Course Contents

Unit-01: Introduction to Microprocessor

Unit-02: Intel 8085

Unit-03: Microoperations

Unit-04: Control Unit and Central Processing Unit

Unit-05: Fixed point Computer Arithmetic

Unit-06: Input and Output Organization

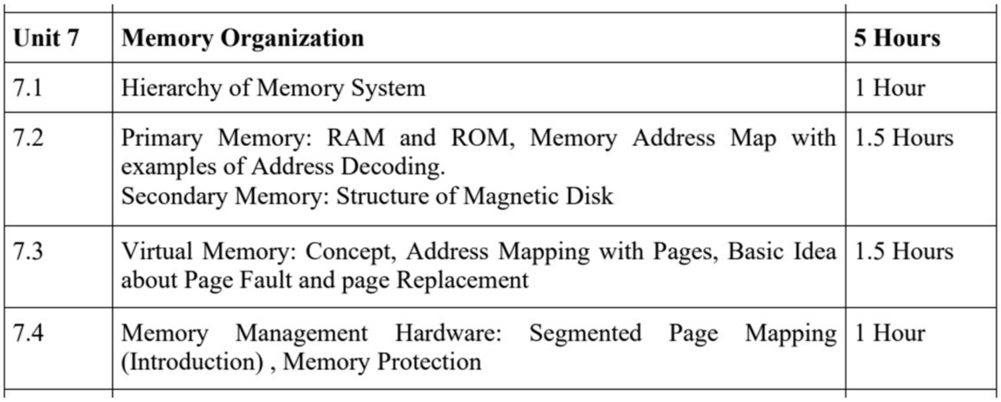
Unit-07: Memory Organization

Unit-08: Pipelining

Course Contents

Unit-07: Memory Organization

Hierarchy of Memory System, Primary and Secondary Memory, Virtual Memory, Memory Management hardware



Unit 7	**Memory Organization**	**5 Hours**
7.1	Hierarchy of Memory System	1 Hour
7.2	Primary Memory: RAM and ROM, Memory Address Map with examples of Address Decoding. Secondary Memory: Structure of Magnetic Disk	1.5 Hours
7.3	Virtual Memory: Concept, Address Mapping with Pages, Basic Idea about Page Fault and page Replacement	1.5 Hours
7.4	Memory Management Hardware: Segmented Page Mapping (Introduction) , Memory Protection	1 Hour

Course Contents

Unit-07: Memory Organization

7.1 Hierarchy of Memory System 1 Hour

7.2 Primary Memory: RAM and ROM, Memory Address Map with examples of Address Decoding. Secondary Memory: Structure of Magnetic Disk 1.5 Hours

7.3 Virtual Memory: Concept, Address Mapping with Pages, Basic Idea about Page Fault and page Replacement 1.5 Hours

7.4 Memory Management Hardware: Segmented Page Mapping (Introduction) , Memory Protection 1 Hour

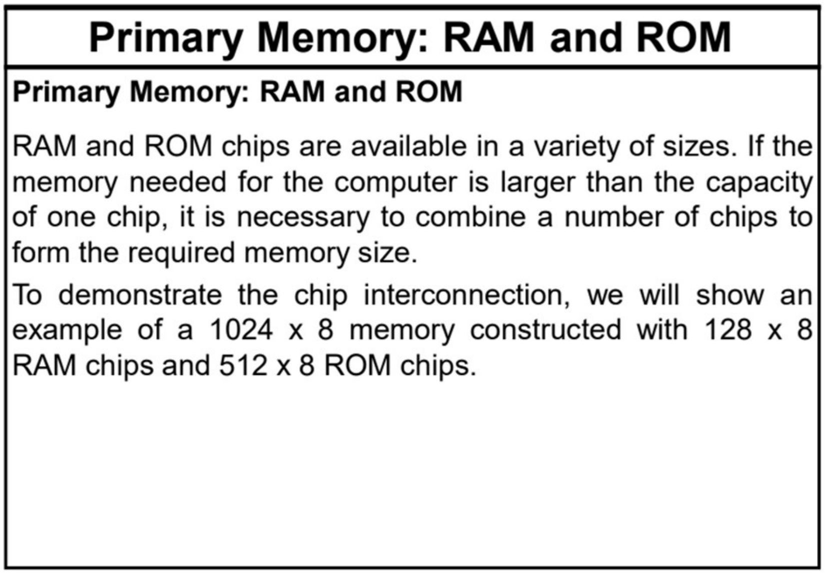
Course Contents

Unit-07: Memory Organization

1. What is Memory Hierarchy?
2. What are the storage devices(memories) employed in a computer system?
3. What are the functions of each memory types in a computer system?
4. What is Memory Management System?
5. What is memory Address map?
6. What is Virtual Memory? Explain in brief.
7. What is Mapping Table?
8. What is Paging? Explain the conversion of virtual address to physical address in paging with example.
9. Explain the terms: Page fault, Page Replacement and Memory Protection.
10. What is Segmentation? Explain.

Course Contents

Unit-07: Memory Organization



Primary Memory: RAM and ROM

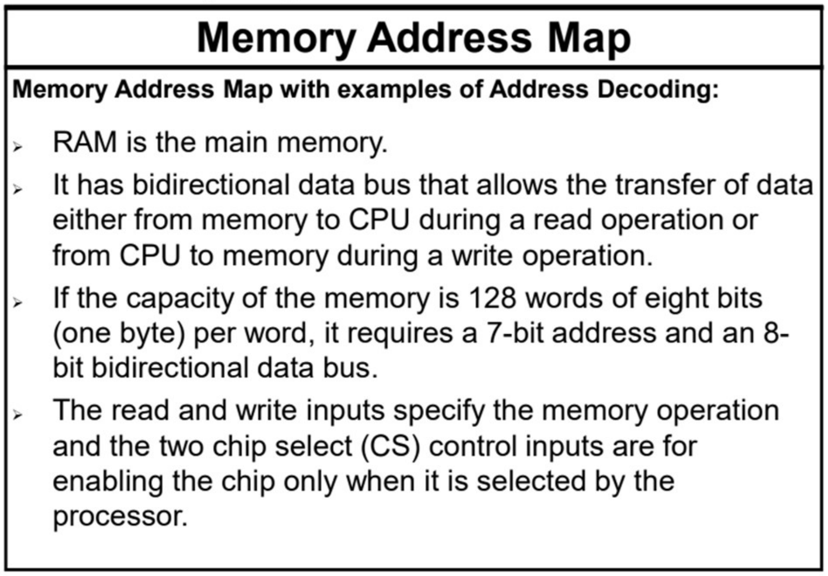
Primary Memory: RAM and ROM

RAM and ROM chips are available in a variety of sizes. If the memory needed for the computer is larger than the capacity of one chip, it is necessary to combine a number of chips to form the required memory size.

To demonstrate the chip interconnection, we will show an example of a 1024 x 8 memory constructed with 128 x 8 RAM chips and 512 x 8 ROM chips.

Course Contents

Unit-07: Memory Organization



Memory Address Map

Memory Address Map with examples of Address Decoding:

- RAM is the main memory.
- It has bidirectional data bus that allows the transfer of data either from memory to CPU during a read operation or from CPU to memory during a write operation.
- If the capacity of the memory is 128 words of eight bits (one byte) per word, it requires a 7-bit address and an 8-bit bidirectional data bus.
- The read and write inputs specify the memory operation and the two chip select (CS) control inputs are for enabling the chip only when it is selected by the processor.

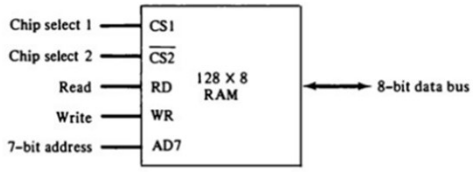
Course Contents

Unit-07: Memory Organization

Memory Address Map

RAM Chip:

Figure 12-2 Typical RAM chip.



