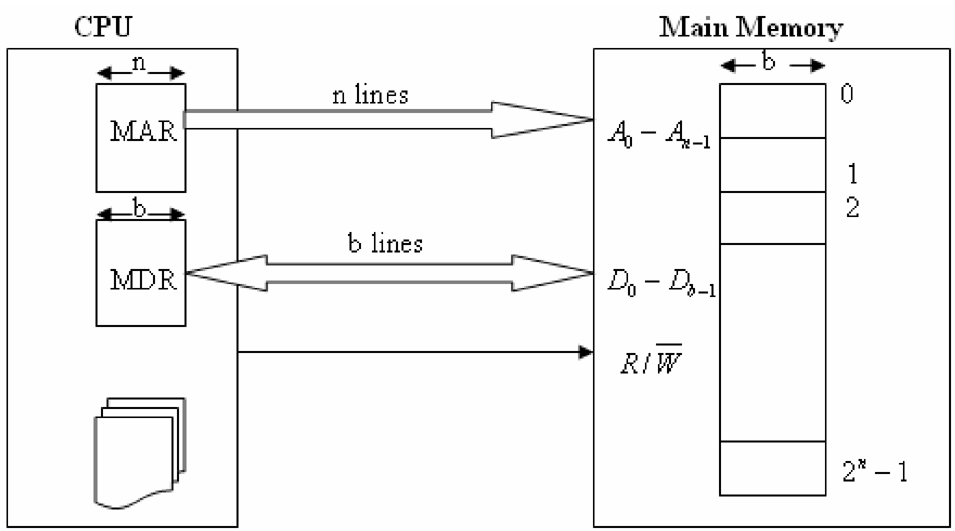
Chapter 7

Memory System Design II

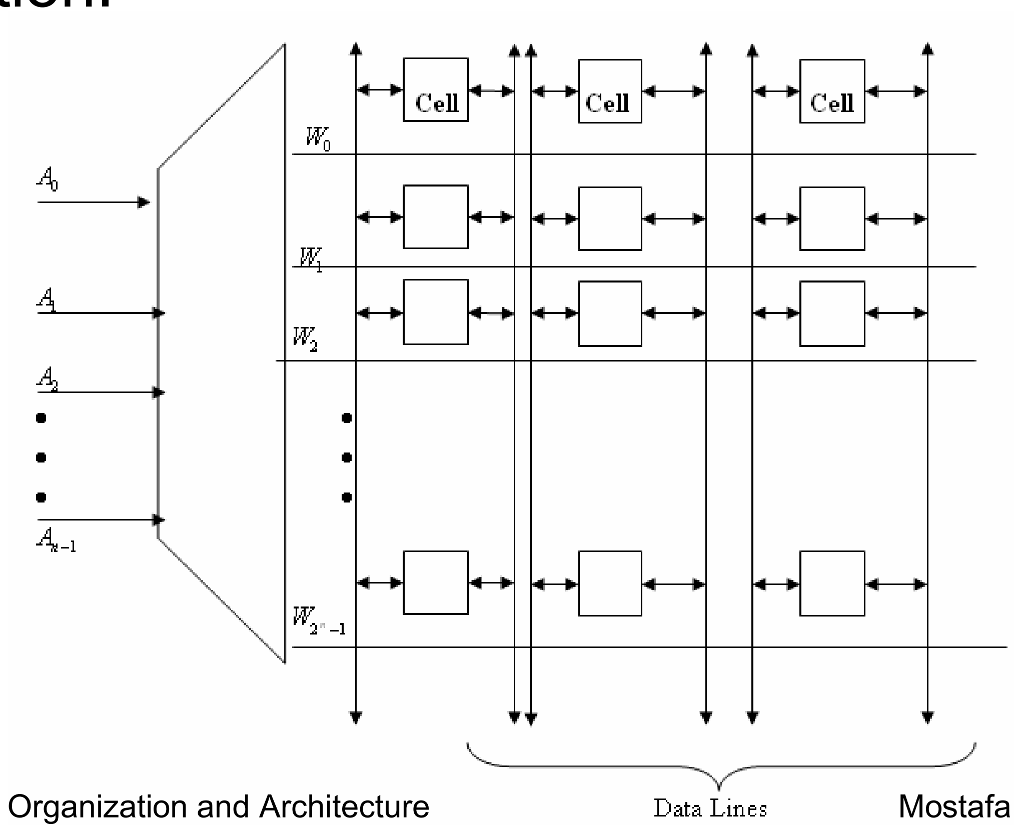
7.1 Main Memory

- The main memory provides the main storage for a computer.
- Two CPU registers are used to interface the CPU to the main memory:
  - The Memory Address Register (MAR) and
  - The Memory Data Register (MDR): is used to hold the data to be stored and/or retrieved in/from the memory location whose address is held in the MAR.



7.1 Main Memory

- It is possible to visualize a typical internal main memory structure as consisting of rows and columns of basic cells. Each cell is capable of storing one bit of information.

