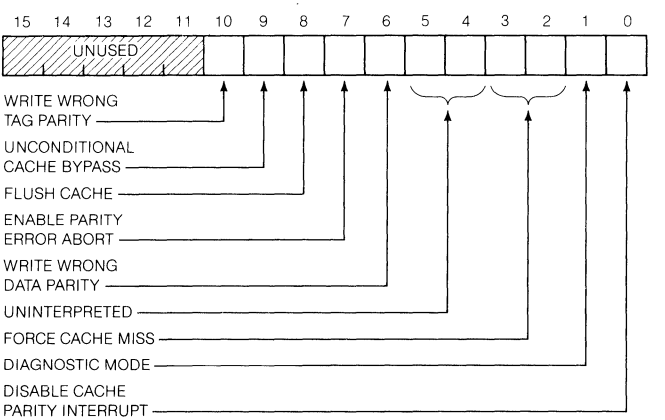


Figure 6-9 ▪ 17 77 746 Cache Control Register

▪ *Cache Control Register (CCR)*

Bits: 15-11

Name: Unused

Function: These bits always read as zeros.

Bit: 10

Name: Wrong Tag Parity

Function: When this read/write bit is set, the CPU and DMA tag parity bits are both written as wrong parity during all operations that update these bits. A cache tag parity error will thus occur on the next access to that location.

Bit: 9

Name: Unconditional Cache Bypass

Function: When this read/write bit is set, all references to memory by the CPU will bypass the cache and go directly to main memory. Read or write hits will result in the invalidation of the corresponding cache location; misses will not affect the cache contents.

Bit: 8

Name: Flush Cache

Function: Writing a one into this write-only bit clears all CPU tag and DMA tag valid bits invalidating the entire contents of the cache. Writing a zero into this bit has no effect. Flush Cache always reads as zero. The CPU requires approximately 1.2 milliseconds to flush the cache.

Bit: 7

Name: Parity Error Abort

Function: This read/write bit is set for diagnostic purposes only. When it is set, a cache parity error (during a CPU cache read) will cause the CPU to abort the current instruction and trap to parity error vector 114. When this bit is clear, a cache parity error (during a CPU read) results in a force miss and data fetch from main memory. The CPU will trap to 114 only if CCR bit 0 is clear. DMA cycle cache parity errors will cause a trap to 114 if CCR 7 is set or CCR 0 is clear. CCR 7 has no effect on main memory parity errors that always cause the CPU to abort the current instruction and trap to 114.

Bit: 6

Name: Write Wrong Data Parity

Function: When this read/write bit is set, both the high and low data parity bits are written with wrong parity during all operations that update these bits. This will cause a cache data parity error to occur on the next access to that location.

Bits: 3-2

Name: Force Miss

Function: When either of these read/write bits is set, CPU reads will be reported as cache misses.

Bit: 1

Name: Diagnostic Mode

Function: When this read/write bit is set, a 10-microsecond nonexistent memory timeout during a word write will not cause a nonexistent memory trap and will not set CPU error register bit 5. All nonbypass and nonforced miss word writes will allocate the cache regardless of the nonexistent memory timeout.

Bit: 0

Name: Disable Cache Parity Interrupt

Function: This read/write bit controls cache parity interrupts when CCR 7 is clear (normal operation). If CCR 7 is clear, a cache parity error (during a CPU cache read) results in a force miss and data fetch from main memory. The CPU will trap to 114 only if CCR bit 0 is clear. DMA cycle cache parity errors will cause a trap to 114 if CCR 7 is set or if CCR 0 is clear.

▪ *Memory System Error Register (MSER)*

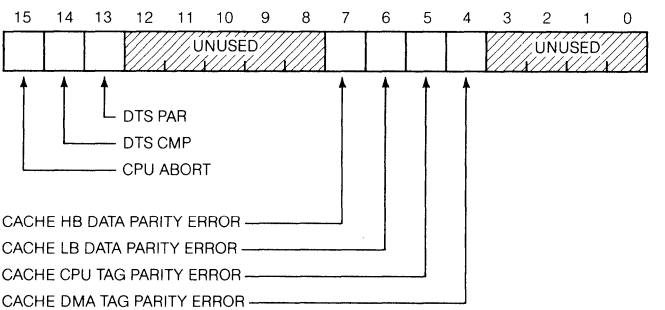


Figure 6-10 ▪ Memory System Error Register

Bit: 15

Name: CPU Abort

Function: This read-only bit is set if a cache or main memory parity error results in an instruction abort, such as during the demand read cycle. Cache parity errors cause an abort only if CCR 7 is set. Main memory parity errors always cause an abort.

Bit: 14

Name: DMA Tag Store Comparator (DTS CMP)

Function: In stand alone mode (boot and diagnostic controller status register [BCSR] bit 8 set), this read-only bit indicates the output of the cache DMA tag store comparator for the previous non-I/O page reference with cache miss. When BCSR 8 is clear, DTS CMP reads as zero.

Bit: 13

Name: DMA Tag Store Parity (DTS PAR)

Function: In stand alone mode (BCSR bit 8 set), this read-only bit indicates the output of the DMA tag store parity check logic for the previous non-I/O page reference with cache miss. When BCSR 8 is clear, DTS PAR reads as zero.

Bits: 12-8

Name: Unused. These bits always read as zeros.

Bit: 7

Name: Cache HB Data Parity Error

Function: This read-only bit is set if a parity error is detected in the high-data byte during a CPU cache read. If CCR 7 is clear, MSER 7 is also set by a low-data byte parity error and by the set condition of MSER 5 or 4.

Bit: 6

Name: Cache LB Data Parity Error

Function: This read-only bit is set if a parity error is detected in the low-data byte during a CPU cache read. If CCR 7 is clear, MSER 6 is also set by a high-data byte parity error and by the set condition of MSER 5 or 4.

Bit: 5

Name: Cache CPU Tag Parity Error

Function: This read-only bit is set if a parity error is detected in the CPU tag field during a CPU cache read. If CCR 7 is clear, MSER 7 is also set by a high- or low-data byte parity error.

Bit: 4

Name: Cache DMA Tag Parity Error

Function: This read-only bit is set if a parity error is detected in the DMA tag field during a DMA write operation.

NOTE

Cache parity errors are ignored (that is, they do not affect MSER 7-4) if either CCR 3 or 2 (force miss) is set or if the CPU tag valid bit is clear.

Bits: 3-0

Name: Unused. These bits always read as zeros.

- *Hit/Miss Register*

Figure 6-11 ▪ Hit/Miss Register

The hit/miss register indicates whether the six most recent CPU memory references resulted in cache hits or cache misses. Bits enter from the right (at bit 0) and are shifted left. A 1 indicates a cache hit, a zero indicates a cache miss.

The hit/miss register is read-only. Bits 15-6 are not used and always read as zeros. The register's value at powerup is undefined and is not affected by console start or a RESET instruction. The hit/miss register will always read zero when the CPU is in console ODT mode.

▪ Backplane

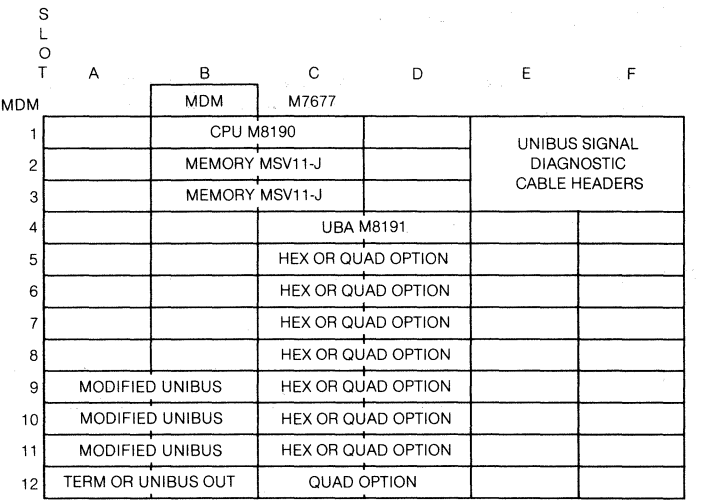


Figure 6-12 ▪ PDP-11/84 Backplane Configuration

The PDP-11/84 backplane (Figure 6-12) consists of 13 slots. Slot MDM is reserved for the monitor and distribution module. Slot 1 is dedicated to the KDJ11-B CPU module. Slots 2 and 3 are reserved for the MSV11-J memory modules. If only one memory module is installed, either slot 2 or 3 can be used. The final dedicated slot, slot 4, is for the UBA. Slots 1 through 4 are connected via the PMI bus on rows A, B, C, and D. Slots 5 through 12 are wired to support any UNIBUS-compatible small peripheral controller (SPC). Quad SPCs are installed in rows C, D, E, and F. In addition to SPC support, slots 9, 10, and 11 support modified UNIBUS devices. A terminator module or UNIBUS-out cable must be installed in slot 12, rows A and B.

The monitor and distribution module has unique features that are essential to the function of the PDP-11/84. It is a quad-height board and includes power supply voltage indicators and test points, a fan/blower rotation monitor with audible alarm that shuts down the system after 60 seconds of air-moving failure, and nonprocessor jumper switches for slots 5 through 12. It has module fingers for row B only and occupies only backplane slot MDM.

▪ Console Functions

Console Serial-Line Unit

The console serial-line unit provides the J-11 microprocessor with a serial interface for the console terminal. There are four console serial-line unit registers—the receiver status register, the receiver data buffer, the transmitter status register, and the transmitter data buffer. These registers cannot be disabled.

▪ *Receiver Status Register (RCSR)*

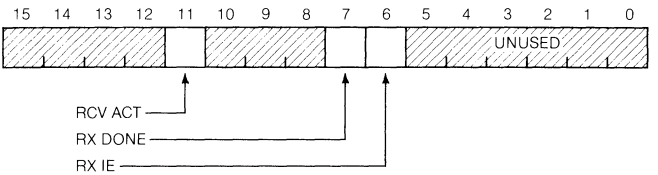


Figure 6-13 ▪ 17 777 560 Receiver Status Register

Bits: 15-12

Name: Unused. They always read as zeros.

Bit: 11

Name: Receiver Active (RCV ACT)

Function: This read-only bit is set at the center of the start bit of the serial input data and is cleared at the expected center of the stop bit at the end of the serial data.

Bits: 10-8

Name: Unused. They always read as zeros.

Bit: 7

Name: Receiver Done (RX DONE)

Function: This read-only bit is set when an entire character has been received and is ready to be read from the RBUF register. This bit is automatically cleared when RBUF is read. It is also cleared by powerup. RX DONE is set one bit time after RCV ACT clears.

Bit: 6

Name: Receiver Interrupt Enable (RX IE)

Function: This read-write bit is cleared by powerup and bus INIT. If both RCVR DONE and RCVR INT ENB are set, a program interrupt is requested at priority level 4.

Bits: 5-0

Name: Unused. They always read as zeros.

▪ *Receiver Data Buffer (RBUF)*

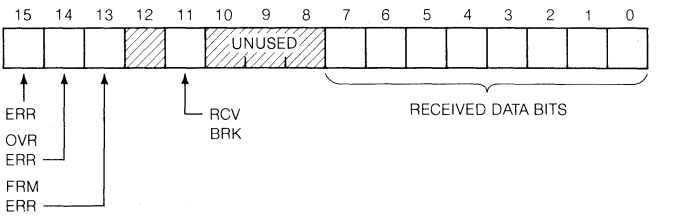


Figure 6-14 ▪ 17 777 562 Receiver Data Buffer

Bit: 15

Name: Error (ERR)

Function: This read-only bit is set if RBUF bit 14 or bit 13 is set. ERR is clear if these two bits are clear.

Bit: 14

Name: Overrun Error (OVR ERR)

Function: This read-only bit is set if a previously received character was not read before being overwritten by the present character.

Bit: 13

Name: Framing Error (FRM ERR)

Function: This read-only bit is set if the present character had no valid stop bit. This bit is used to detect break if the front console panel switch is enabled and the break key is depressed on the console terminal.

NOTE

Error conditions remain present until the next character is received, at which point the error bits are updated. The error bits are not necessarily cleared by powerup.

Bit: 12

Name: Unused. This bit always reads as zero.

Bit: 11

Name: Received Break (RCV BRK)

Function: This read-only bit is set at the end of a received character for which the serial data input remained in the SPACE condition for all 11 bit times. RCV BRK then remains set until the serial data input returns to the MARK condition.

Bits: 10-8

Unused. These bits always read as zeros.

Bits: 7-0

Name: Received Data Bits

Function: These read-only bits contain the last received character.

- *Transmitter Status Register (XCSR)*

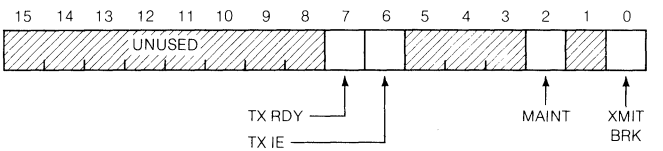


Figure 6-15 ▪ 17 777 564 Transmitter Status Register

Bits: 15-8

Name: Unused. Read as zeros.

Bit: 7

Name: Transmitter Ready (TX RDY)

Function: This read-only bit is cleared when XBUF is loaded and set when XBUF can receive another character. TX RDY is also set by powerup and by bus INIT.

Bit: 6

Name: Transmitter Interrupt Enable (TX IE)

Function: This read-write bit is cleared by powerup and by bus INIT. If both TX RDY and TX IE are set, a program interrupt is requested.

Bits: 5-3

Name: Unused. Read as zeros.

Bit: 2

Name: Maintenance (MAINT)

Function: This read-write bit is used to facilitate a maintenance self-test. When MAINT is set, the DLART internally connects its output to its input. Input from the EIA receivers is ignored. The EIA output drivers are still connected and any characters sent with MAINT set will wrap around to the input and also go to the output drivers. This bit is cleared by powerup and by bus INIT.

Bit: 1

Name: Unused. Read as zero.

Bit: 0

Name: Transmit Break (XMIT BRK)

Function: When this read-write bit is set, the serial output is forced to the SPACE condition. XMIT BRK is cleared by powerup and by bus INIT.

▪ *Transmitter Data Buffer Register (XBUF)*

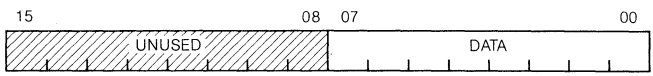


Figure 6-16 ▪ 17 777 566 Transmitter Data Buffer

Bits: 15-8

Name: Unused.

Bits: 7-0

Name: ASCII character.

Function: These eight bits are write-only bits used to load the character to be transmitted.

▪ Line-time Clock

The line-time clock provides the system with timing information at fixed intervals determined by the UNIBUS line-time clock (LTC) signal from the power supply at 50 or 60 Hz, or by one of the onboard J-11 processor frequency signals as programmed by the boot and diagnostic controller status register bits 11 and 10. The three on-board frequencies are 50 Hz, 60 Hz, and 800 Hz.

Clock Status Register (LKS)

The clock status register (LKS) allows line-clock interrupts to be enabled and disabled under program control.

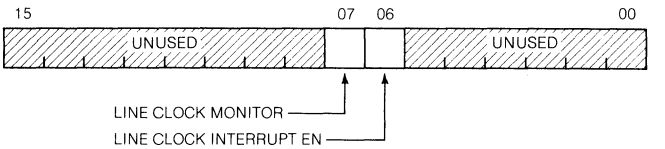


Figure 6-17 ▪ 17 777 564 Clock Status Register

Bits: 15-8

Name: Unused. Always read as zeros.

Bit: 7

Name: Line Clock Monitor (LCM)

Function: This read-only bit is set by the leading edge of the external BEVENT line (or one of the three on-board frequencies) and by bus INIT. Line clock monitor is cleared automatically on processor interrupts acknowledgment. It is also cleared by writes to the LKS with bit 7 equal to zero.

Bit: 6

Name: Line Clock Interrupt Enable (LCIE)

Function: This read-write bit, when set, causes the set condition of LCM (LKS 7) to initiate a program interrupt request. When LCIE is clear, line clock interrupts are disabled. LCIE is cleared by powerup and by bus INIT. LCIE is held set when BCSR 13 (FRC LCIE) is set.

Bits: 5-0

Name: Unused. Always read as zeros.

▪ Console

The console terminal of the PDP-11/84 provides these functions:

- Setup of the system hardware.
- Communications with the operating software.
- Debugging of hardware and software.

While the console terminal is being used to set-up the system hardware, the terminal is said to be in setup mode. While the console terminal is communicating with the operating software, it is said to be in program mode. While the PDP-11/84 is stopped, the console terminal is said to be in console-ODT mode.

The three modes are summarized in Table 6-4. Various conditions can switch the console terminal between modes. These conditions are shown in Figure 6-18.

Table 6-4 ▪ PDP-11/84 Console Terminal Modes

	Setup-mode	Program-mode	Console-mode
Microprogram running?	Yes	Yes	Yes
Macroprogram running?	from ROM	from memory	No
UNIBUS functioning?	No	Yes	Yes

Setup Mode Functions

All setup of the PDP-11/84 CPU, except for the selection of the console baud rate, is performed via the console terminal using console-mode commands. Setup mode is a ROM-based program that can be entered when the ROM code is in dialog mode. Dialog mode allows the user to enter setup mode, map the memory and the I/O page, run ROM-based diagnostics, list available boot programs, and boot a device.

This setup dialog allows the user to determine the actions taken by the ROM code at powerup or restart and includes:

- Selection of one of four powerup or restart modes

  — Enter dialog mode after running tests.
  — Autoboot selected devices after running tests.
  — Enter ODT HALT.
  — Restart operating system through powerup vector 24 for systems with battery backup memory.

- Selection of register values and other parameters

  — Enable/disable the UNIBUS cache.
  — Enable/disable the line clock CSR and select clock source.
  — Select testing.
  — Enable/disable trap on HALT.
  — Override errors for nonstandard boot blocks.
  — Additional parameters...

- Bootstrap device selection for autoboot mode

  — Select one to six devices to try to boot.
  — Identify nonstandard CSR addresses.

- Support of custom bootstraps loaded in the EEROM

  — Create a custom bootstrap.
  — Delete a custom bootstrap.
  — Edit a custom bootstrap.

The setup dialog appears as a series of menu pages that are compatible with hardcopy and softcopy terminals.

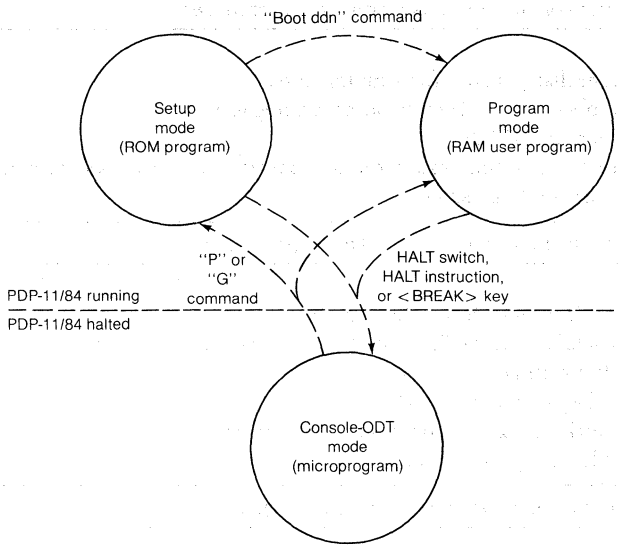


Figure 6-18 ▪ Conditions that Switch Console Terminal Modes

- *Leaving Setup Mode*

  The user exits setup mode by typing CTRL C. Then the user can either boot the operating system manually (i.e., BOOT DU0<CR>) or toggle the restart switch if autoboot mode is selected.

Program Mode Functions

While the console terminal is operating in program mode, it is under the sole control of the operating system software, and most of its functions are defined by that software.

- *Leaving Program Mode*

  The user enters console-ODT in one of three ways: by pressing the <BREAK> key while the PDP-11/84 keyswitch is in the <ENABLE> position, by placing the HALT/RUN/RESTART switch in the HALT position (again, while the keyswitch is in the <ENABLE> position), or by the PDP-11 program executing a HALT instruction while in kernel mode.

  The user enters setup mode by simply pressing the HALT/RUN/RESTART switch to the RESTART position and typing CTRL C after the "Testing in progress—Please wait" message is displayed. Setup mode can be entered by setting the force dialog mode switch ON and pressing the HALT/RUN/RESTART switch to the RESTART position.

Console ODT

The console mode can also be used to examine or modify the contents of memory or UNIBUS device registers. Programs may also be stopped, started, or single-stepped. These functions are implemented by a microprogram contained within the J-11 processor chip. During the execution of these functions, the PDP-11/84 CPU is stopped. That is, although the J-11 microprogram is still running, no PDP-11 macroinstructions are being executed.

These functions are collectively referred to as console ODT (octal debugging technique) and are generally compatible with both ODT-11 and the console ODT functions provided by the LSI-11, the MicroPDP-11/23, and the PDP-11/24.

Console ODT can examine any read-only or read/write address in the entire 22-bit address space. Console ODT can also modify any read/write (not write-only) location. Locations that read all zeros (such as the console XBUF) can also be modified. The ODT commands are summarized in Table 6-5.

Table 6-5 ▪ Console ODT Commands

Command	Symbol	Function
Slash	n/	Opens the specified location (n) and outputs its contents. The n is an octal number.
Carriage Return	<CR>	Closes an open location.
Line Feed	<LF>	Closes an open location and then opens the next contiguous location.
Internal Register Designator	$n or Rn	Opens a specific processor register (n). n is an integer from 0 to 7 or the character S.
Processor Status Word Designator	S	Opens the PSW—must follow an $ or R command.
Go	G	Starts program execution.
Proceed	P	Resumes execution of a program.
Binary Dump	Control S	Manufacturing use only.

▪ *Leaving Console ODT Mode*

The user can leave console-ODT mode via:

- The P command.
- The G command.
- The boot switch.

If the user were previously in setup mode, the P or G command will exit to setup mode.

NOTE

The G command is used for a total restart, while P allows the user to proceed. The G command reinitializes the UNIBUS and resets memory.

If the user were previously in program mode, the P or G command will exit to program mode. The bootswitch's action depends on how you have set up the PDP-11/84.

▪ Specifications

Packaging

PDP-11/84 packaging provides maximum flexibility in system configuration. The basic PDP-11/84 is packaged as a 10.5-inch-by-19-inch rackmount box product or a 19-inch-by-42-inch cabinet product.

The PDP-11/84 line includes a 5.25-inch OEM Design Center, which contains a PDP-11/84 module set with 2 or 4 Mbytes of memory, a 9-slot processor backplane, 650-watt power supply, a 5.25-inch-by-19-inch rackmount enclosure, and five slots for system options.

The 10.5-inch OEM Design Center is also available, and includes the PDP-11/84 module set with 2 or 4 Mbytes of memory with the 9-slot processor backplane. It includes a 6 system unit design, expandable to 27 slots, a 10.5-inch-by-19-inch rackmount enclosure, and an 1100-watt power supply.

Three Kernal Systems, system building blocks based on the 10.5-inch OEM Design Center, are available. All three offer an optional 27-slot expansion capacity, an 1100-watt power supply and an FCC-shielded I/O bulkhead. You can choose one, two, or three rackmount option space systems.

UNIBUS System Cabinets offer two, three, or four 10.5-inch-by-19-inch openings for processors or device options, and allow exceptional configuration flexibility, as well as excellent expansion and power subsystem interconnect possibilities.

Standard Equipment

- PDP-11/84 CPU.
- Memory management.
- Line-frequency clock.
- 8-Kbyte cache memory.
- 1- or 2-Mbyte ECC MOS memory.
- Floating-point accelerator (FPA) processor.

- Console serial-line unit.
- Boot and diagnostic ROM facility.
- Electrically erasable programmable ROMs (EEPROM).
- UNIBUS adapter (UBA) module.
- Monitor and distribution module.

Prewired Expansion Space for Optional Equipment

- 8 SPC slots for UNIBUS compatible peripherals, 7 hex and 1 quad.
- 1-Mbyte or 2-Mbyte ECC MOS memory (up to 4 Mbytes maximum).
- 0 SU open spaces in the box product.
- 2 SU open spaces in the cabinet product.

Other Specifications

ac Power

Box	90–132 Vrms, 47–63 Hz, 1-phase power, 8 A rms maximum @ 120 Vac 180–264 Vrms, 47–63 Hz, 1-phase power, 4.2 A rms maximum @ 240 Vac
Cabinet	93–132 Vrms, 47–63 Hz, 1-phase power, 11.3 A rms maximum @ 120 Vac 186–264 Vrms, 47–63 Hz, 1-phase power, 5.6 A rms maximum @ 240 Vac

Physical Characteristics

Box	48.2 cm wide by 68.6 cm deep by 26.7 cm high (19 in by 27 in by 10.44 in)
Cabinet	53.9 cm wide by 80 cm deep by 105.7 cm high (21.5 in by 31.52 in by 41.64 in)

Weight

Box	44.5 kg (98 lb)
Cabinet	150.0 kg (331 lb)

Operating Environment

- *Box*

Temperature:	10°C–50°C (50°F to 122°F)
Humidity:	10%–90% with maximum wet bulb of 32°C (90°F) (noncondensing)
Altitude:	To 2.4 km (8,000 ft)

- *Cabinet*

Temperature:	10°C–40°C (50°F–104°F)
Humidity:	10%–90% with maximum wet bulb of 28°C (82°F) (noncondensing)
Altitude	To 2.4 km (8,000 ft.)

Nonoperating Environment

- *Box*

Temperature:	-40°C–66°C (-40°F–151°F)

- *Cabinet*

Temperature:	-40°C–66°C (-40°F–151°F)

Chapter 7 ▪ UNIBUS Technical Description

▪ Characteristics of the UNIBUS

All computers must contain some method of interfacing with the outside world. Some computers require a collection of integrated input/output (I/O) controllers with special connections to the processor and/or memory. Other computers require a collection of wires, and a protocol explaining how to use these wires to communicate with the rest of the computer. The set of wires is called a *bus*. The UNIBUS is used by many Digital computers.

Some of the major characteristics of the UNIBUS are listed below, and discussed in subsequent sections.

- Nonmultiplexed bus.
- Strict master/slave relationship.
- Partially distributed arbitration.
- Overlapped arbitration and data transfer.
- Asynchronous operation.
- 18 address bits.
- Word or byte operations.
- Parity error information from slaves.

Nonmultiplexed Bus

Most buses contain the concepts of address and data. All things on the bus are uniquely accessed by means of an *address*, or a series of addresses. The address on the bus is very much like a telephone number or an address on an envelope. Once a unique device has been addressed (selected), *data* may be exchanged with that device. This data corresponds to the phone conversation, or the letter itself.

Some buses first use the same physical set of wires to carry the address, followed by one or more items of data. These buses are called *multiplexed buses* (see Figure 7-1). Digital's Q-bus is an example of a multiplexed bus. Other buses use separate sets of wires to carry the address and data. These buses are called *nonmultiplexed buses* (see Figure 7-2). The UNIBUS is an example of a nonmultiplexed bus.

Because an address and data can be carried simultaneously, a nonmultiplexed bus is usually faster, particularly if each new address bears no rela-

tionship to any preceding address. However, multiplexed buses share the same set of wires for address and data so they tend to be physically smaller and therefore less expensive. If addresses are usually presented in a particular order (for example, ascending), the next address can be predicted rather than explicitly transmitted, and the performance of the multiplexed bus can approach the performance of the nonmultiplexed bus.

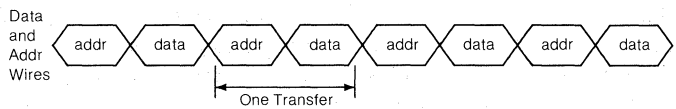


Figure 7-1 ▪ Multiplexed Bus (like the Q-bus)

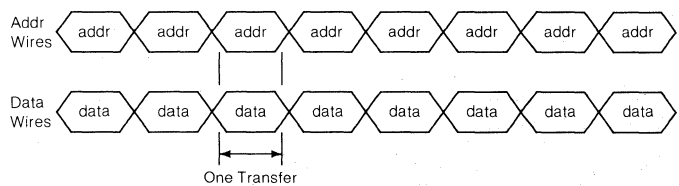


Figure 7-2 ▪ Nonmultiplexed Bus (like the UNIBUS)

Strict Master/Slave Relationship

Most transfers on the UNIBUS consist of a single device (called the *bus master*, or just *master*) requesting another device (the *slave*) to perform some operation. Other buses may use the terms *commander* and *responder* as synonyms for master and slave. No direction of data flow is implied here—the master may be requesting that the slave read or write data.

Partially Distributed Arbitration

The UNIBUS, like most buses, allows only one device at a time to be the bus master. This implies that there must be some method of selecting a bus master from among all of the devices requesting the bus. The process that performs this selection is called *priority arbitration*, and the bus is usually granted to the requester with the highest priority (according to a prearranged scheme).

Arbitration on the UNIBUS is performed using a two-dimensional scheme. A typical example is shown in Figure 7-3. The first dimension is the specific priority level of the request. The UNIBUS may be requested at any of five different priority levels. The highest priority level is named *nonprocessor request* (NPR), so named because devices requesting use of the UNIBUS at this priority level do not need any assistance from the CPU. This is somewhat

analogous to direct memory access (DMA), although the UNIBUS allows NPR transfers between any two devices, not just between a device and memory. The lower four levels are simply called *bus request 7* (BR7) through *bus request 4* (BR4). Devices request use of the UNIBUS via BR levels in order to interrupt a processor (described later).

All requests are passed to a central *bus arbitrator*. This central bus arbitrator is included in the PDP-11 processor and is always located at the front end of the UNIBUS. The arbitrator determines the highest requested level from among the five available and issues a *bus grant* for that level, if appropriate. The grant then travels toward the back end of the UNIBUS. *Non-processor grant* (NPG) is given in response to NPR, bus grant 7 (BG7) in response to BR7, and so forth.

The second dimension of the arbitration is performed when more than one device requests the UNIBUS at the same priority level. Now, the grant is passed from device to device, toward the back end of the UNIBUS. This is referred to as *daisy chaining*. Each device, in turn considers whether it wants to use the UNIBUS. A device that does not want to use the UNIBUS passes the grant on to the next device along the UNIBUS. A device that does want to use the UNIBUS does not pass the grant (*blocks the grant*) and the device wins the arbitration. Thus arbitration among more than one device on a particular level is distributed among the devices.

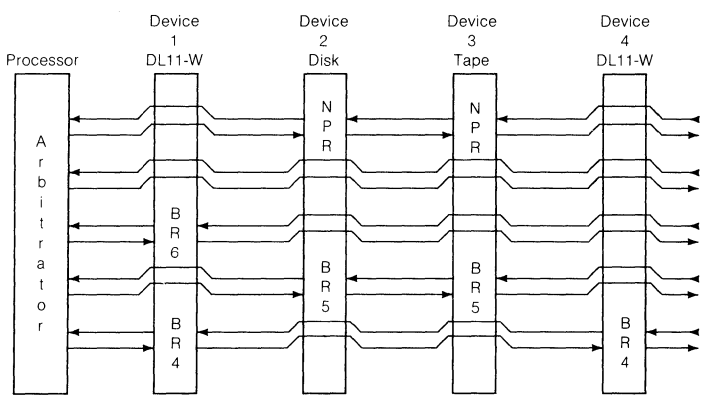


Figure 7-3 ▪ UNIBUS Two-dimensional Priority

Figure 7-3 illustrates the beginning of a typical UNIBUS. Here, the first DL11-W (Device 1) can request the UNIBUS at priority levels BR6 or BR4. Both the disk controller (Device 2) and the tape controller (Device 3) can request at NPR or BR5. The second DL11-W (Device 4) can request only at BR4. No device in this illustration uses BR7.

The UNIBUS arbitrator considers the five levels of priority and grants the highest-level request. So, in our example, the disk or the tape requesting at the NPR level always has priority over either of the DL11-Ws requesting at either BR6 or BR4.

On any given level (for example, NPR), the grant is passed from device to device along the UNIBUS. At the NPR level, the disk has priority over the tape because the disk gets an opportunity to examine (and possibly block) the grant before passing it on to the tape. Similarly, at the BR4 level, the first DL11-W has priority over the second DL11-W.

Overlapped Arbitration and Data Transfer

As mentioned previously, before a prospective bus master may use the bus, it must win the arbitration ensuring that it is the highest priority requester of the bus. Some buses can do only one thing at a time, for example, arbitrate or transfer data, but not both. This is illustrated in Figure 7-4. The UNIBUS, on the other hand, allows the next master to be selected (arbitrated) while the current master is still transferring data. This is referred to as *overlapped arbitration* and greatly contributes to the performance of the UNIBUS. See Figure 7-5.

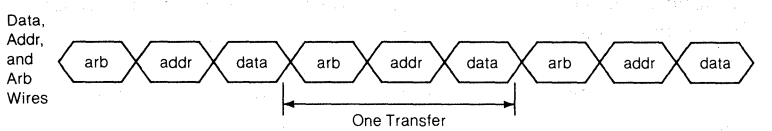


Figure 7-4 ▪ Nonoverlapped Arbitration Bus

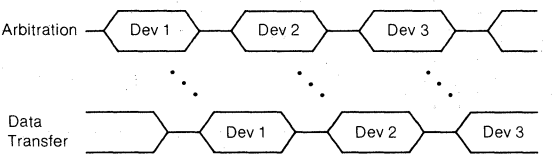


Figure 7-5 ▪ Overlapped Arbitration Bus (like the UNIBUS and Q-bus)

Asynchronous Operation

A bus may be synchronous or asynchronous. Synchronous buses partition time into parcels of fixed duration, usually called cycles. These cycles are defined by one or more clocks on the bus. Every operation on the bus must take place in an integral number of these cycles. Some buses require that every operation must take place within a single bus cycle. A synchronous bus of this sort is illustrated in Figure 7-6. Here, every rising edge of the system

clock implies that a valid address is on the bus. Every falling edge of the system clock implies that the data has replaced the address. Although synchronous buses may be very simple or very fast (although usually not both), this fixed timing imposes severe constraints on the system designer. The duration of the bus cycle must be long enough to allow for the slowest peripheral, yet short enough to get reasonable system performance.

Other synchronous systems allow a device to take multiple bus cycles for one transfer. A bus of this sort is illustrated in Figure 7-7. Here, a *wait* (or *stall*) line freezes the system, allowing slow slaves more time to process the data. This eases the timing constraints somewhat but does so at the expense of a more complex design. Also, when a wait or stall must occur, it must occur in whole-cycle "chunks," even if the data was late by only a few nanoseconds. Stalling by whole cycles degrades the performance of this synchronous bus.

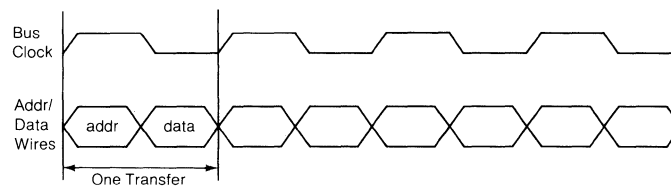


Figure 7-6 ▪ Single-cycle Synchronous Bus

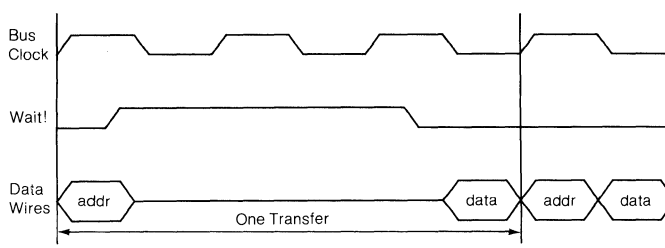


Figure 7-7 ▪ Multicycle Synchronous Bus

The UNIBUS, on the other hand, is completely asynchronous. A master clock does not time the operations of the UNIBUS. Rather, the selected master and slave devices exchange *handshaking* signals to indicate the presence of information on the bus. This handshaking allows each bus cycle to run as quickly as the particular master/slave pair will allow. Figure 7-8 shows a series of read data cycles on the UNIBUS (the master has asked the slave to provide data to the master). Note that the master first asserts the address of the desired slave, then *address strobe* (actually called MASTER SYNC or BUS MSYN L). The slave then looks up the desired data and places it on the data lines along

with a signal that combines the functions of *data strobe* and *address acknowledge* (actually called SLAVE SYNC or BUS SSYN L). No fixed timing is required between BUS MSYN L and BUS SSYN L. Fast slaves can respond quickly, while slow slaves can take more time to produce the requested data.

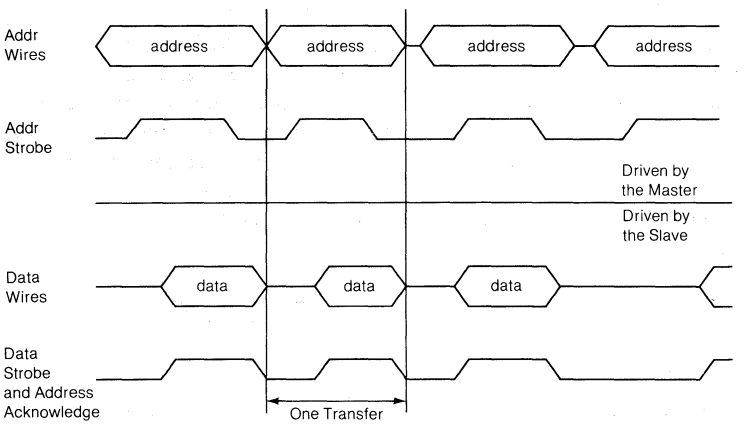


Figure 7-8 ▪ Asynchronous Bus (like the UNIBUS)

18 Address Bits

The total number of unique data that can be addressed on any bus is dependent on the number of bits used to form the address. The UNIBUS uses 18 bits to specify an address. Each address represents a byte of data. Thus the UNIBUS allows 2^{18} = 256K unique byte addresses.

Of these 256K byte addresses, 248K are reserved for system main memory, or access to system main memory, and 8K are used for access to I/O devices.

Word or Byte Operations

The UNIBUS allows data to be accessed as a 16-bit word or an 8-bit byte.

Parity Error Information from Slaves

If the slave contains logic to detect parity errors (for example, in MOS memory), the slave can pass a parity error indication to the master via a predefined UNIBUS signal. The UNIBUS itself does not include parity checking of any bus signals.

▪ UNIBUS Block Diagram

The UNIBUS may be viewed as three separate but cooperating buses. See Figure 7-9.

- Initialization and shutdown bus.
- Arbitration bus.
- Data transfer bus.

The first bus contains signals used to initialize the system or to signal that power has failed and the system must now be shut down in an orderly fashion. Any processor may drive the system initialization signal (BUS INIT L), while power supplies are generally responsible for driving the system shutdown signals (BUS AC LO L and BUS DC LO L). Every device monitors one or more of these signals.

The second bus controls who may have access to the data transfer bus. A central device called the bus arbiter receives requests for use of the UNIBUS, and replies with grants (permission to use the UNIBUS).

The third bus allows the exchange of data between a selected master device and a selected slave device.

The UNIBUS may contain more than one processor, but only one processor is designated to be in charge of handling (servicing) interrupts. This processor is called the *interrupt-fielding processor* (IFP). The arbitrator may, on its own authority, issue an NPG in response to an NPR, because a nonprocessor request by definition does not allow a processor to be interrupted. If, however, the request is a BR (which would interrupt the IFP), the arbitrator must first consult the IFP for permission to issue the BG, because the IFP may not allow the interrupt.

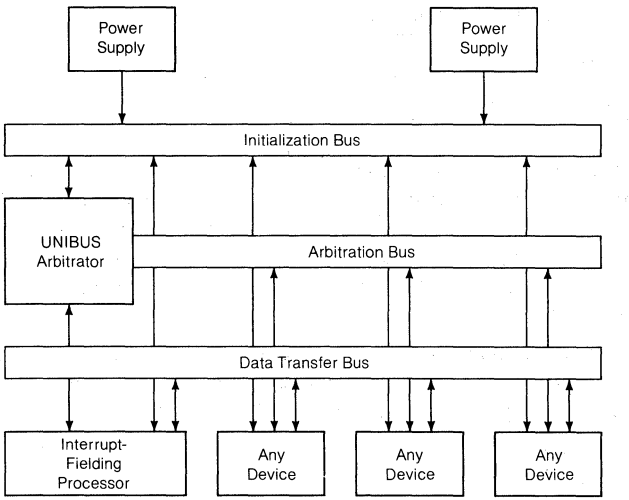


Figure 7-9 ▪ UNIBUS Block Diagram

▪ UNIBUS Data and Address Organization

Addresses

Addresses on the UNIBUS are 18 bits wide and represent the addresses of particular bytes. The bits are numbered as shown in Figure 7-10—address bit <00> is the least significant bit (LSB) and address bit <17> is the most significant bit (MSB).

Words are addressed on even-byte boundaries. That is, address bit <00> is always equal to zero when accessing a word.

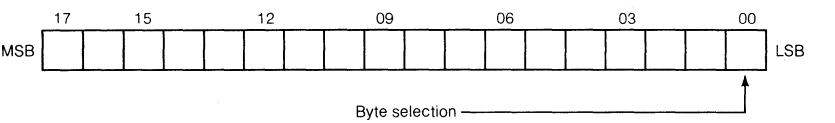


Figure 7-10 ▪ UNIBUS Address Format

Data

Data words on the UNIBUS are 16 bits wide. The bits are numbered as illustrated in Figure 7-11. Bit <00> is the LSB and bit <15> is the MSB. During word operations, address bit <00> must equal zero.

Each data word may also be construed as two data bytes. See Figure 7-12. Data on the UNIBUS is never *justified*. That is, data bits <07:00> always

form the even bytes (bytes with address bit <00> = 0) and data bits <15:08> always form the odd bytes (bytes with address bit <00> = 1). When 2 bytes are construed as a word, the even byte forms the low-order bits of the word and the odd byte forms the high-order bits of the word.

The UNIBUS does not define a byte-read operation. Reads always produce a word (two bytes) of data, and it is the responsibility of the master to select the correct byte, based on address bit <00>. To read from an even byte, data must be taken from the lines representing word bits <07:00>. To read from an odd byte, data must be taken from the lines representing word bits <15:08>. For read operations, address bit <00> is never actually asserted on the bus.

The UNIBUS does define a byte-write operation. Here, it is the responsibility of the slave to update the selected byte of the data word (based on address bit <00>) while preserving the unselected byte. The master must still drive data on the correct byte of the UNIBUS, and must now correctly assert address bit <00>. In fact, the master's logic can be simplified by driving the same data onto both bytes of the UNIBUS simultaneously. The slave will select the correct byte and ignore the other byte.

Not all slave devices implement the byte-write operation for all addresses. Some addresses can be written only as entire words, and an attempt to write to a byte updates the entire word, using whatever data happens to be on the other byte of the UNIBUS data lines. This is generally true of I/O device registers. Memories almost always support byte-write operations.

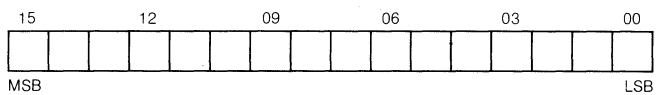


Figure 7-11 ▪ UNIBUS Data as a Word

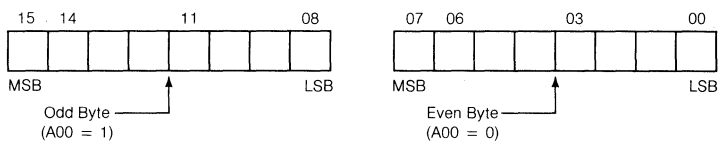


Figure 7-12 ▪ UNIBUS Data as Two Bytes

▪ Types of UNIBUS Data Transfers

The previous sections introduced the three major sub-buses of the UNIBUS, the form of a UNIBUS address, and the two forms of UNIBUS data. The fol-

lowing sections examine in more detail the functions of the data transfer section.

Because the ultimate function of any bus is to move data from place to place, the data transfer bus can be viewed as the central, most important part of the UNIBUS.

The data transfer section implements five distinct types of operations:

- Read word.
- Write word.
- Write byte.
- Read word (with write intent).
- Write vector.

Read Word

The read word operation allows the current UNIBUS master to read one 16-bit word from a particular UNIBUS slave device. In UNIBUS terminology, this operation is called a DATI (data-in). The direction is always taken with respect to the master device. For example, a processor (the master in this case) may wish to read a word of data contained in main memory (the slave). See Figure 7-13.

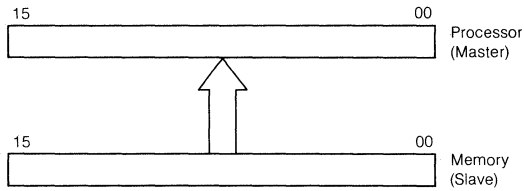


Figure 7-13 ▪ DATI Used to Read a Word

This same operation is also used when the master wishes to read a byte. In this case, the slave still presents an entire word (two bytes) and the master simply selects the correct byte. See Figure 7-14.

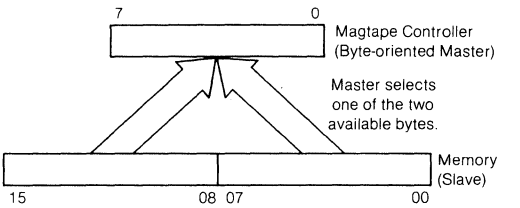


Figure 7-14 ▪ DATI Used to Read a Byte

Write Word

This operation allows the current UNIBUS master to write one 16-bit word from the master to a particular UNIBUS slave device. In UNIBUS terminology, this operation is called a DATO (data-out). The direction is once again taken with respect to the master device. In this example, a disk controller is writing into main memory. See Figure 7-15.

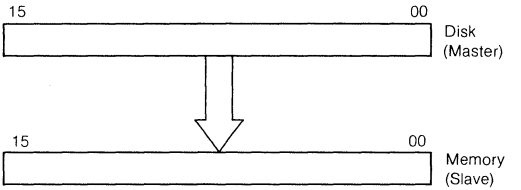


Figure 7-15 ▪ DATO Used to Write a Word)

DATO always writes an entire 16-bit word. DATO cannot write a single byte. The DATO must be directed at a word-address (an even byte address). In any case, most slave devices ignore address bit <00> during word operations. This forces the operation to an even address.

Write Byte

This operation allows the current UNIBUS master to write one 8-bit byte from the master to a particular UNIBUS slave device. In UNIBUS terminology, this operation is called a DATOB (data-out, byte). As usual, the direction ("Out") is taken with respect to the master device. See Figures 7-16 and 7-17.

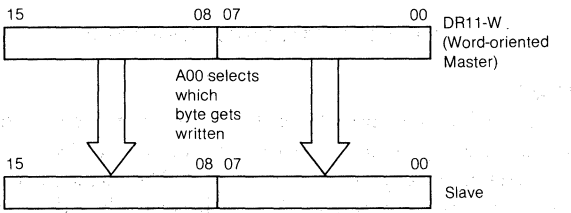


Figure 7-16 ▪ DATOB (Write Byte) for a Word-oriented Master

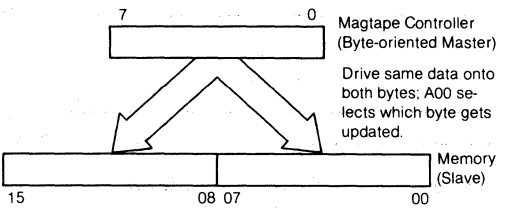


Figure 7-17 ▪ DATOB (Write Byte) for a Byte-oriented Master

DATOB attempts to write a single 8-bit byte. The other byte is to be left unmodified.

Read Word with Write Intent

Like DATI, this operation also allows the current UNIBUS master to read one 16-bit word from a particular UNIBUS slave device. In addition, the slave is informed that a write (DATO/DATOB back to the same address) will soon follow. In UNIBUS terminology, this cycle is called a DATIP (Data-In, Pause).

This cycle originated in the days of magnetic core memory. Reading data from a core memory is a destructive process. That is, the very process of reading the data from the magnetic cores destroys the data, clearing it to zeros (see Figure 7-18). If the core memory location is to retain its data, the read/ destroy operation must immediately be followed by a restore operation to write the data back into the cores.

However, when the master device knows that some form of read/modify/write operation is being performed (for example, an *increment* instruction), there is no need to restore the old data to the cores. In fact, core memory requires that the cores be cleared to zero prior to writing new data into them. Rather than allow the automatic restoration of the data, the master indicates that it will shortly supply new data and that the core memory should pause, rather than restore the old data. Hence the name of the cycle.

DATIP/DATO or DATIP/DATOB requires about half the time compared to DATI/DATO or DATI/DATOB. See Figure 7-19.

