Module III
MEMORY SYSTEM

Basic Concepts

- The maximum size of the memory that can be used in any computer is determined by the addressing scheme.
- For example, a 16-bit computer that generates 16-bit addresses is capable of addressing up to 2^{16}=64K memory locations.
- Similarly, machines whose instructions generate 32-bit addresses can utilize a memory that contains up to 232=4G memory locations.
- Most modern computers are byte addressable.
- Form the system stand point; we can view the memory unit as a block box.
- Data transfer between the memory and processor takes place through the use of two processor registers, MAR and MDR.

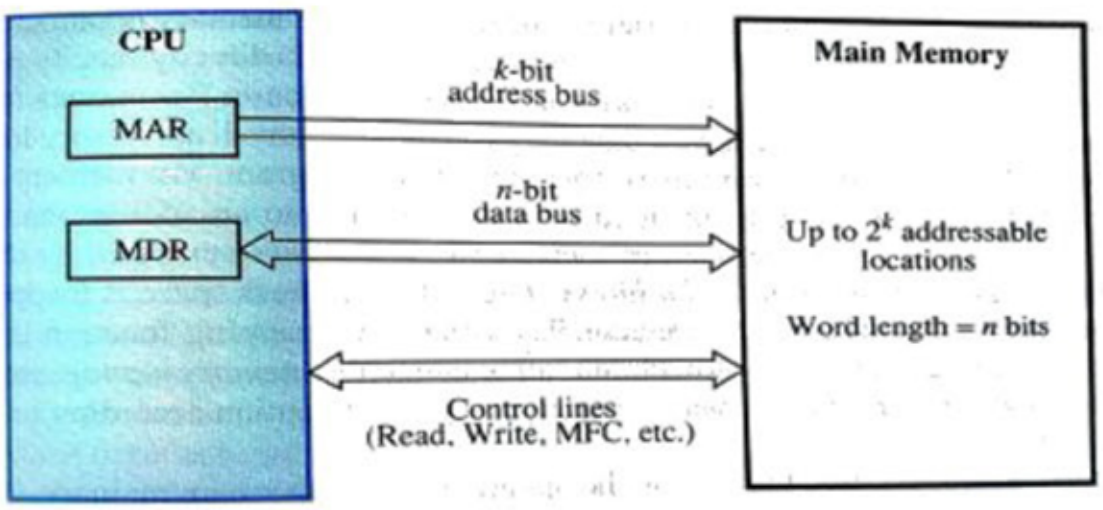


FIGURE 5.2
Connection of the main memory to the CPU.

- As shown in the figure 5.2, if MAR is K bits long and MDR is 'n' bits long, then the Main Memory unit may contain up to 2^k addressable locations and each location will be 'n' bits wide, while the word length is equal to 'n' bits.
- During a "*memory cycle*", n bits of data may be transferred between the Main Memory and CPU.
- This transfer takes place over the processor bus, which has k address lines (address bus), n data lines (data bus) and control lines like Read, Write, Memory Function Completed (***MFC***), Byte specifier etc. (control bus).
- For a ***read operation***, the CPU loads the address into MAR; set READ to 1 and sets other control signals if required. The data from the Main Memory is loaded into MDR and MFC is set to 1.
- For a ***write operation***, MAR, MDR are suitably loaded by the CPU, write is set to 1 and other control signals are set suitably. The MM control circuitry loads the data into appropriate locations and sets MFC to 1.

Memory Access Times

- It is a useful measure of the speed of the memory unit.

- It is the time that elapses between the initiation of an operation and the completion of that operation (for example, the time between READ and MFC).

Memory Cycle Time

- It is an important measure of the memory system.
- It is the minimum time delay required between the initiations of two successive memory operations (for example, the time between two successive READ operations).
- The cycle time is usually slightly longer than the access time.
- A useful measure of the speed of memory units is the time that elapses between the initiation of an operation and the completion of that operation. This is referred to as the memory access time.
- Another important measure is the memory cycle time, which is the minimum time delay required between the initiation of two successive memory operations.
- A memory unit is called random-access memory (RAM) if any location can be accessed for a Read or Write operation in some fixed amount of time that is independent of the location's address.
- The memory cycle time is the bottleneck in the System.
- One way to reduce the memory access time is to use a cache memory.
- Cache memory is a small, fast memory that is inserted between the larger, smaller main memory and the processor.
- Virtual memory is used to increase the apparent size of the physical memory.
- Data are addressed in a virtual address space that can be as large as the addressing capability of the processor. But at any given time, only the active portion of this space is mapped onto locations in the physical memory. The remaining virtual addresses are mapped onto the bulk storage devices used, such as magnetic disks.

SEMICONDUCTOR RAM MEMORIES

Internal organization of Memory chips

- Memory cells are usually organized in the form of an *array*, in which each cell is capable of storing one bit of information.
- Each cell is capable of storing *one bit* of information.
- Each row of cells constitutes a memory word, and all cells of a row are connected to a common line referred to as the *word line,* which is driven by the address decoder on the chip.
- The cells in each column are connected to a *Sense/Write* circuit by two *bit lines,* and the Sense/Write circuits are connected to the data input/output lines of the chip.
- During a Read operation, these circuits' sense, or read, the information stored in the cells selected by a word line and place this information on the output data lines.
- During a Write operation, the Sense/Write circuits receive input data and store them in the cells of the selected word.
- Figure 8.2 is an example of a very small memory circuit consisting of 16 words of 8 bits each. This is referred to as a ***16 × 8 organization***. The data input and the data output of each Sense/Write circuit are connected to a single bidirectional data line that

can be connected to the data lines of a computer. Two control lines, R/W and *CS (Chip Select)*, are provided.

- The R/W (Read/Write) input specifies the required operation, and the CS input selects a given chip in a multichip memory system.

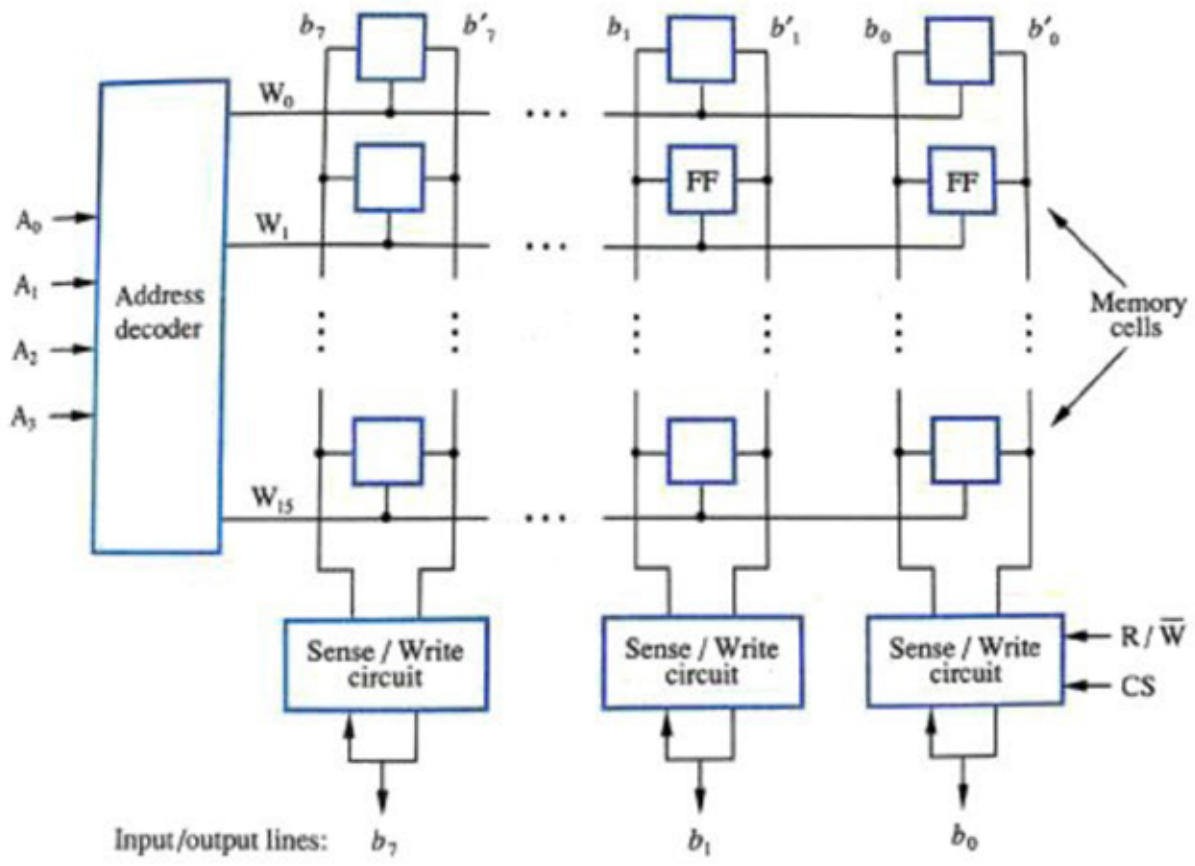


FIGURE 5.3
Organization of bit cells in a memory chip.

- The memory circuit in Figure 5.3 stores 128 bits and requires ***14 external connections*** for ***address (4), data (8), and control lines (2)***. It also needs two lines for power supply and ground connections.
- Consider now a slightly larger memory circuit, one that has 1K (1024) memory cells. This circuit can be organized as a ***128 × 8 memory***, requiring a total of ***19*** external connections, address (*7*), data (*8*), control lines (*2*) and power supply ground (*2*).

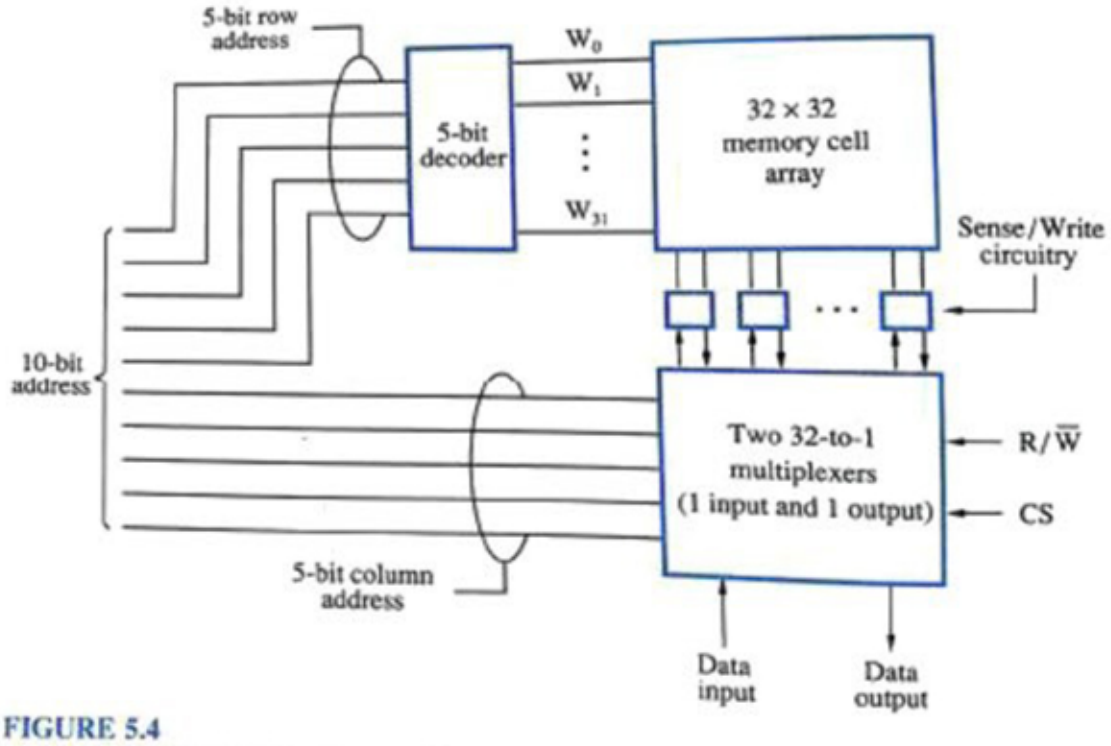


FIGURE 5.4
Organization of a 1K × 1 memory chip.

- Alternatively, the same number of cells can be organized into a 1K×1 format. In this case, a 10-bit address is needed, but there is only one data line, resulting in 15 external connections. Figure 5.4 shows such an organization.
- The required 10-bit address is divided into two groups of 5 bits each to form the row and column addresses for the cell array.
- A row address selects a row of 32 cells, all of which are accessed in parallel. But, only one of these cells is connected to the external data line, based on the column address.

Static Memories

Memories that consist of circuits capable of retaining their state as long as power is applied are known as ***static memories.*** Figure 5.5 illustrates how a *static RAM* (SRAM) cell may be implemented.

