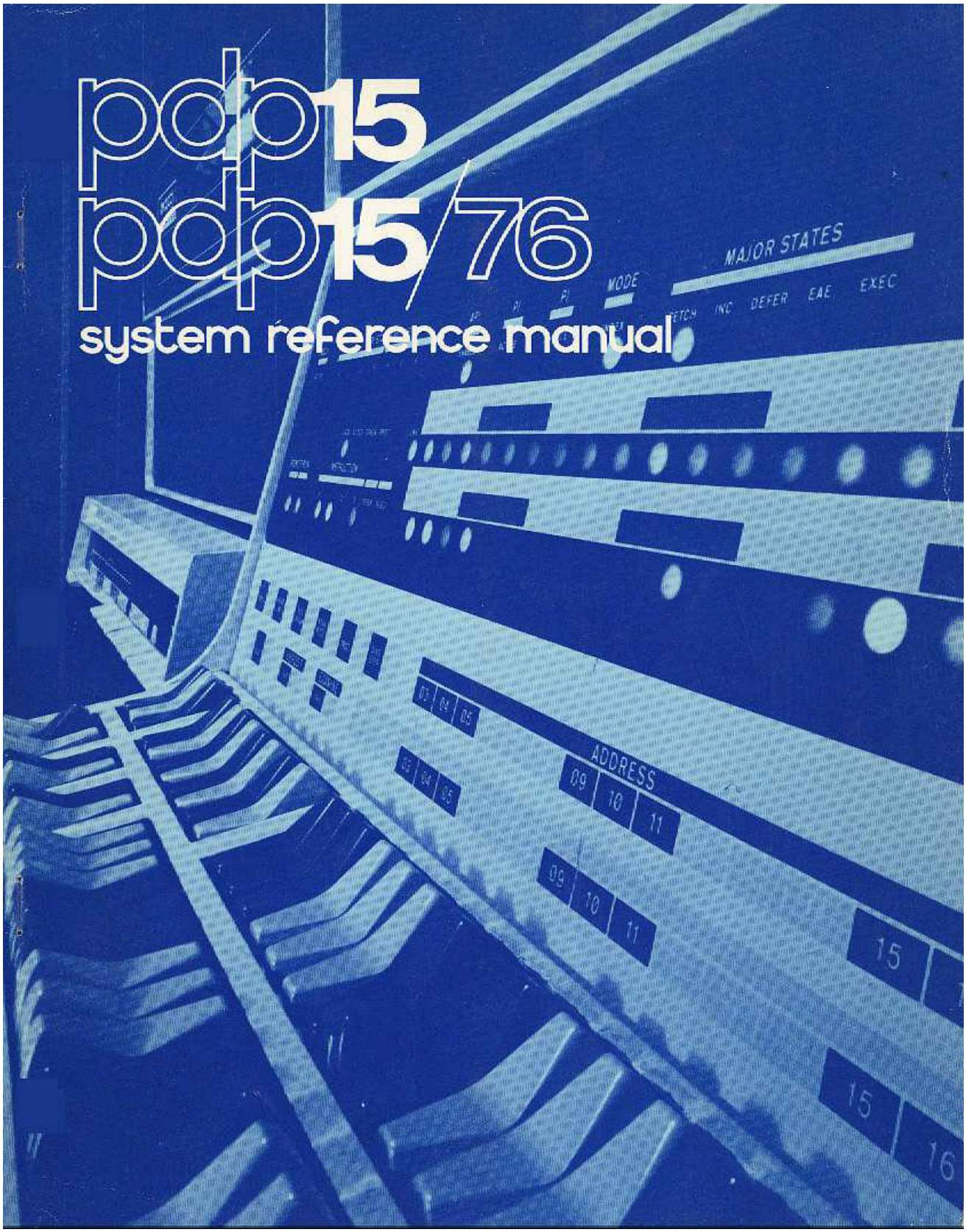
pdp15
pdp15/76

system reference manual



DEC-15-XSRMA-A-D

pdp15 and pdp15/76
SYSTEMS REFERENCE MANUAL

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

1st Edition, April 1969
2nd Printing (Rev), August 1969
3rd Printing (Rev), December 1970
4th Printing (Rev), April 1971
5th Printing, October 1971
6th Printing (Rev), October 1973

CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

ILLUSTRATIONS

ILLUSTRATIONS (Cont)

TABLES

FOREWORD

PDP-15 systems offer comprehensive solutions to data processing needs. They combine new design concepts with a wide array of traditional features that have evolved from Digital's years of leadership in the medium scale scientific computer field. Both elements share the common purpose of simplifying the application of PDP-15 computer systems to demanding environments.

PDP-15 systems excell in applications where hardware and software components are matched to meet the requirements of the user. These matches of hardware and software components can be viewed at two levels. First, PDP-15 systems come with a complete assortment of support software tools to allow complete programming. These tools, such as monitors, compilers, and system utility components, are the base on which all PDP-15 systems are built. Second, for certain specific turn key applications, PDP-15 systems come ready to do applications work with no further programming. At both levels, hardware components allow for modular additions and field upgrades without penalty.

Digital offers a variety of configurations as hardware building blocks to total system capability. These systems differ in their hardware options and peripherals that are required to support various operational software.

The hardware systems are designed with several functional objectives in mind. Among these are the complete autonomy between central processor, input/output processor, and memory, so that processing and I/O operations can occur concurrently in overlapping cycles: A PDP-11 programmable peripheral processor, so that slow speed devices can be spooled to a high performance disk; TTL integrated-circuit construction for high reliability; fast internal speeds, including an 800 ns to 960 ns memory cycle time to meet the demands of real time data processing; core memory expansions to 131,072 words for future growth; floating point hardware for demanding scientific applications; and a sophisticated memory protect system for multiuser integrity in multiprogramming environments. Peripheral device handling and interfacing to instruments are easily accomplished and system growth potential is virtually unlimited with the modular structure of the PDP-15 hardware and software systems.

INTRODUCTION

The PDP-15 18-bit computer has unique capabilities attractively suited to high-speed data acquisition and processing. Expandable core memory (to 128K words) and a full complement of processor options and peripherals enable the PDP-15 to handle virtually any medium scale computing requirement. Figure 1 is a simplified system block diagram.

This manual supplies background information to familiarize the reader with present and potential capabilities of the PDP-15 system.

THE PDP-15 SYSTEM

The basic PDP-15 system is organized into three autonomous subsystems: central processor, memory, and I/O processor, each with independent timing and control logic. Communication between these subsystems occurs through use of an effective asynchronous request scheme. Subsystem autonomy facilitates wide scale expansion of the system, increased throughput, high capacity, reliability, and maintainability. Four other major subsystems of the PDP-15 are: the floating point processor, the peripheral processor, the PDP-15 I/O bus peripherals, and the Unibus peripherals. Figure 2 is a more detailed block diagram of the PDP-15 system.

THE CENTRAL PROCESSOR (CPU)

The CPU controls and executes the system's stored programs. By virtue of its control autonomy, the CPU coordinates its operation with that of other subsystems, thus providing supervisory control over the PDP-15.

As the main unit in this integrated control, the central processor contains arithmetic and control logic hardware for a wide range of operations, including high-speed, fixed point arithmetic and hardware multiply and divide option, extensive test and branch operations implemented with special hardware registers, high-speed input/output instructions, and other arithmetic and control operations.

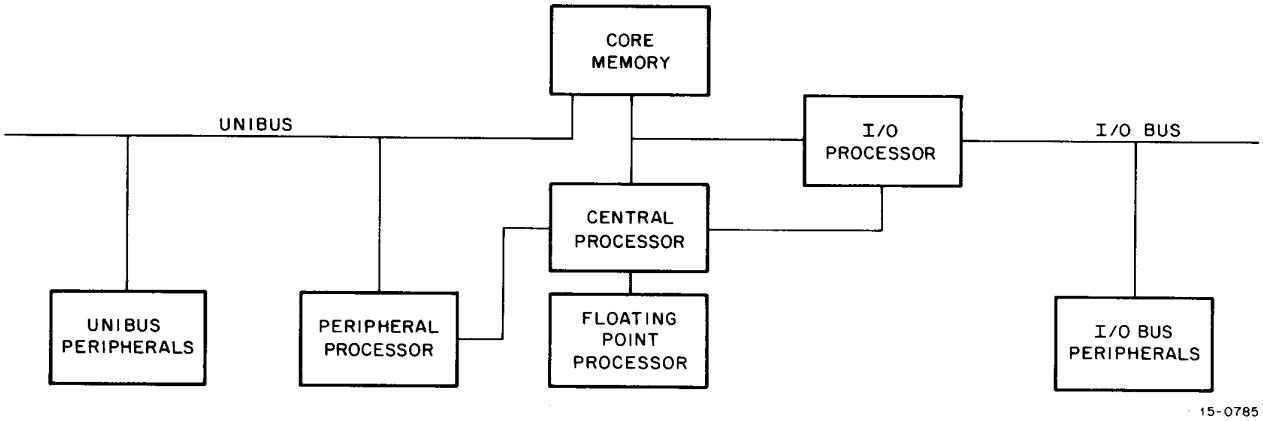


Figure 1 PDP-15, Simplified Block Diagram

The basic processor includes a number of major registers for processor-memory communications; it also includes a program counter, an accumulator, an Index register, and a Limit register. Two 18-bit registers provide memory buffer functions. This allows for processor overlap with memory cycle time and affects faster instruction execution times.

MEMORY

Memory is the primary storage area for computer instructions and system data. Independent read/write control and buffer logic in each memory bank establishes complete autonomy for the memory, i.e., different memories can be accessed simultaneously by different processors. The primary PDP-15 memory system is the ME15: a 980-ns memory system that permits installation and utilization of up to 96K core memory and 8K increments on the back panel of the central processor without requiring the MX15 Memory Bus Multiplexer. Also available is the MM15/MK15 Memory, which is an 800 ns system that is organized into pages which are paired into memory banks. Each page has 4096 18-bit binary words of high-speed random access, making it a core storage. Each bank is an asynchronous unit of 8192 words. To expand beyond 32,768 words, an MX15 Memory Bus Multiplexer must be used. The PDP-15 can address up to 131,768 words of either type core memory.

THE I/O PROCESSOR

The I/O processor satisfies the peripheral data transfer needs. A diverse line of system peripherals available to the PDP-15 require this processor to interface three modes of input/output:

- Single cycle block data transfer; blocks of data transfer at typical rates of up to 1 million words per second.
- Multicycle block data transfer; blocks of data transfer at rates up to 250,000 per second for input and a 181,000 per second for output.
- Program control data transfers; single word transfers to/from the accumulator in the central processor.

The I/O processor provides timing, control, and data lines for information transfers between memory or the central processor and the peripheral devices. It also includes provision for such options as the automatic priority interrupt system and the real-time clock.

FP15 Floating Point Processor

The FP15 enables the PDP-15 to perform arithmetic and logic operations using floating point arithmetic. The prime advantage is increased speed without the necessity of writing complex floating point software routines. The FP15 has single precision and extended integer capability, as well as single and double precision floating point.

The FP15 is an expansion of the central processing unit and increases the PDP-15 instruction set by over 120 instructions. Floating point instructions intermix with PDP-15 instructions. This in-line mode of operation greatly simplifies programming and ensures fast execution.

The Peripheral Processor

The Unichannel 15 (UC15) is peripheral processor for the PDP-15 that uses the PDP-11/05 minicomputer. It provides the PDP-15 with a second general purpose processor and a second high-speed I/O bus: The Unibus is an 18-bit pathway permitting transfer of 18-bit words, 16-bit PDP-11 words, or two 8-bit bytes.

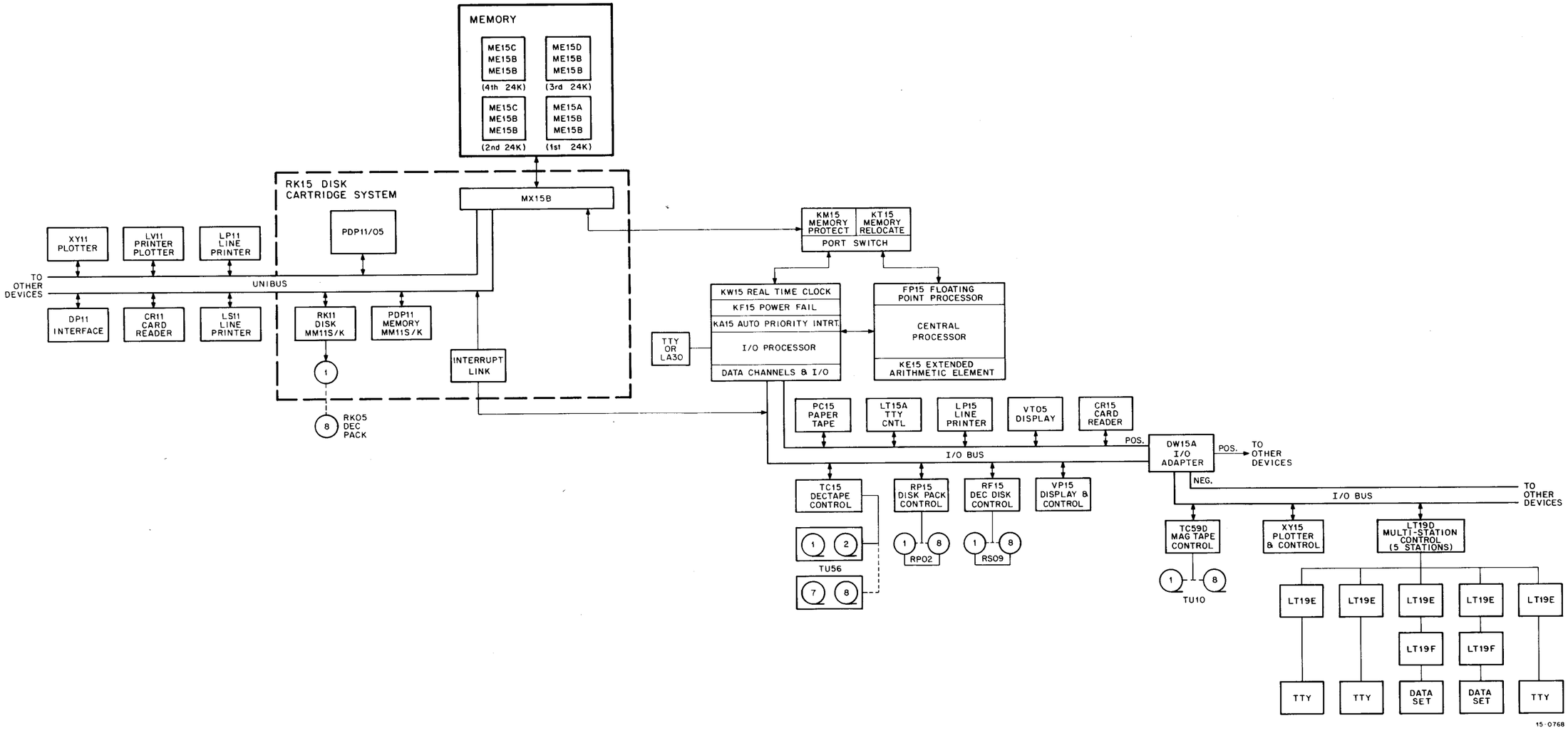


Figure 2 PDP15 System, Detailed Block Diagram

There are three major components for the UC15: 1) a PDP-11/05 Computer with 4K of PDP-11 memory; 2) an MX15-B Memory Bus Multiplexer, which allows both the PDP-15 processor and a PDP-11 processor to share common memory, which is an ordinary 18-bit PDP-15 core memory, and 3) an "interrupt link" to provide a second means of interprocessor communications.

PDP-15 I/O Bus Peripherals

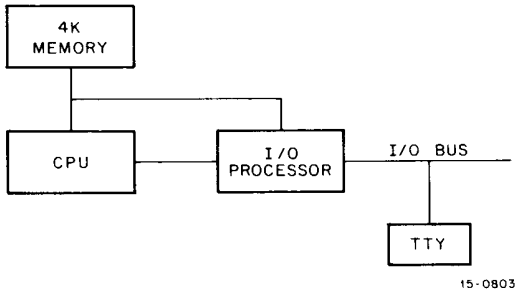
Peripheral equipment on the PDP-15 I/O bus includes Teletypes, PC15 High-Speed Paper Tape Reader and Punch, CR15 Card Readers, TC15 DECtapes, RK15 Cartridge Disk System with Unichannel 15, RF15 RS09 DECdisks, RP02 RP15 DECdisk Pack and Control, line printers, plotters, magtape systems, display devices, graphic systems, A/D and D/A equipment, data acquisition, and communications equipment.

Unibus Peripherals

Peripherals available on the Unibus include LP11 and LS11 Line Printers, LV11 Printer Plotter, CR11 Card Reader, XY11 Plotter, DP11 Asynchronous Interface, RK05/RK11E Disk Cartridge System.

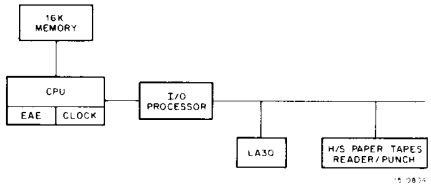
PDP-15 SYSTEM CONFIGURATIONS

The PDP-15 is available in 14 basic configurations as listed in the system configuration table. The simplest PDP-15 system available is the PDP-15/10 which is diagrammed below.



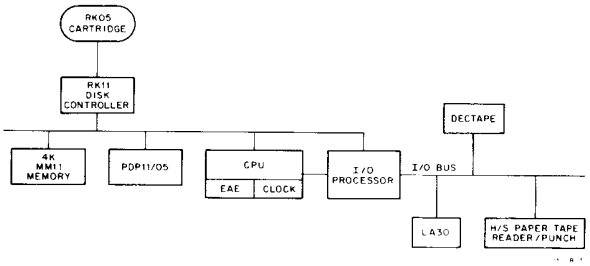
PDP-15/10

The PDP-15/73 is the next step in complexity as indicated below.



PDP-15/73

The PDP-15/76 adds a peripheral processor to the PDP-15. This peripheral processor called Unichannel 15 (UC15) utilizes the PDP-11/05 computer as indicated in the figure below.



PDP-15/76-DA

The full expandability of the PDP-15 is shown in Figure 2.

SYSTEM CONFIGURATIONS TABLE

PDP-15/73-A Computer System (115 V 60 Hz).
PDP-15/73-B Computer System (230 V 50 Hz).

- KP15 Central Processor
- 16,384 words ME15 Core Memory
- LA30 DECwriter
- PC15 High Speed Paper Tape Reader and Punch
- KE15 Extended Arithmetic Element
- KW15 Real Time Clock

PDP-15/75-A Computer System (115 V 60 Hz).
PDP-15/75-B Computer System (230 V 50 Hz).

- KP15 Central Processor
- 16,384 words ME15 Core Memory
- LA30 DECwriter
- PC15 High Speed Paper Tape Reader and Punch
- KE15 Extended Arithmetic Element
- KW15 Real Time Clock
- TC15 DECtape Control
- TU56 Dual DECtape Transport

PDP-15/76-DA Computer System (115 V 60 Hz).
PDP-15/76-DB Computer System (230 V 50 Hz).

- KP15 Central Processor
- 16,384 words ME15 Core Memory
- LA30 DECwriter
- PC15 H/S Paper Tape Reader and Punch
- KE15 Extended Arithmetic Element
- KW15 Real Time Clock
- TC15 DECtape Control
- TU56 Dual DECtape Transport
- RK15 Cartridge Disk System with Unichannel-15 peripheral processor and 4K MM11-K Core Memory

PDP-15/76-DC Computer System (115 V 60 Hz).
PDP-15/76-DD Computer System (230 V 50 Hz).

- KP15 Central Processor
- 16,384 words ME15 Core Memory
- LA30 DECwriter
- PC15 H/S Paper Tape Reader and Punch
- KE15 Extended Arithmetic Element
- KW15 Real Time Clock
- TC15 DECtape Control
- TU56 Dual DECtape Transport
- RK15 Cartridge Disk System with Unichannel-15 peripheral processor and 8K MM11-L Core Memory

PDP-15/76-MA Computer System (115 V 60 Hz).
PDP-15/76-MB Computer System (230 V 50 Hz).

- KP15 Central Processor
- 16,384 words ME15 Core Memory
- LA30 DECwriter
- PC15 H/S Paper Tape Reader and Punch
- KE15 Extended Arithmetic Element
- KW15 Real Time Clock
- TC59 Magnetic Tape Control
- TU10 Magnetic Tape Transport
- RK15 Cartridge Disk System with Unichannel-15 peripheral processor and 4K MM11-K Core Memory

PDP-15/76-MC Computer System (115 V 60 Hz).
PDP-15/76-MD Computer System (230 V 50 Hz).

- KP15 Central Processor
- 16,384 words ME15 Core Memory
- LA30 DECwriter
- PC15 H/S Paper Tape Reader and Punch
- KE15 Extended Arithmetic Element
- KW15 Real Time Clock
- TC59 Magnetic Tape Control
- TU10 Magnetic Tape Transport
- RK15 Cartridge Disk System with Unichannel-15 peripheral processor and 8K MM11-K Core Memory

PDP-15/77-A Computer System (115 V 60 Hz).
PDP-15/77-B Computer System (230 V 50 Hz).

- KP15 Central Processor
- 24,576 words ME15 Core Memory
- LA30 DECwriter
- PC15 High Speed Paper Tape Reader and Punch
- KE15 Extended Arithmetic Element
- KW15 Real Time Clock
- TC15 DECtape Control
- TU56 Dual DECtape Transport
- RF15 DECdisk Control
- RS09 DECdisk Drive, 262,144 words
- KM15 Memory Protect

- KT15 Memory Relocate
- KA15 Automatic Priority Interrupt
- LT15-A Single Teletype Control

PDP-15/79-A Computer System (115 V 60 Hz).
PDP-15/79-B Computer System (230 V 50 Hz).

- KP15 Central Processor
- 16,384 words ME15 Core Memory
- LA30 DECwriter
- PC15 High Speed Paper Tape Reader and Punch
- KE15 Extended Arithmetic Element
- KW15 Real Time Clock
- TC59 Magnetic Tape Control
- TU10 Magnetic Tape Transport
- FP15 Floating Point Processor
- RP15 Disk Pack Control
- RP02 Disk Pack

PDP-15/10 Computer System.

- KP15 Central Processor
- 4,096 words 18-bit, 800-ns Core Memory
- ASR-33 Teletype

PDP-15/20 Computer System.

- KP15 Central Processor
- 8,192 words 18-bit, 800-ns Core Memory
- KSR-35 Teletype
- PC15 High Speed Paper Tape Reader and Punch
- KE15 Extended Arithmetic Element
- TC15 DECtape Control
- TU56 Dual DECtape Transport

PDP-15/30 Computer System.

- KP15 Central Processor
- 16,384 18-bit, 800-ns Core Memory
- KSR-35 Teletype for BACKGROUND use
- KSR-33 Teletype for FOREGROUND use
- LT15-A Single Teletype Control
- PC15 High Speed Paper Tape Reader and Punch
- KE15 Extended Arithmetic Element
- KA15 Automatic Priority Interrupt
- KM15 Memory Protect
- KW15 Real Time Clock
- TC15 DECtape Control
- (2) TU56 Dual DECtape Transports

PDP-15/35 Computer System.

- KP15 Central Processor
- 16,384 words 18-bit, 800-ns Core Memory
- KSR-35 Teletype
- PC15 High Speed Paper Tape Reader and Punch
- KE15 Extended Arithmetic Element
- KA15 Automatic Priority Interrupt
- KW15 Real Time Clock
- TC15 DECtape Control
- TU56 Dual DECtape Transport
- RF15 DECdisk Control
- RS09 DECdisk Drive 262,144 words

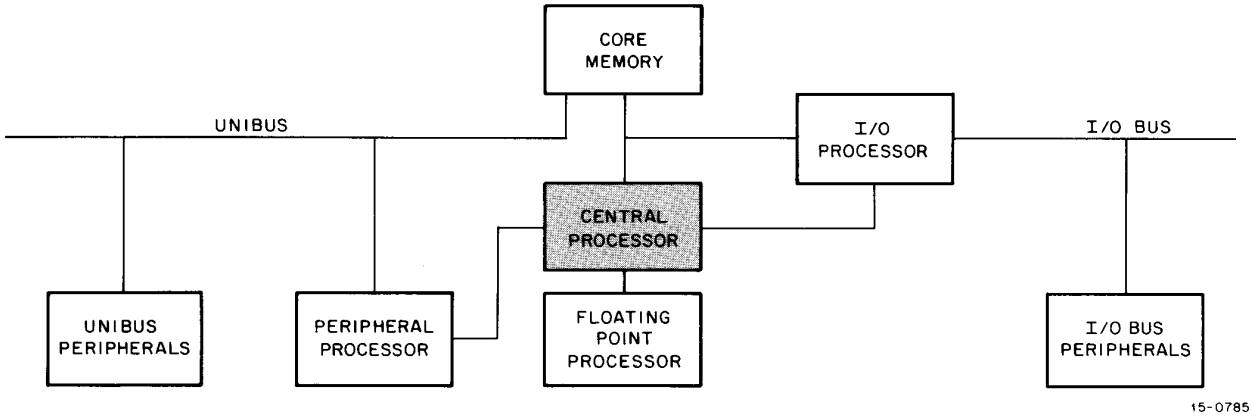
PDP-15/40 Computer System.

- KP15 Central Processor
- 24,576 word 18-bit, 800-ns Core Memory
- KSR-35 Teletype for BACKGROUND use
- KSR-33 Teletype for FOREGROUND use
- LT15-A Single Teletype Control
- PC15 High Speed Paper Tape Reader and Punch
- KE15 Extended Arithmetic Element
- KA15 Automatic Priority Interrupt
- KM15 Memory Protect
- KW15 Real Time Clock
- TC15 DECtape Control
- TU56 Dual DECtape Transport
- RF15 DECdisk Control
- (2) RS09 DECdisk Drives 262,144 words each

PDP-15/50 Computer System.

- KP15 Central Processor
- 16,384 word 18-bit, 800-ns Core Memory
- KSR-35 Teletype
- KE15 Extended Arithmetic Element
- KW15 Real Time Clock
- PC15 High Speed Paper Tape Reader and Punch
- FP15 Floating Point Processor
- RP15 Disk Pack Control
- RP02 Disk Pack Drive, 10.24 million words
- TC59 Magnetic Tape Control
- TU10 Magnetic Tape Transport

CHAPTER 1
CENTRAL PROCESSOR



SUMMARY OF CHARACTERISTICS

Description – 18-bit parallel operation, autonomous operation, fixed point signed and unsigned arithmetic (1's and 2's complement)

Instruction Types –

Memory Reference
Operates
Register Transfer and Control
Extended Arithmetic Element
Input/Output Transfer

Indexing – 1 Index register, 1 Limit register, 8 autoincrement locations

Timing –

Typical Instructions	Execute Time
18-bit TAD	1.6 μs
18-bit Multiply*	7.68 μs
18-bit Divide*	7.42 μs
36-bit Shift*	7.68 μs
36-bit Normalize*	7.68 μs

* With EAE

CENTRAL PROCESSOR DESCRIPTION

The Central Processor (CPU) controls and executes stored programs. By coordinating its

operation with that of other subsystems, it provides supervisory control over the PDP-15 system.

The CPU contains arithmetic and control logic hardware for a wide range of operations. These include high-speed, fixed point arithmetic with a hardware multiply and divide option, extensive test and branch operations implemented with special hardware registers, high-speed input/output instructions, and other arithmetic and control operations.

The PDP-15 CPU contains several major registers for processor-memory communications, a program counter, an Instruction register, and accumulator, an Index register, and a Limit register.

The CPU performs calculations and data processing in a parallel binary mode through step-by-step execution of individual instructions. Both the instructions and the data on which the instructions operate are stored in the core memory of the PDP-15. The arithmetic and logical operations necessary for the execution of all instructions are performed by the arithmetic unit operating in conjunction with central processor registers. Figure 1-1 shows a simplified block diagram of the CPU.

Arithmetic Unit (AU)

The PDP-15 arithmetic unit (AU) handles all Boolean functions and contains an 18-bit, 85-ns adder. The AU acts as the transfer path for inter-register transfers and shift operations.

Instruction Register (IR)

This register accepts the six most-significant bits of each instruction word fetched from memory. Of these bits, the four most-significant constitute the operation code, the fifth signals when the fetched instruction indicates indirect addressing, and the sixth indicates indexing.

Accumulator (AC)

This 18-bit register retains the result of arithmetic/logical operations for the interim between instructions.

For all program-controlled input/output transfers, information is transferred between core memory and an external device through the AC. The AC can be cleared and complemented. Its contents can be rotated right or left with the Link (see below). The contents of the memory, buffered through the Memory Input register, can be added to the contents of the AC with the result left in the AC. The contents of both registers can be combined by the logical operations AND and Exclusive-OR, the result remaining in the AC. The Inclusive-OR can be performed between the AC and the data switches on the operator console (through the Data Switch register) and the result left in the AC.

Data Switch Register

The Data Switch register receives and stores an 18-bit word through the console bus from data switches on the console. This allows programs to accept switch data from the operator console.

Link (L)

This 1-bit register is used to extend the arithmetic capability of the accumulator. In 1's complement arithmetic, the Link is an overflow indicator; in 2's complement arithmetic, it logically extends the accumulator to 19 bits and functions as a carry register. The program can check overflow into the Link to simplify and speed up single and multiprecision arithmetic routines.

The Link can be cleared and complemented and its state sensed independent of the accumulator. It is included with the accumulator in rotate operations and in logical shifts.

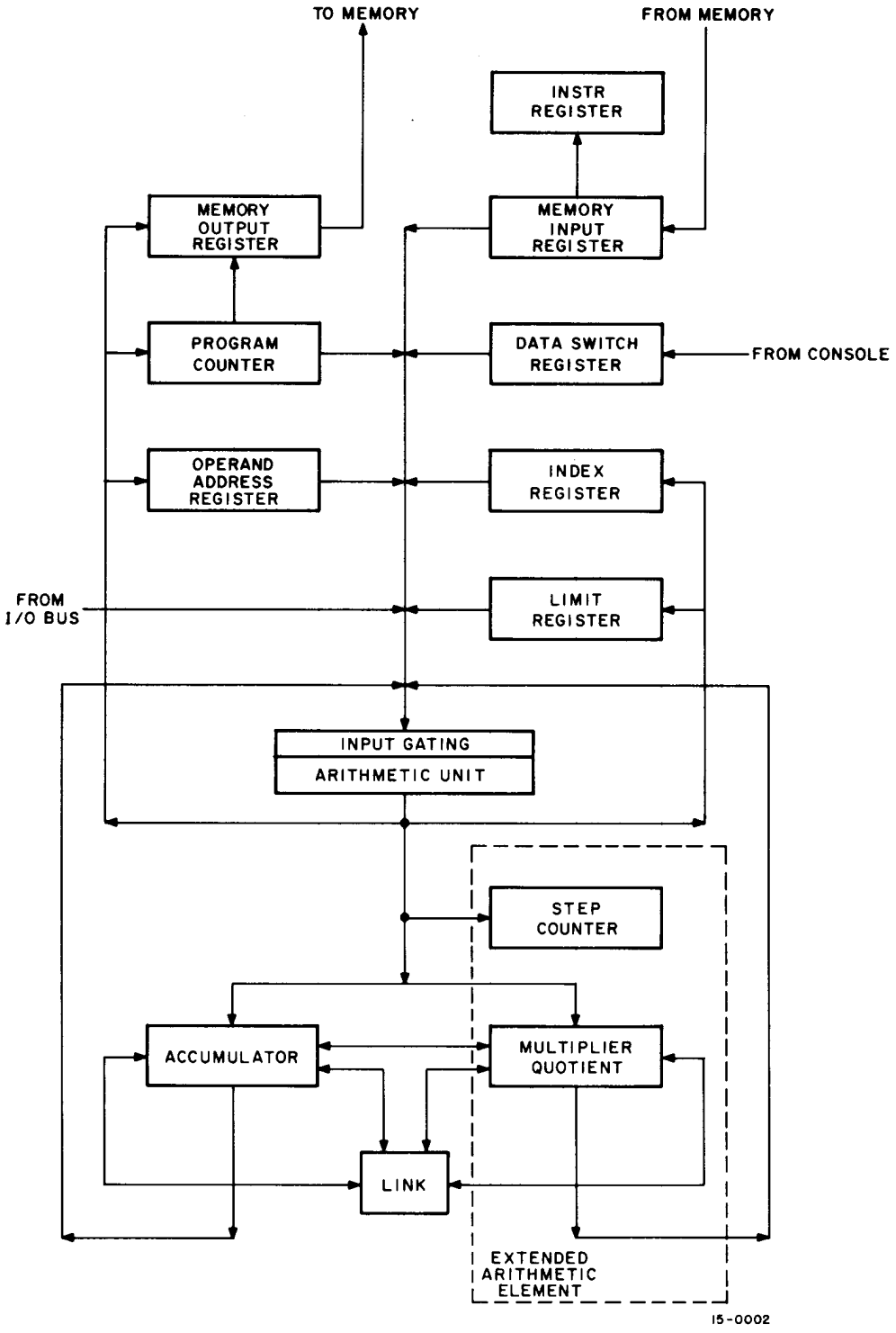


Figure 1-1 Central Processor, Simplified Block Diagram

Program Counter (PC)

The PC determines the program sequence (the order in which instructions are performed). This 18-bit register contains the address of the memory cell from which the next instruction is to be taken. The least-significant 15 bits are used for addressing 32,768 words of core memory. The remaining 3 bits provide the capability to address memory systems greater than 32,768 words.

Operand Address Register (OA)

The Operand Address register contains the effective address of the location where data is currently being fetched.

Memory Input and Output Buffer Register (MI and MO)

Information is read from a memory cell into the Memory Input register and is interpreted as either an instruction or a data word. Information is read from the CPU into memory through the Memory Output register, and is interpreted as either an address or a data word. The use of two 18-bit registers for memory buffer functions allows processor overlap with memory cycle time to decrease execution time and to allow autonomous operation of the CPU and memory. They also allow the I/O processor to gain access to memory between cycles of multicycle instructions. By not having to wait for completion of an instruction, I/O device latency is vastly improved.

Index Register (XR)

This 18-bit register is used to perform indexing operations with no increase in instruction time. An indexed operation adds the contents of the Index register to the address field of the instruction operand, producing an effective address for the data fetch cycle. Index value can be positive or negative in 2's complement form (±131,072). The Index register can be used with the Limit register, described in the following paragraph.

Limit Register (LR)

The 18-bit Limit register enables a program to detect loop completion. The base address of a data array is loaded into the Index register and the ending address is loaded into the Limit register. Within an indexing loop, add to index and skip (AXS) instruction adds a signed value ($-256_{10} \leqslant Y \leqslant +255_{10}$) to the Index register and compares the sum in the index to the contents of the Limit register. If the contents of the Index register are equal to or greater than those of the Limit register, the next instruction is skipped.

PDP-15 Control Console

The PDP-15's control console contains the keys, switches, and lights required for operator initiation, control, and monitoring of the system. Up to twenty-four 18-bit registers can be displayed to provide the user with visual indication of major registers and buses.

The console can be ordered in two different forms: a flush-mounted console which can be covered by cabinet doors for applications where "hands-off" security is paramount, or a tilted console with table.

Some of the features of the console are:

- A READ-IN switch to initiate the hardware readin of paper tapes; REGISTER indicators and REGISTER DISPLAY switches to allow continual monitoring of key points in the system such as the accumulator, Index register, Limit register, Multiplier-Quotient register, program counter, memory address, interrupt status, input/output bus, input/output address, and I/O status.

- Separate ADDRESS and DATA switches to establish an 18-bit data or instruction word to be read into a memory location by the DEPOSIT switch, to be entered into the accumulator by a program instruction, or to be executed by the EXECUTE switch.

- EXAMINE switch to allow the manual examination of the contents of any memory location placed in the ADDRESS switches.

PROCESSOR EXPANSION

The following additional expansions extend processing capabilities of PDP-15.

Extended Arithmetic Element (EAE)

The Extended Arithmetic Element facilitates high-speed arithmetic operations and register manipulations. Installation of the EAE adds an 18-bit Multiplier-Quotient register (MQ) to the system as well as a 6-bit Step Counter register (SC). The Multiplier-Quotient register and accumulator perform as a 36-bit register during shifting, normalizing, and multiplication operations. The contents of the Multiplier-Quotient register are displayed by the REGISTER indicators on the operator's console when the REGISTER DISPLAY control is in the MQ position. The option and the basic computer cycle operate asynchronously, permitting computations to be performed in the least possible time. Moreover, EAE instructions are microcoded so that several operations can be performed by one instruction to simplify arithmetic programming and reduce execution time. Worst case multiplication time is 7.68 μs; division time is 7.42 μs.

KM15 Memory Protect

The KM15 Memory Protect Option provides the PDP-15 core memory with protected memory locations that cannot be accessed by the user. It includes a Boundary register and associated control logic to establish the lower limit of the user's program. It has the facility to trap IOT, HALT OAS, and chained execute instructions and the addressing of a nonexistent memory bank.

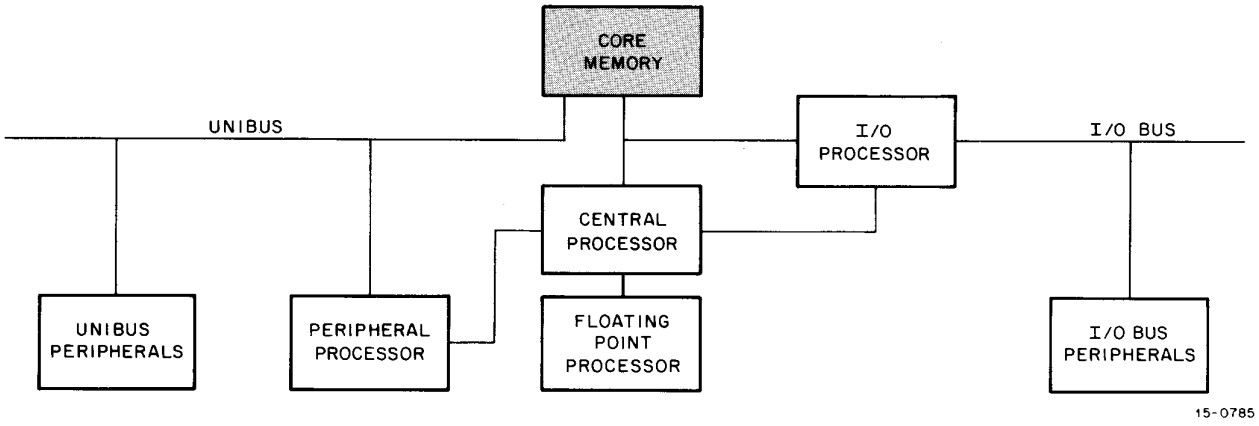
KT15 Memory Protect and Relocate

The memory protect and relocate option is similar to the memory protect option. The KM15 must be used with the KT15. However, it contains a Relocation register as well as a Core Allocation register. The Relocation register provides the lower limit of the user program and relocates the user upward from the real machine location by the quantity contained in the Relocation register. The Core Allocation register indicates the last 256 word increment available to the user. Other features of the memory protect option are also included in this option.

KF15 Power Failure Protection

The basic PDP-15 is not affected by power interruptions of less than 10-ms duration. Active registers in the processor may lose their contents when interrupts of longer duration occur, but memory is not disturbed. The KF15 Power Failure Protection option, available for all PDP-15 systems, provides for saving the active register contents in the event of longer power interrupts and the automatic restart of the system when power is restored. When the line power failure occurs, the system must be operating with the program interrupt facility or the automatic priority interrupt system enabled in order to sense the power failure protection's initiation of a program interrupt in time to save the register contents. When power is restored, the instruction in location 0 is executed.

CHAPTER 2
MEMORY



INTRODUCTION

The magnetic core memory is the primary storage facility of the PDP-15. It provides rapid, random access data instruction storage for both the CPU and the I/O processor.

Memory Data Transfer

The PDP-15 memory communicates directly with the CPU and the I/O processor through the memory bus. Data and instruction words of each blank are read from and written into individual memory cells through a buffered register referred to as the memory data buffer.

Words in a memory bank are selected according to the address in the memory address buffer. The 13-bit capacity of the memory address buffer allows 2^{13} or 8192 words to be referenced in each bank.

The memory address buffer receives the memory cell address from the CPU or I/O processor. The address provides the coordinates for locating a word in a memory bank.

The ME15 Core Memory provides the PDP-15 with a compact, economical storage device capable of storing up to 96K, 18-bit words. The ME15 is mounted on the rear door of the central processor cabinet (Figure 2-1) and may be added to those PDP-15 installations already equipped with MM15-type memory; the MX15 is not required.

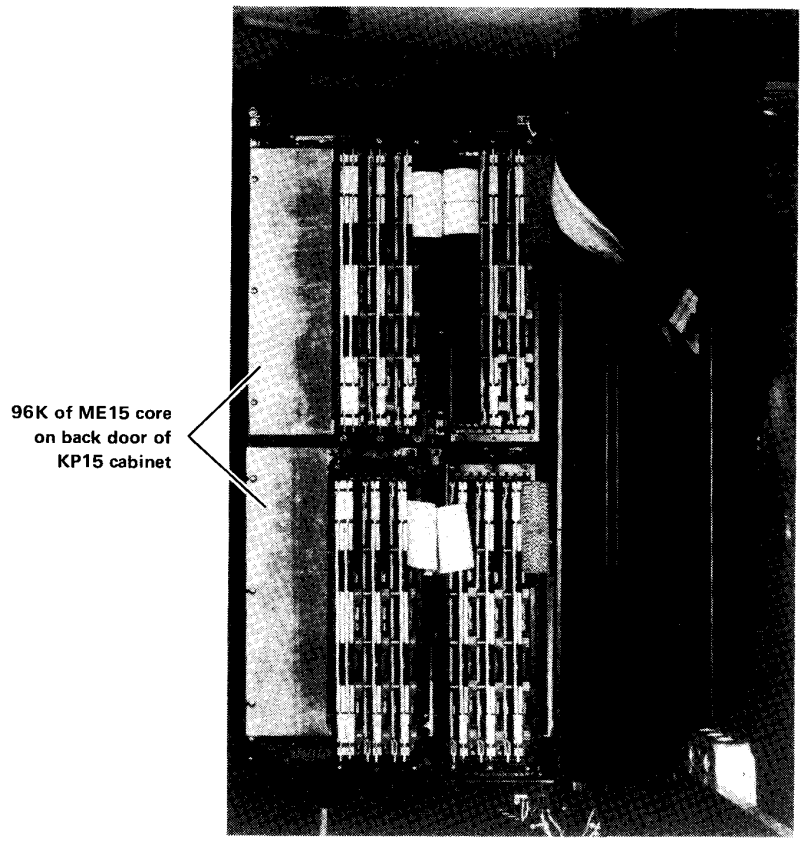


Figure 2-1 ME15 Rear View

Memory Loads

The PDP-15 memory bus can handle four unit loads at the same time. Three kinds of unit loads are available for the PDP-15 and can be intermixed:

- ME15 Memory with 8K to 48K words
- MM/MK15 with an 8K bank of 800-ns memory
- MX15 with up to four 8K banks of 800-ns memory

When MM15/MK15 Memory is addressed through the MX15, a 200-ns delay is added, making total cycle time 1.0 μs.

The entire memory is enclosed in a cooling box with fans at both top and bottom. Each of the four logic panels is six module slots high and nine module slots wide. The ME15 modular complement consists of M7170, G109-YA, G231, and H215 modules. Every 8K of installed memory requires one each of the G109-YA, G231, and H215 modules; one M7170 module is necessary for each 48K (or portion thereof) of memory capacity. The ME15 also requires one H742 Power Supply. The power supply mounts on the right side of the central processor cabinet.

Figure 2-1 illustrates the options required to configure an ME15 up to the maximum 96K

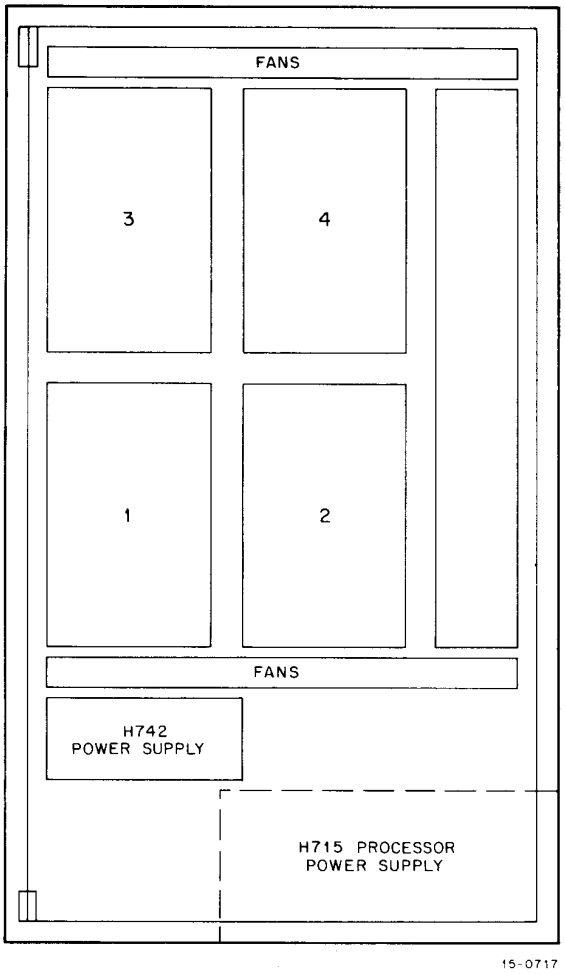


Figure 2-2 ME15 Panel Layout, Wiring Side
(Rear View of CP Cabinet)

memory capacity. Figure 2-1 does not show module positions or the particular modules required when memory is expanded. Figure 2-2 shows the panel layout, in order of installation; Figure 2-3 shows the assigned module slots for one panel.

ME15 MEMORY

The ME15 provides storage for up to 98,504 (96K) 18-bit words. Basic memory size is 8K; however, the memory can be expanded in 8K increments to 96K. Figure 2-4 illustrates a maximum size ME15. The modular complement of the ME15 is determined by the memory capacity. The modules are listed below:

a. M7170 Interface Control. This module functions as the interface between the PDP-15 bus and PDP-11 bus for all signals, data, and addresses exchanged by memory and the processor. The M7170 module is 8-1/2 in. long and of double height. One module is required for each 48K of memory capacity, or portion thereof.

b. G109-YA Memory Control. Hex-height and 8-1/2 in. wide, the G109-YA module contains the control logic, inhibit drivers, sense amplifiers, and an 18-bit data register for one H215 Memory Stack; one module is required for each 8K of memory capacity.

c. G231 Memory Driver. This module is also hex-height and 8-1/2 in. wide. It contains the address selection logic, current generator, and X and Y switches and drivers for 8K (one H215 Memory Stack) of memory.

d. H215 Memory Stack. The H215 provides core storage for 8,192, 18-bit words. It is quad-height and 8-1/2 in. wide.

The M7170 module provides pin-to-pin compatibility with both the PDP-15 bus and the PDP-11 bus. The G109-YA and H231 modules are pin-to-pin compatible with the PDP-11 bus and also with the H215 Memory Stack module connectors.

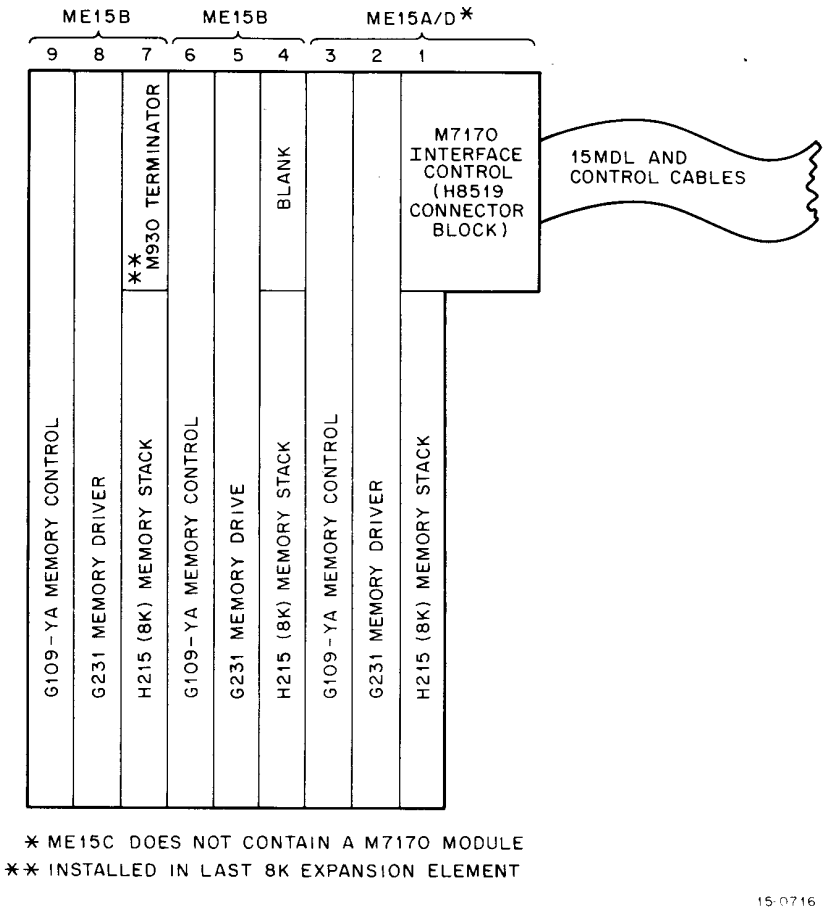


Figure 2-3 Assigned Module Slots (Module Side)

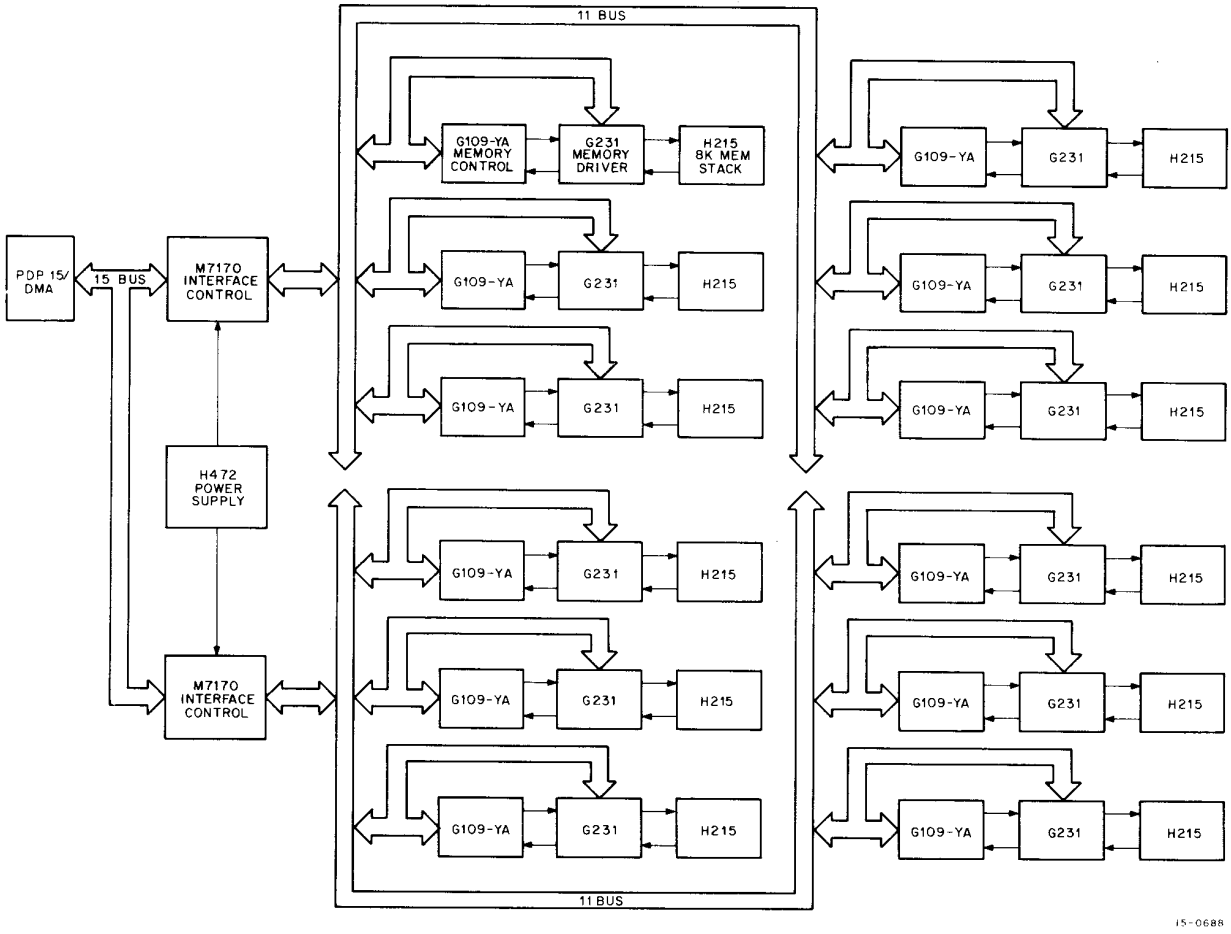


Figure 2-4 ME15 Expander Core Memory, Maximum Configuration (96K X 18 Bit)

ME15 Memory Specifications

Type:	Magnetic core, read/write, coincident current, random access
Organization:	Planar, 3D, 3-wire
Maximum Capacity:	98,504 (96K), 18-bit

Bus Mode	Cycle Time*
Read	980 ns
Write	980 ns
Read/Pause/Write	1.56 μs

* Typical time on a basic PDP-15.

X–Y Current Margins:	±6% @ 0° C, ±7% @ 25° C, ±6% @ 50° C		
Strobe Pulse Margins:	±30 ns @ 0° C, ±40 ns @ 25° C, ±30 ns @ 50° C		
Voltage Requirements:	+5 V ±5% with less than 0.05 V ripple -15 V ±5% with less than 0.05 V ripple		
Average Current Requirements:	Stand by:	+5 V: 2.2 A -15 V: 0.5 A	
	Memory Active:	+5 V: 5.4 A -15 V: 6.0 A	
Power Dissipation (Worst Case):	M7170 Interface Module:		~20 W
	G109-YA Control Module:		~60 W
	G231 Drive Module:		~40 W
	H214 Stack Module:		~20 W
	Total at maximum repetition rate:		140 W
Ambient Temperature:	0° to 50° C (32° to 122° F)		
Relative Humidity:	0–90% (noncondensing)		

PDP-15 Systems with 8K of MM15 Memory (no memory mounted on back door)

A new back door (with the ME15 logic, fans, and memory enclosure door attached) is supplied to these sites. The new back door should be attached to the CP cabinet and then the H742 Power Supply should be bolted to the right side of the cabinet frame so that it is positioned above the H715 Processor Power Supply. Figure 2-2 shows the CP cabinet from the rear with the relative position of the H742 shown. The H742 is mounted so that the breaker switch faces the front of the cabinet. The power harness can then be connected and dressed from the H742 to the ME15. With processor power off, the H742 power cable is connected to the ac outlet on the H715 or into the power control in the next cabinet. The signal cables are next connected to the first 8K of ME15 memory, after removing the M966 terminators from the MM15 and inserting them in the H8519 Connector Block. Cabling connections are specified in the ME15 Configuration Specification (A-SP-ME15-0-6). The address jumpers on the M7170 and the G109-YA modules should be checked to determine if the proper jumpers have been cut (refer to drawings D-CS-M7170-0-1 and D-CS-G109-0-1).

PDP-15 Systems with 16K of MM15 Memory (8K mounted on back door)

A new side panel, memory enclosure door, and fan housings are supplied as replacement parts to those sites already equipped with 8K of MM15 memory on the back door of the CP cabinet. The old fan housings, memory enclosure door, and side panel must be removed from the back door; the replacement parts and the ME15 memory panels are then mounted on the back door. The MM15 power harness is retained.

Once the ME15 power supply and logic panel are attached, the M966 terminators can be removed from the MM15 memory and inserted into the H8519 Connector Block (Figure 2-3). The signal cables from the MM15 are attached to the H8519 on the ME15 as specified in the ME15 Configuration Specification (A-SP-ME15-0-6). The address jumper configuration should be checked for correctness.

PDP-15 Systems with 24K of MM15 Memory (16K mounted on back door)

Installations with 16K of MM15 memory already fastened to the back door of the CP cabinet receive the ME15 mounted in an H963-U, ME15 Expander Cabinet. This cabinet will also contain a pre-mounted H742 Power Supply. The expander cabinet should be bolted to the left side of the CP cabinet in position BAY 1L (drawing D-AR-PDP15-0-2). With the exception of the yellow wire and black twisted pair, the power harness is supplied already connected to the ME15. The signal cables must be connected to the ME15 after repositioning the M966 terminators. Cable connections are specified in the ME15 Configuration Specification (A-SP-ME15-0-6).

PDP-15 Systems with Existing ME15 Memory

Expansion of installed ME15 memories requires the addition of certain units, the number and type of which are determined by the memory capacity before and after the expansion. Figure 2-5 shows the expansion options required when the ME15 is enlarged to any memory capacity up to and including 96K. Changes in cabling connections are specified in the ME15 Configuration Specification (A-SP-ME15-0-6). Depending on the increase in memory capacity, additional voltage regulators may have to be added to the power supply. Increased memory capacity also requires a change in the assigned device addresses. For more information, refer to the *ME15 Core Memory Maintenance Manual,* DEC-15-HMEMA-A-D.

