UNIT – V

MEMORY ORGANIZATION

Q.1. Describe memory hierarchy in a computer system with the help of a diagram.

[UP Tech 2005-2006]

Ans. The memory hierarchy system consists of all storage units placed in a computer system from slow auxiliary memory to fast cache memory. A five level memory hierarchy is shown in figure given below:

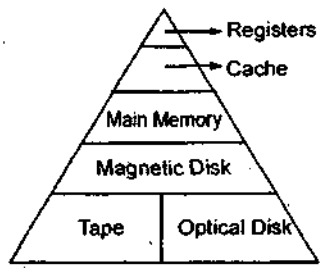


Fig. *Five-level Memory Hierarchy.*

At the top CPU registers are present which are smallest in capacity but fastest in access. Next is cache memory which is normally of the order of 32KB to few MBs. Cache memory is very small, expensive and it has very high access speed. Next is main memory whose size normally in Giga bytes. Next are magnetic disk and finally magnetic tape and optical tape. Magnetic disks and tapes are also called auxiliary memory. They are used for storing large data files and other backup information. The 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 required.

Hierarchy depends upon three parameters.

(i) Access time

(ii) Storage capacity

(iii) Cost

The average time required to reach a storage location in memory and obtain its contents is called access time.

As the storage capacity of the memory increases, the cost per bit for storing binary information decreases. But as storage capacity increases, access time also increases.

Q.2. Discuss the various organization of RAM. A computer uses RAM chip of 1024 * 1 capacities. How many chips are needed and how should their address lines be connected to provide a memory capacity of 1024 bytes?

[UP Tech 2005-2006]

Ans. RAM is of two types: Static random access memory (SRAM) and Dynamic random access memory (DRAM).

Static random access memory (SRAM): memory which consist of circuits capable of retaining their state as long as power is applied are known as Static random access memory. SRAM is made up of flip flops. Data remains in SRAM as long as power remains. As data remains in flip flop till power remains hence no refreshing circuit is required in SRAM. Heat dissipation is less in SRAM. Packing density is high (number of flip-flops placed on a single chip) in SRAM. It is expensive as compared to DRAM.

Dynamic random access memory (DRAM): Memories in which cells do not retain their state indefinitely are known as Dynamic random access memory. Information is stored in a dynamic memory cell in the form of a charge on a capacitor. DRAM is made up of capacitors. DRAM tends to loose data because capacitors discharge with time. Hence refreshing circuits are needed in DRAM to restore the decaying charge. Heat dissipation is more in DRAM because of capacitors. Packing density

is less in DRAM because DRAM is made up of capacitors and capacitors continuously dissipate heat. Hence they cannot be placed close together closely.

Numerical

Total memory capacity required
= 1024 bytes = (1024 * 8) bits
Size of each RAM chip = 1024 * 1 bits
Hence number of RAM chips required
= (1024 * 8)/ (1024 * 1)
= 8

Total 8 RAM chips of size 1024 * 1 are required to provide a memory capacity of 1024 bytes. Address lines are to be connected in parallel.

Q3. Write the differences between Static RAM and Dynamic RAM.
[UP Tech 2004-2005]

Ans. The main memory is the central storage unit in a computer system. It stores programs and data during computer operation. Ram is of two types: SRAM (static random access memory) and DRAM (Dynamic random access memory). There are many differences between SRAM and DRAM which are as follows:

i. SRAM is made up of flip flops where as DRAM is made up of capacitors.
ii. Data remains in SRAM as long as power remains where as DRAM tends to loose data because capacitors discharge with time.
iii. No refreshing circuit is required in SRAM where as refreshing circuits are needed in DRAM to restore the decaying charge.
iv. Heat dissipation is less in SRAM where as heat dissipation is more in DRAM because of capacitors.
v. Packing density is high (number of flip-flops placed on a single chip) in SRAM where as packing density is less in DRAM because DRAM is made up of capacitors and capacitors continuously dissipate heat. Hence they cannot be placed close together closely.
vi. SRAM is expensive as compared to DRAM.

Q.4. How many 128 bytes RAM chips are required to provide a memory of 2048 bytes?
[UP Tech 2004-2005]

Ans. Size of one RAM chip = 128 bytes
Size of memory = 2048 bytes
Total number of RAM chips required = 2048/128 = 16

Hence 16 RAM chips of size 128 bytes are required to form a memory of size 2048 bytes.

Q.5. A computer employees RAM chip of 256*8 and ROM chips of 1024*8. The computer system needs 2K bytes of RAM, 4K bytes of ROM and four interface units, each with four registers. A memory mapped 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: [UP Tech 2007-2008]

(i) How many ROM and RAM chips are needed?
(ii) Draw a memory address map for the system.
(iii) Give the address range in hexadecimal foe RAM, ROM and interface.