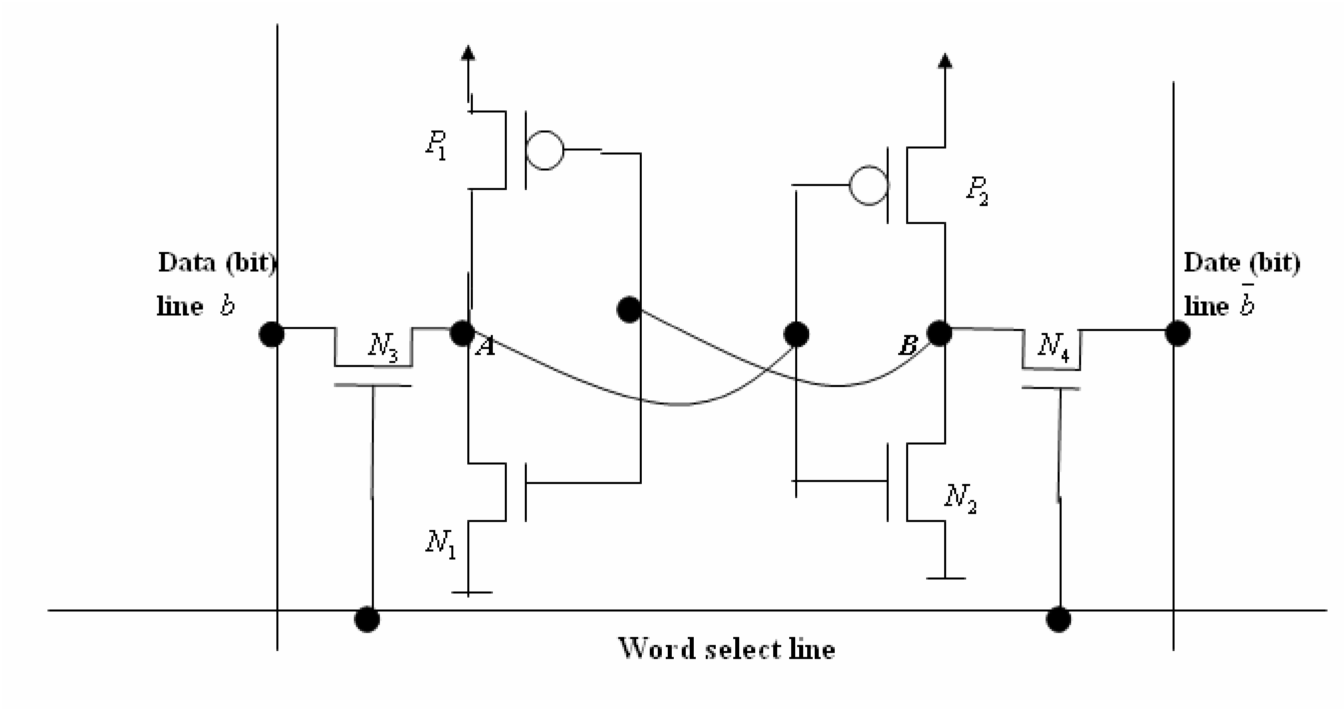


7.1 Main Memory

- In static CMOS technology, each main memory cell consists of six transistors.
- The six transistor static CMOS memory cell consists of two inverters back to back. It should be noted that the cell could exist in one of the two stable states.
- The two transistors N_3 and N_4 are used to connect the cell to the two data (bit) lines.
- If the word select is not activated, these two transistors are turned off, thus protecting the cell from the signal values carried by the data lines.
- The two transistors are turned on when the word select line is activated. What takes place when the two transistors are turned on will depend on the intended memory operation.

7.1 Main Memory

- Static CMOS Memory Cell



7.1 Main Memory

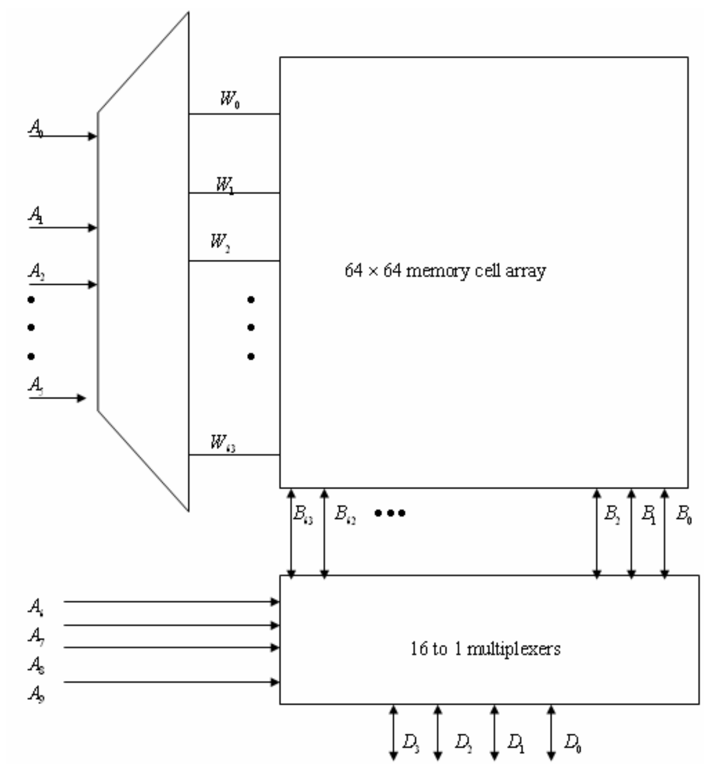
- Read Operation
  - Both lines b and $\bar{b}$ are precharged high.
  - The word select line is activated, thus turning both transistors N_3 and N_4.
  - Depending on the internal value stored in the cell, point A (B) will lead to the discharge of line b($\bar{b}$).

7.1 Main Memory

- Write Operation
  - The bit lines are precharged such that b($\bar{b}$) = 1(0).
  - The word select line is activated, thus turning both transistors N_3 and N_4.
  - The bit line precharged with 0 will have to force the point A (B), which is having 1, to 0.

7.1 Main Memory

- The internal organization of the memory array should satisfy an important memory design factor, i.e. efficient utilization of the memory chip.



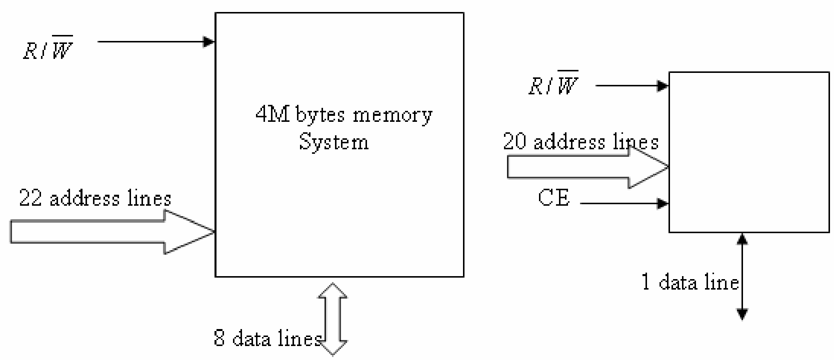
7.1 Main Memory

- Another important factor related to the design of the main memory sub-system is the number of chip pins required in an integrated circuit.
  - Different organization of the same memory capacity can lead to different number of chip pins requirements.

Organization	Number of needed address lines	Number of needed data lines
4K × 1	12	1
1K × 4	10	4
512 × 8	9	8
256 × 16	8	16

7.1 Main Memory

- Another factor pertinent to the design of main memory sub-system is the required number of memory chips.
- It is important to realize that the available per chip memory capacity can be a limiting factor in designing memory sub-systems.