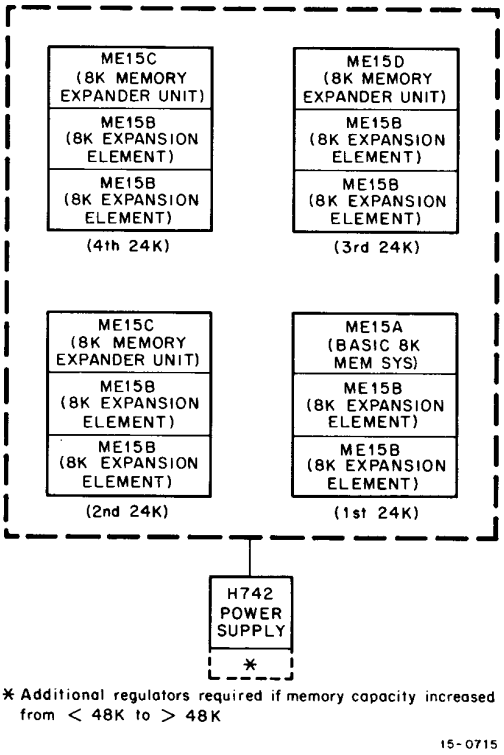


Figure 2-5 ME15 Core Memory Expansion Options (Module Side)

MM15/MK15 MEMORY

The MM15/MK15 Memory system is also available for the PDP-15. The basic subsystem of MM15/MK15 Memory is the bank, which is organized into pages; each bank has two pages of 4096 words each for a total of 8192 words of 3D 3-wire cores. Further, every bank contains a data buffer, an address buffer, and all the necessary read/write, control, and timing circuitry to make it an autonomous unit operating on a request/grant basis with either the port switch or the central or I/O processor. Figure 2-6 illustrates the organization of a memory bank.

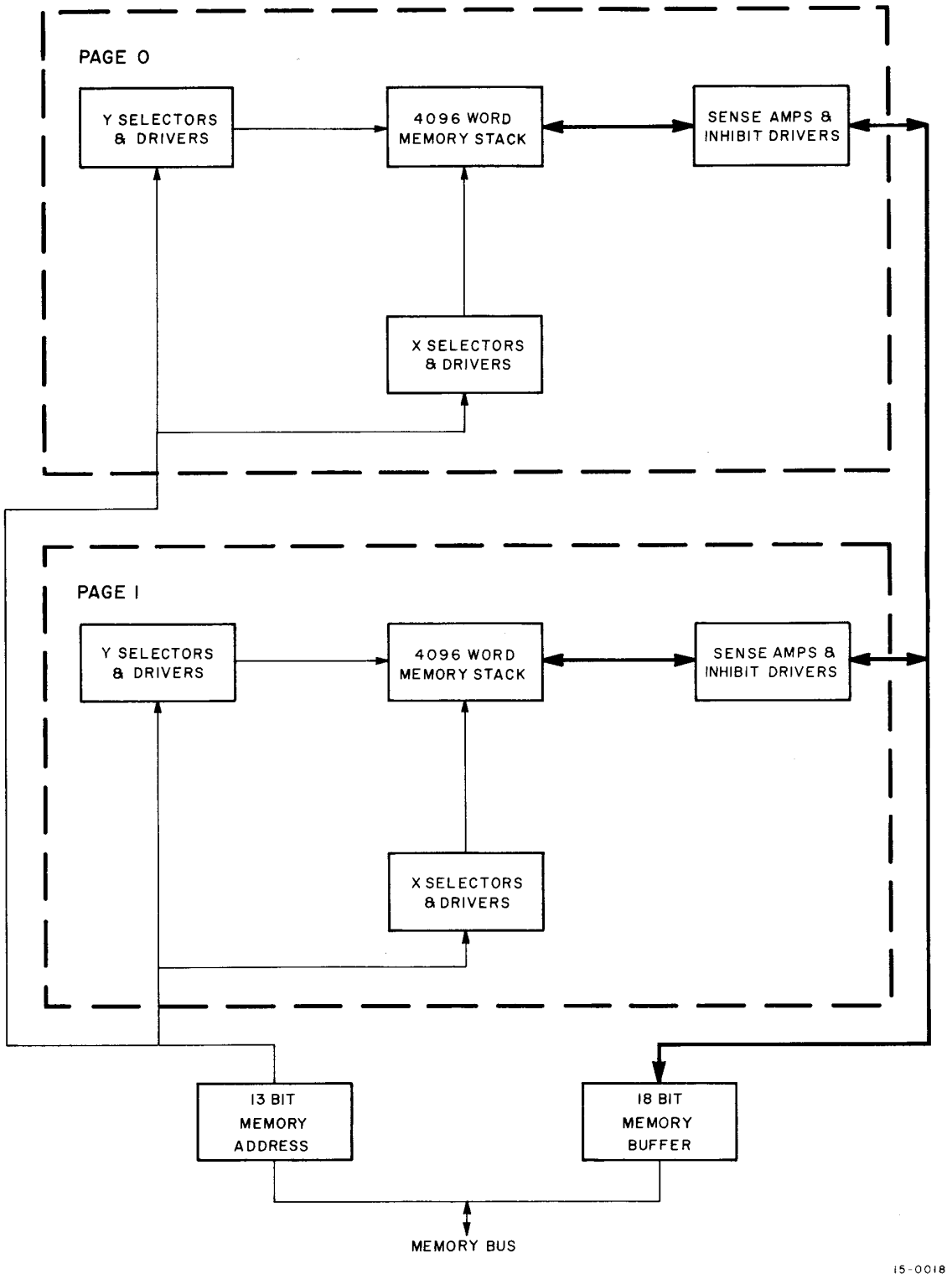


Figure 2-6 PDP-15 MM15/MK15 Memory Bank

MM15/MK15 Memory Specifications

Speed	Cycle time – 800 ns* Access time – 400 ns
Stack	Organization – 3-wire, 3D Core type – extended temperature 18 mil Drive Scheme – dc
Environment	Temperature – 0° to 50° C ambient (basic processor)
Special Features	Single Bus Type – Multiuser, bidirectional Bank Selection
Cycle Types	Read – Restore Clear – Write Read – Pause – Write

*Basic processor, without memory protect, or memory relocate and protect options.

CORE MEMORY

The magnetic core memory is the primary storage facility of the PDP-15. It provides rapid, random access data instructions storage for both the central and the I/O processors. The basic PDP-15/10 memory contains 4096 18-bit word locations. The content of each location is available for processing in 400 ns. The basic subsystem of memory is the bank, which is organized into pages; each bank has two pages of 4096 words each for a total of 8192 words of 3D 3-wire cores.

Further, every bank contains a data buffer, an address buffer, and all the necessary read/write, control, and timing circuitry to make it an autonomous unit operating on a request/grant basis with either the port switch or the central or I/O processor. Figure 2-6 illustrates the organization of a memory bank.

Memory Data Transfer

The PDP-15 memory communicates directly with the CPU and the I/O processor through the memory bus. Data and instruction words of each bank are read from and written into individual memory cells by a buffered register referred to as the memory data buffer (Figure 2-6).

Words in a memory bank are selected by their address in the memory address buffer. The memory address buffer's 13-bit capacity allows 2^{13} or 8192 words to be referenced in each bank.

The memory address buffer receives the memory cell address from the central or I/O processor. The address provides coordinates for locating a word in a memory bank.

Memory Cycles

Words are read from and written into memory by a fixed sequence of events called a *memory cycle.* The memory cycle consists of a read half-cycle and a write half-cycle. Each type of half-cycle requires 400 ns. Thus, the effective cycle time is 800 ns. For most applications, however, the two processors initiate a memory

request and wait until the end of the read half-cycle only. At this time, the desired data is available for reading or has been accepted at the memory data buffer for writing, and the central or I/O processor may proceed to the next step in its logical sequence of operations and perform useful functions while the write half-cycle is taking place. Thus, main memory access operations normally cause the two processors to wait only 400 ns. Delays caused by simultaneous requests by the two processors are discussed later in this section.

Read Half-Cycle – The read half-cycle copies the contents of the memory cell specified by the contents of the memory address buffer into the memory data buffer. If the parity option is present, a parity check bit is copied into the memory data buffer at the same time.

Write Half-Cycle – The write half-cycle copies the contents of the memory data buffer into the addressed memory cell. This half-cycle always occurs in conjunction with and following a read half-cycle, although there may be a "pause" between them, during which the I/O processor can manipulate the data in its add-to-memory mode.

Port Switch

The port switch allows both the central and I/O processors to share core memory. In the event that both request a memory cycle simultaneously, the I/O processor will be serviced first and the CPU must wait. However, if only one processor is using memory, both can process at the same time. For example, the CPU may be executing an EAE instruction while the I/O processor transfers data out of memory to a DECdisk.

MX15 Memory Bus Multiplexer

The MX15 Memory Bus Multiplexer is an option to be used with the PDP-15 computer system. This option adds three important capabilities to the PDP-15: Extended MM15/MK15 memory, Direct Memory Access (DMA), and Multiprocessing systems. The MX15 is a high-speed hybrid switch with three input ports and one output port that can establish one of three communication paths on a priority basis. Up to three memory data lines, each from a separate processor or DMA device, can be connected to 32K of MM15/MK15 core memory via this multiplexer. Granting of priorities for processor access to the multiplexer, thereby gaining access to the core memory, is governed by logic within the MX15. Port 1 has the highest priority and port 3 the lowest. Memory data line loading is such that a processor can drive up to four MX15s while each MX15 is capable of driving 32K of MM15/MK15 Core Memory. Thus, by using four MX15s, MM15/MK15 Core Memory may be extended to 128K (Figure 2-7).

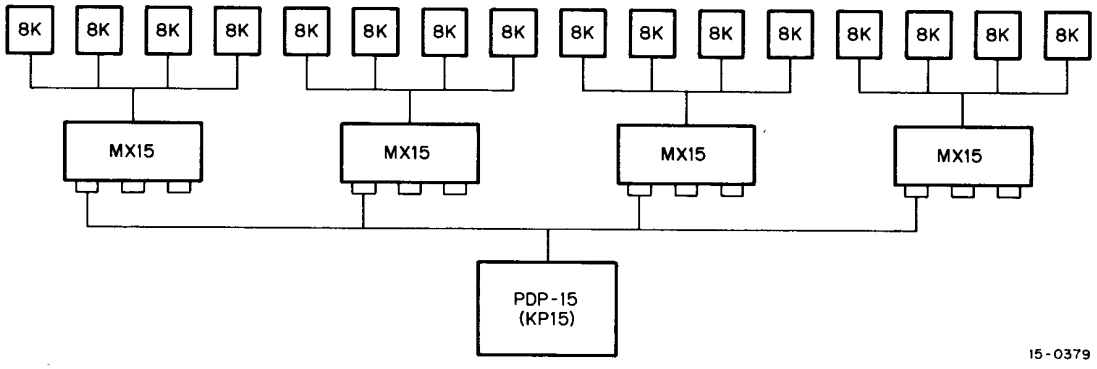


Figure 2-7 PDP-15 with 128K Expanded Memory

DIRECT MEMORY ACCESS

The MX15 also serves as a Direct Memory Access device. Any device, which contains the proper control circuitry, can obtain access to a PDP-15 memory through one of the ports of an MX15 (Figure 2-8).

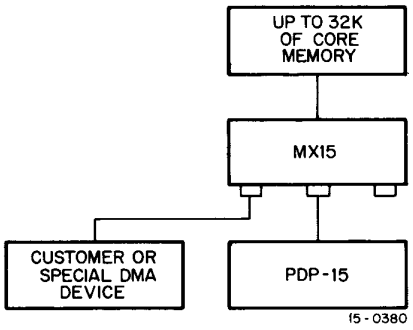


Figure 2-8 DMA Device on PDP-15 Memory

This type of interface provides a faster transfer rate under certain conditions and improved latency. The MX15 enables the device to transmit at memory speed (800 ns), plus worst case delay of 200 ns for transmission through the MX15. Thus, bidirectional I/O transfers of 1 MHz (500 kHz in each direction) are possible; latency is improved; and the MX15 will respond to a priority request within 1 μs.

MULTIPROCESSOR SYSTEMS

The MX15, with its three memory bus inputs, is well adapted to multiprocessor configurations.

Two popular multiprocessor configurations having many useful applications are illustrated in Figures 2-9 and 2-10. In Figure 2-9, both processors can 1) access all of memory and 2) simultaneously access portions of memory on different MX15s. Figure 2-10 illustrates how one (or each) of the processors may have a "private" core by using the MX15's "bank allow" switches, which can allow, or not allow, any input port of a MX15 to access any bank of that MX15.

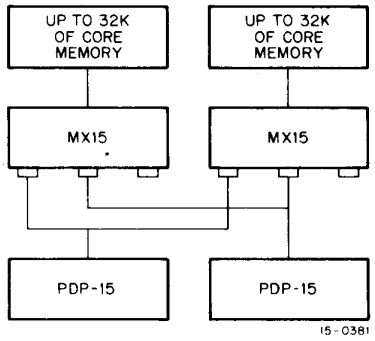


Figure 2-9 Multiprocessor System with Simultaneous Access to Two Sections of Memory

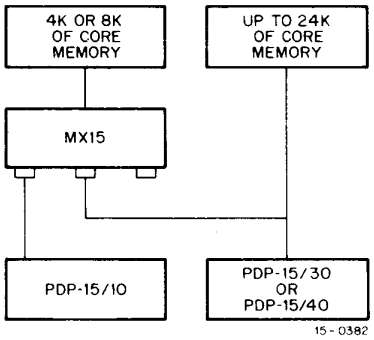
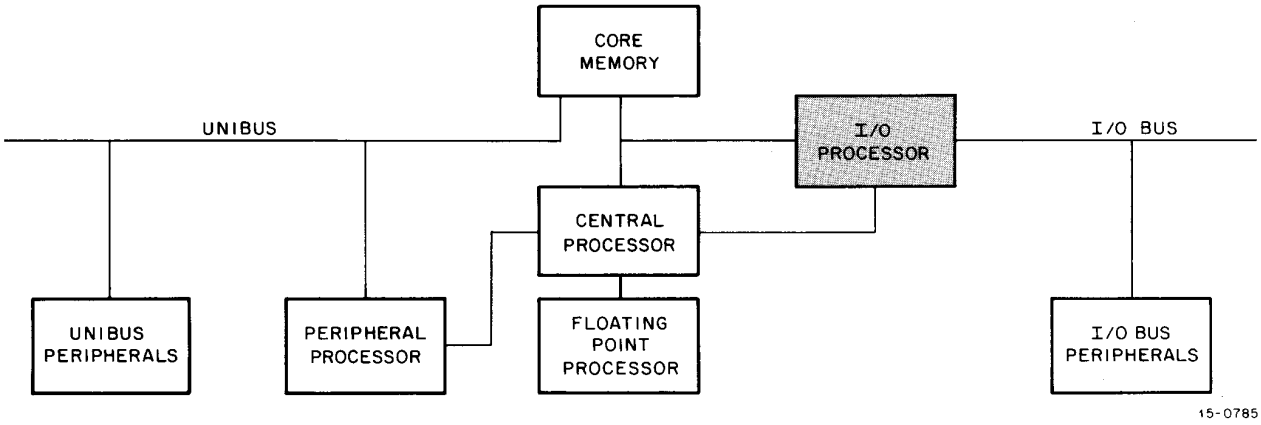


Figure 2-10 Multiprocessor System with One Processor Having Private Core

CHAPTER 3
INPUT/OUTPUT PROCESSOR



SUMMARY OF CHARACTERISTICS

The I/O processor contains two subunits, the data channel controller and the addressable I/O bus.

Data Channel Controller

Data Transfer Modes – Single and multicycle block transfer, memory-increment, add-to-memory

Block Transfer Channels – Eight standard

Options – Real-Time Clock – KW15

Addressable I/O Bus

Features – Two cycle skip line, program interrupt, Teletype interface, console interface

Data Transfer Modes – Program controlled data transfers

Device Ports – A maximum of 50 physical ports shared between the data channels and the addressable I/O bus

Options – API–Eight levels of automatic priority interrupts – Four hardware levels and four software levels

I/O PROCESSOR

The I/O processor (Figure 3-1) contains the control logic and registers necessary to transfer up to 18 bits of parallel data on a common bidirectional I/O bus. Data may be transferred directly between the I/O processor and memory, or between the I/O processor and the accumulator (AC) of the CPU. All transfers are made on a request/grant basis, providing complete autonomy of processors and memory. The I/O processor operates with a 1-μs cycle time. The processor accesses memory through the read-pause-write, read restore, and clear write cycles.

While transfers are being made between memory and the I/O processor, the CPU is free to operate independently. Requests from the I/O processor for memory access are, however, given priority over CPU requests by the port switch; this can cause the CPU an occasional "cycle-stealing" delay. The structure of the I/O processor provides the following benefits to the user:

- The simultaneity of data transfers and CPU computing permits high-speed processing to meet the demands of real-time applications.
- User-designed or special-purpose equipment can be easily and inexpensively interfaced to the system.
- Synchronous and asynchronous devices can be handled with equal ease.
- Direct memory access devices that would otherwise require additional interface logic require only one interface with the PDP-15 single cycle data channel.

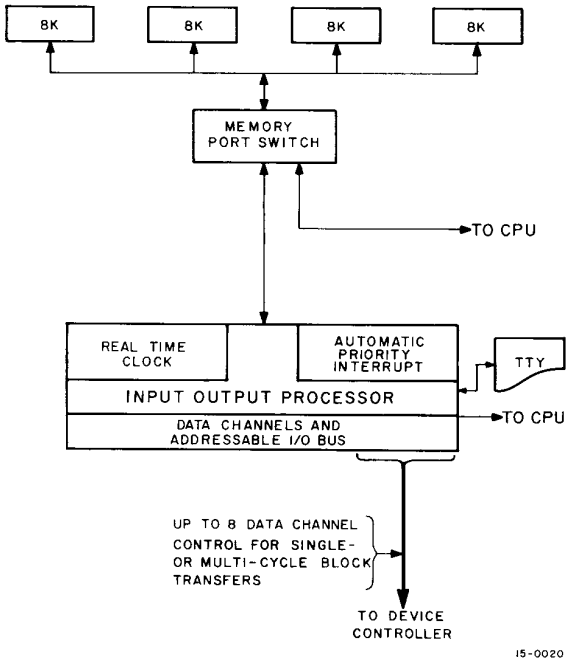


Figure 3-1 I/O Processor

Modes of Data Transfer

Peripheral devices may transfer data in any one of three modes: single cycle block transfers, multicycle block transfers, and program-controlled transfers.

Block Transfer Controller

The block transfer controller implements the first two modes of data transfers and in addition has an add-to-memory mode and an increment memory mode. The real-time clock option is also implemented in this section.

Eight block transfers are standard on all PDP-15 systems and may serve to concurrently transfer data from eight different devices. The channels are designed to accept any mixture of either single or multicycle devices.

Multicycle Block Transfers

A two-word packet in core memory is reserved for each of these channels: locations 30 and 31 for the first, 32 and 33 for the second, 34 and 35 for the third, and 36 and 37 for the fourth in standard PDP-15 software systems. The two words in the packet are used to store the "word count" (number of words to be transferred in the block), and the "current address" (where the data is to be transferred). The I/O processor contains the control logic and an I/O adder to automatically fetch and increment the contents of the two registers.

Data is read into memory and out from memory in three I/O processor cycles. Maximum input rate is 250,000 words/second and maximum output rate is 181,000 words/second, ensuring data transfer integrity.

Prior to initiating a multicycle block transfer (flowcharted in Figure 3-2), the program stores the 2's complement of the word count and the current address minus one in the two appropriate memory locations. The transfer is then initiated by an IOT instruction. During the first cycle, the contents of the Word Count register are incremented by one and restored. During the

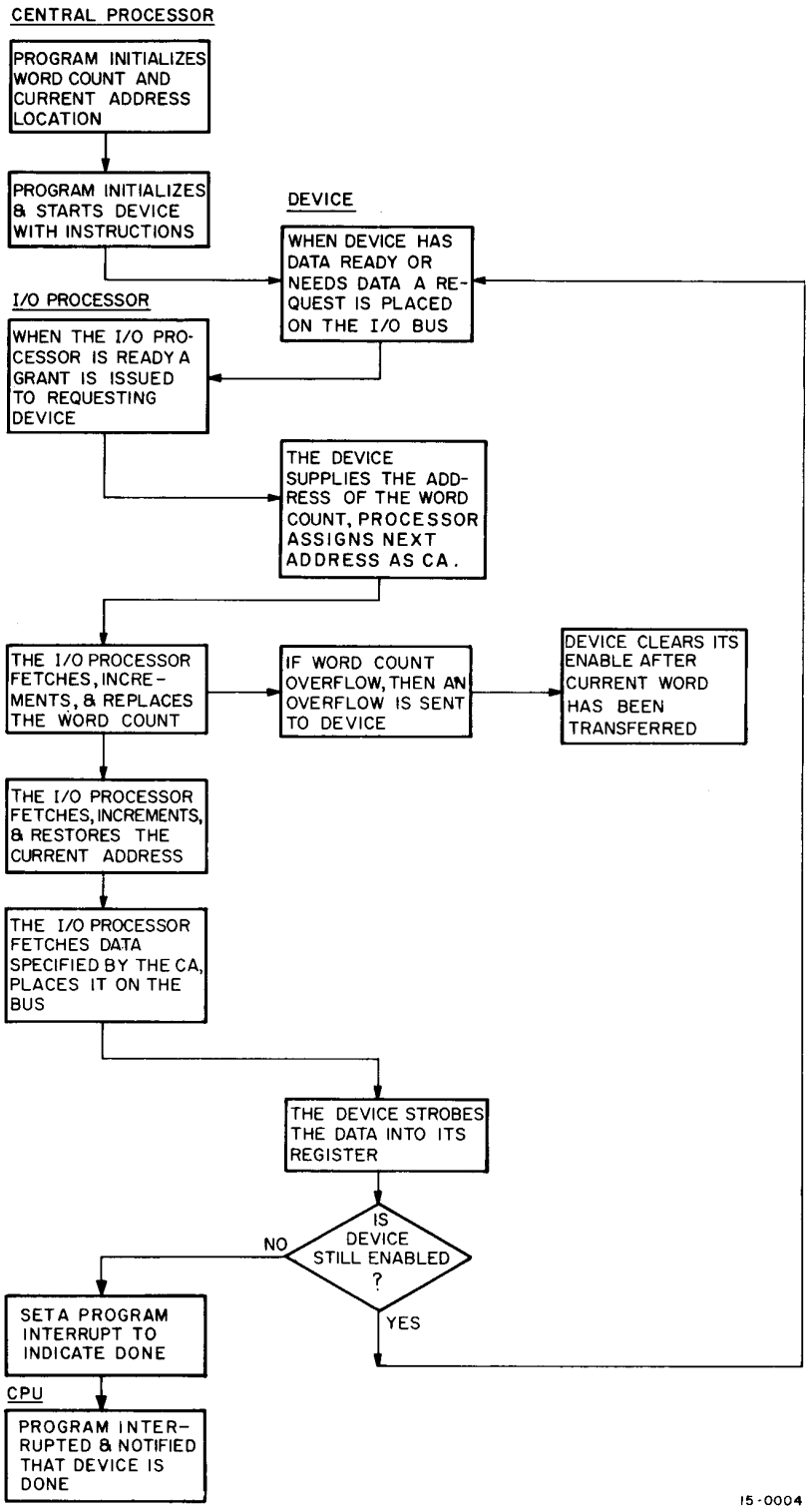


Figure 3-2 Multicycle Block Transfer, Flowchart

second cycle, the current address is incremented by one and restored, in addition to being transferred to the memory address buffer of memory. During the third cycle, the actual data transfer occurs. The I/O processor continues to transfer data sequentially until the Word Count register reaches 0, at which time an interrupt is generated to notify the monitor that the block transfer is complete. Because these multicycle block transfers are completely automatic and do not require any CPU attention except for I/O transfer initialization, the CPU is free to compute while they are taking place. The only limitation on simultaneity lies in the sharing of memory. The I/O processor has first priority on memory requests and effectively "locks out" the CPU for three cycles. As data transfer rates approach maximum, the CPU can be completely locked out.

Figure 3-3 illustrates how the data channel controller registers implement the multicycle transfers.

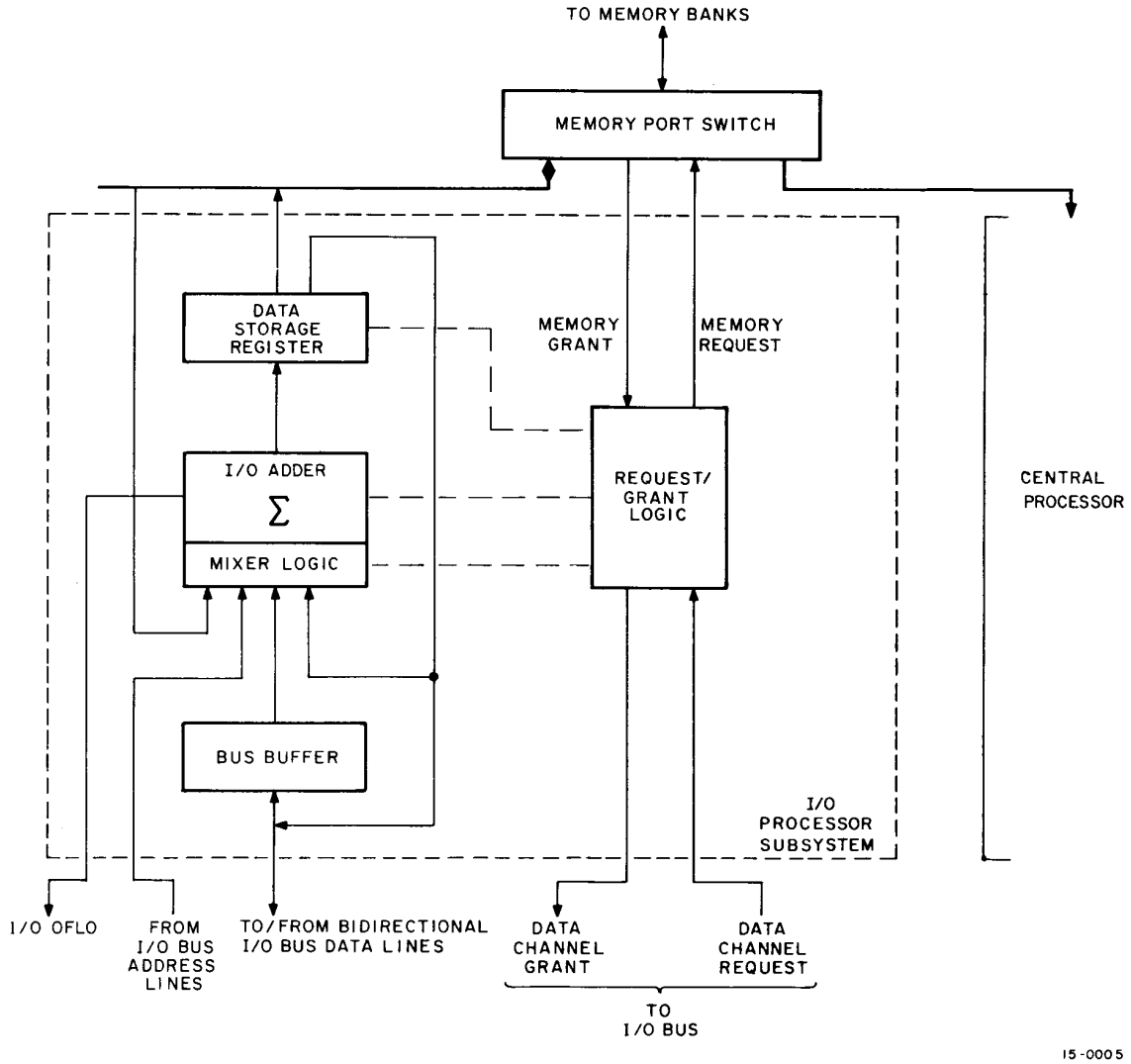


Figure 3-3 Multi-Cycle Transfer Implementation

Assume the two-word core memory packets assigned to a given multicycle data channel have been loaded by the respective I/O service routine. For the case of data input to memory, the following occurs:

An instruction from the service routine enables the device controller. This allows the controller to request a data transfer from the I/O processor.

When the device controller's data buffer registers are full, it issues a "data channel request."

The I/O processor, if not busy, acknowledges the request by returning a "data channel grant."

The device controller then generates a fixed code pointing to its packet address in core memory. This is transmitted over the common I/O bus address lines and is is stored into the Data Storage register of the data channel controller through the I/O adder. The adder is inhibited during this operation.

The I/O processor then generates a "memory cycle request."

The port switch, when ready, acknowledges by returning a "grant."

The address data on the I/O bus address lines is then stored into the Memory Address (MA) register of the memory bank. This address points to the word count, the first word of the packet. This data is transmitted out of memory and into the data channel controller's adder. The word count data word is incremented by one and stored back into memory. If during this incrementing the adder overflows (indicating that the current address was the last), then an I/O overflow pulse is transmitted back to the device to disable future data channel requests from that line and also to post an interrupt to the monitor.

This "word count" operation occurs in one I/O processor cycle, using one memory cycle.

During the second I/O processor cycle, the fixed code from the device controller is gated through the adder and is incremented by 1. It is then transmitted to the MA register to point to the second word in the packet, the "current address," which is then transmitted back to the adder, incremented by 1, and strobed back to memory. During the third I/O processor cycle, the current address is strobed into the MA register to point to the data array word where the I/O data will be transferred. The data is then gated from the device controller, through the adder (which is inhibited during this cycle) and into memory.

A memory request/grant synchronization again occurs and:

The data in the storage register is strobed into the data array ending the cycle.

Data output follows the same sequence, with the exception that one additional I/O processor cycle is required to allow the bus to settle before data is strobed out of memory and into the device.

Single Cycle Block Transfers

Single cycle block transfers, flowcharted in Figure 3-4, are used by high-speed peripherals that normally transfer complete records (blocks) of information, such as disks and CRT devices. A single cycle of the I/O processor takes 1 μs allowing a maximum transfer rate of up to one million 18-bit words per second.

High-speed hardware registers, designed into the device controllers of the high-speed peripherals, store the "current address" (the memory cell where data is currently being transferred), and the "word count" (the number of words remaining to be transferred in a block). These registers are loaded by input/output transfer (IOT) instructions issued by the CPU. Device testing

and initialization are handled by the CPU via IOTs to provide supervisory control. A subsequent IOT initiates the data transfer. The I/O processor uses the current address information to address core memory, then strobes the data between memory and the device controller. Logic within the device controller then increments the Current Address register and the Word Count register to provide sequential block transfer.

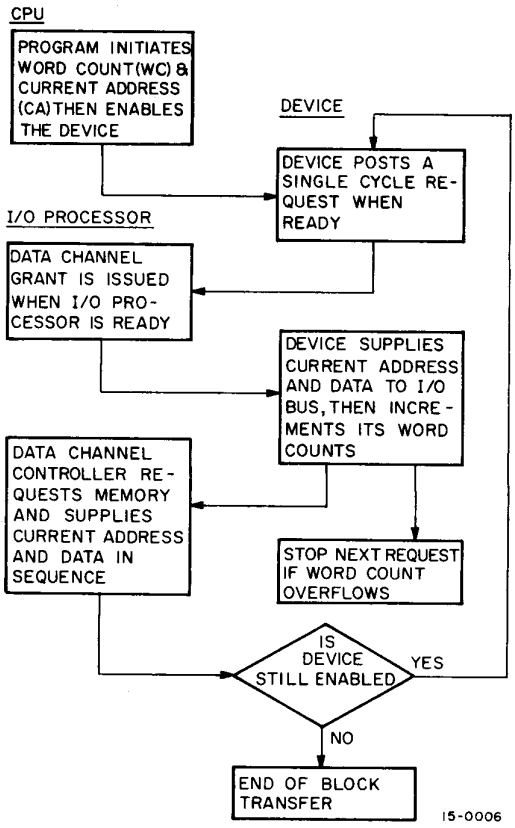


Figure 3-4 Single Cycle Block Transfer, Flowchart

When the Word Count register overflows at the end of a block transfer, an interrupt is generated to allow the monitor system to take further action. Typically, this action will include disconnecting the device from the I/O bus or reloading the device controller registers for another block transfer. The maximum number of words that may be transferred in a single block is 32,768.

Figure 3-4 illustrates how the data channel controller registers handle single cycle transfers.

Assuming that the program has initiated the word count and current address of the device controller, and has then enabled it, the following occurs:

The device controller posts a single cycle data channel request to the I/O processor.

The I/O processor, as soon as it becomes available, acknowledges the request by returning a "Data Channel Grant."

The device then strobes both its current address and its data onto the I/O bus to the I/O processor.

The data channel controller feeds the current address through its adder (which is inhibited throughout the single cycles) to the Data Storage register. A memory cycle is requested, and this address is strobed into a memory bank's address buffer. The data is then strobed off the 18 I/O data lines and into the memory location specified by the current address.

During this operation, the device increments its own word count, and disables itself on overflow. It then posts an interrupt to the monitor to indicate that its operation has been completed.

Increment Memory

The increment memory mode allows an external device to add one to the contents of any memory location in a single cycle; this feature is most commonly employed in the accumulation of data in histogram form. Effectively, the

increment memory mode simply goes through the word count cycle of a multicycle channel transfer, and then stops. The maximum rate at which it can increment is 333 kHz. This feature is particularly useful for in-core scaling and counting in pulse height analysis.

Add-to-Memory

Add-to-memory is a standard feature of the PDP-15 that adds unique capabilities to the already powerful I/O facilities.

In add-to-memory mode, the contents of an external register can be added to the contents of a memory location in four cycles. This feature is extremely valuable in signal averaging and other processes requiring successive sweeps for signal enhancement.

The add-to-memory operation is a combination of multicycle data channel input and multicycle data channel output operations. The data transmitted by the device is added to a word read out of memory as specified by the current address, and the result is rewritten into the same location. It is simultaneously transmitted to the device via the I/O bus. The maximum add-to-memory rate is 188 kHz.

Real-Time Clock

When enabled, the real-time clock counts, in memory location 00007, the number of cycles completed by the line voltage (50 or 60 Hz) or any standard DEC clock module that may be optionally installed. Maximum recommended clock frequency is 10 kHz.

When location 00007 overflows, an internal program interrupt (or API request, if available) is generated, informing the monitor that its preset interval is over. The monitor must either disable the clock or reinitialize location 00007 to the 2's complement of the number of counts it needs to tally.

The incrementing of location 00007 during a real-time clock request occurs via the I/O processor's increment memory facility. A real-time clock request takes priority over API, PI, and IOT requests, but not over block transfers.

ADDRESSABLE I/O BUS

The addressable I/O bus implements the program-controlled transfers. It also contains the program interrupt and the automatic priority interrupt (API) option.

Program-Controlled Transfer

Program-controlled transfers, implemented by input/output transfer (IOT) instructions, can move up to 18 bits of data between a selected device and the accumulator (AC) in the CPU. The devices involved are connected to the addressable I/O bus portion of the I/O processor. A total of up to 50 device controllers may be attached to this bus and to the data channels. IOT instructions are microcoded to effect response only for a particular device. The microcoding includes the issuing of both a unique device selection code and the appropriate processor-generated input/output pulses to initiate a specific operation. For an "out" transfer, the program reads a data word from memory into the AC. A subsequent IOT instruction places the data on the bus, selects the device, and transfers the data to the device. For an "in" transfer, the process is reversed; an IOT instruction selects the device and transfers data into the AC. A subsequent instruction in the program transfers the word from the AC to memory. Maximum transfer rate in this mode is 150,000 words per second.

As previously mentioned, IOT instructions are also used to initialize the single and multicycle channels and the transfer word count and current address information to the single cycle device controllers. In addition, these instructions are used to test or clear device flags, select modes of device operation, and control a number of processor operations.

A PDP-15 IOT instruction, Figure 3-5, contains the following information:

a. An operation code of 70_8.

b. An 8-bit device selection code to discriminate between up to 256 user peripheral devices (selection logic in a device's I/O bus interface responds only to its preassigned code). In normal practice, bits 6 through 11 perform the primary device discrimination between up to 64 device codes with bits 12 and 13 coded to select an operational mode or subdevice.

c. A command code (bits 14 through 17) capable of being microprogrammed to clear the AC and issue up to three pulses via the I/O bus.

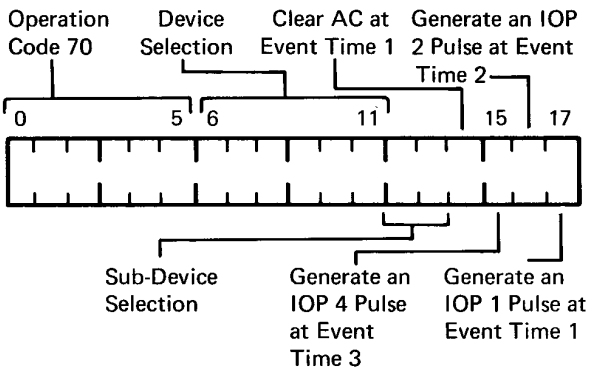


Figure 3-5 IOT Instruction Format

Up to four machine cycles may be required to execute an IOT instruction. These include the IOT fetch from core memory (memory is not accessed thereafter until completion of the IOT), and three sequential cycles each of 1 μs duration designated event times 1, 2, 3 (IOP1, 2, and 4) (Figure 3-6). In IOT skip instructions, however, only IOP1 is used. These are two-cycle instructions. Bits 14 and 17 can be coded to initiate clearing of the AC and generation of an IOP1, respectively, during event time 1. Bits 16 and 15 can be coded to initiate generation of an IOP2 and IOP4 pulse during event times 2 and 3, respectively. IOT skip instructions are microprogrammed to produce an IOP1 pulse for testing a device status flag. IOP2 pulses are normally used to effect programmed transfers of information from a device to the processor. Because the AC serves as the Data Register for both "in" and "out" transfers, the "clear AC" microinstruction (bit 14) is usually microprogrammed with the IOP2 microinstruction; this combination effects clearing the AC during event time 1. IOP4 pulses are normally used to effect programmed transfers of information from the AC to a selected device. These conventions do not, however, preclude use of the IOP pulses to effect other external functions if the following restrictions are observed.

The *usual* uses of IOPs are:

IOP1 – Normally used in an I/O skip instruction to test a device flag. May be used as a command pulse and to load data into a device, but may not be used to initiate either a "load of" or a "read from" device.

IOP2 – Usually used to transfer data from the device to the computer, or to clear device's information register. May not be used to determine a "skip" condition.

IOP4 – Used only to transfer data from the computer to the device. May not be used to determine a "skip" condition or to transfer data to the computer.

PROGRAM INTERRUPT FACILITY

The program interrupt (PI) system, standard on all PDP-15 systems, provides for servicing a peripheral device at rates up to 50 kHz.

The program interrupt (PI) facility, when enabled, relieves the main program of the need for repeated flag checks by allowing the ready status of I/O device flags to automatically cause a program interrupt. The CPU can continue with execution of a program until a previously selected device signals that it is ready to transfer

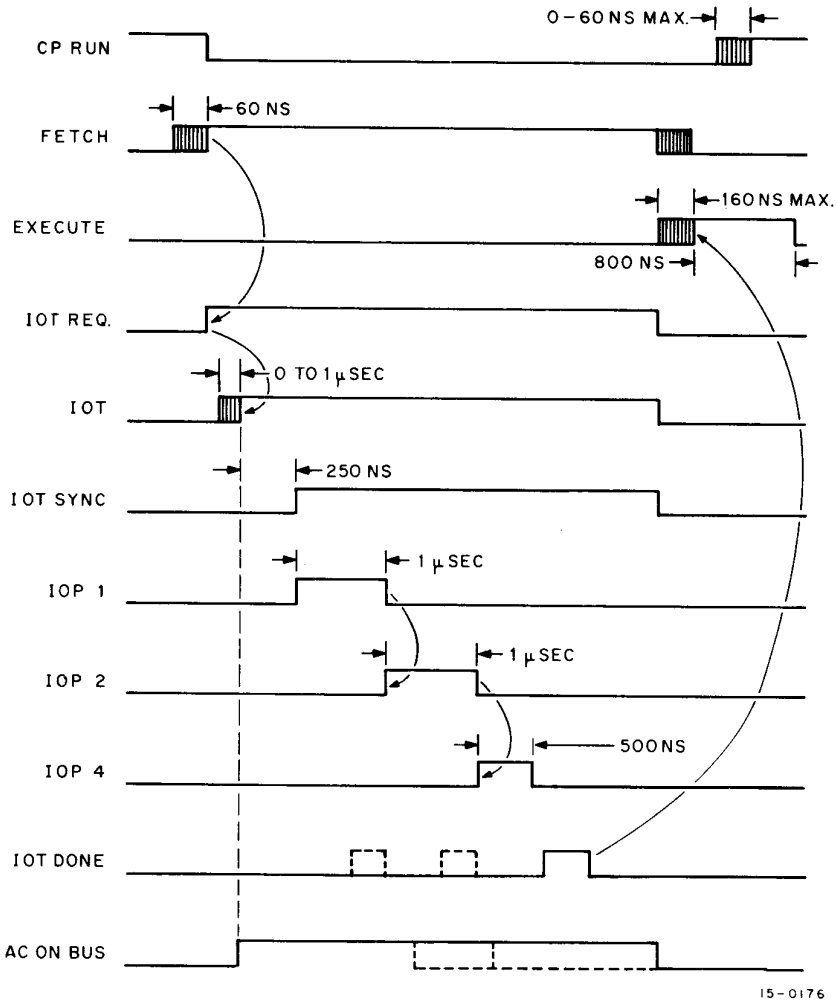
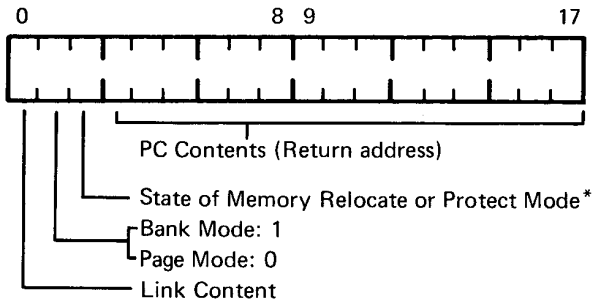


Figure 3-6 IOT Instruction Timing

data. At that time, the program in process is interrupted and automatically the contents of the program counter (15 bits), memory protect mode (1 bit), bank or page addressing mode, and the link bit (1 bit) is stored in location 000000 (Figure 3-7). The instruction in location 000001 is then executed, transferring control to an I/O service routine for IOT instructions. When completed, the routine restores the system to the status prior to the interrupt with a single instruction, allowing the interrupted program segment to continue. Where multiple peripherals are connected to the PI, a search routine containing device status testing (skipping) instructions must be added to determine which device initiated the interrupt request. The program interrupt (PI) control is enabled or disabled by programmed instructions (IOTs). When disabled, the PI ignores all service requests, but such requests normally remain on-line and are answered when the PI is again enabled, unless they are cleared.



*Zero if respective option is not present.

Figure 3-7 Memory Location 0 at Time of Program Interrupt

Mnemonic	Octal Code	Function
ION	700042	Enable the PI
IOF	700002	Disable the PI

The PI is automatically disabled when an interrupt is granted or when the RESET key (on the console) is depressed. The PI is temporarily inhibited while the automatic priority interrupt system is processing a priority interrupt request. The PI ENABLE indicator (on the console) is lighted while the PI is enabled.

Conditional Skip-On Device Status

The PDP-15 order code includes a group of instructions for testing the status of peripherals. Instructions of this type direct the processor to skip the next instruction if the tested condition is true.

This group of instructions allows the testing of peripheral devices at the programmer's option. Normally rather than tying the processor up in a "wait" loop, the device signals that its buffer is ready by generating an interrupt. If it is a program interrupt, the "conditional skip" is used in a "skip chain" to find which device initiated the interrupt. Each skip instruction takes an average of 1.8 μs.

AUTOMATIC PRIORITY INTERRUPT

The automatic priority interrupt (API) system ensures efficient handling of service requests (without any loss of data) at high rates.

The API system contains eight levels of priority. The lower four of these are allocated to the monitor systems, the upper four to the I/O processor. Up to 32_{10} individual interrupts are available at the I/O processor.

A device initiates an interrupt request on its preassigned level by raising a request flag, and identifies itself by posting a unique core address. There is one unique core address for each of the 32_{10} interrupts (44_8 through 77_8 are reserved in the standard software). This address serves as the entry point (trap address) to the device's service routine.

Each monitor API level services one interrupt and uses a single trap address between locations 40_8 and 43_8. The monitor requests are initiated by a program issuing an ISA instruction.

The I/O interrupts permit the asynchronous operation of many devices, each at its proper priority level. The software priority levels are used to establish a priority queue for the processing of real-time data without inhibiting the hardware interrupts to service devices.

API Hardware

Figure 3-8 relates the activity of the Automatic Priority Interrupt (API) system from the initiation and acceptance of the request, the servicing of the accepted request, and the debreak from the serviced priority level.

The API Request register contains eight levels; four levels are activated by the devices (hardware) on the I/O bus, and four are activated under software supervision. The hardware requests are assigned the highest priority, and are

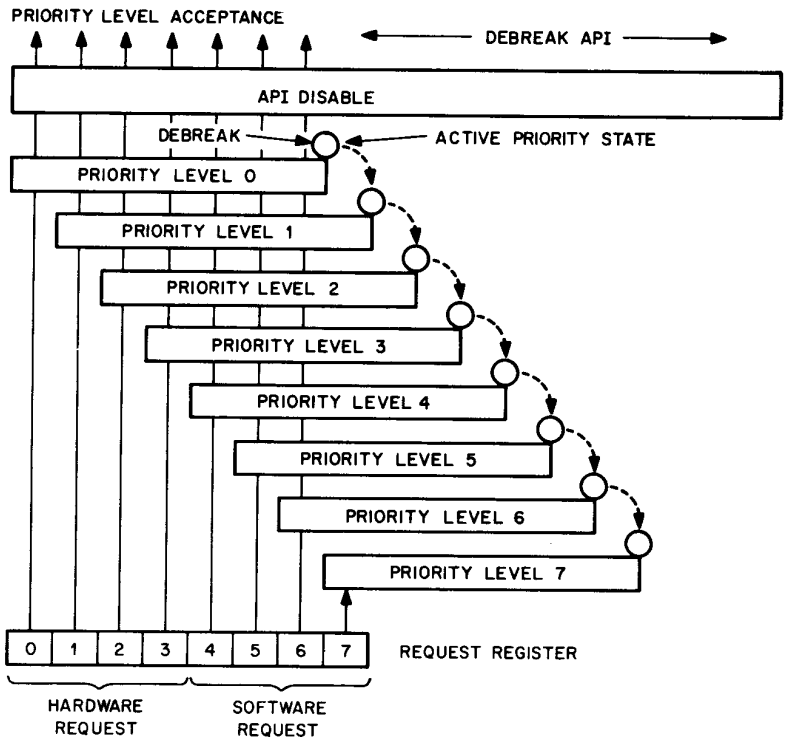


Figure 3-8 API System, Simplified Block Diagram

designated request levels 0, 1, 2, and 3. The software requests are designated request levels 4, 5, 6, and 7, and are initialized by the ISA instruction with the associated AC bits set.

The priority level *bars* depict the priority level that is selected by the ISA instruction, or that is raised by the API control when it has granted a break request on that specific level. The priority level (PL) bars indicate that any request equal to, or less than (in priority), the priority level active will *not* be accepted. At the end of the subroutine currently being performed by an active request, a debreak and restore instruction is issued. This will lower the priority to the next *requesting* priority level. The ball, representing the priority debreaking, will fall as long as there is *no* bar present (i.e., no priority level set). If a lower priority level is set, the debreaking will cease at *that* level.

The API Request register (RR) buffers the inputs from the hardware interrupt on levels 0 through 3 and the inputs from the monitors on levels 4 through 7. If two or more interrupts on one level make simultaneous interrupt requests, the device controller closest to the processor is given priority and interrupts at its unique location. An interrupt request sets a bit in the RR according to its preassigned priority level. When the scanner detects that bit, the API system signals the CPU to stop execution at the completion of its current instruction. It then gates the I/O processor's 15 address lines, which contain the address of the interrupt's unique core location, into the CPU Memory Output register. The CPU then requests a memory cycle and executes the instruction it fetched from that location. During this operation the program counter remains unchanged. The instruction is normally a jump to subroutine (JMS) that stores the contents of the program counter, which, in turn, points to the location where the current program was interrupted in the first location of the subroutine and begins execution at the subroutine's second location.

The API system also sets a bit in the PL corresponding to the level of the interrupt. This prevents interrupts on the same level or lower levels from interrupting the current interrupt. The scanner continues to sample the higher levels so that higher priority devices can interrupt lower priority devices. The JMS instruction allows nesting of all levels. At the completion of the interrupt subroutine, a debreak and restore (DBR) instruction must be issued to reset the bit in the PL and in the RR.

The API hardware ensures that simultaneous requests by multiple devices are handled in the proper priority sequence. If interrupt requests occur at different priority levels, the highest priority requests will be serviced first. Higher priority devices may interrupt lower priority devices. The entire API system may be enabled or disabled with a single instruction; however, most devices provide facilities to connect and disconnect their flags from the interrupt separately. If the API system is disabled, the device will automatically signal the program interrupt to obtain a response at that priority level.

The program interrupt sets level 3 if priority levels PL 0–3 are 0, PL 3 is enabled, and no other API request has been synchronized.

Under program control, the level of a priority request may be raised to provide dynamic priority reallocation. It does this by issuing an ISA.

ISA	705504	Initiate selected activity. The API activity specified by a set bit in the AC is initiated (refer to instruction set Chapter 10).

The ISA instruction places a bit into a priority level specified. This effectively masks all lower priority levels. A debreak instruction (DBK) is used to reset this bit when the higher priority level is no longer required. For example, a priority-2 interrupt routine is designed to enter data in memory locations A through A + 10 during an interim period when the priority-2 device is inactive and, based on a calculation made by a software priority-6 routine, it becomes necessary to move the data to memory locations B through B + 20. The changes in the routine at level 6 must be completed without interrupt once they are started. This is possible by causing the level 6 program to raise itself to level 2 (devices on the same or lower priority may not interrupt), complete the change, and debreak back to level 6. Note that the ISA is also used to trigger the API levels dedicated to software priority queues. This unique advantage of the API system lies in the proper use of its software levels. In real-time environment, it is necessary to maintain data input/output flow, but it is not possible to perform long complex calculations at priority levels which shut out these data transfers. With the API, a high-priority data input routine that recognizes the need for complex calculations can call for a software-level interrupt. Since the calculation is performed at a lower priority than the device handling, the latter can go on undisturbed. The monitor task of establishing a multipriority request queue at the software level is greatly simplified by utilization of the API hardware.

COMMON I/O BUS

The I/O processor contains a common I/O bus (Figure 3-9) to transfer both data and IOT instructions to the block transfer channels and to the addressable I/O bus. The bus is the major communication path for I/O devices. It consists of cables that link all the I/O device controllers to a common interfact point at the I/O processor. All signal lines for command and data transmission arising from the data channels, addressable I/O bus, operation of the multilevel automatic priority interrupt system, program interrupt, I/O status read, and I/O skip facilities, are contained on this bus. The bus length can be up to 75 ft.

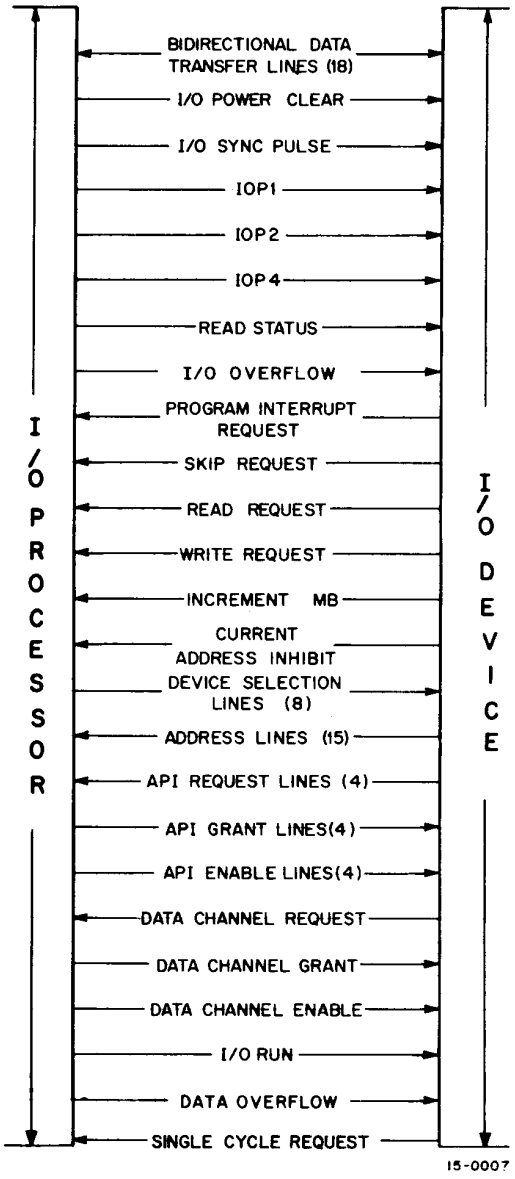


Figure 3-9 Common I/O Bus

Data Lines

Eighteen data lines constitute the bidirectional facility for transferring data bytes of up to 18 bits between the I/O processor and all I/O devices. Transfers are made on a dc basis with the processor or device allowing bus settling time before data on the lines is strobed into the receiving register. The data lines convey data between the Memory Buffer register and a selected Device Buffer register for block transfer channel operation; they transfer data between the accumulator and a selected Device Buffer register for program-controlled transfers.

Output Control Signals

Eight output control signals are generated to effect specific functions in a selected device.

I/O Power Clear

The I/O power clear signal resets all flip-flops storing device-to-processor flag indications (e.g., ready, done, busy). It is issued by power turn on and off, the occurrence of a clear-all-I/O flags (CAF) instruction, and by actuation of the RESET key on the control console. This pulse is also used (in conjunction with the device select lines) to initiate automatic read-in from the selected device.

I/O Sync

I/O sync is used to synchronize I/O device control timing to the processor. It is issued once every I/O processor cycle.

IOP1, IOP2, IOP4

Microprogrammable signals are used to effect IOT instruction-specified operation within a selected I/O device. The I/O processor automatically generates IOP2 or IOP4 for data channel input or output transfers. Although they may be used for other control functions, the common uses of the IOPs are:

IOP1 – to test a device flag (in an I/O skip instruction). It may be used as a command pulse and to initiate loading, but it cannot be used for reading from a Device Buffer register.

IOP2 – to transfer data from a selected device to the processor or load a device. Not used for skip.

IOP4 – to transfer data from the processor to a selected Device register. It may not be used to determine a skip condition or to transfer data to the processor.

Read Status

Read status is issued by execution of the input/output read status (IORS) instruction. It loads the AC with an 18-bit word containing device flag indications for devices interfaced to the read status facility.

Input/Output Read Status

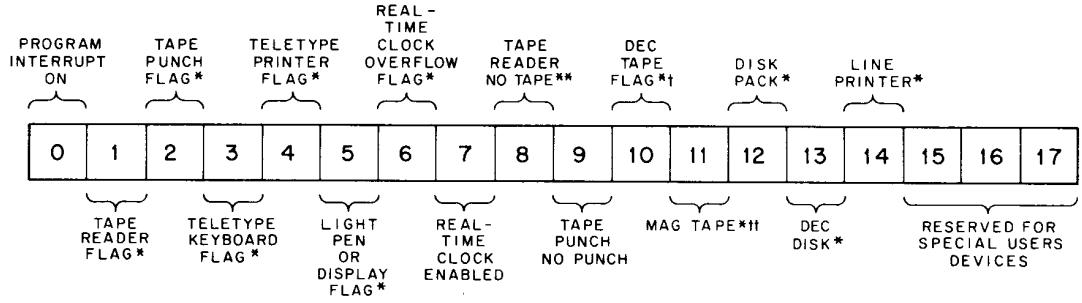
The input/output read status facility provides for programmed interrogation of the status of those external devices using this facility. Upon execution of an input/output read status (IORS) instruction, the states of those device flags (done, busy, not ready, etc.) interfaced to this facility by the I/O bus are transferred to specific assigned bit positions of the AC. This allows the program to check for specific flag conditions or display the flag states on the operator's control console. Figure 3-10 shows the bit positions associated with the commonly interfaced flags.

The IORS word can contain up to 18 flag bits. Those bits not used are zeroed. The presence of a flag is indicated by a 1 state in the corresponding AC bit.

Switching the REGISTER DISPLAY control (on the console) to the STATUS position simulates execution of the IORS instruction (with the processor in the "program stop" condition). The contents of the IORS word (i.e., the states of the device flags) are displayed in the REGISTER indicators (on the console) at this time.

I/O Overflow

I/O overflow is issued during the first cycle of the block transfer operation if the contents (2's complement) of the word counter assigned to the currently active channel device becomes 0 when incremented. This indicates that the program-specified number of words will have been transferred at completion of the channel transfer in progress. It is normally used to turn off the device, thereby preventing further channel action by that device until a service subroutine reinitializes the channel's word counter and



*will cause a program interrupt

†Inclusive OR of transfer completion and error Flags

*††Inclusive OR of MTF and EF

** Will cause an interrupt via the RDR Flag

15-0202

Figure 3-10 I/O Read Status Bit Assignments

Current Address Registers, and the program turns on the device request flag. The overflow signal may also be used to initiate a program interrupt through the program interrupt or automatic priority interrupt facility for access to the initializing subroutine. Additionally, I/O overflow occurs when an I/O increment memory operation causes the location to overflow.

Data Overflow

Data overflow is issued during the third cycle of a four-cycle add-to-memory data transfer when the addition operation generates an arithmetic overflow.