Ans. (i) Size of RAM chip = 256 * 8
Memory size required = 2K bytes = 2 * 1024 bytes = 2 * 1024 * 8
Total number of RAM chips required = (2 * 1024 * 8)/ (256 * 8) = 8 chips
Size of ROM chip = 1024 * 8
Memory size required = 4K bytes = 4 * 1024 bytes = 4 * 1024 * 8
Total number of ROM chips required = (4 * 1024 * 8)/ (1024 * 8) = 4 chips
Hence 8 RAM chips and 4 ROM chips are required.

(ii) and (iii) RAM size = 256 bytes = 2^8
8 bits for RAM address
ROM size = 1024 bytes = 2^{10}
10 bits for ROM address

Memory address map will be as shown below

Component	Address	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
RAM	0000-07FF	0	0	0	0	0	3*3 DEC			X	X	X	X	X	X	X	X
ROM	4000-4FFF	0	1	0	0	2*4DEC		X	X	X	X	X	X	X	X	X	X
Interface	8000-800F	1	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X

There are 8 RAM chips so 3*8 decoder is needed to select 1 RAM chip.

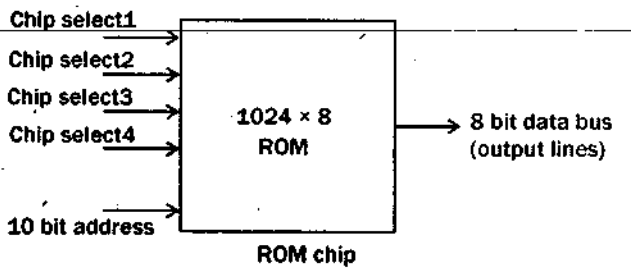
There are 4 ROM chips so 2*4 decoder is needed to select 1 ROM chip.

16 address lines will be in use as shown above. Address are represented in hexadecimal

Q6. A ROM chip of 1024 * 8 bits has four select inputs and operated from a 5-volt power supply. How many pins are needed for the IC package? Draw a block diagram and label all input and output terminals in the ROM. *[UP Tech 2008-2009]*

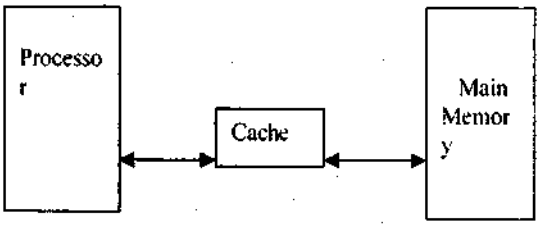
Ans. Total 24 pins are needed for the IC package. These pins are divided as follows:

10 pins for input
4 pins for chip select
8 pins for output
2 pins for power



The block diagram of ROM having all the input and output terminal is as follows:

Q7. Write a short note on cache memory.



Ans. The speed of main memory is low as compared to the speed of modern processors. If processor spends much of its time waiting to access instructions and data in main memory then it affects processor's performance. If active portion of program and data are placed in a fast small memory then average access time can be reduced thus reducing the total execution time of the program. This small memory is called cache memory. It is placed in between CPU and main memory. It is considered as fastest memory. Its size is just fraction of main memory.

Basic operation: when CPU needs to access memory, the cache is examined. It the required word is found in cache, it is read from there otherwise main memory is accessed. Then a block of words including the main request is transferred from main memory to cache. Now cache contains the requested word as well as the future references also.

Performance of cache memory is measured in terms of a quantity called Hit ratio. When CPU refers to memory and finds the word in cache, it is called a *hit*. When it does not find the word then it is a *miss*.

Hit ratio: Ratio of number of hits divided by the total CPU references to memory is called Hit ratio.

In cache memory write operation can be performed in two ways.

- Write through policy
- Write back policy

In *write through policy* the cache location and the main memory location are updated

simultaneously. Means when ever there are any changes in the cache memory those changes are also reflected in the main memory. This policy is simple but it results in unnecessary write operations in the main memory when a given cache word is updated several times.

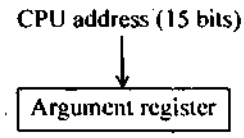
In *write back policy* only cache memory location is updated and the changed location is marked with an associated flag bit commonly known as dirty bit or modified bit. The main memory location is updated later when the marked word is to be removed from cache.

Q8. What are the different procedures for mapping cache memory?

[Up Tech 2005-2006]

Ans. The transformation of data from main memory to cache memory is known as mapping process. There are three mapping procedures.

- Associative mapping
- Direct mapping
- Set associative mapping



Address	data
01020	6754
02555	7564

Associative mapping: In associative memory both address and data are stored. The diagram shows 2 words presently stored in cache. All the numbers are represented in octal.