CE	$R/\overline{W}$	Mode
0	×	Tri-state
1	1	Read
1	0	Write

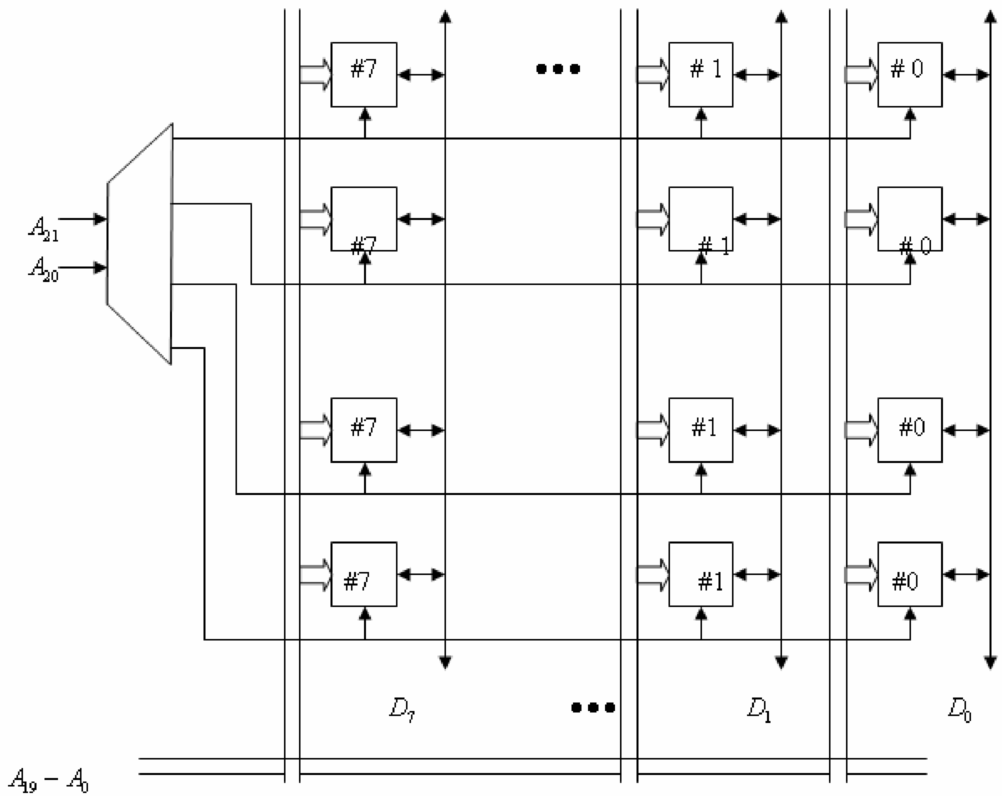
× = don't care
Tri-state = high impedance

(a) Intended memory system.

(b) Basic memory building block.

7.1 Main Memory

- The memory sub-system can be arranged in four rows, each having eight chips.

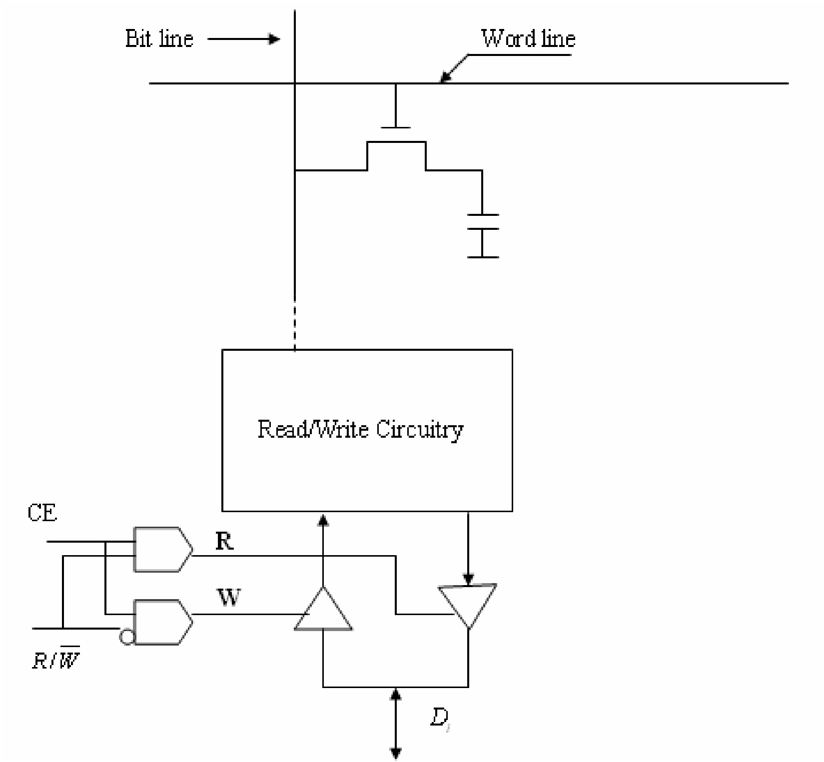


7.1 Main Memory

- It is possible to use a 1-transistor dynamic cell instead of the 6-transistor static random cell.
- Dynamic memory depends on storing logic values using a capacitor together with one transistor that acts as a switch.
- The use of dynamic memory leads to saving in chip area.
- However, due to the possibility of decay of the stored values (leakage of the stored charge in the capacitor), dynamic memory requires periodical (every few milliseconds) refreshment in order to restore the stored logic values.

7.1 Main Memory

- The read/write circuitry performs the functions of sensing the value on the bit line, amplifying it, and refreshing the value stored on the capacitor.