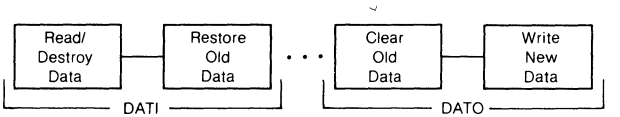


Figure 7-18 ▪ Separate Read and Write Cycles

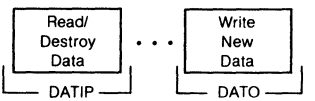


Figure 7-19 ▪ DATIP Followed by Write Cycle

Once the core memory has responded to the DATIP, it locks out all other operations except for the DATO/DATOB to the same location as read by the DATIP.

Write Vector

The final type of UNIBUS cycle has no mnemonic name. It is used by any master that desires to interrupt the interrupt-fielding processor (IFP). Figure 7-20 illustrates the case of a disk controller interrupting the IFP.

The master performs this cycle by using a special UNIBUS signal that always selects the IFP as the slave device and causes the master's data word to be written to the IFP. The IFP then takes the data word as the address of an interrupt service pointer (*vector*) and interrupts through that vector.

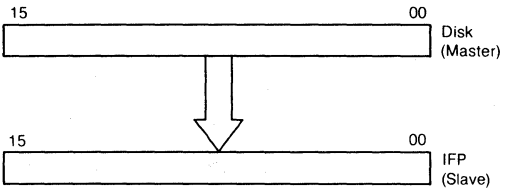


Figure 7-20 ▪ Interrupt Vector Write

By convention, the interrupt vector is in the range [000000 000774] but neither the UNIBUS hardware nor most IFPs impose any limit. If your master device passes 177774 as its vector, the IFP will try to interrupt through that vector.

Interrupt vectors are always located on even-word (quad-byte) boundaries. That is, they always take the form xxxxx0 or xxxxx4. IFPs will respond *unpredictably* to vectors of the form xxxxx2 or xxxxx6.

The UNIBUS actually consists of 56 signal lines and a number of ground (logic reference) lines. The individual lines are detailed in Table 7-1 and discussed in the subsequent paragraphs.

All lines on the bus employ open-collector drivers and resistive pull-ups.

Most of these signals are considered to be asserted (1) while near ground, and are considered to be de-asserted (0) while near 3.3 v. Signals of this type are indicated by the suffix L after the signal name. Any signal which is asserted while near 3.3 v is indicated by the suffix H. All timing diagrams are drawn with up meaning asserted, rather than any particular voltage level.

Three different kinds of termination are used, depending on the particular signal line. The signals BUS AC LO L and BUS DC LO L are terminated by a resistor-capacitor network. In the table, this is indicated as slow termination. The grant lines (BUS NPG H, BUS BGx H) only connect from one module to the next and use a simplified termination on each module. In the table, this is indicated as Grant termination. All other lines are terminated at each end of the UNIBUS into 120 ohms and 3.3 volts. This is indicated as fast termination.

The signals are partitioned into three groups which follow the same grouping as previously described:

- Initialization and shutdown signals.
- Arbitration signals.
- Data transfer signals.

In addition to those signals that are actually part of the UNIBUS and controlled by the UNIBUS specifications, there are other standard signals in most UNIBUS backplanes.

Table 7-1 ▪ UNIBUS Signals

Name	High/Low	Termination Type	Pin(s) In	
			UNIBUS Cable	SPC Backplane
▪ Initialization and Shutdown				
BUS DC LO L	L	Slow	BF2	CN1
BUS AC LO L	L	Slow	BF1	CV1
BUS INIT L	L	Fast	AA1	DL1
▪ Arbitration				
BUS NPR L	L	Fast	AS2	FJ1
BUS BR7 L	L	Fast	AT2	DD2
BUS BR6 L	L	Fast	AU2	DE2
BUS BR5 L	L	Fast	BC1	DF2
BUS BR4 L	L	Fast	BD2	DH2
BUS NPG H	H	Grant	AU1	In-CA1 Out-CB1
BUS BG7 H	H	Grant	AV1	In-DK2 Out-DL2
BUS BG6 H	H	Grant	BA1	In-DM2 Out-DN2
BUS BG5 H	H	Grant	BB1	In-DP2 Out-DR2
BUS BG4 H	H	Grant	BE2	In-DS2 Out-DT2
BUS SACK L	L	Fast	AR2	FT2
▪ Data Transfers				
BUS A00 L	L	Fast	BH2	EH2
BUS A01 L	L	Fast	BH1	EH1
BUS A02 L	L	Fast	BJ2	EF1

Table 7-1 ▪ UNIBUS Signals (Cont.)

Name	High/Low	Termination Type	Pin(s) In	
			UNIBUS Cable	SPC Backplane
BUS A03 L	L	Fast	BJ1	EV2
BUS A04 L	L	Fast	BK2	EU2
BUS A05 L	L	Fast	BK1	EV1
BUS A06 L	L	Fast	BL2	EU1
BUS A07 L	L	Fast	BL1	EP2
BUS A08 L	L	Fast	BM2	EN2
BUS A09 L	L	Fast	BM1	ER1
BUS A10 L	L	Fast	BN2	EP1
BUS A11 L	L	Fast	BN1	EL1
BUS A12 L	L	Fast	BP2	EC1
BUS A13 L	L	Fast	BP1	EK2
BUS A14 L	L	Fast	BR2	EK1
BUS A15 L	L	Fast	BR1	ED2
BUS A16 L	L	Fast	BS2	EE2
BUS A17 L	L	Fast	BS1	ED1
BUS C0 L	L	Fast	BU2	EJ2
BUS C1 L	L	Fast	BT2	EF2
BUS D00 L	L	Fast	AC1	CS2
BUS D01 L	L	Fast	AD2	CR2
BUS D02 L	L	Fast	AD1	CU2, FE2*

*For forwards compatibility, use the first pin rather than the second.

Table 7-1 ▪ UNIBUS Signals (Cont.)

Name	High/Low	Termination Type	Pin(s) In	
			UNIBUS Cable	SPC Backplane
BUS D03 L	L	Fast	AE2	CT2, FL1*
BUS D04 L	L	Fast	AE1	CN2, FN2*
BUS D05 L	L	Fast	AF2	CP2, FF1*
BUS D06 L	L	Fast	AF1	CV2, FF2*
BUS D07 L	L	Fast	AH2	CM2, FH1*
BUS D08 L	L	Fast	AH1	CL2, FK1*
BUS D09 L	L	Fast	AJ2	CK2
BUS D10 L	L	Fast	AJ1	CJ2
BUS D11 L	L	Fast	AK2	CH1
BUS D12 L	L	Fast	AK1	CH2
BUS D13 L	L	Fast	AL2	CF2
BUS D14 L	L	Fast	AL1	CE2
BUS D15 L	L	Fast	AM2	CD2
BUS PA L	L	Fast	AM1	CC1
BUS PB L	L	Fast	AN2	CS1
BUS BBSY L	L	Fast	AP2	FD1
BUS MSYN L	L	Fast	BV1	EE1
BUS INTR L	L	Fast	AB1	FM1
BUS SSYN L	L	Fast	BU1	EJ1, FC1*

Table 7-1 ▪ UNIBUS Signals

Name	High/Low	Termination Type	Pin(s) In	
			UNIBUS Cable	SPC Backplane
▪ Miscellaneous				
BOOT ENABL L	L	N/A	<none>	AR1
HALT REQ L	L	N/A	<none>	CP1
HALT GRANT L	L	N/A	<none>	CR1
LTC	N/A	N/A	<none>	CD1

Initialization and Shutdown Signals

BUS DC LO L	This signal, when asserted, indicates that somewhere on the UNIBUS the ac power has failed long enough that all other bus signals are not to be trusted. Each power supply in the system can drive BUS DC LO L. This signal may also be driven by any UNIBUS device which desires to restart the system as though power had just been applied. Network interfaces use this feature to cause down-line loading to begin.
BUS AC LO L	This signal, when asserted, indicates that somewhere on the UNIBUS, ac power has failed and the system will soon be inoperative. Upon the assertion of BUS AC LO L, the system processor will usually begin an orderly shutdown. Each power supply in the system can drive BUS AC LO L. Note that this signal (and BUS DC LO L, as well) is usually derived from each power supply's bulk dc source, not from direct measurements of the ac power line. This means that a lightly loaded power supply may not assert BUS AC LO L for quite a while after the actual line power fails. This helps the processor ride through line transients but means that a system that must actually monitor the quality of the ac power line must not depend on BUS AC LO L. Other devices in the system may become inoperative prior to the assertion of BUS AC LO L. This signal may also be driven by any UNIBUS device which would like to cause the system to perform its power-fail or power-recovery action. This is also used by network interfaces.
BUS INIT L	This signal, when asserted, indicates that all devices on the UNIBUS should reset themselves to the state they entered upon initial powerup. BUS INIT L is asserted each time the system is started, by the RESET instruction.

Powerup Timing

Powerup timing is quite variable. It depends on the specific power supplies and devices connected to the UNIBUS. A typical powerup sequence is shown in Figure 7-21. As ac power is applied to the system, the output from each power supply becomes stable. As this occurs, each power supply releases BUS DC LO L. When the last power supply (stabilizes and) releases BUS DC LO L,

BUS DC LO L is deasserted on the UNIBUS. At this point, certain devices (such as the CPU) begin their initialization.

As the 5 Vdc power supply voltage to the processor stabilizes, the processor will assert BUS INIT L. BUS INIT L will remain asserted for between 10 microseconds and 120 milliseconds. The assertion of BUS INIT L will initialize the remaining devices on the UNIBUS.

As with BUS DC LO L, each power supply also drives BUS AC LO L. When the last power supply releases BUS AC LO L, BUS AC LO L is deasserted on the UNIBUS. When BUS INIT L deasserts, the processor(s) begin monitoring BUS AC LO L (which very likely has already deasserted). Once the processor detects that BUS AC LO L has deasserted, the processor begins operating as directed by the user. This may mean that the operating system is bootstrapped, or it may mean that a recovery from a power failure is attempted. In any case, the system is now powered up.

In a system with a single power supply, once BUS AC LO L deasserts, it remains deasserted for a minimum of 2 milliseconds. The system guarantees that at least 2 milliseconds of time is available to recover from the previous power failure before the next power failure is signalled. In systems with more than one power supply, this guarantee cannot be made.

BOOT ENABL L is valid only at the instant that BUS DC LO L deasserts. Any device requiring this signal must latch it at this time.

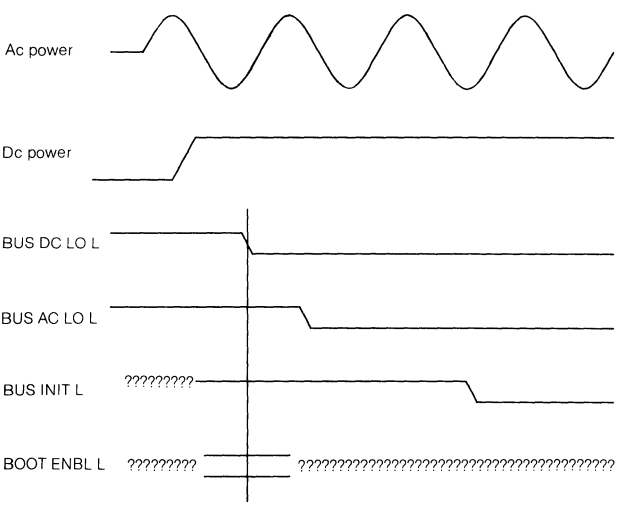


Figure 7-21 ▪ Typical Powerup Timing

Powerdown Timing

Powerdown timing is generally the reverse of powerup timing. It is also quite variable depending on the specific power supplies and devices connected to the UNIBUS. A typical powerdown sequence is shown in Figure 7-22.

Once ac power fails, the storage capacitors in the various power supplies begin discharging. As each power supply recognizes that it will soon have insufficient power to continue operation, it asserts BUS AC LO L. Effectively, BUS AC LO L is asserted by the first power supply to detect the failure. The CPU is notified by means of an interrupt that power is failing. A minimum of 2 milliseconds (5 milliseconds nominal) of runtime remains.

As each power supply discharges further, it asserts BUS DC LO L. Once BUS DC LO L has been asserted by any power supply, all activity in the system is halted. Each of the dc supply voltages actually remains valid for at least 5 milliseconds after the assertion of BUS DC LO L. BUS INIT L also asserts briefly when BUS DC LO L asserts.

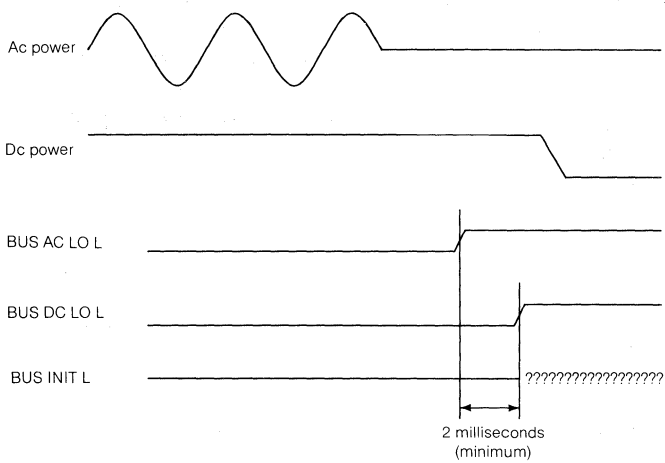


Figure 7-22 ▪ Typical Powerdown Timing

Initialization Timing

Any time the PDP-11/84 executes a RESET instruction, BUS INIT L will be asserted. This signal resets most devices to the state they assumed immediately after powerup.

The BUS INIT L signal is a pulse. Its normal duration is approximately 10 microseconds. Typical timing is illustrated in Figure 7-23. To allow power-fail routines to complete, BUS INIT L may be prematurely terminated by the assertion of BUS AC LO L. In this case, a minimum assertion of BUS INIT L for approximately 1 microsecond is guaranteed.

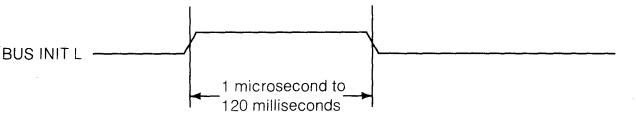


Figure 7-23 ▪ UNIBUS Initialization Timing

Arbitration Signals

BUS BR4 L BUS BR5 L BUS BR6 L BUS BR7 L	These signals request that the interrupt fielding processor (IFP) usually the central processor) be interrupted at priority level 4, 5, 6, or 7, respectively. Priority 7 is the highest priority. If the IFP is running below the priority of the request, the interrupt will be granted at the end of the current instruction. If the IFP is running at or above the priority of the request, the request will be deferred until the processor priority is reduced.
BUS BG4 H BUS BG5 H BUS BG6 H BUS BG7 H	These signals indicate that the IFP is willing to grant the interrupt at priority level 4, 5, 6, or 7, respectively. The interrupting device must now become bus master and pass its interrupt service vector address to the IFP. **Note** These signals are asserted high. They are also not used as wired-OR signals although they are driven with the same type of integrated circuits as all other lines. They are terminated in a slightly different manner than other UNIBUS signals.
BUS NPR L	This signal is used to request use of the UNIBUS (without causing an interrupt of the IFP). This signal is analagous to the DMA request signal defined by other buses.
BUS NPG H	This signal indicates that the bus arbitrator is willing to allow use of the UNIBUS by one of the devices asserting BUS NPR L. **Note** This signal, like the BUS BGx H signals, is asserted high. It is also not used as a wired-OR signal although it is driven with the same type of integrated circuits as all other lines. Also like the BUS BGx H signals, it is terminated in a slightly different manner than other UNIBUS signals.
BUS SACK L	This signal is asserted by a device that has accepted a grant (whether a BG or NPG). This signal means selection acknowledged. As long as BUS SACK L is asserted, further arbitration is stopped. This assures that only one device is designated as the next master of the UNIBUS.

UNIBUS Arbitration Timing

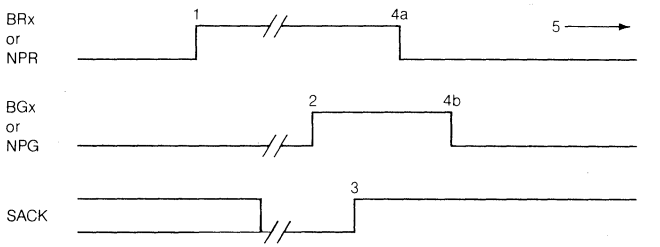


Figure 7-24 ▪ Arbitration Timing

The sequence a device uses to arbitrate for use of the UNIBUS data transfer bus is shown in Figure 7-24 and described as follows:

- Assert Request—The device, wishing to become the bus master, asserts a request line. This request may be ORed with other requests already present on the BRx/NPR line.
- Receive Grant—When the device's NPR or BRx is the highest-priority request presented to the bus arbitrator, the arbitrator will issue NPG, or, with authorization from the IFP, BGx. If no higher priority device at level BRx or NPR wants the grant, the grant will be propagated to our device. Our device will receive the grant and not propagate it (block the grant).
- Assert SACK—As soon as the device sees the grant, it asserts BUS SACK L to indicate that it acknowledges its selection as the next bus master.
- Remove Request—Once BUS SACK L has been asserted, the device may remove its request.
- Removal of the Grant—Once the arbitrator sees the device's assertion of BUS SACK L, the arbitrator will remove the grant. No further grants will be issued to anyone as long as the device holds BUS SACK L.
- Data Cycles—One or more data cycles follow, and are described later.

Abnormal Cycles

Five abnormal arbitration cycles exist:

- Grant refusal.
- No-SACK timeout.
- Passive release.

- Stolen grant.
- Borrowed grant.

- *Grant Refusal*

  After making a request and being given a grant, the device may choose not to take the grant. In this case, the grant passes through the device. If another device further down the bus is waiting for the grant, the arbitration cycle continues normally (from the point of view of the device down the bus). If no device on a particular BRx/NPR level takes the grant, the grant reaches the bus terminator. Some UNIBUS terminators do nothing with the grant—this leads to the no-SACK timeout described below. Other UNIBUS terminators (notably the M9302) take all grants and reply with BUS SACK L. Then, once the grant is removed by the arbitrator, BUS SACK L will be deasserted by the terminator and a passive release occurs.

- *No-SACK Timeout*

  The arbitrator issues only one grant at a time, and the UNIBUS is considered to be busy while a grant is being issued. This means that if a grant is issued, and no one asserts BUS SACK L, the system will hang waiting for its assertion. This situation often occurs if noise is being coupled onto the request lines. In this case, the arbitrator issues a grant in response to the noise but there are no genuine takers on the UNIBUS. The PDP-11/84 therefore contains a feature called no-SACK timeout. If BUS SACK L is not received within 10-25 microseconds of issuing a grant, the arbitrator removes the grant and a passive release occurs. This prevents the system from hanging while waiting for a grant.

- *Passive Release*

  At any time, the device selected to be the next bus master can say "never mind" just by releasing BUS SACK L. Because no actual data cycles occurred, this is referred to as a passive release of the UNIBUS. Passive releases are harmless but should be avoided because they waste time.

- *Stolen Grant*

  The UNIBUS treats NPR as higher in priority than any of the BRx interrupt requests. However, once any grant is issued, the master granted use of the UNIBUS is allowed to take as much time as needed.

  Certain older DMA peripheral devices (for example, TC11 and RK11-C) contain very little buffering, and must be granted immediate use of the UNIBUS once they require that a word be transferred. If the arbitrator has just issued a BG to an interrupting device, the total time required for the processor (the IFP) to get the vector, save the PC and PS, and load the PC and PS from the

vector was too long for the DMA device to wait, and a data-late error occurred in the device.

To alleviate this problem, some old interrupting devices incorporate a feature called steal grant. If a BUS BGx H is propagating down the UNIBUS and reaches one of these devices, the device examines BUS NPR L. If BUS NPR L is asserted (meaning that an NPR device is waiting), the device blocks the BGx grant, and asserts BUS SACK L. In effect, it has stolen the BGx intended for some device on the UNIBUS. Once the arbitrator sees the assertion of BUS SACK L, it deasserts BUS BGx H. The stealing device then deasserts BUS SACK L (a passive release). The arbitrator re-arbitrates and now issues the desired BUS NPG H in response to BUS NPR L.

As processors became more flexible about granting NPRs, and DMA devices have come to contain more buffering, this feature has fallen into disuse. On most Digital modules, the feature is controlled by a jumper named N1.

- *Borrowed Grant*

While an interrupting device can never issue BUS INTR L while using the bus under the authority granted by BUS NPG H, it is perfectly legal for a device to become bus master under the authority of BUS BGx H and then do a number of DMA bus cycles prior to performing the vector write cycle. Stretching this just a little suggests that it is perfectly acceptable for a DMA device to borrow the BUS BGx H intended for an interrupt device, do a few bus cycles, and then pass BUS BGx H on to the interrupting device. The interrupting device generally won't know that its authority to become next bus master has been borrowed. The only possible UNIBUS failure that can occur is a no-SACK timeout should the bad guy DMA device keep the grant too long.

Data Transfer Signals

BUS BBSY L	Also known as BUS BUSY L, this signal indicates that UNIBUS data transfer bus is occupied (by the current bus master). The next bus master monitors this signal to determine when it can seize the data transfer bus.
BUS A17 L through BUS A00 L	These 18 signals are referred to as the UNIBUS address lines and supply the address that selects a slave device. BUS A00 L is the least significant bit and is ignored for all cycles except DATOB (write byte).
BUS C1 L BUS C0 L	These two signals are referred to as the UNIBUS control lines and designate four out of the five types of operations of the data transfer bus. The operations are described in detail later in this chapter.

C1	C0	Cycle Type
0	0	DATI (Read word)
0	1	DATIP (Read with intent to modify)
1	0	DATO (Write word)
1	1	DATOB (Write byte)
X	X	VECTOR WRITE (see BUS INTR L)

BUS D15 L through BUS D00 L	These 16 signals are referred to as the UNIBUS data lines and transmit the data word. BUS D00 L is the least significant bit. Byte data is *not always transmitted on* D07-D00. While the data for bytes at even addresses (A00 = 0) is transmitted on D07-D00, data for bytes at odd addresses (A00 = 1) is transmitted on D15-D08. This generally simplifies the design of word-oriented devices that may also read and write bytes.
BUS PA L BUS PB L	These two signals are used to convey parity error information between the master and slave devices (for example, memories assert BUS PB L to indicate that a parity error was detected within the memory).

PA	PB	Status
0	0	Good parity detected within slave.
0	1	Bad parity detected within slave.
1	X	Master should ignore PA (parity detection is disabled by someone on the UNIBUS). This combination is not used by Digital.
BUS MSYN L		This signal serves two purposes: As the UNIBUS address strobe, it indicates that the UNIBUS address lines contain a valid address and that BUS C1 L and BUS C0 L contain a valid function code. For write word and write byte operations (DATO and DATOB), BUS MSYN L also serves as the data strobe, indicating that the master has placed valid data on the UNIBUS data lines.
BUS SSYN L		This signal also serves two functions: As the UNIBUS address acknowledgment, it serves to inform the bus master that the requested address exists within some slave. For read operations (DATI and DATIP), BUS SSYN L also serves as the data strobe, indicating that the slave has placed valid data on the UNIBUS data lines and valid parity status information on BUS PA L and BUS PB L.

BUS INTR L	This signal provides for a fifth kind of UNIBUS data cycle known as WRITE VECTOR. BUS INTR L acts simultaneously as the address, the address strobe, and the data strobe. The IFP is always the selected slave. The contents of the UNIBUS address lines, BUS C1 L and BUS C0 L, are ignored. The contents of the UNIBUS data lines are accepted by the IFP as an interrupt vector. The IFP then returns BUS SSYN L, just like any other bus slave.
Strobe	This is not a UNIBUS signal. It is typical of signals that exist within real devices. The signal is shown as a pulse indicating the moment that data should be latched into a device's flip-flops.

DATI Timing (Master Reads from Slave)

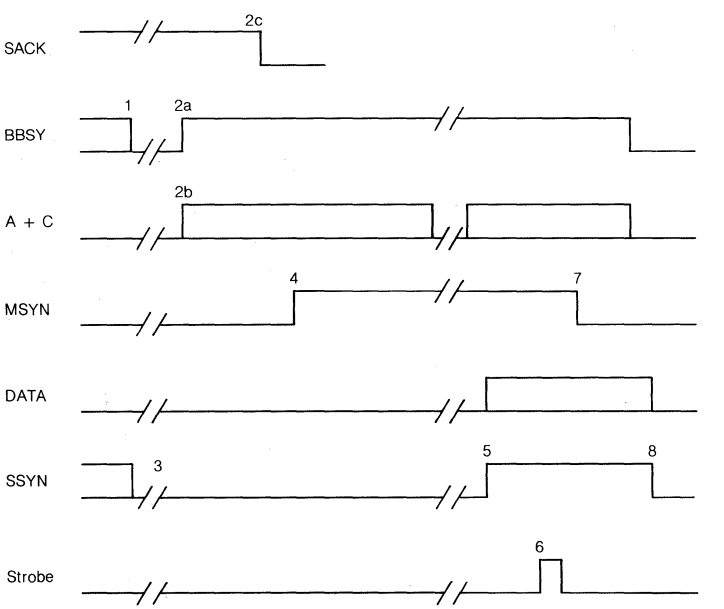


Figure 7-25 ▪ DATI/DATIP Timing

Assuming that the device has won the priority arbitration and asserted BUS SACK L, it can now proceed to observe a data transfer cycle. Refer to Figure 7-25.

1. Receive the deassertion of the previous BBSY—The device begins monitoring the BUS BUSY L line and waits until it sees BUS BBSY L unasserted.

2a. Assert BBSY—As soon as the device sees that BUS BBSY L is unasserted (indicating that the previous master is no longer using the data section of the UNIBUS), the device asserts BUS BBSY L. It is now the bus master.

2b. Assert Address and Control—At the same time that it drives BUS BBSY L, it may also drive the UNIBUS address lines, BUS C1 L, and BUS C0 L. All other devices on the bus immediately begin decoding this information to determine whether they will be the selected slave. A minimum of 150 nanoseconds must elapse prior to the assertion of BUS MSYN L. This time period allows the addressing information time to reach all points on the bus and be decoded. It is referred to as the front end deskew.

2c. Deassert SACK—Once the device asserts BUS BBSY L, it has become the bus master. If it is only transferring a single word, our device can now allow arbitration to resume. This is done by deasserting BUS SACK L.

3. Receive the deassertion of the previous SSYN—Any device on the UNIBUS is allowed to stall the onset of the next bus cycle by holding BUS SSYN L asserted. The new bus master must ensure that BUS SSYN L is deasserted prior to proceeding. See paragraph 8.

4. Assert MSYN—Once the 150 nanoseconds of front-end deskew time has elapsed, and any previous BUS SSYN L has been deasserted, the bus master may now assert BUS MSYN L. This informs everyone on the bus that the addressing information is valid.

5. Receive the assertion of SSYN—The master device now waits for the selected slave to place the requested data onto the UNIBUS data lines (along with parity information on BUS PA L and BUS PB L). When the slave does this, it also asserts BUS SSYN L.

6. Deskew the Data—The master must now allow 75 nanoseconds from the receipt of BUS SSYN L to guarantee that the returned data has all arrived at the master. This is referred to as the data deskew. Once this time has elapsed, the master may strobe in the returned data. Note that Strobe is not a bus signal but rather a typical signal within the master device.

7. Deassert MSYN—Once the master has captured the returned data, BUS MSYN L may be removed. The master must continue to hold the addressing information valid for at least 75 nanoseconds to ensure that all devices are notified that the address is invalid prior to it actually becoming invalid. This period is referred to as tail end deskew.
8. Receive the deassertion of SSYN Once the selected slave sees BUS MSYN L deassert, the slave will eventually deassert BUS SSYN L. The slave is completely within its rights to hold BUS SSYN L asserted for as long as necessary to complete whatever operations it must. For hard-wired devices, BUS SSYN L usually is deasserted very quickly, but for microcoded devices, there may be a delay. Other devices (which are neither the current master nor slave) may also stretch BUS SSYN L once it appears. This capability allows cache memories and bus monitors adequate time to perform their business.

DATIP (Master Reads from Slave, No Restore)

DATIP (Data-In, Pause) is used when the master wishes to read a value from the slave but also expects to write back a new value.

DATIP differs from DATI only in the fact that the control line BUS C0 L is asserted. Most masters also keep BUS SACK L and BUS BBSY L asserted between the DATIP and the subsequent DATO or DATOB. This saves the time that would be spent rearbitrating for the UNIBUS. In the case of interlocked slaves (like the core memory), this also prevents other devices from becoming bus master and subsequently from being frustrated as they try to access the unresponsive memory.

Most I/O devices treat DATIP identically to DATI.

DATO/DATOB (Master Writes to Slave)

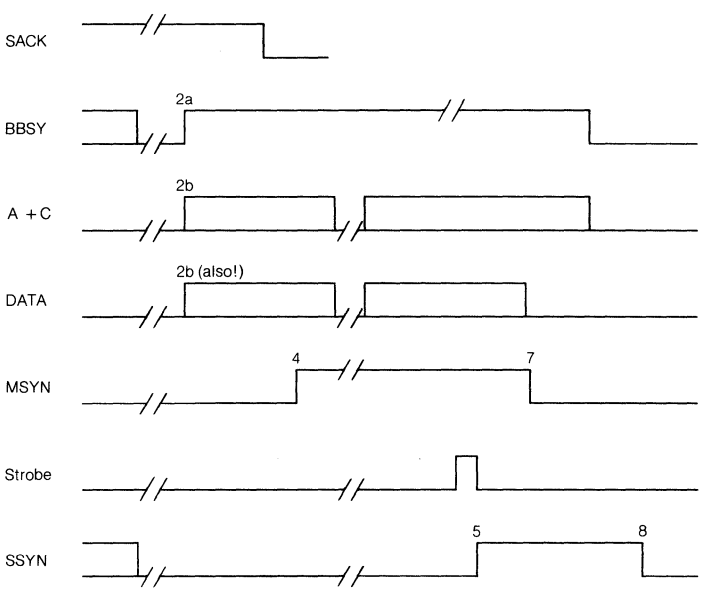


Figure 7-26 ▪ DATO/DATOB Cycles

In many respects, a DATO/DATOB cycle is similar to a DATI cycle. The principal difference is that the master supplies the data to the slave. Please refer to Figure 7-26. Only the following differences between a DATI/DATIP and a DATO/DATOB are documented. All other comments from DATI/DATIP apply to DATO/DATOB as well.

2b. Assert Address, Control, and Data—During a write cycle, the master asserts not only UNIBUS address and control information, but data as well. Note that the 150 nanoseconds of front end deskew is more than adequate to cover the 75 nanoseconds of required data deskew time.

4. Assert MSYN—Once the 150 nanaoseconds of front end deskew have elapsed and any previous BUS SSYN L has deasserted, the master may now assert BUS MSYN L. This informs everyone on the bus that the addressing information and data are valid.

5. Receive the assertion of SSYN—The master device now waits for the selected slave to accept the data it has presented on the UNIBUS data lines. When the slave does this, the slave will assert BUS SSYN L.

7. Deassert MSYN—Upon receiving the returned BUS SSYN L, the master removes BUS MSYN L. The master must hold the addressing information valid for 75 nanoseconds to ensure that all devices are notified that the address is invalid prior to it actually becoming invalid (tail end deskew). The UNIBUS specifications do not require that the data also be held, but it helps add margin to the system.

Write Vector Timing (Master Interrupts IFP)

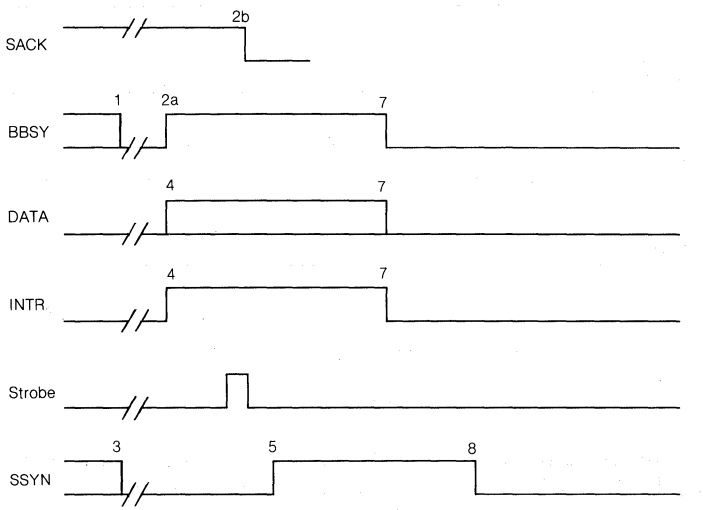


Figure 7-27 ▪ Write Vector Cycle

For the write vector operation, BUS INTR L replaces the functions of the UNIBUS address and control lines and BUS MSYN L. Please refer to Figure 7-27.

1. Receive the deassertion of the previous BBSY—The device begins monitoring the BUS BUSY L line and waits until it sees BUS BBSY L unasserted.
2. Assert BBSY—As soon as the device sees that the previous master is no longer using the data section of the UNIBUS, it asserts BUS BBSY L. The device is now the bus master.

2c. Deassert SACK—Once the device asserts BUS BBSY L, it has become the bus master and can now allow arbitration to resume. This is done by deasserting BUS SACK L.

3. Receive the deassertion of the previous SSYN—Any device on the UNIBUS is allowed to stall the onset of the next bus cycle by holding BUS SSYN L asserted. The new bus master must ensure that BUS SSYN L is deasserted prior to proceeding.
4. Assert D15-D00 and INTR—Once BUS SSYN L is deasserted, the device can immediately drive its interrupt vector onto the UNIBUS data lines and *simultaneously* drive BUS INTR L. This is *the one case where the master is not responsible for the deskewing.*
5. Receive the assertion of SSYN—The IFP, acting as the slave device, receives BUS INTR L and waits 75 nanoseconds to allow for the deskewing of the data (the interrupt vector). The IFP then strobes in the vector and returns BUS SSYN L. Meanwhile, as in any write cycle, the master device has been waiting for the slave to assert BUS SSYN L. Most masters do not contain the logic to timeout if the IFP fails to return BUS SSYN L.
7. Deassert D15-D00, INTR, and BBSY—After the IFP asserts BUS SSYN L, the master may deassert BUS INTR L and remove the vector from the UNIBUS data lines. Note that the IFP must really have meant it when it returned BUS SSYN L because the master is not obliged to provide any tail end deskew of the vector data.
8. Receive the deassertion of SSYN-Once the IFP sees BUS INTR L deassert, it will eventually deassert BUS SSYN L.

Abnormal Bus Cycles

Aside from the abnormalities of arbitration, only one exception can occur.

- *SSYN Timeout*

When an address is placed on the bus, and BUS MSYN L is asserted, the master device begins waiting for BUS SSYN L to be asserted by some slave. If no device exists at the specified address, BUS SSYN L will never be asserted. As described thus far, this situation would hang the bus.

To avoid this problem, it is the responsibility of the current bus master to monitor the amount of time that it has spent waiting for the assertion of BUS SSYN L. If too much time expires, the master must end the current bus cycle (using valid tail end timing). The bus master may then handle the error in whatever manner the designer sees fit. The PDP-11/84 processor traps through the vector at 4 when a UNIBUS timeout error occurs. Most I/O devices set a status bit and begin an error interrupt.

The SSYN timeout value varies from device to device but generally ranges from 10 to 25 microseconds. If the system uses multiported memory, it may be necessary to increase the timeout value. If a one-shot is used as the SSYN

timer, you can simply increase the size of the capacitor. If a counter is used to provide the timeout, a variety of settings should be provided.

The unqualified phrase "bus timeout" usually refers to the specific case of SSYN timeout (rather than any of the other timeouts discussed in this chapter).

Multiple Bus Cycles (Burst Mode)

Once a device has gained UNIBUS mastership, it may do a single bus cycle and then release the UNIBUS, or it may do a series of UNIBUS data cycles while holding BUS BBSY L constantly asserted. Remember, once you gain ownership of the UNIBUS, it is yours without limit until you release BUS BBSY L.

If the device gained control of the UNIBUS under the authority of a BGx, it may perform a write vector as its last cycle.

Performing multiple data cycles during one UNIBUS mastership is generally referred to as *burst mode*. Usually, a small number (1, 2, 4, and sometimes 8 or 16) of data cycles are performed. If the device performs an unlimited number of data cycles, it is referred to as a BUS HOG because this prevents other devices (including the CPU) from using the UNIBUS.

When a device performs multiple data cycles, it is usually best to keep BUS SACK L asserted until the last data cycle is underway. This ensures the fairest arbitration of the UNIBUS (because the priority arbitration will be made based on the most recent requests, not those many microseconds old). Refer to Figure 7-28.

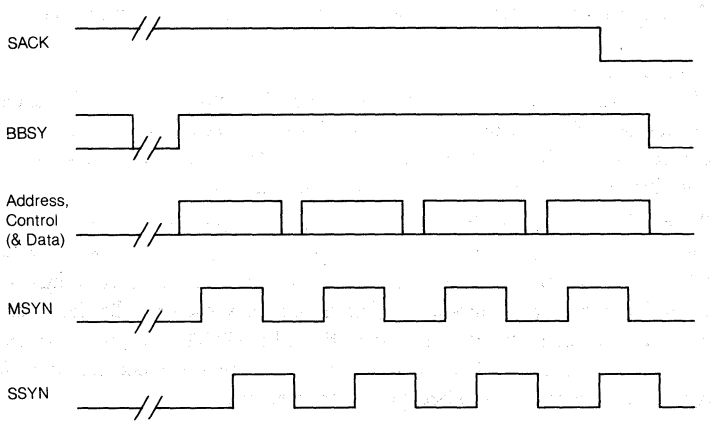


Figure 7-28 ▪ Burst Mode (Showing Four-Cycle Burst)

Miscellaneous Signals

These signals, though not part of the UNIBUS, are sometimes used by UNIBUS devices. They are present in most but not all UNIBUS backplanes.

BOOT ENABL L	This signal is derived from the power supply and indicates whether the system can reasonably expect the contents of a volatile memory (for example, MOS) to be valid. If so (for example, a battery backup unit maintained the contents of memory through a power failure), the signal is unasserted. However, if the contents of the volatile memory are not to be trusted (for example, the battery went dead), then the signal is asserted, and the central processor is thus made aware that the operating software must be reloaded (bootstrapped). BOOT ENABL L is valid only at the instant that BUS DC LO L deasserts.
HALT REQ L	This signal appears in the CPU backplanes of the PDP-11/04, -11/24, -11/34, and -11/44. When asserted, it causes the processor to halt at the completion of the current instruction. *This signal is not available in any other backplane in the system.*
HALT GRANT L	This signal is issued by the PDP-11/04, -11/24, -11/34, and -11/44 when the CPU halts in response to HALT REQ L. *This signal is not available in any other backplane in the system.*
LTC	This signal is a square wave driven from the power supply at the ac power frequency (60 Hz or 50 Hz). Because this signal has a nominal 50 percent duty cycle, and is not related in time to any other bus signal, it has no polarity indicator and may be construed as either logic high or logic low. LTC normally has slow rise and fall times (requiring the use of Schmitt-trigger receivers) and cannot drive as many receivers as other bus signals.

▪ UNIBUS Electrical Characteristics

The UNIBUS has the following major electrical characteristics:

- 56 signal lines, combined with grounds as necessary.
- 120-ohm impedance.
- Open-collector drivers.
- Low-leakage receivers.

- Termination at each end.
  — Thevinen voltage = 3.3 volts
  Thevinen resistance = 120 ohms

The 56 signal lines have already been described. Electrically, they fall into three major classes:

- Electrically bidirectional high-speed lines.
- Electrically unidirectional high-speed lines.
- BUS AC LO L and BUS DC LO L.

Electrically Bidirectional High-speed Lines

Most of the lines on the UNIBUS are electrically bidirectional, although they may be logically unidirectional (for example, BRx and NPR only carry signals *to* the bus arbitrator). A typical line is illustrated in Figure 7-29. All these lines:

- Are terminated at each end of the UNIBUS into 120 ohms.
  — 180 ohms to + 5, 390 ohms to ground
- Contain at least one open-collector driver.
- Contain at least one low-leakage receiver.
- Are asserted by pulling the line to ground (logic low lines).

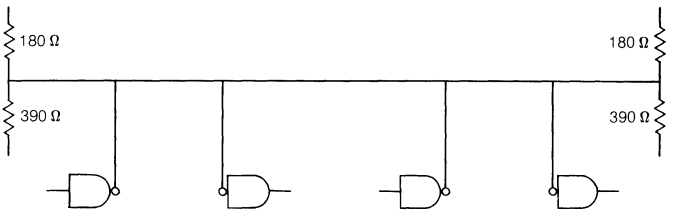


Figure 7-29 ▪ Electrically Bidirectional Line

Electrically Unidirectional High-speed Lines

The UNIBUS grant lines (BUS NPG H, BG7 H, BG6 H, BG5 H, and BG4 H) are electrically unidirectional (BGx and NPG carry only signals *from* the bus arbitrator). A typical segment of a grant line is illustrated in Figure 7-30. All these lines:

- Are daisy chained from module to module.
- Are terminated at each module in the daisy chain with a simplified terminator.
  — Toward arbitrator—180 ohms to + 5
  Away from arbitrator—180 ohms to + 5, 390 ohms to ground
- Contain only one open-collector driver.
- Contain only one low-leakage receiver.
- Are *deasserted* by pulling the line to ground and asserted by allowing the line to float (logic high lines).

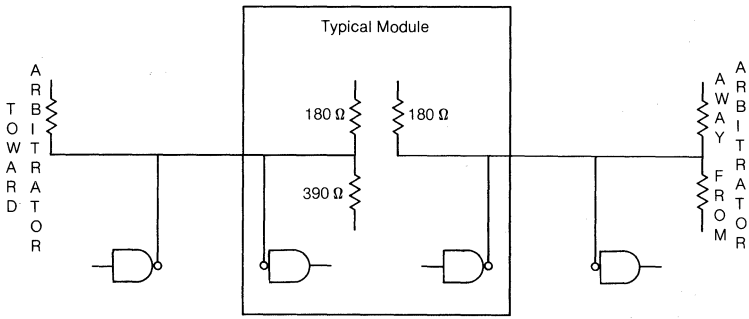


Figure 7-30 ▪ Electrically Unidirectional Line

BUS AC LO L and BUS DC LO L

BUS AC LO L and BUS DC LO L resemble the bidirectional lines. However, connections from these two lines are also carried in wiring harnesses to each of the system's power supplies. This requires slow-down filtering to prevent noise from disrupting the operation of the system. Each power supply must drive the lines with some form of normally-on drivers, so that the lines will be held grounded while the power supply is de-energized. This is often done with a depletion-mode FET (field effect transistor). Other devices use bias from the termination resistors to activate their drivers. A typical line is illustrated in Figure 7-31. Both BUS AC LO L and BUS DC LO L:

- Are terminated at each end of the UNIBUS with a slow terminator
  — 390 ohms to + 5
  1000 picofarads to ground
- Contain at least one open-collector or open-drain driver.
- Contain at least one low-leakage receiver.
- Are asserted by pulling the line to ground (logic low lines).
- Must remain asserted (grounded) when no other power is applied to the system.

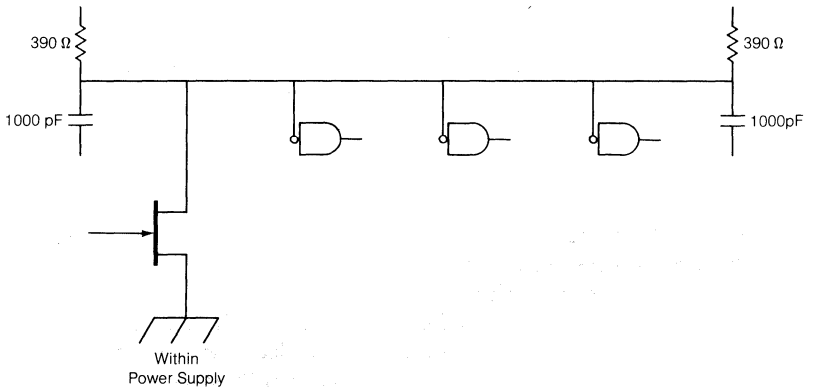


Figure 7-31 ▪ BUS AC LO L or BUS DC LO L

▪ Design Suggestions

Some suggestions for UNIBUS designers are presented below. While not comprising an exhaustive list, these suggestions address the more common errors made in UNIBUS device design.

dc Voltage Levels

The UNIBUS depends on its voltage levels. These voltage levels can be maintained only if the driver and receivers operate optimally, the cable has a controlled resistance, and the ground reference at all points in the system is accurate.

- Use the correct drivers and receivers.
  — Drivers must be able to pull down at least 60 milliamperes with a small voltage (0.6 V or less).
  Drivers and receivers must have little leakage current.
  At the time of this printing, the most recent UNIBUS receivers are Digital part number 19-14987.

- Establish a solid ground reference throughout the system. No driver or receiver can cope with more than a few millivolts of ground potential.

- To minimize voltage drop, keep dc resistances low.

- Limit any one UNIBUS segment to a total length of 15.25 meters (50 feet), including all cable segments, backplanes, and the hidden cable in the M9202 bus jumper modules. A segment of the UNIBUS is defined as that portion of the bus between the front- and back-end terminators (A UNIBUS can have more than a single segment).

- Limit any one UNIBUS segment to a total of 20 dc loads. A dc load is defined as one driver and one receiver, or one transceiver.

- If the UNIBUS must extend beyond 15.25 meters, or contain more than 20 dc loads, the one logical UNIBUS must be broken up into more than one UNIBUS segment, with a UNIBUS repeater between each segment. The UNIBUS repeater contains logic which amplifies the UNIBUS signals, making possible greater length and more bus loads, at the expense of transmission speed. Each of the individual UNIBUS segments between bus repeaters has its own termination.

120-ohm Impedance and ac Signals

The UNIBUS can operate correctly only if its signals are propagated quickly and cleanly from end to end. Signals propagate quickly and cleanly along a line only when the impedance of the line remains constant. Any time the signal encounters a different impedance, a small part of the signal is reflected. These reflections can accumulate and cause errors in the signal.

Each time a branch (or STUB) is added to the bus (for example, a module connected), capacitance is added to the bus. This additional capacitance is referred to as an ac load, because the capacitance affects only ac signals, not dc voltage levels. The capacitance also changes the local impedance of the UNIBUS.

If many modules are connected in a short length of the bus, these many small changes of bus impedance add up to a large change, and a large reflection can be generated. These many small capacitances lumped together are referred to as a lumped load.

- Minimize the distance between the UNIBUS and the drivers and receivers. Minimize the capacitance of the connecting etch run or wire.

- Keep lumped loads small and separate them with 120-ohm bus cable (as in the M9202 2-foot cable in a 1-inch module) or 120-ohm twisted pair (as in the DD11-DK).

- Terminate with accurate, pure resistances and accurate voltages.

Cross-talk Minimization

- Use drivers that generate edges with controlled rise and fall times.
- Be cautious in routing UNIBUS signals. Route the critical timing signals (at a minimum, BUS SACK L, BUS BBSY L, BUS MSYN L, BUS INTR L, and BUS SSYN L) away from the address and data lines.
- Do not allow any signals to glitch, *whether or not* such a glitch violates any timing specification. Glitches cross-talk much more readily than clean signals.
- Use twisted pair with an impedance of 120 ohms to route critical timing signals in backplanes.
- Maintain the 120-ohm impedance of the UNIBUS.

Consideration of All Timing Cases

Design your logic explicitly considering all of the UNIBUS operations, whether you intend to use them or not. For example, whether or not your device honors DATOB, consider what will happen if your device is requested to perform a DATOB. Do not ignore the case.

Remember to consider all of the data deskew and setup times. For devices acting as UNIBUS masters, the most important considerations are:

- The master must assert the UNIBUS address at least 150 nanoseconds prior to asserting BUS MSYN L (frontend deskew time).
- For read operations, the master must not strobe in the slave's data until at least 75 nanoseconds after the reception of BUS SSYN L at the master (data deskew time for reads).
- For write word and write byte, the master must assert the UNIBUS data at least 75 nanoseconds prior to asserting BUS MSYN L (data deskew time for writes).
- For write vector, the master must assert the UNIBUS data (the vector) at or before the assertion of BUS INTR L. This is the one case where the master is not responsible for the data deskew. Here, the IFP performs the deskew.
- The master must hold the UNIBUS address valid for at least 75 nanoseconds after deasserting BUS MSYN L (tail end deskew).