Input Control Levels

Six input control level signals arrive at the I/O processor:

Program Interrupt Request – A device delivers this signal to request interruption of the program in progress. The program traps to location 00000 when no I/O transfer action of higher priority is in progress. The instruction resident in location 00001 is fetched and executed. If more than one device is connected to the program interrupt, this instruction transfers control to a subroutine which determines through a search process (skip chain) the device making the program interrupt request. The appropriate service routine is then accessed.

Skip Request – The return of this signal to the processor indicates that an IOT instruction test for a skip condition in a selected device has been satisfied (e.g., a test of ready status). The program counter is subsequently incremented by one to effect a skip of the next instruction of the program in progress.

Read Request – This signal is used by the device to specify to the I/O processor that an input to the CPU data transfer is required.

Write Request – This signal requests that the processor execute a data-channel-write transfer of a data word into the selected device's Information register.

MB Increment – This level requests that the processor increment (by one) the contents of the memory location address specified by the 15-bit address on the I/O bus address lines. Used in increment memory operation.

Current Address Inhibit – This is a special signal line required by devices which automatically search for records, etc; typical are DECtape and magnetic tape. The presence of this signal level inhibits normal incrementing of the device assigned Current Address register during a data channel transfer.

Device Selection Levels – Identification of the current instruction as an IOT causes the bit pattern placed in the CPU MI 6 through 13 at the fetch of the instruction to be bus-driven and sent via eight bus lines to device selection modules contained in the control logic for each device. These eight levels form a 6-bit device selection code and 2-bit subdevice or mode selection code.

Address Lines

Fifteen lines constitute an input bus for the devices which must deliver address data to the processor. There are three uses for the address bus:

a. When a device interfaced to the multilevel automatic priority interrupt system receives an I/O processor grant of its interrupt request, it delivers to the CPU a hardware-defined address, relating to its API channel assignment. This channel address indicates to the device's service routine the location of the unique transfer vector.

b. When a block transfer channel device receives a processor grant of its transfer request, it delivers to memory a hardware-defined address indicating the memory location of the assigned channel's Word Count register.

c. Single cycle delivers a 15-bit address plus two dual purpose lines for 17-bit 128K accessibility. The dual purpose lines are SKP REQ and WRT REQ.

Multiplexed Control Lines

Sixteen control lines serve as processor-to-device-control information paths, four for the block transfer channel facility and twelve for the priority levels on which the automatic priority interrupt system processes requests for service. Control lines are used in the following ways:

Request – a device transmits a service request to the processor via the appropriate request line. There are four automatic priority interrupt request lines (one for each level) and two data channel request lines (single cycle requests and multicycle requests).

Grant – the processor indicates a grant of the service request. There are four API grant lines (one for each hardware level) and one data channel grant line to answer both single and multicycle requests.

Enable – the enable signals control the priority order for answering service requests of devices interfaced to the block transfer data channels or to one of the API's device channels. Priority for a channel (data or API) is allocated in descending order from the device nearest the processor I/O bus interface. An enable signal permits servicing of the requesting device with the highest priority and inhibits all lower priority devices from making requests during the interval of service. The enable signals are the only I/O signals that are actually broken and regenerated by each device on the associated channel.

I/O Run

The I/O Run signal is available at the interface for use as the interface designer requires. This bus driven level switches to the +2.5 V level and remains there while the I/O RUN flip-flop in the CPU is set. A ground level indicates that the RUN flip-flop has been cleared, and all operations in the CPU have been stopped.

Teletype Interface and Hardware Read-In

The I/O processor includes a Teletype control and Teletype unit as standard input/output equipment. Teletype Models 33 and 35, KSR or ASR, will operate with this controller.

The Teletype is capable of inputting and outputting at a rate of 10 characters per second. Serial transmission and reception of an 8-level character code is over a 4-wire cable connecting the Teletype and the control, which is located in the I/O processor.

For the ASR units, the reader and keyboard are electrically tied together and the punch and printer are mechanically connected. Teletype input functions are logically separated and therefore the keyboard/reader and printer/punch may be considered as individual devices. The program must echo any character from the keyboard it wants printed.

Hardware controlled read-in facilities are provided for reading paper tape into memory via either the ASR or the Type PC15 High-Speed Reader. This method of reading paper tape is accomplished by placing the tape into the reader and pressing the READ-IN key on the console. A hardware program wired into the I/O processor will read the tape into memory. Control is then automatically transferred to the beginning of the read-in program to initiate execution.

Priority

In view of the autonomous substructures of the PDP-15, three types of priority must be con-

sidered: memory access priority, priority on the common I/O bus, and priority on the use of the CPU.

Memory Access – The I/O processor always has priority over the CPU in accessing memory. However, once a CPU memory request has been granted, it will be allowed to complete its cycle before control can be returned to the I/O processor.

Common I/O Bus – Priority on the I/O bus is of concern only when more than one device is transferring information on the I/O bus and the I/O bus requests are received by the I/O processor. The following order of priority occurs:

a. Block Data Transfer Channels – The eight block data transfer channels range in priority from channel 1, which has the highest priority to channel 8, which has the lowest. Normally, single cycle block transfer devices are placed on the high priority channels to give them preference over the multicycle devices. However, if the data transfer rate of a single cycle device is not critical, it can be placed on a lower priority channel to give preference to both single- and multicycle devices. For example, a display processor might be placed on a low priority channel, since a temporary delay of data during the refresh cycle is not critical. The device's priority level and position on the I/O bus establishes its priority.

b. Real-Time Clock – The real-time clock has priority after the block data transfer channels. The real-time clock utilizes the I/O processor to fetch the contents of a reserve core memory cell (000007_8), increment the count, and then restore the new count.

c. Automatic Priority Interrupt – The automatic priority interrupt (API) system adds eight additional levels of priority to the PDP-15. The upper four levels are assigned to devices and are initiated by flags (interrupt requests) from these attached devices. The lower four levels are assigned to the programming system and are initiated by software requests. The priority network ensures that high data rate or critical devices will always interrupt slower device service routines while holding still lower priority interrupt requests off-line until they can be serviced. The API identifies the source of the interrupt directly, eliminating the need for a service routine to flag-search.

d. Program Interrupt – The program interrupt (PI) facility offers an efficient method of I/O servicing, if the API system is not used. The computer continues with execution of a program until a previously selected peripheral device signals that it is ready. At that time, the program in process interrupts and transfers control to a service subroutine. When completed, the subroutine restores the computer to the status existing prior to the interrupt, allowing the interrupted program segment to continue. Where multiple peripherals are connected to the PI, a search routine with device status testing (skipping) instructions must be added to determine which device initiated the interrupt request. The priority is established by the program skip routine.

e. Program – Controlled transfers at the main program level.

CPU – Priority on the use of the CPU is established by the API level, the program interrupt, and the main program, in that order.

Latency

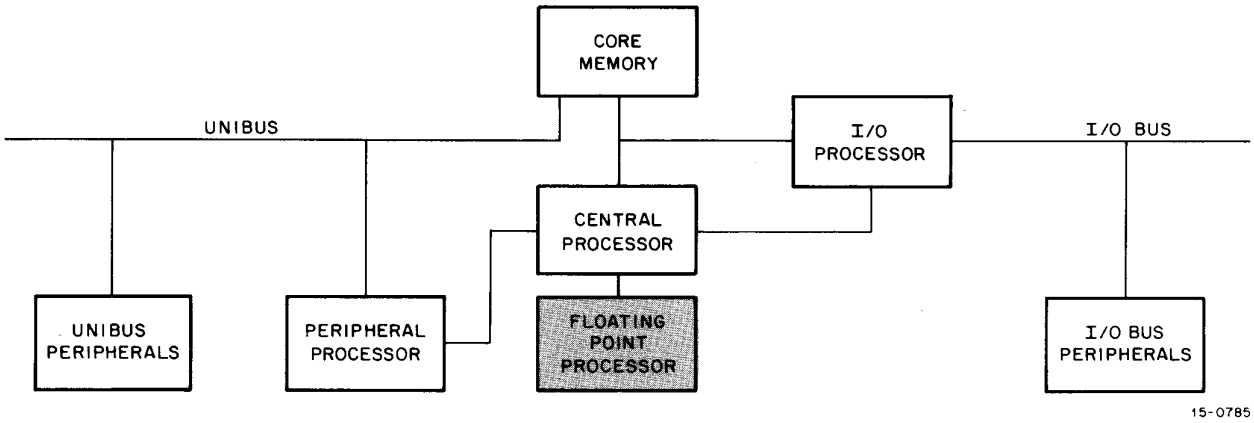
I/O transfer latency is a measure of the time between a device's request for service and the actual performance of that service. Regardless of a device's priority, once its request for service

has been granted, the I/O processor holds all other devices off until the current service request is complete. For example, a single cycle device on data channel 1 requesting service just after the initiation of a multicycle out transfer would have to wait for four I/O processor cycles before using the I/O bus. Finally, clock synchronization can take additional time resulting in a worst case latency of less than 8.5 μs for the requesting single cycle device. Table 3-1 lists typical transfer rates for each type of data channel transfer.

Table 3-1
Data Channel Transfer Rates

Type of Transfer	Frequency			
	Basic 8K	Basic 32K	KM/KT	MX 128K
3 Cycle Input	250 kHz	250 kHz	200 kHz	164 kHz
3 Cycle Output	181 kHz	181 kHz	161 kHz	147 kHz
Increment Memory	500 kHz	333–500 kHz	333 kHz	333 kHz
Single Cycle Out	1 MHz	1 MHz	830 kHz	675 kHz
Single Cycle In	1 MHz	1 MHz	880 kHz	750 kHz

CHAPTER 4
FLOATING POINT PROCESSOR FP15



FP15 FLOATING POINT PROCESSOR

FP15 operations consist of memory transfers to obtain or store data, and arithmetic calculations in the FP15 unit itself. The cycle time of memory, the central processor, and the I/O processor, as well as data channel latency are unaffected by the FP15 option. Data channel transfers to and from memory may occur simultaneously with FP15 arithmetic operations. Program and priority interrupts are inhibited and queued for priority-ordered response upon completion of the FP15 instruction. (The longest FP15 instruction takes 21 μs.)

A PDP-15 computer system was timed running five Fortran programs with and without floating point hardware. The results are as follows:

Program Description	PDP-15	PDP-15 & FP15
One hundred iterations of the analysis of three body final states. A physics application program.	37.0 sec	3.0 sec.
A least squares fit of data to a straight line.	5.1 sec.	0.7 sec.
A matrix inversion.	12.0 sec.	5.0 sec.
A test of all floating point functions.	11.4 sec.	1.4 sec.
A Fourier transform program.	16.9 sec.	2.9 sec.

Functionally, the FP15 contains two operand registers. One of the operand registers, the floating point accumulator, consists of an 18-bit exponent register (EPA), a 35-bit mantissa register (FMA), and a 1-bit sigh register (A SIGN). An additional 35-bit register, designated FMQ, serves as an extension of the floating point accumulator for multiplying and dividing.

The second operand register comprises an 18-bit exponent register (EPB), a 35-bit mantissa register (FMB), and a 1-bit sign register (B SIGN). This second operand register serves as a temporary accumulator to hold the argument fetched from core.

Integer arguments do not use the exponent registers since they are merely integers and not floating point numbers. Basically, if two numbers are to be manipulated, one number is loaded in the floating point accumulator, the second number is loaded in the temporary accumulator, and the result is stored in the floating point accumulator.

FP15 FUNCTIONAL DESCRIPTION

Figure 4-1 is a simplified block diagram of the FP15 Floating Point Processor. The FP15 is in parallel with the CPU on the memory bus, and monitors each instruction fetched by the CPU from core. If bits 00 through 05 of the instruction are equal to 71_8, it is recognized as a floating point instruction; the CPU treats the instruction as an NOP. The FP15 takes control of memory, inhibits the CPU, and then simulates the CPU by completing the normal interface between CPU and memory. After the floating point instruction has been executed, the CPU is enabled, and both the CPU and FP15 are free to monitor the next instruction.

Functionally, the FP15 contains a Memory Buffer register and two operand registers. The Memory Buffer register provides temporary storage for all words transferred to the FP15. One operand register consists of an 18-bit exponent register (EPA), a 35-bit mantissa register (FMA), and a 1-bit sign register (A SIGN). This operand register is referred to as the floating point accumulator. An additional 35-bit register, designated FMQ, serves as an extension to the floating point accumulator.

A second operand register consists of an 18-bit exponent register (EPB), a 35-bit mantissa register (FMB), and a 1-bit sign register (B SIGN). This second operand register, EPB/B SIGN/ FMB, serves as a temporary accumulator to hold the argument fetched from core.

The exponent registers store the exponents associated with floating point numbers and are not used during integer operations. Basically, if two numbers (integer or floating point) are to be manipulated, one number is loaded in the floating point accumulator by a Load type instruction. The second number is normally loaded in the temporary accumulator [EPB (B SIGN) FMB] by an instruction specifying an arithmetic operation. Both numbers are gated into a 36-bit adder, where the arithmetic operation is performed. The result is then transferred to the floating point accumulator. The major registers are described below:

Memory Buffer Register – A 36-bit register that provides the FP15/memory interface. All data transferred into the FP pass through this register.

Adder – A 36-bit arithmetic logic unit (ALU) that serves as the central point in the FP15 and performs all arithmetic and logic operations. The output of the adder is connected to all major registers via an adder bus.

A SIGN – A 1-bit register used to store the polarity of the associated operand (A mantissa).

EPA – An 18-bit register used to store the 2's complement of the exponent associated with the mantissa loaded in the FMA. The most significant bit of the EPA represents the sign of the exponent; in single precision floating arithmetic, the most significant bit of the

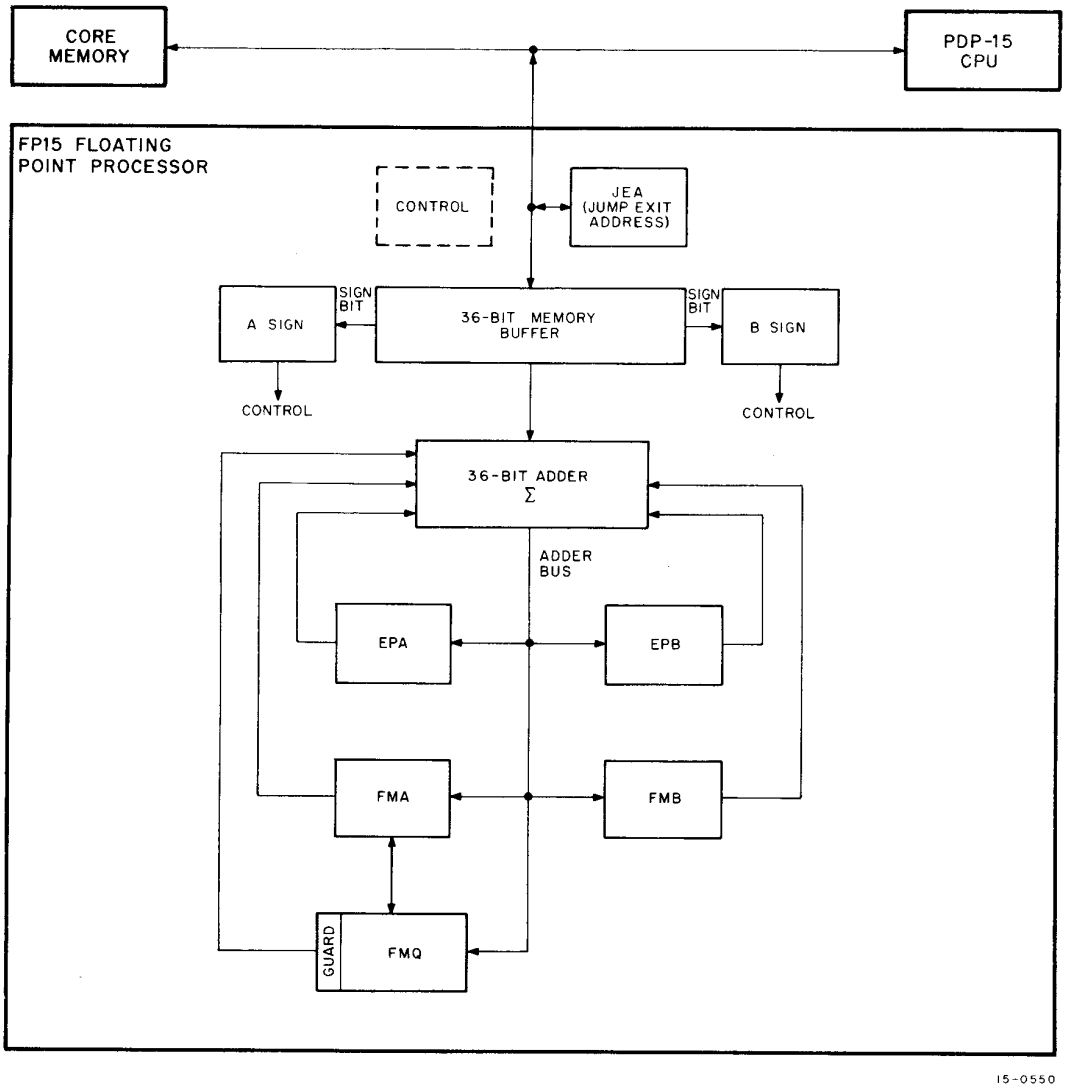


Figure 4-1 FP15 Simplified Diagram

exponent is bit 09. It is therefore necessary to extend the value of this bit from bits 00 through 08. If bit 09 is a 1, bits 00 through 08 in the EPA are forced to 1s, and if bit 09 is a 0, bits 00 through 08 in the EPA are forced to 0s. The EPA and FMA serve as the floating point accumulator.

FMA – A 35-bit register used to store the integer in integer arithmetic, or the mantissa in floating point arithmetic. The binary point is located between bit 00 and bit 01 of the FMA.

FMQ – A 35-bit extension of the FMA register used during multiplication and division operations.

B SIGN – A 1-bit register used to store the polarity of the associated operand (B mantissa).

EPB – An 18-bit register used to store the exponent associated with the mantissa in the FMB. The most significant bit of the EPB represents the sign of the exponent. In single precision arithmetic, where the most signi-

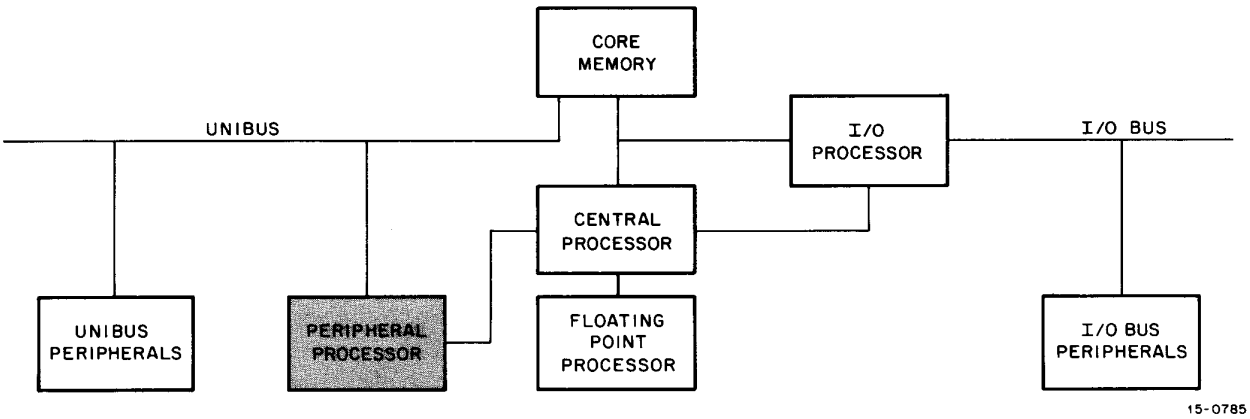
ficant bit in the EPB is bit 09, the value of this bit is extended to bits 00 through 08 (refer to EPA register). The EPB and FMB serve as a temporary accumulator to store the argument fetched from core. The EPB is a dynamic register and is therefore not directly accessible by software.

FMB – A 35-bit register used to store the integer in integer arithmetic or the mantissa argument in floating point arithmetic. The binary point is located between the most significant bit (bit 00) and bit 01 of the FMB. The FMB is a dynamic register and is therefore not directly accessible by software.

JEA (JMS Exit Address) – A 17-bit register used to store two status bits and a 15-bit base exit address for floating point interrupts. When an interrupt condition (overflow, underflow, abnormal division, or memory protect violation) occurs in the FP15, the base exit address (a unique address for each type of interrupt) is returned. This indicates a service routine associated with the interrupt. The guard bit is used in rounding operations.

For more information on the FP15, refer to the *FP15 Floating Point Processor Programmer's Reference Manual,* DEC-15-HQEA-D, and the *FP15 Floating Point Processor Maintenance Manual,* DEC-15-HQFA-D.

CHAPTER 5
PERIPHERAL PROCESSOR UC15



The Unichannel 15 System (UC15) consists of a peripheral processor that interfaces a PDP-15 Central Processor to the PDP-11 Unibus (Figure 5-1). The Unibus contains 18 address lines, various control lines, and 16 data lines. Two additional lines are incorporated, giving a total of 18 data lines, in order to be compatible with the PDP-15 18-bit processor. The PDP-15 functions as the master processor; the peripheral processor functions as a slave in carrying out the tasks initiated by the PDP-15. The peripheral processor will normally be a PDP-11/05 Central Processor. Actually, peripheral control occupies only a small part of the peripheral processor's time. The rest of this time can be used for parallel processing of tasks as conceived by the system designer (*Unichannel 15 Software Manual,* DEC-15-XUCMA-A). The peripheral processor runs its own program with 4K or 8K dedicated memory, hereafter referred to as local memory.

COMMON MEMORY

The Unichannel 15 System allows any Non-Processor Request (NPR) device on the Unibus to access PDP-15 memory so that data can be transferred between I/O devices and common memory.

The use of common memory allows ease of data transfer between common memory and secondary storage (disk, magnetic tape, etc.). The peripheral processor can operate with a maximum 28K of common memory if no relocation option is employed. If local memory contains 4K, the peripheral processor can address the lowest 24K of common memory. With 8K of local memory, the peripheral processor can address only the lowest 20K of common memory. The Unibus can address the combined PDP-15/PDP-11 memory, which can extend up to 124K.

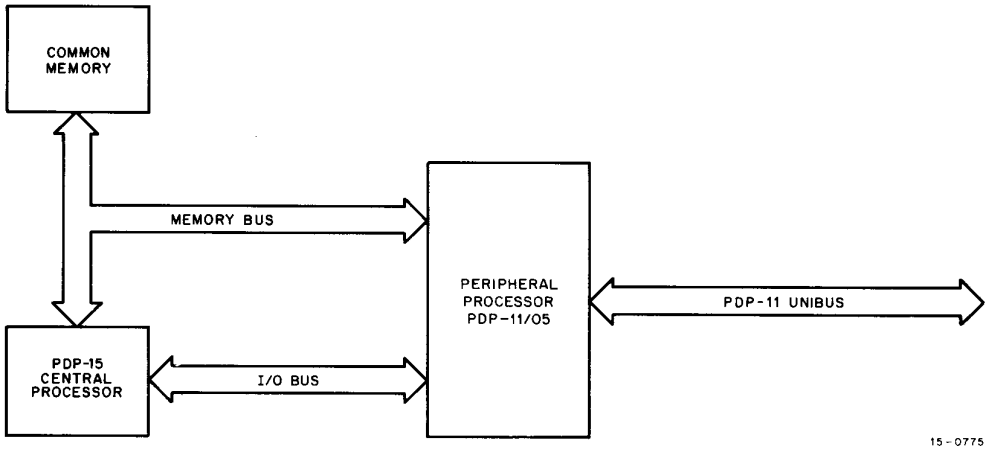


Figure 5-1 UC15 System Concept

INTERRUPT LINK

The PDP-15 and the peripheral processor communicate with each other through device interfaces. The PDP-15 functions as a master processor by initiating and defining tasks, while the peripheral processor functions as a slave device. When the PDP-15 initiates a new task, it interrupts the peripheral processor with a message. The message is designated as a Task Control Block Pointer (TCBP) and points to a table (Task Control Block) in common memory where the task is defined. The peripheral processor performs the task and signifies its completion by sending an interrupt back to the PDP-15 (Figure 5-2).

PERIPHERAL PROCESSOR HARDWARE

The UC15 System, in its standard configuration, consists of the following equipment (Figure 5-3):

- PDP-11 peripheral processor
- DR15-C Device Interface
- Two DR11-C Device Interfaces
- MX15-B Memory Bus Multiplexer
- Local memory

NOTE

The PDP-11, which functions as the peripheral processor, can itself only process 16-bit words but controls peripherals that can process 18-bit words to provide compatibility with the PDP-15.

The DR15-C and the two DR11-C Device Interfaces provide the communication facility between the PDP-15 and the PDP-11. The PDP-15 can interrupt the PDP-11 and send data words to the PDP-11; however, the PDP-11, serving as a peripheral processor, can only interrupt the PDP-15 to indicate an error condition or job completion.

The MX15-B Memory Bus Multiplexer functions as a memory bus switch to allow either the PDP-15 or the PDP-11 to communicate with the common memory.

UC15 CONFIGURATIONS

In the standard UC15 configuration, the peripheral processor is a KD11-B Processor (PDP-11 family) with a 1 μs cycle time and 4K of local memory. Several variations are possible.

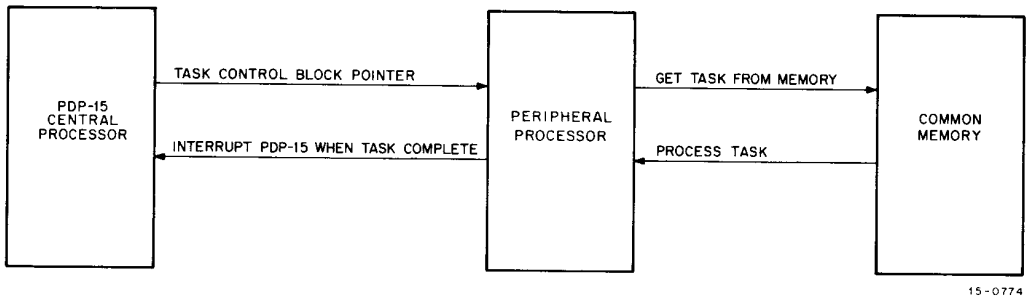


Figure 5-2 Simplified Flow Diagram

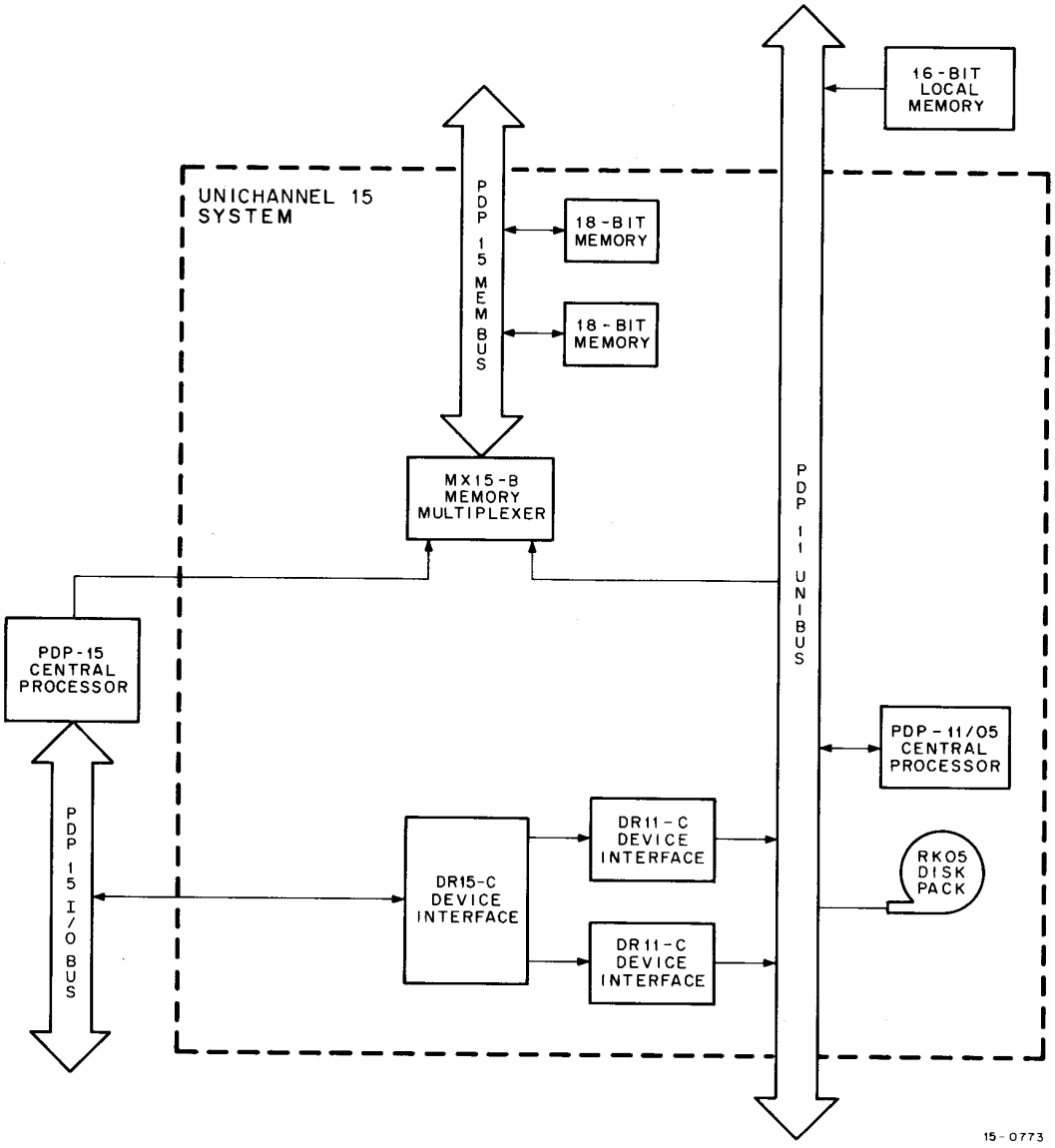


Figure 5-3 UC15 Hardware

A different PDP-11 processor with a faster or slower cycle time can be utilized. However, it would be available only through Computer Special Systems.

Another possible variation is that any Unibus peripheral can be connected into the system.

Sixteen-bit direct memory access (DMA) devices, called NPR devices on the Unibus, can transfer only 16 bits of data to and from memory (the upper two of the 18 bits being defined as 0s). Eighteen-bit NPR Unibus devices, such as the RK15 DECdisk Pack, will transfer the full 18-bit data words to and from common memory.

Byte-oriented Unibus devices, such as DECtape, DECassette, paper tape, magtape, line printers, and communications interfaces, transfer data under control of the PDP-11 peripheral processor, which is limited to 16 bits. Hence, format conversion for these devices from 16-bit or 8-bit formats to 18 bits (and vice versa) must be done by the PDP-15.

SPECIFICATION SUMMARY

- Sum of PDP-11 and PDP-15 memory limited to 124K.
- Normal configuration includes 4K of local memory, allowing 120K of shared memory.
- PDP-11, with 4K of local memory, can address only lowest 24K of common memory to access (1) task control blocks set up the PDP-15, and (2) data for byte-oriented Unibus devices.
- Normal PDP-11/05 processor gives an NPR break a worst-case latency of 7 μs. Total worst-case latency time is 12 μs (7 μs NPR latency plus 5.0 μs for PDP-15 to do three I/O memory cycles, with addition of MX15-B multiplexer).
- Addressable Registers

  DR11-C/#0

CSR (API DONE, ENABLE API DONE INTR)	767770
Output Data Buffer (API 1, API 0 Address)	767772
Input Data Buffer (API 0, 1, 2, 3 DONE, Upper 2 bits of TCBP, LMS 0, 1)	767774

  Interrupt Vector – 300

  Priority Level – BR5

  DR11-C/#1

CSR (new TCBP, ENABLE TCBP INTR)	767760
Output Data Buffer (API 2, API 3 Address)	767762
Input Data Buffer (TCBP)	767764

  Interrupt Vector – 310

  Priority Level – BR7

  DR15-C 18-bit Data Register (loaded by LIOR IOT)

- Task Initiation

  Task Control Block Pointer

  PDP-15 LIOR IOT (706006) – clear flag, load TCBP, and interrupt PDP-11

  PDP-11 interrupt vector 310 at priority level 7

- Task Completion

  PDP-11 loads one of the following four bytes with an API address:

767772	API level 0
767773	API level 1
767762	API level 2
767763	API level 3

  API logic interrupts PDP-15 at API address. If API (Automatic Priority Interrupt) option is not installed, a PI (program interrupt) is created with four skip IOTs for decoding.

- Bus Loading

  MX15-B

  2 PDP-15 memory bus loads
  Drives 4 PDP-15 memory bus loads

  DR15-C/DR11-C

  1 Unibus load
  1 PDP-15 I/O bus load

- Power

DEC Channel 15	7 A at 115 V
(With RK15 DECdisk)	3.5 A at 230 V

- Voltage 115 Vac ± 10%, 230 Vac ± 10%

- Frequency 50 ± 2 Hz, 60 ± 2 Hz

- Environmental

Temperature	10° to 50° C
Relative Humidity	20 to 95%

- Installation

  UC15 hardware includes a UC15 cabinet and space for mounting PDP-11 peripherals. There are two spaces 10-1/2 in. high and one space 5-1/4 in. high. In addition, mounting space for two small peripheral controllers (SPC) exists within the PDP-11/05 processor.

- UC15 Cabinet Dimensions

Depth:	30 in. (0.76 m)
Width:	21 in. (0.53 m)
Height:	72 in. (1.83 m)
Weight:	250 lb (115 kg) – not including peripherals

- Unibus Compatibility

  Can be used with any PDP-11 family processor that does not use parity. On those systems with parity, the parity must be disabled.

- Memory Cycle

  MX15-B normally adds 200 ns to both the PDP-15 and the PDP-11 cycle times.

- DMA Facility to Common Memory

Maximum Transfer Rate	415K words/sec
Worst-Case Latency	6 μs (no DCH transfers in PDP-15) 12 μs (DCH transfers in PDP-15)
Average Latency	2.5 μs

- DMA Facility to PDP-11/05 Local Memory

Maximum Transfer Rate	1 million words/sec
Worst-Case Latency	7.2 μs
Average Latency	2.5 μs

For more information, refer to the *UC15 Unichannel 15 System Maintenance Manual,* DEC-15-HUCMA-B-D.

THE UNIBUS AND THE PDP-11/05 PROCESSOR

The Unibus

All the PDP-11/05 computer system components and peripherals connect to and communicate with each other on a single high-speed bus, the Unibus. All elements of the PDP-11/05 system, including the central processor, communicate with each other in identical fashion via the Unibus; thus, the processor can easily access both peripherals and memory (Figure 5-4).

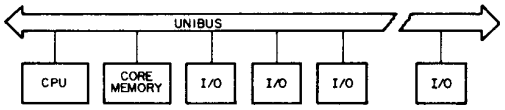


Figure 5-4 PDP-11 System, Simplified Block Diagram

With bidirectional and asynchronous communications on the Unibus, devices can send, receive, and exchange data independently without processor intervention. Because it is asynchronous, the Unibus is compatible with devices operating over a wide range of speeds.

Device communications on the Unibus are interlocked. For each command issued by a "master" device, a response signal is received from a "slave" completing the data transfer. Device to device communication is completely independent of physical bus length and the response times of master and slave devices. Interfaces to the Unibus are not time dependent, and there are no pulse width or rise time restrictions. The maximum rate on the Unibus is one word every 400 ns or 2,500,000 words per second.

I/O devices transferring directly to or from memory are given highest priority and may request bus mastership and steal bus and memory cycles during instruction operations. The processor resumes operation immediately after memory transfer. Multiple devices can operate simultaneously at maximum direct memory access (DMA) rates by "stealing" bus cycles.

PDP-11/05 Processor

Central Processor

The central processor, connected to the Unibus as a subsystem, controls the time allocation of the Unibus for peripherals and performs arithmetic and logic operations and instruction decoding. It contains multiple high speed general purpose registers which can be used as accumulators, address pointers, Index registers, and other specialized functions. The processor can perform data transfers directly between I/O devices and memory without disturbing the processor registers; and it performs both single and double operand addressing and handles both 16-bit word and 8-bit byte data.

Instruction Set

The instruction complement uses the flexibility of the general purpose registers to provide over 400 powerful hard-wired instructions – the

most comprehensive and powerful instruction repertoire of any computer in the 16-bit class. Unlike conventional 16-bit computers, which usually have three classes of instructions (memory reference instructions, operate or AC control instructions, and I/O instructions), all operations in the PDP-11/05 are accomplished with one set of instructions. Since peripheral device registers can be manipulated as flexibly as core memory by the central processor, instructions that are used to manipulate data in core memory may be used equally well for data in peripheral device registers. For example, data in an external device register can be tested or modified directly by the CPU, without bringing it into memory or disturbing the general registers. One can add data directly to a peripheral device register, or compare logically or arithmetically contents with a mask and branch. Thus, all PDP-11/05 instructions can be used to create a new dimension in the treatment of computer I/O and the need for a special class of I/O instructions is eliminated.

Priority Interrupts

A multilevel automatic priority interrupt system permits the processor to respond automatically to conditions outside the system. Any number of separate devices can be attached to each level.

Each peripheral device in the PDP-11/05 system has a hardware pointer to its own pair of memory words (one points to the device's service routine, and the other contains the new processor status information). This unique identification eliminates the need for polling devices to identify an interrupt, since the interrupt servicing hardware selects and begins executing the appropriate service routine after having automatically saved the status of the interrupted program segment.

The device's interrupt priority and service routine priority are independent. This allows adjustment of system behavior in response to real-time conditions, by dynamically changing the priority level of the service routine.

The interrupt system allows the processor to continually compare its own programmable priority with the priority of any interrupting devices and to acknowledge the device with the highest level above the processor's priority level. Servicing an interrupt for a device can be interrupted for servicing a higher priority device. Service to the lower priority device is resumed automatically upon completion of the higher level servicing. Such a process, called nested interrupt servicing, can be carried out to any level without requiring the software to save and restore processor status at each level.

Reentrant Code

Both the interrupt handling hardware and the subroutine call hardware facilitate writing reentrant code for the PDP-11/05. This type of code allows a single copy of a given subroutine or program to be shared by more than one process or task. This reduces the amount of core needed for multitask applications such as the concurrent servicing of many peripheral devices.

Addressing

Much of the power of the PDP-11/05 is derived from its wide range of addressing capabilities. PDP-11/05 addressing modes include sequential addressing forward or backward, address indexing, indirect addressing, 16-bit word addressing, 8-bit byte addressing, and stack addressing. Variable length instruction formatting allows a minimum number of bits to be used for each addressing mode. This results in efficient use of program storage space.

Stacks

In the PDP-11/05, a stack is a temporary data storage area which allows a program to make efficient use of frequently accessed data. The stack is used automatically by program interrupts, subroutine calls, and trap instructions. When the processor is interrupted, the central processor status word and the program counter are saved (pushed) onto the stack area, while the processor services the interrupting device. A new

status word is then automatically acquired from an area in core memory that is reserved for interrupt instructions (vector area). A return from the interrupt instruction restores the original processor status and returns to the interrupted program without software intervention.

Direct Memory Access

All PDP-11/05's provide for direct access to memory. Any number of DMA devices may be attached to the Unibus. Maximum priority is given to DMA devices, thus allowing memory data storage or retrieval at memory cycle speeds. Latency is minimized by the organization and logic of the Unibus, which samples requests and priorities in parallel with data transfers.

Power Fail and Restart

The PDP-11/05's power fail and restart system not only protects memory when power fails, but also allows the user to save the existing program location and status (including all dynamic registers), thus preventing harm to devices, and eliminating the need for reloading programs. Automatic restart is accomplished when power returns to safe operating levels, enabling remote or unattended operations of PDP-11/05 systems. All standard peripherals in the PDP-11/05 family are included in the systemized power-fail protect/restart feature.

For more information, refer to the *PDP-11/05, 11/10 Computer Manual,* (DEC-11-H05AA-B-D).

PDP-11/05 SPECIFICATIONS

Physical

- Dimensions:
  5-1/4 in. high, 19 in. wide, 20 in. deep
  (13.3 cm X 48.3 cm X 50.8 cm)

- Weight:
  65 lb (29.4 kg)

Electrical

- Processor power requirements:
  90–135 V, 47–63 Hz
  (180–270 V model available)

  Unibus Logic Levels:
  Ground and +3 V

  Internal circuit potentials:
  5 V, - 15 V, +15 V

  Logic: Fully integrated TTL and MSI

  Power: 325 W

  Current: 5 A, 115 V; 2.5 A, 230 V

Functional

Word length:
16 bits

Direct Memory Access:

Memory Cycle Time	Unibus Rate w/s	DMA Rate w/s
900 ns	2,500,000	1,100,000

- Multiple device capability without multiplexer

- Calculated MTBF

Central Processor	38,900 hours
Power Supply	31,201 hours
Operator's Console	82,309 hours
Memory (8K)	25,200 hours

- Unibus Data Rate: 2,500,000 words/second

- 4 Levels Automatic Priority Interrupt

- Power fail and Restart

- General Registers:

  Eight high-speed flip-flop registers within central processor.

  Used as accumulators, 16-bit index registers and autoincrement or autodecrement registers. All registers may serve as stack pointers. Register 6 is used as the processor stack pointer. Register 7 is the program counter.

- Instructions:

  Over 400 hard-wired instructions through use of general register address modes.

  Machine directly byte and word addressable to 65,576 bytes or 32,768 words.

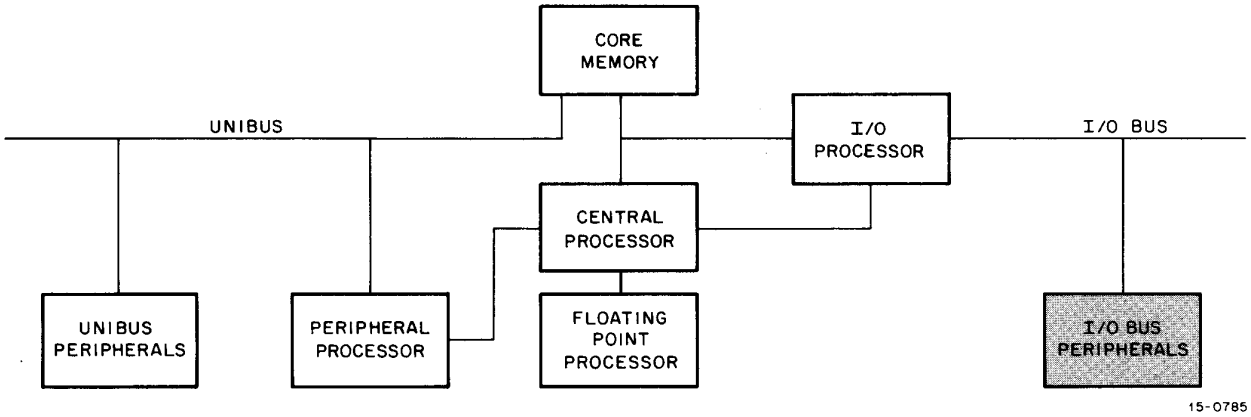
- Environmental (processor):

  Temperature: 0° to 55° C
  Humidity: 10–95% (noncondensing)

- Peripherals:

  Standard PDP-11 family peripherals operate with the PDP-11/05. These include: Teletypes, high-speed paper-tape reader/punch, high-speed line printer, card reader, DECtape and industry compatible magnetic tape units, rotating disk memories, storage displays, oscilloscopes, plotters and point-plot displays.

CHAPTER 6
INPUT/OUTPUT BUS PERIPHERALS



A wide range of peripheral equipment is available on the I/O bus to expand the capabilities of all PDP-15 systems.

STANDARD INPUT/OUTPUT DEVICES

PC15 High-Speed Paper Tape Reader/Punch

The perforated paper tape reader can photoelectrically sense 8-channel paper tape at a rate of 300 characters-per-second. Under program control, data may be read in either alphanumeric (one character) or binary (three character) mode. The use of a paper tape reader buffer and buffer-full flag permits the continuation of processing during the reading functions.

The 50 character-per-second paper tape punch is mounted on the same chassis as the reader. A single output instruction causes an 8-bit character to be transferred from the PDP-15 accumulator to a punch buffer, from which it is punched on the tape. Fanfold paper tape is normally used with the paper tape reader and punch.

CR15 Card Reader and Control

The CR15 reads standard 80-column cards at rates up to 1000 cpm. Interfaced to the PDP-15 multicycle data channel facility, the CR15 transfers card data directly to memory via the PDP-15 I/O processor. This capability allows the system to handle tasks that require overlap between input/output and CPU processing.

Card Reader Specifications

	CR15DA* and CR15DB TABLE TOP	CR15FA* and CR15FB TABLE TOP
Reading Rate (cpm)	1000	300
Input Hopper Capacity (cards)	950–1,000**	550–600**
Output Hopper Capacity (cards)	950–1,000**	550–600**
Size Envelope (H X W X D)		
inches	17 X 24 X 19	13 X 20 X 15
centimeters	43 X 61 X 48	33 X 20 X 38
Weight (max.)		
pounds	100	70
kilograms	220	154
Power Consumption (VA)		
starting	1500	1500
running	460	460
Heat Dissipation		
BTU/hour	1600	1600

* A suffix denotes 60 Hz
B suffix denotes 50 Hz
**Depending on card stock

MASS STORAGE DEVICES

DECtape

The DECtape system provides a unique fixed-address magnetic tape facility for program and data storage and retrieval. In environments where long batch processing queues are not frequently required, DECtape is an excellent substitute for punched cards. A single pocket-sized reel of DECtape can store 150,000 18-bit words; more information than a deck of 5,000 cards. At the same time, it costs considerably less than a reel of IBM-compatible magnetic tape. This compact, inexpensive format allows each user to have his own personal library of programs and data files on a pocket-sized reel, easily mounted on the transport.

DECtape Features

- Stores information at fixed positions on magnetic tape, rather than at unknown or variable positions, as is done on conventional magnetic tape systems. This feature allows replacement of blocks of data on

tape in a random fashion without disturbing other previously recorded data blocks.

- Automatic word transfer via the PDP-15 data channel facility to allow concurrent processing and data transfer.

- Bidirectional operation to allow reading, writing, and searching in either direction.

- Redundant phase recording to ensure transfer reliability, reduce the problem of skewing, and minimize bit dropouts.

- Prerecorded timing and mark tracks to determine the exact position at which to record the information to be written. The same mark and timing information is used to locate data to be read from tape.

TC15 DECtape Control – The TC15 DECtape Control controls up to eight DECtape transports or four TU56 Dual DECtape transports. Binary information is transferred to and from the PDP-15 at the rate of one 18-bit word every 200 μs, using the data channel facility. Mode of operation, function, and direction of motion are controlled by status registers, which can be loaded and read by the computer program.

TU56 Dual DECtape Transport – The TU56 Dual DECtape Transport provides two logically-independent DECtape drives capable of bidirectional reading and writing of DECtape reels. Each 3-in. diameter reel can hold 3,000,000 bits of information (over 150,000 18-bit words) recorded at 375 bpi. Tape moves at 80 ips and requires no vacuum columns, pinch rollers, or capstans.

Magnetic Tape Systems

PDP-15 offers both 7- and 9-channel IBM-compatible magnetic tape systems. Transports are currently available to operate at either 45 or 75 ips and at any of three recording densities, 200, 556, or 800 bpi.

TC59D Magnetic Tape Transport Control – The TC59D Magnetic Tape Transport Control transfers data to and from IBM-compatible transports via the data channel facility. Up to eight transports can be handled by a single control, and both BCD and binary modes are available. One TC59 control can handle both 7- and 9-channel transports at both 45 and 75 ips. Read/write functions, recording density, and tape manipulation functions are controlled by status registers, which can be loaded and read by the PDP-15.

TU10 Magnetic Tape Transports – The TU10-FE, -FJ Magnetic Tape Transport can read and write 7-channel IBM-compatible tapes at 45 ips and 200, 556, or 800 bpi. One 18-bit PDP-15 word is written as three tape characters on the TU10.

Its 9-channel counterpart, the TU10-EE, -EJ Magnetic Tape Transport operates at the same speed at 800 bpi. In two-character mode, the TU10-EE, -EJ reads or writes two 8-bit characters per 18-bit word (ignoring two bits), while in three-character mode, it reads or writes three 6-bit characters (one PDP-15 word) as three 8-bit tape characters.

Disk Systems

RF15/RS09 DECdisk – The fixed head RF15/RS09 Disk System is a mass storage medium for the PDP-15. The basic system consists of one RF15 DECdisk Control and one RS09 DECdisk Drive, providing mass storage capacity of 262,144 18-bit words. Seven additional disk units can be accommodated by the control unit, increasing storage capacity to 2,097,152 words. Data transfers ranging from one word to total capacity are performed via the multicycle data block transfer channel facility. Addressing is by disk number, track, and word.

Format Specifications

Storage Capacity (18-bit words) –	262,144
Expandable to –	2,097,152
Number of Tracks	128
Words Per Track	2,048

Timing Specifications	Power	
	60 Hz	50 Hz
Average Access Time	16.7 ms	20.0 ms
Minimum Access Time	250 μs	250 μs
Worst-Case Access Time	33.3 ms	40.0 ms
Word-Transfer Rate (words/second)		
High	62.5K	50K
Medium	31.2K	25K
Low	15.6K	12.5K

RP15/RP02 Disk Pack and Control – The RP15 Disk Pack Control interfaces the RP02 Disk Pack Drive Unit to the PDP-15. Transfers are made through a single cycle block transfer data channel. Up to eight RP02 drives can be handled by the same control. The total capacity of an RP02 drive is 10.24 million words; the total bulk storage capacity with eight RP02 drives is 81,920,000 18-bit words. Average transfer rate with the RP02 drive is 135K words/second. Average access time, including the rotational latency time of 12.5 ms, is 62.5 ms.

LINE PRINTERS AND PLOTTERS

LP15 Automatic Line Printer

The LP15 Line Printer is available in two models: LP15-F and LP15-J. The LP15-F prints 356 to 1110* lines per minute, has 80 columns, and has a 64 character set. The LP15-J prints 245 to 1110* lines per minute, has 132 columns, and has a 64 character set.

XY15 Incremental Plotter and Control

The Type XY15 Incremental Plotter and Control uses CalComp Model 563 or 565 plotters at rates of 12,000 or 18,000 points-per-minute. Paper width is either 31 in. (Model 563) or 12 0.005 and 0.01 in. are available.

*Minimum speed based in printing all columns available. Printers operate at higher speeds when printing particular lines.

DATA COMMUNICATIONS DEVICES

LT19 Multi-Station Teletype Control

The Multi-Station Teletype Control is available to interface from one to five communications terminals to the PDP-15 central processor (Figure 6-1).

LT19D Teletype Control – The LT19D can control up to five LT19E Teletype Line Units.

LT19E Line Units – LT19E Line Units are full-duplex Teletype interfaces for either KSR or ASR models: each LT19E unit handles one Teletype or LT19 EIA Adapter.

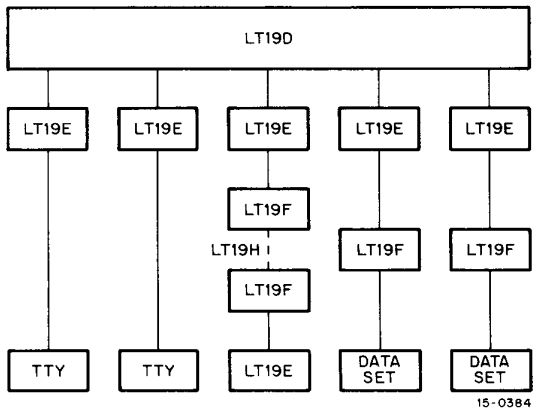


Figure 6-1 Multi-Station Teletype Control

LT19F EIA Adapters – The addition of the LT19F EIA Adapter to an LT19E unit allows the line unit to communicate with low-speed data sets.

LT15 Teletype Control

The LT15-A is a single Teletype control that interfaces a second Teletype-like device to the PDP-15 in addition to the console printer.

Console Printer and Control

Available as the console device are the LA30, KSR 35, and the ASR 33. The KSR 35 and ASR 33 type-in or print-out information at rates up

to 10 cps. The keyboard control includes an 8-bit buffer to hold the last character (ASCII code) struck on the keyboard and a flag to signal the processor of the presence of a character, while the printer control contains an 8-bit buffer to hold one character while it is being printed.

LA30 DECwriter

The LA30 DECwriter is a fast, reliable, low-cost data terminal. The unit prints from a set of 64 characters at speeds of up to 30 characters per second. Data entry is made from either a 96- or 128-character keyboard.

The DECwriter makes a hard-copy original plus one copy on a standard 9-7/8 in. wide, tractor-driven continuous form. Interfaces are available for the entire line of Digital PDP computers.

The DECwriter data terminal comes complete with an attractive stand, and the noise level generated by the terminal is less than an electric typewriter. The basic low-price of the DECwriter makes it uniquely appropriate for systems requiring large numbers of highly reliable printer/terminals.

Specifications

Printer

Printing Speed:	30 characters per second, asynchronous; 300 ms carriage return
Line Length:	80 character positions
Character Spacing:	10 characters per inch
Line Spacing:	6 lines per inch
Paper:	9-7/8 in. wide tractor-driven continuous form original plus one copy
Typeface:	5 X 7 dot matrix
Interface:	Available for all DEC computers
Ribbon:	1/2-inch X 120 feet, nylon

Data Entry

Code:	USASCII-1968 96 characters (128 optional)
Interface:	Available for all DEC computers

Environmental/Physical

Temperature:	50° – 100° F
Humidity:	5–90% (noncondensing)
Power:	Type LA30-A: 115 Vac, 50–60 Hz Type LA30-B: 230 Vac, 50–60 Hz
Dimensions:	20-1/2 in. wide X 31 in. high X 24 in. deep

DP09A Data Communication System

The bit synchronous DP09A Data Communication System provides interface facilities between a PDP-15 and a bit-serial communication device. The DP09A serializes the characters for transmission, and assembles the serial stream into characters for reception. Operation is full duplex. Both the receive and transmit sections are double-buffered to permit one full character transmission time for loading or reading the DP09A.

GT15 Graphics Subsystem

The GT15 Graphics Subsystem consists of four items: VT15 Display Processor, VV15 Arbitrary Vector Generator, VT04 17 in. Display Console (or VT07 21 in. Display Console), and a light pen.

The GT15 utilizes a refresh CRT. The refresh-tube regenerates the information contained in the display buffer continually, typically 30 to

60 times per second. One of the functions of the display controller is to handle the rapid image regeneration. The advantages of this procedure are that it permits the user to see changes in the display (such as rotation or motion simulation) occur very smoothly, and it permits the user to interact dynamically with the display by, for example, drawing on the scope face.

VT15 Graphic Processor

The VT15 Graphic Processor converts digital instruction inputs from the PDP-15 computer into analog signals that drive X- and Y- axis deflection circuits of the display CRT (Figure 6-2). Because the PDP-15 computer and VT15 Graphic Processor share the same core memory, their programs interact through the hardware. Further, their programs interact to the extent that the PDP-15 computer program can directly modify the VT15 Graphic Processor program, which is called the display file.

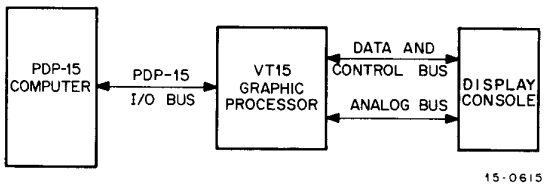


Figure 6-2 Basic VT15 Graphic Display System, Block Diagram

The VT15 Graphic Processor contains a number of registers for central processor-memory communications, a program counter, an accumulator, an Index register, and a Limit register. Two 18-bit registers are used for memory buffer functions which permits overlap for memory cycle times and allows faster instruction execution times. The VT15 has a set of eight basic machine language instructions that give the Graphic 15 System the utmost versatility in the display of points, basic vectors, graph plots, and ASCII characters.

Vectors are drawn on the scope by use of a "stroke vector" technique for maximum drawing speed. Up to four display scopes can be controlled by the VT15 via a VM15 display multiplexer.

Display Processor Specifications

Virtual Paper Size:	12 bits X 12 bits
Screen Display Size:	10 bits X 10 bits
Scales:	4-bit increment register (0–15) characters and vectors
Brightness:	3-bit register (8 levels)
Line Types:	4–solid and 3 types of broken lines
Vector Specification:	Relative
Point Specification:	Absolute
Name Register:	6 bits (228 values)
Synchronization:	Display refresh rate can be synchronized to line frequency or harmonics
Characters:	64 printing characters
	4 special (alt mode [ESC], CR, LF, TAB)

The VV15 Arbitrary Vector Generator enables vectors to be drawn at any specified angle at maximum speeds. The VV15 speeds up display and simplifies memory storage space.

DISPLAY SCOPES

The VT04 and VT07 are rectangular self-contained CRT monitors with CRT power supplies, deflection amplifiers, and six console pushbuttons which generate program interrupts. Provision for implementing a light pen, writing tablet, and keyboard options is also included.

Technical Specifications

	VT07	VT04
Screen Size	21 inches diagonal	17 inches diagonal
CRT Shape	Rectangular, 12 in. X 14 in.	Rectangular 9 in. X 10-1/2 in.
Major Display Area	12 in. X 12 in.	9 in. X 9 in.
Minor Display Area	2 in. X 12 in. (programmable)	1-1/2 in. X 9 in.
Vector Drawing Rate	0.33 in. per μs	.25 in. per μs
Flicker-free Presentation	10,000 vector inches	7,000 vector inches
Character Drawing Capability	2,800 maximum	Same
Spot Size	0.015 in. (within 10-in. diameter circle about CRT center)	Same
Spot Jitter	0.005 in.	Same
Display Drift	1% of full screen (over 24 hours)	Same
Spot Repeatability	± 0.020 in.	Same
Linearity	± 0.5% of full screen	± 1% full screen
Brightness	30 ft lamberts (min) with 100 line raster	50 ft lamberts with 200 line raster P7 standard, others optional
Physical Dimensions	51-1/2 in. X 30-3/4 in. X 45-3/4 in.	
Power Dissipation	At 110 Vac 12 A surge, 6 A running	
Temperature Range	40° F to 95° F	Same
Humidity	20% to 55% relative with no condensation	Same

VL04 Light Pen

The Type 374 Light Pen used in the VL04 is a photosensitive device that detects illuminated phosphors on the CRT screen when programmed to do so. The light pen is interfaced with the VT15 Graphic Processor through a connector on the VT04 Display Console. The light pen can be used to delete unwanted graphic constructions and to draw directly on the face of the CRT through suitable subroutines.

