**UNIT:3**

**THE MEMORY SYSTEM**

**Syllabus:**

* **Various technologies used in memory design,01**
* **Higher order memory design**
* **Multi-module memories and interleaving,**
* **Associative Memory,**
* **Cache memory,**
* **Virtual Memory.**

**Introduction:**

* **Programs and the data are held in the memory of the computer.**
* **There is just not enough space in one memory unit to accommodate all the programs used in a typical computer.**
* **Moreover, most computer users accumulate and continue to accumulate large amounts of data processing software.**
* **Not all accumulated information is needed by the processor at the same time.**
* **Therefore, it is more economical to use low cost storage devices to serve as a backup for storing the information that is not currently used by the CPU.**
* **The memory unit that communicates directly with the CPU is called the main memory.**
* **Devices that provide backup storage are called auxiliary memory. The most common auxiliary memory devices used in computer systems are magnetic disks and tapes.**
* **Only programs and data currently needed by the processor reside in main memory. All other information is stored in auxiliary memory and transferred to main memory when needed.**

**Memory Hierarchy:**



* **The memory hierarchy system consists of all storage devices employed in a computer system from the slow but high capacity auxiliary memory to a relatively faster main memory, to an even smaller and faster cache memory.**
* **At the bottom of the hierarchy are the relatively slow magnetic tapes used to store removable files.**
* **Next are the magnetic disks used as backup storage.**
* **The main memory occupies a central position by being able to communicate directly with the CPU and with auxiliary memory devices through an I/O processor.**
* **When programs not residing in main memory are needed by the CPU, they are brought in from auxiliary memory. Programs not currently needed in main memory are transferred into auxiliary memory to provide space for currently used programs and data.**
* **A special very high speed memory called a cache is sometimes used to increase the speed of processing by making current programs and data available to the CPU at a rapid rate. The cache memory is employed in computer systems to compensate for the speed differential between main memory access time and processor logic. CPU logic is usually faster than main memory access time. A technique used to compensate for the mismatch in operating speeds is to employ an extremely fast, small cache between the CPU and main memory.**
* **While the I/O processor manages data transfers between auxiliary memory and main memory, the cache organization is concerned with the transfer of information between main memory and CPU. Thus each is involved with a different level in the memory hierarchy system.**
* **The reason for having two or three levels of memory hierarchy is economics.**
* **As the storage capacity of the memory increases, the cost per bit for storing binary information decreases and the access time of the memory becomes longer.**
* **The auxiliary memory has a large storage capacity, is relatively inexpensive, but has low access speed compared to main memory.**
* **The cache memory is very small, relatively expensive, and has very high access speed.**
* **Auxiliary and cache memories are used for different purposes. The cache holds those parts of the program and data that are most heavily used, while the auxiliary memory holds those parts that are not presently used by the CPU.**

**Main Memory:**

* **The main memory is the central storage unit in a computer system. It is relatively large and fast memory used to store programs and data during the computer operation.**
* **The principle technology used for the main memory is based on semiconductor integrated circuits.**
* **Integrated circuit RAM chips are available in two possible operating modes, static and dynamic.**

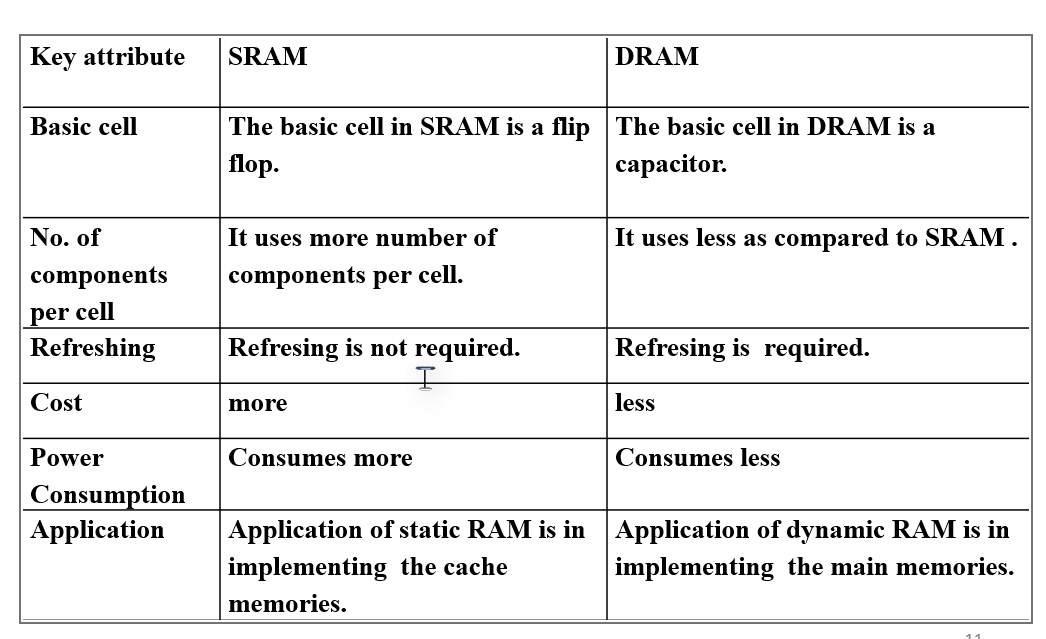
**SRAM:**

* + **SRAM is also known as static RAM.**
  + **In SRAM data will remain stored permanently as along as power is supplied, they need not required rewriting periodically the data.**
  + **The static RAM is easier to use and has shorter read and write cycles. The basic cell in SRAM is a flip flop.**

**DRAM:**

* + **DRAM is also known as dynamic RAM. In DRAM rewriting periodically the data into memory is required. The basic cell in DRAM is a capacitor.**
  + **The dynamic RAM stores the binary information in the form of electric charges that are applied to capacitors.**
  + **The capacitors are provided inside the chips by MOS transistors.**
  + **The stored charge on the capacitors tend to discharge with time and the capacitors must be periodically recharged by refreshing the dynamic memory.**
  + **The dynamic RAM offers reduced power consumption and larger storage capacity in a single memory chip.**

**SRAM Vs. DRAM:**

****

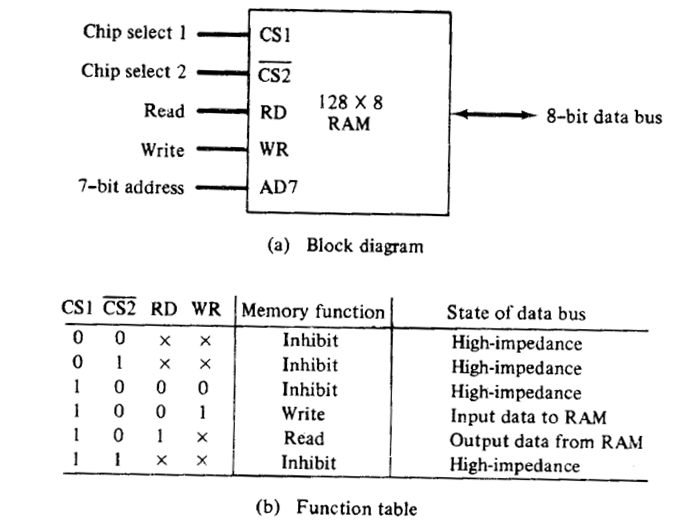
**ROM:**

* **Most of the main memory in a general purpose computer is made up of RAM integrated circuit chips, but a portion of the memory may be constructed with ROM chips.**
* **Originally, ROM was used to refer to a random access memory, but now it is used to designate a read/write memory to distinguish it from a read-only memory, although ROM is also random access.**
* **RAM is used for storing the bulk of the programs and data that are subject to change.**
* **ROM is used for storing programs that are permanently resident in the computer and for tables of constants that do not change in value once the production of the computer is completed.**

**RAM and ROM chips:**

* **RAM and ROM chips are available in a variety of sizes.**
* **If the memory needed for the computer is larger than the capacity of one chip; it is necessary to combine a number of chips to form the required memory size.**
* **To demonstrate the chip interconnection, we will see an example of a 1024 x 8 memory constructed with 128 x 8 RAM chips and 512 x 8 ROM chips.**

