DEPARTMENT
OF
ELECTRONICS & COMMUNICATION ENGINEERING

Digital System Design with Verilog

(Theory Notes)

Autonomous Course

Prepared by

Prof.Manasa R

Module – 4 Memory Organization
Basic Concepts, Semiconductor RAM Memories, Speed, Size, and Cost, Cache Memories – Mapping Functions, Replacement Algorithms, Performance Considerations and Virtual Memories.

Dayananda Sagar College of Engineering

Shavige Malleshwara Hills, Kumaraswamy Layout,
Banashankari, Bangalore-560078, Karnataka
Tel : +91 80 26662226 26661104 Extn : 2731 Fax : +90 80 2666 0789
Web - http://www.dayanandasagar.edu Email : hod-ece@dayanandasagar.edu
(An Autonomous Institute Affiliated to VTU, Approved by AICTE & ISO 9001:2008 Certified)
(Accredited by NBA, National Assessment & Accreditation Council (NAAC) with 'A' grade)

MODULE 4: MEMORY SYSTEM

4.1 BASIC CONCEPTS

• Maximum size of memory that can be used in any computer is determined by addressing mode.

Address	Memory Locations
16 Bit	$2^{16} = 64$ K
32 Bit	$2^{32} = 4$G (Giga)
40 Bit	$2^{40} =$ IT (Tera)

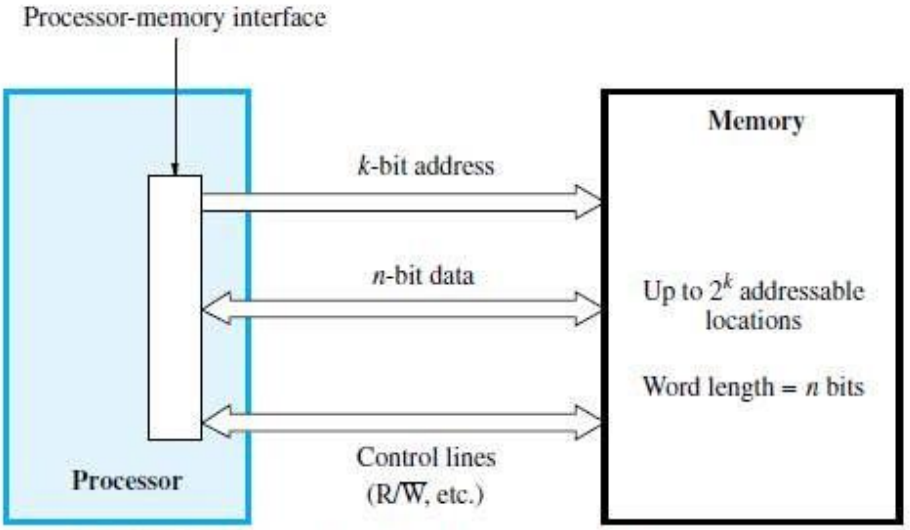


Fig 4.1 Connection of the memory to the processor.

• If MAR is k-bits long then
→ memory may contain upto 2^K addressable-locations
• If MDR is n-bits long, then
→ n-bits of data are transferred between the memory and processor.
• The data-transfer takes place over the processor-bus (Figure 8.1).
• The processor-bus has
1) Address-Line
2) Data-line &
3) Control-Line (R/W", MFC – Memory Function Completed).
• The Control-Line is used for coordinating data-transfer.
• The processor reads the data from the memory by
→ loading the address of the required memory-location into MAR and
→ setting the R/W" line to 1.
• The memory responds by
→ placing the data from the addressed-location onto the data-lines and
→ confirms this action by asserting MFC signal.
• Upon receipt of MFC signal, the processor loads the data from the data-lines into MDR.
• The processor writes the data into the memory-location by
→ loading the address of this location into MAR &
→ setting the R/W" line to 0.
• **Memory Access Time:** It is the time that elapses between
→ initiation of an operation &
→ completion of that operation.
• **Memory Cycle Time:** It is the minimum time delay that required between the initiation of the two successive memory-operations.

4.1.1RAM (Random Access Memory)

- In RAM, any location can be accessed for a Read/Write-operation in fixed amount of time,

Cache Memory

It is a small, fast memory that is inserted between

→ larger slower main-memory and

→ processor.

It holds the currently active segments of a program and their data.

Virtual Memory

The address generated by the processor is referred to as a **virtual/logical address**.

The virtual-address-space is mapped onto the physical-memory where data are actually stored.

The mapping-function is implemented by MMU. (MMU = memory management unit).

Only the active portion of the address-space is mapped into locations in the physical-memory.

The remaining virtual-addresses are mapped onto the bulk storage devices such as magnetic disk.

As the active portion of the virtual-address-space changes during program execution, the MMU

→ changes the mapping-function &

→ transfers the data between disk and memory.

During every memory-cycle, MMU determines whether the addressed-page is in the memory.

If the page is in the memory.

Then, the proper word is accessed and execution proceeds.

Otherwise, a page containing desired word is transferred from disk to memory.

- Memory can be classified as follows:
  1) RAM which can be further classified as follows:
     i) Static RAM
     ii) Dynamic RAM (DRAM) which can be further classified as synchronous & asynchronous DRAM.
  2) ROM which can be further classified as follows:
     i) PROM
     ii) EPROM
     iii) EEPROM &
     iv) Flash Memory which can be further classified as Flash Cards & Flash Drives.

4.2SEMI CONDUCTOR RAM MEMORIES

4.2.1INTERNAL ORGANIZATION OF MEMORY-CHIPS

- Memory-cells are organized in the form of array (Figure 8.2).
- Each cell is capable of storing 1-bit of information.
- Each row of cells forms a memory-word.
- All cells of a row are connected to a common line called as **Word-Line**.
- The cells in each column are connected to **Sense/Write** circuit by 2-bit-lines.
- The Sense/Write circuits are connected to data-input or output lines of the chip.
- During a write-operation, the sense/write circuit
  - → receive input information &
  - → store input info in the cells of the selected word.

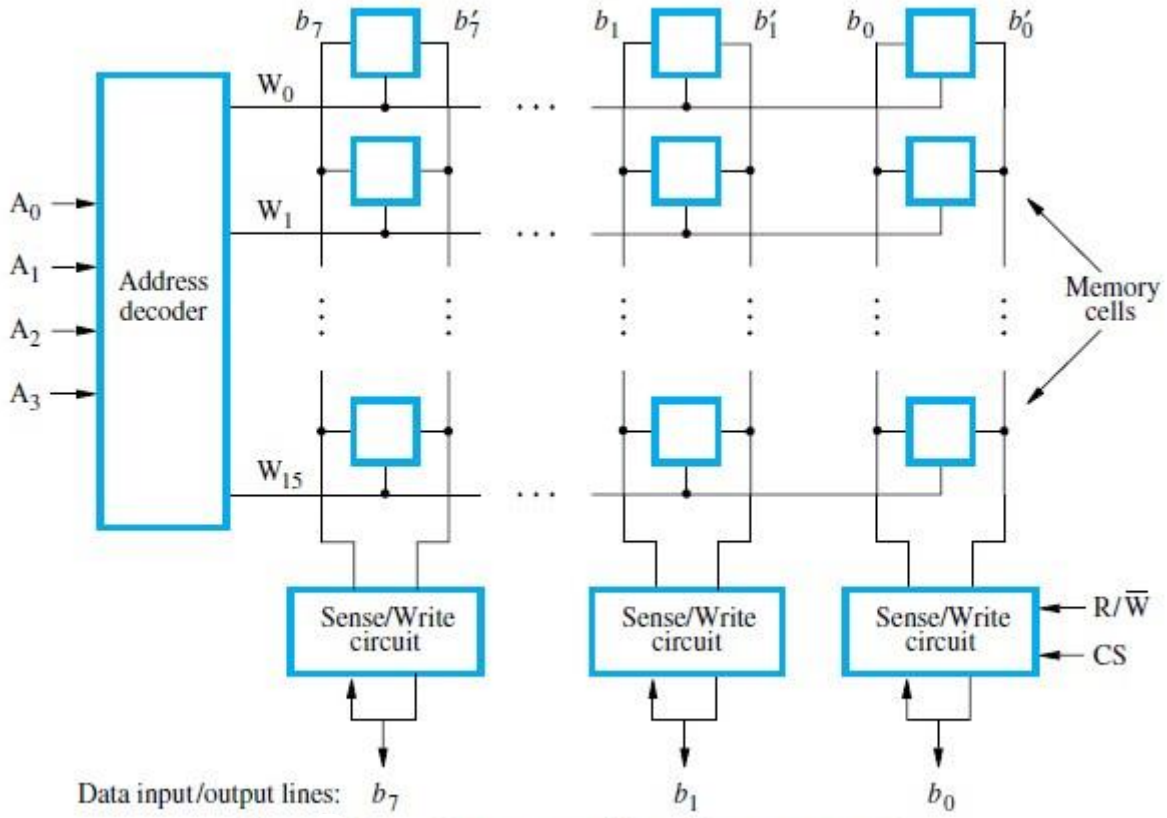


Fig 4.2 Organization of bit cells in a memory chip.

