Sample questions: lecture-5:

RAM internal architecture and design of Memory Module: expanding address and word sizes

What is RAM? Show the simple architecture of RAM

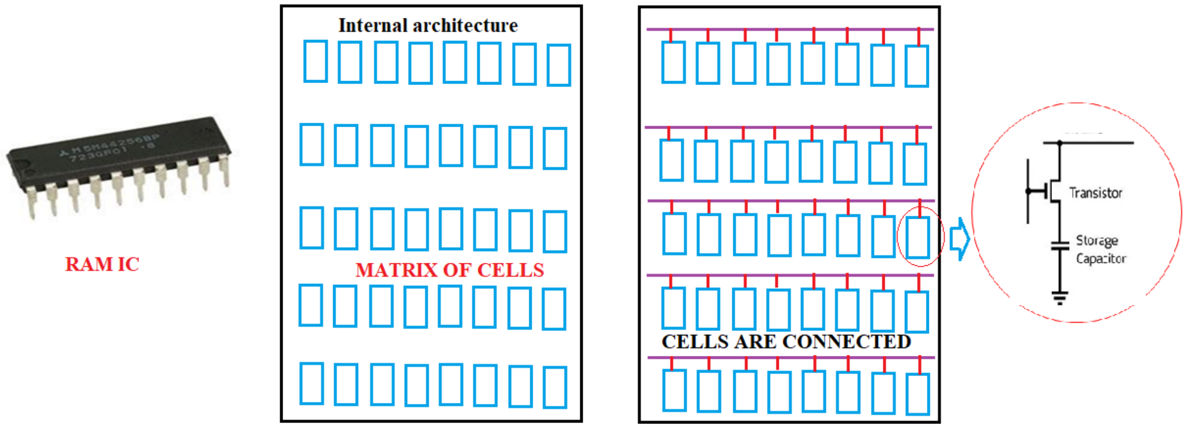
RAM is a semiconductor device, used as main memory in computers. It holds program and data as strings of binary bits: 1’s and 0’s.

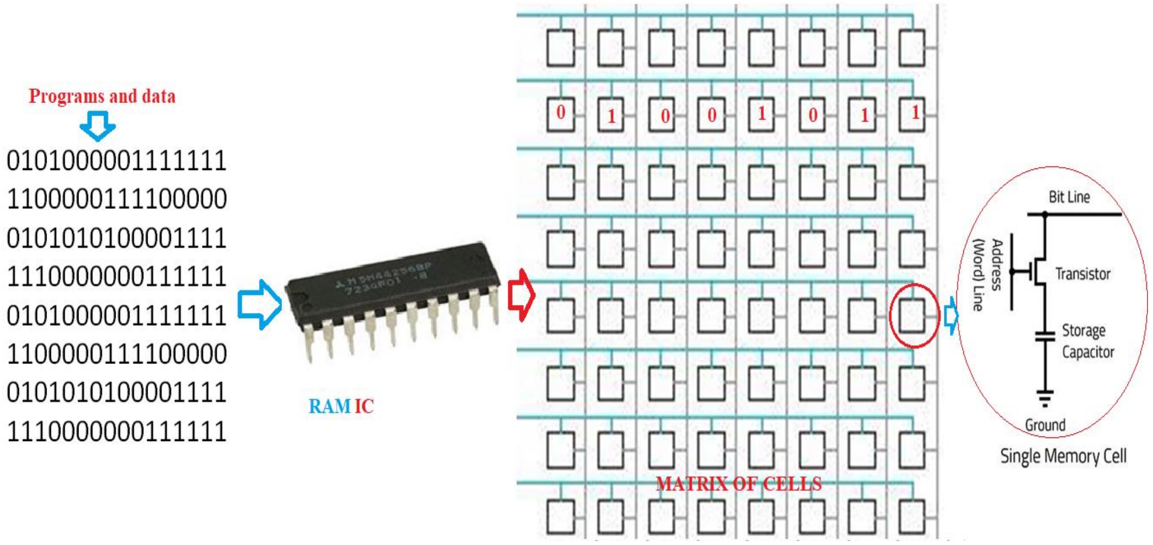
Main memory is designed to as a matrix (rows and columns) of cells and each cell can store only one bit; 1 or 0. Number of bits stored in a memory device depend on the product of row and column. This is also called the capacity of memory device in terms of bits.

Data and instructions are usually represented by 8 bits or multiple of 8 bits, a number of cells (usually minimum 8 cells) are fabricated in each row and these cells are connected together and used together as well.

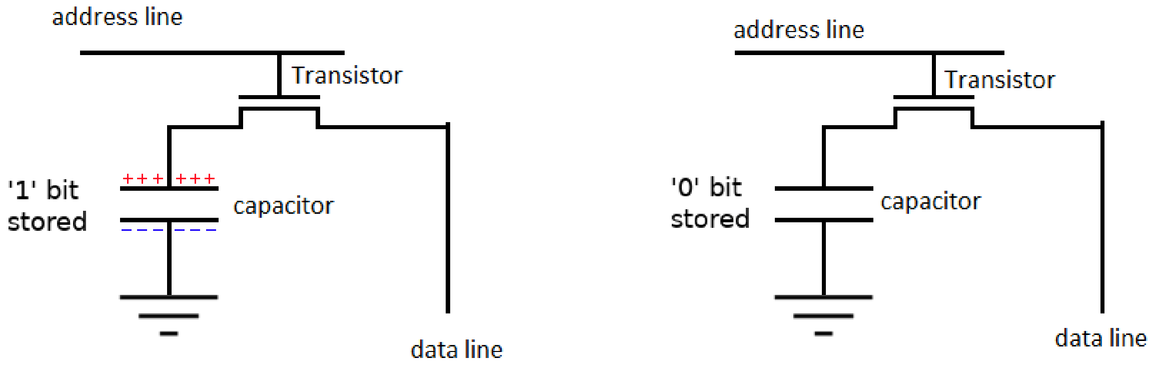
8-bit data is stored in 8-cells in a row. 16-bit data is stored in two consecutive rows where each row having 8-cells. Machine codes are also usually represented by 8 bits or multiple of 8 bits and a row or multiple consecutive rows are used to store instruction in RAM depending on the number of bits in machine codes.

Each cell is designed to have one tiny capacitor. By charging the capacitor, logical 1 is saved whereas by discharging a capacitor, logical 0 is saved in each cell. In order to select a cell, a transistor is also used with it as shown below.

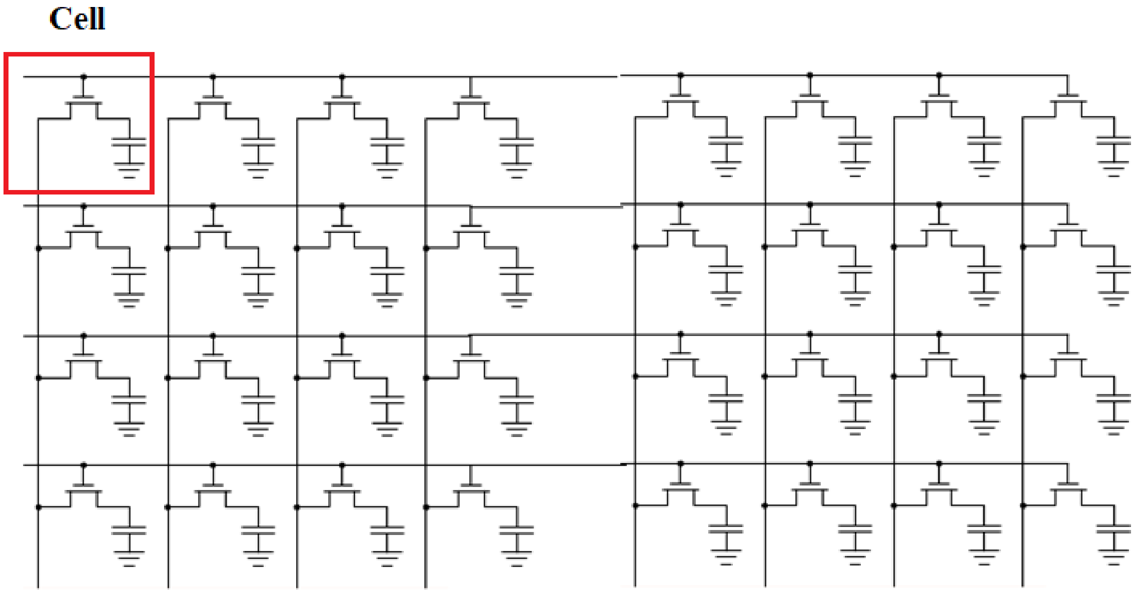




Each cell is designed to have one tiny capacitor. By charging the capacitor, logical 1 is saved whereas by discharging a capacitor, logical 0 is saved in each cell. In order to select a cell, a transistor is also used with it as shown below.