VW01 Writing Tablet

The writing tablet option is an acoustical X-Y digitizer connected directly to the control processor. Its horizontal surface provides free arm movement and its rapid response allows users to take full advantage of the speed of the 21-inch scope.

The VW01 operates in one of two modes: single point or data input. In the single point mode, a single spark is generated each time the pen is pressed against the writing surface. This operation is used when the operator desires to plot specific points in the X- or Y- axis.

In the data input mode, the spark pen produces a continuous series of sparks at a constant rate (normally 200 Hz). This mode allows the user to draw continuous lines, circles, curves, etc. The dual mode capability of the VW01 enables the user to perform a myriad of graphic analytical tasks quickly and accurately.

Drawing Surface:

11 in. X 11 in. (and expandable to 18 in. X 18 in., 24 in. X 24 in., 30 in. X 30 in. and 36 in. X 36 in.)

Data Rate:

Single Tablet:	200 coordinate pairs/ second (maximum)
Multiple Tablets:	100 coordinate pairs/ second
Digital Resolution:	1024 X 1024 (10 bits)
Graphic Resolution:	1000 line pairs
Reproducibility:	1:1000
Environmental Temperature:	40° to 90° F (5° to 30° C)
Humidity:	20 to 50% without condensation
Electrical:	Single phase, 110/220 V, 50/60 Hz

LK35 Keyboard Option

The LK35 Keyboard option is a send-only keyboard, through which an operator at a remote display console station can enter data into the PDP-15 computer. The keyboard is the link between the operator, the VT15 Graphic Processor and the PDP-15 computer. Instructions or data are entered into the PDP-15 computer through the send-only keyboard, are processed by the computer, and stored in core memory as display-file data. A subroutine permits any characters struck on the send-only keyboard to be displayed on the CRT screen.

VM15 Display Console Multiplexer

The VM15 Display Console Multiplexer permits up to 4 Type VT04 Display Consoles, 4 Type VT01 Display Consoles and 4 Type 374 Light Pen options to be interfaced with a single VT15 Graphic Processor. This option permits these equipments to be situated at remote locations and to share the use of the VT15 Graphic Processor and PDP-15 computer.

VP15 Displays

Three low-cost, point-plotting display systems can be provided by the VP15 family of display controllers. The various controller/display device combinations allow the selection of a system particularly suited to a user application.

VP15A Storage Tube Display – The VP15A is a unique and extremely useful point-plotting display terminal that uses a VT01 Storage Tube Display. Points are plotted on a 1024 by 1024-bit matrix on the 8-1/4 by 6-3/8 in. display surface. Two IOT-selectable modes of operation, store and nonstore, are provided. In the store mode of operation, plotted points remain visible up to 15 minutes. No refresh memory is required. In the nonstore mode of operation, a point-refreshing rate of at least 30 Hz is required to keep points visible, but a faster response time is achieved. In either case, only one intensity level is provided.

VP15B/BL X, Y Display – The VP15 B Oscilloscope Display provides a low-cost point-plotting capability. The 1024 by 1024-bit raster is displayed on the Tektronix RM503 X/Y Oscilloscope. Four intensity levels are provided. The VP15BL includes a DEC 370 Light Pen for use with the display.

VP15C/CL Oscilloscope Display – The VP15C version of the basic VP15 display controller is for use with the VR12 display. This provides the user with a display device with useful display dimensions of 7 X 9 in. The VP15C provides various intensity levels. The VP15CL includes a DEC 370 Light Pen for use with the display.

VT05 Alphanumeric Display Terminal

The VT05 is a self-contained, desktop, high capacity CRT display that is completely interchangeable with teletypewriter terminals and compatible with EIA communications interface standard EIA RS-232C. The CRT display terminal with keyboard is used to compose, edit, and input data to the PDP-15, to retrieve and update data and instructions, and to perform on-line program debugging with appropriate software.

The VT05 displays up to 72 characters per line, and has a 20-line capacity. The keyboard transmits full USASCII (128 characters, or a 97-character subset) up to 2400 baud. The CRT display, designed in accordance with EIA standards, can also display graphic information originating from a scan conversion device, and can interface with other closed-circuit TV devices, such as electronic cameras, scanned microscopes, and large screen display devices that use raster scan techniques.

ANALOG-TO-DIGITAL CONVERTERS

DEC supplies analog-to-digital conversion systems with high speed and wide dynamic ranges for up to 1000 channels. Both single-ended and differential systems are available, and amplifier and sample-and-hold options are also available.

AD15 Analog-to-Digital Converter

The AD15 is a 13-bit (12 bit plus sign) bipolar analog-to-digital converter, multiplexing up to 128 single-ended channels. It provides throughput rates of 30 kHz with signal range of ±300 mV to ±10 V. The AD15 converts via program control instructions, direct memory access, or external pulse. It includes a gain-switched amplifier with four ranges and sample-and-hold capability. The add-to-memory feature permits real-time signal averaging.

AM01-A

The AM01-A permits expansion of the AD15 in 32-channel blocks. One is required for each 32-channel group. The AM01-A for the first 32 channels is supplied with the AD15 and up to three additional AM01-A units may be implemented.

ADF15-C

This high-speed A/D converter has a 100 kHz throughput rate with 0–36 analog channel capability wired in 32-channel groups. It has six operating models and otherwise is similar to the AD15.

DIGITAL-TO-ANALOG CONVERTERS

Digital-to-analog converters with 10–12 bit accuracy are available packaged with from 1 to 64 multiplexed channels.

AA15B Digital-to-Analog Multiplexer Control

The AA15B Multiplexer Control houses and controls up to 16 AAC3 Digital-to-Analog Converters and their associated BNC output connectors. The AA15B interfaces to the PDP-15 I/O bus, from which it receives channel selection numbers, digital data, and IOT commands.

The AAC3 Digital-to-Analog Converter offers 12-bit resolution and single-buffered output of 0 to +10 V.

DW15A I/O Bus Adapter

The DW15A I/O Bus Adapter converts a positive PDP-15 I/O bus of +2.5 V and ground signals to a negative bus of -3 V and ground. This adapter interfaces PDP-15 systems with PDP-9 peripherals.

INDUSTRIAL CONTROL SYSTEM

Industrial-accepted digital input/output (UDC-15) and low level analog data acquisition (AFC-15) subsystems are designed to provide high noise immunity and process isolation. The flexibility of the input and output types allows direct interface with most industrial devices, thus allowing real-time digital I/O capabilities. Highly modular in organization, the system is housed in three basic parts:

BD-15 Control
UDC-15 Universal Digital Control subsystem
AFC-15 Automatic Flying Capacitor subsystem

The controller controls both the UDC-15 and the AFC-15 subsystems.

UDC15 Universal Digital Control

The UDC15 Universal Digital Control can control either or both digital input and/or digital output boards. Any slot can accommodate either board. Modules to implement these words are in one word (16-bit) increments.

Digital Output – Fully implemented with digital output boards, the UDC-15 is capable of driving 4096 output points with either level outputs, pulse outputs, momentary closures, sustained contact closures, and latching contact closures.

Digital Input – Fully implemented with digital input boards, the UDC is capable of sensing 4096 digital inputs. The inputs must be supplied with either 6-, 24-, or 48 V field power.

AFC-15 Automatic Flying Capacitor

This device multiplexes up to 2048 differential analog input signals, is field installed, and is a highly modular system employing:

Voltage or current input
  10 mV to 100 V full scale
  1 mA to 50 mA full scale
Integral signal conditioning
Eight program selectable gains per input
Scan rate of 200 channels per second
High system accuracy, ±.05% of input range
Flying capacitor multiplexing
Common mode noise rejection – greater than 120 db
Common mode voltage – greater than 200 V
Economical, simplified field wiring
Screw terminal connectors

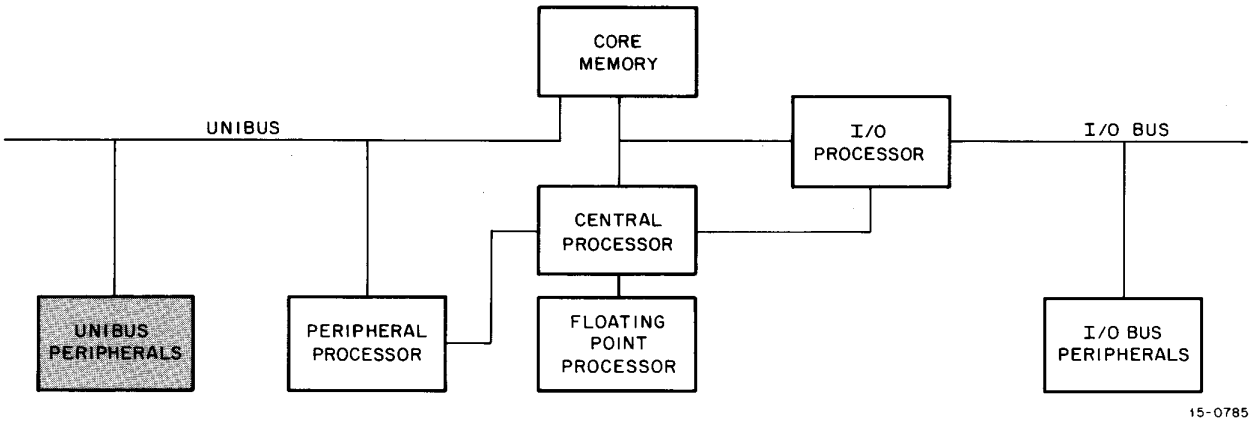
INTERPROCESSOR BUFFER SYSTEMS

Two interprocessor models are available to facilitate communications between computers. Their use is determined by the type of programmed data processor to be interfaced with the PDP-15.

The DB99 Interprocessor Buffer System is used to interface two PDP-15 Programmed Data Processors, or a PDP-15 with a PDP-9. The DB98 Interprocessor Buffer System is used to interface a PDP-15 with a PDP-8 Programmed Data Processor.

The interprocessor buffer system permits one data processor to communicate with a second data processor. The transfer of data may occur on the I/O bus between the accumulator of each processor, or on the data channel. The data channel allows higher transfer rates of blocks of data between the computers, without interrupting program operation. The block transfers use the multicycle data channel facility.

CHAPTER 7
UNIBUS PERIPHERALS



LP11 LINE PRINTER

- 356 lines/minute
- 80 or 132 columns
- 64 or 96 character sets
- low cost

The LP11 is a high-speed line printer available for the PDP-11 computer. The LP11 will print at a maximum of 356 full lines per minute for the 80-column, 64-character model. This model will print more than 1100 lines per minute for 20 columns per line. The LP11 can be ordered with a 64- or 96-character printing set. The printer is an impact type using a revolving character drum and a hammer per column. Up to six-part forms may be used for multiple copies. Fanfold paper from 4-in. to 14-7/8 in. wide may be used with adjustment of the pin-feed tractors.

Operation

80-Column Model

Characters are loaded into the printer memory via the Line Printer Buffer (LPB) serially by character. When the memory becomes full (20 characters), they are automatically printed. This is repeated for the next set of twenty hammers until a full 80-column line has been printed or a special character is recognized. The special characters (nonprinting) are: carriage return, line feed, and form feed.

132-Column Model

The operation is similar to that of the 80-column model except that the printer memory contains 22 characters and there are 132 print hammers.

Model Designation

LP11–x, y

x = F for 80 col. 64 character set
H for 80 col. 96 character set
J for 132 col. 64 character set
K for 132 col. 96 character set

y = A for 60 Hz, 117 V power
B for 50 Hz, 220/240 V power

(example: LP11-FA = 80-col., 64-character set, 117 V model)

LS11 LINE PRINTER

The LS11 Line Printer is a functional and attractive forms-handling unit designed to print clear reports, statements, and listings as required by most users. It is an inexpensive, medium-speed, and reliable impact dot-matrix line printer for the PDP-11 family. The line printer prints at a rate of 165 characters per second at 10 characters per inch with up to 132 characters per line, using the latest state-of-the-art imprint techniques. Printing speeds range from 60 lines per minute on full lines to 200 lines per minute for 20-character lines. The LS11's print assembly is a completely self-contained unit that includes the power supply plus the mechanical and logical components.

Specifications

- Printing Speed – 165 characters per second, 200-ms full-carriage return
- Line Length – 132 columns
- Character Spacing – 10 characters per inch
- Line Spacing – 6 lines per inch
- Paper – Multiple copy, up to five (5) parts with single-shot carbon
- Type Face – 9 X 7 dot matrix
- Size – 27-1/2 X 11-1/4 X 19-1/4 in. (W X H X D)
- Weight – Approximately 155 lb
- Operating Temperature – 40° to 100° F
- Electrical – 115 Vac, 60 Hz, or 230 Vac, 50 Hz

LV11 PRINTER-PLOTTER

The LV11 Printer-Plotter provides quieter and more reliable operation than conventional impact printers and pen plotters, especially under heavy, continuous use. The entire ASCII character set (including upper- and lower-case alphabet) is printed in 132 columns per line at 500 lines per minute. The supplied, program-controlled interface allows both printing and plotting, and accommodates most DEC line printer software. In the plotting mode, the LV11 prints 122,880 dots per second (independent of picture complexity) with a resolution of 10 bits (1024 dots per line). The printer-plotter uses roll paper for continuous plots and printouts (up to 500 ft), or fanfold paper for easier handling.

The electrostatic printing technique employs a fixed writing head with 1024 addressable writing electrodes. As the paper passes over the writing head, any (or all) of the electrodes may be requested to deposit a charge on the coated paper. The charged paper then passes over a liquid toner containing carbon particles; the particles are attracted to the charged areas on the paper, causing the appearance of black dots.

The only moving parts in the LV11 are the paper-moving motor and a small toner pump – simplicity of design that guarantees long, trouble-free operation that more than offsets the small additional cost of the coated paper.

Specifications

Plotting

Plotting area	10.24 in.
Total writing points	1024
Writing point spacing	100/inch
Vertical line spacing	100/inch
Input	8-bit parallel bytes
Data transfer rate	500K bytes/second
Plotting speed	122,880 dots/second
Memory	One-line buffer (1024 bits)

Printing

Columns	132
Character spacing	12.5/inch
Character font	7 X 9 dot matrix
Character generator	Read only memory (ROM)
Print rate	500 lines/minute
Input code	7-bit ASCII (USAS X 3.4-1968) parallel, no parity
Character set	96
Memory	One-line buffer (132 characters)

Dimensions

Width	19 in.
Height	38 in.
Depth	18 in.
Weight	160 lb

Power Input

115 V 600 W single phase
230 V 600 W single phase

(All models operate at 50 or 60 Hz.)

Operating temperature	50° to 110° F
Humidity	20% to 80% noncondensing
Paper drive	Incremental
Paper advance speed	1.20 in./second
Writing spot size	0.0075 in. diameter
Paper	Roll–11 in. wide X 500 ft long Fanfold–11 in. wide X 1000 sheets
Toner supply	2 gallons (enough for over 7000 pages)
Mean time between failure	> 3000 hours

Manual Controls/Indicators

Power ON-OFF (Illuminated)
Paper Advance
Out of Paper Indicator
Contrast Control
Fanfold/Roll Paper Selector

RK15 DISK CARTRIDGE SYSTEM WITH UNICHANNEL 15

The RK15 is a complete subsystem (Figure 7-1) that contains three major components:

- The Unichannel 15 peripheral processor, which uses the PDP-11/05
- The RK11E Disk Controller
- The RK05 Cartridge Disk Drive

The RK15 offers low cost, random access storage expandable from 1.2 to 9.6 million 18-bit words; removable interchangeable DECpacks; up to eight drives per controller; 50 ms average access time; direct to memory operation; easy copying from cartridge to cartridge; write check, track verification, checksum, and maintenance features.

RK11E Disk Controller Features

- The controller accepts and assembles full 18-bit words.
- All transfers are direct to memory and require only one memory reference per word transfer. Word count and current address information is contained within the controller.
- DMA devices can read or write data anywhere in common PDP-15 memory. Data may be treated as 18-bit words, 16-bit words, or as two 8-bit bytes.
- Semiconductor memory allows up to a 50 microsecond transfer latency with no loss of data or "retry" required.
- System may be expanded at any time with up to 8 drives.

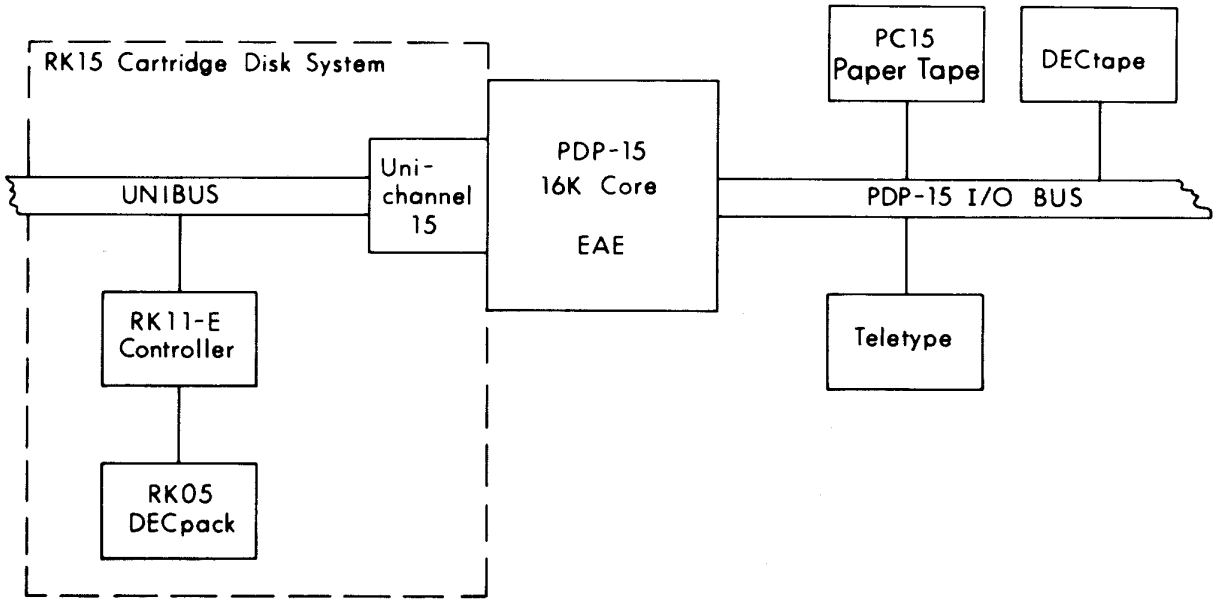


Figure 7-1 RK15 Cartridge Disk System with Unichannel 15

- Operates with or without Automatic Priority interrupt on the PDP-15.

RK11/RK05 DISK AND CONTROL

- Low cost, random access storage up to 9.6 million words.
- Removable interchangeable DECpacks.
- Up to 8 drives per controller.
- Overlapped seeks for high throughput.
- Easy copying from cartridge to cartridge.
- Write check, track verification, checksum, and maintenance features.
- Cartridge case and filtration system protects against dust.

The RK11/RK05 is an economical solution for applications requiring a moderate amount of bulk storage. Disk cartridges are compact, lightweight, and are suitable for storage of backup files. All packs are removable so that the user has almost unlimited off-line storage capacity. In the event of a lost file, the quick-copy facility of the RK05 will get the system back on the air in a matter of minutes.

The PDP-15/76 user can start by purchasing an RK11 controller, a single RK05 drive and one or more interchangeable DECpacks, each with a storage capacity of 1.2 million words. Up to 8 drives can be added without additional power supplies or controllers. Additional RK11s may be added to the system, each controlling up to 8 drives.

Since each DECpack drive is an independent unit and controller electronics are totally separate, the RK05 can overlap its operations. For example, while one drive is performing a READ or a WRITE operation, other disks on the system can be seeking new track locations. Since "seeks" normally take less time than data transfers, the new information is ready for transfer as soon as the previous operation has been completed.

Data can be transferred from one disk to another in 600 ms, including rotational access time. Because the heads are positioned on separate tracks, arms need only be moved when a track has been completely written or read. To copy an entire disk, the user inserts a blank

cartridge in one drive and the cartridge to be copied in another. Copying is then performed in a single step.

To insert a new disk, only the required drive need be turned off-line. Remaining drives can continue to operate without any interruption. As long as one drive remains in service, the system will continue to operate.

To ensure accurate storage and transfer, DEC-pack systems employ a write check function, maintenance features, and hardware features that verify the correct track selection and provide a checksum.

Voice coil positioning and an optical position transducer provide fast access times and accuracy to within 100 millionths of an inch. A unique head carriage design requires no periodic lubrication and allows the head to be supported by its center of gravity for accurate and repeatable tracking. By eliminating mechanical braking, the design removes a major source of wear and critical adjustment.

To protect the disk from damage, an emergency retract power supply automatically forces the heads to their home position if the power fails.

The RK11/RK05 system is cooled with forced air whenever power is on, even when the drive is stopped. Clean cabinet air is introduced into the rear of the drive through a foam prefilter. It then passes over the electronics module through the blower to a filter which removes 99.97 percent of the ambient particles greater than 0.3 micron in size. The clean air is fed to the power supply positioner and disk cartridge at a rate of more than 30 cfm.

RK05 Specifications

Capacities

Disks/drive	1
Surfaces/disk	2
Heads/drive	2
Recording density	2200 bpi
Disk capacity (words-formatted)	1,228,800
Disk capacity (bits-unformatted)	24,400,000
Tracks	400 + 6 spare
Cylinders	200 + 3 spare
Sectors per track	12
Words/sector	256
Words/track	3072

Transfers

Minimum transfer	1 word
Data transfer rate	11.7 μs/word
Bit transfer rate	1.54 million bits per second
Bit transfer code	Double frequency non-return-to-zero

Power

Operating Power	160 W @ 2.1 A
Starting current (to start spindle)	10 A for 2 sec.

CR11 CARD READER

- "Riffle air" card separation to minimize effects of card damage and humidity
- Vacuum picker to prevent "double picking"
- Reliable diode-transistor read station
- Card loading and unloading during reader operation
- Optional automatic reader shutdown
- Quiet operation

Reader design helps prevent card jams and keeps card wear to a minimum. Readers also have a high tolerance to cards that have been nicked, warped, bent or subjected to high humidity.

To keep cards from sticking together, the readers use a special "riffle air" feature. The bottom half inch of cards in the input hopper is subjected to a stream of air which separates the cards and air cushions them from the deck and from each other.

Cards entering the reader are selected through a vacuum picker. The picker and its associated throat block prevent the unit from accepting cards that have been stapled or taped together (unless such taping is on the leading edge). Because the card track is very short, only one card is in motion at any time. This minimizes the chances of cards jamming. Stoppages are also reduced since the reader automatically makes six attempts to process a card before rejecting it.

The read station uses infrared light-emitting diodes as its light source and phototransistors as its sensors. No adjustments are required during the ten-year life expectancy of the diodes.

Because card reader operation is flexible, cards can be loaded and unloaded while the reader is operating. A switch may be set to provide system blower shutdown or continual running after the last card has been read. Automatic shutdown reduces computer room noise level, and indicates that the card hopper is empty.

Specifications

	CR11
Reading Rate (cpm)	300
Input Hopper Capacity (cards)	550–600*
Output Hopper Capacity (cards)	550–600*
Size Envelope (HXWXD)	
inches	11X19X14
centimeters	28X48X36
Weight (max)	
kilograms	32
pounds	70
Power Consumption (VA)	
starting	950
running	400
Head Dissipation	
BTU/hour	1360
Environmental	
Temperature	+10° to 50° C
Humidity	10 to 90%

*Depending on card stock

DP11 SYNCHRONOUS INTERFACE

- Double buffered program interrupt character service
- Full or half duplex operation
- Programmable sync character
- Programmable character size (6, 7, or 8 bits)
- Receive sync character stripping program selectable
- Automatic transmit of sync program selectable
- Speeds to 50,000 baud
- Interfaces to Bell 201 and 303 or equivalent modems
- Auto answering capability
- Dynamic maintenance capability
- Internal clocking source (optional)
- Extended character sizes (10, 11, or 12 bits optional)

The DP11 is a fully character-buffered synchronous serial line interface capable of two-way simultaneous communications. The DP11 translates between serial data and parallel data. Output characters are transferred in parallel from the computer to a buffer register where they are serially shifted to the communication line. Input characters from the modem are shifted into a register, transferred to a buffer register, and made available to the PDP-11 on an interrupt basis.

Both the receiver and the transmitter are double buffered. This allows a full character time in which to service transmitter and receiver interrupts.

The clocking necessary to serialize the data is normally provided by the associated high-speed synchronous modem. Alternately, the internal clocking option can be used for local terminals when no external clocking is available.

The DP11 provides a double-buffered program interrupt interface between a PDP-11 and a serial synchronous line. This interface allows the PDP-11 to be used in remote batch and remote concentrator applications. With the DP11, a PDP-11 can also be used as a front end synchronous line controller to handle remote and local synchronous terminals.

The DP11 interface offers flexibility. It handles a wide variety of terminals and line disciplines (i.e., line control procedures and error control techniques). A programmer can vary sync character, character size, and modem control leads. Automatic sync character stripping and automatic idling are also program selectable. While idling, the DP11 transmits the contents of the sync buffer.

The DP11 design provides individual interrupt vectors and hardware interrupt priority assignments for the transmitter and receiver. Interrupt priority is jumper selectable. This feature, coupled with the automatic transmit idle capability, enables dynamic system adjustment to peak message activity. For example, the programmer can temporarily ignore the transmitter if receive activity is high.

Because the PDP-11's Unibus serves as a multiplexer, multiple synchronous lines can be added to a PDP-11. One PDP-11 system unit's worth of mounting space is used for each independent synchronous line interface unit.

XY11 Plotter Control

The XY11 Plotter Control provides the user with a versatile plotting capability. Plots of either .01-in. or .005-in. steps can be generated at speeds to 300 steps per second maximum. The XY11 control plugs directly into any available PDP-11 Small Peripheral Controller slot. All operations are under program control; either axis (or both axes) can be addressed in positive or negative incremental steps.

Compatible Digital Plotters

A variety of popular plotters can be interfaced to the XY11 Control to provide the user with drum, fan-fold, or flat-bed capabilities. Detailed specifications concerning available plotters can be obtained directly from DEC or from the appropriate manufacturer. The following models are currently available from DEC:

Calcomp 563
Calcomp 565
Houston Complot DP-1
Houston Complot DP-10

Full warranties and maintenance contracts are available for all plotters supplied by DEC.

PLOTTERS

The plotters, whose specifications follow, may be purchased directly from DEC for inclusion in the system. Users may also purchase these plotters, or their equivalents, directly from the manufacturer for use with the XY11.

CalComp-563

(Manufactured by California Computer Products, Inc.).

Mechanical:

Dimensions	9.8 in. h, 39.5 in. w, 14.7 in. d
Type	Drum, tabletop

Electrical:

Input Power	105–125 Vac, 50/60 Hz
Current	1.5 A

Operational:

Plot Size	Y axis = 28.55 in. X axis = 120 ft
Stepping Increments	.01 in. and .005 in.
Stepping Speed	200 and 300 steps/sec

CalComp-565

Mechanical:

Dimensions	9.8 in. h, 18 in. w, 14.7 in. d
Type	Drum, tabletop

Electrical:

Input Power	105–125 Vac, 50/60 Hz
Current	1.5 A

Operational:

Plot Size	Y axis = 11 in. X axis = 120 ft
Stepping Increments	.01 in. and .005 in.
Stepping Speed	300 steps/sec

Complot DP-1 (Manufactured by Houston-Instrument Division of Bausch & Lomb)

Mechanical:

Dimensions	9.5 in. h, 19.75 in. w, 14 in. d
Type	Fanfold, tabletop

Electrical:

Input Power	115/230 Vac, 50/60 Hz

Operational:

Plot Size	Y axis = 11 in. X axis = 8.5 in. (144 ft overall)
Stepping Increments	.01 in. and .005 in.
Stepping Speed	300 steps/sec

Complot DP-10

Mechanical:

Dimensions	6.5 in. h, 19.0 in. w, 15.33 in. d
Type	Flat bed, tabletop or rack mounted

Electrical:

Input Power	115/230 Vac, 50/60 Hz

Operational:

Plot Size	Y axis = 11 in. X axis = 8.5 or 17 in.
Stepping Increments	.005 in.
Stepping Speed	300 steps/sec

CHAPTER 8
ADDRESSING

This chapter describes the PDP-15 addressing scheme and the basic data word formats.

INTERPRETATION OF WORDS FROM MEMORY

Words stored in magnetic core memory are strings of 18 bits. An instruction word is indistinguishable from a data word. The Central Processor which decodes and implements instruction words, differentiates data words from instruction words by the sequence in which they are retrieved from storage. The program counter is used to point at the location of the next instruction. The instruction itself, if a memory reference instruction, then points to the memory location where data is to be fetched or stored. There are two types of instruction words:

a. Those which reference memory by indicating in the operand address field the location of the data necessary to carry out the operation (e.g., Add the contents of memory location 1000 to the accumulator).

b. Those that deal with control and do not require a memory reference cycle (e.g., shift the contents of the accumulator two binary bits to the right).

INFORMATION RETRIEVAL FROM MEMORY

The basic concept involved in retrieving information from storage is that each piece of information has an address, similar in nature to a street address to locate a building, or a zip code to locate a postal zone. Integer addresses ranging from 0 to $32{,}767_{10}$ (or 00000_8 to 77777_8) are used by the PDP-15 to locate information in Memory. Figure 8-1 shows a diagram of the program counter register. Bits 3 through 17 are used to address the first 32,768 core locations in Memory.

Bits 3 and 4 select the memory bank being addressed (banks 0 through 3) and bit 5 determines which page (page 0 or page 1) of that bank the word is in. The remaining positions of the address are used to select one of the 4096 words on that page.

NOTE

The PDP-15 is designed to allow for special expansion to 131,072 words of memory, so that address registers such as the program counter are 18 bits long. Bit 0 is the sign bit, bits 1 through 17 are used to address up to 131,072 words as shown in Figure 8-1.

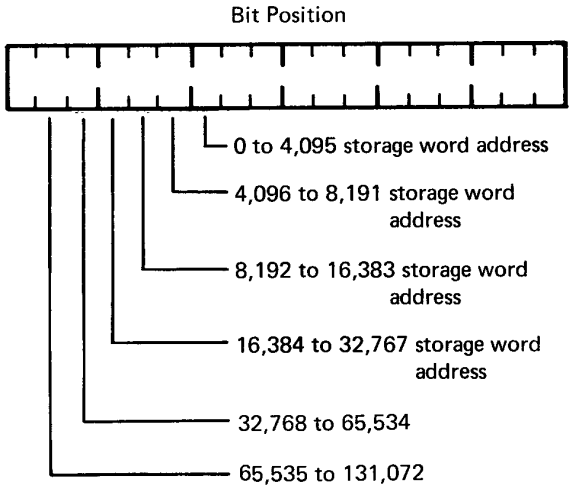
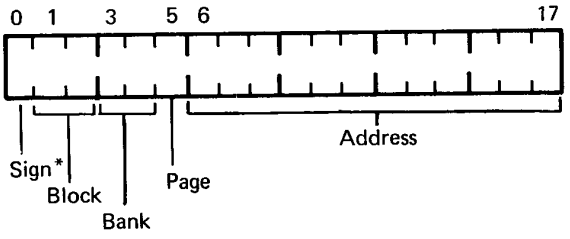


Figure 8-1 Program Counter Register

The *Program Counter* is an 18-bit register which points to the next instruction. This register can be loaded from the console switches to begin execution of the program. At the beginning of each instruction, the program counter is incremented by one to specify the memory location of the next instruction. However, this incrementing is performed modulo 4096. When the PC is incremented, only the 12 low-order bits function as a 12-bit counter. There is no overflow into the high order bits.



*The sign bit is unused in address pointing since the PDP-15 System does not reference negative memory.

To change pages, or banks within a 32K block, a jump indirect is normally used. To change blocks for systems greater than 32K, a jump indexed is used. These instructions replace the contents of the program counter with 15 bits or 17 bits, respectively, to effect both a new bank-page or 32K block address.

MEMORY REFERENCE INSTRUCTIONS

Figure 8-2 shows the format of PDP-15 Central Processor memory reference instructions. The bit positions in the 18-bit word are numbered from 0 to 17, counting to the right with bit 17 as the low-order bit. This convention will be employed throughout this manual. A PDP-15 word will be construed as an instruction word only when it has been retrieved from storage and transferred into appropriate registers in the Central Processor.

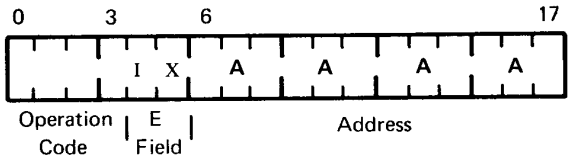


Figure 8-2 Memory Reference Instruction Format

There are three fields in the memory reference instruction word; the operation code, the address mode (E Field) and the operand address field.

Operation Code Field

The 4-bit operation code occupies bits 0 to 3 of the word and specifies 1 of the 13 memory reference class of instructions. The remaining three codes in the operation field are used to specify the non-memory reference instructions and are discussed later in this chapter.

Operand Address Field

The 12-bit operand address field occupies bits 6 to 17. The number contained in the address field is the address of a word located in core storage (which is usually one of the operands of the instruction; e.g., ADD Y) although this address may be modified before it actually references a word in memory. Since this field is 12 bits long, the instruction directly addresses 4096 words of memory, or one full page.

Address Mode

The 2-bit address mode or E Field occupies bits 4 and 5. The 2-bit E Field indicates address modification as illustrated by the flowchart, Figure 8-3. There are four combinations as shown by the following:

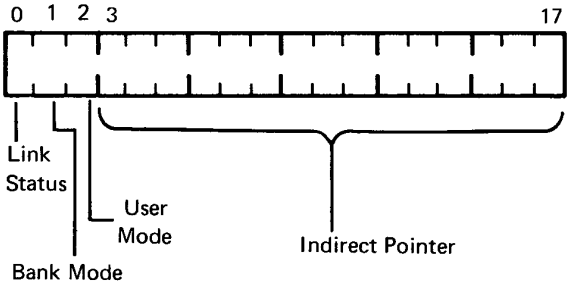
E Field	Value	Operation	Addressable Memory	Δt*
0 0	0	Direct	4K	0
0 1	1	Indexed	128K	0
1 0	2	Indirect	32K	.8
1 1	3	Indirect-Indexed	128K	.8

*Additional time (μs) required for address modification.

E=0 No Address Modification – The 12 bits in the address field point directly to the operand's address in the current page where the data will either be fetched or stored. The address pointer is formed in the operand address register by concatenating the block, bank, and page address stored in the program counter to the 12-bit address field. Since the block, bank, or page address does not change unless a field change instruction is issued (jump indirect or indexed), the term current page address always infers the concatenation process.

E=1 Indexed Address Modification – The content of the index register (18 bits, including sign) is added to the current page address. PC bits 3, 4, and 5 plus the 12-bit page address form the effective address where data will be transferred. An indexed instruction can directly address any portion of core memory and does not add any additional time to an instruction's execution time.

E=2 Indirect Addressing – When indirect addressing is indicated, the current page address is



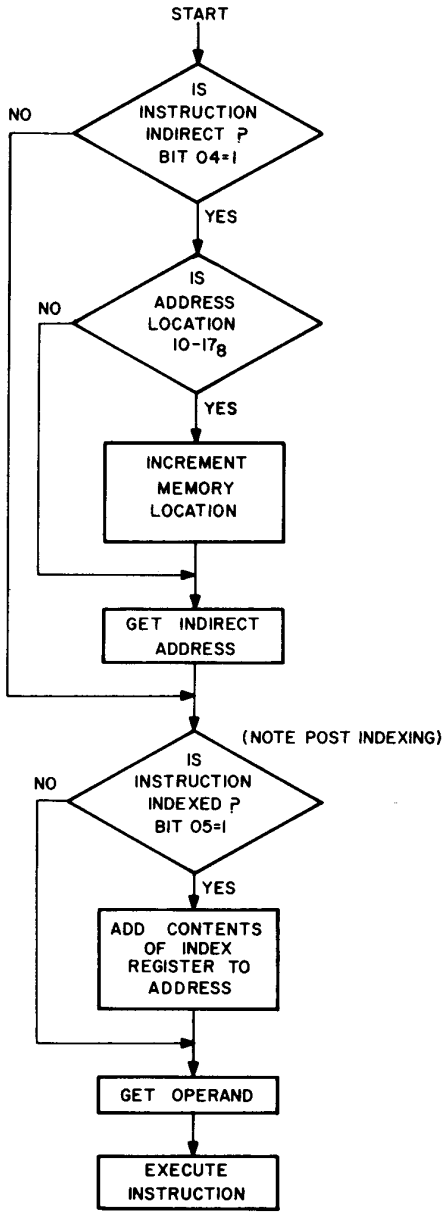


Figure 8-3 Address Modification Flowchart

taken as containing not the operand, but a pointer to a location whose contents are used as the operand address.

The indirect pointer is concatenated with the current block address, bits 1 and 2 of the program counter, in the operand address register to form the effective address where data is transferred. A second memory reference cycle is then required for obtaining the actual data.

Only one level of indirect addressing is permitted. Fifteen bits of the indirect word are used as the address, permitting 32,768 possible words in a current block to be accessed. The first three bits in the indirect pointer are used to store status information and are not used for address formation.

E=3 Indirect and Indexed Address Modification – When both indirect and indexed addressing are indicated, the indirect operation occurs first as shown above. Except for the addition of PC bits 3, 4, and 5, the value of the index register is added to form the effective address.

A second register, called the limit register, (18 bits), is used to test the index register when it is used in a loop. An "Add to Index and Skip" (AXS) instruction causes a signed eight-bit number, contained in the last eight bits of the AXS instruction, to be added to the index register. The index register is then compared to the limit register and if the sum in the index register is equal to, or greater than, the limit register, the next instruction is skipped.

Indexing Examples

The following examples show typical use of the index register. These examples assume that the programs and arrays are in the same page of memory or the programs are in page 0 and the arrays are within the 32K block of memory.

Example 1:

Form the sum of every other word in an array called SAM; the fields are: Tag, Operation, Operand, Comments

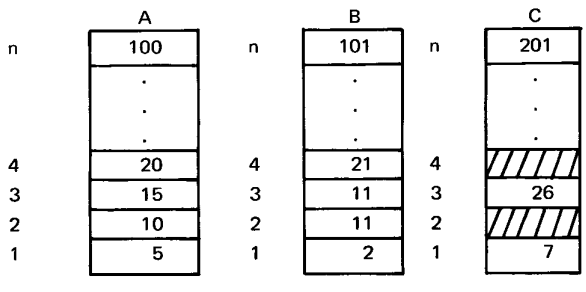
Tag	Operation	Operand	Comments
	.		
	.		
	.		
	LAC	(SAM	/LOAD AC WITH
			/BOTTOM ADDRESS
			/OF ARRAY
	PAX		/DEPOSIT INTO INDEX
			/REGISTER
	LAC	(SAM+N	/LOAD AC WITH
			/NEXT ADDRESS
			/AFTER ARRAY
	PAL		/DEPOSIT INTO LI-
			/MIT REGISTER
BEGIN	CLA!CLL		/CLEAR AC AND
			/LINK
LOOP	TAD	0,X	/FORM SUM
	AXS	2	/ADD TWO TO INDEX
			/REGISTER AND TEST
			/IF COMPLETE
	JMP	LOOP	/CONTINUE LOOPING
	HALT		/LOOP COMPLETE

!indicates a microprogrammed operation

Example 2:

Form the sum of every other element in arrays A and B and store them in array C.

Tag	Operation	Operand	Comments
START	CLX		/CLEAR XR
	LAC	(n	/LOAD ARRAY SIZE
	PAL		/INTO LR*
LOOP	LAC	LOCA,X	/FORM SUM OF EVERY
	TAD	LOCB,X	/OTHER ELEMENT OF
			/ARRAY A AND B AND
	DAC	LOCC,X	/STORE IN EVERY
			/OTHER WORD OF
			/ARRAY C
	AXS	2	/ADD 2 TO XR AND
			/SKIP IF XR $\geq$ LR
	JMP	LOOP	/CONTINUE
	HLT		/DONE
LOCA	A_1		/1ST DATA ELEMENT
	.		OF ARRAY A
	.		
	.		
	A_n		/LAST DATA ELEMENT
			/OF ARRAY A
LOCB	B_1		/1ST DATA ELEMENT
	.		/OF ARRAY B
	.		
	.		
	B_n		/LAST DATA ELEMENT
			/OF ARRAY B
LOCC	C_1		/1ST DATA ELEMENT
	.		/OF ARRAY C
	.		
	.		
	C_n		/LAST DATA ELEMENT
			OF ARRAY C



*LR=LIMIT REGISTER

Auto-Increment Locations – Eight locations (10_8-17_8) of the first 4096-word page in bank 0 act as auto-increment registers. When indirectly addressed, the contents of an auto increment location are incremented by 1 and then taken as the effective address of an instruction. When directly addressed, these locations act like all other memory locations.

The auto-increment locations are used to loop through sequential data arrays. The "increment and skip if zero" (ISZ) instruction is used to test for loop completion. The following example illustrates their use:

```
                Form sum of Am + Bm and store in Cm
                .
                .
                .
BEGIN   CLA!CLL         /CLEAR AC AND LINK
LOOP    LAC* 10         /GET ADDEND
        TAD* 11         /FORM SUM
        DAC* 12         /STORE SUM

                *indicates indirect addressing

        ISZ COUNT       /TEST FOR COMPLETION
        JMP LOOP        /CONTINUE LOOPING
        HALT            /LOOP COMPLETE
COUNT   -N              /NUMBER OF ITERATIONS,
                        /TWO'S COMPLEMENT
10      L(A)-1          /FIRST LOCATION OF ARRAY
                        /A-1
11      L(B)-1          /FIRST LOCATION OF ARRAY
                        /B-1
12      L(C)-1          /FIRST LOCATION OF ARRAY
                        /C-1
```

BANK MODE ADDRESSING

The PDP-15 can be placed into another mode of addressing called bank mode addressing. An "enable bank addressing" (EBA) instruction places the Central Processor in bank address mode. In this mode, 8192 words of memory can be directly addressed by memory reference instructions. This is done by allowing the fifth bit (bit 04) of the instruction to refer to a memory bank rather than to the index register. In this mode, all indexing operations including the use of the limit register for register-to-register compares are eliminated in favor of bank addressing. Indirect addressing can be used.

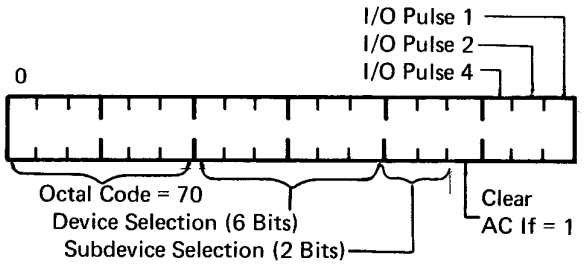
The program counter is incremented modulo 8192. That is, the low-order 13 bits point at the address within the bank specified by the high-order bits. When the program counter is incremented, only the 13 low-order bits function as a counter. There is no carry into the high-order bits. To change banks, a jump indirect instruction is normally used.

A "disable bank addressing" (DBA) instruction places the Central Processor back into the page address mode where indexing can be used.

NONMEMORY REFERENCE INSTRUCTIONS

Input/Output Instruction

IOT instructions are microcoded to effect responses for a particular device. The microcoding includes issuing a unique device selection code and appropriate processor-generated input/output pulses (IOP) to initiate a specific operation. For an "out" transfer, the program reads a data word from memory into the AC. A subsequent IOT instruction places data on the bus, selects the device, and transfers the data to the device. For an "in" transfer, the process is reversed. An IOT instruction selects the device and transfers data into the AC. A subsequent instruction in the program transfers the word from the AC to memory.

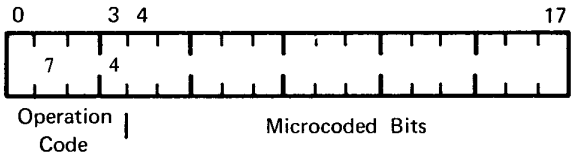


IOT instructions are also used to initialize the single- and multi-cycle channels and to transfer the word count and current address to the single-cycle device controllers. In addition, they are used to test or clear device flags, select modes of device operation and to control a number of processor operations. Within a single IOT instruction, up to 64 unique device-selection codes are available and an additional two bits form up to four subdevice commands. Three microprogrammed pulses (IOP) are also provided to test, initiate transfer, etc. (See the IOT instruction for complete details.)

OPERATE INSTRUCTIONS

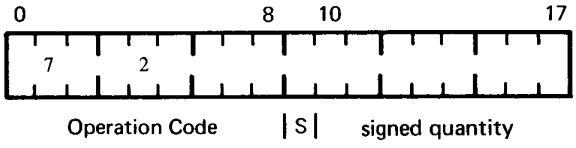
Microcoded Operate Instruction

Microcoded operate instructions (operate code of 74_8) are used to sense and/or alter the contents of the AC and Link. Typical functions are: conditional or unconditional skips and complementing, setting, clearing or rotating the contents of the two registers jointly or independently. A HLT instruction is included. Operates



are fetched and executed in one machine cycle; the actions are specified by the microprogramming of the instruction code. Each of the 13 bits can effect a unique response; hence they are "microinstructions" to the computer. The important feature of the operate class is its microprogramming capability which allows two or three microinstructions to be combined to form one instruction word and, therefore, to be executed during one cycle. Those microinstructions that logically conflict and occur in the same event time should not be microprogrammed.

Index Operates



Index operate instructions (operate code of 72_8 or 73_8) are used to clear, load, and compare the index and limit registers. The PDP-15 Index Register (XR) and Limit Register (LR) instruction set is divided into two groups:

(1) Instructions that have no operand.

(2) Instructions that have a 9-bit immediate operand in 2's complement signed notation; i.e., $-2^8 \leqslant n \leqslant 2^8 - 1$ or $-256_{10} \leqslant n \leqslant 255_{10}$.

Three of the indexed instructions have operands:

AAC n	Add n to AC
AXR n	Add n to XR
AXS n	Add n to XR and skip if XR $\geqslant$ LR

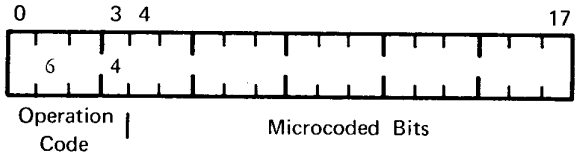
The following example shows typical use of the above instructions.

Form the sum of every other element in arrays A and B and store the sum in array C, using the XR to hold the base address of each array. Clearly, to do this efficiently, one XR per array is required. However, given that arrays A, B, and C coexist in the same page as the procedure and that both the size and position of each array relative to the others is no greater than 256_{10}, the following code could be used:

Example 3:

```
Tag     Operation   Operand      Comments
START   LAC         (LOCA+n      /LOAD NEXT ADDRESS
        PAL                      /AFTER ARRAY A INTO
                                 /LR where n ≤256
        LAC         (LOCA        /LOAD BASE ADDRESS
        PAX                      /OF ARRAY A INTO XR
LOOP    DAC         TEMP         /TEMP STORE POINTER
                                 /TO ELEMENT IN
                                 /ARRAY A
        LAC         0,X          /LOAD AC WITH ELEMENT
                                 /OF ARRAY A
        AXR         LOCB–LOCA    /INCREMENT XR TO
                                 /POINT TO ARRAY B
                                 /ELEMENT
        TAD         0,X          /STORE ELEMENT IN
                                 /ARRAY C
        AXR         LOCC–LOCB    /INCREMENT XR TO
                                 /POINT TO ARRAY C
                                 /ELEMENT
        DAC         0,X
        LAC         TEMP         /RESTORE XR TO POINT
        PAX                      /TO ELEMENT IN
                                 /ARRAY A
        AAC         2            /C(AC)=C(TEMP)+2
        AXS         2            /ADD 2 TO XR AND SKIP
                                 /IF XR ≥LR
        JMP         LOOP         /CONTINUE
        HLT                      /DONE
LOCA    A1                       /1ST DATA ELEMENT
          .                      /OF ARRAY A
          .
        An                       /LAST DATA ELEMENT
                                 /OF ARRAY A
LOCB    B1                       /1ST DATA ELEMENT
          .                      /OF ARRAY B
          .
        Bn                       /LAST DATA ELEMENT
                                 /OF ARRAY B
LOCC    C1                       /1ST DATA ELEMENT
          .                      /OF ARRAY C
          .
        Cn                       /LAST DATA ELEMENT
                                 /OF ARRAY C
TEMP    0                        /TEMPORARY STORAGE
```

EAE Instructions

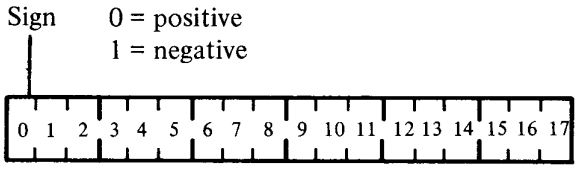


EAE, identified by an operation code of 64_8, performs high-speed data manipulation and multiply divide operations as specified by microprogramming of individual instructions. The microinstructions have capabilities for register set-up, data shift, normalize, multiply, and divide, respectively. EAE is an option on the PDP-15/10 System. For a complete description see the EAE instructions, described in Chapter 10.

DATA WORDS

Memory reference instructions deal with data used for arithmetic and logical operations and are also used for address modifications. The following describes the data formats used in the arithmetic and logical operations.

There are two types of arithmetic instructions in the PDP-15 – 2's complement and 1's complement. Floating point operations are provided in subroutines supplied with the system monitors.



Single-Precision Data

Single-Precision Data

Up to 18 bits of data may be contained in a single precision PDP-15 word. Normally, 2's complement arithmetic is used, because convention has adapted only one representation of 0 in 2's complement notation, namely +0; 1's complement notation has both a +0 and a -0 that can cause ambiguity.

In both complement notations (1's and 2's), the sign indicator (bit 0) is 0 for positive quantities and 1 for negative quantities. The 1's complement of a quantity is equivalent to the logical complement of its magnitude and sign; i.e., all binary 1s are replaced by 0s and all binary 0s are replaced by 1s. The 2's complement of the quantity is equivalent to its 1's complement plus the addition of 1 to the lowest order, or least significant, bit. Positive quantities in either notation have identical representations. For example: $+15_{10}$ (17_8) is represented in a PDP-15 data word as:

s
000 000 000 000 001 111

in either 1's or 2's complement notation. The 1's complement of -15_{10} is represented by:

s
111 111 111 111 110 000

The 2's complement of -15_{10} appears as:

s
111 111 111 111 110 001

A typical PDP-15 instruction sequence for forming the 2's complement of any number is:

```
LAC Y
TCA                 /2's complement the AC
DAC Y
TCA=CMA!IAC         /Complement AC and
                    /increment by 1
```

The TAD (2's complement add) instruction must be used rather than the ADD (1's complement add) instruction as ADD permits an end-around carry into the low-order bit.

Magnitudes of Data Words – For 2's complement signed notation, the permissable magnitude of any quantity, X, is in the range of:

$$-2^{n-1} \leqslant X \leqslant 2^{n-1} - 1$$

where n is again the number of bits allocated to the storage of data. A single-precision data word has the range:

$$-2^{17} \leqslant X \leqslant 2^{17} - 1$$

$$\text{or } -131\,072_{10} \leqslant X \leqslant +131\,071_{10}$$

The position of the decimal point is implied in the above ranges.

For 1's complement signed and sign-and-magnitude notations, the permissable magnitude of any quantity, X, is in the range of:

$$-(2^{n-1} - 1) \leqslant X \leqslant 2^{n-1} - 1$$

where n is the number of bits allocated to the storage of data in a data word. For a single-precision data word (sign bit and 17 data bits), this relationship becomes:

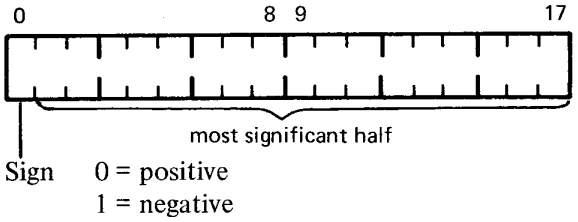
$$-(2^{17}-1) \leqslant X \leqslant 2^{17}-1$$

$$-131\,071_{10} \leqslant X \leqslant +131\,071_{10}$$

Double Precision Data

A signed double-precision data word consists of two computer words for a total of 36 bits (Figure 8-4). The word contains the sign bit and the 17 most-significant bits; the second word contains the 18 least-significant bits. The words are stored in consecutively addressed core memory locations for ease of programming.

1st Word



2nd Word

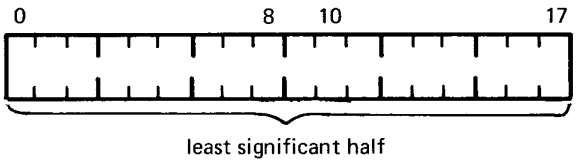


Figure 8-4 Double-Precision Data Word Formats

The magnitude of double precision in 2's complement is

$$-2^{35} \leqslant X \leqslant 2^{35}-1$$

$$-34{,}359{,}738{,}368 \leqslant X \leqslant 34{,}359{,}738{,}367$$

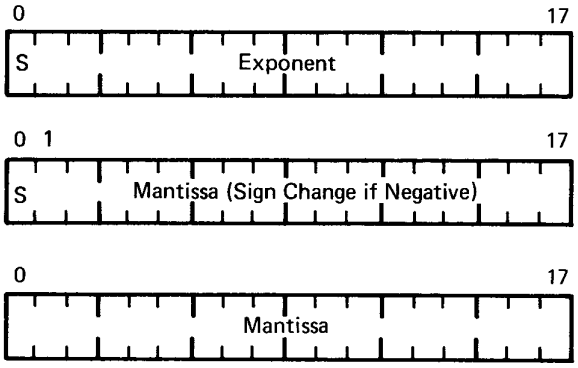
BASIC SOFTWARE FLOATING-POINT FORMATS

Floating-point representation of a binary number consists of two parts: the exponent and the mantissa. The mantissa is a fraction with the binary point assumed to be positioned between the sign bit and the most-significant data bit. The mantissa is always stored in a normalized state, i.e., leading 0s are eliminated from the binary representation so that the high-order bit is always a 1. The exponent, as stored, represents the power of 2 by which the mantissa is multiplied to obtain the number's value for use in computation.

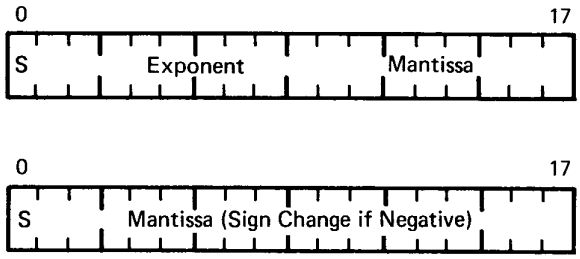
The PDP-15 floating-point software system offers two modes for storage of floating-point numbers: three-word mode and two-word mode.

The three-word mode requires three memory locations for storage of a floating-point binary number (Figure 8-5a). The exponent, a signed 17-bit integer in 2's complement notation, occupies the first word, or memory location. The mantissa, a 35-bit quantity in sign and magnitude notation, is stored in the second and third words. The sign of the mantissa is stored in the high-order bit of the second word. The range is $\pm 10^{39000}$ accurate to 9 decimal digits.

The two-word mode requires two memory locations for storage of a floating-point binary number (Figure 8-5b). The exponent, an 8-bit integer in 2's complement notation, and its sign occupy the 9 high-order bits of the first word. The mantissa, a 26-bit quantity in sign and magnitude notation, is stored in the 9 low-order bits of the first word and in the 17 low-order bits of the second word. The sign of the mantissa is stored in the high-order bit of the second word. The range is $\pm 10^{60}$ with an accuracy to 6 decimal digits.



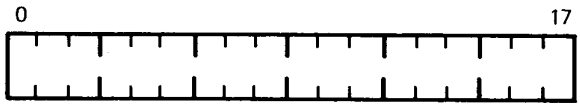
a. Three-Word Mode



b. Two-Word Mode

Figure 8-5 Floating Point Formats

BOOLEAN REPRESENTATION



Boolean operations use unsigned quantities.

Full 18-bit words can be ANDed or exclusive ORed with each other.

CHAPTER 9
PDP-15 CONSOLE

The PDP-15 console (see Figure 9-1) provides access to, or control over, virtually every portion of the PDP-15. All registers, buses, and controls are multiplexed down a single console cable to display the equivalence of 24 registers and 34 control functions. The console provides extensive control for real-time debugging and comprehensive man-machine interaction during checkout.

INFORMATION/CONTROL SWITCHES

The following two-position rocker switches are on the console keyboard:

Number	Designation	Number	Designation
18	Data	1	Repeat
15	Address	1	Protect
1	Clock	1	Single Time
1	Register Group	1	Single Step
1	Bank Mode	1	Single Instruction

Data

These 18 switches may be read indirectly into the accumulator by the execution of a OAS (OR accumulator content with switches content) instruction in an operating program. Data may be inserted manually into the machine with these switches by means of the DEPOSIT and DEPOSIT NEXT keys.