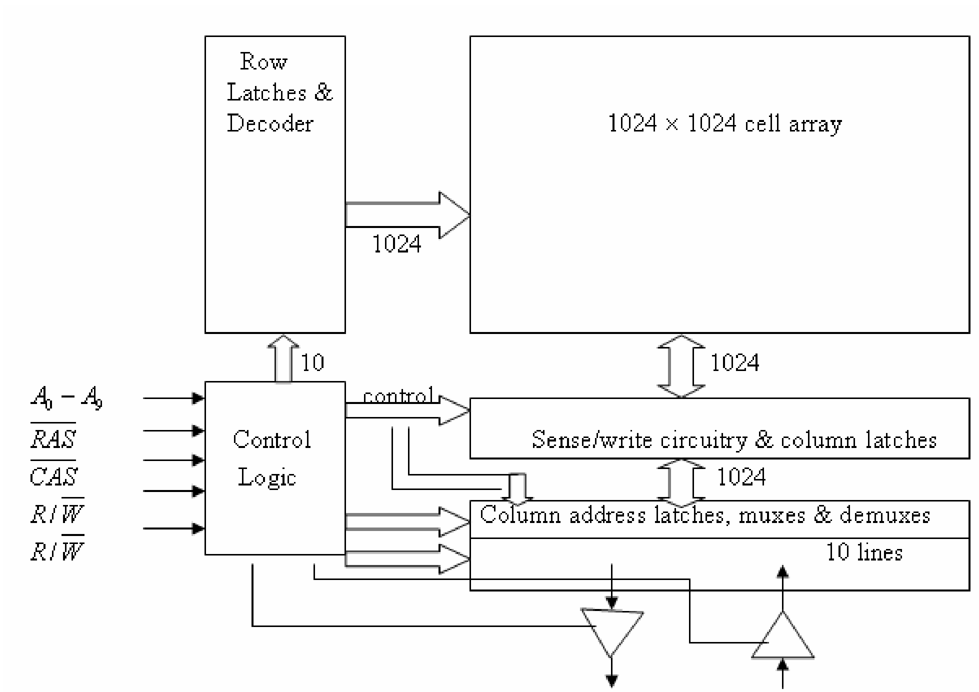


7.1 Main Memory

- In order to reduce the number of pins required for a given dynamic memory sub-system, it is a normal practice to divide the address lines into row and column address lines.
- In addition, the row and column address lines are transmitted over the same pins, one after the other in a scheme known as *time-multiplexing.*
- This can potentially cut the number of address pins required by half. Due to time-multiplexing of address lines, it will be necessary to add two extra control lines, i.e., row address strobe ($\overline{RAS}$) and column address strobe ($\overline{CAS}$).

7.1 Main Memory

- These two control lines are used to indicate to the memory chip when the row address lines are valid and when the column address lines are valid, respectively.

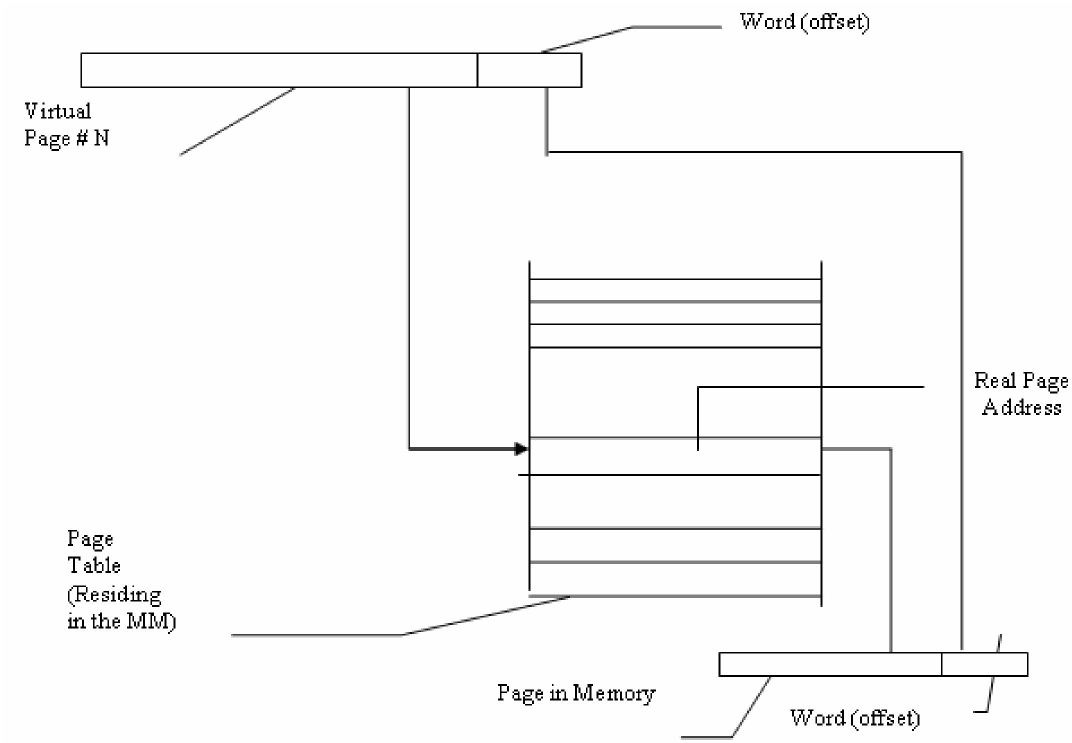


7.2 Virtual Memory

- Direct Mapping
  - The virtual address issued by the processor is divided into two fields:
    - The virtual page number field and
    - The offset field.
  - If the number of bits in the virtual page number field is *N,* then the number of entries in the page table will be 2^N.

7.2 Virtual Memory

- Direct Mapping



7.2 Virtual Memory

- Direct Mapping
  - The virtual page number field is used to directly address an entry in the page table.
    - If the corresponding page is valid (as indicated by the valid bit), then the contents of the specified page table entry will correspond to the physical page address.
    - The latter is then extracted and concatenated with the offset field in order to form the physical address of the word requested by the processor.
    - If, on the other hand, the specified entry in the page table does not contain a valid physical page number, then this represents a page fault.
    - In this case, the *MMU* will have to bring the corresponding page from the hard disk, load it into the main memory, and indicate the validity of the page.

7.2 Virtual Memory

- Direct Mapping
  - The main advantage of the direct-mapping technique is its simplicity measured in terms of the direct addressing of the page table entries.
  - Its main disadvantage is the expected large size of the page table.
    - In order to overcome the need for a large page table, the associative-mapping technique is used.

7.2 Virtual Memory

- Associative Mapping
  - The technique is similar to direct mapping as long as the virtual address issued by the processor is divided in two fields:
    - The virtual page number field and
    - The offset field.
  - However, the page table used in associative mapping could be far shorter than its direct mapping counterpart.
  - Every entry in the page table is divided into two parts:
    - The virtual page number and
    - The physical page number.