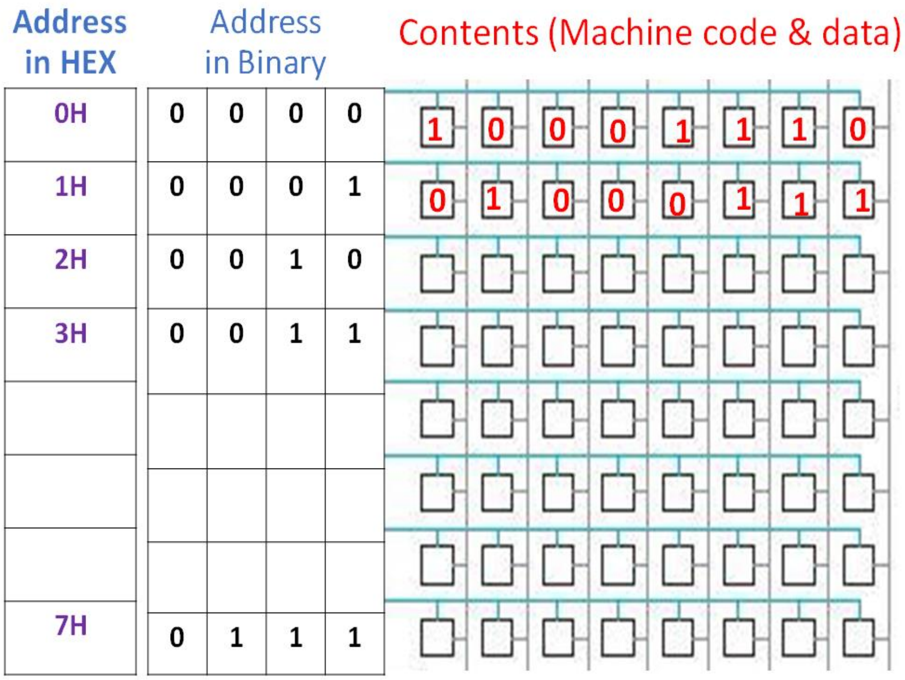
The following diagram shows electronic design of cells in RAM.



All 8-cells in a row are connected together and used together, either to hold data or instructions if they are represented by 8-bits each. Cells of a row do not hold multiple data or instructions, in general, whatever be the number of cells in a row. However, if any data or instruction require multiple of 8-bits, multiple but consecutive rows are used in those cases. For example, if any data requires 16-bits to represent it, 2-consecutive rows, each having 8-cells, are required to save that.

Each row of RAM (main memory) is considered as a location and it is designed to hold 8-bits in most common cases. Data or instruction is stored in a row or in a multiple but consecutive rows depending on number of bits required to represent it. In order to save any data or machine code of instruction to RAM, it is required to specify the row. Similarly, to read any data or instruction from RAM, it is also required to specify the row. Each row is uniquely identified by a code, called address of that row or address of that location in RAM. Address of a row (address of a memory location) is usually the row number which is assigned sequentially, starting with zero and thus it is a unique code for a row as well as for a memory location.

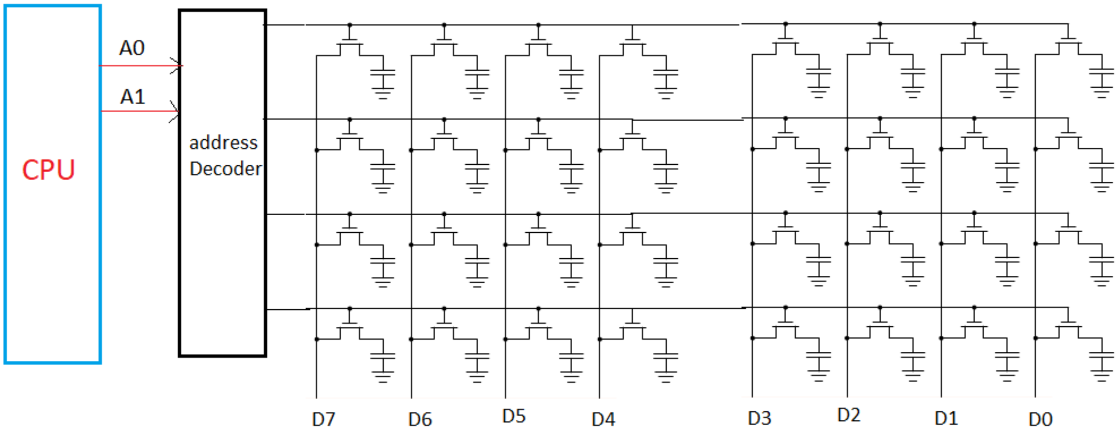
Each row is uniquely identified by a number/code, starting from ‘0’, called Address, usually represented in Hexadecimal form and used in Assembly language programming



Address in HEX	Address in Binary				Contents (Machine code & data)							
0H	0	0	0	0	1	0	0	0	1	1	1	0
1H	0	0	0	1	0	1	0	0	0	1	1	1
2H	0	0	1	0								
3H	0	0	1	1								
7H	0	1	1	1								

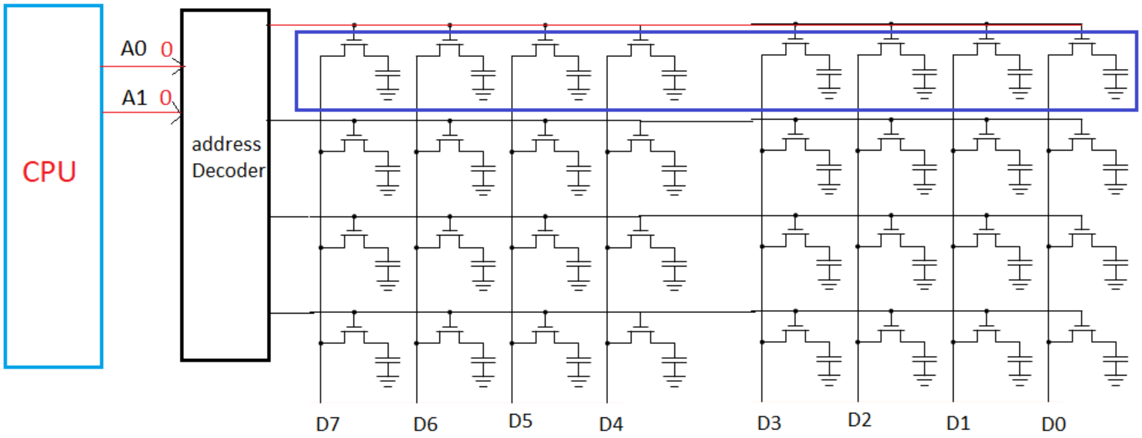
Memory addresses are, in fact, row numbers, assigned sequentially and represented in binary. If the memory capacity if very high, row numbers would be high as well and many binary bits are required to represent memory addresses. For ease of representation, hexadecimal number system is used in assembly language programming to represent memory addresses.

In order to select a particular memory location (particular row) electronically, a decoder is used, called address decoder. This address decoder also acts as interface between CPU and RAM, as shown below:

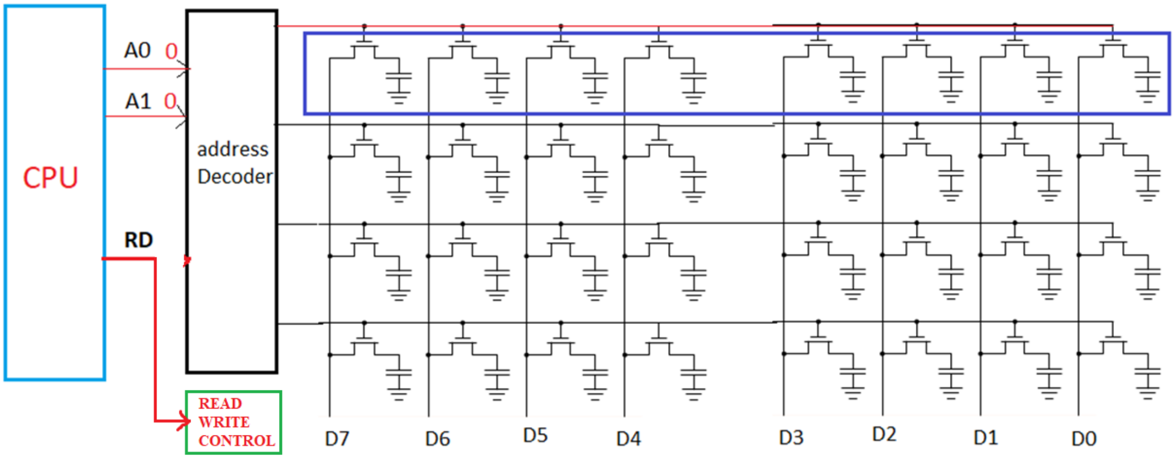


The input lines of decoder are called address bus. Output lines of address decoders are connected to common lines connecting cells of rows. The input lines of address decoder, called address bus of RAM are connected to address buses of CPU.

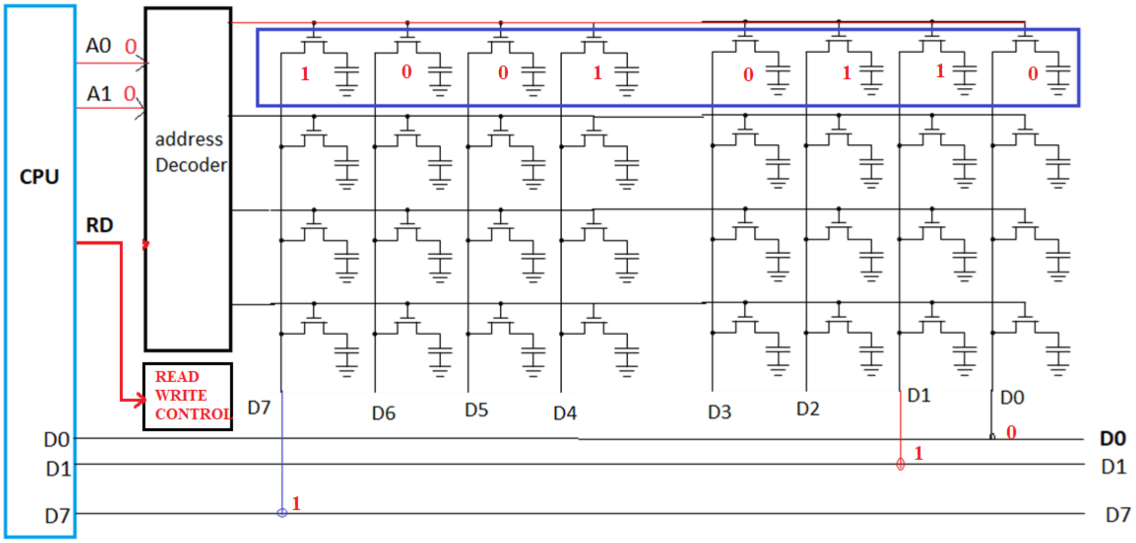
To read information stored in cells of first row, the CPU sends “0 0”, the address of the first row to address decoder. As a result, only the first line of address decoder is activated, which in turn, activates all cells connected to first row. This is how a particular memory location is selected, either for READ or for WRITE. Please note that, depending on address, only cells of a particular row are selected. All other cells remain inactive.