**RAM chip and function table:**



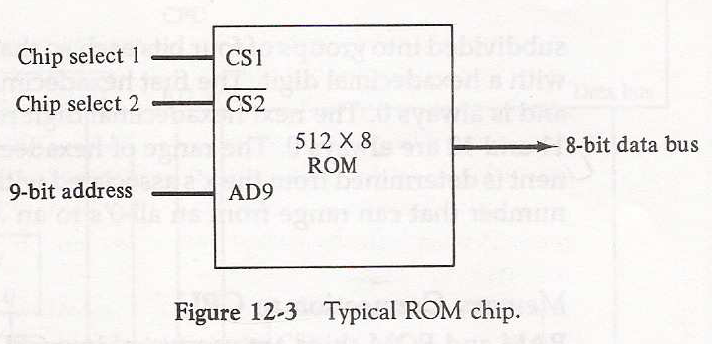
**RAM chips:**

* **In the above slide block diagram of RAM chip is shown. The capacity of the memory is 128 words of eight bits per word.**
* **This requires a 7 bit address and 8-bit bidirectional data bus.**
* **The read and write inputs specify the memory operation and the two chips select (CS) control inputs are for enabling the chip only when it is selected by the microprocessor.**
* **The availability of more than one control input to select the chip facilitates the decoding of the address lines when multiple chips are used in the microcomputer.**
* **The read and write inputs are sometimes combined into one line labeled R/W.**
* **When the chip is selected, the two binary states in this line specify the two operations of read and write.**

**RAM function table:**

* **Function table specifies the operation of the RAM chip. The unit is in operation only when CS1=1 and CS2’=0.**
* **The bar on top of the second select variable indicates that this input is enabled when it is equal to 0.**
* **If the chip select inputs are not enabled, or if they are enabled but the read or write inputs are not enabled, the memory is inhibited and its data bus is in a high impedance state.**
* **When CS1=1 and CS2’=0, the memory can be placed in a write or read mode.**
* **When the WR input is enabled, the memory stores a byte from the data bus into a location specified by the address input lines.**
* **When the RD input is enabled, the content of the selected byte is placed into the data bus.**
* **The RD and WR signals control the memory operation as well as the bus buffers associated with the bidirectional data bus.**

**ROM chips:**



**ROM chips:**

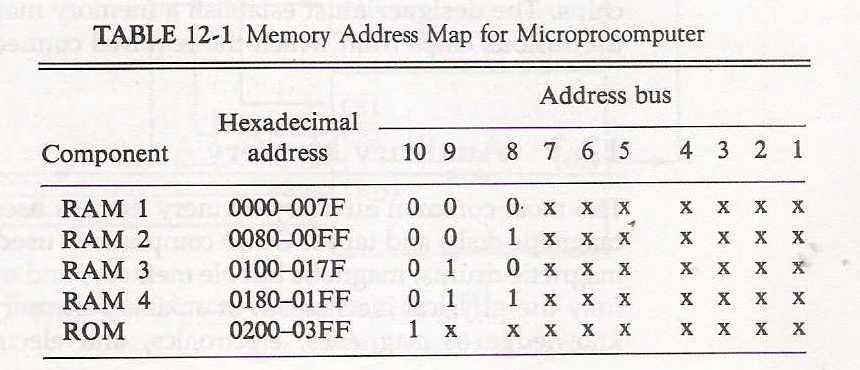
* **A ROM chip is organized externally in a similar manner. However since a ROM can only read, the data bus can only be in an output mode.**
* **For the same size chip, it is possible to have more bits of ROM than of RAM, because the internal binary cells in ROM occupy less space than in RAM.**
* **For this reason, the diagram specifies a 512-byte ROM, while the RAM has only 128 bytes.**
* **The nine address lines in the ROM chip specify any one of the 512 bytes stored in it.**
* **The two chip select inputs must be CS1=1 and CS2’=0for the unit to operate. Otherwise, the data bus is in a high impedance state.**
* **there is no need for a read or write control because the unit can only read. Thus when the chip is enabled by the two select inputs, the byte selected by the address lines appears on the data bus.**

**Memory address map:**

* **The designer of a computer system must calculate the amount of memory required for the particular application and assign it to either RAM or ROM.**
* **The interconnection between memory and processor is then established from knowledge of the size of memory needed and the type of RAM and ROM chips available.**
* **The addressing of memory can be established by means of a table that specified the memory address assigned to each chip.**
* **The table, called a memory address map, is a pictorial representation of assigned address space for each chip in the system.**
* **Memory address mapping for a computer system which needs 512 bytes of RAM and 512 bytes of ROM.**
* **Given : 128 X 8 RAM ,ADR 7 bits**

**: 512 X 8 ROM ,ADR 9 bits**

**Memory address map for microcomputer:**

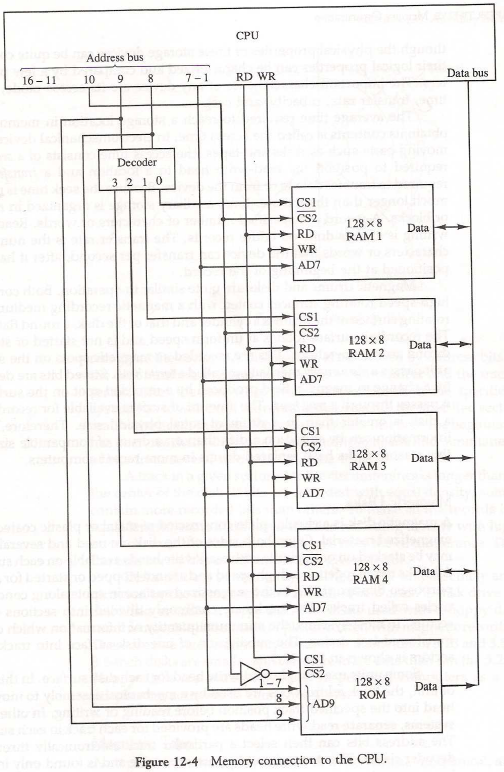


**Memory connection to the CPU:**

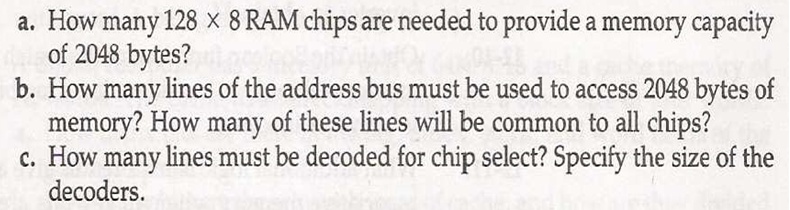
* **RAM and ROM chips are connected to a CPU through the data and address buses. The low order lines in the address bus select the byte within the chips and other lines in the address bus select a particular chip through its chip select inputs. Each RAM receives the seven low order bits of the address bus to select one of 128 possible bytes.**
* **The particular RAM chip selected is determined from lines 8 and 9 in the address bus. This is done through a 2 x 4 decoder whose outputs go to the CS1 inputs in each RAM chip. Thus, when address lines 8 and 9 are equal to 00, the first RAM chip is selected. When 01, the second RAM chip is selected, and so on.**
* **The RD and WR outputs from the microprocessor are applied to the inputs of each RAM chip. The selection between RAM and ROM is achieved through bus line 10. the RAMs are selected when the bit in this line is 0, and the ROM when the bit is 1. the other chip select input in the ROM is connected to the RD control line for the ROM chip to be enabled only during a read operation.**

**Address bus lines 1to 9 are applied to the input address of ROM without going through the decoder. This assigns addresses 0 to 511 to RAM and 512 to 1023 to ROM. The data bus of the ROM has only an output capability, whereas the data bus connected to the RAMs can transfer**

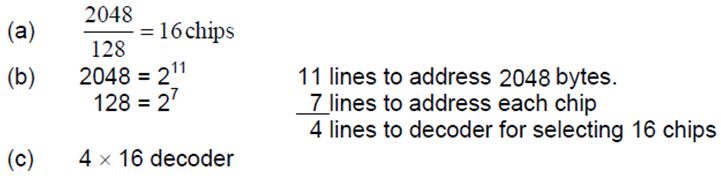
**Memory connection to the CPU:**



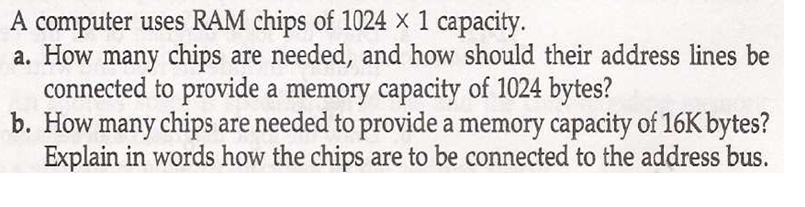
**Questions 1:**



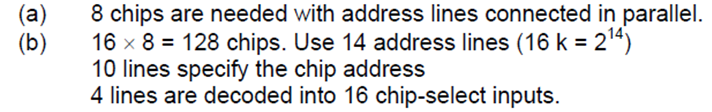
**Solution:**



**Questions 2:**



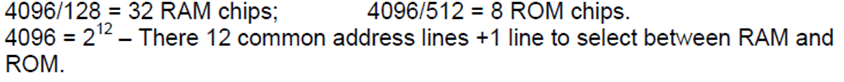
**Solution:**

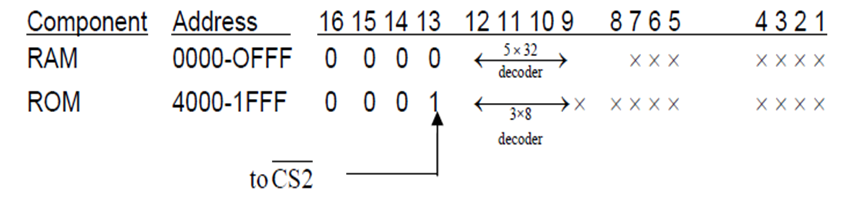


**Questions 3:**

* **Construct the memory of 4096 byte of RAM and 4096 bytes of ROM. List the memory address map and indicate what size decoders are needed.**

