Memory and
Addressing

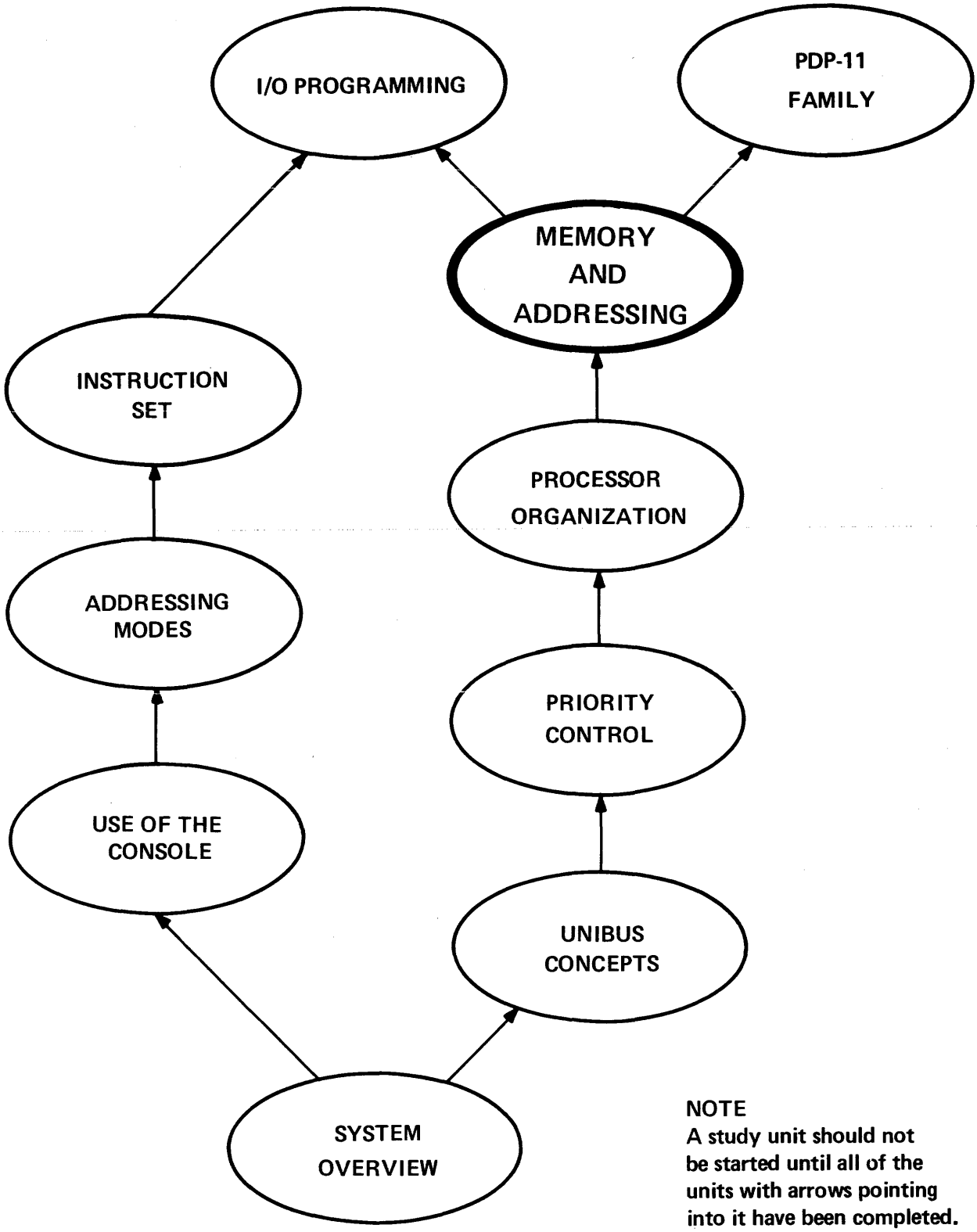
student workbook
introduction to
the pdp11

digital equipment corporation · maynard. massachusetts

1st Printing, March 1974
2nd Printing (Rev), November 1974
3rd Printing (Rev), April 1977

Printed in U.S.A.

course map



NOTE
A study unit should not be started until all of the units with arrows pointing into it have been completed.