To READ from the selected row, a READ control signal is sent from CPU to RAM

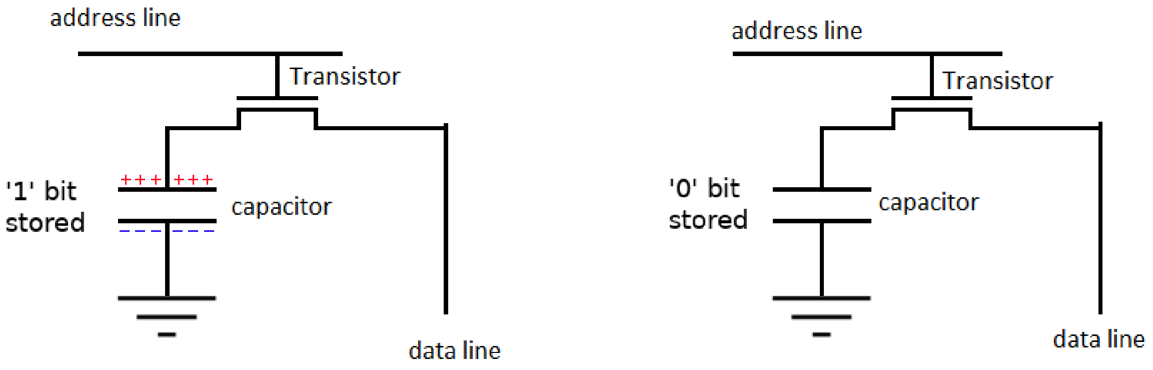


The contents of cells are sensed by read/write control circuit and transferred to data bus.



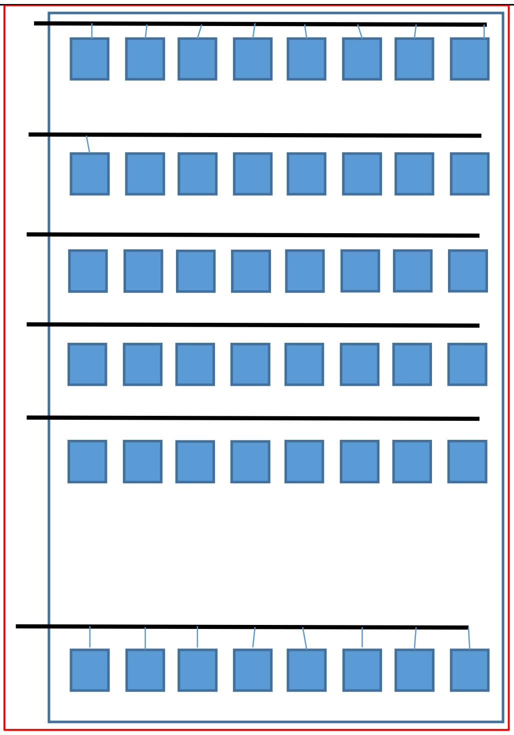
How 1’s and o’s are stored in primary memory?

Each cell is designed to have one tiny capacitor. By charging the capacitor, logical 1 is saved whereas by discharging a capacitor, logical 0 is saved in each cell. In order to select a cell, a transistor is also used with it as shown below.

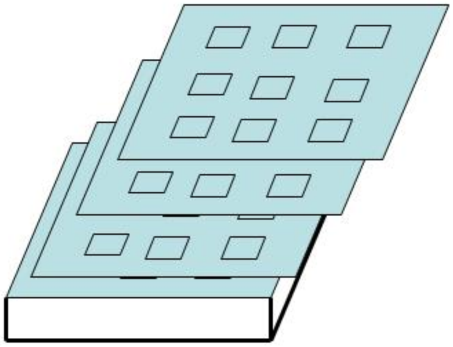


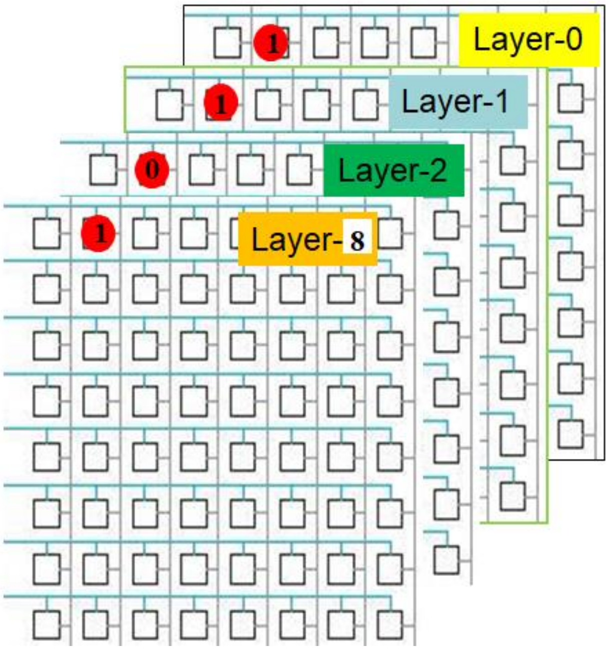
How cells are arranged in semiconductor RAM?

In low capacity RAMs, cells are arranged in rows and columns. For example, in a 1KB RAM, there is a total of 8192 cells and these cells are arranged in 1024 rows and 8 columns).



In a very high capacity RAM, there are multiple surfaces (layers) fabricated one above another. Each surface contains a number of rows and columns. For example, in a 16MB RAM; there are 8 layers and each layer contains 2^{12} x 2^{12} cells. In such structure, an 8-bit data is saved in eight different layers but exactly at same location of each layer (surface).

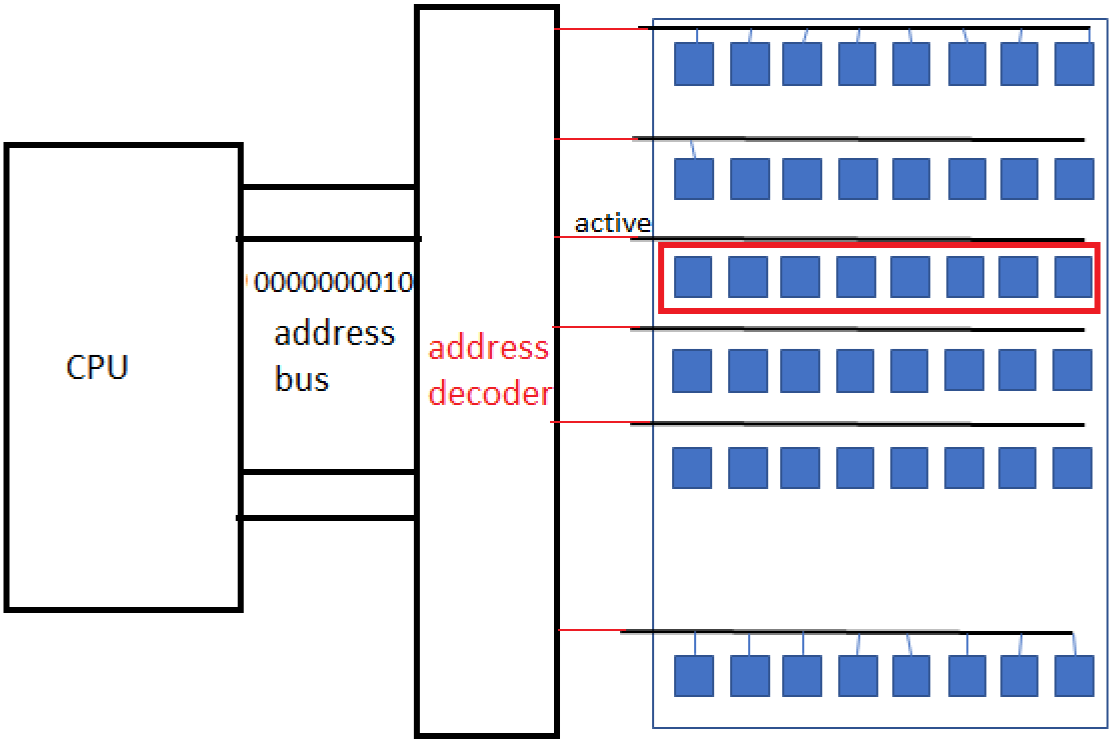




How a memory location is selected?

Each memory device is associated with an address decoder. Inputs of address decoders are connected with address bus of CPU whereas outputs of address decoders are connected with cells in row-wise manner, as shown below.