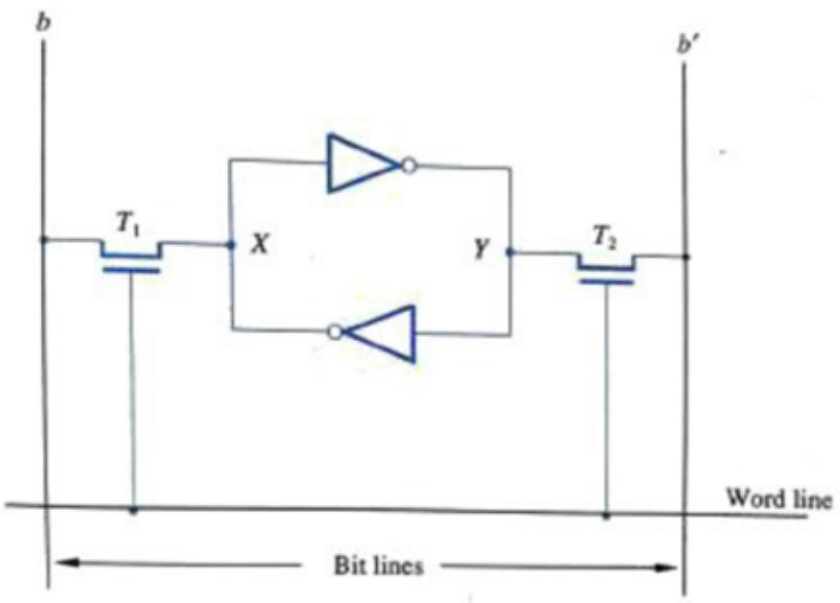


FIGURE 5.5
A static RAM cell.

- Two transistor inverters are cross connected to implement a basic flip-flop.
- The cell is connected to one word line and two bits lines by transistors T1 and T2.
- When word line is at ground level, the transistors are turned off and the latch retains its state.

Read operation: In order to read state of SRAM cell, the word line is activated to close switches T1 and T2. Sense/Write circuits at the bottom monitor the state of b and b'

Write Operation: During a Write operation, the Sense/Write circuit drives bit lines *b* and *b*' instead of sensing their state. It places the appropriate value on bit line *b* and its complement on *b*' and activates the word line.

- Static RAMs can be accessed very quickly.
- Access times on the order of a few nanoseconds are found in commercially available chips.
- SRAMs are used in applications where speed is of critical concern.

Dynamic RAMs

Less expensive and higher density RAMs can be implemented with simpler cells. But, these simpler cells do not retain their state for a long period, unless they are accessed frequently for Read or Write operations. Memories that use such cells are called ***dynamic RAMs*** (DRAMs).

There are two types of DRAM based on their clocking

1. Asynchronous DRAM
2. Synchronous DRAM

Asynchronous DRAM

In the figure 5.7 ADRAM is illustrated, in that:

- Each row can store 512 bytes. 12 bits to select a row, and 9 bits to select a group in a row. Total of 21 bits.
- First apply the row address; RAS signal latches the row address. Then apply the column address, CAS signal latches the address.
- Timing of the memory unit is controlled by a specialized unit which generates RAS and CAS.
- This is asynchronous DRAM.

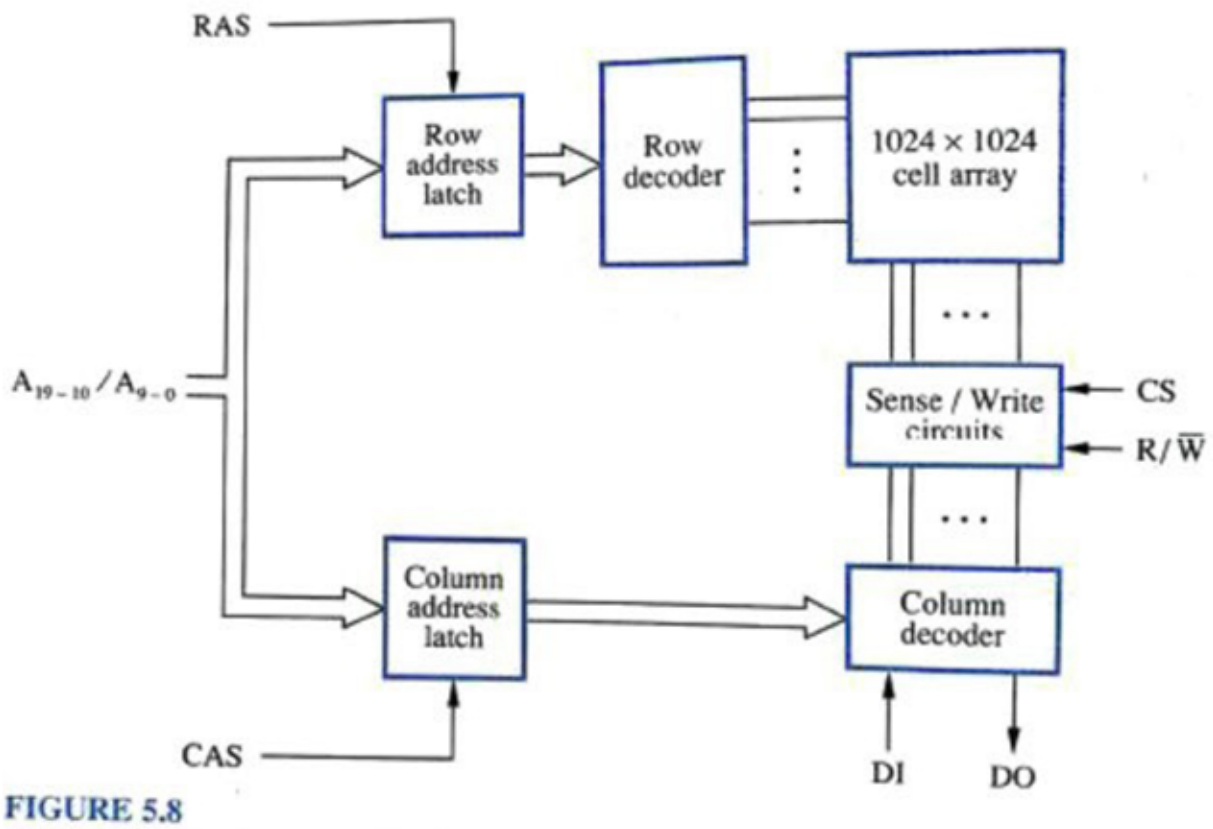


FIGURE 5.8
Internal organization of a 1M × 1 dynamic memory chip.

Fast page mode

- Suppose if we want to access the consecutive bytes in the selected row.
- This can be done without having to reselect the row.
  - ✓ Add a latch at the output of the sense circuits in each row.
  - ✓ All the latches are loaded when the row is selected.
  - ✓ Different column addresses can be applied to select and place different bytes on the data lines.
- Consecutive sequence of column addresses can be applied under the control signal CAS, without reselecting the row.
  - ✓ Allows a block of data to be transferred at a much faster rate than random accesses.
  - ✓ A small collection/group of bytes is usually referred to as a block.

This transfer capability is referred to as the fast page mode feature.

Synchronous DRAMs

Synchronous DRAMs (SDRAMs) structure is shown in Figure 8.8. The cell array is the same as in asynchronous DRAMs. The distinguishing feature of an SDRAM is the use of a clock signal, the availability of which makes it possible to incorporate control circuitry on the chip that provides many useful features.

For example, SDRAMs have built-in refresh circuitry, with a refresh counter to provide the addresses of the rows to be selected for refreshing.

- Operation is directly synchronized with processor clock signal.
- The outputs of the sense circuits are connected to a latch.
- During a Read operation, the contents of the cells in a row are loaded onto the latches.
- During a refresh operation, the contents of the cells are refreshed without changing the contents of the latches.
- Data held in the latches correspond to the selected columns are transferred to the output.
- For a burst mode of operation, successive columns are selected using column address counter and clock .CAS signal need not be generated externally. A new data is placed during raising edge of the clock

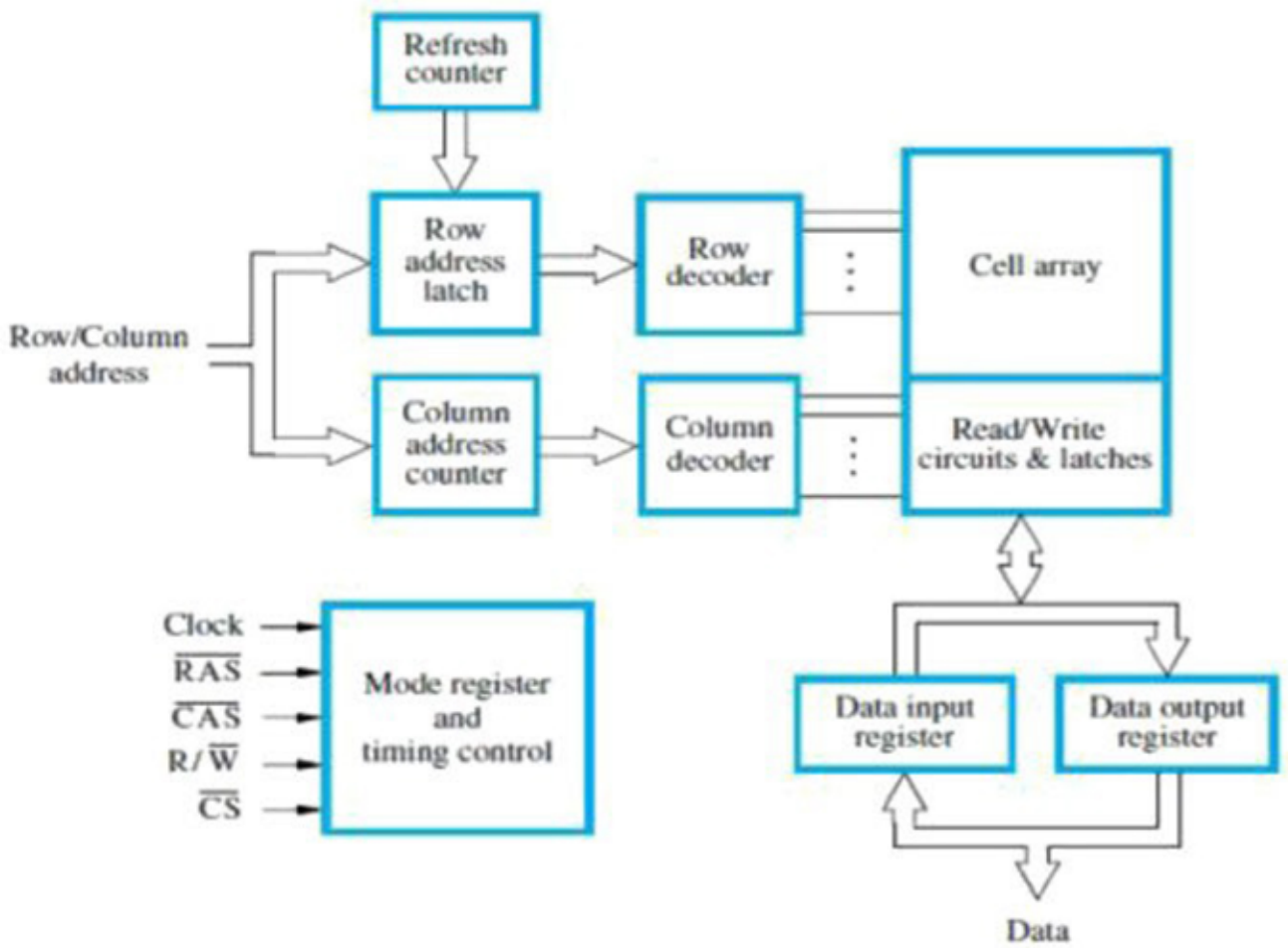


Figure 8.8 Synchronous DRAM.

Latency, Bandwidth, and DDRSDRAMs

- Memory ***latency*** is the time it takes to transfer a word of data to or from memory.
- ***Memory bandwidth*** is the number of bits or bytes that can be transferred in one second.
- ***Double Data Rate SDRAM*** (DDRSDRAM): Cell array is organized in two banks