CPU places 15 bit address in the argument register and associative memory is searched for a matching address. If that address is found, then corresponding 12 bit data is read from it and sent to the CPU. If no match is found then main memory is accessed for the word.

Direct mapping: In direct mapping RAM is used for cache memory as described below.

CPU address is of 15 bits; hence size of main memory is 32K. This address is divided into 2 fields; index and tag.

2^n words in main memory and 2^k words in cache memory. Then k bits are there in index field and (n-k) bits are there in tag field. For example 2^{15} words in main memory and 2^9 words in cache memory so size of index is 9 bits and tag is (15-9)=6 bits.

Addressing relationship between main and cache memory is shown with the help of figure given below.

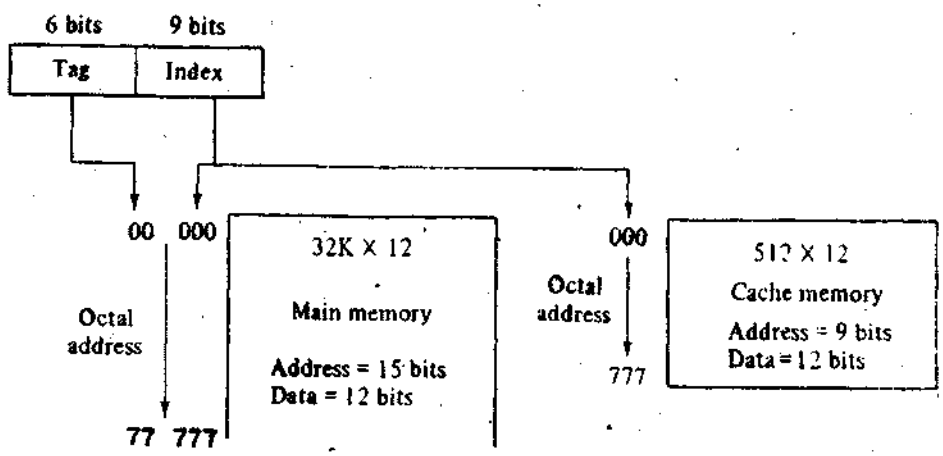


Fig. *Addressing relationships between main and cache memories.*

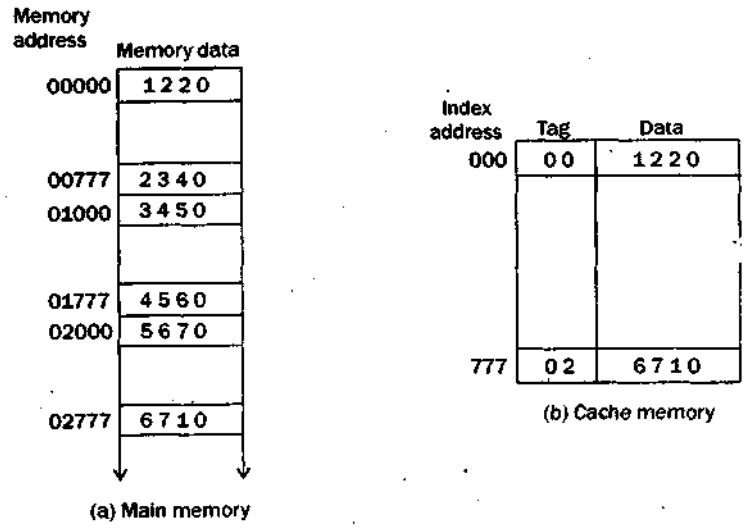


Fig. *Direct mapping chache organization.*

Each word in cache consists of tag field and data. When CPU generates a memory request, the index field is used for the address to access the cache. After that tag field is compared with the tag in the word read from cache. If it matches then it is a hit otherwise miss. If miss then required word is read from main memory and stored in cache, with a new tag replacing previous value.

Set associative mapping: It is an improvement over direct mapping. In this mapping 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. Total number of tag-data items in one word of cache is said to form a set.

Index	Tag	Data	Tag	Data
000	01	3450	02	5670
777	02	6710	00	2340

Fig. *Two-way set-associative mapping cache.*

Each tag requires 6 bits and each data word has 12 bits. If there are two tag fields present in cache then word length of cache will be 2 * (6+12) = 36 bits.

The hit ratio will improve as the set size increases because more words with the same index but different tags can be present in cache.

Q.9. Differentiate Direct Mapping and Associative mapping procedures for organization of cache memory with example. Give merits and demerits of both mapping procedures. **[UP Tech 2004-2005]**

Ans. Differences between associative mapping and direct mapping are as follows:

- Associative mapping technique is expensive than direct mapping because of added logic associated with each cell.
- Hit ratio is low in associative as compared to direct mapping.
- Associative mapping technique is faster in access than direct mapping.