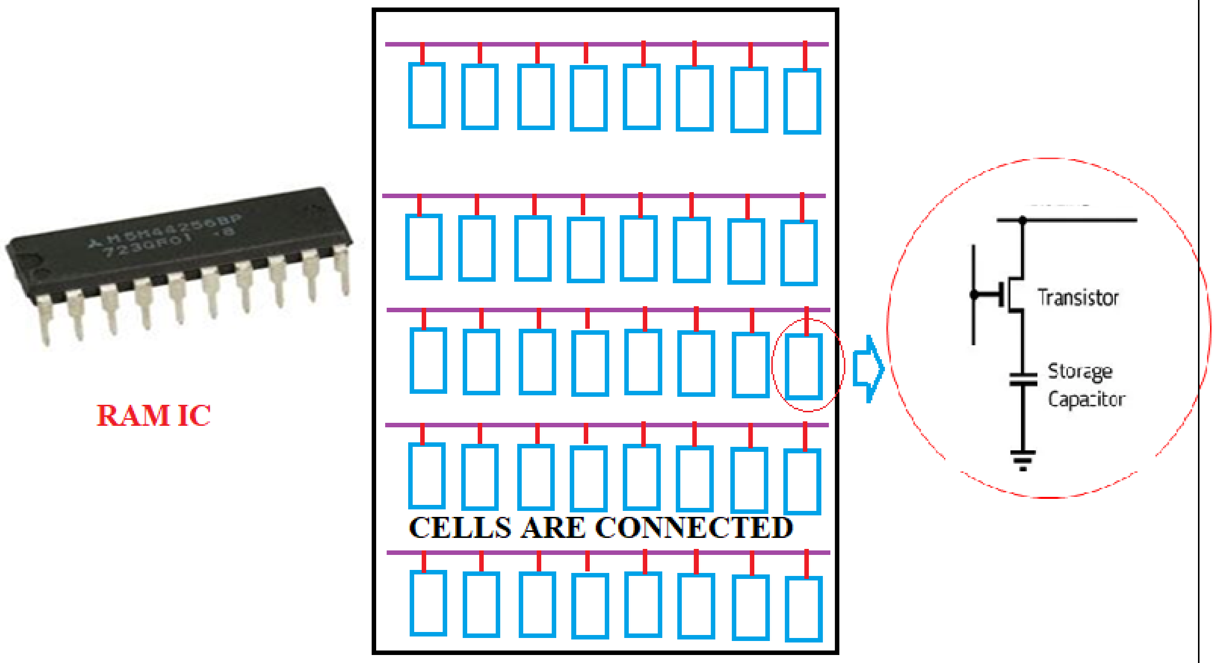
CPU will place address on address bus that is applied at the inputs of address decoder. Only one output of the decoder is selected based on address placed at inputs and cells connected to the output line are selected. For example, when address 0000000010 are placed on address bus, 3rd output of decoder is activated and cells connected to that line are selected.



What is byte-addressable memory?

If a memory is designed to store 8-bits in each addressable location, called byte addressable memory.

There will be 8-cells in each row in memory organization, and these 8-cells can hold 8-bits.



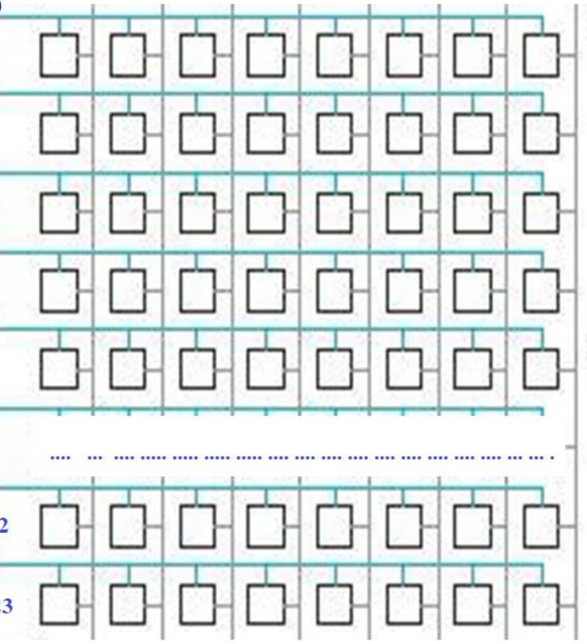
Example of byte-addressable memory: 1KB, 1MB, 1GB, 4MB etc. The first part of memory capacity indicate the number of rows and the 2nd part: B indicated that each row has 8-cells. Memory addresses are simply row numbers assigned sequentially, starting with ‘0’.

Following example shows addresses of 1KB RAM which is a byte-addressable memory. 1KB memory has 8192 cells arranged in 1024 rows and 8 columns.

1KB = 2^{10} (rows) x 8 (columns) (here 1K is $2^{10}$and B is 8)

= 1024(rows) x 8 (columns)

Address(Hex)	Address (Binary)	Address (decimal)
000H	0000000000	0
001H	0000000001	1
002H	0000000010	2
3FEH	1111111110	1022
3FFH	1111111111	1023



In a 1KB memory, there are 1024 rows, so there are 1024 addressable locations. Each row contains 8 cells so the content of each addressable location is 8-bits.

Memory addresses are row numbers, assigned sequentially; starting with 0 and ends with total number of rows-1, which is 1024-1 = 1023. For 1KB memory, starting address is 0 and address of highest addressable location is 1023. To represent addresses in binary, same number of digits (bits) are used for all locations for ease of design and implementation of address decoders to be used with memory devices to select memory locations electronically. For 1KB, the highest address in decimal is 1023 and to convert/represent it to binary 10 bits are required since 1023 in decimal is 1111111111 in binary. So to address all addressable locations of 1KB memory, at least 10 bits are required.

Alternatively, the minimum number of bits required to address all addressable location, the total number of addressable location can be represented as power of 2. Let n be the number of bits required to address all addressable locations of 1K locations.

$2^n = 1K = 2^{10}$

$n = 10$

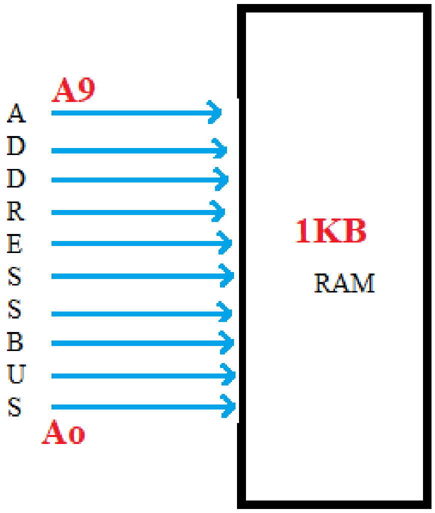
For ease of writing and avoiding errors, hexadecimal number system is used, in general, to write memory addresses in assembly language programming since it requires minimum number of digits. For 1KB, the highest address in binary is 1111111111 whereas it is 3FFH in hexadecimal number system. Here H stands for hexadecimal number system and 3FF is 1023 in decimal, 1111111111 in binary.

It is to be noted that

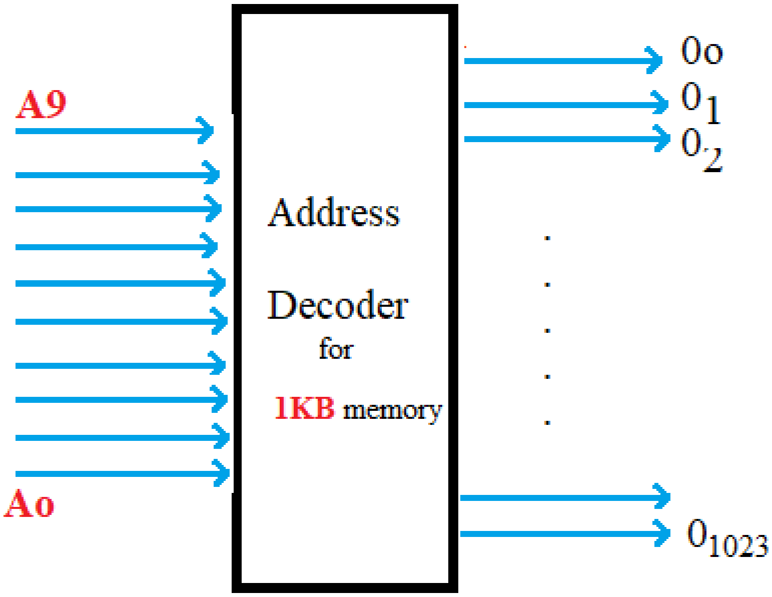
$1K = 2^{10}$	$1M = 2^{20}$	$1G = 2^{30}$
$2K = 2^{11}$	$2M = 2^{21}$	$2G = 2^{31}$
$4K = 2^{12}$	$4M = 2^{22}$	$4G= 2^{32}$
$8K = 2^{13}$	$8M = 2^{23}$	$8G = 2^{33}$
$16K = 2^{14}$	$16M = 2^{24}$	$16G = 2^{34}$

The number of binary bits/digits required to address all addressable locations also decides the number of address lines (pins), a memory IC should have.