7.2 Virtual Memory

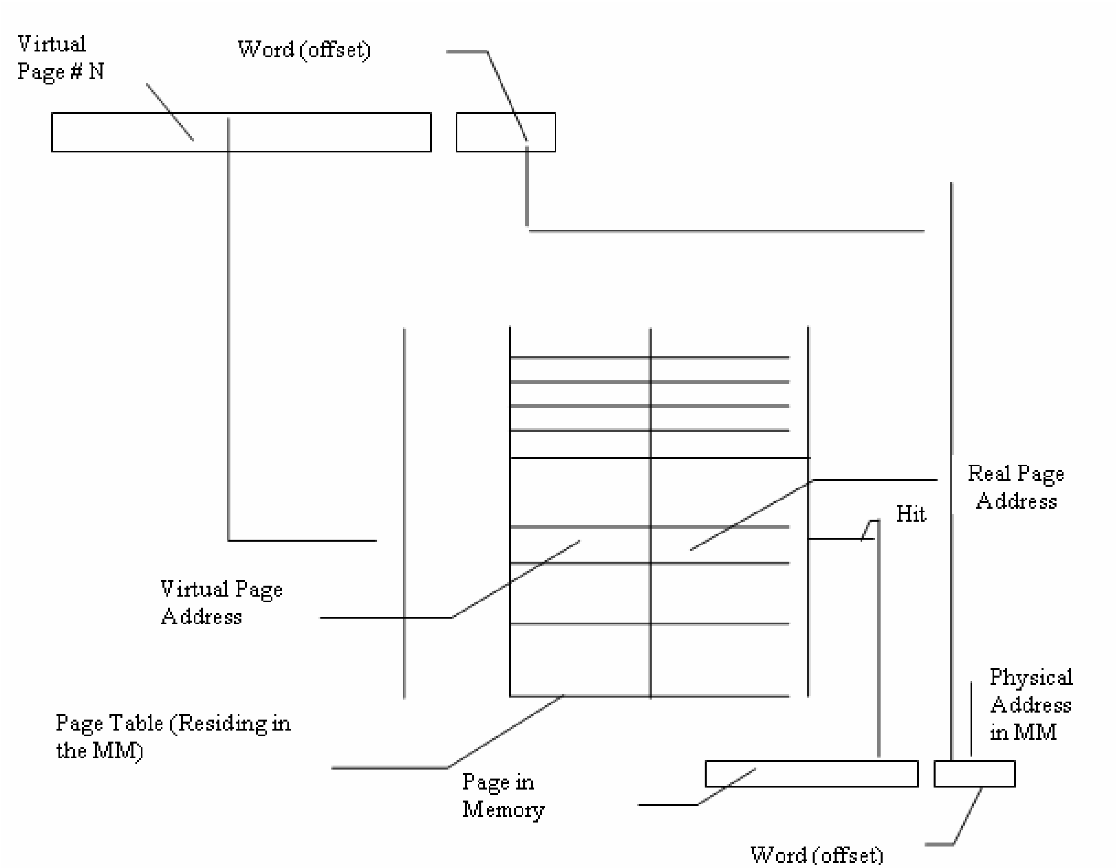
- Associative Mapping
  - A match is searched (associatively) between the virtual page number field of the address and the virtual page numbers stored in the page table.
    - If a match is found, the corresponding physical page number stored in the page table is extracted and is concatenated with the offset field in order to generate the physical address of the word requested by the processor.
    - If, on the other hand, a match could not be found, then this represents a page fault. In this case, the *MMU* will have to bring the corresponding page from the hard disk, load it into the main memory, and indicate the validity of the page.

7.2 Virtual Memory

- Associative Mapping
  - The main advantage of the associative-mapping technique is the expected shorter page table (compared to the direct-mapping technique) required for the translation process.
  - Its main disadvantage is the search required for matching the virtual page number field and all virtual page numbers stored in the page table.
  - Although such search is done associatively, it requires the use of an added hardware overhead.
  - A possible compromise between the complexity of the associative mapping and the simplicity of the direct mapping is the set-associative mapping technique.

7.2 Virtual Memory

- Associative Mapping



7.2 Virtual Memory

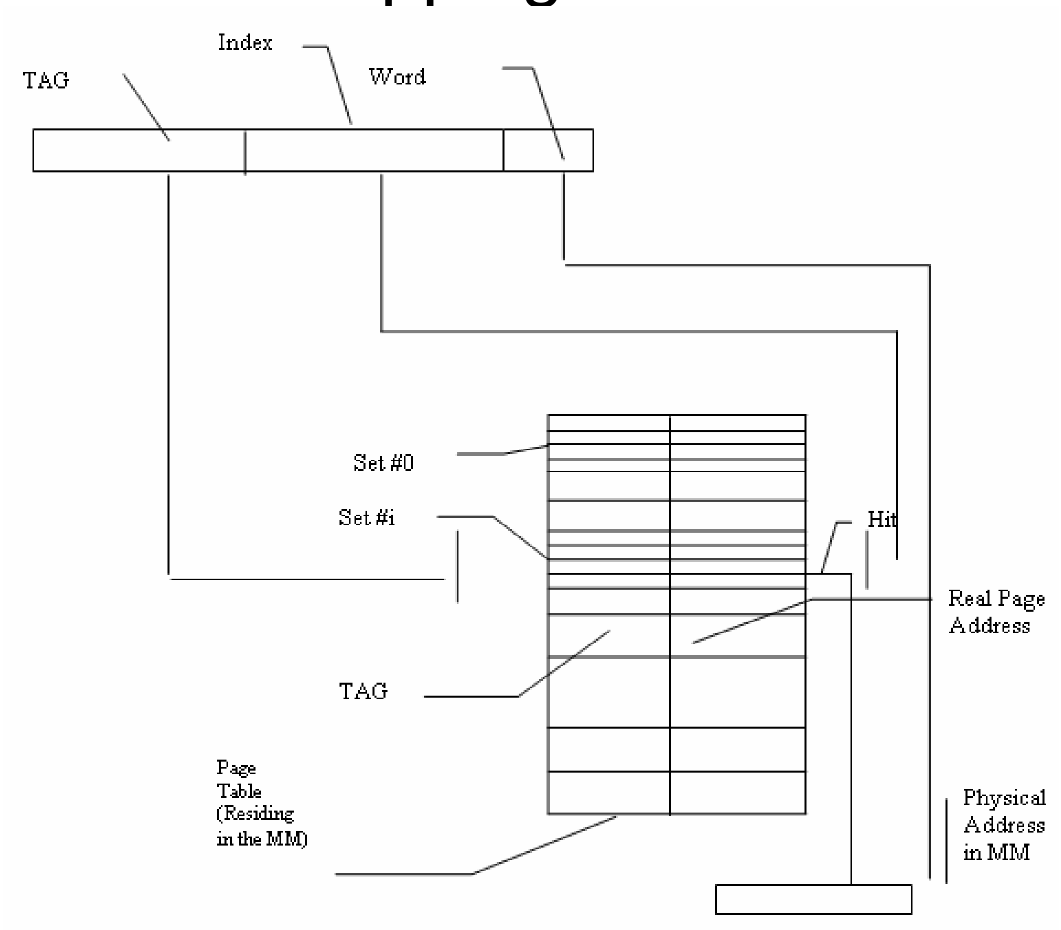
- Set-Associative Mapping
  - The virtual address issued by the processor is divided into three fields:
    - The tag,
    - The index, and
    - The offset.
  - The page table used in the set-associative mapping is divided into sets, each consisting of a number of entries.
    - Each entry in the page table consists of a tag and the corresponding physical page address.
    - Similar to the direct mapping, the index field is used to directly determine the set in which a search should be conducted.
    - If the number of bits in the index field is *S*, then the number of sets in the page table should be 2^S.
    - Once the set is determined, then a search (similar to the associative mapping) is conducted to match the tag field with all entries in that specific set.