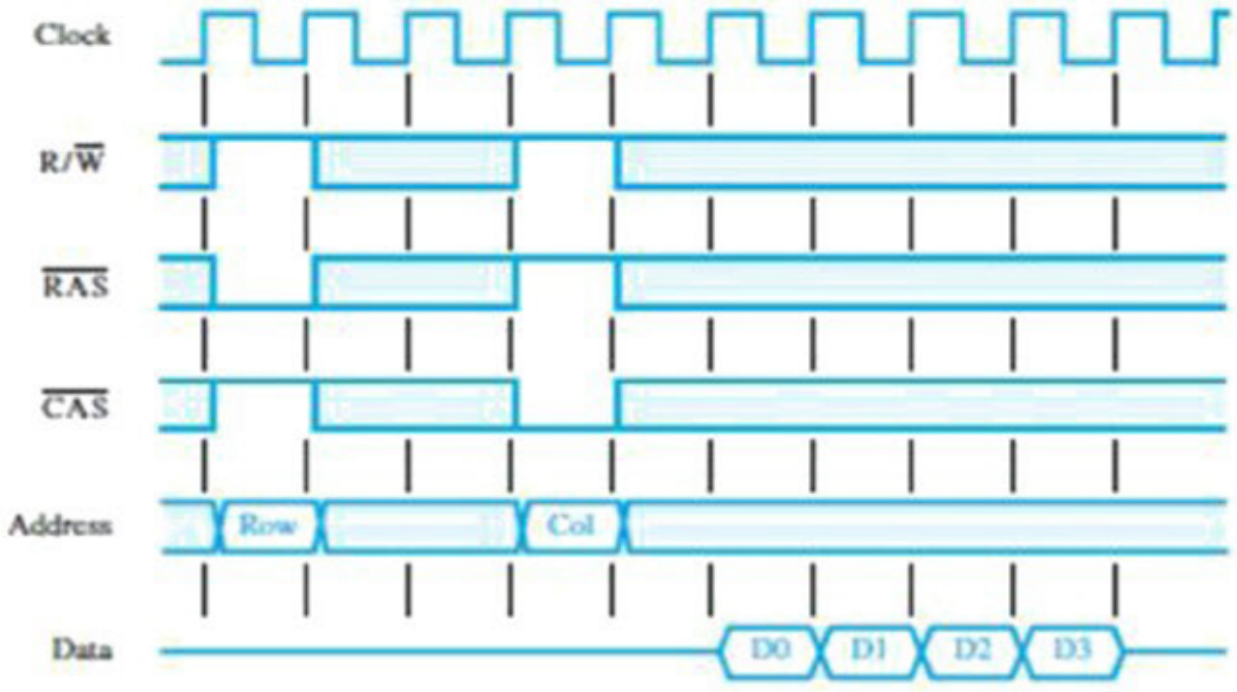
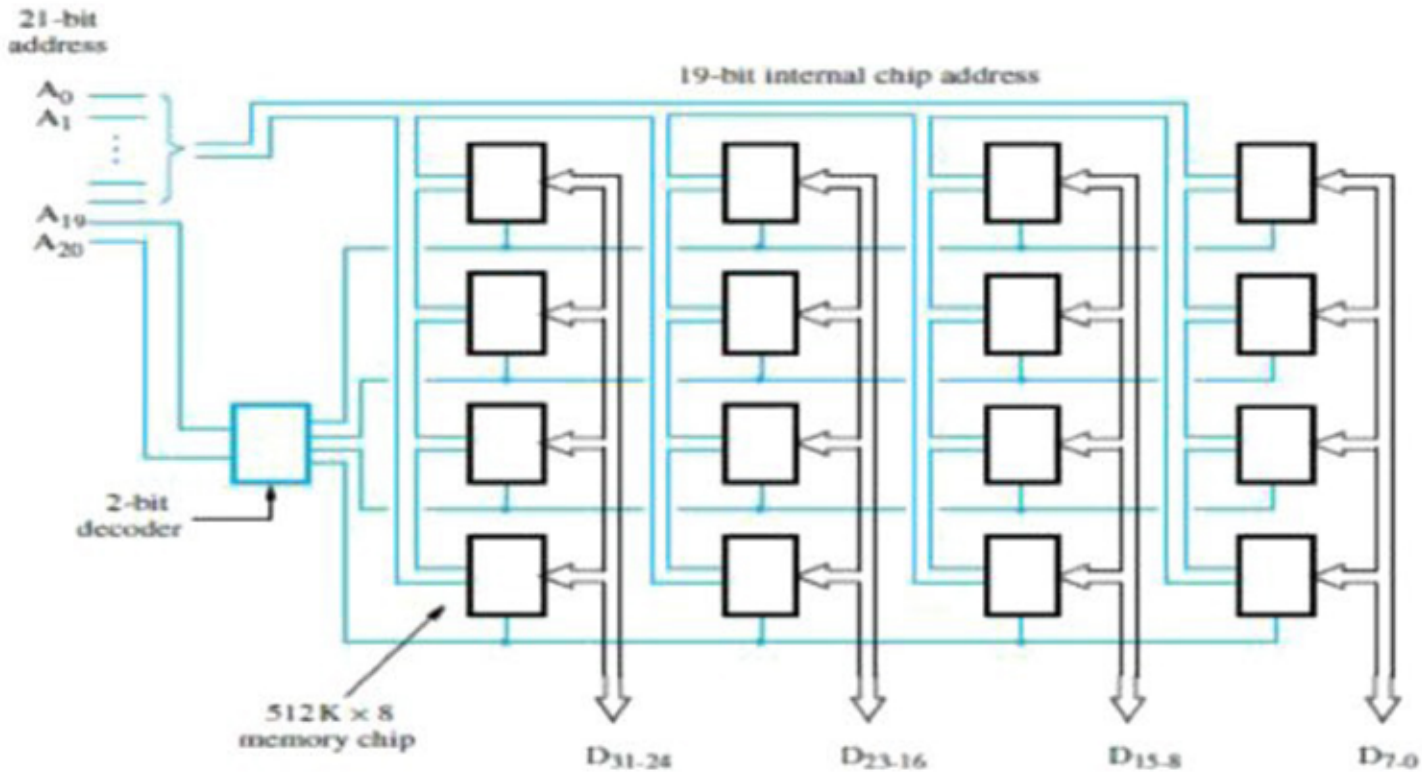


Figure 8.9 A burst read of length 4 in an SDRAM.

The key idea is to take advantage of the fact that a large number of bits are accessed at the same time inside the chip when a row address is applied. Various techniques are used to transfer these bits quickly to the pins of the chip. To make the best use of the available clock speed, data are transferred externally on both the rising and falling edges of the clock. For this reason, memories that use this technique are called *double-data-rate SDRAMs* (DDR DRAMs).

STRUCTURE OF LARGER MEMORIES

Implement a memory unit of 2M words of 32 bits each.



- ✓ Use 512x8 static memory chips.
- ✓ Each column consists of 4 chips.
- ✓ Each chip implements one byte position.
- ✓ A chip is selected by setting its chip select control line to 1.

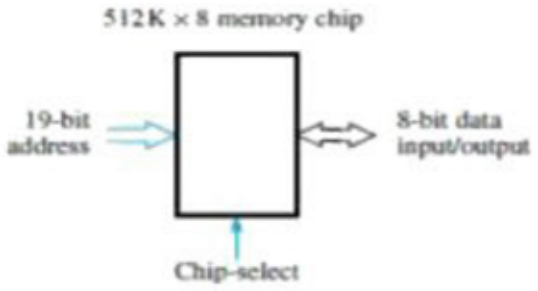


Figure 8.10 Organization of a 2M × 32 memory module using 512K × 8 static memory chips.

- ✓ Selected chip places its data on the data output line, outputs of other chips are in high impedance state.
- ✓ 21 bits to address a 32-bit word.
- ✓ High order 2 bits are needed to select the row, by activating the four Chip Select signals.
- ✓ 19 bits are used to access specific byte locations inside the selected chip.

Exercise question:

Give a block diagram similar to one in figure 8.10 for 8M **X***32 memory using 512* **X** *8 memory chips.*

Solution: The block diagram is essentially the same as in Figure 8.10, except that 16 rows (of four 512 **X** 8 chips) are needed. Address lines A_{18} to A_0 are connected to all chips. Address lines A_{22} to A_{19} are connected to a 4-bit decoder to select one of the 16 rows.

DYNAMIC MEMORIES

- ➢ Large dynamic memory systems can be implemented using DRAM chips in a similar way to static memory systems.
- ➢ Placing large memory systems directly on the motherboard will occupy a large amount of space.
- ➢ Also, this arrangement is inflexible since the memory system cannot be expanded easily.
- ➢ Packaging considerations have led to the development of larger memory units known as ***SIMMs*** (Single In-line Memory Modules) and ***DIMMs*** (Dual In-line Memory Modules).
- ➢ Memory modules are an assembly of memory chips on a small board that plugs vertically onto a single socket on the motherboard.
  - ✓ Occupy less space on the motherboard.
  - ✓ Allows for easy expansion by replacement.

Memory system considerations

Memory controller

- ➢ Recall that in a dynamic memory chip, to reduce the number of pins, multiplexed addresses are used.
- ➢ Address is divided into two parts:
  - ✓ High-order address bits select a row in the array.
  - ✓ They are provided first, and latched using RAS signal.

- ✓ Low-order address bits select a column in the row.
- ✓ They are provided later, and latched using CAS signal.

➢ However, a processor issues all address bits at the same time.

➢ In order to achieve the multiplexing, memory controller circuit is inserted between the processor and memory.

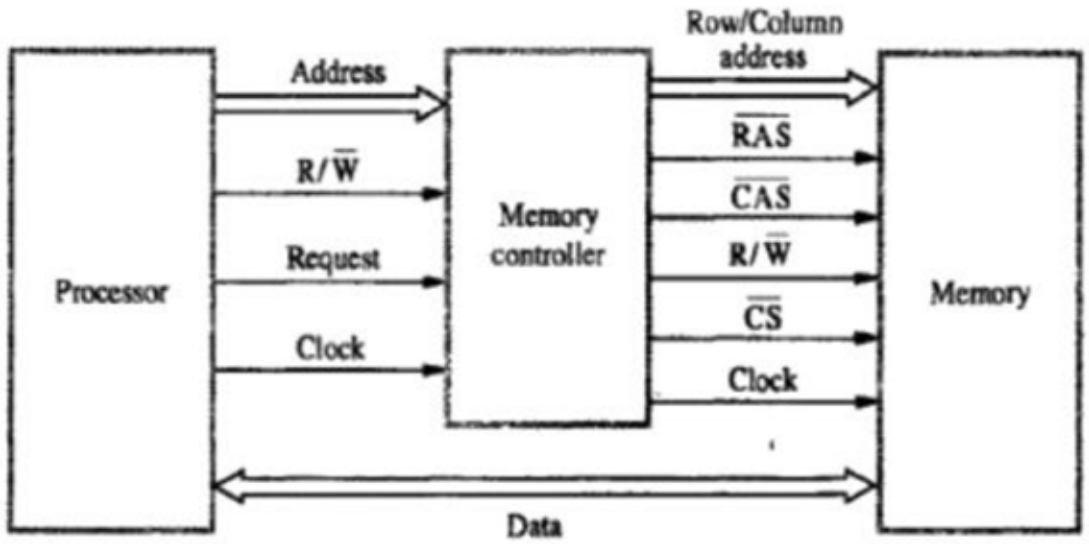


Figure 5.11 Use of a memory controller.

Read-Only Memories (ROMs)

➢ SRAM and SDRAM chips are volatile
- ✓ Lose the contents when the power is turned off.

➢ Many applications need memory devices to retain contents after the power is turned off.
- ✓ For example, computer is turned on; the operating system must be loaded from the disk into the memory.
- ✓ Store instructions which would load the OS from the disk.
- ✓ Need to store these instructions so that they will not be lost after the power is turned off.
- ✓ We need to store the instructions into a non-volatile memory.

➢ Non-volatile memory is read in the same manner as volatile memory.
- ✓ Separate writing process is needed to place information in this memory.

➢ Normal operation involves only reading of data, this type of memory is called Read-Only memory (ROM).

1. Programmable Read-Only Memory (PROM)

- ➢ Allow the data to be loaded by a user.
- ➢ Process of inserting the data is irreversible.
- ➢ Storing information specific to a user in a ROM is expensive.
- ➢ Providing programming capability to a user may be better.

2. Erasable Programmable Read-Only Memory (EPROM)

- ➢ Stored data to be erased and new data to be loaded.
- ➢ Flexibility, useful during the development phase of digital systems.

- Erasable, reprogrammable ROM.
- Erasure requires exposing the ROM to UV light.

3. Electrically Erasable Programmable Read-Only Memory (EEPROM)

- To erase the contents of EPROMs, they have to be exposed to ultraviolet light.
- Physically removed from the circuit.
- EEPROMs the contents can be stored and erased electrically.

4. Flash memory

- Has similar approach to EEPROM.
- Read the contents of a single cell, but write the contents of an entire block of cells.
- Flash devices have greater density.
  - ✓ Higher capacity and low storage cost per bit.
- Power consumption of flash memory is very low, making it attractive for use in equipment that is battery-driven.
- Single flash chips are not sufficiently large, so larger memory modules are implemented using flash cards and flash drives.

SPEED, SIZE, AND COST

The entire computer memory can be viewed as the hierarchy depicted in Figure 8.14.

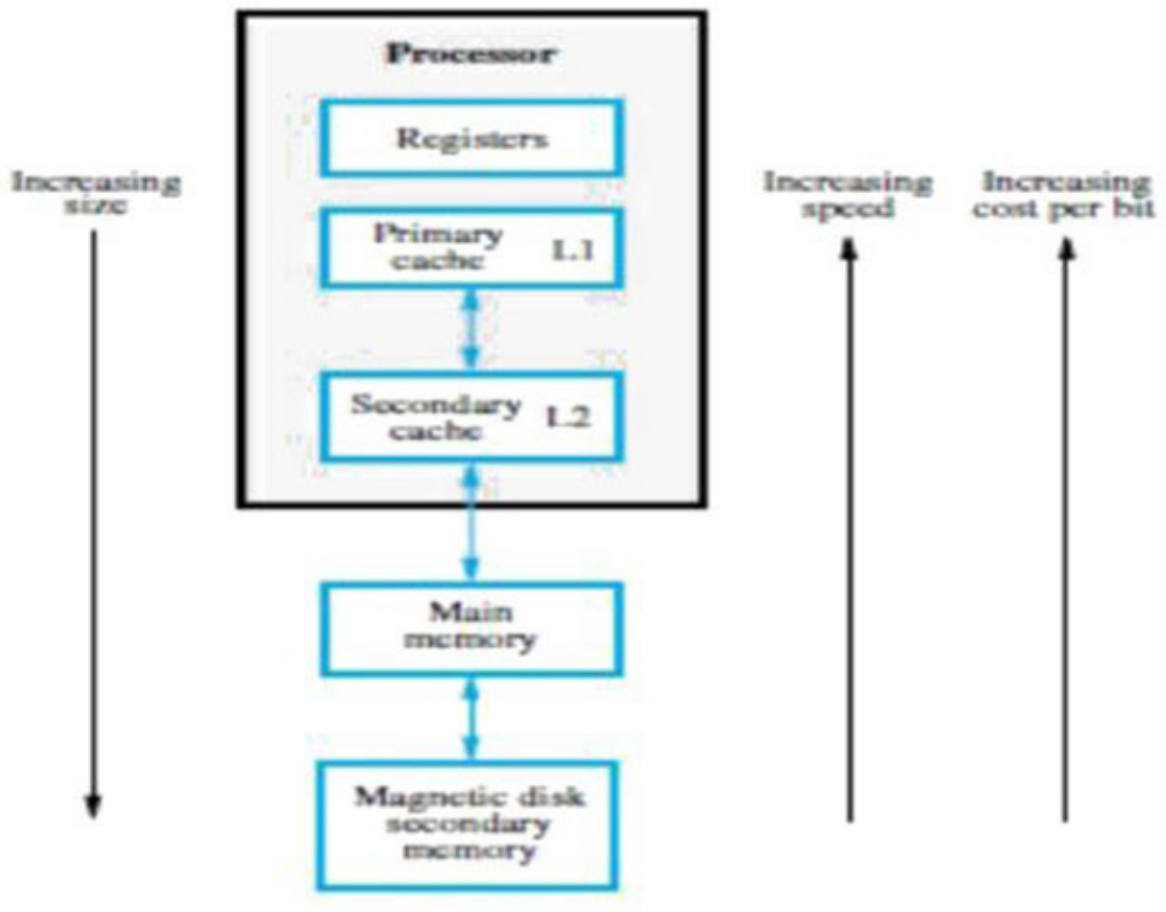


Figure 8.14 Memory hierarchy.