(a) Block diagram

CS1	$\overline{CS2}$	RD	WR	Memory function	State of data bus
0	0	×	×	Inhibit	High-impedance
0	1	×	×	Inhibit	High-impedance
1	0	0	0	Inhibit	High-impedance
1	0	0	1	Write	Input data to RAM
1	0	1	×	Read	Output data from RAM
1	1	×	×	Inhibit	High-impedance

(b) Function table

Course Contents

Unit-07: Memory Organization

Memory Address Map

ROM Chip:

- ROM can only read, the data bus can only be in an output mode.
- For the same size chip, it is possible to have more bits of ROM than of RAM, because the internal binary cells in ROM occupy less space than in RAM.
- For this reason, the diagram specifies a 512 byte ROM, while the RAM has only 128 bytes.
- The nine address lines in the ROM chip specify any one of the 512 bytes stored in it.

Course Contents

Unit-07: Memory Organization

Memory Address Map

ROM Chip:

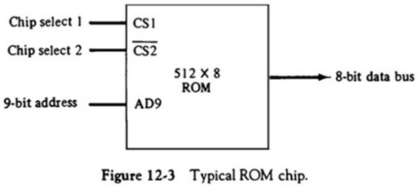


Figure 12-3 Typical ROM chip.

Memory Address Map

Memory Address Map:

- A memory address map, is a pictorial representation of assigned address space for each chip in the system.
- Let us assume that a computer system needs 512 bytes of RAM and 512 bytes of ROM
- The RAM chips have 128 bytes and need seven address lines. The ROM chip has 512 bytes and needs 9 address lines.
- The X's are always assigned to the low order bus lines: lines 1 through 7 for the RAM and lines 1 through 9 for the ROM. It is now necessary to distinguish between 4 RAM chips by assigning to each a different address.

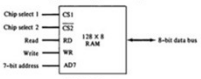
Course Contents

Unit-07: Memory Organization

Memory Address Map

Memory Address Map:

Figure 12-2 Typical RAM chip.



(a) Block diagram

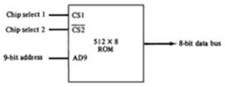
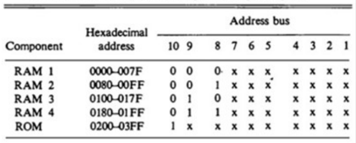


Figure 12-3 Typical ROM chip.

TABLE 12-1 Memory Address Map for Microprocomputer



Component	Hexadecimal address	Address bus									
		10	9	8	7	6	5	4	3	2	1
RAM 1	0000–007F	0	0	0	x	x	x	x	x	x	x
RAM 2	0080–00FF	0	0	1	x	x	x	x	x	x	x
RAM 3	0100–017F	0	1	0	x	x	x	x	x	x	x
RAM 4	0180–01FF	0	1	1	x	x	x	x	x	x	x
ROM	0200–03FF	1	x	x	x	x	x	x	x	x	x

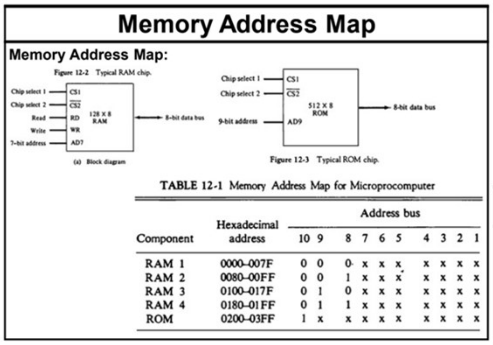
Memory Address Map

MEMORY CONNECTION TO CPU:

- The configuration gives a memory capacity of 512 bytes of RAM and 512 bytes of ROM.
- The particular RAM chip selected is determined from lines 8 and 9 in the address bus. This is done through a 2 x 4 decoder whose outputs goes to the CS1 inputs in each RAM chip.
- The selection between RAM and ROM is achieved through bus line 10.

Course Contents

Unit-07: Memory Organization



Memory Address Map

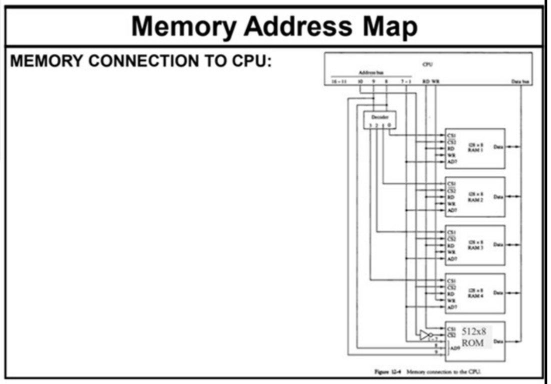
Memory Address Map:

Figure 12-2 Typical RAM chip.

Figure 12-3 Typical ROM chip.

TABLE 12-1 Memory Address Map for Microprocomputer

Component	Hexadecimal address	Address bus									
		10	9	8	7	6	5	4	3	2	1
RAM 1	0000–007F	0	0	0	x	x	x	x	x	x	x
RAM 2	0080–00FF	0	0	1	x	x	x	x	x	x	x
RAM 3	0100–017F	0	1	0	x	x	x	x	x	x	x
RAM 4	0180–01FF	0	1	1	x	x	x	x	x	x	x
ROM	0200–03FF	1	x	x	x	x	x	x	x	x	x



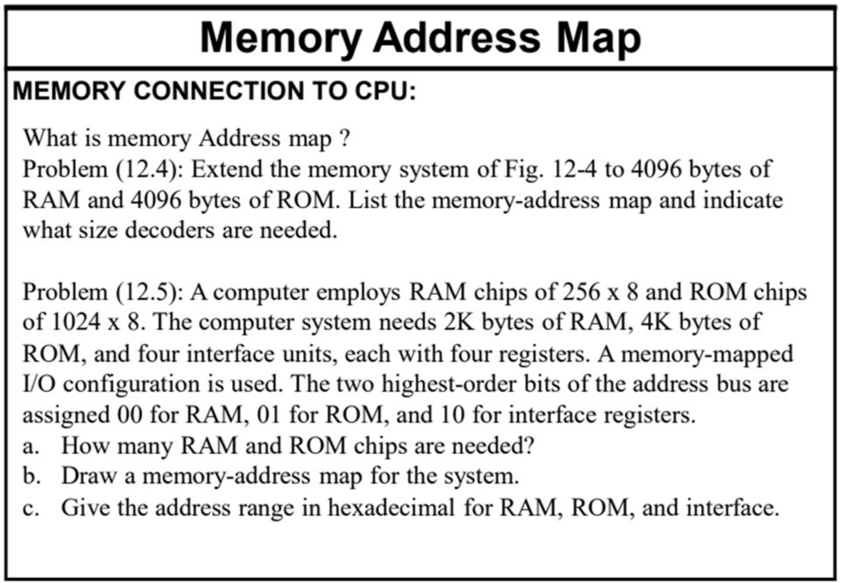
Memory Address Map

MEMORY CONNECTION TO CPU:

Figure 12-4 Memory connection to the CPU.

Course Contents

Unit-07: Memory Organization



Memory Address Map

MEMORY CONNECTION TO CPU:

What is memory Address map ?

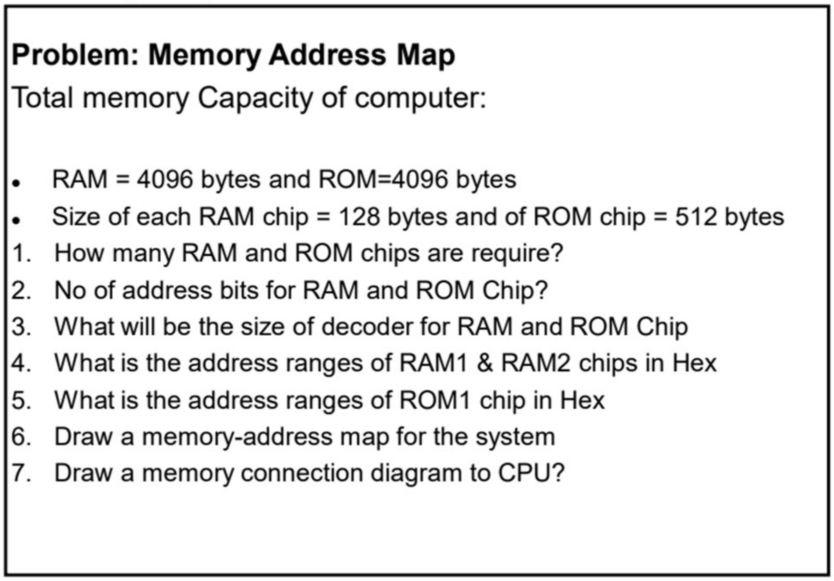
Problem (12.4): Extend the memory system of Fig. 12-4 to 4096 bytes of RAM and 4096 bytes of ROM. List the memory-address map and indicate what size decoders are needed.