read on

READ
LEARNING
OBJECTIVES
(page 1)

NOW RUN FILM
CARTRIDGES A & B

REVIEW
MATERIAL
(pages 3 – 20)

TAKE TEST &
CHECK RESULTS
(pages 21 – 25)

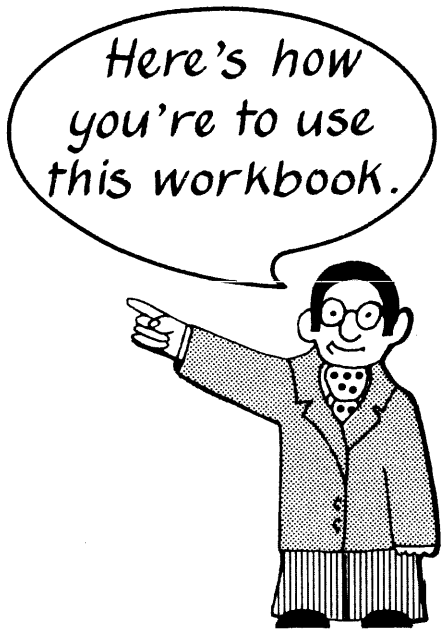
ANY
INCORRECT
ANSWERS

YES

PLEASE REVIEW THE
MATERIAL YOU'RE
HAVING DIFFICULTY WITH.
(ADDITIONAL RESOURCES ARE
LISTED ON PAGE 2.)

NO

GOOD WORK!
NOW GO ON TO THE
NEXT STUDY UNIT.



read on

objectives

After completing this study unit you should be able to . . . .

- ★ Explain how PDP-11 memory is organized into words and bytes. Point out which types of bus cycles are used to store or retrieve words and/or bytes.

- ★ Use octal notation to represent any word, low byte, or high byte.

- ★ Describe the address structure of the PDP-11 including:

  - Significance of odd and even addresses
  - Addresses specified for memory
  - Addresses specified for I/O device registers and CPU registers
  - Memory limits for both 16-bit and 18-bit addresses

- ★ Explain why all register addresses are always in the top 4K of the total address space and describe how the highest octal digit of a 16-bit address is converted from 1 to 7.

- ★ Describe the basic memory map giving the purpose and location of:

  - Trap vectors
  - Interrupt vectors
  - Hardware stack
  - System and Utility Programs

- ★ Show how a hardware stack is built, where its boundary (limit) is located, and what constitutes yellow and red zone stack violations.

- ★ Describe in detail each of the four memory operations including related signals:

  - DATI
  - DATIP
  - DATO
  - DATOB

- ★ Explain core memory address decoding, interleaving, and parity.

- ★ Describe the two types of semiconductor memories (MOS and bipolar) including their prime differences.

- ★ Describe the characteristics of read-only memory (ROM).

- ★ Describe some typical pre-programmed ROM memories, showing how they are used in the PDP-11 system.

additional resources

- **PDP-11/04/05/10/35/40/45 Processor Handbook**

  Read Chapter 2, Paragraph 2.3 (Memory Organization).

- **PDP-11 Peripherals Handbook**

  Refer to Chapter 4 and read the following:

  BM792 Read-Only Memory, pages 4-25 and 4-26.

  BM873 Restart/Loader, pages 4-27 through 4-30.

- **PDP-11 Peripherals Handbook**

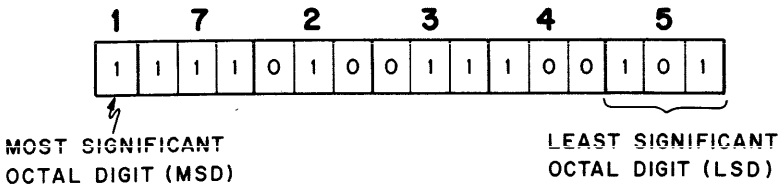
  Appendix A lists the Unibus addresses reserved for interrupt vectors, trap vectors, and device registers.