Address

The memory address required by the START, READ IN, EXAMINE, and DEPOSIT keys is supplied by these switches. When one of these keys is depressed, the address given by the address switches is loaded into the MA (memory address) register and the key instruction is executed.

Register Group

Selects which of two groups of 12 registers to be selected by the register select switch. In the OFF position the normal registers may be viewed while the ON position allows viewing of the maintenance registers. (See Register Select Switch).

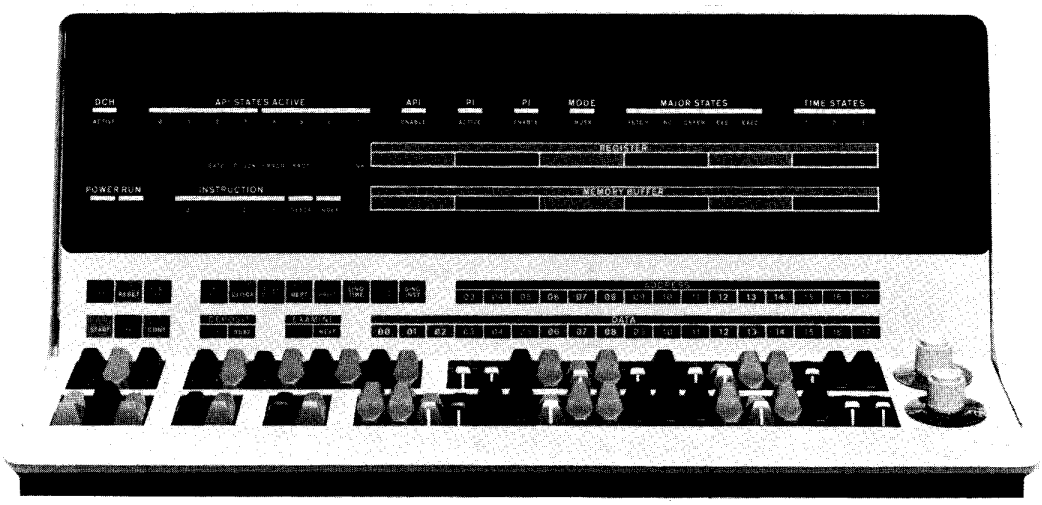


Figure 9-1 PDP-15 Console

Clock

The ON position disables the real-time clock from issuing program interrupt (PI) requests.

Repeat

With this switch in the ON position, the processor will repeat the key function depressed by the operator at the rate specified by the repeat clock.

Start - program execution will restart at the repeat speed after the machine halts.

Execute - the instruction in the data switches will be executed at the repeat clock rate.

Continue - program execution will continue at the repeat speed after halting.

Deposit: This, Next, Examine: This, Next - the Deposit, Deposit Next or Examine Next function will be repeated.

Depressing STOP or turning off the repeat switch will halt the repeat action.

Single Time

Halts execution of the program after each time state has been completed.

Single Step

Halts execution of the program after each major state has been completed.

Single Instruction

Halts program execution after each instruction has been completed.

OPERATE CONTROL SWITCHES

The following momentary contact switches are on the console panel.

Start

Initiates program execution at the location specified by the address switches.

Execute

Executes the instruction in the data switches and halts after execution.

Continue

Resumes program execution from the location specified by the program counter (PC). Also used to advance machine while in single time, step or instruction.

Stop

Halts the processor after the completion of the present instruction.

Reset

Generates a power clear signal which clears all active registers and all control flip-flops except the program counter (PC).

Read-In

Initiates the read-in of paper tape punched in binary code (each set of three 6-bit lines read from tape forms one 18-bit computer word). Storage of words read-in begins at the memory location specified by the ADDRESS switches. At the completion of tape read-in, the processor reads the last word entered and executes it.

Deposit This

Deposits the contents of the data switches into the memory location specified by the address switches. After the transfer, the memory locations address is in the OA (operand address) register and the contents of the data switches are in the MO (memory out) register.

Deposit Next

Deposits the contents of the data switches into the location given by OA + 1. This permits the loading of sequential memory locations without the need of loading the address each time.

Examine This

The address is loaded into the OA (operand address) register. Places the contents of the memory location specified by the address switches into the memory input buffer register.

Examine Next

Places the contents of the memory location specified by the OA + 1 (operand address plus 1) into the memory buffer register. Sequential memory locations may be examined using the Examine-Next switch.

Protect

Used only when KM15 Memory Protect option is installed. Causes system to start in protect (user) mode when START switch is pressed.

Bank Mode

Causes system to be in bank mode when START switch is pressed, permitting 8K direct memory addressing. If not set, system will be in page mode, permitting 4K direct memory addressing.

SPECIAL SWITCHES

Register Select

The 12 position Register Select rotary switch can select the following registers for viewing in the REGISTER indicators (under control of the Register Group switch).

Register Group Switch OFF

AC	Accumulator
PC	Program Counter
OA	Operand Address
MQ	Multiplexer Quotient
PL, SC	Priority Level/Step Counter
XR	Index Register
LR	Limit Register
EAE	EAE
DSR	Data Storage Register
IOB	I/O Bus
STA	I/O Status
MO	Memory Out

Register Group Switch ON

A BU	A Bus
B BU	B Bus
C BU	C Bus
SFT	Shift Bus
IOA	I/O Address
SUM	Sum Bus
M1	Maintenance 1
M2	Maintenance 2
MDL	Memory Data Lines
MA	Memory Address
MB	Memory Buffer
MST	Memory Status

Maintenance 1 and 2

Two positions of the Register Select switch M1 and M2, in conjunction with the Single Time switch, provide a visual display of active control and gating signal levels. For a given instruction, as the processor is stepped through the time states of each major state, the REGISTER indicators display a predictable sequence of enabling and strobing signals. The visual display relieves personnel of a large portion of the signal tracing

normally associated with the maintenance of electronic systems. The sequences or maintenance "status words" are supplied as part of the system maintenance data.

Power/Repeat Rate

A variable-pot/switch provides the POWER ON/OFF control at one end of its rotation and variable repeat speed (approximately 1 Hz to 10 kHz) over the remainder of its rotation.

INDICATORS

The console indicator panel displays the following:

Memory Buffer (18 bits)
Register (18 bits)*
Link
Power
Run
Instruction Register
  IR 0-3
  Defer
  Index
Extended Enable
Clock Enable
Parity Error
Protect
DCH Active
API States Active, 0-7
API Enable
PI Active
PI Enable
Index Mode
Major States
  Fetch
  Increment
  Defer
  EAE
  Execute
Time States
  1
  2
  3

*The "register" indicators display the content of the register, bus or status word that is selected by the setting of the Register Select switch.

CHAPTER 10
SYSTEM SOFTWARE

The PDP-15 software comprises the following major systems:

BACKGROUND/FOREGROUND

ADVANCED MONITOR

DISK OPERATING SYSTEM (DOS-15)

BATCH OPERATING SYSTEM SOFTWARE (BOSS-15)

RESOURCE SHARING EXECUTIVE (RSX–PLUS III)

BACKGROUND/FOREGROUND

Background/Foreground, a two-job DECtape or disk-based real-time monitor system, is an intergrated set of software designed to meet the demands of research, engineering, and industrial environments, where one or more real-time tasks typically require continuous responsiveness from the computer but do not use 100 percent of its capacity.

Under control of the B/F Monitor, real-time tasks are handled in the computer foreground and have immediate call on the system's resources via interrupts. Background time (time left over between service calls for the real-time tasks) is available for program development and testing or other low priority computation in an environment similar to that of the Advanced Software System. The Background/Foreground Monitor contains all the supervisory controls necessary for concurrent processing of background and foreground tasks.

Features of B/F include the following:

DECtape or Disk Resident System Software

All B/F System Software resides on either DECtape or Disk.

Interactive Operation

An interactive keyboard/program monitor permits device-independent programming and automatic calling and loading of system and user programs.

Foreground Priority

Foreground takes precedence over Background in the following ways:

Core allocation: Foreground is loaded first so that it can reserve whatever core it needs.

Peripherals: Foreground, because it is loaded first, stakes out those I/O devices which it will use. Those which cannot be shared with Background are then under exclusive control of the Foreground job.

Execution: Foreground execution is triggered by either clock or external interrupts. Four levels of Foreground priority are defined so that several tasks can be combined into one Foreground job. Background runs only when there is nothing to be done in the Foreground.

Foreground Protected

The Foreground job is protected from the Background job by both software checks and by memory protection hardware.

DECtape/Disk File Structure

Allows DECtape and disk to be treated as directoried (named file oriented) devices or as sequential access (non file oriented) devices.

Dynamic Storage Allocation

The available DECtape and disk storage is automatically allocated for optimum storage utilization.

Conversational Mode

System Utility Programs interact with the operator/user in a simple, conversational manner.

Programming Languages

Several programming languages are offered: FORTRAN IV, FOCAL, MACRO-15.

I/O Device Handlers

Data and file manipulating I/O device handlers are supplied for standard system peripherals, allowing device independence and overlapped computation and I/O. Mass Storage devices, such as DECtape, Disk and Magtape can be shared between the Background and Foreground jobs.

Programmed Monitor Commands

Input/Output programming is simplified by the use of a set of system commands which are standardized for system-supported I/O devices.

Bank and Page Modes

Choice of 8K (Bank Mode) or 4K (Page Mode) direct addressability. Page Mode operation allows address modification via the index register.

ADVANCED MONITOR SYSTEM

Through executive control of DECtape, the Advanced Monitor provides fully automatic operation, including batch processing, keyboard interaction, and real-time control. The Advanced Monitor System uses a large command set to direct system operation. These commands perform three major functions:

a. Provide information about the system such as commands available and their functions; system configuration, error diagnostics, the standard logical-physical I/O device associations, I/O level programs available (device handlers), special memory registers and their functions.

b. Permit the standard physical-logic device associations to be modified; thereby enabling the dynamic allocation of devices at load-time. (This is a natural extension of device independence.)

c. Supervise the loading and execution of all system and user programs, their associated I/O device handlers, and library subroutines, in addition to the generation of error messages and recover procedures.

Coupled with keyboard control of system programs, the Advanced Monitor enables the user to perform editing, assembling-compiling, loading, debugging and running functions in a straightforward manner.

The Advanced Monitor (Figure 10-1) comprises a command decoder, IOPS routines, real-time clock handler, Teletype handler, error detector routine, and device assignment table (.DAT).

The bootstrap loader always resides in upper memory and is responsible for loading the Monitor into lower memory. Return calls from system or user programs cause restoration of control to the Monitor.

The Monitor command decoder recognizes requests for system programs and loads the system loader to bring in the requested program. In response to control cards or keyboard commands, it also manipulates the device assignment table to provide the device independence. The Monitor input/output system routines (IOPS) include data-handling subroutines, device handlers, and an interrupt system as well as the Teletype keyboard and printer. All other IOPS device handlers are stored on the system device until required by object programs.

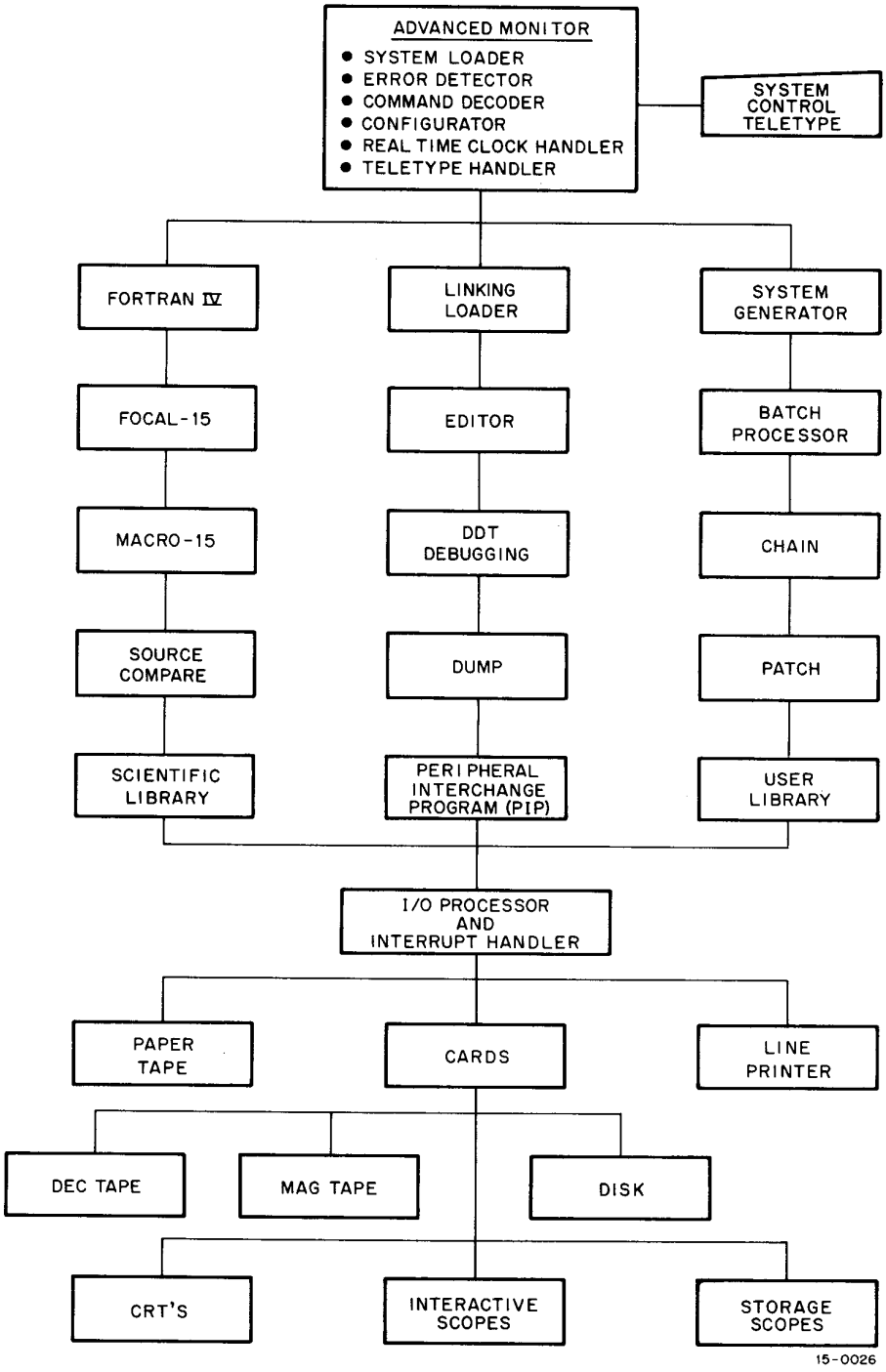


Figure 10-1 Advanced Monitor System

The Monitor also contains a device assignment for each table entry that may be used. Since the contents of the table can be altered by commands to the Advanced Monitor, actual I/O devices may be changed without altering the program references to these devices.

DISK OPERATING SYSTEM (DOS-15)

DOS-15 includes all the features of the Advanced Monitor plus full disk file support, and it is upward compatible with the Advanced Monitor. Programs written for the Advanced Monitor can compile and execute under DOS.

DOS provides the following features over the Advanced Monitor.

- Disk System Executive Control
- Disk Random Access File Capability
- User-to-User Disk File Protection
- Expanded Executive Functions
- Expanded I/O Facilities
- Dynamic Buffer Allocation

With the PDP-15/76, DOS-15 also provides full input/output spooling for devices interfaced to the Unibus.

Minimum DOS Configuration

The minimum PDP-15 system necessary to run DOS-15 is:

a. A PDP-15 central processor with console terminal, EAE, real-time clock, 16K of memory, and paper tape reader/punch.

b. One of the following disk systems: RF15/RS09, RP15/RP02, or UC15/RK05.

c. Either DECtape or magtape.

BATCH OPERATING SYSTEM SOFTWARE (BOSS-15)

BOSS-15 is a comprehensive batch processor and front end processor to DOS-15. DOS-15 and BOSS-15 are fully compatible and share the same basic operating system.

BOSS-15 provides commands to facilitate complete automatic operation in processing streams of programs. BOSS-15 commands are simple to use but allow complete access to all the features of DOS-15.

A PDP-15 operating under BOSS maximizes its throughput by processing jobs as fast as possible, without waiting for operator interaction.

BOSS-15 has a unique procedure file driven command language for maximum flexibility. The user can modify and add his own batch commands easily and efficiently.

Minimum BOSS Configuration

The minimum PDP-15 system necessary to run BOSS-15 is:

a. A minimum DOS-15 configuration.

b. A card reader (If a CR11 series interface, spooling is provided.)

c. A line printer (If a LP11 series interface, spooling is provided.)

COMMON DOS/BOSS SOFTWARE

The following software is provided with all DOS-15 or BOSS-15 systems.

FORTRAN IV

The FORTRAN IV operating system for the PDP-15 is the major method of implementing applications programs. It originates from a base of ASA standardization for compatibility, and moves into extended features to satisfy the

needs of commercial applications and state-of-the-art programming techniques. Some extended features are:

- Free Format I/O
- Encode.Decode
- Multiple Subroutine Entry and Exits
- Part Word Notations for Bit and Byte Manipulations
- Negative Step DO Loops

ALGOL 60

ALGOL 60 is an alternative to FORTRAN for scientific notation. The implementation of ALGOL is a subset of the ETMA 1 level report with some added facilities. Some of the extended features are:

- Externally compiled procedures in either ALGOL, FORTRAN, or MACRO
- Full and mutual recursion between procedures
- BLANK COMMON variables in FORTRAN and OWN variables in ALGOL are directly equivalent
- Data for overlays need not be in COMMON to pass from one overlay to the next
- Free Format I/O

FOCAL-15

FOCAL-15 is an on-line, interactive (conversational) algebraic language designed to help scientists, engineers, and students solve numerical problems. The languages consists of short, easy-to-learn English imperative statements. Mathematical expressions are usually typed in standard notation. FOCAL puts the full calculating power and speed of the PDP-15 under easy conversational control. For example, FOCAL can be used for simulating mathematical models, curve plotting, and handling sets of simultaneous equations in n-dimensional arrays.

MACRO-15

MACRO-15 is the machine language assembler for the PDP-15. It features full MACRO capability including conditionals and repeats. The MACRO assembler allows device independent programming at a machine language level, greatly easing the I/O programming burden. The output of the MACRO-15 assembler is completely compatible with the output of the FORTRAN and ALGOL compilers. Among the features of MACRO-15 are:

- Ability to define and call nested MACROs
- Conditional assembly based on the computational results of symbols or expressions
- Repeat functions
- Boolean manipulation
- Optional octal/symbolic listing
- Two forms of radix control (octal and decimal) and two text modes (ASCII and 6-bit trimmed ASCII)
- Global symbols for easy linking of separately assembled program
- Choice of output format: relocatable, absolute binary (checksummed), or full binary (unchecksummed); capable of being loaded via the hardware READ-IN switch

MACRO-11 (Unichannel systems only)

MACRO-11 assembler is used to develop programs for the PDP-11 peripheral processor of PDP-15. Its primary use is for system tailoring. MACRO-11 is used to assemble the peripheral executive, which, in turn, supervises all I/O activity on the Unibus including disk system I/O and full input/output spooling.

Text Editor

The PDP-15 Text Editor is fully interactive and device independent. Files of text can be input from any medium (such as disk, magnetic tape, paper tape, and cards), edited by using powerful interactive commands, then put out to any medium. The available commands enable searching and modifying all concurrences of specific character strings throughout the file, and of performing substitutions, additions, and deletions at will. The Text Editor is especially valuable in program conversion and modification environments.

Graphic Text Editor

The Graphic Text Editor builds upon the Text Editor to give high-speed verification of output. The user can visually page through a file, modifying text at will. And modifications can be visually verified without waiting for hard copy. A version of the Graphic Text Editor is available for either the refresh interactive display or the storage point plotting display.

Text File Preprocessor

For Batch facility users, text editing of files on mass storage devices is achieved by using the line editing facility. Through simple command cards, line addition, substitution, or deletion can be performed automatically and under Batch control. Original files can be completely replaced by edited files, or new files may be created without disturbing important permanent ones.

Text File Comparator

The file comparator is used when the permanent file storage medium is predominantly mass storage (such as disk pack or magnetic tape). Early versions of files can be compared with later versions to assess the extent of modification. This can be particularly useful when debugging extensively modified files because it enables the user to keep track of possible modifications that could have introduced new problems. The text comparator checks two input files, line for line, for any variations. It has a special look-ahead feature to allow resynchronizing if the files differ by multiple lines.

Peripheral Interchange Program

The Peripheral Interchange Program (PIP) is a general file utility program with extended features. PIP is primarily used to transfer files between system I/O devices, such as cards to disk, disk to tape. It is also used to combine and segment files, verify the integrity of files, rename, duplicate, and delete files. PIP is also used to list mass storage directories and convert data formats.

Magnetic Tape Utility

The Magnetic Tape Utility (MTDUMP) is a general file utility specifically oriented toward industry compatible magnetic tape. Such functions as backspacing over files, skipping files, and creating file markers can be performed with this program, which also allows dumping of files on the printer and other devices.

Interactive Debugger

The Dynamic Debugging Technique program (DDT) is a fully interactive method of controlling the execution of a program while allowing on-line program examination and modification. DDT runs in conjunction with the object program but adds only a small space overhead to the operating system. Programs execute at their full speed except when intercepted by a DDT command or a pre-fixed break point.

Interactive debugging has the advantage of examining events as they happen and being able to place patches to test possible solutions to problems before any permanent changes are made.

Library Update

The PDP-15 Operating System enables a library system of commonly used programs and subroutines to be automatically loaded into core by the systems loaders. The FORTRAN, ALGOL, and MACRO programs are supported by a

library that is supplied with the system. It contains arithmetic routines, data formatting routines, and I/O control routines. The user can create his own libraries from which the systems loaders can automatically search and load.

The maintenance of these libraries is provided by the UPDATE program. UPDATE permits the creation and modification of library files through the addition, substitution, and deletion of routines.

Core Image Save/Restore

The PDP-15 provides full core image dumping on mass storage. This facility can be used in debugging to allow later dumping of core image to line printer listing, or the restoration to core if debugging has to be interrupted prematurely. The core image save/restore facility is also helpful for rapid loading of prepared programs.

Loaders

The PDP-15 is equipped with four loaders to accommodate any eventual type of program development. They are: 1) the hardware bootstrap loader, 2) the absolute loader, 3) the relocatable linking loader, and 4) the absolute PDP-11 loader.

The hardware loader is an integral part of the PDP-15 hardware and is used for cold start loading of bootstrap loaders or monitor systems. If you are designing your own software system or modifying ours, the MACRO-15 assembler allows you to output programs in a format suitable for the hardware loader.

The absolute loader is designed to accommodate the loading of monitor systems and other major items that require special areas in core for their operation. The MACRO-15 assembler provides optional absolute output.

The linking loader is the most commonly used loader. It accommodates output from the FORTRAN and ALGOL compilers or the MACRO-15 assembler. All programs are relocated in memory for greatest efficiency. The user can request a memory map of loaded programs to help in debugging and determining core utilization.

The absolute PDP-11 loader is used to load the PDP-11 peripheral processor with the Peripheral Executive (see section on the I/O system) or to load stand alone programs when the PDP-15 is being run in dual independent mode. The output of the MACRO-11 assembler accommodates the absolute loader.

Systems Generator

Tailoring a complete operating system to your particular hardware configuration and computational needs is accomplished by the fully interactive Systems Generator Program (SGEN).

SGEN operates in conversational mode, asking pertinent questions about specific items in the system.

You can delete particular programs and I/O handlers to conserve disk space. You can rearrange the placement of items on disk to speed up access to the most commonly used system programs. And you can add your specially written device handlers for nonstandard devices.

With SGEN, you can reserve space for programs you wish to be treated as system programs. This feature is used in conjunction with the Systems Patch utility.

Systems Patch

Patch plays several roles in conjunction with SGEN to give you complete control over system tailoring.

Patch allows you to store any user-written programs as system programs on disk space reserved by SGEN. Thus, any user may access these programs via system commands. By defining Procedure Files, the batch user can be given access to these new system programs with BCL commands.

Patch is also used to modify system files to correct small program errors without complete reassembly and resystem generation.

Together SGEN and Patch free the user and his installation from constraints imposed by the operating system, and thus provide increased versatility in many applications.

Extensive System Peripheral I/O Handlers

The PDP-15 provides compatible device-independent I/O drivers and handlers for its peripherals. When using I/O peripherals, this network of handlers gives your applications programs maximum flexibility. Complete programs can be written without regard for the particular I/O devices they will use (this is true for all languages of the PDP-15). The user need only be concerned with specific I/O devices at run time, and not when the program is being written; thus, data can come from tape instead of the card reader, or the output can go to disk instead of spooled to the line printer. Only a single command is necessary, and the appropriate device handler is then loaded and used for I/O.

All the system programs described are device independent. Therefore, all computational aids provided with the PDP-15 can work to and from any peripheral device.

RESOURCE SHARING EXECUTIVE (RSX–PLUS III)

RSX–PLUS is a multifaceted system that incorporates elements of batch and interactive processing, multiprogramming, real-time control, multi-instrument data acquisition, multiple scope interactive graphics, and multiple terminal communications. This system is designed to meet corporate and institutional computer center and laboratory demands for the varied processing capability of a large computer. By tying together various activities, RSX–PLUS provides rapid access to common data bases in a centralized computer facility.

RSX–PLUS derives its power from the PDP-15's multiple processors and extensive CPU/peripheral options. RSX–PLUS is a multiprogramming executive that supports hardware protection and relocation, batch processing for on-line task development, interactive processing for on-line file manipulation and program modification, and extended memory and CPU/peripheral support. It also introduces multiscope interactive graphics and a comprehensive data management system to the shared environment.

The RSX–PLUS communications capability applies to multiple terminal, multiple instrument, and multiple small peripheral computers with the PDP-15 as the centralized host facility.

DOS-15 software and RSX–PLUS are upward compatible. Programs written in FORTRAN or MACRO-15 can be executed in the shared environment with little or no modification. Files created on mass storage need not be modified for use under RSX–PLUS.

For the most demanding requirements and the most flexible operation, RSX–PLUS and DOS-15 software can be co-resident on the same disk. Switching from one system to the other is accomplished instantaneously by keyboard command. In this manner, an installation can take advantage of both systems easily and efficiently and avoid long, drawn-out system generations.

Invisible resource sharing is the key to the RSX–PLUS III executive, i.e., execute different tasks simultaneously but treat each user as if he were sole user of the system.

Basically, the function of RSX–PLUS III is to dynamically allocate system resources. The system allocates core memory, handles interactive terminals, loads and executes programs, outputs to system devices, and operates BATCH. The result is excellent response for critical tasks and optimum throughput and efficiency for other system activities.

With RSX–PLUS III, the number of tasks is limited only by the size of the system's core memory and its disk storage capacity. Task priorities are specified within the program and may be changed on-line, if necessary, by the operator.

Tasks can be written in any one or a combination of three languages: MACRO – a powerful assembly language; FORTRAN IV – an extended ASA standard version of FORTRAN; and RASP – a specialized problem-oriented language for system data management.

RSX–PLUS III Features:

- Fully queued I/O processing
- Protection of tasks from each other through use of hardware memory relocation
- Multiscope interaction graphics
- On-line program editing, compilation, and testing

User tasks can be:

- Permanently or temporarily locked in core memory.
- Disk resident and loaded rapidly into core when needed.
- Structured in overlays.
- Written, compiled, and executed without disturbing on-going operations.
- Written using DIGITAL-supplied subroutines.
- Run in "privileged" executive mode or in "protected" user mode.
- Coupled into processing networks by RASP.
- Swapped out on request.

Task execution is triggered by:

- External interrupts.
- Time delay or time of day.
- Synchronization with another task.
- Request by another task.
- Operator request.
- Batch processor request.
- RASP, when a queue has a specified amount of data or when a high priority item of data becomes available.

Task priority can:

- Be assigned at 512 separate software levels.
- Be modified via operator command.
- Be higher than that of the executive for critical tasks.

Requests to the executive can come from:

- The operator through the console.
- A user task written in FORTRAN or MACRO.
- The RASP operating system.
- Through interrupt level routines (re-entrance).

Core is:

- Expandable to 131,072 words of memory.
- Divided into any number of program partitions and common areas.
- Used to store the executive (6,500 words).
- Divided into partitions that vary in size from 256 words to 32,768 words.

Disk storage includes:

- A high performance fixed-head internal disk up to 2 million words.

- High capacity, high speed disk packs up to 25 million characters per pack.

Disk utilization includes:

- Dynamic allocation and de-allocation of disk space.
- Dynamic creation and deletion of sequential and random files.
- File protection against multiple access conflict.
- Storage of core image task files for rapid loading on demand.
- System save area for rapid restart.
- System device storage of operating system, user tasks, and data.
- Disk space limitation on sequential files.
- File compatibility with DOS on the disk pack.

Related programs communicate via:

- Common blocks of core.
- Disk-resident "named" files.
- RASP-15 intercom channel.

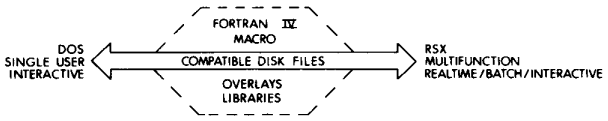
Reentrant Executive can be:

- Entered from any interrupt level.

The full benefits of multiprogrammed operation are available on many configurations. With RSX–PLUS III, the PDP-15 has the power, storage capacity, and multifunction capability to handle present needs and adapt smoothly to future requirements. It is a powerful computing center that does not sacrifice real-time needs.

DOS and RSX share the same advanced disk file structure. Programs running under RSX–PLUS III can read files created by programs running under DOS; and programs running under DOS can read files created by programs running under RSX–PLUS III. This file structure permits an unlimited number of files to be open simultaneously, and provides multiple file directories to protect one user from another. This compatibility extends even further, allowing both DOS and RSX to reside on the same system's disk at the same time. Each monitor is aware of the other, and thus by simple typed commands, systems may be switched from DOS to RSX or RSX to DOS in just 2 seconds.

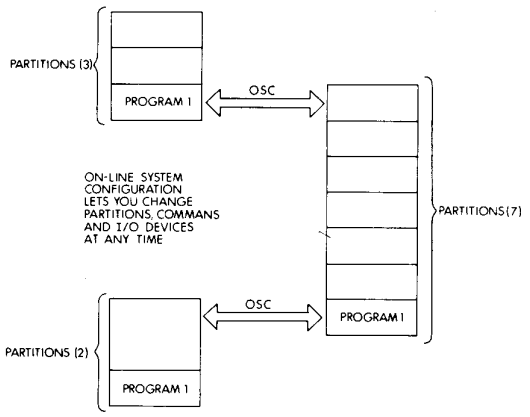
RSX can thus be used in small systems to perform real-time data acquisition and control or be switched to DOS for data analysis.



RSX–PLUS III – Eliminating "System Generation" Procedures

RSX systems are used in virtually every application from data entry to particle physics. Every system will now be even more useful because of a new software technique: On-Line Systems Configuration (OSC).

In seconds, an RSX–PLUS III system can be reconfigured to the proper arrangement of memory partitions, program assignments, common block usage, and I/O devices. And this can be done without disturbing in-progress, real-time tasks. The tedious "system generation" procedures, which have locked conventional real-time monitors into fixed applications, have been entirely eliminated. RSX–PLUS III is a truly interactive real-time system.



On-line systems configuration is a combination of system components that enables the user to modify and monitor the state of the software system. Because of hardware memory relocation, memory partitions can be moved without recompiling, relinking, or reloading user programs.

RSX–PLUS III – Offering Simultaneous BATCH

The PDP-15 solves problems quickly; however, more versatility is available using the BATCH processing capability of RSX–PLUS III.

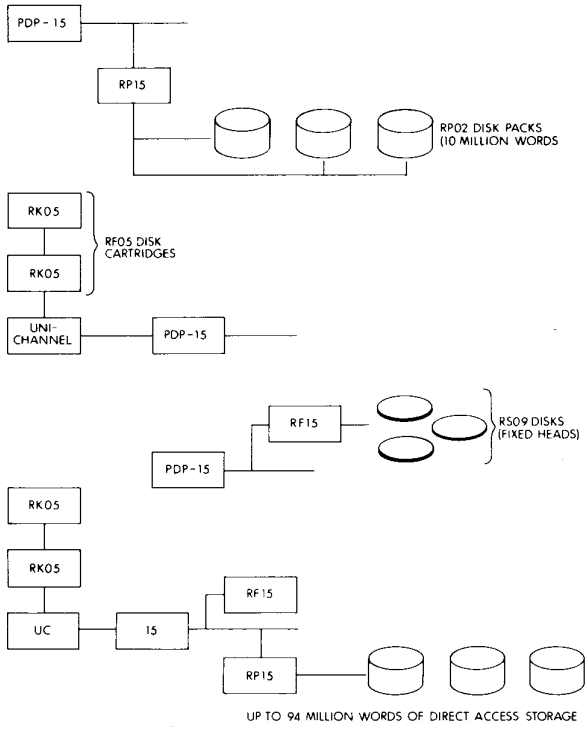
RSX–PLUS III BATCH provides:

- Job header and trailer pages
- Job accounting and account usage summaries
- Full FORTRAN, MACRO, and loader facilities including overlays
- System protection from an undebugged job
- Batch operation from cards, magtape, or disk files
- Convenient operator communications and control
- SLIP – a sophisticated file editing and update program with search capabilities
- Full spooling (with Unichannel 15 ppu)

RSX/BATCH can be used concurrently with interactive on-line processing and real-time experiments or as a high capacity batch-only system in as little as 32K main memory.

RSX–PLUS III supports all disks for PDP-15 computers:

RF15 fixed head disk	250K wds/ea
RP15 disk packs	10 million wds/ea
RK15 disk cartridges	1.2 million wds/ea



Minimum RSX–PLUS III Configuration

The minimum PDP-15 system necessary to run RSX-PLUS III is:

A PDP-15 Central Processor with console terminal, EAE, Real-Time Clock, Automatic Priority Interrupt, Memory Protect/Relocate, Paper Tape Reader/Punch, and 24K of memory.

One of the following disk systems:

RF15/RS09, RP15/RP02, or UC15/RK05

Either DECtape or magtape

UNICHANNEL SOFTWARE (UC15)

UC15 systems feature a fully programmable peripheral processor to handle the processing and time-consuming load of heavy I/O environments. This approach leaves the PDP-15 free to do computation at its highest speed.

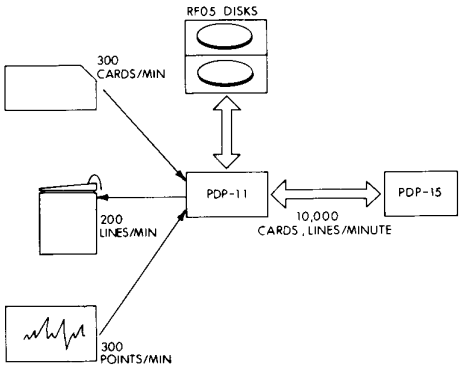
The full spooling features of the UC15 are available on DOS-15, BOSS-15, and RSX–PLUS III.

For DOS-15 users, UC15 spooling eliminates waiting for long printouts to complete before going on to other functions. For BOSS-15 users, UC15 spooling almost doubles the throughput. For RSX–PLUS III, UC15 spooling prevents tasks from tying up partitions to do slow speed I/O.

PERIPHERAL EXECUTIVE

The UC15 software consists of a peripheral executive, PIREX. PIREX resides in the local memory of the PDP-11 peripheral processor. The executive receives requests for I/O from the PDP-15 and initiates the transfer of data to and from shared memory and peripherals interfaced to the Unibus. The executive communicates with the PDP-15 operating system via interrupts, shared memory, and control words that are transferred over the communications interface.

UC15 software provides support for the RK05 disk and the Unibus family of card readers, line printers, and plotters, all of which can be spooled.



MUMPS

MUMPS (MGH Utility Multi-Programming System) is a multiuser software system that provides time-shared use of a common data base management information system. MUMPS includes a multiuser monitor that allows up to 22 simultaneously active 1000-word user partitions. (More than 22 are allowed if user partitions are made smaller.) A simple, high-level, interactive program language provides optimum interface between the users and the system, yet provides powerful random-access storage and retrieval facilities and utility routines that perform system housekeeping functions.

MUMPS is a text-oriented information system that provides users with a solution to a variety of information storage and retrieval problems, such as order entry systems, automating medical or customer records, information directories, and catalogs.

The MUMPS language is similar to FOCAL and BASIC and is easy to learn. The system interacts with users enabling them to write programs, see

the results, make modifications, and re-run their programs in a single session, without costly delays. User programs are always stored and used in their source form to avoid confusion and save valuable tape and core storage space required for other user terminals.

A minimum MUMPS system configuration consists of:

- A standard PDP-15/20 System with the KW15 Real-Time Clock option.
- An RF15 DECdisk Control with two RS09 DECdisk Drives.
- A DC01EB Line Scanner that services user terminals.
- Up to seven user terminals.

This minimum configuration, with the Advanced Monitor and MUMPS software, provides up to seven users with a 500,000-word common data base information system. It can be easily expanded to include additional random-access secondary storage facilities, such as the RP15/RP02 Disk Pack System, and tertiary magnetic tape storage, plus up to 128K of core storage. Equipment expansion requires only the reassembly of the existing software system.

OTHER PDP-15 SOFTWARE

Statistical Library

The PDP-15 statistical library (STATPAC) is a moduler collection of FORTRAN programs that enables the user to perform a variety of descriptive statistical operations on data. With STATPAC, the user obtains accurate results with a minimum knowledge of the computer system. The modular construction of the package makes easy, inexpensive expansion and tailoring possible for unique applications. STATPAC is designed to operate on a standard PDP-15 within the device independent environment of the operating software. Several major advantages are realized:

- Data can be input from many different devices.
- Lack of a particular input/output device can be used. Hard copy can be produced on-line or spooled for later off-line output.

STATPAC consists of three basic modules. The control module performs the executive function of communication between modules and the user. An input module prepares and converts the input data to a format required internally by the analytical modules. Analytical modules consist of descriptive statistics, regression analysis, factor analysis, and variance analysis.

Engineering Library

The PDP-15 engineering library is a collection of FORTRAN and assembler language routines developed by the Division of Automatic Control of the Lund Institute of Technology in Sweden. These routines are particularly applicable to engineering environments where maximum computation speed is required and routines have been optimized for the PDP-15 wherever possible. The package consists of the following five major groupings:

1. Matrix Operations

   Solution of Linear Equations

   Computation of Eigenvalues and Eigenvectors

   Matrix normalization

2. Polynomial Operations

   Test for location and zeros

   Root solvers

   Polynomial manipulations and multiplications

3. Integration of Differential Equations

   Non-Linear Equations

   Linear Equations

4. Signal generation

   Random number generation of gaussian and rectangular distributions

   Random gaussian vector generators

5. Fourier Transforms

   Cooley-Tukey algorithm

   Auto-Covariance computation of a time series

Science Library

The PDP-15 science library (MATH PACKAGE) has been built around Dr. Philip Bevington's book, *Data Reduction and Error Analysis for the Physical Sciences* (McGraw-Hill publication). This book provides a detailed insight into the methods used in experimental research for the reduction and error analysis of physical data. The techniques described are illustrated by FORTRAN routines, which can be used in their entirety as library routines by individuals working in the physical sciences. These routines, provided in the PDP-15 science library, fall into the following categories:

- Distributions

  Binomial

  Factorial

  Poisson

  Gaussian

  Lorentzian

- Estimates of Mean and Errors
- Least-squares Fit to a Polynomial

  Least-squares fit with a polynomial curve

  Least-squares fit with a Legendre polynomial

- Multiple regression
- Goodness of fit
- Least-squares fit to an Arbitrary Function
- Data manipulation

  data smoothing

  interpolation between data points

  integration of the area beneath two data points

- Matrix Inversion

Continuous Systems Modeling Program (CSMP)

Block CSMP is a dynamic continuous system modeling program that operates interactively on System 76. Its primary function is to assist in the accurate simulation of dynamic systems, modeled by ordinary differential and difference equations. It is upward compatible in all major aspects with IBM 1130-CSMP. Block CSMP for PDP-15 is written completely in FORTRAN and can solve problems represented by as many as 75 operational blocks, 25 of which may be integrators.

"Block" is employed in the title of this system to stress the relationship of the input language to the block diagram method of representing dynamic systems (as opposed to equation-oriented systems such as IBM 360-CSMP and IBM 360-CSSL). Because it offers distinct advantages, the block diagram approach to modeling

and simulation of dynamic systems has a long tradition of successful employment in analog and hybrid computation. The operational environment provided by the PDP-15 is essentially the same as that available on the analog computer. However, Block CSMP offers the advantages of far greater precision, accurate documentation of both the model and the simulation, and freedom from scaling.

Commercial Subroutine Package (CSP)

The PDP-15 commercial subroutine package is designed to give the scientific programmer the tools with which to implement applications packages for business data processing. Because FORTRAN was primarily designed for scientific notation, the package consists of a collection of FORTRAN subroutines that allow the creation of commercially-oriented programs with FORTRAN. These subroutines expand the capability of FORTRAN by giving it such features as unlimited precision decimal arithmetic and character manipulation. The subroutines also provide facilities for editing special characters in data arrays and filling in special tabs within text. The commercial subroutine package is compatible with the IBM 1130 commercial subroutine package and provides a similar capability.

Pulse Height Analysis Systems

Two PHS-15 systems are designed to provide a natural growth route for the expanding laboratory. Both systems give the most powerful analyzer capability possible considering the system's size, yet retain the balance and flexibility of an interactive computer system. Hardware consists of a standard nuclear interface, a general purpose computer, a display oscilloscope, a command terminal and selected input/output devices.

Sort/Merge Utility

The PDP-15 Sort/Merge is a family of FORTRAN subroutines, designed to implement an in-core tag sort of any number of sort keys in either ascending or descending order. The Sort/Merge routines make use of all available core storage and will force a multiple phase sort if necessary. The Sort/Merge subroutines are divided into three distinct functions: 1) The first set of routines are those necessary only for the Sort/Merge operations. 2) The second set are subroutines for blocking disk records and may be used for programs other than Sort/Merge. 3) The third set of routines form a collection of routines that may be useful in commercial applications as well as Sort/Merge.

Since this package is a collection of subroutines, it may be easily incorporated into user written applications programs. Since the Sort/Merge phases are called separately, these routines may be overlaid to yield even more working space.

A shell sort algorithm is used in this package to take advantage of any ordering of the data already achieved by previous passes. The final pass of the shell sort is essentially a bubble sort. The sorting of two given records is done in a function subprogram external to the sort, which may be easily modified by the user.

The PDP-15 Sort/Merge also permits user rejection of records before the sort through an external function subprogram supplied by the user.

Hybrid Software

The PDP-15 hybrid software package is a collection of routines that allows parallel operation of the PDP-15/76 digital computer and a comparable analog computer. The package consists of over 60 routines that form the functions of data collection, communication, and control for a fully integrated hybrid system. The routines are callable either from FORTRAN or MACRO and are compatible with standard interfaces supplied by major analog manufacturers.

There are two major packages of hybrid software, one for Applied Dynamics interfaces and one for the Electronic Associates Incorporated interfaces. There is a third package that is part of DECUS (the Digital Equipment Computer Users Society) and interfaces to FOCAL. FOCAL is the interactive scientific language for the PDP-15. This package was developed at Carnegie Mellon University.

DECUS

The Digital Equipment Computer Users Society, encourages the exchange of programs and ideas among its several thousand members. Semi-annual meetings provide a forum for papers, while the program library allows the exchange of programs of common interest.

DIAGNOSTICS

MAINDEC Diagnostic Programs

MAINDEC diagnostic programs are provided for locating hardware malfunctions within the processor, memory, and I/O equipment. They run under a systems exerciser. Simple Teletype commands load and execute requested diagnostics.

The diagnostic programs make troubleshooting fast and straight-forward by selectively exercising every circuit in the machine. Instructions and procedures for loading, operating, and interpreting the results of diagnostic tests are written in clear, simple language, so that beginning maintenance technicians can use them easily.

Among the MAINDEC diagnostics are: The Basic Processor Test and the Extended Processor Test. The Basic Test incrementally checks the entire instruction repertoire, performing 1500 unique tests, and in each case, halts with specific instructions for the troubleshooter. If the Basic Test fails to detect the trouble, the Extended Test uses random number techniques to test the logic for many combinations of data manipulation and addressing problems, runs memory test patterns, performs system tests on I/O devices and controls, and many other tests.

A valuable tool for check-out and troubleshooting, MAINDEC diagnostics contribute to the high productivity of the PDP-15 by minimizing downtime.

CHAPTER 11
INSTALLATION PLANNING

INTRODUCTION

When planning to install any of DEC's computers, it is most important to refer to the *Computer Site Preparation Handbook* (DEC-00-ICSPA–A–D).

A brief synopsis of the content of the *Computer Site Preparation Handbook* follows:

The manual is divided into four chapters: GENERAL CONSIDERATIONS, ENVIRONMENTAL CONSIDERATIONS, ELECTRICAL CONSIDERATIONS and COMMUNICATION CONSIDERATIONS.

- GENERAL CONSIDERATIONS include discussions of how to plan beforehand for computer delivery and what factors to consider when planning the actual computer site.
- ENVIRONMENTAL CONSIDERATIONS cover a variety of important factors which can influence computer operations.
- ELECTRICAL CONSIDERATIONS involve the power and ground requirements of the computer and how to meet them.
- COMMUNICATION CONSIDERATIONS address the question of interfaces required for data transmission. Tables of recommended data sets are included.

PHYSICAL CONFIGURATION

The basic PDP-15 is housed in a standard 19-in. cabinet (Figures 11-1 through 11-3) with overall dimensions of 21-11/16 in. wide, 30 in. deep, and 71-7/16 in. high. The PDP-15 is painted black with grey end panels and two-tone blue console and logos.

All standard PDP-15 system logic is housed in a steel enclosure with cooling fans. Each cabinet uses a large fan which pulls filtered air in from the top of the cabinet – keeping the complete cabinet under pressure.

In the basic PDP-15 cabinet, the console, power supply, API, parity, memory protect, central processor, I/O processor, EAE, real-time clock power fail, and the first 8192 words of MM/MK15 Memory are all mounted in the front portion of the cabinet. Additional memory is mounted on the back door.

Other options are added to the PDP-15 by attaching 19-in. cabinets to either side of the basic system, according to the configuration shown in Figure 11-4.

Several options including disks, displays, industry-compatible tape transports, the card reader, line printers, and plotters are composed, in part, of free standing units.

PLACEMENT OF OPTIONS

Cabinets are numbered 1 through n with the numbers always running from left to right. All cabinets are standard DEC 19-in. type weighing 300 lb with net capacities of 500 lb recommended and 800 lb maximum. Each cabinet can hold eleven standard 5-1/2 in. mounting panels of logic plus a 5-1/2 in. indicator panel at the top. Figure 11-4 shows the placement of options and peripherals. Table 11-1 provides installation data for free-standing cabinets, peripherals, and options. Table 11-2 provides installation data for cabinet-mounted peripherals, controls, and options.

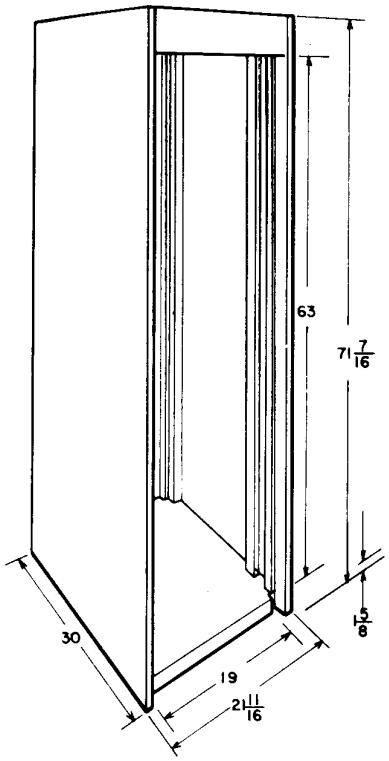


Figure 11-1 Basic PDP-15 Cabinet

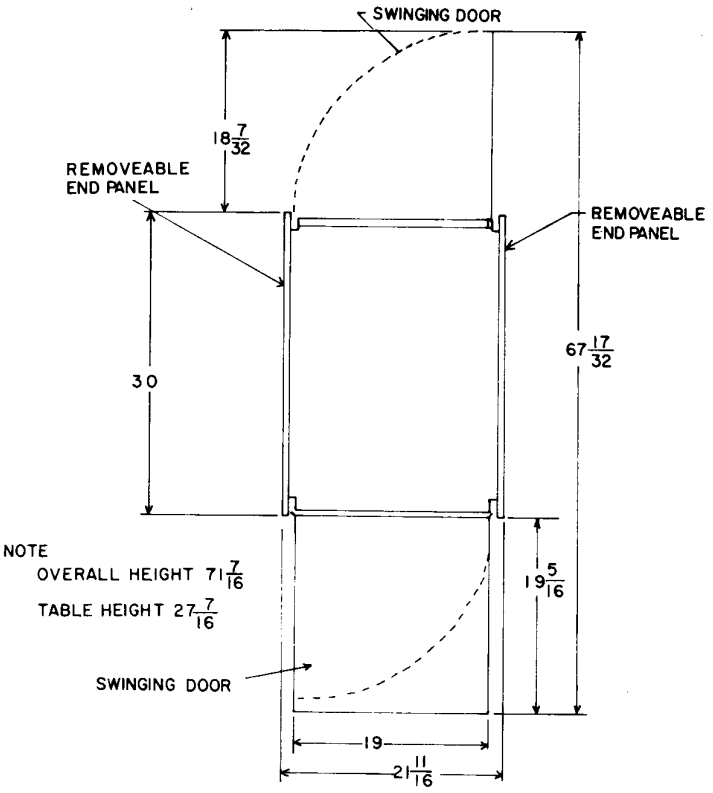


Figure 11-2 Top View of Basic PDP-15 Cabinet

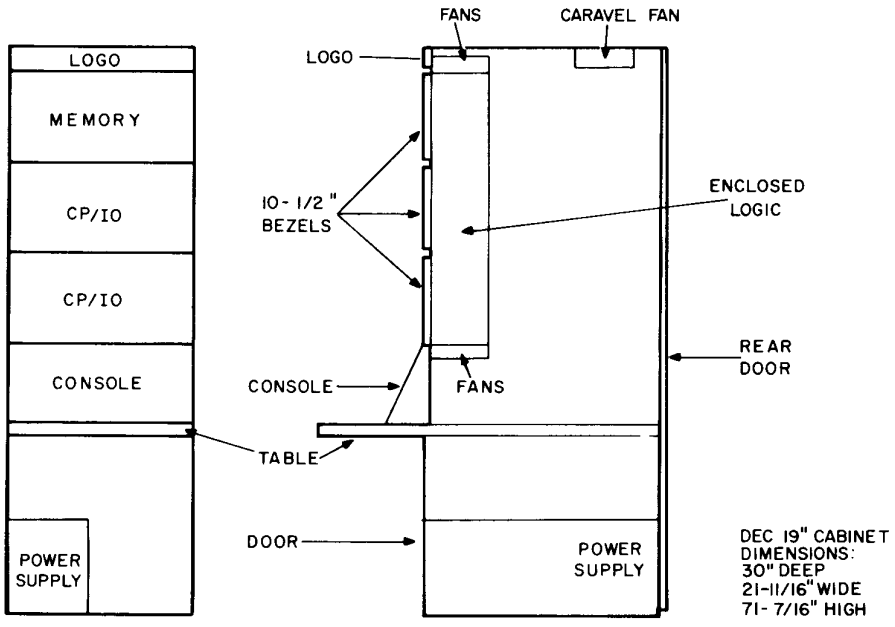


Figure 11-3 PDP-15 Basic Configuration

FRONT VIEW

H963A BAY 6L	H963A BAY 5L	H963B BAY 4L	H963C BAY 3L	H963N BAY 2L	H963U BAY 1L	H963D BAY 00	H963E BAY 1R	H963F BAY 2R	H963H BAY 3R	H963J BAY 4R	H963K BAY 5R	H963P BAY 6R	H963M BAY 7R	H963L BAY 8R	H963S BAY 9R	H963R BAY 10R	H963R BAY 11R
						PDP15											
RS09 #8	RS09 #5	RF15 SEE NOTE 4	INDICATOR	MM/MK15	DR15C	8K MEM MM/MK15	INDICATOR	INDICATOR	BLANK	SEE NOTE 1	AF01B OR ADC1/9	BLANK	INDICATOR	INDICATOR	INDICATOR	BLANK	BLANK
			INDICATOR		MX15B		IND. (FP15)	BLANK					BLANK		BLANK	AFC15 SEE NOTE 3	UDC15 SEE NOTE 3
				MM/MK15	RK05		FANS										
RS09 #7	RS09 #4	RS09 #2	RP15 SEE NOTE 4	FANS		CENTRAL PROCESSOR	API MEM PROT/REL KA15, KM15, KT15	TU56 #3 OR BLANK SEE NOTE 2	BLANK		FAN	AD15	LP15	VT15A	BD-15	BLANK	BLANK
				MM/MK15	RK05						AM09						
							DISPLAY	TU56 #2	TU56 #4		BLANK	BLANK	CR15		BLANK	SCREW TERMINAL CONNECTORS	
RS09 #6	RS09 #3	RS09 #1	FANS	MM/MK15	BLANK	CONSOLE KC15					AA05D						
				FANS	PDP-11/05		PC15	TU56 #1	TU56 #3		FANS						
				MX15A	FANS	TABLE					7006351	BLANK	BLANK	VT15B			
BLANK	BLANK	BLANK	BLANK		841-C RK11-E H740-D	POWER SUPPLY 715	BLANK	TC15	BLANK	TC59	BLANK	AA07			BLANK	BLANK	BLANK
				BLANK			FANS										
							BA15					AA15					
							DW15										
							PWR RECP	PWR SUPP		BLANK		POWER SUPPLY H721	POWER SUPPLY H721		POWER SUPPLY H721	POWER SUPPLY H721	POWER SUPPLY H721
14/6.5	14/6.5	23/8.5	25/7	42/15	41/14.4	150/25	25/10	20/7	18/6	11/3.5	21/7	30/12	30/12	18/6	30/12	30/12	30/12
2542	2542	3324	2737	5620	5630	9775	1475	2737	2346	——	2737	4692	4692	2346	4692	4692	4692
420/190	420/190	400/180	——	——	460/211	——	——	——	——	——	——	——	——	——	——	——	——

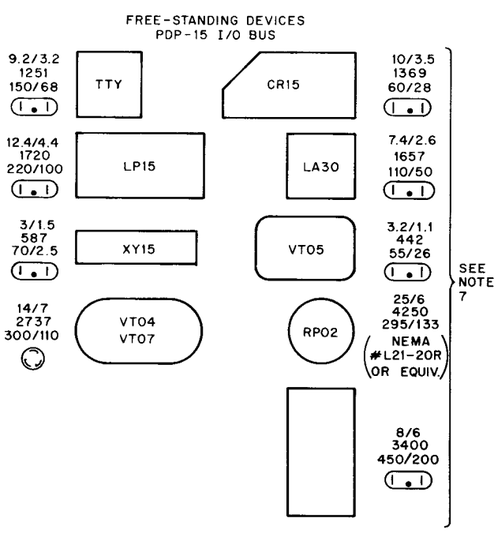
SEE NOTE 7

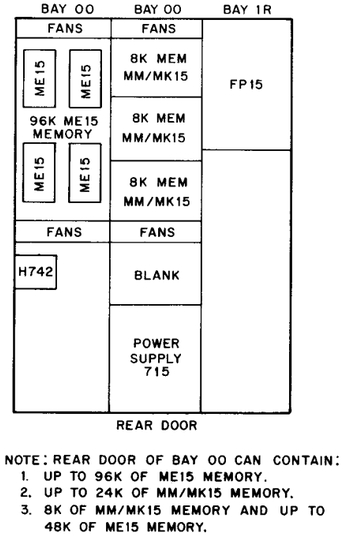
NOTES:

1. OPTIONS LT19, DPO9, XY15 OR DRO9 AND THE DB09 AND DB08 ARE LOCATED IN AVAILABLE SPACE IN THE H963-J CAB. ORDER OF PREFERENCE IS TC59, LT19 DPO9 ETC. PLACED IN THE ORDER LISTED ABOVE FROM BOTTOM TO TOP IN THE EVENT THAT ALL OPTIONS ARE INCLUDED A SECOND CAB. IS REQUIRED.
2. IF FOUR TU56'S ARE ORDERED, TU56 #3 AND #4 ARE INSTALLED IN SECOND CAB. AS SHOWN.
3. AFC-15 AND UDC-15 CABINETS COULD BE A MAX OF 11UDC AND 11AFC CABINETS EACH TO MEET THE REQUIREMENTS OF THE CUSTOMER.
4. IF EXTENSIVE MX15 SYSTEM INSTALLED DISK SYSTEM (RF15 AND/OR RP15) CABINETS MAY BE PHYSICALLY SEPARATED AND REPOSITIONED AT INSTALLATION TO MAINTAIN CABLE LENGTHS.
5. BAY 2L IS ALSO USED FOR EXPANSION OF ME15 MEMORY ACCORDING TO A-SP-ME15-0.
6. WHEN CONFIGURING SYSTEM, WHEN NOT SPECIFIED, AN AVERAGE CABINET WEIGHT OF 500LBS/178KG IS RECOMMENDED.
7. DATA FORMAT

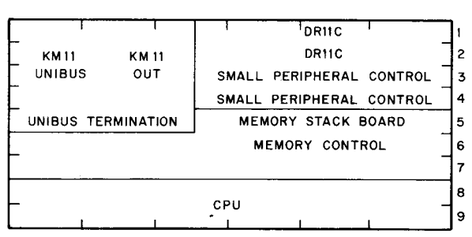
   CURRENT SURGE/NORMAL (MAX)(115V 60Hz).
   MAX BTU/HOUR
   WEIGHT LBS/KGS (MAX)
   TYPE OF PWR PLUG

   = 3 PRONG TWIST LOCK
   = 3 PRONG PARALLEL BLADE





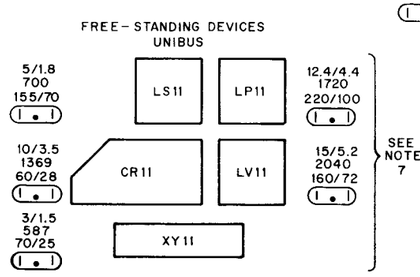
NOTE: REAR DOOR OF BAY 00 CAN CONTAIN:
1. UP TO 96K OF ME15 MEMORY.
2. UP TO 24K OF MM/MK15 MEMORY.
3. 8K OF MM/MK15 MEMORY AND UP TO 48K OF ME15 MEMORY.