**Solution:**



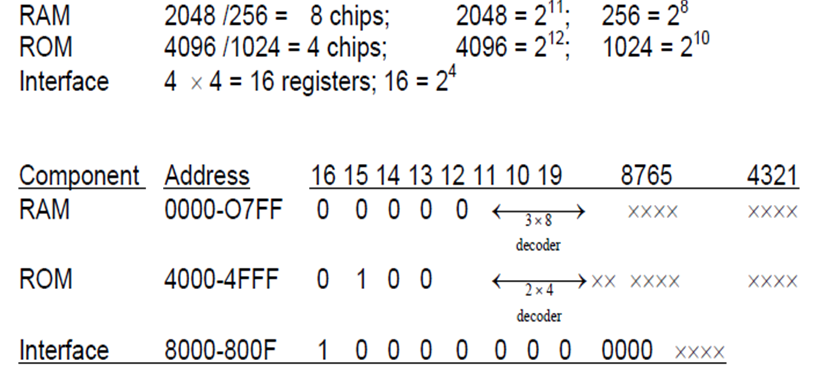


**Questions 4:**

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

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

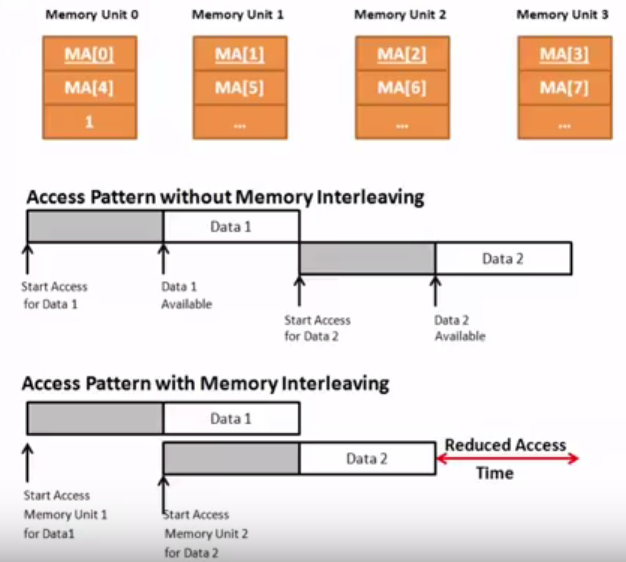
**Solution:**



**Multi-module memory and interleaving**:

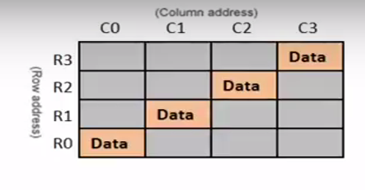
* Interleaving is the technique to interleave successive memory addresses across multiple memory units.
* Memory interleaving is the technique used to increase the throughput.
* The core idea is to split the memory system into independent banks, which can answer read or write requests independents in parallel.
* After a memory chip is accessed usually there is a delay before the chip can be accessed again.
* This delay is long enough to put a wait state in a fast CPU.
* However, with memory interleaving as a subsequent addresses are interleaved among memory units. Each units take turns to handle the request from the CPU, while the other recovering from the access.
* When the first unit sending the data to the CPU the second unit will receive the address from the CPU.
* Similarly, when the second memory unit sending the data to the CPU the first unit will receive the address from the CPU. As a result the delay between the accessing the subsequent words are reduced considerably.

**4 Way interleaved Memory:**



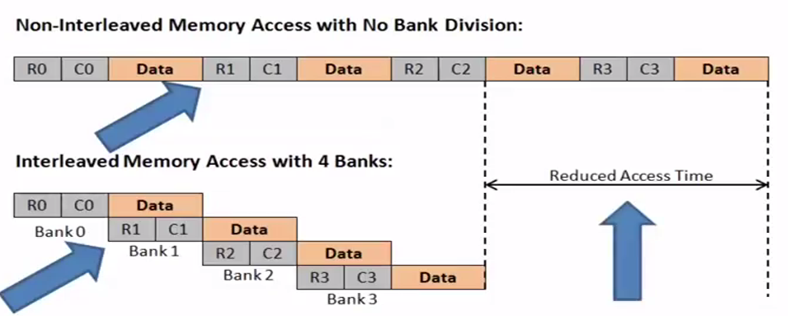
**Main memory:**

* Main memory consists of DRAM chips.
* Each DRAM chip is divided into memory banks and these memory banks are the 2- dimensional array that consist of row and column.
* The intersection of row and column is called cell where data is being stored in matrix form.

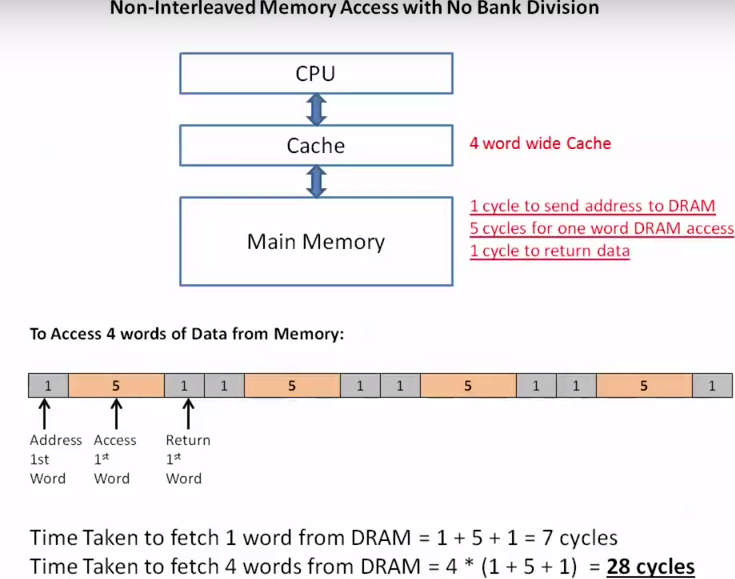


**Non interleaved VS. Interleaved:**

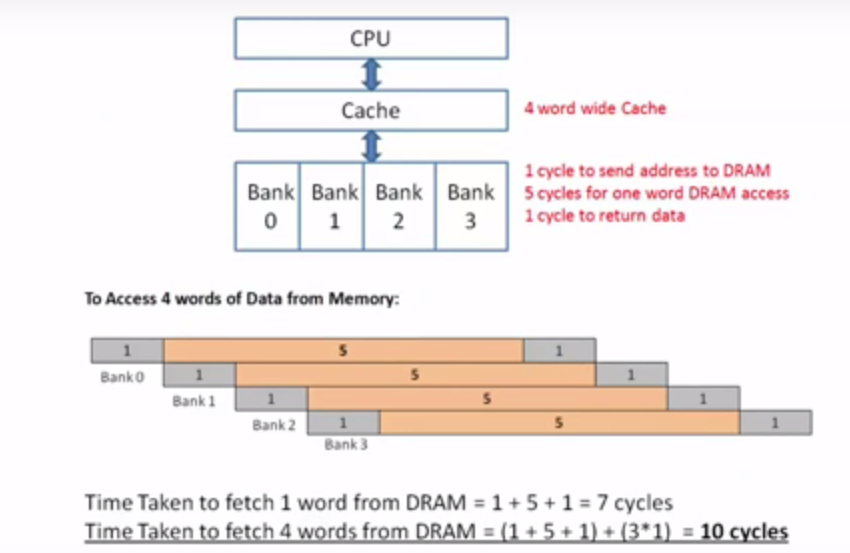
* In an non interleaved memory access, the memory addresses are accessed only when the previous memory access is completed.
* However in an interleaved memory access the access of subsequent memory addresses interleaved across banks can be overlapped.
* As a result, overall access time for same amount of data reduced considerably.