- The data-input and data-output of each Sense/Write circuit are connected to a single bidirectional data-line.
- Data-line can be connected to a data-bus of the computer.
- Following 2 control lines are also used:
  - **1) R/W'** Specifies the required operation.
  - **2) CS'** Chip Select input selects a given chip in the multi-chip memory-system.

Bit Organization	Requirement of external connection for address, data and control lines
128 (16x8)	14
(1024) 128x8(1k)	19

4.2.2 STATIC RAM (OR MEMORY)

• Memories consist of circuits capable of retaining their state as long as power is applied are known**.**

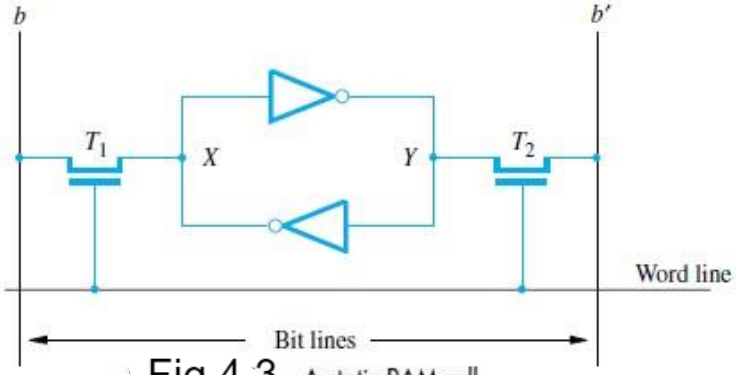


Fig 4.3 A static RAM cell.

• Two inverters are cross connected to form a latch (Figure 8.4).
• The latch is connected to 2-bit-lines by transistors T_1 and T_2.
• The transistors act as switches that can be opened/closed under the control of the word-line.
• When the word-line is at ground level, the transistors are turned off and the latch retain its state.

Read Operation

• To read the state of the cell, the word-line is activated to close switches T_1 and T_2.
• If the cell is in state 1, the signal on bit-line b is high and the signal on the bit-line b‟ is low.
• Thus, b and b‟ are complement of each other.
• Sense/Write circuit
  → monitors the state of b & b‟ and
  → sets the output accordingly.

Write Operation

• The state of the cell is set by
  → placing the appropriate value on bit-line b and its complement on b‟ and
  → then activating the word-line. This forces the cell into the corresponding state.
• The required signal on the bit-lines is generated by Sense/Write circuit.

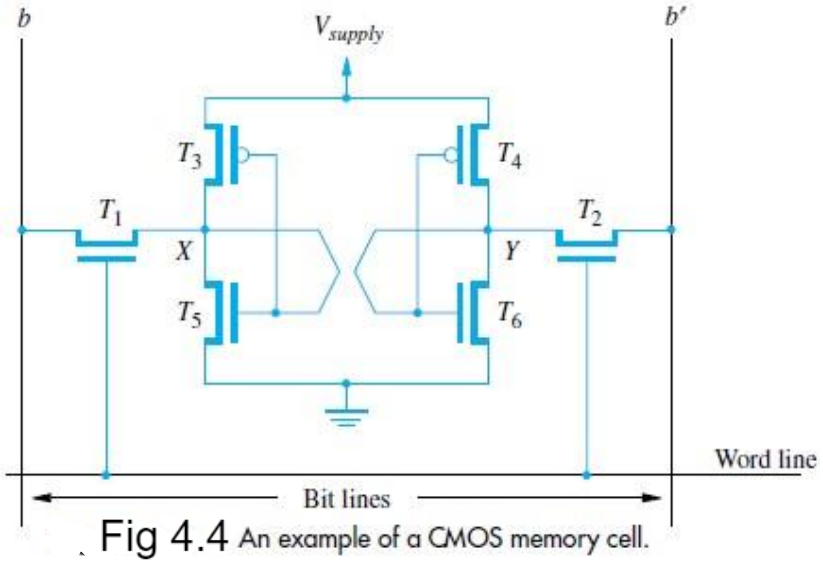


Fig 4.4 An example of a CMOS memory cell.

CMOS Cell

• Transistor pairs (T_3, T_5) and (T_4, T_6) form the inverters in the latch (Figure 8.5).
• In state 1, the voltage at point X is high by having T_5, T_6 ON and T_4, T_5 are OFF.
• Thus, T_1 and T_2 returned ON (Closed), bit-line b and b‟ will have high and low signals respectively.
• **Advantages:**
  1) It has low power consumption „.“ the current flows in the cell only when the cell is active.
  2) Static RAM‟s can be accessed quickly. It access time is few nanoseconds.
• **Disadvantage:** SRAMs are said to be volatile memories „.“ their contents are lost when power is interrupted.

4.2.3 ASYNCHRONOUS DRAM

- Less expensive RAMs can be implemented if simple cells are used.
- Such cells cannot retain their state indefinitely. Hence they are called **Dynamic RAM (DRAM).**
- The information stored in a dynamic memory-cell in the form of a charge on a capacitor.
- This charge can be maintained only for tens of milliseconds.
- The contents must be periodically refreshed by restoring this capacitor charge to its full value.

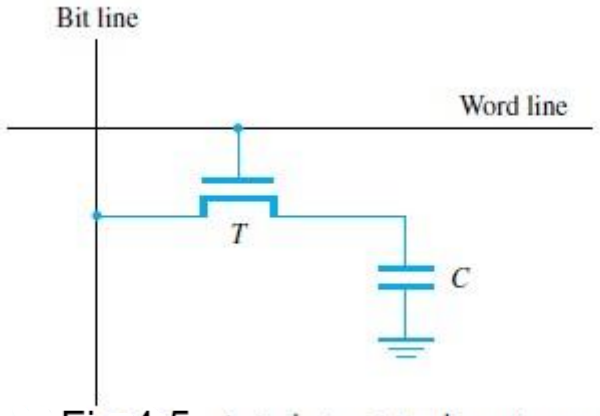
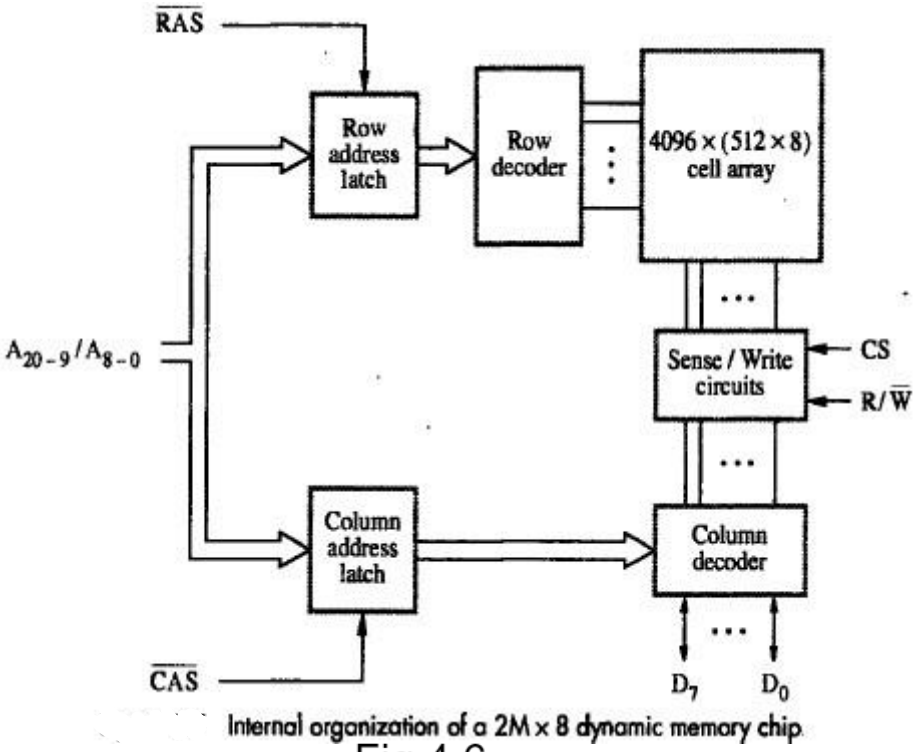


Fig 4.5 A single-transistor dynamic memory cell.