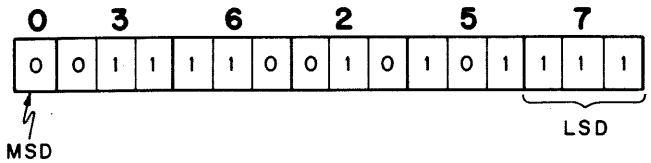
The following material is covered in this study unit:

Topic	*Key Points*	*Visual Ref.*
BASIC CONCEPTS	★ The PDP-11 memory is organized into a series of storage locations each of which holds 8 bits (one byte) of data.	1–9
word structure	• Words are made up of pairs of odd and even-numbered bytes.	
high and low bytes	• Odd-numbered bytes are called "high" bytes; even-numbered bytes "low."	
word handling	• When storing or receiving words, only even addresses are used.	
byte handling	• Odd addresses select only high bytes; even addresses select low bytes.	
addressing	• Memory locations can be addressed by the CPU or by any DMA device.	
slave function	• The memory is a passive device. It can never function as a master; it is always a slave.	
WORD FORMAT	★ The PDP-11 word consists of 16 bits numbered 0 through 15.	12
	• Bit 15 is the most significant bit (MSB) • Bit 0 is the least significant bit (LSB)	
octal numbers	★ To simplify notation, a 6-digit octal number is used to represent the 16-bit word.	13–16
	• Bits 0, 1, and 2 form the least significant octal digit.	
	• Bits 3 through 14 are similarly grouped to form octal digits 2 through 5.	
	• The sixth, most significant, octal digit is formed by a *single* bit. . . .bit 15.	

read on ➧

Topic	*Key Points*	*Visual Ref.*
examples	★ The following examples show octal representation of the 16-bit PDP-11 word.	17



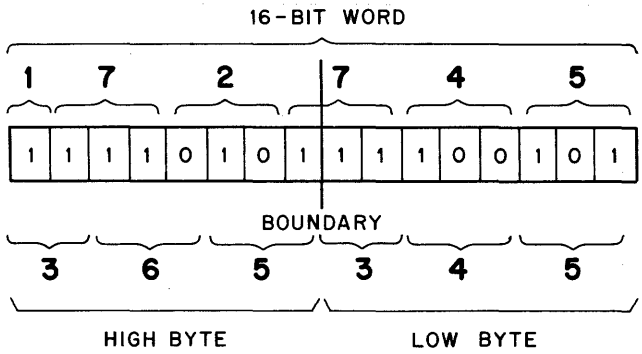


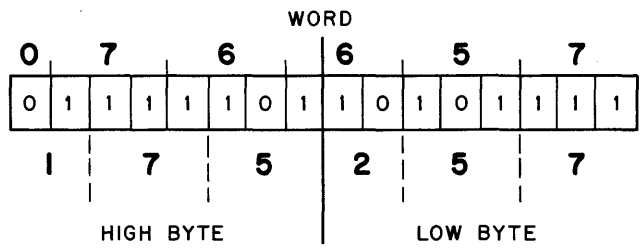
word vs byte	★ Each 16-bit word actually consists of two 8-bit bytes. • Bits 0–7 represent the *low* byte. • Bits 8–15 represent the *high* byte. • Bytes can also be represented in octal notation.	18

read on

Topic	*Key Points*	*Visual Ref.*
byte boundary	★ The byte boundary in a PDP-11 word falls between bits 7 and 8. This boundary affects the octal value of a low or high byte.	19–22

- The *third* octal digit of a low byte is different from the third digit of the word because the byte does not use bit 8.
- All *three* octal digits of a high byte are different from the word because the bit pattern is effectively shifted one bit to the right.
- The following are some examples of word vs byte octal representation:





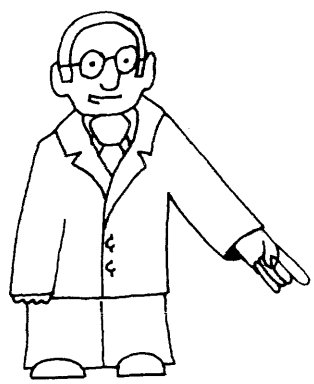
read on

Topic	*Key Points*	*Visual Ref.*
ADDRESS STRUCTURE	★ A unique address is assigned to each memory location. The address is specified by a 6-digit octal number.	23
addressing	• High bytes are designated by addresses ending in an *odd* number.	24–26
	• Low bytes are designated by addresses ending in an *even* number.	
	• Words are also designated by addresses ending in an *even* number.	
	• Consecutive word locations are always addressed in steps of two (0, 2, 4, etc.)	
storing words or bytes	• A DATO and an even address are used to store full 16-bit *words.*	27
	• A DATOB and an *even* address are used to store a *low* byte.	
	• A DATOB and an *odd* address are used to store a *high* byte.	
18-bit addresses	• The standard PDP-11 address has 16 bits. However, when memory management hardware is installed in a PDP-11 system, the address is expanded to 18-bits.	28
	• When an 18-bit address is used, the most significant octal digit is formed by bits 15, 16 and 17.	
ADDRESS ASSIGNMENTS	★ The PDP-11 has only *one* set of addresses. This single address set is used for: • Memory locations, • I/O device registers, • CPU registers	29

read on ➧

review material

Topic	*Key Points*	*Visual Ref.*
16-bit address	★ The standard 16-bit address provides 32K *word* addresses (64K *byte* addresses). • Up to 28K of these word addresses can be used for *memory* locations. • The remaining 4K of word addresses are used for I/O device registers and CPU registers such as the PSW register and the GPRs.	30
18-bit address	★ The optional 18-bit address provides 128K *word* addresses (256K *byte* addresses). • Up to 124K of these word addresses can be used for *memory* locations. • The remaining 4K of word addresses are still used for I/O device registers and CPU registers.	31
memory limits: 16-bit addr.	★ Begins at 0 and increases to address 157 777. This accommodates 28K words or 56K bytes of memory.	32
memory limits: 18-bit addr.	★ Begins at 0 and increases to address 757 777. This accommodates 124K words or 248K bytes of memory.	33



read on

Topic	*Key Points*	*Visual Ref.*
register addresses	★ 16-bit addresses that *exceed* 157 777, are not used for memory; they are reserved for I/O device registers and CPU registers.	34
	★ Whenever one of these registers is addressed, the MSD in the address is forced to a 7 by CPU hardware.	35
	• The CPU monitors bits 13, 14, and 15.	
	• If these three bits are all 1s, it indicates a *register* address because it means that the two most significant bits must be either 16 or 17. In other words, an address beyond 157 777.	
	• The CPU places 1s in bits 16 and 17 of the bus address, thereby converting (or forcing) the most significant address digit from 1 to 7.	
	★ This conversion places all *register* addresses into the *last* (or highest) 4K of the total address space.	36, 37
	★ This conversion results in a gap between the highest *memory* address and the *register* addresses. Therefore, at a later time, we can use the expanded 18-bit address without having to modify programs containing register addresses.	

read on

Topic	*Key Points*	*Visual Ref.*
MEMORY MAPPING	★ A typical PDP-11 memory has specific areas reserved for storage of certain types of information. Defining these areas is usually called "memory mapping." • One area is reserved for *trap* vectors. • Another area is reserved for *interrupt* vectors. • A third area is reserved for the *hardware stack.* • The remainder of memory is available for both the user's programs as well as for system and utility programs.	40
trap vectors	★ Trap vectors are stored in locations 000 to 037. Certain error conditions cause the CPU to automatically go to a predetermined memory location and retrieve the *trap vector.* • The trap vector directs the CPU to an error handling routine stored in memory. • The CPU executes the error routine and then returns to the main program. • Each trap vector occupies *two* word locations. • The first word location is a starting address which is loaded into the PC to direct the CPU to the first instruction in the error routine. • The second word location contains a new PSW to establish initial conditions for the error routine. • A list of trap vectors is given in Appendix A of the PDP-11 Peripherals Handbook.	41–45

read on

Topic	*Key Points*	*Visual Ref.*
interrupt vectors	★ Interrupt vectors are stored in memory locations 040 through 337. When a device interrupts the CPU in order to obtain service, the CPU goes to the indicated location and retrieves the *interrupt vector.* • The interrupt vector directs the CPU to a service routine in memory. • The CPU executes the service routine and then returns to the main program. • Each interrupt vector occupies *two* word locations. • The first word location is a starting address which is loaded into the PC to direct the CPU to the first instruction in the service routine. • The second word location contains a new PSW to establish the initial conditions required for the service routine. • A complete list of interrupt vectors is given in Appendix A of the PDP-11 Peripherals Handbook. In general, these vector assignments are: 040 – 057 system software 060 – 077 basic I/O devices 100 – 266 other I/O devices remainder communications devices	46–49