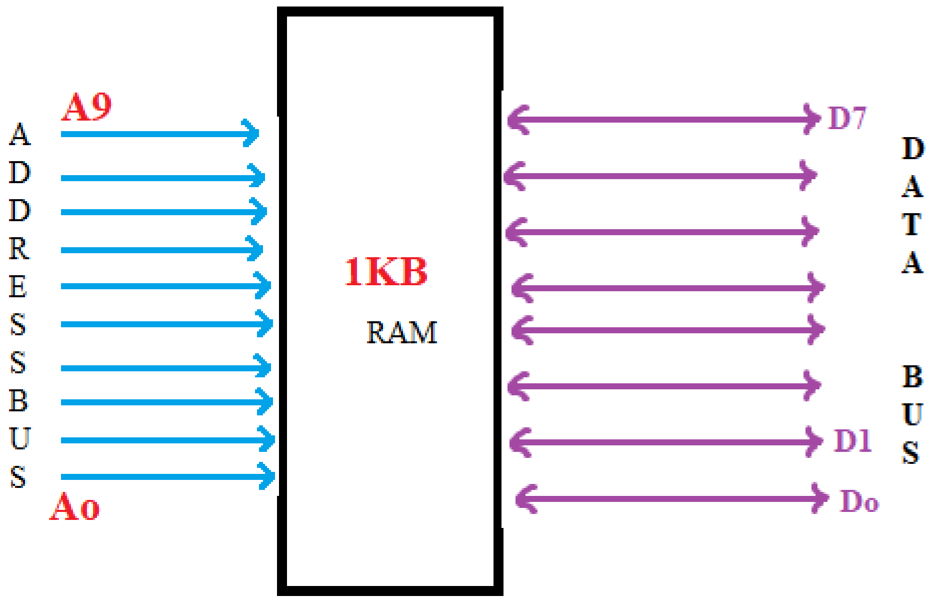
For example, in 1KB memory IC, the number of address lines is 10 since it requires 10 bits to address all addressable locations. These 10 address lines or address bus will receive address of a row/memory location from CPU in cases of READ and WRITE operations.



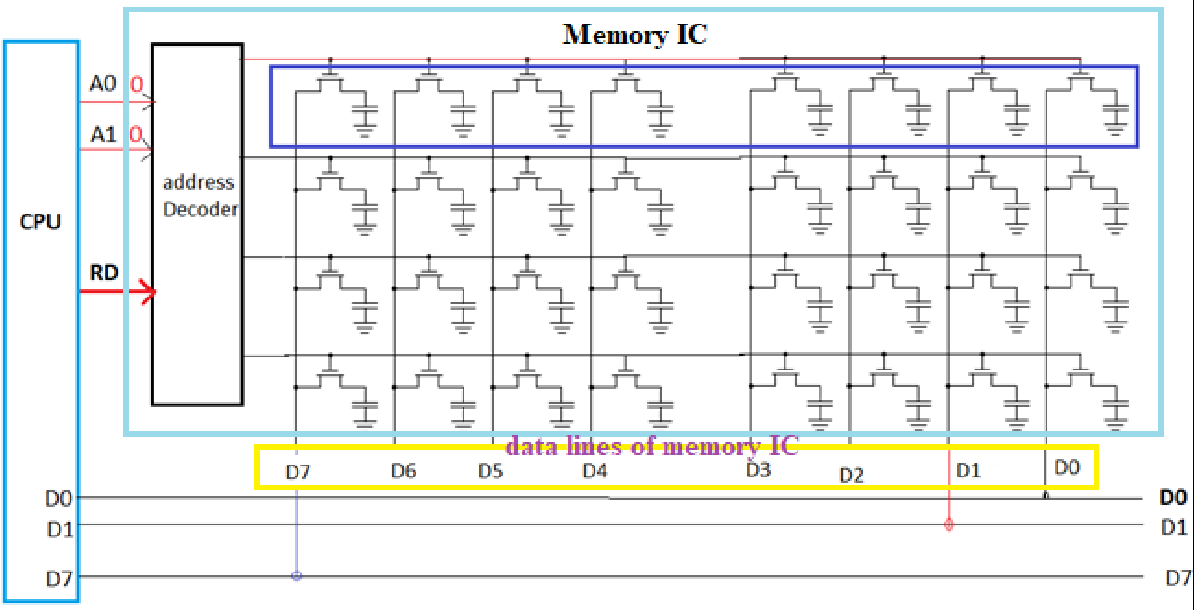
The number of lines in address bus, called size of address bus. The size of address bus, in fact, decides the number of input lines and output lines a decoder should be used with the memory IC. In case of 1KB memory, the address decoder should have 10 input lines and the number of output lines would be 1024 (2^{10}), exactly the number of rows or the number of addressable locations.



The number of cells in each row/location will decide the content of each location in binary bits. The number of cells in each row also decides the number of data lines (pins), a memory IC should have. For 1KB memory, the number of data lines to transfer contents to/from cells is 8 since each row contains 8 cells.



The number of data lines, called size of data bus. For 1KB memory IC, the size of data bus is 8. Inversely, given the size of data bus of a Memory IC, we can find the number of cells in each row as well as the content of each addressable location in terms of binary bits/digits. The data lines of memory IC are connected to data lines of CPU, as shown below.



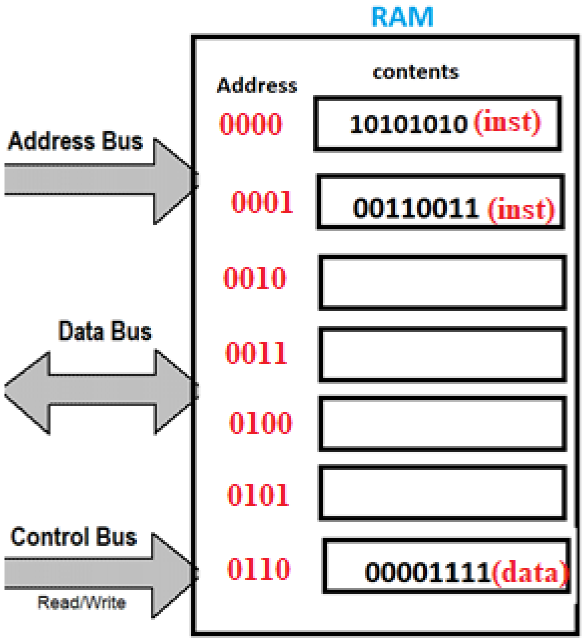
Memory: descriptive

What is primary memory?

RAM (Random Access Memory) is used as primary memory. It is a semiconductor device and it holds programs and data when programs are run. CPU reads instructions and data from RAM, decodes it at the control unit and finally processed at ALU. The results can be stored to RAM temporarily as well.

Briefly discuss Memory (RAM).

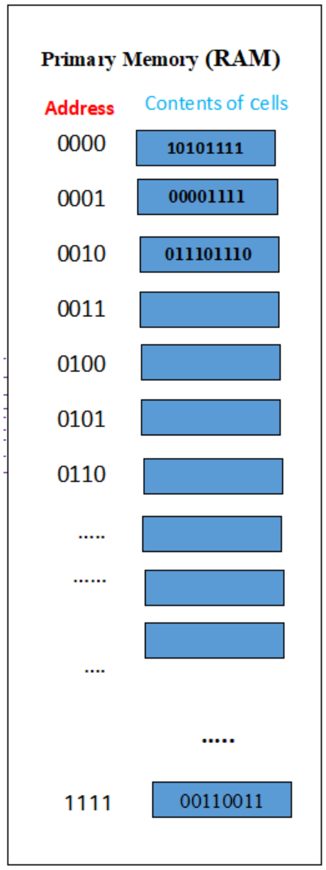
RAM stands for Random Access Memory. RAM is designed as an array of storage location as shown below.



Part of the computer where programs and data are stored in binary. Programs are “loaded” into RAM from a disk prior to execution by the CPU.

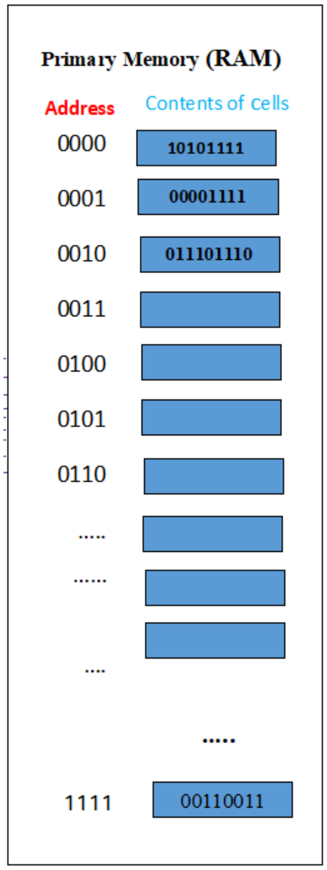
Memory consists of a number of sequential storage locations and each storage location is identified by unique code, called address. Each location is identified by unique code called address and contents of each location is usually 8-bits. The very first location is addressed by 0 and following locations are addressed by numbers sequentially. Both addresses and contents are usually represented in binary. But for ease of writing, large addresses and contents are usually represented in hexadecimal number system.

	Each memory location is capable of storing one data element or one computer instruction (Command for any operation). If content of each location is 8bits or 1Byte, called byte-addressable memory. The contents of RAM are lost if the power is turned off (volatile). The time to read or write is referred to as the **access time** and is constant for all addressable locations (random access). It means, time required to read or write to RAM, called Access time, is independent of physical location of storage which is contrary to sequential or semi-sequential storage devices like, disk, tape drive etc. That's why it is called Random Access memory. Any RAM IC has got a number of pins to receive address from CPU, called address bus, a number of pins to transfer data, called data bus and a few pins to receive/send commands from/to CPU called control bus. These pins or buses are used to interface RAM with CPU and other devices.
	What is RAM and why it is called so? RAM stands for Random Access Memory. RAM is an array of storage locations (shown below) and read/write time, called access time, is same for any location. It means that access time is independent of physical location of storage. The time to read or write is referred to as the **access time** and is constant for all addressable locations (random access). RAM is used to hold both program code (instructions) and data (numbers, strings etc). Programs are "loaded" into RAM from a disk prior to execution by the CPU. Locations in RAM are identified by an **addressing scheme** *e.g.* numbering the bytes in RAM from 0 onwards. The contents of RAM are lost if the power is turned off (volatile). RAM is designed as an array of storage location as shown below. Each location is identified by unique code called address and contents of each location is usually 8-bits. Both addresses and contents are usually represented in binary. But for ease of writing, large addresses and contents are usually represented in hexadecimal number system.