Advantages of associative mapping

- Required data word can be accessed in a fast way as compared to other techniques.

Disadvantages of associative mapping

- Hit ratio is low. Means chances of getting required word in memory are less as compared to other techniques.

- It is expensive because this mapping requires added logic associated with each cell.

Advantages of direct mapping:

- Hit ratio is high. Means chances of getting required word in memory are more as compared to other techniques.
- It is cheaper as compared to other techniques.

Disadvantages of direct mapping:

- Required data word searching is slow as compared to other techniques.

Q10. A block set associative memory consists of 128 blocks divided into four block sets. The main memory consists of 16384 blocks and each block contains 256 eight bit words.

(i) How many bits are required for addressing the main memory?

(ii) How many bits are needed for addressing the cache memory? [UP Tech 2004-2005]

Ans. (i) Size of main memory = 16384 blocks = 2^{14}

Size of one block = 256 words = 2^8

Total number of words that memory contains

$$= 2^{14} * 2^8 = 2^{22}$$

Hence 22 bits are required for addressing main memory.

(ii) Size of cache memory = 128 blocks = 2^7

Size of one block = 256 words = 2^8

Total number of words that cache contains

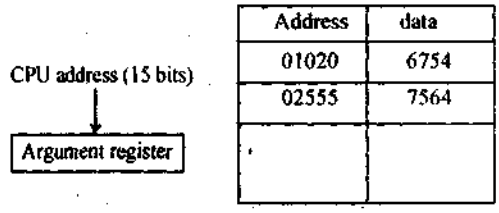
$$= 2^7 * 2^8 = 2^{15}$$

Hence 15 bits are required for addressing cache memory.

Q.11. Explain various cache mapping techniques. A computer system has a 4K word cache organized in block set associative manner with 4 blocks per set, 64 words per block. The main memory contained 65536 blocks. How many bits are there in each of TAG, SET and WORD fields? [UP Tech 2005-2006]

Ans. The transformation of data from main memory to cache memory is known as mapping process. There are three mapping procedures.

- Associative mapping
- Direct mapping
- Set associative mapping



Associative mapping: in associative memory both address and data are stored. The diagram shows 2 words presently stored in cache. All the numbers are represented in octal.

CPU places 15 bit address in the argument register and associative memory is searched for a matching address. If that address is found, then corresponding 12 bit data is read from it and sent to the CPU. If no match is found then main memory is accessed for the word.

Direct mapping: In direct mapping RAM is used for cache memory as described below.

CPU address is of 15 bits; hence size of main memory is 32K. This address is divided into 2 fields; index and tag.

2^n words in main memory and 2^k words in cache memory. Then k bits are there in index field and (n-k) bits are there in tag field.

For example 2^{15} words in main memory and 2^9 words in cache memory so size of index is 9 bits and tag is (15-9)=6 bits.

Addressing relationship between main and cache memory is shown with the help of figure given below.

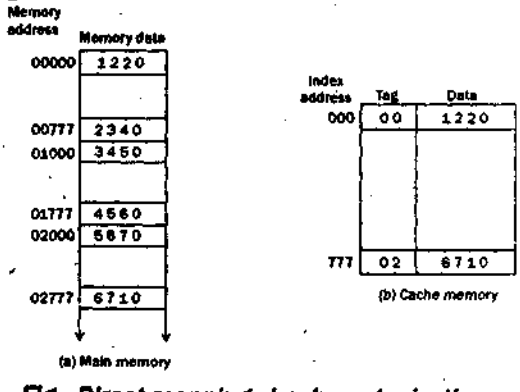


Fig. *Direct mapping chache organization.*

Each word in cache consists of tag field and data. When CPU generates a memory request, the index field is used for the address to access the cache. After that tag field is compared with the tag in the word read from cache. If it matches then it is a hit otherwise miss. If miss then required word is read from main memory and stored in cache, with a new tag replacing previous value.

Set associative mapping: It is an improvement over direct mapping. In this mapping 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. Total number of tag-data items in one word of cache is said to form a set.

Each tag requires 6 bits and each data word has 12 bits. If there are two tag fields present in cache then word length of cache will be 2 * (6+12) = 36 bits.

The hit ratio will improve as the set size increases because more words with the same index but different tags can be present in cache.

Index	Tag	Data	Tag	Data
000	01	3450	02	5670
777	02	6710	00	2340

Fig. *Two-way set-associative mapping*

Numerical

Main memory size = 65536 = 2^{16}

m=16

Cache memory size = 64 blocks = 2^6

n=6

Set size = 4 blocks = 2^2

s=2

TAG = m - n + s = 16 - 6 + 2 = 12 bits

Q.12. Drive the logic of one cell and of an entire word for an associative memory that has an output indicator when the unmasked argument is greater than (but not equal to) to word in the associative memory.