- In order to store information in the cell, the transistor T is turned „ON" (Figure 8.6).
- The appropriate voltage is applied to the bit-line which charges the capacitor.
- After the transistor is turned off, the capacitor begins to discharge.
- Hence, info. stored in cell can be retrieved correctly before threshold value of capacitor drops down.
- During a read-operation,
  - → transistor is turned „ON"
  - → a sense amplifier detects whether the charge on the capacitor is above the threshold value.
    - If (charge on capacitor) > (threshold value) Bit-line will have logic value „1".
    - If (charge on capacitor) < (threshold value) Bit-line will set to logic value „0".

ASYNCHRONOUS DRAM DESCRIPTION

- The 4 bit cells in each row are divided into 512 groups of 8 (Figure 5.7).
- 21 bit address is needed to access a byte in the memory. 21 bit is divided as follows:
  1) 12 address bits are needed to select a row.
     i.e. A_{8-0} → specifies row-address of a byte.
  2) 9 bits are needed to specify a group of 8 bits in the selected row.
     i.e. A_{20-9} → specifies column-address of a byte.



Internal organization of a 2M × 8 dynamic memory chip

Fig 4.6

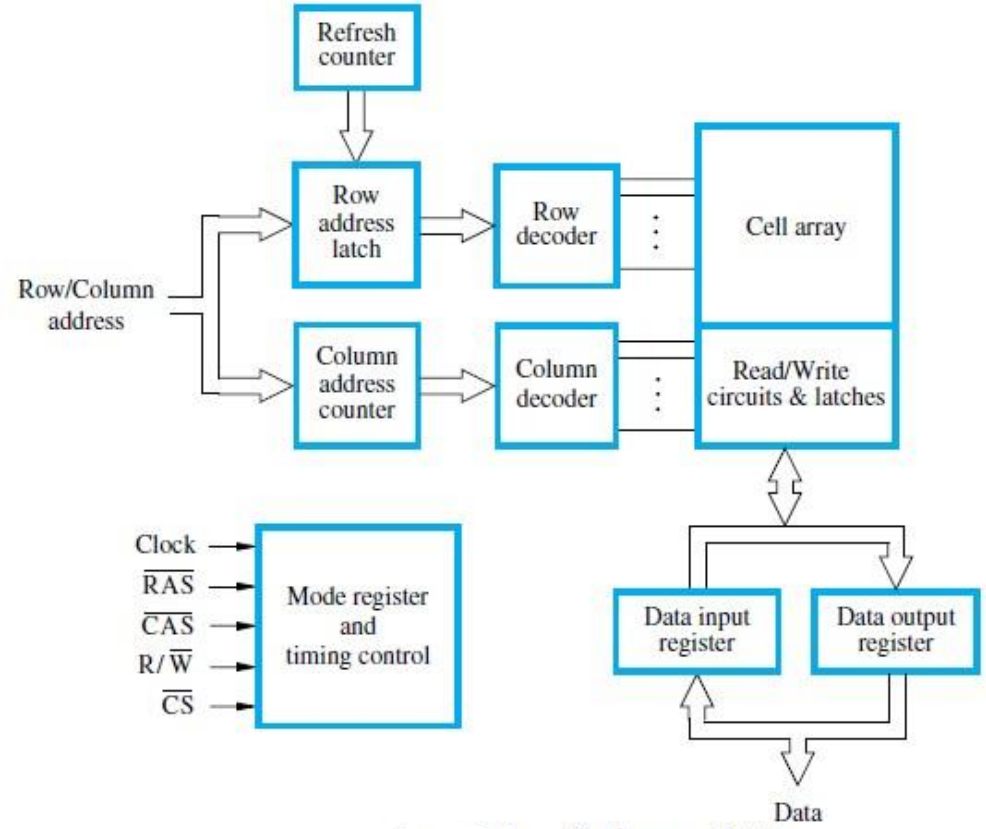
- During Read/Write-operation,
  → row-address is applied first.
  → row-address is loaded into row-latch in response to a signal pulse on **RAS'** input of chip.
  (RAS = Row-address Strobe CAS = Column-address Strobe)
- When a Read-operation is initiated, all cells on the selected row are read and refreshed.
- Shortly after the row-address is loaded, the column-address is
  → applied to the address pins &
  → loaded into **CAS'.**
- The information in the latch is decoded.
- The appropriate group of 8 Sense/Write circuits is selected.
  R/W'=1(read-operation) Output values of selected circuits are transferred to data-lines D_0-D_7.
  R/W'=0(write-operation) Information on D_0-D_7 are transferred to the selected circuits.
- RAS″ & CAS″ are active-low so that they cause latching of address when they change from high to low.
- To ensure that the contents of DRAMs are maintained, each row of cells is accessed periodically.
- A special memory-circuit provides the necessary control signals RAS″ & CAS″ that govern the timing.
- The processor must take into account the delay in the response of the memory.

Fast Page Mode

Transferring the bytes in sequential order is achieved by applying the consecutive sequence of column-address under the control of successive CAS″ signals.
This scheme allows transferring a block of data at a faster rate.
The block of transfer capability is called as *fast page mode***.**

4.2.4 SYNCHRONOUS DRAM

- The operations are directly synchronized with clock signal (Figure 8.8).
- The address and data connections are buffered by means of registers.
- The output of each sense amplifier is connected to a latch.
- A Read-operation causes the contents of all cells in the selected row to be loaded in these latches.
- Data held in latches that correspond to selected columns are transferred into data-output register.
- Thus, data becoming available on the data-output pins.



Synchronous DRAM.

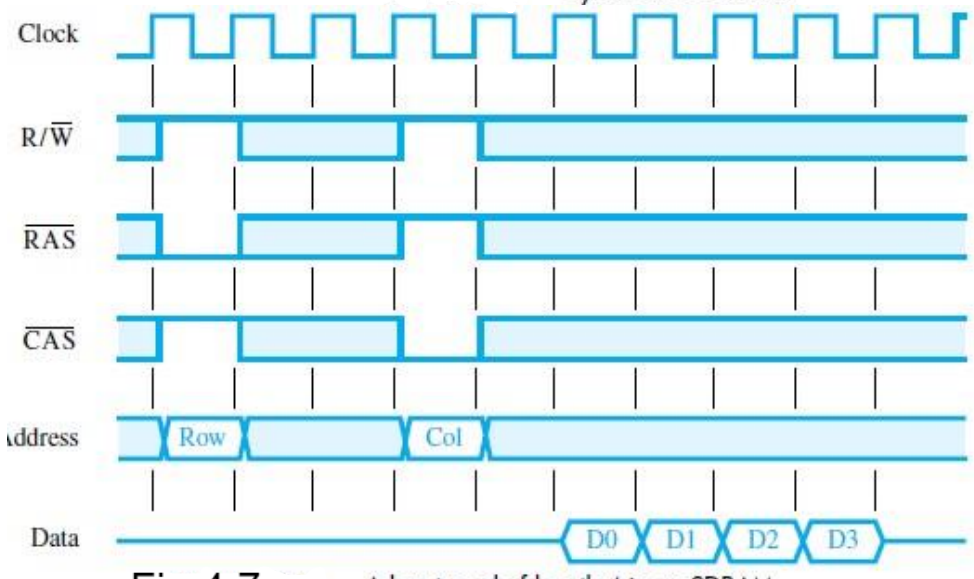


Fig 4.7 A burst read of length 4 in an SDRAM.

- First, the row-address is latched under control of RAS" signal (Figure 8.9).
- The memory typically takes 2 or 3 clock cycles to activate the selected row.
- Then, the column-address is latched under the control of CAS" signal.
- After a delay of one clock cycle, the first set of data bits is placed on the data-lines.
- SDRAM automatically increments column-address to access next 3 sets of bits in the selected row.

LATENCY & BANDWIDTH

• A good indication of performance is given by 2 parameters: 1) Latency 2) Bandwidth.

Latency

• It refers to the amount of time it takes to transfer a word of data to or from the memory.
• For a transfer of single word, the latency provides the complete indication of memory performance.
• For a block transfer, the latency denotes the time it takes to transfer the first word of data.

Bandwidth

• It is defined as the number of bits or bytes that can be transferred in one second.
• Bandwidth mainly depends on
    1) The speed of access to the stored data &
    2) The number of bits that can be accessed in parallel.

DOUBLE DATA RATE SDRAM (DDR-SDRAM)

• The standard SDRAM performs all actions on the rising edge of the clock signal.
• The DDR-SDRAM transfer data on both the edges (loading edge, trailing edge).
• The Bandwidth of DDR-SDRAM is doubled for long burst transfer.
• To make it possible to access the data at high rate, the cell array is organized into two banks.
• Each bank can be accessed separately.
• Consecutive words of a given block are stored in different banks.
• Such interleaving of words allows simultaneous access to two words.
• The two words are transferred on successive edge of the clock.