Design is considerably easier for devices acting as UNIBUS slaves, since the master does most of the deskewing. The major considerations for slaves are:

- For read operations, do not assert BUS SSYN L prior to asserting data.
- For write operations, do not assert BUS SSYN L prior to latching in the data.
- *Never use the falling edge of BUS MSYN L to latch the data.* Do not use transparent latches to capture the data unless the latches are closed prior to or simultaneously with the assertion of BUS SSYN L.

Remember that the UNIBUS is asynchronous and requires careful attention to set up and hold times at flip-flops. Be particularly careful to ensure that all synchronizer circuits account for the possibility of metastability. This is particularly important around the logic that decides whether to pass or block UNIBUS grants, and around circuits that synchronize BUS SSYN L for use by synchronous logic.

Good Bus Citizenship

Design for speed—time a device can save on the UNIBUS increases the system's overall throughput.

- A device should be ready to immediately read or write data once it becomes the bus master.
- During a device's last (and possibly only) data cycle, drop BUS SACK L so that bus arbitration may resume.

If a device transfers many words at great speed (perhaps 100,000 words or more per second):

- Within the device, provide enough FIFO (First-In, First-Out) buffering to avoid data late errors should the device not win mastership of the UNIBUS for some time.
- Transfer 2 words per bus mastership. Better yet, let the user specify the burst size (but default to 2 words).

- Design in sufficient *high-speed* buffering at the bus interface so that the multiple words transfer as quickly as possible. Do not use a microprocessor to produce the words one at a time, at the microprocessor's speed.

- Allow the user some method of throttling the data rate (for example, provide an adjustable delay between the time that your device releases the bus and the next time that it requests the bus).

Chapter 8 ▪ System Software and Layered Products

▪ Introduction

Software is the collection of written procedures and rules that control computer operations. The system software always includes an operating system, which is the intelligence of the computer system.

Layered products are made up of high-level languages, information management products, programmer productivity tools, and communications software products, that all work in conjunction with a specific operating system.

▪ PDP-11 Operating and Development Systems

Extensive PDP-11 system software allows members of the PDP-11 family to address tasks found in realtime, multiuser timesharing, and batch environments. PDP-11 supports eleven operating systems, of which two were developed exclusively for the MicroPDP-11 family. These make the the PDP-11 suitable for both development and applications environments.

Table 8-1 summarizes the key attributes of these systems.

RSX-11 Family

The RSX family is comprised of four compatible, realtime multiprogramming operating systems that are the industry's leading multiuser realtime operating systems. Designed for minimum size and overhead, this family of operating systems can be used in a wide variety of hardware and application environments—from small dedicated laboratory and industrial control systems to large multiuser information management systems.

Micro/RSX, designed specifically for the MicroPDP-11, is a small and easy-to-use operating system that is customer-installable, and in most cases, allows for complete transportability of applications from other RSX family members.

Once considered primarily a tool for technical applications, users are now discovering that RSX systems are ideal for such commercial applications as office automation, banking, airline reservation systems, and stock exchanges.

The RSX family of operating systems provides reliable, high-performance response to realtime demands, as well as to less time-critical activities such as program development. They are designed to execute multiple programs concurrently. A program is allowed to execute (use the CPU as a resource) until its immediate need for the CPU is completed or until an external event with a higher-priority program takes its place. If the higher-priority program needs memory space, the lower-priority program is swapped out to a disk. This ability to respond rapidly to external events makes the RSX family of operating systems an ideal choice where realtime reaction is important. When tasks of equal priority are eligible to execute, a round robin scheduler rotates their selection so that all receive an equal share of CPU time.

RSX systems provide intertask communications facilities for sharing data, synchronizing execution, and sending messages. The sharing of memory areas and the use of shared resident libraries of software routines result in significant memory savings and increased performance.

However, RSX operating systems are not limited to realtime tasks. Micro/RSX, RSX-11M-PLUS, and RSX-11M also provide users with a complete multiuser program development environment. Software development tools provided with the systems include a choice of comprehensive command languages (including DCL, the Digital Command Language), a choice of editors, Record Management System (RMS) supporting sequential, random, relative, and multikeyed indexed sequential processors as well as debugging aids, system libraries, and a wealth of additional program development and maintenance utilities.

Micro/RSX, RSX-11M-PLUS, and RSX-11M offer unsurpassed software compatibility. All nonprivileged tasks that run on RSX-11M and RSX-11S can run on RSX-11M-PLUS and Micro/RSX without change or reassembly. Privileged tasks usually require little or no change. Micro/RSX and RSX-11M-PLUS also provide significant support for migrating applications to VMS environments. By using VAX-11 RSX, VMS and MicroVMS users can run RSX applications, often without modification, on their VAX and MicroVAX II systems.

WPS-PLUS/RSX is document processing software, which runs as a layered product under both the RSX-11M-PLUS and MICRO/RSX operating systems. This GOLD-Key application features menu driven document processing and includes word and list processing with math and sort capabilities. WPS-PLUS offers both American and British edition dictionaries.

- *Micro/RSX*

  Micro/RSX is an extended subset of the multiuser, multitasking RSX-11M-PLUS operating system. As the newest member of the RSX-11 family, Micro/RSX was designed for use with MicroPDP-11 systems and is a customer-installable, easy-to-use system for both realtime and timesharing environments.

  Micro/RSX is offered in two packages. The *Base Kit* provides the full RSX-11M-PLUS executive, appropriate utilities and device drivers, support for user-mode program development in high-level languages, and a user documentation kit. The *Advanced Programmer's Kit* is an add-on to the Base Kit and includes the software and documentation necessary for MACRO-11 and privileged program development. This includes a MACRO assembler, a librarian, an online debugging tool, and system libraries specifically designed to support privileged programming. The Advanced Programmer's Kit also includes the *data terminal emulator* and *file transfer utility*, which allows for easy file transfer between a Micro/RSX system and any other RSX, VMS (with VAX-11 RSX), or P/OS system. Communications among any of these systems is established through a terminal line.

  Micro/RSX also has Professional 350 diskette-exchange capability, which with the use of the Professional Tool Kit allows programs to be developed for use on the Professional 350. Like RSX-11M-PLUS, Micro/RSX also provides a migration path directly to VMS.

- *RSX-11M-PLUS*

  RSX-11M-PLUS provides the optimal multiuser system software for Digital's newest processors in the PDP-11 family. As the superset member of the RSX family of operating systems, RSX-11M-PLUS offers all of this family's capabilities.

  RSX-11M-PLUS takes advantage of the expanded addressing capabilities of Digital's newest PDP-11 processors while retaining the superior reliability and the successful architecture of RSX-11M. RSX-11M-PLUS uses hardware features in these PDP-11 processors that are not available in other members of the PDP-11 family. With the use of supervisor-mode library routines and separate user-mode instruction and data space, an RSX-11M-PLUS task can address as many as 192 Kbytes of memory. In addition, RSX-11M-PL US supports multistream batch, system accounting, dynamic dual-ported disks, additional memory management capabilities, and more simultaneous tasks and terminals than RSX-11M.

- *RSX-11M*

  The RSX-11M operating system is the original member of the RSX family. It offers a large portion of the capabilities contained in RSX-11M-PLUS. RSX-11M excels on small- and medium-size PDP-11 systems and is designed to support factory automation, laboratory data acquisition and control, graphics, process monitoring, process control, communications, and other applications that demand immediate response. Its multiprogramming capabilities permit realtime activities to execute concurrently with such activities as program development, text editing, and data management.

- *RSX-11S*

  RSX-11S is a memory-resident subset of RSX-11M. As a result, a file system is not supported on RSX-11S. RSX-11S is an extremely efficient execute-only system and is generally used under conditions in which a disk cannot safely operate, such as on the floor of a manufacturing plant.

  RSX-11S provides excellent online process control. Because all programs are memory-resident, response is extremely fast. Tasks for an RSX-11S system are developed on computers running the RSX-11M, RSX-11M-PLUS, or VAX/VMS (with VAX-11 RSX) operating system. Such tasks are then loaded into the RSX-11S system by using a supplied host utility, the RSX-11S Online Task Loader (OTL), or by downline-loading if both the host and the RSX-11S system have DECnet or DECdataway support.

RSTS Family

Thousands of users, from financial institutions and schools to manufacturers, insurance companies, and airlines find RSTS/E to be a system that answers their computing needs. It provides what is important to the commercial and administrative environment—reliability, security, consistency, low cost per user, and an efficient base for building and running commercially oriented applications.

These features, in the course of over a decade, have led to the creation and availability of a wealth of applications suited to all dimensions of the commercial and administrative marketplaces. These applications range from general use products, such as word processing, spreadsheets, inventory control, and accounting to specialty products, such as golf handicapping, chromatography graphics, and legal analysis.

The RSTS family includes two members—RSTS/E and Micro/RSTS. RSTS/E is designed to facilitate maximum flexibility in system configuration and a complete, powerful program development environment.

Micro/RSTS is designed for the MicroPDP-11 supermicrosystems and meets the needs of people whose primary use of the system is running RSTS-based applications. Many applications that run on RSTS/E and the MicroPDP-11 hardware will also run unmodified on Micro/RSTS. Micro/RSTS includes the BASIC-PLUS language but does not include all of the powerful development tools and the configuration flexibility that are standard features of RSTS/E.

- *RSTS/E*

  RSTS/E is a multiuser, general purpose timesharing system. It provides interactive timesharing, batch processing, indirect command file processing, program development using a variety of languages and tools, and a wide variety of special purpose applications. As many as 127 concurrent terminal users in both local and remote locations, through multiterminal services, can interact with applications tasks. Without multiterminal services, the maximum number of users is limited to 63. Tasks can share computational, storage, and input/output services provided by the RSTS/E system. Each one of the multiple users can count on an almost immediate response to requests for access to programs, utilities and data, and transactions in process.

  The user is associated with a job and interacts with that job through a terminal. A timesharing job scheduler allows the job to execute until its immediate need for the CPU is completed, or until an allocated period of time expires. The eligible job with the highest priority will then be executed. If several eligible jobs have the same priority, a round robin scheduler rotates their selection so each gets an equal share of CPU time. For example, a batch job can also be submitted to run at a future time after working hours and the batch processor job will supervise its execution in the submitter's absence.

  Each individual's files and file directory can be protected from unauthorized access by other users. Each user can specify who can read a file and who is allowed to modify or update it.

  Communications with the operating system are accomplished through the easy-to-use Digital Command Language (DCL). Individual installations can add their own commands with the Concise Command Language (CCL). BASIC-PLUS is also included with the RSTS/E operating system.

  Because RSTS/E can migrate from one PDP-11 processor to another, system growth is easy. In addition, distributed processing networks can be built using DECnet/E for Digital-only networks and Internets for connection to other vendor's systems.

- *Micro/RSTS*

  Micro/RSTS is a pregenerated subset of RSTS/E and supports all RSTS/E system calls. Micro/RSTS was designed primarily for use with the MicroPDP-11s, and is a customer-installable, easy-to-use system that can support up to 20 jobs and as many as 14 terminals in a timesharing environment.

Micro/RSTS consists of two separate kits. The *Base System Kit* is required for all users, and the *Application Development Kit* is optional. The Base System Kit includes software and documentation for the Micro/RSTS runtime system, for device support, and for BASIC-PLUS program development. BASIC-PLUS is included with the kit. The Application Development Kit includes software and documentation to support native MACRO-11 and high-level language program development. MACRO-11 is included with the Application Development Kit.

Micro/RSTS supports programs developed on any RSTS/E system. Host development requires that files be transferred to and from the Micro/RSTS system. This can be done by using disk or tape media common to both systems.

MicroPower/Pascal Development Toolkit

MicroPower/Pascal is an advanced software toolkit for developing Q-bus-based microcomputer applications. It includes a high-performance Pascal compiler, a modular executive, and a variety of tools to create concurrent, realtime application programs.

- *MicroPower/Pascal Features*

  MicroPower/Pascal has two system environments to accomplish this development. The *host system* creates and builds the software. The *target system* executes the software. Each application is custom-designed for its target system and includes the appropriate set of operating system services. The host, using the symbolic debugger, controls the execution of the target application during development.

  Three MicroPower/Pascal products are available to develop applications using a PDP-11 host system—MicroPower/Pascal-RT, MicroPower/Pascal-Micro/RSX, and MicroPower/Pascal-RSX.

  The host development environment for each of these products includes an extended, realtime Pascal compiler, a symbolic debugger, several build utilities, and a MACRO-11 interface. The target environment includes a library of software modules for process synchronization, communications, scheduling, exception and interrupt handling, timer services, and device and file I/O.

  The application program is created and linked with the appropriate runtime software in the host system. It is then transported to the target system by one of three methods—writing it into read-only memory, downline-loading it over a serial line, or recording it onto removable storage media such as a flexible disk or tape cartridge and then bootstrapping it on the target system.

MicroPower/Pascal is very compact and can reside in as little as 8 Kbytes of memory for small application programs. For complex applications, MicroPower/Pascal can address as much as 4 Mbytes of memory.

ULTRIX-11 Operating System

ULTRIX-11 is Digital's enhanced native-mode UNIX™ system software for PDP-11s, providing a flexible, interactive programming environment for multiple users. ULTRIX-11 is an enhanced version of the UNIX Timesharing System, Seventh Edition (V7) developed by AT&T Bell Laboratories.

- *ULTRIX-11 Features*

ULTRIX-11 includes all of the features found in Version 7, such as the Bourne shell, C shell, shell scripts, pipes, the C compiler, and the Assembler. In addition, it includes a hierarchical file system that can provide a directory of files. Subdirectories can also be created to manage groups of similar files. Input/output can be performed by reading or writing into a special file that is associated with an I/O device. This makes file and device I/O similar for ease of programming. It also allows a program to accept either a file or device without changes, and extends the file protection mechanism to the I/O. The creator of a file can permit or deny read, write, and access protection to other users.

Digital provides many enhancements for ULTRIX-11 for better performance and maintainability. These enhancements include:

- TCP/IP networking via networking.
- Source-level compatibility with UNIX System V.
- System performance improvements.
- Improved fault tolerance.
- Disk bad-block replacement.
- Automated installation and system generation.
- System tuning.
- Processor and peripheral device support.
- VI full-screen editor.
- Terminal Enabling Editor (TED).
- Overlay kernel for CPUs with combined instruction and data space.
- Special files.
- File-system table.

- Crash-dump analyzer.
- System-management commands.
- C shell.
- Kernel floating-point simulator.
- TIP remote login and file transfer.
- Source code control system with job control.
- Certain System V features.
- Certain University of California at Berkeley Version 2.9 features.

The ULTRIX-11 operating system is interfaced through the Bourne or C shell command-line interpreter. Shell commands create processes that can communicate through pipes, create subsidiary processes, and synchronize the offspring processes. These interfaces permit users to create commands for individualized routines that can be run in an interactive environment. This means users can produce their own command lines and command files to assign symbolic names, evaluate numeric and logical expressions, accept parameters, communicate with interactive users invoking the shell script, and perform conditional and branching logic.

ULTRIX-11 languages include:

- C—a high-level language conducive to structured programming.
- FORTRAN-77 and RATFOR—adds a C-type control structure to FORTRAN.
- BASIC-like interpretive language.
- Programmable desk calculator.

Software tools include a compiler writing system, document preparation programs, information handling routines, and graphics support. The customer can easily install ULTRIX-11 and also run the System Exerciser Package to verify that it is functioning properly.

RT-11 and CTS-300 Operating Systems

RT-11 is an operating system for single-user, realtime applications, well-suited for such applications as laboratory and factory instrument control, manufacturing process control, flight management, mapping, and numerous other technical jobs.

CTS-300 is a complete, disk-based, single or multiuser software environment layered on RT-11, designed to support commercial applications. CTS-300 includes DIBOL, a programming language designed for writing business applications.

- *RT-11 Features*

  RT-11 uses the Digital Command Language (DCL). This makes access to operating system services as easy as typing English-like commands. Instead of having to manage system calls directly, services can be called through DCL commands that will prompt for any missing parameters and will offer help if a problem or question arises.

  The keypad editor, KED/KEX, is specially designed for a wide range of videoterminals and takes advantage of all their advanced features. Screen-oriented editing immediately points out any editing problems and makes quick changes to correct errors or to accommodate altered program needs.

  RT-11 systems offer a choice of three different operating-system monitors to accommodate a range of RT-11 users. Digital supplies the system with a single-job monitor, a foreground/background monitor, and an extended-memory monitor. A single-job monitor, called SJ, organizes the system for single-user, single-program conditions. The foreground/background monitor, called FB, takes advantage of the fact that much central processor time is spent waiting for external events such as I/O transfer or realtime interrupts. In the FB monitor, this waiting time is put to good use by allowing the CPU to be used for other jobs while the principal (foreground) job is pausing. The extended-memory monitor, XM, allows both foreground and background jobs to extend their effective logical program space beyond the 64-Kbyte space imposed by 16-bit addresses on PDP-11 computers. The XM monitor contains all the features of FB plus the capability of accessing as many as 4 Mbytes of memory.

  There are three communications utilities that come with RT-11. VTCOM (Virtual Terminal Communications) allows RT-11 to connect to any host system via a serial line and transfer ASCII files between the two systems. TRANSF (Binary File Transfer) uses VTCOM and allows binary files to be transferred between RT-11-based (RT-11, CTS-300, RTEM-11) systems via a serial line. Ethernet Handlers for DEQNA allow users to write their own software to communicate over Ethernet hardware.

  RT-11 provides even more system accessibility to both novice and experienced users alike. RT-11 offers an automatic-installation procedure that installs the operating system simply by conducting an interactive dialogue with the user.

Programs can be written without explicitly identifying the output device. For example, the device selection can be deferred until the program is run so that printer output can be directed to the disk. When a new device is added to the system, any old programs can be adapted easily.

RT-11 programs can also be developed as one of the tasks on an RSX-11 system using RTEM-11. Programs developed with RTEM-11 can execute on appropriately configured RT-11 systems in the same manner as if they had been developed on RT-11. Most programs developed on RTEM-11 can be debugged and tested on RTEM-11. The execution environment supplied with RTEM-11 is foreground/background (FB).

Although RT-11 supports only one command terminal, multiterminal support capability allows programs to control up to 16 additional terminals.

- *CTS-300 Features*

CTS-300 is designed to support commercial applications. It consists of the RT-11 operating system, described before, plus DIBOL (Digital's Business-Oriented Language), and a number of utilities. Third party software can run in a CTS-300 environment only at the RT-11 DCL level and cannot run under SUD, TSD, or XMTSD.

CTS-300, like RT-11, is a single-user system in the sense that there can be only one system command terminal. However, multiple terminals running multiple DIBOL jobs or developing multiple DIBOL programs are supported under the three DIBOL runtime systems—single-user DIBOL (SUD), timeshared DIBOL (TSD), and extended-memory timeshared DIBOL (XMTSD).

DIBOL is an easy-to-learn and easy-to-use language that allows commercial applications to be developed in minimal time. DIBOL has a Data Division and a Procedure Division, like COBOL, and provides the ability to manipulate data, evaluate arithmetic expressions, redefine records, call other programs, spool output, and access files.

Utilities included with CTS-300 are:

- DECform—defines video screen formats, checks entered data for range and type, and totals and validates entered fields. It also supports additions, inquiries, changes, and verifications to DMS-300 files.
- DMS-300—a data management utility that supports sequential access, random access, and keyed access to ISAM files.
- SORT/MERGE—a data management utility that permits users to easily define the parameters for sorting and merging data files.
- Line Printer Spooler Utility—queues and manages files for printed output.

DSM-11 (Digital Standard MUMPS)

DSM-11 is a complete, multiuser system environment with data management capability and the interactive, high-level language, MUMPS. With DSM-11, programs can be quickly written, tested, debugged, or modified to establish a working application.

The MUMPS language, originally developed at Massachusetts General Hospital, has syntax and semantics oriented toward solving database-related applications. A novice programmer can very quickly produce useful working code, although using the full range of MUMPS capabilities does require some programming experience.

MUMPS' text-handling capabilities allow the inspection of any data item for content (such as particular keywords) or for format (letters, numbers, or punctuation characters in a string of text). The capabilities are useful for online data-entry checking and correction.

The DSM-11 hierarchical file structure allows data files to be designed to suit the needs of a particular environment. Dynamic file storage simplifies expansion or modification of the database. The database handler maintains an in-memory cache of disk data for high-performance data access and data sharing.

DSM-11 implements an extension of the 1983 ANSI Standard MUMPS language. DSM-11 allows a MUMPS application to define independent error handlers for each execution level. A MUMPS debugger allows the DSM-11 programmer to set or clear breakpoints, single-step through MUMPS commands, and trace program execution.

- *DSM-11 Features*

DSM-11 is a complete multiuser operating system for the PDP-11 systems that includes the MUMPS language interpreter and powerful, high-performance data management capability. DSM-11 supports the following features:

- Mountable volume sets.
- Interjob communications.
- Memory-resident applications.
- Magnetic-tape streaming.
- IBM binary synchronous communications.
- Autoconfiguration.
- Unattended backup.

- System-level, transparent journal of database modifications can be maintained on either disk or magnetic tape.
- Output to devices (such as a printer) can be spooled.
- Bad-block management for all disk media.
- Online, high-speed database backup, disk-media preparation, and tape-to-tape copying.
- Hardware device-error reporting, system patching utility, and an executive debugger for system maintenance.
- System installation and generation procedures.

Table 8-1 ▪ Operating-system Summary

Feature	Micro/ RSX	RSX-11M-PLUS	RSX-11M	RSX-11S	Micro/ RSTS	RSTS/E	ULTRIX-11	RT-11	CTS-300*	DSM
User Interface										
Shell							X			
DCL	X	X	X	X	X	X		X	X	
MCR		X	X	X						
CCL					X	X		X	X	
User-written	X	X	X					X	X	
Text Editors										
Keypad	X	X	X		X	X		X	X	
Line	X	X	X		X	X	X	X	X	X
Screen	X	X	X	X	X	X		X	X	
Batch Processing	X	X				X		X	X	
File Management										
Multikey ISAM	X	X	X		X	X				
Single-key ISAM	X	X	X		X	X			X	
Sequential	X	X	X		X	X	X	X		
Relative	X	X	X		X	X		X		
Random	X	X	X		X	X		X	X	

*Includes RT-11, DIBOL-83, and DECform.

▪ High-level Languages

Most operating systems need additional software, such as programming languages, to perform more specialized tasks than the operating system can perform alone. The PDP-11's programming languages are well-suited to the needs of industry, science, education, and business. They are typically developed in response to specific functional needs. Some languages, such as FORTRAN, were originally intended for processing enormous amounts of numerical data through complicated formulas at high speeds. Others, such as COBOL and DIBOL, were developed for commercial applications in which data management played a major role. And still others, like BASIC, were invented for use by students who were unfamiliar with computers and needed a simple, easy-to-learn language related to everyday speech. The descriptions in this section show the special strengths of each Digital-supplied language in satisfying specific application needs.

Table 8-2 summarizes the languages supported by each operating system.

BASIC

Three versions of BASIC are available for the PDP-11 systems. All versions use simple English commands, understandable abbreviations, and familiar symbols for mathematical and logical operations. They are readily accessible to programmers who are not computer specialists. They are used extensively in education, small businesses, laboratories, and for personal use.

BASIC-PLUS-2 is an extended BASIC compiler that takes full advantage of the PDP-11 floating-point and integer instruction sets while generating threaded code instructions. A high performance program execution environment is provided for applications development and tinmesharing. BASIC-PLUS-2 combines immediate mode with the power of a structured programming language. BASIC-PLUS-2 runs under the Micro/RSX, RSX-11M-PLUS, RSX-11M, Micro/RSTS, and RSTS/E operating systems.

BASIC-PLUS and BASIC-11 are conversational proBASIC-PLUS-2 programming languages developed at Dartmouth College that use simple English-like statements and familiar mathematical notations to perform operations. BASIC-PLUS is an integral part of the RSTS/E operating system, and is also available for the RT-11 operating system. BASIC-11 is optional for the CTS-300 operating system.

C

C is a concise, expressive, structured programming language designed by AT&T Bell Laboratories for program development on UNIX systems. It is included with ULTRIX-11 for the PDP-11 family.

COBOL-81

COBOL is an industry-standard data-processing language used extensively for business applications because of its orientation toward character, string, and file processing. Digital provides a version of COBOL based on the 1974 ANSI COBOL standard.

CORAL-66

CORAL-66 is a block-structured language developed by the British government for realtime and process-control applications. The language is designed to replace assembly-level programming in modern industrial and commercial applications. It is used for long-life products where ease of maintenance and flexibility are required. CORAL-66 is available only on PDP-11 systems with RSX-11M-PLUS, RSX-11M, or RSX-11S.

DIBOL-83

DIBOL-83 (Digital Interactive Business-Oriented Language) is a high-level, procedural language designed specifically for interactive business data processing. DIBOL-83 is based on the DIBOL Standards Organization definition.

It is represented in two segments—a data division and a procedure division. The data division defines the data that is used by the program. The procedure division contains the executable statements. DIBOL-83 is available on PDP-11 systems under Micro/RSX, RSX-11M-PLUS, RSTS/E, and as part of CTS-300.

FORTRAN

FORTRAN is the most widely used programming language for developing programs dealing with scientific applications. Two FORTRAN compilers, supporting run-time operations, are available for the PDP-11 systems:

- PDP-11 FORTRAN-77 for Micro/RSX, RSX-11M-PLUS, RSX-11M, RSX-11S, Micro/RSTS, VAX/VMS via VAX-11 RSX, and RSTS/E systems. FORTRAN-77 is available for RT-11.
- FORTRAN IV for RSX-11M-PLUS, RSX-11M, RSTS/E, and RT-11 systems.

FORTRAN-77 is an extended implementation of the ANSI subset FORTRAN-77 standard (X3.9-1978). It contains all the features of the ANSI FORTRAN-77 subset, many of the full-set language features, and extensions not included in the standard. The FORTRAN-77 compiler produces direct PDP-11 machine code optimized for execution-time efficiency with the PDP-11 floating point processor.

FORTRAN IV is a fast, one-pass, optimizing compiler that implements an extended superset of the ANSI X3.9 1966 standard for FORTRAN. FORTRAN IV works efficiently in small-memory environments and is capable of producing absolute binary code for stand-alone PDP-11 systems or for loading into ROM or PROM memory.

MUMPS

MUMPS is a language oriented toward database applications. It is an integral part of DSM and is described in the DSM-11 section.

PDP-11 Pascal/RSX

PDP-11 Pascal/RSX is a block-structured language that contains English-like commands and logical grammar.

PDP-11 Pascal/RSX provides all standard Pascal features as well as extensions that are designed to improve the productivity of the Pascal programmer. These extensions make it simple to divide programs into easily managed problem sections and to enhance the computing power of the language. PDP-11 Pascal/RSX is available as an option on Micro/RSX, RSX-11M-PLUS, and RSX-11M.

Table 8-2 ▪ High-level Language Summary

Language	Micro/ RSX	RSX-11M-PLUS	RSX-11M	RSX-11S	Micro/ RSTS	RSTS/E	ULTRIX -11	RT-11	CTS-300	DSM-11
BASIC	X	X	X		X	X		X	X	
C							X			
COBOL-81	X	X	X		X	X				
CORAL-66		X	X	X						
DIBOL-83	X	X				X			X	
FORTRAN	X	X	X	X	X	X		X		
MUMPS										X
PDP-11 Pascal/RSX	X	X	X							

*Includes RT-11, DIBOL-83, and DECform.

▪ Information Management Software

Information management software ensures users and programmers that they have an integrated system of data management capabilities to help them organize their data. Information management products help users do more for themselves and give programmers more time to plan and develop new applications. Some of these products aid the programmer by reducing development time and costs. This section briefly describes each of the information management products that are available for PDP-11 systems.

DATATRIEVE-11

DATATRIEVE-11 is an interactive, query, report generation and data maintenance system designed for the less experienced computer user. DATATRIEVE-11 provides facilities for selective data retrieval, sorting, formatting, updating, and report generation without the need for programming. DATATRIEVE-11 is available on PDP-11 systems running Micro/RSX, RSX-11M-PLUS, RSX-11M, and RSTS/E.

DECgraph-11

DECgraph-11 is an interactive, menu-driven tool for generating graphs from data. It is designed to be used by experienced computer users and novices alike, offering a wide spectrum of capabilities for producing professional quality graphs. DECgraph-11 is available for PDP-11 systems running Micro/RSX, RSX-11M-PLUS, Micro/RSTS, and RSTS/E.

DECmail-11

DECmail-11 is an electronic-message system that can create, edit, send, and process messages, as well as store, search for, and retrieve messages held in user folders. DECmail-11 is menu or command-driven and includes an extensive online help facility. DECmail-11 can be used in a network environment. When DECmail-11 is used with optional DECnet products, its features can be accessed in a multinode environment to communicate with VMS MAIL, A-to-Z Electronic Mail, and ALL-IN-1 mail products via the VMS Message Router or the Message Router VMSmail Gateway, and other nodes running DECmail-11.

DECmail-11 is available on PDP-11 systems running Micro/RSX, RSX-11M-PLUS, Micro/RSTS, and RSTS/E.

FMS-11 (Forms Management System)
FMS-11 is designed to aid in the development of application programs that use video forms. FMS-11 manages the forms for application programs that use the VT100 and VT200-compatible terminals. FMS-11 provides the forms creator with character attributes of reverse video, boldface, blinking, and underline. It provides line attributes of double width, double height, and scrolling. Screenwide attributes such as 80 or 132 column lines and reverse video are included. Alternate character sets including the VT100 special graphics character set for line drawing are supported. FMS-11 is available for PDP-11 systems running Micro/RSX, RSX-11M-PLUS, RSX-11M, RSTS/E, and RT-11.

INDENT
INDENT is a data-entry and forms management product for commercial applications written in DIBOL, COBOL, or BASIC-PLUS-2. INDENT provides reverse video, boldface, underline, 132-column lines, scroll, split screen, reverse screen, and the line-drawing character set. INDENT form definitions are created using a text editor. Data from a form is returned to the application program when an entire form is completed or when an individual field is completed. INDENT is available for PDP-11 systems running RSTS/E.

RMS (Record Management System)
RMS is a straightforward method of creating, updating, and modifying files using sequential, relative, or multikey indexed access methods. RMS is an integral part of Micro/RSX, RSX-11M-PLUS, RSX-11M, Micro/RSTS, and RSTS/E on PDP-11 systems.

▪ Programmer Productivity Tools

Productivity tools allow program developers to simplify their programming and information-management processes. By eliminating extensive maintenance and documentation tasks, more time is left for creating new applications or changing existing ones. This section briefly describes each of the productivity tools available for PDP-11 systems.

ADE (Application Development Environment)
ADE is a software package for the non-programmer who develops small, simple applications requiring the processing of alphabetic, numeric, and data-oriented data such as personnel records, order processing, department budgets, financial/forecasting models, mail lists, and telephone lists. ADE provides easy-to-use facilities and functions for users to create their own databases; add, change, or delete data; produce simple bar graphs; and write reports without waiting for formal programming and report generation. It features total interaction between terminal and user, absence of technical jargon, use of acronymns, easy transfer to and from more powerful application software packages, full-screen handling, user prompts after each input, extensive HELP messages to explain all commands, user protection of data, and automatic sorts, alphabetically, numerically, or chronologically. ADE is available for PDP-11 systems running Micro/RSTS and RSTS/E.

MENU-11
MENU-11 provides the applications developer with the ability to present a simple, menu-driven interface to the user on a videoterminal for both system and application functions. MENU-11 accepts and executes commands related to the menu items, to perform specific functions, without extensive user training in the Micro/RSTS or RSTS/E operating system. Three types of commands provide screen formatting, program execution, and security access. These commands allow the developer to format the menu screen for the user's terminal and to control the interfacing of the user with the system. MENU-11 is available on PDP-11 systems running Micro/RSTS and RSTS/E.

▪ Communications

Communications software makes the networking functional within the Digital Network Architecture (DNA) framework. Digital's network software includes three categories:

- DECnet—for communications among Digital systems.
- Internet—for communications with other manufacturers' equipment.
- Packetnet—for communications with other participants in public packet-switched networks.

For more detailed information on these communications software areas, refer to Chapter 10—Networks. For descriptions of the software that is available in each of these areas, refer to the *Networks and Communications Buyer's Guide*.

▪ Business Applications

The A-to-Z Integrated System is a user-installable, user-maintainable, small business system for MicroPDP-11 computers. A-to-Z software is a layered product on the Micro/RSX operating system. A-to-Z software includes the following features:

- Easy for non-technical computer users to install.
- Easy for the novice end user to maintain.
- Supports an integrated set of layered products.
- Supports the use of INTERRUPT throughout A-to-Z layered products.
- Provides a set of standards for software developers.
- Provides for growth.

A-to-Z is provided in module form for greatest user flexibility in the selection of desired features. These modules include:

- A-to-Z Base System (Prerequisite for running any of the A-to-Z layered products.
- Word Processing
- Data Inquiry
- Business Graphics
- Developer's Kit
- Electronic Mail
- Document Transfer
- Supercomp-Twenty spreadsheet

▪ Additional Documentation

- *PDP-11 Software Handbook*
- *ULTRIX Software Guidebook*
- *Networks and Communications Buyer's Guide*
- *PDP-11 Software Source Book*

Chapter 9 ▪ System Options

▪ Introduction

Once a PDP-11 computer system configuration is selected, the hardware options that best suit the user's application must also be chosen. These options include memory, storage devices, communications devices, terminals, printers, and modems. This chapter provides information about the kind of expansion possibilities available with PDP-11 systems.

Q-bus Memory Options

There are several memory options that are available for MicroPDP-11 systems. Each provides the capability to perform direct-memory access. During direct-memory access operations, data is transferred without processor intervention using block-mode transfers.

▪ *MCV11-DC Memory*

The MCV11-DC is a 32-Kbyte CMOS static random access memory with onboard battery backup for a minimum data retention time of 50 days.

▪ *MSV11-DV*

The MSV11-DV is a set of CMOS static random access memory modules with onboard battery backup. The following are the main features of the MSV11-D series:

- Provides minimum data retention time of 50 days.
- Offers 8 Kbytes (MSV11-DA) or 32 Kbytes (MSV11-DC) of CMOS static random access memory on a single, dual-height module.
- Uses 16-K CMOS static RAM chips.
- Supports 16-, 18-, or 22-bit addressing.
- Offers 1 Mbyte (MSV11-JD) or 2 Mbytes (MSV11-JE) of PMI ECC MOS memory on a single, quad-height module.
- Uses 256-Kbyte RAM integrated circuits.
- Supports 22-bit addressing for as many as 4 Mbytes of physical memory.
- Supports block-mode DMA transfers.
- Has LEDs for parity-error indication.

- *MSV11-M Memory*

The MSV11-M series is a set of dynamic RAM memory modules. The following are the main features of the MSV11- M series:

- Offers 512 Kbytes (MSV11-MA) or 1 Mbyte (MSV11-MB) of dynamic RAM memory on a single, dual-height module.
- Uses 256-Kbyte MOS dynamic RAM integrated circuits.
- Supports 22-bit addressing for as many as 4 Mbytes of physical memory.

- *MSV11-Q Memory*

The MSV11-Q series is a set of random-access memory (RAM) modules. The following are the main features of the MSV11-Q series:

- Offers 1, 2, or 4 Mbytes (MSV11-QA, -QB, -QC) of MOS memory on a single, quad-height module.
- Uses 64-Kbyte (MSV11-QA) or 256-Kbyte (MSV11-QB, -QC) MOS RAM integrated circuits.
- Supports 22-bit addressing for as many as 4 Mbytes of physical memory.
- Supports block-mode DMA transfers.
- Has LEDs for parity-error indication.

- *MSV11-P Memory*

The MSV11-P series is a set of random-access memory (RAM) modules. The following are the main features of the MSV11-P series:

- Offers 256 Kbytes (MSV11-PK) or 512 Kbytes (MSV11-PL) of MOS memory on a single, quad-height module.
- Uses 64-Kbyte MOS RAM integrated circuits.
- Supports 18-bit or 22-bit addressing for as many as 4 Mbytes of physical memory.
- Supports block-mode DMA transfers.
- Has LEDs for parity-error indication.

- *MSV11-SA Memory*

  The MSV11-SA is a dynamic RAM memory module. The following are the main features of the MSV11-SA:

  - Offers 2 Mbyte of dynamic RAM memory on a single, dual-height module.
  - Uses 256-Kbyte MOS dynamic RAM integrated circuits.
  - Supports 22-bit addressing for as many as 4 Mbytes of physical memory.

Q-bus Storage Options

PDP-11 storage options are offered in several different technologies so that the correct choice can be made for the storage application. Whether storage is being added for media backup, for loading software, for main storage, or for software interchange with another system, Digital has the appropriate storage device for the task.

- *RQDX3 Controller*

  The RQDX3 is an intelligent controller that provides data transfers among the Q22 bus, the RX- series of flexible-disk drives, and the RD-series of fixed-disk drives. This controller contains logic that provides the necessary data buffering and control to allow direct-memory access (DMA) transfers using the Mass Storage Control Protocol (MSCP).

  A flat cable attaches to a 50-pin connector mounted at the edge of the module and to the signal distribution board located near the Q22-bus backplane. Signals and data are then transferred from the connectors on the distribution board to the disk-drive assemblies. Four LED indicators are also mounted near the edge of the module to display octal codes during the self-test program operation.

  The following are the main features of the RQX3:

  - Control of up to four logical disk-drive units—one flexible-disk drive and up to two fixed-disk drives or a total of four fixed-disk drives.
  - Support for block-mode DMA data transfers.
  - Dual-height module size.
  - Maintenance self-test programs.
  - LEDs for parity-error indication.

▪ *KDA50 Disk Controller*

The KDA50 is an intelligent controller that interfaces up to four RA series disk drives with the MicroPDP-11/83. Two quad-height modules, the Standard Disk Interconnect (SDI) module and the processor module, make up the KDA50. The SDI module is the communications interface between the KDA50 processor module and the disk drives. The processor module is the control portion of the KDA50.

Two flat cables attached to a 40-pin and 50-pin connector tie together the two modules. An internal SDI cable connects the KDA50 modules to the signal distribution board located near the Q22-bus backplane. Signals and data are then transferred from the connectors on the distribution board to the disk-drive assemblies. Four LED indicators are also mounted near the edge of the module to display octal codes during the self-test program operation.

The following are the main features of the KDA50:

- Controls as many as four RA-series (RA81, RA60) disk drives in a radial connection.
- Supports block-mode DMA transfers.
- Has maintenance self-test programs.
- Uses LEDs for parity-error indication.

▪ *RX50 Flexible Disk*

Programs and data can be moved in and out of the supermicrosystems through the RX50 flexible-disk drives. The RX50 is a single unit that contains two separate drives. Each of the two drives in the RX50 operate with a 5.25-inch flexible disk and provide a storage total on both disks of as many as 819.2 Kbytes of formatted data. Access to the drive is through the two doors located at the front of the drive unit.

The RQDX controller module, located in the Q22-bus backplane, provides the interface between the Q22 bus and the RX50 flexible-disk drive. The controller implements the required MSCP protocol and controls the DMA data transfers. The RX50 operates with dc power supplied from the power supply.

The following are the main features of the RX50:

- 819.2 Kbytes total formatted capacity (409.6 Kbytes formatted capacity per diskette).
- Two surfaces on a single spindle.
- 250 Kbits/second (31.25 Kbytes/second) average transfer rate.

- 164 milliseconds average seek time.
- Available in integrated, tabletop, and rackmount models.

- *RX33*

  The RX33 is a half-height, 5.25-inch 1.2 Mbyte diskette drive. In high density mode, it provides industry standard compatibility using double-sided, high-density diskettes. In standard mode, the RX33 can both read and write RX50-type standard density diskettes on a single side. This dual mode compatibility allows users to access a vast software base without sacrificing RX50 compatibility.

  The RX33 provides three times the capacity per diskette of the RX50, while its half-height form factor requires only half the space. RX33 features include:

  - High diskette capacity.
  - 500 Kbits/second peak transfer rate.
  - Dynamic media clamping.
  - Diskette charged detection device.

- *RD54, RD53, RD32, and RD31 Fixed Disks*

  The RD54, RD53, RD32, and RD31 fixed-disk drives are compact, Winchester disk drives that provide reliable mass-data storage for the supermicrosystems. The drives contain double-sided, 5.25-inch, nonremovable disks enclosed in a sealed assembly. A microprocessor within each of the units controls the data transfers.

  The RD54 has 159 Mbytes of capacity, making it possible to support large applications and/or more users. The drive is packaged in compact housing that conforms to industry standards for 5.25-inch disk media. The RD54 is available as an add-on drive for BA23 and BA123 system enclosures, or for desktop or rackmount enclosures. The RD54 has a peak transfer rate of 5.0 Mbits and average seek time including settling of 30.0 ms.

  The RD53 has 71 Mbytes of formatted capacity. It includes eight recording surfaces and features a 5 Mbits/second peak transfer rate. The RD53 is available in integrated, tabletop, and rackmount models.

  The RD32 is a 42-Mbyte nonremovable half-height Winchester technology disk drive. The drive mechanism is packaged in a single 1.63-inch-high housing, which conforms to industry standards for 5.25-inch disk media. The RD32 offers high capacity and reliability in a compact format.

The RD31 is an integrated 20-Mbyte half-height, fixed-disk drive (11/53 only). The drive mechanism is packaged in a compact housing and conforms to industry standards for 5.25-inch disk media.

- *RA81 Fixed Disk*

The RA81 is a high-capacity (456 Mbytes formatted), rackmounted Winchester fixed-disk drive for the cabinet enclosure. It is known for its outstanding data reliability characteristics, including an industry-leading 170-bit error-correction code (ECC). It requires the KDA50 controller module set.

The following are the main features of the RA81:

- 456 Mbytes formatted capacity.
- Read/write system employing an encoding/decoding scheme that yields one-third more storage capacity than drives using conventional encoding.
- Dual ports.
- 2.2 Mbytes/second peak transfer rate.
- 36.3 milliseconds average access time.

- *RA60 Removable Disk*

The RA60 is a high-capacity (205 Mbytes formatted), rackmounted removable-disk drive for the cabinet enclosure. The RA60 disk drive uses enhanced servo technology to eliminate the need for alignment packs. It also incorporates new recording methods, microprocessor-cont rolled diagnostics, a 170-bit error-correction code, and a modular design for easy maintenance.

The following are the main features of the RA60:

- 205 Mbytes formatted capacity.
- Enhanced servo technology—eliminates the need for alignment packs.
- Dual ports.
- 1.98 Mbytes/second peak transfer rate.
- 50.3 milliseconds average access time.

- *TK50 Cartridge Tape*

  The TK50 is a 95-Mbyte, 5.25-inch, streaming-cartridge tape drive for the MicroPDP-11/73, MicroPDP-11/83, and PDP-11/84 systems. It is a compact and convenient backup, bootstrap, and distribution device that complements the supermicrosystem's disk drives. The TK50 will serially record up to 95 Mbytes on a 0.5-inch by 600-foot tape that is enclosed in a COMPAC-Tape cartridge.

  The following are the main features of the TK50:

  - Up to 95 Mbytes formatted capacity (operating-system dependent).
  - Industry-leading data integrity and reliability.
  - 62.5 Kbytes/second peak transfer rate (45 Kbytes/second for user data).
  - 75 inches/second tape speed.
  - 22 tracks on a 0.5-inch COMPACTape.
  - 6,667 bits/inch recording density.
  - Available in integrated, tabletop, and rackmount models.

- *TSV05 Streaming Tape*

  The TSV05 is an industry-standard, 9-track, magnetic-tape drive for the MicroPDP-11 systems. It includes a tape transport with an integral formatter and a single, quad-height controller module. It features a storage capacity of 40 Mbytes (using 8-Kbyte blocks) and 28 Mbytes (using 2-Kbyte blocks) on a 10.5-inch reel, 25/100 inches/second streaming-tape backup, and front-loading automatic tape threading. The TSV05 supports industry-standard 1,600 bits/inch phase-encoded format (ANSI-compatible). The tape transport occupies only 8.7 inches (22 centimeters) in a 42-inch-high (106-centimeter) cabinet enclosure, and without the cabinet for system integrators.

  The following are the main features of the TSV05:

  - 40 Mbytes formatted capacity (using 8-Kbyte blocks) and 28 Mbytes formatted capacity (using 2-Kbyte blocks).
  - 10.5-inch by 2,400-foot reel.
  - Front loading.
  - Automatic tape threading.
  - 40 or 160 Kbytes/second peak transfer rate.
  - 25/100 inches/second read/write speed (preset by user).
  - 1,600 bits/inch recording density.

Q-bus Communications Options

The communications capability of each of the supermicrosystems can be expanded by communications interface options. These options enable asynchronous, synchronous, and realtime data transfers between two or more systems, and also between a host system and its user terminals, modems, and other external devices. Each option consists of an interface module, internal cables, and a panel insert. The interface module installs into a slot in the Q22-bus backplane, and the device connector panel insert is mounted in the I/O distribution panel at the rear of the supermicrosystem chassis.

Q-bus Asynchronous Interfaces

- *DZQ11*

  The DZQ11 is a four-line, asynchronous multiplexer that provides local or remote interconnection between the supermicrosystem and EIA RS232-C/CCITT V.10 terminals or other systems. The DZQ11 operates at program-selectable speeds up to 9,600 bits/second at full-duplex with limited-modem control on each line.

  The DZQ11 is a single, dual-height module. It is compatible with Digital's family of modems and with the Bell 100 and 200 series of modems and their equivalents.

- *DHV11*

  The DHV11 is an eight-line, asynchronous, direct-memory access multiplexer that provides local or remote interconnection between the supermicrosystem and EIA RS232-C/CCITT V.28 terminals or other systems. Direct-memory access reduces system overhead for terminal-intensive applications. The DHV11 operates at program- or jumper-selectable speeds up to 38,400 bits/second at full-duplex with full-modem control on each line. Split-speed transmit and receive rates are supported on each line making more efficient use of communications facilities by reducing the software demand for the receive line.

  The DHV11 is a single, quad-height module. It is compatible with Digital's family of modems and with the Bell 100 and 200 series of modems and their equivalents.

- *DLVJ1*

  The DLVJ1 is a four-line, asynchronous interface that provides local or remote interconnection between MicroPDP-11 systems and EIA RS232-C/CCITT V.28, EIA RS422/CCITT V.11, and EIA RS423/CCITT V.10 terminals. The DLVJ1 acts as four separate devices, making program operations more convenient than they are with a multiplexer. The DLVJ1 operates at program- or jumper-selectable speeds from 150 to 38,400 bits/second at full

duplex with limited-modem control. Split-speed transmit and receive rates are supported on each line making more efficient use of communications facilities by reducing the software demand for the receive line.

The DLVJ1 is a single dual-height module. It is compatible with Digital's family of modems and with the Bell 100 and 200 series of modems and their equivalents.

Q-bus Synchronous Interfaces

- *DEQNA*

  The DEQNA is a high-performance, synchronous communications controller that connects the supermicrosystem to an Ethernet Local Area Network (LAN). The DEQNA complies fully with the Ethernet specification and operates at 10 Mbits/second.

  The DEQNA provides Ethernet data-link layer functions and a portion of the physical channel functions. The DEQNA is supported under DECnet Phase IV software. Digital also provides documentation and device drivers so that users can write their own higher-level protocols for specialized applications and communications in multivendor environments. The DEQNA allows communications with up to 1,023 addressable devices on an Ethernet. It physically and electrically connects to the Ethernet coaxial cable via Ethernet transceiver cables and an H4000 Ethernet transceiver, or a Local Area Interconnect (DELNI). The Physical Address ROM (DEXMR) is required to downline load software to a diskless Ethernet node with a DEQNA.

  The DEQNA is a single, dual-height module.

- *DMV11*

  The DMV11 is a single-line, synchronous microprocessor-controlled interface that provides local or remote interconnection between the supermicrosystem and other computer systems with EIA RS232- C/CCITT V.28, or RS423/RS449 interfaces. The DMV11 implements DDCMP in hardware and supports direct-memory access data transfers, DECnet point-to-point or multipoint configurations, and full-modem control. It operates at speeds from 19,200 bits/second to 56,200 bits/second (depending on the version selected) at half- or full-duplex.

  Depending on the operating system and layered software, the DMV11 can support up to 12 tributaries. In multipoint configurations, these tributaries can be other DMV11s or DMP11s. In point-to-point configurations, the DMV11 can communicate with other DMV11s, DUP11s, DMR11s, or DMP11s.

The DMV11 is a single quad-height module. It is compatible with Digital's family of modems and with the Bell 200 series of modems and their equivalents.