[UP Tech 2006-2007]

Ans. Let $x_i = A_i F_{ij} + A_i' F_{ij}'$

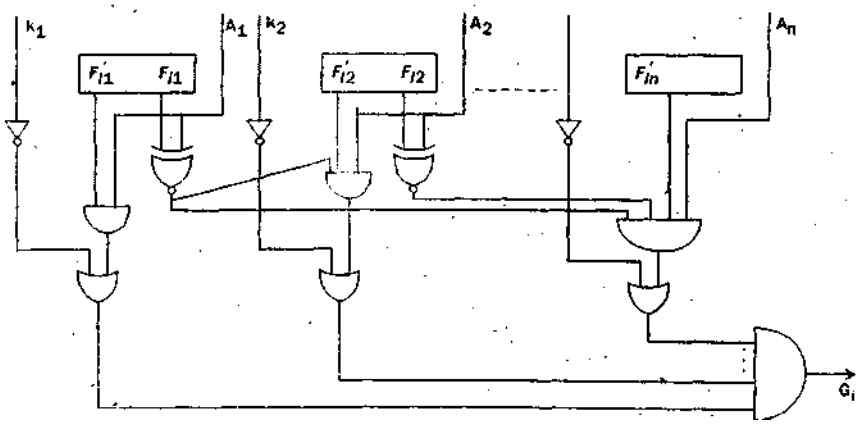
Output indicator $G_i = 1$ if

$A_1 F_{i1}' = 1$ and $k_1 = 1$ (first bit in $A = 1$ while $F_{ij} = 0$)

or if $x_1 A_2 F_{i2}' = 1$ and $k_2 = 1$

(first pair of bits are equal and second bit in $A = 1$ while $f_{i2} = 0$)

$$G_i = (A_1 F_{i1}' + k_1')(x_1 A_2 F_{i2}' + k_2')(x_1 x_2 A_3 F_{i3}' + k_3') \ldots (x_1 x_2 \ldots x_{n-1} A_n F_{in}' + k_n')$$



Q13. A two-way set associative cache memory used blocks of four words. The cache can accommodate a total of 2048 words the main memory. The main memory size is 128 K * 32.

(i) Formulate all pertinent information require constructing the cache memory.

(ii) What is the size of the cache memory? [UP Tech 2006-2007]

Ans. Size of main memory = 128K = 2^{17}

Hence address is of 17 bits.

Cache can accommodate 2048 words.

Set size = 2

(i) Hence cache can accommodate 2048/2 = 1024 words with set size of 2.

1024 words = 2^{10}

Index address needs 10 bits.

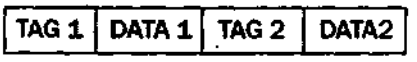
TAG = 7 bits

Index = 10 bits which is divided into Block and word

Block = 8 bits

Word = 2 bits

(ii) One word of cache memory will be like shown below



TAG 1 = TAG 2 = 7 bits

DATA 1 = DATA 2 = 32 bits

There are 1024 words in cache memory.

In one word there are 2(7+32) bits.

Hence size of cache memory = 1024 * 78 bits

Q14. Explain the direct mapping technique. Consider a digital computer has a memory unit of 64 K*16 and a cache memory of 1K words. The cache uses direct mapping with a block size of four words:

(i) How many bits are there in the TAG, index, block and word fields of the address format?

(ii) How many blocks the cache can accommodate? [UP Tech 2007-2008]

Ans. Direct mapping: In direct mapping RAM is used for cache memory as described below.

CPU address is of 15 bits; hence size of main memory is 32K. This address is divided into 2 fields; index and tag.

2^n words in main memory and 2^k words in cache memory. Then k bits are there in index field and (n-k) bits are there in tag field.

For example 2^{15} words in main memory and 2^9 words in cache memory so size of index is 9 bits and tag is (15-9)=6 bits.

Addressing relationship between main and cache memory is shown with the help of figure given below.