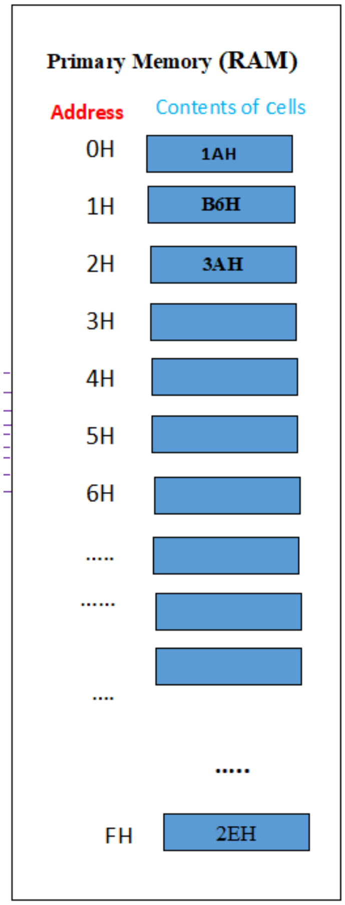
What do you understand by memory address? Explain

Semiconductor RAM is fabricated as an array of storage (shown below). Each storage location is uniquely identified by a code called address and each location can is designed to hold/store a number of bits. Memory addresses are numbers assigned sequentially starting from zero and the highest address depends on total addressable locations (exact value total addressable locations minus one) the memory device has. Number of bits required to form memory address depends on the number of binary bits required to form the highest address of the memory device. For uniformity and ease of hardware design, same number of bits used to address all addressable locations of a memory device. Diagrams below show the organization of a 16Bytes memory. Here the memory device has 16 addressable locations, each can hold 8bits. Addresses are assigned sequentially, starting with 0000. The highest address is 1111(15 in decimal). For ease of writing, both memory addresses and contents are represented in hexadecimal format.



Primary Memory (RAM)

Address	Contents of cells
0000	10101111
0001	00001111
0010	011101110
0011	
0100	
0101	
0110	
…..	
……	
….	
	…..
1111	00110011



Primary Memory (RAM)

Address	Contents of cells
0H	1AH
1H	B6H
2H	3AH
3H	
4H	
5H	
6H	
…..	
……	
….	
	…..
FH	2EH

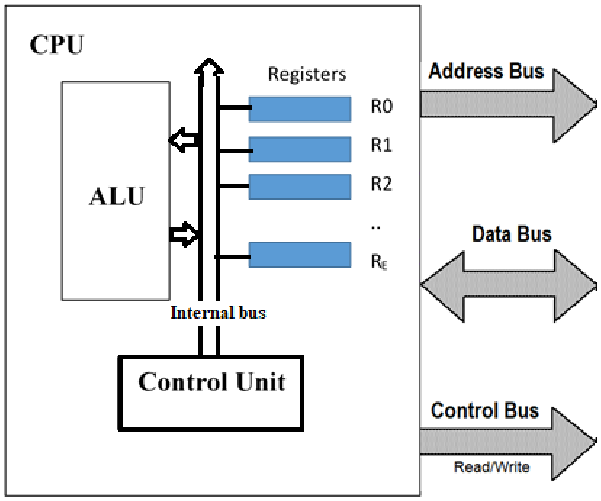
Show the interconnection between CPU and Main Memory.

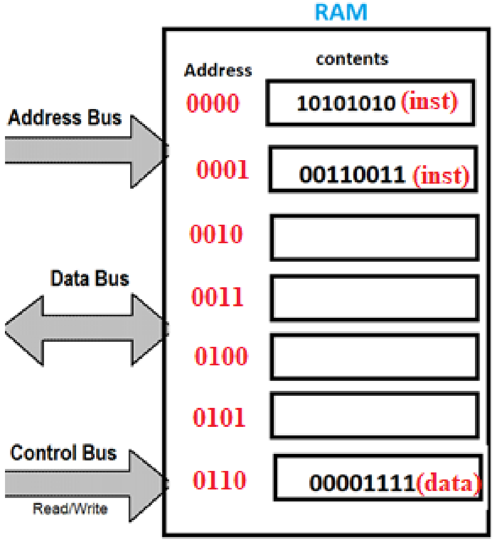
Both CPU and RAM has:

A number of pins to carry address information, called address bus.

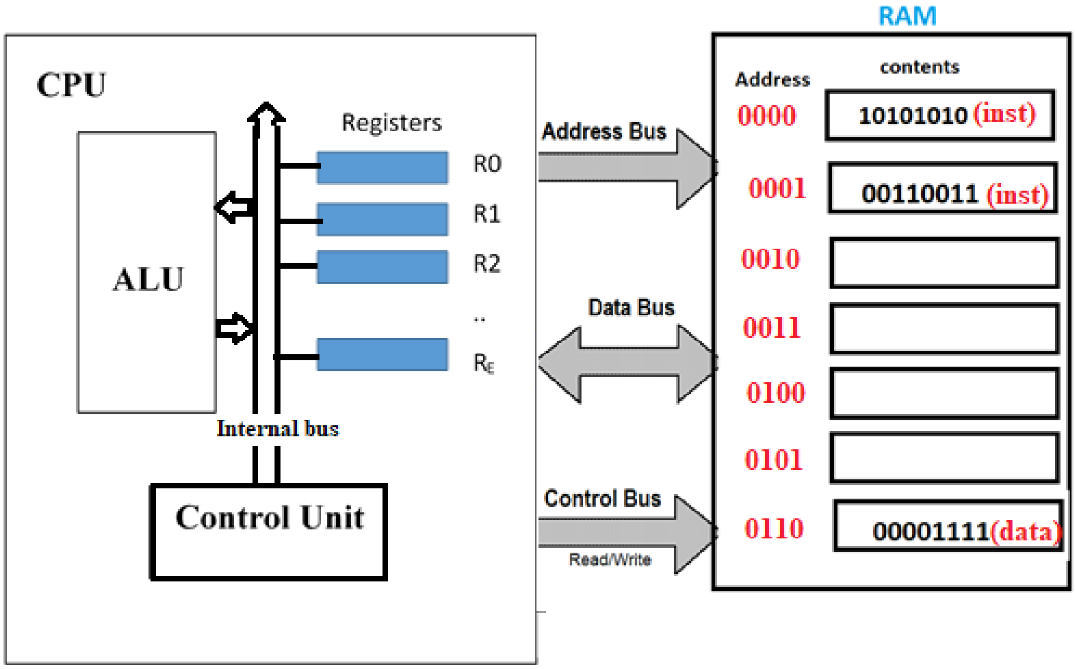
A number of pins to carry data, called data bus.

A number of pins to carry command signals, called control bus





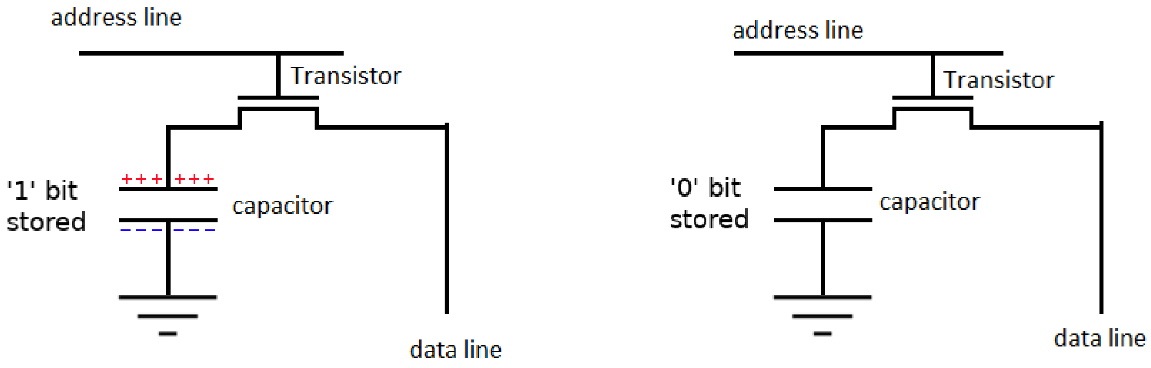
To interconnect/interface RAM to CPU address, data and control pins of RAM are connected to similar pins of CPU through conducting wires. In fact, conducting path are provided on motherboard and sockets/connectors are also designed on motherboard to hold CPU IC and RAM ICs or modules.



--

How 1’s and o’s are stored in primary memory?

Each addressable location of primary memory (RAM) is designed to have a number of cells and each cell can hold only one bit (either 0 or 1). Each cell is designed to have one tiny capacitor. By charging the capacitor, logical 1 is saved whereas by discharging a capacitor, logical 0 is saved in each cell. In order to select a cell, a transistor is also used with it as shown below.



What is a cell? How a cell is designed? What electronic components are used to design a cell?

What do you understand by Byte addressable memory? Give examples

If content of each addressable location of a memory device is 8 bits/one byte, called byte addressable memory.

Example: 1KB, 1MB, 1GB etc.

How many minimum bits are required to address all addressable locations of following memory devices?