read on

Topic	*Key Points*	*Visual Ref.*
hardware stack	★ A portion of the memory is reserved for the hardware stack which is used when the CPU services interrupts and traps. ★ The stack allows the CPU to store its current PC and PSW values before going off to service an interrupt or trap. ★ It is not necessary to keep track of the latest stack entry. This is handled automatically by the stack pointer (SP) which always points to the last entry stored in the stack. ★ Stack boundaries must be considered to prevent the stack from moving into areas of memory reserved for other items. • The stack is started at the highest possible address and builds toward location 400 which is the *stack limit.* • In some PDP-11 systems there are "yellow" and "red" zone spaces. • The yellow zone provides a *warning.* It is placed between locations 337 and 400, and is an overflow space so that operations causing a stack overflow can be completed before the program traps to location 004. • If the stack goes *beyond* its yellow zone, it enters the *red zone* which is a fatal stack error. The operation causing the violation is aborted. • When a *red* zone violation occurs, the old PC and PSW are stored in locations 0 and 2. A new PC and PSW are taken from locations 4 and 6 and a hardware trap is executed.	50–59

read on

Topic	*Key Points*	*Visual Ref.*
user programs	★ The remaining portion of memory (with the exception of the last 4K) is reserved for user programs.	60
utility programs	★ System software and utility programs require at least 4K of storage and are normally loaded into the top 4K of memory so they do not interfere with the user's programs. ★ Paper tape software programs available to the customer are described in the PDP-11 Paper Tape Software Programming Handbook.	
MEMORY OPERATIONS	★ Before a master device can use the memory, it must first select a memory location, and then specify the *type of transfer* to be performed. ★ One of four types of data transfers may be selected. Bus control lines C0 and C1 specify the type of transfer: DATI, DATIP, DATO, or DATOB.	63, 64
DATI operation	★ A DATI is used to retrieve data *from* memory. ★ The master specifies a DATI by issuing MSYN and clearing the C1 and C0 lines. ★ The addressed memory location responds as follows: • Locates the addressed word and reads the word from its storage location. • Stores the word temporarily in its buffer register. • Places word from buffer onto the bus and issues SSYN to inform the master that data is available.	65–69

read on ➧

Topic	*Key Points*	*Visual Ref.*
	• Performs a *restore* cycle by writing the original word back into the memory location. This is necessary because core memory is a destructive readout device; i.e., when data is read, the contents of the location goes to all 0's.	
DATIP operation	★ A DATIP is used to retrieve data from memory when data can be *discarded* after being read. In other words, it is not necessary to *restore* data. The master may take a word from memory, increment it, and then store the *new* value in the original location.	70–75
	★ The master uses the C1 and C0 lines to specify a DATIP (C1=0, C0=1) . It also issues an address and a MSYN signal.	
	★ The memory responds to a DATIP as follows:	
	• Locates the addressed word and reads it from the storage location.	
	• Stores the word temporarily in its buffer register.	
	• Places the word from buffer onto the bus and issues SSYN to inform the master that data is available.	
	• Up to this point, the operation has been identical to a DATI. However, this completes the DATIP operation. There is *no restore* cycle. The cleared memory location is left at all 0's.	
DATI versus DATIP operation	★ A DATI bus operation involves *two* memory cycles: *read* data and *restore* data.	76
	★ A DATIP bus operation involves only *one* cycle: read data. Therefore, total core memory cycle time for a DATIP is *half* that required for a DATI.	