Problem (12.5): A computer employs RAM chips of 256 x 8 and ROM chips of 1024 x 8. The computer system needs 2K bytes of RAM, 4K bytes of ROM, and four interface units, each with four registers. A memory-mapped I/O configuration is used. The two highest-order bits of the address bus are assigned 00 for RAM, 01 for ROM, and 10 for interface registers.

a. How many RAM and ROM chips are needed?
b. Draw a memory-address map for the system.
c. Give the address range in hexadecimal for RAM, ROM, and interface.

Course Contents

Unit-07: Memory Organization



Problem: Memory Address Map

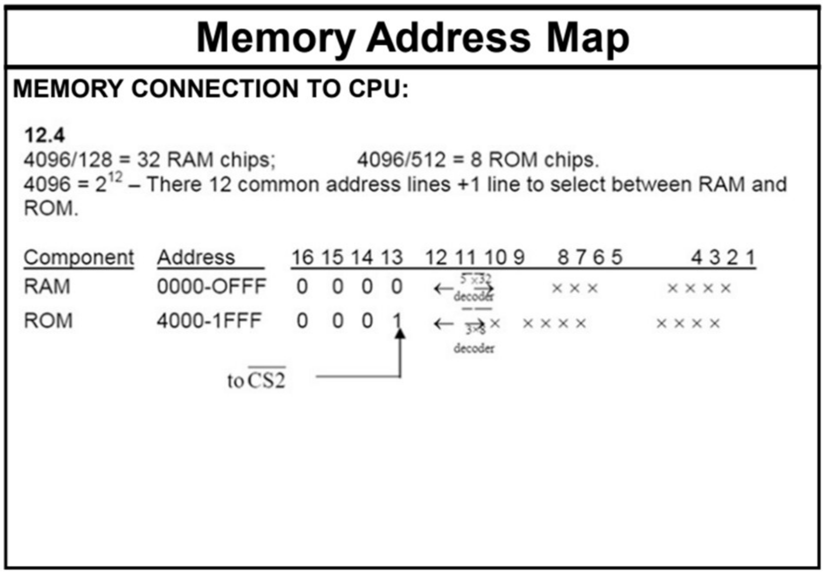
Total memory Capacity of computer:

- RAM = 4096 bytes and ROM=4096 bytes
- Size of each RAM chip = 128 bytes and of ROM chip = 512 bytes

1. How many RAM and ROM chips are require?
2. No of address bits for RAM and ROM Chip?
3. What will be the size of decoder for RAM and ROM Chip
4. What is the address ranges of RAM1 & RAM2 chips in Hex
5. What is the address ranges of ROM1 chip in Hex
6. Draw a memory-address map for the system
7. Draw a memory connection diagram to CPU?

Course Contents

Unit-07: Memory Organization



Memory Address Map

MEMORY CONNECTION TO CPU:

12.4

4096/128 = 32 RAM chips; 4096/512 = 8 ROM chips.

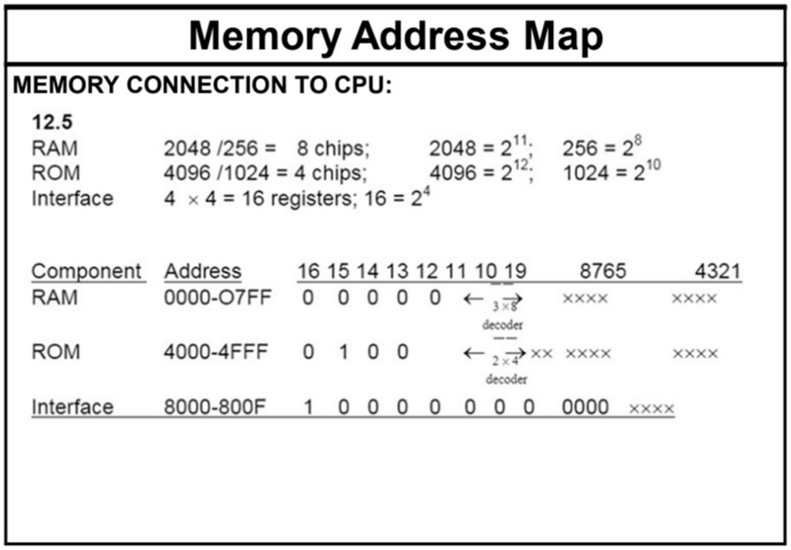
4096 = 2^{12} – There 12 common address lines +1 line to select between RAM and ROM.

Component	Address	16 15 14 13	12 11 10 9	8 7 6 5	4 3 2 1
RAM	0000-OFFF	0 0 0 0	← $\overline{5\times32}$ decoder →	× × ×	× × × ×
ROM	4000-1FFF	0 0 0 1	← $\overline{3\times8}$ decoder → ×	× × × ×	× × × ×

to $\overline{CS2}$

Course Contents

Unit-07: Memory Organization



Memory Address Map

MEMORY CONNECTION TO CPU:

12.5

RAM 2048 /256 = 8 chips; $2048 = 2^{11}$; $256 = 2^{8}$

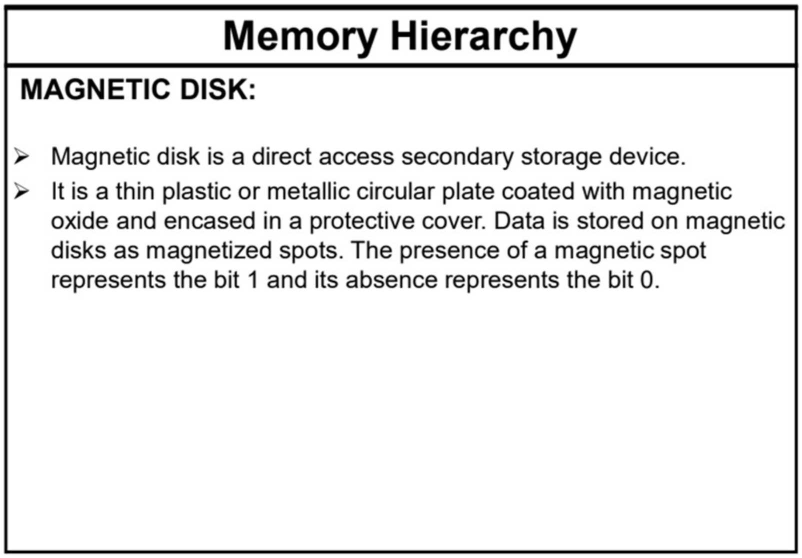
ROM 4096 /1024 = 4 chips; $4096 = 2^{12}$; $1024 = 2^{10}$

Interface 4 × 4 = 16 registers; $16 = 2^{4}$

Component	Address	16	15	14	13	12	11	10	19	8765	4321
RAM	0000-O7FF	0	0	0	0	0	← 3×8 → decoder			××××	××××
ROM	4000-4FFF	0	1	0	0		← 2×4 → decoder		××	××××	××××
Interface	8000-800F	1	0	0	0	0	0	0	0	0000	××××

Course Contents

Unit-07: Memory Organization



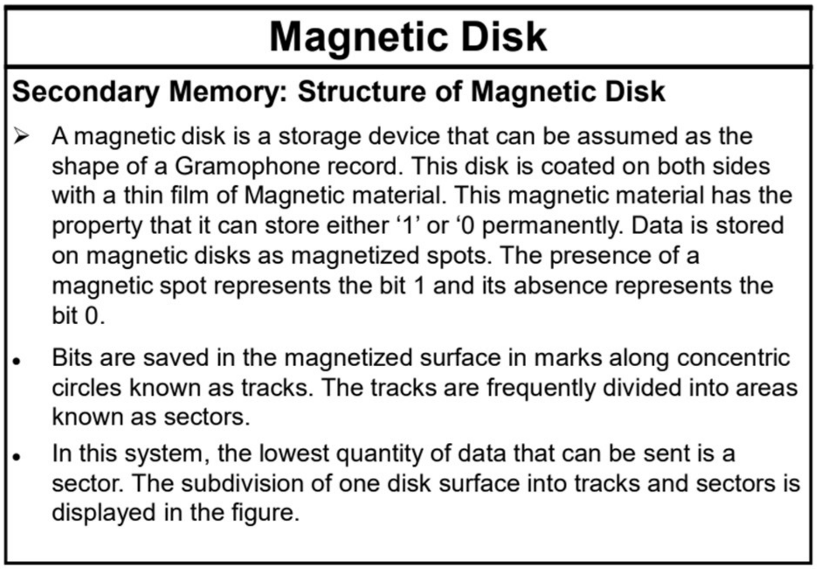
Memory Hierarchy

MAGNETIC DISK:

- Magnetic disk is a direct access secondary storage device.
- It is a thin plastic or metallic circular plate coated with magnetic oxide and encased in a protective cover. Data is stored on magnetic disks as magnetized spots. The presence of a magnetic spot represents the bit 1 and its absence represents the bit 0.