- *DPV11*

  The DPV11 is a single-line, synchronous programmable interface that provides local or remote interconnection between the supermicrosystem and other computer systems with EIA RS232- C/CCITT V.28 or EIA RS232-C/CCITT V.11 interfaces. It operates at speeds as great as 56,000 bits/second at half- or full-duplex with full-modem control. The DPV11 is programmable for either byte-oriented protocols (DDCMP or BISYNC) or bit-oriented protocols (SDLC or HDLC). The DPV11 is suited for interfacing to medium-speed synchronous lines for remote batch and remote job-entry applications.

  The DPV11 is a single, dual-height module. It is compatible with Digital's family of modems and with the Bell 200 series of modems and their equivalents.

- *KMV11*

  The KMV11 is a high-performance, direct-memory access, single-line, programmable communications controller that provides local or remote interconnection between MicroPDP-11 systems and other computer systems with EIA RS232-C/CCITT V.28, EIA RS422/CCITT V.1, or EIA RS423/CCITT V.10 interfaces. It is capable of communications speeds as great as 64,000 bits/second. The KMV11 utilizes the MICRO/T11 processor to perform user-defined communications functions, thereby freeing the host to do more application computations.

  The KMV11 can be programmed in synchronous or asynchronous modes. It is implemented as a single quad-height module. The KMV11 also provides full-modem support for Digital's family of modems, the Bell 200 series of modems or their equivalents, and European-PPT-app roved modems.

Q-bus Realtime Interfaces

- *DRV11-JP*

  The DRV11-JP is a general purpose, program-controlled, parallel-line interface. It contains 64 bidirectional input/output lines configured as four 16-bit ports. It is also bit interruptible on as many as 16 lines. Interrupt vectors may have fixed or rotating priorities. The DRV11-JP is a single, dual-height module.

- *DRV11-WA*

  The DRV11-WA is a general purpose, direct-memory access parallel-line, interface with 22-bit addressing capability. It permits block-mode DMA data transfers at rates as great as 250 Kwords/second in single-cycle mode, and as great as 500 Kwords/second in burst mode. The DRV11-WA is a single, dual-height module.

- *AAV11*

  The AAV11 is a four-channel digital-to-analog (D/A) converter module for MicroPDP-11 systems that includes control and interfacing circuits. It has four D/A converters, a dc-to-dc converter that provides power to the analog circuits, and a precision voltage reference. Each channel has its own holding register that can be addressed separately and provides 12 bits of resolution. The AAV11 is a single, dual-height module and is available as an add-on option only.

- *ADV11*

  The ADV11 is a 12-bit, successive approximation analog-to-digital (A/D) converter module for MicroPDP-11 systems that samples analog data at specified rates and stores the digital equivalent value for processing. A multiplexer section can accommodate as many as 16 single-ended or eight quasi-differential inputs. The converter section uses a patented auto-zeroing design that measures the sample data with respect to its own circuitry offset and, therefore, cancels out its own offset error. The ADV11 is a single dual-height module and is available as an add-on option only.

- *AXV11*

  The AXV11 is an analog input/output module for MicroPDP-11 systems that accepts up to 16 single-ended inputs or as many as eight differential inputs, either unipolar or bipolar. The AXV11 also has two separate digital-to-analog (D/A) converters. Each D/A converter has a write-only register that provides 12-bit input data resolution. On receiving the data, the AXV11 changes it to an analog output voltage. The AXV11 is a single dual-height module and is available as an add-on option only.

- *KWV11*

  The KWV11 is a 16-bit programmable clock counter for MicroPDP-11 systems only that provides a variety of means for determining time intervals or counting events. It can be used to generate interrupts to the processor at predetermined intervals, or to synchronize the processor ratios between input and output events. It can also be used to start the ADV11 analog-to-digital converter either by clock counter overflow or by firing the Schmitt trigger.

The KWV11 can be operated in any of four programmable modes—single interval, repeated interval, external event timing, and external event timing from zero base. The KWV11 is a single dual-height module and is available as an add-on option only.

▪ UNIBUS System Options

UNIBUS Memory Options

There are several memory options available for UNIBUS PDP-11 systems. Each provides the capability to perform direct-memory access in which data is transferred without processor intervention.

With the PDP-11/84 and the MicroPDP-11/83 systems, Private Memory Interconnect (PMI) provides fast memory transfer through serial double-word transfers. PMI shares some of the existing bus hardware, but also has its own bus architecture and data transfer protocols. For further details on PMI, refer to Chapter 6.

- *MSV11-JB/JD*
  The MSV11-JB/JD is a 1-Mbyte ECC PMI MOS memory for the PDP-11/84.

- *MSV11-JC/JE*
  The MSV11-JC/JE is a 2-Mbyte ECC PMI MOS memory for the PDP-11/84.

UNIBUS Storage Options

- *RA81, RA60, and RX50 Disk Storage*
  RA81, RA60, and RX50, described earlier in this chapter, provide a wide range of disk storage options for PDP-11/84.

- *UNIBUS Disk Controllers*
  Digital provides the UDA50 disk controller for RA81 and RA60 fixed-disk drives. The RUX50 UNIBUS controller is provided for the RX50 dual-floppy diskette drive.

- *TSU05 and TK50 Tape Storage*
  The TSU05 is a 40-Mbyte magnetic tape drive and the TK50 is a 95-Mbyte cartridge-tape drive for the PDP-11/84.

  Digital provides the TUK50 UNIBUS controller for TK50 drives.

  See the descriptions of these devices provided under the Q-bus options section above for further details.

- *TU81-Plus*

  The TU81-Plus is an improved version of the TU80 magnetic tape storage subsystem for PDP-11/84 systems. The TU81-Plus brings users faster operation through the use of a 256-Kbyte cache buffer for optimized performance. Automatic speed control and pre-fetching of commands and data further enhance performance.

  High data integrity is provided because the TU81-Plus cache can be turned off for applications in which data is at risk. The TU81-Plus also provides read-after-write verification and automatic two-track error detection and correction during operation.

  The TU81-Plus is ideal for data interchange because it provides industry-standard recording at 6250 bits per inch and 1600 bits per inch.

  The TU81-Plus is designed for the office or the computer room environment. It also is packaged with space for an RA-series Winchester disk drive at the bottom of the cabinet.

UNIBUS Communications Options

- *DUP11, DMR11, DMP11*

  The DUP11, DMR11, and DMP11 are single-line synchronous communications interfaces for the PDP-11/84.

- *DHU11*

  The DHU11 is a 16-line asynchronous DMA multiplexer for local and remote interconnection between the PDP-11 and EIA RS-232-C/CCITT V.28 or EIA RS-423-A/CCIT V.10 terminals.

- *PCL11*

  The PCL11 is a multipoint parallel communications link for the interconnection of up to 16 processors in a local distributed network.

UNIBUS Realtime Interfaces

- *DR11*

  The DR11 is a general purpose digital interface, permitting bidirectional 16-bit parallel transfers between user devices and the UNIBUS.

- *DRS11/DSS11*

  The DRS11/DSS11 input/output devices provide UNIBUS computers with efficient monitoring and control functions useful for a variety of scientific and industrial applications.

- *DRU11*
  The DRU11 is a 16-bit parallel DMA interface that allows users to interface a UNIBUS system to instruments and other devices that require large amounts of data to be passed to or from them at moderate or high speeds.

- *IEEE Interfaces*
  The IEU11 and IEC11 interfaces conform to the IEEE STD. 488-1978 for compliant test equipment. These controllers can support up to 15 instruments, including the controller itself.

- *KW11 Clock*
  The KW11 is the UNIBUS programmable realtime clock. The KW11 permits program-selectable interrupts of 100 kHz, 10 kHz, and line frequency or external signals counted down by 16-bit counters with automatic reload.

▪ Terminals, Printers, and Modems

Digital offers a complete line of videoterminals, printers, and modems for the supermicrosystems. A terminal or device can be selected that incorporates the features that the application requires. A detailed description of all the videoterminals, printers, and printing terminals can be found in the *Terminals and Printers Handbook*.

Videoterminals

Videodisplay terminals use a cathode-ray tube (CRT) screen for output and a typewriter-like keyboard for input. Alphanumeric videoterminals are capable of displaying letters, numbers, and special characters in a fixed format. Graphics videoterminals can individually manipulate picture elements on the display screen and can represent graphs, charts, and pictures.

- *VT300 Videoterminal Series*
  The VT300 series, including the VT330 monochrome and the VT340 color graphics terminals, is Digital's new series of videoterminals. These terminals are especially well-suited to networking environments. A single terminal can connect two lines, log into two hosts, and run two sessions simultaneously.

  VT300 features include:

  - 800 by 500 resolution.
  - Two piece packaging.
  - 14-inch flat viewing surface.
  - 16 colors.

- Fast graphics capabilities.
- Double terminal capabilities.
- No fan noise.

The VT300 series represents significant advancements in both design and performance, and is the best choice for modern network-intensive computing environments.

- *VT200 Videoterminal Series*
  The VT200 series is Digital's newest offering of videoterminals. The VT200 terminals include all of the universal features of the VT100 series of videoterminals. The VT200 units are smaller in size than the VT100 units and include low-profile packaging. A series of setup menus simplifies tailoring the terminal to the application.

- *VT220*
  The VT220, a monochromatic, text-only, videodisplay terminal incorporates full VT100 functionality. The terminal features a 12-inch nonglare screen, low-profile packaging, and an adjustable monitor. The VT220 terminal includes VT52 terminal emulation, advanced-video features, a built-in printer port, and U.S.A./European modem controls. Its international capabilities include a multinational character set, universal power supply, and both 20-milliampere and EIA interfaces. A plain-language setup menu, programmable function keys, and a selective-erase feature combine to make the VT220 terminal easy to use. Operator-oriented features such as split screen, bidirectional smooth scrolling, double-height and double-width characters, and reverse video allow the VT220 terminal to be used for many applications.

- *VT240 and VT241*
  The VT240 is a monochromatic text and graphics videodisplay terminal. It incorporates all the features of the VT100 family and the VT220 terminal, is completely self-contained, and requires no upgrading.

  The VT240 terminal supports the industry's graphics standards by generating full bit-mapped graphics in both ReGIS and Tektronix 4010/4014 emulation. ReGIS is Digital's general purpose graphics descriptor. It allows pictorial data to be created and stored very easily. By connecting this terminal to a graphics printer, such as the LA210, the contents of the display can be reproduced. The VT240 terminal consists of the same keyboard that is used for the VT220 terminal, a monitor, and a system box. The system box contains the power supply, control-circuit board, and electrical connectors.

The VT241 terminal offers all the capabilities of the VT240 terminal and includes a color monitor for four-color text and graphics output.

Printers and Printing Terminals

Several printers and printing terminals are available for operation with the supermicrosystems. These devices include large, free-standing units and small, desktop units.

- *LA210 Letterprinter*

The LA210 Letterprinter is a microprocessor-driven, medium-speed, wide-carriage, receive-only, multimode printer that offers similar functionality to the LA100 personal-computer model plus the addition of IBM personal-computer compatibility. The LA210 uses conventional impact dot-matrix printing technology. Dot formations may be of either letter quality (lower speed/higher density), draft quality (higher speed/lower density) and a binary-print mode that allows the host computer to define all dots printed (graphics mode). The three basic modes of printer operation cover a wide range of applications. The LA210 has been designed with the following standard capabilities:

- 240 characters/second draft printing speed.
- 40 character/second letter-quality printing speed.
- Compatibility with Digital and most IBM bit-map graphics printing applications.
- Compatibility with IBM-PC, IBM-XT, and IBM-AT personal computers and with many IBM personal-computer emulators.
- Equipped with Courier-10 font with over 30 optional font cartridges available.
- Prints in ten languages plus VT100 line-drawing characters.
- Wide carriage prints up to 217 characters on 15-inch paper.
- Styled for the office environment.
- Forms-handling tractor with acoustic shield.
- 2,000-character input buffer.
- Equipped with EIA RS232-C interface.
- 500-million character laminated printhead delivers exceptional print quality.

- *LQP03 Letter-quality Printer*

  The LQP03 letter-quality printer is a desktop, full-character, impact printer especially designed for use with all of Digital's supermicrosystems. The LQP03 includes an expanded character set contained on a single, 130-petal daisywheel. The daisywheel allows use of all of Digital's multinational characters on one wheel, and provides scientific, mathematic, or other special characters on another.

  An optional bidirectional forms tractor is customer-installable and handles a variety of fanfold paper including continuous preprinted forms. It permits the paper to be scrolled forward or backward while printing. The single-tray cutsheet feeder option is designed to automatically feed precut paper to the LQP03 in either portrait or landscape fashion.

  The LQP03 runs on a variety of Digital's operating systems and layered software products, plus software packages from other manufacturers. Depending on the software support, the LQP03 can also perform overprinting, boldfacing, underlining, subscripting, and superscripting. The LQP03 includes a standard serial interface for compatibility with existing Digital printers.

- *LA120 DECwriter III Printer*

  The LA120 freestanding printing terminal is a sturdy, high-performance device with a reputation for reliability. It prints on multiple-copy, tractor-fed paper from 3 inches to 14.8 inches wide, at a maximum speed of 180 characters/second—fast enough to print a typical one-page memo in 20 seconds. Because of its 1000-character buffer, bidirectional printing, and ability to skip quickly across spaces, the LA120 printer maximizes printing throughput.

  The standard LA120 printer character set is U.S.A./United Kingdom. Character sets for Europe or the APL programming language are optional. Characters can be printed in eight sizes and in six choices of vertical-line spacing. Characters are formed by an impact dot-matrix printhead in a 7 by 7 dot format. Both keyboard send/receive (KSR) and receive-only (RO) versions are available.

- *LA100 (Letterprinter 100 and Letterwriter 100)*

  The LA100 printing terminals provide flexibility in a tabletop unit. The Letterwriter 100 is a keyboard send/receive (KSR) terminal, and the Letterprinter 100 is receive-only (RO). Both can print on friction-fed paper or on optional tractor-fed paper.

The impact dot-matrix printer offers a choice of print quality and speed. In draft mode, it prints 240 characters/second maximum with a 7 by 9 print matrix, and letter-quality mode outputs a 33 by 18 print matrix at 30 characters/second. In between these modes is the optional memo-quality mode with its 33 by 9 print matrix and 80 characters/second. Standard fonts included with the LA100 are Courier-10 and Orator-10. Optional fonts include Gothic-10, Symbol-10, Courier-12, and many others. Fonts can be selected by the system or from the keyboard.

- *LA75 Personal Printer*

  The LA75 is a low-cost tabletop printer. It prints at rates of 250, 125, 42, and 32 characters/second plus full graphics.

  LA75 features include:

  - Nine built-in character sets.
  - Compact desktop footprint (4.8-in high by 16.8-in wide by 13.6-in deep.
  - 2047-character input buffer.
  - Quiet operation for busy office environments.

- *LPS40 (PrintServer40)*

  The PrintServer40 is a monochromatic printing subsystem for local area networks, local area systems, and single hosts via Ethernet. The PrintServer40 provides a high-quality print mixture of text, graphics, and natural images.

  The PrintServer40 offers near-typeset print quality and readability, as well as multiple fonts and point sizes.

- *LN03 Laser Printer*

  The LN03 is an eight-page/minute laser printer that can be used as a shared resource for supermicrosystem users and as a remote printer for local area networks.

  The LN03 is a compact printer, packaging the print engine and the controller in one tabletop unit. With resolution of 300 dots by 300 dots/inch, the print quality of the LN03 is outstanding on both cutsheet paper and transparencies. The input paper capacity of the LN03 is 250 sheets, complemented by a paper path that automatically collates the pages in a 250-sheet output tray. In addition to the 16 fonts resident in the LN03, a wide variety of optional fonts will be offered in both host media and cartridge formats. This flexibility reflects the LN03's capability to accept two ROM (precoded typefaces) and/or RAM cartridges, the latter being used to receive fonts or forms that can be downline-loaded from a host.

- *LCG01 Color Printer*

The LCG01 is an intelligent color, ink-jet, graphics printer that produces high-quality presentation graphics on paper and transparencies. The LCG01 contains its own graphics processor that handles the display file processing from the CPU. This feature decreases processing time and frees the host CPU for other tasks.

The LCG01 can store up to five fonts in local memory and offers brilliant output from more than 200 colors. It supports ReGIS, GIDIS, NAPLPS, and BIT MAP IMAGE (Color Sixel format) protocols. It is compatible with VAX DECslide, VAX DECgraph, DATATRIEVE, VAX VTX, and many third-party graphics generation packages.

The following are the main features of the LCG01:

- Print speed: less than 2 minutes per page.
- Print resolution: 154 dots/inch.
- Colors: 216.
- Image size: A size (7.5–9.95) and A4 (7.27–9.95).
- Print colors: yellow, magenta, cyan, red, green, blue, black, and white.
- Tabletop model.
- Quiet (less then 58 db).
- Interface is EIA or 20 mA.

- *LP27 and LP25 System Printers*

The LP27 and LP25 are band printers for medium-duty printing workloads in a one-shift environment. They were designed for applications such as data processing, commercial, scientific, industrial, and educational. The LP27 and LP25 operate at medium speeds—up to 300 and 600 lines per minute.

Modems

Digital provides several modem units to enable the supermicrosystems to communicate with remote terminals and systems over standard telephone lines. These units can be placed on a desk or table or mounted into a cabinet or rack.

- *DF112 Modem*

The DF112 modem connects directly to a modular telephone jack and provides both synchronous and asynchronous communications over dialup and private/leased telephone lines. The DF112 unit is comparable in operation to Bell 103/21A modems.

Over dialup lines, it provides synchronous communication at 1,200 bits/second (full duplex) and asynchronous communication at 300 or 1,200 bits/second (full duplex). Over private/leased lines, the DF112 provides synchronous and asynchronous communication at 1,200 bits/second (full duplex). The DF112 can be used with any standard telephone for manual-call origination or is available with serial asynchronous autodial capability that can be controlled by a computer system or terminal connected to the EIA port.

The following are the main features of the DF112:

- Dual application modem—supports 0 to 300 and 1,200-bit/second asynchronous operation, or 1,200-bit/second synchronous operation.
- Conforms to FCC Part 15 requirements.
- Approved for direct connection to Public-Switched Telephone Network by FCC Part 68.
- Quick and easy installation, maintenance, and operation.
- Automatic originate, answer, and disconnect capabilities.
- Single serial-data port for both automatic dialing with autocall feature and normal data.
- Multiple-modem modules certified for operation in Canada by the Department of Communication.
- Multiple-modem modules support EIA cables up to 200 feet.

▪ *DF124 Modem*

The DF124 modem connects directly to a modular telephone jack and provides both synchronous and asynchronous communications over dialup and private/leased telephone lines. The DF124 is comparable in operation to the CCITT V.22 bis and Bell 212A modems.

Over dialup lines, it provides synchronous and asychronous communication at 1,200 or 2,400 bits/second (full duplex). Over private/leased lines, the DF124 again provides synchronous and asynchronous communication at 1,200 or 2,400 bits/second (full duplex). The DF124 can be used with any standard telephone for manual-call origination or is available with serial asynchronous autodial capability that can be controlled by a computer system or a terminal connected to the EIA port.

The following are the main features of the DF124:

- Dual application modem—supports 1,200-bit/second or 2,400-bit/second synchronous and asynchronous operation

- Compatible with the CCITT V.22 bis modem at 2,400 bits/second. Compatible with the Bell 212A modem at 1,200 bits/second.
- All other features of the DF112.

- *DF100 Modem Enclosure*
  The DF100 modem enclosure houses up to 12 DF100-series modules and can be installed into a 19-inch (48.3-centimeter) cabinet or rack. Can connect to either the Public Switched Telephone Network (PSTN) or the Private/Leased Telephone Network (P/LTN).

  The following are the main features of the DF100:

  - Houses as many as 12 modem modules for cost reduction and space savings.
  - Single complete unit that is easily installed in a cabinet or rack.
  - Contains internal power supply and provides space for optional power regulator.
  - Permits easy servicing with online replacement of modem modules.
  - Device and communication line cables are easily connected by the user.

Additional Options

A wide variety of other options are available for PDP-11 systems. Consult your Digital representative for details.

Additional Documentation

- *PDP-11 Systems and Options Catalog*
- *Networks and Communications Buyer's Guide*
- *Terminals and Printers Handbook*

Chapter 10 ▪ Networks

▪ Introduction

Digital offers extensive capabilities that permit the linking of computers and terminals into flexible configurations called networks. Networks increase the efficiency and cost-effectiveness of data-processing operations.

Networking allows computer systems and terminals, whether located around a facility or around the world, to share resources and exchange information, files, and programs. The smaller computers in a network have access to the powerful capabilities of larger systems, while the larger computers can take advantage of smaller dedicated systems chosen for specific application environments.

Distributed processing is the physical placement of computers where they are needed. As organizations become more complex and develop more sophisticated demands for computer resources, the ability to network processors and share computing resources becomes increasingly important.

▪ Types of Systems

Two general types of systems can be implemented for most processing functions—the stand-alone system and systems connected by a network. With the stand-alone system, all data is entered manually at the system by an operator or from locally connected machines or instruments. In a network-connected system, information can be entered locally or transferred to or from other systems through an electrically connected network link.

Stand-alone Systems

A single-user stand-alone system, shown in Figure 10-1, can process information received from several sources. Data can be entered from the console terminal keyboard, read from a diskette in the disk drive, or received from an external machine or instrument. Only one user at a time can operate the system.

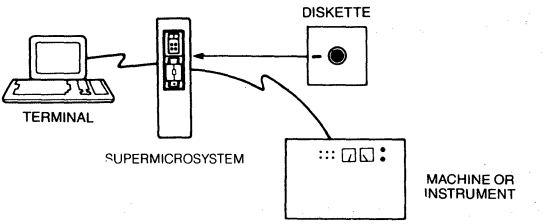


Figure 10-1 ▪ Single-user Stand-alone System

A multiuser stand-alone system, an expansion of the single-user system, allows several users to share a single processor concurrently. Several terminals can be connected to the processor, and the processing is timeshared between users as shown in Figure 10-2. Data is entered from the same sources as the single-user system.

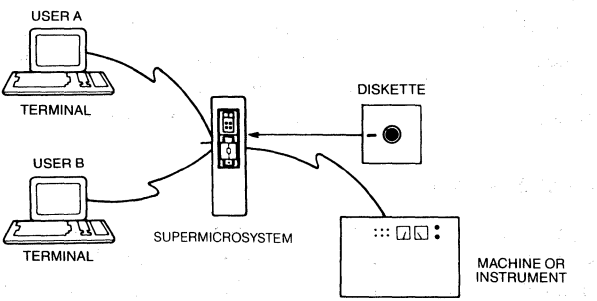


Figure 10-2 ▪ Multiuser Stand-alone System

For realtime or runtime applications requiring that information be shared between systems, several methods of communications are available.

As shown in Figure 10-3, Departments A and B of a small company both need sales figures, and Department B also needs the inventory-level information of Department A. Normally each department would be required to enter the sales figures from the terminal keyboard of each system. Department B would also be required to read the display of Department A and to enter the information manually into the Department B system from the keyboard.

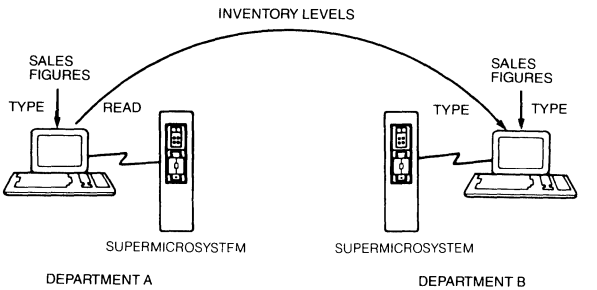


Figure 10-3 ▪ Manual Data Entry

A more efficient method of communications is shown in Figure 10-4. Department A manually enters the sales figures, processes the information, and records both the sales figures and inventory levels onto the diskette. No manual entry would be required by Department B. Once the diskette is received by Department B, the information can be processed.

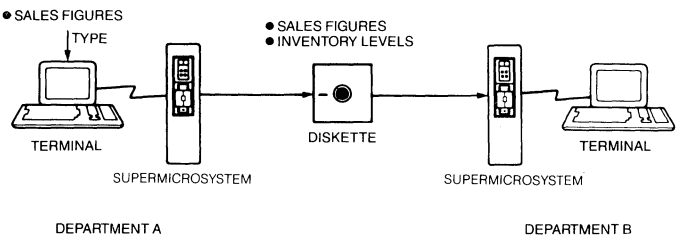


Figure 10-4 ▪ Recorded Data Entry

Network-connected Systems

Two or more stand-alone systems can be electronically connected by a network. The network enables the efficient transfer of information between systems (shown in Figure 10-5).

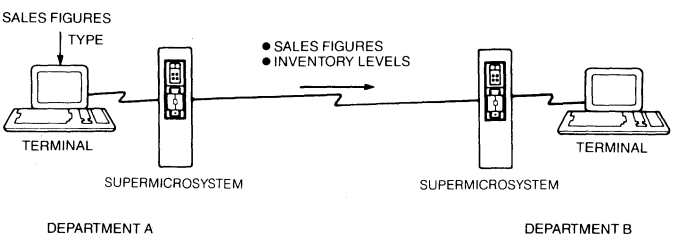


Figure 10-5 ▪ Two-system Network

Networks also enable systems located in different cities to communicate with each other as shown in Figure 10-6. Through the use of modems, information is transferred between offices in different locations over standard telephone lines.

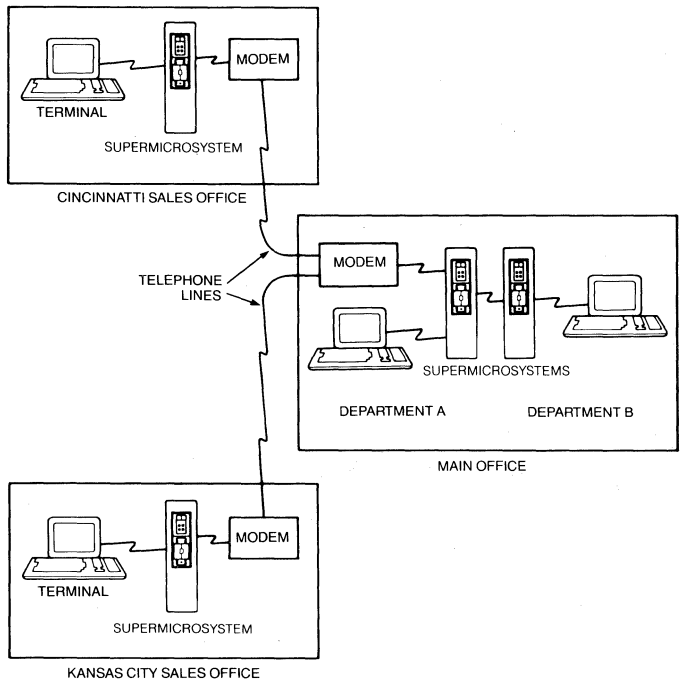


Figure 10-6 ▪ Remote Communications Network

▪ Digital Network Architecture

Digital Network Architecture (DNA) is a set of hardware and software networking capabilities that supports communications between Digital's systems, and between Digital's systems and other manufacturers' systems.

Digital-to-Digital communications are permitted through protocols, or rules, that are defined by the DNA. DNA protocols are based on the architectural models for open-systems interconnection created by the International Standards Organization (ISO). These rules govern the format, control, and sequencing of message exchange among Digital computers.

Internet products provide a means for Digital's systems to communicate with systems built by other manufacturers. These products emulate common communications protocols and are data-transfer facilitators rather than hardware emulators.

DNA Structures

The lowest layer of the DNA structure, shown in Figure 10-7, is the *physical link layer*. This layer governs electrical and mechanical transport of information between systems that are connected. Computer systems can be physically connected by cables, fiber optic lines, microwave transmissions, or switched networks such as telephone lines.

In addition to the physical connection, the electrical signals on the lines must be properly defined. The signal characteristics and data rates are all defined by the hardware comprising the interface module and the transmission link.

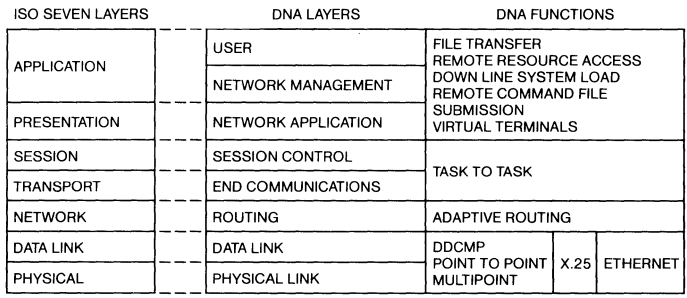


Figure 10-7 ▪ Digital Network Architecture Structure

The next highest layer of the DNA structure is the *data link layer*. The data link layer can prepare messages for transmission according to a specified protocol, check the integrity of received messages, and manage access to the channel. The data link is usually implemented by hardware and software. For a simple asynchronous interface, the hardware contribution to the data link is minimal. With other devices, such as the DEQNA Ethernet interface, almost all of the data link layer is implemented in hardware.

Because of the data link layer, the *routing layer* can rely on error-free connection to adjacent nodes. It addresses messages, routes them across intervening nodes, and controls the flow of messages between nodes. The routing layer and higher layers are implemented in software.

Because the routing layer establishes the path, the *end communications layer* can address the end machine without concern for route, and can perform end-to-end error recovery.

The *session control layer* manages the system-dependent aspects of a communications session. For instance, when the end communications layer reliably delivers a message from another manufacturer's system in the network, the session control layer interprets that message for acceptance by the system software.

The *network application layer* converts data for display on terminal screens and printers.

The *network management layer* monitors network operations by logging events and collecting statistical and error information. It also controls network operations by tuning network parameters and testing nodes, lines, modems, and interfaces.

The *user layer* provides services that directly support the user and application tasks such as resource sharing, file transfers, and remote file access.

▪ Typical Network Configurations

Systems can be interconnected in a network in many different configurations, some of which are described in this section. Each of the participating systems in a network is called a *node*.

Two nodes communicate through a link or connection. Figure 10-8 shows supermicrosystem nodes linked in a network.

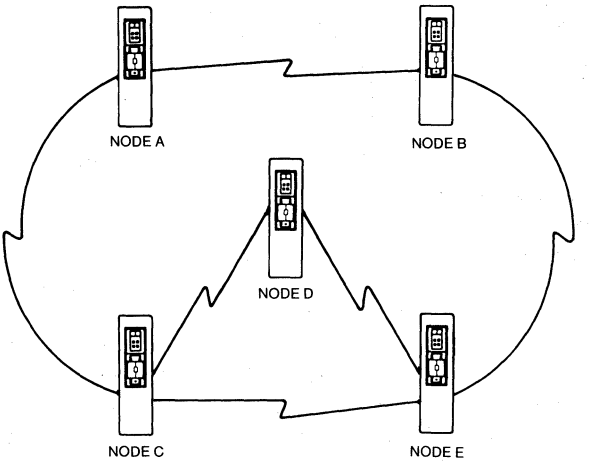


Figure 10-8 ▪ Nodes Linked in a Network

The physical links can be cables, fiber optic lines, microwave transmissions, and other conductors that form the data paths between two nodes such as Node A and B. Logical nodes exist whenever two nodes can communicate through a physical link or through another node such as between Node A and D.

Node C is required to route and transfer the message. This configuration increases the transmission time between nodes but decreases the cost because fewer physical links are required. As the number of physical links increases in a fully connected network, the resulting costs increase. Figure 10-9 shows the relationship of the number of nodes to the physical links.

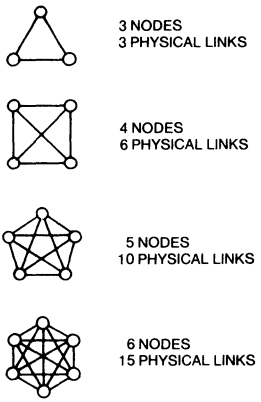


Figure 10-9 ▪ Fully Connected Network Nodes

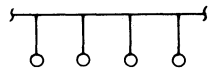
One method of reducing the number of physical links is the multidrop or multipoint network shown in Figure 10-10. More than two nodes can be connected to the same physical link. Each node is required to determine which of the messages are dedicated to that node and to manage access to the links thus avoiding conflict between nodes.

In a network with centralized control, one node such as a mainframe computer is required to determine which of the remaining nodes can send messages, where the messages will be sent, and the length of the messages. In a network with distributed control, each node recognizes a procedure that allows the node to access the network independently.

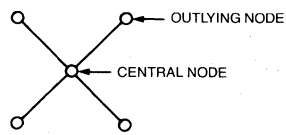
In a star network, shown in Figure 10-10, one node is designated as the central node and all other outlying nodes are physically connected to it. This network is efficient because most of the communications are between the central node and one outlying node, such as a timesharing network or a shared word processor system. Because all messages must be transferred through the central node, the node must process many transactions when the message rate is high. If the central node fails, all transactions halt. In some networks, distributed control may be implemented, thereby decreasing the load on the central node.

In a ring network, each node is physically linked to two adjacent nodes, as shown in Figure 10-10. Messages circulate around the ring and each node retransmits the messages not addressed to itself. This method is less complex than the generalized routing method because each node is required to transmit the message only to the adjacent node.

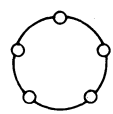
Distributed control in a ring network may be implemented through token passing. A special token message circulates around the ring and a node can claim access to the network by receiving the token message as it passes through.



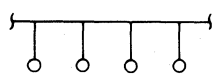
MULTIPOINT-NETWORK LINK



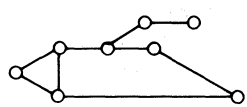
STAR NETWORK



RING NETWORK



BUS NETWORK



UNCONSTRAINED NETWORK

Figure 10-10 ▪ Network Configuration Types

The bus network, shown in Figure 10-10, is similar to a multidrop link. Messages placed on the shared physical link reach all nodes, and the intended receiver must recognize the message's address in order to receive the transmission. None of the nodes, however, have to route or retransmit messages intended for other nodes. A bus network typically uses distributed control.

There is no single point of failure. The Ethernet local area network is a bus configuration.

An unconstrained network is shown in Figure 10-10. The placement of physical links is usually determined by the cost of the physical connections, by the number of messages to be transferred, and by the network reliability requirements. Some of the nodes in this network may have the capability to route messages to other nodes. Long-distance packet-switched networks are often unconstrained.

▪ Types of Links

Several interface options are provided for the supermicrosystems to support the implementation of the physical and data link levels of the DNA.

Ethernet and ThinWire Ethernet Links

As computer systems such as word processors, workstations, personal computers, and departmental systems become more numerous and accessible, an integrated communications network can increase productivity and reduce data processing costs.

Ethernet provides local area network technology through a hardware and software combination to create a physical communications channel between systems. This allows large amounts of data to be exchanged at high rates between systems located within limited distances.

Figure 10-11 shows the physical links and the DNA layers that perform the networking functions of Ethernet. The DEQNA is the interface module that provides the internal connection to the Q22 bus of the supermicrosystems. The hardware elements include the DEQNA Ethernet interface, the transceiver cable, and the transceiver and coaxial cable. The coaxial cable can be in lengths of as great as 500 meters (1,640.5 feet). The transmission rate can be as many as 10 Mbits per second.

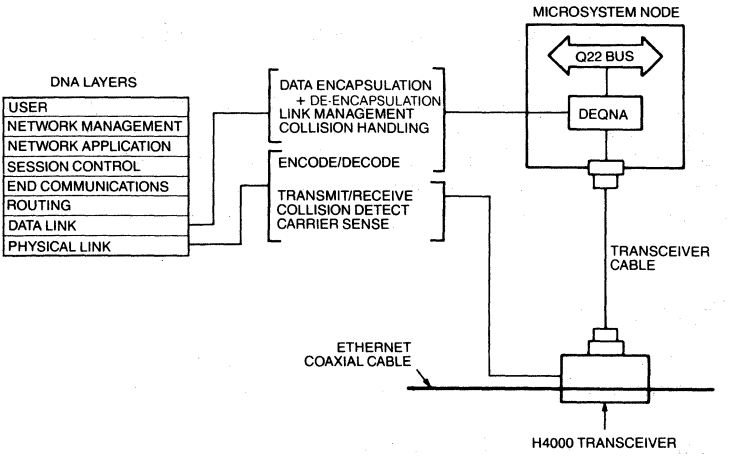


Figure 10-11 ▪ Ethernet Physical Link and Data Link Layers

Figure 10-12 shows a small-scale Ethernet configuration using a single coaxial cable. Each cable segment can include as many as 100 transceivers or nodes.

A transceiver cable, which can be as many as 50 meters (164 feet) in length, connects the transceiver to the DEQNA Ethernet interface.

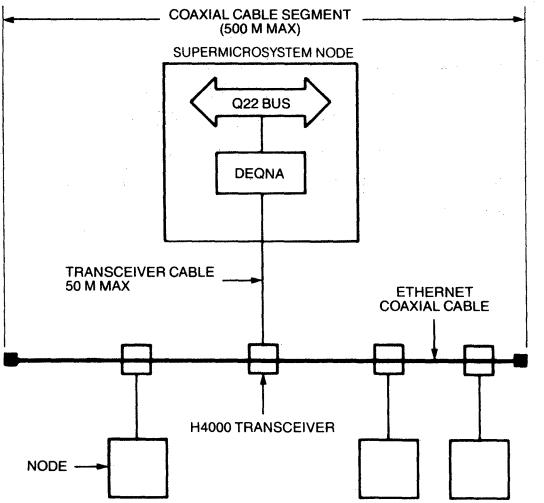


Figure 10-12 ▪ Small-scale Ethernet Configuration

A medium-scale Ethernet configuration is shown in Figure 10-13 and includes a repeater that connects two cable segments. Each coaxial cable can be a maximum of 500 meters (1,640.5 feet) in length and can operate with up to 100 transceivers, including the repeater transceiver.

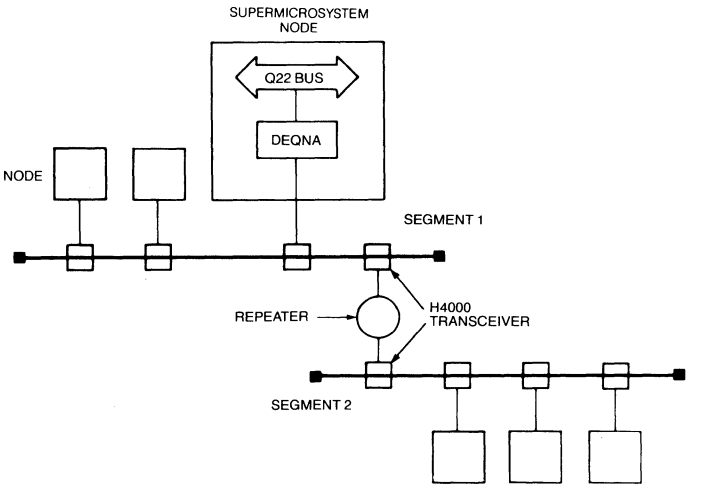


Figure 10-13 ▪ Medium-scale Ethernet Configuration

A large-scale Ethernet configuration is shown in Figure 10-14 and consists of five coaxial cable segments. The segments are connected by repeaters and can attach as many as 1,024 nodes. Each segment is connected by remote repeaters with a maximum distance of 1,000 meters (3,281 feet) between repeaters.

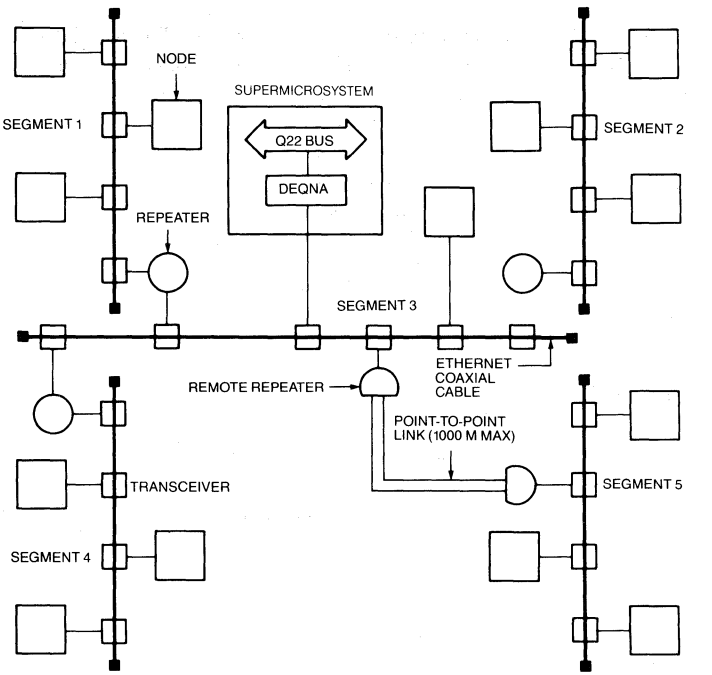


Figure 10-14 ▪ Large-scale Ethernet Configuration

The transceiver and tap can be installed on an operating cable with no disruption of the system operations. The transmit, receive, carrier sense, and collision-detection functions of the physical link layer are implemented in the transceiver.

The access control method used by Ethernet is called Carrier Sense—Multiple Access with Collision Detection (CSMA/CD). To transfer a message, a node monitors the transmissions on the cable to sense a pause between the data packets. The node continues to sense the data while initiating the transmission. If another node transmits simultaneously, both nodes will stop transmission and wait for a random interval of time before initiating another transmission.

The DEQNA Ethernet interface encodes and decodes the data exchanged with the transceiver. In addition, it implements the functions of the data link layer by encapsulating and de-encapsulating messages, handling collisions, and filtering received messages. The DECnet software provides the remaining functions of the message transfer such as routing, end communications, and session control.

As many as eight Ethernet nodes can be connected to a single transceiver using the DELNI Ethernet concentrator as shown in Figure 10-15. These nodes can include Q-bus processors with the DEQNA interface, or UNIBUS PDP-11 processors with the DEUNA Ethernet interface. For localized connections, as many as eight processor nodes can be interconnected using the DELNI concentrator without physical connection to the transceiver.

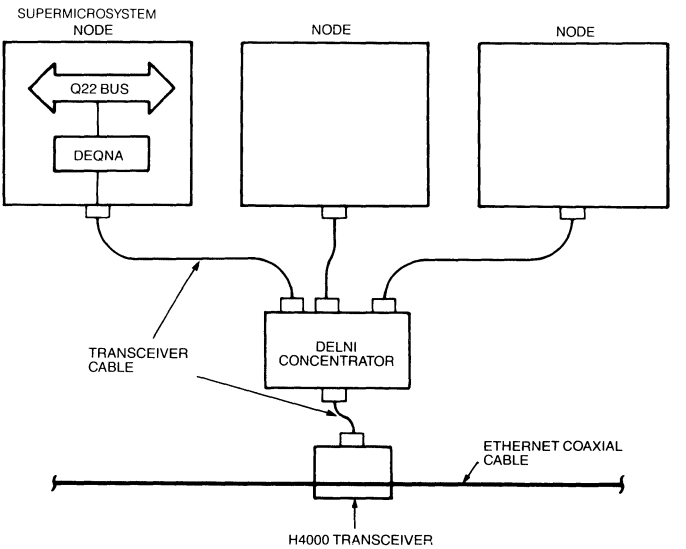


Figure 10-15 ▪ Ethernet Node with DELNI Concentrator

ThinWire Ethernet offers an alternative cabling system compatible with standard Ethernet. ThinWire provides full Ethernet capability for personal computers, workstations, and low-end systems in the office and other local work areas. This low-cost, flexible system uses RG58 C/U-type cable to connect up to 30 stations with one 185-meter segment. The following configurations are available:

- Single segment of up to 30 stations.
- Standalone network of up to eight segments for the interconnection of up to 232 stations.
- Standalone network of up to eight repeater hubs for the interconnection of up to 1,023 stations.
- Network in which any configuration above is connected to a standard Ethernet by means of a transceiver.

Asynchronous Links

The nodes in a network can be connected by asynchronous links when high-speed transfers and efficiency are not required. Data transmitted through the link is character-oriented. The characters can be five to eight bits in length, preceded by a start bit, and followed by stop bits. The time interval between the stop bits of one character and the start bit of the next character can vary, thereby reducing the efficiency of the data communications. The electrical and mechanical characteristics of the signals and interfaces are defined by standards created by the Electrical Industries Association (EIA) and the International Consultative Committee on Telegraphy and Telephony (CCITT).

The physical link can take one of two forms:

- EIA Standard Signals, Remote Connection—Communications between computer systems and devices over long distances can be implemented using modems and telephone lines as shown in Figure 10-16. The modems convert the system and device signal levels into signals acceptable by the telephone lines. The telephone lines can be private, leased, or part of the public-switched network. The remote device can be a terminal or any asynchronous interface of another computer system. The modems of each node must be of a compatible type.

- EIA Standard Signals, Local Connection—For local communications in the same area or within the same building, computer systems and terminals can be connected by EIA null modem cables as shown in Figure 10-17. The null modem cable transfers the serial EIA data and control signals between the nodes. The local device can be a terminal or an asynchronous interface of another computer system.

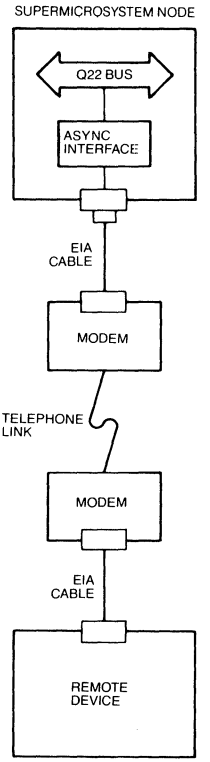


Figure 10-16 ▪ Remote Connection, Asynchronous Link

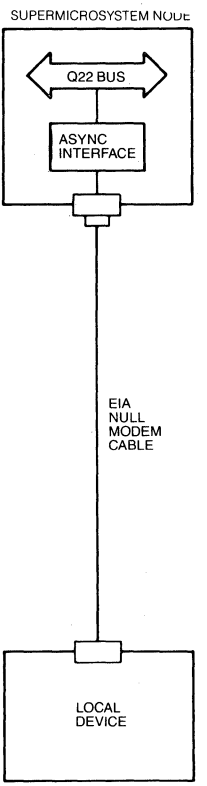


Figure 10-17 ▪ Local Connection, Asynchronous Link

For descriptions of optional asynchronous interface modules, refer to Chapter 9—System Options.

Synchronous Links

Synchronous links are used for communicating where speed and efficiency are important. Synchronous communications send a block of characters enclosed in a frame. The contents of the frame vary from one protocol to another, but they typically consist of text, identification of the beginning and the end of the frame, and information ensuring reliable reception of the text. Because the amount of extra information needed to complete the frame is fixed, the efficiency of synchronous transmission increases as the size of the textblock increases.

Some synchronous protocols, such as Digital's DDCMP and IBM's BISYNC, require the length of the text to be an integral number of characters or bytes. These are called character-oriented protocols. Others, including IBM's SDLC and the International Standards Organization's HDLC, allow the text length to be any number of bits. These are referred to as bit-oriented protocols.

The physical line can take one of three forms:

- Modem Connection, Remote Connection—For synchronous communications over long distances, the interface module is connected to a modem as shown in Figure 10-18. The modem connection can be specified by one of the EIA standards (RS232-C, RS422, or RS423) or by a CCITT standard (V.24, V.28, or V.35). The modem connects to a private line, a leased telephone line, or to the public-switched telephone network. The choice of modem and the line connecting the modems will depend on the speed of the data communications.

- Modem Eliminator, Local Connection—Connecting two local synchronous devices that interface via the EIA or CCITT standards requires a modem eliminator as shown in Figure 10-19. The distance between devices can be from several hundred feet to a few miles, depending on the speed of transmission and other factors.

- Integral Modem, Local Connection—The DMV11 series of synchronous interface modules contains an integrated modem that is compatible with the Digital DDCMP protocol. These interfaces can be used for point-to-point or multidrop-network configurations. Figure 10-20 shows the connections to the local-device interfaces.