read on ➧

Topic	*Key Points*	*Visual Ref.*
DATO operation	★ A DATO is used to *store* (or write) a *full word* into memory. ★ The master device uses the C1 and C0 lines to specify a DATO (C1=1, C0=0). It issues an address and MSYN as before; however, the *data* to be stored is also placed on the bus at this time. ★ The memory responds to a DATO as follows: • Memory first performs a *read* cycle to clear the specified location. • Memory then performs a *write* cycle by taking data from the bus and loading it into the buffer register. It then issues SSYN to inform the master that memory now has the word. • The memory then completes the DATO by storing the data word in the storage location specified by the master.	77–80
DATO operations for semiconductor memory systems	★ Core memories require both a *read* cycle (to clear the location) and a *write* cycle (to store the new word). ★ Semiconductor memories do not have to be cleared prior to storing new data. Because of this, there is no read cycle. When storing new data, the memory simply performs a *write* operation.	81
DATOB operation	★ A DATOB is used to store a *byte,* rather than a word, in memory. (A *low* byte is stored at the even-numbered address; a *high* byte is stored at the odd-numbered address.) ★ Operation of a DATOB is identical to that of a DATO except a byte, rather than a word, is transferred to memory. Remember that the address is *even* for a *low* byte; *odd* for a *high* byte.	82–84

read on

Topic	*Key Points*	*Visual Ref.*
AVAILABLE MEMORIES	★ All PDP-11 memories fall into one of three categories: ferrite *core* memories, *semiconductor* memories, and *read-only* memories (called ROMs). ★ Because of the Unibus concept, memories with different operating speeds and characteristics may be used in the same PDP-11 system.	87–88
core memory	★ Destructive read-out devices. The memory state is switched to zero when read; therefore, a read cycle must be followed by a *restore* cycle to return the core to its original state. ★ Core memory is packaged in individual units (memory banks) that contain their own read-write circuits and address select logic. These units are available in different sizes. Common memory sizes are 8K, 16K, 32K, and 64K.	89–91
addressing (8K banks)	★ Address bits 15 and 14 select the memory bank. • Words within the selected 8K bank are chosen by address bits 13–01. • Address bit 0 permits selection of a low or high byte.	92
addressing (16K banks)	★ With the expanded 18-bit address, bits 15, 16, and 17 designate the desired 16K memory bank. • Words within the 16K bank are selected by address bits 14-01. • Address bit 0 selects a low byte or high byte during DATOB transfers.	93

read on ➧

Topic	*Key Points*	*Visual Ref.*
address decoding	★ Each memory bank has a hard-wired device decoder. ★ When a memory address is placed on the bus, one of the memory banks recognizes its own address and then decodes the remainder of the address in order to locate the specific word or byte location.	94
interleaving (4K banks)	★ Interleaving is used to alternate memory addresses between *two* memory banks. In other words, the first bank would recognize addresses 0, 4, 10; the second bank would recognize 2, 6, and 12. • Interleaving permits overlapping of read and restore cycles, thereby reducing program execution times. • 8K memory banks are interleaved by swapping address bits 1 and 14. • An interleaved memory is transparent to the master device; the master still addresses memory as if it were a single, continuous bank of addresses. • If we are using 8K banks of memory, we can only interleave in consecutive 16K segments. Therefore, if a system has 24K of memory, only the first 16K segment could be interleaved. • If the system has 32K of core memory (four 8K banks), the first 16K segment can be interleaved and the second 16K segment can also be interleaved.	95–100

read on ➧

Topic	*Key Points*	*Visual Ref.*
parity	★ Parity generation and detection logic is included on some PDP-11 memories. • The use of parity expands the *memory word* size from 16 to 18 bits. • Bit 17 is parity for the *high* byte; Bit 16 is parity for the *low* byte. • Parity is generated during a *write* cycle (DATO or DATOB). • Parity is checked during a *read* cycle (DATI or DATIP). • If a parity error is detected, the master device is notified.	102–103
semiconductor memories	★ Semiconductor memories are *non-destructive read out* devices. Therefore, no restore cycle is needed.	105
types	★ Two types of semiconductor memories are available: metal-oxide semiconductor (MOS) and bipolar.	109

read on

Topic	*Key Points*	*Visual Ref.*
MOS memory	• Consist of 1-bit semiconductor cells that are similar to small capacitors. • Each cell is charged to a certain potential when writing and must be *recharged,* or *refreshed,* periodically. • *Precharging,* which is performed in addition to refreshing, is performed prior to reading to ensure that the correct data is readout.	110
bipolar memory	• Consist of semiconductor *flip-flop* cells arranged in a matrix. • Memory cells function in the same manner as conventional flip-flops.	111
comparisons: MOS vs bipolar	★ MOS memory *needs refreshing*; bipolar does not. ★ MOS memory is *slower* than bipolar (450–750 ns versus 300 ns). ★ MOS memory costs less than bipolar memory. ★ MOS is *smaller* and consumes 1/10th the power of bipolar. ★ Neither MOS nor bipolar will retain data if system power is lost.	112–116