- The fastest access is to data held in ***processor registers***. Therefore, if we consider the registers to be part of the memory hierarchy, then the *processor registers* are at

the top in terms of speed of access. Of course, the registers provide only a minuscule portion of the required memory.

- At the next level of the hierarchy is a relatively small amount of memory that can be implemented directly on the processor chip. This memory, called a ***processor cache***, holds copies of the instructions and data stored in a much larger memory that is provided externally.
- There are ***often two or more levels of cache***.
- A primary cache is always located on the processor chip. This cache is small and its access time is comparable to that of processor registers.
- The **primary cache** is referred to as the ***level 1*** (L1) cache. A larger, and hence somewhat slower, **secondary cache** is placed between the primary cache and the rest of the memory. It is referred to as the ***level 2*** (L2) cache. Often, the L2 cache is also housed on the processor chip.
- Some computers have a ***level 3*** (L3) cache of even larger size, in addition to the L1 and L2 caches. An L3 cache, also implemented in SRAM technology, may or may not be on the same chip with the processor and the L1 and L2 caches.
- The next level in the hierarchy is the ***main memory***. This is a large memory implemented using dynamic memory components, typically assembled in memory modules such as DIMMs.
- ***Disk devices*** provide a very large amount of inexpensive memory, and they are widely used as secondary storage in computer systems. They are very slow compared to the main memory. They represent the bottom level in the memory hierarchy.

CACHE MEMORY

- Processor is much faster than the main memory.
  - ✓ As a result, the processor has to spend much of its time waiting while instructions and data are being fetched from the main memory.
  - ✓ Major obstacle towards achieving good performance.
- Speed of the main memory cannot be increased beyond a certain point.
- Cache memory is an architectural arrangement which makes the main memory appear faster to the processor than it really is.
- Cache memory is based on the property of computer programs known as "***locality of reference***".

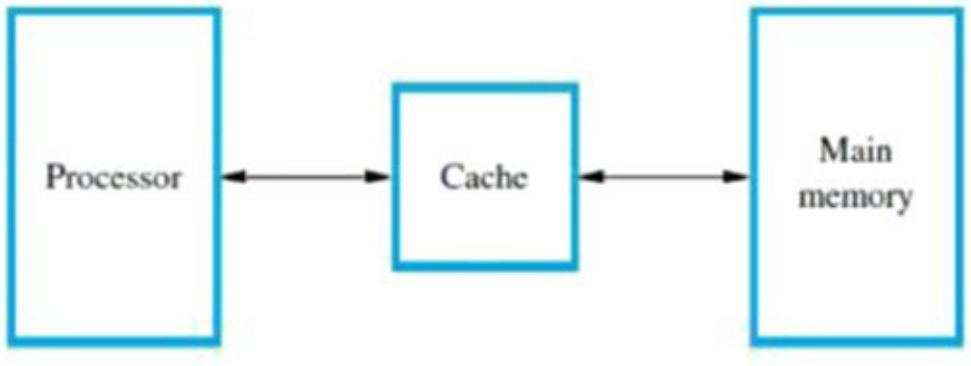


Figure 8.15 Use of a cache memory.

- Analysis of programs indicates that many instructions in localized areas of a program are executed repeatedly during some period of time, while the others are accessed relatively less frequently.
- These instructions may be the ones in a loop, nested loop or few procedures calling each other repeatedly.
    1. ***Temporal locality of reference:***
       Recently executed instruction is likely to be executed again very soon.
    2. ***Spatial locality of reference:***
       Instructions with addresses (proximity) close to a recently instruction are likely to be executed soon.
- Processor issues a Read request a block of words is transferred from the main memory to the cache, one word at a time.
- Subsequent references to the data in this block of words are found in the cache.
- At any given time, only some blocks in the main memory are held in the cache. Which blocks in the main memory are in the cache is determined by **a "*mapping function*".**
- When the cache is full, and a block of words needs to be transferred from the main memory, some block of words in the cache must be replaced. This is determined by a **"*replacement algorithm*".**

Cache hit

- Existence of a cache is transparent to the processor. The processor issues Read and Write requests in the same manner.
- If the data is in the cache it is called a ***Read or Write hit.***
- ***Read hit:*** The data is obtained from the cache.
- ***Write hit:*** Cache has a replica of the contents of the main memory.
- Contents of the cache and the main memory may be updated simultaneously. This is the ***write-through*** protocol.
- Update the contents of the cache, and mark it as updated by setting a bit known as the ***dirty bit or modified bit***.
- The contents of the main memory are updated when this block is replaced. This is ***write-back or copy-back*** protocol.

Cache miss

- If the data is not present in the cache, then a ***Read miss or Write miss*** occurs.
- ***Read miss:***
    - ✓ Block of words containing this requested word is transferred from the memory.
    - ✓ After the block is transferred, the desired word is forwarded to the processor.
    - ✓ The desired word may also be forwarded to the processor as soon as it is transferred without waiting for the entire block to be transferred. This is called ***load-through or early-restart***.
- ***Write-miss:***
    - ✓ Write-through protocol is used, and then the contents of the main memory are updated directly.

- ✓ If write-back protocol is used, the block containing the addressed word is first brought into the cache. The desired word is overwritten with new information.

Cache Coherence Problem

- A bit called as "***valid bit***" is provided for each block.
- If the block contains valid data, then the bit is set to 1, else it is 0.
- Valid bits are set to 0, when the power is just turned on.
- When a block is loaded into the cache for the first time, the valid bit is set to 1.
- Data transfers between main memory and disk occur directly bypassing the cache.
- When the data on a disk changes, the main memory block is also updated.
- However, if the data is also resident in the cache, then the valid bit is set to 0.
- What happens if the data in the disk and main memory changes and the write-back protocol is being used?
- In this case, the data in the cache may also have changed and is indicated by the dirty bit.
- The copies of the data in the cache, and the main memory are different. This is called the ***cache coherence problem***.
- One option is to force a write-back before the main memory is updated from the disk.

MAPPING FUNCTIONS

Mapping functions determine how memory blocks are placed in the cache.

A simple processor example:

- Cache consisting of 128 blocks of 16 words each.
- Total size of cache is 2048 (2K) words.
- Main memory is addressable by a 16-bit address.
- Main memory has 64K words.
- Main memory has 4K blocks of 16 words each.

Three mapping functions:

1. Direct mapping
2. Associative mapping
3. Set-associative mapping.

Direct mapping

- Block ***j*** of the main memory maps to ***j*** modulo 128 of the cache. 0 maps to 0, 129 maps to 1.
- More than one memory block is mapped onto the same position in the cache. May lead to contention for cache blocks even if the cache is not full.
- Resolve the contention by allowing new block to replace the old block, leading to a trivial replacement algorithm.
- Memory address is divided into three fields:
- Low order 4 bits determine one of the 16 **words** in a block.
- When a new **block** is brought into the cache, the next 7 bits determine which cache block this new block is placed in.

- High order 5 bits determine which of the possible 32 blocks is currently present in the cache. These are **tag** bits.
- Simple to implement but not very flexible.

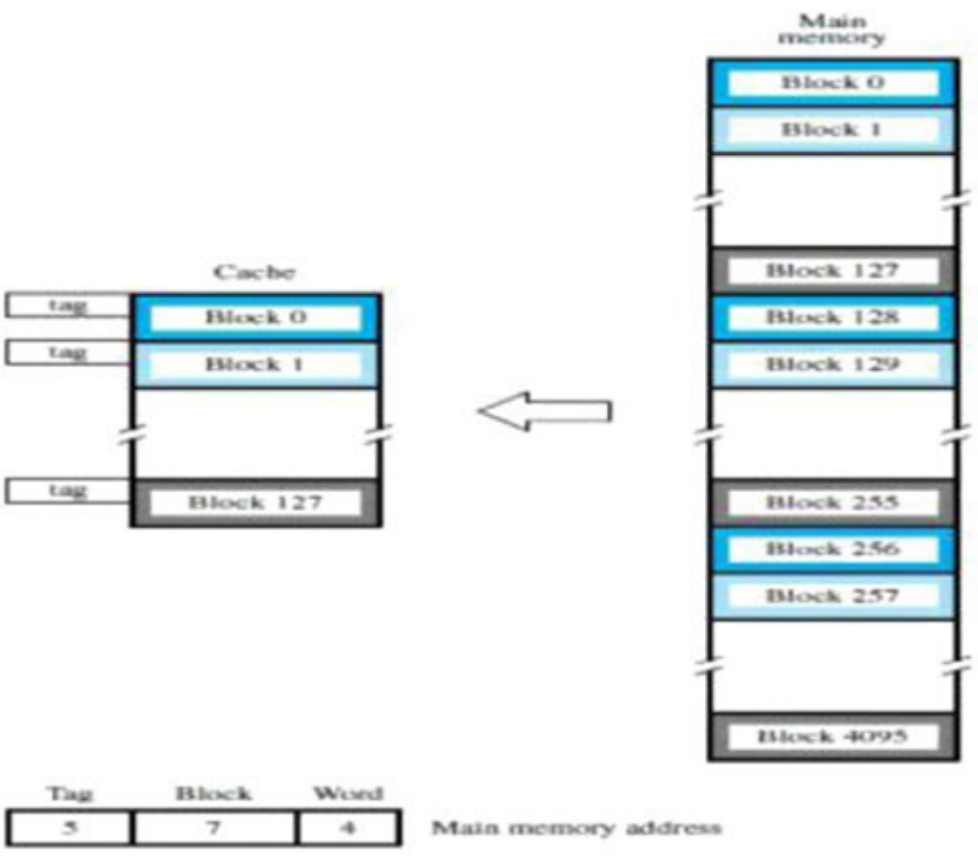


Figure 8.16 Direct-mapped cache.

Associative mapping

- Main memory block can be placed into any cache position.
- Memory address is divided into two fields:

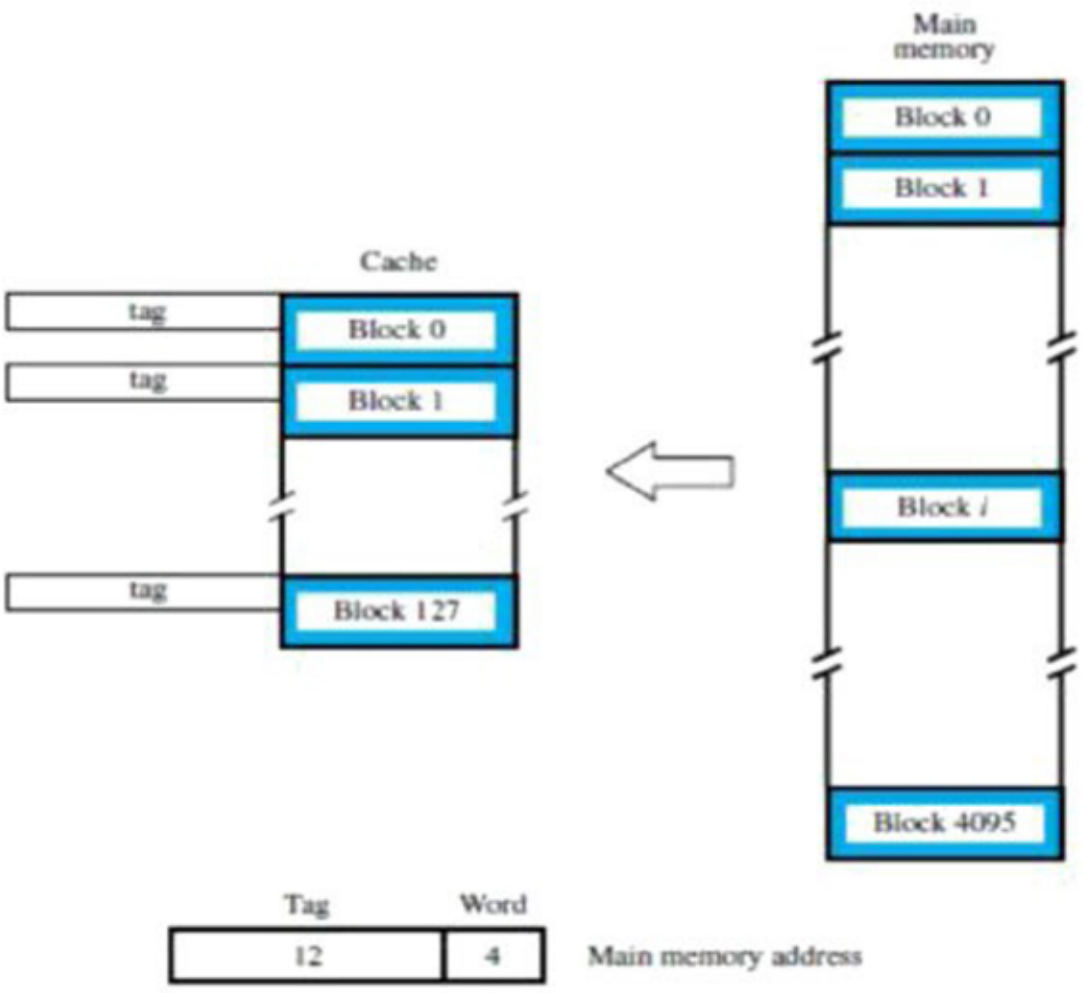


Figure 8.17 Associative-mapped cache.