1KB: 10 bits (1K = 2^{10}, or 1K = 1024, so the addresses would be 0, 1, 2 ……..1023 and binary 10 bits are required to represent decimal 1023 (1111111111))

1Kx16bits = 10 bits

1MB = 20 bits (1M = 2^{20})

16MB = 24bits (16M = 2^{2}x2^{20} = 2^{24})

1GB = 30 bits (1G = 2^{30})

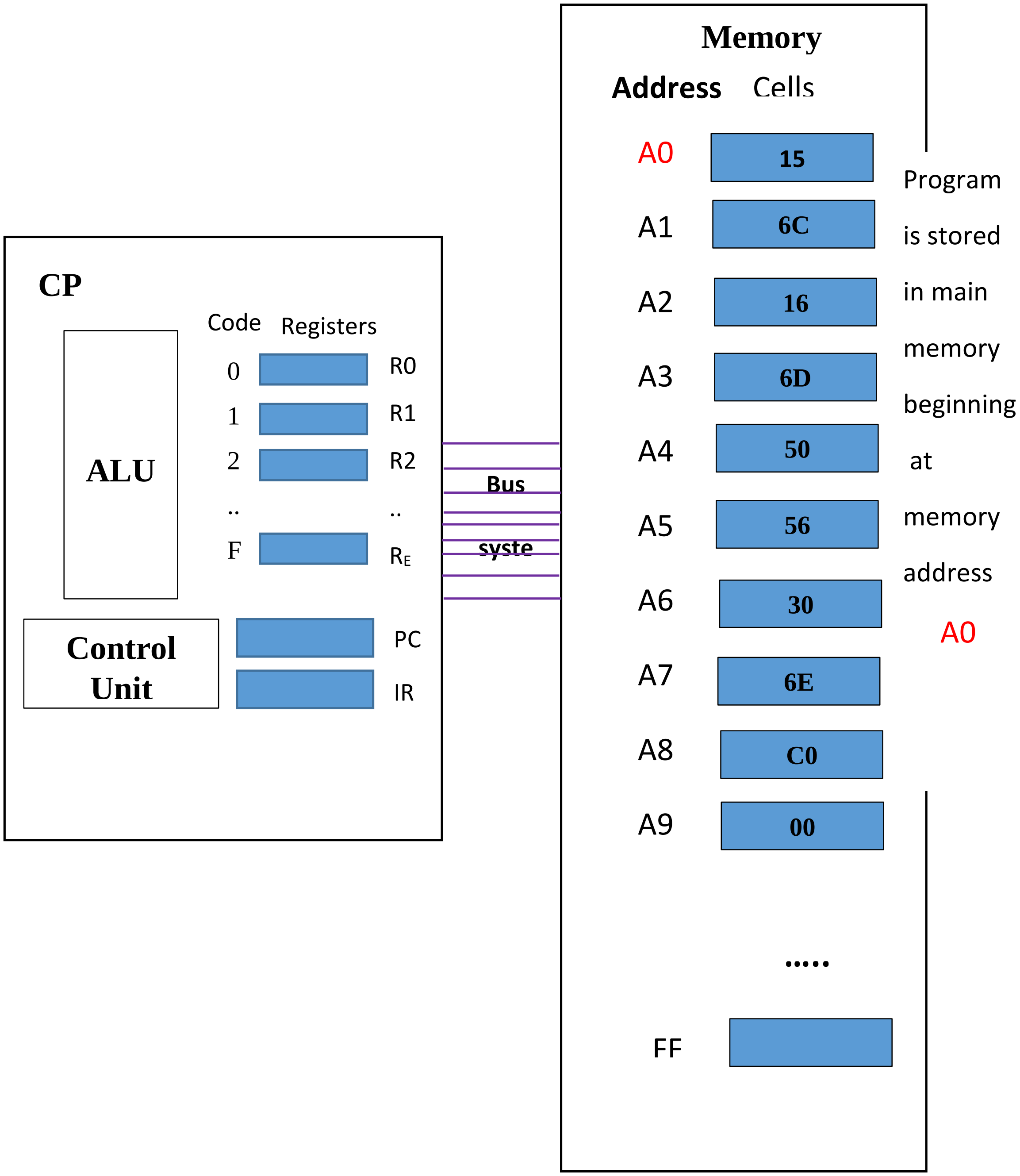
4GB = 32 bits (4G = 2^{2} x 2^{30} = 2^{32})

Can we use more bits than calculated above?

Yes, some address spaces will remain unused.

	Can we use less bits than calculated above? No, a large portion of memory cannot be addressed and used. For example, in case of 1KB memory, if we use 9-bit address instead of 10 bits, we can only address half (512 out of 1024) of total addressable locations.
	What do you understand by Byte addressable memory? Give examples If content of each addressable location of a memory device is 8 bits/one byte, called byte addressable memory. Example: 1KB, 1MB, 1GB etc.
	How many minimum bits are required to address all addressable locations of following memory devices? 1KB: 10 bits (1K = 2^{10}, or 1K = 1024, so the addresses would be 0, 1, 2 ……..1023 and binary 10 bits are required to represent decimal 1023 (1111111111)) 1Kx16bits = 10 bits 1MB = 20 bits (1M = 2^{20}) 16MB = 24bits (16M = 2^{2}x2^{20} = 2^{24}) 1GB = 30 bits (1G = 2^{30}) 4GB = 32 bits (4G = 2^{2} x 2^{30} = 2^{32})
	Can we use more bits than calculated above? Yes, some address spaces will remain unused.
	Can we use less bits than calculated above? No, a large portion of memory cannot be addressed and used. For example, in case of 1KB memory, if we use 9-bit address instead of 10 bits, we can only address half (512 out of 1024) of total addressable locations.
	How a program is loaded into memory? When CPU runs any program, the machine codes of instructions of that program are loaded into RAM in consequtive locations. The starting address is set by operating system or by the user, if program written in low level language(assembly language). The number

	of memory locations used to hold machine codes depend on the length(size in bits) of instruction and size (no of bits) of each memory location. For example, if each instruction of a program is one byte and RAM is byte addressable, then each machine code can be saved in a single addressable location. Assume that the mache codes of inst-1, Inst-2, Inst-3…. are 15H, 6Ch and 16H respectively. The diagram below shows how instructions are loaded into consecutive locations of RAM starting from A0H.



Another example, if each instruction of a program is two bytes and RAM is byte addressable, then each machine code requires two addressable locations of RAM. Assume that the mache codes of inst-1, Inst-2, Inst-3…. are 156CH, 166DH and 5056HH respectively. The diagram below shows how instructions are loaded into consecutive locations of RAM starting from A0H.

	For a 1KB DRAM IC: Calculate the number of cells: 1024x8 How the cells are arranged: row and column; no of rows=1024 and no of columns=8 Calculate the number of total addressable locations: 1K=2^{10} = 1024 What is the minimum number of bits required to form address to access all locations: 1K=2^{10} = 10 bits Write the first address in binary as well as in hexadecimal number system: 0000000000(binary) , 000(in hex) Write the address of highest addressable location in binary as well as in hexadecimal number system 1111111111(in binary), 3FF(hexadecimal) How many address pins this memory IC should have? 10 How many data pins this memory IC should have? 8 What are the control pins this memory IC should have? READ, WRITE, ENABLE
	What is memory hierarchy? What is the top at the hierarchy?
	For a 1KB DRAM IC: Calculate the number of cells How the cells are arranged Calculate the number of total addressable locations What is the minimum number of bits required to form address to access all locations Write the first address in binary as well as in hexadecimal number system Write the address of highest addressable location in binary as well as in hexadecimal number system How many address pins this memory IC should have? How many data pins this memory IC should have? What are the control pins this memory IC should have?

Short questions and MCQ

	What is RAM? What does it hold? Whys it is named so? Random access memory, used as main memory to hold program and data before CPU runs a program.
	What is ROM? What does it hold? Whys it is named so?
	How to measure capacity of a RAM?

	MB(Mega Bytes), GB(Giga Bytes) etc
	What is Cache memory? A fast memory located on CPU and on motherboard to hold frequently/recently used data and instructions.
	How programs and data are saved in memory? In binary format.
	What does memory hold? What are the contents of memory?
	How many bits are used to represent data?
	How many bits are used to represent Instructions?
	How memory devices are designed?
	How memory capacity is defined? Give examples
	Show the architecture of memory?
	How information is stored in memory?
	What do you understand by a byte addressable memory?
	Show the architecture of 1KB memory?
	Identify byte addressable memory: 1KB, 1K x 16bits, 1k x 32 bits.
	How a 16 bit data is stored in a byte addressable memory. Give example. Both types: little endian and big endian with examples
	How a 32 bit data is stored in a byte addressable memory. Give example
	How a 64 bit data is stored in a byte addressable memory. Give example.
	What is little endian system? Give example
	What is big endian system? Give example
	What do you understand by memory address?
	How many bits are used to represent memory address?
	How many bits will require to address all addressable locations of a 1KB memory?
	How many bits will require to address all addressable locations of a 4KB memory?
	How many bits will require to address all addressable locations of a 1MB memory?
	How many bits will require to address all addressable locations of a 1GB memory?
	Show the address of first and highest addressable location of a 1KB memory in binary and hexadecimal format.
	Show the address of first and highest addressable location of a 4KB memory in binary and hexadecimal format.
	Show the address of first and highest addressable location of a 16 KB memory in binary and hexadecimal format.
	Show the address of first and highest addressable location of a 1MB memory in binary and hexadecimal format.
	Show the address of first and highest addressable location of a 1GB memory in binary and hexadecimal format.
	Show the address of first and highest addressable location of a 4GB memory in binary and hexadecimal format.
	What does memory hold? What are the contents of memory?
	How many bits are used to represent data?
	How many bits are stored in a cell?
	Calculate the number of cells in 1KB memory.
	Calculate the number of cells in 1K x 4bits memory.