7.2 Virtual Memory

- Set-Associative Mapping
  - If a match is found, then the corresponding physical page address is extracted and concatenated with the offset field in order to generate the physical address of the word requested by the processor.
  - If, on the other hand, a match could not be found, then this represents a page fault. In this case, the *MMU* will have to bring the corresponding page from the hard disk, load it into the main memory, update the corresponding set and indicate the validity of the page.
  - The set-associative-mapping technique strikes a compromise between the inefficiency of the direct mapping, in terms of the size of the page table, and excessive hardware overhead of the associative mapping.
  - It also enjoys the best of the two techniques: the simplicity of the direct mapping and the efficiency of the associative mapping.

7.2 Virtual Memory

- Set-Associative Mapping

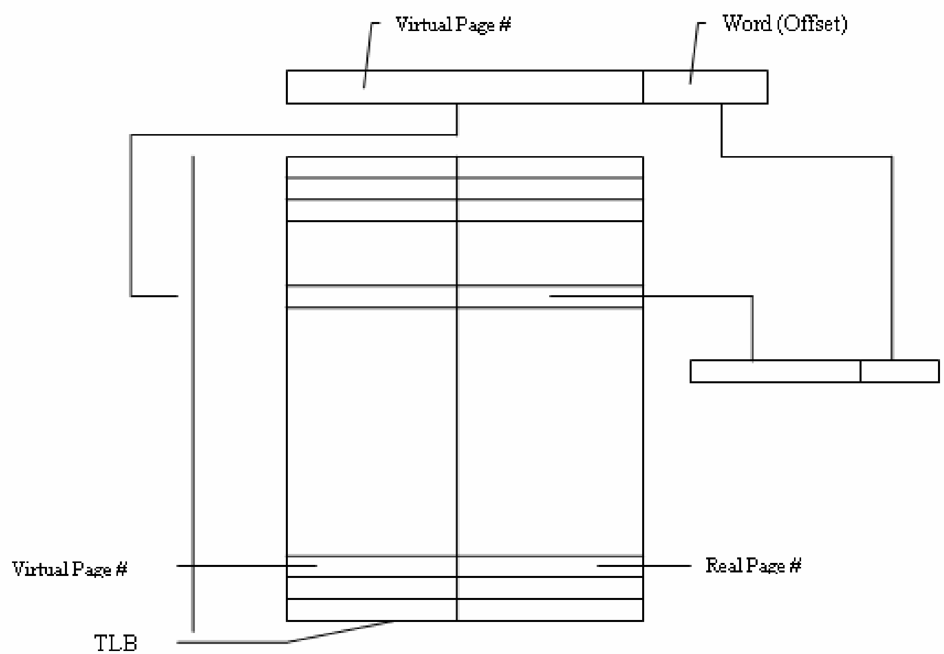


7.2 Virtual Memory

- Translation Look-Aside Buffer (TLB)
  - In most modern computer systems a copy of a small portion of the page table is kept on the processor chip.
  - This portion consists of the page table entries that correspond to the most recently accessed pages.
  - This small portion is kept in the Translation Look-aside Buffer (TLB) cache.
  - A search in the TLB precedes that in the page table.
  - Therefore, the virtual page field is first checked against the entries of the TLB in the hope that a match is found.
  - A hit in the TLB will result in the generation of the physical address of the word requested by the processor, thus saving the extra main memory access required to access the page table.
  - It should be noted that a miss on the TLB is not equivalent to a page fault.

7.2 Virtual Memory

- Translation Look-Aside Buffer (TLB)
  - Typical size of a TLB is in the range of 16 to 64 entries.
  - With this small TLB size, a hit ratio of more than 90% is always possible.
  - Due to its limited size, the search in the TLB is done associatively, thus reducing the required search time.



7.2 Virtual Memory

- Replacement Algorithms (Policies)
  - Random Replacement
    - According to this replacement policy, a page is selected randomly for replacement.
    - This is the simplest replacement mechanism. It can be implemented using a pseudo-random number generator that generates numbers that correspond to all possible page frames.
    - At the time of replacement, the random number generated will indicate the page frame that must be replaced.
    - Although simple, this technique may not result in efficient use of the main memory, i.e. low hit ratio *h.*
    - Random replacement has been used in the Intel *i860* family of RISC processor.

7.2 Virtual Memory

- Replacement Algorithms (Policies)
  - First-In-First-Out Replacement
    - According to this replacement policy, the page that was loaded before all the others in the main memory is selected for replacement.
    - The basis for page replacement in this technique is the time spent by a given page residing in the main memory regardless of the pattern of usage of that page.
    - This technique is also simple.
    - However, it is expected to result in acceptable performance, measured in terms of the main memory hit ratio, if the page references made by the processor are in strict sequential order.

7.2 Virtual Memory

- Replacement Algorithms (Policies)
  - Least Recently Used (LRU) Replacement
    - According to this technique, page replacement is based on the pattern of usage of a given page residing in the main memory regardless of the time spent in the main memory.
    - The page that has not been referenced for the longest time while residing in the main memory is selected for replacement.
    - The LRU technique matches most programs' characteristics and therefore is expected to result in the best possible performance in terms of the main memory hit ratio.
    - It is however more involved compared to other techniques.

7.2 Virtual Memory

- Replacement Algorithms (Policies)
  - Clock Replacement Algorithm
    - This is a modified FIFO algorithm.
    - It takes into account both the time spent by a page residing in the main memory (similar to the FIFO) and the pattern of usage of the page (similar to the LRU).
    - The technique is therefore sometimes called First-In-Not-Used-First-Out (FI-NU-FO).
    - In keeping track of both the time and the usage, the technique uses a pointer to indicate where to place the incoming page and a used bit to indicate the usage of a given page. The technique can be explained using these three steps:
      - If the used bit = 1, then reset bit, increment pointer and repeat.
      - If the used bit = 0, then replace corresponding page and increment pointer.
      - The used bit is SET if the page is referenced after the initial loading.