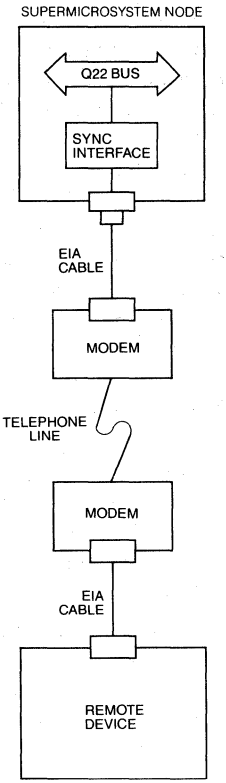


Figure 10-18 ▪ Remote Connection, Synchronous Link

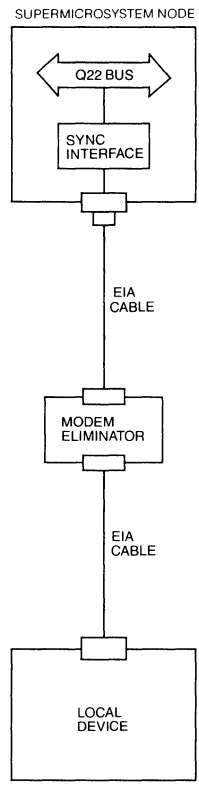


Figure 10-19 ▪ Local Connection, Synchronous Link

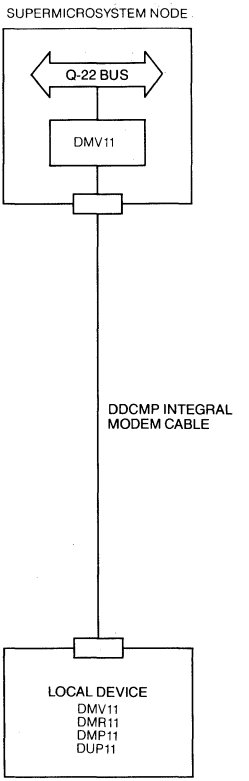


Figure 10-20 ▪ DDCMP Local Connection, Synchronous Link

For descriptions of optional synchronous interface modules, refer to Chapter 9—System Options.

▪ Network Software

After the network links are established, communications software makes the network functional within the DNA framework.

Digital's network software is broadly divided into three categories—DECnet, for communications among Digital systems; Internet, for communications with other manufacturers' equipment; and Packetnet, for communications with other participants in public packet-switched networks.

DECnet Communications

DECnet software supports communications among Digital computer systems. Data on the physical links is independent of system type, and the DECnet software converts the received data into formats that the operating system is prepared to accept. DECnet allows full use of the network by providing higher-level network functions. The following are the key elements:

- *Task-to-task communications* allow programs that are executing in different systems, under different operating systems, and written in different languages, to exchange information.
- *File transfer* supports the exchange of files between different operating systems.
- *Remote file access* allows the user to read, write, or modify files on another system.
- *Remote command file submission and execution* allow one computer system to direct another to execute commands that are resident on the remote system or sent as part of the request.
- *Downline loading* allows programs developed on a system with appropriate peripherals and resources to be transmitted to another system such as a small, memory-only system, for execution.
- The *network virtual terminal* gives a terminal user logical connection to a remote system with the same operating system; the terminal operates as if it were directly connected to the remote system.
- *Network management* provides the tools for monitoring and controlling network operation in a distributed environment.

Refer to Table 10-1 for a complete comparison of DECnet products.

Internet Communications

Digital's supermicrosystems can communicate with another vendor's equipment through software that emulates a protocol supported by that vendor. Although the name of the protocol may correspond to a specific device made by another manufacturer, the supermicrosystems emulate only the communications protocol used by that device and not the capabilities of the device.

Supermicrosystem nodes in an Ethernet link can also communicate with IBM systems through a Systems Network Architecture (SNA) gateway. The SNA gateway is an Ethernet node solely responsible for interfacing to an IBM system using IBM's System Network Architecture. Figure 10-21 shows the SNA gateway connections.

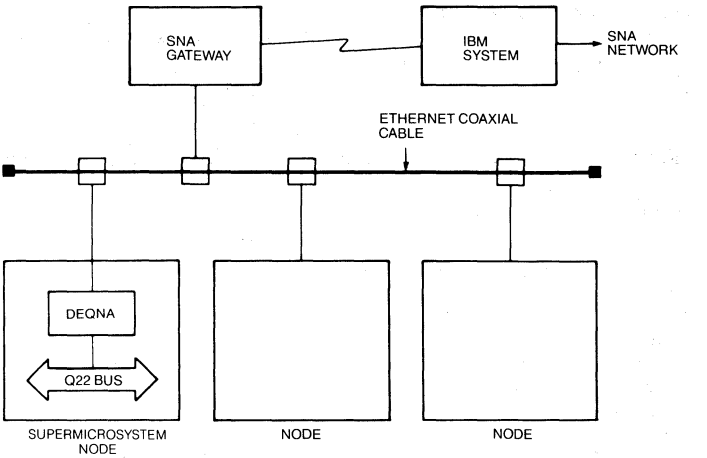


Figure 10-21 ▪ Ethernet to SNA Gateway Node

Packetnet System Interface

Public packet-switched networks can be an alternative to leased or dialup telephone lines for long-distance communications.

The charge is based on the volume of data transmitted rather than the fixed charge for a leased line. Access, speed, and reliability are better than that provided by a dialup line. The network also compensates for differences in transmission speeds between nodes and may offer services in addition to communications.

The Packetnet System Interface (PSI) software can coexist with, or operate as a layered product under, DECnet software. This allows DECnet facilities to be used between nodes connected through the packet-switched network and through leased or dialup lines. PSI makes communications possible with any other system (Digital or non-Digital) connected to the packet-switched network. PSI software supports task-to-task communications and remote terminal access to the supermicrosystems.

The communications protocol, part of the data link layer, is implemented in the hardware of some interface modules. For other interface modules, this protocol is provided by communications software.

Table 10-1 ▪ DECnet Products and Features

Feature	DECnet-VAX	DECnet-Micro/RSX	DECnet-11M-PLUS	DECnet-11M	DECnet-11S	DECnet/E	DECnet-ULTRIX	DECnet-RT
Task-to-task communications	X	X	X	X	X	X	X	X
File transfer	X	X	X	X		X	X	X
Remote file access	X	X	X	X	X[1,2]	X	X	X
Remote command file submission	X	X	X	X		X	X	X[2]
Remote command file execution	X	X	X	X	X	X	X	X[2]
Downline loading	X	X	X	X				
Networkvirtual terminal	X	X	X	X	X[2,3]	X	X	X
Network management	X	X	X	X		X	X	X

[1] Offers local users network access to remote file systems. Does not allow users on remote systems to access local files.

[2] Requester-only function.

[3] DECnet-11S does not support connection from remote virtual terminals.

▪ Digital PC Connection

The Professional 300 series and the Rainbow 100 series can be connected to the supermicrosystems with DECnet. However, the DECmate II and III must be connected to the supermicrosystems over an asynchronous line.

Two modes of communications are supported:

- *File transfer* supervises the transmission of an entire file between the DECmate series and a supermicrosystem without operator intervention. The supermicrosystem must have DX/11M running under RSX-11M or RSX-11M-PLUS, DX/RSTS running under RSTS/E or Micro/RSTS, or DX/VMS running under VMS. DECmate must have the WPS-8 Communications Package.
- *Terminal emulation* provides character-by-character communications that looks like a terminal to the supermicrosystem. DECmate, with the WPS-8 Communications Package, emulates an alphanumeric terminal (VT100 or VT52).

▪ Additional Documentation

- *Networks and Communications Buyer's Guide*
- *Networks Handbook*
- *Networks Guidebook*
- *Microcomputer Products Handbook*

Chapter 11 ▪ Services and Documentation

Like all of Digital's products, the PDP-11 systems and their software have been designed for reliability and manufactured to strict quality control standards. Digital's customer services organization is ready to follow up with quality support if it is required. Digital is the complete service vendor and has the products and tools to back its commitment to customer satisfaction.

▪ Field Service

Digital's Field Service is committed to customer satisfaction through quality delivery of a complete range of service products. This service organization complements Digital's hardware offerings and makes a significant contribution to Digital's position as an industry leader.

The Digital Field Service organization includes service engineers backed by administration and support personnel. Backing each service engineer are resources and support programs that help meet customer needs more effectively. Some of these resources and programs include:

- Computerized logistics network.
- Formalized training programs.
- Automated call-handling system.
- Remote diagnosis and support.
- Site-management guide.
- Action planning and problem escalation program.

The OEM Portfolio

For OEM purchasers of PDP-11s, Digital offers a comprehensive service program called the OEM Portfolio. This program is designed to meet the needs of OEMs whether they require full-service contract protection or backup maintenance support.

The OEM Portfolio gives OEMs a choice between the *Blue Chip Program*, which provides full service from Digital, or the *Partnership Program*, which provides qualified backup maintenance support for OEMs who do their own maintenance.

The *Blue Chip Program* is directed toward OEMs who sell Digital's services directly to their customers or buy our services in volume and then resell them to their end users as part of a total package. The Blue Chip Program lets these OEMs choose from a product menu that includes DECservice, Basic Service, Full System Service for personal computers, or Carry-In service contracts through our Digital Servicenters.

With the Blue Chip Program, OEMs who wish to sell our services directly to their customers can earn commissions. They receive an enrollment kit that contains all the materials they need to begin selling Digital services. OEMs who buy our services in volume can resell them as part of a total package sold to their end users. This may entitle them to a volume dollar discount based on an annual performance review.

OEMs enrolled in the Blue Chip Program can also choose from the following business options:

- Fixed price deinstallation and reinstallation when moving systems to end users' sites.
- New, improved trade show support.
- DECompatible Service for systems incorporating designated non-Digital hardware.
- Option to buy service in 30-day increments.
- OEM installation support.

The *Partnership Program* is designed for OEMs who buy Digital hardware components and maintain their systems and/or their end users' systems. The Partnership Program offers a wide range of onsite and offsite support. This program includes:

- Telephone technical support by senior engineers.
- Guaranteed onsite response by a senior engineer (when dispatched by telephone support).
- Emergency parts exchange service at designated Digital Servicenters.

OEMs in the Partnership Program can also choose from a variety of mail-in services, maintenance support tools, and educational services to maintain Digital hardware themselves.

Onsite Services

Onsite contract services are available for all of the PDP-11 systems, subject to minimum hardware configurations. These services provide corrective maintenance, preventive maintenance, and all applicable engineering changes to ensure that the systems and their options are operational and kept completely up-to-date. In addition to priority service, contractual maintenance allows Digital's customers to budget for their annual maintenance needs. The monthly contract charge covers all travel, labor, and materials. Users have a choice of tailored service agreements. In addition to basic coverage, extended hours are available to customers with critical applications that require special attention.

- *DECservice*

  The DECservice agreement is Digital's most comprehensive onsite service product. It provides committed response time including a 4-hour service response if the system is located within 100 miles of a Digital service location. DECservice also provides continuous repairs until the problem is solved, a program of preventive maintenance, installation of the latest engineering changes, and automatic escalation for complex problems. DECservice also offers the customer a choice of coverage hours—up to 24 hours, seven days per week.

- *Basic Service*

  The Basic Service agreement is the best alternative for customers whose requirements do not demand a fixed response time to calls for remedial maintenance, or for continuous work to resolve system-down situations outside coverage hours. Basic Service offers economical, full-service coverage. Calls for service receive priority status, second only to DECservice calls. It also provides preventive maintenance, installation of the latest engineering changes, and the guarantee that complex problems will be resolved by highly qualified service engineers. The hours of coverage are limited to first-shift business hours.

- *Per Call Service*

  Per Call Service is a noncontractual, time and materials service, available on an onsite and offsite basis. Onsite service is available Monday through Friday during standard business hours, from 8:00 a.m. to 5:00 p.m. Offsite Per Call Service is available through mail-in board replacement and carry-in system repairs.

- *Shared Maintenance Service*
  Shared Maintenance Service combines both onsite and offsite services. It is offered to qualified customers who perform their own preventive and remedial maintenance provided that they meet certain prerequisites. The onsite support provided by Digital is similar to a DECservice agreement except that customers, after paying a fixed monthly fee (a percentage of the DECservice maintenance charge), pay only for labor (at the local Per Call rate) and materials as required. Shared Maintenance Service features include onsite repairs, committed response, branch telephone support (technical), emergency access to branch logistics, extended coverage (optional), and remote diagnosis (optional where available).

- *DECompatible Service*
  DECompatible Service provides standard onsite services to selected non-Digital hardware products that are attached to Digital systems. DECompatible Service is provided under DECservice, Basic Service, or Carry-In agreements, depending on the particular device. The level of service and response time under this program is the same quality service as that available for Digital's hardware products.

- *Recover-All Service*
  Recover-All Service provides full product repair and/or replacement to Digital's hardware products that have been accidentally damaged, or damaged during incidents not covered under the other service agreements. Recover-All service expands the customer's service agreement to cover fire, water damage, natural disasters, power failure, sprinkler leakage, and theft. Recover-All also provides reimbursement for the cost of movement of equipment to a safe place, returning equipment to the site when safe conditions have been restored, removal of damaged equipment, transportation and installation of replacement equipment, replacement of fire protection chemicals, restoration of damaged Digital system software and customer data from backup disks and tapes, and data processing at a temporary location.

Offsite Services

Customers who do not require onsite services can take advantage of the Digital's offsite services that include Digital's Servicenters and DECmailer.

- *Servicenters*
  The Digital Servicenter is a carry-in repair center for Digital's terminals and smaller systems. The Servicenters offer low-cost repairs and have many convenient locations. At the Servicenter, the same quality service is provided that is given to onsite service calls. The Servicenter guarantees 2-day turnaround time. The customer may select from a variety of service offerings—contract, per call, or parts exchange. All Servicenter service and parts come with a 90-day warranty.

- *DECmailer*
  DECmailer is a return-to-factory replacement service for customers who maintain their equipment at the module or subassembly level. It provides 5-day turnaround, free return shipping, 90-day warranty on service and parts, 24-hour emergency service, monthly billing, and quarterly activity reports.

▪ Software Services

Software Services offers a wide range of comprehensive services to support Digital's system customers during any aspect of their system analysis, software development, or implementation efforts. These services start with the personal attention of a Digital software consultant and continue for as long as you own the system.

A Digital software specialist often works with a Digital sales representative to evaluate a prospective user's needs prior to purchase, in order to recommend solutions appropriate to the customer's requirements. A full range of services is available to assist customers throughout the planning, implementation, and production phases of their systems. The services are divided into three groups—Computer Services, Professional Services, and Network Management Services.

Computer Services

Digital's Computer Services provides automated information and software access to its customers. Three principal services are available to deliver the total business solution to your computer resource problem—Service Bureau, Disaster Backup, and Facility Management Services. These services are accessed locally and delivered remotely from Digital's Information Network Center. The network makes access only a local phone call away. Additionally, a telephone support service is available offering 24-hour-a-day, 365-day-a-year hotline staffed by Digital software experts.

- *Service Bureau*

  The Service Bureau provides four types of services—Enhanced Application Network Services, Hardware/Software Evaluation Service, Project Resource Services, and Incremental Computer Resources.

  - Enhanced Application Network Services combines several of Digital's resources with your systems and our nationwide network into an integrated application that is deliverable nationwide.
  - Hardware/Software Evaluation Service makes available to you our hardware and software for you to evaluate before you decide what to buy.
  - Project Resource Services provides computing resources delivered via nationwide network for major Software Services professional consulting projects.
  - Incremental Computer Resources provides computing resources during special needs like peak-load processing.

- *Disaster Backup Services*

  Backup and disaster recovery services allow customers to anticipate and plan for disruptions involving their computer facilities. Part of this plan is the processing of critical applications at computer facilities other than their own.

- *Facility Management Services*

  Facility Management Services is a long-term, customized, dedicated packaging of service offerings consisting of computing resources and operational staff available on the customer's site. Facility Management Services free you from the need to develop operations resources to support information systems.

Professional Services

Digital's Professional Services organization offers a full range of consulting services to help customers analyze, develop, implement, and productively use their Digital Equipment Corporation computer systems. These services benefit customers at all stages of a system life cycle—from planning and design to the development and delivery of solutions through a successful system startup and implementation. Our Professional Services consultants possess extensive practical experience in manufacturing, office automation, information systems, artificial intelligence, and networks.

In addition, Professional Services offers productivity services such as performance monitoring and capacity planning, and migration and conversion services. The services are divided in six groups—Planning and Design Services, Custom Applications Consulting and Projects, DECstart Consulting Services, Office Application Support Services, Performance and Capacity Planning, and Migration and Conversion Services.

- *Planning and Design Services*
  Planning and Design Services assist you in evaluating your needs by determining the best approach to estimate the structure, systems, environment, and cost factors to provide the optimal solution. Areas of concentration include long-range growth planning, networks, office systems, and specific applications.

- *Custom Applications Consulting and Projects*
  Professional Services provides solutions designed for specific applications by working with you so they understand and analyze your unique computing needs and applications. A large-scale project might result in an entire turnkey solution. A smaller scale project could mean the building of a new application or the expansion of an existing one.

- *DECstart Consulting Services*
  DECstart Consulting Services consists of several levels of fixed-price consulting services and automated system management tools that are designed to thoroughly prepare users and operators to effectively use and manage your systems. These services are available for all of Digital's major operating systems, ALL-IN-1, and network software.

- *Office Application Support Services*
  Office Application Support Services provides customized support and individualized onsite consulting for office staff. This includes orientation in the use of office products, support for the transition to an automated office, office procedures consulting, and training on customized applications installed on your system.

- *Performance and Capacity Planning*
  Performance and Capacity Planning helps you monitor your systems, evaluate performance, resolve problems, and make recommendations on how to optimize system utilization. Specific areas of focus are system performance monitoring and capacity planning, and network management control and DECnet monitoring.

- *Migration and Conversion Services*
  Migration Services include the RPG Migration Assistance Service. The service provides assistance in the organization, planning, and implementation of the conversion of RPG software from IBM computer systems to Digital's VAX computer systems. Conversion services enable customers to move from one operating system to another or from other vendor's software to Digital's software.

Network Management Services

Digital's network specialists provide an array of comprehensive network management services. These services include Network Planning and Design Service, Network Consulting Services, Management Services, Office Application Support Services, Project Services, Resident and Advisory Services, and Application Startup Services.

- *Network Planning and Design Service*
  The Network Planning and Design Service provides Digital's communications expertise to define or re-evaluate network requirements. The service develops network designs aimed at meeting the client's business and technical goals.

- *Network Consulting Services*
  Digital's software specialists assist clients at any stage of network planning, operation, implementation, or modification. Digital's consultants provide assistance from overall project management to advice on a specific problem.

- *Management Services*
  Management Services provides the methods to arrive at a solution for problems in four key areas of networking—Network Planning and Design, Office Analysis and Planning, Capacity Planning, and Artificial Intelligence. These four areas of change have been identified by leading experts in the field as critical for business success.

- *Office Application Support Services*
  These customized support services feature individualized, onsite consulting for office staff, office product usage orientation, office automation transition support, office procedures consulting, and user training on customer applications. Clients of this service benefit from an acceleration in productivity gains and improved acceptance of technological changes in the office environment.

- *Project Services*
  Project Services provide the expertise to solve business problems through customized solutions, applications expertise, project management, ongoing support, postimplementation support, and flexible solutions.

- *Resident and Advisory Services*
  If you've an adequate software staff and wish to maintain project management responsibility, the Resident and Advisory Services may be the solution to your problems. The Resident and Advisory consultants guide you to the effective use of Digital's hardware and software. The services feature consulting, short-term staffing, and assistance to personnel using new systems.

- *Application Startup Services*
  The Application Startup Services are directed toward application management and productivity.

▪ Educational Services

Educational Services offers a complete range of training programs and services to support PDP-11 system software and hardware.

Digital maintains worldwide training facilities. Services are centered around fully equipped regional education centers. These centers provide a staff of educators dedicated to providing quality education and training, and a variety of instructional formats that supports the learning pace of individuals as well as classrooms of individuals learning together.

Digital's Educational Services publishes a *Digest* every three months that includes a description and 6-month schedule of all software courses and a 9-month schedule of all hardware courses. It also includes ordering information for the self-paced instruction material. The *Digest* may be ordered by contacting your sales representative or your nearest Digital Training Center.

Courses are regularly scheduled classes offered at training centers. They cover the student range from first-time user to those needing highly specialized training on the theory of operation. Most catalog courses include extensive hands-on laboratory time, and all incorporate the use of a wide range of student workbooks, reference manuals, and other instructional materials.

Specialized training is available for users with unique applications or training situations. This training is designed to give the student the maximum relevant material for specific applications, while minimizing extraneous information. The customized courses are tailored to the individual customer's schedule and typically are presented in a series. The customized courses can be modified from existing courses or can be entirely new programs based on mutually agreed-upon objectives.

Customers with a group of individuals to train may find it more practical to have Educational Services conduct courses at the users' home sites. Onsite instruction of both catalog and customized courses eliminates travel and other expenses incurred by students attending classes at training centers. This method of instruction further enhances training by allowing Digital instructors to emphasize points of particular value to the students' applications and operations.

By taking advantage of the latest in text-based, computer-based, audiovisual-based, and IVIS-based techniques, Educational Services has developed a series of courses that offers self-paced instruction (SPI). These courses are convenient, self-contained, and modular. SPI format allows students to progress at their own rate, to study when and where they wish, and to reread modules for review. SPI course material is available in several forms—computer media (such as magtape and flexible disk), videotape, videocassette, audio/filmstrip cassette, and text—all supported by student guides and workbooks. Computer-based instruction (CBI) refers to courses that students take at their own pace at their terminal.

Training Centers

The Digital Training Centers located in the United States are listed as follows. For additional information on courses, training materials, or other training center locations, call the following customer support number: (617) 276-4373.

- *California*

Los Angeles Training Center
4311 Wilshire Boulevard, Suite 400
Los Angeles, California 90010-3779
Telephone: (213) 937-3870

Santa Clara Training Center
2525 Augustine Drive
Santa Clara, California 95051-7576
Telephone: (408) 748-4048

- *District of Columbia*

Washington, D.C. Training Center
8100 Corporate Drive
Landover, Maryland 20785-2231
Telephone: (301) 577-4300

- *Illinois*
  Chicago Training Center
  5600 Apollo Drive
  Rolling Meadows, Illinois 60008-4063
  Telephone: (312) 640-5521

- *Massachusetts*
  Bedford Training Center
  12 Crosby Drive, Building A
  Bedford, Massachusetts 01730-1493
  Telephone: (617) 276-4111

  Burlington Training Center
  30 North Avenue
  Burlington, Massachusetts 01803-3306
  Telephone: (617) 273-7400

- *Michigan*
  Detroit Training Center
  37735 Interchange Drive
  Farmington Hills, Michigan 48018-1270
  Telephone: (313) 471-6540

- *New Mexico*
  Los Alamos Training Center
  1900 Diamond Drive
  Los Alamos, New Mexico 87544
  Telephone: (502) 662-6905

- *New York*
  New York Training Center
  65 Broadway, Floor 12
  New York, New York 10119-0031
  Telephone: (212) 820-0508

- *Texas*
  Dallas Training Center
  12100 Ford Road, Suite 110
  Dallas, Texas 75234-7231
  Telephone: (214) 888-2500

▪ Documentation

A comprehensive set of documents supports PDP-11 systems. These manuals are periodically updated to include new developments and equipment and can be ordered from Digital's Publishing and Circulation Services.

The following list contains the titles and associated Digital order numbers of some important PDP-11 publications. To order any of these publications, write to:

Digital Equipment Corporation
Publishing and Circulation Services
10 Forbes Road
Northboro, Massachusetts 01532-2597

- KDJ11-B CPU Module User's Guide—EKKDJ1BUG
- KDF11-B CPU Module User's Guide—EKKDJ1ACG
- MicroPDP-11 System Technical Manual—EKMIC11TM
- MicroPDP-11 Owner's Manual—EKMIC11OM
- Microcomputer Products Handbook—EB-26078-41
- Terminals and Printers Handbook—EB-26291-56
- PDP-11 Architecture Handbook—EB-23657-18
- PDP-11 Software Handbook—EB-28783-41
- PDP-11 Software Source Book (Volumes 1 and 2)—EB-29102-41
- PDP-11 Systems and Options Catalog—ED-29632-41
- Networks and Communications Buyer's Guide—ED-26374-42

Glossary

This glossary provides definitions and explanations of common terms and abbreviations used in this handbook.

absolute-indexed mode—An indexed-addressing mode in which the vbase-operand specifier is addressed in absolute mode.

absolute mode—Autoincrement-deferred mode in which the program counter is used as the register and contains the address of the location containing the actual operand.

access mode—Any of the four processor access modes in which software executes. Processor access modes are, in order from most to least privileged and protected—kernel, supervisor, and user. When the processor is in kernel mode, the executing software has complete control of, and responsibility for, the system. In any other mode, the processor is inhibited from executing privileged instructions.

access type—The way in which the processor accesses instruction operands or a procedure accesses its arguments.

acoustics—Characteristics of ambient sound.

address—A number used by the operating system and user software to identify a storage location.

address-access type—The specified operand of an instruction is not directly accessed by the instruction. The address of the specified operand is the actual instruction operand. The context of the address calculation is given by the data type of the operand.

addressing mode—The way in which an operand is specified.

address space—The set of all possible addresses available to a process.

analog—The description of a numerical quanity by means of physical variables.

ANSI—American National Standards Institute.

application—A specific program or task to which a computer solution can be applied.

application program—A computer program designed to meet specific user needs (that is, a program that controls inventory or monitors a manufacturing process).

architecture—Computer architecture refers to the design or organization of the central processing unit (CPU).

argument pointer—It contains the address of the base of the argument list for procedures initiated using the call instructions.

ASCII (American Standard Code for Information Interchange)—A code that assigns a binary number to each alphanumeric character and several nonprinting characters used to control printers and communication devices. ASCII characters are seven or eight bits long and may have an additional parity bit for error detection.

asynchronous transmission—Transmission in which time intervals between transmitted characters may be of unequal length.

autodecrement mode—The contents of the selected register are decremented, and the result is used as the address of the actual operand of the instruction. The contents of the register are decremented according to the data type context of the register.

autodecrement-indexed mode—An indexed-addressing mode in which the base-operand specifier uses autodecrement-mode addressing.

autoincrement mode—The contents of the specified register are used as the address of the operand; then the contents of the register are incremented by the size of the operand.

autoincrement-deferred mode—An addressing mode in which the contents of the specified register is the address of a longword containing the address of the actual operand.

autoincrement-deferred indexed mode—The specified register contains the address of a longword that contains the address of the actual operand. The contents of the register are incremented by four which are the number of bytes in a longword. If the program counter is used as the register, this mode is called absolute mode.

autoincrement-indexed mode—An indexed-addressing mode in which the base-operand specifier uses autoincrement-mode addressing.

automatic-dialing unit—A device capable of automatically generating dialing digits.

automatic-calling unit—A dialing device supplied by the communications common carriers that permits a business machine to dial calls automatically over the communications networks.

backplane—The area of a computer where different logic and control elements are connected.

base-operand address—The address of the base of a table or array reference by indexed-mode addressing.

base-operand specifier—The register used to calculate the base-operand address of a table or array referenced by indexed-mode addressing.

base register—A general register that contains the address of the first entry in a list, table, array, or other data structure.

batch processing—The technique of executing a set of computer programs without human interaction or direction during its execution.

baud—A unit of data transmitting/receiving speed, approximately equal to a single bit per second.

bidirectional printing—A printing terminal technique to increase printing throughput by printing every other line from right to left, thus saving the carriage return time.

binary digit (bit)—In binary notation either of the characters 0 or 1. "Bit" is the commonly used abbreviation for binary digit.

BISYNC—IBM's 1968 Binary Synchronous Communications Protocol (BSC).

bit—Abbreviation for *binary digit.*

bit complement (also called one's complement)—The result of exchanging 0s and 1s in the binary representation of a number. The bit complement of the binary number 11011001 is 00100110. Bit complements are used in place of their corresponding binary numbers in some arithmetic computations in computers.

bit-map graphics—A technology that allows control of individual pixels on a display screen to produce graphic elements of superior resolution, permitting accurate reproduction of arcs, circles, sine waves, or other curved images that block-addressing technology cannot accurately display.

bit string—See *variable-length bit field.*

bit-transfer rate—The number of bits transferred per unit of time, usually expressed in bits per second.

block—A group of bits transmitted as a unit. A coding procedure is usually applied for synchronization or error-control purposes.

block mode—A feature that determines whether a terminal operates as an editing or interactive terminal. With block mode on, the terminal is a local editing terminal; with block mode off, the terminal is an interactive terminal.

branch access type—An instruction attribute that indicates that the processor does not reference an operand address, but that the operand is a branch displacement. The size of the branch displacement is given by the data type of the operand.

buffer—A place where data can be stored temporarily. Terminals can store data in a buffer if data is received faster than it can be processed or displayed.

bus—A group of parallel electrical connections that carry signals between computer components or devices.

byte—Eight contiguous bits starting at an addressable byte boundary.

cache memory—A small, high-speed memory placed between slower main memory and the processor. A cache increases effective memory-transfer rates and processor speed. It contains copies of data recently used by the processor and fetches several bytes of data from memory in anticipation that the processor will access the next sequential series of bytes.

call frame—See *stack frame*.

call instructions—The processor instructions CALLG (call procedure with general argument list).

call stack—The stack, and conventional stack structure, used during a procedure call. Each access mode of each process context has one call stack and interrupt-service context has one call stack.

carrier—A continuous frequency capable of being modulated or impressed with a signal.

CCITT (Commite Consultatif International de Telegraphie et Telephonie)—An international consultative committee that sets international communications usage standards.

character—A symbol represented by an ASCII code. A single, printable letter (a through z), numeral (0 through 9), or symbol (%) (.) ($) (,) used to represent data.

character printer—A printer, similar to a typewriter, that prints one character at a time.

character set—A set of graphics corresponding to the printable characters.

character string—A contiguous set of bytes. A character string is identified by two attributes—an address and a length. Its address is the address of the byte containing the first character of the string. Subsequent characters are stored in bytes of increasing addresses. The length is the number of characters in the string.

chassis—The structure that holds and supports electronic components and their circuitry.

command—An instruction, typed by the user at a terminal or included in a command file, that requests the software monitoring a terminal or reading a command file to perform some well-defined activity.

command procedure—A file containing commands and data that the command interpreter can accept in lieu of the user's typing the commands individually on a terminal.

communications link—The physical connection, typically a phone line, between a terminal and a computer or another peripheral device.

compatibility—The ability of an instruction, source language, or peripheral device to be used on more than one computer.

compatibility mode—A mode of execution that enables the central processor to execute nonprivileged PDP-11 instructions.

computer network—An interconnection of computer systems, terminals, and communications facilities.

concentrator—A communications device that provides communications capability between many low-speed, asynchronous channels and one or more high-speed, synchronous channels.

condition—An exception condition detected and declared by software.

condition codes—Four bits in the processor status word that indicate the results of previously executed instructions.

condition handler—A process that requests the system to execute when an exception condition occurs.

configuration—The arrangement of main components in a system.

console emulation—The use of hardware/software that enables a piece of equipment to be seen as a communications device between the operator and the computer.

console terminal—The terminal connected to the central processor used to control the computer system.

context—Also called process state. See *hardware context*.

context indexing—The ability to index through a data structure automatically because the size of the data type is known and is used to determine the offset factor.

context switching—Interrupting the activity in progress and switching to another activity. Context switching occurs as one process after another is scheduled for execution.

CPU (Central Processing Unit)—Commonly called a computer. A set of electronic components that control the transfer of data and perform arithmetic and logic calculations.

CRC (Cyclic Redundancy Check)—An error-detection scheme in which the check character is generated by taking the remainder after dividing all the serialized bits in a block of data by a predetermined binary number.

CRT terminal—Another name for a video terminal.

current-access mode—The processor access mode of the currently executing software. The current-mode field of the processor status longword (PSL) indicates the access mode of the currently executing software.

cursor—A distinctive mark on a video terminal screen, such as a flashing square or underline, that indicates where the next character will be displayed.

D_floating data—Eight contiguous bytes starting on an addressable-byte boundary, that are interpreted as containing a floating-point number.

daisywheel—A printhead that forms full characters rather than characters formed of dots. It is shaped like a wheel with many spokes, with a letter, numeral, or symbol at the end of each spoke.

database—A collection of interrelated data, organized for efficient inquiry and update.

database management—A systematic approach to the control of data.

data communications—The interchange of data messages from one point to another over communications channels.

data link layer—The higher of the two layers of the Ethernet specification that uses a medium-independent, link-level communicatin facility on top of the physical channel provided by the physical layers.

data type—The way in which bits are grouped and interpreted. In reference to the processor instructions, the data type of an operand identifies the size of the operand and the signficance of the bits in the operand.

DDCMP (Digital Data Communications Message Protocol)—A uniform discipline for the transmission of data between stations in a point-to-point or multipoint-data communications system. The method of physical-data transfer used may be parallel, serial synchronous, or serial asynchronous.

DECnet—Digital communications networks.

device driver—Software that controls all I/O between the system and its devices through a single controller unit.

device interrupt—An interrupt received on processor-priority levels. Device interrupts can be requested only by devices, controllers, and memories.

device name—The field in a file specification that identifies the device unit in which a file is stored.

device register—A location in device controller logic used to request device functions such as I/O transfers and/or to report status.

device unit—One drive, and its controlling logic, of a mass storage device system. A mass-storage system can have several drives connected to it.

diagnostic—A program that tests logic and reports any faults it detects.

digital—The representation of physical quantities by digits.

Digital Command Language (DCL)—The standard command interface to Digital's VAX/VMS operating system.

direct-mapping cache—A cache organization in which only one address comparison is needed to locate any data in the cache because any block of main-memory data can be placed in only one possible position in the cache.

directory—A list of files stored on a system.

disk controller—A device that directs the operation of disk storage devices.

disk drive—A mass storage device.

diskette—A flexible, flat, circular plate permanently housed in a paper envelope with magnetic coating that stores data and software.

displacement mode—The specifier extension is a byte, word, or longword displacement. The displacement is sign extended to 32 bits and added to a base address obtained from the specified register. The result is the address of the actual operand.

displacement-deferred mode—The specifier extension is a byte, word, or longword displacement. The displacement is sign-extended to 32 bits and added to a base address obtained from the specified registers. The result is the address of a longword that contains the address of the actual operand.

displacement-deferred indexed mode—An indexed-addressing mode in which the base-operand specifier uses displacement-deferred mode addressing.

displacement-indexed mode—An indexed-addressing mode in which the base-operand specifier uses displacement-mode addressing.

display—A screen presentation of requested information.

distributed-data processing—A computing approach in which an organization uses computers in more than one location, rather than one large computer in a single location.

DMA (Direct Memory Access)—A facility that permits I/O transfers directly into or out of memory without passing through the processor's general registers; performed either independently of the processor or on a cycle-stealing basis.

DNA (Digital Network Architecture)—A hardware and software scheme for interconnecting Digital's computers in a network. It is composed of three elements—Data Access Protocol (DAP), Network Services Protocol (NSP), and Digital Data Communications Message Protocol (DDCMP). See also *DDCMP*.

dot-matrix printing—A printing technique that forms characters from a two-dimensional array of dots.

double-floating data—See *D_floating data*.

draft-quality printer—A printer that produces characters that are readable, but of less than typewriter quality.

drive—The electromechanical unit of a mass-storage device system on which a recording medium (disk cartridge, disk pack, or magnetic-tape reel) is mounted.

EBCDIC (Extended Binary Coded Decimal Interchange Code)—An 8-bit character code used primarily in IBM equipment. The code provides for 256 different bit patterns.

editor—A program that interacts with the programmer to enter new programs into the computer and edit them as well as modify existing programs. Editors are language-independent and can edit anything in alphanumeric representation.

effective address—The address obtained after indirect- or direct-indexing modifications are calculated.

EIA (Electronic Industries Association)—A standards organization specializing in the electrical and functional characteristics of interface equipment.

enclosure—The physical encasement in which devices reside.

error—Any discrepancy between a computed, observed, or measured quantity and the true, specified, or theoretically correct value or condition.

Ethernet—Coaxial cable-based local communications networking medium for the interconnection of computers, information processing products, and office automation equipment.

event—A change in process status or an indication of the occurrence of some activity that concerns an individual process or cooperating processes. An incident reported to the scheduler that affects a process's ability to execute.

event flag—A bit in an event flag cluster that can be set or cleared to indicate the occurrence of the event associated with that flag. Event flags are used to synchronize activities in a process or among many processes.

exception—An event detected by the hardware (other than an interrupt or jump, branch, case, or call instruction) that changes the normal flow of instruction or set of instructions. There are three types of hardware exceptions—traps, faults, and aborts.

exception condition—A hardware- or software-detected event other than an interrupt or jump, branch, case, or call instruction that changes the normal flow of instruction execution.

exception enables—See *trap enables*.

exception vector—See *vector*.

executive mode—The second most privileged processor-access mode. The Record Management Services (RMS) and many of the operating system's programmed-service procedures execute in executive mode.

F_floating data—Four contiguous bytes starting on an addressable byte boundary. The bits are labeled from right to left 0 to 31. A two-word floating-point is identified by the address of the byte containing bit 0.

fanfold paper—A continuous sheet of paper whose pages are folded accordion-style and separated by perforations.

fault—A hardware-exception condition that occurs in the middle of an instruction and that leaves the registers and memory in a consistent state.

field—(1) See *variable-length bit field*. (2) A set of contiguous bytes in a logical record.

file—A collection of logically related records or data treated as a single item. A file is the means by which data is stored on a disk or diskette so it can be used at a later date.

file management—The processing of files and records within files by operating system procedures.

floating (point) data—See *F_floating data*.

floppy disk—See *diskette*.

font—A complete set of letters, numerals, and symbols of the same typestyle of a given typeface.

formfeed—A device that automatically advances a roll of fanfold paper to the top of the next page or form when the printer has finished printing the previous form.

frame pointer (FP)—FP contains the base address of the most recent call frame on the stack.

full-duplex—Describes a communications channel on which simultaneous two-way communications are available.

function key—A key that causes a computer to perform a function such as clearing the screen or executing a program.

G_floating data—Eight contiguous bytes starting on an arbitrary-byte boundary. The bits are labeled from the right 0 through 63. A G_floating data is specified by its address A, the address of the byte containing bit 0.

general register—Any of the registers used as the primary operands of the native-mode instructions.

graphics—The use of lines and figures to display data in contrast with the use of printed characters.

half-duplex—A communications channel on which only one-way communications are permitted at a time. The line can be "turned around" to allow data to flow the other way. Some half-duplex links provide a special "reverse channel" in the direction opposite to the flow of data that permits transmission of control signals only.

hardcopy—Hardcopy refers to paper printout, as opposed to video displays that cannot be saved.

hardware—The physical part or parts of a computer system.

hardware context—The values contained in the following registers while a process is executing—the program counter, the processor status longword, the general registers, the processor registers that describe the process virtual-address space, the stack pointer for the current-access mode in which the processor is executing, plus the contents to be loaded in the stack pointer for every access mode other than the current-access mode.

Hertz (Hz)—A unit of frequency equal to one cycle per second. Cycles are referred to as Hertz in honor of the physicist Heinrich Hertz.

high-level language—A computer programming language designed for applications and requiring a translation system such as an interpreter or compiler for execution.

host computer—A computer attached to a network that provides such services as computation, database access, or special programs or programming languages.

H_floating data—An extended range floating point number 16 bytes long having a range of + or-*.84*10**-4932 to + or-0.59*10**4932 and a precision of pproximately 33 decimal digits.

image—Procedures and data that have been bound together by the linker. There are three types of images—executable, shareable, and system.

immediate mode—Autoincrement-mode addressing in which the program counter is used as the register.

index mode—A data structure within a hierarchical index that provides additional speed of access.

indexed-addressing mode—Two registers are used to determine the actual instruction operand—an index register and a base-operand specifier. The contents of the index register are used as an index (offset) into a table or array. The base-operand specifier supplies the base address of the array (called the base-operand address or BOA).

index register—A register used to contain an address offset.

information management—The control of data available to a system.

initialation—The preparation, through the establishment of an empty file directory, of a new disk for use with a computer or the erasure of material on an existent disk.

I/O (Input/Output)—Pertaining to devices that accept data for transmission to a computer system (input) and that accept data from a computer system for transmission to a user or process (output).

input stream—The source of commands and data. One of either the user's terminal, the batch stream, or an indirect-command file.

instruction—A command that tells the computer what operation to perform next.

instruction buffer—An 8-byte buffer in the processor used to contain bytes of the instruction currently being decoded and to prefetch instructions in the instruction stream. The control logic continously fetches data from memory to keep the 8-byte buffer full.

instruction set—A unique set of instructin characters that specifies computer operation.

integer data—Data expression as whole positive or negative numbers.

integral modem—A modem built into a terminal rather than packaged separately.

integrated circuit (IC)—A complete electrical circuit on a single chip.

interface—An electronic assembly that connects an external device, such as a printer, to a computer.

interleaving—Assigning consecutive physical memory addresses alternately between two memory controllers.

ISO (International Standards Organization)—An organization for the establishment of international standards for network architectures.

Internet—A network in which Digital computers are connected to those of other manufacturers.

interrupt—An event other than an exception or branch, jump, case, or call instruction that changes the normal flow of instruction execution. Interrupts are generally external to the process executing when the interrupt occurs.

interrupt-service routine—The routine executed when a device interrupt occurs.

interrupt stack—The systemwide stack used when executing in interrupt-service context. At any time the processor is either in a process context executing in user, supervisor, executive, or kernel mode, or in systemwide interrupt-service context operation with kernel privileges, as indicated by the interrupt stack and current mode bits in the processor status longword. The interrupt stack is not context-switched.

interrupt-stack pointer—The stack pointer for the interrupt stack. Unlike the stack pointers for process-context stacks, the interrupt stack pointer is stored in an internal register.

interrupt vector—See *vector.*

kernel mode—The most privileged processor access mode. The operating system's most privileged services, such as I/O drivers and the pager, run in kernel mode.

laser printer—A printer that produces printing through laser technology.

letter-quality printer—A printer that produces printing comparable in quality to that achieved by a typewriter.

linefeed—The printer operation that advances the paper by one line.

literal mode—The instruction operand is a constant whose value is expressed in a 6-bit field of the instruction.

local—Hardwired connection of a computer to another computer, terminal, or peripheral device, such as in a local area network.

local area network—A privately owned data communications system with high speed channels optimized for information processing equipment over a limited geographic area.

longitudinal redundancy check (LRC)—An error-checking technique based on an accumulated exclusive-OR of transmitted characters.

longword—Four contiguous bytes starting on an addressable byte boundary. Bits are numbered from right to left 0 through 31. The address of the longword is the address of the byte containing bit 0.

main memory—See *physical memory*.

mass-storage device—A device capable of reading and writing data on mass-storage media such as a diskpack or a magnetic-tape reel.

memory management—The system functions that include the hardware's page mapping and protection and the operating system's image activator and pager.

mnemonic—A short, easy-to-remember name or abbreviation.

modem (Modulator/Demodulator)—A device that converts digital data from a terminal or CPU into analog signals for transmission over telephone lines and convert the receiver data back to digital format.

module—A separate unit or part of a computer program.

MOS (Metal-Oxide Semiconductor)—The most common form of LSI technology.

monitor—The master program that controls, checks, or verifies the operation of a computer system.

multiplexer—A device for connecting a number of communications lines to a computer.

multidrop network—A single communications line to which a number of nodes, terminals, or control units can be connected.

multiprogramming—A scheduling technique that allows more than one job to be in an executable state at any one time, so even with one CPU more than one program can appear to be running at a time because the CPU is giving small slices of its time to each executable program.

multitasking—Concurrent execution of different applications.

multiuser—A system capable of serving more than a single user at one given time.

native mode—The processor's primary execution mode.

network—A group of computers that are connected to each other by communications lines to share information and resources.

network application layer—A network protocol layer that provides generic services to the user layer.

network management layer—A network protocol layer that provides user control of and access to operational parameters and counters in lower layers.

node—An end point of any branch of a network, or a junction common to two or more branches of a network.

numeric string—A contiguous sequence of bytes representing up to 31 decimal digits (one per byte) and possibly a sign.

octaword—A set of 16 contiguous bytes starting at an arbitrary-byte boundary.

offset—A fixed displacement from the beginning of a data structure.

one's complement—See *bit complement*.

operation code—The pattern of bits within an instruction that specifies the operation to be performed.

operand specifier—The pattern of bits in an instruction that indicates the addressing mode, a register, and/or displacement, that taken together identify an instruction operand.

operand-specifier type—The access type and data type of an instruction operand(s).

operating system—A collection of computer programs that control the overall operation of a computer and perform such tasks as assigning places in memory to programs and data, processing interupts, scheduling jobs, and controlling the overall input/output of the system.

packed decimal—A method of representing a decimal number by storing a pair of decimal digits in one byte.

packed-decimal string—A contiguous sequence of up to 16 bytes interpreted as a string of nibbles. Each nibble represents a digit, except the low-order nibble of the highest-addressed byte, which represents the sign.

Packetnet—A product that allows Digital operating systems to participate in a packet switching environment.

packet switching—A data transmission process that utilizes addressed packets in which a channel is occupied only for the duration of transmission of the packet.

page—A set of 512 contiguous byte locations used as the unit of memory mapping and protection or the data between the beginning of file and a page marker, between two markers, or between a marker and the end of a file.

paging—The action of bringing pages of an executing process into physical memory when referenced. When a process executes, all of its pages are said to reside in virtual memory.

parity—A common technique for error detection in data transmission. Parity-check bits are added to the data so that each group of data bits include an even number of "ones" for even parity and an odd number for odd parity.

peripheral—A device that is external to the CPU and main memory (that is, printer, modem, or terminal), but connected to it by appropriate electrical connections.

physical address—The address used by hardware to identify a location in physical memory or on a directly addressable secondary-storage device such as a disk.

physical-address space—The set of all possible physical adresses that can be used to refer to locations in memory space or I/O space.

physical memory—The memory contained in the CPU memory modules.

pin assignment—The electrical connector of an active circuit component.

pixels—Definable locations on a display screen that are used to form images on the screen. For graphics displays, screens with a large number of pixels generally provide higher resolution.

point-to-point connection—A network configuration in which a connection is established between two terminal installations.

polling—A technique for determining the order in which nodes take turns accessing the network. This is done so that access collision can be avoided.

port—A location on the CPU where physical connection is made between the central computer and a terminal, printer, modem, another computer, or a communications line.

position-dependent code—Code that can execute properly only in the locations in virtual-address space that are assigned to it by the linker.

position-independent code—Code that can execute properly without modification wherever it is located in virtual-address space.

printhead—The element in a printer that forms a printed character.

printout—An informal expression referring to almost anything printed by a peripheral device; any computer-generated hardcopy.

printwheel—See *daisywheel.*

privileged instructions—Any instruction intended for use by the operating system or privileged-system programs.

procedure—A routine entered via a call instruction. See also *command procedure.*

process—The basic entity scheduled by the system software that provides the context in which an image executes. A process consists of an address space and both hardware and software contexts.

process address space—See *process space.*

process context—The hardware and software contexts of a process.

process priority level—The priority assigned to a process for scheduling purposes.

process space—The lowest-addressed half of virtual-address space, where process instructions and data reside. Process space is divided into a program region and a control region.

processor—The functional part of the computer system that reads, interprets, and executes instructions. See also *central processing unit.*

processor register—A part of the processor used by the operating system software to control the execution states of the computer system.

processor status longword (PSL)—A system-programmed processor register consisting of a word of privileged-processor status and the processor status word.

processor status word (PSW)—The low-order word of the processor status longword.

program—A complete sequence of instructions and routines needed to solve a problem or to execute directions in a computer.

program counter (PC)—At the beginning of an instruction's execution, the program counter normally contains the address of a location in memory from which the processor will fetch the next instruction it will execute.

program disk—A disk containing the instructions of a program.

programming language—The words, mnemonics, and/or symbols, along with the specific rules allowed in constructing computer programs. Some examples are BASIC, FORTRAN, and COBOL.

protection code—A code in each file that indicates who can access the file.

protocol—A formal set of conventions governing the format and relative timing of message exchange between two communicating processes.

PSTN (Private Switched Telephone Network)—Generic term for European telephone carriers.

quadword—Eight contiguous bytes (64 bits) starting at an addressable-byte boundary.

queue—(n.) A circular, doubly linked list. (v.) To make an entry in a list or table.

RAM (Random Access Memory)—Memory that can both be read and written into (that is, altered) during normal operation. RAM is the type of memory used in most computers to store the instructions of programs currently being run.

read-access type—An instruction- or procedure-operand attribute indicating that the specified operand is only read during instruction or procedure execution.

realtime—Refers to computer systems or programs that perform a computation during the actual time that a related physical process transpires.

ReGIS (Remote Graphics Instruction Set)—Digital's graphics command interface to terminals for putting shapes on the terminal screen.

register—A storage location in hardware logic other than main memory. See also *general register; processor register; device register.*

register-deferred indexed mode—An indexed-addressing mode in which the base-operand specifier uses register-deferred mode addressing.

register-deferred mode—In register-deferred mode addressing, the contents of the specified register are used as the address of the actual instruction operand.

register mode—In register-mode addressing, the contents of the specified register are used as the actual instruction operand.

remote—Communications between computer and terminals via switched lines such as telephone lines.

reverse video—A feature on a display unit that produces the opposite combination of characters and background from that which is usually employed, that is, white characters on a black screen, if having black characters on a white screen is normal.