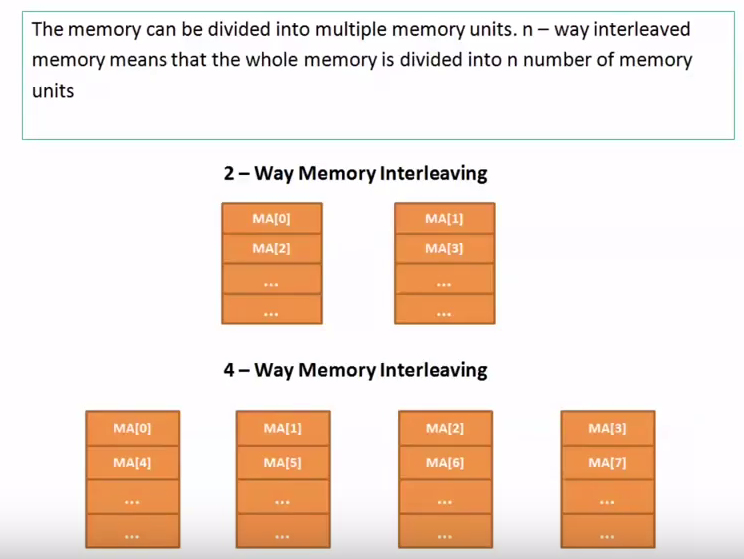
**Non interleaved Memory:**



**Interleaved Memory:**



**N-way interleaving:**



**The Address Space for Interleaving:**

* When a memory is N–way interleaved, we always find that N = 2K.    
  This is due to the structure of the memory address.
* For K = 1, we have 2–way interleaving.
* For K = 2, we have 4–way interleaving.
* For K = 3, we have 8–way interleaving.
* For K = 4, we have 16–way interleaving.
* For each scheme, the K bits of the address select the Module.

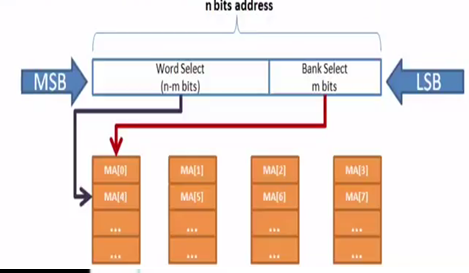
**Types of interleaving:**

There are two-address format for memory interleaving the address space.

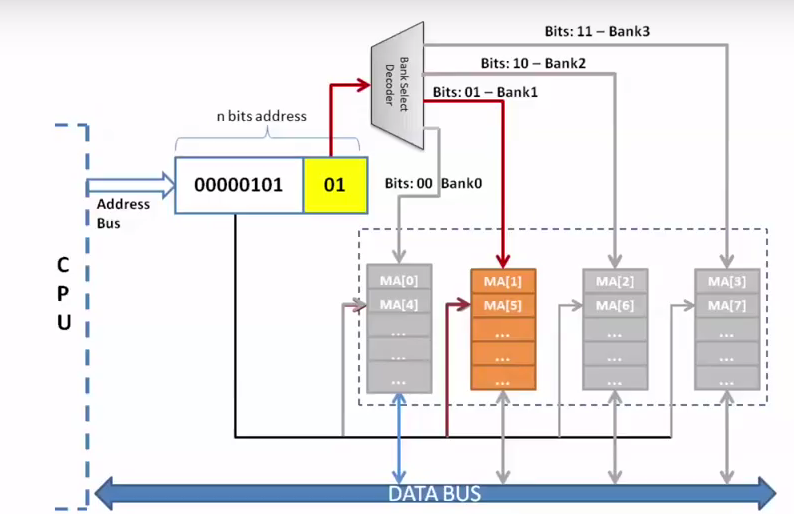
**Low order interleaving**: Low order interleaving spreads contiguous memory location across the modules horizontally. This implies that the low order bits of the memory address are used to indentify the memory module. High order bits are the word addresses within each module

**High order interleaving:** High order interleaving spreads contiguous memory location across the modules vertically. It uses the high order bits as the module address and the low order bits as the word address within each module.

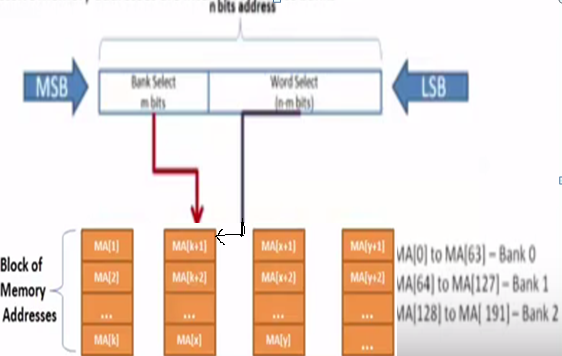
**Low order interleaving:**



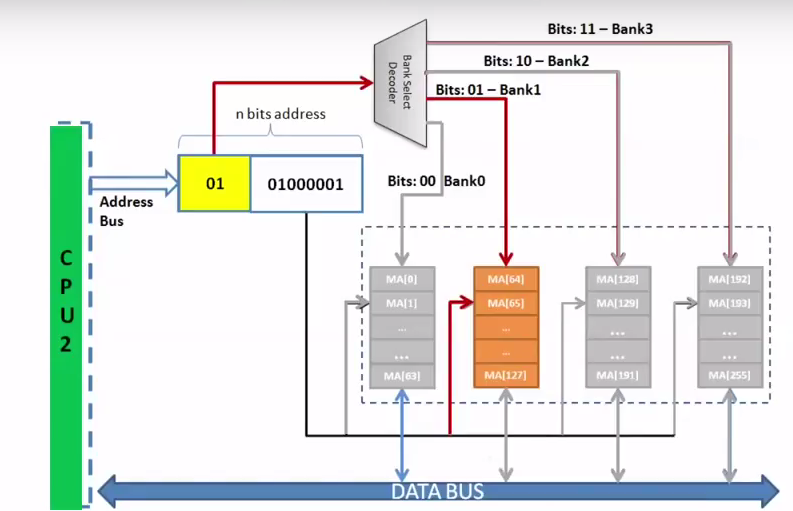
**Example: Low order interleaving:**



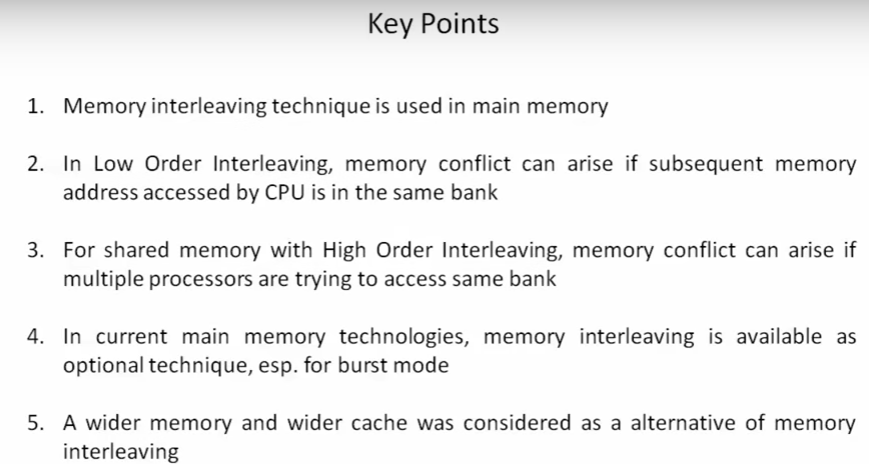
**High order interleaving:**



**Example: High order interleaving:**



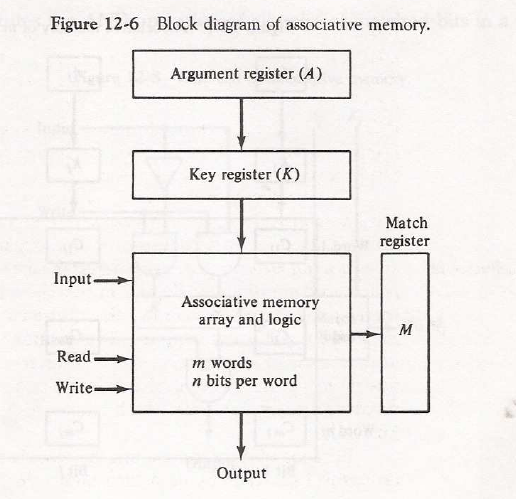




**Associative memory:**

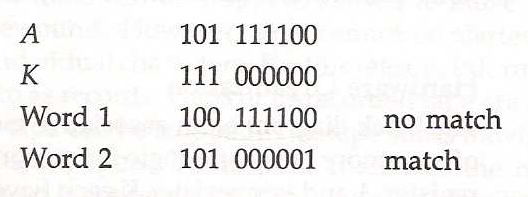
* The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by the content of the data itself rather than by an address.
* A memory unit accessed by content is called an **associative memory or content addressable memory (CAM**).
* This type of memory is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location.
* When a word is written in an associative memory, no address is given. The memory is capable of finding an empty unused location to store the word.
* When a word is to be read from an associative memory, the content of the word, or part of the word, is specified.
* The memory locates all words which match the specified content and marks them for reading.
* The entire argument is compared with each memory word if the key register contains all 1’s. Otherwise, only those bits in the argument that have 1’s in their corresponding position of the key register are compared.
* Thus the key provides a mask or identifying piece of information which specifies how the reference to memory is made.