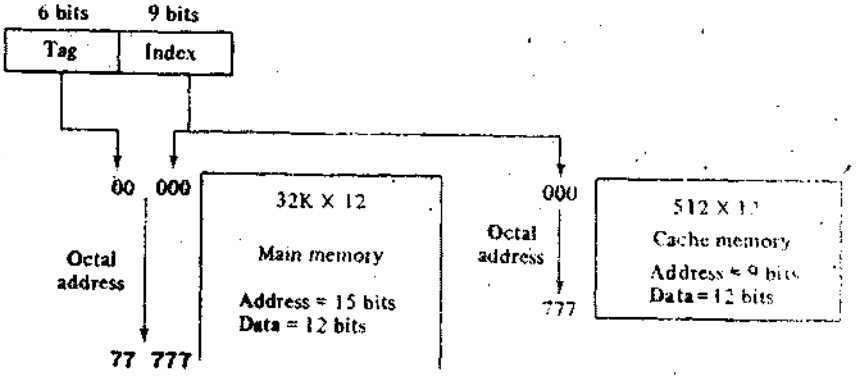


Fig. *Addressing relationships between main and cache memories.*

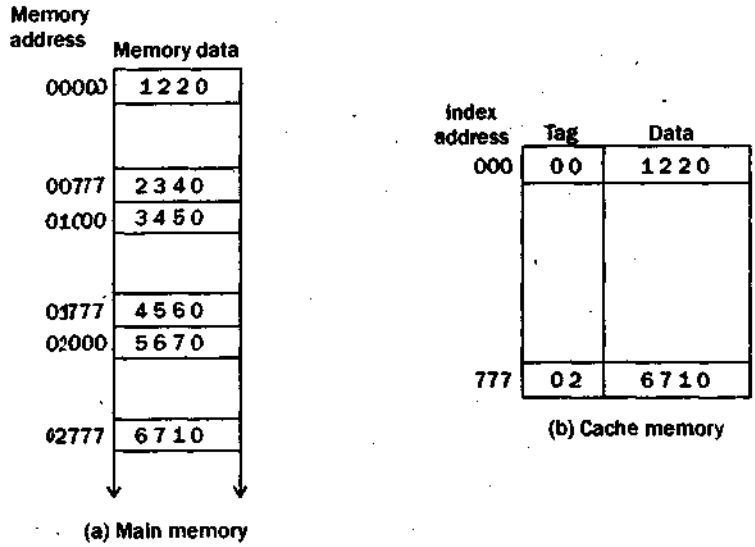
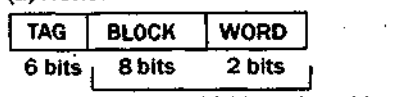


Fig. *Direct mapping cache organization.*

Each word in cache consists of tag field and data. When CPU generates a memory request, the index field is used for the address to access the cache. After that tag field is compared with the tag in the word read from cache. If it matches then it is a hit otherwise miss. If miss then required word is read from main memory and stored in cache, with a new tag replacing previous value.

Size of main memory = 64 k × 16
Address lines = 16
Word length = 16 so, data lines = 16
Size of cache memory = 1 K × 16
address = 10 bits

(*a*) Hence



TAG = 6 bits
BLOCK = 8 bits
WORD = 2 bits
INDEX = 10 bits

(*b*) Size of cache memory = 1 k words = 1024 × 16
Size of one block = 4 words = 4 × 16

Hence cache can accommodate = $\frac{1024 \times 16}{4 \times 16}$ = 256 blocks

Cache memory can accommodate 256 blocks of 4 words each.

Q.15. Define the terms address space and memory space. An address space is specified by 24 bits and the corresponding memory space by 16 bits.

(a) How many words are there in the address space?
(b) How many words are there in the memory space?
(c) If a page consists of 2K words, how many pages and blocks are there in the system? **[UP Tech 2008-2009]**

Ans. Address space: An address used by programmer is called Virtual address. The set of virtual addresses is called address space.

Memory space: An address in main memory is called physical address and set of such addresses is called memory space.

The address space is allowed to be larger than the memory space in computers with the help of virtual memory.

(a) Address space = 2 bits

Number of words present in address space

$= 2^{24} = 16$ M words

(b) Memory space = 16 bits

Number of words in memory space

$= 2^{16} = 64$ k words

(c) One page = 2 k words

$$\text{Totalnumber of pages} = \frac{16M}{2k} = 8 \text{ k pages}$$

Total 8 k pages are present

$$\text{Total number of blocks} = \frac{16k}{2k} = 32 \text{ blocks}$$

Total 32 blocks are present in the system.

Q16. Write short notes on the following:

(a) Auxiliary memory

(b) Memory hierarchy

(c) Cache memory

(d) Virtual memory

[UP Tech 2008-2009]

Ans. Auxiliary memory: Memories which are used for backup storage are called auxiliary memory. They are used for storing system programs, large data files and other backup information. During execution, programs and data required by CPU reside in main memory and other information is stored in auxiliary memory and transferred to main memory only when required. Most commonly auxiliary memory devices are magnetic disks and magnetic tapes. Some other examples of auxiliary memory devices are magnetic drums, magnetic bubble memory and optical disks. The important properties of any memory device are its access time, transfer rate capacity and its cost. The average time required to reach a storage location in memory and obtain its contents is called access time. Number of words transferred per unit time is transfer rate. Time required to transfer data to and from any device is transfer time.