PDP-11/05 CONFIGURATION GUIDE

NOTE:
TWO (2) SLOTS IN THE 11/05 ARE AVAILABLE FOR SMALL PERIPHERAL CONTROLLERS (LS11, LP11, XY11, ETC.). IF MORE THAN TWO (2) SLOTS ARE NEEDED, A BA11ES, H720, AND D11 ARE REQUIRED. ONE (1) DD11 CONTAINS LOGIC FOUR (4) SMALL PERIPHERAL CONTROLLERS.

STANDARD CABLE LENGHS (IN FEET)	LS11	CR15 CR11	LP15 LP11	LV11	XY15 XY11	LA30	VT05	TTY	VT04 VT07	RP02
	25	25	25	25	10	12	12	12	25	20



15-0787

Figure 11-4 System Configuration Diagram

Table 11-1
Free-Standing Cabinets, Peripherals, and Options

Processor Cabinet or Free-Standing Peripheral	15/10	15/20	15/30	15/35	15/40	Dimensions (in) Height	Width	Depth	Weight (lb)	Input Current Nominal	Surge	Heat Dissipation (Btu/hr)	Power Dissipation (W)	Cable Length (ft)	Comments
H963D (Bay 00)	1	1	1	1	1	71-1/2	21-3/4	30		25	150	9775	2875		Includes 32K memory, KE15, KF15, and KW15.
H963E (Bay 1R)		1	1	1	1	71-1/2	21-3/4	30		10	25				Refer to Table A-2 for cabinet-mounted options.
H963F (Bay 2R)		1	1	1	1	71-1/2	21-3/4	30		2.5	14				Refer to Table A-2 for cabinet-mounted options.
H963B (Bay 2L)				1	1	71-1/2	21-3/4	30		8.5	23	3325	978		Includes one RF15 and one RS09.
TU10 DECmagtape						71-1/2	21-3/4	30	450	8.5		3420	1000	15 std.	In H950 Cabinet 25 ft. max. cable length between transports.
RP15 Disk Pack Control H963C (Bay 1L)						71-1/2	21-3/4	30	305	7	25	2747	805		
RP02 Disk Drive						39	30	24	295	6	25	3100	900	20 std. 50 max.	
CR15 Card Reader						56	64	36	300	16	–	5800	1700	15 max.	
MX15 Memory Bus Multiplexer						71-1/2	21-3/4	30	500	12		4692 85	1380 25		For single cabinet with 32K memory and MX15A Multiplexer.
AD15 Analog Subsystem						71-1/2	21-3/4	30		1		342	100		In H963P Cabinet.
BD-15 Control						71-1/2	21-3/4	30	750	12	30				
AFC-15 Automatic Flying Capacitor Subsystem						71-1/2	21-3/4	30	750	12	30	1700	500		Data listed for one cabinet. Installation may include up to 11 cabinets.
UDC-15 Universal Digital Control Subsystem						71-1/2	21-3/4	30	750	12	30				Data listed for one cabinet. Installation may include up to 11 cabinets.
VT15 Graphic Processor H963L (Bay 6R)						71-1/2	21-3/4	30	300	17	30	2346	690		
VT04 Graphic Display Console						51-1/2	21-1/2	30	260	8	15	2737	805	25 max.	

Table 11-1 (Cont)
Free-Standing Cabinets, Peripherals, and Options

Processor Cabinet or Free-Standing Peripheral	Dimensions (in)			Weight (lb)	Input Current		Heat Dissipation (Btu/hr)	Power Dissipation (W)	Cable Length (ft)	Comments
	Height	Width	Depth		Nominal	Surge				
VT05 Alphanumeric Display Terminal	12	19	30	55			800	130		
LP15 Line Printer	55	56	30	600	25		10200	3000	25 max.	
XY15AA, AB CalComp Plotters	9-3/4	18	14-3/4	53	1.5	3	595 626	175 184	10 max.	XY15AA XY15AB
XY15BA/BB CalComp Plotters	9-3/4	39-1/2	14-3/4	53	1.5	3	626	184	10 max.	
VP15A Storage Tube Display	14-3/4	9-3/4	21-5/8	51	2.4		932	274	15 max.	
LA30 DECwriter	31	20-1/2	24	110	3	–	1020	300	12	

Table 11-2
Cabinet-Mounted Peripherals, Controls, and Options

Cabinet-Mounted Peripheral or Control Option	Location			Weight (lb)	Prerequisite	Input Current		Heat Dissipation (Btu/hr)	Power Dissipation (W)	Comments
	Cabinet	Height	No. of Panels			Nominal	Surge			
TC15 DECtape Control	H963F	15-3/4	3	38	---	0.8		327	96	TU56 3 and 4 located in H963H if 4 are implemented.
TU56 Dual DECtape Transport	H963H	10-1/2	2		TC15	3	9	1190	350	
TC59D Magnetic Tape Transport Control	H963J	21	4	50	DW15	2	5	1000	230	
BA15 Peripheral Option Expander	H963E	5-1/4	1	13						Control for LT15A, PC15, and VP15.
BB15 Internal Option Expander	H963E	10-1/2	2	25						Accommodates KA15, KM15, KT15, Options.
VP15B, BL X-Y Oscilloscope Display	H963E	10-1/2	2	28	BA15	1.6		592	174	B
								598	176	BL with Light Pen
VP15C, CL Point Plot Display	H963E	10-1/2	2	68	BA15	1.6		592	174	C
								598	176	CL with Light Pen
PC15 High-Speed Paper-Tape Reader/Punch	H963E	10-1/2	2	60	BA15			1040	306	
FP15 Floating-Point Processor	H963E	21	4	50		18	30	13650	4000	Main wired assembly mounts on rear door, indicator panel below BB15 indicator panel.
XY15 Incremental Plotter Control	H963J	5-1/4	1	16	DW15					
CP15 Card Reader Control		5-1/4	1	16		<1		170	50	
LT19 Multi-Station Teletype Control	H963J	10-1/2	2	25	DW15			47	14	For one LT19D
								51	15	For each LT19E
DP09A Data Communication Channel	H963J	10-1/2	2	25	DW15			253	75	
DB99, DB98 Interprocessor Buffers	H963J	10-1/2	2	25	DW15			333	97	
AA15 Digital-to-Analog Multiplexer Control	H963K	10-1/2	2							

APPENDIX A
BASIC INSTRUCTION SET

This chapter describes the instruction set of the basic PDP-15 by function, and describes the use and action of each operation. The format is as follows:

Mnemonic	Three to six alphabet characters which represent the operation code in the MACRO-15 Assembler.
Operation Name	A description of the instruction.
Octal Code	The machine language code in octal notation. The symbols listed below will be used where applicable.
A	The bits of the Address Field
E	The E Field (Address Mode)
I	Indirect Addressing
X	Index Addressing
Execute Time	The times shown herein are average times for a system without memory parity or memory protect.

INSTRUCTION GROUPS

Memory Reference Instructions

The specific operations of the PDP-15 instruction repertoire are categorized into the following groups:

Transfer Instructions

DAC	Deposit Accumulator
LAC	Load Accumulator
DZM	Deposit Zero in Memory

Arithmetic Instructions

ADD	Add, 1's Complement
TAD	Add 2's Complement
ISZ	Increment and Skip if Zero

Logical Instructions

XOR	Exclusive OR
AND	AND (Logical Product)
SAD	Skip if Accumulator different from Memory

Jump and Skip Instructions

CAL	Call Monitor
JMP	Unconditional Jump
JMS	Jump to Subroutines

Control Instructions

XCT	Execute

Operate Instructions

Rotate Instructions

RAR	Rotate Accumulator and Link 1 Right
RTR	Rotate Accumulator and Link 2 Right
RAL	Rotate Accumulator and Link 1 Left
RTL	Rotate Accumulator and Link 2 Left

Control Instructions

OPR or NOP	No Operation
HLT	Halt
CLL	Clear Link
CML	Complement Link
STL or CCL	Set Link
GLK	Get the Link
CLA	Clear Accumulator
CLC	Clear and Complement Accumulator
OAS	OR Console Data Switches to Accumulator
IAC	Increment the Accumulator
SWHA	Swap Halves of the Accumulator
LAW	Load Accumulator with this Instruction
CMA	1's complement the Accumulator
TCA	2's complement the Accumulator
SKP	Skip Unconditionally

Skip on Register Condition Instructions

SPA	Skip if Accumulator is Positive
SMA	Skip if Accumulator is Negative
SNA	Skip if Accumulator Not Zero
SZA	Skip if Accumulator is Zero
SNL,SML	Skip if Link Not Zero
SZL,SPL	Skip if Link Equals Zero

Microcoded Instructions

! is a MACRO 15 assembly code to define "and."

LAS=CLA!OAS	Load Accumulator from Console
LAS!CMA	Load Accumulator from 1's Complement of Console
CLC=CLA!CMA	Set Accumulator to all 1's
STL=CLL!CML	Set the Link to 1
TCA=CMA!IAC	2's Complement the Accumulator
STL!CLC	Set Link and Accumulator to all 1's
RCL=CLL!RAL	Clear Link and Rotate Left
RCR=CLL!RAR	Clear Link and Rotate Right
GLK=CLA!RAR	Place Link in Accumulator

Index and Limit Register Instructions

Register Transfer Instructions

PAX	Place Accumulator in Index Register
PAL	Place Accumulator in Limit Register
PXA	Place Index Register in Accumulator
PXL	Place Index Register in Limit Register
PLA	Place Limit Register in Accumulator
PLX	Place Limit Register in Index Register

Register Control Instructions

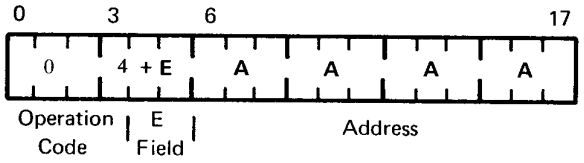
AXS n	Add n to Index Register and skip if result is equal to or greater than the Limit Register
AXR n	Add n to the Index Register
AAC n	Add n to Accumulator
CLX	Clear the Index Register
CLLR	Clear the Limit Register

Input/Output Instructions

IORS	Read Flags
CAF	Clear All Flags
IOF	Turn Interrupt Off
ION	Turn Interrupt On
EBA	Enable Bank Addressing
INH	Inhibit (API, PI)
ENB	Enable (API, PI)
SBA	Skip if in Bank Addressing
DBA	Disable Bank Addressing
TTS	Test TTY and Skip if Connected to Computer
SK15	Skip if Processor is a PDP-15

TRANSFER INSTRUCTIONS

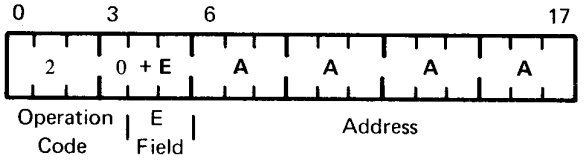
DAC **Deposit Accumulator**



Execute Time: 1.6 μs

The content of the accumulator is deposited in the memory location specified by the effective address. The accumulator remains unchanged.

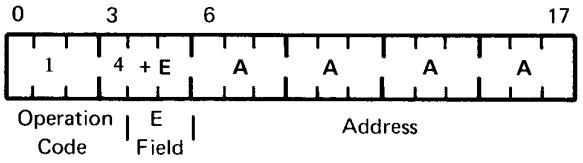
LAC **Load the Accumulator**



Execute Time: 1.6 μs

The content of the memory location specified by the effective address replaces the contents of the accumulator. The content of the memory location is unchanged.

DZM **Deposit Zero in Memory**

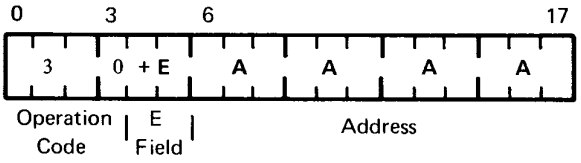


Execute Time: 1.6 μs

The content of the memory location specified by the effective address is zeroed. All other registers remain unchanged.

ARITHMETIC INSTRUCTIONS

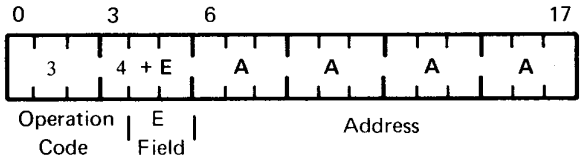
ADD **ADD, 1's Complement**



Execute Time: 1.6 μs

The content of the effective address and the content of the accumulator are added in 1's complement arithmetic. The results are put into the accumulator. An arithmetic overflow sets the link to the binary 1 state.

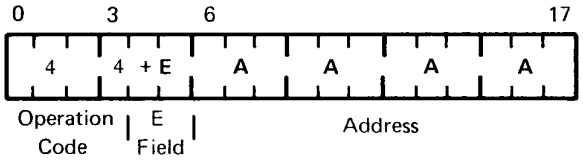
TAD **ADD, 2's Complement**



Execute Time: 1.6 μs

The content of the effective address and the content of the accumulator are added in 2's complement arithmetic. The results are put into the accumulator. An arithmetic carry from AC00 complements the link.

ISZ **Increment and Skip if Zero**

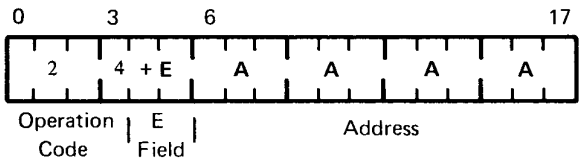


Execute Time: 2.4 μs

The content of the memory location specified by the effective address is incremented by 1. Two's complement arithmetic is used for the algebraic sum. If the result in memory is equal to 0, the next instruction is skipped. If the result is not 0, the next sequential instruction is executed. The content of the accumulator is unchanged.

LOGICAL INSTRUCTIONS

XOR **Exclusive OR (Half Add)**



Execute Time: 1.6 μs

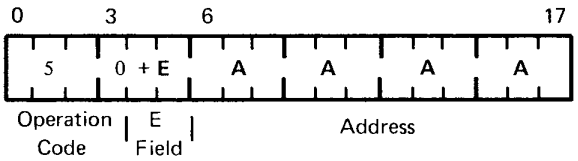
The Exclusive ORed bits of the accumulator and the content of the memory location specified by the effective address, C(EA), replace the content of the accumulator.

Programming Note – If corresponding memory and AC bits are in the same binary state, the AC bit is cleared to the 0 state. If the corresponding bits differ in state, the AC bit is set to the 1 state. The contents of the memory location are unchanged. The truth table for the XOR instruction is:

(AC)		C(EA)*		RESULT
0	+	0	=	0
0	+	1	=	1
1	+	0	=	1
1	+	1	=	0

*Content of the Effective Address

AND **AND (Logical Product)**



Execute Time: 1.6 μs

The contents of the effectively-addressed memory location are logically ANDed with the contents of the AC on a bit-by-bit basis. The result is left in the AC. If corresponding memory and AC bits are in the 1 state, the AC bit remains a 1; otherwise the AC bit is cleared to the 0 state. The contents of the memory location are unchanged, the previous contents of the AC are lost.

Programming Note – A logical product (AND) is used to select or mask specific portions of an operand. If only a selected portion of an operand is required, it is subjected to a mask. The mask is composed of patterns of 0s and 1s.

The AND instruction causes the bits of the operand to appear unchanged in the accumulator only in the area of the mask of 1 bits.

For example, to obtain the displacement address for an instruction (least-significant 12 bits), the following operation would be performed.

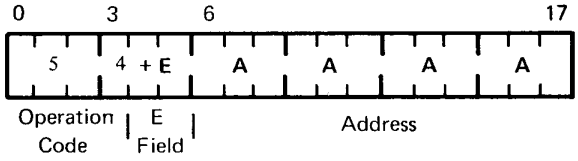
Operand	010110111011110101
Mask	000000111111111111
Accumulator Result	000000111011110101

The truth table for the AND instruction is as follows:

(AC)		C(EA)*		RESULT
0	x	0	=	0
0	x	1	=	0
1	x	0	=	0
1	x	1	=	1

*Content of the Effective Address

SAD **Skip if Accumulator Different From Memory**

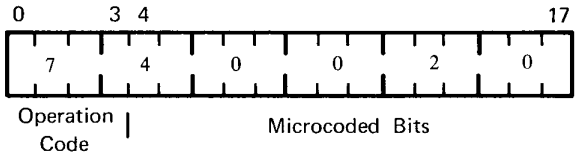


Execute Time: 1.6 μs

The content of the memory location specified by the effective address is compared bit-by-bit with the content of the accumulator. If they are not equal, the program counter is incremented by 1, and thus the next instruction is skipped. If the contents are the same, the next instruction is executed. Neither the contents of the accumulator nor the contents of the specified memory location are altered.

ROTATE INSTRUCTIONS

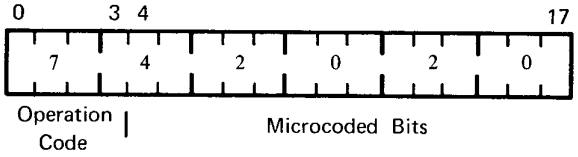
RAR **Rotate Accumulator and Link One Right**



Execute Time: 800 ns

The contents of the accumulator and Link are rotated one bit to the right. The content of the Link moves into accumulator bit 0, and accumulator bit 17 moves into the Link. All other bits move accordingly.

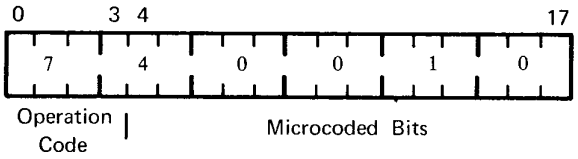
RTR **Rotate Accumulator and Link Two Right**



Execute Time: 800 ns

The contents of the accumulator and Link are rotated two bits to the right. The content of the Link moves into accumulator bit 1, accumulator bit 17 moves into bit 0 and bit 16 moves into the Link. All other bits move accordingly.

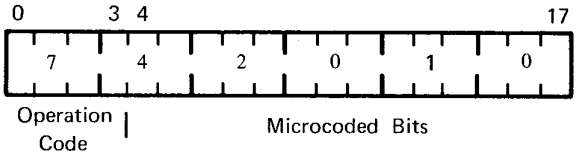
RAL **Rotate Accumulator and Link One Left**



Execute Time: 800 ns

The contents of the accumulator and Link are rotated one bit to the left. Accumulator bit 0 moves into the Link and the Link moves to accumulator bit 17. All other bits move accordingly.

RTL **Rotate Accumulator and Link Two Left**

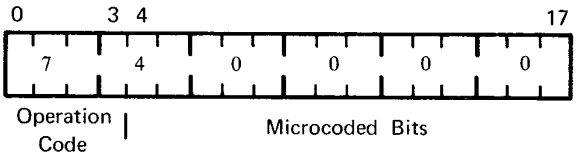


Execute Time: 800 ns

The contents of the accumulator and Link are rotated two bits to the left. The Link moves to accumulator bit 16, accumulator bit 0 moves to bit 17 and accumulator bit 1 moves to the Link. All other bits move accordingly.

CONTROL INSTRUCTIONS

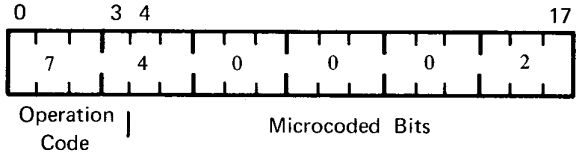
NOP **No Operation**



Execute Time: 800 ns

The NOP instruction causes a delay of one cycle. The program counter is incremented and processing continues. It is a "do nothing" cycle. NOP can be used to synchronize program timing by delaying program execution until the appropriate time.

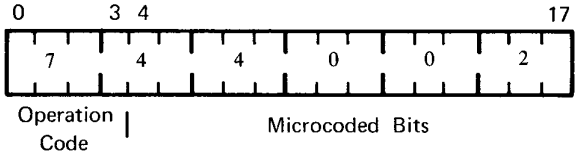
CML **Complement the Link**



Execute Time: 800 ns

The content of the Link is complemented.

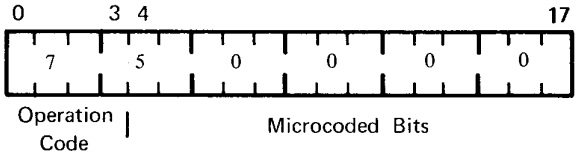
STL or (CCL) **Set the Link (Clear and Complement the Link)**



Execute Time: 800 ns

The content of the Link is set to a one.

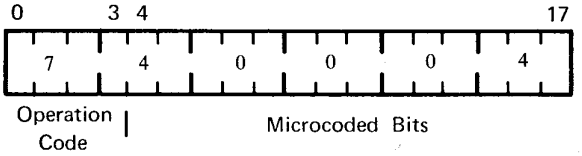
CLA **Clear the Accumulator**



Execute Time: 800 ns

The content of the AC is zeroed.

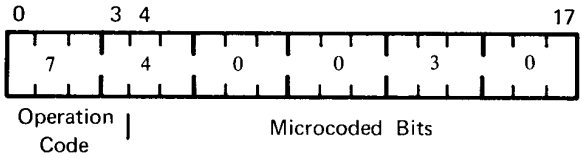
OAS **OR Console Data Switches to the Accumulator**



Execute Time: 800 ns

The contents of the console accumulator switches are inclusive ORed with the contents of the accumulator and the results are put into the accumulator.

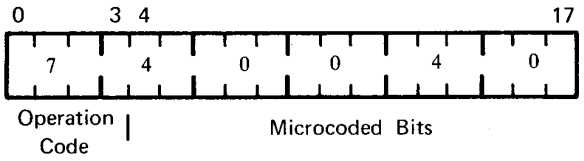
IAC **Increment the Accumulator**



Execute Time: 800 ns

The content of the accumulator is incremented by 1 in 2's complement. A carry-out complements the Link.

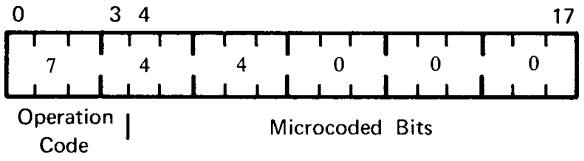
HLT **Halt**



Execute Time: 800 ns

The computer is stopped and must be restarted manually by depressing the console CONTINUE or START buttons. CONTINUE begins execution at the instruction immediately following the Halt instruction and START loads the contents of the address switches into the location counter and begins execution at that address.

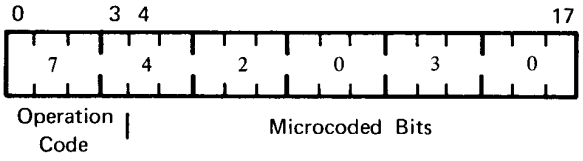
CLL **Clear Link**



Execute Time: 800 ns

The content of the Link is set to zero.

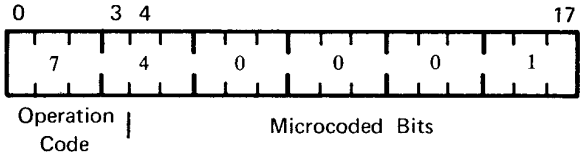
SWHA **Swap Halves of the Accumulator**



Execute Time: 800 ns

The high-order 9 bits of the accumulator (0-8) are exchanged with the low-order bits (9-17). The content of the Link is unchanged.

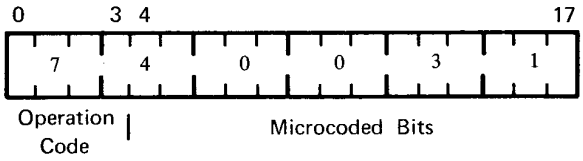
CMA **One's Complement the Accumulator**



Execute Time: 800 ns

The content of the accumulator is replaced by its 1's complement; i.e., each bit in the accumulator is inverted.

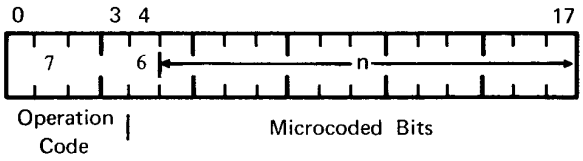
TCA **Two's Complement the Accumulator**



Execute Time: 800 ns

The content of the accumulator is replaced by its 2's complement.

LAW **Load Accumulator with This Instruction**



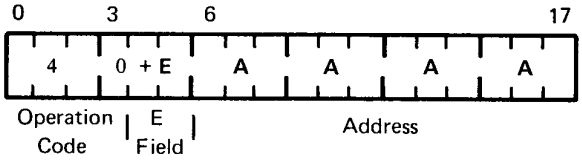
Execute Time: 800 ns

This instruction loads itself into the accumulator. The accumulator will contain the operation code 76 and a number "n."

$$0 \leqslant n \leqslant 17777$$

Programming Note – This instruction is used for loading alphanumeric character codes into the accumulator for transfer to peripherals and to preset the real-time clock counter. LAW should not be used for address formulation since program control by this instruction is only for the lowest 8K of memory.

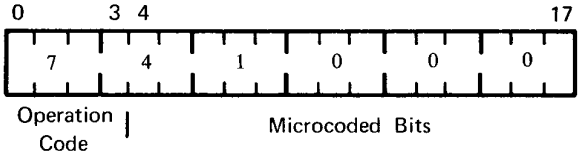
XCT **Execute**



Execute Time: 800 ns + Instruction Time
[e.g. XCT (LAC A) = 0.8 + 1.6 = 2.4 μs]

The computer executes the instruction located at the effective address. The contents of the PC are unchanged unless the effective memory address contains a JMS, CAL, JMP, or SKP instruction.

SKP **Skip Unconditionally**

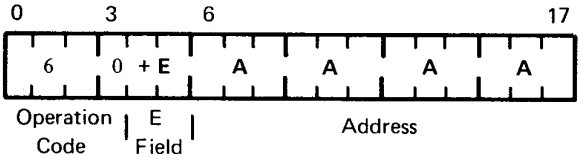


Execute Time: 800 ns

The next sequential instruction is unconditionally skipped.

JUMP INSTRUCTIONS

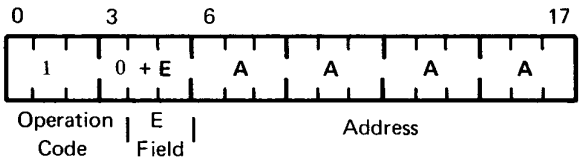
JMP **Unconditional Jump**



Execute Time: 800 ns

The next instruction is read from the C(EA). The previous contents of the PC are lost as the effective address enters the PC. The contents of the AC are unchanged.

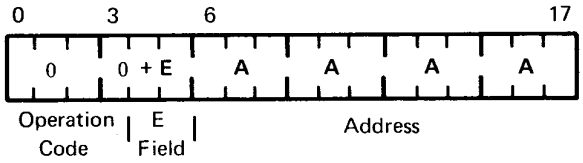
JMS **Jump to Subroutine**



Execute Time: 1.6 μs

The content of the program counter, the Link, and the status of the memory protect mode replace the content of the memory location specified by the effective address. The content of the program counter is stored in bits 3 through 17, user mode in bit 2, Bank or Page bit 1, and Link in bit 0. The effective address plus one replaces the content of the program counter (see INH instruction).

CAL **Call Monitor**

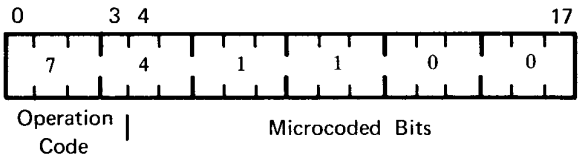


Execute Time: 1.6 μs

CAL specifies a JMS 20 to page 0 regardless of the content of the address field. This instruction is used to call the monitor while the system is in user mode. The address field stores information for the monitor from the user and puts the computer in monitor mode. CAL automatically sets the API to level 4.

If an index bit is set in a CAL instruction, it is ignored; i.e., no indexing is done. An indirect bit in the CAL instruction, however, will cause a JMS I 20. Bits 0, 1, and 2 are stored similar to the JMS instruction (see INH instruction).

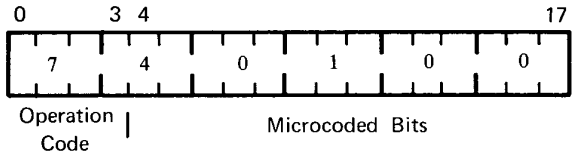
SPA **Skip if Accumulator is Positive**



Execute Time: 800 ns

The sign (bit 0) of the content of the accumulator is tested and, if positive (0), the next instruction is skipped. If the sign is negative (1), the next instruction is executed. The content of the accumulator remains unchanged.

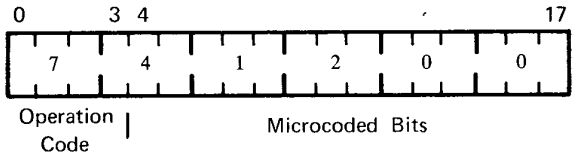
SMA **Skip if Accumulator is Negative**



Execute Time: 800 ns

The sign (bit 0) of the content of the accumulator is tested and, if negative (1), the next instruction is skipped. If the sign is positive, the next sequential instruction is executed. The content of the accumulator is not altered.

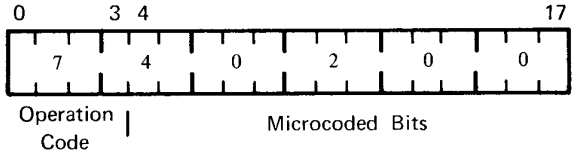
SNA **Skip if Accumulator is Not Zero**



Execute Time: 800 ns

The content of the accumulator is tested. If any bits are 1s, the next instruction is skipped. If all bits are 0s, the next instruction is executed. The content of the accumulator remains unchanged.

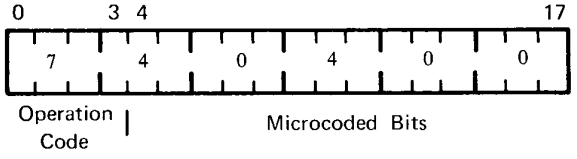
SZA **Skip if Accumulator is Zero**



Execute Time: 800 ns

The content of the accumulator is tested and if all bits are 0, the next instruction is skipped. If any bits are a 1, the next instruction is executed. The contents of the accumulator remain unchanged.

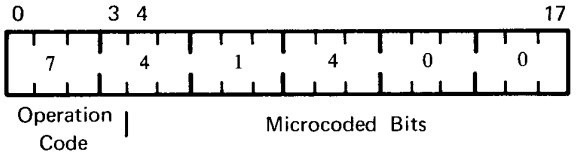
SNL or SML **Skip if Link Not Zero Skip on Minus Link**



Execute Time: 800 ns

The content of the Link is tested. If it is not 0, then the next instruction is skipped. If it is a 0, the next instruction is executed. The content of the Link remains unchanged.

SZL or SPL **Skip if Link Equal Zero Skip on Positive Link**



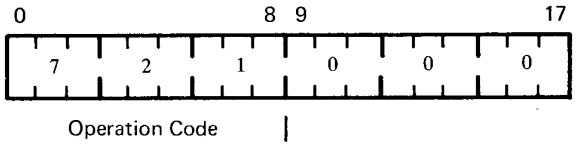
Execute Time: 800 ns

The content of the Link is tested. If it is a 0, then the next instruction is skipped. If it is a 1, the next instruction is executed. The content of the Link remains unchanged.

INDEX AND LIMIT REGISTER INSTRUCTIONS

Register Transfer Instructions

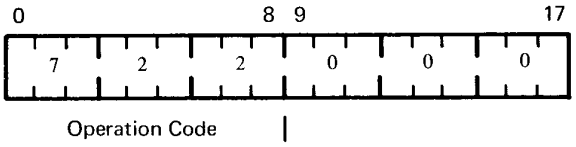
PAX **Place Accumulator in Index Register**



Execute Time: 1.6 μs

The content of the AC replaces the content of the index register. The content of the AC remains unchanged.

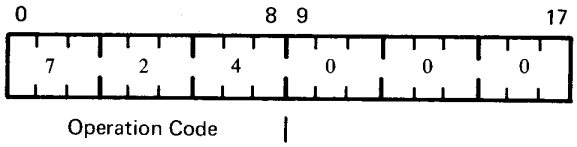
PAL **Place Accumulator in Limit Register**



Execute Time: 1.6 μs

The content of the limit register is replaced by the content of the accumulator. The content of the accumulator remains unchanged.

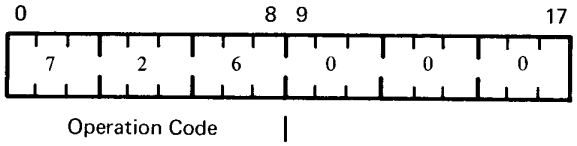
PXA **Place Index Register in Accumulator**



Execute Time: 1.6 μs

The content of the index register replaces the content of the accumulator. The content of the index register remains unchanged.

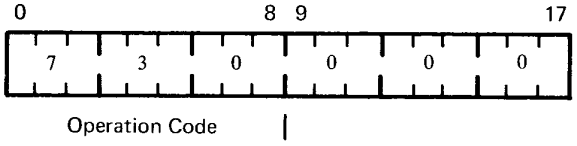
PXL **Place Index Register in Limit Register**



Execute Time: 1.6 μs

The content of the index register replaces the content of the limit register. The content of the index register remains unchanged.

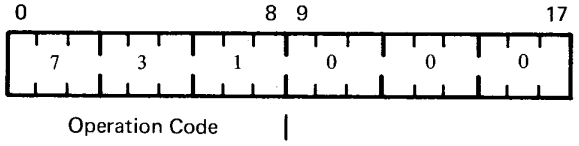
PLA **Place Limit Register in Accumulator**



Execute Time: 1.6 μs

The content of the limit register replaces the content of the accumulator. The content of the limit register remains unchanged.

PLX **Place Limit Register in Index Register**

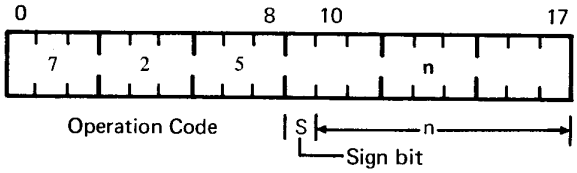


Execute Time: 1.6 μs

The content of the limit register replaces the content of the index register. The content of the limit register remains unchanged.

REGISTER CONTROL INSTRUCTIONS

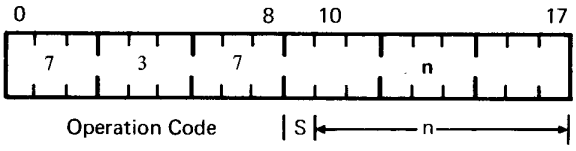
AXS n **Add n to Index Register and Skip if the Result Equal to or Greater than the Limit Register**



Execute Time: 1.6 μs

n, a signed 9-bit (8 bits plus sign) 2's complement integer is added to the content of the index register, and the results are placed in the index register. If the sum is greater than or equal to the content of the limit register, then the program counter is incremented by 1 and thus the next instruction skipped.

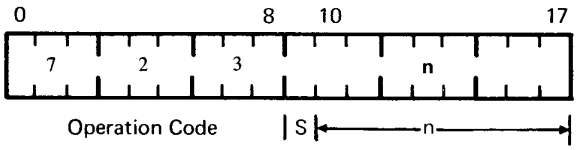
AXR n **Add to Index Register**



Execute Time: 1.6 μs

n, a signed 9-bit (8 bits plus sign) 2's complement integer is added to the content of the index register, and the result is placed in the index register.

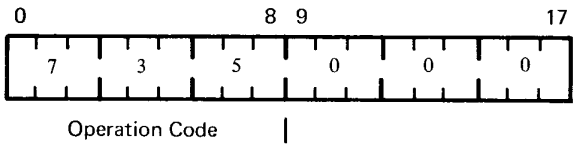
AAC n **Add n to Accumulator**



Execute Time: 1.6 μs

n, a signed 9-bit (8 bits plus sign) 2's complement binary number, is added to the content of the accumulator, and the result is placed into the accumulator.

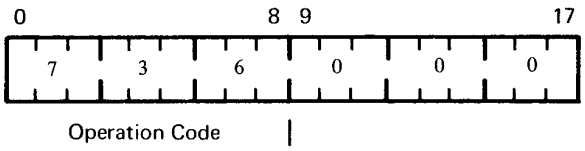
CLX **Clear the Index Register**



Execute Time: 1.6 μs

The content of the index register is replaced with all 0s. Former content is lost.

CLLR **Clear the Limit Register**



Execute Time: 1.6 μs

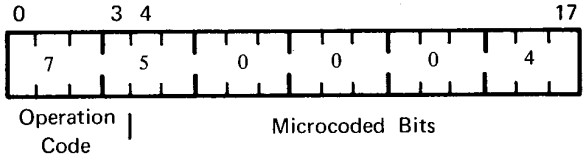
The content of the limit register is replaced with all 0s. The former content is lost.

MICROCODING

Some of the preceding instructions, which do not reference memory, can be combined by inclusive ORing their codes. Figure A-1 gives the set of all possible pairs of these instructions, and indicates which can be meaningfully combined to form higher order sets. From this figure, sets of triplets can also be found, to form even more complex instructions. A summary of the more useful combinations follows.

Microcoded Instructions

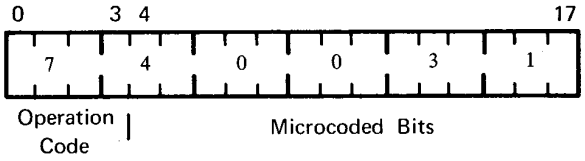
LAS=CLA!OAS **Load Accumulator from Console**



Execute Time: 800 ns

The content of the console data switches replaces the content of the accumulator.

TCA=CMA!IAC **2's Complement the Accumulator**



Execute Time: 800 ns

The content of the accumulator is replaced by its 2's complement.

Order of Events	Column 1	Column 2	Column 3		Column 4
Level 1	SNL SZA SMA	CLA CLL	OAS CMA	CML IAC	HLT
Level 2	SZL SNA SPA		RAR or RAL		
Level 3	SKP		RTF or RTL or SWHA		

1. Combine instructions from left to right.
2. Any instructions in a box can be combined, except the rotate instructions.
3. Instructions on different levels cannot be combined if they are in the same column. Instructions on any level can be combined if they are in different columns. (e.g., SZA!SMA!CLA!OAS!HLT! is legal – SZA!SPA is not legal.)
4. CML and IAC cannot be combined. Either one can be combined with OAS and/or CMA (e.g., OAS!CMA!CML or OAS!CMA!IAC).
5. Instructions occur in order from column 1 to column 4.

NOTE

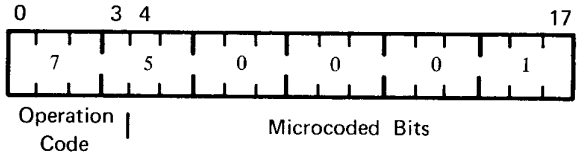
Level 1 skips (SNL, SZA, SMA) will occur if any one of the combined tests is satisfied (an OR condition).

Level 2 skips (SZL, SNA, SPA) will occur only if all the combined tests occur (an AND condition).

Combined rotates become a SWHA or an IAC, depending on bit 7.

Figure A-1 Microcoding PDP-15 Control Instructions

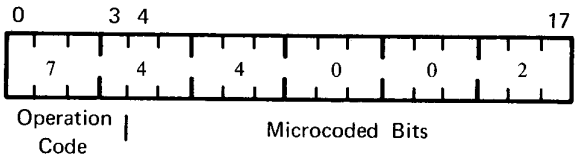
CLC=CLA!CMA **Set the Accumulator to all Ones**



Execute Time: 800 ns

The content of the accumulator is set to all 1's. The Link is unchanged.

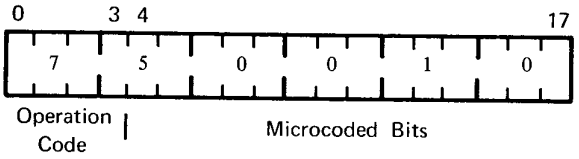
STL=CLL!CML **Set the Link to One**



Execute Time: 800 ns

The content of the Link is set to a 1.

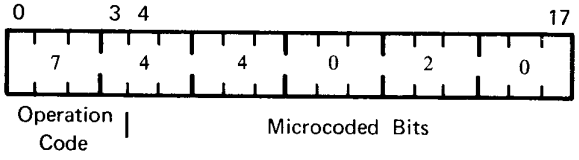
GLK=CLA!RAL **Get the Link**



Execute Time: 800 ns

The accumulator is cleared and the Link shifts into accumulator bit 17.

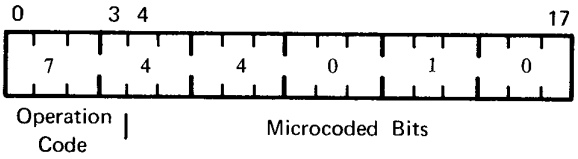
RCR=CLL!RAR **Clear Link and Rotate Right**



Execute Time: 800 ns

The Link is cleared to 0, and then the Link and accumulator are rotated one bit to the right.

RCL=CLL!RAL **Clear Link, Rotate AC and Link Left**



The Link is cleared to 0, and then the Link and accumulator are rotated one bit to the left.

INPUT/OUTPUT INSTRUCTION GROUP

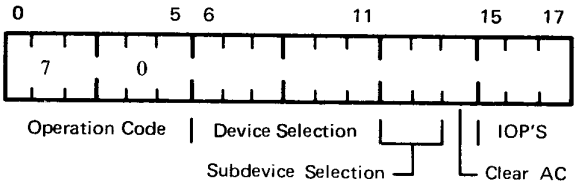
Each internal or peripheral device is assigned a subgroup of the group of instructions called Input/Output Instructions. The format for this group is shown below:

(1) The operate code is 70.

(2) The device and subdevice bits select the device's code or address.

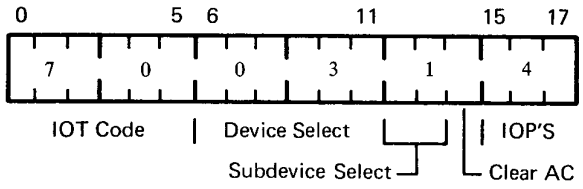
(3) Bit 14 asserted clears the accumulator before a transfer.

(4) Bits 15, 16 and 17 select three control pulses, IOP4, IOP2, and IOP1, respectively.



Details concerning the use of input/output instructions are given in the PDP-15 User Handbook or the PDP-15 Interface Manual. The instructions used in the basic PDP-15 are:

IORS **Read Flags**



Execute Time: 4 μs

The read flags instruction causes the transfer of an 18-bit system status word from the I/O bus to the accumulator. During this instruction, each of the internal and external system devices gates its status register onto preassigned data lines. The I/O Processor transfers these bits to the accumulator. Figure A-2 shows the word/status/bit assignment.

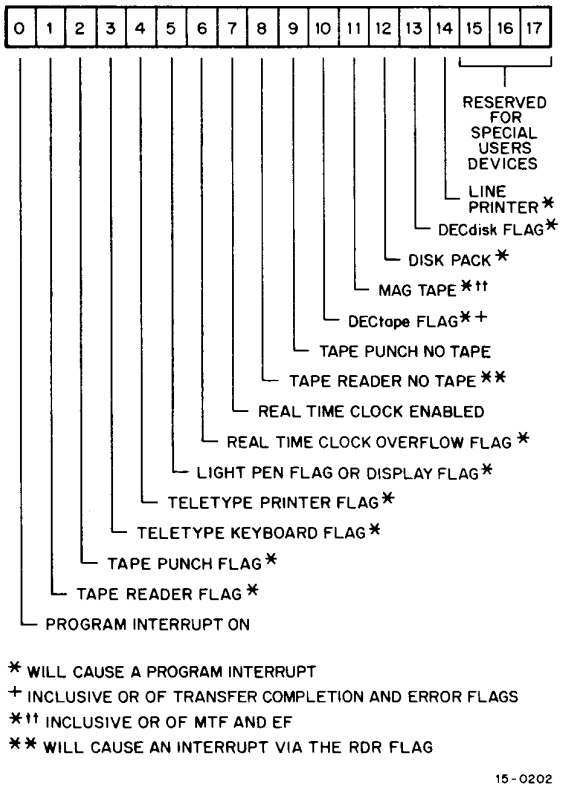
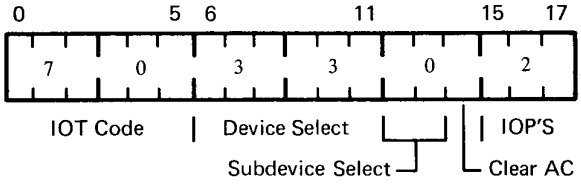


Figure A-2 IORS Word Status Bit Assignments

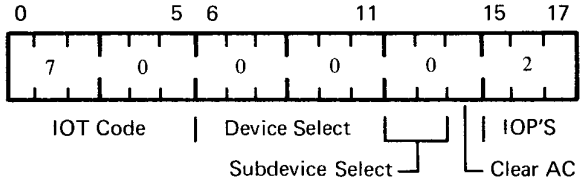
CAF **Clear all Flags**



Execute Time: 3 μs min., 4 μs max.

This instruction gates a pulse to the I/O bus to initialize (clear) all flags of any device that can call for interrupt service. Customer-installed equipment should make use of this pulse to reset flags and registers that must be cleared for system initiation.

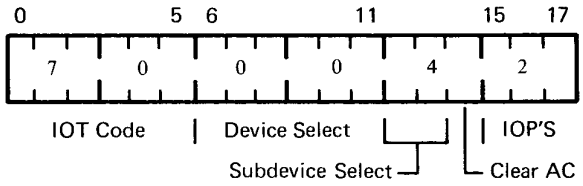
IOF **Turn Interrupt OFF**



Execute Time: 3 μs min., 4 μs max.

This input/output instruction turns off the program interrupt facility of the exchange.

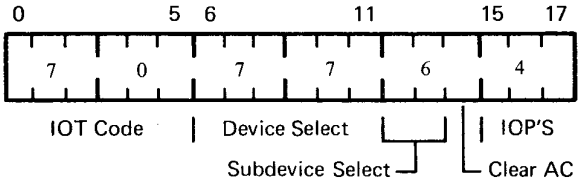
ION **Turn Interrupt ON**



Execute Time: 3 μs min., 4 μs max.

The program interrupt facility is enabled.

EBA **Enable Bank Addressing**



Execute Time: 2 μs min., 3 μs max.

The index register is disabled and the sixth bit (bit 5), normally used to indicate an indexed operation, is gated to the memory address field, permitting direct addressing of 8192 words of memory.

DBI **Disable Breaks**

Execute Time: 3 μs min.
4 μs max.

Inhibits API and PI. Incorporated to make reentrant programming more convenient. When in monitor mode one free instruction will be granted after CAL, JMS, PI; two free instructions after NORM. "Free instructions" means executable instructions that are performed before the computer goes into the interrupt mode. See example program.

EBI **Enable Breaks**

Execute Time: 3 μs min.
4 μs max.

Enable API or PI. See example program.

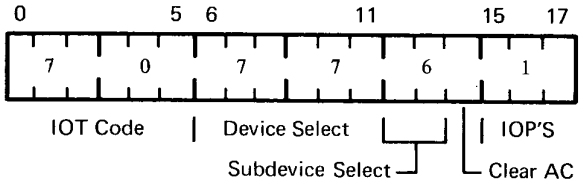
Example:

Each of the sequences listed is expected to be uninterruptable (except for data breaks).

```
        JMS A or JMS* (A)     /INTERRUPT
                              /FLAG OCCURS
A       0
        LAC A                 /INTERRUPT IS
                              /SERVICED


        JMS B or JMS* (B)
B       0
        DBI
        LAC X                 /INTERRUPT
                              /FLAG OCCURS
        DAC XX
        LAC Y
        EBI
        DAC YY                /INTERRUPT
                              /WILL BE
                              /SERVICED
```

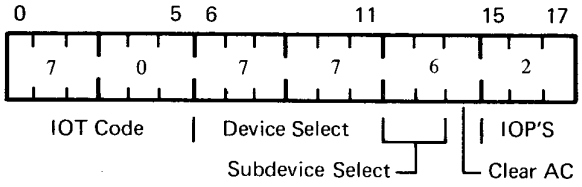
SBA **Skip if in Bank Addressing**



Execute Time: 2 μs min., 3 μs max.

If in bank addressing mode the next instruction is skipped.

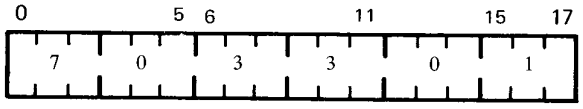
DBA **Disable Bank Addressing**



Execute Time: 4 μs

Bank address mode is disabled, and the PDP-15 operates with indexing and addresses 4096 words of memory directly.

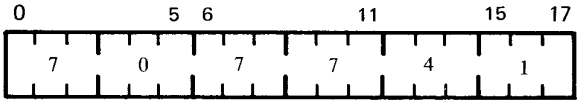
TTS **Test TTY and Skip**



Execute Time: 2 μs min.
3 μs max.

Test if KSR or ASR Teletype is connected to PDP-15. Skip next instruction if it is.

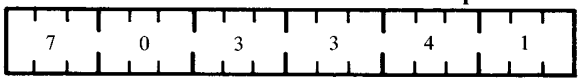
SKP15 **Skip if PDP-15**



Execute Time: 2 μs min.
3 μs max.

Skip the next instruction if the processor is a PDP-15.

SPCO **Skip if PC15**



Execute Time: 2 μs min.
3 μs max.

Skip next instruction if a PC15 is connected to the system.

APPENDIX B
INTERNAL OPTIONS INSTRUCTION SET

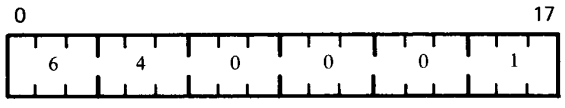
The options available with PDP-15 systems and the instruction sets for each option are described in this chapter.

KE15 EXTENDED ARITHMETIC ELEMENT (EAE)

The KE15 Extended Arithmetic Element, a Central Processor option, facilitates high-speed multiplication, division, shifting, normalizing and register manipulation. Installation of this element as shown in Figure B-1 adds an 18-bit multiply quotient register, step counter and additional shift logic to the arithmetic unit. The instruction set associated with this unit is as follows.

Control Instructions

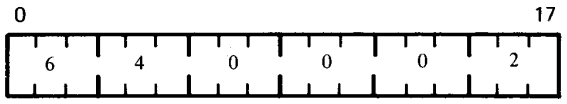
OSC **Inclusive OR the Step Counter with the Accumulator**



Execute Time: 1.325 μs

The content of the step counter is inclusively ORed with accumulator bits 12 to 17, and the result replaces bits 12 to 17 of the accumulator. The contents of accumulator bits 0 to 11 and the step counter are unchanged.

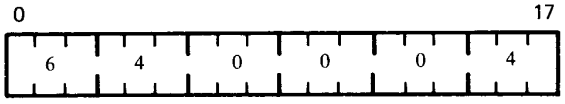
OMQ **Inclusive OR Multiplier Quotient Register with the Accumulator**



Execute Time: 1.325 μs

The content of the multiply quotient register is inclusively ORed with the content of the accumulator, and the results replace the former contents of the accumulator. The content of the multiplier quotient register is unchanged.

CMQ **Complement the Multiplier Quotient Register**



Execute Time: 1.325 μs

The 1's complement (logical inversion) of the multiplier quotient register replaces the former content of the multiplier quotient register.

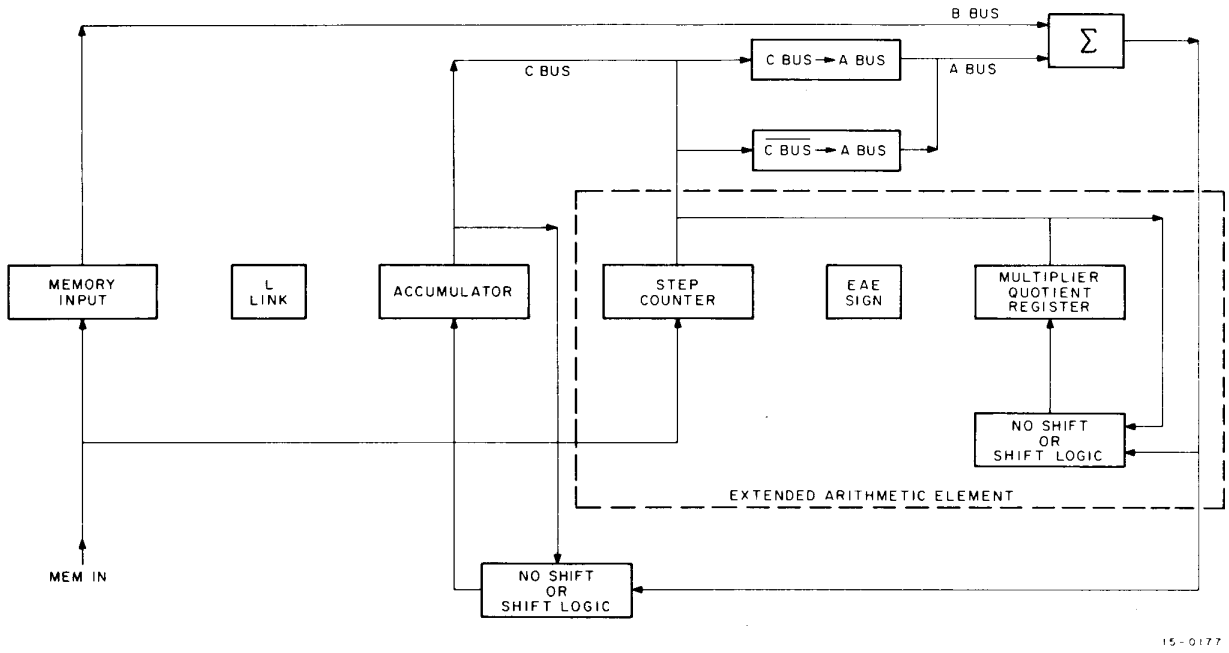
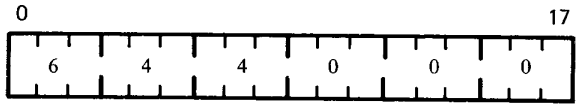


Figure B-1 Simplified Arithmetic Unit With The Extended Arithmetic Element

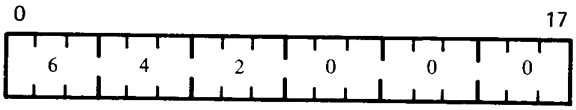
ABS **Replace the Contents of the Accumulator with the Absolute Value**



Execute Time: 1.325 μs

If the content of the accumulator is negative – that is, if bit 0 is a 1, then the content is 1's complemented (logical inversion).

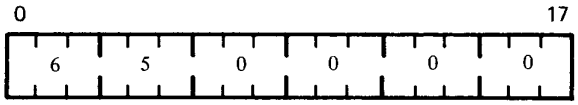
OAC **Inclusive OR Accumulator and Multiplier Quotient Register**



Execute Time: 1.325 μs

The contents of the accumulator and the multiplier quotient register are inclusive ORed, and the result is placed into the multiplier quotient register. The content of the accumulator is unchanged.

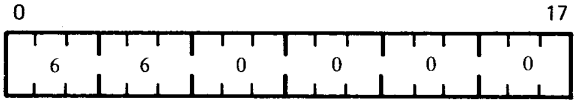
CLQ **Clear the Multiplier Quotient Register**



Execute Time: 1.325 μs

The content of the multiplier quotient register is replaced with 0.

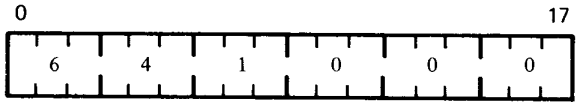
SHAL **Shift Accumulator Bit Zero into the Link**



Execute Time: 1.325 μs

The content of accumulator bit 0 is shifted into the Link. The content of accumulator bits 1 through 17 are unchanged.