ROM (Read-only Memory)—Memory containing fixed data or instructions that is permanently loaded during the manufacturing process.

routing layer—Modules in the routing layer used to route user data to its destination.

ruggedized—The construction of a device specifically for use in hazardous areas where special protection of computer equipment is needed.

scrolling—When a video terminal's screen is full, a new line of data can be displayed by adding it at the bottom of the screen and shifting all the previous lines upward, discarding the top line. When the upward movement is continuous rather than in line steps, it is called smooth scrolling.

serial transmission—A method of information transmission in which each bit of information is sent sequentially on a single path rather than simultaneously as in parallel transmission.

session control layer—A layer that defines the system-dependent aspects of logical link communications.

slave—A component of a system that operates under the control of another system component.

SNA (System Network Architecture)—A network architecture of IBM.

software—A set of computer programs, procedures, rules and associated documentation concerned with the operation of network computers (that is, compilers, monitors, editors, utility programs).

software interrupt—An interrupt generated on processor-priority levels that can be requested only by software.

stack—An area of memory set aside for temporary storage or for procedure- and interrupt-service linkages.

stack frame—A standard data structure built on the stack during a procedure call, starting from the location addressed by the frame pointer and going to lower addresses, and popped off during a return from procedure. Also called call frame.

stack pointer (SP)—A general register that contains the address of the top (lowest address) of the processor-defined stack. Reference to stack pointer will access one of the five possible stack pointers—kernel, executive, supervisor, user, or interrupt—depending on the value in the current mode and interrupt stack bits in the processor status longword.

standalone—A system designed to be used independently.

status code—A longword value that indicates the success or failure of a specific function.

storage—A place for keeping information.

supervisor mode—The third most privileged processor access mode. The operating system's command interpreter runs in supervisor mode.

symbolic debugger—A program that helps a programmer find errors in other programs.

synchronous—A technique in which data bits are sent at precisely timed intervals. Synchronous channels are capable of higher data rates than asynchronous ones, often running at 56,000 bits per second.

system-address space—See *system space*.

system space—The higher-addressed half of virtual-address space.

system-virtual address—A virtual address identifying a location mapped by an address in system space.

system-virtual space—See *system space*.

tape drive—A magnetic tape storage device.

terminal—The general name for those peripheral devices that have keyboards and video screens or printers.

timesharing—A mode of data processing that allows many terminal users to utilize a computer's resources to perform a variety of tasks simultaneously.

tool kit—The software and hardware components, including documentation, that are manufactured by Digital to help software developers create application programs that can be fully intergrated into computers.

tractorfeed—An attachment used to move paper through a printer. The roller that moves the paper has sprockets on each end that fit into the fanfold paper's matching pattern of holes.

translation buffer—An internal processor cache containing translations for recently used virtual addresses.

trap—An exception condition that occurs at the end of the instruction that caused the exception. The program counter saved on the stack is the address of the next instruction that would normally have been executed. All software can enable and disable some of the trap conditions with a single instruction.

trap enables—Bits in the processor status word that control the processor's action on certain arithmetic exceptions.

two's complement—A binary representation for integers in which a negative number is one greater than the bit complement of the positive number.

typeface—See *font*.

user layer—The layer that contains most of the user-supplied functions.

user mode—The privileges granted a user by the system manager.

variable-length bit field—A set of 0 to 32 contiguous bits located arbitrarily with respect to byte boundaries.

vector—An interrupt or exception vector is a storage location that contains the starting address of a procedure to be executed when a given interrupt or exception occurs. The system defines separate vectors for each interrupting device controller and for classes of exceptions.

video—A screen representation.

virtual address—A 16-bit or 32-bit integer identifying a byte location in virtual-address space. The memory-management hardware translates a virtual address to a physical address. The term virtual address may also refer to the address used to identify a virtual block on a mass-storage device.

virtual-address space—The set of all possible virtual addresses that an image executing in the context of a process can use to identify the location of an instruction of data.

virtual memory—The set of storage locations in physical memory and on disk that are referred to by virtual addresses.

virtual-page number—The virtual address of a page of virtual memory.

word—Two contiguous bytes (16 bits) starting at an addressable-byte boundary.

X.25—A protocol that implements the X.25 packet level and X.25 frame level of the CCITT X.25 recommendation for public data network interfaces.

Appendix A – PDP-11 Family Differences Table

The table that follows illustrates the issues involved in software migration between different members of the PDP-11 family. Each member of the family has some small difference in the way it executes instructions. Any program developed using PDP-11 operating systems with higher level languages will migrate from one system to another with very little difficulty. However, some applications written in assembly language may have to be modified slightly.

The table also details the slight differences between the various PDP-11 family processors and modules. Available hardware options and memories occasionally vary from processor to processor. Sometimes there is a small difference in the specific feature of one processor or another (e.g., a given feature may be standard on one module, optional on another, or not applicable). These variances are categorized under seven major headings—instructions or instruction sets, memory management expansion and relocation, interrupts, buses, general purpose registers, error handling, and consoles.

The VAX column refers to the PDP-11 Compatibility Mode available on VAX-11 processors.

Key to the table:

Y	—Yes
N	—No
▩	—Not applicable to that processor or module
Std	—Standard
Opt	—Optional
With "option"	—This feature is available only if the named option is available.

Instructions or Instruction sets

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
BASIC	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOB, SXT	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y	N	Y	N	Y	Y	Y	Y
RTT	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
MARK	Y	N	N	Y	Y	Y	Y	Y	N	N	Y	Y	N	Y	N	Y	Y	Y	N
XOR	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
ASH, ASHC, MUL, DIV	With KEV11 or KEV11-B	N	N	Y	Y	Y	Y	Y	N	N	Y	Y	With KE11-E	Y	Y	Y	Y	Y	Y
46 Floating-point instructions	N	N	N	With KEF11-A or FPF11	With KEF11-A or FPF11	Y	Y	Y	N	N	With KEF11-A or FPF11	With FP11-A	N	With FP11-A	With FP11-B or FP11-C	Y	With FP11-B or FP11-C	Y	N
MFPT	N	Y	Y	Y	Y	Y	Y	Y	N	N	Y	N	N	Y	N	N	N	Y	N
MTPS	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	Y	N
KE11-A, -B available for MUL, DIV, SHIFT	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
CIS	*	N	N	With KEF11-B	With KEF11-B	N	N	N	N	N	With KEF11-B	N	N	With KE44	N	N	N	N	N
MFPI, MFPD, MTPI, MTPD	N	N	N	Y	Y	Y	Y	Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
SPL	N	N	N	N	N	Y	Y	Y	N	N	N	N	N	Y	Y	N	Y	Y	N
CSM	N	N	N	N	N	Y	Y	Y	N	N	N	N	N	Y	N	N	N	Y	N

*Limited subset
(DIS) with KEV11-C

Instructions or Instruction sets

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
Illegal Instruction Actions																			
OP codes 000100-000107 JMP %n (jump to register)	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_4	T_4	T_4	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_4	T_4	T_{10}	T_4	T_4	T Native
OPcode 000000 (halt) in supervisor or user mode				T_{10}	T_{10}	T_4	T_4	T_4			T_{10}	T_{10}	T_{10}	T_4	T_4	T_{10}	T_4	T_4	T Native
In kernel mode	Halt	$T_{pup}+4$	$T_{pup}+4$	Halt or T_{10}	Halt or T_{10}	Halt or T_{10}	Halt or T_{10}	Halt or T_{10}	Halt	Halt	Halt or T_{10}	Halt	Halt	Halt	Halt	Halt	Halt	Halt or T_{10}	T Native
OPcodes 075040-75377 OPcodes 075400-75777		T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T Native
OPcode 076600	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	Med	T_{10}	T_{10}	T Native
OPcodes 170000-177777	User Microcode or T_{10}	T_{10}	T_{10}	FPP or T_{10}	FPP or T_{10}	FPP	FPP	FPP	T_{10}	T_{10}	FPP or T_{10}	FPP or T_{10}	T_{10}	FPP or T_{10}	FPP or T_{10}	FPP	FPP or T_{10}	FPP	T Native
OPcodes 210-227	Maint. Instruc.	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T_{10}	T Native
JMP,JRS																			
For JMP or JRS Mode 2, PC will receive	R	R	R	R	R	R	R	R	R	R+2	R	R	R	R	R	R	R	R	R

Instructions or Instruction sets

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
MOV(B),CMP(B),BIT(B) BIC(B),BIS(B),ADD,SUB:																			
If an instruction uses a register as the source operand, and the same register in autoincrement, autodecrement, auto-increment-deferred, or autodecrement-deferred mode (e.g., MOV R,(R) +), the destination will receive	R	R±2	R±2	R±2	R±2	R±2	R±2	R±2	R	R	R±2	R	R±2	R	R	R±2	R	R±2	R
If an instruction uses the PC as the source operand and indexed or indexed-deferred mode for the destination operand (e.g., MOV PC, HERE), the destination will receive	PC	PC+2	PC+2	PC+2	PC+2	PC+2	PC+2	PC+2	PC	PC	PC+2	PC	PC+2	PC	PC	PC+2	PC	PC+2	PC
SWAB:																			
V-bit action	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EIS Differences:																			
EIS exists	With KEV11	N	N	Y	Y	Y	Y	Y	N	N	Y	Y	With KE11-E	Y	Y	Y	Y	Y	Y
If R = Rv1 (in other words, the register is odd and a 16-bit result is being produced), are the condition codes based on the 16-bit or 32-bit result?				32	32	32	32	32			32	16		16	32		32	32	

Instructions or Instruction sets

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
FIS Differences (floating-instruction set):																			
FIS exists	With KEV11	N	N	N	N	N	N	N	N	N	N	N	With KE11-F	N	N	N	N	N	N
FMUL and FDIV require one word of RB stack	Y												N						
Is interruptible	Y												N						
Condition codes after FIS is interrupted are indeterminate	Y																		
MFPT Differences:																			
MFPT exists	N	Y	Y	Y	Y	Y	Y	Y	N	N	Y	N	N	Y	N	N	N	Y	N
MFPT result		4	4	3	3	5	5	5			3			1				5	
FPP Differences:																			
FPP microcode	N	N	N	With KEF11-A	With KEF11-A	Y	Y	Y	N	N	With KEF11-A	N	N	N	N	Y	N	Y	N
FPP hardware	N	N	N	With FPF11	With FPF11	N	N	With FPJ11	N	N	With FPF11	With FP11-A	N	With FP11-F	With FP11-B FA11-C	With FP11-E	With FP11-B FP11-C	Y	N
Hardware FPP is synchronous or asynchronous				Sync	Sync			Async			Sync	Sync		Sync	FP11-B Async FP11-C Sync	Async	FP11-B Async FP11-C Sync	Async	
FPP instructions can be interrupted				Y with KEF11-A N with FPF11-A	Y with KEF11-A N with FPF11-A	N	N	N			Y with KEF11-A N with FPF11-A	N		N	N	N	N	N	

Instructions or Instruction sets

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
Reset Instruction:																			
Nominal Duration of assertion of BUS INIT L	10μs + 90μs idle	8.4μs + 0.15μs idle	8.4μs + 0.15μs idle	10μs + 90μs idle	10μs + 90μs idle	15μs + 130μs idle	15μs + 130μs idle	12μs + 10μs idle	25 ms	150 ms	100 ms	100 ms	20ms + 70ms idle	100 ms	25 ms	10 ms	25 ms	12μs + 10μs idle	
Detection of power failure ends RESET instruction	N	N	N	N	N	N	N	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	
Minimum assertion of BUS INIT L (as shortened by power failure)									1μs	300ns		1μs		1μs	1μs		1μs		
Power failure during fetch of RESET instruction is fatal: no power down is executed	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	
MTPS/MFPS:																			
MTPS/MFPS exists	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	Y	PSL only affected in native mode
While in user or supervisor mode, MTPS changes PSW <3-0>	Y	Y	Y	Y	Y	Y	Y	Y			Y	Only if 777776 mapped						Y	
While in user or supervisor mode, MTPS changes PSW <7-4>	N	N	N	N	N	N	N	N			N	Only if 777776 mapped						N	
While in user or supervisor mode, MFPS accesses PSW <7-0>	Y	Y	Y	Y	Y	Y	Y	Y			Y	Only if 777776 mapped						Y	

Memory Management Expansion and Relocation

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
Number of physical Address bits	16	16	16	16, 18, or 22	22	22	22	22	16	16	18 22	18	16 18 with KT11-D	22	16 18 with KT11-C	18	22	22	Various
Maximum physical memory (assuming 8-Kbyte I/O Page)	56	56	56	56, 248, or 4088	4088	4088	4088	4088	56	56	248 or 3840	248	56 248 with KT11-D	3840	56 248 with KT11-C	248	3840	4088	Various
Kernel mode?	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Supervisor mode?	N	N	N	N	N	Y	Y	Y	N	N	N	N	N	Y	Y with KT11-C	N	Y	Y	N
User mode?	N	N	N	Y	Y	Y	Y	Y	N	N	Y	Y	Y with KT11-D	Y	Y with KT11-C	Y	Y	Y	N
Instruction space	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Separate data space	N	N	N	N	N	Y	Y	Y	N	N	N	N	N	Y	Y	N	Y	Y	N
Action on any access with PSW bits <15-14> (current mode field) set to:																			
00 Kernel mode				Kernel	Kernel	Kernel	Kernel	Kernel			Kernel	Kernel	Kernel	Kernel	Kernel	Kernel	Kernel	Kernel	
01 Supervisor mode				User	User	Super-visor	Super-visor	Super-visor			User	T250		Super-visor	Super-visor	Kernel	Super-visor	Super-visor	
10 Reserved				User	User	T250	T250	T250			User	T250		T250	T250		T250	T250	
11 User mode				User	User	User	User	User			User	User	User	User	User	User	User	User	
Which SP do MFPI, MFPD, MTDI, and MTPD use when PSW <13-12> = 10?				Unpredictable		User	User	User					Unpredictable					User	
PSW <15-12>, multiple SPS, and MTPI, MFPI, MTPD, and MFPD exist with or without MMU option				Y	Y	Std	Std	Std			Y	Std	N	Std	Y	Std	Std	Std	

Memory Management Expansion and Relocation

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
MMU Registers																			
MMR0 <08> (maintenance) implemented				N	N	N	N	N			N	Y	Y	Y	Y	Y	Y	N	
MMR0 <12> MMU trap flag implemented				N	N	N	N	N			N	N	N	N	Y	N	Y	N	
MMR1 exists but always reads 000000				Y	Y	N	N	N			Y	N	N	N	N	N	N	N	
MMR1 exists and operates				N	N	Y	Y	Y			N	N	N	Y	Y	N	Y	Y	
MMR2 tracks instruction fetches				Y	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	
MMR2 tracks interrupt vectors				N	N	N	N	N			N	N	N	N	Y	N	Y	N	
MMR3: Bits <00-02> (ENABLES DSPACE) exists and functions				N	N	Y	Y	Y			N	N	N	Y	Y	N	Y	Y	
Bit <03> (ENABLE) exists and functions				N	N	Y	Y	Y			N	N	N	Y	N	N	N	Y	
Bit <04> (ENABLE 22-bit) exists and functions					Y	Y	Y	Y			Y	N	N	Y	N	N	Y	Y	
Bit <05> (ENABLE UB map) exists and functions						Exists only					With KT24	N	N	Y	N	N	Y	Y	
PARs:																			
Width (bits)				16	16	16	16	16			16	12	12	16	12	12	16	16	
Program can execute from PAR				Y	Y	N	N	N			Y	12-bits only	12-bits only	Y	12-bits only	12-bits only	Y	N	
Bit <00> (trap any access) implemented				N	N	N	N	N			N	N	N	N	Y	N	Y	N	
Bit <07> (any access flag) implemented				N	N	N	N	N			N	N	N	N	Y	N	Y	N	
Bit <15> (bypass cache) implemented				N	N	Y	Y	Y			N	N	N	Y	N	N	N	Y	

Interrupts

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
Number of interrupt (BR) levels	1	4		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
Does expected interrupt occur if PSW <07-05> are lowered for only one instruction?	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	
Can an interrupt service routine itself be interrupted prior to executing its first instruction?	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
EIS	N			N	N	N	N	N			N	N	N	N	N	N	N	N	N
FIS	Y												N						
FPP				Y with KEF11-A	Y with KEF11-A	N	N	N			Y with KEF11-A	N		N	N	N	N	N	
CIS (DIS)	Y			Y	Y						Y			Y					
Exists	N	N	N	N	N	Y	Y	Y	N	N	N	N	N	Y	Y	N	Y	Y	N
Is cleared by reset						Y	Y	Y						N	Y		Y	Y	

Buses

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
Buses Available:																			
Q-bus	Y	N	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N
UNIBUS	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	1-4
Special Memory Bus						OBM					EUB			EUB	Digital Fast Bus		Cache Bus	PMI	Various
Bus Cycles Utilized:																			
CLR, SXT, do only DATO for last bus cycle (alternative: DATIP-DATO)	N	N	N	Y	N	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	Varies
MOV does only DATO for last bus cycle	Y	DATI, DATO	DATI, DATO	Y	Y	Y	Y	Y	DATIP, DATO	DATIP, DATO	Y	Y	Y	Y	Y	Y	Y	Y	Varies
EIS does DATI to fetch source operand	DATIP, DATO	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Varies
UNIBUS/Q-bus timeout value	10μs		10μs	10μs	10μs	10μs	10μs	10μs	22μs	22μs	25μs	25μs	15μs	25μs	7μs	35μs	7μs	15μs	Various
NPRs (DMA) will be granted during CPU instructions	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Console SLU accessible from Q-bus/UNIBUS	Y		N	Y	N	N	Y	N	Y	N	N	Y	Y	N	Y	Y	Y	N	N
Line clock register accessible from a Q-bus/UNIBUS	*		N	Y	N	N	Opt	N	Y	N	N	Y	Y	N	Y	Y	Y	N	N
Bootstrap ROMs accessible from Q-bus/	Y		N	*	N	N	*	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	1	N

* If LTC/Bootstrap exists at all.
1 M9312 Emulation – Yes.
Internal ROMs – No.

Processor Status Word

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
Bits <15-12> mechanized (Current mode, Previous mode)	N	N	N	Y	Y	Y	Y	Y	N	N	Y	Y	With KT11-D	Y	Y	Y	Y	Y	
Bit <11> mechanized (Register set selection)	N	N	N	N	N	Y	Y	Y	N	N	N	N	N	N	Y	N	Y	Y	
Bit <08> mechanized (CIS instruction suspended)	N	N	N	Y	Y	Y	Y	Y	N	N	Y	N	N	Y	N	N	N	Y	
Bit <07> mechanized (High-order priority bit)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	

Processor Status Word

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
Bits <06-05> mechanized (Low-order priority bits)	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
Bit <04> mechanized (T)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
Bits <03-00> mechanized (N, Z, V, and C)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Processor Status Word can be accessed via Reads/Writes to location 17 777 776	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N
MTPS/MFPS instructions (Bits <07-00> or <03-00>	Y	N	N	Y	Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	Y	N
SPL instruction (Bits <07-05>	N	N	N	N	N	Y	Y	Y	N	N	N	N	N	Y	Y	N	Y	Y	N
Condition code instructions (bits <03-99>	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
T-bit Differences:																			
Can explicit PSW reference (by program or console) Set/Clear T-bit?	N	N	N	N	N	N	N	N	Y	Y	N	N	N	N	N	N	N	N	N
Number of instructions executed between RTI setting T-bit and T-bit TRAP:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Number of instructions executed between RTT setting T-bit and T-bit trap	1	1	1	1	1	1	1	1	1	No RTT Instruction	1	1	1	1	1	1	1	1	1
T-bit TRAP immediately ends WAIT instruction	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N	Y	
T-bit TRAPS have higher priority than interrupts (are taken prior to interrupts)				Y	Y	Y	Y	Y		Y	Y	Y	Y		N		N	Y	

General Purpose Registers

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
Number of sets of R0-R5	1	1	1	1	1	2	2	2	1	1	1	1	1	1	2	1	2	2	1
Number of stack pointers	1	1	1	3 (2 useful)		3	3	3	1	1	(2 useful)	2	2	3	3	2	3	3	1
Can program code be executed from GPRs?	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N
Can GPRs be accessed by program as 17777700-17777717?	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N
Can GPRs be accessed by console as 17777700-17777717?	N	N	N	N	N	N	N	N	Y	Y	N	Y	Y	Y	Y	Y	Y	N	N

Error Handling

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
Odd Address/Bus Errors																			
Odd address errors detected by CPU	N	N	N	N	N	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y
If odd address error occurs while autoincrementing/ autodecrementing a register, register will have been modified.						Y	Y	Y	N	N		N	Y	N	Y		Y	Y	
Bus error (timeouts) detected by CPU	Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
If bus error occurs while using autoincremented/ autodecremented-mode addressing, register will have been modified.	Y			Y	Y	Y	Y	Y	N	Y	Y	N	Y	N	Y	Y	Y	Y	
If bus error occurs while reading I-stream using PC, PC will have been incremented.	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	N	

Error Handling

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
Errors Using (SP):																			
Odd address/Bus error while using (SP)	HALT	No error detected		SP ← 4 T4[1]	SP ← 4 T4[1]	SP ← 4 T4	SP ← 4 T4	SP ← 4 T4	HALT	HALT	SP ← 4 T4[1]	HALT	SP ← 4 T4	HALT	SP ← 4 T4	SP ← 4 T4	SP ← 4 T4	SP ← 4 T4	
Stack Overflow Errors:																			
Yellow-zone trap implemented?	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N
Yellow-zone trap programmable (else fixed at 000400)				N	N	N	N	N	N	N	N	N	With KJ11	N	Y	Y	Y	N	
Yellow-zone action				Execute then T4	Execute then T4	Execute then T4	Execute then T4	Execute then T4	T4	T4	Execute then T4	T4	Execute then T4	Execute then T4	Execute then T4	Execute then T4	Execute then T4	Execute then T4	
Separate red-zone trap implemented?				N	N	N	N	N	N	N	N	N	Y	N	Y	Y	Y	N	
Red-zone action													T4		SP ← 4 T4	T4	SP ← 4 T4		
Vector Errors:																			
Error while fetching error, vector hangs processor.	Y	No error not detected		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
Halt switch processor.	N			N	N	Y	Y	Y	N	N	N	N	N	Y	N	N	N	Y	
Console initialize able to halt processor.									Y	Y		Y	Y	Y	Y		Y		
Console RESTART switch able to restart processor.	Y			Y	Y	Y	Y	Y	Y		Y	Y		Y				Y	

[1]Odd SP not detected.

Consoles and Serial Ports

Processor	03	T11	21	23		53	73/83		04	05/10	24	34	35/40	44	45/50/55	60	70	84	VAX
Module	**Various**	**DCT11**	**KXT11-A**	**KDF11-A**	**KDF11-B**	**KDJ11-D**	**KDJ11-A**	**KDJ11-B**	**KD11-D**	**KD11-B**	**KDF11-UA**	**KD11-E**	**KD11-A**	**KD11-Z**	**KD11-A KD11-D**	**KD11-K**	**KB11-B KB11-C**	**KDJ11-B**	**Various**
Programmer's Console:																			
Lights and switches	N			N	N			N	With KY11-LB	With KY11-J	N	With KY11-LB	Y	N	Y	Y	Y	N	
ASCII console	Micro ODT	N	Y With Boot ROM	Micro ODT	Micro ODT	Micro ODT	Micro ODT	Micro ODT	With M9312	N	Micro ODT	With M9312	N	MSD	N	N	With F.S. Contract	Micro ODT	
ASCII console memory addressing range in bits	16		16	18	18	22	22	22	18		18	18		22			22	22	
ASCII console could access GPRs	Y		Y	Y	Y	Y	Y	Y	N		Y	N		Y			Y	Y	
Operator's Console:																			
Exists	Std			Std	Std	Std		Std	With KY11-LA	Opt	Std	With KY11-LA	Opt	Std	Opt	N	Opt	Std	
Remote Diagnosis:																			
Available from Field Service	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	Y	N	Y
Serial Ports:																			
Number on CPU module	0	0	2	0	2	0	1	0	1	0	2	0	0	2	0	0	0	2	

Appendix B ▪ Console ODT Command Languages

▪ Console ODT Command Set for MicroPDP-11/53, MicroPDP-11/73, MicroPDP-11/83, and PDP-11/84

A description of the PDP-11/84 console terminal and a table of the console ODT commands are found in Chapter 6 (PDP-11/84).

To describe the use of a command, other commands are mentioned before they have been defined. For the novice user, the following paragraphs should be skimmed first for familiarization and then reread for detail. The word "location" refers to a bus address, processor register, or the processor status word (PSW).

Whenever these systems are halted, the console terminal can be used as the hardware console of the machine. The contents of the main memory, the general registers, or I/O device registers can be examined or modified and the processor can be initialized or started.

The PDP-11 processor can be halted by:

- Pressing the <BREAK> key on the terminal (if the BREAK HALT feature is enabled on the CPU module).
- Pressing the HALT button, located on the front panel.
- Executing the PDP-11 HALT instruction.
- Causing certain PDP-11 internal errors.

Once the PDP-11 processor is halted, you are prompted for input. The console prompt character (@) indicates that the console commands described below are available for use.

NOTE

In the following examples, the user's entry is printed in **boldface** (dark) type.

/(ASCII 057) Slash

This command (/) is used to open a bus address, processor register, or the processor status word and is normally preceded by other characters that specify a location. In response to /, console ODT prints the contents of the location (six characters) and then a space (ASCII 40). After printing is complete, console ODT waits either for new data for that location or for a valid close command. The space character is issued so that the location's contents and possible new contents entered by the user are legible on the terminal.

Example:	@**001000**/012525<SPACE>
@	= console ODT prompt character.
001000	= octal location in the bus address space desired by the user (leading zeros are not required)
/	= command to open and print contents of location
012525	= contents of octal location 1000
<SPACE>	= space character generated by console ODT

The / command can be used without a location specifier to verify the data just entered into a previously opened location. The / is recog-nized only if it is entered immediately after a prompt character. A / issued immediately after the processor enters ODT mode causes a ?<CR><LF> to be printed because a location has not yet been opened.

Example:	@**1000**/012525<SPACE>1234<**CR**><CR><LF> @/001234<SPACE>
first line =	new data of 1234 entered into location 1000 and location closed with <CR>
second line =	a / was entered without a location specifier and the previous location was opened to reveal that the new contents were correctly entered into memory

<CR>(ASCII 015) Carriage Return

This command <CR> is used to close an open location. If a location's contents are to be changed, the user should precede the <CR> with the new data. If no change is desired, <CR> closes the location without altering its contents.

Example:	@**R1**/004321<SPACE> <**CR**><CR><LF> @

Processor register R1 was opened and no change was desired so the user issued <CR>. In response to the <CR>, console ODT printed <CR><LF>@.

Example:	@**R1**/004321<SPACE> **1234**<**CR**><CR><LF> @

In this case the user desired to change R1, so new data, 1234, were entered before issuing the <CR>. Console ODT deposited the new data in the open location and then printed <CR><LF>@.

Console ODT echoes the <CR> entered by the user and then prints an additional <CR>, followed by a <LF>, and @.

<LF> (ASCII 012) Line Feed

This command <LF> is used to close an open location and then open the next contiguous location. Bus addresses and processor registers are incremented by 2 and 1 respectively. If the PSW is open when a <LF> is issued, the PSW is closed and a <CR><LF>@ is printed; no new location is opened. If the open location's contents are to be changed, the new data should precede the <LF>. If no data is entered, the location is closed without being altered.

Example:	@**R2**/123456<SPACE> <**LF**><CR><LF> R3/054321<SPACE>

In this case, the user entered <LF> with no data preceding it. In response, console ODT closed R2 and then opened R3. When a user has the last register, R7, open and issues <LF>, console ODT opens the beginning register, R0. When the user has the last bus address of a 32-Kword segment open and issues <LF>, console ODT opens the first location of that same segment. The user who wishes to cross the 32-Kword boundary must reenter the address for the desired 32-Kword segment (i.e., console ODT is module 32 Kword). This operation is the same as that found on older PDP-11 consoles

Example:	@**R7**/000000<SPACE> <**LF**><CR><LF> R0/123456<SPACE> or @**577776**/000001<SPACE> <**LF**><CR><LF> 400000/125252<SPACE>

Unlike other commands, console ODT does not echo the <LF>. Instead it prints <CR>, then <LF> so the printing terminals operate properly. In order to make this easier to decode, console ODT does not echo ASCII 0, 2, or 10 either, but responds to these three characters with ?<CR><LF>@.

$ (ASCII 044) or R (ASCII 122) Internal Register Designator
Either character ($ or R) when followed by a register number, 0 to 7, or the PSW designator, S, will open that specific processor register.

The $ character is recognized to be compatible with ODT-11. The R character was introduced because it requires only one keystroke and because it is representative of what it does.

Example:	@**$0**/000123<SPACE> or @**R7**/000123<SPACE>**<LF>** R0/054321<SPACE>

If more than one character is typed (digit or S) after the R or $, console ODT uses the last character as the register designator. There is an exception, however, if the last three digits equal 077 or 477. ODT interprets this to mean the PSW rather than R7.

S (ASCII 123) Processor Status Word (PSW)
This designator is for opening the PSW and must be employed after the user has entered an R or $ register designator.

Example:	@**RS**/100377<SPACE>**0<CR>**<CR><LF> @/000020<SPACE>

Note the trace bit (bit 4) of the PSW cannot be modified by the user. This is done so that PDP-11 program debug utilities (e.g., ODT-11) that use the T bit for single-stepping are not accidentally harmed by the user.

If the user issues a <LF> while the PSW is open, the PSW is closed and ODT prints a <CR><LF>@. No new location is opened in this case.

G (ASCII 107) Go
This command (G) is used to start program execution at a location entered immediately before the G. This function is equivalent to the LOAD ADDRESS and START switch sequence on older PDP-11 consoles.

Example:	@**200G**<NULL><NULL>

The console ODT sequence for a G, after echoing the command character follows.

1. Print two nulls (ASCII 0) so that the bus initialization that follows does not flush the G character from the double-buffered UART chip in the serial line interface.
2. Load R7 (PC) with the entered data. If no data is entered, 0 is used. (In the example above, R7 is set equal to 200 and that is where program execution begins.)

3. The PSW and floating-point status register contained in the MMU are cleared to 0.
4. The bus is initialized.
5. The service state is entered by the processor. If there is anything to be serviced, it is processed. If the HALT signal is asserted, the processor reenters the console ODT state. This feature is used to initialize a system without starting a program (R7 is altered). If the user wants to single-step a program, it can be executed by issuing a G and then successive P commands, all done with the HALT signal asserted via the HALT switch.

P (ASCII 120) Proceed

This command (P) is used to resume execution of a program and corresponds to the CONTINUE switch on older PDP-11 consoles. No programmer-visible machine state is altered using this command.

Example: **@P**

The PDP-11 processor is started immediately after the transmission of the P to the terminal console has begun. If a RESET instruction is executed while the P is transmitting, the echo of the P may be lost.

Program execution resumes at the address pointed to by R7. After the P is echoed, the console ODT state is left and the processor immediately fetches the next instruction. If the HALT signal is asserted, it is recognized at the end of the instruction (during the service state) and the processor enters the console ODT state. Upon entry, the content of the PC (R7) is printed. In this fashion, a user can single-instruction step through a program and get a PC trace displayed on the terminal.

Control-S (ASCII 023) Binary Dump

This command (Control-S) is used for manufacturing test purposes and is not a normal user command. It is described here to explain the machine's response if the command is accidentally invoked. Control-S is intended to more efficiently display a portion of memory than the / and <LF> commands can display it. The protocol follows.

1. After a prompt character, console ODT receives a control-S command and echoes it.
2. The host system at the other end of the serial line must send two 8-bit bytes which console ODT interprets as a 16-bit starting address. These two bytes are not echoed. The first byte specifies starting address <15:8> and the second byte specifies starting address <7:0>. Bus address bits <17:16> are always forced to be 0; the dump command is restricted to the first 32 Kwords of address space.

3. After the second address byte has been received, console ODT outputs 10 bytes to the serial line starting at the address previously specified. When the output is finished, console ODT prints <CR><LF>@.

If a user accidentally enters this command, it is recommended, in order to exit from the command, that the user resets the terminal and enters two @ characters (ASCII 100) as a starting address. After the binary dump, an @ prompt character is printed.

Appendix C ▪ Instruction Timing

It is often useful to know the amount of time required to execute a particular instruction or series of instructions. The calculation of this time is straightforward but dependent on a variety of factors. These factors include two main categories—speed of the hardware and complexity of the instruction.

This appendix examines the hardware features that affect speed and presents tables that break down the instructions to allow you to calculate the execution time for any instruction as executed by the various hardware systems.

▪ Speed of the Hardware

A computer system is built of many separate hardware subsystems. Each of these can affect the rate at which instructions are processed. Some of the most prominent factors are:

- The processor microcycle rate.
- The amount of work performed with each microcycle.
- The main memory access time.
- The number of main memory cycles that can be avoided entirely.
- Whether there is DMA activity on the system bus.
- Timing variations peculiar to a given.

Processor Microcycle Rate

If the time required for a processor microcycle is shortened, the amount of time required to execute a complete macroinstruction should also be shortened, assuming nothing else stalls the processor. For example, the basic clock time of the PDP-11/53 and 73 microcycle is 267 nanoseconds; the basic clock time of the PDP-11/83 and 84 is 222 nanoseconds. If all other things were equal, the PDP-11/83 and 84 would be faster for this reason alone.

The Amount of Work Performed with Each Microcycle

A processor that can accomplish more useful work per microcycle will also execute macroinstructions faster than a processor that accomplishes less per

microcycle. For example, these processors' microcycle can accomplish more work per microcycle than previous PDP-11 processors.

Main Memory Access Time
While the processor is waiting for data from memory, it is not accomplishing useful work. Faster memories mean less waiting time.

Minimizing the Number of Main Memory Cycles
The inclusion of fast buffer memories between the processor and main memory can mean that fewer references need to go all the way to main memory for data. This speeds execution while freeing up main memory cycles for use by DMA devices. The PDP-11/73, 83, and 84 use two methods to minimize the number of main memory cycles—cache memory and the instruction buffer. The PDP-11/53 uses a dedicated memory bus and the instruction buffer.

PDP-11/73, 83, and 84 Cache Memory
The PDP-11/73, 83, and 84 contain a buffer memory between the processor and the main memory. This buffer memory attempts to store the data and instruction words most frequently required by the processor. This buffer is referred to as the *cache memory*. Any processor data request satisfied by the cache memory takes much less time than a request that can be satisfied only from main memory. When the requested data is found in the cache, the operation is referred to as a *cache hit*. When the requested data is not found in the cache, the operation is referred to as a *cache miss*, and the request automatically passes to main memory.

In addition, every time these processors must go to main memory, two words will be returned to the cache. This is useful because there is a high probability that the second word will be used soon.

The PDP-11/53 does not contain a cache memory. All memory requests result in a read or write of main memory, however, main memory is located on-board, on its own bus for fast access.

The Instruction Buffer
In addition to the speed improvement provided by its cache, the PDP-11/84 attempts to predict the address of the next instruction word required by the processor. Because the processor is normally accessing instruction words in ascending order (as directed by the Program Counter), this prediction is generally successful. Separate hardware within the processor performs this prediction and uses spare memory or cache cycles to access the next instruction word. This next instruction word is then stored in the *instruction buffer*. Should the prediction prove true, the word is already waiting and the processor need not stop to fetch that next word. If, however, the program

branches, the prediction will prove false and the processor will need to stop while the real next instruction word is fetched from memory.

Any read access using the PC is referred to as an Instruction-stream (I-stream) read. Read accesses that do not use the PC cannot be predicted in this fashion, and are referred to as data reads.

Effective Memory Access Time

Table C-1 indicates the time to access data from each of the different data sources for each of the different processors.

Table C-1 ▪ Effective Memory Access Time

Processor		I-stream Reads	Data Reads	Data Writes
PDP-11/84—data in:				
	Instruction buffer	0 ns		
	Cache memory	222 ns	222 ns	
	Main memory	1000 ns	1000 ns	680 ns
PDP-11/83—data in:				
	Instruction buffer	0 ns		
	Cache memory	222 ns	222 ns	
	Main memory	1000 ns	1000 ns	700 ns
PDP-11/73—data in:				
	Instruction buffer	0 ns		
	Cache memory	267 ns	267 ns	
	Main memory	1000 ns	1000 ns	1000 ns
PDP-11/53—data in:				
	Instruction buffer	0 ns		
	Main memory	542 ns	542 ns	542 ns

Direct Memory Access (DMA)

If the processor must read or write a main memory word, and the bus is already busy servicing a direct memory access (DMA), then the processor must wait until the DMA completes. This can greatly extend the processor's effective memory access time.

In most circumstances, the Dual Tag Store of the PDP-11/73, 83, and 84 allows the processor to continue to read from the cache memory unimpeded by DMA activity. The processor must wait only if a cache miss or a main memory write occurs or the DMA attempts to alter data already stored in the processor's cache memory.

Timing Variations in a Given System

Finally, the speed of the various hardware systems varies from unit to unit. The clock of all these processors is timed by a quartz crystal and is very accurate, but the memory system is not so precise.

▪ Instruction Complexity

PDP-11 processors fetch instructions from main memory. These instructions can change the state of the CPU, or manipulate data stored in main memory. The instructions vary in complexity and, therefore, in execution speed. The execution of some PDP-11 instructions is examined below.

Increment (INC)

This instruction causes the operand to be replaced by the operand plus one, that is, the operand is incremented by one. The operand may be a CPU general register or a byte or word of main memory.

The first step in the execution of any instruction is to fetch the opcode from main memory. Contained within the opcode is the field specifying the mode to be used in addressing the operand. Also specified is the general register that will combine with the addressing mode to tell the CPU where the operand will be found.

If the operand is contained within a general register (address mode 0), then the register can be quickly incremented, and the INC instruction is complete.

If any other addressing mode was selected, the operand is contained in main memory. First, the address in memory of the operand must be determined. This requires zero, one, or two reads of main memory, depending on the particular addressing mode. Then the operand can be read from that address. The data is incremented within the CPU and the updated data written back into main memory. The INC instruction always returns the incremented data to the same location, so the address calculation need not be repeated.

Move (MOV)

The MOV instruction is used to copy data from one location to another. The location that supplies the data is called the *source*. The location to receive the data is called the *destination*.

As in all instructions, the opcode is first fetched from memory. This time, the opcode contains two addressing fields. One specifies the addressing mode and register for the source operand, and the other specifies the addressing mode and register for the destination operand.

The exact order of operations varies from one member of the PDP-11 family to another but, in general, the following operations take place: the source field is evaluated to supply the address of the source operand (that is, where the MOV instruction will find the data). Remember that this requires zero, one, or two memory reads, depending on the addressing mode. The destination field is then evaluated to determine the address of the destination operand (i.e., where the MOV instruction will put the copy of the data). The source operand itself is then read, and the destination operand written. If the source operand is in main memory, this requires reading main memory. If the destination operand is in main memory, this requires writing main memory.

Add

This is another two-operand instruction. It differs from the MOV instruction only in that the destination operand is replaced with the sum of the source and destination operands. In other words, both the source and destination operands must be read, then the two data words added together, and the result rewritten to the destination operand. Thus, if the destination operand is in main memory, ADD requires one more read of main memory than an equivalent MOV instruction.

Emulator Trap (EMT)

This instruction does not require the specification of any addressing modes. This is because all of the operands are implicitly specified by the opcode itself.

The instruction allows the software to perform a trap (similar to a hardware interrupt). This instruction is generally used to call monitor or kernel routines.

As always, the opcode is first fetched from memory. Then the old program counter (PC) and processor status (PS) are pushed onto the SP stack. The PC and the PS are then loaded from the EMT trap vector (at location 000 030). (This is a simplified explanation, not taking into account memory management.)

Beyond the opcode fetch, EMT therefore requires two main memory writes as well as two main memory reads to execute completely.

General Method to Calculate Instruction Timing

In general, all instructions executed by the PDP-11 require an opcode fetch, and most require that some data be manipulated. The data manipulation consists of one or more of source operand access, destination operand access, destination operand write-back, and miscellaneous implied reads and writes.

The total execution time of an instruction can be found by summing up the time required for each of the basic operations listed above. The tables that follow provide the specific timings for each of these basic operations.

The three-operand CIS instructions are different and are not covered in this appendix.

Overlap of Phases

Certain simple addressing modes allow much work to be accomplished in a single microcycle. For example, most PDP-11 processors can execute the instruction ADD R1, R0 in a single microcycle. In that single microcycle, R1 and R0 are presented to the adder and the result returned to R0. Thus, the source, destination, execution, and write-back all take place concurrently.

Addressing Modes

The various addressing modes require various amounts of work. Mode 0 (operand in general register) requires essentially no work and is the fastest addressing mode. Mode 7 (indexed, deferred) is the most complex addressing mode, requiring the most work and execution time. Use of the various addressing modes is described in the *PDP-11 Architecture Handbook*. The tables below document the number of I-stream reads, data reads, and data writes performed by each of the source and destination addressing modes as well as their contribution (in microseconds) to the overall instruction execution time. Separate tables document:

- Source addressing modes.
- Destination modes for CMP(B), TST(B).
- Destination modes for JMP.
- Destination modes for JSR.
- Destination modes for MOV(B) and CLR(B).
- Destination modes for all other instructions.

Source Addressing Modes

The source operand is never written, so a source operand access simply consists of read operations.

Source Mode	Macro-11 syntax	I-stream reads	Data reads	Data writes	CPU activities required	11/84/83	73/53
00-07	R	0	0	0	<None>	0	0
10-17	@R or (R)	0	1	0	Access operand	.44	.53
20-26	(R) +	0	1	0	Access operand Increment R	.44	.53
27	(PC) + or #n	1	0	0	Access operand Increment PC	.22	.26
30-36	@(R) +	0	2	0	Access addr of operand Access operand Increment R	.89	1.07
37	@(PC) + or @#n	1	1	0	Access addr of operand Access operand Increment PC	.67	.80
40-46	-(R)	0	1	0	Decrement R Access operand	.67	.80
47	-(PC)	0	1	0	Decrement PC Access operand	1.33	1.60
50-56	@-(R)	0	2	0	Decrement R Access addr of operand Access operand	1.11	1.33
57	@-(PC)	0	2	0	Decrement PC Access addr of operand Access operand	1.78	2.14
60-67	INDEX(R)	1	1	0	Access index word Sum with R Access operand	.89	1.07
70-77	@INDEX(R)	1	2	0	Access index word Sum with R Access addr of operand Access operand	1.33	1.60

*0 if destination mode also equals 00-07.

Destination Addressing Modes for BIT(B), CMP(B) and TST(B) (read-only)

These instructions never modify the destination operand. Therefore most PDP-11s never write the destination operand. Other older models write back the exact same data as read. See the PDP-11 difference list (Appendix A) for details.

Destination Mode	Macro-11 syntax	I-stream reads	Data reads	Data writes	CPU activities required	11/84/83 Single Op	73/53
00-07	R	0	0	0	<None>	0	0
10-17	@R or (R)	0	1	0	Access operand	.44	.53
20-26	(R) +	0	1	0	Access operand Increment R	.44	.53
27	(PC) + or #n	1	0	0	Access operand Increment PC	.22	.26
30-36	@(R) +	0	2	0	Access addr of operand Access operand Increment R	.89	1.07
37	@(PC) + or @#n	1	1	0	Access addr of operand Access operand Increment PC	.67	.80
40-46	-(R)	0	1	0	Decrement R Access operand	.67	.80
47	-(PC)	0	1	0	Decrement PC Access operand	1.55	1.86
50-56	@-(R)	0	2	0	Decrement R Access addr of operand Access operand	1.11	1.33
57	@-(PC)	0	2	0	Decrement PC Access addr of operand Access operand	2.00	2.40
60-67	INDEX(R)	1	1	0	Access index word Sum with R Access operand	.89	1.07
70-77	@INDEX(R)	1	2	0	Access index word Sum with R Access addr of operand Access operand	1.33	1.60

*0 if source mode also equals 00-07.

Destination Addressing Modes for JMP

The JMP instruction performs one less read than an equivalent MOV to PC instruction and is implemented with its own microroutines.

Destination Mode	Macro-11 syntax	I-stream reads	Data reads	Data writes	CPU activities required	11/84/83 Single Op	73/53
00-17	R	N/A	N/A	N/A	Illegal instruction – Trap through 10	N/A	N/A
10-11	@R or (R)	0	0*	0	PC←R	.89	1.07
20-26	(R) +	0	0*	0	PC←R Increment R	1.33	1.60
27	#n	0	0*	0	PC←PC Increment PC	1.33	1.60
30-36	@(R) +	0	1*	0	Access operand PC←operand Increment R	1.11	1.33
37	@#n	1	0*	0	Access operand PC←operand	1.11	1.33
40-47	-(R)	0	0*	0	Decrement R PC←R	1.11	1.33
50-57	@-(R)	0	1*	0	Decrement R Access operand PC←operand	1.33	1.60
60-66	Index(R)	1	0*	0	Access index word PC←R + index	1.33	1.60
67	Symbol	1	0*	0	Access symbol word PC←PC + symbol + 2	1.11	1.33
70-76	@Index(R)	1	1*	0	Access index word $Addr_T$←R + index Access T PC←T	1.55	1.80
77	@ Symbol	1	1*	0	Access symbol word $Addr_T$←PC + symbol + 2 Access T PC←T	1.55	1.86

*PDP-11/84 performs one additional read (e.g., 1 instead of 0)

Destination Addressing Modes for JSR

The Jump-to-Subroutine instruction is similar to the JMP instruction, except that JSR must push the old contents of the destination register onto the SP stack. This requires extra microcycles and one data write.

Destination Mode	Macro-11 syntax	I-stream reads	Data reads	Data writes	CPU activities required	11/84/83 Single Op	73/53
00-17	R	N/A	N/A	N/A	Illegal instruction– Trap through 10	N/A	N/A
10-11	@R or (R)	0	0*	1	PC←R	2.00	2.40
20-26	(R) +	0	0*	1	PC←R Increment R	2.22	2.66
27	#n	0	0*	1	PC←PC Increment R	2.22	2.66
30-36	@(R) +	0	1*	1	Access operand PC←operand Increment R	2.22	2.66
37	@#n	1	0*	1	Access operand PC←operand	2.00	2.40
40-47	-(R)	0	0*	1	Decrement R PC←R	2.22	2.66
50-57	@-(R)	0	1*	1	Decrement R Access operand PC←operand	2.22	2.66
60-66	Index(R)	1	0*	1	Access index word PC←R + index	2.22	2.66
67	Symbol	1	0*	1	Access symbol word PC←PC + symbol + 2	2.00	2.40
70-76	@Index(R)	1	1*	1	Access index word $Addr_T$←R + index Access T PC←T	2.66	3.19
77	@ Symbol	1	1*	1	Access symbol word $Addr_T$←PC + symbol + 2 Access T PC←T	2.66	3.19

*PDP-11/84 performs one additional read (e.g., 1 instead of 0)

Destination Addressing Modes for MOV(B) and CLR(B) (write-only)
The Move and Clear instructions never need to access the destination data, because that data is merely overwritten. Therefore, some PDP-11 processors suppress the reading of the destination data for MOV(B), CLR(B), or both. This speeds execution. Other processors may read the data and then discard it. See the PDP-11 difference list (Appendix A) for details.

Destination Mode	Macro-11 syntax	I-stream reads	Data reads	Data writes	CPU activities required	11/84/83	73/53
00-07	R	0	0	0	<None>	1.11	1.33
10-16	@R or (R)	0	0	1	Write operand	.44	.53
17	@PC or (PC)	0	0	1	Write operand	1.33	1.60
20-26	(R) +	0	0	1	Write operand Increment R	.44	.53
27	(PC) + or #n	0	0	1	Write operand Increment PC	1.33	1.60
30-36	@(R) +	0	1	1	Access addr of operand Write operand Increment R	.89	1.07
37	@(PC) + @#n	1	0	1	Access addr of operand Write operand Increment PC	.67	.80
40-46	-(R)	0	0	1	Decrement R Write operand	.67	.80
47	-(PC)	0	0	1	Decrement PC Write operand	1.55	1.86
50-56	@-(R)	0	1	1	Decrement R Access addr of operand Write operand	1.11	1.33
57	@-(PC)	0	1	1	Decrement PC Access addr of operand Write operand	2.00	2.40
60-67	INDEX(R)	1	0	1	Access index word Sum with R Access operand	.89	1.07
70-77	@INDEX(R)	1	1	1	Access index word Sum with R Access addr of operand Access operand	1.33	1.60

*0 if source mode also equals 00-07.