4.2.5 STRUCTURE OF LARGER MEMORIES Dynamic Memory System

• The physical implementation is done in the form of memory-modules.
• If a large memory is built by placing DRAM chips directly on the Motherboard, then it will occupy large amount of space on the board.
• These packaging consideration have led to the development of larger memory units known as SIMM"s & DIMM"s.
    1) SIMM Single Inline memory-module
    2) DIMM Dual Inline memory-module
• SIMM/DIMM consists of many memory-chips on small board that plugs into a socket on motherboard.

4.2.6 MEMORY-SYSTEM CONSIDERATION MEMORY CONTROLLER

- To reduce the number of pins, the dynamic memory-chips use multiplexed-address inputs.
- The address is divided into 2 parts:
  - **1) High Order Address Bit**
    Select a row in cell array.
    It is provided first and latched into memory-chips under the control of RAS" signal.
  - **2) Low Order Address Bit**
    Selects a column.
    They are provided on same address pins and latched using CAS" signals.
- The Multiplexing of address bit is usually done by **Memory Controller Circuit** (Figure 5.11).

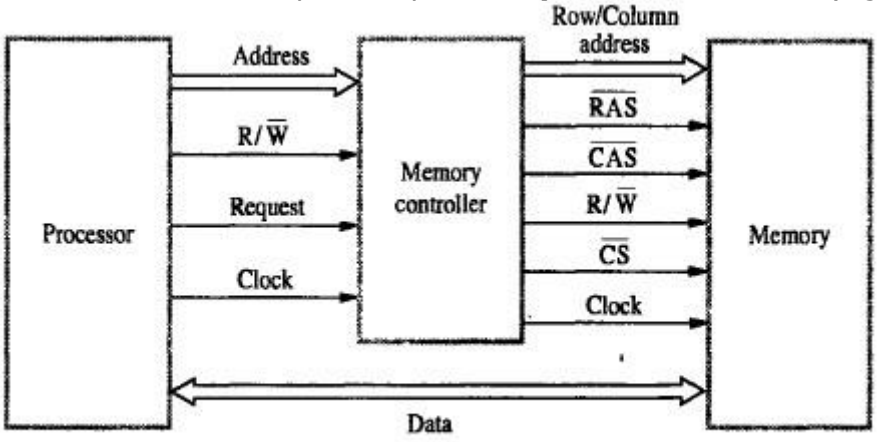


Fig 4.8 Use of a memory controller.

- The Controller accepts a complete address & R/W" signal from the processor.
- A Request signal indicates a memory access operation is needed.
- Then, the Controller
  - → forwards the row & column portions of the address to the memory.
  - → generates RAS" & CAS" signals &
  - → sends R/W" & CS" signals to the memory.

RAMBUS MEMORY

- The usage of wide bus is expensive.
- Rambus developed the implementation of narrow bus.
- Rambus technology is a fast signaling method used to transfer information between chips.
- The signals consist of much smaller voltage swings around a reference voltage V_{ref}.
- The reference voltage is about 2V.
- The two logical values are represented by 0.3V swings above and below V_{ref}.
- This type of signaling is generally is known as **Differential Signalling**.
- Rambus provides a complete specification for design of communication called as **Rambus Channel**.
- Rambus memory has a clock frequency of 400 MHz.
- The data are transmitted on both the edges of clock so that effective data-transfer rate is 800MHZ.
- Circuitry needed to interface to Rambus channel is included on chip. Such chips are called **RDRAM.**
  (RDRAM = Rambus DRAMs).
- Rambus channel has:
  1) 9 Data-lines (1^{st}-8^{th} line ->Transfer the data, 9^{th} line->Parity checking).
  2) Control-Line &
  3) Power line.
- A two channel rambus has 18 data-lines which has no separate Address-Lines.
- Communication between processor and RDRAM modules is carried out by means of packets transmitted on the data-lines.
- There are 3 types of packets:
  1) Request
  2) Acknowledge &
  3) Data

4.3 SPEED, SIZE COST

Characteristics	SRAM	DRAM	Magnetis Disk
Speed	Very Fast	Slower	Much slower than DRAM
Size	Large	Small	Small
Cost	Expensive	Less Expensive	Low price

Memory	Speed	Size	Cost
Registers	Very high	Lower	Very Lower
Primary cache	High	Lower	Low
Secondary cache	Low	Low	Low
Main memory	Lower than Seconadry cache	High	High
Secondary Memory	Very low	Very High	Very High

- The main-memory can be built with DRAM (Figure 8.14)
- Thus, SRAM"s are used in smaller units where speed is of essence.
- The Cache-memory is of 2 types:
  - **1) Primary/Processor Cache** (Level1 or L1 cache)
    It is always located on the processor-chip.
  - **2) Secondary Cache** (Level2 or L2 cache)
    It is placed between the primary-cache and the rest of the memory.
- The memory is implemented using the dynamic components (SIMM, RIMM, DIMM).
- The access time for main-memory is about 10 times longer than the access time for L1 cache.

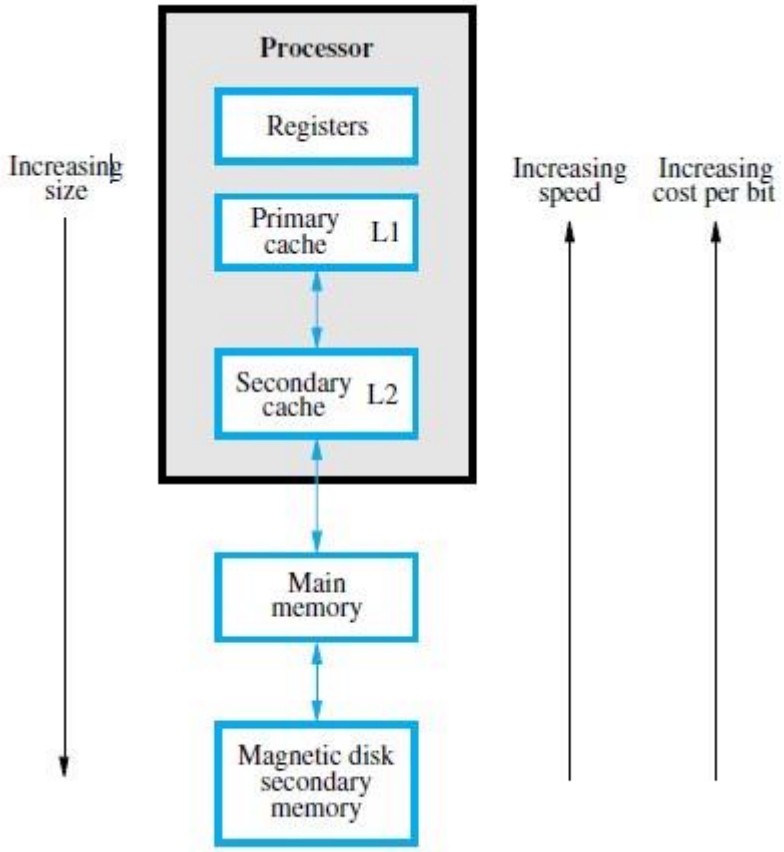


Fig 4.9 Memory hierarchy.

4.4 CACHE MEMORIES

• The effectiveness of cache mechanism is based on the property of „**Locality of Reference'.**

Locality of Reference

• Many instructions in the localized areas of program are executed repeatedly during some time period
• Remainder of the program is accessed relatively infrequently (Figure 8.15).
• There are 2 types:

1) Temporal

The recently executed instructions are likely to be executed again very soon.

2) Spatial

Instructions in close proximity to recently executed instruction are also likely to be executed soon.

• If active segment of program is placed in cache-memory, then total execution time can be reduced.
• **Block** refers to the set of contiguous address locations of some size.
• The cache-line is used to refer to the cache-block.

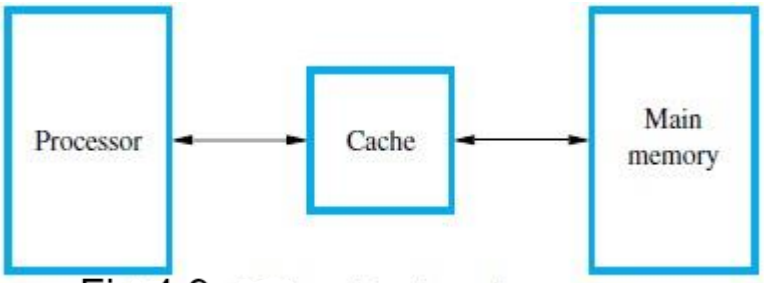


Fig 4.9 Use of a cache memory.

• The Cache-memory stores a reasonable number of blocks at a given time.
• This number of blocks is small compared to the total number of blocks available in main-memory.
• Correspondence b/w main-memory-block & cache-memory-block is specified by mapping-function.
• Cache control hardware decides which block should be removed to create space for the new block.
• The collection of rule for making this decision is called the **Replacement Algorithm.**
• The cache control-circuit determines whether the requested-word currently exists in the cache.
• The write-operation is done in 2 ways: 1) Write-through protocol & 2) Write-back protocol.