ECLA **Clear the Accumulator**



Execute Time: 1.325 μs

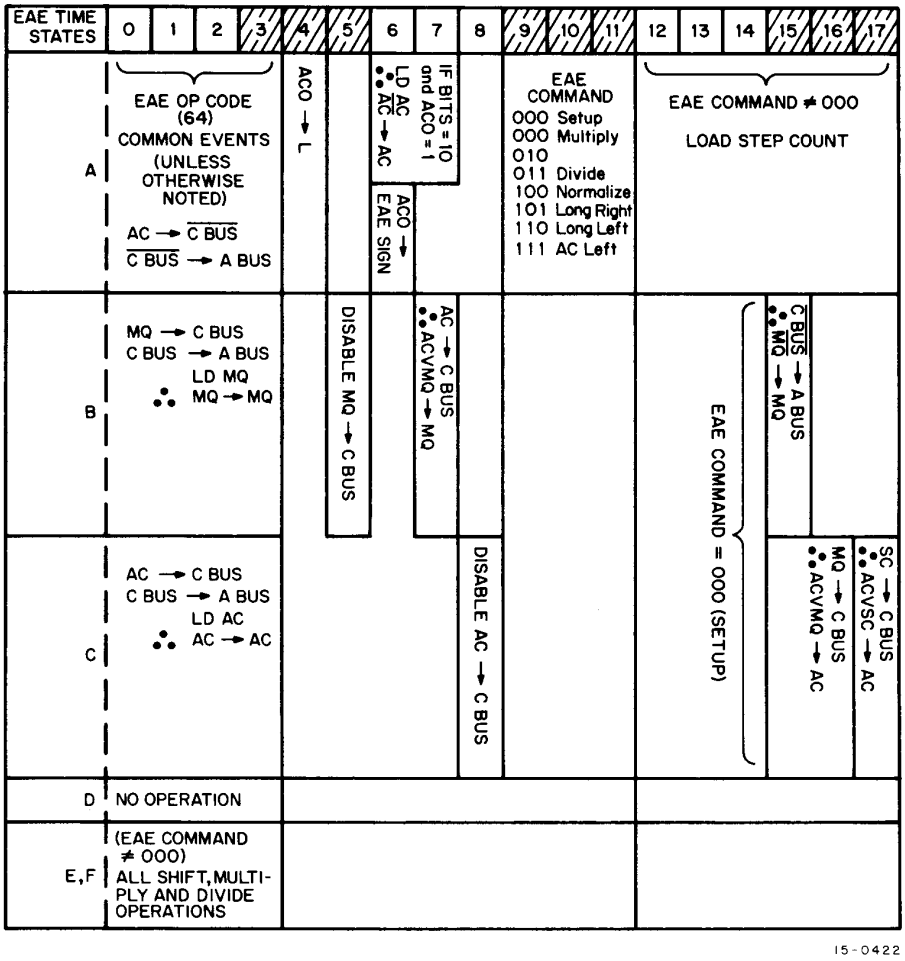


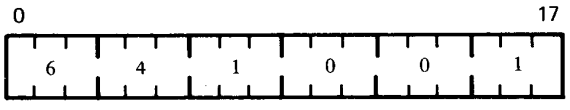
Figure B-2 EAE Control Instruction Microcoding

The content of the accumulator is set to 0. The Link remains unchanged.

Microcoding

The control instructions of the extended arithmetic element can be microcoded by inclusive-ORing their instruction codes according to the chart in Figure 10-2. The most useful combinations are explained in detail.

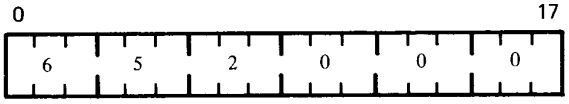
LACS= OSC!ECLA — **Load Accumulator From Step Counter**



Execute Time: 1.325 μs

The accumulator is cleared and the content of the 6-bit step counter is loaded into accumulator bits 12 to 17.

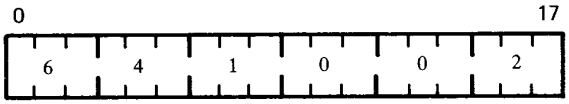
LMQ= OAC!CLQ — **Load Multiplier Quotient Register from Accumulator**



Execute Time: 1.325 μs

The content of the multiplier quotient register is replaced with the content of the accumulator. The accumulator remains unchanged.

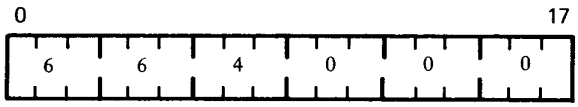
LACQ= OMQ!ECLA — **Load Accumulator From Multiplier Quotient Register**



Execute Time: 1.325 μs

The content of the accumulator is replaced with the content of the multiplier quotient register. The multiplier quotient register remains unchanged.

GSM= ABS!SHAL **Get Sign and Magnitude of Accumulator**



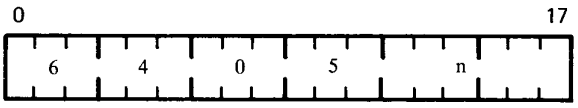
Execute Time: 1.75 μs

The content of accumulator bit 0 is entered into the Link, leaving the accumulator unchanged. Then, if the sign bit (bit 0) is a 1 (negative), the accumulator is 1's complemented.

An instruction which could prove useful in diagnostics is a combination of CLQ OAC ECLA OMQ. In this case, the content of the accumulator is loaded into the multiplier quotient register, and then back to the accumulator. The content of the accumulator now reflects what was loaded into the multiplier quotient register. The code is 653002.

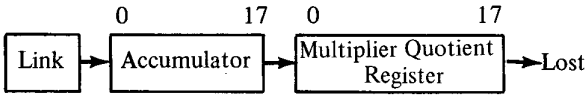
BASIC SHIFT INSTRUCTIONS

LRS n **Long Right Shift**

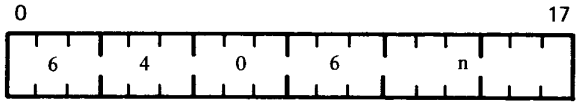


Execute Time: $2.915 + 0.133(n-1)$
$1 \leqslant n \leqslant 36$
2.915 if n = 0

The accumulator and multiplier quotient registers together function as a 36-bit shift register. Their contents are shifted n-bits to the right, where n is specified by the six low-order bits of the instruction. The step counter is automatically initialized to the 2's complement of n, and shifting stops when the step counter reaches 0. For each step, the content of accumulator bit 17 enters bit 0 of the multiplier quotient register, and bit 17 of the multiplier quotient register is lost. The content of the Link, usually initialized to 0, remains unchanged, but enters bit 0 of the accumulator at each step. The action is represented below:

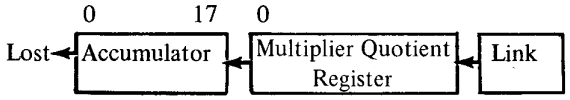


LLS n **Long Left Shift**

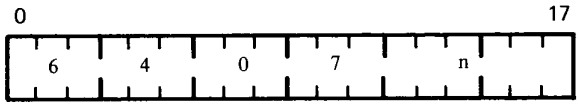


Execute Time: $2.915 + 0.133(n-1)$
$1 \leqslant n \leqslant 36$
2.915 if n = 0

The accumulator and multiplier quotient registers together function as a 36-bit shift register. Their contents are shifted n bits to the left where n is specified by the six low-order bits of the instruction. The step counter is automatically initialized to the 2's complement of n, and shifting stops when the step counter reaches 0. For each step, bit 0 of the multiplier quotient register enters bit 17 of the accumulator, the content of accumulator bit 0 is lost, and the content of the Link, usually initialized to 0, remains unchanged but enters bit 17 of the multiplier quotient register. Shifting is illustrated below.

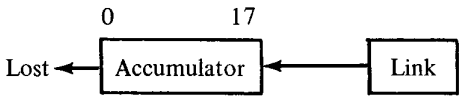


ALS n **Accumulator Left Shift**



Execute Time: $2.915 + 0.133(n-1)$
$1 \leqslant n \leqslant 18$
2.915 if n = 0

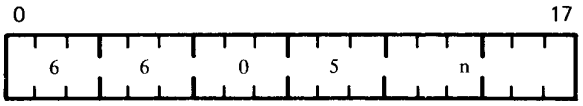
The content of the accumulator is shifted n positions to the left; where n is specified by the six low-order bits of the instruction word. Shifting stops when the contents of the step counter, automatically initialized to the 2's complement of n, reaches 0. For each shift, the content of the Link, usually initialized to 0, enters bit 17 of the accumulator. The bits shifted out of accumulator bit 0 are lost. This action is illustrated below:



Microcoding

The shift instructions can be microcoded with certain functions to form more complex instructions. Figure 10-2 lists all possible combinations. Some of the most useful combinations are described below.

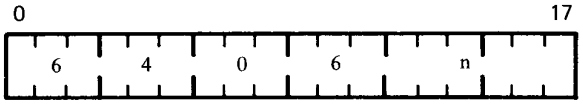
LRSSn= **Long Right Shift Signed**
LRSn!SHAL



Execute Time: $2.915 + 0.133(n-1)$
$1 \leqslant n \leqslant 36$

The content of accumulator bit 0 enters the Link. Then a long right shift instruction is carried out. Note that the content of the Link enters accumulator bit 0.

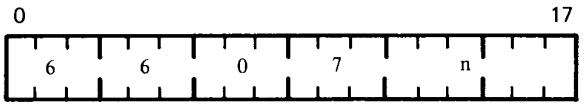
LLSSn= **Long Left Shift Signed**
LLSn!SHAL



Execute Time: $2.915 + 0.133(n-1)$
$1 \leqslant n \leqslant 36$

The content of the accumulator bit 0 enters the Link. Then a long left shift instruction is carried out. Note that the content of the Link enters multiplier quotient bit 17.

ALSSn= **Accumulator Left Shift Signed**
ALSn!SHAL

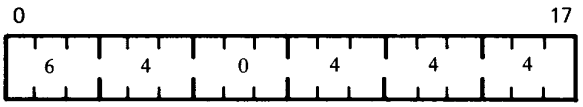


Execute Time: $2.915 + 0.133(n-1)$
$1 \leqslant n \leqslant 18$

The content of accumulator bit 0 enters the Link. Then a long accumulator left shift takes place. Note that the content of the Link enters accumulator bit 17.

NORMALIZE INSTRUCTIONS

NORM **Normalize Unsigned**

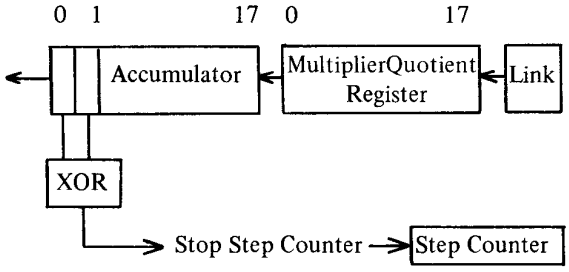


Execute Time: $2.915 + 0.133(n-1)$

The contents of the AC and MQ are shifted left, where n is the number of shifts required to complete the normalize. The accumulator and multiplier quotient registers together function as a 36-bit shift register. Their contents are shifted left until:

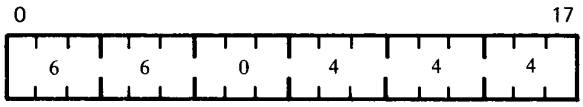
a. The content of accumulator bit 0 is different from the content of accumulator bit 1, or
b. The content of the step counter reaches 0.

The content of the 6-bit step counter, which specifies the step count, is automatically initialized to 44_8 or 36_{10} by the content of the six low-order bits of the instruction. For each shift step, the content of bit 0 of the multiplier quotient register enters bit 17 of the accumulator, and the content shifted out of accumulator bit 0 is lost. The content of the Link, usually initialized to 0, enters bit 17 of the multiplier quotient register. If shifting halts because accumulator bit 0 does not equal bit 1, the step counter reflects the number of steps executed to reach that state. Its content (2's complement of step count plus the steps executed) is accessible with either the OSC or LACS instruction. The NORM operation is indicated below (see INH instruction in Appendix A):



This normalize instruction can be combined with the control instruction SHL to form a normalize signed instruction.

NORMS **Normalize Signed**



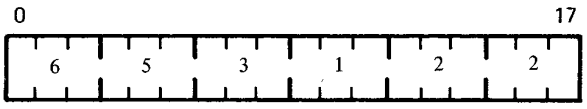
Execute Time: $2.915 + 0.133(n-1)$

The content of accumulator bit 0 enters the Link. The accumulator remains unchanged. Then a normalize is carried out. Note that the content of the Link enters bit 17 of the multiplier quotient register.

ARITHMETIC INSTRUCTIONS

Multiply

MUL **Multiply**



Execute Time: 7.420 μs

The content of the memory location which follows immediately after the MUL instruction is multiplied by the content of the accumulator. A double length (36-bit) product is formed in the accumulator and the multiplier quotient register. The most significant 18 bits are contained in the accumulator, and the least significant bits in the multiplier quotient register.

Prior to this instruction, the Link must be zeroed, the multiplier entered into the accumulator, and the multiplicand into the address following the MUL instruction.

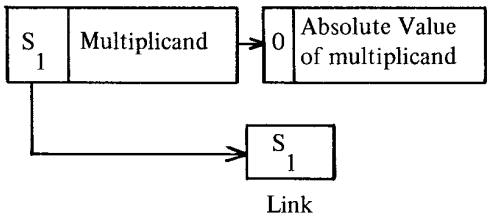
At the beginning of a MUL execution, the multiplier is transferred to the multiplier quotient register, the accumulator is cleared, and the step counter is initialized to the 2's complement of 22_8 (18_{10} steps) from the six low-order bits of the MUL instruction word. The execution, a multiplication of one unsigned quantity by another (18 bits, binary point of no consequence) halts when the step counter overflows. The content of the Link remains 0, the content of the register containing the multiplicand, the one following the MUL instruction, remains unchanged, and the program continues from the following instruction (two locations after MUL). The process is illustrated below:

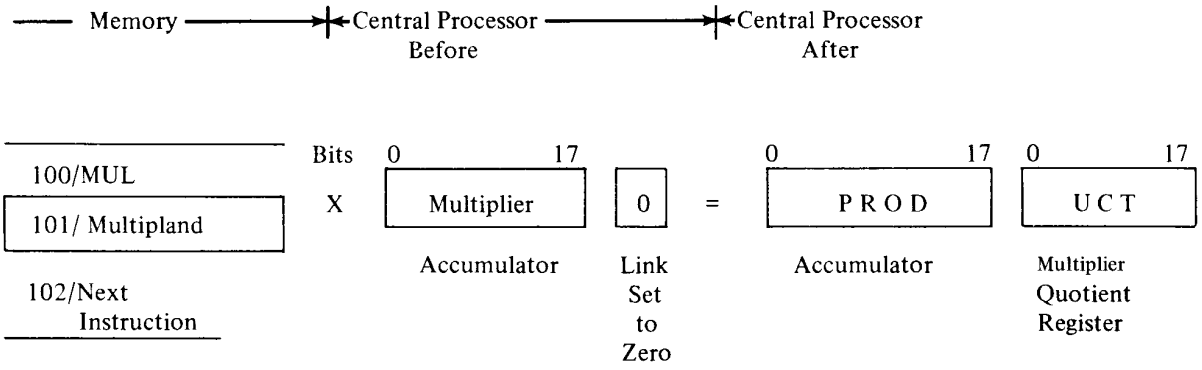
Signed multiplication can be carried out by using the MUL instruction with the ABS instruction and converting the multiplicand into its absolute value. This is done as follows:

Register	Content	
Y-6	LAC Multiplicand	Put the Multiplicand into the accumulator
Y-5	GSM	Stored sign in Link and Convert
Y-4	DAC Y	Store absolute value of multiplicand in Y
Y-3	LAC Multiplier	Load multiplier into accumulator
Y-2	ABS	Store sign in EAE sign bit and convert Multiplier to absolute value
Y-1	MUL	Multiply two absolutes; 1's complement results if Link differ.
Y	Multiplicand	

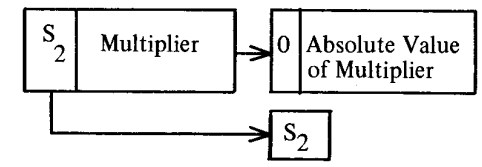
The results are illustrated below:

a. Memory Location Y Location Y

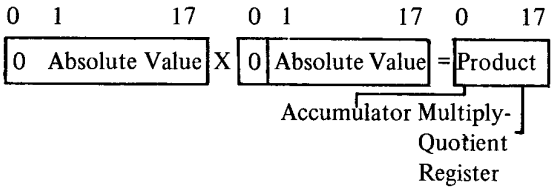




b. Memory Location Accumulator



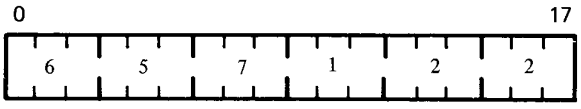
c.



If the EAE sign bit ≠ Link, 1's complement the product. Clear EAE sign bit in either case.

The MUL and ABS instructions can be micro-coded to become the MULS instruction which is called Signed Multiply.

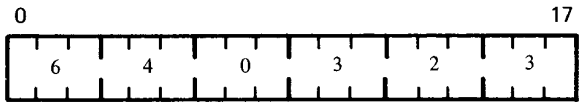
MULS **Signed Multiply**



Execute Time: 7.4 μs

This is a combination of MUL and ABS. In signed multiplication, it is assumed that a GSM instruction has already been performed on the multiplicand. See explanation for MUL.

DIV **Divide Unsigned**



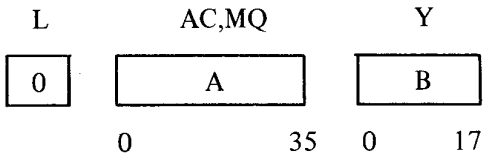
Execute Time: 7.65 μs

The content of the memory location Y which follows immediately after the DIV instruction divides the contents of the accumulator and multiplier quotient register (an unsigned 36-bit dividend). The resulting quotient appears in the multiplier quotient register, and the remainder is in the accumulator.

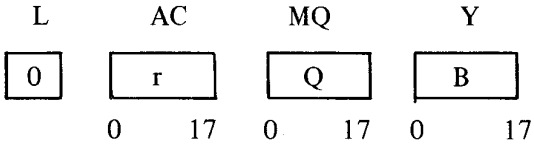
Prior to this instruction, the Link must be zeroed, and the dividend must be entered into the accumulator and multiplier quotient register. This is done with a LAC (least significant half), LMQ, LAC (most significant half) sequence. If the divisor is not greater than the accumulator portion of the dividend, divide overflow occurs (the magnitude of the quotient exceeds the 18-bit capacity of the multiplier quotient register) and the Link is set to a 1 signaling this overflow condition. In this case, the data in the accumulator and multiplier quotient registers is of no value.

A valid division halts when the step counter, initialized to the 2's complement of 23_8 (19_{10} steps) by the six low-order bits of the instruction, counts to 0. The divisor remains unchanged in core, and the program resumes at the next instruction – two after the DIV. The process is indicated below.

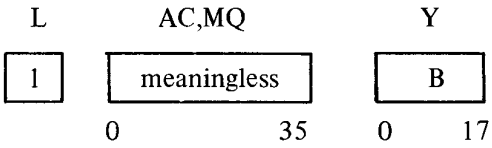
Pre-execution



Post-execution
(no overflow)



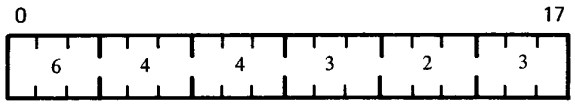
(overflow)



Instruction Sequence:

Memory Location	Contents
Y - 4	LAC Dividend (least significant)
Y - 3	LMQ
Y - 2	LAC Dividend (most significant half)
Y - 1	DIV
Y	Divisor
Y + 1	Next instruction

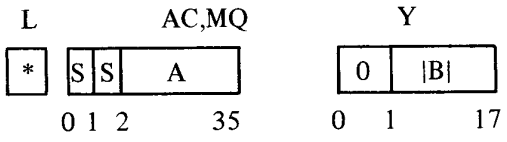
DIVS **Signed Division**



Execute Time: 7.65 μs

Divide the contents of the AC and MQ (a 36-bit signed dividend with the sign in AC_0 and bits 0 and 01, and the remaining 34 bits devoted to magnitude) by the contents of memory location Y (the divisor). The resulting quotient appears in the MQ with the algebraically determined sign in MQ bit 0 and the magnitude (1's complement) in MQ bits 1 through 17. The remainder is in the AC with AC bit 0 containing the magnitude (1's complement). The address of Y is taken to be sequential to the address of the DIVS instruction word. The contents of Y are taken to be the absolute value of the divisor; the contents of the Link are taken to be the orignal sign of the divisor (DIVS assumes previous execution of an EAE GSM instruction). Prior to this DIVS instruction, the dividend must be entered in the AC and MQ (LAC of least significant half, LMQ, and LAC of most significant half). The MQ portion of a negative dividend is 1's complemented prior to the division. If the divisor is not greater than the AC portion of the dividend, divide overflow occurs (magnitude of the quotient exceeds the 17 bit plus sign capacity of the MQ), and the Link is set to one to signal the overflow condition; data in the AC and the MQ are of no value. A valid division halts when the step counter, initialized to the 2's complement of 23_8 (19_{10} steps), counts up to 0 (the six low-order bits of the DIVS instruction word specify the step count). The content of the Link is cleared to 0. The contents of Y are unchanged. The program resumes at the next instruction (memory location Y + 1).

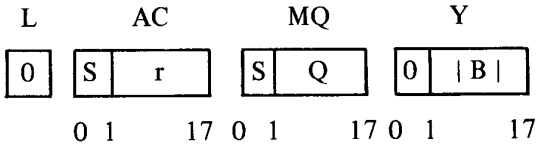
Pre-execution



*original sign of B

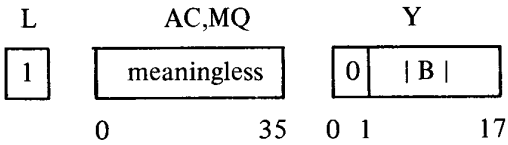
Post-execution

(no overflow)



(S=Sign A) (S=Sign A ∀ L)

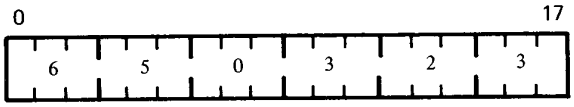
(overflow)



Instruction Sequence:

Memory Location	Contents
Y - 7	LAC Divisor
Y - 6	GSM
Y - 5	DAC Divisor in Y
Y - 4	LAC Dividend (least significant half)
Y - 3	LMQ
Y - 2	LAC Dividend (most significant half)
Y - 1	DIVS
Y	Divisor (absolute value)
Y + 1	Next Instruction

FRDIV **Fraction Divide Unsigned**

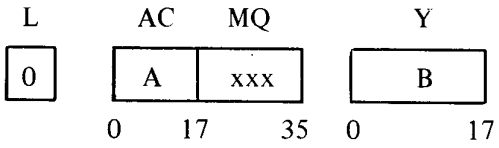


Execute Time 7.65 μs

Divide the contents of the AC and the MQ (AC contains an 18-bit fractional dividend, MQ is zeroed at setup) by the contents of memory location Y (the divisor). The binary point is assumed at the left of AC (bit 0). The quotient

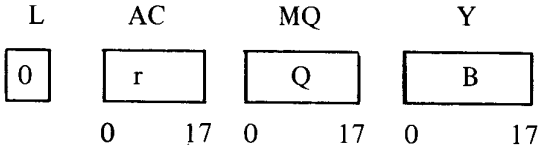
appears in the MQ; the remainder is in the AC. The address of Y is taken to be sequential to the address of the FRDIV instruction word. Prior to this instruction, the contents of the Link must be 0, and the dividend must be entered in the AC (the set-up phase of FRDIV clears the MQ). If the divisor is not greater than the dividend, divide overflow occurs (magnitude of quotient exceeds the 18-bit capacity of the MQ), and the Link is set to 1 to signal the overflow condition; data in the AC and the MQ are of no value. A valid division halts when the step counter, initialized to 23_8 (19_{10} steps), counts up to 0 (the six low-order bits of the FRDIV instruction word specify the step count). The contents of the Link remain 0. The contents of Y are unchanged. The program resumes at the next instruction (memory location Y + 1).

Pre-execution

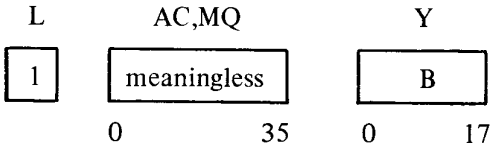


Post-execution

(no overflow)

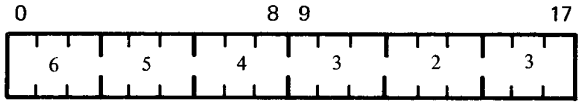


(overflow)



Memory Location	Contents
Y - 2	LAC Dividend
Y - 1	FRDIV
Y	Divisor
Y + 1	Next Instruction

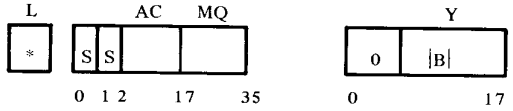
FRDIVS **Fraction Divide Signed**



Execute Time: 7.65 μs

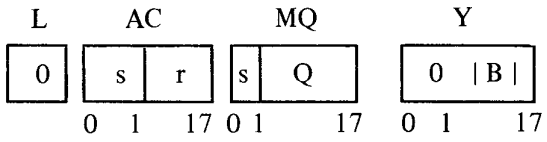
Divide the content of the AC and the MQ (AC contains a signed fractional dividend, MQ is zeroed at set-up) by the contents of memory location Y (the divisor). The binary point is assumed between AC bit 1 and AC bit 2. The resulting quotient appears in the MQ with the algebraically determined sign in MQ bit 0 and the magnitude (1's complement) in MQ bits 1 through 17. The remainder is in the AC with bit 0 containing the original sign of the dividend and bits 1-17 containing the magnitude (1's complement). The address of Y is taken to be sequential to the address of the FRDIVS instruction word. The contents of Y are taken to be the absolute value of the divisor; the contents of the Link are taken to be the original sign of the divisor (FRDIVS assumes previous execution of an EAE GSM instruction). Prior to this FRDIVS instruction, the dividend must be entered in the AC (the set-up phase of FRDIVS clears the MQ and 1's complements the dividend, if negative, prior to the division). If the divisor is not greater than the dividend, divide overflow occurs (magnitude of the quotient exceeds the 18-bit capacity of the MQ) and the Link is set to 1, to signal the overflow condition. Data in the AC and the MQ are of no value. A valid division halts when the step counter, initialized to the 2's complement of 23_8 (19_{10} steps), counts up to 0 (the six low-order bits of the FRDIVS instruction word specify the step count). The contents of the Link are cleared to 0. The contents of Y are unchanged. The program resumes at the next instruction (memory location Y + 1).

Pre-execution



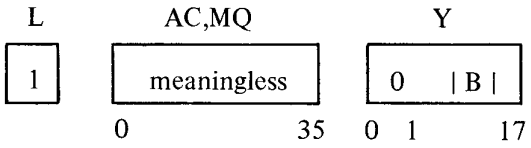
*original sign of B

Post-execution
(no overflow)



(s = Sign A) (s = L ⊻ Sign A)

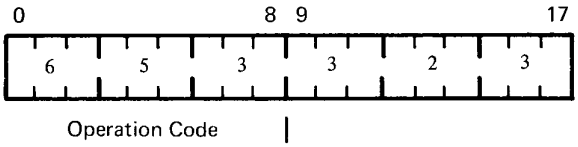
(overflow)



Instruction Sequence:

Memory Location	Contents
Y - 5	LAC Divisor
Y - 4	GSM
Y - 3	DAC Divisor (absolute value) in Y
Y - 2	LAC Dividend
Y - 1	FRDIVS
Y	Divisor (absolute value)
Y + 1	Next Instruction

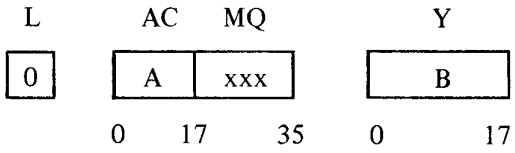
IDIV **Integer Divide Unsigned**



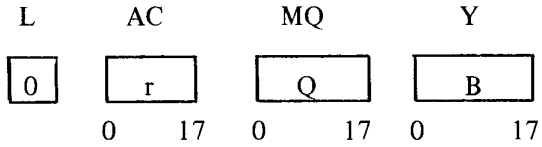
Execute Time: 7.65 μs

Divide the contents of the AC and the MQ (AC is 0, MQ contains a 18-bit integer dividend) by the contents of memory location Y (the divisor). The resulting quotient appears in the MQ; the remainder is in the AC. The address of Y is taken to be sequential to the address of the IDIV instruction word. Prior to this instruction, the contents of the Link must be 0, and the dividend must be entered in the AC (the set-up phase of IDIV transfers the dividend to the MQ and clears the AC). Division overflow occurs only if division by 0 is attempted; i.e., the quotient's magnitude will not exceed the 17-bit plus sign capacity of the MQ. The division halts when the step counter, initialized to the 2's complement of 23_8 (19_{10} steps), counts up to 0 (the six low-order bits of the IDIV instruction word specify the step count). The content of the Link is cleared to 0. The contents of Y are unchanged. The program resumes at the next instruction (memory location Y + 1).

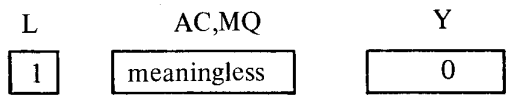
Pre-execution



Post-execution



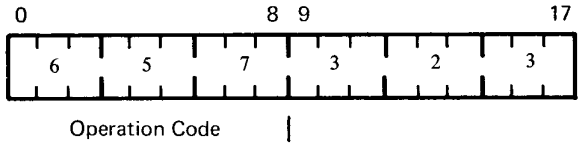
If Y = 0 (overflow)



Instruction Sequence:

Memory Location	Contents
Y - 2	LAC Dividend
Y - 1	IDIV
Y	Divisor
Y + 1	Next Instruction

IDIVS **Integer Divide Signed**

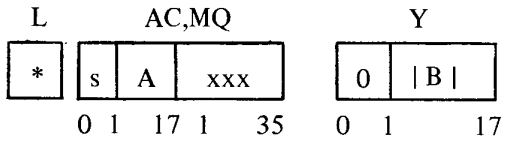


Execute Time: 7.65 μs

Divide the contents of the AC and the MQ (AC is 0, MQ contains a signed integer dividend) by

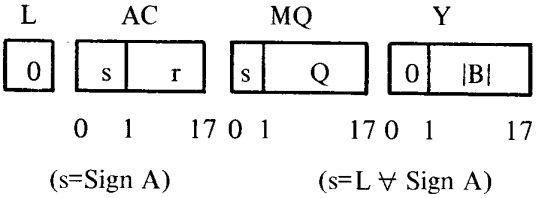
the contents of memory location Y (the divisor). The resulting quotient appears in the MQ with the algebraically determined sign in MQ in bit 0 and the magnitude (1's complement) in MQ bits 1-17. The remainder is in the AC with AC bit 0 containing the sign of the dividend and AC bits 1-17 containing the magnitude (1's complement). The address of Y is taken to be sequential to the address of the IDIVS instruction word. The contents of Y are taken to be the absolute value of the divisor; the contents of the Link are taken to be the original of the divisor (IDIVS assumes previous execution of an EAE GSM instruction). Prior to this IDIVS instruction, the dividend must be entered in the AC (the set-up phase of IDIVS transfers the dividend to the MQ, clears the AC, and 1's complements the MQ if the dividend is negative). Divide overflow occurs only if division by 0 is attempted; i.e., the quotient's magnitude will not exceed the 17-bit plus sign capacity of the MQ. The division halts when the step counter, initialized to the 2's complement of 23_8 (19_{10} steps), counts up to 0 (the six low-order bits of the IDIVS instruction word specify the step count). The contents of the Link are cleared to 0. The contents of Y are unchanged. The program resumes at the next instruction (memory location Y + 1).

Pre-execution

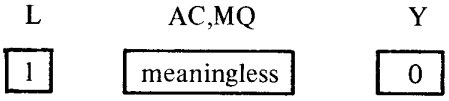


*original sign of B

Post-execution



If Y = (overflow)



Instruction Sequence:

Memory Location	Contents
Y - 5	LAC Divisor
Y - 4	GSM
Y - 3	DAC Divisor (absolute value) in Y
Y - 2	LAC Dividend
Y - 1	IDIVS
Y	Divisor (absolute value)
Y + 1	Next Instruction

FLOATING-POINT PROCESSOR – FP15

The FP15 Floating-Point Processor performs single- and double-precision floating-point arithmetic. The FP15 is a hardware option for systems that can perform arithmetic operations ten times faster than existing software routines. It features 9-digit precision arithmetic on numbers within the $10^{-131,072}$ to $10^{131,071}$ range.

The FP15 is a complete processor with its own instruction set that interfaces directly with up to 128K of memory and monitors every instruction fetched by the KP15 Central Processor. When it recognizes a floating-point instruction, the FP15 inhibits the KP15 and begins the specified function. This in-line mode of operation simplifies programming and minimizes execution time.

All FP15 instructions are two 18-bit word instructions. The first word is the specified operation and the second word is the direct or indirect address of the data in memory. The instructions involved in performing a typical floating-point operation, such as a double-precision floating-point multiply, normalized and rounded, are as follows:

First Instruction DLD (Load double-precision floating-point number)

The first word of this two-word instruction specifies the operation (load) and the format (double-precision, floating-point). The second word is the address of the data (a three-word buffer).

The operation requires five memory cycles (six if indirect addressing is specified). Total time required is 4.0 μs.

Second Instruction — DMP (Multiply double-precision, floating-point number, normalized and rounded)

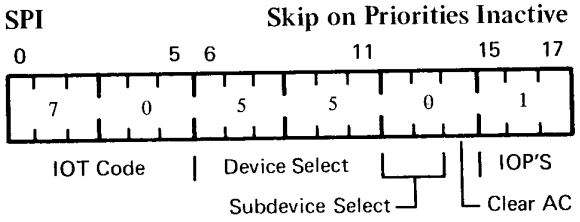
The first word of this instruction specifies the operation (multiply), type of data (double-precision floating-point) and the conditions (normalized, rounded). The second word is the address of the data. The operation requires a maximum of 20 μs (typically 15), including instruction and data fetch. However, only five memory cycles are required (six if indirect addressing is specified). Total time required if direct addressing is specified is 4.0 μs.

Representative double-precision, floating-point execution times, in microseconds, are:

Operation	Max.	Typ.
Floating-point add	15	10
Floating-point subtract	15	10
Floating-point multiply	20	15
Floating-point divide	21	18
Floating-point load	15	10
Floating-point store	15	10

A complete description of the FP15 Floating-Point Processor instruction set is provided in the *FP15 Floating-Point Processor Reference Manual,* DEC-15-HQEA-D.

AUTOMATIC PRIORITY INTERRUPT – KA15 INSTRUCTION SET



Execute Time: 2 μs

This instruction compares a condition code in the accumulator with part of the ENABLE bit and priorities active register. If any bit of the condition code matches the corresponding bit of the ENABLE or priority active register and both are set, the next instruction is skipped. Otherwise the next instruction is executed. The corresponding bits are shown in Figure B-3.

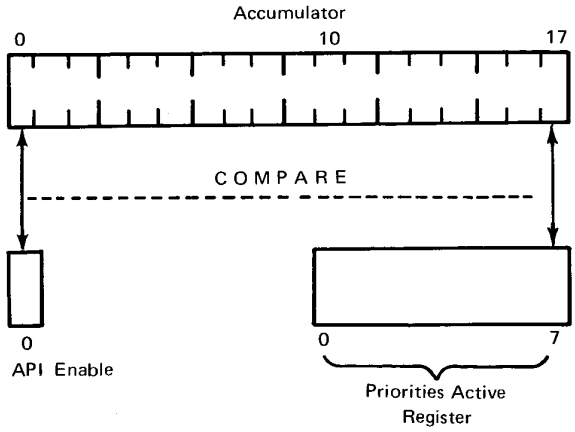
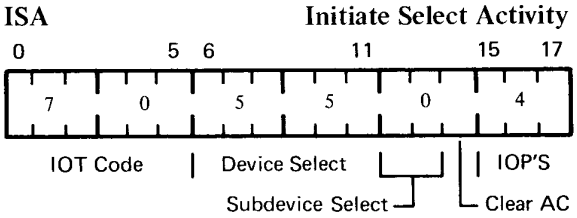
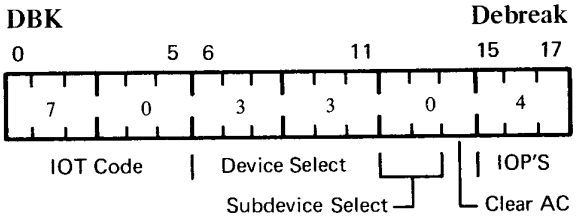


Figure B-3 Skip on Priorities Inactive



Execute Time: 4 μs

The content of accumulator bit 0 is placed into the ENABLE flip-flop; accumulator bits 6 through 9 are ORed into bits 4 to 7 of the API request register, and accumulator bits 10 through 17 are ORed into bits 0 through 7 of the API priorities active register. A diagram of this follows.

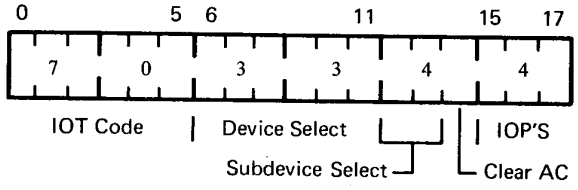


Execute Time: 4 μs min.
5 μs max.

This instruction is used to release the highest active priority level. Its use is to return a subroutine's priority to the normal assignment after the requirement for an interim ISA-

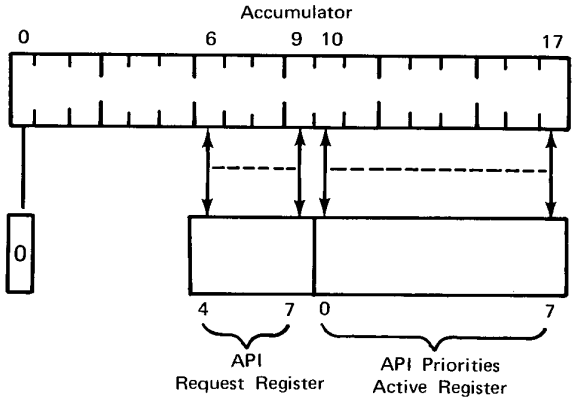
initiated raising of priority has been satisfied. DBK should not be used to terminate a subroutine as it does not provide for restoration of the PC, Link, etc.

DBR **Debreak and Restore**



Execute Time: 4 μs

This instruction zeroes the highest priority presently in the priority active register, thus clearing the way for future requests. It also primes the PDP-15 to restore the Link, the program counter and user mode to their status at the time the API request was honored. The actual restoration occurs at the execution of any indirect instruction. However, a JMP indirect is usually used exiting the subroutine which must immediately follow the DBR instruction.

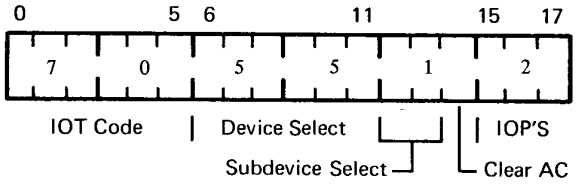


Programming Note – Normally, the SPI and ISA instructions are used sequentially to first test that the program segment currently in progress is not already at the requested priority level and then if not to initiate a raising of priority to the requested level. Hence, if a program segment cannot raise its priority, the segment must be already at the requested level or higher. The ISA instruction cannot be used to lower the priority level of an active program segment. The hardware will not recognize the priority change.

RES **Restore**

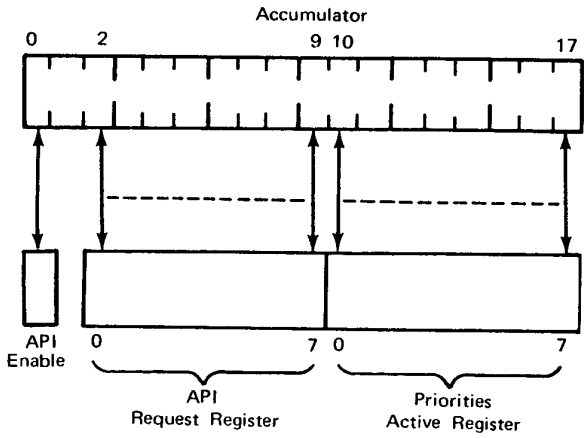
Restore the status of the Link, Bank Mode, and User Mode, at the first indirect instruction after it is executed. The RES does not, however, affect the API priority levels.

RPL **Read Priority Levels**



Execute Time: 3 μs min.
4 μs max.

The contents of the API ENABLE flip-flop is read into accumulator bit 0, the content of the API request register is read into accumulator bits 2 through 9, and the content of the priorities active register is read into accumulator bits 10 through 17, as shown in the following illustration.



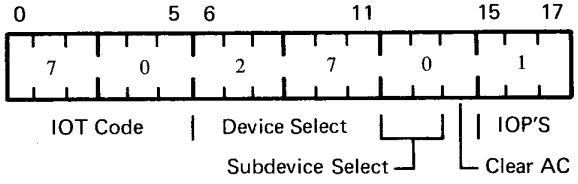
MP15 MEMORY PARITY

The MP15 Memory Parity option adds a 19th bit to each of the words in a 4096-word core memory module. This bit retains the parity indication for the associated word. Parity is generated when a word is written into memory

and checked when the word is read from the memory location. Detection of a parity error can initiate a program interrupt. Systems with the MP15 Memory Parity option have a 1.1 μs memory cycle time. Table B-1 shows PDP-15 cycle times for various combinations of MP15, KM15, and KT15 options.

The IOT instructions associated with memory parity are as follows:

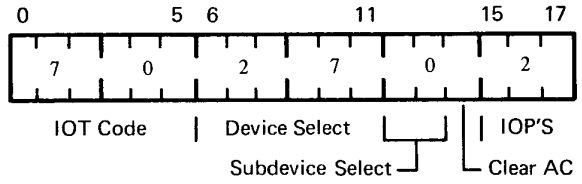
SPE **Skip on Parity Error**



Execute Time: 2 μs min.
3 μs max.

If the parity error flag is posted, the program counter is incremented by 1 and the next instruction is skipped.

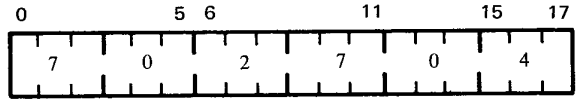
CPE **Clear Parity Error**



Execute Time: 3 μs min.
4 μs max.

The parity error flag is cleared to 0 when this instruction is executed.

FWP **Force Wrong Parity**



Used for maintenance purposes only. Permits check of parity detection logic by forcing wrong parity bit to be written into memory.

KW15 REAL-TIME CLOCK

The real-time clock, when enabled, counts in memory location 00007 the number of cycles completed by any one of three inputs:

a. The line voltage (50 or 60 Hz)

b. An M401 R-C Clock (offered as standard) which can be set to any frequency from 0 to 10 kHz.

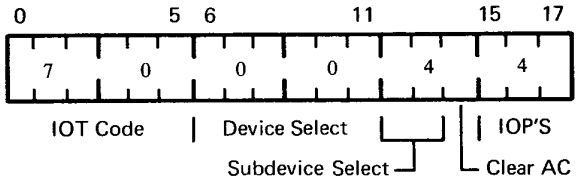
c. A user supplied TTL compatible signal that is fed to a point on the PDP-15 logic.

When location 00007 overflows, an interval program interrupt or API request, if available, is generated informing the monitor that its preset interval is over. The monitor must either disable the clock or reinitialize location 00007 to the 2's complement of the number of counts it needs to tally.

The incrementing of location 00007 during a real-time clock request occurs via the I/O Processor, using its increment-memory facility. A real-time clock request takes priority over API, PI and IOT requests.

The following IOT instructions are provided for use with the clock:

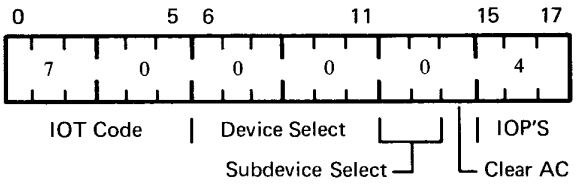
CLON **Clock On**



Execute Time: 4 μs min.
5 μs max.

The real-time clock is enabled to begin incrementing location 00007, and its flag is cleared.

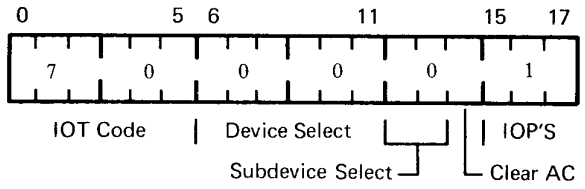
CLOF **Clock Off**



Execute Time: 4 μs min.
5 μs max.

The real-time clock is disabled, preventing it from incrementing location 00007.

CLSF **Skip on Clock Flag**



Execute Time: 2 μs

The program counter is incremented and the next instruction skipped if the clock flag is set.

While the facility is enabled, requests for clock breaks have priority of acceptance over API and PI requests. The first clock break may occur at any time up to 17 ms after the facility has been enabled. The clock switch on the console can inhibit the clock from incrementing location 7.

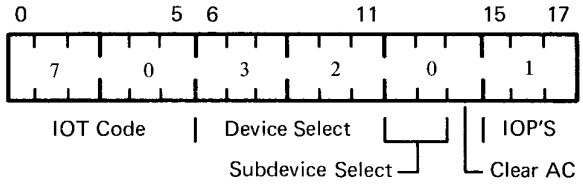
KF15 POWER FAIL

The PDP-15 contains circuitry which provides optimum protection of programs during machine turn-on or turn-off, whether accidental or intended. In the basic machine, the circuitry is designed to protect the contents of memory. However, the addition of the power fail option, an option of the I/O Processor, further allows time to store active registers before the system stops during a power failure, and a system restart and the subsequent restoration of these registers when system power is reapplied.

The basic PDP-15 is not affected by power interruptions of less than 10 ms duration. Active registers in the processor (AC, AR, PC, etc.) will lose their contents when interrupts of longer duration occur but memory will not be disturbed. The power failure protection option provides for saving the contents of active registers in the event of longer power interrupt and for automatic restart of the system when power is restored. The restart feature is switch-selected by the operator to be enabled or disabled. When enabled, the program in progress resumes execution at location 00000. The system must be operating with the program interrupt facility (or the API) enabled to sense the option's initiation of a program interrupt to save the register contents at the time of the line power failure. If API is available the power failure option interrupts on its highest level and traps to memory address 52.

There is only one instruction associated with the Power Failure Option. That is:

PFSF **Skip On Power Low Flag**



Execute Time: 2 μs

The state of the power low flag is tested, and if set, indicating that system line voltage has dropped and that this flag has posted an interrupt, then the reset instruction is skipped. The flag is cleared by the power clear signal when the power interruption is over.

KM15 MEMORY PROTECTION

The memory protection feature establishes a foreground/background environment for PDP-15 processing activity by specifying the boundary between protected (lower) and unprotected (upper) regions of system core memory. Allocation of memory locations (in increments of 256 words) to the protected region is dynamic and program controlled. A boundary register, added by the option, stores the location of the upper limit of the protected region. It is loaded from bits 1 through 9 of the AC by a MPLD (Load Boundary Register) instruction.

The KM15 checks the instruction about to be executed, and the protect feature transfers control to a monitor program should the instruction be in the category of "illegal instructions", before the instruction is executed. If a program tries to reference a nonexistent memory bank, the KM15, if it has been enabled, transfers control to the monitor program. The protect feature increases all memory cycle times as indicated in Table B-1.

The memory protect (or user mode) may be enabled either by programmed instruction, or by depressing the PROT switch on the console, and pressing the START key. When enabled, the PROT indicator lights and the KM15 will trap the following instructions:

Table B-1
Memory Cycle Times*

PDP-15 Configuration	In User Mode				Not In User Mode			
	Read		Write		Read		Write	
	Max.	Typ.	Max.	Typ.	Max.	Typ.	Max.	Typ.
Basic PDP-15	800	800	800	800	800	800	800	800
KM15 Memory Protect option	830	800	975	920	830	800	830	800
KM15 Memory Protect and KT15 Memory Protect/ Relocate options	1,165	1,080	1,165	1,080	965	880	965	880
MP15 Memory Parity option	1,100	1,050	1,100	1,050	1,100	1,050	1,100	1,050
MP15 Memory Parity and KM15 Memory Protect options	1,130		1,255		1,130		1,130	
MP15 Memory Parity, KM15 Memory Protect, and KT15 Memory Protect/ Relocate options	1,355		1,355		1,155		1,155	

*Time indicated in nanoseconds

IOT	Input/Output
CAF	Clear All Flags
XCT of XCT	Chained Execute Instructions
HLT	Halt
OAS	OR AC From Data Switches

References to nonexistent memory

References to locations below the boundary limit

Trapping causes the execution of an effective JMS instruction after the machine cycle that attempts to violate. The address referenced by the effective JMS instruction will be either location absolute 20, if the program interrupt facility is disabled, or location absolute 0, if the program interrupt facility is enabled. The violation flag is set.

The nonexistent memory flag is also set if the violation was caused by a program or I/O Processor reference to nonexistent memory. A single cycle I/O processor reference to nonexistent memory will be detected by the KM15.

User mode is disabled in the following ways:

RESET Key

The detection of a violation

CAL Instruction (which never causes a violation)

A Program Interrupt

An API Interrupt

If user mode is enabled when an API break starts, and the API channel address contains a HLT, OAS, or IOT – rather than the normal JMS – that instruction will be inhibited, user mode will be disabled in the normal fashion and no violation will be detected.

If user mode is disabled when a reference to nonexistent memory is made, the nonexistent memory flag is set and the processor hangs. STOP and RESET must be pressed to clear this condition. If a reference to nonexistent memory occurs when user mode is enabled, the violation flag is also set and the trap occurs.

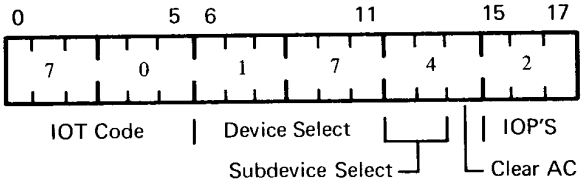
HLT, OAS, and IOT instructions are totally inhibited when the memory protect option is enabled. If the HLT or OAS is combined with any other operate group instruction (microprogramming), the other parts of the operate group instruction are executed before the trap. (The exception is SKP which is not executed.) The second XCT in a chain of XCT instructions is trapped before execution.

The state of the protect mode (a 1 for user mode) is stored in bit 2 of the storage word by the operations that save the state of the machine (CAL, JMS, PI). The stored PC will contain one more than the location of the violating instructions, except for JMP to a protected area. In this case, the stored PC will contain the protected address.

The sole operator control is the PROT switch, which has an indicator above it. This indicator lights when in user mode. The PROT switch is used in conjunction with the START key to establish the proper mode at the beginning of program execution. If the switch is up, then the program is started in user mode. The switch has no further effect.

The RESET key clears the violation and nonexistent memory flags, and user mode (i.e., memory protect is turned off).

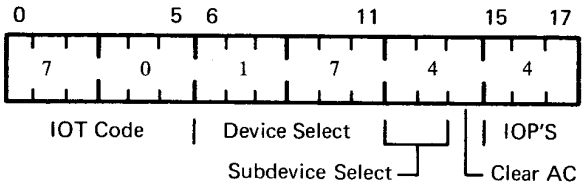
MPEU **Enter User (Protect) Mode**



Execute Time: 4 μs

Memory protect mode will be entered during the fetch cycle of the instruction following MPEU.

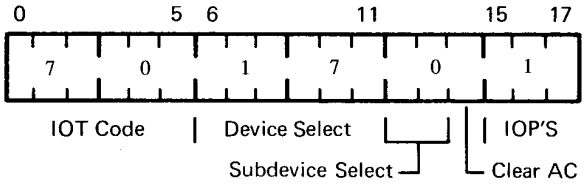
MPCNE **Clear Nonexistent Memory Flag**



Execute Time: 4 μs

The nonexistent memory flag posted when nonexistent memory has been referenced, is cleared by the IOT.

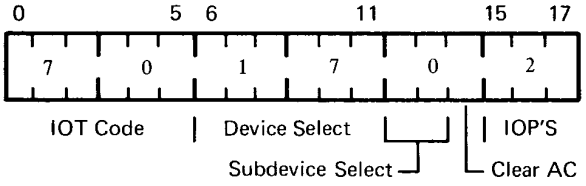
MPSK **Skip On Violation Flag**



Execute Time: 2 μs min.
3 μs max.

The memory protect violation flag will be set whenever the execution of an instruction has violated the provision of memory protection (see above).

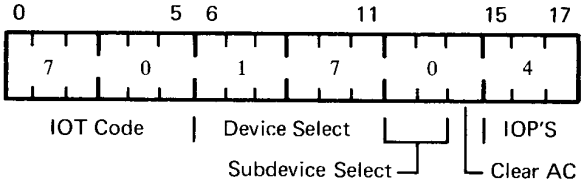
MPCV **Clear Violation Flag**



Execute Time: 3 μs min.
4 μs max.

The violation flag, set if the boundary has been violated or an illegal instruction attempted, is cleared by this IOT.

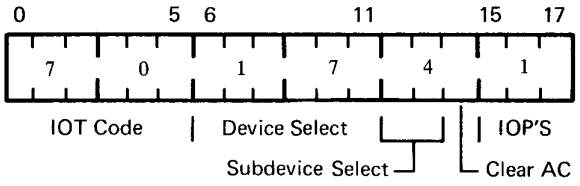
MPLD **Load Memory Protect Boundary Register**



Execute Time: 4 μs min.
5 μs max.

Load the memory protection register with the contents of AC 1 through 9. The boundary register will store the number of 256 word blocks to be protected.

MPSNE **Skip On Nonexistent Memory Flag**



Execute Time: 2 μs min.
3 μs max.

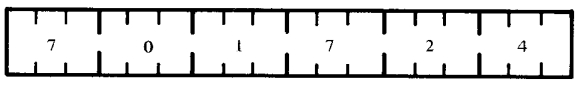
The nonexistent memory flag is set whenever the processor attempts to reference a nonexistent area of core.

KT15 MEMORY RELOCATION AND PROTECT

The relocation and protect hardware consists of two registers: the relocation register and the core allocation register. The relocation register provides the lower limit of the user program and relocates the user upward from real machine-location 0 by the quantity contained in the register. The core allocation register assigns the quantity of locations available to the user. Any attempt by the user to reference core locations exceeding the limit of the core allocation register will cause a protect violation. Memory relocation and protect adds 165 ns to each memory reference cycle, when not in user mode and 365 ns max. when in user mode.

The KT15 option responds to the same IOT instructions as the KM15, plus one additional IOT that loads the relocation register.

MPLR **Load Relocation Register**



Load the relocation register with the contents of AC 1 through 9. The relocation register will be set to the first address to which the user is to be relocated.

Execute Time: 3.96 μs min.
5.02 μs max.

There are two modes of operation in the PDP-15 protect system: user mode and monitor mode. In monitor mode, the relocation and protect hardware is disabled and the machine functions as it would without protect hardware. The program running in monitor mode addresses real locations within the system. In user mode, the relocation and protect hardware is enabled. The machine is programmed as though the user had a machine all to himself. His memory begins from location 0 and goes up to the last 256 page specified by the core allocation register. In the real machine, the program is located from the contents of the relocation register up, however, with the exception of I/O operations and the CAL instructions; the user has the machine virtually to himself and the programs that way. The following special cases occur in user mode.

CAL Instruction

When in user mode, and the CAL is given, the user mode is disabled, (monitor mode invoked). The CAL goes to location 20 in the real machine (not the virtual or relocated machine). The virtual PC link, bank mode, and protect bits are saved in real 20, and real 21 is executed.

Program Interrupt

When a program interrupt occurs, the user mode is disabled, the virtual PC link, bank mode, and protect bits are saved in real location 0, and real location 1 is executed.

Automatic Priority Interrupt

When in user mode, an API break causes monitor mode to be entered, and an instruction in the real machine specified by the I/O device is executed. This instruction will usually be a JMS to the device handler. The device handler will run in monitor mode. The device handler entry will receive the virtual PC, link, bank mode, and protect bits.

Data Channel

Data Channel operations are never relocated.

Real-Time Clock

The real-time clock always increments location 7 in the real machine; attempts to reference the

contents of location 7 in the real machine must be handled through the monitor (with a CAL instruction).

Auto Increment Registers

Each user will have a complete set of auto-increment registers located at locations 10 to 17 of the user's virtual machine. In addition, the monitor can utilize the auto-increment registers in 10 to 17 of the real machine in monitor mode.

DBR Instruction

The debreak and restore instruction (DBR) is used when returning from monitor mode to user mode. The instruction primes the PDP-15 to return the Link and the program counter to their status in user mode. The protect bit, if set, will initiate relocation. Since CAL, PI and API JMS save the virtual PC (not the real PC), the set protect bit causes return to the correct location in real memory, the virtual PC is restored, and user mode is invoked.

THE HARDWARE

The hardware involves a box between the processor and memory which computes relocated addresses and detects protect violations (see Figure B-4).

Two registers of up to 9 (bits 1 to 9) bits are utilized in this scheme. One register is the relocation register whose contents are added to each processor-supplied address when in user mode. The user core-allocation register is subtracted from each processor address and if the result is negative, a violation has occurred. The relocation register is initialized to the user's lower address and the core allocation register is initialized to the quantity of core assigned to the user.