- Low order 4 bits identify the **word** within a block.
- High order 12 bits or **tag** bits identify a memory block when it is resident in the cache.
- Flexible, and uses cache space efficiently.
- Replacement algorithms (LRU) can be used to replace an existing block in the cache when the cache is full.
- Cost is higher than direct-mapped cache because of the need to search all 128 patterns to determine whether a given block is in the cache.

Set-Associative mapping

- Blocks of cache are grouped into sets.
- Mapping function allows a block of the main memory to reside in any block of a specific set.
- Divide the cache into 64 sets, with two blocks per set.
- Memory block 0, 64, 128 etc. map to block 0, and they can occupy either of the two positions.
- Memory address is divided into three fields:
  1. 6 bit field determines the **set** number.
  2. High order 6 bit fields are compared to the **tag** fields of the two blocks in a **set**.
  3. 4 bits are used to represent he **word**.
- Set-associative mapping is a combination of direct and associative mapping.
- Number of blocks per set is a design parameter.
- One extreme is to have all the blocks in one set, requiring no set bits (fully associative mapping).
- Other extreme is to have one block per set, is the same as direct mapping.

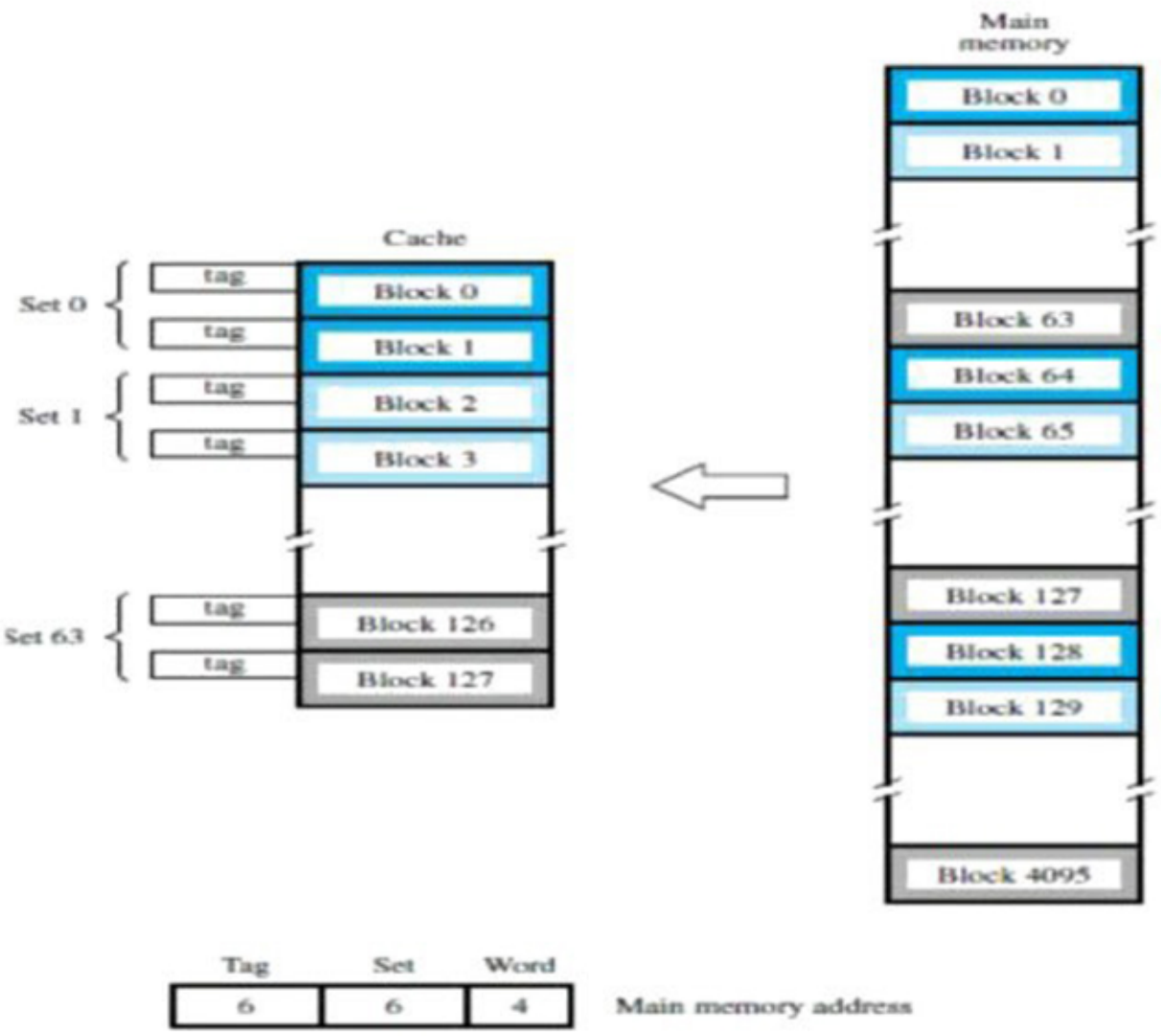


Figure 8.18 Set-associative-mapped cache with two blocks per set.

REPLACEMENT ALGORITHM

- In a direct-mapped cache, the position of each block is fixed, hence no replacement strategy exists.
- In associative and set-associative caches, when a new block is to be brought into the cache and all the Positions that it may occupy are full, the cache controller must decide which of the old blocks to overwrite.
- This is important issue because the decision can be factor in system performance.
- The objective is to keep blocks in the cache that are likely to be referenced in the near future.
- It's not easy to determine which blocks are about to be referenced. The property of locality of reference gives a clue to a reasonable strategy.
- When a block is to be over written, it is sensible to overwrite the one that has gone the longest time without being referenced. This block is called the ***Least Recently Used (LRU)*** block, and technique is called the ***LRU Replacement algorithm.***
- To use LRU Algorithm, the cache controller must track references to all blocks as computations proceeds. Suppose it is required to track the LRU block of a four-block set in a set associative cache.
- A 2 bit counter can be used for each cache. When a ***hit*** occurs, the counter of the block that is referenced is set to 0 and when a ***miss*** occurs and if the set is not full, the counter associated with the new block loaded from the main memory is set to 0, and value of all other counters are increased by 1. Otherwise the block with the counter value 3 is removed, and the new block is put in its place and its counter is set to 0.
- The LRU algorithm has been used extensively for many access patterns, but it can lead to poor performance in some cases.
- For example, it produces disappointing results when accesses are made to sequential elements of an array that is slightly too large to fit into the cache.
- Performance of LRU algorithm can be improved by introducing a small amount of randomness in deciding which block to replace.

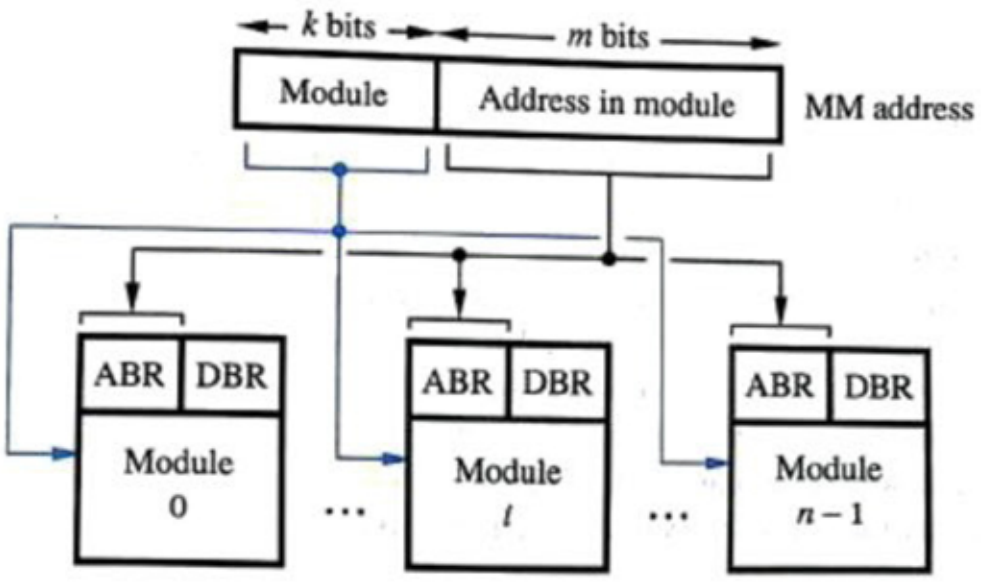
PERFORMANCE CONSIDERATIONS

- A key design objective of a computer system is to achieve the best possible performance at the lowest possible cost.
  - ✓ Price/performance ratio is a common measure of success.
- Performance of a processor depends on:
  - ✓ How fast machine instructions can be brought into the processor for execution.
  - ✓ How fast the instructions can be executed.

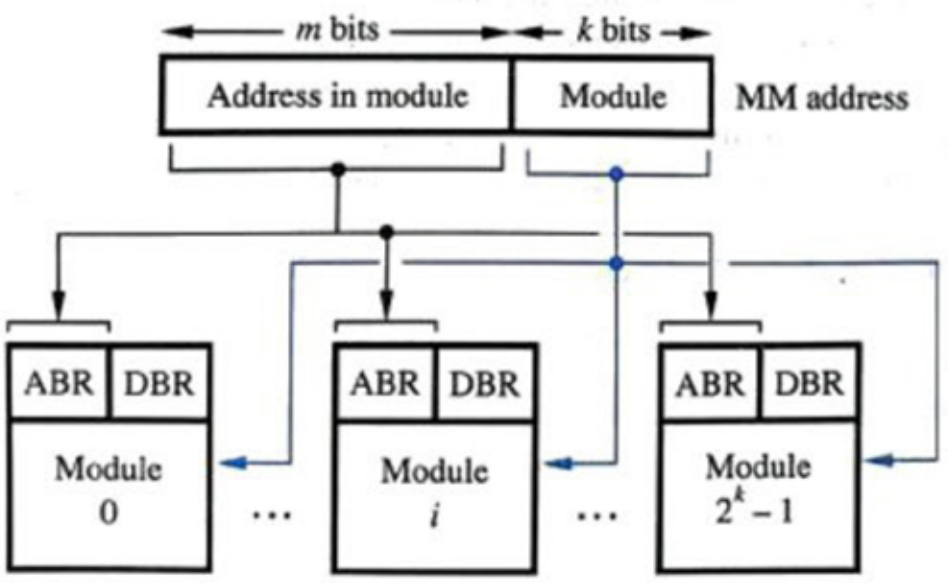
Interleaving

- Divides the memory system into a number of memory modules. Each module has its own ***Address Buffer Register (ABR)*** and ***Data Buffer Register (DBR)***.
- Arranges addressing so that successive words in the address space are placed in different modules.

- When requests for memory access involve consecutive addresses, the access will be to different modules.
- Since parallel access to these modules is possible, the average rate of fetching words from the Main Memory can be increased.



(a) Consecutive words in a module



(b) Consecutive words in consecutive modules

FIGURE 5.24
Addressing multiple-module memory systems.

- Consecutive words are placed in a module.
- High-order k bits of a memory address determine the module.
- Low-order m bits of a memory address determine the word within a module.
- When a block of words is transferred from main memory to cache, only one module is busy at a time.
- Consecutive words are located in consecutive modules.
- Consecutive addresses can be located in consecutive modules.
- While transferring a block of data, several memory modules can be kept busy at the same time.

Example 5.1

Consider the time required to transfer a block of data from the main memory to cache when a read miss occurs. Suppose that a *cache with 8-words* blocks is used. On a read miss, the block that contains the desired word must be copied from the memory into the cache. Assume that hardware has following properties: -

- It takes ***1*** *clock cycle* to send an address to the main memory
- The main memory is built with relatively slow DRAM chips that allow the *first word* to be accessed in ***8*** *cycles*, but subsequent words of the block are accessed in ***4*** *clock cycles per word.*
- Also ***1*** *clock* cycle is needed to send one word to the cache

If the *single memory module* is used, then the time needed to load the desired block into the cache is

$$1 + 8 + (7*4) + 1 = 38\text{cycles}$$

Suppose now that the memory is consumed as ***4*** *interleaved* modules. When the starting address of the block arrives at the memory, all ***4*** modules begin accessing the required data, using higher-order bits of the address. After ***8*** *clock cycles*, each module has one word of data in its DBR. These words are transferred to the cache, one word at a time, during next ***4*** clock cycles. During this time, the next word in each module is accessed. Then it takes another ***4*** *cycles* to transfer these words to the cache.