Course Contents

Unit-07: Memory Organization



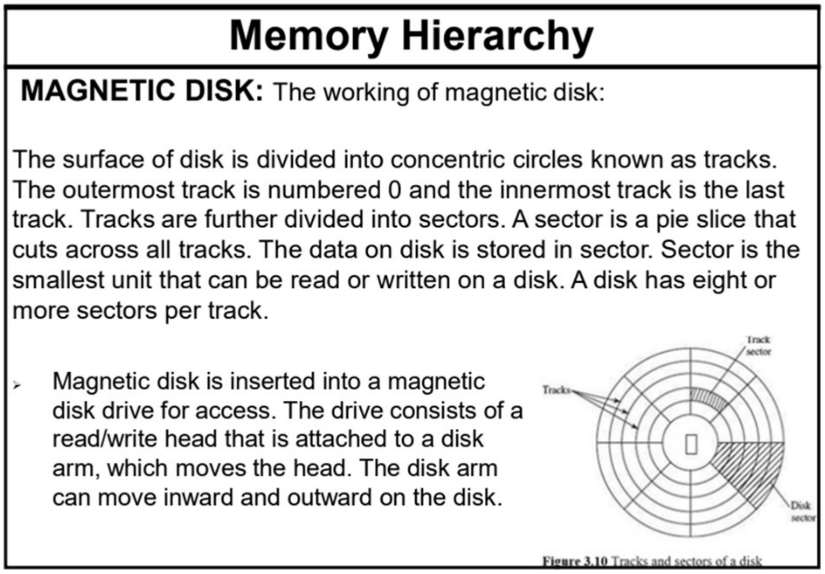
Magnetic Disk

Secondary Memory: Structure of Magnetic Disk

- A magnetic disk is a storage device that can be assumed as the shape of a Gramophone record. This disk is coated on both sides with a thin film of Magnetic material. This magnetic material has the property that it can store either ‘1’ or ‘0 permanently. Data is stored on magnetic disks as magnetized spots. The presence of a magnetic spot represents the bit 1 and its absence represents the bit 0.
- Bits are saved in the magnetized surface in marks along concentric circles known as tracks. The tracks are frequently divided into areas known as sectors.
- In this system, the lowest quantity of data that can be sent is a sector. The subdivision of one disk surface into tracks and sectors is displayed in the figure.

Course Contents

Unit-07: Memory Organization



Memory Hierarchy

MAGNETIC DISK: The working of magnetic disk:

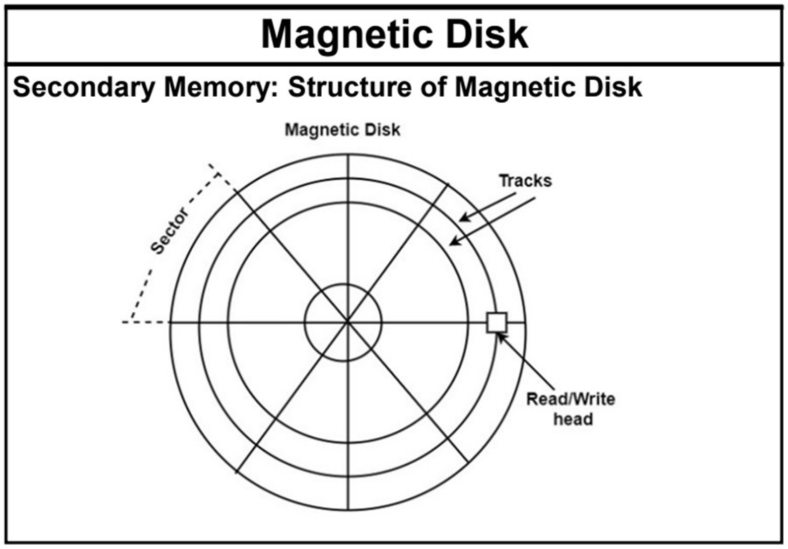
The surface of disk is divided into concentric circles known as tracks. The outermost track is numbered 0 and the innermost track is the last track. Tracks are further divided into sectors. A sector is a pie slice that cuts across all tracks. The data on disk is stored in sector. Sector is the smallest unit that can be read or written on a disk. A disk has eight or more sectors per track.

- Magnetic disk is inserted into a magnetic disk drive for access. The drive consists of a read/write head that is attached to a disk arm, which moves the head. The disk arm can move inward and outward on the disk.

Figure 3.10 Tracks and sectors of a disk

Course Contents

Unit-07: Memory Organization

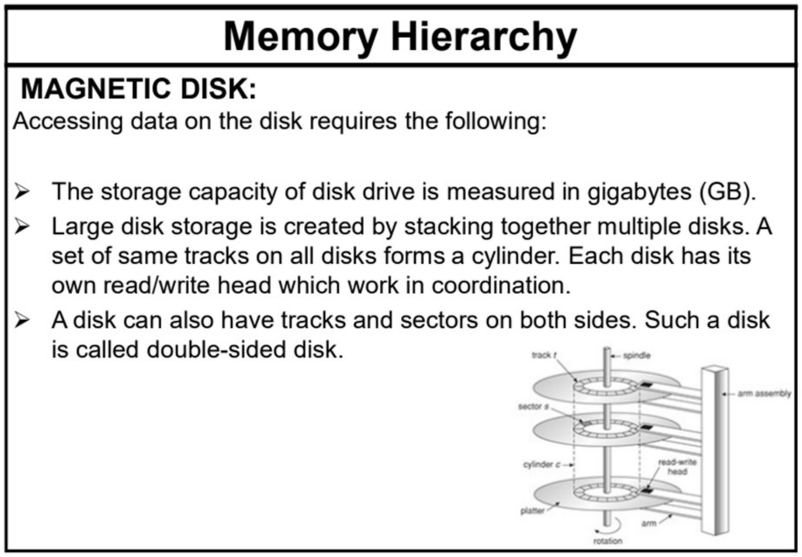


Magnetic Disk

Secondary Memory: Structure of Magnetic Disk

Course Contents

Unit-07: Memory Organization



Memory Hierarchy

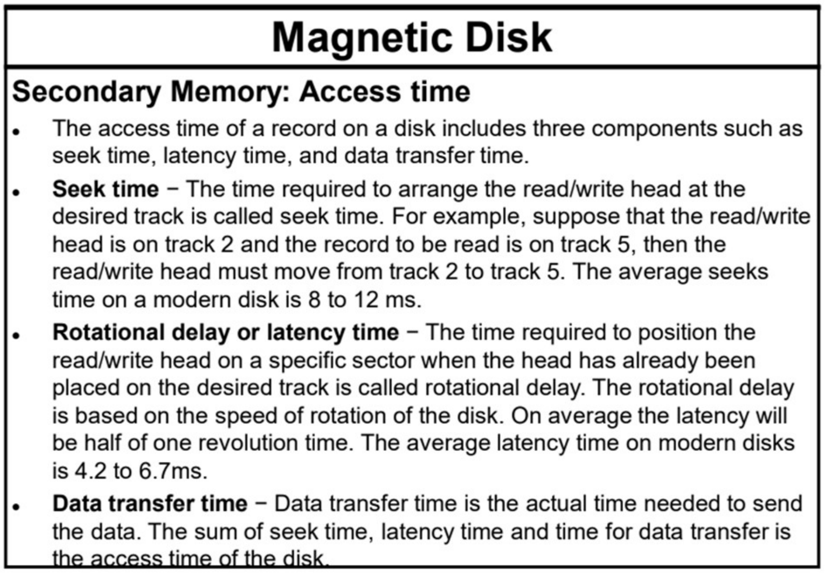
MAGNETIC DISK:

Accessing data on the disk requires the following:

- The storage capacity of disk drive is measured in gigabytes (GB).
- Large disk storage is created by stacking together multiple disks. A set of same tracks on all disks forms a cylinder. Each disk has its own read/write head which work in coordination.
- A disk can also have tracks and sectors on both sides. Such a disk is called double-sided disk.