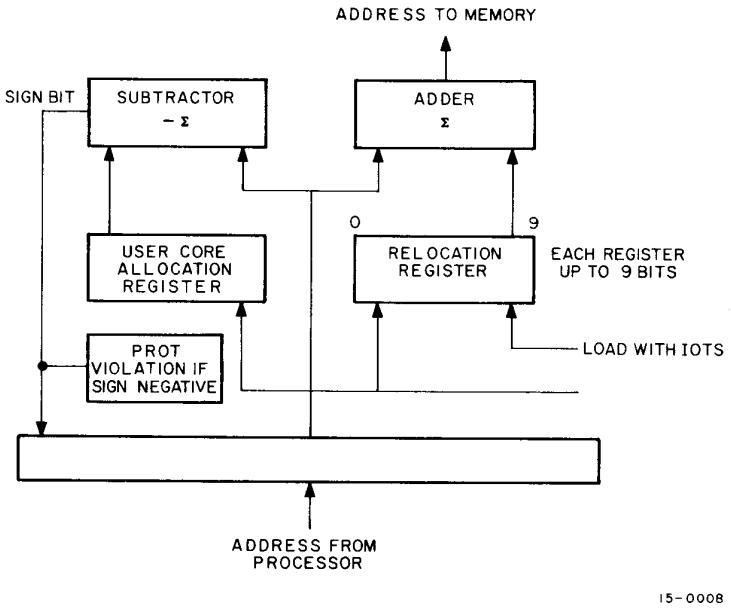


Figure B-4 Hardware

APPENDIX C
EXTERNAL OPTIONS INSTRUCTION SET

CONSOLE TELETYPE KEYBOARD

KSF	700301	Skip on keyboard flag	2.21-3.27
KRB	700312	Read keyboard to AC10-17	3.21-4.27
KRS	700332	Read Full Duplex (AC10-17) and select Reader	3.21-4.27

CONSOLE TELETYPE TELEPRINTER

TSF	700401	Skip on teleprinter flag	2.21-3.27
TCF	700402	Clear teleprinter flag	3.21-4.27
TLS	700406	Clear teleprinter flag, load teleprinter buffer and print	3.96-5.02
SPCO	703341	Skip if PC15	3.21-4.27

TYPE PC15 PAPER TAPE READER

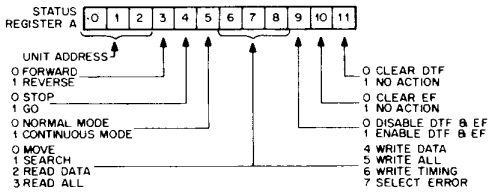
RSF	700101	Skip on reader flag	2.21-3.27
RCF	700102	OR reader buffer to AC	3.21-4.27
RSA	700104	Select alphanumeric mode	3.96-5.02
RRB	700112	Read reader buffer	3.21-4.27
RSB	700144	Select binary mode	3.96-5.02

TYPE PC15 PAPER TAPE PUNCH

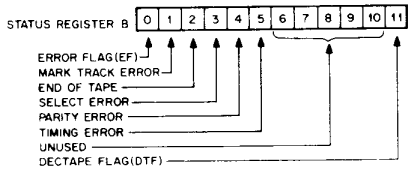
PSF	700201	Skip on punch flag	2.21-3.27
PCF	700202	Clear punch flag	3.21-4.27
PSA	700204	Punch, alphanumeric	3.96-5.02
PSB	700244	Punch, binary	3.96-5.02

INPUT/OUTPUT TRANSFER INSTRUCTIONS (CONT'D)

Mnemonic	Code	Operation	Execute Time (Microseconds)
TYPE TC15 DECTAPE CONTROL			
DTCA	707541	Clear status register A	2.21-3.27
DTXA	707544	XOR AC into status register A	3.96-5.02
DTLA	707545	Load status register A	3.96-5.02
DTRA	707552	Read status register A→AC (0-11)	3.21-4.27

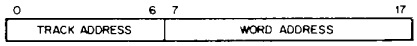


DTEF	707561	Skip on error flag	2.21-3.27
DTRB	707572	Read status register B	3.21-4.27



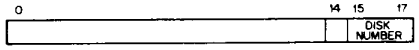
DTDF	707601	Skip on DECtape flag	2.21-3.27
RF15/RS09 DECdisk			
DSSF	707001	Skip on disk flag	2.21-3.27
DSCC	707021	Clear the disk control and disable the "freeze" status. A "freeze" is caused either by a timing or data track hardware error or an address parity error. It forces the control to abort the operation in progress.	2.21-3.27
DRAL	707022	OR the contents of Address Pointer 0 (AP0) into the AC	3.21-4.27

APO



DRAH	707062	OR the contents of Address Pointer 1 (AP1) into the AC	

AP1



DLAL	707024	Load the contents of the AC into AP0	3.96-5.02
DLAH	707064	Load the contents of the AC into AP1	3.96-5.02

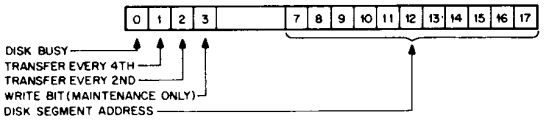
INPUT/OUTPUT TRANSFER INSTRUCTIONS (CONT'D)

Mnemonic	Code	Operation	Execute Time (Microseconds)
DSCF*	707041	Clear the function register and the interrupt mode	2.21-3.27
DSFX*	707042	XOR the contents of AC15, 16, 17 into the function register	3.21-4.27

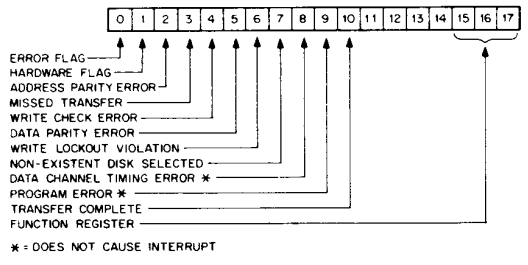
Function Register

15	16	17	
F0	F1	INT	
0	0	X	no effect
0	1	X	read
1	0	X	write
1	1	X	write check

DSCN*	707044	Execute the condition held in the function register	3.96-5.02
DLOK	707202	OR the disk segment address (ADS) register into the AC	3.21-4.27



DSCD	707242	Clear the Status Register and Disk Flag	3.21-4.27
DSRS	707262	OR the Status Register with the AC	3.21-4.27



DRBR	707002	OR the contents of the Buffer register with the accumulator. Primarily for maintenance	3.21-4.27
DLBR	707004	Load the contents of the accumulator into the Buffer Register. Primarily for maintenance	3.96-5.02
DGHS	707204	Generate Simulated Head signals. Primarily for maintenance	3.96-5.02
DGSS	707224	Generate Simulated Disk signals. Primarily for maintenance	3.96-5.02

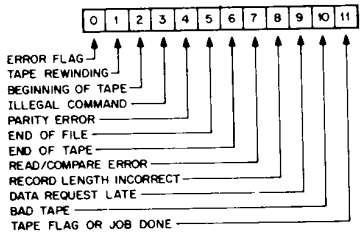
*These instructions may be micro-coded in any combination.

INPUT/OUTPUT TRANSFER INSTRUCTIONS (CONT'D)

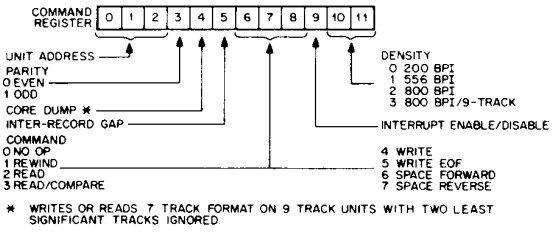
Mnemonic	Code	Operation	Execute Time (Microseconds)

TYPE TC59D MAGNETIC TAPE CONTROL

MTSF	707341	Skip on error flag (EF) or magnetic tape flag (MTF)	2.21-3.27
MTCR	707321	Skip on tape control ready (TCR)	2.21-3.27
MTTR	707301	Skip on tape transport ready (TUR)	2.21-3.27
MTAF	707322	Clear status and command registers and EF and MTF if TCR. If not, TCR only clears EF and MTF.	3.21-4.27



MTCM	707324	OR AC0-5 and 9-11 into command register. Jam AC6-8	3.96-5.02
MTLC	707326	Load AC0-11 into command register. MTCL is the summation of MTAF and MTCM	3.96-5.02
——	707342	OR status register into AC0-11	3.21-4.27
MTRS	707352	Read status register into AC0-11	3.21-4.27
——	707302	OR command register into AC0-11	3.21-4.27
MTRC	707312	Read command register into AC0-11	3.21-4.27
MTGO	707304	Execute command in command register	3.96-5.02

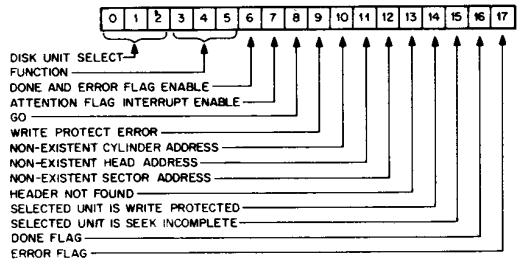


INPUT/OUTPUT TRANSFER INSTRUCTIONS (CONT'D)

Mnemonic	Code	Operation	Execute Time (Microseconds)

TYPE RP15 DISK PACK CONTROL

DPSF	706301	Skip on disk flag	2.21-3.27
DPOSA	706302	OR status register A into the AC	3.21-4.27
DPRSA	706312	Read status register A into the AC	3.21-4.27

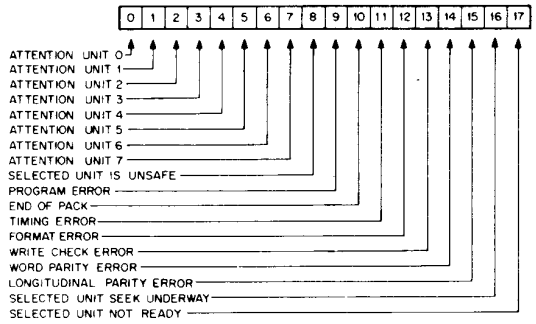


	706402	OR the unit cylinder address register into the AC	3.21-4.27
DPRU	706412	Read the unit cylinder address register into the AC	3.21-4.27
DPLF	706464	Load status register A from AC0-8 and execute	3.96-5.02
DPLA	706304	Load the cylinder, head, and sector address registers from the AC	3.96-5.02
DPCA	706344	Load current address register	3.96-5.02

DPWC	706364	Load word count register	3.96-5.02
DPOA	706422	OR the cylinder, head, and sector address register into the AC	3.21-4.27
DPRA	706432	Read the cylinder, head, and sector address into the AC	3.21-4.27
DPOC	706442	OR the current address register into the AC	3.21-4.27
DPRC	706452	Read the current address register into the AC	3.21-4.27
DPOW	706462	OR the word count register into the AC	3.21-4.27
DPRW	706472	Read the word count register into the AC	3.21-4.27
DPSA	706321	Skip on attention flag	2.21-3.27
DPOSB	706322	OR status register B into the AC	3.21-4.27
DPRSB	706332	Read status register B into the AC	3.21-4.27
DPSJ	706341	Skip on job done flag	2.21-3.27
DPSE	706361	Skip on error flag	2.21-3.27

INPUT/OUTPUT TRANSFER INSTRUCTIONS (CONT'D)

Mnemonic	Code	Operation	Execute Time (Microseconds)



DPLZ	706424	Load AC0-8 zeros into status register A bits 0-8 and execute	3.96-5.02
DPLO	706444	Load the AC0-8 ones into status register A bits 0-8 and execute	3.96-5.02
DPCN	706454	Execute the function register; clear the accumulator	3.96-5.02
DPCS	706324	Clear status	3.96-5.02
DPCF	706404	Clear function	3.96-5.02

TYPE VP15A STORAGE TUBE DISPLAY

CXB	700502	Clear x-coordinate buffer	3.21-4.27
CYB	700602	Clear y-coordinate buffer	3.21-4.27
LXB	700504	Load x-coordinate buffer from AC8-17	3.96-5.02
LYB	700604	Load y-coordinate buffer from AC8-17	3.96-5.02
EST	700724	Erase storage tube	3.96-5.02
SDDF	700521	Skip on display done flag	2.21-3.27
CDDF	700722	Clear display done flag	3.21-4.27
LXBD	700564	Load x-coordinate buffer and display the point specified by XB and YB (store mode)	3.96-5.02
LYBD	700664	Load the y-coordinate buffer and display the point specified by XB and YB (store mode)	3.96-5.02
LXDNS	700544	Load the x-coordinate buffer and display the point specified by XB and YB (non-store mode)	3.96-5.02
LYDNS	700644	Load the y-coordinate buffer and display the point specified by XB and YB (non-store mode)	3.96-5.02

TYPE VP15M STORAGE TUBE DISPLAY MULTIPLEXER
(Identical to VP15A Plus)

LUDR	700764	Load unit designation register. AC10-17 correspond to units 0-7	3.96-5.02

INPUT/OUTPUT TRANSFER INSTRUCTIONS (CONT'D)

Mnemonic	Code	Operation	Execute Time (Microseconds)

TYPE VP15B, VP15BL, VP15C, VP15CL
OSCILLOSCOPE DISPLAYS

DXL	700504	Load the x-coordinate buffer from AC8-17	3.96-5.02
DXS	700544	Load the x-coordinate buffer and display the point specified by the XB and YB	3.96-5.02
DYL	700604	Load the y-coordinate buffer from AC8-17	3.96-5.02
DYS	700644	Load the y-coordinate buffer and display the point specified by the XB and YB	3.96-5.02
DXC	700502	Clear the x-coordinate buffer	3.21-4.27
DYC	700602	Clear the y-coordinate buffer	3.21-4.27
DLB	700704	Load the brightness register from bits 16-17 of the AC. Note: This instruction clears the display flag associated with the light pen	3.96-5.02
DSF	700501	Skip if display (light pen) flag is a 1	2.21-3.27
DCF	700702	Clear display (light pen) flag	3.21-4.27

TYPE LT15 TELETYPE CONTROL

Teleprinter

TSF1	704001	Skip on transmit flag	2.21-3.27
TCF1	704002	Clear transmit flag	3.21-4.27
TLS1	704004	Load transmit buffer and send character to teleprinter	3.96-5.02

Keyboard

KSF1	704101	Skip on receiver flag	2.21-3.27
KRB1	704102	Read receiver buffer and clear receiver flag	3.21-4.27

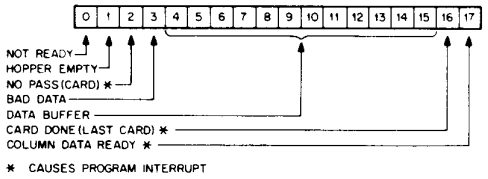
Mnemonic	Code	Operation	Execute Time (Microseconds)
		TYPE LT19D TELETYPE CONTROL	
		Teleprinter	
	704XX1*	Skip on transmit flag	2.21-3.27
	704XX2*	Clear transmit flag	3.21-4.27
	704XX4*	Load transmit buffer and send character	3.96-5.02
		Keyboard	
	704XX1*	Skip on receiver flag	2.21-3.27
	704XX2*	Read receiver buffer and clear receiver flag	3.21-4.27

UNIT #	XX (Keyboard)	XX (Teleprinter)
1	10 (or LT15)	00 (or LT15)
2	12	02
3	14	04
4	16	06
5	30	20

(see PDP-15 User Handbook Vol. II for additional units)

*Where XX is code for Teletype unit (LT19E)

		TYPE CRO3B CARD READER	
CRSC	706722	Clear status register and data buffer; select a card	3.21-4.27
CROR	706712	Load data and status into AC	3.21-4.27



CRCS	706704	Clear status register and data buffer	3.96-5.02
CRSI	706721	Skip if card reader interrupt is set	2.21-3.27
CRLA	706734	Load status from AC (maintenance only)	3.96-5.02
		LP15F LINE PRINTER CONTROL	
LPP1	706541	Print one Line	2.21-3.27
LPPM	706521	Print Multilines	2.21-3.27
PLSF	706501	Skip on done and error	2.21-3.27
LPRS	706552	Read status	3.21-4.27
LPEI	706544	Enable interrupt	3.96-5.02
LPDI	706561	Disable interrupt	2.21-3.27
LPCD	706621	Clear done flag	2.21-3.27
LPCS	706641	Clear status and error flags	2.21-3.27

INPUT/OUTPUT TRANSFER INSTRUCTIONS (CONT'D)

Mnemonic	Code	Operation	Execute Time (Microseconds)
		TYPE XY15 INCREMENTAL PLOTTER CONTROL	
PLSF	702401	Skip if plotter flag is a 1	2.21-3.27
PLCF	702402	Clear plotter flag	3.21-4.27
PLPU	702404	Plotter pen up. Raise pen off paper	3.96-5.02
PLPR	702421	Plotter pen right	2.21-3.27
PLDU	702422	Plotter drum (paper) upward	3.21-4.27
PLDD	702424	Plotter drum (paper) downward	3.96-5.02
PLPL	702441	Plotter pen left	2.21-3.27
PLUD	702442	Plotter drum (paper) upward (same as 702422)	3.21-4.27
PLPD	702444	Plotter pen down. Lower pen onto paper	3.96-5.02

PDP-15 IOPS ASCII CHARACTER SET

Listed below are the ASCII characters interpreted by the ADVANCED Monitor and system programs as meaningful data input or as control characters.

	00-37	40-77	100-137	140-177	
	ASCII CHAR.	**ASCII CHAR.**	**ASCII CHAR.**	**ASCII CHAR.**	
0	NUL	SP	@		0
1	SOH (CTRL A)	!	A		1
2		"	B		2
3	ETX (CTRL C)	#	C		3
4	EOT (CTRL D)	$	D		4
5		%	E		5
6		&	F		6
7		'	G		7
10		(	H		10
11	HT	)	I		11
12	LF	*	J		12
13	VT	+	K		13
14	FF	,	L		14
15	CR	-	M		15
16		.	N		16
17		/	O		17
20	DLE (CTRL P)	0	P		20
21	DC1 (CTRL Q)	1	Q		21
22	DC2 (CTRL R)	2	R		22
23	DC3 (CTRL S)	3	S		23
24	DC4 (CTRL T)	4	T		24
25	NACK (CTRL U)	5	U		25
26		6	V		26
27		7	W		27
30	CNCL (CTRL X)	8	X		30
31		9	Y		31
32	SS (CTRL Z)	:	Z		32
33	ESC (ALTMODE)	;	[		33
34		<	~		34
35		=	]	ESC (ALTMODE)	35
36		>	∧ or ↑	ESC (ALTMODE)	36
37		?	←or an (–) underscore	delete (RO)	37

Codes 33, 176, 175 are interpreted as ESC (ALT Mode) and are converted on input to code 175 by IOPS handlers.

API Addresses

Break Address	Standard Device	Suggested Priority Level
40	Software channel 0	4
41	Software channel 1	5
42	Software channel 2	6
43	Software channel 3	7
44	DECtape (TC15)	1
45	MagTape (TC59)	1
46		
47		
50	Paper Tape Reader (PC15)	2
51	Clock Overflow (KW15)	3
52	Power Fail (KF15)	0
53	Parity (MP15)	0
54	VT15	2
55	Card Readers (CR03B)	2
56	Line Printer (LP15)	3
57	A/D (AF01)	0
60	DB99A/DB98A	3
61		
62	Data Phone (DP09A)	2
63	RF15	1
64	RP15	1
65		
66		
67		
70		
71		
72		
73		
74	LT19 & LT15 Teleprinter	3
75	LT19 & LT15 Keyboard	3
76		
77		

Multi-Cycle Data Channel Addresses

Device	Word Count	Current Address
DECtape TC15	30	31
MagTape TC59	32	33
Interprocessor Buffers		
DB99, 98 Transmit	22	23
DB99, 98 Receive	24	25
RF15	36	37
LP15	34	35

FP-15 INSTRUCTION SUMMARY

Mnemonic	Instruction Type	Octal Code	Average Execution Time (μsec)
FPT	Floating Point Test	710314	5.0
ISB	Single Integer Subtract	710400	6.2
ESB	Extended Integer Subtract	710500	7.3
FSB	Single Floating Subtract	710440	8.4
URFSB	Unrounded, Single Floating Subtract	710450	8.4
UNFSB	Unnormalized, Single Floating Subtract	710460	8.3
UUFSB	Unrounded, Unnormalized Single Floating Subtract	710470	8.3
DSB	Double Floating Subtract	710540	11.2
URDSB	Unrounded, Double Floating Subtract	710550	11.2
UNDSB	Unnormalized, Double Floating Subtract	710560	11.2
UUDSB	Unrounded, Unnormalized Double Floating Subtract	710570	11.2
IRS	Single Integer Reverse Subtract	711000	6.2
ERS	Extended Integer Reverse Subtract	711100	7.3
FRS	Single Floating Reverse Subtract	711040	8.6
URFRS	Unrounded, Single Floating Reverse Subtract	711050	8.6
UNFRS	Unnormalized, Single Floating Reverse Subtract	711060	8.5
UUFRS	Unrounded, Unnormalized, Single Floating Reverse Subtract	711070	8.5
DRS	Double Floating Reverse Subtract	711140	11.6
URDRS	Unrounded, Double Floating Reverse Subtract	711150	11.6
UNDRS	Unnormalized, Double Floating Reverse Subtract	711160	11.2
UUDRS	Unrounded, Unnormalized, Double Floating Reverse Subtract	711170	11.2
IMP	Single Integer Multiply	711400	14.1
EMP	Extended Integer Multiply	711500	17.0
FMP	Single Floating Multiply	711440	16.6
URFMP	Unrounded, Single Floating Multiply	711450	16.6
UNFMP	Unnormalized, Single Floating Multiply	711460	16.2
UUFMP	Unrounded, Unnormalized, Single Floating Multiply	711470	16.2
DMP	Double Floating Multiply	711540	18.6
URDMP	Unrounded, Double Floating Multiply	711550	18.6
UNDMP	Unnormalized, Double Floating Multiply	711560	18.2
UUDMP	Unrounded, Unnormalized, Double Floating Multiply	711570	18.2
IDV	Single Integer Divide	712000	11.8
EDV	Extended Integer Divide	712100	14.4
FDV	Single Floating Divide	712040	15.6
URFDV	Unrounded, Single Floating Divide	712050	15.6
DDV	Double Floating Divide	712140	18.3
URDDV	Unrounded, Double Floating Divide	712150	18.3
IRD	Single Integer Reverse Divide	712400	11.8
ERD	Extended Integer Reverse Divide	712500	14.4
FRD	Single Floating Reverse Divide	712440	15.6
URFRD	Unrounded, Single Floating Reverse Divide	712450	15.6
DRD	Double Floating Reverse Divide	712540	18.3
URDRD	Unrounded, Double Floating Reverse Divide	712550	18.3
ILD	Single Integer Load	713000	6.6
ELD	Extended Integer Load	713100	7.8
FLD	Single Floating Load	713050	8.3
UNFLD	Unnormalized, Single Floating Load	713070	7.9
DLD	Double Floating Load	713150	9.5
UNDLD	Unnormalized, Double Floating Load	713170	9.3
IST	Single Integer Store	713600	6.6
EST	Extended Integer Store	713700	7.8
FST	Single Floating Store	713640	7.9
URFST	Unrounded, Single Floating Store	713650	7.9
UNFST	Unnormalized, Single Floating Store	713660	7.7
UUFST	Unrounded, Unnormalized, Single Floating Store	713670	7.7
DST	Double Floating Store	713750	9.1
UNDST	Unnormalized, Double Floating Store	713770	8.9
ILF	Single Integer Load and Float	714010	11.2
UNILF	Unnormalized, Single Integer Load and Float	714030	6.6
ELF	Extended Integer Load and Float	714110	11.0
UNELF	Unnormalized, Extended Integer Load and Float	714130	7.9
FLA	Float FMA	714210	8.2
UNFLA	Unnormalized, Float FMA	714230	5.3
FLX	Single, Floating Load and Fix	714460	11.0
URFLX	Unrounded, Single Floating Load and Fix	714470	11.0
DLX	Double Floating Load and Fix	714560	12.4
URDLX	Unrounded, Double Floating Load and Fix	714570	12.4
FXA	Fix EPA, FMA	714660	8.3
URFXA	Unrounded, Fix EDA, FMA	714670	8.3
ILQ	Single Integer Load FMQ	715000	6.6
ELQ	Extended Integer Load FMQ	715100	7.9
FLQ	Single Floating Load FMQ	715050	14.0
UNFLQ	Unnormalized, Single Floating Load FMQ	715070	7.9
DLQ	Double Floating Load FMQ	715150	9.5
UNDLQ	Unnormalized, Double Floating Load FMQ	715170	9.3
SWQ	Swap FMA and FMQ	715250	5.5
UNSWQ	Unnormalized Swap FMA and FMQ	715270	5.3
LJE	Load JEA Register	715400	6.6
SJE	Store JEA Register	715600	6.6
IAD	Single Integer Add	716000	6.6
EAD	Extended Integer Add	716100	7.9
FAD	Single Floating Add	716040	8.2
URFAD	Unrounded, Single Floating Add	716050	8.2
UNFAD	Unnormalized, Single Floating Add	716060	8.3
UUFAD	Unrounded, Unnormalized Single Floating Add	716070	8.3
DAD	Double Floating Add	716140	9.3
URDAD	Unrounded, Double Floating Add	716150	9.3
UNDAD	Unnormalized, Double Floating Add	716160	9.3
UUDAD	Unrounded, Unnormalized Double Floating Add	716170	9.3
BZA	Branch on Zero FMA	716601	5.2
BMA	Branch on Minus FMA	716602	5.2
BLE	Branch if FMA ≤ 0	716603	5.2
BPA	Branch on Positive FMA	716604	5.2
BRU	Branch Unconditional	716606	5.2
BNA	Branch on Non-Zero FMA	716610	5.2
BAC	Branch if Carry Out of FMA	716620	5.2
FZR	Zero EPA (A SIGN) FMA	711200	5.2
FAB	Make A SIGN positive (Absolute Value)	713271	5.2
FNG	Make A SIGN Negative	713272	5.2
FCM	Complement A SIGN	713273	5.2
FNM	Normalize EPA (A SIGN) FMA	713250	8.4
DMF	Diagnostic Mode Off	717200	5.3
DMN	Diagnostic Mode On	717300	5.3
DRR	Diagnostic Read Registers	710000	16.1
DSR	Diagnostic Step and Read Registers	710100+n	17.6
DBK	Debreak	703304	

VT15 GRAPHIC & PROCESSOR INSTRUCTIONS

Mnemonic	Octal Code	Operation	Function
SPSF	703001	Skip on stop flag.	Causes the computer program to skip next instruction when the stop flag is raised.
SPLP	703021	Skip on light pen flag.	Causes the computer program to skip next instruction when the light pen flag is raised.
SPPB	703041	Skip on pushbutton flag.	Causes the computer program to skip next instruction when any one of six pushbuttons on the display console is depressed.
SPEF	703061	Skip on edge flag.	Causes the computer program to skip next instruction when a vector exceeds a previously defined paper size edge.
SPDF	703101	Skip on any flag.	Causes the computer program to skip next instruction when any flag in the VT15 is raised.
SPDI	703121	Skip on any interrupting flag.	In the VT15 instruction set there is an interrupt and two skip parameter words. The parameter 3 instruction enables a program interrupt on the occurrence of any one of five flags. This IOT instruction will cause a skip when any of the five flags is raised only when the parameter 3 interrupt is also enabled. When the flag is raised and the interrupt enabled, the computer program will be interrupted and a program skip will occur.

VT15 GRAPHIC & PROCESSOR INSTRUCTIONS (Cont)

Mnemonic	Octal Code	Operation	Function
SPES	703161	Skip on external stop flag.	One of the IOTs in this instruction set is an external stop instruction (STPD). This stop originates from the PDP-15 computer program via the STPD IOT. When the external stop IOT is received by the VT15 Graphic Processor, the display will stop and the external stop flag will be raised. This IOT permits the PDP-15 computer program to skip when this flag is raised.
LSD	703004	Load and Start display.	This IOT is used to initialize a single cycle word transfer sequence by the PDP-15 computer and entry into the display file by the VT15. The transfer is initialized by an 18-bit address word which is first loaded into the accumulator. This address is the location of the first instruction in the display file. It is loaded into the VT15 PC when this IOT occurs.
STDP	703044	External stop display.	Provides an external stop of the display. Refer to the SPES instruction described above.
RES	703064	Resume display after flag.	This IOT causes the display to clear all flags and continue after a flag occurs.
SIC	703024	Set initial conditions.	This IOT is used to set initial conditions for computer interrupts, to clear flags, and to establish paper size. The initial conditions are determined by an 18-bit word formed in the PDP-15 accumulator.

PDP-15 IOT INSTRUCTIONS FOR UNICHANNEL 15 INTERRUPT LINK

PDP-15 IOTs	Description
706001	SIOA – Skip I/O Accepted. Tests whether the TCBP done flag is set, indicating the PDP-11 has read the TCBP and skips the next location if the done flag is a 1.
706002	CIOD – Clear I/O Done. Clears the TCBP done flag.
706006	LIOR – Load I/O Register and clear TCBP done flag. Places the contents of the PDP-15 AC into an 18-bit buffer register. The output of the buffer register is seen by the PDP-11 as TCBP at location 767764 and bits 0 and 1 767764. This IOT also causes the TCBP done flag to be cleared and in the PDP-11 causes bit 7 to be set in location 767760, which, in turn, causes the PDP-11 to perform an interrupt at BR 7 to TV location 310.
706112	RDRS – Read Status Register. Clears the AC and loads the contents of the DR15-C Status register into the AC. (This effectively moves the DR15-C enable interrupt bit into bit 17 of the AC.)
706122	LDRS – Load Status Register. Loads the contents of the AC into the DR15-C Status register. (Places value of AB bit 17 in the DR15-C enable interrupts bit.)
706104	CAPI0 – Clear API0 flag in DR15-C.
706124	CAPI1 – Clear API1 flag in DR15-C.
706144	CAPI2 – Clear API2 flag in DR15-C.
706164	CAPI3 – Clear API3 flag in DR15-C.
706101	SAPI0 – Tests the API0 flag in the DR15-C and skips the next instruction if the flag is 1.
706121	SAPI1 – Tests the API1 flag in the DR15-C and skips the next instruction if the flag is 1.
706141	SAPI2 – Tests the API2 flag in the DR15-C and skips the next instruction if the flag is 1.
706161	SAPI3 – Tests the API3 flag in the DR15-C and skips the next instruction if the flag is 1.

APPENDIX D

TECHNICAL REFERENCE LITERATURE

TECHNICAL REFERENCE LITERATURE--SOFTWARE

REF. NO.	TITLE	DESCRIPTION	ORDER NUMBER	STATUS--PRICE
1	PDP-15/10 USERS GUIDE-COMPACT and BASIC I/O Monitor Software Systems	Quick reference summary of operating procedures on PDP-15/10 with Basic 4K or Expanded 8K systems.	DEC-15-GG1A-D	In Stock--$2.00
2	PDP-15/10 SOFTWARE SYSTEM-COMPACT and Basic I/O Monitor Software	COMPACT describes complete on-line preparation of symbolic programs, their assembling and execution in paper tape environment. Basic I/O Monitor describes complete system for creation and operation of relocatable programs; all I/O capabilities between system and/or user's programs and peripheral devices.	DEC-15-GR1A-D	In Stock--$5.00
3	PDP-15/10 COMPACT SYSTEM CHECKOUT PACKAGE	Specifies step-by-step procedures by users which detail operations required to load general-purpose software and to develop and checkout new system tapes. (Line printer output format.)	DEC-15-CRYA-D	In Stock--$5.00 (Order from Program Library)
4	PDP-15/10E BASIC I/O MONITOR SYSTEM CHECKOUT PACKAGE	Specifies step-by-step procedures by users which detail operations required to load general-purpose software and to develop and checkout new system tapes. (Line printer output format.)	DEC-15-CRZA-D	In Stock--$5.00 (Order from Program Library)
5	PDP-15/20 USERS GUIDE-ADVANCED MONITOR SOFTWARE SYSTEM	Quick reference summary of operating procedures, excluding those for SGEN and 8TRAN, for Advanced Monitor System (V5A).	DEC-15-MG2B-D	In Stock--$4.50
6	PDP-15/20/30/40 ADVANCED MONITOR SOFTWARE SYSTEM	Operating and application details of Advanced Monitor System (V5A) to allow user to prepare, compile, assemble, debug and operate his programs; includes system language applications.	DEC-15-MR2B-D	In Stock--$5.00
7	PDP-15/20/30/40 ADVANCED MONITOR SYSTEM CHECKOUT PACKAGE	Specifies step-by-step procedures by users which detail operations required to load general-purpose software and to develop and checkout new system tapes. (Line printer output format.)	DEC-15-CGFA-D	In Stock--$5.00 (Order from Program Library)

TECHNICAL REFERENCE LITERATURE--SOFTWARE (Cont)

REF. NO.	TITLE	DESCRIPTION	ORDER NUMBER	STATUS--PRICE
8	PDP-15/30 & 15/40 BACKGROUND/FOREGROUND MONITOR SOFTWARE SYSTEM	System program language, preparation and application for on-line data acquisition and control (Foreground) and off-line program development and data reduction (Background) in DECtape (15/30) – or Disk (15/40) – oriented environment.	DEC-15-MR3A-D DEC-15-MR3A-DN1 (Erata Sheet) DEC-15-MR3A-DN2 (Change Notice)	In Stock--$6.50
9	PDP-15/30/40 BACKGROUND/FOREGROUND MONITOR SYSTEM CHECKOUT PACKAGE	Specifies step-by-step procedures by user which detail operations required to load general-purpose software and to develop and checkout new system tapes. (Line printer output format.)	DEC-15-QRZA-D	In Stock--$5.00 (Order from Program Library)
10	MACRO-15 ASSEMBLER	Macro assembly language for PDP-15 Pass 1 symbol location assigned, table constructed Pass 2 final object program Pass 3 listing of all symbols, where defined and each program line where used.	DEC-15-AMZC-D (Basic)	In Stock--$3.00 Revision
11	PDP-15 FORTRAN IV	FORTRAN IV language and compiler system for 8K PDP-15 Systems; consists of: Part 1 Basic Language Part 2 Object Time System Part 3 Science Library	DEC-15-KFZB-D (Basic) DEC-15-KFZB-DN (Update)	In Stock--$2.50
12	FOCAL PROGRAMMING MANUAL	Describes structure and use of FOCAL language, demonstration programs and advanced user-library storage, retrieval functions and user defined FOCAL functions.	DEC-15-KJZB-D	In Stock--$3.00
13	STATPAC USER'S GUIDE	FORTRAN-coded program used to perform statistical analysis on user-supplied data.	DEC-15-UFZB-D	In Stock--N/C
14	8TRAN-PDP-8 to PDP-15 TRANSLATOR	Special purpose relocatable program which assists in translating (converting) PDP-8 programs for operation on PDP-15. Describes three versions for operating in PDP-15 MONITOR or COMPACT software environments.	DEC-15-ENZA-D	In Stock--$1.60
16	DDT-dynamic debugging technique utility program	Debugging language, commands and operations.	DEC-15-YWZA-DN1	In Stock--$2.20
17	CHAIN & EXECUTE UTILITY PROGRAM	Allows user to generate system of core overlays.	DEC-15-YWZB-DN2	In Stock--$3.00
18	SGEN UTILITY PROGRAM (ADVANCED MONITOR ONLY)	Allows user under Advanced Monitor control, to re-configure PDP-15 software to generate a resident software system unique to his requirement and installation.	DEC-15-YWZA-DN3	In Stock--$2.00
19	MTDUMP UTILITY PROGRAM	Allows user to output, on any available listing device, specified core locations stored on a bulk storage device.	DEC-15-YWZB-DN4	In Stock--$1.50

TECHNICAL REFERENCE LITERATURE--SOFTWARE (Cont)

REF. NO.	TITLE	DESCRIPTION	ORDER NUMBER	STATUS--PRICE
20	PATCH UTILITY PROGRAM	Allows user to examine and modify system programs, binary format, on DECtape or Disk.	DEC-15-YWZB-DN5	In Stock--$3.50
21	EDIT UTILITY PROGRAM	Allows user to modify and create symbolic source programs and other ASCII text material; most frequently used to modify MACRO and FORTRAN IV source programs. Will replace EDIT section of basic manual.	DEC-15-YWZB-DN6	In Stock--$3.50
22	UPDATE UTILITY PROGRAM	Allows user to examine and update the contents of binary library files stored on a file-oriented media.	DEC-15-YWZB-DN7	In Stock--$2.00
23	LINKING LOADER UTILITY PROGRAM	Allows user to load and link relocatable or absolute binary program units as produced by the FORTRAN IV Compiler and the MACRO Assembler.	DEC-15-YWZB-DN8	In Stock--$2.00
24	PIP-ADVANCED MONITOR-UTILITY PROGRAM	Allows user to transfer data files from one standard peripheral device to another; operates under Advanced Monitor control using Monitor I/O device handlers. Will replace PIP section of basic manual.	DEC-15-YWZB-DN9	In Stock--$1.75
25	SRCCOM UTILITY PROGRAM	Allows user to compare, under Advanced Monitor control, any two symbolic (IOPS ASCII) programs and indicates differences between the compared programs. Will replace SRCCOM section of basic manual.	DEC-15-YWZB-DN11	In Stock--$2.00
26	SGEN UTILITY PROGRAM (DOS)	Allows user, under Disk Based Monitor control, to re-configure PDP-15 software to generate a resident software system unique to his requirements and installation. (New manual, not revision.)	DEC-15-YWZB-DN12	In Stock--$3.00
27	PIP UTILITY PROGRAM (DOS)	Allows user to transfer data files from one standard peripheral device to another; operates under Disk Based Monitor control. (New manual, not revision.)	DEC-15-YWZB-DN13	In Stock--$5.00
28	VP15A GRAPHICS SOFTWARE	Describes VP15A Display Systems programs to compile display commands, define display elements and direct linking, displaying of the elements and operation of VP15 Display controller, VT01 Storage Tube interfaced to a PDP-15 or PDP-9 computer.	DEC-15-UXSB-D	In Stock--$1.00
29	GRAPHIC-15 PROGRAMMING MANUAL	Describes VT15 Graphics Software programs to compile display commands, define display elements and direct linking, displaying and deleting of the elements and operation of VT15 Graphics Processor, VT04 Display Console interfaced with a PDP-15 computer.	DEC-15-ZFSB-D	In Stock--$3.50

TECHNICAL REFERENCE LITERATURE--SOFTWARE(Cont)

REF. NO.	TITLE	DESCRIPTION	ORDER NUMBER	STATUS--PRICE
30	HYBRID SOFTWARE FOR THE PDP-15/AD-4 SYSTEM	Functional descriptions and user instructions for general control, mode control, analog data and discrete data linkage routines to link a PDP-15 through hybrid interface to an Applied Dynamics AD-4 analog computer, includes supporting subroutines, FORTRAN IV function sub-programs and flow charts. Functional description of interactive block Continuous System Modeling Program (CSMP) with language descriptions, argument and statement configurations and user instructions.	DEC-15-GR2B-D	In Stock--$9.00
31	HYBRID SOFTWARE FOR THE PDP-15/EAI-680 SYSTEM	Functional descriptions and user instructions for linkage routines, test program and function generation routines to link a PDP-15 through hybrid interface to Electronic Associates EAI-680 analog system. Functional description of interactive block Continuous System Modeling Program (CSMP) with language descriptions, argument and statement configurations and user instructions.	DEC-15-GR3A-D	In Stock--$8.00
32	RSX15 REAL-TIME EXECUTIVE REFERENCE MANUAL	Complete information for program preparation, compilation, assembly, debugging and operation of a real-time monitor system.	DEC-15-GRQA-D	In Stock--$8.00
33	RSX-PLUS REAL-TIME EXECUTIVE REFERENCE MANUAL	Complete information for program preparation, compilation, assembly, debugging and operation of the RSX-PLUS system.	DEC-15-IRSXA-A-D	$10.00
34	RSX-PLUS PROCUREMENT BUILDING AND CHECK-OUT MANUAL	Instructions for configuring, building and checking the RSX-PLUS system.	DEC-15-IREXA-A-D	$5.00
35	MUMPS PROGRAMMING LANGUAGE	Provides user with information to acquire working knowledge of MUMPS language for user-oriented, general-purpose programs integrated into interactive time-sharing system.	DEC-15-GXZB-D	In Stock--$4.00
36	MUMPS-15 OPERATORS GUIDE	Use programming and application of the MUMPS operating system.	DEC-15-MMUPA-A-D DEC-15-MMOSA-A-D (Supplement)	$10.00
37	SCOLDS PROGRAMMING REFERENCE MANUAL	Description of a programming language and routines for spark chamber applications.	DEC-PH-UWZA-D	In Stock--$1.75
38	DOS MONITOR USER'S MANUAL	Provides all programming and operating information necessary for user to create and execute his programs in a Disk Based Monitor environment. Its scope is limited, however, to a "black-box" approach.	DEC-15-MRDA-D	In Stock--$10.00

TECHNICAL REFERENCE LITERATURE--SOFTWARE (Cont)

REF. NO.	TITLE	DESCRIPTION	ORDER NUMBER	STATUS--PRICE
39	DOS SYSTEM MANUAL	Provides detailed technical information on internal and external operations of the Disk Based Monitor and its routines. Its scope is directed to systems level programmer who may want to create and incorporate changes in standard software.	DEC-15-NRDA-D	In Stock--$10.00
40	BOSS-15 BATCH REFERENCE MANUAL	Description of batch operating software system (BOSS) language, operating procedures, procedural library (tables) and interaction with DOS Monitor system; provides information for user to modify procedural library for own requirements.	DEC-15-GUDA-D	In Stock--$8.50
41	PDP-15 COMMERCIAL SUBROUTINE PACKAGE	Describes Commercial Subroutines that interface with FORTRAN IV Operating System and allow user to program business applications in FORTRAN IV.	DEC-15-FZ1A-D DEC-15-FZ1A-UC Package 1 DEC-15-FZ2A-D DEC-15-FZ2A-UC Package 2 (order from Program Library)	$5.00 $35.00 $5.00 $35.00
42	DR. PHILLIP BEVINGTON'S "DATA REDUCTION AND ERROR ANALYSIS FOR THE PHYSICAL SCIENCES"	Description of software routines developed by Dr. Philip R. Benington for the reduction of data and error analysis commonly used in the Physical Sciences.	DEC-15-ZFTA-D	$6.00
43	PDP-15 MATH PACKAGE USERS GUIDE (USED IN CONJUNCTION WITH DEC-15-ZFTA-D)	Operation and use of routines described in DEC-15-ZFTA-D.	DEC-15-DFTA-DL	N/C
44	RASP-15 MULTIPRO-GRAMMING LANGUAGE MANUAL	Description operation and use of RASP-15 (Real Time Aids for Scientific Programming) RAST-15 is used in conjunction with RSX-PLUS (and RSX-PLUS) III only.	DEC-15-KGZA-D	$7.50
45 through 57 Reserved				
58	FORTRAN IV LANGUAGE MANUAL	This manual describes the syntax and use of the new FORTRAN compiler offered for users having a 16K PDP-15 system.	DEC-15-GFWA-D	In Stock--$5.00
59	FORTRAN IV OPERATING ENVIRONMENT	This manual describes the system software and hardware features which support the new FORTRAN compiler (16K). Descriptions of the OTS and Science Library are included.	DEC-15-GFZA-D	In Stock--$6.00
60	DOS KEYBOARD COMMAND GUIDE	A pocket-sized summary of the monitor and utility program keyboard commands. Provides a complete reference source for the keyboard commands permitted the DOS user.	DEC-15-NGKA-D	Scheduled for 2nd quarter (Estimated price--$5.00)

TECHNICAL REFERENCE LITERATURE--SOFTWARE (Cont)

REF. NO.	TITLE	DESCRIPTION	ORDER NUMBER	STATUS--PRICE
61	DOS SYSTEM REFERENCE CARD	An accordian-folded, pocket-sized card containing a tabular summary of monitor keyboard commands, system MACROS, IOPS errors and .SCOM Table entries. This is a duplicate of the tear-out card distributed with the DOS Users Manual.	DEC-15-NRZA-D	In Stock--$0.50
62	BOS-15 SYSTEM REFERENCE CARD	An accordian-folded, pocket-sized card containing a tabularized summary of BOS-15 system commands and other information useful to the BATCH system user.	DEC-15-NRXB-D	In Stock--$0.50
63	A Guide to ALGOL PROGRAMMING	Text book by D.D. McCraken; distributed by permission of the publisher.	DEC-15-GAZA-D	In Stock
64	ALGOL USERS MANUAL	Describes syntactical difference between ALGOL, as implemented on PDP-15, and ALGOL-60 as described in "A Guide to ALGOL Programming" listed above. Operating System and relation to Advanced and DOS monitors is also discussed.	DEC-15-GARA-D DEC-15-GARA-DN (Change Notice)	In Stock--$5.00
65	PDP-15 SYSTEMS REFERENCE MANUAL	Overview of hardware and software systems, internal options, addressing, instruction sets, system expansion capabilities, peripheral equipment and installation planning data.	DEC-15-BRZC-D	In Stock--$2.50
66	USER'S HANDBOOK VOLUME 1	User's guide to basic processor and options; includes system, subsystems and option features, functional descriptions, machine language programming considerations, instruction repertoire and system expansion data.	DEC-15-H2DC-D Volume 1	In Stock--$15.00
67	USER'S HANDBOOK VOLUME 2	User's guide to peripheral devices; includes device features, functional descriptions and programming considerations; covers: LP15 Line Printer TC15 DECtape Controller PC15 High Speed Paper Tape Punch LT15 Single-Teletype Control LT19 Multi-Station Teletype Control RF15 DECdisk Controller	DEC-15-H2DC-D Volume 2	In Stock--$10.00
68	OPERATOR'S GUIDE	Information and step-by-step instructions for operating the PDP-15 systems and associated peripheral devices.	DEC-15-H2CB-D	In Stock--$5.00
69	INTERFACE MANUAL	Provides parameters required to interface command, data and status information between all PDP-15 systems and their associated peripheral devices.	DEC-15-HOAC-D	In Stock--$5.00
70	INSTALLATION MANUAL	Provides data for all phases of site planning, predelivery preparation, installation, checkout and final acceptance. Specifies both DEC and customer responsibilities to ensure successful installation.	DEC-15-H2AB-D	In Stock--$4.00

TECHNICAL REFERENCE LITERATURE--SOFTWARE (Cont)

REF. NO.	TITLE	DESCRIPTION	ORDER NUMBER	STATUS--PRICE
71	MODULE MANUAL	Functional description, specifications and detailed characteristics of modules used in PDP-15 processor and options, including schematics and component lay-out (where available).	DEC-15-H2EB-D	In Stock--$15.00
72	PDP-15 CUSTOMER ACCEPTANCE PROCEDURE	Describes various acceptance forms, their intent and distribution to customers and DEC; specified test programs to be run for basic PDP-15 and standard options.	DEC-15-CUSTOMER ACCEPTANCE PRO-CEDURE	Available from Program Library only.
73	PDP-15 MAINTENANCE MANUAL, VOLUME 1	Provides functional descriptions of basic processor and prewired options; discusses instructions and associated logic circuits; prescribes preventive and corrective maintenance procedures to module-replacement level.	DEC-15-H2BB-D Volume 1	In Stock--$25.00

Note: All maintenance manuals, volume 1 (basic manual) are 8-1/2 × 11 inch format.
All maintenance manuals, volume 2 (engineering drawings) are 11 × 17 inch format.

*Because of the block and flow diagrams required, Volume 1 of the Floating Point Maintenance Manual will be on an 11 × 17 inch format.

TECHNICAL REFERENCE LITERATURE--HARDWARE

REF. NO.	TITLE	DESCRIPTION	ORDER NUMBER	STATUS--PRICE
74	PDP-15 MAINTENANCE MANUAL, VOLUME 2	Contains complete engineering drawings for basic processor and prewired options with index and parts lists; includes signal indexes and mnemonic glossary.	DEC-15-H2BB-D Volume 2	In Stock--$15.00
75	GRAPHIC 15 REFERENCE MANUAL	Overall system description, programming methods, applications and uses of the basic Graphic-15 processor and its interfacing with the PDP-15 computer.	DEC-15-GWSB-D	In Stock--$3.00
76	VT15 GRAPHIC PROCESSOR MAINTENANCE MANUAL, VOLUME 1	General description, installation block diagram, and maintenance data.	DEC-15-H2JB-D Volume 1	In Stock--$15.00
77	VT15 GRAPHIC PROCESSOR MAINTENANCE MANUAL, VOLUME 2	Engineering drawing set for VT15.	DEC-15-H2JB-D Volume 2	In Stock--$10.00
78	VT04 GRAPHIC DISPLAY CONSOLE MAINTENANCE MANUAL, VOLUME 1	Functional description and maintenance restricted to VT04. Includes user guide information pertaining to VT04 use with graphic processor other than VT15.	DEC-15-H2GB-D Volume 1	In Stock--$8.00
79	VT04 GRAPHIC DISPLAY CONSOLE MAINTENANCE MANUAL, VOLUME 2	Engineering drawing set for VT04.	DEC-15-H2GA-D Volume 2	In Stock--$4.00
80	VW01 SONIC DIGITIZER MAINTENANCE MANUAL, VOLUME 1	Writing tablet description and applications, operating and programming instructions, installation and maintenance procedures.	DEC-00-H4AA-D Volume 1	In Stock--$
81	VW01 SONIC DIGITIZER MAINTENANCE MANUAL, VOLUME 2	Engineering drawing set for VW01.	DEC-00-H4AA-D Volume 2	In Stock--$

TECHNICAL REFERENCE LITERATURE--HARDWARE (Cont)

REF. NO.	TITLE	DESCRIPTION	ORDER NUMBER	STATUS--PRICE
82	VT05 ALPHANUMERIC DISPLAY TERMINAL REFERENCE MANUAL	Overall system description, programming, applications and interfacing to the PDP-15.	DEC-00-H4AB-D	In Stock--$
83	VT05 ALPHANUMERIC DISPLAY TERMINAL MAINTENANCE MANUAL, VOLUME 1	System description and applications, operating and programming instructions, installation, and maintenance procedures.	DEC-00-H4BA-D Volume 1	Scheduled for 2nd quarter
84	VT05 ALPHANUMERIC DISPLAY TERMINAL MAINTENANCE MANUAL, VOLUME 2	Engineering drawing set for VT05.	DEC-00-H4BA-D Volume 2	Scheduled with Volume 1
85	RF15/RS09 DECdisk SYSTEM MAINTENANCE MANUAL, VOLUME 1	System description, recording format, block diagram description, instructions timing and logic, maintenance and installation.	DEC-15-H2IC-D Volume 1	In Stock--$15.00
86	RF15/RS09 DECdisk SYSTEM MAINTENANCE MANUAL, VOLUME 2	Module and engineering drawing set for RF15/RS09.	DEC-15-H2IB-D Volume 2	In Stock--$10.00
87	RP15 DISK PACK CONTROL MAINTENANCE MANUAL, VOLUME 1	System description, recording format, block diagram description, instructions timing and logic maintenance, installation.	DEC-15-H3GB-D Volume 1	In Stock--$25.00
88	RP15 DISK PACK CONTROL MAINTENANCE MANUAL, VOLUME 2	Module and engineering drawing set.	DEC-15-H3GB-D Volume 2	In Stock--$15.00
89	RK11-E DISK DRIVE CONTROL MANUAL	System description, block diagram, drawing set of RK11-E disk drive control system.	DEC-11-HRKDA-A-D	$10.00
90	RK05 DISK DRIVE MAINTENANCE MANUAL	Mechanical drawings, logic diagrams, and installation of the RK05 disk drive.	DEC-00-RK05-DB	
91	TC15 DECtape CONTROL MAINTENANCE MANUAL, VOLUME 1	Equipment description, operating and programming instructions, installation and maintenance procedures and engineering drawing set.	DEC-15-H2KB-D Volume 1	In Stock--$15.00
92	TC15 DECtape CONTROL MAINTENANCE MANUAL, VOLUME 2	Engineering Drawing Set	DEC-15-H2KB-D Volume 2	In Stock--$15.00
93	TC59 MAGNETIC TAPE CONTROL INSTRUCTION MANUAL, VOLUME 1	Equipment description, operating and programming instructions, installation and maintenance procedures.	DEC-9A-I3BC-D Volume 1	In Stock--$10.00
94	TC59 MAGNETIC TAPE CONTROL INSTRUCTION MANUAL, VOLUME 2	Engineering drawing set for TC59.	DEC-9A-I3BC-D Volume 2	In Stock--$4.00
95	MX15 MEMORY MULTIPLEXER MAINTENANCE MANUAL, VOLUME 1	System description and applications, operating and programming instructions, installation and maintenance procedures.	DEC-15-H2XB-D Volume 1	In Stock--$3.00

TECHNICAL REFERENCE LITERATURE--HARDWARE (Cont)

REF. NO.	TITLE	DESCRIPTION	ORDER NUMBER	STATUS--PRICE
96	MX15 MEMORY MULTIPLEXER MAINTENANCE MANUAL, VOLUME 2	Engineering Drawing Set	DEC-15-H2XB-D Volume 2	In Stock
97	AD15 ANALOG SUBSYSTEM MAINTENANCE MANUAL, VOLUME 1	System description, operating and programming instructions, installation and maintenance procedures.	DEC-15-H3AB-D Volume 1	In Stock--$3.00
98	AD15 ANALOG SUBSYSTEM MAINTENANCE MANUAL, VOLUME 2	Engineering Drawing Set	DEC-15-H3AB-D Volume 2	In Stock--$3.00
99	LP15C LINE PRINTER MAINTENANCE MANUAL	System description, operating and programming instructions, installation and maintenance procedures and engineering drawing set.	DEC-15-H2LA-D	In Stock--$3.00
100	ME15 CORE MEMORY MAINTENANCE MANUAL	Module and engineering drawing set, system description and block diagrams of ME15 memory.	DEC-15-HMEMA-A-D	$10.00
101	LP15F LINE PRINTER MAINTENANCE MANUAL	System description, operating and programming instructions, installation and maintenance procedures and engineering drawing set.	DEC-15-H1LB-D	In Stock--$3.00
102	UC15 UNICHANNEL 15 SYSTEM MAINTENANCE MANUAL	Block diagrams, programming, logic description and installation of the Unichannel 15 System.	DEC-15-HUCMA-B-D	$10.00
103	FP15 FLOATING POINT PROCESSOR REFERENCE MANUAL	Describes floating point arithmetic, address formats, instruction set, FP15 processor operation; includes programming samples.	DEC-15-HQEA-D	In Stock--$5.00
104	FP15 FLOATING POINT PROCESSOR MAINTENANCE MANUAL, VOLUME 1*	Provides functional description of floating point processor, interfacing to PDP-15, flow diagram analysis of instruction execution; special FP15 modules; installation and maintenance procedures.	DEC-15-HQFA-D Volume 1	In Stock--$15.00
105	FP15 FLOATING POINT PROCESSOR MAINTENANCE MANUAL, VOLUME 2	Engineering drawing set for FP15.	DEC-15-HQFA-D Volume 2	In Stock--$15.00
106	INDUSTRIAL CONTROL SYSTEM MAINTENANCE MANUAL, VOLUME 1	Covers the BD15, AFC15 and UDC15 units for industrial applications, provides programming information and IOT instructions, describes applications, options and organization of each unit, special system and functional I/O modules; includes installation and maintenance procedures.	DEC-15-H2QA-D	In Stock--$5.00
107	INDUSTRIAL CONTROL SYSTEM MAINTENANCE MANUAL, VOLUME 2	Module and engineering drawing set for Industrial Control System.	DEC-15-H2QA-D Volume 2	In Stock--$5.00

TECHNICAL REFERENCE LITERATURE--HARDWARE (Cont)

REF. NO.	TITLE	DESCRIPTION	ORDER NUMBER	STATUS--PRICE
108	CR03B CARD READER MAINTENANCE MANUAL	System description, operating and programming instructions, installation and maintenance procedures and engineering drawing set.	DEC-09-H2CA-D	In Stock--$10.00
109	DP09A BIT-SYNCHRONOUS DATA COMMUNICATION CHANNEL	System description, operating and programming instructions, data flow and timing charts, installation and maintenance procedures, engineering drawings and signal glossary.	DEC-09-H8AA-D	In Stock--$15.00
110	DB88, 98, 99 INTERPROCESSOR BUFFER SYSTEM MAINTENANCE	Describes three interprocessor buffer systems, associated equipments and interfacing with other systems*; includes operating and programming instructions, installation and maintenance procedures and engineering drawing set. * DB88A Interfaces a PDP-8 with a PDP-8 DB98A Interfaces a PDP-9/15 with a PDP-8 DB99A Interfaces a PDP-9/15 with a PDP-9/15	DEC-89-HOAA-D	In Stock--$25.00
111	PDP-9 INTERPROCESSOR DATA BUFFER DB09A MAINTENANCE MANUAL	Describes data buffer section of interprocessor buffer system which interfaces two PDP-9/15's (DB99A) or PDP-9/15 to PDP-8 (DB98A); includes operating and programming instructions, control block and logic diagrams; installation and maintenance procedures, module component location figures, repair parts lists and engineering drawing set.	DEC-09-H9PA-D	In Stock--$15.00
112	PDP-8 DB08A INTERPROCESSOR DATA BUFFER MAINTENANCE MANUAL	Describes data buffer section of interprocessor buffer system which interfaces a PDP-9/15 to PDP-8 (DB98A) or two PDP-8's (DB88A); includes operating and programming instructions, control and logic diagrams, installation and maintenance procedures, module component location figures, repair parts lists and engineering drawing set.	DEC-09-H8PH-D	In Stock--$15.00
113	350C INCREMENTAL PLOTTER MAINTENANCE MANUAL	System description, operating and programming instructions, installation and maintenance procedures and engineering drawing set.	DEC-09-I6CA-D	In Stock--$10.00
114	PDP-11/05 COMPUTER MANUAL	PDP-11/05 Computer Handbook	DEC-11-H05AA-B-D	

Note: All maintenance manuals, volume 1 (basic manual) are 8-1/2 X 11 inch format.
All maintenance manuals, volume 2 (engineering drawings) are 11 X 17 inch format.

*Because of the block and flow diagrams required, Volume 1 of the Floating Point Maintenance Manual will be on an 11 X 17 inch format.

PDP-15 and PDP-15/76
SYSTEM REFERENCE MANUAL
DEC-15-XSRMA-A-D

Reader's Comments

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

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

_____

_____

_____

What features are most useful? _____

_____

_____

_____

What faults do you find with the manual? _____

_____

_____

_____

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

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

_____

_____

_____

Would you please indicate any factual errors you have found. _____

_____

_____

_____

Please describe your position. _____

Name _____ Organization _____

Street _____ Department _____

City _____ State _____ Zip or Country _____