Magnetic disks: It consists of a circular plate made up of metal or plastic which is coated with magnetized material. Many disks are placed together on one spindle with read/write heads on each surface because normally both sides of disk are used. Bits are stored in the magnetized surface in spots along concentric circles called tracks. Tracks are divided into sections which are called sectors. Disks which are permanently attached in computer system are called hard disks.

Magnetic Tape: It is a plastic strip with a coating of magnetic medium. Bits are recorded as magnetic spots on the tape along several tracks. Read/write tracks are mounted one in each track so that data can be recorded and read as a sequence of characters. In tapes information (data) is recorded in blocks. A tape unit is addresses by specifying the record number and the number of characters in the record. Records can be of fixed length or they can be of variable length.

Memory hierarchy: The memory hierarchy system consists of all storage units placed in a computer system from slow auxiliary memory to fast cache memory.

At the top CPU registers are present which are smallest in capacity but fastest in access. Next is cache memory which is normally of the order of 32KB to few MBs. Cache memory is very small, expensive and it has very high access speed. Next is main memory whose size normally in Giga bytes. Next are magnetic disk and finally magnetic tape and optical tape. Magnetic disks and tapes are also called auxiliary memory. They are used for storing large data files and other backup information. The 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 required.

There are three parameters in memory to compare between different types of memory which are access time, storage capacity and cost. The average time required to reach a storage location in memory and obtain its contents is called access time.

As the storage capacity of the memory increases, the cost per bit for storing binary information decreases. But as storage capacity increases, access time also increases.

Cache memory: If active portion of program and data are placed in a fast small memory then average access time can be reduced thus reducing the total execution time of the program. This small memory is called cache memory. It is placed in between CPU and main memory. It is considered as fastest memory. Its size is just fraction of main memory.

Basic operation: when CPU needs to access memory, the cache is examined. It the required word is found in cache, it is read from there otherwise main memory is accessed. Then a block of words including the main request is transferred from main memory to cache. Now cache contains the requested word as well as the future references also.

Performance of cache memory is measured in terms of a quantity called Hit ratio. When CPU refers to memory and finds the word in cache, it is called a *hit*. When it does not find the word then it is a *miss*.

Ratio of number of hits divided by the total CPU references to memory is called Hit ratio.

Virtual memory: This concept allows its users to construct programs as if large amount of memory is available. It gives illusion to its programmers as if they have a large memory for their use, where there is relatively small main memory. It is a combination of hardware and software techniques. A virtual memory system provides a mechanism for translating program generated addresses into correct main memory locations. This is dynamically, while programs are being executed in the CPU. The translation or mapping is handled automatically by the hardware by means of a mapping table.

An address used by programmer is called Virtual address. The set of virtual addresses is called address space. An address in main memory is called physical address and set of such addresses is called memory space. The address space is allowed to be larger than the memory space in computers with the help of virtual memory.

Q17. A virtual memory has page size of 1K words. There are eight pages and four blocks. The associative memory page table contains the following entries.

Page	Block
0	3
1	1
4	2
6	0

Make a list of all virtual addresses (in decimal) that will cause a page fault if used by CPU.

[UP Tech 2006-2007]

Ans. Page size = 1K words

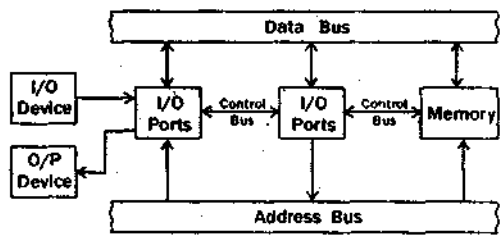
Total 8 pages are there out of which only 4 pages (0, 1, 4, and 6) are present in main memory.

Pages which are not present are 2, 3, 5, and 7.

Page no.	Address	Address that will cause page fault
2	2K	2048-3071
3	3K	3072-4095
5	5K	5120-6143
7	7K	7168-8191

Q.18. Draw and explain the block diagram of simple μp based system.

Ans.



Microprocessor: It is a semiconductor device consisting of electronic logic circuits manufactured by using either LSI or VLSI technique. It is capable of performing computing functions and making decisions to change the sequence of program executions.

The three sequents of μp are:

Arithmetic/Logic Unit: Computing functions are performed on data in this area. It performs operations such as addition, subtraction, logical operations etc.

Register unit: These are used primarily to store data temporarily during the execution of a program.

Control unit: In provides the necessary timing and control signals to all the operations.

Input Device: The input section transfers data and instructions in binary from the outside world to the μp. It includes devices like keyboards, teletypes etc.

Output Device: It transfers data from the μp to the output device like LED, CRT, pointers, mag. tape etc.

Memory: it stores binary information such as instruction and data and provides that info. to the μp. To execute prograss, the μp reads instructions and data from memory performs the operatioins in ALU section. These are RAM and ROM.