read on

Topic	*Key Points*	*Visual Ref.*
read-only memories (ROM)	★ There are two distinct characteristics of *all* ROM memories: • Data is stored permanently; it is not erased or destroyed if power is removed from the computer system. • The CPU can read information out of a ROM, but cannot write information into the ROM.	122
types of ROMs	★ In some PDP-11 systems, read-only memory is an integral part of the CPU. For example, the 11/04 and 11/34 central processors contain a ROM. Diagnostic routines and bootstrap loader programs are permanently stored in this ROM. ★ Another type of ROM is packaged as a separate unit that connects to the PDP-11 Unibus. • This ROM only responds to DATIP transfers. It cannot respond to DATO, DATOB, or DATI cycles because they involve a write operation. • This ROM employs diodes for storage; one diode equals one bit of data. • Information is stored by cutting out specific diodes. When a diode is removed, it means that bit position is read as a binary 0; if the diode is left in, it reads as a binary 1. • Each ROM holds 32 words of data and up to 8 ROMs can be used in a system.	122a–127
addressing diode ROMs	★ Address bits 6, 7, and 8 select one of the eight ROM memories. • Address bits 1–5 select a specific word in the 32-word ROM memory. • Only words, not bytes, can be addressed. Therefore, address bit 0 is always 0.	128–130

read on ➧

Topic	*Key Points*	*Visual Ref.*
pre-programmed ROM memories	★ Pre-programmed ROMs are available for use with PDP-11 systems. • These ROMs contain loading instructions which cause the CPU to automatically read in programs from input devices for loading into memory. • These pre-programmed ROMs eliminate the need for manually entering *each* program instruction using the console. • The operator simply specifies the ROM starting address and then depresses the console LOAD address and START switches to start the program running.	131–133
typical pre-programmed ROM memories	★ *Paper Tape Bootstrap loader* – when a computer is first installed, its memory does not contain any meaningful information. Before the computer can input information, it must be given the necessary instructions. The paper tape bootstrap loader is a minimal instruction program that can be stored in a ROM. This program permits loading of larger programs from a paper tape reader.	134–136
	★ *Bulk Storage Bootstrap loader* – similar to paper tape bootstrap loader except that this program permits loading of programs from bulk storage devices such as disks and tapes.	137
	★ *Card Reader Bootstrap loader* – again similar to paper tape bootstrap loader except permits loading of programs stored on punched or mark-sense cards.	138
typical ROM address assignments	★ Each ROM has a unique set of addresses. These addresses are as follows: • Paper-Tape loader 773000–773076 • Bulk Storage loader 773100–773176 • Card Reader loader 773200–773276	139

read on ➧

test—addressing & memory

When you have completed the study unit, please take this self-scoring test. Then compare your answers against the "answer sheet" which can be obtained from your supervisor. Based on your test results, either review the appropriate material in this study unit or proceed to the next unit in the series.

1 Which of the following statements are true?

(a) T – F PDP-11 memory consists of 8-bit bytes; words are formed by combining pairs of bytes.

(b) T – F Even addresses select high bytes; odd addresses select low bytes or words.

(c) T – F PDP-11 memory cannot initiate data transfers between itself and another device.

2 Below are some 16-bit words. For each word, give the octal value of the word, the high byte, and the low byte.

	Word	High Byte	Low Byte
1 111 010 011 100 101	_____	_____	_____
0 101 111 110 011 110	_____	_____	_____
1 001 011 101 111 011	_____	_____	_____

3 When storing information into memory, two items are needed: an address and a bus cycle. Give the appropriate type of address and bus cycle required to store the following.

	Address (odd or even)	Bus Cycle
full word	_____	_____
high byte	_____	_____
low byte	_____	_____

read on ➧

4 When the 18-bit memory management hardware is installed, the maximum memory size is increased to:

(a) 104K words.
(b) 128K words.
(c) 124K words.
(d) 32K words.