Therefore, the total time needed to load the block from the interleaved memory is

$$1+8+4+4=17\text{cycles}$$

Transfer time reduced by more than factor of 2.

Hit Rate and Miss Penalty

The data in a cache is called a ***hit***. The number of hits stated as a fraction of all attempted accesses is called the ***hit rate***, and the ***miss rate*** is the number of misses stated as a fraction of attempted accesses.

The total access time seen by the processor when a miss occurs as the ***miss penalty***.

- Hit rate can be improved by increasing block size, while keeping cache size constant
- Block sizes that are neither very small nor very large give best results.
- Miss penalty can be reduced if load-through approach is used when loading new blocks into cache.

Consider a system with only one level of cache. In this case, the miss penalty consists almost entirely of the time to access a block of data in the main memory.

Let ***h*** be the hit rate,

M the miss penalty and

C the time to access information in the cache.

Thus, the average access time experienced by the processor is

$$t_{ave} = hC + (1 - h)M$$

Continuation of example 5.1 If the computer has no cache, then using a fast processor and typical DRAM main memory, it takes *10 clock* cycles for each memory read access. Suppose

the computer has a cache that holds 8-word blocks and an interleaved main memory, it takes *17 cycles* to load block into cache.
Assume that *30%* of the instructions in a typical program perform a read or a write operation, which means that there are *130 memory* access for every *100 instructions* executed. Assume that hit rates in the cache are *0.95* for instructions and *0.9* for data.
Let's assume that the miss penalty is the same for both read and write access. Then a rough estimate of the improvement in performance that results from using the cache can be obtained as

$$\frac{\text{Time without cache}}{\text{Time with cache}}$$

$$\frac{130 * 10}{100\,(0.95 * 1 + 0.05 * 17) + 30\,(0.9 * 1 + 0.1 * 17)} = 5.04$$

0.05 → Penalty

This result suggests that the computer with the cache perform 5 times better.
Consider how effective this cache compared to an ideal cache that has a hit rate of 100%

$$\frac{100\,(0.95 * 1 + 0.05 * 17) + 3\,(0.9 * 1 + 0.1 * 17)}{130} = 1.98$$

Caches on the processor chip

- In high performance processors 2 levels of caches are normally used.
- Average access time in a system with 2 levels of caches is

$$T_{ave} = h1c1 + (1-h1)\,h2c2 + (1-h1)(1-h2)M$$

OTHER PERFORMANCE ENHANCEMENTS

Write buffer

1. **Write-through**
   - Each write operation involves writing to the main memory.
   - If the processor has to wait for the write operation to be complete, it slows down the processor.
   - Processor does not depend on the results of the write operation.
   - Write buffer can be included for temporary storage of write requests.
   - Processor places each write request into the buffer and continues execution.
   - If a subsequent Read request references data which is still in the write buffer, then this data is referenced in the write buffer.
2. **Write-back**

- Block is written back to the main memory when it is replaced.
- If the processor waits for this write to complete, before reading the new block, it is slowed down.
- Fast write buffer can hold the block to be written, and the new block can be read first.

Prefetching

- New data are brought into the processor when they are first needed.
- Processor has to wait before the data transfer is complete.
- Prefetch the data into the cache before they are actually needed, or a before a Read miss occurs.
- Prefetching can be accomplished through software by including a special instruction in the machine language of the processor.
  - ✓ Inclusion of prefetch instructions increases the length of the programs.
- Prefetching can also be accomplished using hardware:
  - ✓ Circuitry that attempts to discover patterns in memory references and then prefetches according to this pattern.

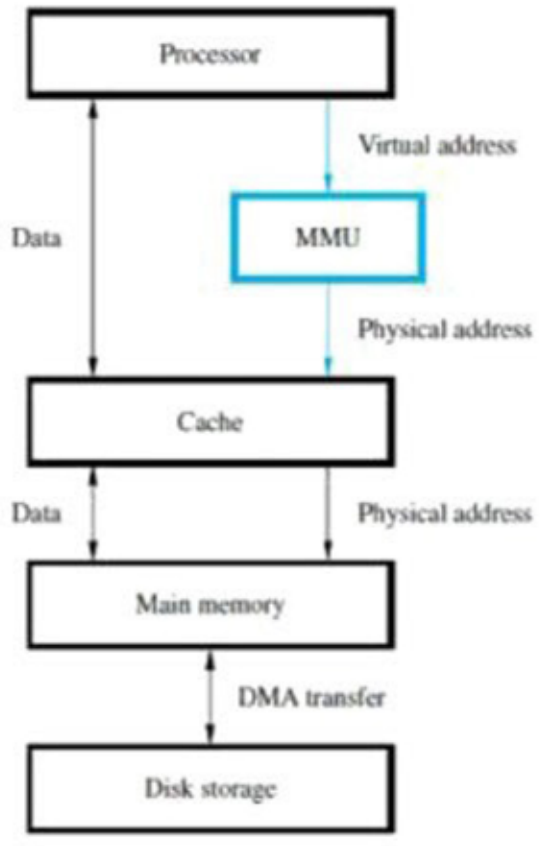


Figure 8.24 Virtual memory organization.

- *Memory management unit (MMU) translates virtual addresses into physical addresses.*
- *If the desired data or instructions are in the main memory they are fetched as described previously.*
- *If the desired data or instructions are not in the main memory, they must be transferred from secondary storage to the main memory.*
- *MMU causes the operating system to bring the data from the secondary storage into the main memory.*
- The program and data are composed of fixed-length units called pages.
- A page consists of a block of words that occupy contiguous locations in the main memory.
- Page is a basic unit of information that is transferred between secondary storage and main memory.
- Size of a page commonly ranges from 2K to 16K bytes.
  - ✓ Pages should not be too small, because the access time of a secondary storage device is much larger than the main memory.
  - ✓ Pages should not be too large, else a large portion of the page may not be used, and it will occupy valuable space in the main memory.
- Each virtual or logical address generated by a processor is interpreted as a virtual page number (high-order bits) plus an offset (low-order bits) that specifies the location of a particular byte within that page.
- Information about the main memory location of each page is kept in the *page table*.
  - ✓ Main memory address where the page is stored.
  - ✓ Current status of the page.
- Area of the main memory that can hold a page is called as *page frame*.
- Starting address of the page table is kept in a *page table base register.*
- Virtual page number generated by the processor is added to the contents of the page table base register.

- ✓ This provides the address of the corresponding entry in the page table.

➢ The contents of this location in the page table give the starting address of the page if the page is currently in the main memory.

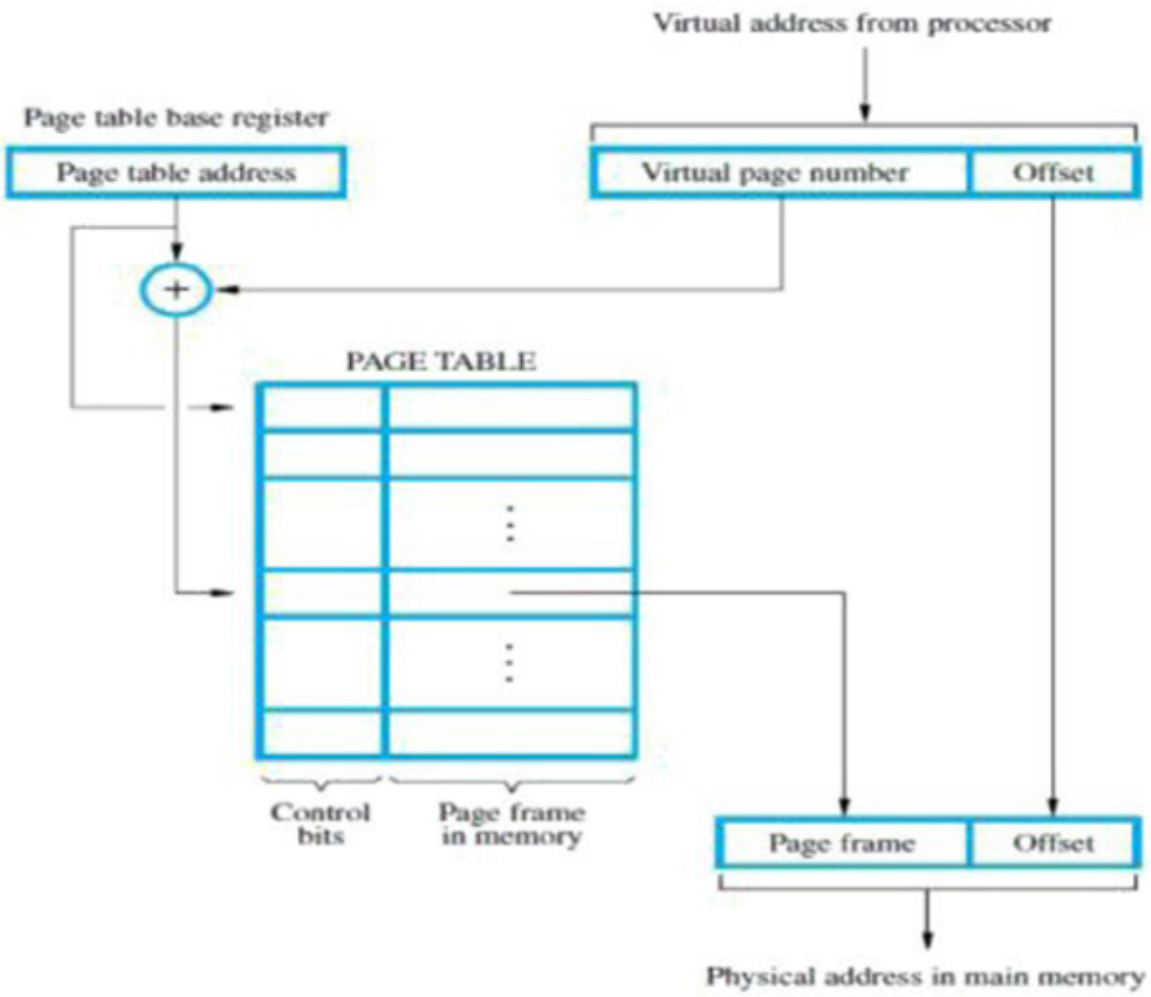


Figure 8.25 Virtual-memory address translation.

➢ Page table entry for a page also includes some control bits which describe the status of the page while it is in the main memory.

➢ One bit indicates the validity of the page.
- ✓ Indicates whether the page is actually loaded into the main memory.
- ✓ Allows the operating system to invalidate the page without actually removing it.

➢ One bit indicates whether the page has been modified during its residency in the main memory.
- ✓ This bit determines whether the page should be written back to the disk when it is removed from the main memory.
- ✓ Similar to the dirty or modified bit in case of cache memory.

➢ Other control bits for various other types of restrictions that may be imposed.
- ✓ For example, a program may only have read permission for a page, but not write or modify permissions.

➢ Page table is kept in the main memory.

➢ A copy of a small portion of the page table can be accommodated within the MMU.
- ✓ Portion consists of page table entries that correspond to the most recently accessed pages.

➢ A small cache called as *Translation Lookaside Buffer (TLB)* is included in the MMU.
- ✓ TLB holds page table entries of the most recently accessed pages.

➢ Recall that cache memory holds most recently accessed blocks from the main memory.

- ✓ Operation of the TLB and page table in the main memory is similar to the operation of the cache and main memory.

➢ Page table entry for a page includes:
- ✓ Address of the page frame where the page resides in the main memory.
- ✓ Some control bits.

➢ In addition to the above for each page, TLB must hold the virtual page number for each page.

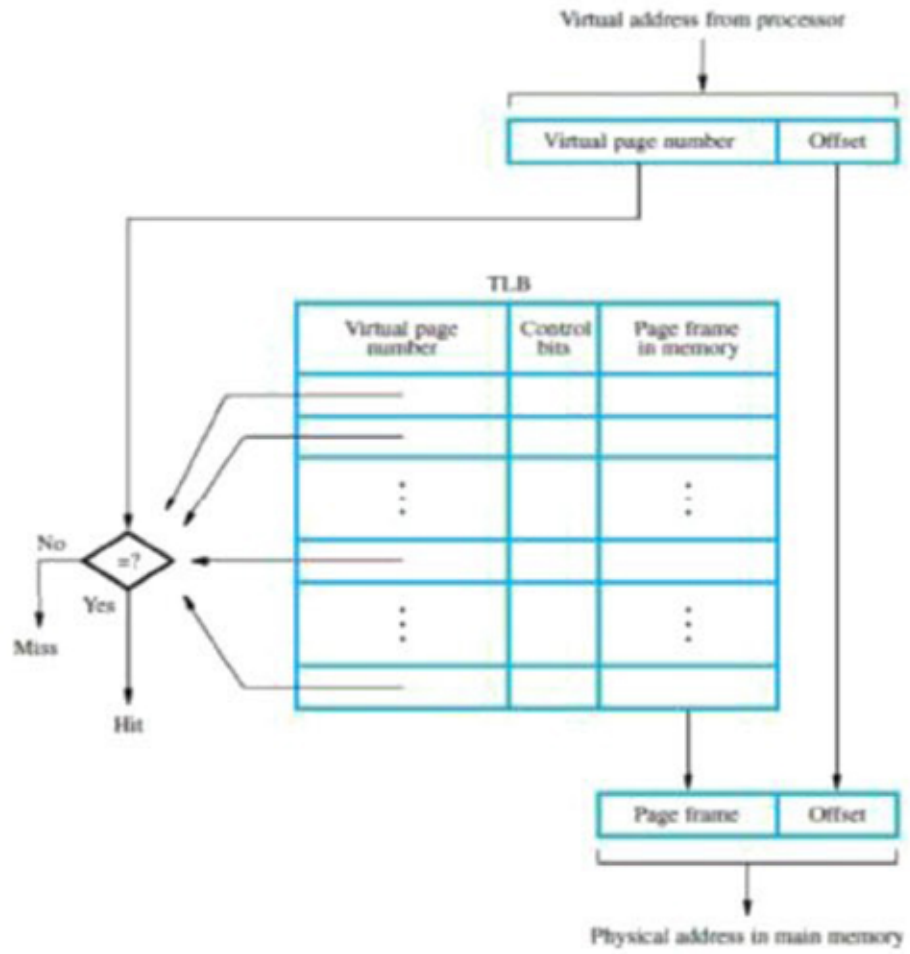


Figure 8.26 Use of an associative-mapped TLB.