Destination Addressing Modes for All Other Instructions (read-write)
All other instructions read the old destination operand, modify the data, and then write back new data.

Destination Mode	Macro-11 syntax	I-stream reads	Data reads	Data writes	CPU activities required	11/84/83	73/53
00-06	R	0	0	0	<None>	0	0
07	R	0	0	0	<None>	1.11	1.33
10-16	@R or (R)	0	1	1	Access operand Write-back operand	.67	.80
17	@PC or (PC)	0	1	1	Access operand Write-back operand	1.55	1.86
20-26	(R) +	0	1	1	Access operand Increment R Write-back operand	.67	.80
27	(PC) + or #n	0	1	1	Access operand, Write-back operand Increment PC	1.55	1.86
30-36	@(R) +	0	2	1	Access addr of operand Access operand Write back operand Increment R	1.11	1.33
37	@(PC) + @#n	1	1	1	Access addr of operand Access operand Write-back operand Increment PC	.89	1.07
40-46	-(R)	0	1	1	Decrement R Access operand Write-back operand	.89	1.07
47	-(PC)	0	1	1	Decrement PC Access operand Write-back operand	1.78	2.14
50-56	@-(R)	0	2	1	Decrement R Access addr of operand Access operand Write-back operand	1.33	2.60
57	@-(PC)	0	2	1	Decrement PC Access addr of operand Access operand Write-back operand	2.22	2.66
60-67	INDEX(R)	1	1	1	Access index word Sum with R Access operand Write-back operand	1.11	1.33
70-77	@INDEX(R)	1	2	1	Access index word Sum with R Access addr of operand Access operand Write-back operand	1.55	1.86

*0 if source mode also equals 00-07.

Fetch and Execution Timing

Operand access and update has already been described. The rest of the work performed by an instruction consists of the opcode fetch, the actual data manipulation(s), and any implicit reads or writes.

The table below documents the number of microcycles required times the basic clock time of the processor. Thus, the table directly lists the duration of the fetch and execution phase in microseconds.

The table also indicates the number of implicit data reads and writes that must be performed to complete execution of the operation, including the opcode fetch.

Notes:

1 The execution time of the ASH and ASHC instructions depends on the number of shifts performed. The table states the minimum time (for 0 bits shifted) plus the incremental time per bit shifted, or the minimum-maximum times.

2 The execution time for the branch instructions varies depending on whether or not the branch is taken. The table states the values for branch-not-taken and branch-taken, respectively. In addition, on the PDP-11/84, taking the branch flushes the instruction buffer.

3 On a given processor, all cases of the CLx and SEx condition code operators execute in the same amount of time.

4 Special case—see JMP Destination Table.

5 Special case—see JMP Destination Table.

6 1.33 if other than RTS, PC.

7 These instructions flush the instruction buffer, requiring an extra instruction to refill the instruction buffer.

8 These instructions flush the instruction buffer, requiring an extra instruction to refill the instruction buffer.

9 The value stated in the table is the minimum execution time for the WAIT instruction. WAIT actually continues executing until the next interrupt request is received.

▪ Floating-Point Instructions

The same basic methodology can be applied to floating point instructions, with a few additional concerns.

Operand Length

PDP-11 floating-point operands may be 32- or 64-bits in length. In addition, mode 27 operands result in the access of just 16 bits. This means that one,

two, or four words must be read or written from or to memory to access or write floating-point operands. This affects the execution time of the floating-point instructions.

Data Dependencies

Most of the basic PDP-11 instructions require a fixed number of microcycles to execute. ASH and ASHC (the long-shift instructions) vary in the number of microcycles based on the number of bits shifted, while BRANCH instructions vary slightly depending on whether or not the branch is taken.

On the other hand, the number of microcycles required to execute a floating-point instruction can vary greatly depending on the specific floating-point data used.

This is due to two facts:

- Before two floating-point numbers can be added or subtracted, their radix points must be aligned.
- Before a floating-point number can be stored, it must be normalized (so that its mantissa is in the range of 0.5 to 0.999...).

▪ PDP-11/83 and 84 Floating Point Instruction Timing: FPJ11

Because the FPJ11 is a co-processor operating in parallel with the J-11 chipset, the calculation of floating point instruction times (for J-11 systems that utilize the FPJ11) must take this parallel processing into account. The following are the terms and definitions for those terms used in describing the timing.

FPJ11 cycle—two clock periods (110 ns at 18 MHz).

J-ll nonstretched cycle—two FPJ11 cycles (220 ns at 18 MHz).

J-11 read cycle—J-11 nonstretched cycle if cache hit. Dependent on read access time of the system if cache miss, the minimum is two J-11 nonstretched cycles, after which the J-11 stretches in ½ cycle increments until MCONT is asserted.

J-11 write cycle—dependent on write access time of the system. (Two J-11 cycles + ½ cycles until MCONT).

Instruction Decode—a decode/prefetch cycle followed by a MOV microinstruction which allows the FPJ11 to assert DMR prior to the start of the next microinstruction (INPR for REG mode). This time equals two nonstretched cycles if the prefetch is a cache hit, else one nonstretched cycle plus one read cycle.

Address Calculation Time—J-11 time required to calculate the address of the operand. This time is dependent on the addressing mode of the instruction, the frequency of the system clock, and whether any indirect data required is present in the cache.

Argument Transfer Time—J-11 time required to load or store floating-point operands. This time is one nonstretched cycle (address relocation microcycle) plus one read cycle per 16-bit word read from memory for load class instructions—or one nonstretched cycle plus one write cycle per 16-bit word written to memory for store class instructions.

INPR (FEATEMP, TEMP)—J-11 support code microinstruction executes for all FPJ11 instructions. It moves the PC of the previous FPJ11 instruction to a TEMP register in case that instruction resulted in a floating-point exception. If the FPJ11 is still executing the previous floating-point instruction when the J-11 reaches its INPR microinstruction, the FPJ11 asserts STALL causing the J-11 INPR microcycle to stretch. The J-11 then WAITs for the FPJ11 to deassert STALL, signaling the system interface to assert MCONT before executing the next microinstruction (OUTR).

WAIT—J-11 time waiting for the completion by the FPJ11 of the previous floating-point instruction. For load class or REG mode instructions, this time is from when the J-11 INPR cycle stretches at the trailing edge of MALE until the FPJ11 deasserts STALL. This time equals zero if a stall was not required or if the FPJ11 deasserted the STALL signal after the INPR cycle began, but prior to the trailing edge of MALE. Although the WAIT time for this latter case is zero, RESYNC time is required. For store class instructions the WAIT time equals the time between the assertion of SCTL (when the system interface is ready to execute the first write cycle of a floating-point store) and the assertion of FPA-RDY (data ready) by the FPJ11.

RESYNC—For load class and REG mode instructions this is the time required to continue a stretched INPR. This is the time for the SYSTEM INTERFACE to recognize the deassertion of STALL and assert MCONT, plus the time required for the J-11 to synchronize MCONT and advance to the next microinstruction. Store class instructions normally do not have RSYNC time because the J-11 is waiting in a stretched write cycle and the continuation time is part of the write cycle. However, if the FPJ11 is executing a previous MODF/D or DIVD, the FPJ11 will assert STALL in order to stretch a non-I/O cycle prior to the first bus write. This allows the SYSTEM INTERFACE to service DMA thus limiting the worst case DMA latency when waiting for FPJ11 output. In this case a WAIT and RESYNC time associated with the stretched non-I/O cycle is added to the effective execution time of the store class instruction.

OUTR (PC,FEATEMP), TESTPLA FPE—last J-11 support microinstruction unless there is a floating-point exception from the previous floating-point instruction. This saves the address of PC in FEATEMP.

PRDC SYNC—time required by FPJ11 to decode a floating-point instruction and begin execution after receiving PRDC. This time equals two or three FPJ11 cycles depending upon synchronization. PRDC SYNC is not added to FPJ11 instruction execution times when the FPJ11 is executing a previous floating-point instruction at the assertion of PRDC.

Floating-Point Execution Time—time required by FPJ11 to complete a floating-point instruction once it has received all arguments. For store class instructions, floating-point execution time includes the time from the start of the instruction until the FPJ11 asserts FPA-RDY indicating the first 16-bit word is available for output.

Effective Execution Time—total J-11 time required to execute a floating-point instruction.

Load class—Instruction Decode + Address Calculation + Argument Transfer + INPR + WAIT + RSYNC + OUTR.

REG mode—Instruction Decode + INPR + WAIT + RSYNC + OUTR.

Store class—Instruction Decode + Address Calculation + INPR + Argument Transfer + WAIT + OUTR.

Load class instructions require input data and deposit results to the destination floating-point accumulator. REG mode instructions are floating-point accumulator to floating-point accumulator.

Execution of a load class floating-point instruction by the FPJ11 occurs in parallel with J-11 operation and can be overlapped. See Figure C-2.

Store class instructions can be overlapped by the J-11 as the FPJ11 will complete a previously started load class or REG mode instruction and then continue to store the instruction. Execution of the store class instruction must be completed before the result can be stored to memory, thus eliminating further parallel processing for store class floating-point instructions.

Table C-2 ▪ Address Calculation Times

MODE	LOAD CLASS	STORE CLASS
0	0	0
1	3	3
2	3	2
3	3 + RD	2 + RD
4	4	4
5	3 + RD	3 + RD
6	3 + RDI	2 + RDI
7	3 + RDI + RD	3 + RDI + RD
27	2	2
37	2 + RDI	1 + RDI
67	3 + RDI	2 + RDI
77	4 + RDI + RD	4 + RDI + RD

RDI = J-11 Istream request RD = J-11 Read cycle

Table C-3 ▪ FPJ11 Instruction Times

INSTRUCTION	MINI-MUM CYCLES	TYPICAL CYCLES	MAXI-MUM CYCLES	STRETCH CYCLES	18 mHz TYPICAL (usec)
ADDF/SUBF	7	9	19	5	1.0
ADDD/SUBD	15	9	30	5	1.0
MULF	26	15	16	11	1.7
MULD	17	26	27	22	2.9
DIVF	33	24	30	25	2.7
DIVD	28	48	62	57	5.4
MODF	39		43	15	
MODD	3		71	26	
CMPF/D	3		6	2	
LDF/D	2		3	0	
LDEXP	10		2	0	
LDCIF/D	10		10	3	
LDCLF/D	4		10	3	
LDCFD	4		4	1	
LDCDF	3		8	1	
STF/D	8		3	0	
STCFI	8		13	1	
STCFL	4		16	1	
STCFD	6		4	0	
STCDF	5		6	1	
STEXP	3		5	0	
TSTF/D, LDFPS	4		3	0	
STFPS, CFCC, SET ABSF/D, NEG7			5	0	

NOTE

Stretch cycles indicate the number of cycles out of maximum cycles that a data dependent stretch of one additional cycle could occur with probability less than 1% for each additional cycle.

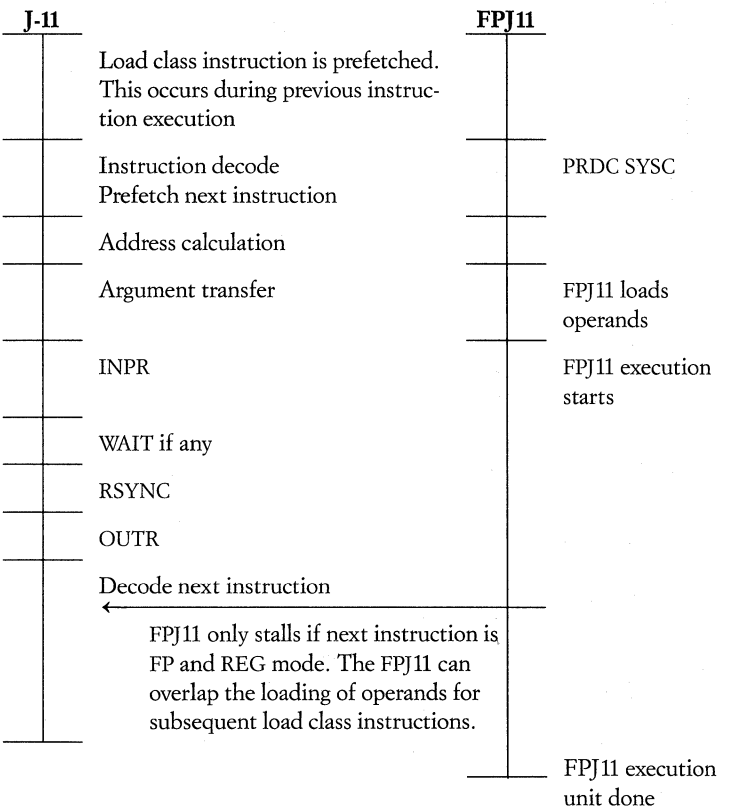


Figure C-1 ▪ J11/FPJ11 Interaction for Load Class Instructions

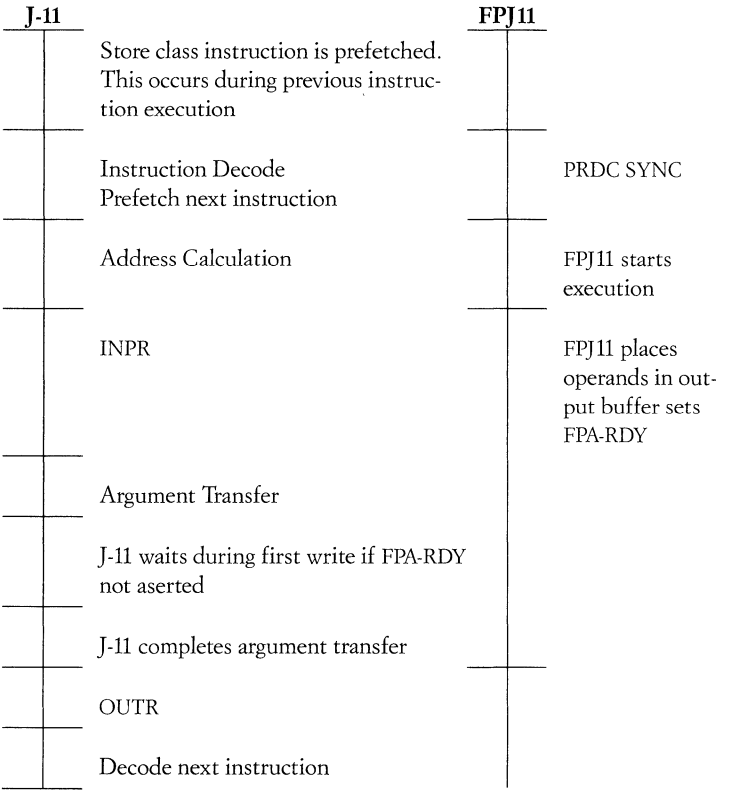


Figure C-2 ▪ J-11/FPJ11 Interaction for Store Class Instructions

Appendix D ▪ MicroPDP-11/23 System and Upgrades

▪ Introduction

Digital's proud PDP-11 history includes a number of systems that continue to serve customers but that are no longer actively marketed. This appendix discusses one such system: the MicroPDP-11/23. More detailed examination of this system is available in other handbooks.

▪ MicroPDP-11/23 Features

The MicroPDP-11/23 system is ideally suited to the small business or departmental computing environment. The large number of MicroPDP-11/23 configurations serves wide variety of customer requirements. A low-cost, entry-level system, the MicroPDP-11/23 fits on or under a desktop and includes in a single package the processing components and mass storage. Because the MicroPDP-11/23 is a full member of Digital's family of 16-bit supermicrocomputers, the MicroPDP-11/23 can use most of the software already written for PDP-11s, including a wide variety of operating systems, languages, application software, and communications packages. The MicroPDP-11/23 can also link with existing corporate mainframes and other computer systems via Digital's powerful DECnet networking software. Users can add more power as needs grows.

Over 50 configurations of the MicroPDP-11/23 have been offered since the system was introduced providing configuration flexibility for a number of computer solutions. These configurations include:

- RD51-based systems.
- RD52-based systems.
- RX50-based systems.
- RC25-based systems.
- Packaged automation systems.
- Standard systems.

The current configurations, including RD52-based systems and RX50-based systems, incorporate diagnostics with the latest in component technology for highly reliable system use under demanding conditions.

The current MicroPDP-11/23 features include:

- Compact size.
- Accommodation of two to four active users and up to 14 devices such as terminals and printers.
- Support for a wide variety of operating systems, languages, and protocols.
- 256- or 512-Kbyte parity memory system.
- 31-Mbyte formatted Winchester fixed hard-disk.
- 800-Kbyte removable dual-diskette storage.
- Additional external fixed hard-disk expandability.
- Proven Digital KDF11-B CPU.
- Simple plug-in connections for terminals and external devices.

The MicroPDP-11/23 executes the full set of PDP-11 instructions and uses the same F-11 chipset as some larger PDP-11 systems. It provides 256 or 512 Kbytes of parity memory, expandable to a maximum of 4 Mbytes for large program capacity.

A wide variety of storage options are available, including internal disks, cartridge tape drives, and floppy disk drives.

The MicroPDP-11/23 disk controller uses the Mass Storage Control Protocol (MSCP), a Digital standard that:

- Accommodates all disk accesses, data transfers, error, recovery, and performance optimization.
- Permits the operating system to deal with MSCP, not with specific disk drives.

Because it's a PDP-11, the MicroPDP-11/23 can run many of the application packages available from Digital and from third-party software developers.

The MicroPDP-11/23 supports all PDP-11 operating systems. Micro/RSX, RSX11M, RSX11M-PLUS, and RSX-11S offer multiuser, multitasking commercial and realtime applications. Micro/RSTS AND RSTS/E provide multiuser timesharing and interactive programming development. MicroPower/Pascal offers realtime microcomputer applications development. ULTRIX-11, a commodity timesharing operating system, is Digital's enhanced native-

mode UNIX software. RT-11 is a single user realtime operating system. CTS-300 is an RT-11 based, small business timesharing system. DSM-11 is the MUMPS language and integrated operating system for hierarchical data management.

Additionally, a number of data management utilities and languages provide the MicroPDP-11/23 with enhanced utility. Refer to the Chapter 1 and the PDP-11 System Comparison Chart for further details.

The MicroPDP-11/23 can be linked with other computers. DECnet as well as SNA, X.25, and UNIVAC 1004 for multivendor connections, and the traditional 2780/3780 protocol products are available for MicroPDP-11/23 systems.

Depending on the operating system used, the MicroPDP-11/23 can connect to Ethernet local area networks for high-speed local area communications among, for example, departments in an organization.

The MicroPDP-11/23 can use up to 14 terminals, printers, and other devices.

The MicroPDP-11/23 uses the 16-bit KDF11-B central processing board with the F11 chipset, discussed below. Optional floating-point microcode chips add 46 additional instructions for floating-point mathematics and FORTRAN-77. A commercial instruction set chip for enhanced COBOL execution is also available.

▪ Central Processor

The KDF11-B series is a quad-height central processor module implemented in the MicroPDP-11/23 and earlier PDP-11/23 Plus systems. It is designed for moderate-speed, realtime applications and for multiuser, multitasking environments.

The KDF11-B module provides:

- F-11 microprocessor chip.
- Complete PDP-11 instruction set including the Extended Instruction Set (EIS).
- Floating-point and commercial instruction sets (optional).
- Q22-bus interface that supports block-mode DMA transfers and as many as 4 Mbytes of physical memory.
- Line-frequency clock.
- Four levels of interrupts.
- Powerfail/autorestart
- Console emulator in microcode.

- Two serial-line units—one for use as a console terminal and the other for use with a user terminal or serial printer.
- 8-Kbyte bootstrap/diagnostic read-only memory (ROM).

The KDF11-B module supports up to 256 Kbytes of memory on an 18-bit Q-bus backplane or up to 4 Mbytes of memory on a Q22-bus backplane. When used with the Q22 bus, the KDF11-B uses four level interrupt protocol. The 22-bit memory management is provided for both instruction and data references in two protection areas—kernel and user. The KDF11-B is fully downward compatible with older PDP-11s that have 18-bit memory management.

The KDF11-B supports console emulation (micro octal debugging tool, or ODT). This allows users to interrogate and write to main memory, CPU registers, and I/O devices.

Self-diagnostic LEDs indicate the status of the module and the system when the module is powered. The LEDs aid in troubleshooting module failures. The LEDs also appear on the I/O distribution panel.

The KDF11-B module supports all MicroPDP-11 operating systems.

▪ System Chassis and Backplane

The MicroPDP-11/23 chassis contains the following components:

- 8-slot Q22-bus backplane.
- dc power supply.
- Front control panel.
- I/O distribution panel.
- Two cooling fans.
- CPU module(s).
- System modules (optional).
- 5.25-inch fixed-disk drive (optional).
- 5.25-inch dual-diskette drive.

H2978-A Backplane

The logic module for the MicroPDP-11/23 is installed in the H2978-A backplane assembly, included with the system chassis. The assembly consists of four rows by eight slots of prewired connectors and a mounting frame that allows quad- or quad-height logic modules to be easily inserted and removed.

A card guide also permits the latches on the quad-height modules to hold securely onto the backplane.

The backplane incorporates the Q22-bus wiring in rows A and B of connector slots one through eight and in rows C and D of connector slots four through eight. The Q22 bus supports an interrupt and DMA grant-continuity scheme for the logic modules installed in the backplane. The backplane provides the 120-ohm far-end termination for the system.

Four connectors on the backplane, J1 through J4, receive voltages and signals from the H7864 power supply and provide signals and voltages to the front control panel. Maximum ratings are 36A at + 5V and 6A at + 12V.

Table D-1 ▪ Logic-module Assignments

Slot	MicroPDP-11/23
1	CPU
2	Memory
3	Additional memory or communications option
4	Two dual-height or one quad-height option
5	See Slot 4
6	See Slot 4
7	See Slot 4
8	See Slot 4

H7864 Power Supply

The H7864 modular power supply is common to the basic microsystem chassis. It drives 230 watts of power to the logic modules mounted in the system backplane, to the control panel switches and indicators, and to the two dc fans.

Power supply features include:

- Universal supply with switchable inputs for 88-128 V RMS at 120V/60 Hz and 176-256V RMS at 240 V/50 Hz.
- Separate output circuit for fans.
- Line voltage conditioning.
- Q22-bus compatible power sequencing signals.
- 12 V dc power rating of 7 A maximum.

Additional H7864 features include thermal shutdown, overvoltage protection, and ac input transient suppression.

Control Panel

Control switches and indicator lights are located at the front of the system chassis. These controls allow you to apply and remove power, start or stop the current program operation, and protect the data stored on the fixed disk drive.

- ac power—is controlled by the ac power rocker switch marked 0 and 1.
- Halt—halts the operation of the current program.
- Restart—initiates a process restart.
- Fixed Disk 0 Write Protect—prevents data from being written to the fixed disk.
- Fixed Disk 0 Ready—disables disk-drives to prevent read or write.
- Run Indicator—indicates a process in operation.
- DC OK Indicator—indicates correct dc voltages.
- Removable Disk Write Protect 1—indicates that data cannot be written to removable disk 1.
- Removable Disk Write Protect 2—indicates that data cannot be written to removable disk 2.

▪ Memory Systems

The MicroPDP-11/23 uses the MSV11-P series of random-access memory(RAM) modules that provide 256 Kbytes (MSV11-PK) or 512 Kbytes (MSV11-PL) of main memory storage. The following are the main features of this series:

- 256 Kbytes or 512 Kbytes of MOS memory on a single, quad-height module.
- 64-Kbyte MOS RAM integrated circuitry.
- Support for 18-bit or 22-bit addressing for as many as 4 Mbytes of physical memory.
- Support for block-mode DMA transfers.
- LEDs for parity-error indication.

▪ Performance Options

Three options are available to enhance the performance of the MicroPDP-11/23. A floating-point processor and a floating-point accelerator are available for applications that require a great deal of calculation. A character-string instruction set (Commercial Instruction Set) is offered for business applications.

KEF11-AA Floating-point Processor
KEF11-AA is a single- and double-precision floating-point option that implements PDP-11 floating-point instructions. The microcode resides in two chips in one 40-pin package that mounts directly on the CPU module. This option performs operations on 32-bit and 64-bit floating-point numbers and provides up to 17 digits of precision. KEF11-AA also provides integer-to-floating-point conversion.

FPF11 Floating-point Accelerator
FPF11 is a single-precision and double-precision floating-point hardware option that executes instructions approximately six times faster than the KEF11-AA. This option is a single, quad-height module and mounts adjacent to the CPU board. FPF11 performs hardware operations on 32-bit and 64-bit floating-point numbers and provides up to 17 digits of precision. Like the KEF11-AA, it provides integer-to-floating-point conversions.

KEF11-BB Character-string Instruction Set
KEF11-BB implements a set of 27 commercial instructions on a variety of data types, including character-string, packed-decimal, and numeric formats. Because these data types closely resemble those used in COBOL, they are referred to as the Commercial Instruction Set (CIS). KEF11-BB mounts directly on the CPU board.

▪ Storage Options

Many storage options are available for the MicroPDP-11/23. The RQDX series of disk controllers coordinates the activities of these options by providing data transfers between the Q22 and the RX50 and RD-series of drives. These controllers contain logic that provides the necessary data buffering and control to allow direct-memory access (DMA) transfers using the Mass Storage Control Protocol (MSCP).

The following are the main features of the RQDX series:

- Control for up to four logical disk drive units.
- Support for block-mode DMA data transfers.

- Quad-height module size for the RQDX1 and RQDX2. Dual-height module for the RQDX3.
- Maintenance self-test programs.
- LED parity-error indication.

▪ Specifications

Identification	M8189
Size	Quad
Power Requirements	5 V ± 5%, 6.4 A max (4.5 A typ) 12 V ± 5%, 0.7 A max (0.3 A typ)
Bus Loads ac dc	21
Instruction Timing	Based on 75 ns intervals
Interrupt Latency	5.7 μsec (typ) 12.6 μsec (max), except EIS 54.225 μsec (max), including EIS
Interrupt Service Time	8.625 μsec (memory management off) 9.750 μsec (memory management on)
DMA Latency	1.35 μsec (max)

NOTE

Interrupt and DMA latencies assume a KDF11-B with memory management enabled and using MSV11-P Memory.

Nonstandard Environmental Specifications	
Operating temperature	5°C to 50°C (41°F to 122°F)
Operating altitude	Up to 2.44 km (8,000 ft)

▪ Board Configuration

Jumper and Switch Configuration

The MicroPDP-11/23 CPU board contains two dual inline pack (DIP) switchpacks (E102 and E114) and several jumpers that allow the user to

select the module features desired. The boot/diagnostic switchpack (E102) consists of eight switches that let the user select boot and diagnostic programs. The second switchpack (E114) selects the baud rate for the console SLU and the second SLU. The module contains both wirewrap jumper pins and soldered-in jumpers. The jumpers are divided into the following functional groups:

- Test jumpers.
- CPU option jumpers.
- Device selection jumpers.
- Boot and diagnostic ROM jumpers.
- SLU character format jumpers.
- Internal/external SLU clock jumpers.
- Q-bus backplane jumpers.

CPU Option Jumpers

Four wirewrap pins provide user-selectable features associated with the operation of the CPU. The ground pin can be connected to any combination of the other three pins to select the available features. Two powerup mode pins select one of three powerup modes. The halt/trap pin selects the halt/trap options.

Powerup Mode Selection

Three powerup modes are available for user selection. Selection is made by installing or removing wirewrap jumpers between jumper pins (J17, J19) and the ground pin (J18) in various combinations.

Table D-2 ▪ LSI-11/23-PLUS Powerup Mode Jumpers

Mode	Name	J18 to J19	J18 to J17
0	PC @24, PS @26	Removed	Removed
1	Console ODT	Removed	Installed
2	Bootstrap	Installed	Removed
3	Not Implemented	Installed	Installed

Only the powerup mode is affected—not the power-down sequence.

▪ System Upgrade

Common features in system hardware, software, enclosures, and bus architectures among MicroPDP-11 systems allow for a variety of growth paths for the MicroPDP-11/23, either within the PDP-11 family or as entry into the VAX family. Upgrade kits make it easy to enhance performance or capacity and preserve investment in both hardware and software.

MicroPDP-11/23 to MicroPDP-11/73

11/73-UA, the upgrade kit to the MicroPDP-11/73, provides an ability to upgrade to the higher performance J11 technology while preserving investments in software and Q-bus peripherals. The 11/73-UA provides a KDJ11-BB(11/73) CPU board and cabinet kit, diagnostics on RX50, documentation, and Field Service installation package.

MicroPDP-11/23 to MicroPDP-11/83

11/83-UB, the upgrade kit to the MicroPDP-11/83, provides an ability to upgrade to the higher performance and higher capacity MicroPDP-11/83 while preserving investments in hardware and peripherals. The 11/83-UB consists of a KDJ11-BF(11/83) CPU board and cabinet kit, 2 Mbytes PMI memory, diagnostics on RX50, documentation, and Field Service installation package.

MicroPDP-11/23 to MicroVAX II

The MicroPDP-11/23 can also be upgraded to an entry level VAX system, the Micro VAX II. The MicroVAX II Upgrade Kit, 630VR-BB, offers the power and reliability of a VAX, while preserving investments in Q-bus peripherals. 630VR-BB consists of a MicroVAX II CPU, 2 Mbytes of memory, RQDX3 disk controller, documentation, diagnostics on RX50, MicroVMS 1 to 8-user license and key, Field Service installation and integration package.

Appendix E ▪ PDP-11/24 and PDP-11/44 Systems

▪ Introduction

Digital's proud PDP-11 history includes a number of systems that continue to serve customers but that are not currently marketed. This appendix discusses two such systems: the PDP-11/24 and PDP-11/44. More detailed examination of these systems is avalable in other handbooks.

▪ PDP-11/44 and PDP-11/24

The PDP-11/44 processor delivers the capability needed to satisfy a wide range of application requirements. Many outstanding features such as a high-performance central processor, access up to four Mbytes of main memory, and a large 8-Kbyte parity cache memory are standard on the PDP-11/44. Available options include the floating-point processor, commercial instruction set processor and the battery backup unit.

The PDP-11/24 is a fourth-generation member of the UNIBUS PDP-11 processor family. Designed as a single-hex module UNIBUS processor, the PDP-11/24 provides the basis for compact, low-cost application solutions. Offering an extended 22-bit memory addressing capability, the PDP-11/24 can address up to four full Mbytes of memory. The PDP-11/24 optional floating-point unit and commercial instruction set provide programming compatibility with other PDP-11s.

Integral to the PDP-11/24 central processor unit are hardware features and expansion capabilities that are common to the PDP-11/84, PDP-11/44, PDP-11/70 and PDP-11/34A.

Many of the hardware features and expansion capabilites of the PDP-11/44 are common to other PDP-11s. Appendix A illustrates the similarities and differences between the PDP-11/44, PDP-11/84, PDP-11/70, PDP-11/24, PDP-11/34A, and other PDP-11 systems.

▪ System Architecture

The PDP-11/44 is a medium-scale, general purpose computer that is designed according to an enhanced, upwardly compatible version of the basic PDP-11 architecture. A block diagram is shown in Figure E-1.

Memory management is standard with the basic computer, allowing expanded memory addressing, relocation, and protection. Also standard is a UNIBUS map that translates 18-bit UNIBUS addresses to 22-bit physical memory addresses. The cache contains 8192 bytes of fast, static MOS memory that buffers the processor data from main memory.

The PDP-11/44 system has an expanded internal implementation of the PDP-11 architecture for greatly improved system throughput. All memory is on its own high-data-rate bus. The processor has a direct connection to the cache memory system for very rapid memory access.

The UNIBUS remains the primary control path in the PDP-11/44 system. It is conceptually identical with all other PDP-11 systems; the memory in the system still appears to be on the UNIBUS to all UNIBUS devices through the UNIBUS map. This expanded internal implementation of the PDP-11 architecture is generally compatible with earlier PDP-11/70 programs.

The PDP-11/24 is a minicomputer designed for both multitasking and dedicated applications. A block diagram of the computer is shown in Figure E-2.

The central processor performs all arithmetic and logical operations required in the system. Memory management is standard with the basic computer, allowing expanded memory addressing, relocation, and protection. The UNIBUS map, which translates UNIBUS addresses to physical memory address, is program compatible with PDP-11/44 and PDP-11/84 UNIBUS maps. The UNIBUS remains the primary control path in the PDP-11/24 system. Memory addresses are passed on a separate 22-bit wide bus. This bus provides reduced memory access times. It is conceptually identical with previous PDP-11 systems; the memory in the system still appears to be on the UNIBUS to all UNIBUS devices through the UNIBUS map.

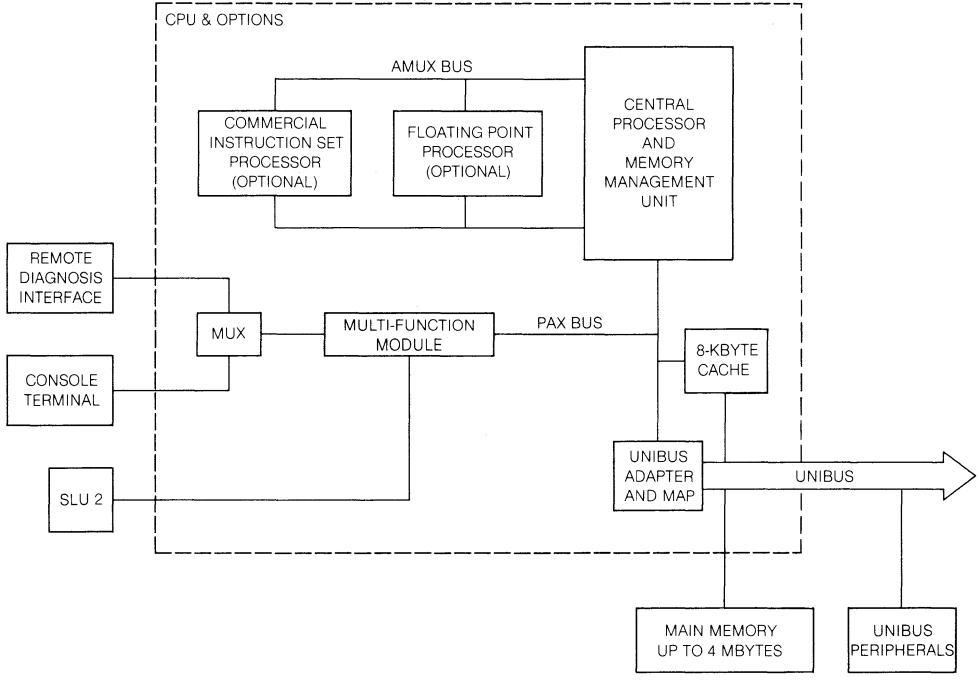


Figure E-1 ▪ PDP-11/44 Block Diagram

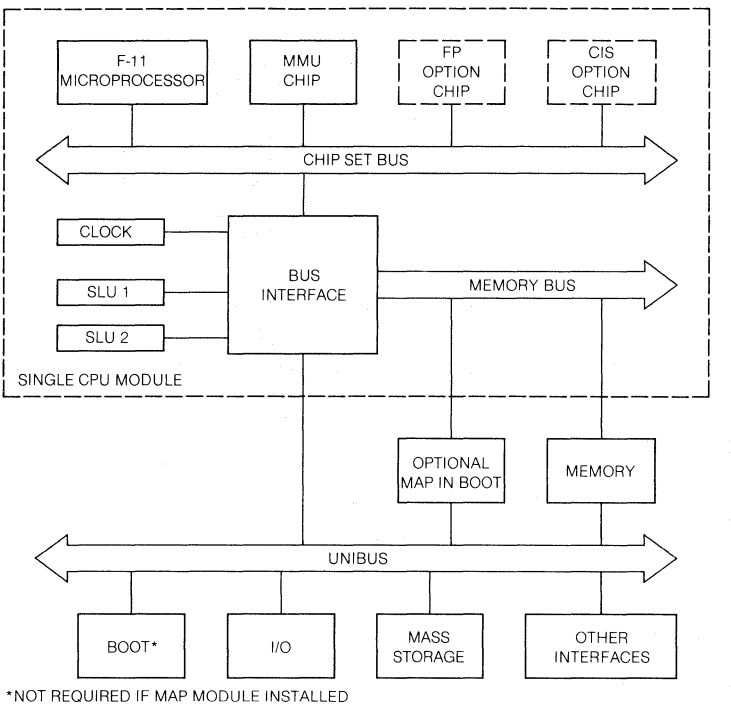


Figure E-2 ▪ PDP-11/24 Block Diagram

▪ PDP-11/44 and PDP-11/24 Central Processor

The PDP-11/44 processor acts as the arbitration unit for UNIBUS control by regulating bus requests and transferring control of the bus to the requesting device with the highest priority.

The central processor contains arithmetic and control logic for a wide range of operations. These include fixed point arithmetic with hardware multiply and divide, extensive test and branch operations, and other control operations. It also provides room for the addition of the floating-point processor, commercial instruction set, and UNIBUS options.

The machine operates in three modes—kernel, supervisor, and user. When the machine is in kernel mode, a program has complete control of the machine; when the machine is in any other mode, the processor is inhibited from executing certain instructions and can deny direct access to the peripherals on the system. This hardware feature can be used to provide complete executive protection in a multiprogramming environment.

The central processor contains six general registers that can be used as accumulators, index registers, or stack pointers. Stacks are extremely useful for nesting programs, creating reentrant coding, and as temporary storage where a last-in/first-out structure is desirable. An additional register is used as the PDP-11/44's program counter. Three additional registers are used as processor stack pointers, one for each operational mode.

The CPU performs all of the computer's computation and logic operations in a parallel binary mode through step-by-step execution of individual instructions.

The PDP-11/24 processor is the arbitration unit for UNIBUS control. It regulates bus requests and transfers control of the bus to the request device with the highest priority.

The central processor contains arithmetic and control logic for a wide range of operations. These include fixed-point arithmetic with hardware multiply and divide, extensive test and branch operations, and other control operations. It also provides room for the addition of the floating point unit, commercial instruction set, and UNIBUS options.

The machine operates in two modes—kernel and user. When the machine is in kernel mode, a program has complete control of the machine; when the machine is in user mode, the processor is inhibited from executing certain instructions and can be denied direct access to the peripherals on the system. This hardware feature can be used to provide complete executive protection in a multiprogramming environment.

The PDP-11/24 processor is implemented using three chips. Two MOS/LSI chips, called the data chip and control chip, implement the basic processor. The memory management unit (MMU), the third chip, provides a software-compatible memory management scheme.

The data chip (DC302) performs all arithmetic and logical functions, handles data and address transfers with the external world, and coordinates most interchip communication. The control chip (DC303) does micro-program sequencing for PDP-11 instruction decoding and contains the control store ROM. The data and control chips are both contained on one 40-pin package. The MMU chip (DC304) contains the registers for 18-bit or 22-bit memory addressing and also includes the FP11 floating-point registers and accumulators.

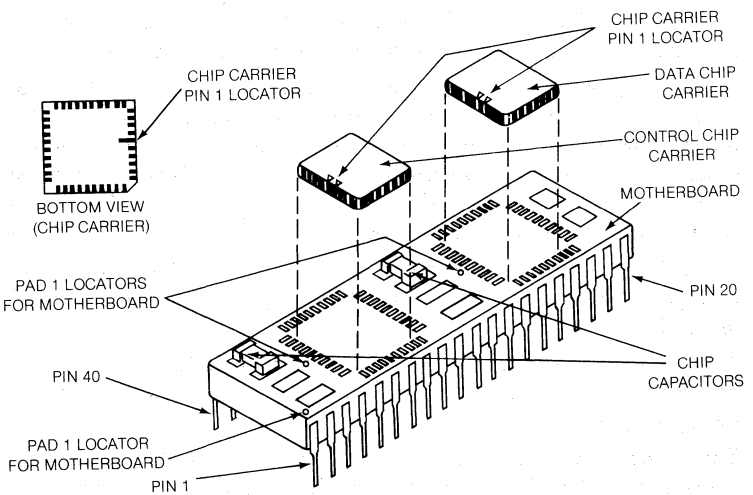


Figure E-3 ▪ PDP-11/24 Data and Control Chip

General Registers

The general registers, the same for both PDP-11/44 and PDP-11/24 systems except as noted, can be used in many ways, the uses varying with requirements. The general registers can be used as accumulators, index registers, autoincrement registers, autodecrement registers, or as stack pointers for temporary storage of data. The *PDP-11 Architecture Handbook* chapter on Addressing Modes describes these uses of the general registers in more detail. Arithmetic operations can be done from one general register to another, from one memory location or device register to another, or between memory or a device register and a general register.

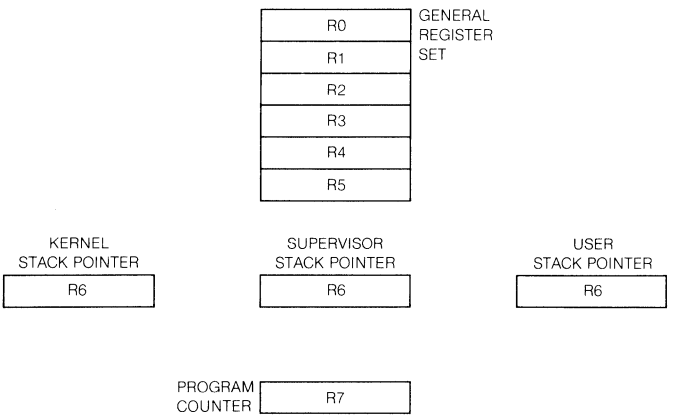


Figure E-4 ▪ The General Registers

R7 is used as the machine's program counter (PC) and contains the address of the next instruction to be executed. It is a general register normally used for addressing purposes and not as an accumu-lator for arithmetic operations.

The R6 register is normally used as the processor stack pointer indicating the last entry on the current mode's hardware stack. (For information on the programming uses of stacks, please refer to the *PDP-11 Architecture Handbook*.) The three stacks are called the kernel stack, the supervisor stack, and the user stack. When the central processor is operating in kernel mode, it uses the kernel stack; in supervisor mode, the supervisor stack; and in user mode, the user stack. When an interrupt or trap occurs, the current status on the stack selected by the service routine is saved automatically. This stack-based architecture facilitates reentrant programming. The remaining six registers are R0-R5.

Registers can be used to increase the speed of realtime data handling or facilitate multiprogramming. Each of the six general registers could be used as an accumulator or index register for a realtime task or device.

Processor Status Word (PSW)

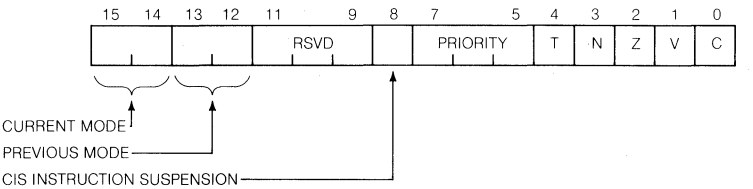


Figure E-5 ▪ 17 777 776 Processor Status Word

The processor status word (PSW), shown in Figure E-5, contains information on the current status of the PDP-11. This information includes current and previous operational modes, an indicator that is used to show that a Commercial Instruction Set (CIS) instruction was suspended by an interrupt, current processor priority, an indicator for detecting the execution of an instruction to be trapped during program debugging, and condition codes describing the results of the last instruction.

Modes

Mode information includes the present mode, either user, supervisor, or kernel (bits 15, 14), and the mode the machine was in before the last interrupt or trap (bits 13, 12).

The three (two in the PDP-11/24) modes permit a fully protected environment for a multiprogramming system by providing the user with three (two in the PDP-11/24) distinct sets of processor stacks and memory management registers for memory mapping.

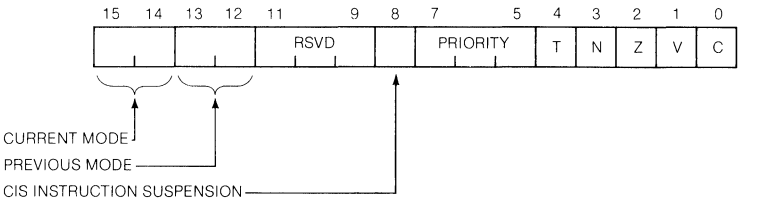


Figure E-6 ▪ 17 777 776 Processor Status Word

In user and supervisor modes, a program is inhibited from executing a HALT instruction, and the processor will trap through location 4 if an attempt is made to execute this instruction. Furthermore, the processor will ignore the RESET and SPL (Set Priority Level) instructions, and will execute No Operation. In kernel mode, the processor will execute all instructions.

A program operating in kernel mode can map users' programs anywhere in memory and thus explicitly protect key areas (including the device registers and the processor status word) from the user operating environment.

CIS Instruction Suspension

In both the PDP-11/44 and the PDP-11/24, bit 8, when set, indicates that a commercial instruction is in process. Because commercial instructions can be suspended (interrupted), this bit will be pushed onto the stack with the rest of the processor status word so that when control is returned to the routine, the commercial instruction can continue where it left off. Bit 8 may be used in future nonCIS instructions.

Processor Priority

The central processor operates at any of eight levels of priority, 0-7. When the CPU is operating at level 7, an external device cannot interrupt it with a request for service. The central processor must be operating at a lower priority than the priority of the external device's request in order for the interruption to take effect. The current priority is maintained in the processor status word (bits 5-7). The eight processor levels provide an effective interrupt mask, which can be dynamically altered by the kernel-mode program through use of the SPL instructions. (For more information on the instructions see the *PDP-11 Architecture Handbook*.) In the PDP-11/44 only, this SPL instruction allows a kernel mode program to alter the central processor's priority without affecting the rest of the processor status word.

Trace Trap

The trace trap (T) bit 4, can be set or cleared under program control. When the trace trap bit is set, a processor trap will occur through location 14 on completion of instruction execution and a new processor status word and program counter will be loaded. This bit is especially useful for debugging programs because it provides an efficient method of single-stepping the program.

Interrupt and trap instructions both automatically cause the previous processor status word and program counter to be saved and replaced by the new values corresponding to those required by the routine servicing the interrupt or trap. The user can thus cause the central processor to automatically switch modes (context switching), alter the CPU's priority, or disable the trace trap bit whenever a trap or interrupt occurs.

Condition Codes

The condition codes contain information on the result of the last CPU operation. They include a negative bit 3 (N), set if the result of the previous operation was negative; a zero bit 2 (Z), set if the result of the previous operation was zero; an overflow bit 4 (V), set if the result of the previous operation caused an arithmetic overflow, and a carry bit © set by the previous operation if the operation caused a carry out of its most significant bit.

Stack Limit

The PDP-11/44 and PDP-11/24 have a kernel stack overflow boundary at location 400. Once the kernel stack exceeds this boundary, the processor will complete the current instruction and then trap through location 4, stack overflow in the CPU error register.

▪ CPU Registers

The following CPU registers are accessed by program or console control.

PDP-11/44 CPU Error Register

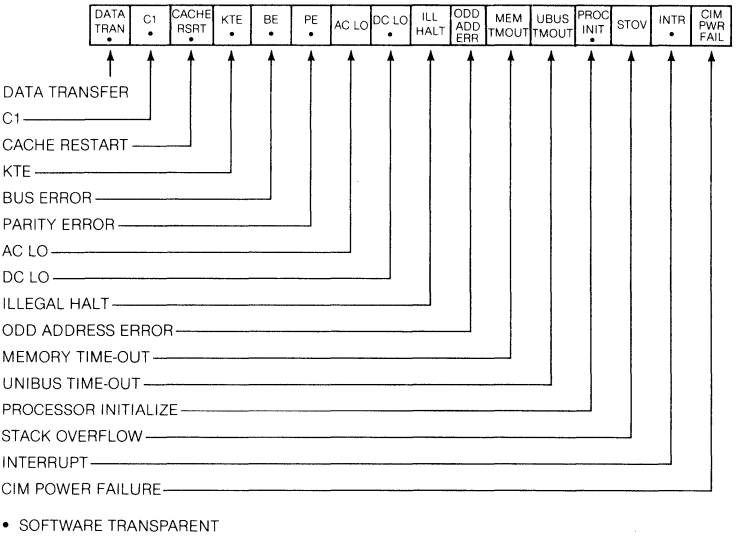


Figure E-7 ▪ 17 777 766 PDP-11/44 CPU Error Register

The CPU error register (shown in Figure E-7) identifies the source of the abort or trap that caused a trap through the vector at location 4. Bits 7–4, bit 2 and bit 0 are cleared when the CPU error register is written. When set, bit 9 indicates to software that a software powerdown is in progress. The remaining bits are software transparent and are accessible only when the console has control. They serve to provide diagnostic visibility into the processor.

Bit: 15

Name: DATA TRANSFER

Function: Monitors the DATA TRAN line of the processor. When clear, this bit indicates the processor is initiating a data transfer on the UNIBUS.