RAM used to store user programs and data. The info can be read and altered easily.

ROM: It stores programs that do not need alternations.

System Bus: It is a common path between μp and peripherals. It may be address bus, control bus or data bus.

Address Bus: It is undirectional from μp to peripheral devices. The μp uses the address bus to perform the first function i.e. to identify a pheripheral or a memory locatien.

Data Bus: These lines are bidirectional data flow in boths directions between μp and devices. The μp uses the data bus to perform the function of transferring the data.

Control Bus: It comprises of various single lines that carry synchronization signals. These lines are used to provide timing signals.

Q.19. Draw the functional stock diagram of 8085 μp. Explain briefly the different blocks.

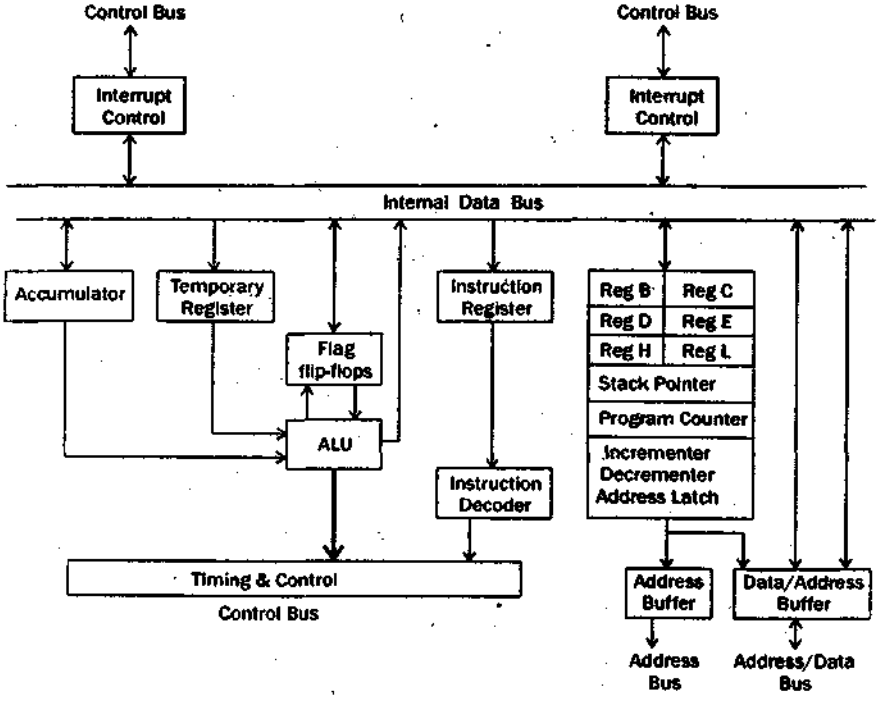
Ans. 1. Data and Address Bus: 8085 is an 8 bit μp. It has 8 bit bidirectional data bus used for data input as well as data output in the form of 8 bit words.

The address bus is 16 bit wide i.e. memory address space is 64 k bytes. The address bus is in two sections lower order eight bits of address and higher order eight bits of addresses.

2. Arithmetic logic units (ALU): It has 8 bit ALU. The arithmetic and logical operatioins are perofmred in ALU.

3. Accumulator Register: For arithmetic and logical operations, one of the operands is contained in this register. The result of these operations is also stored here.

4. Temporary Register: These are involved only in internal operations and are not accessible to the user.



5. Instruction Register: It gets the operation code of the instruction in op-code fetch m/c cycle and passes it on to the decoder.

6. Instruction decoder: The function is to interpret and enable the control section to produce proper signals to carry out the tasks required to be performed in the instruction.

7. Flags flip-flops: These serve to indicate conditions that arise during arithmetic and logical operations, (*i*) Zero (*z*), (*ii*) Sign (*s*), (*iii*) in parity (*P*), (*iv*) carry (CY), (*v*) Auxilliary carry (AC).

8. General purpose registers: There are 6 such registers B, C, D, E, M & L. These registers can be used as single in pairs. These registers are used for temporary storage of operands or intermediate data in calculation.

9. Stock pointer (SP): 16 bit register, used to store the address of the stack top.

10. Program counter (PC): It is a 16 bit special purpose register that that stores the address of the next instruction to be executed of a program so it acts as a pointer to the next instruction.

11. Interrept system: This is used to stop the µp to postpone the routine job in order to attend© to some urgent task.

12. Timing and control: This unit synchronizes all the µp operations with the clock and generates the control signals necessary for communication.

13. Power Supply & Clock Frequency: V_{CC} = + 5 V, V_{SS} = Ground. The clock freq. is interually divided by two system, one is operated at 3 MHz, whereas crestal has frequency of 6 MHz.