**Block diagram of Associative memory:**

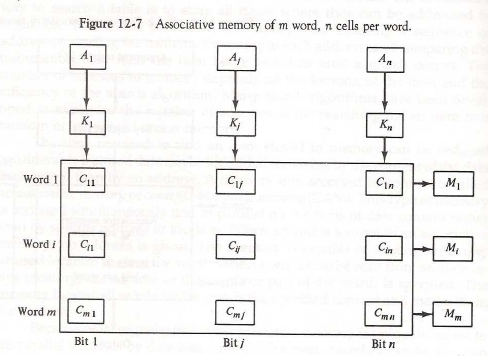


**Example:**

* **Suppose that the argument register A and the key register K have the bit configuration shown below. Only the three leftmost bits of A are compared with memory words because k has 1’s in these positions.**
* **Word 2 matches the unmasked argument field because the three leftmost bits of the argument and the word are equal.**



**Relation between memory array and external registers:**



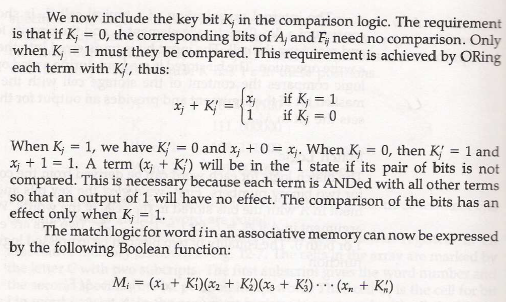
* The relation between the memory array and external registers in an associative memory is given in previous slide.
* The cells in the array are marked by the letter C with two subscripts. The first subscript gives the word number and the second specifies the bit position in the word.
* Thus cell C IJ is the cell for bit j in word i. A bit Aj in the argument register is compared with all the bits in column j of the array provided that K j =1.
* This is done for all columns j=1,2,3…..n. if a match occurs between all the unmasked bits of the argument and the bits in word I, the corresponding bit M I in the match register is set to 1.
* If one or more unmasked bits of the argument and the word do not match, M I is cleared to 0.

**Match logic:**

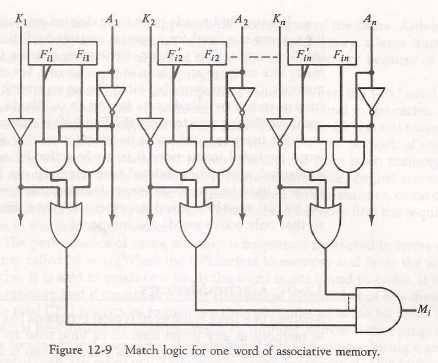
* The match logic for each word can be derived from the comparison algorithm for two binary numbers.
* First , we neglect the key bits and compare the argument in A with the bits stored in the cell of the words. Word I is equal to the argument in A if

Ai = F IJ for j=1,2,….n.

* Two bits are equal if they are both 1 or both 0. The equality of two bits can be expressed logically by the Boolean function
* Where XJ  = 1 if the pair of bits in position j are equal; otherwise, Xj  = 0.
* For a word I to be equal to the argument in A we must have all Xj variables equal to 1. This is the condition for setting the corresponding match bit Mi to 1. The Boolean function for this condition is
* Mi = X I X 2 X 3 ………. X n
* And constitutes the AND operation of all pairs of matched bits in a word.

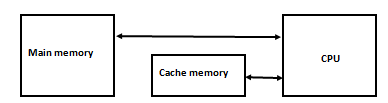






**CACHE MEMORY:**

Cache is a fast, small capacity memory that should hold those information which are most likely to be accessed



**Example of cache memory**

**Operation of cache:**

* If the CPU needs to access memory, the cache is examined. If the word is found in the cache, it is read from the fast memory.
* If the word addressed by the CPU is not found in the cache, the main memory is accessed to read the word.
* A block of words containing the one just accessed is then transferred from main memory to cache memory.

**Performance of Cache Memory System:**

* The performance of cache memory is frequently measured in terms of a quantity called **hit ratio**.
* When the CPU refers to memory and finds the word in cache, it is said to produce a **hit**. If the word is not found in cache , it is in main memory and it counts as a **miss**.
* The ratio of the number of hits divided by the total CPU references to memory is the **hit ratio**.

Te: Effective memory access time in Cache memory system

Tc: Cache access time

Tm: Main memory access time

Example: Tc: =100 ns , Tm: 1000 ns, h=0.9

Te = Tc + (1 - h) Tm

=100 + (1-0.9) 1000

=200 ns

This is a considerable improvement over a similar computer without a cache memory, whose access time is 1000 ns.

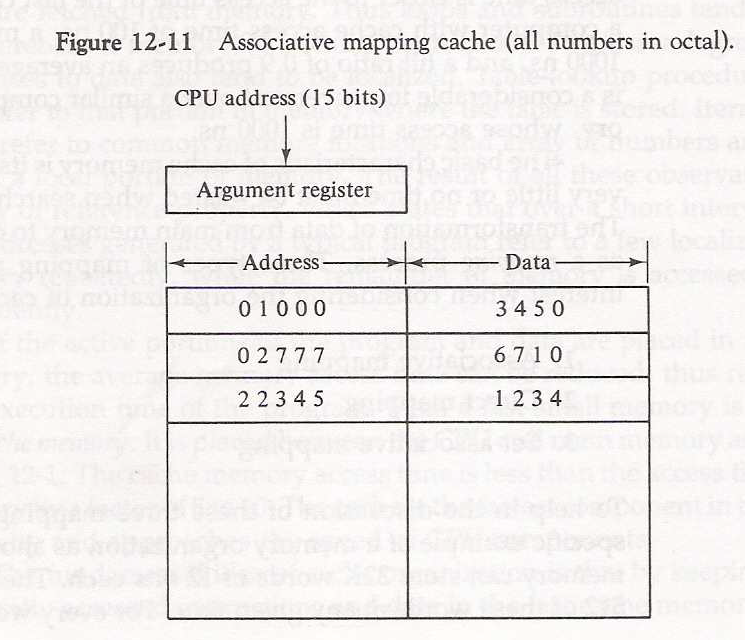
**Cache mapping:**

* The transformation of data from main memory to cache memory is referred to as a **mapping process**. Three types of mapping are
  1. Associative mapping
  2. Direct mapping
  3. Set associative mapping

**Associative Mapping:**

* The fastest and most flexible cache organization uses an associative memory.
* The associative memory stores both the address and content of the memory word.
* This permits any location in cache to store any word from main memory.
* The diagram shows three words presently stored in the cache.
* The address value of 15 bits is shown as a 5 digit octal number and its corresponding 12 bit word is shown as a four digit octal number.
* A CPU address of 15 bits is placed in the argument register and the associative memory is searched for a matching address.
* If the address is found, the corresponding 12 bit data is read and sent to the CPU. If no match occurs, the main memory is accessed for the word.
* The address-data pair is then transferred to the associative cache memory. If the cache is full, an address-data pair must be displaced to make room for pair that is needed and not presently in the cache.