	What is address decoder?
	What is the size of address decoder of 1KB memory?
	What is the size of address decoder of 4KB memory?
	What is the size of address decoder of 1MB memory?
	What is the size of address decoder of 8MB memory?
	What do you understand by byte addressable memory? If the content of each addressable location of a memory device is one byte (8-bits), called byte addressable memory. Example: 1KB, 1MB, 16MB memory
	How a 16 bit data is saved in a byte-addressable memory? Requires two addressable locations and saved in consecutive locations.
	How a 32 bit data is saved in a byte-addressable memory? Requires four addressable locations and saved in consecutive locations.
	For the following memory ICs, find the number of addressable locations and word size (contents of each location: 1KB, 4K x 16 bits, 4MB, 4M x 32 bits
	What is memory address?
	What do you understand by byte addressable memory?

Sample Questions:

1.	What does memory hold? What are the contents of memory?
2.	How many bits are used to represent data?
3.	How many bits are used to represent Instructions?
4.	How memory devices are designed?
5.	How memory capacity is defined? Give examples
6.	Show the architecture of memory?
7.	How information is stored in memory?
8.	What do you understand by a byte addressable memory?
9.	Show the architecture of 1KB memory?
10.	Identify byte addressable memory: 1KB, 1K x 16bits, 1k x 32 bits.
11.	How a 16 bit data is stored in a byte addressable memory. Give example. Both types: little endian and big endian with examples
12.	How a 32 bit data is stored in a byte addressable memory. Give example
13.	How a 64 bit data is stored in a byte addressable memory. Give example.
14.	What is little endian system? Give example

15.	What is big endian system? Give example
16.	What do you understand by memory address?
17.	How many bits are used to represent memory address?
18.	How many bits will require to address all addressable locations of a 1KB memory?
19.	How many bits will require to address all addressable locations of a 4KB memory?
20.	How many bits will require to address all addressable locations of a 1MB memory?
21.	How many bits will require to address all addressable locations of a 1GB memory?
22.	Show the address of first and highest addressable location of a 1KB memory in binary and hexadecimal format.
23.	Show the address of first and highest addressable location of a 4KB memory in binary and hexadecimal format.
24.	Show the address of first and highest addressable location of a 16 KB memory in binary and hexadecimal format.
25.	Show the address of first and highest addressable location of a 1MB memory in binary and hexadecimal format.
26.	Show the address of first and highest addressable location of a 1GB memory in binary and hexadecimal format.
27.	Show the address of first and highest addressable location of a 4GB memory in binary and hexadecimal format.
28.	
29.	What is a cell? How a cell is designed? What electronic components are used to design a cell?
30.	How many bits are stored in a cell?
31.	Calculate the number of cells in 1KB memory.
32.	Calculate the number of cells in 1K x 4bits memory.
33.	
34.	What is address decoder?
35.	What is the size of address decoder of 1KB memory?
36.	What is the size of address decoder of 4KB memory?
37.	What is the size of address decoder of 1MB memory?
38.	What is the size of address decoder of 8MB memory?
39.	Show, how address decoder is connected to cells.

questions

1.	What is RAM? Why it is called so?
2.	How a high level program is stored in RAM?
3.	What is a cell? Show the organization of a cell.
4.	What electrical/electronic components are used to design a cell? Show the diagram.
5.	What is bit line of a cell? What is the function of bit line?
6.	What is address line of a cell? What is the function of address line?
7.	How 1 and 0 are stored in cells?
8.	For a 2-dimensional cell array, how the address lines are connected.
9.	For a 2-dimensional cell array, how the bit lines are connected.
10.	What do you understand by a byte-addressable memory?
11.	What do you understand by little endian and big endian system?

12.	How a 32 bit data is saved in a byte-addressable memory? Give example
13.	How the following data (hexadecimal) will be saved in a byte-addressable memory in little endian system: 2AH, 2A45H, 7E2A459CH
14.	How the following data (hexadecimal) will be saved in a byte-addressable memory in big endian system: 2AH, 2A45H, 7E2A459CH
15.	Sow the internal architecture of a RAM.
16.	What is memory address? How it is formed, give example.
17.	What is address decoder?
18.	How an address select a memory location?
19.	How does CPU read from memory? List the steps.
20.	How does CPU write to from memory? List the steps.
21.	What do you understand by capacity of RAM, give example.
22.	What are important pins and signals found in a RAM IC?
23.	Show the interfacing of a RAM IC to a processor.
24.	For the following memory ICs, find the number of addressable locations and word size (contents of each location: 1KB, 4K x 16 bits, 4MB, 4M x 32 bits
25.	For the following memory modules, find the number of addressable locations and word size (contents of each location: 1GB, 4G x 16 bits, 4GB, 4G x 32 bits
26.	For the following memory ICs, find the size of address bus and data bus: 1KB, 4K x 16 bits, 4MB, 4M x 32 bits
27.	For the following memory ICs, find the size of address bus and data bus: 1GB, 4G x 16 bits, 4GB, 4G x 32 bits
28.	For the following memory ICs, show the address of first location and highest addressable locations in decimal, binary and hexadecimal number systems: 1KB, 4K x 16 bits, 4MB, 4M x 32 bits
29.	For the following memory ICs, show the address of first location and highest addressable locations in decimal, binary and hexadecimal number systems: 1GB, 4G x 16 bits, 4GB, 4G x 32 bits
30.	Design memory modules of following capacity using 4KB ICs and other components: 32KB, 32K x 32bits, 64K x 64bits,
31.	What do you understand by Byte ordering or endianness? Byte ordering or endianness, is another major architectural consideration. If we have a two-byte integer, the integer may be stored so that the least significant byte is followed by the most significant byte or vice versa. In little endian machines, the least significant byte is followed by the most significant byte. Big endian machines store the most significant byte first (at the lower address).
32.	How a 16-bit data/instruction is saved in a byte addressable memory? Two consecutive memory locations are used to save 16-bit data/instruction is saved in a byte addressable memory. However, there are two methods: a) Little endian and b) big endian Example: show how 10101010 00110011 (AA33H) will be saved a) Little endian (lower byte in lower address, higher byte in higher address)

<table>
<tr><td></td><td>
<table>
<tr><th>Memory address</th><th>contents</th></tr>
<tr><td></td><td></td></tr>
<tr><td>405H (randomly chosen)</td><td>00110011 (33H)</td></tr>
<tr><td>406H</td><td>10101010 (AAH)</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>
b) Big endian (higher byte in lower address, lower byte in higher address)
<table>
<tr><th>Memory address</th><th>contents</th></tr>
<tr><td></td><td></td></tr>
<tr><td>405H (randomly chosen)</td><td>10101010 (AAH)</td></tr>
<tr><td>406H</td><td>00110011 (33H)</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>33.</td><td>How the following 32 bit data will be save in a byte addressable memory at 10000H? 5A2B3C4DH. Show for little endian and big endian.</td></tr>
<tr><td>34.</td><td>For an 8M x 4 bit RAM, what is minimum number of address bits required to address all addressable locations? What is the size of data bus?</td></tr>
<tr><td>35.</td><td>Which is byte addressable RAM?
a) 8K x 16 bit
b) 16K x 8bit
c) 4K x 2bit
d) 2K x 4bit
e) none</td></tr>
<tr><td>36.</td><td>If the size of address bus of a CPU is 22 bits and size of data bus is 16bits, calculate the maximum capacity of RAM to be used with the CPU?
Solution:
Maximum capacity of RAM = 2^{22} x 16 bits = 2^{2} x 2^{20} x 16 bits = 4M x 16 bits</td></tr>
</table>

<table>
<tr><td></td><td>A 16-bit microprocessor uses byte addressable memory and can address 16MB RAM. Show, how the CPU will save the following 32 bit data: 12 AB 23 CD H to highest addressable location of RAM in Little endian as well as in Big Endian systems.</td></tr>
<tr><td></td><td>For a 16M x 16 bits DRAM IC:
i. Calculate the number of cells
……………………………………………………………………………………………………….
ii. How the cells are arranged…………………………………………………………………………………………………..
iii. Calculate the number of total addressable locations
…………………………………………………………………………….
iv. What is the minimum number of bits required to form address to access all locations
…………………………….</td></tr>
</table>

v. Write the first address in binary as well as in hexadecimal number

system………………………………………………

……

……………………….

vi. Write the address of highest addressable location in binary as well as in hexadecimal

number system ……

……

……………………..

vii. How many address pins this memory IC should have?
…………………………………………………………………………….

viii. How many data pins this memory IC should have?
…………………………………………………………………………………….

ix. What are the control pins this memory IC should have?
……………………………………………………………………………

x. What would be the size of address decoder?
……………………………………………………………………………………………..

A 16-bit microprocessor uses byte addressable memory and can address 16KB RAM. Show, how the CPU will save the following 32 bit data: 2312CDAB H to highest addressable location of RAM in Little endian as well as in Big Endian systems.

For a 32M x 32 bits DRAM IC:

i. Calculate the number of cells
…….

ii. How the cells are
arranged………..

iii. Calculate the number of total addressable locations
…………………………………………………………………………………..

iv. What is the minimum number of bits required to form address to access all locations
…………………………….

v. Write the first address in binary as well as in hexadecimal number
system………………………………………………

vi. Write the address of highest addressable location in binary as well as in hexadecimal
number system ……

vii. How many address pins this memory IC should have?
…………………………………………………………………………….

viii. How many data pins this memory IC should have?
…………………………………………………………………………………….

	ix. What are the control pins this memory IC should have? ……………………………………………………………………………… x. What would be the size of address decoder? ………………………………………………………………………………………………..