7.2 Virtual Memory

- Virtual Memory Systems with Cache memory
  - A typical computer system will contain a cache, a virtual memory, and a TLB.
  - When a virtual address is received from the processor, a number of different scenarios can occur, each is dependent on the availability of the requested item in the cache, the main memory, or the secondary storage.
  - The first level of address translation checks for a match between the received virtual address and the virtual addresses stored in the TLB.
  - If a match occurs (TLB hit) then the corresponding physical address is obtained:
    - This physical address can then be used to access the cache.
    - If a match occurs (cache hit) then the element requested by the processor can be sent from the cache to the processor.

7.2 Virtual Memory

- Virtual Memory Systems with Cache memory
    - If, on the other hand, a cache miss occurs, then the block containing the targeted element is copied from the main memory into the cache and the requested element is sent to the processor.
  - If a TLB miss occurs, then the page table (PT) is searched for the existence of the page containing the targeted element in the main memory:
    - If a PT hit occurs, then the corresponding physical address is generated and a search is conducted for the block containing the requested element. This will require updating the TLB.
    - If on the other hand a PT miss takes place (indicating a Page Fault), then the page containing the targeted element is copied from the disk into the main memory, a block is copied into the cache, and the element is sent to the processor.
      - This last scenario will require updating the page table, the main memory, and the cache.

7.2 Virtual Memory

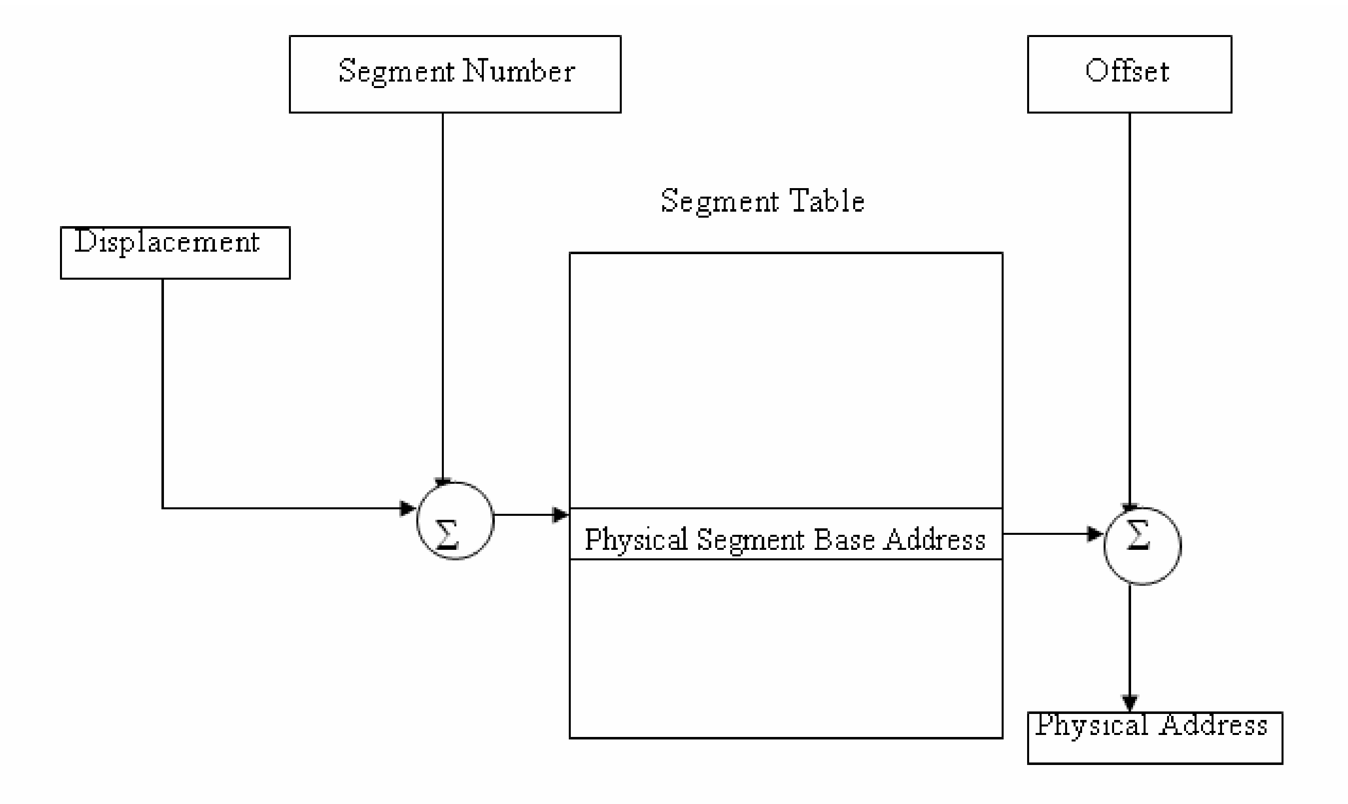
- Segmentation
  - A segment is a block of contiguous locations of varying size.
  - Segments are used by the Operating System (O.S.) to relocate complete programs in the main memory and the disk.
  - Segments can be shared between programs. They provide means for protection from unauthorized access and/or execution.
  - It is not possible to enter segments from other segments unless the access has been specifically allowed.
  - Data Segments and Code Segments are separated.
  - It should not also be possible to alter information in the Code Segment while fetching an instruction nor should it be possible to execute data in a Data Segment.

7.2 Virtual Memory

- Segment Address Translation
  - In order to support segmentation, the address issued by the processor should consist of a Segment Number (Base) and a Displacement (or an offset) within the segment.
  - Address translation is performed directly via a Segment Table.
  - The starting address of the targeted segment is obtained by adding the Segment Number to the contents of the Segment Table Pointer.
  - One important content of the Segment Table is the Physical Segment Base address.
  - Adding the latter to the Offset yields the required Physical Address.