➢ High-order bits of the virtual address generated by the processor select the virtual page.
➢ These bits are compared to the virtual page numbers in the TLB.
➢ If there is a match, a hit occurs and the corresponding address of the page frame is read.
➢ If there is no match, a miss occurs and the page table within the main memory must be consulted.
➢ Set-associative mapped TLBs are found in commercial processors.
➢ What happens if a program generates an access to a page that is not in the main memory?
➢ In this case, a ***page fault*** is said to occur.
- ✓ Whole page must be brought into the main memory from the disk, before the execution can proceed.

• Upon detecting a page fault by the MMU, following actions occur:
- ✓ MMU asks the operating system to intervene by raising an exception.
- ✓ Processing of the active task which caused the page fault is interrupted.
- ✓ Control is transferred to the operating system.
- ✓ Operating system copies the requested page from secondary storage to the main memory.
- ✓ Once the page is copied, control is returned to the task which was interrupted.

SECONDARY STORAGE

Magnetic Hard Disks

- The storage medium in a magnetic-disk system consists of one or more disk platters mounted on a common spindle. A thin magnetic film is deposited on each platter, usually on both sides.
- The assembly is placed in a drive that causes it to rotate at a constant speed. The magnetized surfaces move in close proximity to read/write heads, as shown in Figure 8.27*a*.
- Data are stored on concentric tracks, and the read/write heads move radially to access different tracks.
- Each read/write head consists of a magnetic yoke and a magnetizing coil, as indicated in Figure 8.27*b*.

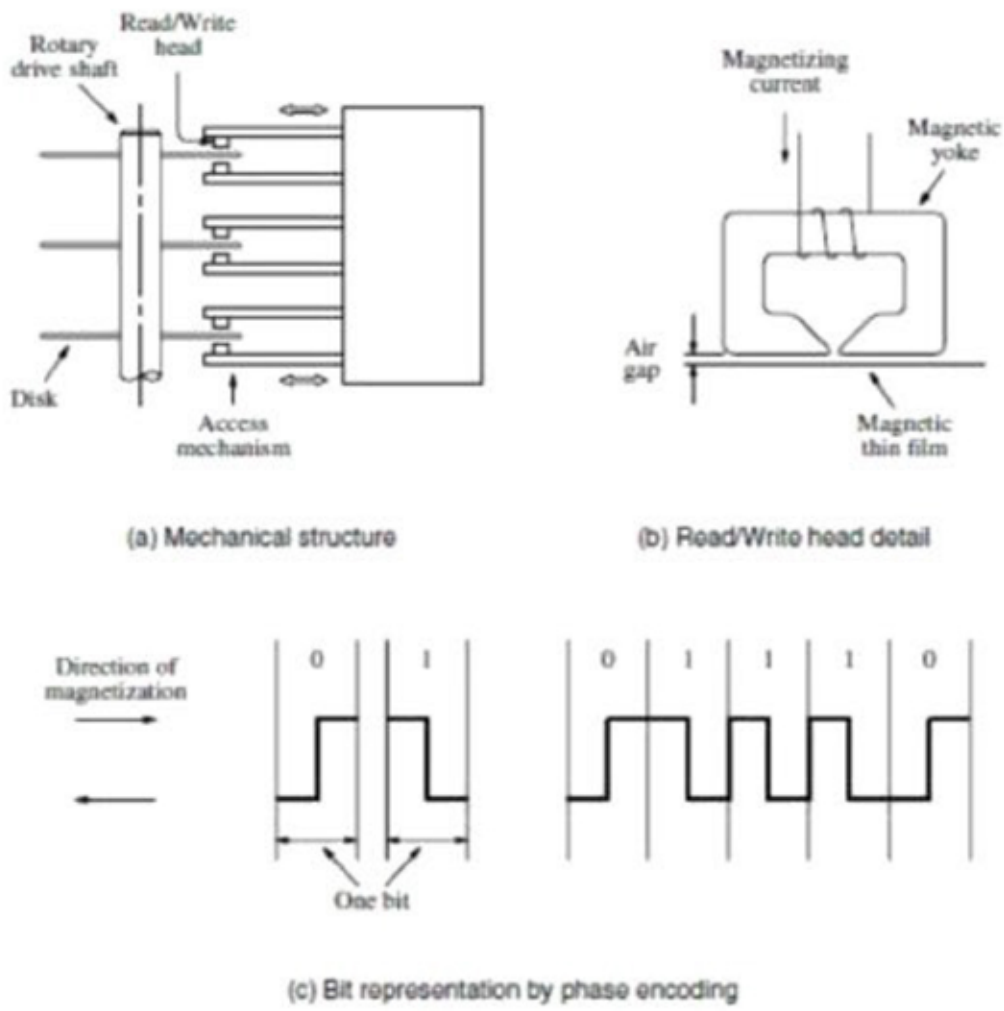


Figure 8.27 Magnetic disk principles.

- Digital information can be stored on the magnetic film by applying current pulses of suitable polarity to the magnetizing coil. This causes the magnetization of the film in the area immediately underneath the head to switch to a direction parallel to the applied field. The same head can be used for reading the stored information.
- Figure 8.27*c*, is known as ***phase encoding* or *Manchester encoding***. In this scheme, changes in magnetization occur for each data bit.
- Clocking information is provided by the change in magnetization at the midpoint of each bit period.
- The drawback of Manchester encoding is its poor bit-storage density. The space required to represent each bit must be large enough to accommodate two changes in magnetization.

- In most modern disk units, the disks and the read/write heads are placed in a sealed, air-filtered enclosure. This approach is known as ***Winchester technology***. In such units, the read/write heads can operate closer to the magnetized track surfaces, because dust particles, which are a problem in unsealed assemblies, are absent.
- Winchester disks have a larger capacity for a given physical size compared to unsealed units. Another advantage of Winchester technology is that data integrity tends to be greater in sealed units, where the storage medium is not exposed to contaminating elements.

The disk system consists of three key parts.

1. One part is the assembly of disk platters, which is usually referred to as the ***disk***.
2. The second part comprises the electromechanical mechanism that spins the disk and moves the read/write heads; it is called the ***disk drive***.
3. The third part is the ***disk controller***, which is the electronic circuitry that controls the operation of the system.

Organization and Accessing of Data on a Disk

The organization of data on a disk is illustrated in Figure 8.28.

- Each surface is divided into concentric *tracks*, and each track is divided into ***sectors***. The set of corresponding tracks on all surfaces of a stack of disks forms a logical ***cylinder***.
- All tracks of a cylinder can be accessed without moving the read/write heads. Data are accessed by specifying the surface number, the track number, and the sector number. Read and Write operations always start at sector boundaries.
- Data bits are stored serially on each track. Each sector may contain 512 or more bytes.
- The data are preceded by a ***sector header*** that contains identification (addressing) information used to find the desired sector on the selected track. Following the data, there are additional bits that constitute an ***error-correcting code (ECC).***

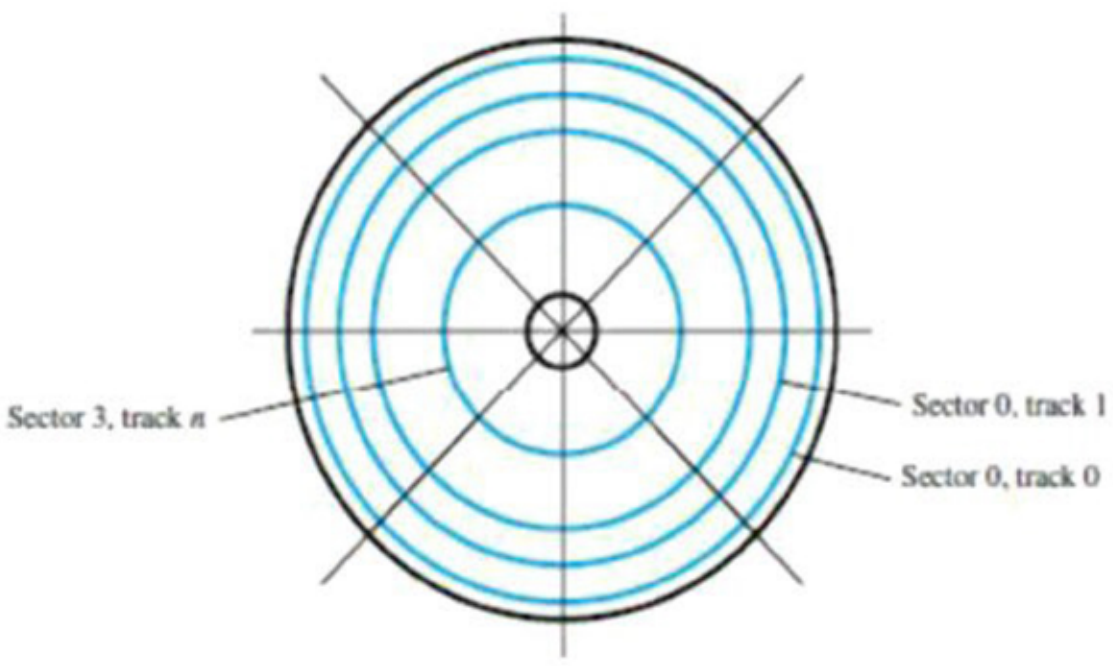


Figure 8.28 Organization of one surface of a disk.

Access Time

There are two components involved in the time delay between the disk receiving an address and the beginning of the actual data transfer.

1. The first, called the ***seek time***, is the time required to move the read/write head to the proper track. This time depends on the initial position of the head relative to the track specified in the address. Average values are in the 5 to 8-ms range.
2. The second component is the ***rotational delay***, also called ***latency time***, which is the time taken to reach the addressed sector after the read/write head is positioned over the correct track. On average, this is the time for half a rotation of the disk.
3. The sum of these two delays is called the disk ***access time***.

Data Buffer/Cache

A disk drive is connected to the rest of a computer system using some standard interconnection scheme, such as SCSI or SATA. The interconnection hardware is usually capable of transferring data at much higher rates than the rate at which data can be read from disk tracks. An efficient way to deal with the possible differences in transfer rates is to include a ***data buffer*** in the disk unit.

Disk Controller

- Operation of a disk drive is controlled by a *disk controller* circuit, which also provides an interface between the disk drive and the rest of the computer system.
- One disk controller may be used to control more than one drive.

The OS initiates

- The transfers by issuing Read and Write requests, which entail loading the controller's registers with the necessary addressing and control information. Typically, this information includes:
  - ✓ ***Main memory address***—the address of the first main memory location of the block of words involved in the transfer.
  - ✓ ***Disk address***—the location of the sector containing the beginning of the desired block of words.
  - ✓ ***Word count***—the number of words in the block to be transferred.

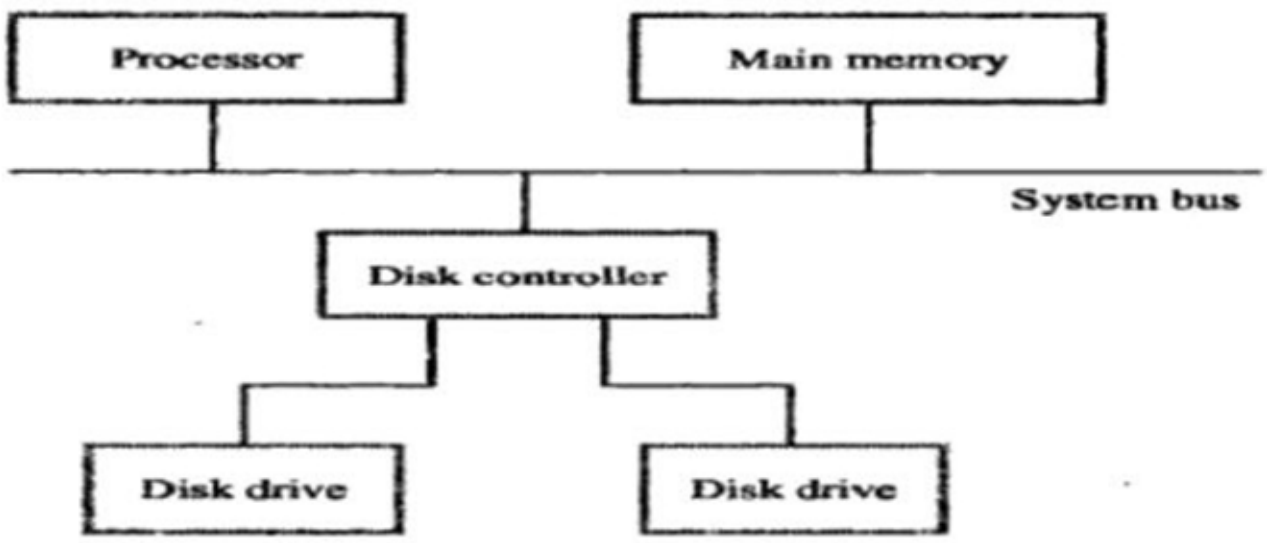


Figure 5.31 Disks connected to the system bus.