5 Explain the reason and the method of converting the most significant octal digit of a 16-bit address from a 1 to a 7.

(a) Reason:

(b) Method:

6 Each trap or interrupt vector actually occupies two word locations in memory. What information is stored in these two locations?

(a) First Word:

(b) Second Word:

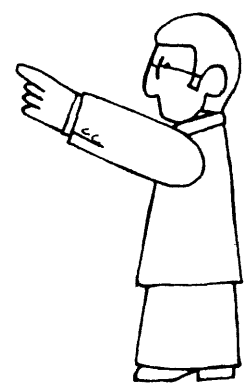
read on

7 Explain the primary function of the hardware stack and specify the difference between a "yellow" and a "red" zone violation.

(a) Function:

(b) Yellow Zone:

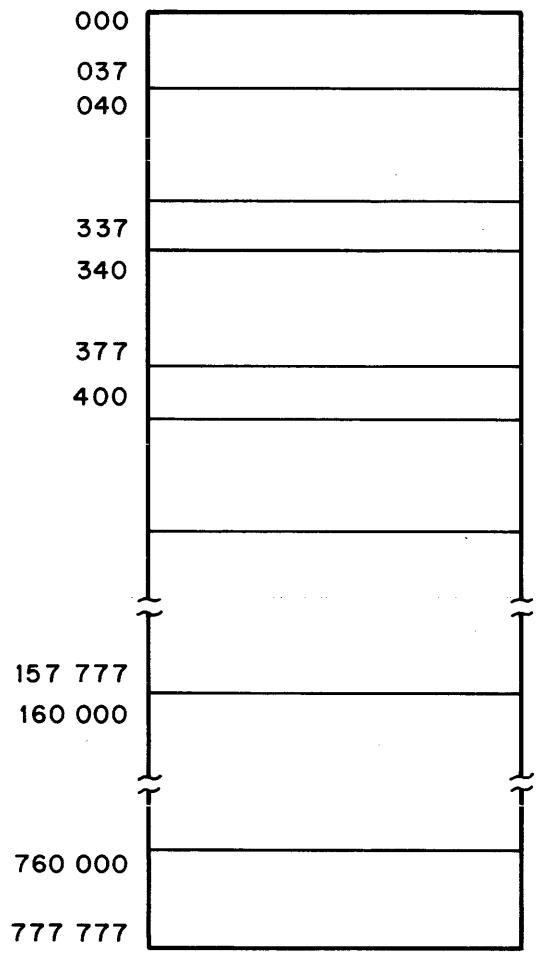
(c) Red Zone:



read on

8 Place each of these items in its appropriate address space.

(a) memory expansion
(b) yellow zone
(c) hardware stack
(d) I/O & CPU registers
(e) user programs
(f) interrupt vectors
(g) red zone
(h) trap vectors
(i) stack limit



9 Match each operation with the corresponding bus cycle or cycles (DATO, DATOB, DATI, DATIP).

(a) _____ The processor retrieves an instruction from memory.

(b) _____ The processor retrieves a number from core memory, increments the number, and stores the new value in the original memory location.

(c) _____ A disk transfers data into memory location 703.

read on

10 What is the reason for interleaving banks of core memory?

(a) Allows use of the 18-bit address option.

(b) Does away with the need for the restore operation when successive memory locations are read.

(c) Reduces program execution times by overlapping memory cycles.

(d) Allows greater core density in alternate memory banks.

11 Assume that the first and second 4K memory banks have been interleaved. Which of these addresses now select the first bank and which addresses select the second bank? What two locations are *physically* adjacent in the same bank?

1st or 2nd: 000 206

1st or 2nd: 012 640

1st or 2nd: 012 642

1st or 2nd: 016 440

1st or 2nd: 016 444

1st or 2nd: 034 262

12 Match each subject in the left column with the most appropriate statement in the right column.

a.	Bipolar Memory	( )	destructive readout
b.	Core Memory	( )	must be refreshed
c.	ROM	( )	flip-flop
d.	MOS Memory	( )	only responds to DATIPs.

notes

notes

notes