Write-Through Protocol

Here the cache-location and the main-memory-locations are updated simultaneously.

Write-Back Protocol

This technique is to

→ update only the cache-location &

→ mark the cache-location with associated flag bit called **Dirty/Modified Bit.**

The word in memory will be updated later, when the marked-block is removed from cache.

During Read-operation

• If the requested-word currently not exists in the cache, then **read-miss** will occur.
• To overcome the read miss, *Load–through/Early restart protocol* is used.

Load–Through Protocol

The block of words that contains the requested-word is copied from the memory into cache.
After entire block is loaded into cache, the requested-word is forwarded to processor.

During Write-operation

• If the requested-word not exists in the cache, then **write-miss** will occur.

1) If **Write Through Protocol** is used, the information is written directly into main-memory.
2) If **Write Back Protocol** is used,

→ then block containing the addressed word is first brought into the cache &

→ then the desired word in the cache is over-written with the new information.

4.4.1 MAPPING-FUNCTION

• Here we discuss about 3 different mapping-function:

1) Direct Mapping
2) Associative Mapping
3) Set-Associative Mapping

DIRECT MAPPING

- The block-j of the main-memory maps onto block-j modulo-128 of the cache (Figure 8.16).
- When the memory-blocks 0, 128, & 256 are loaded into cache, the block is stored in cache-block 0.
  Similarly, memory-blocks 1, 129, 257 are stored in cache-block 1.
- The contention may arise when
  1) When the cache is full.
  2) When more than one memory-block is mapped onto a given cache-block position.
- The contention is resolved by
  allowing the new blocks to overwrite the currently resident-block.
- Memory-address determines placement of block in the cache.

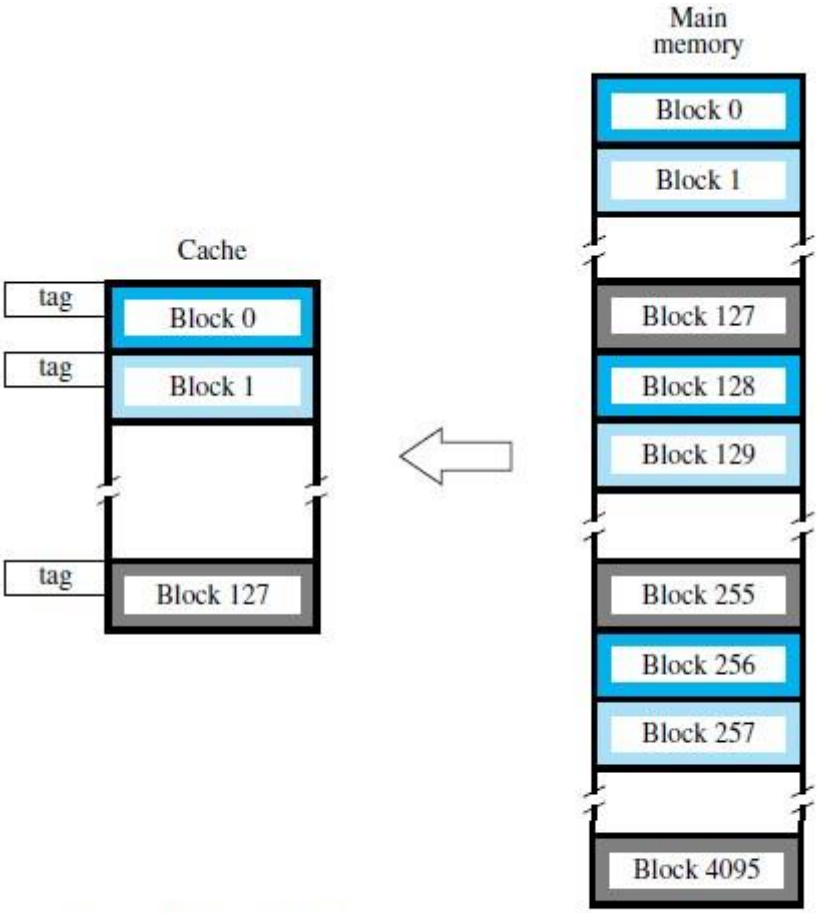


Fig 4.10

- The memory-address is divided into 3 fields:

1) Low Order 4 bit field

Selects one of 16 words in a block.

2) 7 bit cache-block field

7-bits determine the cache-position in which new block must be stored.

3) 5 bit Tag field

5-bits memory-address of block is stored in 5 tag-bits associated with cache-location.

- As execution proceeds,
  5-bit tag field of memory-address is compared with tag-bits associated with cache-location.
  If they match, then the desired word is in that block of the cache.
  Otherwise, the block containing required word must be first read from the memory.
  And then the word must be loaded into the cache.

ASSOCIATIVE MAPPING

- The memory-block can be placed into any cache-block position. (Figure 8.17).
- 12 tag-bits will identify a memory-block when it is resolved in the cache.
- Tag-bits of an address received from processor are compared to the tag-bits of each block of cache.
- This comparison is done to see if the desired block is present.

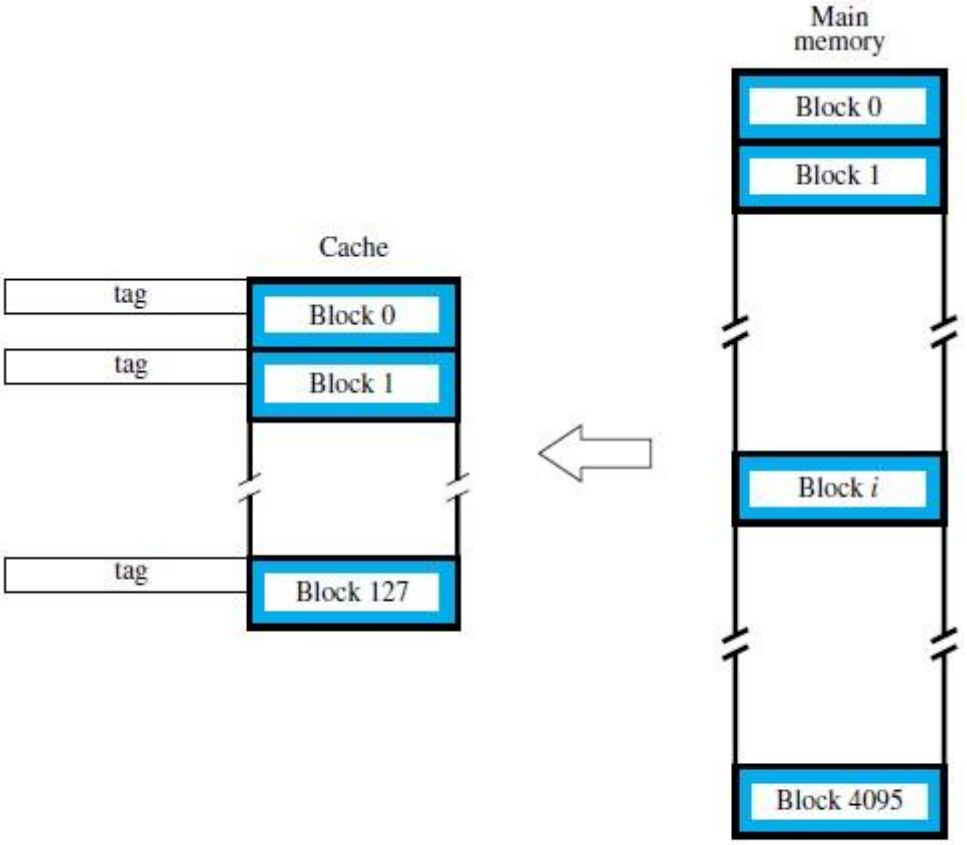


Fig 4.11 Associative-mapped cache.

- It gives complete freedom in choosing the cache-location.
- A new block that has to be brought into the cache has to replace an existing block if the cache is full.
- The memory has to determine whether a given block is in the cache.
- **Advantage:** It is more flexible than direct mapping technique.
- **Disadvantage:** Its cost is high.

SET-ASSOCIATIVE MAPPING

- It is the combination of direct and associative mapping. (Figure 8.18).
- The blocks of the cache are grouped into sets.
- The mapping allows a block of the main-memory to reside in any block of the specified set.
- The cache has 2 blocks per set, so the memory-blocks 0, 64, 128…….. 4032 maps into cache set „0".
- The cache can occupy either of the two block position within the set.

6 bit set field

Determines which set of cache contains the desired block.

6 bit tag field

The tag field of the address is compared to the tags of the two blocks of the set.
This comparison is done to check if the desired block is present.