Course Contents

Unit-07: Memory Organization



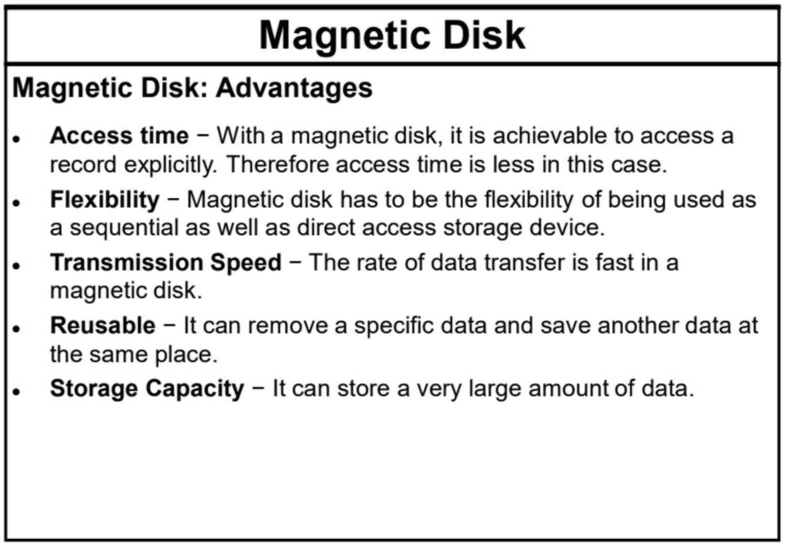
Magnetic Disk

Secondary Memory: Access time

- The access time of a record on a disk includes three components such as seek time, latency time, and data transfer time.
- **Seek time** − The time required to arrange the read/write head at the desired track is called seek time. For example, suppose that the read/write head is on track 2 and the record to be read is on track 5, then the read/write head must move from track 2 to track 5. The average seeks time on a modern disk is 8 to 12 ms.
- **Rotational delay or latency time** − The time required to position the read/write head on a specific sector when the head has already been placed on the desired track is called rotational delay. The rotational delay is based on the speed of rotation of the disk. On average the latency will be half of one revolution time. The average latency time on modern disks is 4.2 to 6.7ms.
- **Data transfer time** − Data transfer time is the actual time needed to send the data. The sum of seek time, latency time and time for data transfer is the access time of the disk.

Course Contents

Unit-07: Memory Organization



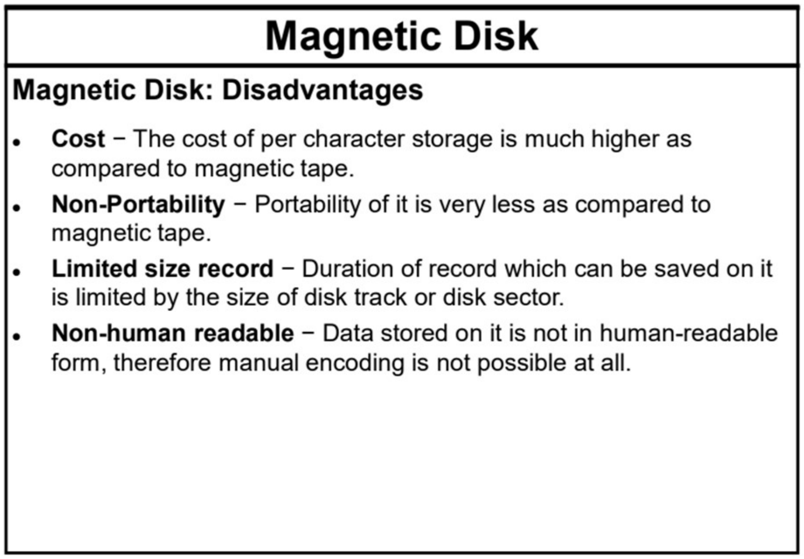
Magnetic Disk

Magnetic Disk: Advantages

- **Access time** − With a magnetic disk, it is achievable to access a record explicitly. Therefore access time is less in this case.
- **Flexibility** − Magnetic disk has to be the flexibility of being used as a sequential as well as direct access storage device.
- **Transmission Speed** − The rate of data transfer is fast in a magnetic disk.
- **Reusable** − It can remove a specific data and save another data at the same place.
- **Storage Capacity** − It can store a very large amount of data.

Course Contents

Unit-07: Memory Organization



Magnetic Disk

Magnetic Disk: Disadvantages

- **Cost** − The cost of per character storage is much higher as compared to magnetic tape.
- **Non-Portability** − Portability of it is very less as compared to magnetic tape.
- **Limited size record** − Duration of record which can be saved on it is limited by the size of disk track or disk sector.
- **Non-human readable** − Data stored on it is not in human-readable form, therefore manual encoding is not possible at all.

Course Contents

Unit-07: Memory Organization

Problem1: A computer employs 4096 bytes of RAM and 4096 bytes of ROM. The RAM chips have 128 bytes and the ROM chip has 512 bytes. List the memory-address map and indicate what size decoders are needed.

Problem2: A computer employs RAM chips of 256 x 8 and ROM chips of 1024 x 8. The computer system needs 2K bytes of RAM, 4K bytes of ROM, and four interface units, each with four registers. A memory-mapped I/O configuration is used. The two highest-order bits of the address bus are assigned 00 for RAM, 01 for ROM, and 10 for interface registers.

a. How many RAM and ROM chips are needed?
b. Draw a memory-address map for the system.
c. Give the address range in hexadecimal for RAM, ROM, and interface.

Course Contents

Unit-07: Memory Organization

Problem3: Memory Address Map
Total memory Capacity of computer:
RAM = 4096 bytes and ROM=4096 bytes
Size of each RAM chip = 128 bytes and of ROM chip = 512 bytes

1. How many RAM and ROM chips are require?
2. No of address bits for RAM and ROM Chip?
3. What will be the size of decoder for RAM and ROM Chip
4. What is the address ranges of RAM1 & RAM2 chips in Hex
5. What is the address ranges of ROM1 chip in Hex
6. Draw a memory-address map for the system
7. Draw a memory connection diagram to CPU?

Course Contents

Unit-07: Memory Organization

1. What is memory Address map?
2. What is Virtual Memory? Explain in brief.
3. What is Mapping Table?
4. What is Paging? Explain the conversion of virtual address to physical address in paging with example.
5. Explain the terms: Page fault, Page Replacement and Memory Protection.
6. What is Segmentation? Explain.

Course Contents

Unit-07: Memory Organization

Virtual Memory: Concept

Virtual Memory: Concept, Address Mapping with Pages,

Basic Idea about Page Fault and page Replacement

Memory Management Hardware: Segmented Page Mapping

(Introduction) , Memory Protection

Virtual Memory Concept

How Virtual Memory Works?

Demand Paging

Virtual memory management system

Course Contents

Unit-07: Memory Organization

Virtual Memory: Concept

- Program needs to be available in main memory for the processor to execute it.
- In a memory hierarchy system, programs and data are first stored in auxiliary memory. Portions of a program or data are brought into main memory as they are needed by the CPU.
- Amount of RAM may not enough to run all of the programs (operating system, an e-mail, a Web browser and word processor into RAM simultaneously) that most users expect to run at once.

Course Contents

Unit-07: Virtual Memory: Concept

- With virtual memory, we do not view the program as one single piece. We divide it into pieces, and only the one part that is currently being referenced by the processor need to be available in main memory. The entire program is available in the hard disk. As the copying between the hard disk and main memory happens automatically, we don't even know it is happening, and it makes your computer feel like is has unlimited RAM space.
- Therefore, most modern computers use a combination of both hardware and software to allow the computer to address more memory than the amount physically present on the system. This extra memory is actually called Virtual Memory.
- Virtual memory is a concept used in some large computer systems that permit the user to construct programs as though a large memory space were available, equal to the totality of auxiliary memory.