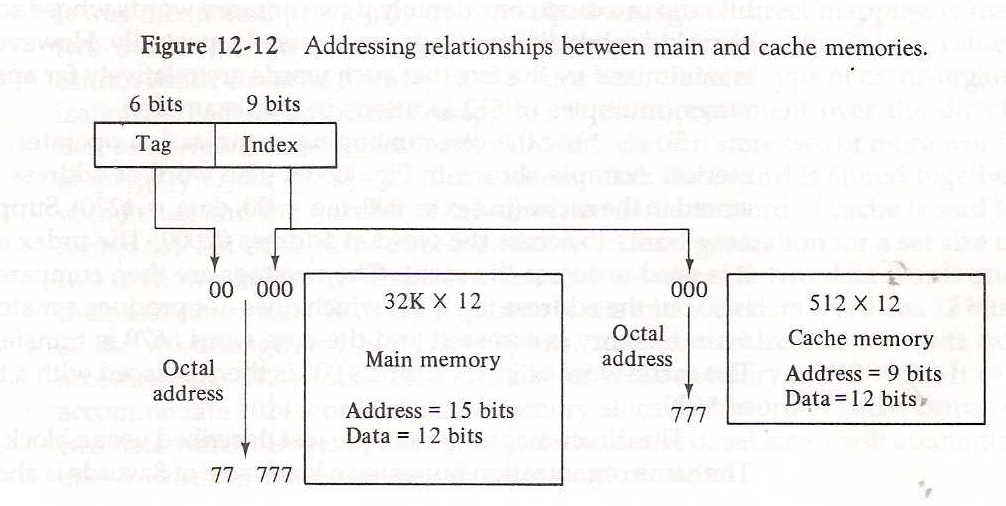
**Associative Mapping cache(all numbers in octal):**



**Direct mapping:**

* Associative memories are **expensive** compared to random access memories because of the added logic associated with each cell.
* The possibility of using a random access memory for the cache is shown in slide 47.
* The CPU address of 15 bits is divided into two fields. The nine LSB constitute the index field and the remaining six bits from the tag field.
* The figure shows that main memory needs an address that includes both the tag and the index bits. The number of bits in the index field is equal to the number of address bit required to access the cache memory.
* In the general case, there are 2k  words in cache memory and 2n words in main memory. The n-bit memory address is divided into two fields: k bits for the index field and n-k bits for the tag field.
* The direct mapping cache organization uses the n bit address to access the main memory and the k-bit index to access the cache.
* The internal organization of the words in the cache memory is shown in slide 49.
* Each word in cache consists of the data word and its associated tag.
* When a new word is first brought into the cache, the tag bits are stored alongside the data bits.
* When the CPU generates a memory request, the index field is used for the address to access the cache. The tag field of the CPU address is compared with the tag in the word read from the cache. If the two tags match, there is a hit and the desired data word is in cache.
* If there is no match, there is a miss and the required word is read from main memory. It is then stored in the cache together with the new tag, replacing the previous value.
* The **disadvantage** of direct mapping is that the hit ratio can drop considerably if two or more words whose addresses have the same index but different tags are accessed repeatedly.

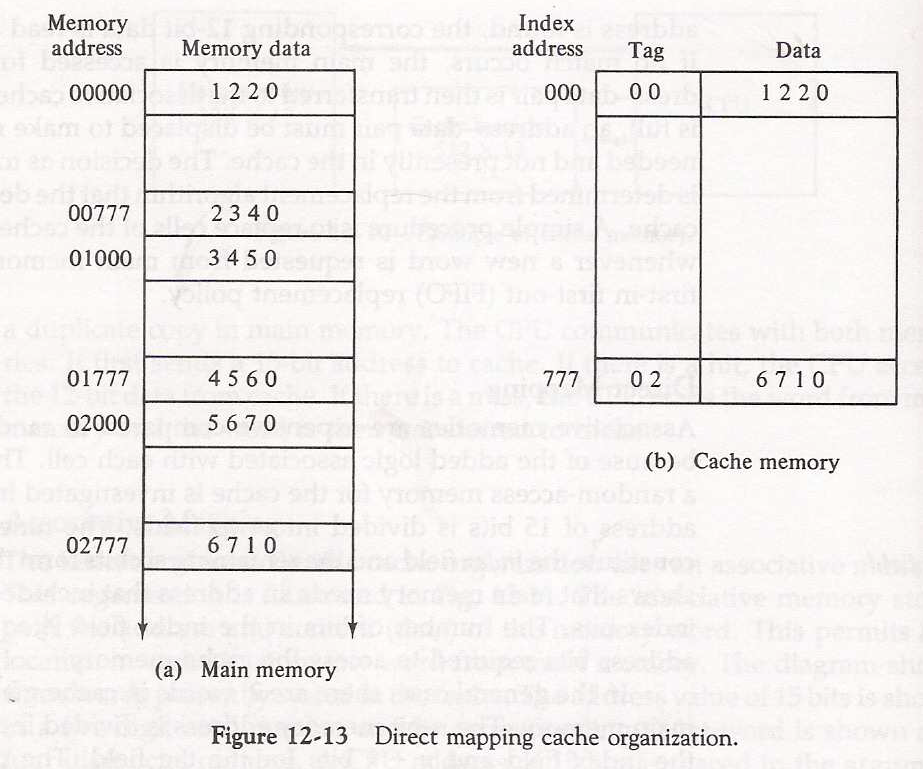
**Addressing relationships between main and cache memories:**



**Direct mapping organization operation using one word:**

* In figure, the word at address zero is presently stored in the cache (index=000, tag=00, data=1220) .
* Suppose that the CPU now wants to access the word at address 02000. the index address is 000, so it is used to access the cache. The two tags are then compared. The cache tag is 00 but the address tag is 02, which does not produce a match.
* Therefore the main memory is accessed and the data word 5670 is transferred to the CPU. The cache word at index address 000 is then replaced with a tag of 02 and data of 5670.

**Direct mapping cache organization:**



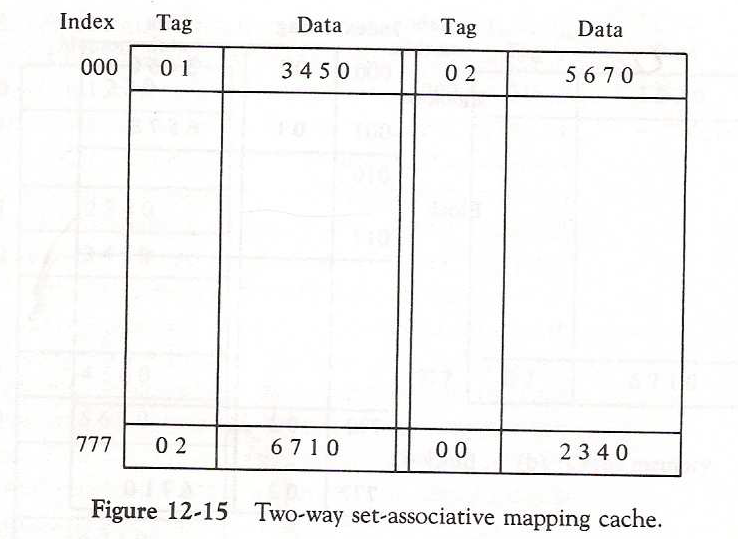
**Set associative mapping:**

* The **disadvantage** of direct mapping is that two words with the same index in their address but with different tag values cannot reside in cache memory at the same time.
* A third type of cache organization, called set-associative mapping, is an improvement over the direct mapping organization in that each word of cache can store two or more words of memory under the same index address.
* Each data word is stored together with its tag and the number of tag data items in one word of cache is said to form a set.

**Example of set associative mapping:**

* In the slide 52, each address refers to two data words and their associated tags, each tag requires six bits and each data word has 12 bits, so the word length is 2(6+12)=36 bits.
* An index address of nine bits can accommodate 512 words. Thus, the size of cache memory is 512 x 36.
* The octal numbers listed in figure are with reference to the main memory contents. The words stored at addresses 01000 and 02000 of main memory are stored in cache memory at index address 000.similarly, the words at addresses 02777 and 00777 are stored in cache at index address 777.
* When the CPU generates a memory request, the index value of the address is used to access the cache. The tag field of the CPU address is then compared with both tags in the cache to determine if a match occurs.
* The comparison logic is done by an associative search of the tags in the set similar to an associative memory search: thus the name “set associative”.

**Two-way set associative mapping cache:**



**CACHE COHERENCE:**

For higher performance in a multiprocessor system, each processor will usually have its own cache. Cache coherence refers to the problem of keeping the data in these caches consistent. The main problem is dealing with writes by a processor.

There are two general strategies for dealing with writes to a cache:

**Write-through -** all data written to the cache is also written to memory at the same time.

**Write-back -** when data is written to a cache, a dirty bit is set for the affected block. The modified block is written to memory only when the block is replaced.

**SOLUTION FOR CACHE COHERENCE:**

**SNOOPY PROTCOL:**

* Snoopy protocols distribute the responsibility for maintaining cache coherence among all of the cache controllers in a multiprocessor system.
* A cache must recognize when a line that it holds is shared with other caches.
* When an update action is performed on a shared cache line, it must be announced to all other caches by a broadcast mechanism.
* Each cache controller is able to “snoop” on the network to observed these broadcasted notification and react accordingly.
* Snoopy protocols are ideally suited to a bus-based multiprocessor, because the shared bus provides a simple means for broadcasting and snooping.
* Two basic approaches to the snoopy protocol have been explored: Write invalidates or write- update (write-broadcast)
* With a write-invalidate protocol, there can be multiple readers but only one write at a time.
* Initially, a line may be shared among several caches for reading purposes.
* When one of the caches wants to perform a write to the line it first issues a notice that invalidates that tine in the other caches, making the line exclusive to the writing cache. Once the line is exclusive, the owning processor can make local writes until some other processor requires the same line.
* With a write update protocol, there can be multiple writers as well as multiple readers. When a processors wishes to update a shared line, the word to be updated is distributed to all others, and caches containing that line can update it.