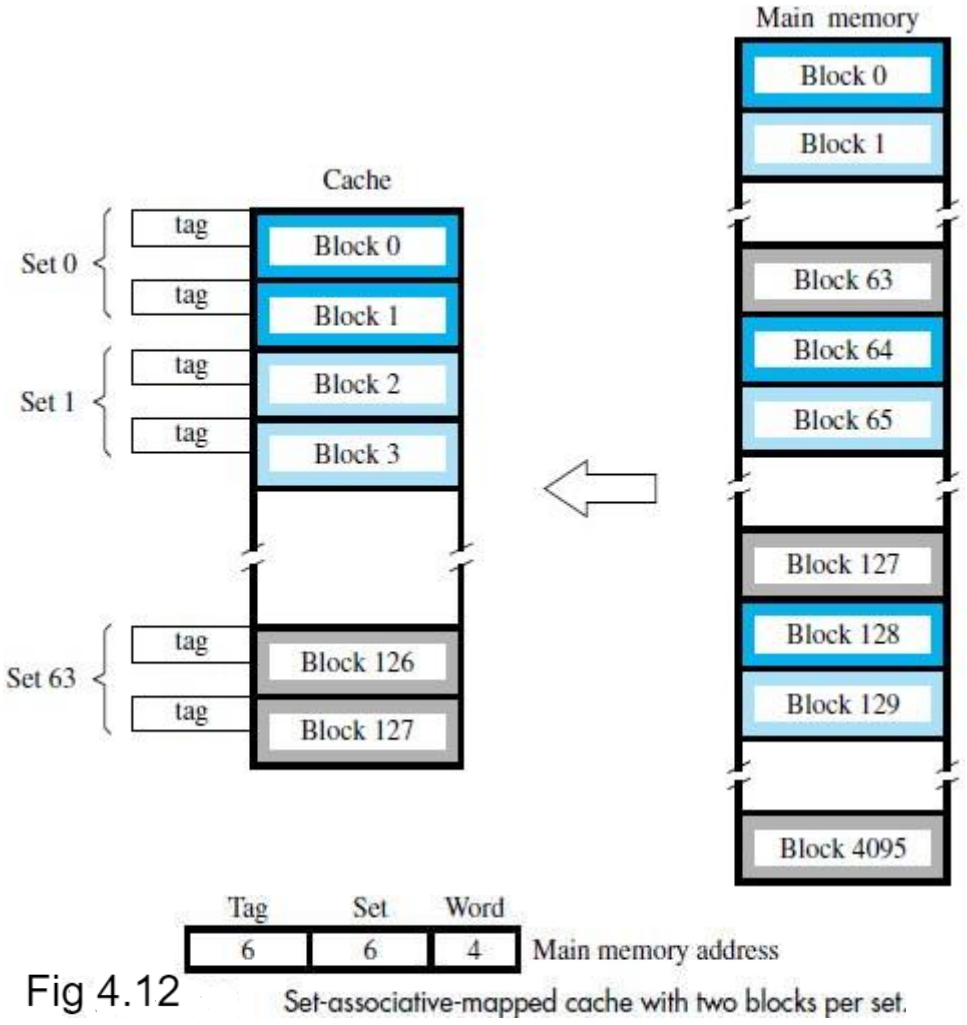


Fig 4.12 Set-associative-mapped cache with two blocks per set.

- The cache which contains 1 block per set is called **direct mapping.**
- A cache that has „k" blocks per set is called as "**k-way set associative cache**".
- Each block contains a control-bit called a **valid-bit**.
- The Valid-bit indicates that whether the block contains valid-data.
- The dirty bit indicates that whether the block has been modified during its cache residency.

Valid-bit=0 When power is initially applied to system.
Valid-bit=1 When the block is loaded from main-memory at first time.

- If the main-memory-block is updated by a source & if the block in the source is already exists in the cache, then the valid-bit will be cleared to "0".
- If Processor & DMA uses the same copies of data then it is called as **Cache Coherence Problem**.
- **Advantages:**

1) Contention problem of direct mapping is solved by having few choices for block placement.
2) The hardware cost is decreased by reducing the size of associative search.

4.4.2 REPLACEMENT ALGORITHM

- In direct mapping method,
  the position of each block is pre-determined and there is no need of replacement strategy.
- In associative & set associative method,
  The block position is not pre-determined.
  If the cache is full and if new blocks are brought into the cache,
  then the cache-controller must decide which of the old blocks has to be replaced.
- When a block is to be overwritten, the block with longest time w/o being referenced is over-written.
- This block is called **Least recently Used (LRU) block** & the technique is called **LRU algorithm**.
- The cache-controller tracks the references to all blocks with the help of block-counter.
- **Advantage:** Performance of LRU is improved by randomness in deciding which block is to be over-written.

Eg:

Consider 4 blocks/set in set associative cache.

2 bit counter can be used for each block.

When a **'hit'** occurs, then block counter=0; The counter with values originally lower than the referenced one are incremented by 1 & all others remain unchanged.

When a **'miss'** occurs & if the set is full, the blocks with the counter value 3 is removed, the new block is put in its place & its counter is set to "0" and other block counters are incremented by 1.

4.5 PERFORMANCE CONSIDERATION

- Two key factors in the commercial success are 1) performance & 2) cost.
- In other words, the best possible performance at low cost.
- A common measure of success is called the **Pricel Performance ratio.**
- Performance depends on
  - → how fast the machine instructions are brought to the processor &
  - → how fast the machine instructions are executed.
- To achieve parallelism, *interleaving* is used.
- Parallelism means both the slow and fast units are accessed in the same manner.

INTERLEAVING

- The main-memory of a computer is structured as a collection of physically separate modules.
- Each module has its own
  - 1) ABR (address buffer register) &
  - 2) DBR (data buffer register).
- So, memory access operations may proceed in more than one module at the same time (Fig 5.25).
- Thus, the aggregate-rate of transmission of words to/from the main-memory can be increased.

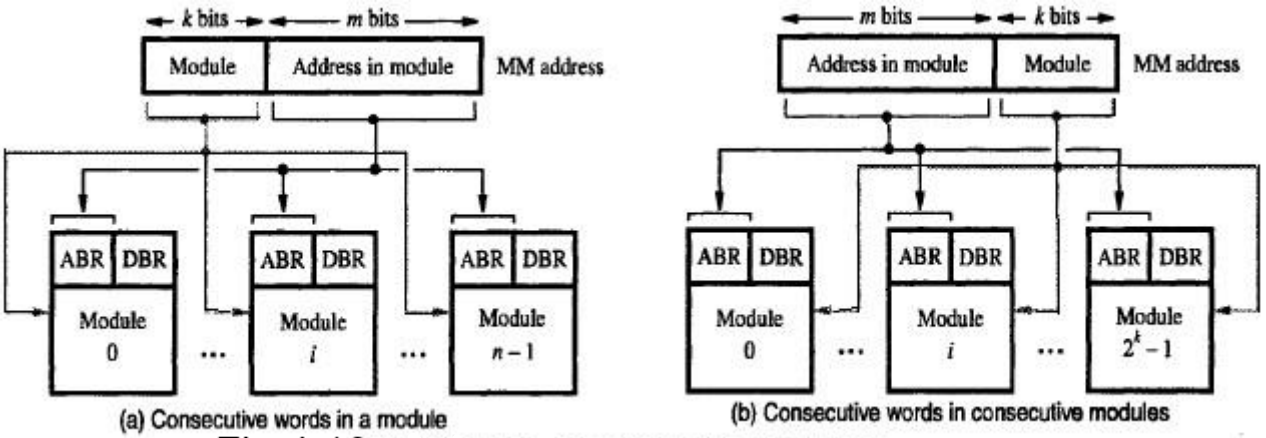


Fig 4.13 Addressing multiple-module memory systems.

- The low-order k-bits of the memory-address select a module.
  - While the high-order m-bits name a location within the module.
    - In this way, consecutive addresses are located in successive modules.
- Thus, any component of the system can keep several modules busy at any one time T.
- This results in both
  - → faster access to a block of data and
  - → higher average utilization of the memory-system as a whole.
- To implement the interleaved-structure, there must be 2k modules;
  - Otherwise, there will be gaps of non-existent locations in the address-space.