Course Contents

Unit-07: Virtual Memory: Concept

- Each address that is referenced by the CPU goes through an address mapping from the so-called virtual address to a physical address in main memory. Virtual memory is used to give programmers the illusion that they have a very large memory at their disposal, even though the computer actually has a relatively small main memory.
- A virtual memory system provides a mechanism for translating program-generated addresses into correct main memory locations. This is done dynamically, while programs are being executed in the CPU. The translation or mapping is handled automatically by the hardware by means of a mapping table.

Course Contents

Unit-07: Page Fault and page Replacement

Address Mapping with Pages:

Address Space and Memory Space:

1. What is Virtual address and Address Space?
2. What is Physical address and Memory Space?

Course Contents

Unit-07: Page Fault and page Replacement

Address Space and Memory Space:

- Processor reference an instruction and data space that is independent of the available physical main memory space.
- The binary addresses that the processor issues for either instructions or data are called virtual or logical addresses. These addresses are translated into physical addresses by a combination of hardware and software components.
- If a virtual address refers to a part of the program or data space that is currently in the physical memory, then the contents of the appropriate location in the main memory are accessed immediately. On the other hand, if the referenced address is not in the main memory, its contents must be brought into a suitable location in the memory before they can be used.
- Therefore, an address used by a programmer will be called a virtual address, and the set of such addresses the address space.

Course Contents

Unit-07: Page Fault and page Replacement

Address Space and Memory Space:

- An address in main memory is called a location or physical address. The set of such locations is called the memory space, which consists of the actual main memory locations directly addressable for processing.

Course Contents

Unit-07: Page Fault and page Replacement

Address Space and Memory Space:

- An address used by a programmer will be called a virtual address, and the set of such addresses the address space.
- An address in main memory is called a location or physical address. The set of such locations is called the memory space.
- Thus the address space is the set of addresses generated by programs as they reference instructions and data; the memory space consists of the actual main memory locations directly addressable for processing.
- In most computers the address and memory spaces are identical. The address space is allowed to be larger than the memory space in computers with virtual memory.

Course Contents

Unit-07: Page Fault and page Replacement

Address Space and Memory Space:

- As an illustration, consider a computer with a main-memory capacity of 32K words (K = 1024). Fifteen bits are needed to specify a physical address in memory since 32K = 2^{15} . Suppose that the computer has available auxiliary memory for storing 2^{20} = 1024K words. Thus auxiliary memory has a capacity for storing information equivalent to the capacity of 32K main memories. Denoting the address space by N and the memory space by M, we then have for this example N = 1024K and M = 32K.

Course Contents

Unit-07: Page Fault and page Replacement

Address Space and Memory Space:

What is Mapping Table?

- In a multiprogramming computer system, programs and data are transferred to and from auxiliary memory and main memory based on demands imposed by the CPU.
- Suppose that program-1 is currently being executed in the CPU.
- Program-1 and a portion of its associated data are moved from auxiliary memory into main memory.
- Portions of programs and data need not be in contiguous locations in memory since information is being moved in and out, and empty spaces may be available in scattered locations in memory.

Course Contents

Unit-07: Page Fault and page Replacement

Address Space and Memory Space:

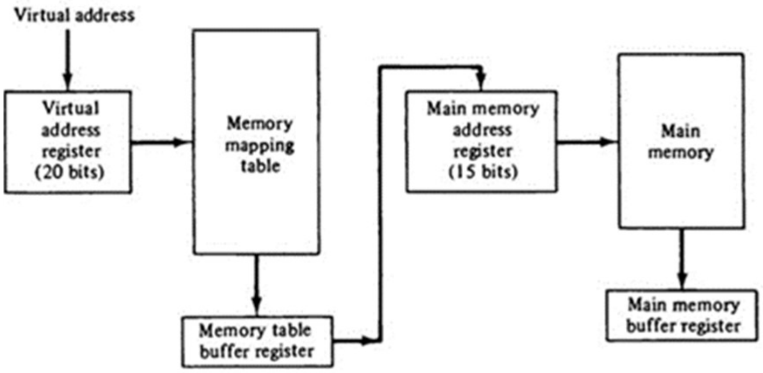
- In a virtual memory system, the address field of the instruction code has a sufficient number of bits to specify all virtual addresses. In our example, the address field of an instruction code will consist of 20 bits but physical memory addresses must be specified with only 15 bits. Thus CPU will reference instructions and data with a 20-bit address, but the information at this address must be taken from physical memory because access to auxiliary storage for individual words will be prohibitively long.
- A table is then needed to map a virtual address of 20 bits to a physical address of 15 bits. The mapping is a dynamic operation, which means that every address is translated immediately as a word is referenced by CPU which is called Mapping Table.

Course Contents

Unit-07: Page Fault and page Replacement

Address Space and Memory Space:

Figure 12-17 Memory table for mapping a virtual address.


Virtual address
Virtual address register (20 bits)
Memory mapping table
Memory table buffer register
Main memory address register (15 bits)
Main memory
Main memory buffer register

Course Contents

Unit-07: Page Fault and page Replacement

Address Space and Memory Space:

Virtual Memory can be implemented using two methods :

- Paging
- Segmentation

Page Fault and page Replacement

Unit-07: Page Fault and page Replacement

Address Mapping Using Pages:

- The table implementation of the address mapping is simplified if the information in the address space and the memory space are each divided into groups of fixed size.
- The physical memory is broken down into groups of equal size called blocks, which may range from 64 to 4096 words each. The term page refers to groups of address space of the same size.

Page Fault and page Replacement

Unit-07: Page Fault and page Replacement

Address Mapping Using Pages:

- For example, if a page or block consists of 1K words and a computer with a main-memory capacity of 32K words and auxiliary memory 1024K words. The address space is divided into 1024 pages and main memory space is divided into 32 blocks.
- Although both a page and a block are split into groups of 1K words, a page refers to the organization of address space, while a block refers to the organization of memory space.
- The programs are also considered to be split into pages. Portions of programs are moved from auxiliary memory to main memory in records equal to the size of a page. The term "page frame" is sometimes used to denote a block.

Page Fault and page Replacement

Unit-07: Page Fault and page Replacement

Address Mapping Using Pages:

- Consider a computer with an address space of 8K and a memory space of 4K. If we split each into groups of 1K words, we obtain eight pages and four blocks as shown in Fig. 12-18. At any given time, up to four pages of address space may reside in main memory in any one of the four blocks.

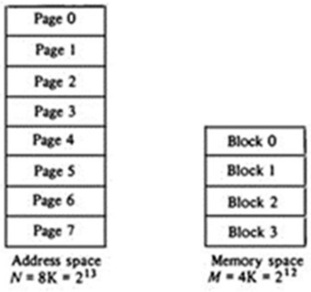


Figure 12-18 Address space and memory space split into groups of 1K words.

Page Fault and page Replacement

Unit-07: Page Fault and page Replacement

Address Mapping Using Pages:

- The mapping from address space to memory space is facilitated if each virtual address is considered to be represented by two numbers: a page number address and a line within the page.
- In a computer with 2^p words per page, p bits are used to specify a line address and the remaining high-order bits of the virtual address specify the page number.
- In the example of Fig. 12-18, a virtual address has 13 bits. Since each page consists of $2^{10} = 1024$ words, the high order three bits of a virtual address will specify one of the eight pages and the low-order 10 bits give the line address within the page.
- Note that the line address in address space and memory space is the same; the only mapping required is from a page number to a block number.

Page Fault and page Replacement

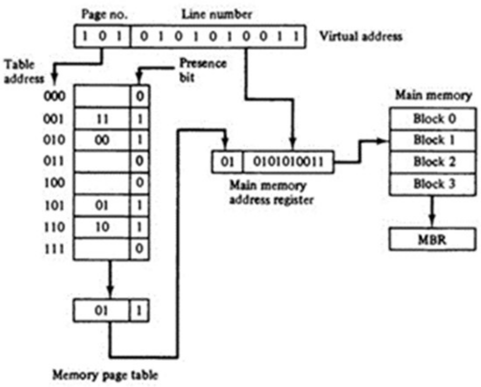
Unit-07: Page Fault and page Replacement

Address Mapping Using Pages:

How Virtual Address to Physical Memory address?