**Directory protocols:**

* Directory protocols collect and maintain information about where copies of lines reside. Typically, there is centralized controller that is part of the main memory controller, and a directory that is stored in main memory.
* The directory contains global state information about the contents of the various local caches.
* When an individual cache controller makes a request, the centralized controller checks and issues necessary commands for data transfer between memory and caches or between caches themselves.
* It is also responsible for keeping the state information up to date, therefore, every local action that can effect the global state of a line must be reported to the central controller.
* The controller maintains information about which processors have a copy of which lines.
* Before a processor can write to a local copy of a line, it must request exclusive access to the line from the controller.
* Before granting thus exclusive access, the controller sends a message to all processors with a cached copy of this time, forcing each processors to invalidate its copy.
* After receiving acknowledgement back from each such processor, the controller grants exclusive access to the requesting processor.
* When another processor tries to read a line that is exclusively granted to another processors, it will send a miss notification to the controller.
* The controller then issues a command to the processor holding that line that requires the processors to do a write back to main memory.
* Directory schemes suffer from the drawbacks of a central bottleneck and the overhead of communication between the various cache controllers and the central controller.

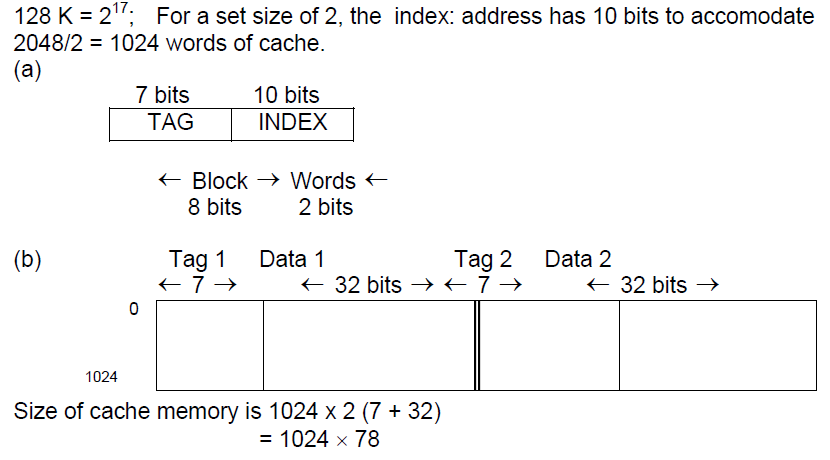
**Numerical on cache memory:**

**Question 1:**

* A tow-way set associative cache memory uses block of four words. The cache can accommodate a total of 2048 words from main memory. The main memory size is 128K X 32.

1. Formulate all pertinent information required to construct the cache memory.
2. What is the size of the cache memory?

**Solution 1:**



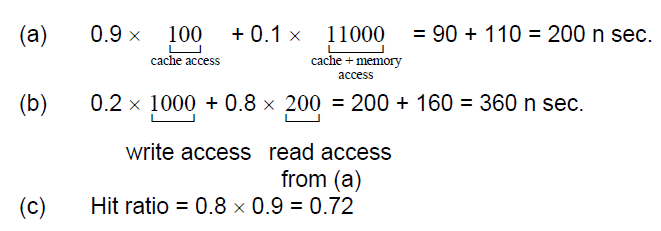
**Question 2:**

* The access time of a cache memory is 100 ns and that of main memory 1000 ns. It is estimated that 80 % of the memory requests are for read and the remaining 20% for write. The hit ratio for read accesses only is 0.9. A write through procedure is used.

1. What is the average access time of the system considering only memory read cycles.
2. What is the average access time of the system for both read and write requests?
3. What is the hit ratio taking into consideration the write cycles?

**Solution 2:**

1. **0.9 X 100+ 0.1 X 1100 =90 +110=200 ns**

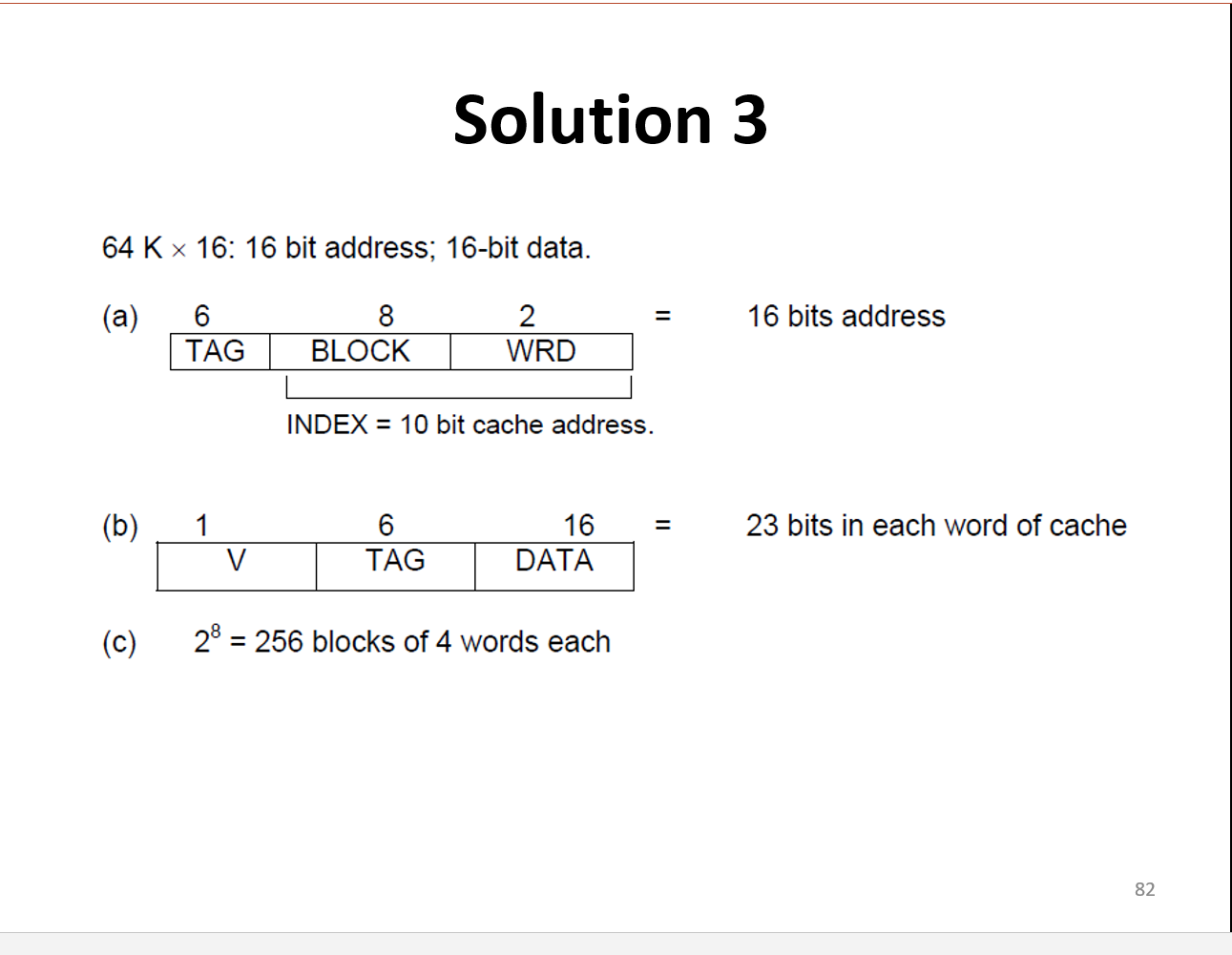




**Question 3:**

* A digital computer has a memory unit of 64K x16 and a cache memory of 1K words. The cache uses direct mapping with a block size of four words.

1. How many bits are there in the tag, index, block and word fields of the address format?
2. How many bits are there in each word of cache, and how are they divided into functions? Include a valid bit.
3. How many blocks can the cache accommodate?