Hit Rate & Miss Penalty

- The number of hits stated as a fraction of all attempted accesses is called the **Hit Rate**.
- The extra time needed to bring the desired information into the cache is called the **Miss Penalty**.
- High hit rates well over 0.9 are essential for high-performance computers.
- Performance is adversely affected by the actions that need to be taken when a miss occurs.
- A performance penalty is incurred because
  - of the extra time needed to bring a block of data from a slower unit to a faster unit.
- During that period, the processor is stalled waiting for instructions or data.
- We refer to the total access time seen by the processor when a miss occurs as the miss penalty.
- 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

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

4.6 VIRTUAL MEMORY

• It refers to a technique that automatically move program/data blocks into the main-memory when they are required for execution (Figure 8.24).
• The address generated by the processor is referred to as a **virtual/logical address**.
• The virtual-address is translated into physical-address by **MMU** (Memory Management Unit).
• During every memory-cycle, MMU determines whether the addressed-word is in the memory.
    If the word is in memory.
        Then, the word is accessed and execution proceeds.
        Otherwise, a page containing desired word is transferred from disk to memory.
• Using DMA scheme, transfer of data between disk and memory is performed.

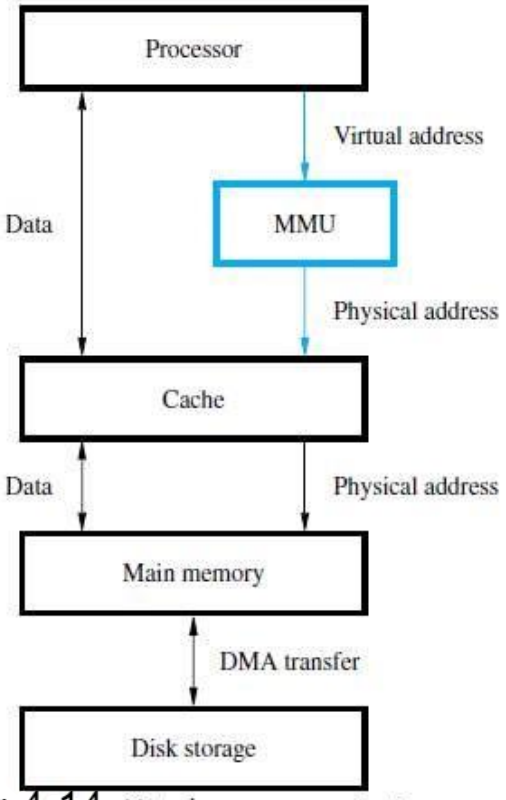


Fig 4:14 Virtual memory organization.

VIRTUAL MEMORY ADDRESS TRANSLATION

• All programs and data are composed of fixed length units called **Pages** (Figure 8.25).
The Page consists of a block-of-words. The words occupy contiguous locations in the memory.
The pages are commonly range from 2K to 16K bytes in length.

• **Cache Bridge** speed-up the gap between main-memory and secondary-storage.

• Each virtual-address contains
1) Virtual Page number (Low order bit) and
2) Offset (High order bit).
Virtual Page number + Offset specifies the location of a particular word within a page.

• **Page-table:** It contains the information about
→ memory-address where the page is stored &
→ current status of the page.

• **Page-frame:** An area in the main-memory that holds one page.

• **Page-table Base Register:** It contains the starting address of the page-table.

• *Virtual Page Number + Page-table Base register* Gives the starting address of the page if that page currently resides in memory.

• **Control-bits in Page-table:** The Control-bits is used to
1) Specify the status of the page while it is in memory.
2) Indicate the validity of the page.
3) Indicate whether the page has been modified during its stay in the memory.

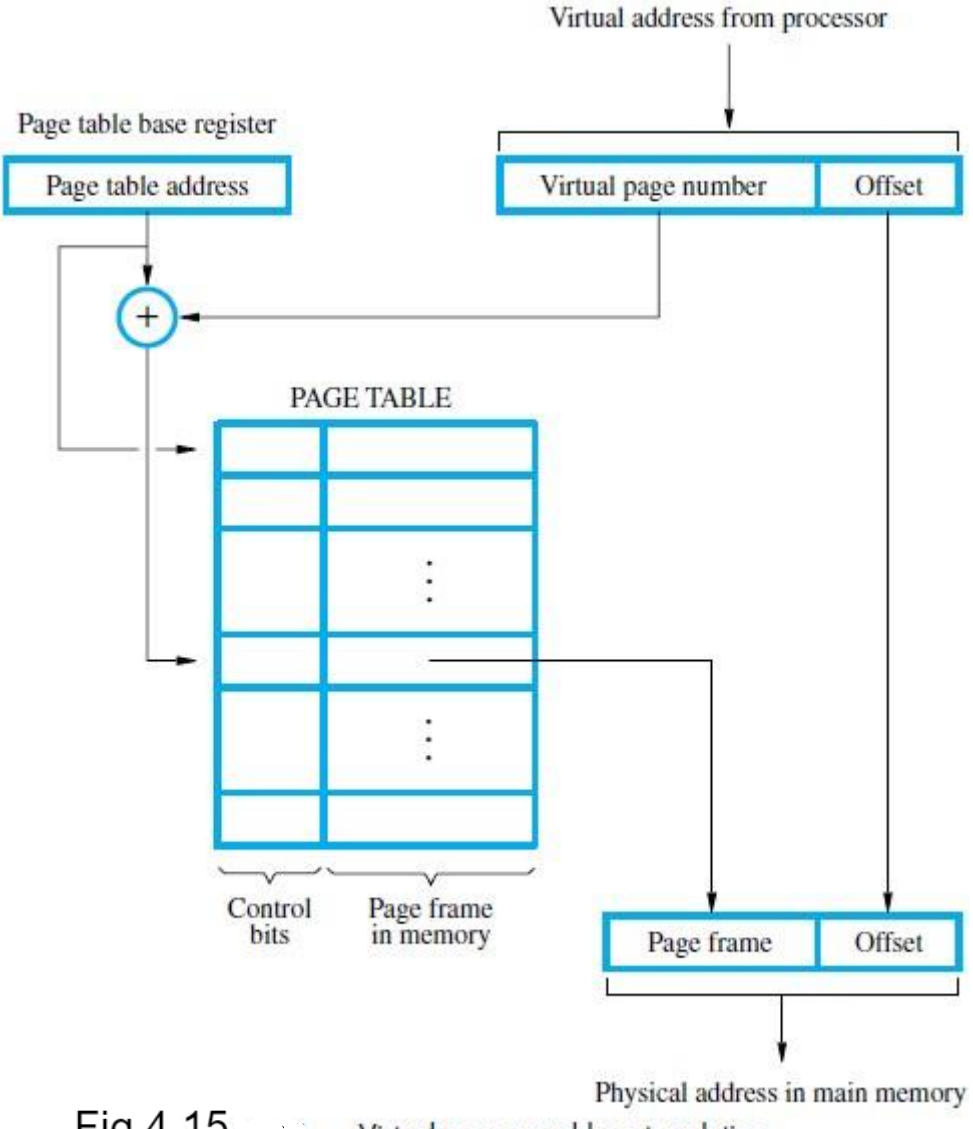


Fig 4.15 Virtual-memory address translation.

Problem 1:

TRANSLATION LOOKASIDE BUFFER (TLB)

• The Page-table information is used by MMU for every read/write access (Figure 8.26).

• The Page-table is placed in the memory but a copy of small portion of the page-table is located within MMU. This small portion is called **TLB** (Translation LookAside Buffer).

TLB consists of the page-table entries that corresponds to the most recently accessed pages.

TLB also contains the virtual-address of the entry.