Bit: 14

Name: C1

Function: Set when the control signal Bus C1 is asserted, indicating that a DATO or DATOB transfer is being performed.

Bit: 13

Name: CACHE RESTART

Function: Set when the cache has generated the signal necessary to restart the processor clock.

Bit: 12

Name: KTE

Function: Set when a memory management error (nonresident, page length, or read-only abort) has occurred.

Bit: 11

Name: BUS ERROR

Function: Set when processor has attempted to access nonexistent memory, odd address during word reference, or if there was no response on the UNIBUS within approximately 20 microseconds.

Bit: 10

Name: PARITY ERROR

Function: Set when processor has received an indication of a memory parity error.

Bit: 9

Name: AC LO

Function: Set when UNIBUS AC LO is asserted. To software, when this bit is set, a powerdown is in progress. This signal is not latched and therefore bit 9 is not affected by a processor INIT.

Bit: 8

Name: DC LO

Function: Set when UNIBUS DC LO is asserted. This signal is not latched and therefore bit 8 is not affected by a processor INIT.

Bit: 7

Name: ILLEGAL HALT

Function: Set when a HALT instruction was attempted while the processor was in user or supervisor mode.

Bit: 6

Name: ODD ADDRESS ERROR

Function: Set when the program attempts a word reference of an odd address.

Bit: 5

Name: MEMORY TIMEOUT

Function: Set when the processor attempts to read/write from a non-existent main memory location. This does not include UNIBUS addresses.

Bit: 4

Name: UNIBUS TIMEOUT

Function: Set when the processor attempts to read/write from a non-existent UNIBUS location.

Bit: 3

Name: PROCESSOR INITIALIZE

Function: Set when the processor initialize signal is asserted.

Bit: 2

Name: STACK OVERFLOW

Function: Set when the kernel hardware stack is below virtual address 400 octal.

Bit: 1

Name: INTERRUPT

Function: Set when the PAX interrupt line is asserted.

Bit: 0

Name: CIM Power Failure

Function: Set after dc power to the machine has exceeded voltage tolerance limits for a period of 1.5 microseconds or greater.

PDP-11/24 CPU Error Register

The PDP-11/24 CPU error register is available only when the UNIBUS map is installed. The CPU error register contains one bit, bit 0. This bit when set indicates that one or more power supply voltages has exceeded its tolerance. This bit is set when a voltage error occurs and is cleared either by RESET or by writing a zero to the bit.

Bit: 0

Name: CPU Power Failure

Function: See explanation above.

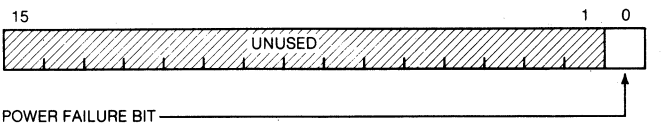


Figure E-8 ▪ 17 777 776 PDP-11/24 CPU Error Register

Processor Traps

Processor traps are a series of errors and programming conditions that will cause the central processor to trap through a set of fixed locations. These include power failure, odd addressing errors, stack errors, timeout errors, nonexistent memory references, parity errors, memory management violations, floating point processor exception traps, use of reserved instructions, use of the T bit in the processor status word, and use of the IOT, EMT, and TRAP instructions. PDP-11/24 processor traps are a subset of PDP-11/44 processor traps.

Power Failure

Whenever ac power drops below 90 volts for 120 V power (180 volts for 240 V) or outside a limit of 47 to 63 Hz, as measured by rectified dc power, the powerfail sequence is initiated. The central processor automatically traps through location 24 and the user's powerfail program has 2 (5 in the PDP-11/24) milliseconds to save all volatile information (data in registers).

If battery backup is present, and the batteries are not depleted when power is restored, the processor traps again through location 24 and executes the user's powerup routine to restore the machine to its state prior to power failure. If batteries are not present, a bootstrap of the default device is executed.

Odd Addressing Error

In the PDP-11/44, this odd addressing error occurs whenever a program attempts to execute a word instruction on an odd address (between word boundaries). The instruction is aborted and the CPU traps through location 4.

Timeout Error

In both the PDP-11/24 and the PDP-11/44, this timeout error occurs when a MSYN pulse is placed on the UNIBUS or main memory bus and there is no SSYN pulse within 20 microseconds. This error usually occurs during attempts to address nonexistent or peripherals or memory. The instruction is aborted and the processor traps through location 4.

Reserved Instruction

There is a set of illegal and reserved instructions that causes the processor to trap through location 10. If no floating-point processor is installed, execution of a floating-point instruction results in a trap through location 10.

Trap Handling

The *PDP-11 Architecture Handbook* includes a list of the reserved trap vector locations and system error definitions that cause processor traps. When a trap occurs, the processor follows the same procedure for traps as it does for interrupts (saving the PC and PSW on the new processor stack, for example).

Priority sequencing differs on PDP-11/44 and PDP-11/24 systems. In cases in which traps and interrupts occur concurrently, the processor will service the conditions according to the priority sequence following

PDP-11/44 Trap Priorities

1. HALT (instruction, switch, or command)
2. Memory management fault
3. Memory parity errors
4. Bus error traps
5. Floating-point traps
6. TRAP instruction
7. Trace trap
8. Stack overflow trap
9. Powerfail trap
10. Console bus request (console NEXT command or on-the-fly EXAMINE)
11. Program interrupt request (PIR) level 7
12. Bus request (BR) Level 7
13. PIR 6
14. BR 6
15. PIR 5
16. BR 5
17. PIR 4
18. BR 4
19. PIR 3
20. PIR 2
21. PIR 1
22. WAIT LOOP

PDP-11/24 Trap Priorities

1. DC LO (powerup)
2. Reserved instruction trap
3. Memory management fault
4. Bus error traps
5. Memory parity errors
6. Trace trap
7. Stack overflow trap
8. Powerfail trap
9. Bus request (BR) level 7
10. Line clock (highest B6 device)
11. BR level 6
12. BR level 5
13. BR level 4
14. HALT REQUEST
15. WAIT LOOP

Stack Limit Violations

In both the PDP-11/44 and the PDP-11/24, when instructions cause the kernel stack virtual address to go lower than 400 octal, a stack violation occurs. When the operation that caused the stack violation is completed, then a bus error trap is effected (trap through 4). The error trap, which itself uses the stack, executes without causing an additional violation.

PDP-11/44 Program Interrupt Requests (PIR)

Figure E-9 shows the layout of the program interrupt request register. A request is booked by setting one of bits 15–9 (for PIR 7–PIR 1) in the program interrupt register. The hardware sets Bits 7–5 and 3–1 to the encoded value of the highest PIR bit set. This program interrupt active (PIA) field should be used to set the processor level and also index through a table of interrupt vectors for the seven software priority levels.

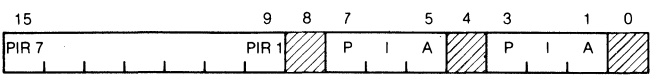


Figure E-9 ▪ 17 777 772 Program Interrupt Request Register

When the PIR is granted, the processor will trap through location 240 and pick up the PC from 240 and the PSW from 242. It is the interrupt service routine's responsibility to queue requests within a priority level and to clear the PIR bit before the interrupt is dismissed.

The following sample shows how the actual interrupt dispatch program should look.

```
MOVB PIR,PS          places bits 7–5 in PSW
                     priority level bits
MOV R5,-(SP)         save R5 on the stack
MOV PIR,R5
BIC #177761,R5       gets bits 3–1
JMP @DISPAT(R5)      use to index through table which
                     requires 15 core locations
```

CPU and I/O Device Registers and Addresses

The following, Table E-1, summarizes the PDP-11/44 registers and their addresses. Table E-2 summarizes the PDP-11/24 registers and their addresses. Note that commonalities exist.

Table E-1 ▪ PDP-11/44 CPU and I/O Device Registers and Addresses

Address	Register
17 777 776	Processor Status Word (PSW)
17 777 772	Program Interrupt Request (PIRQ)
17 777 766	CPU Error
17 777 707—17 777 700	CPU General Registers
17 777 676—17 777 660	User Data PAR, Reg. 0-7
17 777 656—17 777 640	User Instruction PAR, Reg. 0-7
17 777 636—17 777 620	User Data PDR, Reg. 0-7
17 777 616—17 777 600	User Instruction PDR, Reg. 0-7
17 777 576	MM Status Register 2 (SR2)
17 777 574	MM Status Register 1 (SR1)
17 777 572	MM Status Register 0 (SR0)
17 777 570	Switch Register
17 777 566—17 777 560	Console Terminal SLU
17 777 776—17 760 000 (switch-selectable)	SLU 2 DECtape (Normally 17 776 500)
17 777 516	MM Status Register 3 (SR3)
17 772 376—17 772 360	Kernel Data PAR, Reg. 0-7
17 772 356—17 772 340	Kernel Instruction PAR, Reg. 0-7
17 772 336—17 772 320	Kernel Data PDR, Reg. 0-7
17 772 316—17 772 300	Kernel Instruction PDR, Reg. 0-7
17 772 276—17 772 260	Supervisor Data PAR, Reg. 0-7
17 772 256—17 772 240	Supervisor Instruction PAR, Reg. 0-7
17 772 236—17 772 220	Supervisor Data PDR, Reg. 0-7
17 772 216—17 772 200	Supervisor Instruction PDR, Reg. 0-7
17 770 372—17 770 200	UNIBUS Map Registers

Table E-2 ▪ PDP-11/24 CPU and I/O Device Registers and Addresses

Address	Register
17 777 776	Processor Status Word (PSW)
17 777 766	CPU Error (Optional with UNIBUS map)
17 777 707—17 777 700	CPU General Register (not accessible by address)
17 777 656—17 777 640	User Instruction PAR, Reg. 0-7
17 777 616—17 777 600	User Instruction PDR, Reg. 0-7
17 777 576	MM Status Register 2 (SR2)
17 777 574	MM Status Register 1 (SR1)
17 777 572	MM Status Register 0 (SR0)
17 777 570	Display Register
17 777 566—17 777 560	Console Terminal SLU
17 776 500—17 776 506	SLU 2
17 772 516	MM Status Register 3 (SR3)
17 772 356—17 772 340	Kernel Instruction PAR, Reg. 0-7
17 772 316—17 772 300	Kernel Instruction PDR, Reg. 0-7
17 770 372—17 770 200	UNIBUS Map Registers (optional with UNIBUS map)

▪ Memory Systems

MOS Memory With Error Correcting Code and Optional Battery Backup
Both PDP-11/44 and PDP-11/24 employ MOS memory with error correcting code (ECC) identical to the PDP-11/84 system. Refer to system technical manuals for details.

Memory Management
The memory management hardware is standard with the PDP-11/44 and the PDP-11/24 computer. It is a hardware relocation and protection facility that can convert the 16-bit program virtual addresses to 22-bit physical addresses. The unit may be enabled or disabled under program control. There is a small speed advantage when the unit is in the 16-bit mode. For a more detailed description of memory management techniques, refer to the *PDP-11 Architecture Handbook*.

UNIBUS Map
The UNIBUS map is the hardware relocation facility for converting the 18-bit UNIBUS addresses to 22-bit addresses. The relocation mapping may be enabled or disabled under program control. Once again, there is a slight speed advantage when the UNIBUS map is disabled (off).

▪ PDP-11/44 Cache Memory

PDP-11/44 cache memory is integral to the PDP-11/44 processor and is designed to increase the CPU performance by decreasing the CPU-to-memory read access time. It is an 8192-byte, high-speed RAM memory, organized as a direct-mapped cache with write-through. Functionally, main memory and cache can be treated as a single unit.

Physical Description
The PDP-11/44 cache memory interfaces to the processor through the processor backplane. Two user-accessible switches (S1 and S2) enable the cache to be shut off by causing a forced-miss condition in either upper or lower cache address space. Software bits for enabling or disabling cache are also provided in the processor's cache-control registers discussed later in this chapter.

General System Architecture

The cache operates as an associative memory in parallel with the main memory, and is connected to the CPU by the high-speed internal data path in the PDP-11/44 (the PAX Data Bus). This high-speed data path is separate from the internal data path that is shared by the floating-point and commercial instruction set options (the AMUX data bus). The cache is logically connected to the PAX address and memory address buses, but is isolated from them by a set of independent receivers. When a memory read transfer is initiated by the CPU, the cache is strobed 100 nanoseconds later to determine if the data is in the cache and is error free. If so, this is referred to as a cache hit. If the access results in a cache hit, the processor clock is immediately restarted and clocks in the cache data that ends the transfer from the CPU. If access results in a cache miss, then main memory MSYN is asserted and the access is to main memory with the cache performing an automatic write-through to update itself. During write transfers, a write is performed to main memory with the cache updating itself if that location is presently cached. DMA write transfers from the UNIBUS are monitored by the cache and result in invalidation of cached locations. Only CPU transfers which access main memory are cached. Any data stored in memory appearing on the UNIBUS will not be cached.

CPU Bypass of the Cache

Besides having the capability of disabling half the cache, or the entire cache, the CPU can also disable caching of data based on the virtual address (virtual page) of the data. This is useful in two circumstances.

- If a multiported main memory (not supplied by Digital) is shared among two or more processors, it is possible for a particular word of main memory data to be cache in all of the CPUs. If one CPU then alters this word, only the copy of the data in main memory and that particular CPU's cache is updated. The other CPUs still have the old data in their caches. The old data is referred to as stale. In order to avoid using stale data, each CPU that accesses shared data must do so bypassing its cache. This ensures that the CPU gets the copy of the data stored in main memory, which is fresh (currently valid).

- Bypassing the PDP-11/44 cache is also useful if the CPU is sweeping through a large amount of data, with no intent of soon rereading the data. If the CPU is caching all the data, the cache will simply become full of useless data, meanwhile forcing out the program and other useful data. This is particularly true if the amount of data the CPU sweeps through exceeds 8192 bytes. This technique of bypassing the cache while accessing large data lists does not apply to the PDP-11/70 or PDP-11/84.

Table E-3 ▪ Cache Response Matrix

	CPU Hit	Miss	DMA Hit	Miss
Read Bypass	Nothing or Invalidate	Nothing	Nothing	Nothing
Write Bypass	Invalidate	Invalidate	Nothing	Nothing
Write	Update	Nothing	Invalidate	Nothing

The response of the cache to a CPU read bypass hit is jumper selectable. In its normal configuration, jumper W1 (M7097 module) is inserted and jumper W2 is removed to allow a forced miss to occur only for a CPU read hit bypass. If the PDP-11/44 and the KK11-B cache are to be used in a multiported memory system, jumper W1 is removed and jumper W2 is inserted to allow a CPU read hit with bypass to cause an invalidation to occur to that location. This allows the software to clear potentially stale cache data that might arise in a multiported memory system.

Cache Memory Organization

The cache memory array (Figure E-10) consists of thirty 4096 X 1 RAM chips arranged as follows.

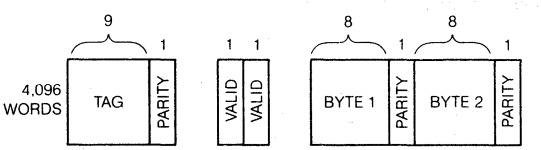


Figure E-10 ▪ Cache Memory Array

TAG	Consists of nine tag store bits plus one bit of parity.
VALID	Consists of two bits, one of which is currently active, allowing the other bit to be cleared concurrently. By having two bits, a fast flush may be accomplished by switching to the set which has been previously cleared.
DATA	Consists of two 8-bit bytes plus a parity bit for each byte.

Cache-Control Registers

The following cache-control registers are implemented on the PDP-11/44 cache. All bits are cleared by processor INIT, but not by a CPU RESET instruction.

- *Cache Data Register (CDR)*

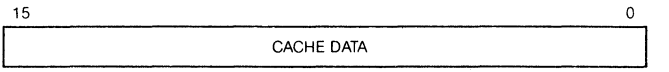


Figure E-11 ▪ 17 777 754 Cache Data Register

Bits: 15–0 (Read Only)

Name: Cache Data Register Bits

Function: These bits are loaded from the 16-bit data array section of the cache RAM on every read access to main memory space, except the top 256 Kbytes, which are reserved for the UNIBUS address space. This register can be used with the hit on destination only bit to aid the cache diagnostics in identifying failures in the data section of the cache array.

- *Cache Memory Error Register (CMER)*

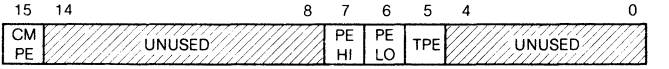


Figure E-12 ▪ 17 777 744 Cache Memory Error Register

Bit: 15

Name: Cache Memory Parity Error (CMPE)

Function: Set if a cache parity error is detected while the cache parity abort, bit 7, is set, or if a memory parity error occurs. If set, cache will force a miss. Clear by any write to the CMER or by console INIT. This bit must be cleared before the disable cache parity interrupt (DCPI) is cleared. If the cache detects a parity error in itself, the cache error LED (mounted on the cache module) will light.

Bit: 7

Name: Parity Error High Byte (PEHI)

Bit: 6

Name: Parity Error Low Byte (PELO)

Bit: 5

Name: Tag Parity Error (TPE)

Function: These bits are set individually when a parity error occurs in the high-data byte, low-data byte, or tag field, respectively, if the cycle is aborted (cache parity abort bit is set). If the cycle is not aborted, all three bits, 5, 6 and 7, are set upon any cache parity error occurrence as an aid to system software compatibility. Cleared by any write to the CMPE register or by console INIT.

▪ *Cache Control Register (CCR)*

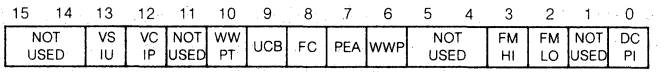


Figure E-13 ▪ 17 777 746 Cache Control Register

Bit: 13 (Read Only)

Name: Valid Store in Use (VSIU)

Function: This bit indicates which set of valid store bits is currently being used to determine the validity of the contents of the tag store memory. It is complemented each time that the cache is flushed. When set, valid bit set B is in use. When clear, valid bit set A is in use.

Bit: 12 (Read Only)

Name: Valid Clear in Progress (VCIP)

Function: This is set to indicate that the cache is currently in the process of clearing a valid store set. The clear cycle occurs on powerup and when the flush cache bit is set.

NOTE

The hardware clear cycle takes approximately 800 microseconds. While a valid store set is being cleared, the other set is in use allowing the cache to continue functioning. If the cache is flushed a second time within 800 microseconds, then the CPU will pause until the first flush completes (800 microseconds from the time the first flush command was issued).

Bit: 10 (Read/Write)

Name: Write Wrong Tag Parity (WWTP)

Function: This bit when set causes tag parity bits to be written with wrong parity on CPU read misses and write hits. A parity error will thus occur on the next access to that location.

Bit: 9 (Read/Write)

Name: Unconditional Cache Bypass (UCB)

Function: When this bit is set, all references to memory by the CPU will be forced to go to main memory. Read or write hits will result in invalidation of those locations in the cache and misses will not change the contents.

Bit: 8 (Write Only)

Name: Flush Cache (FC)

Function: This bit will always read as zero. Writing a one into it will cause the entire contents of the cache to be declared invalid. Writing a zero into this bit will have no effect.

Bit: 7 (Read/Write)

Name: Parity Error Abort (PEA)

Function: This bit controls the response of the cache to a parity error. When set, a cache parity error will cause a forced miss and an abort to occur (asserts UNIBUS signal PB L). When cleared, this bit inhibits the abort and enables an interrupt through location 114. All cache parity errors result in forced misses.

Bit: 6 (Read/Write)

Name: Write Wrong Data Parity (WWDP)

Function: This bit when set causes high and low parity bytes to be written with wrong parity on all update cycles (CPU read misses and write hits). This will cause a cache parity error to occur on the next access to that location.

Bit: 3 (Read/Write)

Name: Force Miss High (FMHI)

Function: This bit when set causes forced misses to occur on CPU reads of addresses where physical address bit 12 is a one. This bit can also be set by moving the toggle switch S1 to the right side of the board. The bit cannot be cleared via the toggle switch.

Bit: 2 (Read/Write)

Name: Force Miss Low (FMLO)

Function: This bit when set causes forced misses to occur on CPU reads of addresses where physical address bit 12 is a zero. This bit can also be set by moving the toggle switch S2 to the right side of the board. The bit cannot be cleared via the toggle switch.

NOTE

Setting bits 3 and 2 will cause all CPU reads to be misses.

Bit: 0 (Read/Write)

Name: Disable Cache Parity Interrupt (DCPI)

Function: This bit when set overrides the cleared condition of the parity error abort bit, disabling the interrupt through location 114. The cache memory parity error bit must be cleared before disable cache parity interrupt (DCPI) is cleared.

Bit 7	Bit 0	Result of Cache Parity Error
0	0	Interrupt to 114 and force miss
0	1	Force miss only
1	X	Abort and force miss

▪ *Cache Maintenance Register (CMR)*

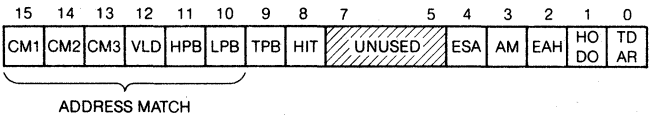


Figure E-14 ▪ 17 777 750 Cache Maintenance Register

Bits: 15–10 (Write Only)

Name: Address Match Bits 21–16

Function: This register is used to set bits 21–16 of the address-match register. The contents of the address-match register are constantly compared to the CPU physical address bus (PAX address). When an address-match occurs, the processor can 1) stop its microprogram; 2) halt; 3) supply an oscilloscope trigger pulse. This feature is useful for troubleshooting the PDP-11/44 system and is used with the console BREAK command.

Bit: 15

Name: Compare 1 H

Bit: 14

Name: Compare 2 H

Bit: 13

Name: Compare 3 H

Bit: 12

Name: Valid H

Bit: 11

Name: High parity bit H

Bit: 10

Name: Low parity bit H

Bit: 9

Name: Tag parity bit H

Bit: 8

Name: Hit L

Function: These bits are key points in the cache that the diagnostic can use to help localize errors. This register is loaded on any read to main memory. Like the cache data register, these bits can be used with the hit on destination only bit to aid the cache diagnostic in tracing cache failures.

Bit: 4

Name: Enable Stop Action

Function: This bit can be set to allow the cache to stop the CPU clock upon detection of a cache parity error or address match condition. This stops the CPU microprogram.

Bit: 3 (Read/Write)

Name: Address Matched (AM)

Function: This bit is set when the 22-bit address match register is equal to the 22-bit cache address. The address-match LED (on the cache module) also lights.

Bit: 2

Name: Enable Halt Action

Function: This bit can be set to allow the cache to halt the CPU upon detection of a cache parity error or address match condition.

Bit: 1 (Read/Write)

Name: Hit on Destination Only (HODO)

Function: When set, this bit causes the cache to be enabled only during the final memory access of an instruction. Read hits and updates will happen only during the final access. This feature is a very powerful tool for cache diagnostics. When cleared, this bit has no effect on the cache. This bit should be used with caution because it can cause stale data in the cache.

Bit: 0 (Read/Write)

Name: Tag Data from Address Match Register (TDAR)

Function: When set, this bit enables the tag field of the cache to be written with data from bits 8–0 of the address match register. Once this bit is set, it will cause all cache writes to clear the valid bit in these locations. This feature allows the cache diagnostics to identify failures in the tag field of the cache array.

▪ *Cache Hit Register (CHR)*

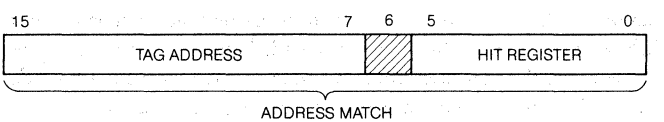


Figure E-15 ▪ 17 777 752 Cache Hit Register (CHR)

Bits: 15–0 (Write Only)

Name: Address Match Bits

Function: This register is used to set bits 15–0 of the address match register. It is used in conjunction with bits 15–10 of the cache maintenance register.

Bits: 15–7 (Read Only)

Name: Tag Address Bits

Function: Tag address bits contain the nine bits of the tag store memory of the last access by the CPU to main memory (except the top 256 Kbytes). When used with the hit on destination only and tag data from address match register bits, this field will allow the cache diagnostics to read any tag field of any location in the array.

Bits: 5–0 (Read Only)

Name: Hit Register

Function: This six-bit field shows the number of cache hits (read and write hits) on the last six CPU accesses to noncache-control memory. The bits flow from the least significant bit to the most significant bit of the field with a one indicating a hit and a zero indicating a miss.

▪ Other PDP-11/44 and PDP-11/24 Processor Equipment

Floating-Point Processor (FP11-A)

The PDP-11/44 floating point processor module fits integrally into the central processor. It provides a supplemental instruction set for performing single- and double-precision floating-point arithmetic operations and floating-integer conversion in parallel with the CPU. The floating-point processor provides both speed and accuracy in arithmetic computations. It provides 7 decimal digit accuracy in single-precision calculations and 17 decimal digit accuracy in double-precision calculations. For a detailed discussion on the PDP-11 floating-point processors, refer to the *PDP-11 Architecture Handbook*.

PDP-11/44 Backplane

Figure E-16 illustrates the PDP-11/44 CA Backplane. In this diagram, the standard and optional hardware features are seen in their corresponding slots in the backplane.

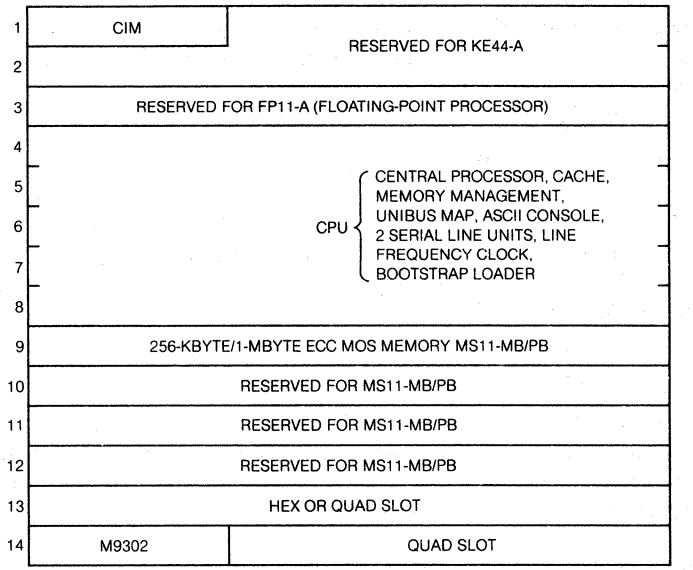


Figure E-16 ▪ PDP-11/44 Backplane Configuration

PDP-11/24 Backplane Configuration

The PDP-11/24 backplane consists of nine slots. Slot 1 is reserved for the M7133 CPU module. Slot 2 can contain memory or the UNIBUS map module. Additional memory can be configured in slots D-6. (In the 5.25-inch box, the total number of MS11-L memory modules cannot exceed three; only one MS11-P memory module can be configured in the 5.25-inch box). The UNIBUS map is generally required for configurations with more than 256 Kbytes of memory. Slot 9 contains either the M9312 bootstrap/terminator, the M9302 terminator (if the UNIBUS map option is installed), or the UNIBUS expansion cable.

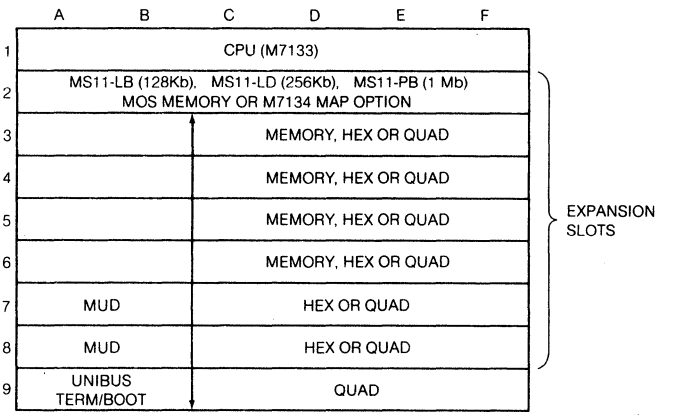


Figure E-17 ▪ PDP-11/24 Backplane Configuration

▪ Serial Line Unit Registers

Both the PDP-11/44 and PDP-11/24 contain two serial line interfaces as a standard feature. The first interface (the console interface) is used to control the PDP-11/44 hardware and the operating system software.

While controlling the PDP-11/44 hardware, the console interface is said to be in console mode and is operated by a fixed program in both systems. While controlling, and in turn being controlled by, operating system software, the console interface is said to be in program mode. The selection of console or program mode is made via the front panel and by special characters typed at the console terminal.

Serial Line Unit Timing Considerations

The UART (Universal Asynchronous Receiver/Transmitter) is an asynchronous subsystem. The transmitter accepts parallel characters and converts them to serial asynchronous output. The receiver accepts asynchronous serial characters and converts them to parallel output.

The serial line unit timing considerations for the PDP-11/24 are identical to those of the PDP-11/44.

- *Receiver*

  The RECEIVER DONE bit sets when the UART has assembled a full character, which occurs approximately at the middle of the first stop bit. Because the UART is double-buffered, data remains valid until the next character is received and assembled. This allows one full character time for servicing the RECEIVER DONE bit or the interrupt caused by it.

- *Transmitter*

  The UART's transmitter section is also double-buffered. After initialization, the TRANSMITTER READY bit is set. When the buffer is loaded with the first character, the bit clears but sets again within a fraction of a character transmission time period. A second character can then be loaded, clearing the bit again. This time the bit remains clear until the first character and its stop bit(s) have been transmitted (about one character time).

- *Break Generation*

  Setting the break bit causes the transmission of a continuous space. Because the TRANSMITTER READY bit continues to function normally, the duration of the break can be timed by the pseudo-transmission of a number of characters. However, because the transmitter is double-buffered, a null character (all zeros) should precede transmission of the break to ensure that the previous character completes transmission. Likewise, the last pseudo-transmitted character under break should be a null.

Terminal Serial Line Unit Control Registers (SLU 1)

There are four terminal SLU registers that follow. All unused or write only bits are zero when examined.

- *Receiver Control Status Register (TERM RCSR)*

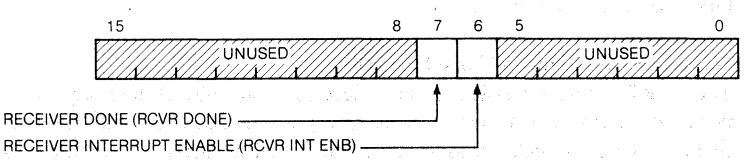


Figure E-18 ▪ 17 777 560 Receiver Control Status Register

Bits: 15–8

Function: Unused

Bit: 7 (Read Only)

Name: RECEIVER DONE

Function: Set during the program mode when an entire character has been received and is ready for transfer to the CPU. Cleared by INIT or by addressing (read only) RBUF. Starts an interrupt sequence when set if RECEIVER INTERRUPT ENABLE is also set.

Bit: 6 (Read/Write)

Name: RECEIVER INTERRUPT ENABLE

Function: Cleared by INIT. When set, a priority 4 interrupt sequence will start each time RECEIVER DONE is set.

Bits: 5–0

Function: Unused

- *Receiver Data Buffer (TERM RBUF)*

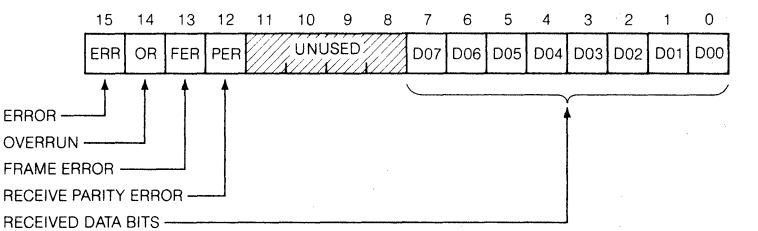


Figure E-19 • 17 777 562 Receiver Data Buffer

Bit: 15 (Read Only)

Name: ERROR

Function: Logical OR of OVERRUN ERROR, FRAMING ERROR and PARITY ERROR. ERROR is not tied to the interrupt logic, but RECEIVER DONE is.

Bit: 14 (Read Only)

Name: OVERRUN ERROR

Function: Set if the previously received character is not read (RECEIVER DONE not cleared) before another character is received.

Bit: 13 (Read Only)

Name: FRAMING ERROR

Function: Set if the character received has no valid stop bit(s). Also used to detect a break character.

Bit: 12 (Read Only)

Name: PARITY ERROR

Function: Set if received parity does not agree with the expected parity. Always cleared if no parity is selected.

NOTE

Error bits remain set until the next character is received, at which time the error bits are updated. INIT does not clear the console terminal error bits. However, in the PDP-11/44 a powerup sequence does clear them. Error bits may be disabled by removing a jumper on the M7096 module.

Bits: 11–8

Function: Unused

Bits: 7–0 (Read Only)

Name: RECEIVED DATA

Function: These bits contain the character just received. If fewer than eight bits are selected, the buffer will be right-justified with the unused bits read as 0. Not cleared by INIT.

- *Transmitter Control Status Register (TERM XCSR)*

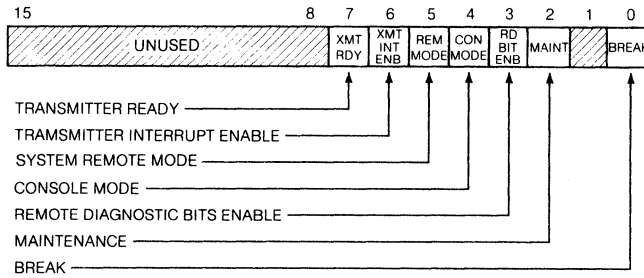


Figure E-20 • PDP-11/44 77 777 564 Transmitter Control Status Register

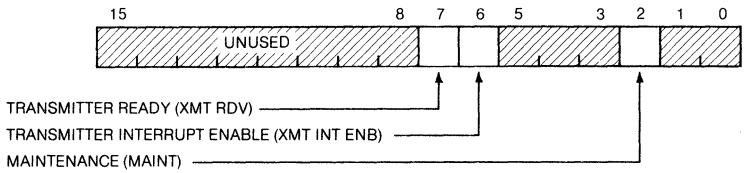


Figure E-21 • PDP-11/24 17 777 564 Transmitter Status Register

Bits: 15–8

Function: Unused

Bit: 7 (Read Only)

Name: TRANSMITTER READY

Function: Set during the program mode only by INIT or when XBUF can accept another character. Cleared when a character is written into the XBUF. Starts an interrupt sequence if TRANSMITTER INTERRUPT ENABLE is also set.

Bit: 6 (Read/Write)

Name: TRANSMITTER INTERRUPT ENABLE

Function: Cleared by INIT. When set, a priority 4 interrupt sequence will start each time TRANSMITTER READY is set.

Bit: 5 (Read Only)

Name: SYSTEM REMOTE MODE

Function: Set when CPU is operating in the remote diagnostic mode.

Bit: 4 (Read Only)

Name: CONSOLE MODE

Function: Set to indicate that the CPU is operating in the console mode.

Bit: 3 (Read Only)

Name: REMOTE DIAGNOSTIC BITS ENABLE

Function: Set by turning on switch #2 of E79 on the M7096 module. When set, the statuses of bits 4 and 5 are entered into this register. When cleared (switch off), all three bits will be zero.

Bit: 2 (Read/Write)

Name: MAINTENANCE

Function: Cleared by INIT. When set, it connects the serial output of the TRANSMITTER into the serial input of the RECEIVER, in place of the normal serial input from the terminal. It also forces the receiver to run at the same speed as the transmitter.

Bit: 1

Function: Unused

Bit: 0 (Read/Write)

Name: BREAK**Function:** Cleared by INIT. When set, a continuous space is transmitted, equivalent to sending a null character with no stop bits (framing error). May be disabled by removing a jumper on the M7096 module.

▪ *Transmitter Data Buffer (TERM XBUF) 17 777 566*

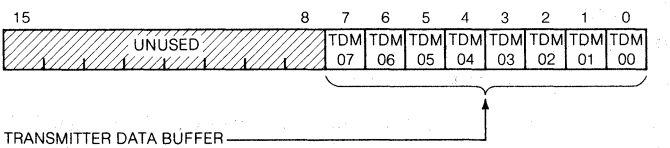


Figure E-22 ▪ 17 777 566 Transmitter Data Buffer

Bits: 15–8

Function: Unused

Bits: 7–0 (Write Only)

Name: TRANSMITTER DATA BUFFER

Function: If fewer than eight bits are jumper selected, the character must be right-justified.

Second Serial Line Unit Registers (SLU 2)

The second serial line unit is a general-purpose serial line interface. It may be used for a variety of purposes:

- Connection of a serial line-printer
- Connection of a TU58 cartridge tape drive
- Connection of a modem

This interface is not recomended for use with a high-speed (>1200 baud) interactive terminal, such as a VT220 or VT240/Z. The four SLU2 registers follow.

- *Receiver Control/Status Register (SLU 2 RCSR)*

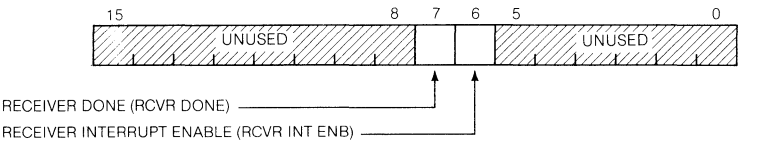


Figure E-23 ▪ Receiver Control/Status Register

Bits: 15–8

Function: Unused

Bit: 7 (Read Only)

Name: RECEIVER DONE

Function: Set when an entire character has been received and is ready for transfer to the CPU. Cleared by INIT or addressing (read-only) RBUF. Starts an interrupt sequence when set if RECEIVER INTERRUPT ENABLE is also set.

Bit: 6 (Read/Write)

Name: RECEIVER INTERRUPT ENABLE

Function: Cleared by INIT. When set, a priority 4 interrupt sequence will start each time RECEIVER DONE is set.

Bits: 5–0

Function: Unused

▪ *Receiver Data Buffer (SLU 2 RBUF)*

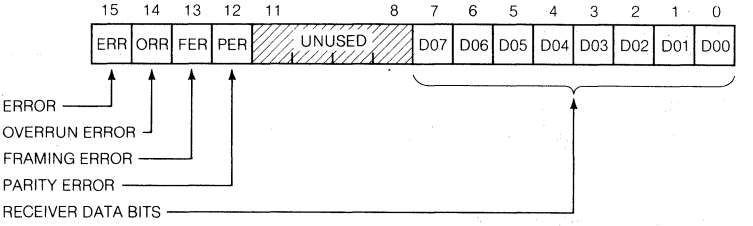


Figure E-24 ▪ Receiver Data Buffer

Bit: 15 (Read Only)

Name: ERROR

Function: Logical OR of OVERRUN ERROR, FRAMING ERROR and PARITY ERROR. ERROR is not tied to the interrupt logic, but RECEIVER DONE is cleared by INIT. Bits 12 through 15 may be disabled and cleared by removing a jumper on the M7096 module.

Bit: 14 (Read Only)

Name: OVERRUN ERROR

Function: Set if previously received character is not read (RECEIVER DONE not cleared) before another character is received. Cleared by INIT or reading before receiving another character.**Bit**: 13 (Read Only)

Name: FRAMING ERROR

Function: Set if character received has no valid stop bit(s). Cleared by INIT or when a valid character is received. This bit indicates an error in transmission or the reception of a break character.

Bit: 12 (Read Only)

Name: PARITY ERROR

Function: Set if received parity does not agree with expected parity. Cleared by INIT or when the parity of the next character is valid. Always cleared if no parity is selected.

Bits: 11–8

Function: Unused

Bits: 7–0 (Read Only)

Name: RECEIVED DATA

Function: These bits contain the character just received. If fewer than eight bits are selected, the buffer will be right-justified with the unused bits read as zeros. Not cleared by INIT.

- *Transmitter Control/Status Register (SLU 2 XCSR)*

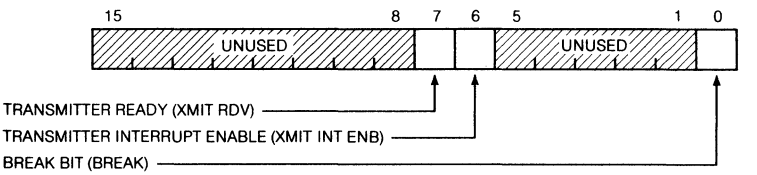


Figure E-25 ▪ Transmitter Control/Status Register

Bits: 15–8

Function: Unused

Bit: 7 (Read Only)

Name: TRANSMITTER READY

Function: Set by INIT or when the XBUF can accept another character. Starts an interrupt sequence when set if TRANSMITTER INTERRUPT ENABLE is also set. Cleared when a character is written into the XBUF.

Bit: 6 (Read/Write)

Name: TRANSMITTER INTERRUPT ENABLE

Function: Cleared by INIT. When set, a priority 4 interrupt sequence will start each time TRANSMITTER READY is set. Cleared by the program or by the initialization sequence.

Bits: 5–3

Function: Unused

Bit: 2 (Read/Write)

Name: MAINTENANCE

Function: Cleared by INIT. When set, it connects the serial output of the transmitter into the serial input of the receiver, in place of the normal serial input from the terminal. It also forces the receiver to run at the same speed as the transmitter.

Bit: 1

Function: Unused

Bit: 0 (Read/Write)

Name: BREAK

Function: Cleared by INIT. When set, a continuous space is transmittd equivalent to sending a null character with no stop bits (framing error). May be disabled by removing a jumper on the M7096 module.

▪ *Transmitter Data Buffer (SLU 2 XBUF)*

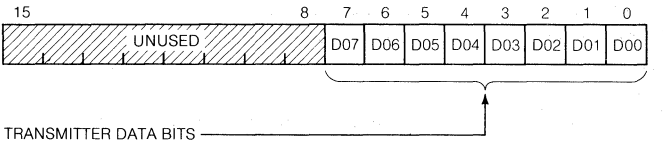


Figure E-26 ▪ Transmitter Data Buffer

Bits: 15–8

Function: Unused

Bits: 7–0 (Write Only)

Name: TRANSMITTER DATA

Function: If fewer than eight bits are selected, the character must be right-justified.

▪ Line Clock

Both the PDP-11/44 and PDP-11/24 include a line-time clock as standard equipment. This clock provides interrupts synchronized with the cycles of the ac power line (mains). By counting these interrupts, this allows the operation system software to keep realtime.

Line Clock Status Register

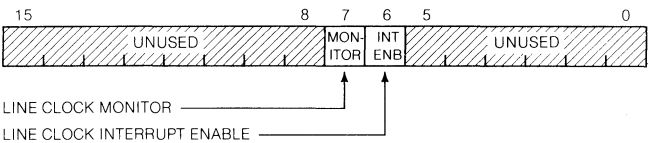


Figure E-27 ▪ 17 777 546 Line Clock Status Register

Bits: 15–8

Function: Unused

Bit: 7 (Read/Write, clear only)

Name: LINE CLOCK MONITOR

Function: Set by INIT or by the line-frequency clock signal (LTC). Cleared only by the program.

Bit: 6 (Read/Write)

Name: LINE CLOCK INTERRUPT ENABLE

Function: Cleared by INIT. When set, starts a priority 4 interrupt sequence each time LINE CLOCK MONITOR is set.

Bits: 5–0

Function: Unused

▪ Address and Vector Assignments

Integral to the PDP-11/44 and PDP-11/24 CPUs are the two serial line units and a the line-clock. The serial line units and clock follow the same address and vector assignments as the KL11, DL11-A, B, C, D, and W, and the KW11-L, respectively. SLU #1 is used for the system console and has fixed addresses and vectors. SLU #2, which may be used for asynchronous devices, has switch-selectable contiguous addresses and vectors. The realtime clock has a fixed address and vector.

Table E-4 ▪ PDP-11/44 Address and Vector Assignments

	Address	Vector	Priority
Console	17 777 560		BR4 (fixed)
(SLU #1)	17 777 562	60	
	17 777 564		
	17 777 566	64	BR4 (fixed)
(SLU #2)	17 7YX XX0		BR4 (fixed)
	17 7YX XX2	XX0	
	17 7YX XX4		
	17 7YX XX6	XX4	BR4 (fixed)
	Where Y=6 or 7 and X=0-7 (vector)		
Line Clock	17 777 546	100	BR6 (fixed)

Table E-5 ▪ PDP-11/24 Address and Vector Assignments

	Address	Vector	Priority
Console	17 777 560	60/64	BR4
(SLU #1)	17 777 562		
	17 777 564		
	17 777 566		
	17 776 500	300/304	BR4
(SLU #2)	17 776 502		
	17 776 504		
	17 776 506		
Line Clock	17 777 546	100	BR6

NOTE

Recommended address and vector assignments for SLU #2 when used for a TU58 are:

Address: 17 776 500

Vector: 300

These are the settings as received from Digital.

▪ PDP-11/44 Specifications

Packaging
A basic PDP-11/44 consists of a 10.5-inch box with a 14-slot backplane, power supply, CPU, and 1-Mbyte memory.

Component Parts
The basic PDP-11/44 system includes:

▪ Standard Equipment

- PDP-11/44 CPU
- Memory management
- Bootstrap loader
- Line-frequency clock
- Asynchronous console terminal interface
- Second asynchronous serial interface
- 8-Kbyte cache memory
- 1-Mbyte ECC MOS Memory
- BA11-A box with power supply

▪ Prewired Expansion Space for Optional Equipment

- Floating-point processor
- Commercial instruction set processor
- 2 SPC slots for peripherals (1 hex, 1 quad)
- 3-Mbyte ECC MOS memory (up to 4 Mbytes maximum)
- 3 SU open space in CPU box

Other PDP-11/44 Specifications

▪ *AC Power*
Power is 90-128 Vrms, 47-63 Hz, 1 phase power, 19 amp rms maximum @ 120 Vac or 180-256 Vrms, 47-63 Hz, 1 phase power, 9.5 amp rms maximum @ 240 Vac.

▪ *Size*
Each cabinet is 26.4 cm high by 42.2 cm wide by 66.0 cm deep (10.4-inches by 16.6-inches by 26.0 inches).

▪ *Weight*
The PDP-11/44 CPU box weights 40.5 kg (90lb).

▪ PDP-11/24 Specifications

Packaging

A basic PDP-11/24 consists of either a 5.25-inch or a 10.5-inch box with a 9-slot backplane, power supply, CPU, 128-Kbyte, 256-Kbyte, or 1-Mbyte memory. Prewired areas within the backplane provide for expansion with optional equipment.

Component Parts

The basic PDP-11/24 system contains:

- Standard Equipment PDP-11/24 CPU

  Memory management

  Bootstrap loader

  Line-frequency clock

  Second serial interface

  Terminal interface

  128-Kbyte, 256-Kbyte, or 1-Mbyte ECC MOS Memory

  BA11-l or BA11-A box with power supply

- Prewired Expansion Space for Optional Equipment Floating-point processor

  SPC slots for peripherals, up to 6 hex height and 1 quad height

  768-Kbyte parity MOS memory (up to 1024 Kbytes maximum)

  4 system units of open space in BA11-A Box

Other PDP-11/24 Specifications

- *AC Power*

  In the 5.25-in box, ac power is 104-127 Vrms, 47-63 Hz, 1 phase power, 5 amps rms maximum @ 120 Vac.

  In the 10.5-in box, ac power is 90-128 Vrms, 47-63 Hz, 1 phase power, 9 amps rms maximum @ 240 Vac.

- *Size*

  The 5.25-in box cabinet is 13.5 cm high by 48 cm wide by 69 cm deep (5.25-in by 19-in by 25-in).

  The 10.5-in box cabinet is 26.3 cm high by 42.4 cm wide by 66 cm deep (10.35-in by 16.62-in by 26-in).

▪ PDP-11/44 and PDP-11/24 Environmental Data

Operating Environment

Temperature:	5°C to 50°C (41°F to 122°F)
Humidity:	10% to 95% with max. wet bulb of 32°C (89.6°F) and minimum dew point of 2°C (36°F)
Altitude:	To 2.4 km (8,000 ft.) noncondensing

Nonoperating Environment

Temperature:	-40°C to 80°C (-14°F to 176°F)
Humidity:	To 95% noncondensing
Altitude:	To 9.1 km (30,000 ft.)

Index to PDP-11 Hardware Handbook

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

3601

26A

PDP-11 Systems Handbook

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

What is your general reaction to this handbook? (format, accuracy, completeness, organization, etc.)

What features are most useful?

Does the publication satisfy your needs?

What errors have you found?

Additional comments

Name

Title

Company Dept.

Address

City State Zip

(staple here)

(staple here)

786-3543

(please fold here)

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD, MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Corporate Communications Group
CFO 1-2/M92
200 Baker Avenue
West Concord, MA 01742

digital