- ➢ The memory-page table consists of eight words, one for each page. The address in the page table denotes the page number and the content of the word gives the block number where that page is stored in main memory.
- ➢ The table shows that pages 1, 2, 5, and 6 are now available in main memory in blocks 3, 0, 1, and 2, respectively.
- ➢ A presence bit in each location indicates whether the page has been transferred from auxiliary memory into main memory.
- ➢ A 0 in the presence bit indicates that this page is not available in main memory.

Figure 12-19 Memory table in a pagei system.



Page Fault and page Replacement

Unit-07: Page Fault and page Replacement

Address Mapping Using Pages:

- The CPU references a word in memory with a virtual address of 13 bits. The three high-order bits of the virtual address specify a page number and also an address for the memory-page table.
- The content of the word in the memory page table at the page number address is read out into the memory table buffer register.
- If the presence bit is a 1, the block number thus read is transferred to the two high-order bits of the main memory address register. The line number from the virtual address is transferred into the 10 low-order bits of the memory address register. A read signal to main memory transfers the content of the word to the main memory buffer register ready to be used by the CPU.

Page Fault and page Replacement

Unit-07: Page Fault and page Replacement

Address Mapping Using Pages:

- If the presence bit in the word read from the page table is 0, it signifies that the content of the word referenced by the virtual address does not reside in main memory. A call to the operating system is then generated to fetch the required page from auxiliary memory and place it into main memory before resuming computation.

Page Fault and page Replacement

Unit-07: Page Fault and page Replacement

Page Replacement:

- Paging is a process of reading data from, and writing data to, the secondary storage. It is a memory management scheme that is used to retrieve processes from the secondary memory in the form of pages and store them in the primary memory. The main objective of paging is to divide each process in the form of pages of fixed size. These pages are stored in the main memory in frames(blocks). Pages of a process are only brought from the secondary memory to the main memory when they are needed.
- When an executing process refers to a page, it is first searched in the main memory. If it is not present in the main memory, a page fault occurs.

** Page Fault is the condition in which a running process refers to a page that is not loaded in the main memory.

Page Fault and page Replacement

Unit-07: Page Fault and page Replacement

Page Replacement:

** Page Fault is the condition in which a running process refers to a page that is not loaded in the main memory.

- In such a case, the OS has to bring the page from the secondary storage into the main memory. This may cause some pages in the main memory to be replaced due to limited storage. A Page Replacement Algorithm is required to decide which page needs to be replaced.

Page Fault and page Replacement

Unit-07: Page Fault and page Replacement

Page Replacement Algorithm:

- Page Replacement Algorithm decides which page to remove, also called swap out when a new page needs to be loaded into the main memory. Page Replacement happens when a requested page is not present in the main memory and the available space is not sufficient for allocation to the requested page.
- When the page that was selected for replacement was paged out, and referenced again, it has to read in from disk, and this requires for I/O completion. This process determines the quality of the page replacement algorithm: the lesser the time waiting for page-ins, the better is the algorithm.

Page Fault and page Replacement

Unit-07: Page Fault and page Replacement

Page Replacement Algorithm:

- A page replacement algorithm tries to select which pages should be replaced so as to minimize the total number of page misses. There are many different page replacement algorithms. These algorithms are evaluated by running them on a particular string of memory reference and computing the number of page faults. The fewer is the page faults the better is the algorithm for that situation.

** If a process requests for page and that page is found in the main memory then it is called page hit, otherwise page miss or page fault.

Page Fault and page Replacement

Unit-07: Page Fault and page Replacement

Page Replacement Algorithm:

- Some Page Replacement Algorithms :
  1. First In First Out (FIFO)
  2. Least Recently Used (LRU)
  3. Optimal Page Replacement
- First In First Out (FIFO) - This is the simplest page replacement algorithm. In this algorithm, the OS maintains a queue that keeps track of all the pages in memory, with the oldest page at the front and the most recent page at the back.
- When there is a need for page replacement, the FIFO algorithm, swaps out the page at the front of the queue that is the page which has been in the memory for the longest time.

Page Fault and page Replacement

Unit-07: Page Fault and page Replacement

Page Replacement Algorithm:

- Least Recently Used (LRU) - Least Recently Used page replacement algorithm keeps track of page usage over a short period of time. It works on the idea that the pages that have been most heavily used in the past are most likely to be used heavily in the future too.
- In LRU, whenever page replacement happens, the page which has not been used for the longest amount of time is replaced.

Page Fault and page Replacement

Unit-07: Page Fault and page Replacement

Page Replacement Algorithm:

- Optimal Page Replacement - Optimal Page Replacement algorithm is the best page replacement algorithm as it gives the least number of page faults. It is also known as OPT, clairvoyant replacement algorithm, or Belady's optimal page replacement policy.
- In this algorithm, pages are replaced which would not be used for the longest duration of time in the future, i.e., the pages in the memory which are going to be referred farthest in the future are replaced.
- This algorithm was introduced long back and is difficult to implement because it requires future knowledge of the program behaviour. However, it is possible to implement optimal page replacement on the second run by using the page reference information collected on the first run.

Page Fault and page Replacement

Memory Management Hardware:

- A memory management system is a collection of hardware and software procedures for managing the various programs residing in memory in a multiprogramming environment.
- The memory management software is part of an overall operating system available in many computers. Here we are concerned with the hardware unit associated with the memory management system.
- The basic components of a memory management unit are:
  1. A facility for dynamic storage relocation that maps logical memory references into physical memory addresses
  2. A provision for sharing common programs stored in memory by different users.
  3. Protection of information against unauthorized access between users and preventing users from changing operating system functions.

Page Fault and page Replacement

Memory Management Hardware:

- The dynamic storage relocation hardware is a mapping process similar to the paging system. The fixed page size used in the virtual memory system causes certain difficulties with respect to program size and the logical structure of programs. It is more convenient to divide programs and data into logical parts called segments. A segment is a set of logically related instructions or data elements associated with a given name. Segments may be generated by the programmer or by the operating system.
- The sharing of common programs is an integral part of a multiprogramming system. Other system programs residing in memory are also shared by all users in a multiprogramming system without having to produce multiple copies.
- The third issue in multiprogramming is protecting one program from unwanted interaction with another. An example of unwanted interaction is one user's unauthorized copying of another user's program.

Page Fault and page Replacement

Memory Management Hardware:

- The address generated by a segmented program is called a logical address. This is similar to a virtual address except that logical address space is associated with variable-length segments rather than fixed-length pages.
- The logical address may be larger than the physical memory address as in virtual memory, but it may also be equal, and sometimes even smaller than the length of the physical memory address.
- In addition to relocation information, each segment has protection information associated with it.
- Shared programs are placed in a unique segment in each user's logical address space so that a single physical copy can be shared.
- The function of the memory management unit is to map logical addresses into physical addresses similar to the virtual memory mapping concept.

Page Fault and page Replacement

Segmented Page Mapping (Introduction):

- In Segmented Paging, the main memory is divided into variable size segments which are further divided into fixed size pages.
- The length of each segment is allowed to grow and contract according to the needs of the program being executed. One way of specifying the length of a segment is by associating with it a number of equal-size pages.
- The logical address is partitioned into three fields. The segment field specifies a segment number. The page field specifies the page within the segment and the word field gives the specific word within the page.
- A page field of k bits can specify up to 2^k pages. A segment number may be associated with just one page or with as many as 2^k pages. Thus the length of a segment would vary according to the number of pages that are assigned to it.

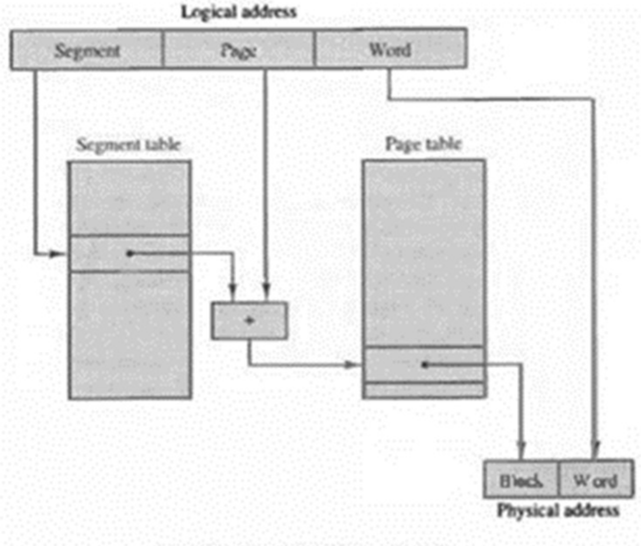
Page Fault and page Replacement

Segmented Page Mapping (Introduction):

- The mapping of the logical address into a physical address is done by means of two tables, as shown in Fig. 12-21(a). The segment number of the logical address specifies the address for the segment table. The entry in the segment table is a pointer address for a page table base.
- The page table base is added to the page number given in the logical address. The sum produces a pointer address to an entry in the page table. The value found in the page table provides the block number in physical memory. The concatenation of the block field with the word field produces the final physical mapped address.