Floppy Disks

- *Floppy disks* are smaller, simpler, and cheaper disk units that consist of a flexible, removable, plastic ***diskette*** coated with magnetic material. The diskette is enclosed in a plastic jacket, which has an opening where the read/write head can be positioned.
- A hole in the center of the diskette allows a spindle mechanism in the disk drive to position and rotate the diskette.
- The main feature of floppy disks is their low cost and shipping convenience. However, they have much smaller storage capacities, longer access times, and higher failure rates than hard disks.

RAID Disk Arrays

One way to reduce access time is to use multiple disks operating in parallel. In 1988, researchers at the University of California-Berkeley proposed such a storage system. They called it RAID, for Redundant Array of Inexpensive Disks.

- Using multiple disks also makes it possible to improve the reliability of the overall system. Different configurations were proposed, and many more have been developed since.
- The basic configuration, known as RAID 0, is simple. A single large file is stored in several separate disk units by dividing the file into a number of smaller pieces and storing these pieces on different disks. This is called *data striping*.
- RAID 1 is intended to provide better reliability by storing identical copies of the data on two disks rather than just one. The two disks are said to be mirrors of each other. If one disk drive fails, all Read and Write operations are directed to its mirror drive.
- RAID2, 3, 4 – increased reliability
- RAID5 – parity-based error-recovery

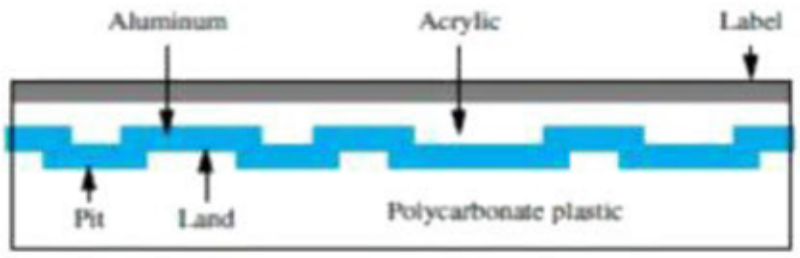
Optical Disks

Storage devices can also be implemented using optical means. The familiar compact disk (CD), used in audio systems, was the first practical application of this technology. Soon after, the optical technology was adapted to the computer environment to provide a high capacity read-only storage medium known as a CD-ROM.

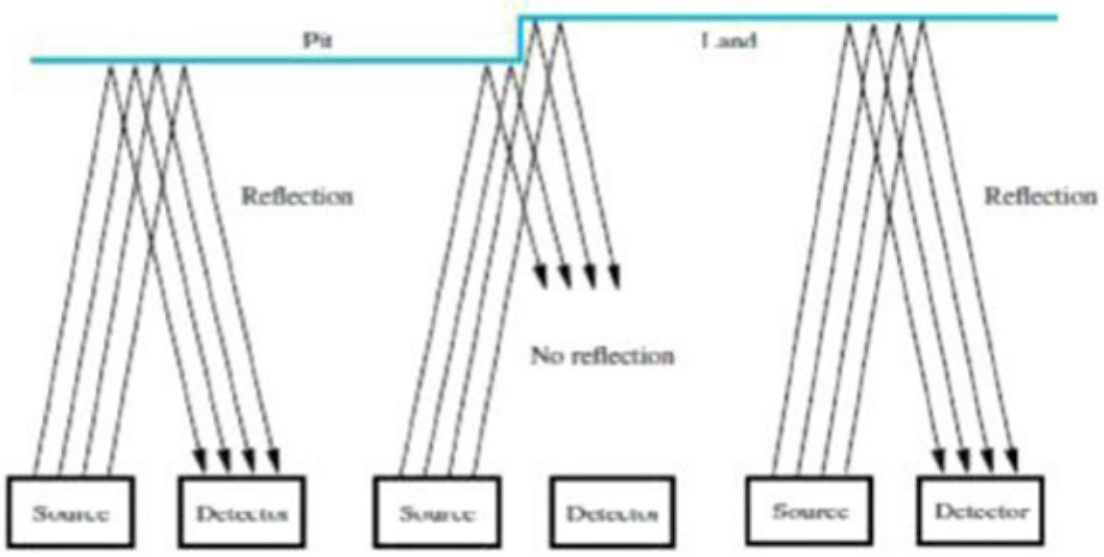
CD Technology

- The optical technology that is used for CD systems makes use of the fact that laser light can be focused on a very small spot.
- A laser beam is directed onto a spinning disk, with tiny indentations arranged to form a long spiral track on its surface. The indentations reflect the focused beam toward a photo detector, which detects the stored binary patterns.
- The laser emits a coherent light beam that is sharply focused on the surface of the disk. Coherent light consists of synchronized waves that have the same wavelength.
- If a coherent light beam is combined with another beam of the same kind, and the two beams are in phase, the result is a brighter beam.

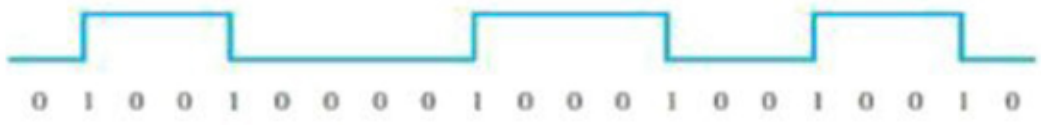
- But, if the waves of the two beams are 180 degrees out of phase, they cancel each other. Thus, a photo detector can be used to detect the beams.
- It will see a bright spot in the first case and a dark spot in the second case.
- Across-section of a small portion of a CD is shown in Figure 8.29*a*.
- The bottom layer is made of transparent polycarbonate plastic, which serves as a clear glass base.
- The surface of this plastic is programmed to store data by indenting it with *pits*. The unintended parts are called *lands*.



(a) Cross-section



(b) Transition from pit to land



(c) Stored binary pattern

Figure 8.29 Optical disk.

- A thin layer of reflecting aluminum material is placed on top of a programmed disk. The aluminum is then covered by a protective acrylic. Finally, the topmost layer is deposited and stamped with a label.
- The total thickness of the disk is 1.2 mm, almost all of it contributed by the polycarbonate plastic. The other layers are very thin.
- The laser source and the photo-detector are positioned below the polycarbonate plastic.
- The emitted beam travels through the plastic layer, reflects off the aluminum layer, and travels back toward the photo-detector.

- Figure 8.29*b* shows what happens as the laser beam scans across the disk and encounters a transition from a pit to a land.
- Three different positions of the laser source and the detector are shown, as would occur when the disk is rotating.
- When the light reflects solely from a pit, or from a land, the detector sees the reflected beam as a bright spot. But, a different situation arises when the beam moves over the edge between a pit and the adjacent land.
- Figure 8.29*c* depicts several transitions between lands and pits. If each transition, detected as a dark spot, is taken to denote the binary value 1, and the flat portions represent 0s, then the detected binary pattern will be as shown in the figure.
- CDs use a complex encoding scheme to represent data. Each byte of data is represented by a 14-bit code, which provides considerable error detection capability. We will not delve into details of this code.

CD-ROM

- The CDs used to store computer data are called *CD-ROMs*, because, like semiconductor ROM chips, their contents can only be read.
- Stored data are organized on CD-ROM tracks in the form of blocks called *sectors*.
- There are several different formats for a sector. One format, known as Mode 1, uses 2352 byte sectors. There is a 16-byte header that contains a synchronization field used to detect the beginning of the sector and addressing information used to identify the sector. This is followed by 2048 bytes of stored data. At the end of the sector, there are 288 bytes used to implement the error-correcting scheme. The number of sectors per track is variable; there are more sectors on the longer outer tracks.
- With the Mode 1 format, a CD-ROM has a storage capacity of about 650 Mbytes.
- For Error detection and correction, each byte of information stored on a CD is encoded using a 14-bit code that has some error-correcting capability. This code can correct single-bit errors.
- Errors that occur in short bursts, affecting several bits, are detected and corrected using the error-checking bits at the end of the sector.
- CD-ROM drives operate at a number of different rotational speeds. Form 1X to 56X CD-ROM.
- Their attraction lies in their small physical size, low cost, and ease of handling as a removable and transportable mass-storage medium.

CD-Recordable

CD-Recordable (CD-R) is a shiny spiral track covered by an organic dye is implemented on a disk during the manufacturing process. Then, a laser in a CD-R drive burns pits into the organic dye. The burned spots become opaque. They reflect less light than the shiny areas when the CD is being read. This process is irreversible, which means that the written data are stored permanently. Unused portions of a disk can be used to store additional data at a later time.

CD-Rewritable

- The most flexible CDs are those that can be written multiple times by the user. They are known as CD-RWs (CD-ReWritables).
- The basic structure of CD-RWs is similar to the structure of CD-Rs. Instead of using an organic dye in the recording layer, an alloy of silver, indium, antimony, and tellurium is used. This alloy has interesting and useful behavior when it is heated and cooled. If it is heated above its melting point (500 degrees C) and then cooled down, it goes into an amorphous state in which it absorbs light. But, if it is heated only to about 200 degrees C and this temperature is maintained for an extended period, a process known as ***annealing*** takes place, which leaves the alloy in a crystalline state that allows light to pass through.
- If the ***crystalline state*** represents land area, pits can be created by heating selected spots past the melting point. The stored data can be erased using the annealing process, which returns the alloy to a uniform crystalline state. A reflective material is placed above the recording layer to reflect the light when the disk is read.
- A CD-RW drive uses three different laser powers. The highest power is used to record the pits. The middle power is used to put the alloy into its crystalline state; it is referred to as the "***erase power***." The lowest power is used to read the stored information. CD drives designed to read and write CD-RW disks can usually be used with other compact disk media.

DVD Technology

The success of CD technology and the continuing quest for greater storage capability has led to the development of DVD (Digital Versatile Disk) technology.

Its storage capacity is made much larger than that of CDs by several design changes:

- A red-light laser with a wavelength of 635 nm is used instead of the infrared light laser used in CDs, which has a wavelength of 780 nm. The shorter wavelength makes it possible to focus the light to a smaller spot.
- Pits are smaller, having a minimum length of 0.4 micron.
- Tracks are placed closer together; the distance between tracks is 0.74 micron.

Using these improvements leads to a DVD capacity of 4.7 Gbytes.

MAGNETIC TAPE SYSTEMS

Magnetic tapes are suited for off-line storage of large amounts of data. They are typically used for backup purposes and for archival storage. Magnetic-tape recording uses the same principle as magnetic disks.

The main difference is that the magnetic film is deposited on a very thin 0.5- or 0.25-inch wide plastic tape. Seven or nine bits (corresponding to one character) are recorded in parallel across the width of the tape, perpendicular to the direction of motion. A separate read/write head is provided for each bit position on the tape, so that all bits of a character can be read or written in parallel. One of the character bits is used as a parity bit.

- Data on the tape are organized in the form of ***records*** separated by gaps, as shown in Figure 8.30.

- Tape motion is stopped only when a record gap is underneath the read/write heads.
- The record gaps are long enough to allow the tape to attain its normal speed before the beginning of the next record is reached.
- The large amounts of data, a group of related records is called a ***file***. The beginning of a file is identified by a ***file mark***, as shown in Figure 8.30.
- The file mark is a special single- or multiple-character record, usually preceded by a gap longer than the inter-record gap.
- The first record following a file mark can be used as a ***header*** **or** ***identifier*** for the file. This allows the user to search a tape containing a large number of files for a particular file.

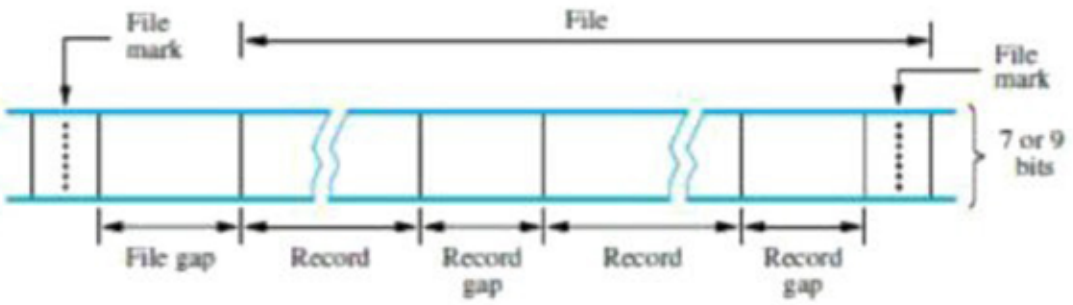


Figure 8.30 Organization of data on magnetic tape.