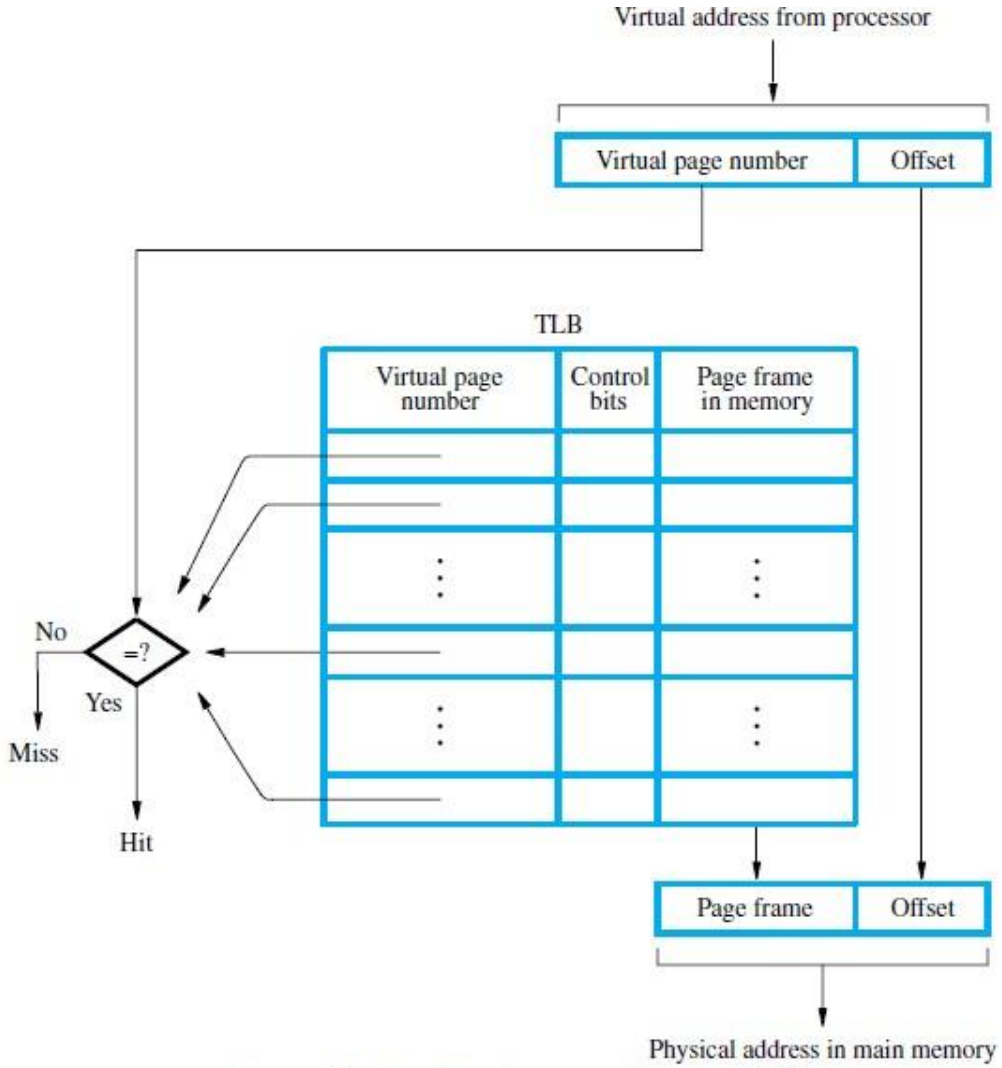


Fig 4.16 Use of an associative-mapped TLB.

• When OS changes contents of page-table, the control-bit will invalidate corresponding entry in TLB.

• Given a virtual-address, the MMU looks in TLB for the referenced-page.

If page-table entry for this page is found in TLB, the physical-address is obtained immediately.

Otherwise, the required entry is obtained from the page-table & TLB is updated.

Page Faults

• Page-fault occurs when a program generates an access request to a page that is not in memory.

• When MMU detects a page-fault, the MMU asks the OS to generate an interrupt.

• The OS

→ suspends the execution of the task that caused the page-fault and

→ begins execution of another task whose pages are in memory.

• When the task resumes the interrupted instruction must continue from the point of interruption.

• If a new page is brought from disk when memory is full, disk must replace one of the resident pages.

In this case, **LRU algorithm** is used to remove the least referenced page from memory.

• A modified page has to be written back to the disk before it is removed from the memory.

In this case, **Write–Through Protocol** is used.

Problem 1:

Consider the dynamic memory cell. Assume that C = 30 femtofarads (10^{-15} F) and that leakage current through the transistor is about 0.25 picoamperes (10^{-12} A). The voltage across the capacitor when it is fully charged is 1.5 V. The cell must be refreshed before this voltage drops below 0.9 V. Estimate the minimum refresh rate.

Solution:

The minimum refresh rate is given by

$$\frac{50 \times 10^{-15} \times (4.5 - 3)}{9 \times 10^{-12}} = 8.33 \times 10^{-3}\ \text{s}$$

Therefore, each row has to be refreshed every 8 ms.

Problem 2:

Consider a main-memory built with SDRAM chips. Data are transferred in bursts & the burst length is 8. Assume that 32 bits of data are transferred in parallel. If a 400-MHz clock is used, how much time does it take to transfer:

(a) 32 bytes of data

(b) 64 bytes of data

What is the latency in each case?

Solution:

(a) It takes 5 + 8 = 13 clock cycles.

$$\text{Total time} = \frac{13}{(133 \times 10^6)} = 0.098 \times 10^{-6}\ \text{s} = 98\ \text{ns}$$

$$\text{Latency} = \frac{5}{(133 \times 10^6)} = 0.038 \times 10^{-6}\ \text{s} = 38\ \text{ns}$$

(b) It takes twice as long to transfer 64 bytes, because two independent 32-byte transfers have to be made. The latency is the same, i.e. 38 ns.

Problem 3:

Give a critique of the following statement: "Using a faster processor chip results in a corresponding increase in performance of a computer even if the main-memory speed remains the same."

Solution:

A faster processor chip will result in increased performance, but the amount of increase will not be directly proportional to the increase in processor speed, because the cache miss penalty will remain the same if the main-memory speed is not improved.

Problem 4:

A block-set-associative cache consists of a total of 64 blocks, divided into 4-block sets. The main-memory contains 4096 blocks, each consisting of 32 words. Assuming a 32-bit byte-addressable address-space,

(a) how many bits are there in main-memory address

(b) how many bits are there in each of the Tag, Set, and Word fields?

Solution:

(a) 4096 blocks of 128 words each require 12+7 = 19 bits for the main-memory address.

(b) TAG field is 8 bits. SET field is 4 bits. WORD field is 7 bits.

Problem 5:

The cache block size in many computers is in the range of 32 to 128 bytes. What would be the main advantages and disadvantages of making the size of cache blocks larger or smaller?

Solution:

Larger size

Fewer misses if most of the data in the block are actually used

Wasteful if much of the data are not used before the cache block is ejected from the cache

Smaller size

More misses

Problem 5:

Consider a computer system in which the available pages in the physical memory are divided among several application programs. The operating system monitors the page transfer activity and dynamically adjusts the number of pages allocated to various programs. Suggest a suitable strategy that the operating system can use to minimize the overall rate of page transfers.

Solution:

The operating system may increase the main-memory pages allocated to a program that has a large number of page faults, using space previously allocated to a program with a few page faults

Problem 6:

In a computer with a virtual-memory system, the execution of an instruction may be interrupted by a page fault. What state information has to be saved so that this instruction can be resumed later? Note that bringing a new page into the main-memory involves a DMA transfer, which requires execution of other instructions. Is it simpler to abandon the interrupted instruction and completely re-execute it later? Can this be done?

Solution:

Continuing the execution of an instruction interrupted by a page fault requires saving the entire state of the processor, which includes saving all registers that may have been affected by the instruction as well as the control information that indicates how far the execution has progressed. The alternative of re-executing the instruction from the beginning requires a capability to reverse any changes that may have been caused by the partial execution of the instruction.

Problem 7:

When a program generates a reference to a page that does not reside in the physical main-memory, execution of the program is suspended until the requested page is loaded into the main-memory from a disk. What difficulties might arise when an instruction in one page has an operand in a different page? What capabilities must the processor have to handle this situation?

Solution:

The problem is that a page fault may occur during intermediate steps in the execution of a single instruction. The page containing the referenced location must be transferred from the disk into the main-memory before execution can proceed.

Since the time needed for the page transfer (a disk operation) is very long, as compared to instruction execution time, a context-switch will usually be made.

(A context-switch consists of preserving the state of the currently executing program, and ”switching” the processor to the execution of another program that is resident in the main-memory.) The page transfer, via DMA, takes place while this other program executes. When the page transfer is complete, the original program can be resumed.

Therefore, one of two features are needed in a system where the execution of an individual instruction may be suspended by a page fault. The first possibility is to save the state of instruction execution. This involves saving more information (temporary programmer-transparent registers, etc.) than needed when a program is interrupted between instructions. The second possibility is to ”unwind” the effects of the portion of the instruction completed when the page fault occurred, and then execute the instruction from the beginning when the program is resumed.

Problem 8:

Magnetic disks are used as the secondary storage for program and data files in most virtual-memory systems. Which disk parameter(s) should influence the choice of page size?

Solution:

The sector size should influence the choice of page size, because the sector is the smallest directly addressable block of data on the disk that is read or written as a unit. Therefore, pages should be some small integral number of sectors in size.

Problem 5:

Problem 9:

A disk unit has 24 recording surfaces. It has a total of 14,000 cylinders. There is an average of 400 sectors per track. Each sector contains 512 bytes of data.

(a) What is the maximum number of bytes that can be stored in this unit?
(b) What is the data-transfer rate in bytes per second at a rotational speed of 7200 rpm? (c) Using a 32-bit word, suggest a suitable scheme for specifying the disk address.

Solution:

(a) The maximum number of bytes that can be stored on this disk is 24 X 14000 X 400 X 512 =
68.8×10^9 bytes.
(b) The data-transfer rate is (400 X 512 X 7200)/60 = 24.58×10^6 bytes/s.
(c) Need 9 bits to identify a sector, 14 bits for a track, and 5 bits for a surface.

Thus, a possible scheme is to use address bits A_{8-0} for sector, A_{22-9} for track, and A_{27-23} for surface identification. Bits