Page Fault and page Replacement

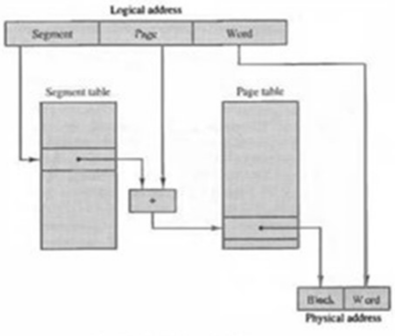
Segmented Page Mapping (Introduction):



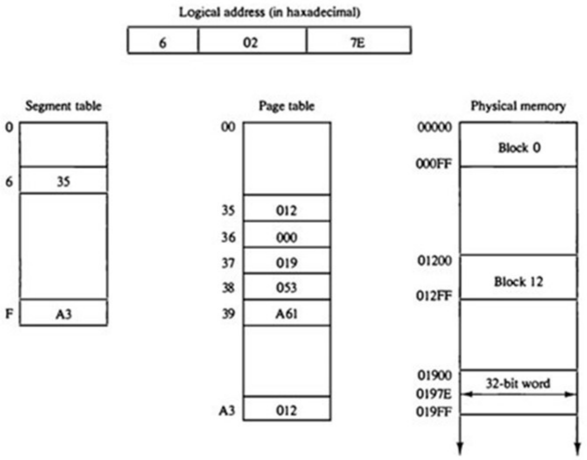
(a) Logical to physical address mapping

Page Fault and page Replacement

Segmented Page Mapping (Introduction):



(a) Logical to physical address mapping



(a) Segment and page table mapping

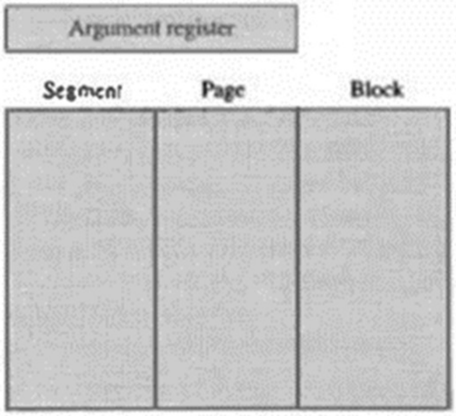
Page Fault and page Replacement

Segmented Page Mapping (Introduction):

- The two mapping tables may be stored in two separate small memories or in main memory. In either case, a memory reference from the CPU will require three accesses to memory: one from the segment table, one from the page table, and the third from main memory. This would slow the system significantly when compared to a conventional system that requires only one reference to memory.
- To avoid this speed penalty, a fast **associative memory** is used to hold the most recently referenced table entries. (This type of memory is sometimes called a translation lookaside buffer, abbreviated TLB.) The first time a given block is referenced, its value together with the corresponding segment and page numbers are entered into the associative memory as shown in Fig. 12-21(b). Thus the mapping process is first attempted by associative search with the given segment and page numbers. If it succeeds, the mapping delay is only that of the associative memory. If no match occurs, the slower table mapping of Fig. 12-21(a) is used and the result transformed into the associative memory for future reference.

Page Fault and page Replacement

Segmented Page Mapping (Introduction):



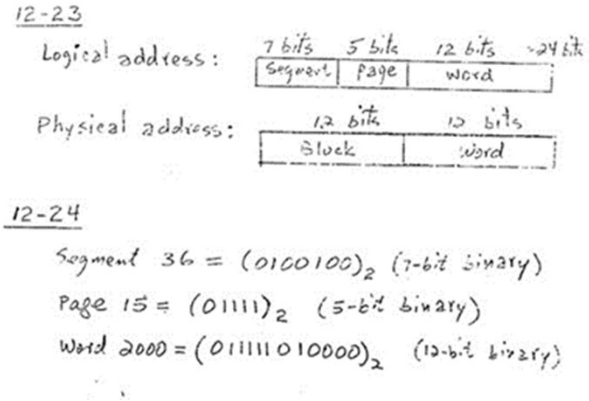
(b) Associative memory translation look-aside buffer (TLB)

Figure 12-21 Mapping in segmented-page memory management unit.

Page Fault and page Replacement

Segmented Page Mapping (Introduction):

Problem (12-23): The logical address space in a computer system consists of 128 segments. Each segment can have up to 32 pages of 4K words in each. Physical memory consists of 4K blocks of 4K words in each. Formulate the logical and physical address format.



12-23

Logical address:

7 bits	5 bits	12 bits	=24 bits
Segment	Page	Word	

Physical address:

12 bits	12 bits
Block	Word

12-24

Segment 36 = $(0100100)_2$ (7-bit binary)

Page 15 = $(01111)_2$ (5-bit binary)

Word 2000 = $(011111010000)_2$ (12-bit binary)

Page Fault and page Replacement

Memory Protection:

Memory protection is a way to control memory access rights on a computer, and is a part of most modern instruction set architectures and operating systems. The main purpose of memory protection is to prevent a process from accessing memory that has not been allocated to it.

Page Fault and page Replacement

Memory Protection:

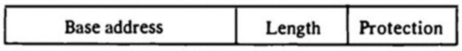
Memory protection can be assigned to the physical address or the logical address. The protection of memory through the physical address can be done by assigning to each block in memory a number of protection bits that indicate the type of access allowed to its corresponding block. Every time a page is moved from one block to another it would be necessary to update the block protection bits. A much better place to apply protection is in the logical address space rather than the physical address space. This can be done by including protection information within the segment table or segment register of the memory management hardware.

Page Fault and page Replacement

Memory Protection:

- The content of each entry in the segment table or a segment register is called a descriptor. A typical descriptor would contain, in addition to a base address field, one or two additional fields for protection purposes.
- A typical format for a segment descriptor is shown in Fig. 12-25. The base address field gives the base of the page table address in a segmented-page organization or the block base address in a segment register organization. This is the address used in mapping from a logical to the physical address.
- The length field gives the segment size by specifying the maximum number of pages assigned to the segment. The length field is compared against the page number in the logical address. A size violation occurs if the page number falls outside the segment length boundary. Thus a given program and its data cannot access

Figure 12-25 Format of a typical segment descriptor.



Base address	Length	Protection

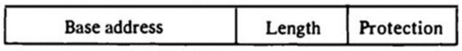
Page Fault and page Replacement

Memory Protection:

The protection field in a segment descriptor specifies the access rights available to the particular segment. In a segmented-page organization, each entry in the page table may have its own protection field to describe the access rights of each page. The protection information is set into the descriptor by the master control program of the operating system. Some of the access rights of interest that are used for protecting the programs residing in memory are:

1. Full read and write privileges
2. Read only (write protection)
3. Execute only (program protection)
4. System only (operating system protection)

Figure 12-25 Format of a typical segment descriptor.



Base address	Length	Protection

Page Fault and page Replacement

Memory Protection:

- Full read and write privileges are given to a program when it is executing its own instructions. Write protection is useful for sharing system programs such as utility programs and other library routines. These system programs are stored in an area of memory where they can be shared by many users. They can be read by all programs, but no writing is allowed. Thus protects them from being changed by other programs.
- The execute-only condition protects programs from being copied. It restricts the segment to be referenced only during the instruction fetch phase but not during the execute phase. Thus it allows the users to execute the segment program instructions but prevents them from reading the instructions as data for the purpose of copying their content.
- Portions of the operating system will reside in memory at any given time. These system programs must be protected by making them inaccessible to unauthorized users. The operating system protection condition is placed in the descriptors of all operating system programs to prevent the occasional user from accessing operating system segments.