7.2 Virtual Memory

- Segment Address Translation

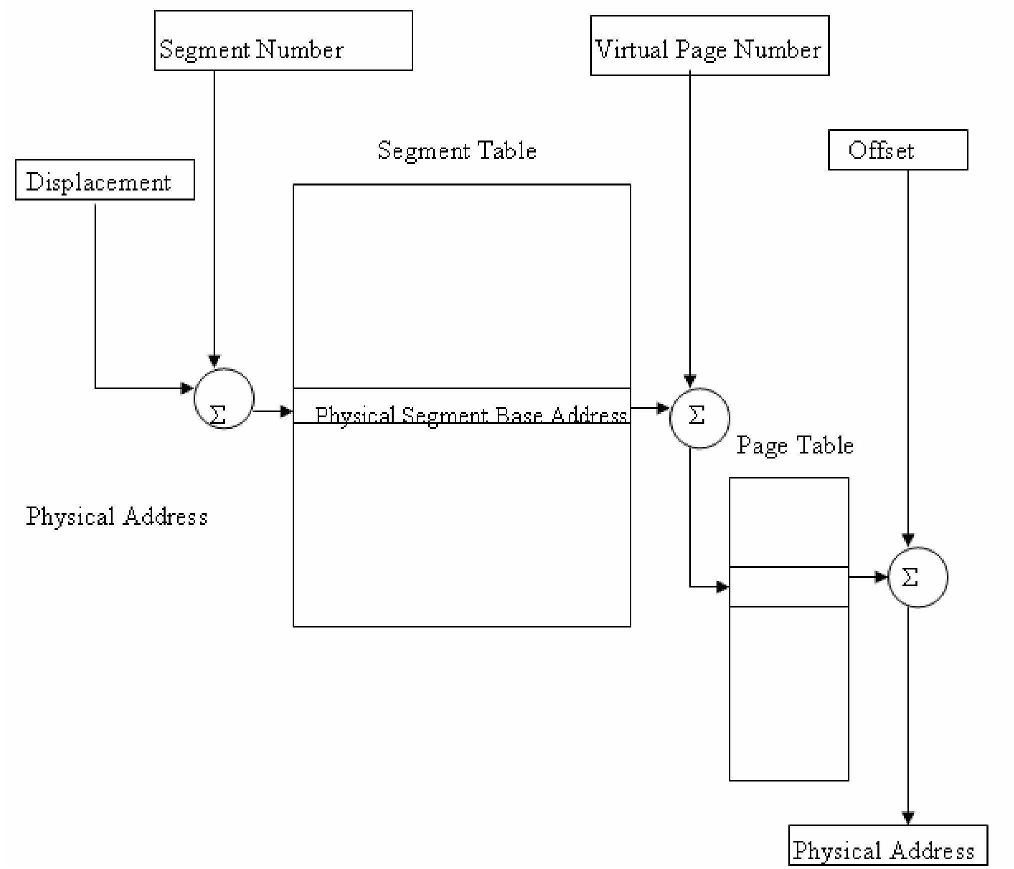


7.2 Virtual Memory

- Paged Segmentation
  - Both Segmentation and Paging are combined in most systems.
  - Each segment is divided into a number of equal sized pages.
  - The basic unit of transfer of data between the main memory and the disk is the page.
  - In this case, the virtual address is divided into a segment number, a page number, and displacement within the page.
  - The output of the page table is the page physical address which when concatenated with the word field of the virtual address results in the physical address.

7.2 Virtual Memory

- Paged Segmentation



7.2 Virtual Memory

- Pentium Memory Management
  - In the Pentium processor, both segmentation and paging are individually available and can be disabled as well.
  - Four distinct views of the memory exist.
    - Unsegmented Unpaged Memory
    - Unsegmented Paged Memory
    - Segmented Unpaged Memory
    - Segmented Paged Memory

7.2 Virtual Memory

- Pentium Memory Management
  - For segmentation, 16-bit segment number (two of which are used for protection) and 32-bit offset will give a segmented virtual address space equal to 2^{46} = 64 terabytes.
  - The virtual address space is divided into two parts; one half, i.e. 8K $\times$ 4 GByte, is global and shared by all processes, and the other half is local and is distinct for each process.
  - For paging, a two-level table lookup paging system is used. First level is a page directory with 1024 entries, i.e. 1024 page groups, each with its own page table and each 4 MB in length.
  - Each page table contains 1024 entries; each entry corresponds to a single 4-Kbyte page.

7.3 Virtual Memory

- Read-Only Memory
  - Random access as well as cache memories are examples of *volatile* memories.
  - A volatile storage is defined as the one which loses its contents when power is turned off.
  - *Nonvolatile* memory storages are those that retain the stored information if power is turned off.
  - As there is a need for volatile storage there is also a need for nonvolatile storage.
  - Computer system boot subroutines, microcode control, video game cartridges are few examples of computer software that require the use of nonvolatile storage.

7.3 Virtual Memory

- Read-Only Memory
  - ROM can also be used to realize combinational logic functions.
  - The technology used for implementing ROM chips has evolved over the years.
  - Early implementations of ROMs were called *mask-programmed ROMs.* In this case, a made-to-order one time ROM is programmed according to a specific encoding pattern supplied by the user.
  - If the user would like to program his/her ROM on site, then a different type of ROM, called the Programmable ROM (PROM) should be used.

7.3 Virtual Memory

- Read-Only Memory
  - Although it allows for some added flexibility, yet PROM is still restricted by the fact that it can only be programmed once (by the user).
  - A third type of ROM, called Erasable PROM (EPROM) is reprogrammable, i.e., allows stored data to be erased and new data to be stored.
  - Flash EPROMs (FEPROMs) have emerged as strong contenders to EPROMs.
  - This is because FEPROMs are more compact, faster, and removable as compared to EPROM.
  - A different type of ROM that overcomes the drawback of the EPROM is the Electrically EPROM or EEPROM.
  - In this case, the erasure of the EPROM can be done electrically and selectively.

7.3 Virtual Memory

- Read-Only Memory

ROM type	Cost	Programmability	Typical Applications
Mask-programmed ROM	Truly inexpensive	Once at manufacture	Microcode
PROM	Inexpensive	Once on site	Prototyping
EPROM	Moderate	Many times	Prototyping
FEPROM	Expensive	Many times	VCR & TVs
EEPROM	Truly Expensive	Many times	VCR & TVs

7.4 Summary

- The design aspects that relate to the internal and external organization of the main memory were covered.
- The design of a static RAM cell was introduced with emphasis on the read and the write operations.
- Three address translation techniques were discussed and compared.
- The use of a TLB to improve the average access time was explained.

7.4 Summary

- Three replacement techniques were introduced.
- Segmented paged systems were also introduced.
- The discussion on virtual memory ended up with an explanation of the virtual memory aspects of the Pentium-IV processor.
- Toward the end of the chapter, a number of implementations for ROMs were visited.