****

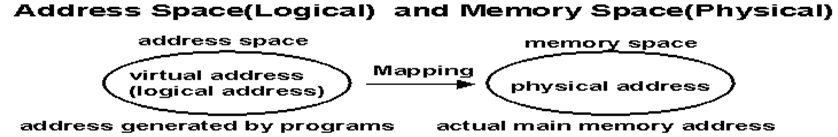
**Virtual Memory:**

* **Computers have a finite amount of RAM so memory can run out, especially when multiple programs run at the same time.**
* **A system using virtual memory can load larger programs or multiple programs running at the same time, allowing each one to operate as if it has infinite memory and without having to purchase more RAM.**
* **Virtual memory give the programmer the illusion that the system has a very large memory, even though the computer actual has a relatively small physical memory.**

**Address space and memory space:**

* **An address used by a programmer will be called a virtual address, and the set of such addresses the address space.**
* **An address in main memory is called a location or physical address. The set of such locations is called the memory space.**

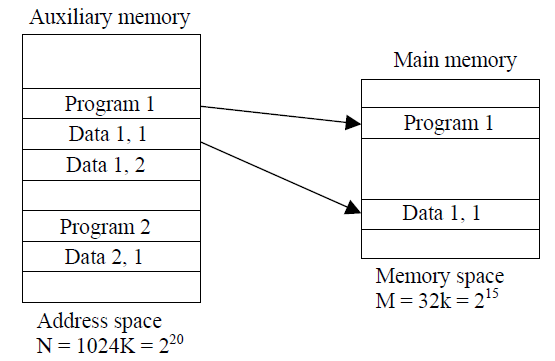
**Memory Mapping:**



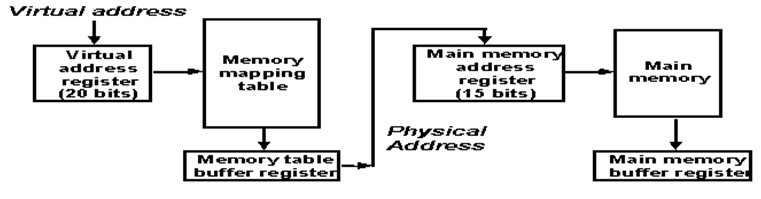
**Relation between memory and address space:**

* **Consider a computer with a main-memory capacity of 32K words (K =1024).**
* **Fifteen bits are needed to specify a physical address in memory since 32K = 215.**
* **Suppose that the computer has available auxiliary memory for storing 220 = 1024K words. Thus auxiliary memory has a capacity for storing information equivalent to the capacity of 32 main memories.**
* **Denoting the address space by N and the memory space by M, we then have for this example N = 1024K and M = 32K.**
* **In a multi-program computer system, programs and data are transferred to and from auxiliary memory and main memory based on demands imposed by the CPU.**
* **Suppose that program 1 is currently being executed in the CPU. Program 1 and a portion of its associated data are moved from auxiliary memory into main memory as shown in next slide.**
* **Portions of programs and data need not be in contiguous locations in memory since information is being moved in and out, and empty spaces may be available in scattered locations in memory.**

**Relation between memory and address space:**



**Memory table for mapping a virtual address:**

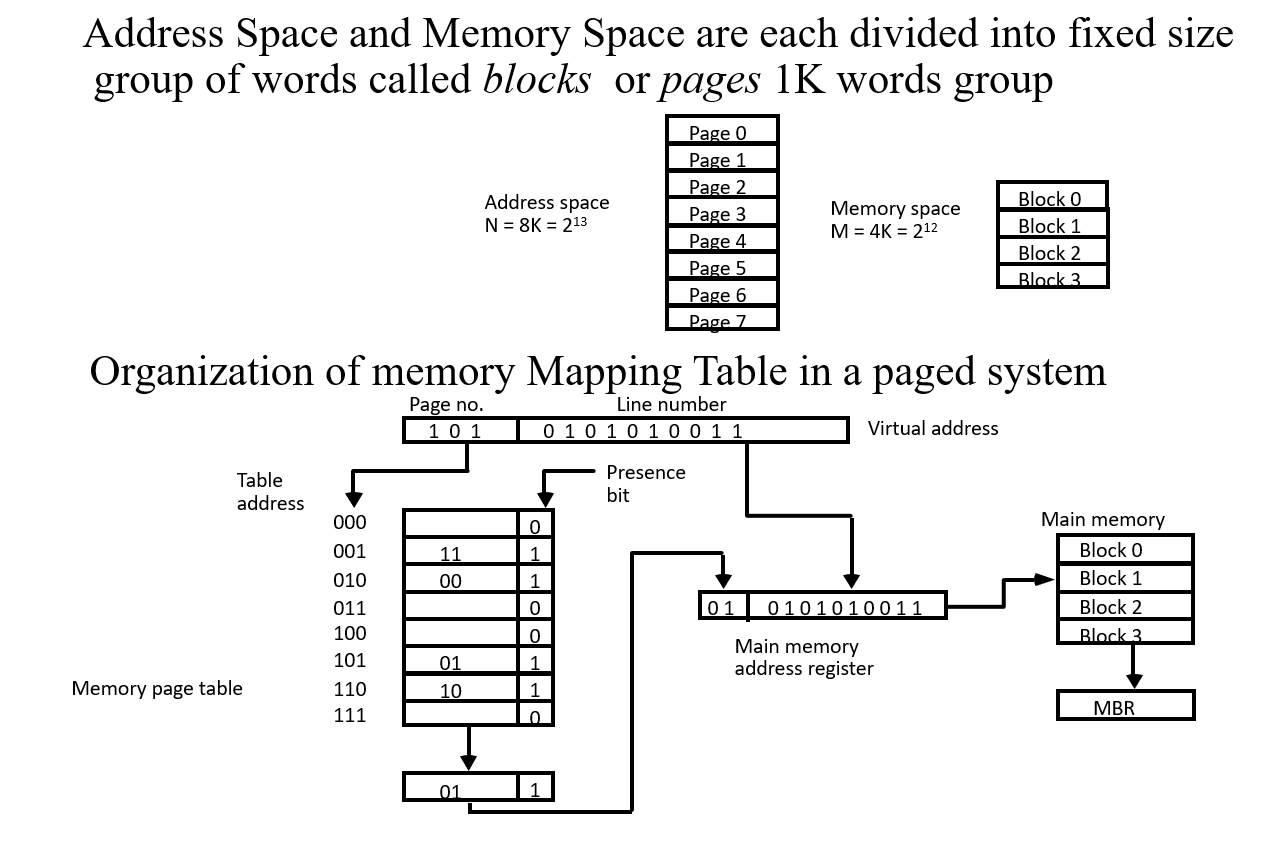


**Memory Mapping Table for Virtual Address to Physical Address:**

* **In a virtual memory system, programmers are told that they have the total address space at their disposal. Moreover, the address field of the instruction code has a sufficient number of bits to specify all virtual addresses.**
* **In the above example, the address field of an instruction code will consist of 20 bits but physical memory addresses must be specified with only 15 bits.**
* **Thus, CPU will reference instructions and data with a 20 bit address, but the information at this address must be taken from physical memory.**
* **A table is then needed, to map a virtual address of 20 bits to a physical address of 15 bits.**
* **The mapping is a dynamic operation, which means that every address is translated immediately as a word is referenced by CPU.**
* **The mapping table may be stored in a separate memory or in main memory. In the first case, an additional memory unit is required as well as one extra memory access time.**

**In the second case, the table takes space from main memory and two accesses to memory are required with the program running at half speed.**

**ADDRESS MAPPING:**

****

**Dirty bit (presence bit):**

* **A dirty bit is a flag that indicates whether an attribute needs to be updated. Such situations usually occur when a bit in a memory cache or virtual memory page that has been changed by a processor but has not been updated in the storage.**

**Address mapping:**

* **The table implementation of the address mapping simplified if the information in the address space and the memory space are each divided into groups of fixed size.**
* **The physical memory is broken down into groups of equal size called blocks.**
* **The term page refers to groups of address space of the same size.**
* **Portion of programs are moved from auxiliary memory to main memory in records equal to the size of a page. The term “page frame” is sometimes used to denote a block.**
* **The mapping from address space to memory space is facilitated if each virtual address is considered to be represented by two numbers: a page number address and line within the page.**
* **The organization of the memory mapping table in a paged system is shown in above figure.**
* **The memory page table consists of eight words, one for each page. The address in the page table denotes the page number and the content of the word gives the block number where that page is stored in main memory. The table shows that pages 1, 2, 5 and 6 are now available in main memory in blocks 3,0,1 and 2 respectively.**
* **A presence bit in each location indicates whether the page has been transferred from auxiliary memory into main memory.**
* **A 0 in the presence bit indicates that this page is not available in main memory. If the presence bit is a 1, the block number thus read is transferred to the two high order bits of the main memory address register.**
* **The line number from the virtual address is transferred into the 10 low order bits of the memory address register. A read signal to main memory transfers the content of the word to the main memory buffer register ready to be used by the CPU.**
* **If the presence bit in the word read from the page table is 0, it signifies that the content of the word referenced by the virtual address does not reside in main memory. A call to the operating system is then generated to fetch the required page from auxiliary memory and place it into main memory before resuming computation.**

**Question 1:**

* **An address space is specified by 24 bits and the corresponding memory space by 16 bits.**

1. **How many words are there in the address space?**
2. **How many words are there in the memory space?**
3. **If a page consists of 2K words, how many pages and blocks are there in the system?**

**Solution:**

